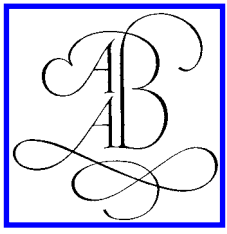
NEURAL NETWORKS FOR HYDROLOGICAL MODELLING

Neural Networks for Hydrological Modelling

ROBERT J. ABRAHART
University of Nottingham

PAULINE E. KNEALE
University of Leeds

LINDA M. SEE
University of Leeds



A.A. BALKEMA PUBLISHERS Leiden / London / New York / Philadelphia / Singapore

Library of Congress Cataloging-in-Publication Data
A Catalogue record for this book is available from the Library of Congress

Cover design: Miranda Bourgonjen
Typesetting: Charon Tec Pvt. Ltd, Chennai, India
Printed in the Netherlands by Wilco

Published by: A.A. Balkema Publishers (Leiden, The Netherlands), a member of Taylor & Francis Group plc.
http://balkema.tandf.co.uk and www.tandf.co.uk

ISBN 90 5809 619 x

Contents

List of Contributors VII

Preface IX

Chapter 1 Why Use Neural Networks? 1
Pauline E. Kneale, Linda M. See and Robert J. Abrahart

Chapter 2 Neural Network Modelling: Basic Tools and Broader Issues 15
Robert J. Abrahart

Chapter 3 Single Network Modelling Solutions 39
Christian W. Dawson and Robert L. Wilby

Chapter 4 Hybrid Neural Network Modelling Solutions 61
Asaad Y. Shamseldin

Chapter 5 The Application of Time Delay Neural Networks to River Level Forecasting 81
Linda M. See and Pauline E. Kneale

Chapter 6 The Application of Cascade Correlation Neural Networks to River Flow Forecasting 97
Claire E. Imrie

Chapter 7 The Use of Partial Recurrent Neural Networks for Autoregressive Modelling of Dynamic Hydrological Systems 115
Henk F. P. van den Boogaard

Chapter 8 RLF/1: Flood Forecasting via the Internet 139
Simon A. Corne and Stan Openshaw

Chapter 9 Rainfall-Runoff Modelling 157
Anthony W. Minns and Michael J. Hall

Chapter 10 A Neural Network Approach to Rainfall Forecasting in Urban Environments 177
James E. Ball and Kin Choi Luk

Chapter 11 Water Quality and Ecological Management in Freshwaters 197
Pauline E. Kneale

Chapter 12 Neural Network Modelling of Sediment Supply and Transfer 225
Susan M. White

Chapter 13 Nowcasting Products from Meteorological Satellite Imagery 243
George S. Pankiewicz

Chapter 14 Mapping Land Cover from Remotely Sensed Imagery for Input to Hydrological Models 267
Giles M. Foody

Chapter 15 Towards a Hydrological Research Agenda 289
Robert J. Abrahart, Pauline E. Kneale and Linda M. See

List of Contributors

Robert J. Abrahart School of Geography, University of Nottingham, Nottingham NG7 2RD, UK

James E. Ball Water Research Laboratory, School of Civil and Environmental Engineering, The University of New South Wales, Sydney, New South Wales, Australia

Christian W. Dawson Department of Computer Science, Loughborough University, Leicestershire, LE11 3TU, UK

Simon A. Corne correspondence c/o Prof Pauline Kneale, School of Geography, University of Leeds, Leeds, LS2 9JT, UK

Michael J. Hall UNESCO-IHE Institute for Water Education, Delft, The Netherlands

Giles M. Foody School of Geography, University of Southampton, Highfield, Southampton, SO17 1BJ, UK

Claire E. Imrie Department of Environmental Science and Technology, Imperial College of Science, Technology and Medicine, Prince Consort Road, London SW7 2AZ, UK

Pauline E. Kneale School of Geography, University of Leeds, Leeds, LS2 9JT, UK

Kin Choi Luk correspondence c/o Prof James Ball, Water Research Laboratory, School of Civil and Environmental Engineering, The University of New South Wales, Sydney, New South Wales, Australia

Anthony W. Minns WL|Delft Hydraulics, Delft, The Netherlands

Stan Openshaw correspondence c/o Prof Pauline Kneale, School of Geography, University of Leeds, Leeds, LS2 9JT, UK

George S. Pankiewicz Satellite Applications Section, The Meteorological Office, London Road, Bracknell, RG12 2SZ, UK

Linda M. See School of Geography, University of Leeds, Leeds, LS2 9JT, UK

Asaad Y. Shamseldin School of Engineering, Department of Civil Engineering, University of Birmingham, Edgbaston, Birmingham, B15 2TT

Henk van den Boogaard WL|Delft Hydraulics, Delft, The Netherlands

Susan M. White Institute of Water and Environment, Cranfield University, Silsoe, Bedfordshire, MK45 5DT, UK

Robert L Wilby Environment Agency, Trentside Offices, Nottingham, NG2 5FA, UK

Preface

This book is intended as an introduction to those who are new to neural network hydrological modelling and as a useful update for those who have been experimenting with different tools and techniques in this area. The scope for applying neural network modelling to hydrological forecasting and prediction is considerable and it is only really in the last five to ten years that it has been tried and tested. The various chapters show that while rainfall runoff forecasting is the main area of research, neural networks are also used in ecological, fisheries, water quality, sediment, groundwater and many other water related applications. The scope is considerable because a neural network works in an equation free environment so that economic, social, hydrological and chemical data can be integrated on an equal basis. Neural networks are often denigrated as black box solutions, but they are sophisticated black boxes, which can produce very useful results. We hope that this book will encourage further users to get involved and experiment.

Each of the chapters has been the subject of an independent review and we are grateful for the many comments and time involved. We are also grateful to the authors for responding to our comments and the reviewers' input and for making the changes requested.

Robert Abrahart
Nottingham

Pauline Kneale
Leeds

Linda See
Leeds

1

Why Use Neural Networks?

PAULINE E. KNEALE, LINDA M. SEE
School of Geography, University of Leeds, UK

ROBERT J. ABRAHART
School of Geography, University of Nottingham, UK

ABSTRACT: Neural networks are one computational methodology for hydrological forecasting. Although widely used in other research and application fields they are employed less by hydrologists than might be expected given the data driven nature of the applied problems to be solved. Neural networks provide a modelling route that can be helpful when there is enough data to link x to y and especially where results are needed in real time. This chapter introduces neural network issues generally, setting them in a wider modelling context and provides a framework link to later chapters which handle neural network topics in detail.

1 INTRODUCTION

Neural networks (NN) are an alternative and complementary set of techniques to traditional models. NN can be thought of as computational pattern searching and matching procedures that permit forecasting without an intimate knowledge of the physical or chemical processes. For the hydrologist this technique has considerable appeal, provided the absence of a detailed process explanation can be borne.

NN rely on the provision of adequate data sets, and where these are available, NN may be programmed to search for patterns within the data. On the basis of this pattern-matching, forecasts are made on independent data sets first for model validation and then for operational purposes. NN are one approach within the broader hydroinformatics framework which emerged in the 1990s as a route to managing information overload in an effective way (Govindaraju & Rao, 2000). Price (2000) recognises four strands to hydroinformatics, the mathematical and physical sciences understanding, the handling

of data and the human cultural element. One of the significant strengths of the NN approach is that it can handle all data types.

The challenge of managing water in its many dimensions and applications calls for techniques which can link a myriad of components, from the complexity of hydraulics and water quality, to financial planning and social agendas. This is a move towards a holistic or integrated approach to modelling. The techniques available to the hydrologist are many and varied, each with their own advantages and drawbacks. The vision of 1970s modellers (Freeze & Harlan, 1969) that forecasting problems would be cracked when computers became powerful enough to handle very complex equations and infinitely large data sets has become a receding, but by no means disappearing, target. Natural environmental variability, the uniqueness of catchments, system chaos and the complexities of scale integration, together with the expense of data acquisition, make the forecasting task challenging. Flood modelling at the basin scale with fine mesh models requires prodigious amounts of computer time, but Beven and Feyen (2002) consider that these goals are coming nearer as visualisation and virtual game technologies advance, so the ambitions for catchment-wide 4D modelling are getting closer. NN do not in any way aim to replace such models but they can provide a very fast forecasting system that is operationally available in very short time frames. NN do not compete with distributed models but rather offer alternative and complementary ways of tackling forecasting problems.

This text is aimed both at those using NN in research for the first time and at those wanting to review recent examples of NN hydrological applications. It is not intended as a manual but should be used as a supportive guide for anyone wanting to experiment with this type of modelling. This chapter introduces some of the basic ideas and background behind the NN approach, particularly for those who are new to this methodology. If you are already familiar with NN techniques then skipping to later chapters may be helpful. The sections that follow provide a link to the more detailed materials in the main chapters and to broader applications.

2 THE BASIC IDEAS OF THE NN APPROACH

To understand the basic ideas behind the NN approach, let's look at a simple hydrological example. Imagine that you could access data banks of hydrological information. Suppose that the databases contain stage data recorded every ten minutes at 4 points on a river (C–F), precipitation data from a gauge (A) collated every fifteen minutes from a radar system (B) and weekly soil moisture data (G) (Fig. 1.1). Your first task is to decide which station you want to model. In a conventional approach you might choose to model stage at F using all the data sets including past records from site F as inputs. Alternatively you might (eccentrically) decide to forecast precipitation at B using all the data (Fig. 1.2). The point is that the NN has no knowledge of the spatial relationship between the sites as seen on a map nor any idea about what it is being modelled. The NN only seeks the relationships between the input and output data and then creates its own equations to match the patterns in an iterative manner.

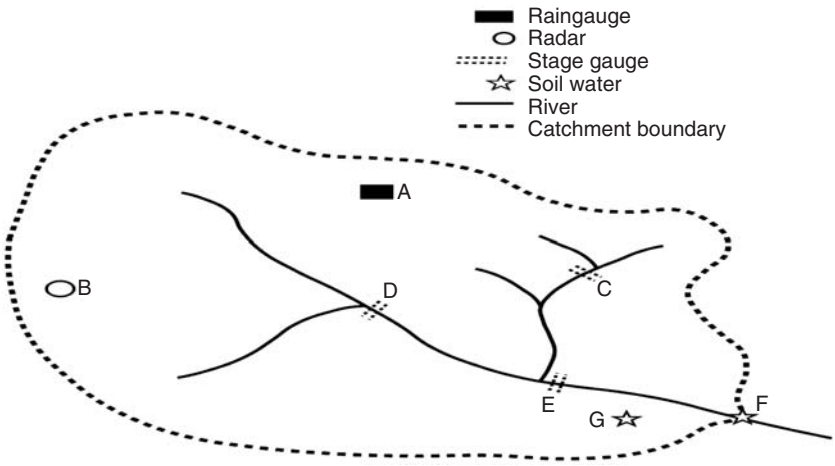


Fig. 1.1. Catchment X.

Input Data	Potential Forecast	Potential Forecast	Potential Forecast
Raingauge A (Continuous)	Stage at F	Stage at C	Radar B
Radar B (15 minute)			
Stage C (15 minute)			
Stage D (15 minute)			
Stage E (15 minute)			
Stage F (15 minute)			
Soil water deficit G (weekly)			

Fig. 1.2. Potential models.

Continuing with this example, a forecaster might choose to start modelling with all the data, and look to eliminate those data sets that are not contributing significantly to the output to find the most parsimonious approach, thereby saving data collection and data processing time. It may be that operationally the best forecasting model for stage at station E is the stage at station E in previous time steps. In forecasting terms this may be the cheapest and most accurate model, but a user might choose a less optimal model that includes an upstream site in real-time forecasting in case there are data transfer problems during a real-time event. The 'best' computational solution may not be the ideal practical solution. This is a forecasting approach where there are many decisions to be made by the user.

NN models are variously described as mimicking the parallel-information processes of the brain. However, a typical human brain is thought to contain 10^{11} neurons, each receiving input from an additional 5000 to 15000 neurons. The average worm has approximately 10^3 connections. A NN is likely to have connection numbers in the 10–1000 range so a NN would be considered to be

of sub-worm complexity (Openshaw, 1997). Comparisons of this kind are illuminating at one level but do not inspire immediate confidence in the technical merits of NN as a sophisticated analytical technique. It is important to see why they are so described and evaluate this description (see Chapters 2, 3 and 4).

The brain analogy is helpful for new users. NN are a mathematical representation of a process that operates like nerve cells. Each network is made up of nodes and links, much like the nerve cells and messages in a nervous system. The user defines the architecture of the network and following trial and error runs, this mathematical representation of the NN becomes the model framework. For example, trials may show that the radar data at B (Fig. 1.2) may not be contributing useful additional information so that node would then be removed in further model trials.

Forecasting should follow in three clearly separate stages of NN development, stages that are kept separate to make comparisons as accurate as possible. In 'training mode' the output pattern at say station F (Fig. 1.1) is linked to as many of the input nodes (A–G) as desired and the patterns are defined. In the training phase, part of the total data set is used. Conjusingly, NN scientists may also refer to a 'validation' dataset used at this stage to ensure the model is not overtrained. The data may be temporally contiguous or it may be selected as being representative across the whole period. This can be important if it is thought that there may be systematic change on a catchment across the whole period, arising for example from land use change. This is followed by a 'testing phase' when the model is tested using data sets that were not used in training. If the forecasts are satisfactory then the model may be used in an 'operational' or 'real time mode' to generate live forecasts. These live forecasts are evaluated against real events. Measures of accuracy of a model should ideally refer to forecast performance in the real-time mode or independent validation mode.

Once established the NN can be developed or updated as more data become available. In this sense NN are dynamic in that the operator can adjust and adapt them as change occurs, which makes them potentially very valuable in hydrological operational modes. In this simple hydrological example it would be logical for an operator to update the forecasting networks at the end of each wet season to take account of recent precipitation events and thus give the users additional confidence in the modelling. Because the processing speeds of NN are very high, in practice a model can be updated and redeveloped in real time to take account of new or changing circumstances if required (Abrahart, 2003).

NN may be regarded as data driven techniques but it is argued here that their flexibility in data handling and the ability to solve problems where it is effectively impossible to get primary data, as in groundwater modelling solutions (Ouenes, 2000; Zio, 1997) and with the added complexity of groundwater chemistry (Gumrah *et al.*, 2000), or where processes are highly non-linear and spatially and temporally variant (Islam & Kothari, 2000), makes these techniques well worth exploring. If a distributed modelling solution is not available but the data are, then this may be a useful approach. Certainly many NN applications have been prompted by unsatisfactory results with regression and time series techniques.

3 SOME ANTECEDENTS

The pattern for classifying hydrological modelling approaches was articulated by Dooge (1977). His three phase black box empirical, lumped and physically-based distributed model distinction is very widely recognised. This lead to an acceptance of an apparent hierarchy in quality of approach with the 'simple black box' considered to be less acceptable than the more mathematically rigorous, theoretically based distributed approach. While this distinction is academically valid, it is not always helpful in practical terms. The advice to use the simplest tool that will do the job is appropriate in practical and operational modelling. If the data are available and the problem is linear then using linear regression is fine. The unit hydrograph and rational formulas survive because they are practical tools that supply useful answers.

While NN are a relatively recent technique for hydrologists they have an established antecedence which Govindaraju and Rao (2000) acknowledge as starting in the 1940s. NN concepts arrived with McCulloch and Pitts' (1943) work but their practical use followed Rumelhart *et al.*'s (1986) development of the back propagation neural network (BPNN) algorithm which lead to a plethora of applications in many subjects. Various text books in the 1990s generated some interest (Masters, 1993; Cruz, 1991) and the first hydrological applications were probably Daniell (1991), French *et al.* (1992) and Hall and Minns (1993). So for hydrologists this is a young technique with a short pedigree. But there has been a rapid uptake and a positive blossoming in conferences and publications. Good generic texts on the subject include Bishop (1995), Haykin (1998) and Picton (2000) but there are many other sources available.

Various authors describe NN models as black box and dismiss them as empirical, and therefore by definition, as inferior. Certainly the calculations are 'set-up' by the modeller but the nature of the relationship between variables is found by the computer (see Chapter 2). So in this sense NN are input-output models. They are therefore vulnerable to the problems of inadequate data and a less than thoughtful forecaster. However, they have the strength when compared for example with ARMA and regression approaches that non-linearity in relationships will be captured (Hsu *et al*., 1995) and the black box can be looked into in detail if the forecaster wishes (Abrahart *et al*., 2001; Wilby *et al.*, 2003). The early hydrological literature is dominated by rainfall-runoff forecasting applications, probably because these represent a conceptually straight-forward starting point. There are some lengthy records for both variables for training and validation, and the solutions are evidently non-linear; this theme is well reviewed in this text in Abrahart (Chapter 2), by Minns and Hall (Chapter 9), and in the GIS application of rainfall modelling discussed by Ball and Luk (Chapter 10).

Alternative introductions to NN modelling in hydrological contexts include Maier and Dandy (2001b) who provide a sound introductory overview in the context of cyanobacterium and salinity modelling in River Murray, and Dawson and Wilby (2001). In a Special Issue of *Computers and Operations Research,* Gupta and Smith (2000) cover a significant range of non-hydrological examples,

and the business applications considered are of interest to those considering modelling economic and management aspects of water supply and water management.

The hydrological applications from the last seven years fall into a series of broad categories and styles of modelling. There are three main types of NN: backpropagation (BPNN), radial basis function network (RBFN) and the self-organising feature map (SOFM). Abrahart addresses each style in detail in Chapter 2 and as later chapters will indicate, backpropagation neural networks (BPNN) dominate for forecasts at specified points such as river stage, whereas SOFM mapping algorithms are employed to predict spatial patterns.

Running models with multiple inputs implies the availability of appropriate data sets, a problem for any field-based hydrological work. However where data are captured in remote sensing operations and GIS programmes the NN approach can be very powerful as Foody's Chapter 14 indicates. Gautam *et al*. (2000) have the advantage of a well instrumented catchment at Tono, Japan, providing meteorological, runoff and soil moisture content data for their stream flow forecasts. This is a luxury not available in most areas; however, the results are satisfactory indicating that the NN technique may be of benefit for small catchment forecasts and perhaps in agricultural applications. To forecast soil texture from remotely sensed maps Chang and Islam (2000) use brightness temperature and remotely sensed soil moisture. The soils are classified into six classes. Forecasting the permeability of oil reservoirs, Bruce and Wong (2002) use an evolution NN algorithm to solve a forecasting problem bedevilled by solutions that can be trapped in local minima using backpropagation.

NN are not necessarily run in isolation. In linking NN within their models Maskey *et al*. (2000) for example show how NN can be used with process models to calculate travel times of groundwater pollution plumes in response to well injections and pumping in an experiment to optimise a groundwater clean up programme. The flexibility to use a NN within a broader modelling framework is an attractive use of the technology.

While hydroinfomatics primarily concentrates on aquatic forecasting, for some authors NN technologies assist in the objective inclusion of social and economic dimensions. Jonoski (2002) looks towards a sociotechnological role for the hydrological forecaster where these additional dimensions are an integral part of the modelling process in what they define as Network Distributed Decision Support Systems.

The reported use of NN models is broad and considerable in statistical and engineering applications (Ma *et al*., 2001; Venkateswaran *et al*., 2002). Their operational rather than research use is also extensive in a wide range of industries: in mining to identify rocks that can be obstructive (Cabello *et al*., 2002), converting speech to text (Wang *et al*., 2002), monitoring wear on machine tools (Scheffer & Heynes, 2001), automating wastewater treatment and chemical monitoring (Zyngier *et al*., 2001), forecasting sea ice conditions in Canadian waters (El-Rabbany *et al*., 2002), coffee bean blending (Tominaga *et al*., 2002), flavour of blackcurrants (Boccorh & Paterson, 2002) and identifying corrosion rates on aircraft parts (Pidaparti *et al*., 2002).

Govindaraju and Rao (2000) suggest that the adoption of NN techniques by hydrologists has been constrained by the relative newness of the technique, and its position as an empirical methodology in a subject which struggled to get rid of its soft empirical subject image and emerge as an accepted physics-based discipline. Maier and Dandy (2000) reiterate the essential need for thoughtful applications: 'In many applications, the model building process is described poorly, making it difficult to assess the optimality of the results obtained'. Flood and Kartam (1994) also add a relevant observation: 'There is a tendency among users to throw a problem blindly at a NN in the hope that it will formulate an acceptable solution'. Maier and Dandy's (2000) paper would be a great place for many modellers to start. The authors review the issues for modelling with a wide range of practical examples.

Much of this text exemplifies the need for a systematic approach to thinking through the methodological approaches and constraints. Then to apply these approaches to relevant hydrological issues. We would argue that it represents an opportunity to model with greater freedom and speed some of the 'difficult' multifaceted problems in hydrology.

However, it is important to point out that NN are not magic boxes. There is an extensive mathematical background and theory that has underpinned their development and for those mathematically inclined this is a rich area of investigation. The NN technique cannot be criticised as theoretically unsupported and therefore unsound. Users can decide to try the NN approach without exploring the mathematics in detail and to take advantage of the plethora of freeware or shareware off-the-shelf packages. This is really no different to users taking some of the more advanced codes in SPSS for partial canonical correlation. *Caveat emptor* always applies, and as the authors of Chapters 2–5 which look at the basics of different types of modelling approaches emphasise, it is vital to understand the data and programming decisions involved. But these are explained in practical terms and are on a par with understanding that 3 samples are not enough for multiple regression with 6 variables and that ANOVA values require significance tests. In other words try it for yourself.

4 WHERE DO I FIND THEM? NN PLATFORMS

Individual chapters in the book direct you to specific software sources, while this section provides a brief overview of the sites available. There are a very substantial number of companies and web sites offering NN software and a range of product support packages. The most useful starting point might be ftp://ftp.sas.com/pub/neural/FAQ.html; a users site that is updated monthly. As it says: 'its purpose is to provide basic information for individuals who are new to the field of neural networks'. There are software programs to download via ftp sites, for use on multiple platforms. Table 1.1 provides a short starting list of websites that you might check out, while later chapters point users to particular software packages.

You can also find NN embedded within data mining software such as Clementine or IBM's Intelligent Data Miner. Data mining is a popular term in

Table 1.1. NN software suppliers and web sites, a starting list.

Software name and company	Web sites
Free or Share ware	
Ainet – Freeware Neural Network	www.ainet-sp.si/
GENESIS and PGENESIS 2.2	http://www.bbb.caltech.edu/GENESIS
KarstenKutza –	http://www.geocities.com/CapeCanaveral/1624/
NEURALFUSION –	http://www.neuralfusion.com/
PDP Plus, MIT Press	http://www.cnbc.cmu.edu/PDP++/PDP++.html
SNNS, Stuttgarter Neural Network Simulator, University of Tuebingen, Germany	http://www-ra.informatik.uni-tuebingen.de/SNNS
Commercial packages	
Brain Maker, California Scientific Software Company	www.calsci.com
Cortex-Pro	www.reiss.demon.co.uk/webctx/detail.html
IBM Neural Network Utility, IBM Company	nninfo@vnet.ibm.com
NeuralWorks Professional II Plus, NeuralWare Inc.	http://www.neuralware.com/
Neuro Genetic Optimizer (NGO), Bio Comp Systems Inc.	www.bio-comp.com/pages/neuralnetworkoptimizer.htm
Neuro Shell Predictor, Ward Systems Group Inc.	www.wardsystems.com
NeuroSolutions v3.0, Neuro Dimension, Inc.	http://www.nd.com/
QNET v2000	www.qnetv2k.com
STATISTICA: Neural Networks version 4.0, Statsoft Inc.	http://www.statsoft.com/
Neural Connection, SPSS Inc.	http://www.spss.com/

the business world for all techniques that can be used to turn large amounts of data into useful information, of which NN are only one example. Clearly any package needs evaluation and for the novice the array of software available is confusing. The hydrological NN literature is not awash with citations of software used; some users will have written their own programmes but given the availability of packages this seems as unnecessary today as writing a program to calculate regression. A starter suggestion is the SNNS, Stuttgarter Neural Network Simulator which is well documented and user friendly.

Rather than re-inventing program codes for backpropagation it would seem to be more useful for hydrological forecasters to develop a suitable suite of quality testing procedures. Kneale and See (2003) testing Time Delay Neural Network (TDNN) forecasts use ten tests to compare hydrograph forecast accuracy. It is critical that the tests chosen include those normally used in hydrological model evaluation, such as the Nash and Sutcliffe (1970) index. This permits users to evaluate the forecasts in a consistent and objective manner and compare them to results obtained from traditional hydrological forecasting procedures.

5 SUMMARY

Essentially NN are one of many tools at the disposal of the hydrological researcher. The user defines the independent and dependent variables and has all the normal modelling problems of locating suitable data sets to develop, test and validate the models.

One major advantage the NN approach has over traditional input-output modelling is that it makes fewer demands on the data. Unlike multiple regression, where the constraints of normality in the data distributions are often simply ignored, NN do not make assumptions about the statistical properties of a data set. Data for different variables can be of all types and available on different time or spatial scales. This allows for a flexible approach to data collection and model development. In management models for example weather related, soil dynamic, crop development and agricultural management information can be used as inputs using parameters that are recorded on hourly, weekly, monthly, m^2, hectare and currency scales.

A second major advantage is that in searching for patterns and links in the data sets there are no assumptions of linearity. NN are non-linear pattern identification tools, which is why they are potentially so attractive for tackling the non-linear problems of hydrology.

The powerful potential of NN models to solve 'hard computational problems' including those where the underlying ecological relations are not understood was cited by Lek and Guegan (1999). There is a wealth of understanding of hydrological processes at a range of scales from laboratory to hillslope and catchment. But it is not always clear how to write the equations to link processes that are understood at the m^2 scale so that they scale up to the basin scale. NN search for the patterns in the data and therefore have the potential to create the equations that describe the processes operating on the catchment under study. As with all modelling an ill-specified NN will generate inadequate to useless forecasts. A good hydrological understanding of the relevant field processes is a pre-requisite of good modelling. That together with enough understanding of the NN to have the confidence to eliminate inessential variables and so define, through experimentation, the most parsimonious but efficient model. The relationship that a NN defines must be sought again in data for different catchments, the chosen model reflects the complex interactions within the specified data sets. However the final selection of parameters, model architectures and training times for any model will be helpful guidance for forecasters applying the NN approach in comparable catchments, speeding up the development of future models.

The potential speed of model development is a factor that most NN users find attractive. Forecasting algorithms are available from a range of web and shareware sources. Data acquisition is part of every modelling process but the forecaster then moves into model development and testing. Our experience of river stage modelling is that computational run times are a matter of minutes and validation and independent forecasting is effectively instantaneous (Kneale *et al.*, 2000). A forecaster should not find this element of the hydrologist's toolbox more

difficult to apply than partial Canonical Correspondence Analysis, a GIS system, an ARMA model or complex process-based software applications.

It may be that the role of NN is as part of a larger modelling framework, where the NN is one element in a data handling and management tool. Most of this text is concerned with the application of NN to solving specific hydrological problems with the NN as the primary technique, but this is just one potential role. The considerable scope for links to GIS models is made explicit in Foody's Chapter 14. There is a dominance of rainfall runoff applications which are explored more fully in various chapters. NN were developed to mirror biological activities, their non-linear flexibility makes them very attractive for forecasting complex multi-disciplinary hydrological problems like crop and fish stock management, pesticide leaching and runoff from hillslopes, and groundwater pollution and abstraction interactions (Freissinet *et al*., 1999; Tansel *et al*., 1999; Morshed & Powers, 2000; Tingsanchali & Gautam, 2000).

Where the NN fits in the mosaic of techniques for the hydrologist is still uncertain but we hope these chapters will encourage each reader to see its relevance in a range of applications and to try the techniques.

REFERENCES

Abebe, A.J., Solomatine, D.P. & Venneker, R.G.W. 2000. Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events. *Hydrological Sciences Journal* 45(3): 425–436.

Abrahart, R.J. 2003. Neural network rainfall-runoff forecasting based on continuous resampling. *Journal of Hydroinformatics* 5(1): 51–61.

Abrahart, R.J., See, L. & Kneale, P. 2001. Applying saliency analysis to neural network rainfall-runoff modelling. *Computers and Geosciences* 27: 921–928.

Berry, M.J. & Linoff, G. 1997. *Data Mining Techniques: For Marketing, Sales and Customer Support*. Chichester: Wiley.

Berry, M.J. & Linoff, G. 2000. *Mastering Data Mining: The Art and Science of Customer Relationship Management*. Chichester: Wiley.

Beven, K. & Feyen, J. 2002. The future of distributed modelling – Special Issue editorial. *Hydrological Processes* 16(2): 169–172.

Bishop, C.M. 1995. *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Boccorh, R.K. & Paterson, A. 2002. An artificial neural network model for predicting flavour intensity in blackcurrant concentrates. *Food Quality and Preference* 13(2): 117–128.

Bruce, A.G. & Wong, P.M. 2002. Permeability prediction from well logs using an evolutionary neural network. *Petroleum Science and Technology* 20(3–4): 317–331.

Cabello, E.E., Sanchez, M.A. & Delgado, J. 2002. A new approach to identify big rocks with applications to the mining industry. *Real-Time Imaging* 8(1): 1–9.

Chang, D-H. & Islam, S. 2000. Estimation of soil physical properties using remote sensing and artificial neural network. *Remote Sensing of Environment* 74(3): 534–544.

Corne, S. Kneale, P.E., Openshaw S. & See L. 1998. The use and evaluation of artificial networks in flood forecasting. *Proceedings 33rd MAFF Annual Conference*.

Cruz, C. 1991. *Understanding Neural Networks: A Primer.* Cutter Information Corporation, Arlington, MA: 74(3).

Daniell, T.M. 1991. Neural networks – applications in hydrology and water resources engineering. *Proceedings of the International Hydrology and Water Resources Symposium.* National Conference Publication 91/22 Institute of Engineering, Barton, Australia, 3: 797–802.

Dawson, C.W. & Wilby, R.L. 2001. Hydrological modelling using artificial neural networks. *Progress in Physical Geography* 25(1): 80–108.

Dooge, J.C.I. 1977. Problems and methods of rainfall-runoff modeling. In T.A. Cirani, U. Maione & J.R. Wallis (Eds.) *Mathematical Models for Surface Water Hydrology* (pp. 71–108). Chichester: John Wiley.

El-Rabbany, A., Auda, G. & Abdelazim, S. 2002. Predicting sea ice conditions using neural networks. *Journal of Navigation* 55(1): 137–143.

Flood, I. & Kartam, N. 1994. Neural networks in civil engineering I Principles and understanding. *Journal of Computing in Civil Engineering* 8(2): 131–148.

Freeze, R.A. & Harlan, R.L. 1969. Blueprint for a physically-based, digitally simulated hydrologic response model. *Journal of Hydrology* 9: 237–258.

Freissinet, C., Vauclin, M. & Erlich, M. 1999. Comparison of first-order analysis and fuzzy set approach for the evaluation of imprecision in a pesticide groundwater pollution screening model. *Journal of Contaminant Hydrology* 37(1–2): 21–43.

French, M.N., Krajewski, W.F. & Cuykendall, R.R. 1992. Rainfall forecasting in space and time using a neural network. *Journal of Hydrology* 137: 1–31.

Gautam, M.R., Watanabe, K. & Saegusa, H. 2000. Runoff analysis in humid forest catchment with artificial neural network. *Journal of Hydrology* 235(1–2): 117–136.

Govindaraju, R.S. & Rao, A.R. (Eds.) 2000. *Artificial Neural Networks in Hydrology.* Dordrecht: Kluwer Academic.

Gumrah, F., Oz, B., Guler, B. & Evin, S. 2000. The application of artificial neural networks for the prediction of water quality of polluted aquifer. *Water Air and Soil Pollution* 119(1–4): 275–294.

Gupta, J.N.D. & Smith, K.A. 2000. Neural networks in business, Special Issue. *Computers & Operations Research* 27: 1017–1247

Hall, M.J. & Minns, A.W. 1993. Rainfall-runoff modelling as a problem in artificial intelligence: experience with a neural network. *Proceedings of the Fourth National Hydrological Symposium,* Cardiff, 5.51–5.57.

Haykin, S.S. 1998. *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall.

Hong, Y-S. & Rosen, M.R. 2001. Intelligent characterisation and diagnosis of the groundwater quality in an urban fractured-rock aquifer using an artificial neural network. *Urban Water* 3(3): 217–228.

Hsu, K.L., Gupta, H.V. & Sorooshian, S. 1995. Artificial neural network modelling of the rainfall runoff process. *Water Resources Research* 31: 2517–2530.

Islam, S. & Kothari, R. 2000. Artificial neural networks in remote sensing of hydrologic processes. *Journal of Hydrologic Engineering* 5(2): 138–144.

Jonoski, A. 2002. *Hydroinformatics as Sociotechnology – Promoting Individual Stakeholder Participation by Using Network Distributed Decision Support Systems*. Rotterdam: Balkema.

Kneale, P.E. & See, L. 2003. Forecasting river stage with artificial neural networks. In G. Clarke & J.C.H Stillwell, (Eds.), *Applied GIS and Spatial Analysis*. Chichester: John Wiley.

Kneale, P.E., See, L., Cameron, D., Kerr, P. & Merrix, R. 2000. Using a prototype neural net forecasting model for flood predictions on the Rivers Tyne and Wear. *Proceedings of the British Hydrological Society, Seventh National Hydrology Symposium,* 3.45–3.49.

Lek, S. & Guegan, J.F. 1999. Artificial neural networks as a tool in ecological modelling, an introduction. *Ecological Modelling* 120(2–3): 65–73.

Ma, Q.C., Wang, J.T.L., Shasha, D. & Wu, C.H. 2001. DNA sequence classification via an expectation maximization algorithm and neural networks: A case study. *IEEE Transactions on Systems Man and Cybernetics Part C – Applications and Reviews* 31(4): 468–475.

Maier, H.R. & Dandy, G.C. 2001a. Neural network based modelling of environmental variables: A systematic approach. *Mathematical and Computer Modelling* 33 (6–7): 669–682.

Maier, H.R. & Dandy, G.C. 2001b. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling and Software* 15(1): 101–124.

Maskey, S., Dibike, Y.B., Jonoski, A. & Solomatine, D. 2000. Groundwater model approximation with artificial neural network for selecting optimum pumping strategy for plume removal. In O. Schleider & A. Zijderveld (Eds.), *AI Methods in Civil Engineering* (pp. 45–58). Rotterdam: Cottbus

Masters, T. 1995. *Neural, novel & hybrid algorithms for time series prediction.* New York: John Wiley & Sons.

McCulloch, W.S. & Pitts, W. 1943. A logic calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5: 115–133.

Morshed, J. & Powers, S.E. 2000. Regression and dimensional analysis for modeling two-phase flow. *Transport In Porous Media* 38(1–2): 205–221.

Nash, J.E. & Sutcliffe, J.V. 1970. River flow forecasting through conceptual models, I, A discussion of principles. *Journal of Hydrology* 10: 282–290.

Ouenes, A. 2000. Practical application of fuzzy logic and neural networks to fractures reservoir characterization. *Computers and Geosciences* 26(8): 953–962.

Picton, P. 2000. *Neural networks* (2nd ed.). Basingstoke: Palgrave.

Pidaparti, R.M., Jayanti, S. & Palakal, M.J. 2002. Residual strength and corrosion rate predictions of aging aircraft panels: Neural network study. *Journal of Aircraft* 39(1): 175–180.

Pokrajac, D. 2001. Groundwater in urban areas. *Urban Water* 3(3): 155.

Price, R.K. 2000. Hydroinformatics and urban drainage: an agenda for the beginning of the 21st century. *Journal of Hydroinformatics* 2: 85–86.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. 1986. Learning internal representations by error propagations. In D.E. Rumelhart & J.L. McClelland, (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition* 1 (pp. 318–362). Cambridge, MA: MIT Press.

Scheffer, C. & Heyns, P.S. 2001. Wear monitoring in turning operations using vibration and strain measurements. *Mechanical Systems and Signal Processing* 15(6): 1185–1202.

Tansel B., Jordahl C. & Tansel, I. 1999. Mapping of subsurface contaminant profiles by neural networks. *Civil Engineering and Environmental Systems* 16(1): 37–50.

Thuraisingham, B. 1998. *Data Mining: Technologies, Techniques, Tools, and Trends*. New York: CRC Press.

Tingsanchali, T. & Gautam, M.R. 2000. Application of tank, NAM, ARMA and neural network models to flood forecasting. *Hydrological Processes* 14(14): 2473–2487.

Tominaga, O., Ito, E., Hanai, T., Honda, H. & Kobayashi, T. 2002. Sensory modeling of coffee with a fuzzy neural network. *Journal of Food Science* 67(1): 363–368.

Venkateswaran, S., Rai, M.M., Govindan, T.R. & Meyyappan, M. 2002. Neural network modeling of growth processes. *Journal of the Electrochemical Society* 149(2): G137–G142.

Wang, W.J., Liao, Y.F. & Chen, S.H. 2002. RNN-based prosodic modeling for mandarin speech and its application to speech-to-text conversion. *Speech Communication* 36(3–4): 247–265.

Wilby, R.L., Abrahart, R.J. & Dawson, C.W. 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrological Sciences Journal* 48(2): 163–181.

Zio, E. 1997. Approaching the inverse problem of parameter estimation in groundwater models by means of artificial neural networks. *Progress in Nuclear Energy* 31(3): 303–315.

Zyngier, D., Araujo, O.Q.F. & Lima, E.L. 2001. Hybrid estimator for a wastewater treatment process. *Latin American Applied Research* 31(5): 507–511.

2

Neural Network Modelling: Basic Tools and Broader Issues

ROBERT J. ABRAHART
School of Geography, University of Nottingham, UK

ABSTRACT: The purpose of this chapter is to define and illustrate the basic terms and concepts involved in neural network modelling. The main neurohydrological modelling tools used to date are introduced. The chapter also includes an insight for new users into the scope and function of potential neural network hydrological modelling applications with respect to the broader hydrological picture.

1 INTRODUCTION

This chapter discusses the main elements in the neural network (NN) toolbox; it also addresses the 'what' and 'when' of NN hydrological modelling. Section 2 contains a brief introduction to the mechanisms and procedures involved – which includes a discussion on architectures and learning; while Section 3 contains a detailed description of the most popular tools that have been used in the field of water related research. Sections 2 and 3 are intended to complement one another and are designed to impart the minimum amount of information that would be required to understand the various operations and processes that are adopted in neurohydrological modelling. There are several respected sources that can be consulted for a more authoritative and comprehensive discussion on generic NN modelling items or issues of interest. Bishop (1995) and Masters (1995) are good academic texts; each book contains a copious amount of in-depth material. Reed and Marks (1999) is oriented towards the developer and practitioner. It describes selected techniques in sufficient detail, such that real-world solutions could be implemented, and technical issues or operational problems could be resolved. Section 4 illustrates the range of different hydrological possibilities and potentials that exist in which to develop and implement

a neural solution. Section 5 highlights the numerous opportunities and benefits that are on offer and further strengthens the argument for increased research into the provision of data-driven models. Sections 4 and 5 are thus intended to bolster appeal and to encourage uptake amongst interested parties; the exploration and testing of unorthodox strategies and alternative mindsets can indeed be a rewarding experience that leads to fresh insights and discoveries.

2 WHAT IS A NEURAL NETWORK?

NN are structures which forecast and predict through pattern matching and comparison procedures. NN tools are, in most cases, non-linear adaptive information processing structures that can be 'described mathematically' (Fischer, 1998). NN can exist as real-time hard-wired mechanisms, software simulators, optical processors and specialized neurocomputing chips (Taylor, 1993) and their computational elements are generic. NN software simulation programs, written in a standard high-level language, are the most common form.

There are a number of commercial and public domain simulators that users can select from, depending upon their preferred computer platform, and the sophistication offered in such packages provides a significant attraction. Catalogues of established software and shareware can be found on the World Wide Web e.g. NEuroNet (2001) or Sarle (2002). See Table 1.1 for a more comprehensive list. It is both an advantage and a potential drawback that users can download and install powerful NN products and packages with little or no real effort e.g. Stuttgart Neural Network Simulator (SNNS Group, 2003). Trained NN solutions can also be converted into dedicated 3GL (Third Generation Language e.g. C++) functions, for amalgamation into home-grown software products, or linked to commercial applications using a run-time connection based on standard software libraries (e.g. DLL). This is a major advantage for users, and especially new users wishing to experiment with the technique, but all users must be clear about the pros and cons of this modelling procedure.

2.1 *Network architecture*

NN are constructed from two basic building blocks: processing units (also referred to as elements or nodes or neurons) and weighted connections (also referred to as arcs or edges or links). These components and their respective organisation, into a set of interconnected items, form the 'network architecture'.

Maren (1991) has suggested that the architectural configuration can be described at three basic levels and this framework is used to explain the components here:

(a) ***Microstructure***. The characteristics of each processing unit in a network.
(b) ***Mesostructure***. The manner in which a network is organised, including such features as the number of layers, the pattern of connections, and the flow of information.

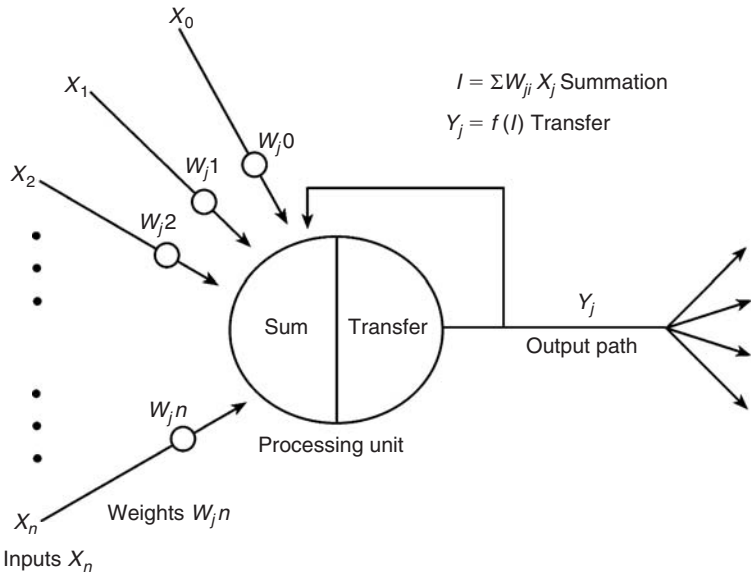


Fig. 2.1. The microstructure of a neural network model in terms of processing units.

(c) ***Macrostructure***. The manner in which a number of 'networks' are linked together, interacting with each other to build a more complex solution, for more demanding tasks.

Figure 2.1 illustrates the standard organization of an individual processing unit – which is the microstructure. Each processing unit can have numerous incoming connections, that arrive from other processing units, or from the 'outside world' $X_1 \ldots X_n$. The 'outside world' could be raw input data, or outputs produced from another forecasting model, that exports data to the NN. The connections function as unidirectional paths that conduct signals or data, and transmit their information in a predetermined direction. These are the user-defined 'input connections' and there is no upper limit on their number. There is also a program default input, termed bias, that is a constant $X_0 = 1$. Each processing unit first computes an intermediate value that comprises the weighted sum of all its inputs $I = \Sigma W_{ji} X_j$. This value is then passed through a transfer function $f(I)$, which performs a non-linear 'squashing operation'. The user can opt for default transfer functions or in certain software packages define their own – the standard options being logistic, sigmoid, linear, threshold, gaussian and hyperbolic tangent – with the selection of an appropriate transfer function being dependant upon the nature of each specific problem and its proposed solution. Shamseldin *et al.* (2002) explored the application of several different transfer functions to the amalgamation of multi-model hydrological forecasts and found that in most cases a logistic function provided the best results and an arctan function produced the worst results. Each processing unit can have numerous output connections, that lead to other processing units or to the 'outside world', and again there is no restriction on their number. Each

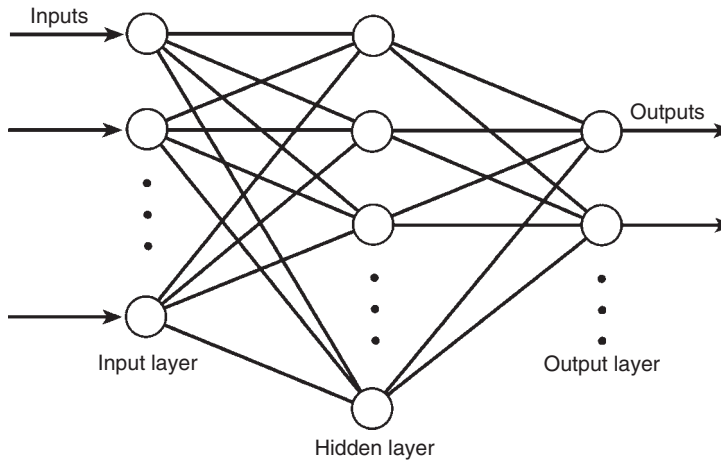


Fig. 2.2. The mesostructure of a neural network model in terms of processing units.

output connection carries identical copies of each numerical output, or signal, which is the state, or activation level, of that processing unit Y_j. The weights are termed 'connection parameters'. It is these weights that are adjusted during the learning process, to determine the overall behaviour of the neural solution, and that in combination generate the so-called 'network function'.

Figure 2.2 illustrates the standard organisation of a network architecture – which is the mesostructure. The basic structure consists of a number of processing units, arranged in a number of layers, and connected together to form a network. Data enters the network through the input units (left). It is then passed forward, through successive intermediate hidden layers, to emerge from the output units (right). The outer layer, where information is presented to the network, is called the input layer and contains the input units. These units disperse their input values to units in the next layer and serve no other function or purpose. The layer on the far side, where processed information is retrieved, is called the output layer and contains the output units. The layers in between the two outer layers are called hidden layers, being hidden from direct contact with the outside world, and contain the hidden units. Full connection is said to exist if each node in each layer is connected to all nodes in each adjacent layer. To avoid confusion the recommended method for describing a NN is based on the number of hidden layers. Figure 2.2 thus depicts a one-hidden-layer feedforward architecture with no feedback loops. However, it is also possible to have connections that transfer information backwards from output units to input units, from output units to hidden units, or from a unit to itself. These are termed partial-recurrent networks (PRNN) – see Van den Boogaard (Chapter 7) and Ball and Luk (Chapter 10). If the internal connections circulate information from each node to all other nodes then it is a recurrent network.

The use of storage tanks and chronological updating procedures is a familiar concept to the hydrologist and such items comprise an integral part of most

conceptual models and distributed modelling solutions. Thus far, however, in direct contrast most published NN hydrological modelling applications have been based on static models that contain no explicit consideration of time, previous events, antecedent conditions or state-space evolution – with no attempt being made to account for the complex interaction that should in fact occur between sequential representations of different but related input-output 'snapshots'. It is therefore argued that feedback loops could perhaps be used to address this issue, through the addition and circulation of dedicated variables that change or update specific factors in response to previous computations, and thus provide a dynamic and responsive solution that is better suited to modelling hydrological processes.

The number of processing units in the input and output layers is fixed according to the number of variables in the training data and is specific to each individual problem depending on the number of predictors and predictands. But the selection of an optimal number of hidden layers and hidden units will in all cases depend on the nature of the application. Intuition suggests that 'more is better' – but there are limits on the extent to which this is true. In certain instances a small(er) number of hidden units is advantageous. The number of hidden units and layers is important, since a larger architecture will extend the power of the model to perform more complex modelling operations, but there is an associated trade-off between the amount of training involved and the level of generalisation achieved. The use of large hidden layers can also be counterproductive since an excessive number of free parameters encourages the overfitting of the network solution to the training data, and so reduces the generalisation capabilities of the final product (Fig. 2.3). The other question that needs to be addressed is the number of hidden layers and the relative organisation of their hidden units. Practical methods to establish an 'optimum' set of hidden features range from best guess (e.g. Cheng & Noguchi, 1996) or trial and error (e.g. Shamseldin, 1997) to the application of sophisticated computational solutions e.g. cascade correlation which is a constructive algorithm (Imrie *et al.*, 2000; Lekkas *et al.*, 2001); weight or node based pruning which is a destructive algorithm (Abrahart *et al.*, 1999); or evolution-based approaches using a dedicated genetic algorithm package (Abrahart *et al.*, 1999). In the first instance inexperienced users might opt for one hidden layer with the number of hidden units equal to the number of inputs. More experienced users might match the number of hidden units to an anticipated number of empirical functions.

2.2 *Learning considerations*

NN 'learning' is defined as 'deliberate or directed change in the knowledge structure of a system that allows it to perform better or later repetitions of some given type or task' (Fischler & Firschein, 1987). Specific information on a particular topic or task is thus encoded in order for the solution to produce a suitable response on subsequent occasions. The two most common types of learning are supervised and unsupervised: the difference between them is that supervised learning requires each input pattern to have an associated output pattern. In

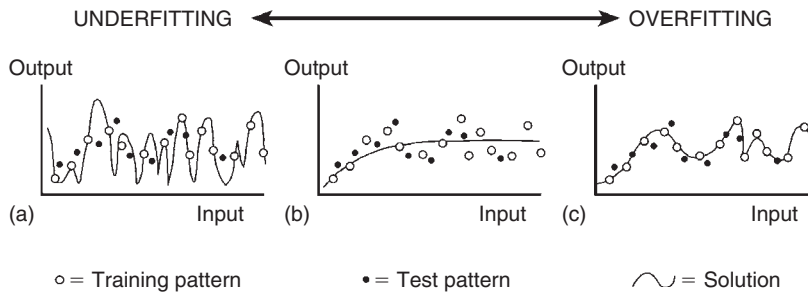


Fig. 2.3. The training trilemma (adapted from Flood & Kartam, 1994).

supervised training the model input might be discharge data collected at one or more upstream gauges with the output being forecast discharge at a downstream station. Cameron *et al*. (2002), for example, used a combination of river stage at two upstream stations and two local variables to estimate future river stage at a downstream station. In unsupervised training the output is in most cases a set of clusters; for instance river level series can be partitioned into different categories of event (Abrahart & See, 2000); rainfall and river series records can be partitioned to establish combined clusters that span the total input space (Hsu *et al*., 2002); catchments can be clustered into homogeneous categories that possess similar geomorphological and climatological characteristics (Hall *et al.*, 2002).

Each combination of input and output data is referred to as a training pair and the complete set of training pairs is the training set. The training period for the presentation of an entire training set is one epoch. The goal of training is to minimise the output error, which is achieved through the use of different algorithms that 'search the error surface' and 'descend the gradient'. Inputs (predictors) are passed through the network to become the outputs (predictands) and through the learning process the internal connection weights are modified in response to computed error – the equation that specifies this change is termed the 'learning law' or 'learning rule'. There are a large number of different learning methods and the learning process is often complex, with numerous options, variables, and permutations to choose from.

The learning process is continued until such time as an acceptable solution is arrived at. This is accomplished through numerous repeated iterations of data presentation and weight updates, until such time as an acceptable prespecified stopping condition is met, and the underlying function has been 'discovered'. However, it is important to ensure that the network does not become over-familiarised with the training data, and thus lose its power to generalise to unseen data sets. Figures 2.3 and 2.4 illustrate the basic problem of underfitting (undertraining) and overfitting (overtraining). The data set used in this process may be referred to as a 'validation' set.

If a neural solution has insufficient complexities, or has been underfitted, it will fail to detect the full signal in a complicated data set. If the neural solution is too complex, or has been overfitted, it will fit the noise as well as the signal. To differentiate between these opposing situations in an effective manner

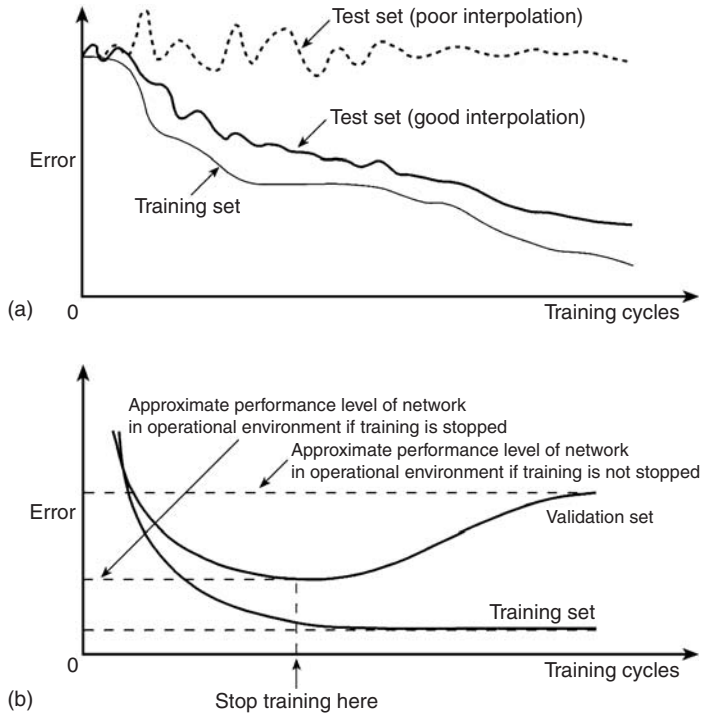


Fig. 2.4. Two possible scenarios for a plot of network error against training cycles. In each case overfitting arises when the solution learns the exact nuances of each individual case in the training data such that the final product has limited or no real interpolation capabilities (a) after Flood and Kartam (1994) (b) after Caudhill and Butler (1992).

is problematic and continuous assessment would be required throughout the different stages of construction and development. Several techniques are available to prevent overfitting:

(a) **Jitter:** addition of artificial noise to the input data during training that will produce a smoother final mapping between inputs and outputs e.g. Abrahart and White (2001).

(b) **Weight Decay:** addition of an extra term to the error function that penalises large weights in order to create a smoother final mapping between inputs and outputs – but no hydrological modelling investigation of this method has been reported.

(c) **Early Stopping:** use of split-sample validation, cross-validation or bootstrapping techniques to determine that point at which a sufficient degree of learning has taken place. For a comparison between continuous cross-validation and continuous bootstrapping applied to discharge forecasting see Abrahart (2003).

(d) **Structural Control:** restrict the number of hidden units and weighted connections such that a limited number of free parameters is available during the 'fitting process'. Each hidden node in each solution will attempt to

represent a discrete input-output association; so in the case of discharge forecasting simple functions such as 'quickflow' and 'baseflow' will be assigned to specific hidden nodes, whereas more complex entities such as 'soil moisture switches', would be assigned to one or more of the unclaimed units. Wilby *et al*. (2003) illustrate the inner workings of this mechanism, in a series of river-level forecasting experiments, in which a conceptual model is cloned with a number of neural solutions.

3 MAIN CATEGORIES OF MODEL

Neural networks are often promoted as a one-stop-shop but *caveat emptor* applies; users must recognise that there are several important decisions that must be taken to select an appropriate class of model. Certain forms of solution might be better suited to modelling specific hydrological functions or processes – although this notion is still quite novel and extensive testing will be required before indicative outcomes could be converted into a set of definitive guidelines. Different types of solution can nevertheless be differentiated in terms of:
(a) node characteristics i.e. properties of the processing units;
(b) network topologies i.e. the pattern of connections; and
(c) the learning algorithm and its associated parameters.

The number of possible combinations and permutations that could be implemented is enormous and to perform a detailed and comprehensive analysis is impractical. However, for hydrological modelling purposes, the three most common tools are:
(a) BPNN – backpropagation neural network;
(b) RBFN – radial basis function network;
(c) SOFM – self-organising feature map.

3.1 *Backpropagation neural network (BPNN)*

The most popular 'default' training algorithm is 'backpropagation' (Rumelhart *et al.*, 1986; Tveter, 2003). This technique offers an efficient computational procedure for evaluating the derivatives of the network performance function, with respect to a given set of network parameters, and corresponds to a propagation of errors backwards through the network. The term has however been extended to describe feedforward multi-layered networks trained using the backpropagation algorithm – which causes confusion. BPNN have emerged as major workhorses in various areas of business and commerce; such tools are also the most popular mechanism that has been applied to 'difficult' hydrological modelling problems (Maier & Dandy, 2000). The first port-of-call for most users will be a standard backpropagation network – but should initial trials prove inadequate – then see Imrie *et al*. (2000) who adopted non-standard activation functions or Mason *et al.* (1996) who examined different options in terms of the number of epochs to convergence.

Backpropagation for a multi-layered network works as follows:

The weighted input to each processing unit is:

$$I_i = \sum_{j=1}^{n} w_{ij}\, x_j \tag{1}$$

The output from each processing element is a sigmoid function, the most common being:

$$f(I) = \frac{1}{1 + e^{-I}} \tag{2}$$

with the derivative:

$$f'(I) = \frac{\mathrm{d}f(I)}{\mathrm{d}I} = f(I)(1 - f(I)) \tag{3}$$

Weight updates are based on a variation of the generalised delta rule:

$$\Delta w_{ij} = \beta\, Ef(I) + \alpha \Delta w_{ij}^{\text{previous}} \tag{4}$$

where β is the learning rate, E is the error, $f(I)$ is the output from a processing unit in the previous layer (incoming transmission), α is the momentum factor, and where $0.0 < \beta < 1.0$ and $0.0 < \alpha < 1.0$. Error for the output layer is desired output minus actual output:

$$E_j^{\text{output}} = y_j^{\text{desired}} - y_j^{\text{actual}} \tag{5}$$

whereas error for a hidden processing unit is derived from error that has been passed back from each processing unit in the next forward layer. This error is weighted using the same connection weights that modified the forward output activation value, and the total error for a hidden unit is thus the weighted sum of the error contributions from each individual unit in the next forward layer. To ensure a stable mathematical solution the total error for each unit is then multiplied using the derivative of the output activation, for that unit, in the forward pass:

$$E_i^{\text{hidden}} = \frac{\mathrm{d}f\,(I_i^{\text{hidden}})}{\mathrm{d}I} \sum_{j=1}^{n} (w_{ij} E_j^{\text{output}}) \tag{6}$$

which is an operation that is propagated backwards across the network. Following training, input data are then passed through the trained network in recall mode, where the presented data are transformed within the hidden layers to provide the required modelling output values.

BPNN are used to perform non-linear regression operations; such mechanisms will develop a function that relates a set of inputs to a set of outputs in

a data-driven environment. This tool is well suited to several different kinds of hydrological modelling, with reported applications that range from rainfall-runoff forecasting (Minns & Hall, 1996) and algal growth prediction (Whitehead *et al*., 1997), to the construction of rating curves (Tawfik *et al*., 1997; Jain, 2001) and the provision of error updates (Shamseldin & O'Connor, 2001) or multi-model data fusions (Abrahart & See, 2002). For more information on the mathematics involved see Bishop (1995).

3.2 *Radial basis function network (RBFN)*

The second most popular model is a radial basis function network (RBFN). This form of network has three layers: input, hidden, and output. The main architectural differences between an RBFN and a standard BPNN is that in the former the connections between the input units and the hidden units are not weighted and the transfer functions in the hidden units possess radial-symmetric properties (as opposed to sigmoidal). The hidden units perform a fixed non-linear transformation with no adjustable parameters and the output layer combines these results in a linear fashion, in most cases, using a simple summation (Leonard *et al*., 1992).

Figure 2.5 provides a schematic diagram of an RBFN. The activation function in each hidden unit is not critical to the performance of the network (Chen *et al*., 1991) and several forms of basis function have to date been considered. The most common form is the gaussian kernel. There are N inputs, L hidden layer units, and M output layer units for the general transformation of ND points $X(X^1, \ldots, X^i, \ldots, X^{ND})$ in the input space to ND points $Y(Y^1, \ldots, Y^i, \ldots, Y^{ND})$ in the output space. The parameters of the network consist of the centers (U_j) and the spread (σ_j) of the basis functions in the hidden layer units and the synaptic weights (E_{kj}) of the output layer units. The function centres are also points in the input space; so to have a unit at each distinct point on the input space would be the ideal solution. But for most problems a few inputs points will be selected from the full set of all available points using a clustering process.

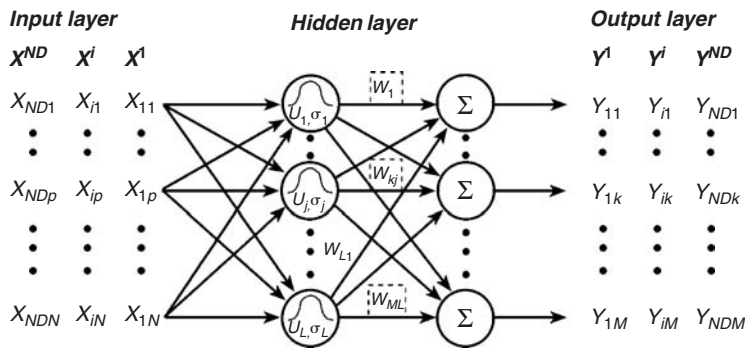


Fig. 2.5. Schematic diagram of a radial basis function network (after Jayawardena & Fernando, 1998).

For an input vector X^i the jth hidden unit produces the following response:

$$h_j = \exp\left\{\frac{-\left\|X^i - U_j\right\|}{2\sigma_j^2}\right\} \tag{7}$$

where $\|X^i - U_j\|$ is the distance between the point representing the input X^i and the centre of the jth hidden unit as measured according to some norm. Euclidian distance is the common measure.

The output of the network at the kth output unit will then be:

$$Y_{ik} = \sum_{j=1}^{L} h_j W_{kj} \tag{8}$$

Training is carried out using a hybrid learning procedure that contains both unsupervised and supervised methods. First, in the unsupervised phase, statistical clustering techniques are used for the estimation of kernel positions and kernel widths e.g. a k-means based clustering algorithm. Training of the hidden layer involves determination of the radial basis functions by specifying appropriate U_j and σ_j values for each unit. Clustering provides an effective method to reduce the number of centres from the ideal *ND* input points such that each point then falls into one of several hyperspheres, which on a collective basis, span the entire input space. The value of σ_j is computed as the mean distance from the centre of a cluster to the other points that form that cluster. The number of hidden units is therefore equal to the number of clusters.

Second, in the supervised learning phase, adjustment of the second layer of connections is implemented using linear regression or a gradient-descent technique. The output unit functions are in most cases linear, so the application of an initial non-iterative algorithm is commonplace, and often sufficient. However, if need be, a supervised gradient-based algorithm can be utilised in a further step to refine the connection parameters. The appropriate connection weights, between units in the hidden and the output layers, would thus be determined for example using either least mean squares or the backpropagation algorithm.

RBFN (in comparison to BPNN) have fast convergence properties and do not suffer from the problematic effects of local minima. However, although the training process could indeed be orders of magnitude faster, such networks require more training data and more hidden units, to achieve the same levels of approximation. The nature of their two non-linear activations is also quite different: RBFN non-linearities are derived from data in the training set; BPNN non-linearities are a fixed function of the pre-determined sigmoid transfer equation.

RBFN are also used to perform non-linear regression operations and can be applied to several different kinds of hydrological modelling task. RBFN and BPNN are thus similar in the sense of what can or cannot be achieved; the main difference between the two algorithms is in the method of construction. RBFN

are much quicker to train, which could save a vast amount of time and effort, and will be of particular benefit for solutions that must be developed on massive (e.g. global) data sets. The two are often compared and contrasted; the initial findings from such investigations are mixed, but comparative results suggest that the fast build solution is a viable option in terms of forecasting skill. Example applications in the field of runoff forecasting can be found in Mason *et al.* (1996); Jayawardena and Fernando (1998); and Dawson and Wilby (1999). For more information on the mathematics involved, again see Bishop (1995).

3.3 *Self-organising feature map (SOFM)*

The third most popular model is a self-organising feature map (SOFM) (Kohonen, 1995). This network algorithm is based on unsupervised classification in which the processing units compete against each other, to discover the important relationships that exist within a data set. There is no prior knowledge to assist in the clustering process. The traditional architecture contains two layers of processing units, a one-dimensional input layer and a two-dimensional competitive layer. The competitive layer, or feature map, is organised into a regular grid of processing units and each unit in the input layer is connected to each unit in the competitive layer. The feature map has connections between the competitive units and each competitive unit also has one or more additional weights, or reference vectors, that will be trained to represent the fundamental pattern associated with each class group. Training consists of (i) random weight initialisation; (ii) presenting a data pattern to the network; (iii) determining which unit has the closest match; and then (iv) updating both the winning unit and those around it. This process is repeated over numerous epochs until a stopping condition is reached. The training rule is:

$$\Delta w_i = \beta(x_i - w_i^{\text{old}}) \tag{9}$$

where w_i is the weight on the ith reference vector, β is the learning rate, x_i is the transmission along the ith weighted reference vector and where $0.0 < \beta < 1.0$.

The winning unit and its neighbours will adapt their reference vectors to better fit the current pattern, in proportion to the strength of the learning coefficient, whereas other units are either inhibited or experience no learning whatsoever. Lateral interaction is introduced between neighbouring units within a certain distance using excitors; beyond this area, a processing unit either (i) inhibits the response of other processing units or (ii) does not influence them at all. Two popular examples are (i) the Mexican Hat Function (Caudill & Butler, 1992 p.84) and (ii) the Square Block Function (Openshaw, 1994 p.64). The weight adjustment of the neighbouring units is instrumental in preserving the topological ordering of the input space. The neighbourhood for updating is then reduced, as is the learning coefficient, in two broad stages: (i) a short initial training phase in which a feature map is trained to reflect the coarser and more general details; and (ii) a much longer, fine-tuning stage, in which the local details of the

organisation are refined. This process continues until the network has stabilised and weight vectors associated with each unit define a multidimensional partitioning of the input data.

SOFM will perform unsupervised clustering and classification operations, to produce a set of clustered inputs, and for each cluster a set of internal vectors ordered in either one- or two-dimensional topological space. Two main applications are evident: 'divide and conquer' clustering in which a bigger problem is split into several smaller problems; and 'partner identification' clustering in which similar items are matched together for various purposes. The use of divide and conquer clustering involves the construction of less challenging and easier to model relationships, that can be resolved on an individual basis, taking a multi-model approach. For instance, river level series data can be partitioned into different categories of event, that are then modelled to produce a combined forecast (Abrahart & See, 2000); rainfall and river series records can be partitioned to establish combined records that span the total input space and which are then modelled to produce a combined forecast (Hsu *et al*., 2002). The use of partner identification clustering is about common properties and mutual associations. For instance, in regional flood estimation activities, catchments can be clustered into homogeneous categories possessing similar climatological, geomorphological, vegetation or soil characteristics (Hall *et al*., 2002). Clothiaux and Batchmann (1994) explore the relevant mathematics in detail.

4 WHAT USE IS A NEURAL NETWORK?

NN can be described as a generic 'solution in-waiting'; the burden is thus placed on the hydrologist to discover what can and cannot be done with these tools. This section examines the broad range and nature of the potential opportunities that are on offer to the hydrologist in terms of:

(a) implementation strategies with respect to standard tools (Section 4.1).
(b) the types of processing operation that can be undertaken and the manner in which specific problems can be resolved (Section 4.2).

4.1 *Different implementation methods*

There are three different methods of implementation that can be envisaged with respect to a traditional hydrological modelling solution. This breakdown is intended to provide a formal nomenclature and categorisation of several alternative possibilities, which is important, since given that neural solutions are at this time being brought into hydrological modelling, not just for straightforward linking of input (e.g. rainfall) to output (e.g. discharge at outfall, level of flooding, etc.), but also for example to clone traditional modelling mechanisms. Such replication, in turn, speeds up the process of computation so that items such as long time series analyses or Monte Carlo simulations can be done in a much faster and more efficient manner (Abebe *et al*., 2000; Gautam, 1998; Khindker *et al*., 1998; Liu & Lin, 1998). Neural solutions also demonstrate considerable

promise for efficient real time operational control, in which optimised scenarios can be selected at a particular stage/moment, depending on rainfall or other types of input-output forecast (Price, 2000; Bazartseren, 1999; Rahman, 1999).

4.1.1 *Replacement option*

Direct replacement involves using a neural model to perform an identical operation to something that would otherwise be performed with a traditional solution. Neural alternatives can for instance be developed to formulate rating curves that link stage and discharge under conditions of strong hysteresis (Tawfik *et al*., 1997), to simulate catchment response from environmental inputs associated with simple conceptual models (Shamseldin, 1997), or to produce a direct data-driven solution to the complex task of forecasting typhoon-related runoff events in a heterogeneous watershed (Chang & Hwang, 1999). Neural replacements can also facilitate the inclusion of additional data sources, irrespective of our comprehension about their relative hydrological roles. This offers considerable potential for exploring meaningful relationships where process data are inadequate for conventional deterministic forecasting. In situations where the theoretical understanding is limited, where equations are difficult to code, where data are nonstationary (Minns, 1996) or where data are limited to isolated points rather than areal distributions (Lorrai & Sechi, 1995), neurocomputation provides modelling alternatives that deserve exploration.

4.1.2 *Enhancement option*

Enhancement is the process of building hybrid mechanisms that contain an integrated bundle of traditional solution(s) and neural model(s). Enhancement could involve intelligent data pre-processing or post-processing operations, or be used to connect two or more existing models, as a simple filter which corrects inconsistencies in either input or output data streams, or to combine multi-model forecasts (Xiong *et al*., 2001; Abrahart & See, 2002; Shamseldin *et al*., 2002). The neural model might also function as a direct replacement for some internal component, or could be trained on residual data from an existing model, and then run in parallel acting as an error corrector or real-time output updating mechanism (Shamseldin & O'Connor, 2001). The basic idea is that through the adoption of neural plug-ins or add-ons it is possible to improve upon traditional techniques. Modern neural software will also facilitate the inclusion of trained solutions in other software packages, through the use of dynamic links, or will export the finished product as a function that can be compiled as part and parcel of a bespoke program (Van den Boogaard & Kruisbrink, 1996; Abrahart, 1998; Abrahart *et al*., 2001).

4.1.3 *Cloning option*

Cloning is the process of using a neural model to mimic an existing equation-based solution – including inherent imperfections. In addition to offering rapid

improvement in processing speed and data handling capabilities, for instance in an integrated optimisation procedure (Rogers & Dowla, 1994; Solomatine & Avila Torres, 1996), such clones could also be used to reduce existing model calibration time through building a response surface that relates internal parameters to original output (Liong & Chan, 1993). Clones can also be constructed to include additional variables or to omit certain variables in those instances where one or more standard inputs are not available (Abrahart, 2001). It is also possible to mimic the internal functions of an existing model, for model reduction purposes, or for rapid prototyping, sensitivity analysis, and bootstrapping operations. Less obvious is the use of neural clones to mimic spatial distributions, thus making redundant our existing problems of storing and accessing copious amounts of spatial input, and enabling models to switch from file-based data retrieval (slower) to chip-based data computation (faster) operations.

4.2 *Different types of processing operations*

Fischer and Abrahart (2000) proposed a five-fold classification of NN applications based on the type of modelling that is being performed. This breakdown provides a useful insight into some of the different kinds of functions and operations that could be considered as suitable targets for a NN hydrological modelling implementation. Each model could operate alone, or in combination with other modelling constructs, as part and parcel of a hybrid formulation.

4.2.1 *Pattern classification/recognition*

The task of pattern classification is to assign an input pattern to one of several pre-specified class groups, (either user-defined, or determined from an automated clustering exercise). This is a decision making process that will often involve the use of 'hard' boundaries in which the class groups represent a set of discrete and exclusive entities. However, hard boundaries are not appropriate for continuous hydrological variables that inter-grade. These demand a 'softer' approach – one that is able to encapsulate a certain degree of natural fuzziness.

Classification is employed in spectral pattern recognition where ground-truth information is utilised for the transformation of individual pixel values on multispectral images into given land cover categories (Kanellopoulos *et al*., 1997; for an appraisal of progress with respect to remote sensing of hydrological processes see Islam and Kothari 2000 or Foody (Chapter 14)). It is also possible to perform pattern recognition of an entire image to determine different weather patterns, or to differentiate between cloud patterns that are cumuliform and stratiform, in order to switch between different short range meteorological forecasting tools (Pankiewicz, Chapter 13).

4.2.2 *Clustering/categorisation*

In clustering operations (also known as unsupervised pattern classification) there are no pre-specified, or accepted, class group labels attached to the training data.

The clustering algorithm is used to explore the data and to determine the intrinsic similarities that exist between the various patterns which make up a particular data set. Each item in that data set is then identified as belonging to a cluster of similar patterns. Popular clustering applications would include data mining, data compression, and partitioning of the hydrograph.

Hall and Minns (1999) performed neural clustering on flood data to produce homogeneous regions that facilitated the subsequent transfer of information from gauged to ungauged catchments (not between neighbouring catchments based on geographical proximities). Abrahart and See (2000) used neural clustering to divide river level data into different hydrograph sectors or components. These component clusters were then used to illustrate the benefits of multi-network modelling applications.

4.2.3 *Function approximation*

Numerous hydrological problems require non-linear function approximation. The task of function approximation is to create a generalised model of some known or unknown function. Suppose a set of n training patterns (input-output pairs) have been collected and are associated with an unidentified function $f(x)$ which are subject to noise. Then the task of function approximation would be to build a (descriptive or mathematical or computational) model which is able to reproduce that, in most cases, continuous function.

Examples include the provision of numerical solutions for: flood quantiles in ungauged catchments based on data from neighbouring areas (Liong *et al*., 1994); two-year peak stream discharge based on geographical and meteorological variables (Muttiah *et al*., 1997); soil water retention curves based on soil sample data (Schaap & Bouten, 1996): unit hydrographs related to small catchments in Bavaria (Lange, 1999); and the relationship between rainfall-runoff and sediment transfer in Malawi (Abrahart & White, 2001).

4.2.4 *Time series forecasting*

In mathematical terms given a set of n samples Yt_1, Yt_2,…, Yt_n in a time sequence t_1, t_2,…, t_n then the role of forecasting would be to estimate the sample $Y(t)$ at some future point in time (often t_{n+1}). Time series prediction is an important requirement and the final product can have a significant impact on decision-making with respect to operational forecasting, for example in reservoirs and power plants, and in life threatening real-time flood warning assessment. The estimation of associated sediment, ecological and pollution variables is also important.

Examples include modelling stream output from a conceptual model, based on a time series of simulated storm events, in which the sequence generator was calibrated to represent linear and non-linear catchment response (Minns & Hall, 1996); modelling real hydrological and meteorological time series data on a fifteen minute time step with a six hour lead time for flood forecasting purposes (Dawson & Wilby, 1998); and a solution to address particular hydrological issues associated with modelling low-flow periods (Campolo *et al*., 1999).

4.2.5 *Optimisation*

Certain types of hydrological modelling can be posed as a (non-linear) optimisation problem, in which the goal is to find a solution that satisfies a set of constraints, such that some objective function is maximised or minimised. The solution will depend on a number of factors or possibilities; with an enormous number of possible combinations often rendering such problems either insoluble or too time consuming for more conventional methods of analysis and modelling. Traditional model calibration procedures are a case in point.

Liong and Chan (1993) describe a neural optimisation process for the calibration of a traditional model based on the inherent relationship between model parameters and output. Hybrid implementations are also possible in which the speed of a neural solution can be exploited e.g. in combination with a genetic algorithm to establish optimal groundwater remediation (Rogers & Dowla, 1994; Rao & Jamieson, 1997); or as an integrated part of a larger multi-criteria reservoir optimisation exercise (Solomatine & Avila Torres, 1996).

5 POTENTIAL BENEFITS TO BE EXPLOITED

Neurocomputing will never become a universal panacea and there is no question of it making conventional computer methodologies or traditional modelling procedures obsolete. These modern tools are not in all cases better than conventional statistical methods or traditional mathematical models; although it is often possible to get equivalent good results, in a fast and efficient manner, working with minimum sized data sets. Neural solutions do however possess a number of distinctive features and properties – which should, at least in principle, enable these automated tools to exceed various inherent limitations or drawbacks associated with standard information processing techniques. It is the strength of these specific items that should therefore be investigated, understood, and exploited to their full potential. The main potential with respect to neural modelling opportunities in this field of science is considered to fall within the following categories.

5.1 *Power to handle complex non-linear functions*

If a function can be represented as a mapping between two or more vectors then it can also be approximated, albeit with uncertain precision, using a neural solution. Such opportunities would be of particular benefit in all cases where relationships are difficult to determine or in situations where hard and fast rules cannot be applied.

5.2 *Power to perform model-free function estimation*

Nothing is assumed or enforced in NN. There are no pre-specified relationships, constraints, or single solution conditions. This has obvious advantages for modelling situations in which there is limited available information on the nature of the data distribution functions, when the data are non-gaussian, or when different levels of generalisation are required.

5.3 *Power to learn from training data*

No prior knowledge of the underlying mechanisms is needed. The modelled relationships are derived from a synergistic combination of training data and learning procedures. This is important since certain functions and processes are not fully understood in the field or laboratory. Some processes also suffer from poor definition or are misunderstood. In other cases the process might be well understood but algorithms for their implementation do not exist, or full data sets to operationalise such equations cannot be obtained at a commensurate level of detail.

5.4 *Power to adapt to changing environments*

The solution is not static. It can adapt to accommodate alternative data, processes or constraints, to produce a different model. This form of dynamic adaptation would be useful in a fast-changing environment, to resolve a problem using anticipated trends, to take account of changes in the nature of the problem, or help to resolve an old problem in a fresh manner.

5.5 *Power to handle incomplete, noisy and fuzzy information*

Distributed processing means that each specific component is responsible for one small part of the total input-to-output 'mapping operation'. The neural solution will thus possess substantial 'fault tolerant' characteristics; it would generate an acceptable response, plus exhibit graceful degradation, based on incomplete, imprecise or noise-ridden data.

5.6 *Power to effect multi-level generalisation*

Solutions are developed through the process of 'construction'. This process can be halted at different stages as the solution moves from a more general, to a more specific, representation. If the relationship can be described using some sort of rule then it will tend to discover that rule; with high levels of noise it will extract the central trend, or prototype, of that particular data set.

5.7 *Power to perform high speed information processing*

Neural models are fast. Their inherent parallelism and simple algorithms can provide rapid throughput. Speed is crucially important in most real-time hydrological applications where real-time response is required to events. Other operations can be made viable with this technique: production of bootstrap confidence intervals; replacement of 'number crunching' with 'model crunching'; detailed research on vast data sets for example at the scale and magnitude of global warming.

6 CONCLUSIONS

There can be little doubt, following a decade or so of essential groundwork, that neural solutions are well suited to the challenging task of hydrological forecasting and prediction. The most important item for the successful implementation of each solution will be the acquisition of appropriate and representative

data (Tokar & Johnson, 1999; Smith & Eli, 1995) and the proper division of this material into training, validation and testing data sets (Bowden *et al*., 2002). From then on the implementation of an effective model will be to a large extent dependent upon the skill and experience of each individual neurohydrologist. The modeller is faced with a number of potential opportunities and alternative strategies at each stage of the model building process and this search space must henceforth be reduced. Future advances will thus be contingent upon (i) the creation of detailed working protocols that (ii) contain objective guidelines for the development and application of each individual modelling solution. These guidelines must distinguish between those circumstances, under which a particular approach should be adopted, and the manner in which to best optimise the numerous procedures and parameters that exist therein. Effort should also be directed towards the identification of specific tasks and circumstances in which particular strategies might under-perform or might perhaps fail – to delimit the boundaries of their application.

REFERENCES

Abebe, A.J., Solomatine, D.P. & Venneker, R.G.W. 2000. Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events. *Hydrological Sciences Journal* 45(3): 425–436.

Abrahart, R.J. 1998. Neural networks and the problem of accumulated error: an embedded solution that offers new opportunities for modelling and testing. In V. Babovic & C.L. Larsen (Eds.), *Proceedings Third International Conference on Hydroinformatics, Copenhagen, Denmark, 24-26 August 1998*. Vol. 2. 725–731. Rotterdam: A.A. Balkema.

Abrahart, R.J. 2001. *Neural network methodologies for hydrological prediction and forecasting*. Unpublished PhD Thesis, University of Leeds, UK.

Abrahart, R.J. 2003. Neural network rainfall-runoff forecasting based on continuous resampling. *Journal of Hydroinformatics* 5(1): 51–61.

Abrahart, R.J & See, L. 2000. Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes* 14: 2157–2172.

Abrahart, R.J. & See, L. 2002. Multi-model data fusion for river flow forecasting: an evaluation of six alternative methods based on two contrasting catchments. *Hydrology and Earth System Sciences* 6(4): 655–670.

Abrahart, R.J., See, L. & Kneale, P.E. 1999. Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model. *Journal of Hydroinformatics* 1(2): 103–114.

Abrahart, R.J., See, L. & Kneale, P.E. 2001. Investigating the role of saliency analysis with a neural network rainfall-runoff model. *Computers and Geosciences* 27(8): 921–928.

Abrahart, R.J. & White, S. 2001. Modelling Sediment Transfer in Malawi: comparing backpropagation neural network solutions against a multiple linear regression benchmark using small data sets. *Physics and Chemistry of the Earth (B)* 26(1): 19–24.

Bazartseren, B. 1999. *Use of artificial neural networks and fuzzy adaptive systems in controlling polder water levels in the Netherlands*. Unpublished MSc Thesis, IHE Delft, the Netherlands.

Bishop, C.M. 1995. *Neural Networks for Pattern Recognition*. Oxford: Clarendon Press.

Bowden, G.J., Maier, H.R. & Dandy, G.C. 2002. Optimal division of data for neural network models in water resources applications. *Water Resources Research* 38(2): 1–11.

Cameron, D., Kneale, P. & See, L. 2002. An evaluation of a traditional and a neural net modelling approach to flood forecasting for an upland catchment. *Hydrological Processes* 16: 1033–1046.

Campolo, M., Soldati, A. & Andreussi, P. 1999. Forecasting river flow rate during low-flow periods using neural networks. *Water Resources Research* 35(11): 3547–3552.

Caudill, M. & Butler, C. 1992. *Understanding Neural Networks: Computer Explorations.* Vol. 1: Basic Networks. Cambridge, MA: MIT Press.

Chang, F. & Hwang, Y. 1999. A self-organization algorithm for real-time flood forecast. *Hydrological Processes* 13: 123–138.

Chen, S., Cowan, C.F.N. & Grant, P.M. 1991. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks* 2(2): 302–309.

Cheng, X. & Noguchi, M. 1996. Rainfall-runoff modelling by neural network approach. *Proceedings International Conference on Water Resources and Environment Research: Towards the 21st Century, Kyoto, Japan, 29–31 October 1996*. Vol. 2. 143–150.

Clothiaux, E.E. & Batchmann, C.M. 1994. Neural networks and their applications. Chapter 2. In B.C. Hewitson & R.G Crane (Eds.), *Neural Nets: Applications in Geography*. Dordrecht: Kluwer Academic Publishers.

Dawson, C.W. & Wilby, R.L. 1998. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal* 43(1): 47–66.

Dawson, C.W. & Wilby, R.L. 1999. A comparison of artificial neural networks used for river flow forecasting. *Hydrology and Earth System Sciences* 3(4): 529–540.

Fischer, M.M. 1998. Computational neural networks – a new paradigm for spatial analysis. *Environment and Planning A* 30: 1873–1892.

Fischer, M.M. & Abrahart, R.J. 2000. Neurocomputing – Tools for Geographers. Chapter 8. In S. Openshaw & R.J. Abrahart (Eds.), *GeoComputation* (pp. 187–217). London: Taylor & Francis.

Fischler, M.A. & Firschein, O. (Eds.) 1987. *Readings in computer vision: Issues, problems, principles, and paradigms*. Morgan Kaufmann Publishers.

Flood, I. & Kartam, N. 1994. Neural Networks in Civil Engineering. 1: Principles and Understanding. *Journal of Computing in Civil Engineering* 8(2): 131–148.

Gautam, D.K. 1998. *Regressive neural networks for the modelling of time series*, Unpublished MSc Thesis, IHE Delft, the Netherlands.

Hall, M.J. & Minns, A.W. 1999. The classification of hydrologically homogeneous regions. *Hydrological Sciences Journal* 44(5): 693–704.

Hall, M.J., Minns, A.W. & Ashrafuzzaman, A.K.M. 2002. The application of data mining techniques for the regionalisation of hydrological variables. *Journal of Hydrology and Earth System Sciences* 6(4): 685–694.

Hsu, K.-L., Gupta, H.V., Gao, X., Sorooshian, S. & Iman, B. 2002. Self-organizng linear output map (SOLO); An artificial neural network suitable for hydrologic modeling and analysis. *Water Resources Research* 38(12): 1302, doi: 10.1029/2001WR000795.

Imrie, C.E., Durucan, S. & Korre, A. 2000. River flow prediction using artificial neural networks: generalisation beyond the calibration range. *Journal of Hydrology* 233: 138–153.

Islam, S. & Kothari, R. 2000. Artificial Neural Networks in Remote Sensing of Hydraulic Processes. *Journal of Hydraulic Engineering* 5(2): 138–144.

Jain, S.K. 2001. Development of Integrated Sediment Rating Curves Using ANNs. *Journal of Hydraulic Engineering* 127(1): 30–37.

Jayawardena, A.W. & Fernando, D.A.K. 1998. Use of radial basis function type artificial neural networks for runoff simulation. *Computer-aided Civil and Infrastructure Engineering* 13(2): 91–99.

Kanellopoulos, I., Wilkinson, G.G., Roli, F. & Austin, J. 1997. (Eds.) *Neurocomputation in Remote Sensing Data Analysis*. Berlin: Springer.

Khindker, M.U.L-H., Wilson, G. & Klinting, A. 1998. Application of neural nets in real time flash flood forecasting. In V. Babovic & C.L. Larsen (Eds.), *Proceedings Third International Conference on Hydroinformatics, Copenhagen, Denmark, 24–26 August 1998*. Vol. 2. 777–782. Rotterdam: A.A. Balkema.

Kohonen, T. 1995. *Self-Organizing Maps*. Heidelberg: Springer-Verlag.

Lange, N.T. 1999. New mathematical approaches in hydrological modeling – an application of artificial neural networks. *Physics and Chemistry of the Earth* 24(1–2): 31–35.

Lekkas, D.F., Imrie, C.E. & Lees, M.J. 2001. Improved non-linear transfer function and neural network methods of flow routing for real-time forecasting. *Journal of Hydroinformatics* 3(3): 153–164.

Leonard, J.A., Kramer, M.A. & Ungar, L.H. 1992. Using radial basis functions to approximate a function and its error bounds. *IEEE Transactions on Neural Networks* 3(4): 624–627.

Liong, S.Y. & Chan, W.T. 1993. Runoff Volume Estimates with Neural Networks. In B.H.V. Topping & A.I. Khan (Eds.), *Neural Networks and Combinatorial Optimization in Civil and Structural Engineering, Proceedings Third International Conference on the Application of Artificial Intelligence to Civil and Structural Engineering, Edinburgh, UK, 17–19 August 1993,* 67–70.

Liong, S.Y., Nguyen, V.T.V., Chan, W.T. & Chia, Y.S. 1994. Regional Estimation of Floods for Ungaged Catchments with Neural Networks. In H-F. Cheong, N.J. Shankar, E-S. Chan & W-J. Ng (Eds.), *Developments in Hydraulic Engineering and their impact on the Environment, Proceedings Ninth Congress of the Asian and Pacific Division of the International Association for Hydraulic Research, Singapore, 24–26 August 1994,* 372–378.

Liu, C.L. & Lin, S.C. 1998. Artificial neural network forecasting for chaotic dynamical processes. In V. Babovic & C.L. Larsen (Eds.), *Proceedings Third International Conference on Hydroinformatics, Copenhagen, Denmark, 24–26 August 1998*. Vol. 2. 791–796. Rotterdam: A.A. Balkema.

Lorrai, M. & Sechi, G.M. 1995. Neural nets for modelling rainfall-runoff transformations. *Water Resources Management* 9: 299–313.

Maier, H.R. & Dandy, G.C. 2000. Neural networks for the prediction of water resources variables: a review of modelling issues and applications. *Environmental Modelling and Software* 15: 101–124.

Maren, A.J. 1991. A logical topology of neural networks. *Paper presented at the Second Workshop on Neural Networks: Academic/Industrial/NASA/ Defense, Auburn University, Auburn, Alabama, February 1991.*

Mason, J.C., Tem'me, A. & Price, R.K. 1996. A neural network model of rainfall-runoff using radial basis functions. *Journal of Hydraulic Research* 34(4): 537–548.

Masters, T. 1995. *Neural, Novel & Hybrid Algorithms for Time Series Prediction.* New York: John Wiley & Sons.

Minns, A.W. 1996. Extended rainfall-runoff modelling using artificial neural networks. In A. Müller (Ed.), *Proceedings 2nd International Conference on Hydroinformatics, Zurich, Switzerland, 9–13 September 1996*. Vol. 1. 207–213. Rotterdam: A.A. Balkema.

Minns, A.W. & Hall, M.J. 1996. Artificial neural networks as rainfall-runoff models. *Hydrological Sciences Journal* 41(3): 399–417.

Muttiah, R.S., Srinivasan, R. & Allen, P.M. 1997. Prediction of two-year peak stream discharges using neural networks. *Journal of the American Water Resources Association* 33(3): 625–630.

NEuroNet. 2001. *Neural Network Products*. Last modified: 23/1/2001 URL = http://www.kcl.ac.uk/neuronet/products/index.html (August 2003).

Openshaw, S. 1994. Neuroclassification of spatial data. Chapter 3. In B.C. Hewitson & R.G. Crane (Eds.), *Neural Nets: Applications in Geography*. Dordrecht: Kluwer Academic Publishers.

Price, R.K. 2000. Hydroinformatics and urban drainage: an agenda for the beginning of the 21st Century. *Journal of Hydroinformatics* 2(2): 133–147.

Rahman, S.M.M. 1999. *Analysis and prediction of chaotic time series*. Unpublished MSc Thesis, IHE Delft, the Netherlands.

Rao, Z. & Jamieson, D.G. 1997. The use of neural networks and genetic algorithms for design of groundwater remediation schemes. *Hydrology and Earth System Sciences* 1(2): 245–356.

Reed, R.D. & Marks, R.J. 1999. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press.

Rogers, L.L. & Dowla, F.U. 1994. Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling. *Water Resources Research* 30(2): 457–481.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. 1986. Learning internal representations by error propagations. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructures of Cognition.* Vol. 1. 318–362. Cambridge, MA: MIT Press.

Sarle, W.S. 2002. *FAQ posting for Usenet newsgroup comp.ai.neural-nets*. Last modified: 17/5/2002. URL = ftp://ftp.sas.com/pub/neural/FAQ.html (August 2003).

Schaap, M.G. & Bouten, W. 1996. Modelling water retention curves of sandy soils using neural networks. *Water Resources Research* 32(10): 3033–3040.

Shamseldin, A.Y. 1997. Application of a neural network technique to rainfall-runoff modeling. *Journal of Hydrology* 199: 272–294.

Shamseldin, A.Y., Ahmed, E.N. & O'Connor, K.M. 2002. Comparison of different forms of the multi-layer feed-forward neural network method used for river flow forecast combination. *Hydrology and Earth System Sciences* 6(4): 671–684.

Shamseldin, A.Y. & O'Connor, K.M. 2001. A non-linear neural network technique for updating of river flow forecasts. *Hydrology and Earth System Sciences* 5(4): 577–597.

Smith, J. & Eli, R.N. 1995. Neural-network models of rainfall-runoff process. *Journal of Water Resources Planning and Management* 121: 499–509.

SNNS Group. 2003. *Stuttgart Neural Network Simulator.* URL = http://www-ra.informatik.uni-tuebingen.de/SNNS/ (August 2003).

Solomatine, D.P. & Avila Torres, L.A. 1996. Neural network approximation of a hydrodynamics model in optimizing reservoir operation. In A. Müller (Ed.), *Proceedings 2nd International Conference on Hydroinformatics, Zurich, Switzerland, 9–13 September 1996.* Vol. 1. 201–206. Rotterdam: A.A. Balkema.

Tawfik, M., Ibrahim, A. & Fahmy, H. 1997. Hysteresis sensitivity neural network for modeling rating curves. *Journal of Computing in Civil Engineering* 11(3): 206–211.

Taylor, J.G. 1993. *The Promise of Neural Networks*. London: Springer-Verlag.

Tokar, S.A. & Johnson, P.A. 1999. Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering* 4(3): 232–239.

Tveter, D.R. 2003. *Backpropagator's Review.* Last modified: 14/1/2003. URL = http://www.dontveter.com/bpr/bpr.html. (August 2003).

Van den Boogaard, H.F.P & Kruisbrink, A.C.H. 1996. Hybrid modelling by integrating neural networks with numerical models. In A. Müller (Ed.), *Proceedings 2nd International Conference on Hydroinformatics, Zurich, Switzerland, 9–13 September 1996.* Vol. 2. 471–478. Rotterdam: A.A. Balkema.

Whitehead, P.G., Howard, A. & Arulmani, C. 1997. Modelling algal growth and transport in rivers: a comparison of time series analysis, dynamic mass balance and neural network techniques. *Hydrobiologia* 349: 39–46.

Wilby, R.L., Abrahart, R.J. & Dawson, C.W. 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrological Sciences Journal* 48(2): 163–181.

Xiong, L., Shamseldin, A.Y. & O'Connor, K.M. 2001. A non-linear combination of the forecasts of rainfall-runoff models by the first-order Takagi-Sugeno fuzzy system. *Journal of Hydrology* 245(1–4): 196–217.

3

Single Network Modelling Solutions

CHRISTIAN W. DAWSON
Department of Computer Science, Loughborough University, UK
ROBERT L. WILBY
Environment Agency, Trentside Offices, Nottingham NG2 5FA, UK

ABSTRACT: Artificial neural networks have been applied to the problem of rainfall-runoff modelling and flood forecasting since the late 1980s. However, despite the extensive literature base which has built up in this field, there is not a systematic method that the neural network engineer can use to develop such models. This chapter introduces such a method and explains, through the use of a case study experiment, how each of the stages of this process are executed to develop a single network modelling solution.

1 INTRODUCTION

Before developing artificial neural network (NN) rainfall-runoff models, it is important to have a fundamental understanding of both the application area (hydrology) and the modelling tool. The field of NN research is diverse, which makes it difficult but nevertheless essential to: (i) grasp the fundamental workings of this 'tool' and (ii) develop an appreciation of the broad range of NN models available, the different training approaches that can be used and, perhaps most importantly, understand their strengths and limitations.

Hydrological forecasting and NN are discussed elsewhere within this book. It is the purpose of this chapter to provide an 'instruction handbook' on how to build a NN model of rainfall-runoff processes and how to evaluate the model(s) thus developed.

The NN field is constantly evolving and new network types and training algorithms are constantly being discovered, defined and refined (for example, 'support vector machines' are an area of current research – see Haykin, 1999). However, while these areas provide interesting fields of exploration, such tools

are of limited value to the 'neurohydrologist' until they become established and accessible. For this reason, the case study presented in this chapter is based on an established NN model and training algorithm – the multi-layer perceptron (MLP), trained using the popular error backpropagation algorithm (BPNN). This is used in the majority of hydrological applications and is probably the best starting point for building NN models of a specific river catchment.

When developing NN to model rainfall-runoff processes, a number of decisions must be made. For example, as implied above, one must choose a suitable network from the plethora of different types available and then go on to choose an associated training algorithm. One must also decide how to process the available data. A number of considerations must be made in terms of partitioning the data into training, validation and test data sets, identifying predictors, defining network outputs and on standardising or normalising the data. While some of these operations may be automated using appropriate modifications to the training algorithm, many decisions must still be made through a process of trial-and-error. A full discussion of these topics is beyond the scope of this chapter and interested readers are directed towards authoritative texts such as Bishop (1995).

The remainder of this chapter is divided as follows. First, an overview of the NN model development process is introduced. Second, an introduction to the case study is presented. Third, this case study is used as a vehicle to explain the process stages outlined. Finally, a summary is presented and areas for future research are discussed.

2 THE PROCESS

Dawson and Wilby (2001) introduced a seven stage process to develop NN models for hydrological applications. This has been reduced to the following six stages:

1. Data selection: gather an appropriate data set.
2. Select an appropriate predictand: decide what is to be modelled (flow volume, flow stage or depth, changes in flow, etc.) and the accompanying lead time.
3. NN selection: select an appropriate type of network and choose a suitable training algorithm.
4. Data preprocessing: process the original data in terms of identifying suitable network inputs (predictors) and perform data cleansing (i.e. remove or reduce problematic artefacts) as appropriate – for example, if necessary, remove trends or seasonal components. In addition, one must normalise or standardise the data and split the data into training, validation and testing data sets.
5. Training: train a number of networks using the chosen training algorithm and preprocessed data.
6. Using appropriate assessment criteria, evaluate the model(s) produced and select the 'best' solution for subsequent implementation.

This process is represented in Figure 3.1, adapted from Dawson and Wilby (2001). Solid lines show the process flow, dashed lines indicate influences of one process stage on another, and subprocesses are identified by rectangles with double-lined sides.

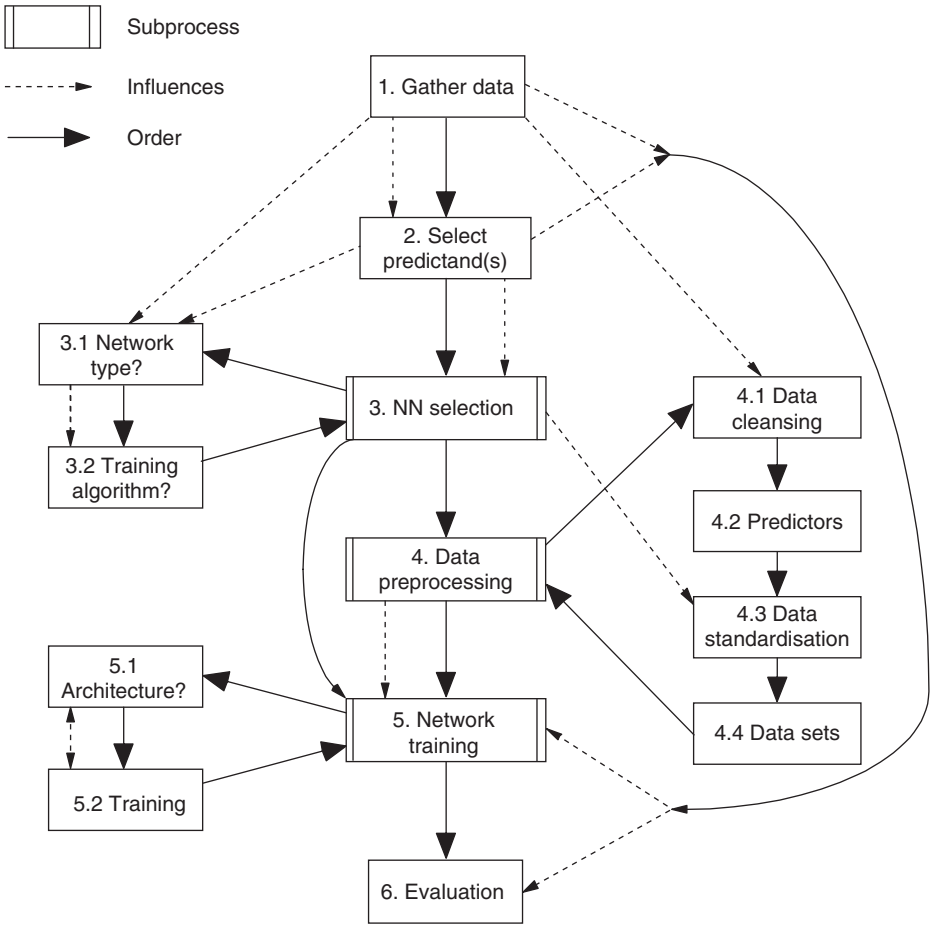


Fig. 3.1. NN model development process advocated by Dawson and Wilby (2001).

It is difficult to identify which of these stages is the most important. Without an adequate data set (gathered in Stage 1), a meaningful study would be impossible. However, even if a suitable data set is available, should any of the other stages of the process be performed badly, unsuitable models and poor evaluation results will be produced. Each of these stages is discussed in more detail in the case study below.

3 CASE STUDY

3.1 *Introduction*

The purpose of this section is to apply the NN development process model outlined above to a case study in order to show how the stages are implemented in practice. This case study is based on the ANNEXG (Artificial Neural Network Experiment Group) experiments undertaken during 2001/2002 by thirteen

Table 3.1. Case study catchment descriptors.

Descriptor	Description
Catchment area (Ha)	66.2
Elevation (m)	150–250
Geology	Pre-Cambrian outcrops comprising granites, pyroclastics, quartzites and syenites
Soils	Brown rankers, acid brown soils and gleys
Land use	Bracken heathland (39%), mixed deciduous woodland (28%), open grassland (23%), coniferous plantation (6%), open deciduous and bracken under-storey (2%), surface waters (<1%), urban (<1%)
Annual rainfall (mm)	700
Annual runoff (mm)	120
Runoff (%)	17
Drainage	Mainly open channel, with tile drains and soil piping at 50 cm

neurohydrologists worldwide. The concept behind the ANNEXG experiments was to provide a benchmark data set to a number of participants in an attempt to compare and contrast several different modelling approaches that are used by neurohydrologists and to evaluate the forecasting skill of the different models produced. Initial results from the ANNEXG experiments are reported in Dawson (2002). The results of the authors' contributions to the ANNEXG case study and the thinking behind the model development is presented in this chapter. Two contrasting approaches to data preprocessing operations – a complex method and a simple method – using the ANNEXG data set are described.

3.2 *Catchment description*

Table 3.1 provides the characteristics of the river catchment used in the ANNEXG study. The modelling was undertaken 'blindly' by all groups in order that none were disadvantaged through lack of first hand knowledge of the catchment. The site receives on average 700 mm of precipitation per year, distributed evenly across the seasons. However, the drainage network is restricted to the lowest part of the catchment, and comprises an ephemeral system of small inter-connected ponds and subsurface tile drains. Furthermore, a network of naturally occurring soil pipes at about 50 cm below the surface promotes rapid lateral flow during winter storms. The flow regime, therefore, ranges from zero flow during dry summer months to a 'flashy' response following rainfall (and occasional snow-melt) events in winter.

3.3 *Stage 1: Gather data*

> 'The success of an ANN application depends both on the quality and the quantity of data available' (ASCE, 2000:121)

It is paramount to the effective development of data driven models, such as NN, that sufficient data are available for model development. Sufficient in this

context means that the data are of high quality (i.e. free of errors, omissions, conflicts) and available in adequate quantity for a meaningful model. In other words, the information content is paramount (Tokar & Johnson, 1999). There is little point, for example, in using a data set that contains few historical flood events if flood forecasting is the primary objective of the model. In this case the NN model that is developed will not have been exposed to sufficient training examples to adequately predict flood events in the future. However, how we measure 'information content' and how we define 'sufficient examples' is unclear and is an area for further research. Other examples where data may be inappropriate for training include: direct transfer of data from another catchment; using data from different seasons; or using data gathered prior to fundamental changes in the catchment's properties (for example, urbanisation or deforestation). How models can be adopted and adapted, having been developed using a non-representative data set, is also an area for future research.

In the example presented in this chapter, three years of daily data were made available for a small experimental catchment in central England (see previous section). These data included stage (mm), precipitation (mm) and maximum daily air temperature (°C) for the period 1 January 1988 to 31 December 1990. This data series contained some missing values for each of the three variables – represented in the data set as -999. Two years of data were used for model development (1989 and 1990) and one year for independent testing (1988). 1988 was chosen as the testing period as this contained values outside the range of the data used in model development and thus provides a severe test of model skill. How the remaining data for 1989 and 1990 were split for training and validation purposes is discussed later.

3.4 *Stage 2: Select predictand(s)*

It is important to understand exactly what is to be modelled within the catchment and this must be stated clearly at the outset. Is it the intention, for example, to model flow volume or flow depth? If a flood forecasting model is required, what is an appropriate lead time – days or possibly weeks for large catchments? It might be better to model 'changes in flow' if the data contain a large variance (and model development data may be unrepresentative of long-term extremes). Some authors use change in flow rather than flow *per se* to reduce the likelihood of problems in extrapolating beyond the range of the model development data (e.g. Minns & Hall, 1997).

In the ANNEXG experiments, the intention was to evaluate model skill for forecasting depth of flow (or stage) at $t + 1$ and $t + 3$ days ahead. For simplicity and clarity, in this chapter only a $t + 1$ day ahead model is reproduced and presented as the case study.

3.5 *Stage 3: NN selection*

Since NN were repopularised in the 1980s, a plethora of different types (e.g. BPNN, radial basis function networks (RBFN), support vector machines,

probabilistic neural networks, etc.) and training algorithms (e.g. backpropagation, conjugate gradient, cascade correlation, skeletonisation, etc.) have been developed or rediscovered. In addition, there are numerous NN tools available to the neurohydrologist that automate the development and implementation of different networks and provide the user with a selection of training algorithms. Many software packages are available as either shareware or freeware that can be easily downloaded from the Internet. Further details are provided in Table 1.1. Alternatively, many researchers develop their own tools in their favourite programming language – enabling them to adapt their models more easily to different problems and ensuring that they understand, more clearly, the inner workings of the model.

It is beyond the intended scope of this chapter to discuss the full range of alternative architectures, training algorithms and tools that are available. However, as a starting point, it is best for the 'budding' neurohydrologist to choose a tool that is tried and tested. We advocate a simple BPNN, which can be used as a 'baseline' that should only be modified or replaced if more sophisticated models prove to be more accurate. This type of network and training algorithm are included within most NN tools and packages or can, if preferred, be programmed using no more than one or two pages of code. This is the approach taken in the case study reported in this chapter – a simple BPNN has been implemented (written in Pascal) and has been trained using backpropagation.

3.6 *Stage 4: Data preprocessing*

Data preparation involves a number of processes such as data 'cleansing', identifying appropriate predictors (using data reduction techniques), standardising or normalising the data, and finally, dividing the data into training, validation and testing data sets.

3.6.1 *Data 'cleansing'*

NN can, theoretically, handle incomplete, noisy and non-stationary data (Zealand *et al*., 1999), but with suitable data preparation beforehand, it is possible to improve their performance (Masters, 1995). Data cleansing involves identifying and removing trends and non-stationary components (in terms of the mean and variance) within a data set. For example, trends can be removed by differencing the time series and the data can be centred using rescaling techniques. Seasonal variability can be accommodated by using, for example, moving averages (Janacek & Swift, 1993). To date, data cleansing techniques have not been widely used in NN rainfall-runoff modelling so there is much scope for development in this area (Maier & Dandy, 1996).

As the test data used in the case study covered an entire year (1988), it was not necessary to remove any seasonal components from the data set. In addition, because the testing and model development periods were adjacent (1988, 1989–1990 respectively), there was no need to remove long term trends or cycles. However, the data set did contain missing values that required some preprocessing and two approaches were used. First, during training and validation,

any days containing missing values were simply ignored and removed from the data set. During testing, if a day contained missing data, then no prediction was attempted. This is a simple solution but one must question the acceptability of such an approach for real time implementation if the model cannot provide a prediction when data are unavailable (for example, if a rain gauge should fail).

As an alternative, a second, more complicated solution was investigated. In this case an additional input driver was added to the data set called 'missing data identifier'. This was set to zero when all predictors were available and one when one or more predictors were missing. Thus, during development, it was anticipated that the NN model would 'learn' to deal with missing inputs having been warned that data were missing by the extra input parameter.

Another approach that can be used is to infill missing data by interpolating between the last and next available data point. This is acceptable during model development but during testing in real time there is no 'next available' data point to work with. In such cases, monthly averages or a moving average could be used or even a value from the same day on a previous year. Each of these techniques has advantages and disadvantages and there is no one 'right' solution for dealing with missing data.

3.6.2 *Identifying suitable predictors*

In order to improve model performance it is useful if the most 'powerful' predictors can be identified. The majority of studies in the literature focus on predicting flow (as either discharge or stage) using antecedent or concurrent catchment conditions. In this case the NN is attempting to model a process of the form:

$$Q_{t+x} = f\left(Q_{t-n}, R_{t-n}, X_{t-n}\right) \tag{1}$$

in which Q_{t+x} is future flow (at x time steps in the future), Q_{t-n} is antecedent flow (at t, $t-1$, $t-2$, . . . , $t-n$ time steps), R_{t-n} is antecedent rainfall (at t, $t-1$, $t-2$, . . . , $t-n$), and X_{t-n} represents any other factors identified as affecting Q_{t+x}, e.g. *year type* such as wet or dry (Tokar & Johnson, 1999); *percentage impervious area* (Minns, 1996); or *storm occurrence* (Dawson & Wilby, 1998).

In order to improve performance, the neurohydrologist must first establish the optimal lag-interval between input and response. This can be achieved through the use of NN (e.g. Furundzic, 1998), ARMA models (Refenes *et al*., 1997) or autocorrelation functions. Auto Regressive Moving Average (ARMA) models are often used to determine appropriate variables, lead times and the optimal window(s) for averaging (Maier & Dandy, 2000). Alternatively, correlation testing may be used to identify the strongest causal relationships from a set of possible predictor variables (as in Dawson & Wilby, 1998). The chosen predictor variables are then applied as either individual inputs to multiple nodes (e.g. predictors are Q, Q_{t-1}, Q_{t-2}, Q_{t-3}, etc.) and/or as lumped averages (in which case an input node receives a moving average).

If the available data comprise many input variables but few points, it is important to attempt some form of data reduction. Otherwise, the model will

Table 3.2. Fourteen model predictors (M14) identified by preprocessing analysis for the $t + 1$ day ahead model.

Precipitation (t)
Temperature ($t - 7$)
Stage (t)
Sin Clock
Cos Clock
Precipitation minus temperature ($t - 5$)
Stage plus precipitation (t)
Stage minus temperature (t)
10 day moving average precipitation (t)
10 day moving average temperature ($t - 9$)
10 day moving average precipitation minus temperature (t)
6 day moving average stage plus precipitation (t)
6 day moving average stage minus temperature (t)
Missing data identifier (0 – no data missing, 1 – data missing)

have more free parameters to establish than data to constrain individual parameter values. Data reduction techniques might involve statistical manipulations, such as extracting principal components (e.g. Masters, 1995), or reducing physical data sets, by averaging rainfall data from several rain gauges (e.g. Chang & Hwang, 1999).

In our case study, two approaches have been used to determine appropriate input drivers for flow forecasting in the test catchment. First, data preprocessing and analysis were used in which moving averages (ranging from one to fifty days were calculated) and lags (also from one to fifty days) were correlated against flow. Those lagged and moving average variables that were most strongly correlated with flow were chosen as inputs to the model. In addition, two clock variables were included (Sine and Cosine of the Julian day number) to represent the seasonal cycle, along with a missing data flag (as discussed in the previous section). Further preprocessing of the data was undertaken to incorporate the interaction between variables. In this case, precipitation minus temperature, and stage minus temperature, were calculated to capture some sense of net loss through evapotranspiration from the catchment. Also, precipitation plus stage was calculated to provide some indication of 'wetness' within the catchment. These analyses led to the selection of the 14 input drivers shown in Table 3.2 (henceforth referred to as the M14 inputs).

A second, less sophisticated approach to identifying suitable inputs was used for comparison. In this case it was decided that antecedent variables for the previous three days would be used as input drivers (stage, precipitation and maximum air temperature). No attempt would be made, in this case, to deal with missing data and so missing data days were simply removed from the training, validation and testing data sets (as discussed in the previous section). This led to the nine inputs shown in Table 3.3 (henceforth referred to as the M9 inputs). No analyses were performed with these data – the lagged inputs were merely chosen 'intuitively'.

Table 3.3. Nine model predictors (M9) chosen for the $t + 1$ day ahead model.

Stage	$(t, t-1, t-2)$
Precipitation	$(t, t-1, t-2)$
Temperature	$(t, t-1, t-2)$

3.6.3 *Data standardisation*

All variables should be standardised to ensure that they receive equal weight during the training process (Maier & Dandy, 2000). Without standardisation, input variables measured on different scales will dominate training to a greater or lesser extent because initial weights within a network are randomised to the same finite range.

Data standardisation is also important for the efficiency of training algorithms. For example, error backpropagation used to train BPNN is particularly sensitive to the scale of the data used. Due to the nature of this algorithm, large values slow training because the gradient of the sigmoid function at extreme values approximates zero. To avoid this problem, data are rescaled using an appropriate transformation. In general, data are rescaled to the intervals $[-1, 1]$, $[0.1, 0.9]$ or $[0, 1]$ (referred to as *standardisation*). Another approach is to rescale values to a Gaussian function with a mean of zero and unit standard deviation (referred to as *normalisation*). The advantage of using $[0.1, 0.9]$ for runoff modelling is that extreme (high and low) flow events, occurring outside the range of the training and validation data, may be accommodated (Hsu *et al*., 1995). Other authors advocate $[0.1, 0.85]$ (e.g. Shamseldin, 1997) or $[-0.9, 0.9]$ (e.g. Braddock *et al*., 1998). In the present study, data were standardised to the range $[0.1, 0.9]$ as logistic sigmoid transfer functions were used.

3.6.4 *Data sets*

Provided one does not become caught in local error minima (which can be avoided using certain techniques), theoretically one could continue training NN indefinitely, steadily improving the accuracy of the model with respect to the training data. However, while the network might become very adept at modelling the training data, it may well lose its ability to generalise to new and unseen situations. Figure 3.2 highlights this situation and shows that after training a network for e_o epochs, the network begins to lose its ability to generalise and its performance with respect to unseen data begins to deteriorate. Thus, as training progresses, a validation set is used to halt training if it appears that the network is becoming over-trained on the training data. Ideally then, three data sets should be used for a rigorous analysis of NN skill: a training set, a validation set and a test set (called 'cross validation'). The training set is used to develop a number of different NN model configurations. The validation set is used to decide when to stop training (to avoid over-fitting) and also to determine which of the networks is the most accurate. Finally, the test set is used to evaluate the chosen model against independent data.

Lachtermacher and Fuller (1994) identify a number of problems with using three data sets. First, if there are limited data available, it can be impractical to

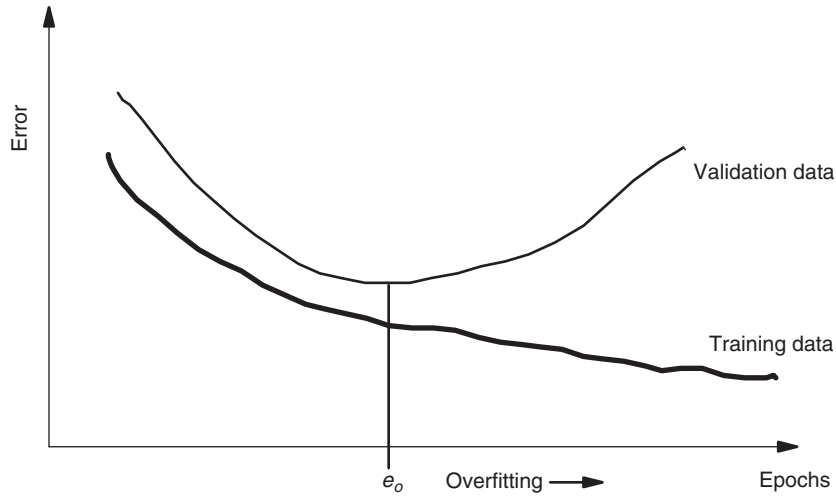


Fig. 3.2. Overfitting on the training data.

create three independent data sets. Second, the method of dividing the data can significantly affect the results. Third, when using a validation set to cease training, it is not always clear when a network is beginning to 'learn' the noise inherent to the time series.

With finite data availability, it is often prudent to use a 'cross training' technique. This method involves splitting the available data into S equal sized segments. NN models are then trained using all the data in S-1 of these segments and validated on the remaining segment of unseen data. The procedure is repeated S times so that S models are built and tested for each model type and configuration. This ensures that each data segment is used only once for validation. Thus, when the validation segments are recombined, one has a validation set equal to the entire data set. Typical values for S are 5 and 10 (Schalkoff, 1997). An alternative is to use the 'hold-one-out' or 'jack-knife' method, in which $S = n - 1$ (where n is the number of data points in the entire data set). Thus, for a data set containing n data points, one would have to create and test n NN models.

In the case study presented here, three years of daily data were available – 1989 and 1990 were used for model development, and 1988 for model testing purposes. In order to create a training and validation set, to avoid over-training, it was decided that the model development data should be split thus: 1989 was used as training data and 1990 as validation data. By using a full year of data for training, all seasonal variations were captured.

3.7 *Stage 5: Network training*

During training, the NN developer is trying to 'optimise' a number of things – the network architecture (number of hidden layers, number of nodes in each layer), the weights connecting neurons and the biases that are applied. As discussed earlier, there are a number of algorithms that exist that can be used

to (i) identify an appropriate architecture (quite often a trial-and-error approach is used) and/or (ii) optimise the interconnecting weights.

During training with the case study data, a simple trial-and-error approach was used to identify an optimal network structure. Thus, a number of networks were constructed with 3, 5, 10, 15, 20, 30 hidden nodes (single hidden layer), each being trained from 100 to 2000 epochs in steps of 100 epochs. After each period of 100 epochs, the network weights were saved and evaluated with respect to the unseen validation data. The network configuration (in terms of hidden nodes and epochs) that performed best against the validation data was chosen as the best NN model and thus used for the final evaluation (with the test data set). In this exercise, using the M14 predictor set, a network with five hidden nodes trained for 100 epochs proved most accurate with respect to the validation data. For the M9 predictor set, a network with 15 hidden nodes trained for 1300 epochs proved to be most accurate with respect to the validation data. In all cases BPNN was used, trained using error backpropagation, with the learning parameter fixed at 0.1 and the momentum value set at 0.9.

3.8 *Stage 6: Evaluation*

Dawson and Wilby (2001) provide a detailed discussion of assessment criteria and the evaluation of rainfall-runoff models. This discussion is reproduced here with some modification.

There is a general lack of objectivity and consistency in the way in which rainfall-runoff models are assessed or compared (Legates & McCabe, 1999). This also applies to the more specific case of NN model assessment and arises for several reasons. First, there are no standard error measures (although some have been more widely applied than others). Second, the diversity of catchments studied (in terms of area, topography, land use, climate regime, etc.) hinders direct comparisons. Third, different aspects of flow may be modelled (e.g. discharge, stage, change in discharge, etc.). Finally, there are broad differences between studies with respect to lead times (ranging from 0 to +24 model time steps) and the temporal granularity of forecasts (from seconds to months).

When NN are trained using algorithms such as backpropagation, they are generally optimised in such a way as to minimise their global error. While this is a useful general target, it does not necessarily lead to a network that is proficient for both low flow (drought) and high flow (flood) forecasting. The squared error, which is used in many training algorithms, does provide a general measure of model performance, but it does not identify specific regions where a model is deficient. Other error measures are, therefore, employed to quantify these deficiencies (see the review of Watts, 1997).

The most commonly employed error measures are: the mean squared error (MSE), the mean squared relative error (MSRE), the coefficient of efficiency (CE), and the coefficient of determination (r^2) (see Equations 2, 3, 4, and 5 respectively):

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2}{n} \qquad (2)$$

$$\mathrm{MSRE} = \frac{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2 / Q_i^2}{n} \tag{3}$$

$$\mathrm{CE} = 1 - \frac{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2}{\sum_{i=1}^{n}(Q_i - \overline{Q}_i)^2} \tag{4}$$

$$r^2 = \frac{\sum_{i=1}^{n}(Q_i - \overline{Q})(\hat{Q}_i - \tilde{Q})^2}{\sqrt{\sum_{i=1}^{n}(Q_i - \overline{Q})^2 \sum_{i=1}^{n}(Q_i - \tilde{Q})^2}} \tag{5}$$

where Q_i are the n modelled flows, $\hat{Q}_i$ are the n observed flows, $\overline{Q}$ is the mean of the observed flows, and $\widetilde{Q}$ is the mean of the modelled flows.

According to Karunanithi *et al*. (1994), squared errors (MSE) provide a good measure of the goodness-of-fit at high flows, whilst relative errors (MSRE) provide a more balanced perspective of the goodness of fit at moderate flows. However, the appropriateness of these measures are strongly affected by the nature of the flow regime, and so care must be taken when comparing results between sites.

CE and r^2, on the other hand, provide useful comparisons between studies since standardised measures are independent of the scale of the data used (i.e. flow, catchment, temporal granularity, etc.). These are correlation statistics that assess the goodness of fit of modelled data with respect to observed data. CE is referred to by some authors as the 'determination coefficient' (e.g. Cheng & Noguchi, 1996), the 'efficiency index', E (Sureerattanan & Phien, 1997; Abrahart & Kneale, 1997), the F index (Minns & Hall, 1996), and R^2 (Nash & Sutcliffe, 1970). Care must be taken not to confuse R^2 with the coefficient of determination, r^2, which some authors also refer to as R^2 (e.g. Furundzic, 1998, Legates & McCabe, 1999, Lorrai & Sechi, 1995).

The CE statistic provides a measure of the ability of a model to predict flows that are different from the mean [i.e. the proportion of the initial variance accounted for by the model (Nash & Sutcliffe, 1970)], and r^2 measures the variability of observed flow that is explained by the model (see the evaluation of Legates & McCabe, 1999). CE ranges from negative infinity in the worst case to +1 for a perfect correlation. According to Shamseldin (1997), a CE of 0.9 and above is very satisfactory, 0.8 to 0.9 represents a fairly good model, and below 0.8 is deemed unsatisfactory. r^2 ranges from −1 (perfect negative correlation), through 0 (no correlation) to +1 (perfect positive correlation).

Legates and McCabe (1999) highlight a number of deficiencies with relative measures such as CE and r^2. They note that r^2 is particularly sensitive to outliers and insensitive to additive and proportional differences between modelled and observed data. For example, a model could grossly, but consistently, overestimate the observed data values and still return an acceptable r^2 statistic.

Although CE is an improvement over r^2 (in that it is more sensitive to differences in modelled and observed means and variances), it is still sensitive to extreme values. The *index of agreement* measure, *d* (Equation 6), has been proposed as a possible alternative (Legates & McCabe, 1999) but owing to the use of squared differences, it is still sensitive to extreme values. Modified versions of *d* and CE have also been described which are both baseline adjusted (adjusted to the time series against which the model is compared) and adapted from squared to absolute differences. The second adaptation reduces the sensitivity of these measures to outliers. For a more thorough discussion, the interested reader is directed to Legates and McCabe (1999).

$$d = 1 - \frac{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2}{\sum_{i=1}^{n}(|\hat{Q}_i - \overline{Q}| + |Q_i - \overline{Q}|)^2} \tag{6}$$

Another error measure that has been used is S4E (presented as MS4E in Equation 7) by Abrahart and See (2000). This higher order measure places more emphasis on peak flows than the lower order MSE. Alternatively, the mean absolute error (MAE, Equation 8), which computes all deviations from the original data regardless of sign, is not weighted towards high flow events.

$$\text{MS4E} = \frac{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^4}{n} \tag{7}$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|Q_i - \hat{Q}_i|}{n} \tag{8}$$

Other measures that have been employed in only a limited number of cases include RMSE/μ [RMSE as a percentage of the observed mean (Fernando & Jayawardena, 1998; Jayawardena *et al*., 1997)]; %MF [percent error in modelled maximum flow relative to observed data (Hsu *et al*., 1995; Furundzic, 1998)]; %VE [percent error in modelled runoff volume (Hsu *et al*., 1995)]; $\%N_{RMSE}$ [percentage of values exceeding the RMSE (Campolo *et al*., 1999)]; and RMSNE [root mean squared normalised error (Atiya *et al*., 1996) defined as the square root of the sum squared errors divided by the square root of the sum squared desired outputs (Equation 9)].

$$\text{RMSNE} = \frac{\sqrt{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2}}{\sqrt{\sum_{i=1}^{n}Q_i^2}} \tag{9}$$

Lachtermacher and Fuller (1994) identified other measures for time series analysis such as the Average Relative Variance (Nowlan & Hinton, 1992) and the Mean Error (Gorr *et al*., 1992). Another measure often used in time series

analysis is Theil's U-statistic (Theil, 1966) which provides a relative basis for comparing complex and naïve models. However, these measures have yet to be used in the evaluation of NN rainfall-runoff models.

Classification approaches are also used to evaluate predictive models. For example, Colman and Davy (1999) used a classification technique to evaluate seasonal weather forecasts. In this technique the observed data were assigned to one of three equiprobable sets, or *terces* (in this case, below average, average and above average temperatures). Model skill (relative to chance) is then assessed using a Chi-square test of the modelled versus expected frequencies in each category. Similarly, Abrahart and See (2000) classified predictions according to: % correct; % predictions within ± 5, 10, 25% of observed and % predictions greater than $\pm 25\%$ of observed. This allows direct comparisons to be made between different models irrespective of the predictand and model time step.

While the above discussion relates more generally to rainfall-runoff modelling, flood forecasting systems need to employ additional error measures. For example, P-P (Dawson *et al*., 2000) is a measure of the error in the timing of a predicted flood peak [Chang and Hwang (1999) refer to this as Et_p]. Abrahart and See (2000) use MAEpp and RMSEpp which measure equivalent values to MAE and RMSE for all predicted peak flood events in a data set. These authors also employed a classification criteria which measures % early, % late and % correct occurrences of individual predicted peaks (although they do not indicate what discrepancy constitutes a 'late' peak). A further measure used for flood forecasting is total volume but this measure provides no indication of temporal accuracy (Zealand *et al*., 1999).

The measures introduced above take no account of the parsimony of the models. One would expect a model with many parameters to provide a better 'fit' to the data than one with fewer degrees of freedom. However, more complex models do not necessarily lead to proportionate increases in accuracy and one must question whether the additional effort is justifiable. Too many parameters may also result in over-fitting to the training data. Fortunately, several performance measures take into account the number of parameters used in a model. For example, the A Information Criteria – AIC (Akaike, 1974); the B Information Criteria – BIC (Rissanen, 1978); the Schwarz Information Criteria – SIC (Schwarz, 1978); the Vapnik-Chervonenkis dimension (Abu-Mostafa, 1989); or the Network Information Criteria – NIC (Murata *et al*., 1994). The AIC and BIC measures are defined as follows:

$$\mathrm{AIC} = m \ln(\mathrm{RMSE}) + 2p \tag{10}$$

$$\mathrm{BIC} = m \ln(\mathrm{RMSE}) + p \ln(m) \tag{11}$$

in which m is the number of data points and p is the number of free parameters in the model. These measures take into account the number of parameters used within a model and give credit to models that are more parsimonious. In both cases, lower scores indicate a more parsimonious model.

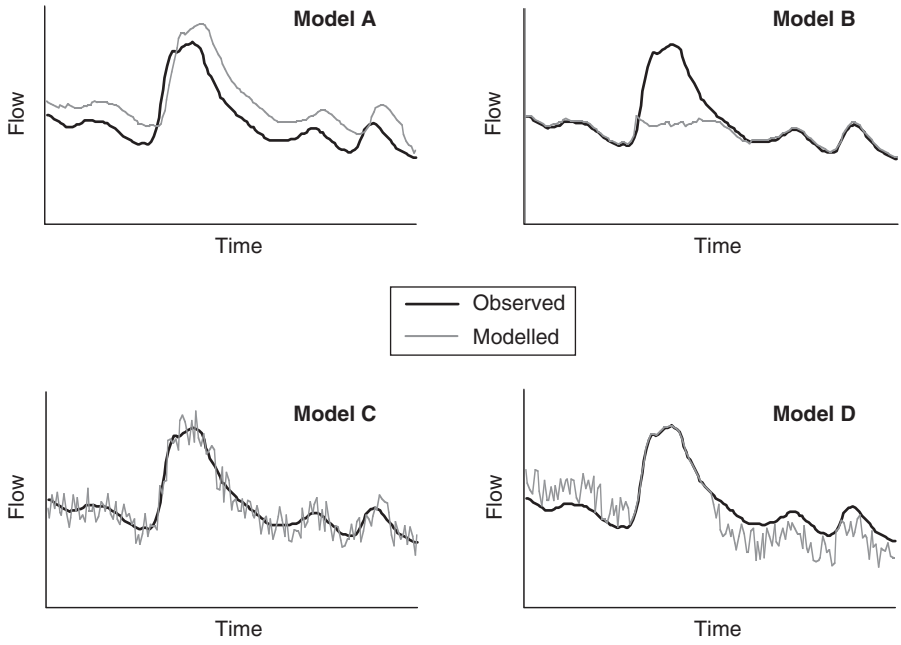


Fig. 3.3. Hydrographs of four hypothetical models (from Dawson & Wilby, 2001).

Given this assortment of performance measures, the problem then becomes one of deciding which (if any) are most appropriate to a particular application. For example, Figure 3.3 shows different types of model error produced by four hypothetical rainfall-runoff models. Model A, which is somewhat naïve, predicts the shape of the hydrograph well but consistently overestimates flow and predicts the flood peak late. Model B predicts low flows accurately but returns poor estimates of the flood peak. Model C simulates flow generally well but contains a lot of 'noise', and Model D reproduces flood events very well but performs poorly for low flows. Table 3.4 reports the error measures associated with each model. Model B may be selected in preference to model D based on the MSRE or MAE statistic. However, model D would be selected in preference to model B from the RMSE, CE, d and r^2 statistics. Model C consistently outperforms all other models based on the error statistics, but it is not as accurate as Model B during low flow periods, or Model D during flood events. Model A appears relatively weak when assessed using most of the error statistics, but it performs very well according to r^2. This echoes the results of Legates and McCabe (1999) who point out the imperfections of the r^2 statistic (which does not penalise additive and proportional differences).

The results in Table 3.4 emphasise the importance of not relying on individual error measures to assess model performance. Thus, goodness-of-fit error measures (e.g. CE, d, and r^2) and absolute error measures (RMSE and MAE) should be used in combination (Legates & McCabe, 1999).

Table 3.4. Error measures of four hypothetical models (reproduced from Dawson & Wilby, 2001).

	MSRE	RMSE	r^2	CE	d	MS4E	MAE
Model A	0.0510	61.24	0.827	44%	0.871	215	56
Model B	0.0243	72.28	0.397	21%	0.558	2003	34
Model C	0.0123	29.78	0.885	87%	0.968	14	26
Model D	0.0430	50.89	0.785	61%	0.922	152	39

Table 3.5. Comparative results of case study models.

Predictors	Parameters	RMSE	r^2	CE	AIC	BIC
M14	81	1.3517	0.948	93.96%	252	552
M9	166	1.3065	0.944	94.37%	409	1018

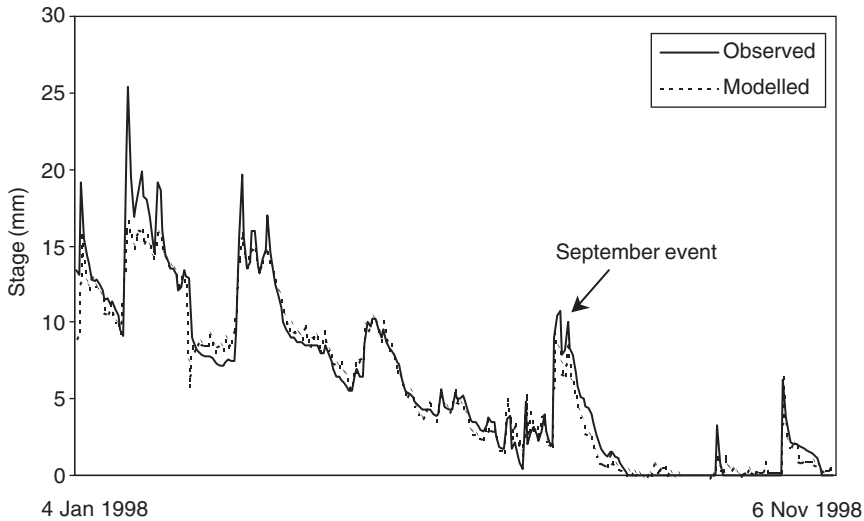


Fig. 3.4. Time series hydrograph of M14 model versus observed during test period.

3.9 *Case study results*

Table 3.5 presents summary statistics for the two NN models developed from the case study data. Note that the MSRE has not been calculated in this case because extremely low values of observed stage (often zero) overly skew the results. The M9 model was not tested on days when data were missing; hence, the M14 model was also evaluated on the same (reduced) testing period in order to provide a fair comparison of both models.

Figures 3.4 and 3.5 show the hydrographs of these two models for the test period. Comparing these two hydrographs, it appears that the M9 model is predicting flood peaks more closely than the more complex M14 model. Because

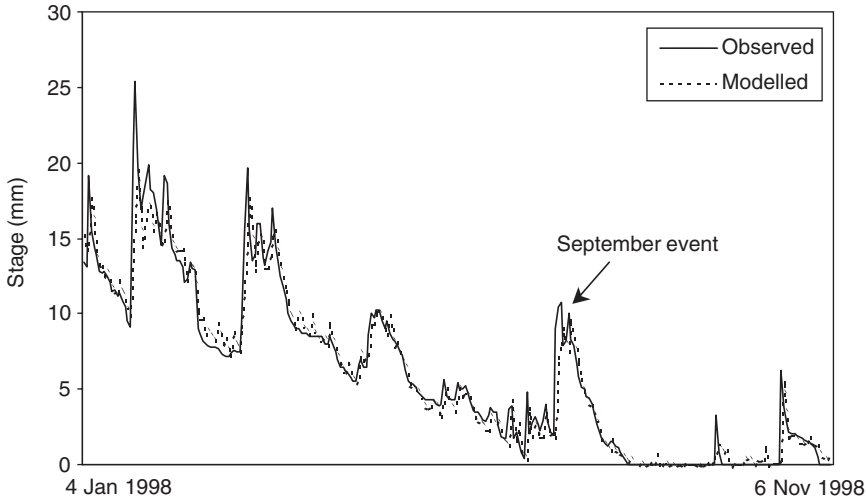


Fig. 3.5. Time series hydrograph of M9 model versus observed during test period.

the M14 model is based on antecedent moving average values, it has tended to underestimate the falling limb of the September event as the predictors from the drier summer period are influencing the model ahead of more recent events. This is not a problem with the M9 model as it is using data from only three days beforehand. These results emphasise the reliance such models have on the data upon which they are developed and tested. In other words, the models become dependent upon the relationships 'identified' during data preprocessing which may lead to the models being 'tuned' to specific seasons and catchment conditions. When the models are then applied to test data covering a range of events, overemphasis on particular situations becomes apparent and a model's ability to generalise appears to be weakened.

4 SUMMARY/CONCLUSIONS

The results of the case study indicate that it is not always necessary to employ complex data preprocessing techniques in order to improve model performance. Indeed, in this example, additional model predictors actually degraded the model according to certain evaluation statistics (i.e. RMSE and CE). This could be due to the fact that certain combinations of predictor variables were not in hydrological terms a sensible selection at particular times of the year – for example, precipitation minus temperature. More complex preprocessing is only helpful when valid hydrological insight is conveyed through data manipulation and/or input selection. The relatively poor performance of the M14 model may be due to poor combinations or non-sensible combinations of predictors rather than the act of preprocessing *per se*. However, this may not always be the case and the

NN developer should always try alternative predictors and different NN models before selecting the 'best' one.

Having identified an accurate NN model, the final stage (not discussed as part of the process here) is the implementation of this model in the field. How such models are integrated within packages, linked to telemetry data and implemented with appropriate graphical user interfaces is beyond the intended scope of this chapter.

In summary, a development process model has been presented, which consists of, at the top level, six simple to follow, sequential stages. However, what cannot be explicitly defined are the subjective rules and the complex decisions that the NN developer must consider at each stage of the process. Due to the complex nature of the problem domain and the extensive range of NN tools that are discussed in the literature, with countless variations and minor modifications, the NN developer is also plagued with a number of options at each stage. For this reason it is recommended that those new to the practice should attempt to implement simple, tried and tested solutions such as a BPNN. Indeed, as shown in the example presented here, it is not necessarily the case that more complex models provide more accurate results.

While the literature accumulates more examples of NN applications in different catchments worldwide (with increasingly minor adjustments to the basic approaches), there has been relatively little attention paid to more fundamental questions. For example, does the internal configuration of a trained NN in anyway represent the physical processes that are at work within a catchment (see Wilby *et al*., 2003)? The answer to this question may lead to improved confidence in network performance and increasing respect for such models across the wider hydrological community. Can NN – once trained – be transferred to other (ungauged) catchments without extensive retraining – perhaps using generalised catchment indices? Finally, could network weights be modified in the light of new information in real time, for example, through an understanding of the physical processes represented by the neurons? In this way, such tools could become much more appealing to environmental organisations attempting to estimate flow in real time with useful lead times and forecasting horizons.

NN models of the river flow process, as far as we are aware, have still not been implemented as full working versions in a real time operational context. Indeed, until the above raised fundamental issues have been properly addressed in a thorough and consistent manner, the use of neural tools for water-related applications will continue to be restricted to a small band of proactive neurohydrologists and computer scientists working in a research environment.

ACKNOWLEDGEMENTS

Grateful thanks are extended to Arnold Publishers (London) for permission to reproduce some of the text and figures from Dawson and Wilby (2001) in this chapter.

REFERENCES

Abrahart, R.J. & Kneale, P.E. 1997. Exploring Neural Network Rainfall-Runoff Modelling. *Proceedings of the 6th British Hydrological Society Symposium, University of Salford, 15–18 September*: 9.35–9.44.

Abrahart, R.J. & See, L. 2000. Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes* 14: 2157–2172.

Abu-Mostafa, Y.S. 1989. The Vapnik-Chervonenkis dimension: information versus complexity in learning. *Neural Computation* 1(3): 312–317.

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automotive Control* AC-19: 716–723.

ASCE. 2000. Artificial neural networks in hydrology. I: Preliminary concepts. *Journal of Hydrologic Engineering* 5(2): 115–123.

Atiya, A., El-Shoura, S. Shaheen, S. & El-Sherif, M. 1996. River flow forecasting using neural networks. *World Congress Neural Networks, San Diego, CA, September*: 461–464.

Atiya, A. El-Shoura, S. Shaheen, S.I. & El-Sherif, M.S. 1999. A comparison between neural-network forecasting techniques – case study: river flow forecasting. *IEEE Transactions on Neural Networks* 10(2): 402–409.

Bishop, C.M. 1995. *Neural Networks for Pattern Recognition*. Oxford, UK: Clarendon Press.

Braddock, R.D., Kremmer, M.L. & Sanzogni, L. 1998. Feed-forward artificial neural network model for forecasting rainfall run-off. *Environmetrics* 9(4): 419–432.

Campolo, M., Andreussi, P. & Soldati, A. 1999. River flood forecasting with a neural network model. *Water Resources Research* 35(4): 1191–1197.

Chang, F. & Hwang, Y. 1999. A self-organization algorithm for real-time flood forecast. *Hydrological Processes* 13: 123–138.

Cheng, X. & Noguchi, M. 1996. Rainfall-runoff modelling by neural network approach. *Proc. Int. Conf. on Water Resour. & Environ. Res.* 2: 143–150.

Colman, A. & Davey, M. 1999. Prediction of summer temperature, rainfall and pressure in Europe from preceding winter north Atlantic ocean temperature. *International Journal of Climatology* 19: 513–536.

Dawson, C.W. 2002. An international comparative study of artificial neural network techniques for river stage forecasting: ANNEXG–artificial neural network experiment group. *Proceedings of British Hydrological Society 8th National Hydrology Symposium*, Birmingham, UK. 75–81.

Dawson, C.W. & Wilby, R. 1998. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal* 43(1): 47–66.

Dawson, C.W. & Wilby, R.L. 2001. Hydrological modelling using artificial neural networks. *Progress in Physical Geography* 25(1): 80–108.

Dawson, C.W., Brown, M. & Wilby, R. 2000. Inductive learning approaches to rainfall-runoff modelling. *International Journal of Neural Systems* 10(1): 43–57.

Fernando, D.A.K. & Jayawardena, A.W. 1998. Runoff forecasting using RBF networks with OLS algorithm. *Journal of Hydrologic Engineering* 3(3): 203–209.

Furundzic, D. 1998. Application of neural networks for time series analysis: rainfall-runoff modeling. *Signal Processing* 64(3): 383–396.

Gorr, W., Nagin, D. & Szcypula, J. 1992. The relevance of artificial neural networks to managerial forecasting; an analysis and empirical study. *Technical Report 93-1, Heinz School of Public Policy Management, Carnegie Mellon University, Pittsburgh, USA*.

Haykin, S. 1999. *Neural networks: a comprehensive foundation* (2nd Ed.). Prentice Hall.

Hsu, K., Gupta, H.V. & Sorooshian, S. 1995. Artificial Neural Network Modeling of the Rainfall-Runoff Process. *Water Resources Research* 31(10): 2517–2530.

Janacek, G. & Swift, L. 1993. *Time series forecasting, simulation, applications.* London: Ellis Horwood.

Jayawardena, A.W., Fernando, D.A.K. & Zhou, M.C. 1997. Comparison of multilayer perceptron and radial basis function networks as tools for flood forecasting. *Destructive Water: Water-Caused Natural Disaster, their Abatement and Control (Proceedings of the Conference at Anaheim, CA, June)*, IAHS Publication Number 239: 173–181.

Karunanithi, N., Grenney, W.J., Whitley, D. & Bovee, K. 1994. Neural networks for river flow prediction. *Journal of Computing in Civil Engineering* 8(12): 201–220.

Lachtermacher, G. & Fuller, J.D. 1994. Backpropagation in hydrological time series forecasting. In K.W. Hipel, *et al*. (Eds.) *Stochastic and Statistical Methods in Hydrology and Environmental Engineering*, Vol 3, 229–242.

Legates, D.R. & McCabe, G.J. 1999. Evaluating the use of 'goodness-of-fit' measures in hydrologic and hydroclimatic model validation. *Water Resources Research* 35: 233–241.

Lorrai, M. & Sechi, G.M. 1995. Neural nets for modelling rainfall-runoff transformations. *Water Resources Management* 9(4): 299–313.

Maier, H.R. & Dandy, G.C. 1996. Neural network models for forecasting multivariate time series. *Neural Network World* 6(5): 747–771.

Maier H.R. & Dandy G.C. 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling and Software* 15(1): 101–123.

Masters, T. 1995. *Neural, novel and hybrid algorithms for time series prediction*. John Wiley.

Minns, A.W. 1996. Extended rainfall-runoff modelling using artificial neural networks. In A. Muller, (Ed.) *Hydroinformatics 96, Proceedings 2nd International Conference on Hydroinformatics, Zurich*, Vol. 1, 207–213.

Minns, A.W. & Hall, M.J. 1996. Artificial Neural Networks as Rainfall-Runoff Models. *Hydrological Sciences Journal* 41(3): 399–417.

Minns, A.W. & Hall, M.J. 1997. Living with the ultimate black box: more on artificial neural networks. *Proceedings of the 6th British Hydrological Society Symposium, University of Salford, 15–18 September*: 9.45–9.49.

Murata, N., Yoshizawa, S. & Amari, S. 1994. Network information criteria – determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks* 5: 865–872.

Nash, J.E. & Sutcliffe, J.V. 1970. River flow forecasting through conceptual models part 1 – a discussion of principles. *Journal of Hydrology* 10: 282–290.

Nowlan, S.J. & Hinton, G.E. 1992. Simplifying neural networks by soft weight-sharing. *Neural Computation* 4: 473–493.

Refenes, A., Burgess, A.N. & Bents, Y. 1997. Neural networks in financial engineering: a study in methodology. *IEEE Transaction on Neural Networks* 8: 1223–1267.

Rissanen, J. 1978. Modeling by short data description. *Automation* 14: 465–471.

Schalkoff, R.J. 1997. *Artificial Neural Network.* New York: McGraw Hill.

Schwarz, G. 1978. Estimating the dimension of a model. *Annals of Statistics* 6: 461–464.

Shamseldin, A.Y. 1997. Application of a neural network technique to rainfall-runoff modelling. *Journal of Hydrology* 199: 272–294.

Sureerattanan, S. & Phien, H.N. 1997. Back-propagation networks for daily streamflow forecasting. *Water Resources Journal* December: 1–7.

Theil, H. 1966. *Applied economic forecasting*. Amsterdam: North-Holland.

Tokar, S.A. & Johnson, P.A. 1999. Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering* 4(3): 232–239.

Watts, G. 1997. Hydrological modelling. Chapter 5 in R.L. Wilby, (Ed.), *Contemporary hydrology: towards holistic environmental science*. Chichester, UK: Wiley.

Wilby, R.L., Abrahart, R.J. & Dawson, C.W. 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrological Sciences Journal* 48(2): 163–181.

Zealand, C.M., Burn, D.H. & Simonovic, S.P. 1999. Short term streamflow forecasting using artificial neural networks. *Journal of Hydrology* 214(1–4): 32–48.

4

Hybrid Neural Network Modelling Solutions

ASAAD Y. SHAMSELDIN
School of Engineering, Department of Civil Engineering, University of Birmingham, UK

ABSTRACT: This chapter deals with hybrid neural network models in which neural networks are integrated with other models to produce better solutions. The different hybrid neural network combinations that are used in the field of river flow forecasting and the underlying theory behind their development are discussed. Hybrid neural network models can be classified into modular or non-modular depending on the manner in which the modelling problem is partitioned and examples taken from both sub-sets are considered. Broader issues related to the practical application of hybrid neural network models are also discussed with particular emphasis on training issues and the assessment of uncertainties.

1 INTRODUCTION

Artificial neural networks (NN) can be applied in the form of stand-alone models, or combined with other tools including other NN, to provide hybrid modelling solutions. Hybrid neural network (HNN) models can be defined as the integration of a number of different models, but with the proviso that one or more of the constituent models is a NN. The underlying principle of the hybrid model is that it exploits the strength of the individual component models in a synergistic manner to produce a better forecasting solution. The hybrid model offers opportunities for integrating conventional hydrological models with those based on artificial intelligence techniques, such as neurocomputing and fuzzy logic. Moreover, in hybrid modelling, conventional models and the artificial intelligence solutions are intended to complement rather than compete with each other.

Hybrid models can be modular or non-modular. In the case of modular hybrid models a complex forecasting problem is divided into a number of simpler

modelling sub-tasks, appropriate models are used to solve these sub-tasks, and their results are integrated to produce the hybrid forecast. In non-modular hybrid modelling, a number of component models are used to provide an independent solution to the exact same problem and the hybrid forecast (temporal) or prediction (non-temporal) is a combination of individual outputs from the individual component models.

The idea of developing HNN models was discussed in Van den Boogaard and Kruisbrink (1996). The authors presented several avenues for the integration. Some of these approaches were discussed in Chapter 2 and include using NN to estimate some of the inputs to the numerical model and the enhancement of the numerical model outputs by using NN to forecast the errors in the numerical model outputs.

This chapter provides several examples that illustrate the potential benefits and use of HNN. The river flow forecasting applications discussed here, cover both non-modular and modular approaches to problems associated with river flow forecasting, and the results of their use on catchments of various sizes and different climatic conditions in different parts of the world. The final section is a discussion on training issues and uncertainties related to modelling with HNN.

2 NON-MODULAR NEURAL NETWORK SOLUTIONS

2.1 *Neuro-combination technique for river flow forecasting*

The essence of a combination non-modular river flow forecasting system is the synchronous use of the discharge forecasts from a number of individual competing river flow forecasting models in order to provide an overall combined forecast which is more accurate and reliable than each individual model output (see Fig. 4.1). The objective of the combination system is not the development of a new single individual model, based on combining the different structural features of various models, but the provision of a shell for blending the forecasts of the different models.

The theoretical background behind this combination process is that each constituent model is regarded as providing an important source of information, which in certain aspects might provide different information from that contained in the other models. Thus, it is logical to assume that the intelligent combination of

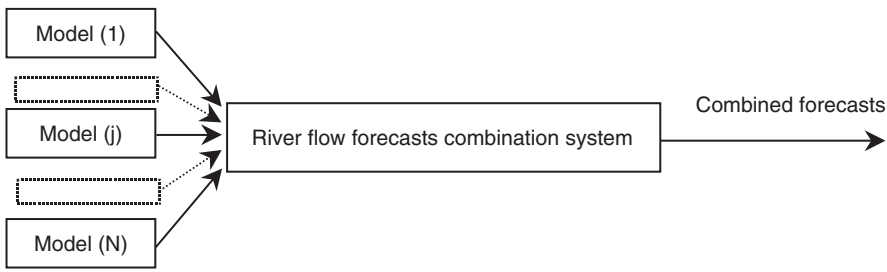


Fig. 4.1. River flow combination system.

information from these different sources can be used to provide more reliable discharge forecasts.

There is also a practical justification for the use of combination river flow forecasting systems since, at the present time, there is no superior individual river flow forecasting model providing discharge forecasts that are better under all circumstances with respect to alternative competing models. This fact has been echoed in a number of inter-comparison studies of river flow forecasting models, which showed that neither simple nor complex models are free from failure (WMO, 1992; Ye *et al.*, 1997; Perrin *et al.*, 2001). These inter-comparison studies have also affirmed the equifinality principle of river flow forecasting models, which is a further illustration of the fact that the construction of a single superior river flow forecasting model is at best improbable. This principle states that different models with identical or different structures and complexities can generate quite equivalent results (Loague & Freeze, 1985; Hughes, 1994; Franchini & Pacciani, 1991; Michaud & Sorooshian, 1994; Ye *et al.*, 1997; Beven & Freer, 2001; Mein & Brown, 1978; Kachroo, 1992; O'Connor, 1995; Lidén & Harlin, 2000).

The recurrent emergence of publications on improvements to existing models and on the development of new models is further evidence that a single superior model has still to be developed. Consequently, there is considerable risk in depending on the results of one river flow model in an operational flood forecasting system, and the failure of a model to produce reliable flood forecasts will weaken the integrity of such tools. Moreover, unreliable river flow forecasts can lead to unreliable flood warnings, which could prove to be expensive in terms of socio-economic or environmental damage and losses (as reported in Chapter 10).

The combination river flow forecasting system addresses such deficiencies in traditional flood forecasting systems, which exclusively depend upon the output from a single model, through the simultaneous application of a suite of alternative individual models. However, the use of combination forecasting, although very well established and regarded as standard practice in diverse fields such as economics, business, statistics and meteorology (Clemen, 1989), has seen limited uptake in the field of river flow forecasting (Shamseldin *et al.*, 2002). There are a small number of publications in this area notwithstanding the fact that the first attempt at river forecasting combinations can be traced back to the pioneering efforts of McLeod *et al.* (1987). This initial work sought to combine monthly river flows obtained from different time series models; such ideas were further developed and investigated to combine daily discharge forecasts from different rainfall-runoff models in Shamseldin (1996), Shamseldin *et al.* (1997), Shamseldin and O'Connor (1999) and Xiong *et al.* (2001), and to combine hourly discharge forecasts from different rainfall-runoff models in See and Openshaw (2000), See and Abrahart (2001) and Abrahart and See (2002).

The recent studies mentioned above, although limited in number, demonstrate the tremendous potential and capabilities of combination methodologies to provide more accurate and more reliable forecasts. The results show that in most cases the combined forecasts were more accurate than the forecasts of the best individual model used in producing the combined solution.

The most popular approaches to produce the combined river flow forecasts that users will find useful include linear-based, fuzzy-based and NN-based methods. To produce a linear combination, the combined forecast is obtained from the forecasts of the individual models using either a simple average (SAM) or a weighted average (WAM) method. The combined forecast obtained by simple averaging is often regarded as a naïve forecast and used as a benchmark against which the forecasts of the more sophisticated combination methods can be compared; persistence is another well-liked universal benchmark.

Fuzzy-based combination methods are based on fuzzy logic and fuzzy systems, which are powerful in the approximation of non-linear time variant functions, and for dealing with imprecise and uncertain knowledge (Goonatilake & Khebbal, 1995; Khan, 1999). Each fuzzy system is a collection of if-then rules that transform a set of inputs into a set of outputs. The rules and sets are developed using the fuzzy set theory of Zadeh (1965), which is regarded as an extension to classical set theory, and the use of such methods for the combination of river flow forecasts is established in See and Openshaw (2000), Xiong *et al.* (2001) and Abrahart and See (2002).

In the NN combination method, as the name indicates, NN are used to build combination river flow forecasting systems. This offers a novel approach that differs from other hydrological applications. In combination systems, NN work synergistically with the constituent models to produce better river flow forecasts, while in other hydrological applications NN contend with traditional hydrological models. Shamseldin *et al.* (1997) advocated the use of NN as a complex method for the combination of river flow forecasts obtained from different rainfall-runoff models. NN combinations are in most cases better than those of WAM and SAM (Shamseldin *et al.*, 1997; Abrahart & See, 2002). Xiong *et al.* (2001) also found that neuro-combination forecasting was often more accurate and more reliable than linear and fuzzy-based combination systems, although Abrahart and See (2002) report that neuro-combinations perform well on certain categories of error, but do less well on others, given shorter time steps and forecasting horizons. NN combinations of differenced data were also observed to provide the best solution for a stable regime – whereas a fuzzified probabilistic solution produced superior results in a more volatile environment.

NN, as described in Chapter 2, can be classified into different types depending upon the arrangement of their internal neurons and the pattern of the interconnections between them. The neurons are the computational processing elements of the network, which operate on the external inputs to produce the final network outputs, and, in the context of river flow combination, the external inputs at each time step are the forecasts of the individual models while the network output is the combined forecast.

There are various different types of NN that can be used in a neuro-combination river flow forecasting system. However, the common multi-layer perceptron (MLP) has to date been the only NN type used in river flow forecast combination systems, in part due to the dominance of this solution in all other forms of river flow forecasting applications (Dawson & Wilby, 2001; Maier & Dandy, 2000, Chapter 2).

Each neuron can have multiple inputs but produces a single output, which becomes an input to other neurons in the next layer. The process of input-output transformation is governed by a mathematical function known as the neuron transfer function. A linear transfer function is used in the input layer while a non-linear transfer function is used in the hidden and output layers. The logistic function is the most common non-linear transfer function. Shamseldin *et al.* (2002) examined the significance of using different non-linear transfer functions for the hidden and output layers in an MLP when used in the context of the river flow forecast combination method. Five neuron transfer functions were used in the investigation: the logistic function; the bipolar function; the hyperbolic tangent function; the arctan function; and the scaled arctan function. The results of the investigation showed that the logistic function generated the best results while the arctan function often produced the worst results.

The field of combination river flow systems is a new and challenging area of research. There is tremendous scope for investigation, e.g. to test if the use of different types of NN, other than MLP, will lead to further improvements in the performance of the neuro-combination method. Future applications of neuro-combination river-flow forecasting systems will also need to be tested over a wide range of catchments with different characteristics to develop general guidelines about their use in different catchment types with respect to area, land use, soil type and relief. The complexity of the combination system will also be increased, commensurate with an increase in the number of constituent models, and there is a need to determine the optimum number of the constituent models for each combination system. However, the increase in complexity does not always guarantee a significant improvement in the system performance, and it is necessary to provide guidance on the optimal number of models beyond which the performance of the system does not substantially improve. This optimal number would therefore maintain a balance between the complexity of the system and the performance of the system, which is a recurring theme in the field of applied hydrological science, and which is of particular significance with respect to neurocomputation where there is a strong danger of over-modelling.

The results obtained so far with the combination river flow forecasting system are very encouraging in terms of forecast reliability and accuracy. The implication of these results is that users of modern flood forecasting systems, who use a suite of river flow forecasting models, should seriously consider switching to the synchronous combination of alternative forecasts. Indeed, the implementation of combination procedures in flood forecasting centres that use an existing suite of forecasting models would be a simple and inexpensive operation.

3 MODULAR HYBRID NEURAL NETWORK SOLUTIONS

3.1 *Neuro-updating technique for river flow forecasting*

In real-time river flow forecasting systems, the substantive simulation rainfall-runoff model operates on-line on the basis of the latest available data, with auxiliary updating procedures being used to compensate for the errors that arise between the simulated and observed discharge hydrographs. These errors are

due to numerous factors, which include inadequacy of the model structure, poor estimation of the model parameters and inherent errors (systematic or random) in both rainfall and discharge data.

There are different categories of updating procedure but there is no common or universal agreement on the best method to be used in river flow forecasting (WMO, 1992; Refsgaard, 1997). The common task of different updating procedures is to provide the substantive simulation model with feedback information based on the latest river flow data observed prior to the time of issuing the forecast. This feedback information is then used to offset a significant proportion of the errors that occur between the observed and the simulated discharge hydrographs, thereby refining the discharge forecasts. In general, the use of an updating procedure considerably improves the forecasting accuracy for short-term forecasting. Such considerable improvements are essential for the reliable management of the routine operation of real-time river flow forecasting systems. Such systems provide reliable information that can be used to help mitigate the impacts of floods for real-time operation.

The substantive rainfall-runoff model, coupled to its discharge updating procedure, is termed a real-time river flow forecasting model (Becker & Serban, 1990). The estimated discharges produced by the substantive simulation model prior to the application of the updating procedures are known as simulation-mode (or design mode) forecasts; the estimated discharges obtained after applying the updating procedures are known as updating-mode forecasts. The real-time forecasting model is an example of a modular hybrid model in which the real-time forecasting problem is divided into two modelling sub-problems: (i) simulation and (ii) updating. NN can be used for modelling either or both of these two sub-problems.

Moore (1986) classified real-time river flow updating procedures into four types depending on the nature of the substantive model variables to be modified. These variables are the input variables to the model, the water content of the various storage elements in the model, the internal parameters of the model, and the output variables from the model. The four possible updating procedure types are thus: (i) input updating procedures; (ii) storage content updating procedures; (iii) parameter updating procedures; and (iv) output updating procedures. There is no restriction on the number or type of procedures that can be applied to updating and more than one type of updating procedure can be used to produce a set of output forecasts (Becker & Serban, 1990; Serban & Askew, 1991; WMO, 1992). NN could also, therefore, be used to update one or more such variables, either alone, or in a combination.

Input updating is based on adjusting the hydro-meteorological input variables to the model, e.g. rainfall or upstream discharge. The input adjustment is made such that re-running the substantive river flow forecasting model with the modified inputs produces near-correct discharge forecasts. The input updating is in essence an inverse modelling problem in which the model inputs are expressed as an explicit or implicit function of the model outputs. However, since most river flow forecasting models have complex structures, it is not always possible to derive the implicit inverse function. For this reason, input

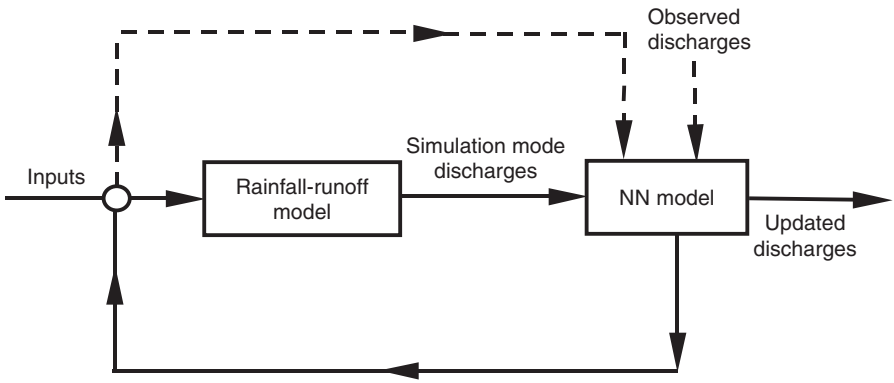


Fig. 4.2. The NN input updating procedure (adapted from Refsgaard, 1997).

updating is normally performed on a trial-and-error basis. NN could also be trained to explicitly approximate the inverse function. Figure 4.2 shows a schematic diagram of how NN can be used for an input updating procedure in conjunction with a substantive rainfall-runoff model.

The operation of storage content updating depends on real-time or run-time recursive adjustment of the water content in the various storage elements to enhance the forecasts. In a quasi-physical conceptual rainfall-runoff model, such enhanced forecasts could be achieved by modifying the water content of the water balance part and/or of routing reservoir elements. The raison d'être of storage updating is that errors in the physical input variables to the model accumulate and appear as corresponding errors in the water content of the storage elements. Therefore, if these water contents are adjusted in accordance with identified discrepancies between simulated and observed discharge, then better discharge forecasts can be obtained (Moore, 1986). The storage contents are usually adjusted using two methods: (i) the Kalman filter which has a strong hydrological tradition and (ii) empirical state updating procedures. In both methods the adjusted storage content is the sum of the current storage content value and the model error multiplied by a gain coefficient. In the case of the Kalman filter, the gain coefficient is statistically defined, while in the case of the empirical adjusting procedures, the empirical gain coefficient is found from off-line optimisation. NN can also be used for storage updating, as such tools can be trained to produce either the gain coefficient, or the final adjusted content itself.

The parameter updating procedure involves real-time or run-time adjustments to parameter values of the model using the recursive least squares method and the Kalman filter (O'Connell & Clarke, 1981). This updating procedure is con-sidered less attractive than the other updating procedure categories and the degree of parameter variation normally demonstrates the extent of structural unsuitability of the substantive model (Moore, 1986). It is also hard to substantiate that such considerable variations in parameter values can occur in such a short time as the observation interval. NN-based parameter updating

procedures can be developed in a similar fashion to the NN storage updating procedures to produce the final adjusted parameters.

In the output updating procedures, the updated discharge forecasts are obtained from external adjustment of the simulation mode discharges. This external adjustment is performed without interfering with the internal operation of the simulation model in the sense that the updates do not alter the model parameters. Furthermore, there is no need to re-run the substantive simulation model to obtain the updated forecasts. For the aforementioned reasons, in comparison to other types of updating procedure, the application of output-updating procedures is regarded as a simple and straightforward operation.

Output updating procedures can be classified as direct or indirect. The indirect output updating procedure is based on forecasting the errors in simulation mode discharge forecasts. The final updated discharge forecast is then the sum of non-updated (simulation-mode) discharge values and the corresponding error forecast. The forecast updating via error predictions is perhaps the most popular updating procedure, which has been extensively used in applied hydrology in conjunction with numerous different rainfall-runoff models (Serban & Askew, 1991). Such procedures, which are normally used for indirect updating, are the uni-variate linear stationary time series solutions of Box and Jenkins (1976), e.g. the autoregressive (AR) models and autoregressive moving average (ARMA) models. These time series models generally exploit the time persistence structure of the error series to forecast future errors. NN can thus be implemented as a more general non-linear form of these time series models, which are used for error forecasting. For example, Xiong and O'Connor (2002) developed an NN error-forecast updating procedure based on the structure of a multi-layer feed forward neural network (MLFN). The external inputs to the NN, for one-step ahead error forecasting, are the simulation mode errors of the substantive model up to the time of issuing the forecasts. Figure 4.3 shows a schematic diagram of the NN updating procedure operating in parallel in conjunction with a substantive rainfall-runoff model. Babovic *et al.* (2001) used the NN error updating procedure for updating the forecast of a hydrodynamic model. The authors concluded that the NN error updating procedure 'provides very good forecasting skills that can be extended over a forecasting horizon of a significant length' (p. 181). Xiong and O'Connor (2002) compare the NN error updating procedure with the standard AR error updating procedure, using the simulation mode discharges of a conceptual rainfall-runoff model on eleven selected catchments. The results of their comparison illustrated that the complex NN error updating procedure offers no significant merits in terms of enhancing the real-time flow forecast performance over the simple AR error updating procedure.

The direct procedure is in most applications based on the input-output structure of the linear and non-linear Auto-Regressive eXogenous-input Model (ARXM), which encompasses a self-correcting mechanism that can be exploited for direct updating of the forecasts of the substantive model. Some of the indirect output can be regarded as a special case of the direct output updating procedure. For example, Shamseldin and O'Connor (1999) show that the AR updating procedure is a limiting case of the linear ARXM updating procedure.

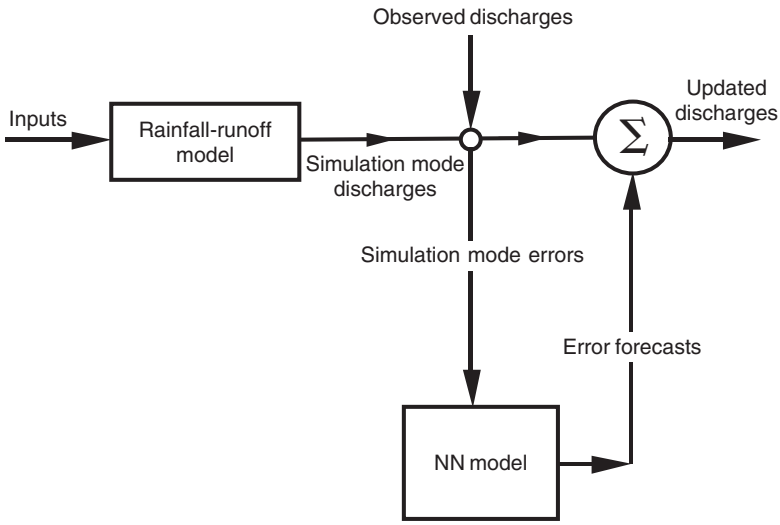


Fig. 4.3. The NN error updating procedure.

In general, the linear ARXM updating procedure can lead to better real-time forecasting results than those produced from AR and ARMA error updating procedures (Shamseldin, 1996).

Traditionally, the linear ARXM is used as a river flow forecasting model in its own right where the exogenous inputs to the model are the conventional physical inputs such as rainfall and the upstream inflow hydrograph. In the case of direct output updating, the autoregressive component of the ARXM provides a feedback information mechanism in the form of the latest observed discharges. In the direct output updating procedures, the exogenous inputs to the ARXM are the simulation mode discharge forecasts of the substantive model. These exogenous inputs are used in combination with the latest observed discharges prior to the time of issuing the forecast to produce the updated discharge forecasts as direct outputs of the linear ARXM. The first apparent application of an implicit use of the linear ARXM in the context of output updating was in the unpublished work of Peetanonchai (1995). Further investigation into the use of the ARXM in this context was later reported in Abdelrahman (1995), Shamseldin (1996), Suebjakla (1996), Shamseldin and O'Connor (1999) and Shamseldin and O'Connor (2001).

Shamseldin and O'Connor (2001) developed a Non-linear Auto-Regressive eXogenous-input Model (NARXM), which is based on the structure of the MLP. This NARXM updating mechanism presents new avenues for the integration of substantive rainfall-runoff models with NN. The overall operation of a NARXM-NN output updating procedure is shown in Figure 4.4. For one-step ahead forecasting, the external inputs to the NN are simulation-mode discharge forecasts, up to the time of issuing the forecasts, and the current or the latest observed discharges, prior to the time of issuing the forecasts. The NN output, after suitable transformation, constitutes the updated discharge forecast. The performance of

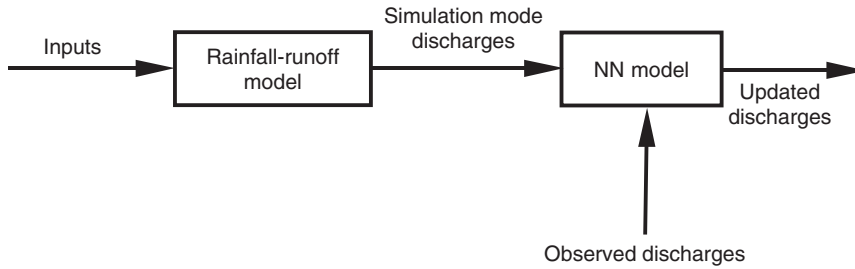


Fig. 4.4. The NARXM-NN output updating procedure (adapted from Shamseldin *et al.*, 2002).

the NARXM-NN output updating procedure was tested using the daily discharge forecasts of a conceptual rainfall-runoff model for five catchments that had different climatic conditions. The results of the NARXM-NN output updating procedure were compared with a linear ARXM updating procedure. The results of this comparison indicate that the NARXM generally performed better than the linear ARXM. The results also show that the highest improvement in performance was obtained for semi-arid catchments where the time persistence structure (i.e. autocorrelation) of the simulation mode error time series was very weak.

3.2 *Modular neural networks*

Modular neural networks MNN are based on the concept of *divide and conquer*. According to this concept, a complex non-linear computational problem is divided into a number of simple computational sub-tasks thus splitting the input space into regions, with the solution for one or more specific sub-tasks or modules being assigned to a different NN. It is assumed that different constituent solutions will work best in different regions of the solution space (Haykin, 1999). Modular solutions will also have an integrating unit. The main function of the integrating unit is to combine the results of the constituent modules. MNN, undoubtedly, will be very complex in contrast to standard NN. MNN will also have a large number of parameters (i.e. they are not parsimonious) and can therefore be quite difficult to train. There are several publications that deal with the use of the MNN in rainfall-runoff studies (e.g. See & Openshaw, 1999; Zhang & Govindaraju, 2000; Hu *et al.*, 2001; Hsu *et al.*, 2002).

See and Openshaw (1999) developed a MNN model for river flow forecasting on the River Ouse in the UK. In this case the 'divide and conquer' element involved breaking up the hydrographs into their component parts: rising flow limb, peak flow, falling limb and low flows. Independent NN models were then produced for each hydrograph section and their outputs reintegrated. The integration of the different MNN modules was achieved using a sophisticated fuzzy logic rule-based model. It was concluded that the developed MNN model 'may provide a well performing, low-cost solution, which may be readily integrated into existing operational flood forecasting and warning systems' (See & Openshaw, 1999 p. 763).

Zhang and Govindaraju (2000) used MNN to perform rainfall-runoff transformations for three medium sized catchments in the USA. There were three NN modules in each MNN to simulate low, medium and high flow events. The justification for using three modules is that, in the chosen catchments, different runoff generating mechanisms are dominated by different physical processes depending on the magnitude of each runoff event. The MNN used a linear function to combine the three individual outputs and the results were compared against those obtained using a standard MLP. This comparison showed that, while MLP are capable of adequately simulating the average events, such tools fail to capture extreme events, whereas MNN appear to be more successful at simulating such items. Similarly, Hu *et al.* (2001) found that MNN were more successful than MLP in reproducing low flow events.

Hsu *et al.* (2002) also developed an MNN in which the input domain is partitioned into different regions using a Self-Organising Feature Map (SOFM) (Kohonen, 1984), which is a specific type of NN – for more details see Chapter 2. These regions correspond to different hydrological situations such as base flow, increasing rainfall, peaking hydrograph, etc. The output is then calculated using a set of piecewise linear regression equations, one for each node in the SOFM, which relates the model inputs to an estimation of flow at the next time step. The linear regression coefficients were calculated using a least square error solution. Hsu *et al.* (2002) found that the performance of the MNN was better than those of other commonly used river flow forecasting models such as MLP, ARMX and conceptual rainfall-runoff models.

4 BROADER ISSUES

This section of the chapter discusses issues of training and uncertainties in the development of modelling solutions based on HNN. It outlines practical difficulties associated with these issues and provides an outline of several methods that can be used for training and the quantification of uncertainty. The discussion is of a general nature since most of the items considered in this section are also valid for both traditional hydrological models and NN.

4.1 *Training of hybrid solutions*

The use of standard and hybrid NN models for river flow forecasting in a specific catchment requires the estimation of numerical values for the parameters that control the overall operation of the model. This is achieved through training. For MLP, the connection weights assigned to the connection paths that link the neurons, together with the neuron threshold values, constitute the parameters of the network. Hence, the objective of the training process is to find a set of parameter values for a particular catchment that can provide the best possible fit between the simulated and the observed outputs. Thus, the training process is essentially equivalent to the traditional process of model calibration.

For hydrological applications of NN and HNN models, the issue of their training has received less attention, to the extent that most publications do not

quote the number of model parameters. The calibration process has a considerable role in testing the appropriateness of the model for a specific catchment. The use of an inadequate set of model parameter values would generally produce poor predictions and, consequently, this may lead the model developer to abandon the use of a model when, properly optimised, it might be quite adequate.

Estimation of the parameters entails the specification of an estimation criterion to quantify how good the simulated outputs are in replicating the actual observed outputs (i.e. a goodness-of-fit criterion). The estimation criterion is also known as the objective function and could be a single criterion or a combination or weighting of several criteria for goodness-of-fit. Several forms of objective function have been used in a hydrological context (Sorooshian & Gupta, 1995). However, the most widely used objective function for the estimation of parameters in the context of NN and hydrological models is the least squares criterion. The least squares objective function is the sum of the squares of deviations between the estimated model outputs and the actual observed outputs. The choice of a particular objective function will, in most cases, mainly depend on the purpose of the model and on what results the model is expected to produce. As an example, if the emphasis is on the estimation of the flood peaks, then a greater weight to peaks could be incorporated in the objective function to reflect this emphasis.

The training of a hybrid model involves the calibration of both the constituent models and the integration unit. In most cases, the constituent models and the integration unit are developed as individual and unrelated solutions (Shamseldin *et al.*, 1997). However, this separate development may not yield the best results. On the other hand, combined calibration of both constituent models and the integration unit will most often be a problematic undertaking that involves the estimation of a large number of parameters.

In most of the cases, the constituent models and the integration unit in the hybrid model are non-linear, and their calibration or training with observed input-output data involves the optimisation of a non-linear objective function. There are several different deterministic local search methods and global stochastic search methods that can be used to solve this non-linear optimisation problem.

'Local search methods are defined as those that are designed to efficiently find the minimum of unimodal functions – functions for which any strategy that seeks to continuously proceed downhill (a direction of improving function value) must eventually arrive at the location of the function minimum, irrespective of where in the parameters space the search procedure is started' (Sorooshian & Gupta, 1995, pp. 31–32). Depending on whether or not the derivatives of the objective function are used, the local search methods can be further classified as direct search methods or gradient search methods. The gradient search methods (e.g. the backpropagation and conjugate gradient algorithms) are the most widely used methods for training NN. Direct search methods such as the Simplex Method (Nelder & Mead, 1965) and the Rosen-brock Method (Rosenbrock, 1960), which have been widely used for decades in calibrating rainfall-runoff models, can also in principle be used for calibrating HNN.

Stochastic global optimisation methods offer efficient tools that can be used to discover a global minimum, and adopt probabilistic rules to search the response surface of the objective function. These methods will guarantee the convergence of each solution to the global optimum in a probabilistic sense (Duan *et al.*, 1992). Examples of stochastic global optimisation methods are genetic algorithms and simulated annealing algorithms. There are also global optimisation methods, which are developed based on the integration of concepts drawn from local search (deterministic) and stochastic optimisation methods, which have been found to be successful in calibrating rainfall-runoff models. For example, Duan *et al.* (1992) developed the complex shuffled evolution method to combine the strength of the simplex method with the concept of information sharing and concepts drawn from evolution-based biological theories that are similar to those used in genetic algorithms.

The parameters of the hybrid model can also be estimated by a sequential optimisation procedure. This sequential optimisation involves the successive use of a stochastic global search method and a local search method. The final optimised parameters of the stochastic global search method are used to produce the initial starting values for a local search method (Shamseldin *et al.*, 2002). The sequential optimisation procedure combines the strength of the stochastic method in locating the global solution with the efficient convergence of a local search technique, which is used to fine-tune the results of the stochastic method. Wang (1991) found that further sequential tuning of the final optimised parameters from a genetic algorithm, using the simplex method, is a robust and efficient method for the calibration of conceptual models.

The NN and HNN models are in most cases too complex, having a large number of parameters (i.e. not parsimonious), in contrast with traditional river flow forecasting models. For example, a simple MLP rainfall-runoff model can have more than a dozen or so parameters. Such over complex solutions can cause numerous difficulties in the process of model identification and in finding the true optimum set of parameters. The extent of these difficulties is much greater for NN and HNN models, compared to those reported for traditional conceptual rainfall-runoff models, which possess far less parameters. Difficulties in the calibration of traditional models can be attributed to (Ibbitt & O'Donnell, 1974; Johnston & Pilgrim, 1976; Moore & Clarke, 1980): i) parameter interdependence, in which numerous parameters produce the same optimum value from the objective function, i.e. equifinality ii) discontinuities in the objective function causing numerical problems for gradient-based optimisation methods iii) scaling of the parameters resulting in narrow elongated valleys in objective function response surfaces, along which the search progress is generally very slow iv) indifference of the objective function to the parameter values causing the optimisation algorithm to end in a premature fashion v) local minima causing the search algorithm to be terminated at a point at which the objective function is lower than all surrounding points in a local neighbourhood, but at a higher value than a point in another region of the objective response surface, which is the true minimum.

4.2 Uncertainties of hybrid solutions

Most reported NN hydrological forecasting applications, including HNN applications, do not provide a measure of forecast uncertainties (Maier & Dandy, 2000). The uncertainty analysis on the forecast of each HNN and NN model, however, is quite important given the internal complexities involved. It will be interesting to see whether or not the use of such models can lead to a reduction in forecast uncertainties when compared to traditional hydrological models.

The uncertainties in the forecasts of river flow models can be attributed to a number of different factors such as model structure, poor estimation of the model parameters and errors in the input data (Lei & Schilling, 1996). The estimation of such uncertainties is important because a single deterministic model prediction will be always be wrong (Beven *et al.*, 2001). The analysis of river flow forecast uncertainties will enhance the utility of the forecasting system as it will enable the estimation of potential flooding probabilities, which has numerous benefits, such as to enable the setting of risk-based criteria for flood warning codes (e.g. flooding possible, flooding expected, severe flooding expected and all clear) and emergency response plans (Krzysztofowicz, 2001).

There are various simple and complex methods, such as the Mean Value First Order Method (MVFOM) and Monte Carlo simulation methods, which can be used to perform uncertainty analysis (Melching, 1992; Yu *et al.*, 2001). MVFOM is the most common method as it is simpler to apply than the other methods of uncertainty analysis. MVFOM is based on a linearisation of the model output equation around the mean value or other convenient central value, or around other basic variables such as parameters and inputs (Melching, 1992; Lei & Schilling, 1996). This method yields information about the expected value and the variance of the forecast as well as the contribution of each variable to the variance (output uncertainties) of the forecast. However, its application requires the specification of the mean and the variance of the basic variables.

The Monte Carlo method involves randomly generating a large number of basic variable sets from their corresponding probability distributions. The model is run repeatedly with a new randomly generated set of basic variables in each run. In this way, a large number of model forecast samples are generated and these samples are used to calculate the mean and the variance of the forecast. Thus, the operation of this method requires knowledge of the probability distribution of the basic variables. The results of the Monte Carlo method are often used as a benchmark against which the results from other uncertainty analysis methods can be compared.

As noted above, the use of uncertainty methods such as MVFOM and the Monte Carlo method requires the specification of the statistical moments (i.e. mean and variance) and the probability distributions of the basic variables. In the case of model parameters, the statistical moments can be estimated by repeated model calibration to different time periods. The nature of the probability distributions can also be inferred from the results of the repeated model calibration to different time periods (Melching, 1992). From the results of the uncertainty analysis, a model reliability index can be calculated, and used to compare the results of different models. The results of this approach, together

with an appropriate assumption about the distribution of the model forecasts, can also be used to construct a set of forecast confidence intervals.

5 CONCLUSIONS

In this chapter, the use of hybrid neural network (HNN) models in the context of river flow forecasting is examined. HNN models offer frameworks for the integration of traditional hydrological models with modern soft computing models, such as NN models and fuzzy logic models. They also offer good opportunities for enhancing the forecast accuracy. The literature review presented in the chapter suggests that the use of HNN models is not wide spread. However, the number of publications dealing with their applications is increasing.

The literature review reveals that the issue of training HNN models is often overlooked. The primary focus of most HNN modelling studies has been to demonstrate the potential capabilities of HNN models for improving the output forecasts, compared to stand-alone models. This chapter has discussed the various methods that can be used for training HNN models. It is envisaged that many of the long-reported problems faced during the calibration of traditional rainfall-runoff models could be exacerbated in the calibration of HNN rainfall runoff models due to their inherent complexities. There is a need to determine whether or not the current procedure for calibrating HNN, which involves the separate calibration of the individual constituent units, is the best calibration solution.

The forecasts of HNN and NN models are not at present usually issued with measures of forecast uncertainties. This is not acceptable as a practice; rectifying this omission is thus a research priority that is needed to give confidence to future modellers. This omission could be due in part to the structure of existing NN software source codes taken from 'off-the-shelf' packages, which does not allow room for further development as source code is often concealed from the user. Using off-the-shelf code is sensible but the adoption of open source code would permit further refinement or the addition of user-constructed modules, which can systematically examine the output quality. Many of the methods used for uncertainty analysis in traditional hydrological models can and should be used in conjunction with HNN models to aid the critical evaluation of the results, and to provide better modelling developments.

REFERENCES

Abdelrahman, E.A. 1995. Real-time streamflow forecasting for single-input single-output systems. MSc. thesis, National University of Ireland, Galway. (Unpublished).

Abrahart, R.J. & See, L. 2002. Multi-model data fusion for river flow forecasting: an evaluation of six alternative methods based on two contrasting catchments. *Hydrology and Earth System Sciences* 6(4): 655–670.

Babovic, V., Caňizares, R., Jensen, H.R. & Klinting, A. 2001. Neural networks as routine for error updating of numerical models. *Journal of Hydraulic Engineering* 127(3): 181–193.

Becker, A. & Serban, P. 1990. *Hydrological Models for Water-resources System Design and Operation*. Operational Hydrology. Report. 34, WMO No. 740, Geneva.

Beven, K. & Freer, J. 2001. Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology. *Journal of Hydrology* 249(1–4): 11–29.

Beven, K., Romanowicz, R. & Hankin, B. 2001. Mapping the probability of flood inundation (even in real time). In M. Lees & P. Walsh (Eds.), *Flood Forecasting: What does Current Research Offer the Practitioner?* Occasional paper no. 12, British Hydrological Society.

Box, G.E.P. & Jenkins G.M. 1976. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco.

Clemen, R.T. 1989. Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting* 5: 559–583.

Dawson, C.W. & Wilby, R.L. 2001. Hydrological modelling using artificial neural networks. *Progress in Physical Geography* 25(1): 80–108.

Duan, Q., Sorooshian, S. & Gupta, V.K. 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research* 28(4): 1015–1031.

Franchini, M. & Pacciani, M. 1991. Comparative analysis of several conceptual rainfall-runoff models. *Journal of Hydrology* 122: 161–219.

Goonatilake, S. & Khebbal, S. 1995. *Intelligent Hybrid Systems*. John Wiley & Sons Ltd., UK.

Haykin, S. 1999. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, USA.

Hsu, K., Gupta, H.V., Gao, X., Sorooshian, S. & Imam, B. 2002. Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis. *Water Resources Research* 38(12): 1–17.

Hu T.S., Lam, K.C. & Ng, T.G. 2001. River flow forecasting with a range-dependent network. *Hydrological Sciences Journal* 46(5): 729–745.

Hughes, D.A. 1994. Soil Moisture and runoff simulation using four catchment rainfall-runoff models. *Journal of Hydrology* 158: 381–404.

Ibbitt, R.P. & O'Donnell, T. 1971. Fitting methods for conceptual models. *J. Hydr., Div., Proc. ASCE* 97(HY9): 1310–1342.

Johnston, P.R. & Pilgrim, D.H. 1976. Parameter optimization of watershed models. *Water Resources Research* 9(2): 447–486.

Kachroo, R.K. 1992. River flow forecasting. Part 5. Applications of a conceptual model. *Journal of Hydrology* 133: 141–178.

Khan, E. 1999. Neural fuzzy based intelligent systems. In L.C. Jain & N.M. Martin (Eds.), *Fusion of Neural Networks, Fuzzy Sets and Genetic Algorithm Industrial Applications*. CRC Press.

Kohonen, T. 1984. *Self-Organization and Associative Memory*. Berlin: Springer.

Krzysztofowicz, R. 2001. The case of probabilistic forecasting in hydrology. *Journal of Hydrology* 249: 2–9.

Lei, J.H. & Schilling, W. 1996. Preliminary uncertainty analysis – a prerequisite for assessing the predictive uncertainty of hydrologic models. *Water Science and Technology* 33: 79–90.

Lidén, R. & Harlin J. 2000. Analysis of conceptual rainfall-runoff modelling. *Journal of Hydrology* 238(3–4): 231–247.

Loague, K.M. & Freeze, R.A. 1985. A comparison of rainfall-runoff modelling techniques on small upland catchments. *Water Resources Research* 21(2): 229–249.

Maier, H.R. & Dandy, G.C. 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling and Software* 15(1): 101–124.

McLeod, A.I., Noakes, D.J., Hipel, K.W. & Thompstone, R.M. 1987. Combining hydrologic forecasts. *Water Resources Planning and Management* 113(1): 29–41.

Mein, R.G. & Brown, B.M. 1978. Sensitivity of optimized parameters in watershed models. *Water Resources Research* 14(2): 229–303.

Melching, C.S. 1992. An improved first order reliability approach for assessing the uncertainties in hydrologic modelling. *Journal of Hydrology* 132: 157–177.

Michaud, J. & Sorooshian, S. 1994. Comparison of simple versus complex distributed models on mid-sized semiarid watersheds. *Water Resources Research* 30(3): 539–605.

Moore R.J. & Clarke, R.T. 1980. A distribution function approach to rainfall-runoff modelling. *Water Resources Research* 17(5): 1367–1382.

Moore, R.J. 1986. *Advances in real-time forecasting practice*. Invited paper, Symposium on Flood Warning Systems, Winter Meeting of the River Engineering Section, The Institution of Water Engineers and Scientists, 23 pp.

Nelder, J.A. & Mead R. 1965. A simplex method for function optimisation. *Computer Journal* 7: 308–313.

O'Connell, P.E. & Clarke, R.T. 1981. Adaptive hydrological forecasting – a review. *Hydrological Science Bulletin* 26(2): 179–205.

O'Connor, K.M. 1995. *River flow forecasting: The Galway experience*. Invited Lecture, Symposium on River Flow Forecasting and Disaster Relief, Haikou City, China, Nov. 20–25, 1995.

Peetanonchai, B. 1995. *A modification of the AR model for updating of model forecasts*. Final Report for the 6th International advanced Course/Workshop on River Flow Forecasting, 24–30th June, 1995, Department of Engineering Hydrology, University College Galway, Ireland.

Perrin, C., Michel, C. & Andréassian, V. 2001. Does a large number of parameters enhance model performance? Comparative assessment of common catchment model structures on 429 catchments. *Journal of Hydrology* 242(3–4): 275–301.

Refsgaard, J.C. 1997. Validation and intercomparsion of different updating procedures for real-time forecasting. *Nordic Hydrology* 28: 65–84.

Rosenbrock, H.H. 1960. An automatic method for finding the greatest or least value of a function. *Computer Journal* 7: 175–184.

See, L. & Abrahart, R.J. 2001. Multi-model data fusion for hydrological forecasting. *Computers and Geosciences* 27(8): 987–994.

See, L. & Openshaw, S. 1999. Applying soft computing approaches to river level forecasting. *Hydrological Sciences Journal* 44(5): 763–778.

See, L. & Openshaw, S. 2000. A hybrid multi-model approach to river level forecasting. *Hydrological Sciences Journal* 45(4): 523–536.

Serban, P. & Askew, A.J. 1991. Hydrological forecasting and updating procedures. *IAHS Publication. No.* 201: 357–369.

Shamseldin, A.Y. 1996. Fundamental studies in rainfall-runoff modelling. Ph.D thesis, National University of Ireland, Galway. (Unpublished).

Shamseldin, A.Y., O'Connor, K.M. & Liang, G.C. 1997. Methods for combining the outputs of different rainfall-runoff models. *Journal of Hydrology* 197: 203–229.

Shamseldin, A.Y. & O'Connor, K.M. 1999. A real-time combination method for the outputs of different rainfall-runoff models. *Hydrological Sciences Journal* 44(6): 895–912.

Shamseldin, A.Y. & O'Connor, K.M. 2001. An non-linear neural network technique for updating river flow forecasts. *Hydrology and Earth System Sciences* 5(4): 577–597.

Shamseldin, A.Y., Ahmed, E.N. & O'Connor, K.M. 2002. Comparison of different forms of the Multi-layer Feed-Forward Neural Network Method used for River Flow Forecast Combination. *Hydrology and Earth System Sciences* 6(4): 671–684.

Sorooshian, S. & Gupta, V.K. 1995. Chapter 2: Model calibration. In V.P. Singh (Ed.), *Computer Models of Watershed Hydrology* (pp. 23–68). Fort Collins, CO: Water Resources Publications.

Suebjakla, W. 1996. Modification of the Autoregressive (AR) error updating model for real-time streamflow forecasting. M.Sc. thesis, National University of Ireland, Galway. (Unpublished).

Van den Boogaard. H.F.P. & A.C.H. Kruisbrink 1996. Hybrid modelling by integrating neural networks and numerical models. In Müller (Ed.), *Hydroinformatics '96* (pp. 471–477). Rotterdam: Balkema.

Wang, Q.J. 1991. The genetic algorithm and its application to calibrating conceptual rainfall-runoff models. *Water Resources Research* 27(9): 2467–2471.

World Meteorological Organization (WMO). 1992. *Simulated real-time intercomparison of hydrological models*. Oper Hydrol. Rep. 38, WMO No. 779, Geneva.

Xiong, L. & O'Connor, K.M. 2002. Comparison of four updating models for real-time river flow forecasting. *Hydrological Sciences Journal* 47(4): 621–639.

Xiong, L.H., Shamseldin A.Y. & O'Connor, K.M. 2001. A non-linear combination of the forecasts of rainfall-runoff models by the first-order Takagi-Sugeno fuzzy system. *Journal of Hydrology* 245(1–4): 196–217.

Ye, W., Bates, B.C., Viney, R., Sivapalan, M. & Jakeman, A.J. 1997. Performance of conceptual rainfall-runoff models in low-yielding ephemeral catchments. *Water Resources Research* 33(1): 153–166.

Yu, P.S., Yang, T.C. & Chen, S.J. 2001. Comparison of uncertainty analysis methods for a distributed rainfall-runoff model. *Journal of Hydrology* 244: 43–59.

Zadeh, L.A. 1965. Fuzzy sets. *Information Control* 8: 338–353.

Zhang, B. & Govindaraju, R.S. 2000. Modular neural network for watershed runoff. In R.S. Govindaraju & R.A. Ramanchandra (Eds.), *Artificial Neural Networks in Hydrology*. Kluwer Academic Publishers, The Netherlands.

5

The Application of Time Delay Neural Networks to River Level Forecasting

LINDA M. SEE AND PAULINE E. KNEALE
School of Geography, University of Leeds, UK

ABSTRACT: This chapter considers the use of time delay neural networks for river flow forecasting. These tools have an advantage over feedforward networks trained with backpropagation because such mechanisms are responsive to time-varying behaviours. This has relevance for the inclusion of information from upstream stations, in which the input data set must be lagged, using average travel times, when used with a feedforward network trained with backpropagation. However, this average travel time is not an accurate representation of the time-varying behaviour of the flood wave as it moves down the channel. To compare the performance of both types of neural network, a series of models to forecast lead times of 2 and 4 hours ahead were developed for the River Tyne in Northumbria. The networks were trained on 22 storm events and validated on an additional 14 events over a 10 year period. Global and flood specific evaluation measures were used to assess and compare the performance of the different models. The time delay neural networks performed slightly better overall relative to neural networks trained with backpropagation at a 2 hour lead times but showed a greater overall improvement at 4 hour lead times, suggesting that these models become more effective at longer forecasting horizons. These preliminary results suggest that time delay neural network models should be examined in more depth for use in river flow forecasting.

1 INTRODUCTION

There are many different artificial neural networks (NN) (Shepherd, 1997) but as mentioned in Chapters 2 and 9, NN for forecasting river flow are almost always trained using backpropagation (BPNN). This may be due in part to the fact that BPNN were the first successful models to be implemented

(Rumelhart *et al*., 1986), and because the algorithm is simple to program and apply. Within hydrology there are many examples of the successful application of these network types, e.g. as rainfall-runoff models (Abrahart & Kneale, 1997; Campolo *et al*., 2003; Minns & Hall, 1996; Salas *et al*., 2000; Shamseldin, 1997; Smith & Eli, 1995), for predicting water quality (Brion *et al.*, 2001; Gumrah *et al.*, 2000, Maier & Dandy, 1996) and in estimating rainfall (Dell'Acqua & Gamba, 2003; French *et al.*, 1992; Hsu *et al*., 2000).

Despite the frequent use of BPNN, a major limitation of the standard back-propagation algorithm is that it can only learn an input-output mapping that is static (Haykin, 1994), rendering it well suited to time-independent pattern recognition problems. When time is added in the form of time series data, it is possible to develop a forecasting model using a BPNN provided that the data are stationary, i.e. there are no time varying behaviours in the mean or the variance of the series. River level data are generally nonstationary but this can usually be corrected by pre-processing the data using a single point differencing operation (Masters, 1995). The real difficulty arises when data from upstream gauging stations are used as inputs in the forecasting model. It is then necessary to calculate a travel time between stations which represents the average time difference for a peak flow passing between the two stage gauges. This average travel time is used to lag the upstream station inputs before training the network. If the average travel time does not vary with each storm event, the BPNN should not have any problems in forecasting levels. However, this is not necessarily the case. For example, the average travel time between Bywell and Reaverhill on the River Tyne (Fig. 5.1)

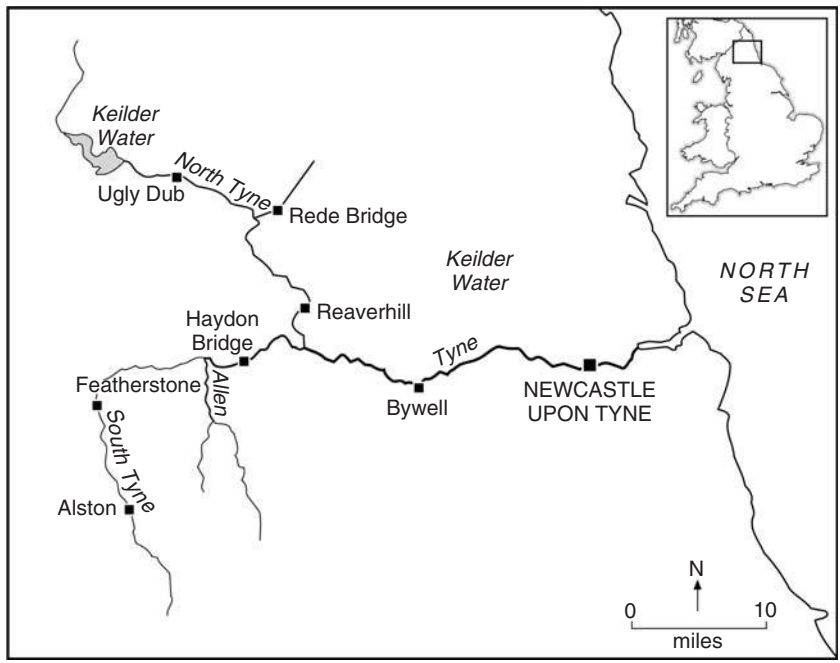


Fig. 5.1. Location map of the River Tyne.

has been calculated at 2 hours but this can vary by ±1 hour. Travel times vary with storm event size and duration; therefore, instead of using a BPNN for forecasting river levels, a time delay neural network (TDNN) (Lang & Hinton, 1988) might be more appropriate. TDNN are dynamic feedforward networks that are designed to capture time-varying behaviour through the addition of time delays in the architecture of the network. These time delays are analogous to adding memory capability (Elman, 1990). This is another example of the application of biological inspiration to NN, as signal delays are important in processing information in the brain (Haykin, 1994).

This chapter considers the feasibility of using TDNN for river level forecasting and contrasts them to BPNN. Both types of model were developed and tested on the River Tyne at Bywell in Northumbria (Fig. 5.1), forecasting levels at lead times of 2 and 4 hours ahead. The networks were trained and validated on a series of storm events over a 10 year period. Global and flood specific evaluation measures were used to compare the different model types. Hydrographs of validation events were also examined.

2 TIME DELAY NEURAL NETWORKS (TDNN)

TDNN have been used to date for a range of different applications including time series analysis of both the stock market (Sitte & Sitte, 2000) and ionospheric conditions (Wintoft & Cande, 1999), image sequence analysis (Cancelliere & Gemello, 1996; Wöhler & Anlauf, 1999), speech recognition and analysis (Waibel *et al*., 1989; Lavagetto, 1997), the regulation of anesthetic dosages to patients (Vefghi & Linkens, 1999), traffic in Banff National Park (Lingras *et al*., 2003) and in wastewater treatment (Zhu *et al*., 1998; Belanche *et al*., 1999). Previous hydrological research includes the use of TDNN to estimate rainfall (Luk *et al*., 2001) and some preliminary explorations of river flow forecasting on the River Tyne (Smith, 2000). Luk *et al.* (2001) developed BPNN, TDNN and PRNN (partial recurrent neural network) models to forecast rainfall one step ahead. These authors found that the models produced comparable results. It was thought that these rainfall time series have very short-term memory characteristics, which might explain why neither TDNN nor PRNN improved upon the performance of BPNN. Smith (2000) compared the performance of TDNN and BPNN for a range of forecasting horizons ranging from 3 to 24 hours ahead. He developed models to predict levels at Bywell using only stations from the North Tyne. This simultaneously addressed the question of whether NN developed on information from the North Tyne would be good enough for flood prediction in the event that the telemetry on the South Tyne failed. The results showed that for the majority of cases, TDNN outperformed BPNN.

2.1 *Network architecture*

TDNN are a type of feedforward network, i.e. the information travels forward so there are no feedback loops as one would find, for example, in recurrent neural networks (Pineda, 1987). TDNN, as with BPNN, have neurons arranged

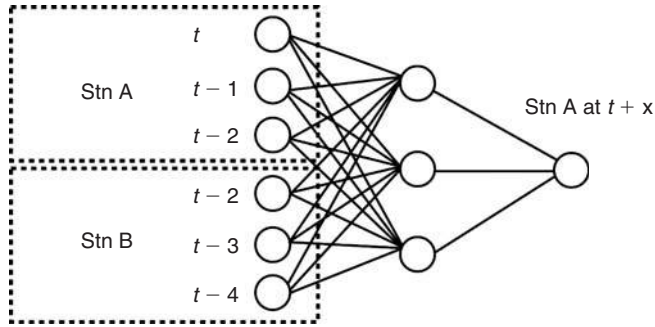


Fig. 5.2. Backpropagation neural network.

in three layers: the input, hidden and output layer, and there are weighted interconnections between the neurons in each layer; these get updated during the training phase. The difference between the two network types can be found in:

- the inputs to the network
- the interconnectivity between neurons
- the training algorithm

As mentioned previously, when using upstream station information as inputs to the BPNN, the average travel time is used to lag the inputs. Figure 5.2 illustrates an example of using two input stations (one at the point of prediction, station A, and one upstream, station B) to predict the level at time $t + x$. If we calculate the average travel time to be 3 hours between station A and station B, we might input the current and two previous level readings at station A and the same number of inputs at the upstream station but lagged by three hours. Abrahart *et al.* (2001) showed using saliency analysis that the three most recent river level inputs are the most important in predicting future river levels. Thus, there are 6 inputs to the model and all neurons in each layer are fully interconnected. This would be the configuration for the BPNN.

Figure 5.3 shows a three dimensional representation of a TDNN for the same stations in Figure 5.2, which illustrates the difference in network inputs between the BPNN and TDNN. The inputs are arranged in a matrix where the x-axis is the station and the y-axis is time, and the inputs are not lagged. The number of elements in the y-axis is referred to as the total delay length. In the example given in Figure 5.3a, the total delay length is five, which covers the same inputs as provided to the BPNN in Figure 5.2.

The second difference between the network types involves neuron interconnectivity and is also illustrated in Figure 5.3, which shows three views of the TDNN. These have been separated out to allow a clearer view of the interconnections but such items are all part of a single TDNN. Connectivity is determined by the size of the receptive field, which is shown as a dashed box in Figure 5.3. The size of the receptive field is determined by the input to hidden delay length, which is chosen to be three in this situation. This again matches the findings of Abrahart *et al.* (2001). The first receptive field is completely

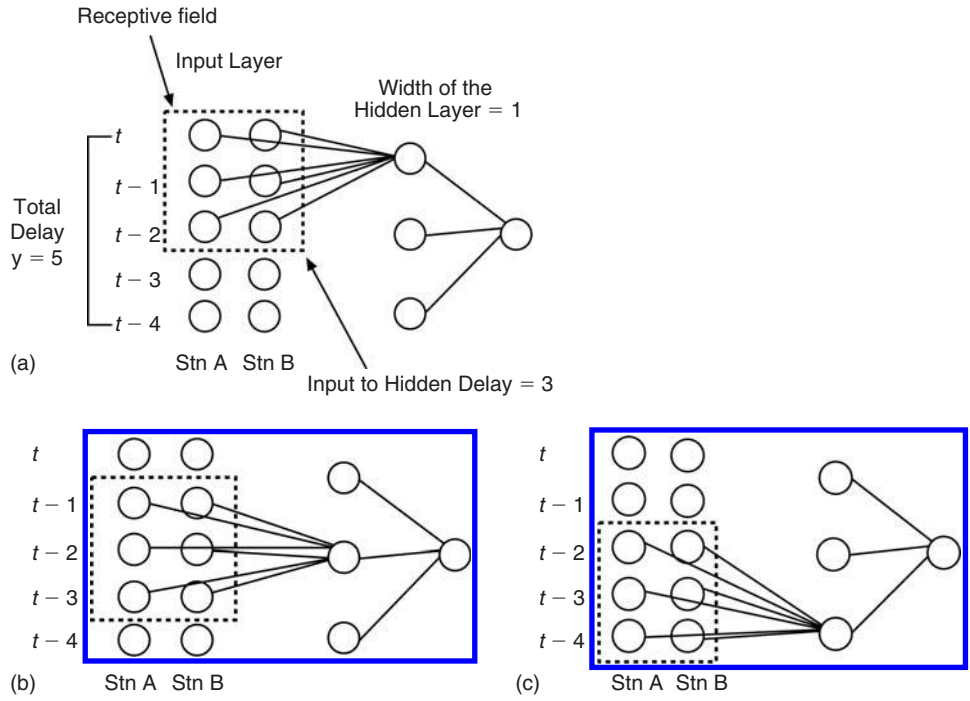


Fig. 5.3. Time delay neural network.

connected to the first neuron in the hidden layer. The second receptive field, which is shown in Figure 5.3b, can be thought of as a window in time that has moved backwards one step. This window is only connected to the second neuron of the hidden layer. Finally in Figure 5.3c, the third and final receptive field is connected to the third neuron of the hidden layer. The length of the hidden layer is specifically chosen to match the number of receptive fields in the network but the width, which in this case is 1, can vary.

2.2 *Network training*

TDNN are trained using an algorithm similar to backpropagation. The modification to this algorithm is necessary to account for what are referred to as coupled links. These are the links that are duplicated in each receptive field and are treated as a single link during the network training. As in standard backpropagation, a forward pass is performed and an output value is calculated. This is compared to the actual output value and the difference or error term is propagated backwards. A single error term is then calculated for each set of coupled links, by averaging the error terms, and this is then applied to the entire set of coupled links. These links are, therefore, changed according to the average of the changes they would experience if treated separately. This method does tend to lead to slower convergence than backpropagation as applied to BPNN and hence a longer training period is required.

3 NEURAL NETWORK EXPERIMENTS

This section outlines the NN experiments undertaken on the River Tyne data. Information on the study area is provided as well as the nine evaluation measures used to compare the models.

3.1 *Study area*

The study area comprises the non-tidal section of the River Tyne, North East England (Fig. 5.1). The Tyne basin has an area of approximately 2,920 km^2. The lowest non-tidal stage gauge is located at Bywell, west of Newcastle. Historic data of stage gauge and rainfall values are available, measured at six stations throughout the Tyne catchment. Telemetered data at 15 minute intervals for the years 1992 to 2001 were provided by the Environment Agency for Bywell and three upstream stations on the River Tyne: Reaverhill, Haydon Bridge and Featherstone.

The average travel times between stations were calculated by plotting hydrographs of historical storm events. The average travel time from Reaverhill to Bywell is 2 hours, from Haydon Bridge to Bywell is 4 hours and from Featherstone to Haydon Bridge 1 hour. Data from the upstream stations were offset by the average travel time before training with backpropagation (BPNN), but the TDNN were not provided with lagged data.

Every major river under the jurisdiction of the Environment Agency has between one and four flood risk warning levels associated with its stage gauges. Each warning level indicates a point where forecasters may need to take action, from alerting emergency services to evacuation of residents. The alarm levels at Bywell are 3.5 m, 4.4 m, 5.3 m and 5.8 m, respectively. These alarm levels were used in devising the TDNN operational evaluation measures.

3.2 *Level vs. flow modelling*

In this chapter the NN models use river levels. River forecasting may be based on modelling level (stage) or flow volumes. The choice for the modeller is likely to depend on data availability and the reason for forecasting. Stage is the height of flow above a datum at the gauging station and is the variable measured by traditional gauges. The stage data are then converted to flow volumes via the gauging station rating curve. This has problems of accuracy and is transforming the data. Flow has the advantage that it is a variable that increases downstream as the river widens and tributaries add to the discharge. Stage in likely to be a more accurate value, and for the flood forecaster it is helpful because at a specific height above the datum the river will start to flow over bank. In practical terms this is very useful. However, for the modeller looking at changing regimes downstream, the datum heights will not necessarily increase (or increase in proportion) to the flow volume. A narrow weir in the upper reaches may record higher absolute stages than a wider, shallower weir downstream with higher flow volumes. The advantage of NN modelling is that it treats each set of inputs independently so modelling on the stage data is practical, eliminating errors associated with the rating curve. If the modeller has doubts

about the rating curve, then using stage data is sensible. It is an advantage that NN independently treat data since a series of forecasts along a channel could be based on a combination of stage data from some traditional gauges and volumetric data from, for example, a sonic gauging station.

3.3 *Outline of experimental runs*

A series of feedforward BPNN and TDNN were trained on historical data covering the period 1992 to 2001 to forecast levels at Bywell for lead times of 2 and 4 hours ahead. From this time period, 47 flood events were selected where a flood event was defined as an event which reached the first alarm level at Bywell of 3.5 m. A three day period around the peak of the flood event was selected. These 47 flood events were then divided into training and validation data sets where 23 storms were selected for the training data set and 14 storms were selected for the validation data set. The highest flood event was placed in the training data set to ensure that each network would see the full range of events and not be forced to extrapolate to heights never seen before. When choosing the training and validation data sets, flood events were selected from each year and from both winter and summer to ensure a good balance between the two.

NN model inputs for all experiments were the levels at Bywell and upstream stations on both the North and South Tyne including Reaverhill, Haydon Bridge and Featherstone. No rainfall information was used. The BPNN had 12 inputs, which covered three previous levels at Bywell and 3 lagged inputs from the three upstream stations. There were 6 neurons in the hidden layer and the output was to predict the level at Bywell at 2 and 4 hours ahead. The TDNN was given a matrix of inputs from the 4 stations at the current time t to $t - 5$. This covered the entire travel time of the BPNN. The width of the hidden layer was 6 neurons and the length was 4, to match the number of receptive fields in the input data set with an input to hidden delay of 3 and a total delay length of 6.

Two different data pre-processing operations were applied prior to training. The first involved normalising the absolute values of the input data between the range 0.1 to 0.9 while the second applied a single point differencing operation prior to normalisation. A continuous river level data set is normally nonstationary but storm events were selected over a 10 year period. Therefore, it is more difficult to determine whether the data set is still nonstationary so both data sets were provided to the different models. NN were trained using backpropagation with momentum (in the BPNN) and backpropagation for a time delay neural network (in the TDNN). Training was stopped when the errors in both the training and validation data sets were at a minimum to avoid problems with overfitting of the data.

3.4 *Evaluation measures*

A series of global and flood specific evaluation measures were calculated on the predictions made by the BPNN and TDNN for the training and validation

data sets. Hydrographs of events from the validation data set were also examined. The following global goodness-of-fit statistics were calculated:

(a) Root Mean Squared Error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(O_i - P_i)^2}{N}} \tag{1}$$

where O_i is the observed value at time i, P_i is the predicted value at time i and N is the total number of observations.

(b) Mean Absolute Error (MAE)

$$\mathrm{MAE} = N^{-1}\sum_{i=1}^{N}|O_i - P_i| \tag{2}$$

(c) A modified Coefficient of Efficiency

$$E_1 = 1 - \frac{\sum_{i=1}^{N}|O_i - P|}{\sum_{i=1}^{N}|O_i - \overline{O}|} \tag{3}$$

where $\overline{O}$ is the mean of O over N and $\overline{P}$ is the mean of P over N. This measure was introduced by Nash and Sutcliffe (1970) and is widely used in the hydrological literature but Legates and McCabe (1999) suggest a modification to the coefficient in which the squared terms are changed to absolute values. They argue that this modification will reduce the influence of outliers.

(d) A modified Index of Agreement

$$d_1 = 1 - \frac{\sum_{i=1}^{N}|O_i - P|}{\sum_{i=1}^{N}\left[|P_i - \overline{O}| + |O_i - \overline{O}|\right]} \tag{4}$$

which was originally proposed by Willmott (1981) as an adaptation of the Nash and Sutcliffe efficiency index but with the squared terms still in place. The change to the denominator acts to penalise the differences in the mean predicted and observed values. Legates and McCabe (1999) once again suggest changing the squared term to an absolute value to reduce the sensitivity to outliers in the data set.

(e) Difference of means

$$\Delta m = \overline{O} - \overline{P} \tag{5}$$

(f) Difference of standard deviations

$$\Delta s = \sigma_o - \sigma_p \tag{6}$$

These measures, although not fully descriptive goodness-of-fit statistics, do provide a useful measure of the model performance. These are also recommended by Legates and McCabe (1999).

The following are a list of additional evaluation measures that are specifically aimed at measuring flood forecasting performance:

(g) Proportion of false alarms

$$E = N^{-1}\sum_{i=1}^{N} f(O_i, P_i)$$
$$f(O_i, P_i) = \begin{cases} 0 & O_i, P_i \text{ are at the same EA alarm level} \\ 1 & O_i, P_i \text{ are at different EA alarm levels} \end{cases} \quad (7)$$

It is useful to assess the proportion of the time that a model falsely predicts an alarm level. The lower this value, the more useful the model will be for operational purposes.

Measures (h) and (i) were originally suggested by Smith (2000) in his investigation of NN forecasting models for the River Tyne. These were found to be useful indicators in combination with the other measures outlined above.

(h) Root Mean Flow Weighted Error (RM_FWE)

$$\text{RM\_FWE} = \sqrt{\frac{\sum_{i=1}^{N} O_i \,|O_i - P_i|}{N}} \quad (8)$$

The absolute difference between the observed and predicted values is weighted by the observed level. Although this weighting assumes a linear importance, the error term will be biased towards storm events. There are very few measures designed to capture an accurate picture of errors at high flows as most global statistics average out the results and are biased towards low flow performance.

(i) Root Mean Gradient Weighted Error

$$\text{RM\_GWE} = \sqrt{\frac{\sum_{i=2}^{N} |O_i - O_{i-1}| \cdot |O_i - P_i|}{N-1}} \quad (9)$$

This is another measure designed to characterise errors at high flows. It adds an extra dimension beyond the RM_FWE, which is insensitive to the errors in the lower part of the rising limb of the hydrograph. This measure takes the gradient, which is the absolute difference between the observed and predicted value, and multiplies it by the absolute error. In this way the sensitivity of the error calculation is increased for periods of rapid change in levels and reduced for periods of stable flow. However, this measure also penalises a model that inaccurately predicts the falling limb of the hydrograph, which is less important for flood forecasting purposes.

4 RESULTS AND DISCUSSION

The goodness-of-fit statistics are provided in Table 5.1 corresponding to lead times of 2 and 4 hours. Note that for the BPNN, the statistics for the undifferenced data are provided while for the TDNN, the statistics for the differenced data are provided, as they produced the best overall results for comparison. However, there was not a great deal of difference between them.

The results for both models generally show an increase in RMSE, MAE, the false alarm rate, RM_FWE and FM_GWE as the lead time increases, which is to be expected. However, the TDNN had lower values for both lead times compared to the BPNN, especially at the longer lead time, indicating that these networks are handling the data better. The RM_FWE and RM_GWE are more difficult measures to interpret than the RMSE and MAE so such items are best used for comparing model performance. It might be more useful to calculate the RMSE and MAE at certain river level intervals so that the variation in the ability of the model to predict at different levels would be clearer.

E_1 and d_1 also follow the same expected pattern, decreasing as the lead time increases. Similarly, the TDNN generally had higher values for all lead times compared to the BPNN. The average differences are very small for the entire data set so there is little indication of a general over or underprediction. The difference in standard deviations indicates a difference in the variation of predictions relative to the observed data. These numbers are also very small,

Table 5.1. Error measures for BPNN and TDNN with a 2 and 4 hour lead time. The top value in each row is the statistic for the training data set while the bottom value is the statistic for the validation data set.

	BPNN		TDNN	
Measure	2 hr	4 hr	2 hr	4 hr
RMSE (m)	0.0764	0.2413	0.0735	0.1376
	0.0960	0.2766	0.0754	0.1366
MAE (m)	0.0432	0.1335	0.0374	0.0773
	0.0537	0.1687	0.0403	0.0814
E_1	0.9499	0.8444	0.9565	0.9090
	0.9376	0.8034	0.9531	0.9042
d_1	0.9748	0.9216	0.9783	0.9542
	0.9686	0.9023	0.9765	0.9516
False Alarms	0.1243	0.3148	0.1000	0.1588
	0.1474	0.3626	0.1158	0.2210
Difference in Mean	−0.0007	0.0028	0.0037	−0.0077
	−0.0056	−0.0648	0.0089	0.0034
Difference in Std Dev	0.0053	0.0279	0.0039	0.0113
	0.0053	−0.0361	0.0063	0.0196
RM_FWE (m)	0.3384	0.6072	0.3207	0.4597
	0.3871	0.6907	0.3419	0.4865
RM_GWE (m)	0.1024	0.1948	0.0937	0.1368
	0.1117	0.1871	0.0968	0.1340

showing a difference in variation of a few millimetres or less. These measures did not provide very much information to differentiate model performance.

Overall, the goodness-of-fit statistics, with the exception of the differences in the average and standard deviation, showed that the TDNN performed better than the BPNN but that the increase in performance was more noticeable at the longer lead time. This has positive implications for operational flood forecasting.

Figures 5.4 to 5.7 show hydrographs from two events in the validation data set. Figure 5.4 shows an event in the winter of 1995 for a lead time of 2 hours. Both the BPNN and TDNN capture the hydrograph well but the BPNN

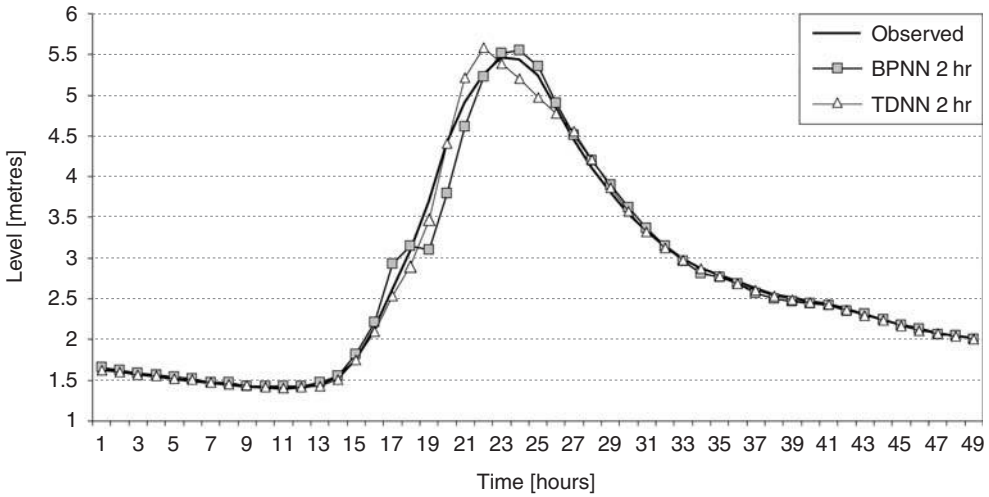


Fig. 5.4. Validation hydrograph (Feb 21 to 23 1995) 2 hr lead time.

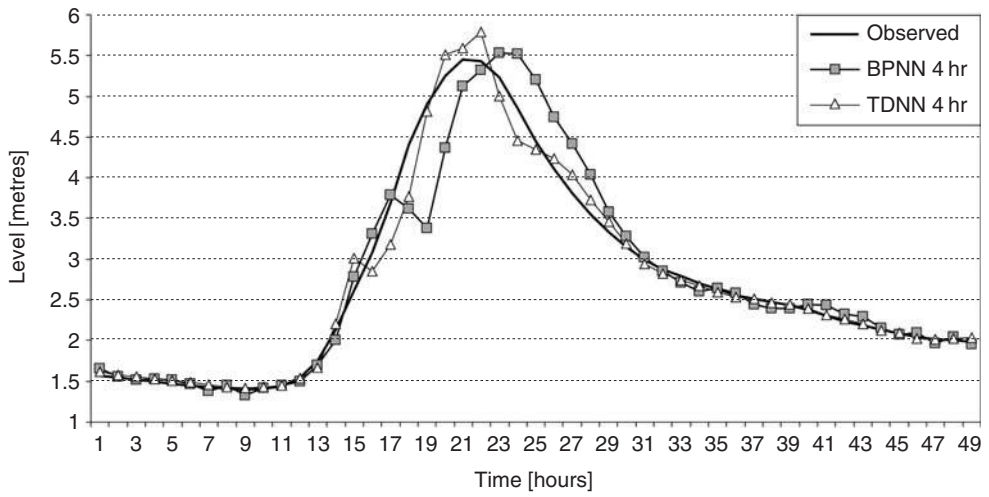


Fig. 5.5. Validation hydrograph (Feb 21 to 23 1995) 4 hr lead time.

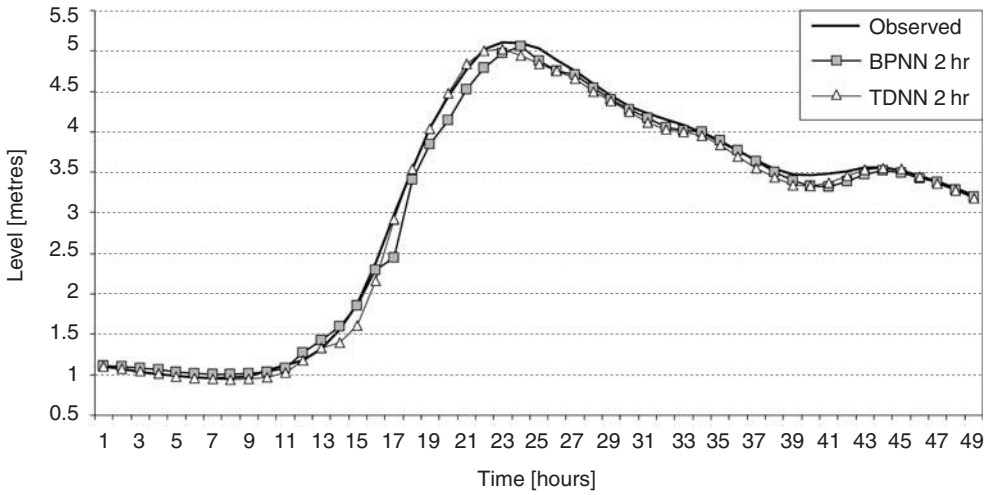


Fig. 5.6. Validation hydrograph (Nov 5 to 7 2000) 2 hr lead time.

is better in terms of the timing of prediction of the first and second alarm (i.e. 3.5 m and 4.4 m) levels. The BPNN predicts the next level (5.3 m) early while the TDNN predicts it on time. The highest level (5.8 m) is not reached. The peak is better predicted by the TDNN, and both models predict the falling limb well although the BPNN is early in its predictions.

Figure 5.5 shows the same event for a lead time of 4 hours. The alarm level is predicted well by the BPNN and one hour late by the TDNN but after this point the behaviour of the BPNN is very poor. The timing is completely wrong with the predicted hydrograph shifted by a few hours. The TDNN predicts the third alarm level (5.3 m) on time but overpredicts the peak. The TDNN is better at predicting the falling limb than the BPNN. Thus, the TDNN shows better performance at a 4 hour lead time relative to the BPNN. This is also reflected in the goodness-of-fit statistics in Table 5.2.

Figures 5.6 and 5.7 show an event in the winter of 2000 for both 2 and 4 hour lead times. The 2000/2001 winter period was one of the worst for flooding in the UK on record. This event does not reach the maximum ever recorded but is nevertheless still an event of considerable magnitude. Figure 5.6 shows that both models predict the hydrograph well although in overall terms the TDNN is better than the BPNN (which is a little bit late). Alarm levels are predicted very well by both models but there is a slight underprediction of the peak by the TDNN. The falling limb is also well predicted by both models.

Figure 5.7 shows the same event for a lead time of 4 hours. The TDNN is a bit late on the rising limb, missing the alarm by an hour but predicts the next alarm level on time. The higher alarm levels were not triggered. The peak is well predicted but there is a sharper decline in the falling limb. For the BPNN, the rising limb is well predicted although the alarm level is out by one hour. After this point, the same behaviour occurs as in Figure 5.5. The hydrograph is shifted and

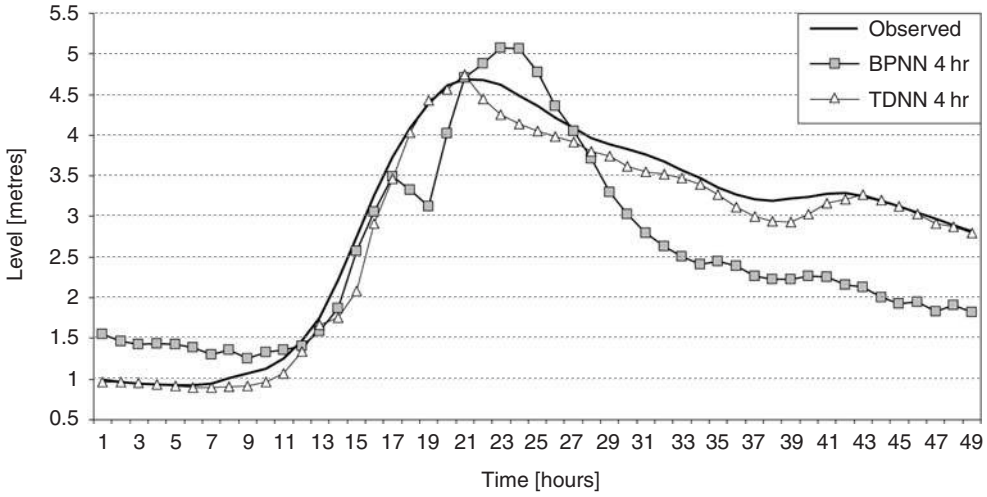


Fig. 5.7. Validation hydrograph (Nov 5 to 7 2000) 4 hr lead time.

the peak is over predicted, with a false prediction of the third alarm level (5.3 m) that did not occur in practice. Finally, the falling limb is poorly characterised.

The BPNN is better than the TDNN at predicting the initial alarm level (3.5 m) but after this point it appears to break down at a lead time of 4 hours. There may be a change in the travel time of the flood wave as the storm progresses. The TDNN is able to pick up this behaviour but the BPNN with its static mapping is incapable of adapting to this change. As both models are quick to train on storm events, as opposed to a continuous data set, it would be possible to use both models in a multi-modelling approach whereby the early part of the rising limb, which appears to be better predicted by the BPNN, could be used to make initial predictions, with a switch to the TDNN when later predictions or longer lead times are needed. More investigation is required to determine whether this pattern is consistent.

5 CONCLUSIONS

This chapter has compared the performance of TDNN with conventional feed-forward networks trained with backpropagation (BPNN). These networks were trained to forecast stage on the River Tyne, Northumbria using only upstream stations. The networks were trained on a series of flood events over a 10 year period for lead times of 2 and 4 hours. An examination of the goodness-of-fit statistics showed that the TDNN marginally outperformed the BPNN at a 2 hour lead time. This was confirmed when examining the hydrographs of 2 validation events although the BPNN appeared to be better at predicting the initial alarm level. However, at a 4 hour lead time, the TDNN considerably outperformed the BPNN. The validation hydrographs illustrated the poor performance of the

BPNN after reaching the alarm levels, which may indicate a change in the travel time between stations and the inability of the BPNN to respond to this change. These results are encouraging and suggest that TDNN may be able to play an important operational role in forecasting floods at longer lead times.

REFERENCES

Abrahart, R.J. & Kneale, P.E. 1997. Exploring neural network rainfall-runoff modelling. *Proceedings of the Sixth National Hydrology Symposium,* University of Salford, 9.35–9.44.

Abrahart, R.J., See, L. & Kneale, P.E. 2001. Investigating the role of saliency analysis with a neural network rainfall-runoff model. *Computers and Geocsciences* 28: 921–928.

Belanche, L.A., Valdes, J.J., Comas, J., Roda, I.R. & Poch, M. 1999. Towards a model of input-output behaviour of wastewater treatment plants using soft computing techniques. *Environmental Modelling and Software* 14: 409–419.

Brion, G.M., Neelakantan, T.R. & Lingireddy, S. 2001. Using neural networks to predict peak Cryptosporidium concentrations. *Journal of the American Water Works Association* 93: 99–105.

Campolo, M., Soldati, A. & Andreussi, P. 2003. Artificial neural network approach to flood forecasting in the River Arno. *Hydrological Sciences Journal-Journal Des Sciences Hydrologiques* 48: 381–398.

Cancelliere, R. & Gemello, R. 1996. Efficient training of Time Delay Neural Networks for sequential patterns. *Neurocomputing* 10: 33–42.

Dell'Acqua, F. & Gamba, P. 2003. Pyramidal rain field decomposition using radial basis function neural networks for tracking and forecasting purposes. *IEEE Transactions on Geoscience and Remote Sensing* 41: 853–862.

Elman, J. 1990. Finding structure in time. *Cognitive Science* 14: 179–211.

French, M.N., Krajewski, W.F. & Cuykendall, R.R. 1992. Rainfall forecasting in space and time using a neural network. *Journal of Hydrology* 137: 1–31.

Gumrah, F., Oz, B., Guler, B. & Evin, S. 2000. The application of artificial neural networks for the prediction of water quality of polluted aquifer. *Water Air and Soil Pollution* 119: 275–294.

Haykin, S. 1994. *Neural Networks, A comprehensive foundation*. New York: Macmillan.

Hsu, L.-L., Gupta, H.V., Gao, X. & Sorooshian, S. 2000. Rainfall estimation from satellite imagery. In R.S. Govindaraju & A.R. Rao (Eds.), *Artificial Neural Networks in Hydrology* (pp. 23–52). London: Kluwer Academic Publishers.

Lang, K. & Hinton, G. 1988. *The development of the time-delay neural network architecture for speech recognition*. Tech. Rep. CMU-CS-88-152, Carnegie-Mellon University.

Lavagetto, F. 1997. Time-Delay Neural Networks for estimating lip movements from speech analysis: A useful tool in audio-video synchronization. *IEEE Transactions on Circuits and Systems for Video Technology* 7: 786–800.

Legates, D.R. & McCabe, G.J. 1999. Evaluating the use the 'goodness-of-fit' measure in hydrologic and hydroclimatic model validation. *Water Resources Research* 35: 233–241.

Lingras, P., Sharma, S. & Zhong, M. 2003. Prediction of recreational travel using genetically designed regression and time-delay neural network models. *Travel Demand and Land Use 2002, Transportation Research Record* 1805: 16–24.

Luk, K.C., Ball, J.E. & Sharma, A. 2001. An application of artificial neural networks for rainfall forecasting. *Mathematical and Computer Modelling* 33: 683–693.

Maier, H.R. & Dandy, G. 1996. The use of artificial neural networks for the prediction of water quality parameters. *Water Resources Research* 32: 1013–1022.

Masters, T. 1995. *Neural, Novel & Hybrid Algorithms for Time Series Prediction*. New York: John Wiley & Sons.

Minns, A.W. & Hall, M.J. 1996. Artificial neutral networks as rainfall-runoff models. *Hydrological Sciences Journal* 41: 399–417.

Nash, J.E. & Sutcliffe, J.V. 1970. River flow forecasting through conceptual models, I, A discussion of principles. *Journal of Hydrology* 10: 282–290.

Pineda, F.J. 1987. Generalisation of back-propagation to recurrent neural networks. *Phys. Rev. Letters*, 59: 2229–2232.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. 1986. Learning internal representation by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (pp. 318–362), Vol. 1. Cambridge MA: MIT Press.

Salas, J.D., Markus, M., & Tokar, A.S. 2000. Streamflow forecasting based on artificial neural networks. In R.S. Govindaraju & A.R. Rao (Eds.), *Artificial Neural Networks in Hydrology* (pp. 23–52). London: Kluwer Academic Publishers.

Shamseldin, A. 1997. Application of a neural network to rainfall-runoff modelling. *Journal of Hydrology* 199: 272–294.

Shepherd, A.J. 1997. *Second-Order Methods for Neural Networks*. London: Springer-Verlag Limited.

Sitte, R. & Sitte, J. 2000. Analysis of the predictive ability of Time Delay Neural Networks applied to the S&P500 Time Series. *IEEE Transactions on Systems, Man and Cybernetics Part C* 30: 568–572.

Smith, A. 2000. *A comparison of neural network designs with regards to their capability for river level forecasting, for the Tyne River, NE England*. Unpublished MSc Dissertation. School of Geography, University of Leeds: Leeds, UK.

Smith, J. & Eli, R.N. 1995. Neural network models of rainfall-runoff process. *Journal of Water Resources Planning and Management* 121: 499–509.

Vefghi, L. & Linkens, D.A. 1999. Dynamic monitoring and control of patient anaesthetic and dose levels: time-delay, moving-average neural networks and principal components analysis. *Computer Methods and Programs in Biomedicine* 59: 91–106.

Waibel, A. 1989. Modular construction of time-delay neural networks for speech recognition. *Neural Computing* 1: 39–56.

Willmott, C.J. 1981. On the validation of models. *Physical Geography* 2: 184–194.

Wintoft, P. & Cander, L.R. 1999. Short-term prediction of foF2 using Time-Delay Neural Network. *Phys. Chem. Earth* 24: 343–347.

Wöhler, C. & Anlauf, J.K. 1999. An adaptable Time-Delay Neural-Network Algorithm for Image Sequence Analysis. *IEEE Transactions on Neural Networks* 10: 1531–1536.

Zhu, J., Zurcher, J., Rao, M., Meng, M.Q-H. 1998. An on-line wastewater quality prediction system based on time-delay neural networks. *Engineering Applications of Artificial Intelligence* 11: 747–758.

6

The Application of Cascade Correlation Neural Networks to River Flow Forecasting

CLAIRE E. IMRIE
Department of Environmental Science and Technology, Imperial College of Science, Technology and Medicine, Prince Consort Road, London, SW7 2AZ UK

ABSTRACT: The cascade correlation learning architecture (Falman & Lebiere, 1990) is a constructive neural network algorithm which automatically determines the structure of the neural network by adding hidden neurons throughout the training process. At present, it is more commonly used for classification tasks than for regression tasks, as the learning process tends to saturate the units (Hwang *et al*., 1996). However, despite this limitation, it has been successfully employed in a number of river flow forecasting applications (Karunanithi *et al*., 1994; Muttiah *et al*., 1997; Imrie *et al*., 2000; Lekkas *et al*., 2001).

This chapter provides an insight into the background and the motivation that led to the development of cascade correlation. The mechanics of the algorithm are described in some detail, as is the quickprop (Falman, 1988) update rule that is often used in conjunction with cascade correlation. This is followed by a number of suggested modifications that may render the algorithm more suitable for regression tasks such as river flow forecasting. Finally there is a flow-routing case study using the River Trent in the UK.

1 BACKGROUND AND MOTIVATION

Artificial neural networks (NN) provide a potential alternative to statistical methods as a data-driven approach to environmental modelling. Multi-layer feed-forward networks (MLFN), which in most environmental applications are used for prediction and forecasting purposes (Maier & Dandy, 2000), are able to

approximate non-linear functions through learning procedures such as the error backpropagation algorithm (Rumelhart *et al*., 1986). However, although NN have the computational ability to outperform statistical techniques, there are a number of disadvantages that have restricted their application to the research environment.

If the backpropagation neural network (BPNN) architecture is used for training, the network's size and structure must be predefined, although network pruning can be incorporated as part of the training algorithm (Abrahart *et al*., 1999). The optimal architecture will have sufficient parameters to capture the relevant processes, but too few for the network to readily over-train. However, it is impossible to identify the correct size for a NN without having prior knowledge of the rules it will have to learn (Lee, 1997). Although a number of studies have been conducted, there is at present no standard method of determining the most suitable configuration of hidden units. As such, a trial and error approach is usually employed in finding the optimum architecture (Maier & Dandy, 2000). However, this process may be somewhat frustrating and time-consuming (Karunanithi *et al*., 1994).

One approach towards surmounting this problem may be to search the space of network structures using a genetic algorithm (Miller *et al*., 1989; Yao, 1993; Blanco *et al*., 2000). This process, however, would be time and CPU intensive (Russell & Norvig, 1995). Maren *et al*. (1990) suggest that where the outputs must be continuous functions of the input, two layers of hidden units should be used. A number of guidelines for determining the optimum number of hidden units empirically have arisen through experimentation. For example, Hecht-Nielsen (1987) suggests an upper limit of $2I + 1$, where I is the number of input units. Other suggestions relate the number of hidden units to the number of training patterns available (Weigend *et al*., 1990).

Various 'pruning' methods have been developed whereby training commences with a large network structure and weights are gradually removed, reducing the network's size. Optimal brain damage (Le Cun *et al*., 1990), for example, calculates the saliencies (importance) of the weights when a specified error level is reached and removes those connections with the lowest values. The network is trained again and this process of elimination is repeated until no further improvement is observed. Karnin (1990) also suggests beginning with a large number of hidden units and pruning these until an optimal architecture is found. A method for weight pruning using genetic algorithms was more recently forwarded by Bebis *et al*. (1997).

In practice, however, it is more computationally efficient and therefore more practical to begin with a minimal network and add units one at a time (Hsu *et al*., 1995). To locate an optimal architecture in this manner, a number of 'constructive' algorithms have been developed (Fahlman & Lebiere, 1990; Hirose *et al*., 1991; Setiono & Hui, 1995).

Besides the need to pre-specify the architecture of the NN before training, another problem that is associated with the error backpropagation algorithm is caused by the fact that all the weights in the network are adjusted at the same time. Each unit tries to detect a feature defined by the error signal propagated backwards. However, since a unit's weights change independently to those

of the others, the error signal, and hence the problem, are constantly being redefined. This leads to a 'complex dance' amongst the units, increasing the time taken to reach a stable condition (Fahlman & Lebiere, 1990).

The problems discussed above inspired the development of the cascade correlation learning architecture (Fahlman & Lebiere, 1990), which constructively builds the network by adding one hidden unit at a time. Once fully installed in the network each new hidden unit acts as an individual 'feature detector', thus eliminating the moving target problem.

A final point to make here concerns another aspect of generalisation. An important criterion when developing environmental forecasting models is that the models can perform well in the event of an extreme occurrence. However, it has been found in previous studies (Minns & Hall, 1996; See *et al*., 1997; Dawson & Wilby, 1998; Campolo *et al*., 1999) that NN tend to perform poorly outside the calibration range, and therefore cannot be reliably used in situations where significant events are of the most concern. Obviously, flow forecasting is one such application since we are often interested in the extremes and are often faced with a limited amount of calibration data. The main reason for the poor performance of the popular BPNN is that all the data are routed through one or more layers of sigmoidal functions, which ultimately means that the maximum output value attainable is proportional to the number of hidden units in the final layer. Although the cascade correlation algorithm is largely overlooked by NN modellers, it surmounts this problem to a large degree as the input units have direct connections to the output units, and so the restriction does not apply.

2 THE CASCADE CORRELATION LEARNING ARCHITECTURE

2.1 *Training procedure*

The training of a NN using the cascade correlation (CC) learning architecture proceeds as follows:

1. The algorithm begins with a one-layer network comprising an input layer and an output layer. The interconnecting weights are trained until a pre-specified or minimum error level is reached.
2. The input-output weights are then frozen, and a pool of 'candidate' units is connected to the input layer.
3. The data patterns are propagated forwards through the CC network both to the output layer and to the layer of candidate units.
4. The activation of each candidate c_p is compared with the residual error E_o summed over the output layer upon the presentation of each pattern p.
5. The covariance C between each candidate's activation and the error signal is calculated as follows:

$$C = \sum_o \left| \sum_p (c_p - \bar{c})(E_{p,o} - \bar{E}) \right| \tag{1}$$

where $\bar{c}$ and $\bar{E}_o$ are the values of c and E_o averaged over the pattern set.

6. The input-candidate weights are trained so that the covariance C is maximised, using the following update rule:

$$\frac{\partial C}{\partial w_i} = \sum_{p,o} \sigma_o (E_{p,o} - \overline{E}) f'_p I_{i,p} \quad (2)$$

where w_i are the candidate's incoming weights, and for each pattern p, f'_p is the derivative of the candidate's activation with respect to the sum of its inputs, and $I_{i,p}$ is the input each receives from unit i. The value of σ_o is $[-1]$ if C is negative and $[+1]$ if C is positive. The weights are adjusted until no further improvement is observed, or until a specified maximum number of iterations has been reached.
7. The covariances calculated for each candidate are compared, and the candidate which is deemed most highly correlated with the residual error is installed into the network as a hidden unit. Its input weights are frozen and new weights connect it to the units in the output layer.
8. A second round of error minimisation is undertaken, as in Step 1. After these weights have been frozen, a second pool of candidate units is connected to both the input layer and the newly-installed hidden unit, and the procedure continues as before.

Hidden units are incorporated in this way until the output error has reached a satisfactory level. The final network will therefore have a multi-layer structure, with each hidden layer containing a single hidden unit. The topology of a cascade correlation neural network is illustrated in Figure 6.1.

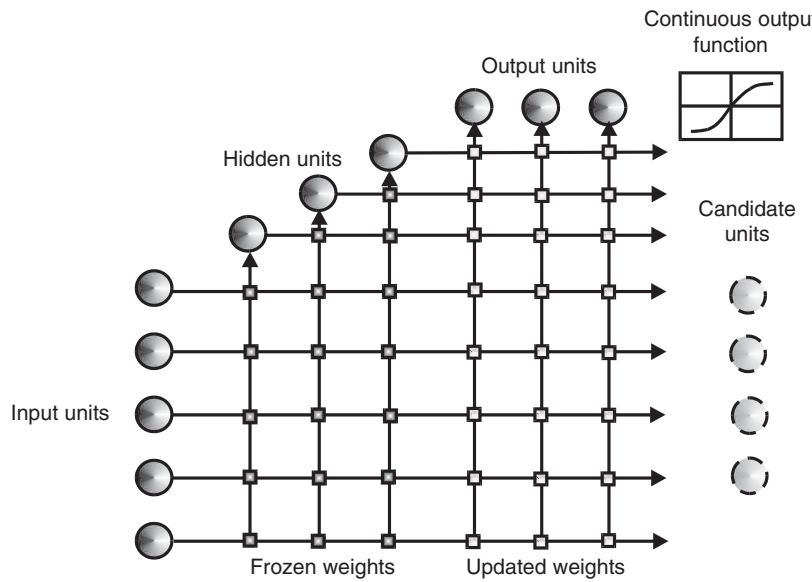


Fig. 6.1. The cascade correlation learning architecture.

As can be seen in Figure 6.1, the input units of a CCNN have direct connections with the output units, and as such the data are not forced through hidden layers of limiting sigmoidal functions. If linear activation functions are used at the output layer, an indirect advantage of this is that there is no limit to the network's output.

A standard rule such as the gradient descent method can be used to update the network's weights. However, due to the fact that only one layer of weights is trained at a time, the learning algorithm Quickprop (Fahlman, 1988) can be used instead. This allows faster convergence to be observed (Fausett, 1994). This update rule is described in detail below.

2.2 *Quickprop*

An alternative to the gradient descent method of calculating the weight increments is the use of quickprop (Fahlman, 1988). This algorithm was developed to speed up the NN training process. In brief, the quickprop algorithm uses information about the error surface curvature to take larger steps towards the solution. For each weight, a copy of the previous error derivative, as well as the difference between the current and previous weight value, are stored. Fahlman assumed that the error versus weight curve could be approximated by a parabola for each weight. Another assumption was that the change in the slope of the error curve is not affected by the simultaneous changing of the other weights. The weight increments $\Delta w(t)$ are calculated as follows:

$$\Delta w(t) = \frac{\frac{\partial E}{\partial w}(t)}{\frac{\partial E}{\partial w}(t-1) - \frac{\partial E}{\partial w}(t)} \Delta w(t-1) \tag{3}$$

A number of parameters were added to the algorithm in order to make it work properly. For example, to avoid the possible scenario whereby an infinite step is taken, or the algorithm begins instead to search for a local maximum, the 'maximum growth factor', μ, was introduced. The algorithm specifies that no step greater than μ multiplied by the previous step for that weight can be taken.

An additional parameter, ε, is used as a momentum factor, and a weight decay term is used to avoid excessive growth of the weight values. Due to the inclusion of the previous weight update in the algorithm, the quickprop learning process has to be started by making a single update using the gradient descent method.

It should also be noted that a number of alternative update rules may be used, besides quickprop and the gradient descent method. One such rule is resilient backpropagation, which is a local adaptive learning scheme (Riedmiller & Braun, 1993) whereby the weight update values change according to the behaviour or the error function.

The cascade correlation learning architecture and the quickprop update rule are provided as part of the well-established neural network software package SNNS. The Stuttgart Neural Network Simulator (SNNS) is a freeware package (SNNS Group, 2002). The package includes a wide range of neural network architectures and algorithms, and provides a graphical user interface through which neural networks can be designed and trained. Its use has been reported in a number of applications (Abrahart & Kneale, 1997; Tchaban *et al*., 1998; See & Openshaw, 1998; Campolo *et al*., 1999; Hasenauer *et al*., 2001).

The cascade correlation algorithm provided in SNNS does not include an automatic method for early stopping using cross-validation. This means that a manual process of cross-validation must be employed to ensure generalisation, whereby training is temporarily stopped after the addition of each hidden neuron so that the performance of the NN can be checked against a separate data set. Such a procedure was described by Hasenauer *et al*. (2001). This tedious process can however be automated, as demonstrated by Imrie *et al*. (2000) with a number of additional modifications to improve model generality. These modifications are described in the next section.

3 MODIFIED CASCADE CORRELATION

3.1 *Ensuring model generality*

In order to ensure that the network will generalise and the final model will perform adequately when confronted with fresh data, a 'guidance system' may be incorporated (Imrie *et al.*, 2000). This involves the use of a cross-validation method whereby the available data are split into three sets: a *training* set, used by the algorithm to build the network and update the weights; a *testing* set, used to periodically check the performance of the network during the training procedure; and a *validation* set, by which the performance of the final NN model is assessed.

The training and testing patterns are first loaded into the computer's memory. Input-output weight arrays are constructed and initialised with random values between -1.0 and 1.0. The training process is kick-started by adjusting the input-output weights with a single round of gradient descent. The weights are then trained using the quickprop algorithm over a pre-specified maximum number of weight updates. The procedure to automatically check for over-training has been incorporated as follows. Before training commences, a parameter *fail* is set to zero. When the weights have been adjusted 24 times, the residual error E_{24} is calculated over the testing pattern array. This is repeated after the next weight update, and if the test set error E_{25} is greater than E_{24}, *fail* is incremented by 1. The update counter is reset to 1 and training is allowed to proceed for another 24 updates, whereupon the error E_{24} is calculated and compared with the previous E_{25}. If the new E_{24} is greater than the old E_{25} *fail* is incremented by 1, otherwise it is reset to zero. Each time a lower test set error is encountered the current weights are saved in a temporary array. The training process is stopped when the condition (*fail* $= 2$ AND $E_{25} > E_{24}$) is satisfied.

The first hidden unit can then be added to the network. An array of input-candidate weights is initialised so that their values lie between –1.0 and 1.0. The procedure of maximising the covariance between the output residual error and the activation of the candidate units is kick-started as before with a round of gradient descent. Quickprop is then used to update the candidate weights. Each update involves two 'sweeps' through the training patterns: the candidate covariance is calculated first, and this is then used to compute the weight increments during the second sweep.

An automatic stopping procedure is also included in the candidate training phase. After every 25 weight updates the testing patterns are used to calculate the covariance between the candidate activations and the residual output errors. Each time a higher test set covariance is observed, the corresponding weights are saved in a temporary array. Once the test set covariances for all the candidate units have stopped increasing, the temporary weight array is used to install a new hidden unit.

A hidden unit activation array is then constructed for the training and testing patterns. An array of hidden-output weights is initialised randomly with values between -1.0 and 1.0, and a round of output weight training commenced. The training phase proceeds as described above, adjusting both the input-output weight array and the hidden-output weights to minimise the residual error at the output layer.

The algorithm continues to build the network architecture in this way until a specified maximum number of hidden units have been installed. A parameter *best_error* is updated every time a lower total error is calculated for the testing data. The configuration of weights that has given rise to this network should be saved in an external network file, so that this 'best model' can be retrieved at the end of training. The performance of this model can then be evaluated using the validation data set.

Further modifications can be implemented to increase the chance of an optimal model being obtained. One such modification involves the undertaking of a number of 'trials' at each stage of the algorithm. The network saved at stage t is used as a starting point for the trials undertaken in stage $t + 1$, where each stage commences at the covariance update phase. Figure 6.2 outlines the progress of the cascade correlation learning algorithm with these changes implemented.

3.2 *Improved Candidate Selection Procedure*

Through experimentation with the SNNS software and the version of cascade correlation with the modifications implemented, it has been observed that the performances of the resulting NN were relatively insensitive to the number of candidates involved in the correlation phase. In addition, it appeared that the magnitude of the covariances calculated with respect to the test set had little bearing on the ensuing reduction in the test set residual error. Indeed, Prechelt (1997) noted that the "covariance is an ill-suited target function for training the candidates". This was attributed to the resulting tendency for the algorithm to

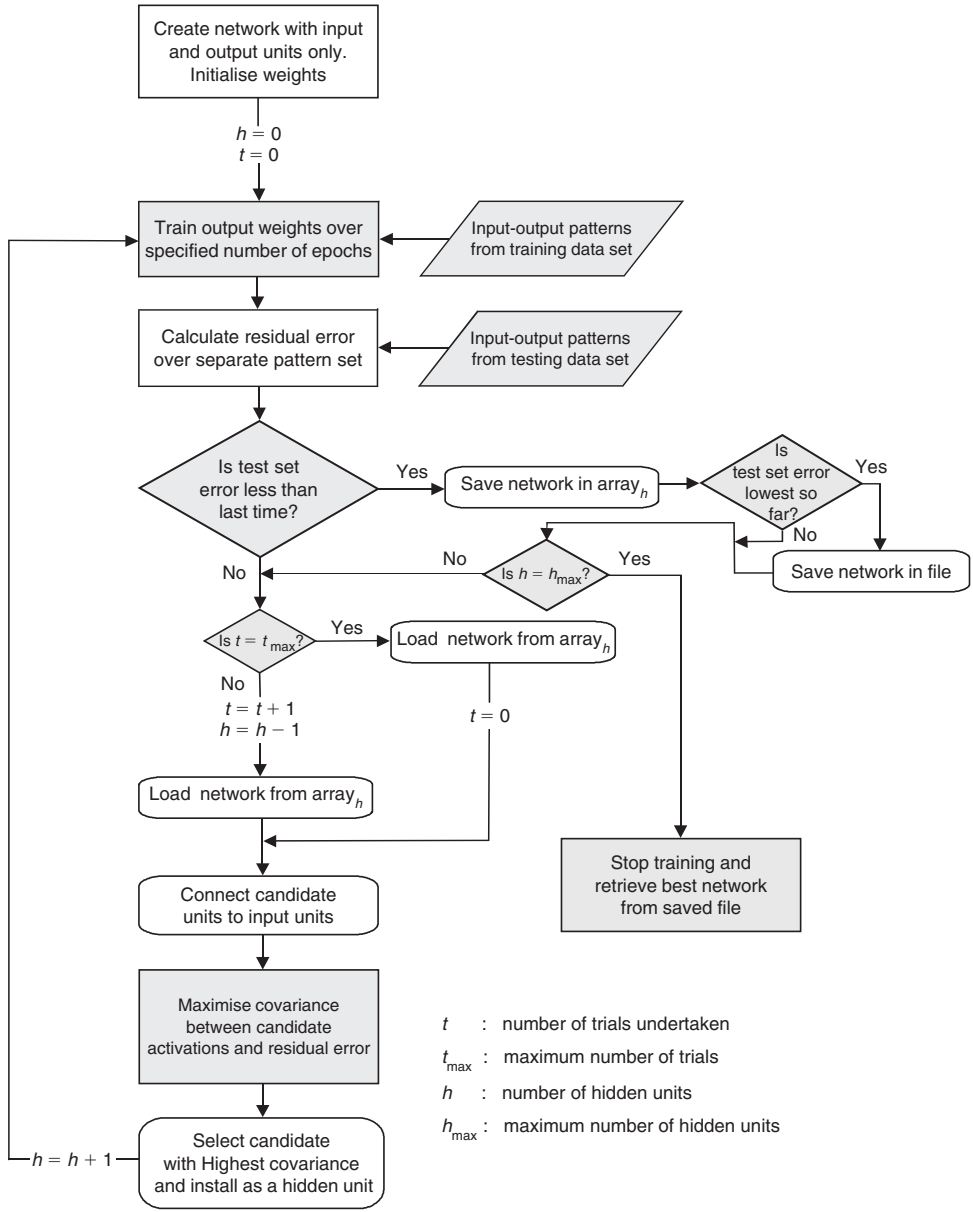


Fig. 6.2. Flow chart showing training procedure with modified cascade correlation algorithm.

over-compensate for errors, leading to the conclusion that cascade correlation was more suitable for classification tasks than for regression.

Furthermore, the correlation phase of the cascade correlation learning architecture requires two passes through the training data prior to each weight update and is therefore time-consuming. Reducing the size of the candidate

pool and the number of trials undertaken would allow the training process to progress at a greater speed. It was therefore considered that if an improved method of candidate selection could be identified, the modified cascade correlation algorithm would be rendered more efficient.

Maximisation of the cross-correlation between the candidate activations and the residual error would be extremely computationally expensive, and therefore impractical. The original cascade correlation algorithm of Fahlman and Lebiere (1990) assumes that maximising the covariance will make an acceptable alternative. Experimentation with the algorithm, however, revealed that the candidate unit with the highest covariance is not necessarily that with the highest cross-correlation. Bearing this in mind, together with the previous observation that the candidate unit with the highest covariance was not necessarily the best choice, an alternative candidate selection procedure was developed.

Intensive experimentation revealed that the candidate covariance was typically positively correlated with the test set residual error, suggesting that the higher the covariance of a candidate, the worse it was likely to perform when installed as a hidden unit. The highest negative correlation was obtained for a combination of two parameters: the sum of the training and test set cross-correlations between the candidate activation and the residual error. The candidate selection procedure implemented in the cascade correlation algorithm can therefore be altered to operate as follows:

- After the 25th weight update, the cross-correlations with respect to the training and testing data are calculated and summed for each candidate unit.
- Once all the cross-correlation sums have stopped increasing, the covariance-maximising phase is stopped.
- The candidate with the highest cross-correlation sum is installed as a hidden unit.

Furthermore, the algorithm was found to converge more quickly when output weights for the new hidden units were initialised with the cross-correlation coefficient calculated with respect to the training data. A similar procedure was suggested by Phatak and Koren (1994). Liang and Dai (1998) suggest using a genetic algorithm to search for the optimum weights, although this method is likely to be time-consuming and may encourage NN over-training. As a final note, Prechelt (1997) suggests that for regression tasks, the covariance maximisation procedure should be replaced by a direct error minimisation. This is an option for future research.

4 RECURRENT MODIFIED CASCADE CORRELATION ALGORITHM

The majority of NN forecasting applications in hydrology involve the construction of input patterns that contain a length of lagged values representing time series windows of the determinant of interest and other pertinent variables (e.g. Hsu *et al*., 1995; Minns & Hall, 1997; Campolo *et al*., 1999; Zealand *et al*., 1999).

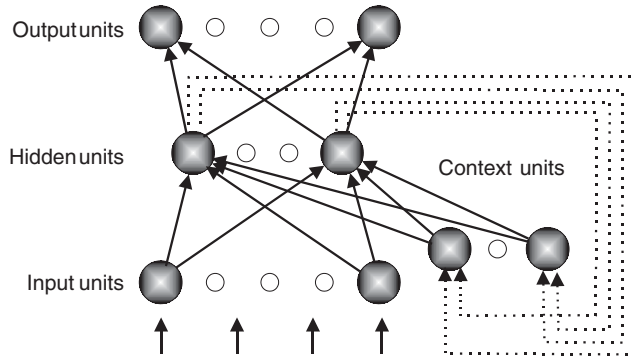


Fig. 6.3. The recurrent architecture of Elman networks.

However, when the forecast lead-time is greater than one time-step, it may be useful to use the NN forecast of the modelled variable as an additional input to the next time step. This principle is used in recurrent neural networks, which were first conceived by Jordan (1986). These tools are now commonly employed to do temporal processing tasks (Wang *et al*., 1996), although their application in hydrological modelling is not widely reported.

The simplest form of a partial recurrent NN is the Elman network (Elman, 1988), whose architecture is presented in Figure 6.3. These networks assume that the NN operates in discrete time-steps. The activations of the hidden units at time t are fed backwards and used as inputs to 'context units' at time $t + 1$, representing a kind of short-term memory. The importance and influence of these lag 1 inputs are determined during the training of the network.

A recurrent version of the original cascade correlation algorithm has also been developed (Fahlman, 1991). In this case the hidden unit activations are no longer fed back to all of the other hidden units. Instead, every hidden unit has only one self-recurrent link, which is trained along with the candidate unit's other input weights to maximize the correlation. When the candidate unit is added to the active network as a hidden unit, the recurrent link is frozen along with all other links.

The majority of recurrent NN algorithms were originally designed for tasks associated with temporal sequences, such as natural language processing and recognising characters from Morse Code (Fahlman, 1991; Wang *et al*., 1996). As such, the hidden unit activations are recycled as internal state variables, and the resulting NN are used to map sequences of inputs into desired corresponding sequences of outputs. The problem posed in river flow forecasting differs in that the aim is to provide a continuous sequence of forecasts with lead times of greater than one time step. For this reason, the recurrent modified cascade correlation algorithm developed in this chapter recycles the *output* of the network instead of the activations of the hidden units. There are a number of advantages to this simple implementation: the number of input units does not grow as the hidden units are added; and it would be possible to directly determine the relative importance of the recycled values in a sensitivity analysis.

It should be noted that there are also a number of possible drawbacks to the use of recurrent NN. Firstly, the procedure of training the weights in recurrent neural networks is much less orderly than in simple feedforward networks (Russell & Norvig, 1995). The networks can become unstable and chaotic. In particular, for a NN that uses its outputs as additional inputs on the next pattern, each input pattern will change after each weight update. This constitutes a moving target problem, as the error surface is continually changing as training proceeds. Furthermore, the benefits of recycling the output predictions will ultimately depend on the quality of the predictions themselves. However, results obtained in previous research showed that the recurrent version performed better in various river flow prediction applications than the modified cascade correlation algorithm alone (Lekkas *et al*., 2001).

5 CASE STUDY: 12-HOUR FLOW FORECASTING ON THE RIVER TRENT, UK

All the modifications to the original CC algorithm suggested above are assessed in this case study. The example was first discussed in Lekkas *et al*. (2001) where the cascade correlation neural networks (CCNN) were found to perform well in comparison with traditional ARMA and state-of-the-art transfer function methods of river flow routing. Here, the following different NN algorithms will be compared:

- SNNS backpropagation
- SNNS cascade correlation
- Modified cascade correlation
- Recurrent modified partial cascade correlation.

River flow data for 1996, 1997 and 1998 were obtained from the Environment Agency of England and Wales for a number of gauging stations located within the catchment of the River Trent, as shown in Figure 6.4.

The aim was to create models that could forecast the flow at Colwick with a lead-time of 12 hours. The size of the catchment upstream of Colwick is 7486 km^2. Drought during 1995 and 1996 (Smith & Crymble, 1998) means that the flows during this period were unusually low. Although 1997 saw a greater number of high flow events, the highest and most numerous flood peaks were observed in 1998. Therefore, in order to test the performance of the methods for significantly higher flows than those present in the calibration period, it was considered most informative to use the years 1996 and 1997 as calibration data, and to validate the models using the data from 1998. The Colwick flow time-series for all three years is plotted in Figure 6.5, which also shows the division of the data into training, testing and validation sets.

A correlation analysis was performed on the data, to identify suitable lags to be applied to each upstream gauging station time series in order to form the NN input patterns. The intention was to provide the models with a snapshot of the current ($t = 0$ hours) and antecedent ($t = -1, -2, \ldots -n$ hours) conditions

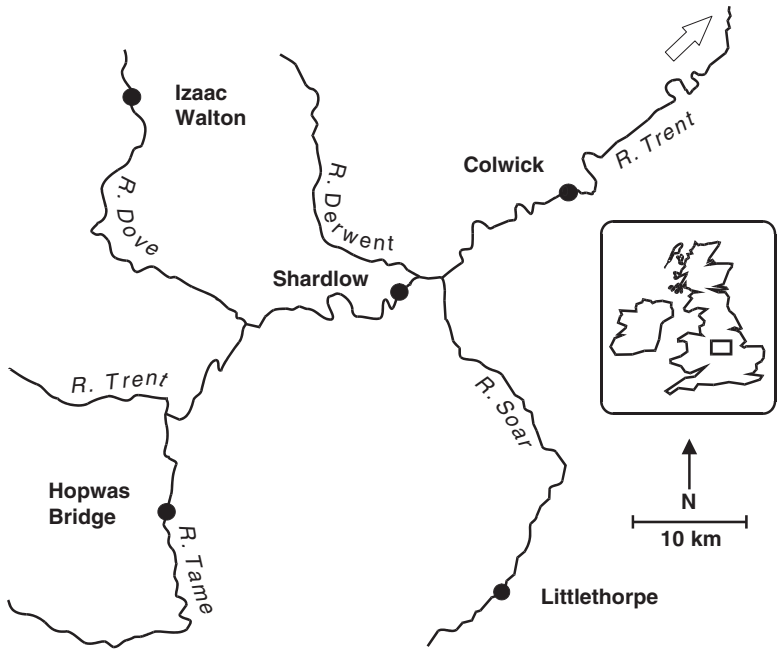


Fig. 6.4. Map of River Trent catchment showing positions of gauging stations.

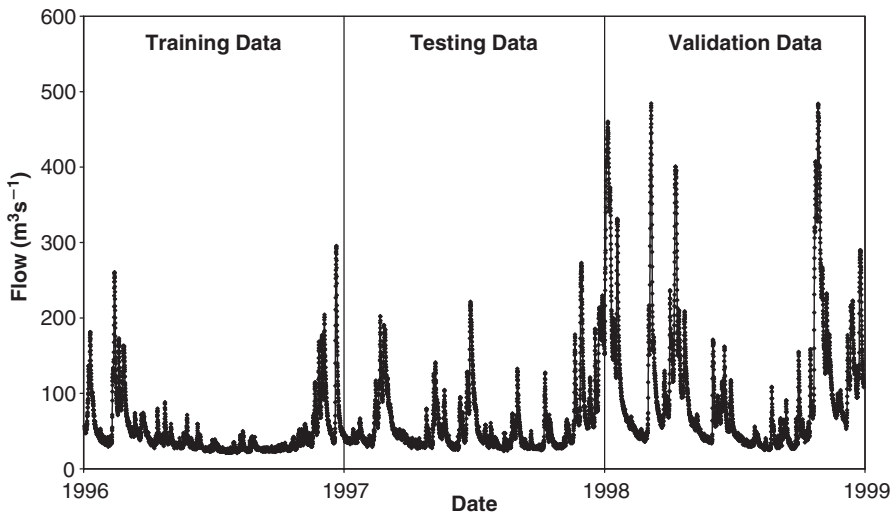


Fig. 6.5. Discharge data record for Colwick gauging station.

at each of the selected gauging stations, which could then be used to predict the flow at Colwick at $t = 12$ hours. For Hopwas Bridge and Izaac Walton, lags up to $t = -14$ hours were considered appropriate, whereas at Littlethorpe, which is closer to Colwick, lags up to $t = -12$ hours were used.

Four NN models were developed, based upon the input data described above. All the NN incorporated linear activation functions at the output layer.

- SNNS backpropagation: the traditional feed-forward BPNN was trained with the gradient descent method and used the cross-validation method implemented in SNNS. This involved a periodic quantification of the residual error over the test set data. When the test set error was observed to increase, the training procedure was stopped. The best BPNN obtained had one layer of 15 hidden units.
- SNNS cascade correlation: This was developed using the original CC learning architecture as implemented in SNNS. The software provides no means of performing automatic cross-validation and no attempt was made to do it manually. The maximum number of hidden units was limited to three, in a simple attempt to reduce the possibility of over-training. It should be noted that with CC the hidden units are connected to the previously installed hidden units, so that a network with three hidden units is comparable to one with three hidden layers containing one unit each, and therefore allows a greater level of model complexity.
- Modified cascade correlation: This implementation of the CC algorithm containing the procedures for automatic cross-validation developed above. The algorithm was set to install a maximum of twenty hidden units, and include five trials at each stage.
- Recurrent modified partial cascade correlation: The maximum number of hidden units was again set to twenty and five trials were performed at each stage. The algorithm was found to perform best when it included one recurrent output, that is, the forecast representing time $t + 11$ was appended to the input pattern for forecasting the flow at time $t + 12$.

6 RESULTS AND DISCUSSION

The overall performance of each model obtained was judged with respect to the validation data on the basis of the coefficient of efficiency, R^2, defined as follows:

$$R^2 = 1 - \frac{\sum_p (y_p - d_p)^2}{\sum_p (d_p - \bar{d})^2} \tag{4}$$

where y_p, and d_p are the model predictions and target values for each pattern (sample) p respectively, and $\bar{d}$ is the mean target output. The R^2 coefficient is a useful statistic in that it provides a measure of the proportion of variance that is explained by the model. The closer its value is to unity, the better the fit of the model.

The results obtained using each of the NN algorithms over the training, testing and validation data periods are presented in Table 6.1. It can be seen that in this application, there is little difference in the performances obtained using

Table 6.1. Flow forecasting results at Colwick with a lead-time of 12 hours.

NN Algorithm	R^2 (Training)	R^2 (Testing)	R^2 (Validation)
SNNS backpropagation	0.984	0.792	0.906
SNNS cascade correlation	0.991	0.762	0.917
Modified cascade correlation	0.963	0.892	0.955
Recurrent modified partial cascade correlation	0.962	0.897	0.961

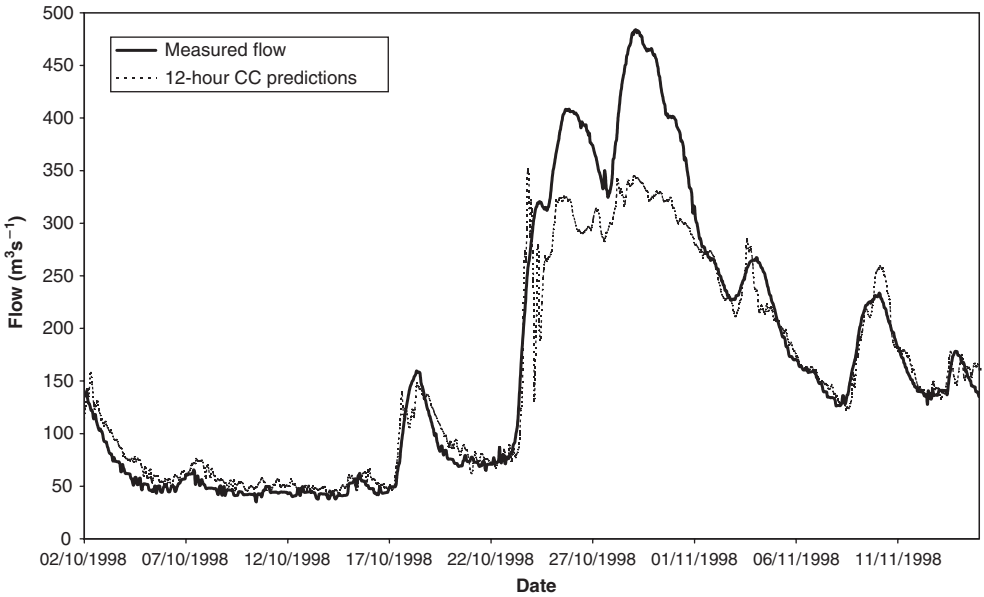


Fig. 6.6. Graph of measured flow against 12-hour predictions made by original cascade correlation NN.

the implementations of the commonly used backpropagation algorithm and cascade correlation. This suggests that the cascade correlation algorithm can be used as effectively as backpropagation to model river flow.

The BPNN and original CCNN give the poorest results with respect to the validation data set, while the modified CCNN give the best results. Correspondingly, the performance over the training data set is poorer with the modified CCNN than it is using the BPNN and original CCNN. It is possible that a modeller with more time and patience, might produce better models from these algorithms, but the results nevertheless emphasise the benefits of an effective and automated method for ensuring model generalisation.

The best model was obtained using the recurrent version of the modified cascade correlation algorithm, whereby the network's output is recycled as an additional input for the next prediction. Although the improvement here is small, the algorithm can be adapted so that the NN can continuously produce output when the input data form an irregular or incomplete series.

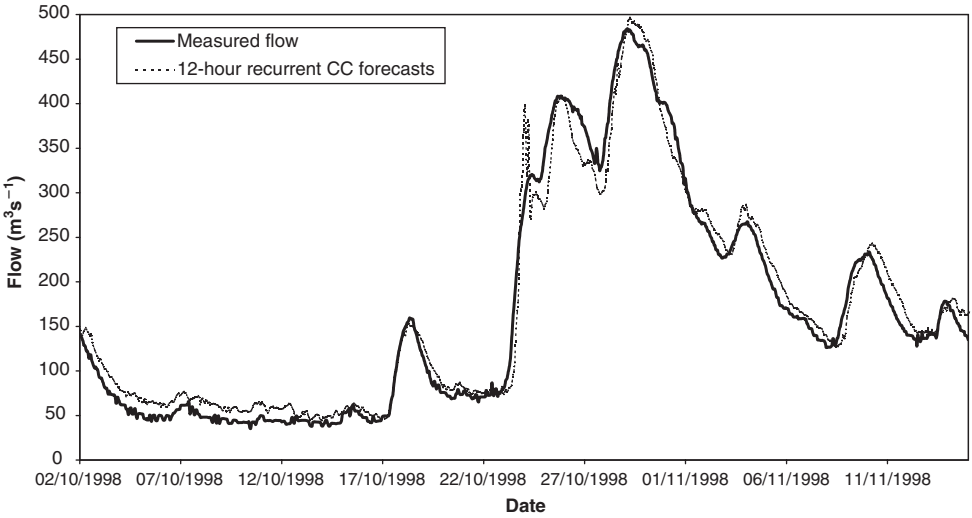


Fig. 6.7. Graph of measured flow against 12-hour forecasts made by recurrent modified CCNN.

Figure 6.6 shows the 12-hour flow predictions made by the original CCNN, while Figure 6.7 plots those made by the recurrent modified partial CCNN. Only a portion of the time series is shown for reasons of clarity. Although both models perform similarly during the period of low flow, the recurrent modified partial CCNN performs far better over the peak flows at the end of October. These results further demonstrate the importance of implementing a good generalisation procedure.

7 SUMMARY AND DISCUSSION

This chapter has described the cascade correlation learning architecture, and offers a number of possible modifications to render it more suitable for environmental modelling. The potential advantages of using CCNN are immediately clear in that by using a constructive algorithm there is no need to implement a trial and error procedure for finding the optimal network architecture. A further advantage was identified: that the structure of the CC network allows direct connections between the input and output units, which avoids saturation of the hidden units. This means that the resulting models are less likely to perform poorly when predicting events which lie outside the range of values included in the calibration data.

One of the factors that have caused cascade correlation to be largely overlooked in river flow forecasting is their reputation to be ill-suited to regression problems. This stems from the fact that the hidden units are prone to saturation. However, the modifications to the CC algorithm suggested above include early-stopping criteria, which should alleviate this problem to some extent. The case study demonstrates that when a suitable method for ensuring generalisation

was implemented, the algorithm could produce highly effective river flow prediction models.

ACKNOWLEDGEMENTS

The Environment Agency of England and Wales supplied the data used in this paper. The funding for this research was provided by the UK Engineering and Physical Sciences Research Council and by the European Commission.

REFERENCES

Abrahart, R.J. & Kneale, P.E. 1997. Exploring neural network rainfall-runoff modelling. *BHS National Hydrology Symposium*, Salford, UK, 9.35–9.43.

Abrahart, R.J., See, L. & Kneale, P.E. 1999. Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model. *Journal of Hydroinformatics* 1: 103–114.

Bebis, G., Georgiopoulos, M. & Kasparis, T. 1997. Coupling weight elimination with genetic algorithms to reduce network size and preserve generalization. *Neurocomputing* 17: 167–194.

Blanco, A., Delgado, M. & Pegalajar, M.C. 2000. A genetic algorithm to obtain the optimal neural network. *International Journal of Approximate Reasoning* 23: 67–83.

Campolo, M., Andreussi, P. & Soldati, A. 1999. River flood forecasting with a neural network model. *Water Resources Research* 35: 1191–1197.

Dawson, C.W. & Wilby, R. 1998. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Science Journal*, 43: 47–66.

Elman, J.L. 1988. Finding structure in time. *CRL Technical Report 8801*, University of California at San Diego, Centre for Research in Language.

Fahlman, S.E. 1988. An Empirical Study of Learning Speed in Back-Propagation Networks. *Carnegie Mellon University Technical Report CMU-CS-88-162*.

Fahlman, S.E. & Lebiere, C. 1990. The Cascade Correlation Learning Architecture. *Carnegie Mellon University Technical Report CMU-CS-90-100*.

Fahlman, S.E. 1991. The Recurrent Cascade Correlation Architecture. *Carnegie Mellon University Technical Report CMU-CS-91-100*.

Fausett, L. 1994. *Fundamentals of Neural Networks – Architectures, Algorithms and Applications*. New Jersey: International Editions, Prentice Hall.

Hasenauer, H., Merkl, D. & Weingartner, M. 2001. Estimating tree mortality of Norway spruce stands with neural networks. *Advances Environmental Research* 5: 405–414.

Hecht-Nielsen, R. 1987. Kolmogorov's mapping neural network existence theorem. *Proceedings of the 1st IEEE International Joint Conference on Neural Networks*, San Diego, California.

Hirose, Y., Yamashita, K. & Hijayi, S. 1991. Back-propagation algorithm which varies the number of hidden units. *Neural Networks* 4: 61–66.

Hsu, K., Gupta, H.V. & Sorooshian, S. 1995. Artificial neural network modeling of rainfall-runoff process. *Water Resources Research* 31(10): 2517–2530.

Hwang, J., You, S., Lay, S. & Jou, I. 1996. The cascade correlation learning: A projection pursuit learning perspective. *IEEE Transactions Neural Networks* 7: 278–289.

Jordan, M.J. 1986. Attractor dynamics and parallelism in a connectionist sequential machine. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society* (pp. 531–546). Hillsdale, NJ: Erlbaum .

Imrie, C.E., Durucan, S. & Korre, A. 2000. River flow prediction using artificial neural networks: generalisation beyond the calibration range. *Journal of Hydrology* 233: 138–153.

Karnin, E.D. 1990. A simple procedure for pruning back-propagation trained neural networks. *IEEE Transactions Neural Networks* 1: 239–242.

Karunanithi, N., Grenney, W.J., Whitley, D. & Bovee, K. 1994. Neural networks for river flow prediction. *Journal of Computing in Civil Engineering* 8: 201–219.

Le Cun, Y., Denker, J.S. & Solla, S.A. 1990. Optimal Brain Damage. In D.S. Touretzky (Ed.), *Advances in Neural Information Processing Systems*, 2: 598–605. San Mateo, CA: Morgan Kaufmann.

Lee, C.W. 1997. Training feedforward neural networks: an algorithm giving improved generalization. *Neural Networks* 10: 61–68.

Lekkas, D.F., Imrie, C.E. & Lees, M.J. 2001. Improved non-linear transfer function and neural network methods of flow routing for real-time forecasting. *Journal of Hydroinformatics* 3: 153–164.

Liang, H. & Dai, G. 1998. Improvement of cascade correlation learning algorithm with an evolutionary initialization. *Information Systems* 112: 1–6.

Maier, H.R. & Dandy, G.C. 2000. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling Software* 15: 101–124.

Maren, A., Harston, C. & Pap, R. 1990. *Handbook of Neural Computer Applications*. San Diego, CA: Academic Press.

Miller, G.F., Todd, P.M. & Hegde, S.U. 1989. Designing neural networks using genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms and their Applications* (pp.379–384). San Mateo, CA: Morgan Kaufmann.

Minns, A.W. & Hall, M.J. 1996. Artificial neural networks as rainfall-runoff models. *Journal of Hydrological Science* 41: 399–417.

Minns, A.W. & Hall, M.J. 1997. Living with the ultimate black box: more on artificial neural networks. *BHS 6th National Hydrology Symposium*, Salford, UK, 9.45–9.49.

Muttiah, R.S., Srinivasan, R. & Allen, P.M. 1997. Prediction of two-year peak stream discharges using neural networks. *Journal of American Water Resources. Association* 33: 625–630.

Phatak, D.S. & Koren, I. 1994. Connectivity and performance tradeoffs in the cascade correlation learning architecture. *Technical Report TR-92-CSE-27*, ECE Dept., University of Massachusetts.

Prechelt, L. 1997. Investigation of the CasCor Family of Learning Algorithms. *Neural Networks* 10: 885–896.

Reidmiller, M. & Braun, H. 1993. A direct adaptive method for faster back-propagation learning: the Rprop algorithm. In *Proceedings of the IEEE International Conference on Neural Networks*, San Francisco, CA, 586–591.

Rumelhart, D.E., Hinton, E. & Williams, J. 1986. Learning internal representation by error propagation. *Parallel Distributed Processing*, 1: 318–362.

Russell, S.J. & Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice-Hall International, Inc.

See, L., Corne, S., Dougherty, M. & Openshaw, S. 1997. Some initial experiments with neural network models of flood forecasting on the River Ouse. *Proceedings of the 2nd International Conference on GeoComputation*. Dunedin, New Zealand.

See, L. & Openshaw, S. 1998. Using soft computing techniques to enhance flood forecasting on the River Ouse. *Proceedings of the 1998 Hydroinformatics Conference*, Copenhagen, Denmark.

Setiono, R. & Hui, L. C. K. 1995. Use of a quasi-Newton method in a feedforward neural-network construction algorithm. *IEEE Transactions Neural Networks* 6: 273–277.

Smith, D.J. & Crymble, S. 1998. Water quality considerations in developing a new resource for water supply within the Midlands of England. *Water Science and Technology* 38: 201–208.

SNNS Group. 2002. Stuttgart Neural Network Simulator. Online http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/snns.html

Tchaban, T., Taylor, M.J. & Griffin, J.P. 1998. Establishing impacts of the inputs in a feedforward neural network. *Neural Computing and Applications* 7: 309–317.

Wang, D., Liu, X. & Ahalt, S. C. 1996. On temporal generalization of simple recurrent networks. *Neural Networks* 9: 1099–1118.

Weigend, A.S., Rumelhart, D.E. & Huberman, B.A. 1990. Predicting the future: A connectionist approach. *Int. Journal of Neural Systems* 1: 193–203.

Yao, X. 1993. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems* 8: 539–567.

Zealand, C.M., Burn, D.H. & Simonovic, S.P. 1999. Short term streamflow forecasting using artificial neural networks. *Journal of Hydrology* 214: 32–48.

7

The Use of Partial Recurrent Neural Networks for Autoregressive Modelling of Dynamic Hydrological Systems

HENK F.P. VAN DEN BOOGAARD
WL|Delft Hydraulics, Delft, Netherlands

ABSTRACT: This chapter deals with partial recurrent neural networks and their application to the non-linear autoregressive modelling of dynamic systems. This particular type of neural network can be seen as a data-driven model in state space, where a standard neural network model is used for the description of the non-linear transfer function. Each solution will thus possess the same generic form of time propagation mechanism that appears in conceptual dynamic models and linear black box models of type ARMAX. In this chapter, the theoretical background to partial recurrent neural networks is provided, together with two applications that demonstrate the practical relevance of these tools for modelling dynamic hydrological systems.

1 INTRODUCTION

In numerous disciplines related to applied science and technology, an increasing trend is observed towards the use of artificial neural networks (NN) for non-linear modelling, control, optimisation, design, data analysis and classification. For recent applications in areas such as meteorology, oceanography, hydraulics, hydrology and ecology the reader is referred to: Wüst (1995), Minns (1996), Minns and Hall (1996), Scardi (1996), Abrahart and Kneale (1997), Recknagel *et al.* (1997), Clair and Ehrman (1998), Hsieh and Tang (1998), Lange (1998), Sanchez *et al.* (1998), Shen *et al.* (1998), Van Gent and Van den Boogaard (1998), Wen and Lee (1998) or See and Openshaw (1999). NN are in most cases used to perform an identification of input–output relations and multi layer perceptrons (MLP) or radial basis function networks

(RBFN) are the two most popular types of tool for studies; see Haykin (1994), Beale and Jackson (1990) or Abrahart (Chapter 2).

Data-driven modelling, which includes NN, is often used when there is a lack of system understanding or process knowledge that prevents the development of *conceptual* models. Conceptual modelling requires system description and physical principles that must be formulated and quantified. The black box approach, with an appropriate level of robustness, can provide a more flexible and efficient solution with associated savings – in terms of time, cost and effort – for defining and refining the detailed information that is required to implement a conceptual model. NN can be used either as part of a conceptual model or to establish an alternative standalone solution. NN can also offer an attractive technique for emulating existing conceptual models since such tools will perform faster digital processing at higher computational speeds. Emulators make good sense (i) when for some reason a large number of model evaluations must be carried out, e.g. to perform sensitivity or uncertainty analysis, scenario evaluation, risk assessment, optimisation, or inverse modelling and/or (ii) when single model runs must be super fast e.g. to perform rapid assessment, decision support, real-time forecasting, or operational management and control. For examples of model emulation in practice see Solomatine and Avila Torres (1996) and Proaño *et al.* (1998).

The flexibility, efficiency and emulation capacity of data-driven models is particularly relevant within the fields of hydrology, meteorology and hydraulics. However, in these disciplines, an important modelling issue is the need to deal with dynamic systems and processes that can evolve over time. This leads to a serious limitation for 'standard' NN – these models are static and there is no explicit mechanism that accounts for time propagation or facilitates 'system memory' in terms of previous states and past events. Indeed, in both network training, and network application operations, the input-output patterns are processed in an independent manner such that the modelling response to previous input patterns has no direct influence on the modelling response to current input patterns. The input-output data set must be available beforehand and the order of pattern presentation is not important although random order is preferred. This situation also applies to common linear or non-linear regression methods and to interpolation techniques where curves or manifolds are fitted to a given set of data points. Thus 'standard' NN are in fact little more than universal function approximators (Cybenko, 1989; Hornik *et al*., 1989) and as such form part of a set of generalised regression techniques. Such tools are static non-linear regression models that offer no explicit consideration of time, past events, time evolution and/or the complex interaction that should occur between different or sequential patterns of input-output material. This problem of adopting static solutions for time series modelling is also addressed in other chapters; Chapter 5, for instance, considers the use of time delay neural networks (TDNN).

For dynamic systems, neither dependencies, nor the interaction of input-output patterns should be ignored. This is perhaps most recognisable in the state space (or phase space) mechanism of dynamic models. In discrete time, a state

space model is characterised with a time propagation mechanism, wherein the system state at time t is based upon a combination of the system state at time $t - 1$ and/or previous time steps, together with the set of external forcings that impact on this system. In continuous time, typical state space models will have the form of one or more coupled partial differential equations, such as flow models, based on the Saint Venant equations, or transport models, which use advection diffusion equations.

In this chapter the state space mechanism of dynamic models is extended into the neural modelling environment to obtain data-driven models that are better suited to non-linear modelling and for the analysis of dynamic systems and time series data. This extension results in the development of partial recurrent neural networks (PRNN). Such models are equipped with a time propagation mechanism that involves feedback of *computed* outputs as opposed to *observed* outputs. Sections 2.1 and 2.2 consider the state space properties of discrete time dynamic conceptual models and linear black box models of type ARMAX. PRNN are discussed in Section 3 and their practical relevance is demonstrated using two hydrological applications in Section 4. Section 5 contains summarised conclusions and provides some suggestions for related research and development in this field.

2 THEORETICAL BACKGROUND

2.1 *Dynamic conceptual models in discrete time*

Many dynamic conceptual models (as used in the fields of oceanography, meteorology, hydraulics, hydrology, ecology) are state space mechanisms that offer state space solutions. For continuous spatial and temporal coordinates the evolution of the 'system state' is governed by one or more coupled partial differential equations. The usual order of presentation in such equations is: left-hand side contains first and/or higher order temporal derivatives for all state variables; right-hand side contains all other model terms.

The complex nature and extent of these partial differential equations will in most instances prevent the derivation of analytical solutions. Difficulties will often arise from, or be related to, the occurrence of marked non-linearities, non-constant and/or non-uniform coefficients, irregular spatial domain, etc. Therefore, one must instead turn to numerical techniques, where the differential equations are discretised in terms of temporal and spatial coordinates. This results in a discrete time model that is of the following generic form:

$$\vec{X} = \Phi_t\left(\vec{X}_{t-1}, \ldots, \vec{X}_{t-2}, \ldots, \vec{X}_{t-M}; \vec{U}_t, \vec{U}_{t-1}, \ldots, \vec{U}_{t-N} \,\middle|\, \vec{\Theta}\right) \qquad (1)$$

The discretised model of Equation 1 is in state space form and can be solved through numerical integration of the state space equations. The output vector $\vec{X}_t$ denotes the system state at discrete time t. It consists of the state variables at all spatial grid points and could have a large dimension. Note that the 'old' states $\{\vec{X}_{t-1}\}_{m=1}^{M}$, on the right hand side of Equation 1, can be seen as part of the model input which will be propagated to the 'new' state $\vec{X}_t$ at time t.

The orders M and N in the discretised model are related to the order of the temporal derivatives in each partial differential equation. If present, partial derivatives with respect to the spatial coordinates, are absorbed in the transfer function $\Phi_t(\cdot)$ as well as in the model coefficients and in the other terms of this equation.

The vectors $\{\vec{U}_{t-1}\}_{n=1}^{N}$ are also inputs for the model and represent the non-autonomous part of the equation. These vectors originate from one or more external system forcings: wind drag in flow or wave models, sources or sinks in transport models, etc. It is also common for boundary conditions to be modelled as external inputs.

The (non-linear) function $\Phi_t(\cdot)$ is a transfer function that governs the time propagation of the system state from discrete time $t-1$ to time t. For conceptual models this function should be derived from physical principles such as the laws of conservation with respect to mass, momentum, energy, heat, etc. The transfer function $\Phi_t(\cdot)$ thus attempts to incorporate all system and physical knowledge, although such activities will to some extent also depend on the degree to which encapsulation is possible, and on the numerical scheme that was adopted for the discretisation of each continuous process.

The vector $\vec{\Theta}$ represents one or more uncertain parameters in the conceptual model. These uncertainties may include unknown parameters in the bed and/or surface friction coefficients, parameters in dispersion, reaction or exchange coefficients, parameters in the boundary or initial conditions, etc. Such parameters are in practice often determined from calibration based on observations of the model state. For conceptual models the number of unknown parameters will in general be small compared to the dimension of the system state and/or the total number of state variables. The calibrated model is in consequence, still for the most part based on physical principles and system knowledge, and to a minor extent on data.

2.2 *Linear black box models in discrete time (ARMAX)*

ARMAX (Autoregressive Moving Average with eXterior or eXogeneous input) models are often used to provide linear dynamic black box solutions for the processing and analysis of time series data. For a general discussion on such matters see Ljung and Söderström (1983); for hydrological applications of ARMAX models see Gourbesville and Lecluse (1994). In discrete time, and for a univariate case, such models will have the following form:

$$X_t = \sum_{m=1}^{M} \alpha_m \cdot X_{t-m} + \sum_{n=0}^{N} \beta_n \cdot U_{t-n} + \sum_{i=0}^{I} \gamma_i \cdot Z_{t-i} \qquad (2)$$

where X_t is the autoregressive model output, U_t is the deterministic external forcing input, and Z_t is a write random process with zero mean that is included to account for model uncertainties, i.e. Z_{t_1} and Z_{t_2} are independent for $t_1 \neq t_2$. This random noise represents non-modelled sub-grid processes, discretisation errors, or errors in the external forcings.

The model in Equation 2 is time invariant if its parameters $\vec{\Theta} = \{\alpha_m, \beta_n, \gamma_i\}$ are constant over time. Generalisation of this univariate ARMAX model to a multivariate (and/or time variant) system is straightforward, leading to:

$$\vec{X}_t = F_t\left(\begin{matrix}\vec{X}_{t-1}, \vec{X}_{t-2}, \ldots, \vec{X}_{t-M}; \; \vec{U}_t, \vec{U}_{t-1}, \ldots, \vec{U}_{t-N}; \\ \vec{Z}_t, \vec{Z}_{t-1}, \ldots, \vec{Z}_{t-I} \,\big|\, \vec{\Theta}\end{matrix}\right) \tag{3}$$

This is again a state space model, albeit that the revised model is stochastic, and involves a *linear* transfer function $F_t(\cdot \mid \vec{\Theta})$. So, in contrast to the conceptual model that is described in Section 2.1, all aspects of the transfer function are derived from parameterisation and nothing is based on physical principles. The end product from this data-driven modelling operation could also contain a large number of unknown parameters $\vec{\Theta}$, that must be identified using calibration procedures, based on an appropriate set of observed input-output patterns. For further details and methods of parameter identification on ARMAX models the interested reader is again referred to Ljung and Söderström (1983).

3 PARTIAL RECURRENT NEURAL NETWORKS

NN with feedback connections are not new and can be labelled as either 'recurrent networks' or 'partial recurrent networks'; recurrent networks exhibit full connection between each node and all other nodes in the network, whereas partial recurrent networks contain a limited number of specific feedback loops. The term 'lattice networks' is also used when source nodes are included that supply external inputs. Haykin (1994) notes the interest in such networks for dealing with time varying inputs or outputs; recurrent or sequential networks are also considered by Hertz *et al.* (1989) who assign the name Jordan Networks, for a one step delayed feedback of outputs, and Elman Architecture when the feedback to the network input is from one or more hidden layers. These networks and their names originate from Jordan (1986) and Elman (1990).

It should also be noted that the author would prefer to use the expression 'autoregressive neural network'. This might help to retain (as much as possible) the common terms and meanings that are used in system theory, conceptual models and time series analysis. Data-driven approaches, which include all types of regression modelling, form a small subset of the total modelling spectrum – which varies from a full conceptual approach to a total data-driven approach – such that data-driven neurocomputation is not identified to be a toolbox of distinct techniques deserving specialist consideration. From this viewpoint the name autoregressive neural network might perhaps better emphasise the non-linear generalisation of linear autoregressive moving average models that are common tools in time series analysis e.g. ARMA or ARMAX. However, to be consistent with a book on neural network modelling techniques, the term partial recurrent neural networks will be used throughout this chapter.

3.1 *PRNN basics*

Equations 1 and 3 are closely related with regard to the mechanism that governs the time evolution of the corresponding models. Both are in state space form and involve the feedback of earlier computed previous system states. The transfer function in Equation 1 is based on physical principles – to the extent that such things are possible – and will often possess a strong non-linear component. In Equation 3 the transfer function is linear and must be identified from observed data. These concepts form the basis for the definition of a PRNN. The idea is to replace the transfer functions $\Phi_t(\cdot)$ and/or $F_t(\cdot)$, with an NN, which leads to the following dynamic model in discrete time:

$$\vec{X}_t = \mathrm{NN}\left(\hat{X}_{t-1}, \hat{X}_{t-2}, \ldots, \hat{X}_{t-M};\ \vec{U}_t, \vec{U}_{t-1}, \ldots, \vec{U}_{t-N} \,\middle|\, \vec{w} \right) \tag{4}$$

'NN', on the right hand side of this equation, serves as a black box model for the transfer function, which is of a standard input-output configuration. MLP or RBFN solutions would both offer a suitable architecture. The vector $\vec{w}$ represents the weights of the connections in the NN. These weights equate to the uncertain parameters of each model, analogous to $\vec{\Theta}$ in the conceptual model of Equation 1, or $\vec{\Theta}$ in the ARMAX model of Equation 3.

Equation 4 introduces a dynamic model in which the complete time propagation mechanism is derived from the process of parameterisation; no system understanding or physical process knowledge has been incorporated. The PRNN version of Equation 4, thus follows a data-driven approach, albeit that the state space concept which is inherent to generic dynamic conceptual models has been incorporated. PRNN models thus form an important extension to standard static NN models, and can be seen as a first step in importing the properties or components of conceptual models, into the domain of neurocomputation. This also provides some interesting opportunities for hybrid modelling (see Thompson & Kramer, 1994; Van den Boogaard & Kruisbrink, 1996).

PRNN must encapsulate the dynamic properties of each system that is considered in an accurate and efficacious manner. This could require the development of complex architectures, albeit that more complex architectures will contain a larger number of uncertain parameters $\vec{w}$. The dimension of $\vec{w}$ will, in particular, be much larger than the dimension of the model state, such that this large number of uncertain parameters is in contrast to the situation for conceptual models, where the number of uncertain parameters is in most cases quite small.

The uncertain parameters $\vec{w}$ must again be identified using training procedures, and given a fully parameterised representation of the transfer function, this calibration process will often require a large number of observed input-output patterns. The assessment of a suitable network architecture will also form an important aspect of such operations, since this item will be used to represent the transfer function, and as such the complexities of each dynamic model.

PRNN models, compared to ARMAX models, can also be viewed as a non-linear generalisation with respect to the latter type of linear black box model.

The PRNN model does in fact reduce to an ARMAX model if a linear NN is used to represent the transfer function. It must be noted, however, that random model components have so far been omitted from consideration in the PRNN model, i.e. the stochastic part of an ARMAX model. This then introduces a restriction to the deterministic approach that will in consequence possess no statistical representation of the model error. This important aspect of the development process is considered in more detail in Section 3.5.

If possible, the orders M and N of the (auto)regressive parts of the modelling input, should be determined on the basis of system knowledge or from correlation analysis applied to the different processes and lag times involved. However, a proper (and minimal) estimate for both factors can also be obtained through the process of calibration.

The components $X_t^{(k)}$ and $U_t^{(l)}$ of the K-dimensional state vector $\vec{X}_t$ and L-dimensional external forcing $\vec{U}_t$ will in practice often refer to processes of different origin, type or character. Therefore, the (minimum required) orders of M and K can differ for the individual components suggesting different 'optimal' orders M_k for $X_t^{(k)}$ and N_l for $U_t^{(l)}$. To make the model as 'compact' as possible, it also makes sense to take these different orders into account, and the PRNN model of Equation 4 can thus be further generalised to the following form:

$$\begin{pmatrix} X_t^{(1)} \\ X_t^{(2)} \\ \cdot \\ \cdot \\ \cdot \\ X_t^{(k)} \\ \cdot \\ \cdot \\ \cdot \\ X_t^{(k)} \end{pmatrix} = \mathrm{NN} \begin{pmatrix} X_{t-1}^{(1)}, X_{t-2}^{(1)}, \cdots, X_{t-M_1}^{(1)}, \\ X_{t-1}^{(2)}, X_{t-2}^{(2)}, \cdots, X_{t-M_2}^{(2)}, \\ \cdot, \cdot, \cdots, \cdot, \\ X_{t-1}^{(K)}, X_{t-2}^{(K)}, \cdots, X_{t-M_K}^{(K)}, \\ U_t^{(1)}, U_{t-1}^{(1)}, \cdots, U_{t-N_1}^{(1)}, \\ U_t^{(2)}, U_{t-1}^{(2)}, \cdots, U_{t-N_2}^{(2)}, \\ \cdot, \cdot, \cdots, \cdot, \\ U_t^{(L)}, U_{t-1}^{(L)}, \cdots, U_{t-N_L}^{(L)} \mid \vec{w} \end{pmatrix} \tag{5}$$

In this case the architecture of a (standard) NN consists of $\sum_{k=1}^{K} M_k + \sum_{l=1}^{L}(1 + N_l)$ neurons in the input layer and K neurons in the output layer. Not all components $X_t^{(k)}$ of the output vector $\vec{X}_t$ must be autoregressive. For a non-autoregressive component k, the order M_k is 0, and this output is not then subject to feedback at later time steps.

Figure 7.1 illustrates a 'simple' PRNN model in which the basic NN is of type MLP. The model output $\vec{X}_t$ is two-dimensional, the first output component $X_t^{(1)}$ is order two autoregressive, and the second output component $X_t^{(2)}$ is non-autoregressive. Therefore $K = 2, M_1 = 2$ and $M_2 = 0$. The external input $\vec{U}_t$ of the model is one dimensional, but is order two regressive, so that $L = 1$ and $N_1 = 2$.

This figure shows that a PRNN model can also be viewed as a complex NN. The latter contains a large number of layers, part of the output from certain layers is not fed forward to other layers, whilst other parts of the output skip

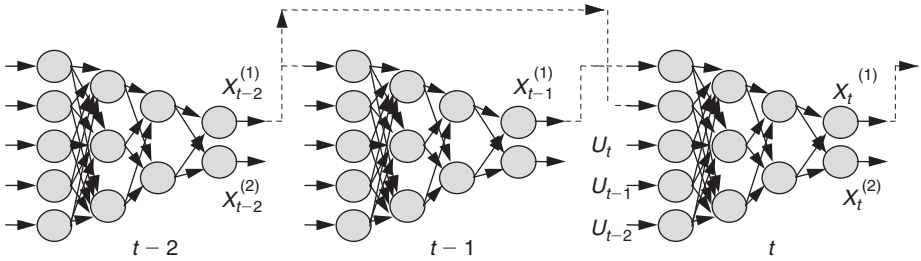


Fig. 7.1. The architecture of a partial recurrent neural network.

over adjacent layers and are fed forward to non-adjacent layers. This architecture differs from a feedforward solution, with a standard set of connections, where the outputs from each node in each layer are always fed forward to all nodes in the next layer and not to nodes in other layers.

Summarising, the PRNN of Equation 5 is found after embedding the time propagation mechanism of dynamic conceptual models into a data-driven modelling environment – such that a non-linear black box model with state space properties is obtained. PRNN models are thus seen to be a non-linear generalisation of ARMAX models. Some other important aspects of the PRNN model will be discussed in Section 3.

The main issue in PRNN is the feedback of outputs, from the NN at time t, to provide inputs at time $t + 1$ and/or later. This feedback might also involve self-feedback loops, at the level of an individual neuron, albeit that in the reported applications no such loops were present when hidden layers existed. Thus, seen in time, there is a strict direction for subsequent feedback such that at each time step the output is an explicit function of the input and the 'numerical scheme' is in this regard explicit rather than implicit.

3.2 *Open loop versus closed loop systems*

The PRNN model is a closed loop system where a 'new' system state follows from one or more system states computed at one or more previous time steps. Under certain conditions, a closed loop architecture can be reorganised into an equivalent open loop arrangement, and in this situation the modelling input will involve external forcings alone. The model in Equation 4 will thus become:

$$\vec{X}_t = \mathrm{NN}\left(\vec{U}_t, \vec{U}_{t-1}, \ldots, \vec{U}_{t-N} \mid \vec{w}\right) \tag{6}$$

The model is no longer autoregressive and 'system memory' in terms of previous states and past events must therefore be made available using a sufficient set of historical time series forcings N'. However, the required series could be infinite, as can be verified from the following one-dimensional, first order,

time invariant and autoregressive model:

$$X_t = \Theta \cdot X_{t-1} + U_t\left(|\Theta| < 1\right) \;\leftrightarrow\; X_t = \sum_{n=0}^{\infty} \alpha_n \cdot U_{t-n} \qquad (7)$$
$$\text{with } \alpha_n = \Theta^n$$

The order N' in an open loop representation can indeed become infinite even when the order N in the closed loop architecture is finite or small. However, in practice, a fair approximation will often be possible with a finite order N', although this N' will still be much larger than the sum of the orders K and N, in the original autoregressive model. The number of parameters in the open loop model will also be much larger, which can significantly handicap parameter identification, and in consequence decrease the predictive capabilities of the calibrated model. Hence the preference for a closed loop solution.

The use of an autoregressive or closed loop model, under certain conditions, will often be more of a necessity than an option. Here one must think of dynamic systems with an infinite memory, leading to non-stationary processes and responses, where initial conditions are not 'gradually forgotten'. Let us consider the case of a simple and pure (non-leaky) integrator. In mathematical terms, this system is described in the model of Equation 7 when $\Theta = 1$. In hydrological terms, it represents the filling of a reservoir with no output mechanism, although this configuration is not very realistic and some form of discharge mechanism must in practice be included. For instance, a weir is a simple passive emptying mechanism, which produces an overflow when the upstream water depth exceeds its crest. The overflow is then a non-linear function of water depth, and for this modified and more realistic reservoir model, it can be verified that an autoregressive tool is a better-suited solution than an open loop description. Proaño *et al*. (1998) dealt with such overflows in sewerage systems and used a PRNN for the emulation of a detailed conceptual model. Their approach and results are reviewed and further discussed in Section 4.1.

3.3 *Calibration of a PRNN using adjoint modelling*

PRNN encapsulate a standard NN that is applied at each time step and which will, through recursion, generate a temporal sequence of changes in the model state over time. However, due to the occurrence of feedback over one or more time steps, the application of common training approaches based on backpropagation of error must be customised to provide an appropriate generalisation. The adjoint formalism that is used for the calibration of large scale deterministic dynamic models is well suited to such a task. The advanced calibration or data assimilation facilities that have been developed for, and applied to, deterministic numerical models can thus be applied to PRNN, which is to be expected since the latter tool is a direct adaptation from the generic structure of such earlier modelling methodologies.

Model calibration in general, and NN training in particular, involves the definition of a cost function or a goodness-of-fit criterion that is used to assess the level of (dis)agreement that occurs between model predictions and real

world observations. This cost function must be minimised with respect to the control variables. The control variables will in most instances entail a combination of uncertainties in model parameters, in initial conditions, or in external forcings. For large scale minimisation problems, gradient descent techniques offer a practical solution, provided that such gradients are not too difficult or too time consuming to extract. Efficient gradient descent techniques include conjugate gradient and quasi-Newton methods (see Press *et al*., 1989). Several tools are available to provide an estimate of the gradient at each stage in the calibration process; one important and efficient method is adjoint modelling. In this approach the desired gradient is obtained from the results of two model runs and is independent of the number of control variables. The first run involves the propagation of an 'original' model over the time interval whilst the second run involves the construction of a so called 'adjoint solution'. The latter is intended to be a tangent model of the original solution; it thus equates to a linear dynamic system that has the same spatial and temporal dimensions as the original model. However, in contrast to the original model, the adjoint solution must be solved backwards in time. The cost function gradient is expressed in the states and prediction residuals of the original model, which together with the states of the adjoint model, provides an exact set of analytical derivatives, as opposed to an imperfect numerical approximation. Each gradient, in association with previous gradients computed during earlier iterations, is then used to provide the inputs for a gradient descent operation that will find an update for the uncertain parameters and in so doing will provide a better performance for the model. The adjoint model is thus applied more than once, the updating procedure is repeated until the cost function is minimised, and the parameters are updated in batch mode, after each forward (model) and backward (adjoint) sequence and not at separate time steps.

Within the context of dynamic models, adjoint modelling is also known as variational data assimilation and for a detailed introduction to adjoint modelling and its practical significance with respect to the calibration of large scale dynamic numerical models the reader is referred to: Chavent (1980), Long (1989), Panchang and O'Brien (1990), Van den Boogaard *et al.* (1993), Lardner *et al.* (1993), or Ten Brummelhuis *et al.* (1993). The relevance of variational data assimilation techniques to NN models is also mentioned in Hsieh and Tang (1998).

The adjoint formalism can be derived for PRNN and such adjoint PRNN would in turn offer a generalisation of the backpropagation rule for standard MLP. The adjoint PRNN model reduces to standard backpropagation if the model is not autoregressive.

3.4 *Calibration of a PRNN using data insertion*

Static NN that model time series data often use a different training procedure to the one that is advocated in the work that is reported in this chapter. In such cases, the NN is fed with observations, rather than system states that were computed in previous time steps so that the model is propagated and calibrated

according to:

$$\vec{X}_t = \mathrm{NN}\left(\hat{X}_{t-1}, \hat{X}_{t-2}, \ldots, \hat{X}_{t-M};\ \vec{U}_t, \vec{U}_{t-1}, \ldots, \vec{U}_{t-N} \,\middle|\, \vec{w}\right) \qquad (8)$$

Time propagation in the above equation is based on the process of data insertion. Equation 8 is in consequence not autoregressive and the solution thus equates to a 'one-step-ahead forecasting' operation. The 'advantage' of data insertion, in a strict sense, is that standard procedures can be used in the training process e.g. backpropagation, which is described in Chapter 2. The development and implementation of advanced calibration techniques that deal with strict autoregressiveness is thus avoided, e.g. the adjoint method, which is described in Section 3.3.

For the following reasons, however, it is argued that data insertion is often considered to be an improper form of modelling and/or data assimilation:

1. Replacing computed system states with their corresponding observations means that a user has no confidence in the model. Prediction of one or more future system states, based on auto/cross-correlations with observed data, could be an inaccurate method for longer time horizons. It would in consequence be reasonable to expect improved accuracies over longer prediction horizons from an autoregressive model that had been trained and applied in a strict and proper sense.
2. Data insertion is not an acceptable method of data assimilation because the true estimate of a 'system state' can never be more accurate than the last data point inserted. Data insertion gives no possibilities to average redundant information over time in order to reduce the effects of noise; it is thus impossible to improve the estimated 'system state'. Moreover, overwriting model states with observations can leave the model dynamically unbalanced. This is a particular problem when the observed data are sparse, noisy and/or inconsistent with the modelling predictions. Each insertion might typically inject bursts of noise into an evolving modelling solution with obvious consequences in the form of undesirable modelling results. For a more detailed discussion see Long (1989).
3. In practice, observed data sets are often sparse and irregularly distributed in space and time. With data insertion, outputs cannot be generated when one of the required antecedent observations is missing. Moreover, such gaps in the data set can significantly reduce the quantity and quality of the training and/or verification patterns and thus limit applications to those cases where state space observations are not missing. True autoregressive models do not suffer from these limitations since such tools propagate computed rather than observed model states. In fact, this pinpoints the main issue of modelling, which should be to generate estimates of the full system state, and, in particular, for positions and times where observations are missing.
4. Data insertion introduces a strong difference between the use of a model in the calibration stage and the application of that same model to real-world forecasting operations. For the purposes of calibration each output is

replaced with a corresponding observation at the next time step; but this cannot be done in forecast mode, since future states are still an unknown factor, and there is no alternative other than to propagate the model as a strict autoregressive solution with feedback of computed outputs. Hence, calibration based on data insertion has optimised a model that is based on different conceptions, with respect to the one that is used for forecasting operations which could reduce the quality of forecasts.

3.5 *On-line or sequential data assimilation for PRNN models*

Data assimilation facilities are often used to improve the performance of (deterministic) dynamic conceptual models. The data insertion process, described above, represents a simple form of on-line sequential assimilation. However, in a dynamic environment, it is also important that each model is able to adapt to changing conditions and as such to improve its skill or power with respect to short or medium term forecasting horizons. Thus other factors must be considered and assimilated as each model proceeds from time step to time step, such that the consistent integration of changing conditions, within an adaptive solution, will produce an operational product that is superior in comparison to a corresponding non-adaptive deterministic solution.

Even a well-calibrated standard model will not provide a perfect set of predictions in forecast mode and such predictions will in most cases become less accurate over modelled time. Gradual increases in error will arise from errors in the calibration data set, unresolved modelling miscalculations, fluctuations or long term trends in the model parameters and/or external forcings. These uncertainties can be described as changing internal or external conditions, such that to provide improved forecasting power, it is essential that the model must adapt to pertinent trends, oscillations or environmental switches. To allow for this, and to make the model better suited to operational and/or real time prediction, on-line or sequential data assimilation techniques are important. The usual approach is to describe all model and observation uncertainties using random noise, such that the uncertainties are modelled in a statistical sense, as opposed to the imposition of a strict physical relationship. The original deterministic model is thus embedded in a stochastic environment and the actual process of data assimilation then involves the consistent integration of all sources of modelled or observed information so far available, whilst also taking the statistics of their uncertainties into account. Thus, following the integration of model and data, an optimal instantiation of the model is obtained for forecasting purposes. The most common procedure that is used to perform this type of data assimilation exercise is Kalman filtering; for theoretical background material see Gelb (1974), Jazwinski (1970) or Maybeck (1979); for applications in tidal flow models with emphasis on storm surge forecasting see Heemink (1986), Heemink and Kloosterhuis (1990) or Verlaan (1998).

Most on-line data assimilation facilities have been developed for, and applied to, dynamic conceptual models. PRNN, however, with their state space architecture, also provide a suitable basis to develop data-driven models that

are equipped with on-line data assimilation procedures. So, in addition to the inclusion of state space form, another important and substantial component of dynamic conceptual models is imported into the sphere of neurocomputation. This will provide new and important opportunities with respect to the development of operational or real-time applications. For example, in hydrological modelling, real time flood forecasting, prediction of water loads in drainage systems, forecasting and controls of structures such as sluices, weirs or barriers, can be mentioned as relevant applications for on-line real-time or run-time data assimilation. Similar examples can be given for other disciplines where operational modelling is important.

From the above it is clear that for the integration of on-line data assimilation facilities, random noise must be incorporated to allow for the statistical modelling of uncertainties, in the form of extended and stochastic PRNN. Such mechanisms are not considered in this chapter but will require a much stricter generalisation of linear ARMAX models in comparison to deterministic PRNN models. For calibration and/or data assimilation procedures with respect to stochastic PRNN, the same mechanisms that are used for conceptual or ARMAX models, could also be used to develop neural solutions. Technical details on the construction of such tools can be found in Van den Boogaard *et al*. (2000).

4 PRNN APPLICATIONS

Two hydrological applications are presented. The first case involves using a PRNN to emulated a dynamic conceptual model of a sewerage system. In the second case a PRNN is applied to model the water balance of Lake IJsselmeer in the Netherlands.

4.1 *PRNN emulation of a sewerage system*

This section outlines the potential use of PRNN for model emulation. Further details on the reported application can be found in Proaño *et al.* (1998).

One major problem with respect to sewerage systems is the overflows that can and do occur during severe rain storm events. Sewerage overflow occurs when the capacities of either sewerage systems or treatment plants are exceeded during a rainfall event. The overflow devices are often nothing more than a storage chamber, with a weir that acts to control the flow, such that an overflow occurs when the upstream water depth exceeds its crest.

In the Netherlands, some 90% of all sewerage systems are combined systems, which spill diluted sewage into open water systems during extreme rainfall events. For most towns, these systems have either no slope or mild slopes, such that the resultant network of interconnected flows will contain a large number of loops. Moreover, when all street sewer lines are included, the total structure could contain several thousands or even tens of thousands of pipes.

Models based on a numerical solution to the Saint Venant equations can be used to simulate sewerage systems. The SOBEK-URBAN modelling system is

the main tool that is used to perform such simulations in the Netherlands; for a numerical description of this mechanism see Stelling (2000). The main point to note is that this tool can provide an integrated approach for the simulation of 1-D processes in rivers, sewers and drainage systems, where integrated means that flows in the pipe network and the receiving waters can be combined.

Legislation regarding sewage spilling into open water systems has in recent times placed more and more constraints on the frequencies and quantities of permitted overflow that is allowed to occur during storm events. Moreover, to deal with current legislation, and to assess whether or not a particular case satisfies the regulations, simulations must be performed for a 10 year period, and in the near future this period is expected to be extended from 10 to 25 years.

Simulation of sewerage systems over such a long period are performed on recorded series of historical rain storm events. For a current design, or one or more proposed rehabilitation designs (e.g. dealing with additional storage capacities), this implies that the set of simulated storm events must contain storms with the potential to produce overflows. However, for a time period of 10 to 25 years, this will require at least a few hundred such events.

For large cities, with several tens of thousands of pipes, the computational burden associated with such simulations will remain extreme, despite anticipated computer improvements in terms of faster processors and more powerful memories (Verwey, 1994). Even when restricted to subsets of events with potential overflow, the computational effort will still be large, in particular when numerous designs must be compared. It is for this reason that a number of alternative methods have been investigated; see Proaño *et al*. (1998) or Price *et al*. (1998). NN, under such circumstances, offer two possible opportunities: (i) to act as fast model emulators and/or (ii) to perform model reduction operations. PRNN, in contrast to static standard NN, would in both cases be expected to provide a more suitable solution since the solution would in both operations be required to encapsulate numerous dynamic systems and dynamic processes.

Proaño *et al.* (1998) used a PRNN for the dynamic emulation of overflow discharges. The external input consisted of rainfall time series data for a storm event. The output of the PRNN was a time series of overflow discharges at one or more overflow structures. This output must be zero when no overflow occurs. The PRNN was calibrated on a data set of rainfall-overflow combinations, the overflow time series was generated from a numerical conceptual model, and the output time series of a few nodes in the numerical model were emulated as opposed to the production of a solution that would model the complete state of the sewerage system. It was also ensured that the calibration and verification data sets, each contained rain storm events that produced overflow, as well as events without overflow.

It was important in the research to assess whether or not the emulation could be based on a limited subset of the large 'original' set of events. If not, then the input and output patterns for all events would have to be derived from the conceptual model, and the desired reduction of computational cost would not be achieved. In addition, an emulation would no longer make sense, since

the frequencies and quantities of overflow would be contained in the numerical model predictions.

This model emulation and reduction exercise was applied to the sewerage system of Maartensdijk, a town with about 10000 inhabitants, in the centre of the Netherlands. Based on system emptying time and storage/pump capacities, 200 potential overflow events were selected from a rainfall series of 10 years, and an overflow model produced from 89 of the 200 events.

The selection of representative calibration and validation data sets was based on scatter plots of three parameters that characterised the main features of the rainfall time series: storm duration, maximum rainfall depth and total rainfall depth. 20 overflow events and 7 non-overflow events were selected for the calibration set; a verification set of 17 events was likewise constructed. Then, for all 44 calibration and verification events, simulation with a SOBEK-URBAN model of the Maartensdijk sewerage system was carried out. This provided a time series of water depths and discharges at three overflow structures and from which a PRNN model was then constructed. This PRNN model was calibrated and verified in batch mode, which means that the connection weights were optimised for all training events in concert as a collective item, and that individual events were not treated as separate entities for model development purposes.

The external inputs for this PRNN consisted of current and antecedent rainfall samples. There were two outputs: current discharge and current water depth at the weir. The latter was needed because at such structures, the water depth determines the state of the system and not discharge. In fact, discharge is a rating curve function of the water depth, but this relation is not one-to-one, e.g. discharge is zero when water depth falls below the crest of a weir. Therefore, discharge cannot be used to represent the 'system state'. This also illustrates the fact that, within black box procedures, physical principles or system knowledge must often be used in order to ensure that there is a meaningful set up for each model.

Similar physical considerations, as well as test experiments, also revealed the need for PRNN models to deal with non-transient effects for the accumulation of rain depths into water depths. Open loop models (see Section 3.2) or standard NN models are thus seen to be less suitable approaches for the sewerage emulations reported herein.

Several training experiments were undertaken, a suitable architecture was assessed, and accurate PRNN emulations of the SOBEK-URBAN model obtained. Validation tests on events not contained in the training and verification data sets also demonstrated the capabilities of a PRNN to generalise to unseen data sets.

4.2 *Water balance in Lake IJsselmeer*

This section summarises the potential use of a PRNN to model water balance. The reported application is for Lake IJsselmeer in the Netherlands. Further details on this investigation can be found in Van den Boogaard *et al.* (1998).

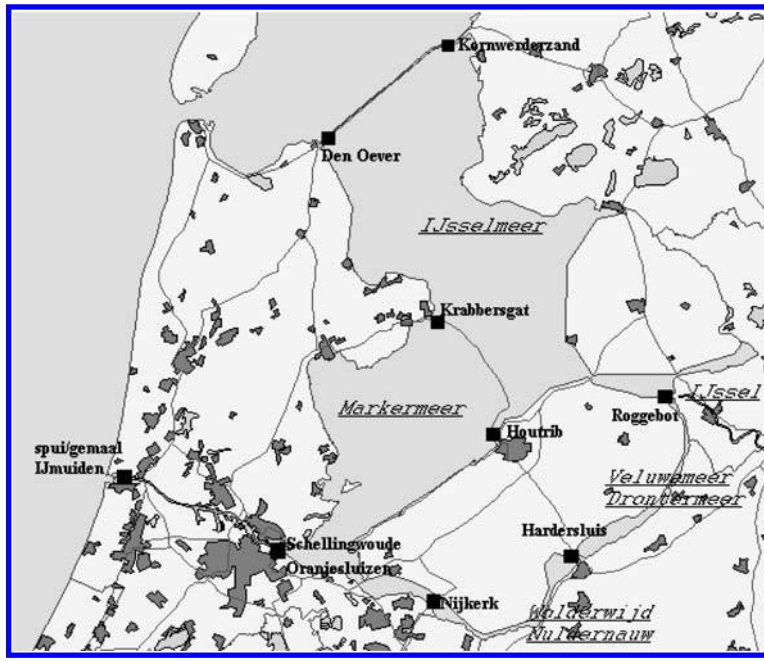


Fig. 7.2. Plan view of Lake IJsselmeer.

The most important factors that affect the water balance of Lake IJsselmeer are the inflow of the River IJssel and the outflows from the sluices at Den Oever and Kornwerderzand. Figure 7.2 provides a map of the area which marks the position of the River IJssel and the locations of both sluices. The sluices are used to discharge excess water and form an interface between the lake and the tidal Dutch Wadden Sea. Such spilling is possible during low tides, when water levels outside the lake are lower than water levels within the lake, and spill volume is estimated in terms of the difference between these two water levels. The wind can also have a significant effect on the difference between the inner water level and the outer water level.

Water balance modelling, in most instances, will involve building some type of autoregressive mechanism. This can be recognised from the manner in which the water volume V_t, changes over a time increment Δt, which can be expressed as:

$$\Delta V = V_{t+\Delta t} - V_t = (Q_{\text{in}}(t) - Q_{\text{out}}(t) \cdot \Delta t) \tag{9}$$

where $Q_{\text{in}}(\cdot)$ and $Q_{\text{out}}(\cdot)$ denote the inflow and outflow discharges. If the total area remains more or less constant, the water level h_t, will then evolve according to:

$$h_{t+\Delta t} = h_t = (q_{\text{in}}(t) - q_{\text{out}}(t) \cdot \Delta t) \tag{10}$$

Typical evolutions will occur in state space form. The transfer function will be non-linear when the relationship between the currents $q_{in}(\cdot)$ or $q_{out}(\cdot)$ and the water levels is non-linear. However, non-linear mechanisms will often impact on the inflows and outflows, and marked non-linearities are evident for the overflow at sluices.

In the reported work a PRNN was used to perform dynamic modelling of the water levels of Lake IJsselmeer as a function of five external system forcings: discharge of the River IJssel, the North-South and East-West components of the wind, and the outside tidal water levels at the sluices of Kornwerderzand and Den Oever. The output was the water level of Lake IJsselmeer. The time step in this model is $\Delta t = 1$ day, which corresponds to the sampling period of observed discharges for the River IJssel, and the sampling period of water levels for Lake IJsselmeer. Daily time series of the maximum wind speed and the minimum outer tidal water levels were also included in this modelling operation.

In this research, only winter seasons were considered, since the dynamic behaviour of the system at these times is the most interesting. The data set consisted of daily samples, for the period October 1 to March 31, taken from the 15 winter seasons between 1978/9 and 1992/3. From this data set, nine seasons were selected for training, whilst the other six seasons were used for testing. The training and testing data sets were each considered to be more or less representative of the whole data set and the chosen architecture was calibrated to find a set of weights $\vec{w}$ that provided the best model performance as follows. For each season, initial conditions were derived from one or more observations taken on October 1, and the model was then propagated in a strict autoregressive sense through to the following March 31. The standard least squares criterion provided a cost function and quantified the level of (dis)agreement between modelled and observed water levels. Figure 7.3 shows the performance of the calibrated solution for four representative periods. For all periods, including ones not shown here, a reasonable level of performance was obtained. However, during certain periods, considerable prediction errors were still present. These are thought to be related to errors in the external forcings and not a result of poor training. PRNN performance was also better than a conceptual water balance model that had been developed in earlier work – albeit that the actual difference was considered to be minor.

In Section 3.4 it was argued that data insertion is not a proper form of calibration for PRNN and the effect of data insertion on such models for Lake IJsselmeer is considered in Van den Boogaard *et al*. (1998). PRNN output, where the model is applied and calibrated in a strict autoregressive sense, was compared against the outcomes of the corresponding model trained with data insertion. For all periods in the training data set, the model based on data insertion reproduced the observed data with near-perfect levels of skill, and performed much better than the autoregressive solution trained with feedback of computed outputs. The weights of the model based on data insertion were then substituted into an autoregressive model that was used to predict the water levels for the periods of the verification data set. This model was run in prediction mode as an autoregressive forecaster and the outputs were found to be

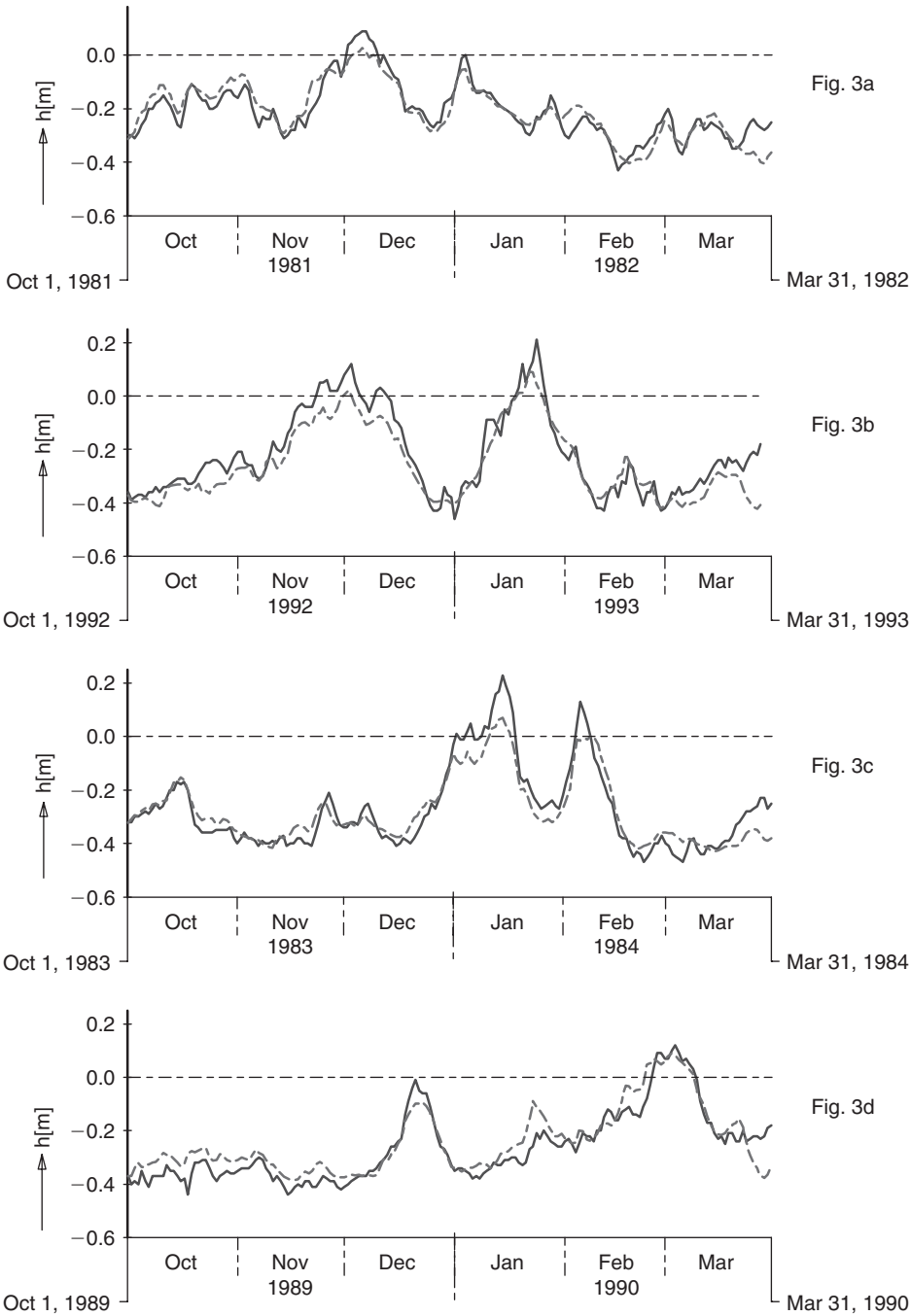


Fig. 7.3. Observed water levels (solid curves) and water levels computed with a calibrated PRNN (dashed curves) for 4 winter seasons. The seasons of Figures 7.3a,b were part of the calibration data set, while the seasons of Figures 7.3c,d were within the verification data set.

systematically less accurate than those produced from the corresponding autoregressive solution that had been trained with strict computed feedback.

The reported comparison was based on the root mean square error of both forecasts. Such findings support the theoretical arguments that strict autoregressive solutions will contain better predictive skills than models based on data insertion. The observed accuracies of the data insertion model for the training data set is misleading since this could reflect strong temporal correlations between the various observations. PRNN trained with feedback of computed outputs will, in general, better mimic the dynamics of the physical system and as such offer strong potential for superior modelling.

It should also be noted that major uncertainties were associated with the discharge figures for the River IJssel. These discharges were not direct observations but estimates derived from empirical relationships using water level registrations from the River Rhine at Lobith near the Dutch-German border. These relationships could contain errors, especially for winter seasons, when water levels and/or discharges are large or even extreme. In all other external forcings, such as the wind, large uncertainties will also have been present. For applications under operational or real time conditions, such as short to medium term water level forecasting or sluice controls, these uncertainties must be taken into account in a more explicit form. Section 3.5 argued that this could be done in a statistical sense, through the use of on-line data assimilation techniques, to improve the forecasting capabilities of the model. The application of on-line data assimilation to this situation is presented in Van den Boogaaard *et al.* (2000).

5 DISCUSSION AND CONCLUSIONS

This chapter has considered the role of PRNN – in hydrological science and for dynamic systems modelling. Each PRNN is a non-linear dynamic model in state space form where the transfer function is modelled with a standard NN. The complete time propagation mechanism is thus a parameterised solution that must be identified from observed data. Each PRNN is in essence a data-driven model, but with the incorporation of a state space architecture, an important generic aspect of dynamic conceptual models is included. PRNN models can be viewed as a non-linear generalisation of ARMAX models – which are common tools for the modelling and analysis of time series data. PRNN are also observed to be a dynamic extension of standard static NN, since the former is able to deal with time varying inputs and outputs, whereas the latter contains no explicit provision for 'system dynamics' or 'system memory'.

The state space mechanism of a PRNN involves the feedback of outputs computed in one or more previous time steps. This closed loop architecture can be written in an equivalent open loop form, leading to a non-autoregressive model. The transfer function in this situation can be modelled with a standard NN. However, to account for 'system memory', a much longer series of historical external forcings must be provided in the shape of numerous inputs. This

will produce a significant increase in the number of unknown parameters in terms of additional connection weights that will cloud the issue and reduce the power of the network to establish a clear model. For some solutions, the demand for historic factors could be infinite, which leaves little real choice other than to use an autoregressive model or PRNN. Indeed, this is not a theoretical issue, as can be seen from the sewerage experiment presented in Section 4.1.

The feedback of computed outputs requires a proper generalisation of the training rules that are used in a standard NN. It was also argued, that for the calibration of a PRNN, gradient descent techniques and adjoint modelling were appropriate. The latter is a variational data assimilation technique, which provides an extremely efficient and elegant means of computing the gradient of a cost function and is one that is widely used for the calibration of large scale meteorological models and for 2-D or 3-D hydraulic flow modelling. PRNN share a common state space architecture with large scale conceptual models, such that the adjoint method has direct application, and the adjoint formalism is in fact a generalisation of the standard backpropagation rule.

Sometimes an alternative training procedure is followed using data insertion. This means that real observations are used for feedback, rather than the actual model outputs, that were computed in one or more previous time steps. It was argued that, in general, this is not a proper form of modelling and an inappropriate mechanism for calibration or data assimilation purposes. In the case of data insertion, the modelling and/or forecasting is based on the temporal correlations of the observed processes, rather than the consistent identification of the physical mechanism that governs the time propagation operation. Each calibrated autoregressive solution represents a model in a strict sense that is better suited to forecasting operations. This was confirmed in the practical application provided under Section 4.2. In fact, data insertion should not be used for model calibration purposes, unless the circumstances are such that there is an absence of both model error and observation noise.

This chapter has considered deterministic PRNN. The implication of this is that both the model and the external forcings, which form the modelling input, are assumed to be perfect. In Section 4.1, these conditions are clearly satisfied, as the input-output is adopted from the results of a deterministic dynamic conceptual model. The PRNN was shown to be a feasible solution for model emulation and/or model reduction purposes. Such findings would be applicable to the general case as, within emulations, external forcings and observations are consistent and without uncertainties. Moreover, the number of observations can be as large as demanded or desired, and fast neural emulators of dynamic conceptual models will provide important opportunities for rapid assessment, real time control and management, scenario optimisation and uncertainty/sensitivity/risk analysis. However, the main problem still lies in finding the right architecture.

In Section 4.2 the deterministic PRNN was used to model the water balance of Lake IJsselmeer based on observed hydrological and meteorological data. For a test data set of winter seasons, the calibrated model produced reasonable water level predictions, although in certain periods the prediction errors were

still large and could well be due to uncertainties in the observations. Section 3.5 suggested that to achieve improved performance, such uncertainties should be taken into account, and that the uncertainties could be modelled in an explicit statistical manner. Sequential data assimilation techniques could then be used for the optimal integration of both model and observations. PRNN combined with sequential data assimilation facilities will also provide a robust solution in terms of errors and adaptation to changing conditions. This could lead to significant improvements in forecasting skill and further increase the potential benefits for operational and/or real time applications.

Sequential data assimilation techniques, applied to conceptual dynamic models, can also be imported into a PRNN. The state space form of such models means that this a simple operation; so, after the state space architecture and the adjoint formalism, this third important feature of dynamic conceptual models could also be embedded within a neural solution.

The other chapters in this book describe numerous feasible and adequate hydrological modelling applications based on standard static NN, but for forecasting purposes, such solutions are in most cases limited to the simple production of one-step-ahead outputs. Hence one future challenge will be to ascertain the extent to which this window of prediction or forecasting horizon can be extended with a PRNN, and, if so, what then would be the most appropriate architectures and training methods?

ACKNOWLEDGEMENTS

The author wishes to thank Arthur Mynett, Matthew Turner and Micha Werner of WL|DELFT HYDRAULICS for their support, stimulating discussions and valuable comments during the preparation of this manuscript. Important contributions to the PRNN application section were supplied by Oswaldo Proaño and Dilip Gautam of IHE-Delft.

REFERENCES

Abrahart, R.J. & Kneale, P.E. 1997. Exploring neural network rainfall-runoff modelling. *Proceedings Sixth National Hydrology Symposium, University of Salford, 15–18 September 1997*: 9.35–9.44.

Beale, R. & Jackson, T. 1990. *Neural Computing. An Introduction*. Bristol: IOP Publishing Ltd.

Chavent, G. 1980. Identification of distributed parameter systems: about the output least square method, its implementation, and identifiability. In R. Iserman (Ed.), *Proc. 5th IFAC Symposium on Identification and System Parameter Estimation.* I: 85–97. New York: Pergamon Press.

Clair, T.A. & Ehrman, J.M. 1998. Using neural networks to assess the influence of changing seasonal climates in modifying discharge, dissolved organic carbon, and nitrogen export in eastern Canadian rivers. *Water Resources Research* 34(3): 447–455.

Cybenko, G. 1989. Approximation by Superpositions of a sigmoidal function. *Math. Control Signals Systems* 2: 303–314.

Elman, J.L. 1990. Finding structure in time. *Cognitive Science* 14: 179–211.

Gelb, A. 1974. *Applied Optimal Estimation*. Cambridge, MA: The MIT Press.

Gourbesville, P. & Lecluse, Y. 1994. STORAGE: A Stochastic multimOdel pRocedure for dischArGe forEcasting. In Verwey, Minns, Babovic & Maksimovic (Eds.), *Hydroinformatics '94* (pp. 147–154). Rotterdam: Balkema.

Haykin, S. 1994. *Neural Networks. A Comprehensive Foundation*. New York: Macmillan Publishing Company.

Heemink, A.W. 1986. *Storm surge prediction using Kalman filtering*. Ph.D. Thesis, Enschede: Twente University of Technology.

Heemink, A.W. & Kloosterhuis, H. 1990. Data assimilation for non-linear tidal models. *International Journal for Numerical Methods in Fluids* 11: 1097–1112.

Hertz, J., Krogh, A. & Palmer, R.G. 1991. *Introduction to the Theory of Neural Computation*. Reading, MA: Addison Wesley.

Hornik, K., Stinchcombe, M. & White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2: 359–366.

Hsieh, W.W. & Tang, B. 1998. Applying neural network models to prediction and data analysis in Meteorology and Oceanography. *Bulletin of the American Meteorological Society*: 1855–1870.

Jazwinski, A.H. 1970. *Stochastic Processes and Filtering Theory*. Volume 64 of *Mathematics in Science and Engineering*. New York: Academic Press.

Jordan, M.I. 1986. Attactor Dynamics and Parallelism in a Connectionist Sequential Machine. In *Proceedings of the Eight Annual Conference of the Cognitive Science Society* (Amherst, Massachusetts): 531–546. Hillsdale: Erlbaum.

Lange, N.T.G. 1998. Advantages of unit hydrograph derivation by neural networks. In Babovic & Larsen (Eds.), *Hydroinformatics '98* (pp. 783–789). Rotterdam: Balkema.

Lardner, R.W., Al-Rabeh, A.H. & Gunay, N. 1993. Optimal estimation of parameters for a two-dimensional model of the Arabian Gulf. *Journal of Geophysical Research*. 98(C10): 229–242.

Ljung, L. & Söderström T. 1983. *Theory and Practice of Recursive Identification.* New York: MIT Press.

Long, R.B. 1989. *Notes on assimilating observations into numerical models*. Report Z274. Delft: WL|Delft Hydraulics.

Maybeck, P.S. 1979. *Stochastic Models, Estimation, and Control*. Volume 141-1 of *Mathematics in Science and Engineering*. London: Academic Press, Inc. Ltd.

Minns, A.W. 1996. Extended rainfall-runoff modelling using artificial neural networks. In Müller (Ed.), *Hydroinformatics '96* (pp. 207–213). Rotterdam: Balkema.

Minns, A.W. & Hall, M.J. 1996. Artificial neural networks as rainfall-runoff models. *Hydrological Sciences Journal* 41(3): 399–417.

Panchang, V.G. & O'Brien, J.J. 1990. On the determination of hydraulic model parameters using the adjoint state formulation. In A.M. Davies (Ed.), *Modelling Marine Systems*, Volume I, Chapter 2: 5–18. Boca Raton, FL: CRC Press, Inc.

Press, W.H., Flannery, H., Teukolsky, S.A. & Vetterling, W.T. 1989. *Numerical Recipes. The art of scientific computing.* Cambridge: Cambridge University Press.

Price, R.K., Samedov, J.N. & Solomatine, D.P. 1998. An artificial neural network model of a generalised channel network. In Babovic & Larsen (Eds.), *Hydroinformatics '98* (pp. 813–818). Rotterdam: Balkema.

Proaño, C.O., Verwey, A., Van den Boogaard, H.F.P & Minns, A.W. 1998. Emulation of a sewerage system computational model for the statistical processing of large number of simulations. In Babovic & Larsen (Eds.), *Hydroinformatics '98*(2) (pp. 1145–1152). Rotterdam: Balkema.

Recknagel, F., French, M., Harkonen, P. & Yanunaka, K.I. 1997. Artificial neural network approach for modelling and prediction of algal blooms. *Ecological Modelling* 96: 11–28.

Sanchez, L., Arroyo, V., Garcia, J., Koev, K. & Revilla, J. 1998. Use of neural networks in design of coastal sewage systems. *Journal of Hydraulic Engineering*: 457–464.

Scardi, M. 1996. Artificial neural networks as empirical models for estimating phytoplankton production. *Marine Ecol. Progress Series* 139: 289–299.

See, L. & Openshaw, S. 1999. Applying soft computing approaches to river level forecasting. *Hydrological Sciences Journal* 44(5): 763–778.

Shen, Y., Solomatine, D.P. & Van den Boogaard, H.F.P. 1998. Improving performance of chlorophyll concentration time series simulation with artificial neural networks. *Annual Journal of Hydraulic Engineering, JSCE* 42: 751–756.

Solomatine, D.P. & Avila Torres, L.A. 1996. Neural network application of a hydrodynamic model in optimizing reservoir operation. In Müller (Ed.), *Hydroinformatics '96* (pp. 201–206). Rotterdam: Balkema.

Stelling, G.S., 2000. A numerical method for inundation simulations. In Yoon, Jun, Seoh & Choi (Eds.), *Proceedings of the 4th International Conference on Hydro-Science and Engineering*. Seoul, Korea.

Ten Brummelhuis, P.G.J., Heemink, A.W. & Van den Boogaard, H.F.P. 1993. Identification of shallow sea models. *International Journal for Numerical Methods in Fluids* 17: 637–665.

Thompson, M.L., Kramer, M.A. 1994. Modelling Chemical Processes Using Prior Knowledge and Neural Networks. *AIChE Journal* 40(8): 1328–1340.

Van den Boogaard, H.F.P., Hoogkamer, M.J.J. & Heemink, A.W. 1993. Parameter identification in particle models. *Stochastic Hydrology and Hydraulics* 9(2): 109–130.

Van den Boogaard, H.F.P. & Kruisbrink, A.C.H. 1996. Hybrid Modelling by integrating neural networks and numerical models. In Müller (Ed.), *Hydroinformatics '96* (pp. 471–477). Rotterdam: Balkema.

Van den Boogaard, H.F.P., Gautam, D.K. & Mynett, A.E. 1998. Auto-regressive neural networks for the modelling of time series. In Babovic & Larsen (Eds.), *Hydroinformatics '98* (pp. 741–748). Rotterdam: Balkema.

Van den Boogaard, H.F.P., Ten Brummelhuis, P.G.J. & Mynett, A.E. 2000. On-Line Data Assimilation in Auto-Regressive Neural Networks. *Hydroinformatics 2000 Conference, The University of Engineering, IOWA City, USA, July 23–27, 2000.*

Van Gent, M.R.A. & Van den Boogaard, H.F.P. 1998. Neural network modelling of forces on vertical structures. In *Proc. ICCE 1998 Copenhagen*, 2: 2096–2109. ASCE

Verlaan, M. 1998. *Efficient Kalman filtering algorithms for hydrodynamic models*. Ph.D. Thesis, Delft: Delft University of Technology.

Verwey, A. 1994. Linkage of Physical and Numerical Aspects of Models Applied in Environmental Studies. Keynote Lecture in: *Proceedings of the Conference on Hydraulics in Civil Engineering.* Brisbane, Australia.

Wen, C.-G. & Lee, C.-S. 1998. A neural network approach to multiobjective optimization for water quality management in a river basin. *Water Resources Research* 34(3): 427–436.

Wüst, J.C. 1995. Current prediction for shipping guidance. In Kappen & Gielen (Eds.), *Neural Networks: Artificial Intelligence and Industrial Applications*. London: Springer Verlag.

8

RLF/1: Flood Forecasting via the Internet

SIMON A. CORNE AND STAN OPENSHAW
School of Geography, University of Leeds, UK

ABSTRACT: This chapter introduces a web-based neural network flood forecasting toolbox called RLF/1. This system comprises a set of software tools written in C++, neural network models, data sets and documentation, and runs on a SUN-Solaris workstation. The system is integrated using UNIX shell scripts, which display forms, upload data, create databases, train networks, control a simulation and display results. Output including results files, session logs and trained models are available for downloading. Three default neural network models, which are stored on the system and can be experimented with, were developed for flood forecasting on stable, flashy and intermediate river systems. They use historical data for gauging stations around the Ouse catchment in Yorkshire, UK. Using the models, a simulation of a real-time forecast may be run by supplying historical time-series data. Alternatively, the time series may be used to construct new models.

1 INTRODUCTION

Following a successful pilot project to demonstrate the application of artificial neural network (NN) methods to flood forecasting (Openshaw *et al*., 1998), the main barrier to the practical implementation and use of NN based flood forecasting systems was considered to be one of communication. In order for the river engineers and hydrologists who are responsible for flood warning and control to place their trust in such tools, the world of NN must first be demystified. To help address this issue, an integrated NN based flood forecasting system, called RLF/1, was developed using web-based technologies. RLF/1 comprises a suite of software tools running on a SUN-Solaris workstation. The tools have been written in C++ and are integrated via HTML and UNIX shell scripts. The system can be accessed and run on-line from the following address:
http://www.ccg.leeds.ac.uk/simon/intro.html

The aim of RLF/1 is to demonstrate that NN solutions are a practical tool for real-world flood forecasting operations. The Internet offers a system that can be openly accessed at minimal cost by engineers and scientists from around the world who can experiment with the NN models at their leisure. Time series data can be uploaded and a real-time forecast simulated using stored NN models. New models can also be constructed, trained on-line and then used to provide simulated forecasts. The use of web-based technologies means that background material, on-line help, instructions and links to relevant information are fully integrated within the software package.

Initial work on the application of NN to flood forecasting was applied to river flow data for the Yorkshire Ouse catchment (Openshaw *et al*., 1998). Input data for NN training were selected on the advice of hydrologists. A database was constructed from historical time series comprising immediate river-level history for principal and upstream gauging stations at fifteen-minute resolution, mean hourly river-level histories, mean daily rainfall, averaged long-term rainfall history, averaged daily temperatures, averaged daily evapotranspiration, and the length of daylight. In the current work, NN models have been developed using a minimal set of input data (river level time series for a principal gauging station combined with suitably lagged upstream time series) in conjunction with a method of pre-classification. Models have been trained for stable, flashy and intermediate types of river system. These models are then integrated to produce a computer system that can be adapted to fit a similar network of rivers and tributaries where comparable information is available.

2 MODELLING APPROACH

The RLF/1 system may be used in two ways. The first approach uses existing NN models that were developed for stable, intermediate and flashy types of river network based upon a simple routing model. For each of the three types of river network, two different sets of models were built: one producing a single predicted river level or flow at some future time (e.g. three hours ahead, six hours ahead, twelve hours ahead, etc.); the other produced a predicted time series of river levels/flows up to some future time. These models may be used to predict river levels for comparable river systems. When supplied with historical time series data by the user, RLF/1 can simulate a forecast by using the raw data as input variables to the models. The software allows for predictions to be compared with observations to give a measure of the success.

The second approach involves using RLF/1 as an interactive model-building tool, creating and training user-defined NN with user-supplied data.

2.1 *Basic model*

The RLF/1 system uses a very simple model of a river network. The system aims to forecast the river level (or flow) at a principal gauging station. The main input data for this station is derived from high-resolution historical river level time series for that station. Additional inputs, suitably lagged for mean

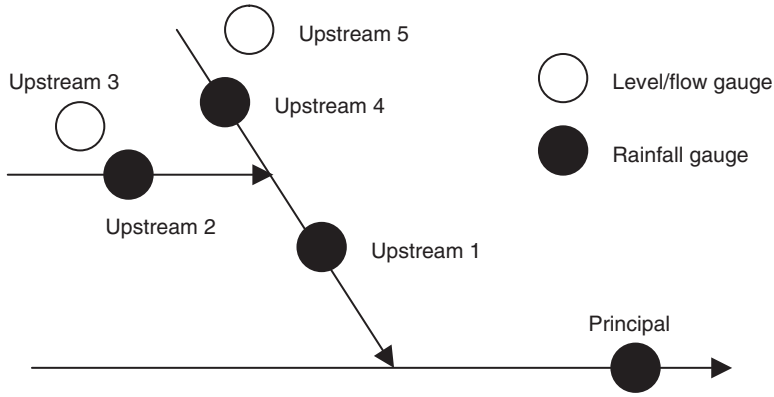


Fig. 8.1. Example of a simple routing model used by RLF/1.

time of transport, include: river level time series for the upstream gauging stations, rainfall time series for relevant upstream rainfall gauges, coarser-resolution and/or averaged time series of other upstream data, e.g. soil moisture (explicit measurement or implicit from long-term mean rainfall), temperature, evapotranspiration, etc. A simple model with two upstream gauging stations and two upstream rainfall gauges is illustrated in Figure 8.1.

RLF/1 may then be used to model the response at the principal gauging station by supplying the time series for the principal station and the four upstream gauges and their mean time of transport to the principal station.

2.2 *Pre-classification*

In earlier flood forecasting experiments (Openshaw *et al*., 1998), it was found difficult to train networks to cover the entire dynamic range of the data. The database, constructed from a long historic record, is mostly composed of vectors from periods of very low activity, punctuated with relatively short periods of interest: rising, falling and peak levels. Rather than attempting to train a single NN using randomly selected vectors from the entire database, a Kohonen (1988) self-organizing feature map (SOFM) was used to identify clusters in the data. The sixteen clusters produced by the optimum SOFM (a four-by-four map) were found to fall into five general categories (rising, falling, low and flat, moderate and flat, peak). Separate databases were created according to these five categories and NN were generated for each. For practical forecasting, the category of an input vector was identified and the forecast was generated using the most appropriate model.

In the web-based version of RLF/1, this procedure has been simplified. Instead of the five separate classes, RLF/1 makes use of either two (high, low) or three (high, intermediate, low) categories. The selection of thresholds to distinguish categories may be automatic or interactive. Separate network models are then used for each of the categories.

2.3 *NN for stored models*

Conventional multilayer feedforward NN were used as the basis for stored models. These were constructed, trained and tested using the Stuttgart Neural Network Simulator (SNNS Group, 2002) from the University of Stuttgart, Germany. Models were trained using the resilient propagation modification (Riedmiller & Braun, 1993) to the backpropagation training algorithm and also using the scaled conjugate gradient method (Moller, 1993).

2.4 *NN for on-line model building*

The principal concern for on-line model building was to create adaptable NN models. Conventional multilayer feedforward networks are prone to a problem commonly known as catastrophic forgetting, i.e. once trained, the NN cannot assimilate new training information without losing the ability to represent its original training information. We have used an approach for adaptive modelling developed for process control (Mills *et al*., 1995). These methods are based upon conventional multilayer feedforward models but incorporate some architectural modifications and make use of new training algorithms.

2.4.1 *Architectural modifications*

Two modifications to the conventional multilayer networks have been implemented. The first is the use of linear nodes at the output layer. This has the advantage of a wide dynamic range with respect to the strict range enforced by the standard sigmoidal transfer function (Mills *et al*., 1995). A modified gain parameter for updating hidden-to-output connection weights is also required during training. Following Mills *et al*. we have used a fixed value of 0.1 for this parameter.

Some features of the data being modelled may have an approximately linear relationship. Conventional feedforward networks, which by nature use non-linear transfer functions, cannot always make a very close approximation to linear functions. But it is possible to represent a near linear relationship by combining a non-linear network with linear component mappings. The second architectural modification is the incorporation of linear bypass connections, with trainable weights, between the input and linear output layers (Mills *et al*., 1995); this is illustrated in Figure 8.2.

2.4.2 *History stack adaptation*

It has been suggested that the computational requirements of simulated NN are too great for real-time application (Sanner & Slotine, 1991). The history stack adaptation method (Mills *et al*., 1995) can enhance convergence characteristics for adaptive identification of non-linear processes using multilayer feedforward networks (MLFN). The algorithm is based upon a short history of input patterns that can represent an approximation to the non-linear mapping. The history stack is a first-in-first-out (FIFO) stack containing n_p patterns. The method requires warming-up for several time steps as the stack is filled. At each subsequent time

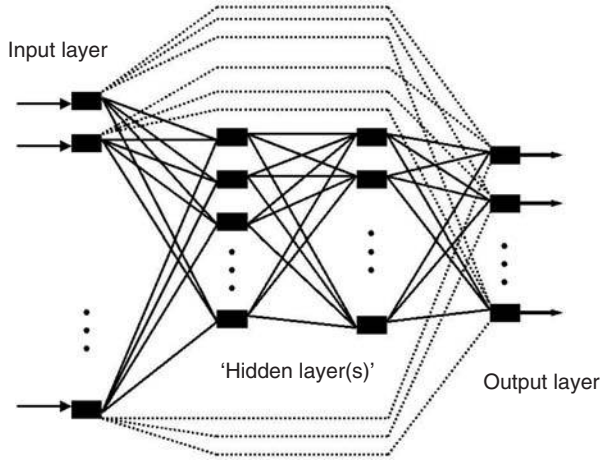


Fig. 8.2. Schematic diagram of a modified multilayer network. Standard weighted interconnections are shown by solid lines. Linear bypass connections are shown by dotted lines.

step a new pattern is loaded onto the stack while the oldest pattern is discarded; also, network weights are adjusted over n_c cycles using the patterns in the stack. Supervised learning using all patterns in the stack, in random order, at each time step results in a network that minimizes the error corresponding to the approximate mapping represented in the stack patterns.

The algorithm uses each pattern $n_p n_c$ times before it is discarded, with a concomitant increase in the load on computational resources. The selection of these two parameters is critical to the adaptation process: the number of cycles must be greater than one in order to achieve a large learning increment at each time step; the number of patterns should be small enough to be both representative and to avoid the retention of out-of-date information. The combination of n_p and n_c is critical to performance. Different combinations can be used for a variety of purposes: e.g. fast adaptation, low computational cost, complex mapping and noisy data.

As well as the architectural modifications described above, further extensions to the history stack adaptation algorithm can enhance performance. For time-varying processes it may be appropriate to attach more weight to more recent patterns. This can be achieved by exponential time weighting of the NN learning rate parameter. For stability it may be necessary to exclude some patterns from the stack: an entry criterion may be used to measure the 'novelty' of each pattern. If a new pattern is too similar to those already present in the stack, there is little to gain from its inclusion. A sufficiently novel pattern will make more efficient use of the history stack method.

2.4.3 *Reinforcement backpropagation*

A truly adaptable NN system must use current data to modify the behaviour of the model. Reinforcement backpropagation (Mills *et al*., 1995) uses reinforcement

learning, in which feedback is provided by a scalar performance measure given by a so-called 'critic function,' and the history stack algorithm to provide good learning performance.

The reinforcement learning system receives a time-varying input vector and generates a time-varying output vector. The output vector is adjusted by a random dither vector with elements selected from a Gaussian distribution. After a delay, the system receives a time-varying scalar signal related to the performance resulting from the output vector. This performance measure is based upon the comparison of calculated $u(k)$ and observed $y(k)$ outputs:

$$p(k) = -0.2 \log([y(k) - u(k)]^2 + 0.001) \tag{1}$$

The reinforcement learning algorithm aims to optimize the performance scalar for future cases. The performance measure is passed through a discrete-time low-pass first order filter that estimates the current average performance against which new performances are measured. This is called the improvement comparison: a positive value indicating improvement, a negative value indicating diminished performance. Given the delay in acquiring the performance measure, the input, dither and output vectors are passed through a shift register corresponding to the delay. A new target vector based upon the suitably delayed output and dither vectors is created and combined with delayed input vector to create a new training pattern which can be entered into the history stack. A modification to this algorithm only accepts the new training pattern if the improvement comparison is positive. The method proceeds using the history stack algorithm described in the previous section.

2.5 *Bootstrapping*

The RLF/1 software includes a simple bootstrapping algorithm to give a statistical measure of the errors in forecasted values. At each step of the bootstrapping algorithm, some input variables are discarded at random and are replaced by duplicates of remaining variables. The output is calculated according to the modified input vector and the process is repeated. This method permits the calculation of a mean output with confidence limits. The statistical significance of the results is increased by increasing the number of steps although this will also increase the overall computation time. The method also simulates the effect of missing input values, a problem commonly faced by real-time computer applications. The number of steps may be set to zero, which results in a single calculation of output without statistics.

2.6 *Historical data*

The RLF/1 system has been constructed and tested on a set of data supplied by the UK Environment Agency, North-Eastern Division. The data cover the years 1982–1996. River levels at fifteen-minute intervals were available for nineteen gauging stations in the Ouse catchment in North Yorkshire. Rainfall data at fifteen-minute

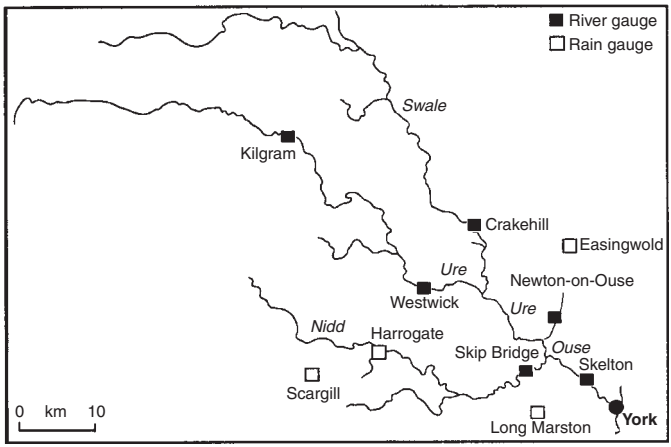


Fig. 8.3. Location of the Ouse catchment.

intervals were available for eleven stations in the same geographical area. In addition, daily mean rainfall data were available for thirty-five different locations. Daily climate data for Leeming in North Yorkshire were obtained from the UK Meteorological Office archive maintained by the British Atmospheric Data Centre at the Rutherford Appleton Laboratory in Chilton, Oxfordshire. The data files for use by RLF/1 may be supplied in Hydrolog and Rainark formats (used by the Environment Agency) or as comma-separated variable files.

The Ouse catchment (Fig. 8.3) incorporates several river sub-networks that exhibit quite different characteristics. The gauging station at Skelton, near the city of York, has a very stable, slowly-evolving response, where the flood-peak can take a week to rise and fall. At the top of the catchment, the gauging station near Low Houses exhibits a very flashy response, where a flood peak might rise and fall within a few hours. The gauging station near Boroughbridge, roughly midway between Low Houses and Skelton, displays characteristics intermediate to these two extremes, with a flood peak that may last two or three days. The data for these three principal stations, together with their upstream stations, were used to construct the stable, flashy and intermediate models, respectively.

3 THE WEB-BASED INTERFACE

The RLF/1 system uses web-based technologies to provide a relatively seamless transition between descriptive text (both background and help screens), data uploading, parameter specification, program control and the evaluation of results. The system encompasses C++ software tools, HTML documentation, SUN-Solaris shell scripts and GNU utilities. It contains five main sets of tools for data file conversion, simple statistics, database construction, NN and flood forecasting. Step-by-step guides to using the flood forecaster and model builder are also available on-line.

3.1 *Using the flood forecaster*

The RLF/1 system is comprised of a series of forms, which are listed in Table 8.1. Form 1 is the RLF/1 home page, which contains background information written in HTML. Several choices are provided. Selection of the flood forecasting or model building options leads to a registration page (name, address, e-mail address). The next stage is to define the river system, its associated data files (i.e. river level/flow and rainfall time-series) and the formats. Form 2 (Specification of the basic model), shown in Figure 8.4, allows the construction of the routing model in terms of transport times from upstream gauging stations to the principal station. Several other parameters may be specified: the look-ahead time (*n* hours), the data resolution (quarter-hourly, hourly, etc.), the catchment type (i.e. stable, flashy, intermediate) and the type of model (i.e. trained network, trained network with some adaptation, the history stack/reinforced learning model). The flood forecasting models normally generate a single number, which is the river level at

Table 8.1. Forms in RLF/1.

Form	Function
1	RLF/1 home page
2	Specification of the basic model
3	Specification of the pre-classification thresholds
4	Modified version of Form 3
5	Specification of model architectures and training parameters

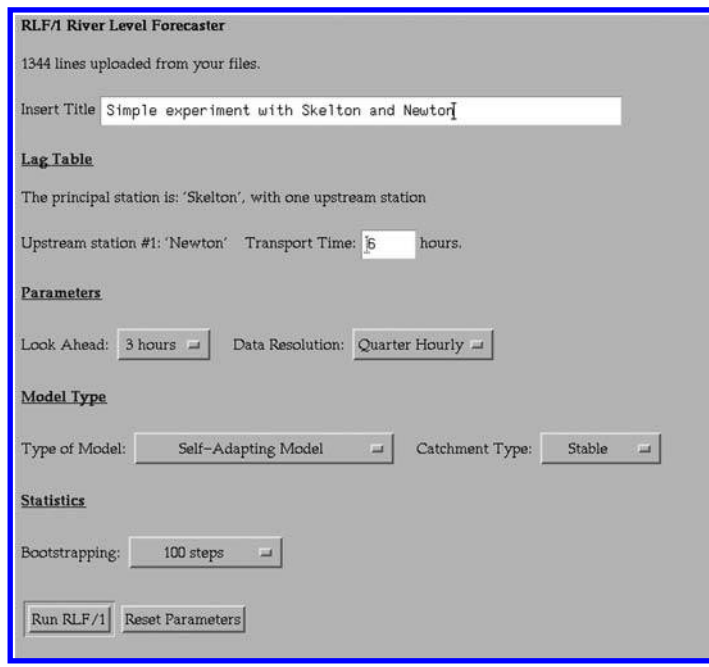


Fig. 8.4. Form 2: Specification of the basic model.

a time *n* hours in the future. The use of bootstrapping to measure statistical error in the forecast can be specified. This method also provides an idea of how well the models would perform if one or more input variables were missing.

Once the model is run, Form 3 (Specification of the pre-classification thresholds) appears, as shown in Figure 8.5, which displays the time series for the principal station and allows the specification of a level threshold to distinguish high and low levels, as well as the expected minimum and maximum levels. These levels are estimated automatically but may be overwritten as required. Following a parameter check the forecaster is run, displaying the current time, time of prediction, predicted value and confidence limits (if bootstrapping has been specified) and the prediction error. The results are displayed graphically, with error bars on the predictions if bootstrapping has been specified. A series of expanded plots allows a closer examination of the quality of the predictions. Separate pages provide a listing of the time series of predictions, observations, comparison between the predictions and observations and a log file, all of which can be downloaded. The final form allows the extension of the forecast beyond the supplied time series. The last few levels/flows for each of the stations are displayed and the values at the next time step may be supplied to generate a new forecast.

A step-by-step guide is also available. To examine the capabilities of the system, or if the interested user has no suitable data, a demonstration is provided using data stored on the web-server.

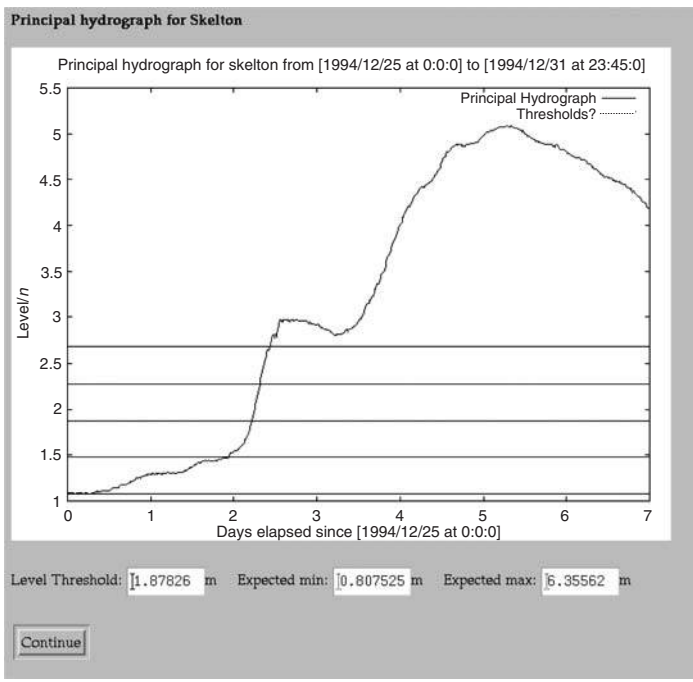


Fig. 8.5. Form 3: Specification of the pre-classification thresholds.

3.2 *Using the model builder*

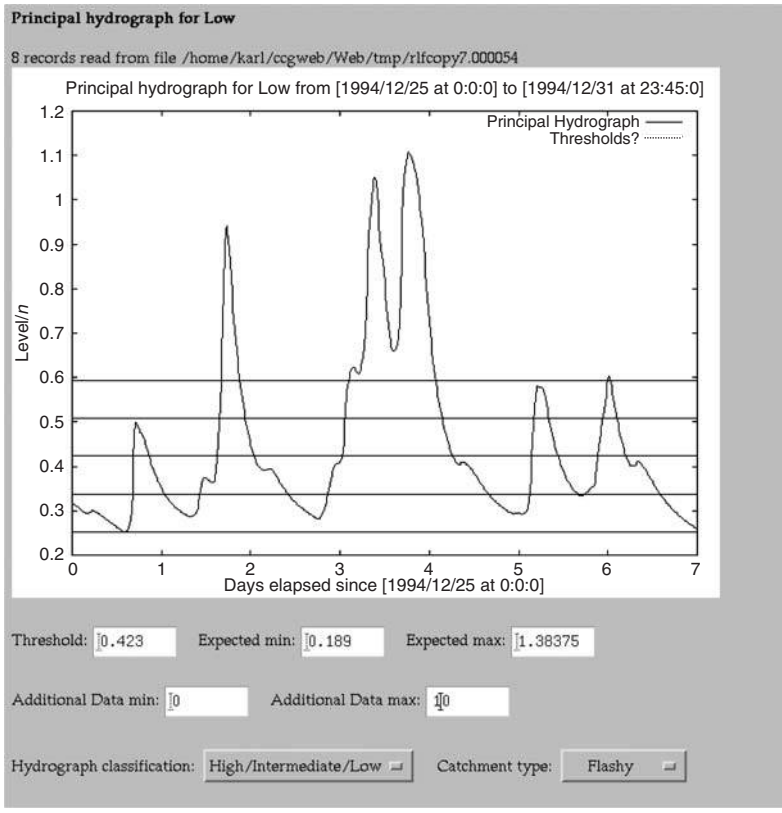
The interface to the model builder is largely the same as that used by the flood forecaster. The data specification form following registration does have one major difference: an additional low-resolution (e.g. daily, monthly) time-series may be included in the modelling, such as soil moisture, average temperature or long-term average rainfall.

Next is a modified version of the flood forecaster time-series display form. The major differences, shown in Figure 8.6a, are the specification of the expected minimum and maximum for the additional low-resolution data and the high/low (H/L) or high/intermediate/low (H/I/L) classification; Figure 8.6b shows the specification of the parameters for constructing the databases. The input window and look-ahead time determine the size of input and target vectors. By default, half of each database is used for training, a quarter for validation and a quarter for testing but these proportions can be changed. Generally, training vectors are selected in random order from the database, but this can also be changed to sequential selection, if required.

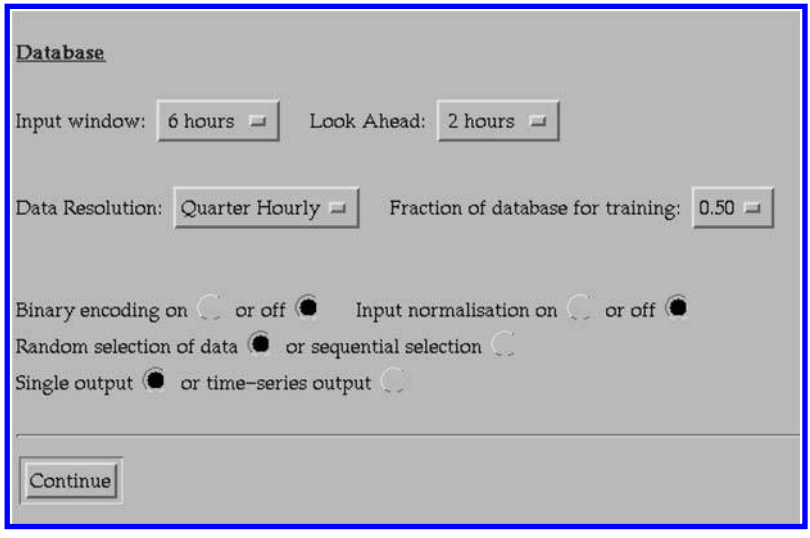
Using the parameters specified in the previous form, the two (H/L) or three (H/I/L) databases are created and then split into training, validation, and testing files. Further events may be added to the databases using a modified data specification form, before the NN models are created. The final form (Form 5), as shown in Figure 8.7, is used for the specification of the architecture and training parameters for the H/L or H/I/L NN models. NN architecture is specified by the number of hidden layers (one or two), the size of the hidden layer(s), the use of linear bypass connections and the use of linear output gain. Training is controlled by the number of training cycles, gain and momentum parameters, initial weight range and exponential time weighting factor. The next two screens serve to display useful feedback information: the first lists the parameters and the second displays the classification of input vectors into the two or three categories selected. The two or three NN models are then trained according to the parameters. The training and validation data sets are used during the NN training. On completion, i.e. at the end of the specified number of iterations, the unseen data set is passed through the model and the calculated and observed values are listed.

The results of training are made available in a number of ways, mostly via links to files that can be displayed and downloaded. Such files include the progress of training, the trained network represented as a text file, as well as a session log. Graphical displays are also provided, showing the progress of training and a comparison of the calculated and observed values.

If the models are sufficient for the end user, they may be saved. It is then possible to test the saved models with new, unseen data or a new training exercise may be started. If training is not satisfactory, the models may be trained further with the option of changing one or more parameters (number of cycles, gain, momentum and exponential time weighting). To test a trained network, the data specification form is again used to upload time series files for the stations comprising the routing model. The flood forecaster is then run using the models and the test data. The display of results is as described in the previous section.



(a)



(b)

Fig. 8.6. Form 4: Modified version of Form 3 (a) The RLF/1 model builder form for database construction. Setting pre-classification thresholds. (b) The RLF/1 model builder form for database construction. Setting database parameters.

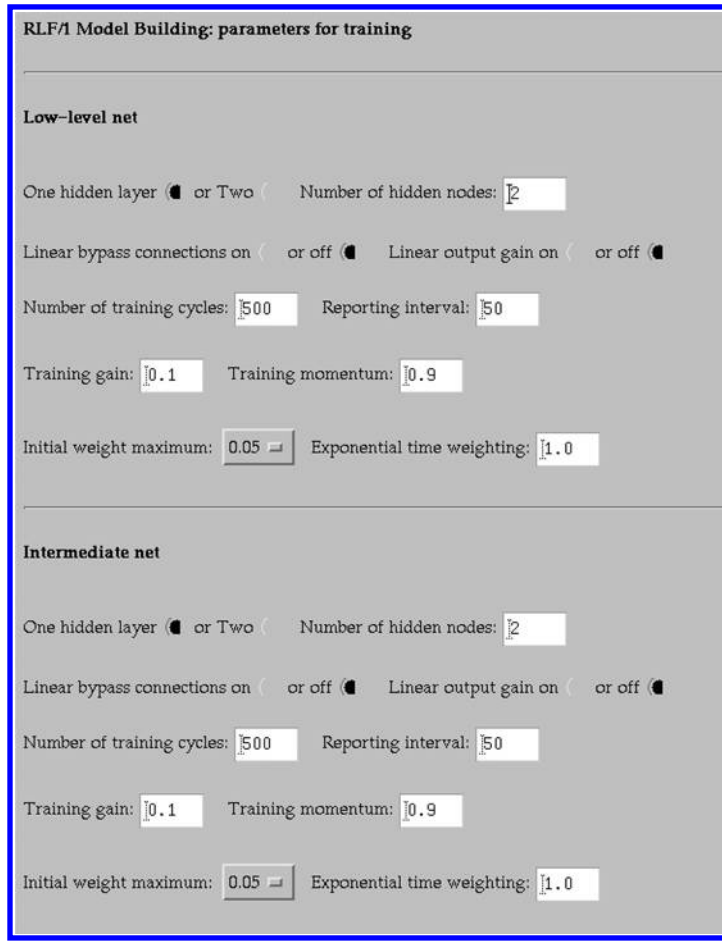


Fig. 8.7. Form 5: Specification of model architectures and training parameters.

4 WORKING EXAMPLES

The model builder in conjunction with the flood forecaster has been tested with several examples derived from two different data sets, one from the Ouse catchment and the second from Northumbria; both data sets were supplied by the Environment Agency. The results of the tests with both catchments are available on the website. In this chapter, examples are provided for a stable, intermediate and flashy regime from the Ouse catchment.

4.1.1 *Stable response testing*

The aim of this stable event experiment was to predict the river level at the Skelton gauging station, near the city of York, using historical level time series for Skelton only. A six-hour input window was used to predict the future level with a forecasting horizon of six hours. A threshold of 2 m was used to perform

Table 8.2. Model architectures for the stable example.

Model	Input size	Hidden layers	Hidden size	Output size	Training cycles
Low	24	1	6	1	250
Intermediate	24	2	3	1	500
High	24	2	3	1	500

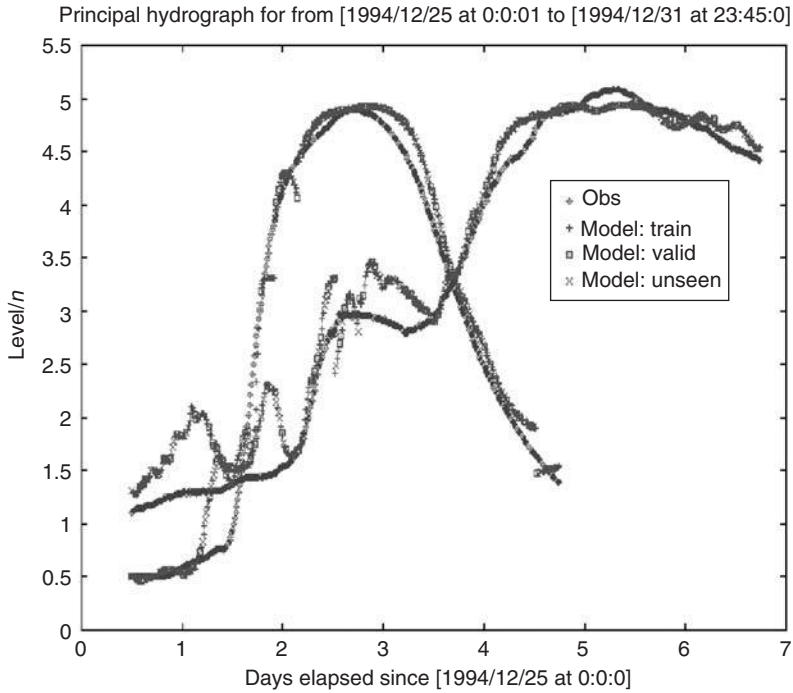


Fig. 8.8. Results for training events at Skelton from the summer of 1986 and the winter of 1994.

a pre-modelling classification of the data into three categories. All input data were standardised and the expected data range was set to 0–7 m. The three models are described in Table 8.2.

Training data were derived from events at Skelton during the summer of 1986 and the winter of 1994. The results of training are shown in Figure 8.8, where the predicted values are plotted alongside the observed levels. The trained models were tested with a flood event at Skelton from the winter of 1995. The test results are provided in Figure 8.9, which show good correspondence with the first peak and an under prediction of the second peak. This is probably due to the fact that the second event is particularly large and exceeds the levels that were used to train the model. To improve the model performance, very large events, either real or simulated, should be presented to the model during the training process.

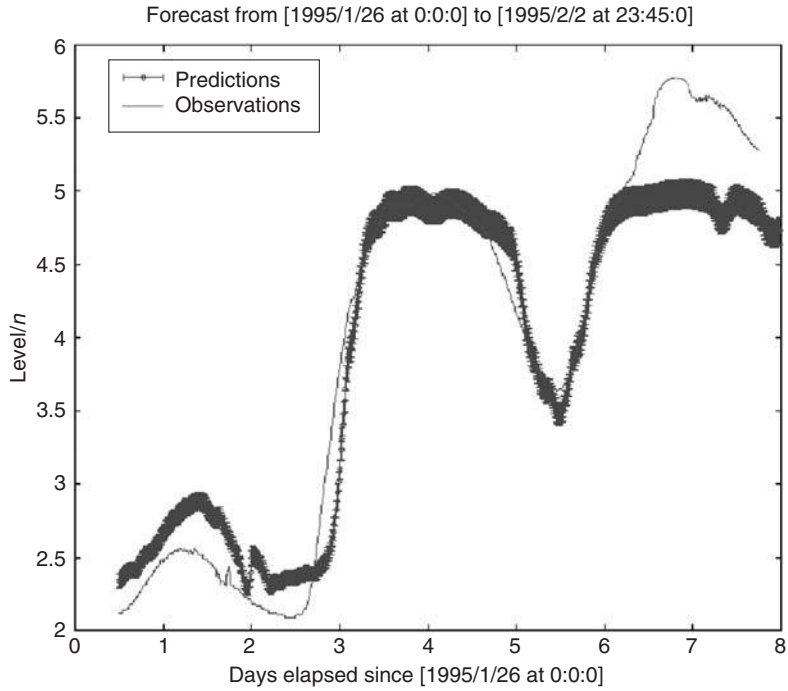


Fig. 8.9. Results for a test event at Skelton from the winter of 1995.

Table 8.3. Model architectures for the intermediate example.

Model	Input size	Hidden layers	Hidden size	Output size	Training cycles
Low	80	1	4	1	250
Intermediate	80	1	8	1	400
High	80	2	4	1	500

4.1.2 *Intermediate response testing*

In this experiment, the aim was to predict the river level at Boroughbridge in North Yorkshire based upon historical level time series for the nearby gauging station at Westwick, together with upstream gauging stations at Alma Weir, Kilgram and Low Houses, as well as the rainfall gauge at Tow Hill, near Low Houses. A four-hour input window was used to predict the future level with a forecasting horizon of three hours. A threshold of 1.12 m was used to perform a pre-modelling classification of the data into three categories. All input data have been standardised. The expected data range was set to 0–4.25 m. The three models are described in Table 8.3.

Training data were derived from events in the region during the winters of 1994 and 1995. The results of training are shown in Figure 8.10, where the predicted values are plotted alongside the observed levels. The trained models

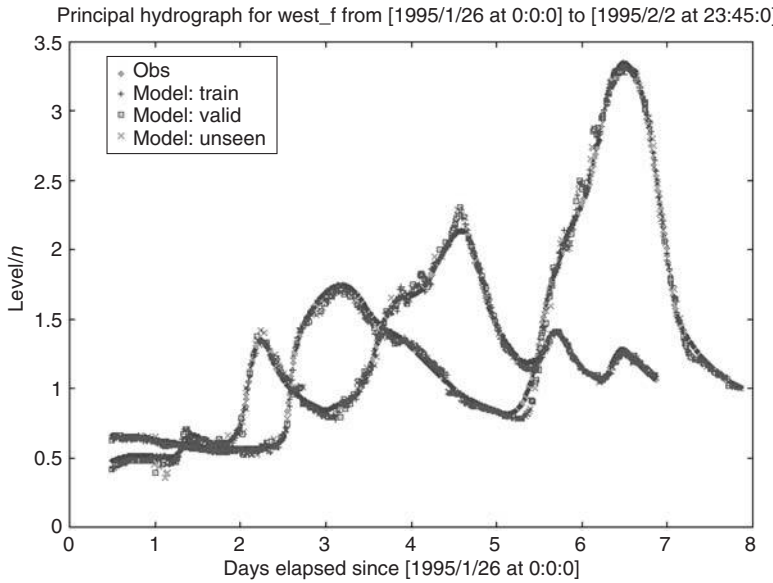


Fig. 8.10. Results for training events at Boroughbridge from the winters of 1994 and 1995.

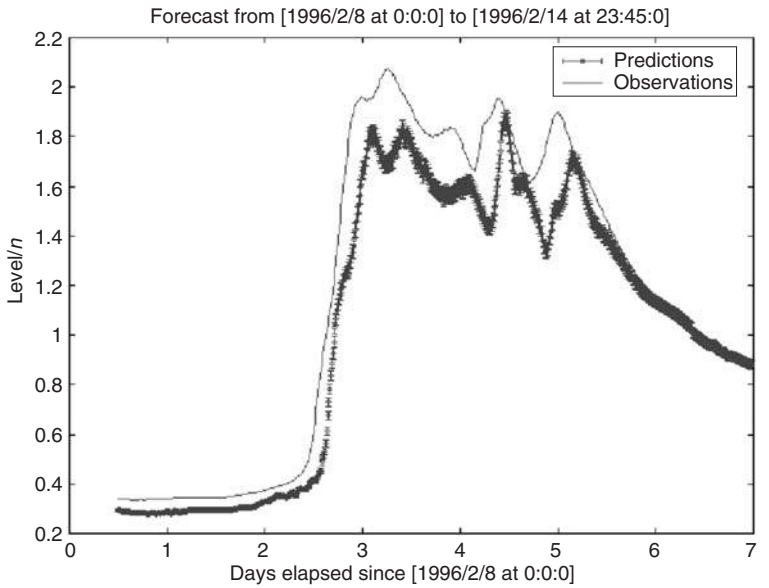


Fig. 8.11. Results for a test event at Boroughbridge from the winter of 1996.

were tested with an event in the region during the winter of 1996. The test results are shown in Figure 8.11.

Although the training hydrographs are predicted perfectly, the validation event shows a lag and a small under prediction although the general shape of

Table 8.4. Model architectures for the flashy example.

Model	Input size	Hidden layers	Hidden size	Output size	Training cycles
Low	32	1	4	1	500
Intermediate	32	2	2	1	500
High	32	2	2	1	500

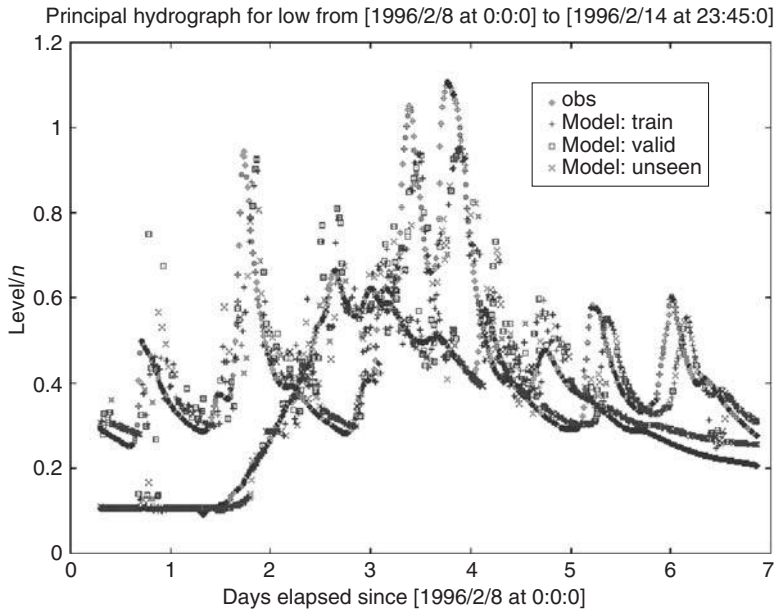


Fig. 8.12. Results for training events at Low Houses for the winters of 1994 and 1996.

the curve is well represented. The absolute errors in level are quite small so further improvements might require training on more events of this size.

4.1.3 *Flashy response testing*

The aim of this flashy event experiment was to predict the river level at Low Houses gauging station in North Yorkshire, using historical level time series for Low Houses, together with time series for the rainfall gauge at nearby Tow Hill. A four-hour input window was used to predict with a forecasting horizon of three hours. A threshold of 0.2 m was used to perform a pre-modelling classification of the data into three categories. The expected data range was set to 0–2.5 m. In contrast to the other two examples, all input data were retained as raw values. The three models are described in Table 8.4.

Training data were derived from events at Low Houses during the winters of 1994 and 1996. The results of training are shown in Figure 8.12, where the predicted values are plotted alongside the observed levels. The trained models

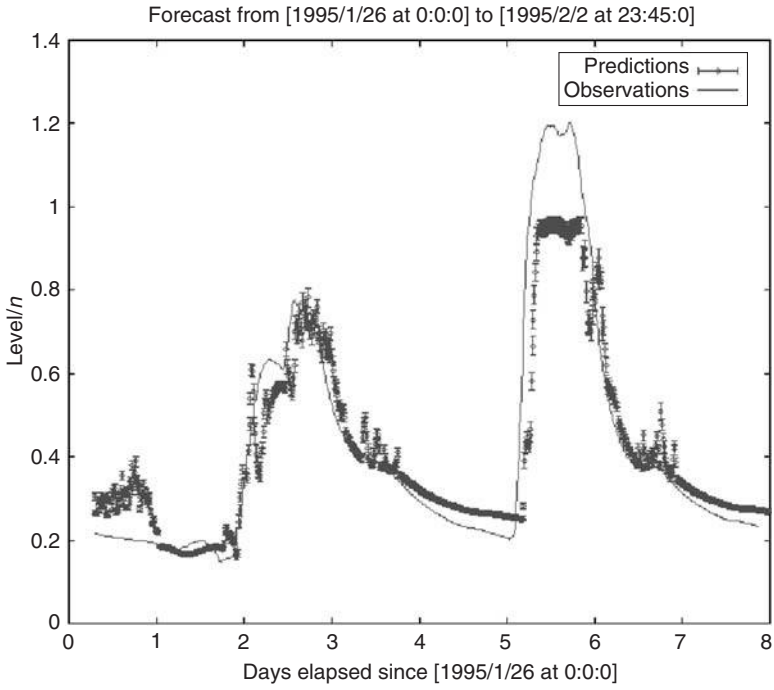


Fig. 8.13. Results for a test event at Low Houses from the winter of 1995.

were tested with an event at Low Houses during the winter of 1995. The test results are shown in Figure 8.13.

The results for the test event exhibit similar patterns to the first experiment, i.e. good forecasts for smaller hydrographs and under prediction for larger ones although once again, the absolute errors are not large. However, the model was trained to see events close to 1.2 m so the problems with under prediction may simply reflect the more difficult nature of this hydrological regime.

5 NEXT STEPS

The aim of RLF/1 is to demonstrate to a wider audience that NN are a practical tool for flood forecasting. The implementation of this tool as an Internet application means that interested engineers and scientists can experiment with NN, at little or no cost, and see whether or not these tools can provide a suitable solution for the task in hand.

RLF/1, as the name implies, is a first release and therefore still has some limitations. For example, the underlying routing model is very simple, the maximum number of upstream gauging stations is restricted to five and only one additional long-term input variable can be specified. The selection of certain training parameters is also limited to menu choices. It should, however,

be stressed that the software beneath the interface is not limited so it will be possible to provide a more flexible tool in future versions. Another limitation is that the only NN supplied are multilayer perceptrons with either one or two hidden layers, although some extensions to the basic models are provided. Additional NN architectures will be added in later versions. Finally, the system does not permit prediction of hydrographs with different lead times or forecasting horizons. RLF/2 is being developed and will incorporate alternative methods for time series prediction (ARMA and fuzzy logic modelling) as well as hybrid systems that will provide increased flexibility.

The technologies underlying RLF/1 can be customized for particular river networks and climatic regimes and can therefore be used as a practical flood forecasting tool. It is left to hydrologists and river engineers to experiment with RLF/1 to see if it answers (some of) their requirements for flood forecasting. Please visit the web site and test our software.

REFERENCES

Kohonen, T. 1988. *Self-Organization and Associative Memory*. Springer-Verlag.

Mills, P.M., Zomaya, A.Y. & Tade, M.O. 1995. *Neuro-adaptive process control: a practical approach*. Chichester: John Wiley & Sons.

Moller, M.F. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, v. 6, 525–533.

Openshaw, S., Kneale, P., Corne, S.A. & See, L. 1998. Artificial neural networks for flood forecasting. Final report, MAFF project OCS967P, University of Leeds.

Riedmiller, M. & Braun, H. 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks 1993 (ICNN 93)*.

Sanner, R.M. & Slotine, J.E. 1991. Gaussian networks for direct adaptive control. *Proc. 1991 American Control Conference*: 2153–2159.

SNNS Group 2002. Stuttgart Neural Network Simulator, online http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/snns.html

9

Rainfall-Runoff Modelling

ANTHONY W. MINNS
WL/Delft Hydraulics, Delft, The Netherlands

MICHAEL J. HALL
UNESCO-IHE, Institute for Water Education, Delft, The Netherlands

ABSTRACT: Artificial neural networks (NN) are recognised as being universal approximators and are capable of extracting the underlying relationship between any input, or stimulus, and its subsequent output, or response. This property is particularly attractive when dealing with the complex natural systems that are commonly encountered in hydrology, hydrogeology and water resources planning and management. Among the many other applications, the last few years have seen increasing attention being paid to rainfall-runoff modelling. However, the effective use of NN in this context is seen to demand almost as much, if not more, hydrological insight than conceptual/physical rainfall-runoff models. Particular attention must be paid to the choice of input variables, and to the patterns of inputs and outputs upon which the NN is trained and validated.

1 INTRODUCTION

In this chapter we look at the position of rainfall-runoff models in the broader context of the already wide choice of models for hydrological forecasters. Why should a modeller choose this approach? The last decade has seen the gradual introduction of informatic tools, such as artificial neural networks (NN), into hydrology, hydrogeology and water resources planning and management. The applications of these techniques have been many and various, but a broad appreciation of their potential has been slow to develop. One reason for the apparent reluctance to consider such an approach may be unfamiliarity with the concept of, and nomenclature associated with, NN. Essentially, NN are excellent computational devices for pattern recognition and classification; it is

their origins in artificial intelligence and cognitive sciences that have given rise to what is unfamiliar terminology in the hydrological sciences.

The structure of a NN, which consists of multiple interconnections and their parallel processing architecture have an obvious analogy with biological systems, as explained in any introductory textbook on the subject (e.g. Alexander & Morton, 1990; Beale & Jackson, 1990; Hertz *et al.*, 1991). There are, however, many different types of NN, the choice between which can add to the discomfort of the potential user. However, the purpose of this commentary is not to elaborate upon the different forms of NN (see Chapters 2–7), although inevitably some reference has to be made to the architectures and calibration methods that have been applied to date. In preference, the emphasis is placed on the applications. In this context, a cursory review of the literature is sufficient to reveal the wide variety of problems to which NN have been applied successfully. The following listing is intended to be illustrative rather than exhaustive:

- as sub-models of complex processes within a larger physically-based framework; the estimation of daily solar radiation from daily maximum and minimum air temperatures and precipitation (Elizondo *et al*., 1993), and the modelling of drying water retention curves for sandy soils by Schaap and Bouten (1996) provide ready examples of this approach;
- as a replacement for, or for modelling the results obtained from, more complex, physically-based computer models that impose heavy demands on computing resources; examples include the river salinity forecasting model of Maier and Dandy (1996), and the determination of optimum pumping scenarios for groundwater remediation schemes by Rogers and Dowla (1994) and Rogers *et al.* (1995), and Aly and Peralta (1999);
- as models of analytically-intractable relationships, such as the approximation for the confidence limits to the quantiles from a flood frequency distribution derived by Whitley and Hromadka (1999);
- as a method of avoiding the constraints associated with standard techniques, such as multiple linear regression analysis, in deriving relationships between the parameters of a regional flood frequency distribution and catchment characteristics (Muttiah *et al.*, 1997; Hall & Minns, 1998; Hall *et al.*, 2000, 2002);
- as a screening model, e.g. for the identification of critical realisations of log conductivity fields for a single realisation groundwater remediation management model (Ranjithan *et al.*, 1993);
- as a method of pattern completion, e.g. the neural kriging method of Rizzo and Dougherty (1994);
- as a model of the preferences of decision-makers in multi-objective optimisation (see Wen & Lee, 1998 for an example);
- as a forecasting device, such as that for predicting rainfall fields proposed by French *et al.* (1992) and Luk *et al.* (1998);
- as an alternative to parameter-intensive physical/conceptual models in applications that do not require a detailed understanding of the system dynamics.

The problem that seemingly has attracted the most attention in this last category has been that of modelling the relationship between rainfall and runoff. However, as the following review is intended to demonstrate, the application of

NN to the development of a rainfall-runoff model demands as much, if not more, hydrological insight than the calibration of a standard physical/conceptual model (see also Maier & Dandy, 1999).

The details of a representative selection of 16 studies in which NN have been employed to develop rainfall-runoff models are presented in Table 9.1. A more

Table 9.1. Summary of previous studies in which NN have been employed to model rainfall-runoff relationships

Reference	Time unit	Catchment	Area (km^2)	Technique
Abrahart and Kneale (1997)	Hour	Wye, UK	11	MLP/BPNN
Campolo *et al.* (1999)	Hour	Tagliamento, Italy	2480	MLP/BPNN
Carriere *et al.* (1996)	30 second	Laboratory catchment	2.1 m^2	MLP/BPNN
Dawson and	15 minute	Amber, UK	139	MLP/BPNN
Wilby (1998)	15 minute	Mole, UK	142	MLP/BPNN
Fernando and Jayawardena (1998)	10 minute	Kamihonsha, Japan	3.12	RBF/OLS; MLP/BPNN
Hall and Minns (1993)	5 second	Laboratory catchment	26.8 m^2	MLP/BPNN
	one minute	Doncaster, UK	5.14 ha	MLP/BPNN
Hsu *et al.* (1995)	Day	Leaf River, USA	1949	MLP/LLSSIM
Jayawardena and Fernando (1998)	Hour	Kamihonsha, Japan	3.12	RBF/OLS; MLP/BPNN
Lange (1998)	Hour	Zeller Bach, Germany	20	MLP/BPNN
	Hour	Windach, Germany	344.7	MLP/BPNN
Lorrai and Sechi (1995)	Month	Araxisi, Italy	121	MLP/BPNN
Minns and Hall (1997)	30 minute	Dollis Brook, UK	24	MLP/BPNN
	30 minute	Silk Stream, UK	31.25	MLP/BPNN
Poff *et al.* (1996)	Day	Independence, USA	230	MLP/BPNN
	Day	Little Patuxent, USA	97	MLP/BPNN
See and Openshaw (1998)	Hour	Ouse, UK	3286	MLP/BPNN with KN
Shamseldin (1997)	Day	Sunkosi, Nepal	18000	MLP/CG
	Day	Shiquan, China	3092	MLP/CG
	Day	Yanbain, China	2350	MLP/CG
	Day	Bird Creek, USA	2344	MLP/CG
	Day	Wolombi Creek, Australia	1580	MLP/CG
	Day	Brosna, Ireland	1207	MLP/CG
Teegavarapu (1998)	10 day	Malaprabha, India	?	MLP/BPNN; RBF/OLS
Zealand *et al.* (1999)	Quarter-month	Namakan	19270	MLP/BPNN

Notation: MLP – multi-layer perceptron; BPNN – error back-propagation; RBF – radial basis function; OLS – ordinary least squares; LLSSIM – linear least squares with multi-start simplex operation; CG – conjugate gradient; KN – Kohonen network.

comprehensive tabulation has been provided by Dawson and Wilby (2001), and the details of several of the cited studies have been reviewed by the ASCE Task Committee on Application of Neural Networks in Hydrology (2000). The approach that has been overwhelmingly favoured has been the multi-layer perceptron (MLP) network with the error back-propagation learning algorithm (BPNN). Several authors reported favourable results using radial basis function (RBF) networks due primarily to the fact that the training time of RBF networks was usually significantly less than for equivalent MLP networks. Furthermore, the RBF networks appear to provide a superior performance over MLPs when dealing with only small numbers of input data sets (see Dibike *et al.*, 1999). However, the generalisation properties of the RBF networks deteriorate as the number of input data sets increases and RBF networks are subsequently out-performed by MLP networks (Y B Dibike personal communication, 1997, see also Minns, 1998, p. 33).

2 RAINFALL-RUNOFF RELATIONSHIPS

> The shape of a hydrograph for any given stream is a function of total available overland flow supply, subsurface flow, groundwater flow, slope of the overland and stream segments, roughness characteristics of flow elements, and geometry of channels. (Bras, 1990, p. 385)

The above quotation, from a well-known hydrological textbook, provides a succinct description of the physics of hydrograph generation. The forcing function to the catchment system is obviously precipitation in general, and the variations of rainfall intensity over time in particular. However, the relationship between the rainfall intensity and the response of the catchment in terms of changes in discharge at the outlet is primarily dependent on the action of the intervening processes within the hydrological cycle. Overland flow supply and subsurface flow are essentially functions of the soils and vegetative cover, and therefore dependent on the state of wetness of the catchment prior to the rainfall. The contribution of groundwater is a function of geology and the height of the phreatic surface in relation to the channel system. The slope, roughness and geometry of the latter then shape the formation of the outlet hydrograph.

Such influences are well appreciated in a qualitative sense, and models of individual processes, such as those for unconfined/confined groundwater flow, are readily available. Their integration into spatially-distributed, physically-based models of the land phase of the hydrological cycle has been pursued vigorously for almost two decades. However, as noted by Abbott and Refsgaard (1996), their application in practice belies the need for the type of results that such models are capable of providing. Nevertheless, not all hydrological problems which depend on rainfall-runoff modelling for their solution require an in-depth knowledge of individual components of the hydrological cycle. This differentiation was recognised many years ago by Amorocho and Hart (1964), who coined the terms *physical hydrology* and *systems investigation* to describe the two different approaches. Whereas the former is directed at a complete synthesis of the hydrological cycle, in the spatially-distributed and physically-based sense

described above, systems investigation is concerned with the solution of technological problems within the constraints imposed by the available data.

Perhaps the most widely-known of the modern generation of physically-based, distributed catchment modelling systems is the *Système Hydrologique Européen* (SHE), the original structure of which was described by Abbott *et al.* (1986). Details of a case study in which SHE was applied to a river basin of some 4955 km^2 in India have been provided by Refsgaard *et al.* (1992). Those authors provided a frank discussion of the substantial data requirements and supplementary fieldwork required to implement their model. They acknowledged that their use of 2 km × 2 km grid squares still did not provide a fully physically-based and fully-distributed description of the basin, even though it was entirely sufficient for the practical application in question. Even in these systems there remained a certain degree of empiricism in the representation of particular hydrological processes. Consequently process identification and the associated determination of parameter values by direct measurement continued to require extensive calibration procedures.

From this it can be concluded that for many problems of rainfall-runoff modelling simpler approaches have serious merit. Less detailed models would in most situations be equally accurate and much cheaper to apply. For example for forecasting at sites without any significant changes in land use, or for forecasting over a certain range and distribution of antecedent soil conditions. But to be fair to the distributed, physically-based models it should be pointed out that many of these aim to solve more complex process problems than simple rainfall-runoff. In practical terms problems of waste disposal, erosion, changes in vegetation and so on, are much more important than rainfall-runoff alone although such models might include a NN approach as part of its structure.

In contrast, systems investigation is a data-driven approach and hydrology in general and rainfall-runoff modelling in particular provides ample opportunities to take advantage of informatic techniques, such as NN. The principal advantage of NN is that, even if the precise relationship between input and output data streams is unknown but is acknowledged to exist, the network can be trained to learn that relationship. The use of the data *as recorded*, i.e. the total rainfall volumes instead of the rainfall excess volumes and the recorded stages or discharges instead of the direct runoff rates is an added incentive to avoid unnecessary empiricism. However, the user must be assured at the outset that the relevant input and output data sets have been selected in the first place.

To date, NN have been applied to model the rainfall-runoff relationships of anything from laboratory catchments (Hall & Minns, 1993; Carriere *et al.*, 1996) to drainage areas in excess of 19000 km^2 (Zealand *et al.*, 1999) – see Table 9.1. For the larger sizes of catchment, the use of stage or flow records for sites upstream of the outlet may be possible, so that the NN are implicitly routing hydrographs as part of the learning process (e.g. See & Openshaw, 1998). Indeed, NN have been applied directly for the routing of flood and stage hydrographs by Zhu and Fujita (1994), Raman and Sunilkumar (1995), Thirumallaiah and Deo (1998) and Teegavarapu (1998). Minns (1998, 2000) showed that for these types of simple advection and dispersion processes, a NN is capable of encapsulating

the same knowledge that is contained in the governing partial differential equations. In fact, the governing continuum equations could actually be restored by analysing the weights of the NN that had been trained with measured data.

The following discussion, however, is more specifically concerned with the art of rainfall-runoff modelling, which despite being a fertile area for exploration offers several less-than-obvious traps for the unwary. The first choice to be made by the NN modeller is the mode of presentation of the data to the network, i.e. how can the input and output patterns be defined? One possibility is to take the n successive ordinates of the rainfall hyetograph and feed these into the n input nodes of a network whose m output nodes carry the m successive ordinates of the flow hydrograph. This was the approach followed by both Smith and Eli (1995) and Lange (1998). In the former case the outputs were the coefficients from a truncated harmonic series representation of the hydrograph, which had the added advantage of already being standardised within the interval ± 1.

An alternative method of defining patterns is the so-called dynamic approach. In this case the input is a set of concurrent ordinates of (say) the rainfall totals from p raingauges within the catchment, and the output is the concurrent rate of outflow. Each time step defines a pattern, and therefore a single storm event provides as many exemplars as there are runoff ordinates, rather than only a single input-output pairing. This is the approach that has been adopted by the majority of writers on rainfall-runoff modelling using NN, but requires a much higher level of hydrological insight into the working of the catchment system. Within the dynamic approach, three types of model can be identified:

– naïve dynamic model;

$$Q(t) = f(r(t),\ r(t-1),\ldots) \tag{1}$$

– rainfall-runoff simulation model;

$$Q(t) = f(r(t),\ r(t-1),\ldots,Q(t-1),\ Q(t-2),\ldots) \tag{2}$$

– 'auto regressive' model;

$$Q(t) = f(Q(t-1),\ Q(t-2),\ldots) \tag{3}$$

where $Q(t)$ is the outflow at time level t and $r(t)$ is the rainfall ordinate at time level t.

The simplest, naïve dynamic rainfall-runoff model (1) would consist of a NN with inputs from one or more raingauges at time t, and an output of concurrent flow. A simple scatter plot of input(s) against output is sufficient in this case to indicate that the description of the input pattern is inadequate. Some improvement is obtainable by allowing for the time lag between the occurrence of the flow and the incidence of the causative rainfall. Since the flow at any instant is effectively composed of contributions from different sub-areas with highly variable travel times to the outlet, both the concurrent and antecedent rainfalls can be considered to be contributing to the outflow. Use of a moving window of rainfall at time

t and the k previous intervals provides some improvement (e.g. Karunanithi *et al.,* 1994).

The choice of length for the moving window of rainfall can significantly affect the accuracy of the resulting NN model. If the window is too short, the input data does not contain enough information about the entire rain event that is contributing to the concurrent outflow. The square shapes of the output hydrographs in Figure 9.1(a) seem to represent only the shape of the rainfall bursts and there is a very poor representation of the rising limbs and the recession limbs. The equilibrium discharges are also underestimated. Conversely, if the window is too long, the input contains too much information, which may even include historical rainfall events whose effects have long since passed out of the catchment and so are no longer contributing to the concurrent flow. Figure 9.1(c) is indicative of a NN that can no longer 'generalise' the relationship between rainfall and runoff. There is simply *too much* data being presented at the input layer and – to maintain the biological analogy – the NN gets 'confused'. The most accurate results, shown in Figure 9.1(b), are obtained using a moving window length that broadly encompasses the range of centroid-to-centroid lag times of the training data; a result that has some intuitive appeal. Details of the laboratory catchment experiment depicted in Figure 9.1 are given in Hall and Minns (1993).

A further problem with model (1) is that the simplistic input patterns may easily result in ambiguous results. More specifically, intervals with zero rainfall inputs are encountered in two contrasting situations. First immediately prior to the beginning of the storm and the start of the rising limb, and secondly some time after the end of the storm event when flows are moderately high and in recession. The NN has no information to discriminate between these two 'no-rainfall' conditions and once more becomes 'confused'. These conditions have to be differentiated by the addition of another input if the NN is to achieve the correct mapping. The most obvious candidate is adding a flow ordinate. This is most easily provided by making the output at time t an input at time $t + 1$ (Hall & Minns, 1993; Minns & Hall, 1996, and similar use of stage outputs by Campolo *et al.*, 1999). This approach is described above as model (2). In effect, the flow (or stage) ordinate is employed as a crude measure of catchment wetness. Figure 9.2 demonstrates the remarkable improvement in the NN model obtained by simply adding several antecedent flow ordinates to the input.

Model (3) is then the logical extension of models (1) and (2) into purely 'auto-regressive' time series prediction. This model does not make use of any rainfall data at all but uses only antecedent outflow values as input to the NN to predict the concurrent outflow. Figure 9.3 depicts the results of a NN that uses only the 5 antecedent flow values to predict the concurrent outflow.

Although the rising limb of the hydrograph in Figure 9.3 is not reproduced very accurately and the equilibrium flows are slightly underestimated, this NN model provides only slightly poorer performance than the results from model (2) in Figure 9.2. The overshoot at the top of each rising limb is caused by the fact that the network has no other information to tell it at which level the rising limb should stop until the actual measurements indicate that this has happened. That is, at the top of the rising limb, the output from the NN continues rising in

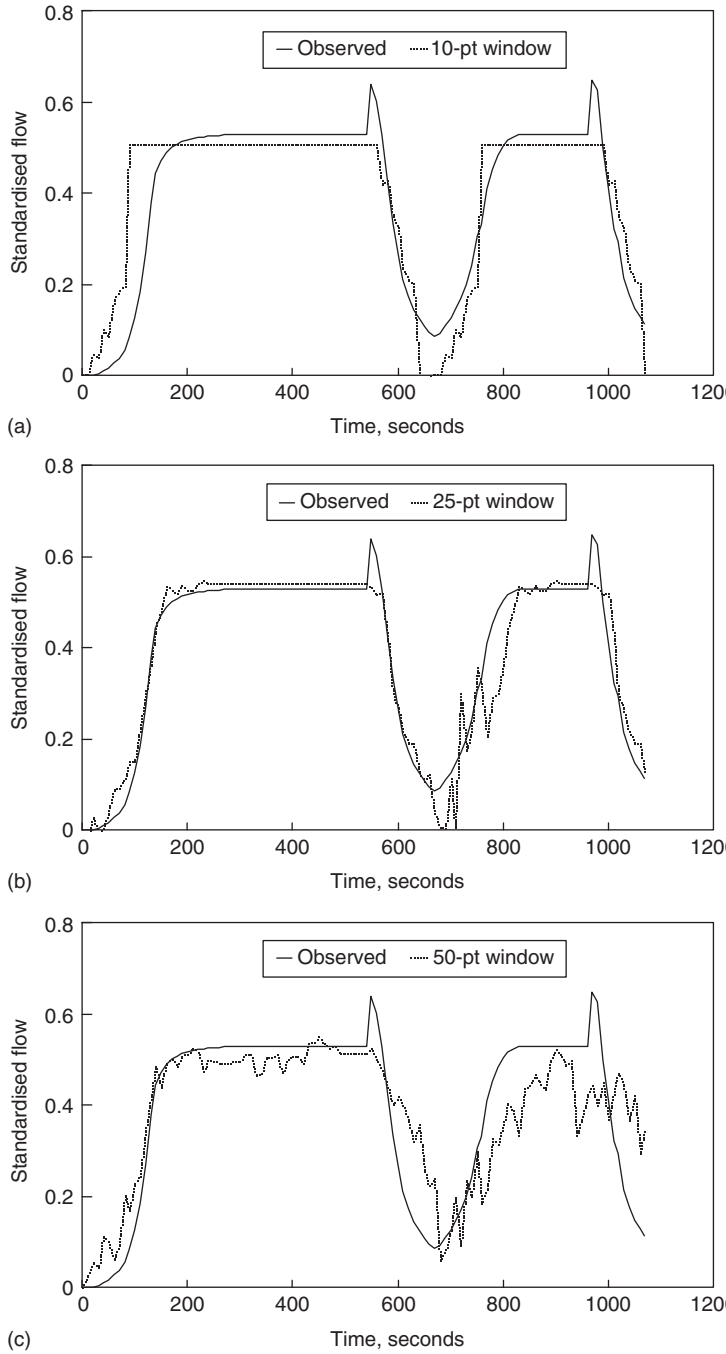


Fig. 9.1. Performance of NN on the verification event for a laboratory catchment using only 10-second-interval rainfall data as input for (a) 10-interval; (b) 25-interval; and (c) 50-interval windows.

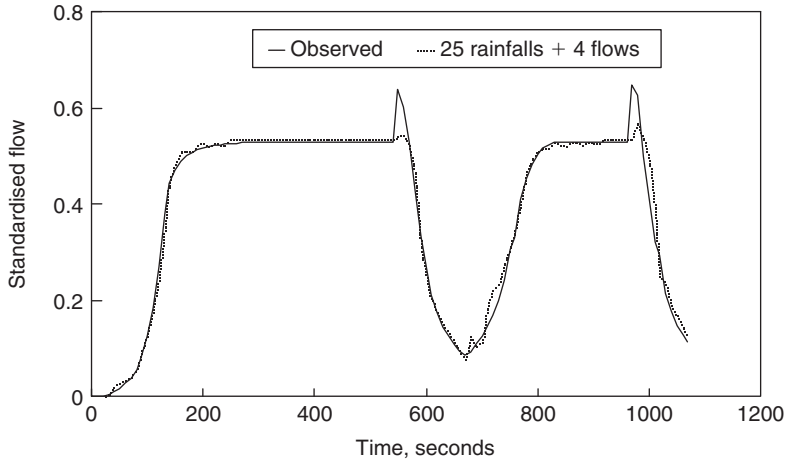


Fig. 9.2. Performance of NN on verification event for laboratory catchment using 25 rainfall ordinates and 4 antecedent flow ordinates at 10-second intervals as input.

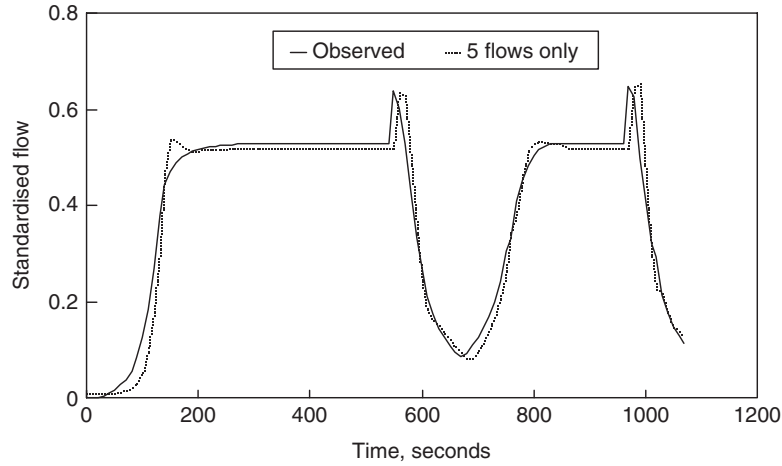


Fig. 9.3. Performance of NN on verification event for laboratory catchment using 5 antecedent flow ordinates only as input.

magnitude reflecting the pattern of the preceding flows. It is not until several time steps have passed for which the measured values are all constant, that the NN 'recognises' that the equilibrium level has been reached. Similarly, the phase error that occurs at the beginning of the recession limb is caused by the fact that the NN has no knowledge about the cessation of the rainfall until one or two time steps after the actual measured values start to decrease.

Figures 9.1–9.3 depict respectively the results of models (1), (2) and (3) when applied to the problem of one-step-ahead time series prediction. In general, the plethora of literature involving the application of NN to rainfall-runoff

modelling confirms the exceptional accuracy of NN models for short forecasting intervals. Longer forecasting intervals may be obtained by utilising $Q(t + 1)$, $Q(t + 2)$, … etc. as outputs during the training of the NN. Unfortunately, the performance of the NN decreases quite rapidly with an increasing prediction time horizon (Campolo *et al.*, 1999; Zealand *et al.*, 1999). Another approach is to use a trained NN with a 'feedback' loop in which the predicted output is used directly as input data for the subsequent time step. Unfortunately, the error accumulation associated with this approach also means that the performance of the NN deteriorates quite rapidly after only one or two iterations (van den Boogaard *et al.,* 1998). Although Abrahart (1998) describes a method to deal with the accumulated error, the most promising approach would appear to involve the use of partial recurrent neural networks (PRNN), which contain feedback loops in both the training and recall modes of the network. Hertz *et al.* (1991) describe the architecture of these so-called Jordan/Elman Recurrent Networks. Proaño *et al.* (1998) and van den Boogaard *et al.* (1998) show significant improvements in long-term predictions using PRNN (see Chapter 7), which they also refer to as 'auto regressive' NN.

In terms of the number of patterns that can be extracted from a given data set, the dynamic models described above are superior to those based simply on the rainfall hyetograph as the input and the complete hydrograph as the output. However, the problem then arises as to the set of time points for which those patterns are extracted. Here a clear perspective is required as to the purpose of the modelling, since with any series of discharges the (positive) skewness of its marginal distribution tends to increase as the time interval at which the data are recorded reduces. This effect is manifested in the appearance of sustained recessions to the hydrographs as the time interval becomes shorter, such that with (say) daily data they become the dominant features of the time series. In these circumstances, the rising limbs and peaks of the storm hydrographs form only a small portion of the total number of patterns in the time series, and the mapping of inputs to outputs is biased in favour of the recession behaviour. If therefore the purpose of the modelling is to capture the essence of the flood regime of the catchment, then the inputs should be restricted to the major storm events. For convenience, these inputs and outputs may be arranged in the form of an artificial time series in which the flow transitions between successive events are smoothed to provide continuity. In the absence of seasonal influences, this approach has been found to work satisfactorily (Hall & Minns, 1993; Minns & Hall, 1996, 1997; Minns, 1996; Campolo *et al.*, 1999). If the full range of flow behaviour is of interest, then an alternative approach might be to carry out a prior classification of event types or hydrograph features – perhaps employing a Kohonen network – and to develop a separate NN rainfall-runoff model for each class (e.g. See & Openshaw, 1998).

Dawson and Wilby (1998) have concluded that NN for long flow series at short time intervals should ideally be calibrated and validated on data for a common period of the year. Seasonal influences can, of course, be incorporated by extending the list of input variables. For example, Abrahart and Kneale (1997) and Abrahart (1998) employed an annual hour count converted into its sine and cosine equivalents to denote 'time of year'. Alternatively, Zealand *et al.* (1999)

added a 'period of the year' (in effect, a week number) and cumulative precipitation since the previous 1st November to the current time up to 1st April, to their inputs. The latter being intended as a measure of winter snowpack accumulation. Yet another approach to incorporating a seasonal variable is to use temperature data as an additional input (Lorrai & Sechi, 1995; Poff *et al.*, 1996; Zealand *et al.*, 1999).

Despite the potential significance of seasonal influences, the majority of NN rainfall-runoff models rely on combinations of current and antecedent rainfall totals and antecedent flow (or stage) ordinates as inputs (see Hsu *et al.*, 1995; Minns & Hall, 1996, 1997; Jayawardena & Fernando, 1998; Fernando & Jayawardena, 1998; Campolo *et al.*, 1999). More elaborate inputs derived from the basic records have been introduced in some studies. Mason *et al.* (1996) for example uses the derivative of the rainfall intensity and the integral of the rainfall intensity over the previous five time steps. Shamseldin (1997) defined a series of rainfall indices consisting of weighted sums of previous rainfall ordinates, the weights being derived from the ordinates of a gamma distribution.

The predominant objective of the rainfall-runoff models that have been developed using some form of NN has been that of forecasting future flows given the knowledge of past flows, rainfalls and other relevant variables. Unfortunately, the MLP-type of NN is not ideally suited to this application. This is largely because the sigmoidal activation function adopted by many authors imposes a scaling on the network output such that the network is incapable of predicting a flow larger in magnitude than that contained within the training data set. This effect is amply demonstrated by Minns and Hall (1996) based upon trials with synthetic data. NN modellers must be aware that when a NN is applied to a real catchment, even if the training data included all the available measurements, there remains a small, but non-negligible probability that an extreme event, beyond the range of recorded experience, might occur in the future, and not be correctly forecast.

An alternative approach suggested by Minns (1996) and Minns and Hall (1997) is to use the change in flow as the output rather than the absolute magnitude of the discharge. This variable was used independently by Zhu and Fujita (1994) for forecasting purposes, but without explanation. Change in stage was adopted as the forecast variable by See and Openshaw (1998), presumably because of the scaling problem outlined above. However, Karunanithi *et al.* (1994) claim that a (clipped) linear activation function allows extrapolation. This property arises due to the unbounded nature of the linear activation output. However, if this type of NN is applied with little or no hydrological insight, this apparent luxury of 'unlimited' extrapolation may lead to quite unacceptable linear extrapolations of some very non-linear hydrological processes. The results may then be not only meaningless, but also quite dangerous to apply! More recently, Imrie *et al.* (2000) have proposed the addition of a 'guidance system' to the output layer of a cascade correlation architecture of NN in order to assist with extrapolation beyond the range of the training data set. However, this approach depends upon information from the testing data set, and thereby disrupts the cycle of training and independent verification (see also Chapter 6).

3 EXAMPLE: A SMALL SEWERED CATCHMENT

In order to minimise the possible effects of seasonal variations in losses, Hall and Minns (1993) constructed a NN model using records from an urban catchment area. The Cantley Estate in Doncaster was gauged for a 3-year period in the late 1950s as part of a research programme carried out by the then Road Research Laboratory (see Watkins, 1962). The catchment, which has a gross area of 5.14 ha, is served by a separate surface water drainage system having an outfall 610 mm in diameter. The details of 16 storm events were kindly supplied by the Institute of Hydrology (now the Centre for Ecology and Hydrology), Wallingford. Twelve of these events were randomly selected for training and the remaining 4 reserved for verification. With data at one-minute intervals, there were therefore 985 data sets for training and 270 for verification.

Several different network configurations were trained and tested. A series of runs was carried out exploring the effect of changing the length of the rainfall window, and then adding antecedent flows to the input. Simultaneously, the number of nodes in the hidden layer was adjusted to eventually obtain the smallest possible network with the best generalisation properties. Rainfall-only input (i.e. model (1)) produced very noisy outputs, with peak flows significantly underestimated on some events and overestimated on others. In addition, the lower limbs of the recessions were too steep. The addition of 3 antecedent flows (i.e. model (2)) removed most of these undesirable features, although the highest peak flow rate was both underestimated in magnitude and late in timing. The results of the best-performing network configuration, with 18 inputs, 10 hidden nodes and 1 output, are presented in Figure 9.4.

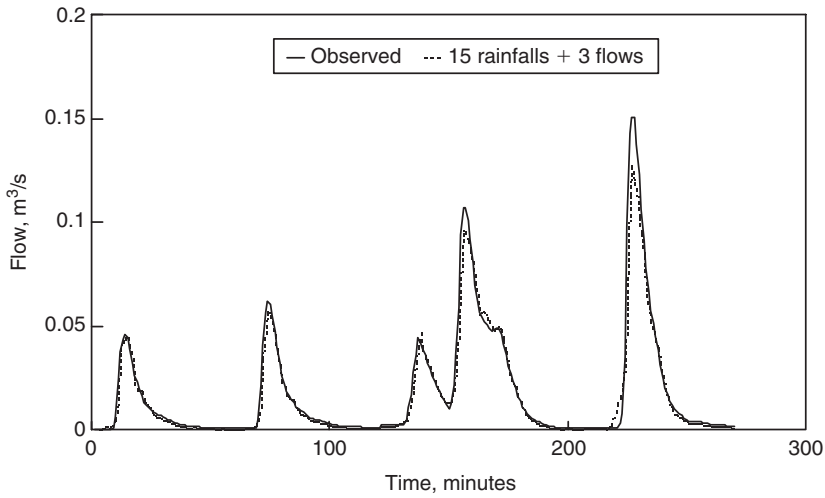


Fig. 9.4. Performance of NN on verification events for Cantley Estate sewerage system using 15 rainfall ordinates and 3 antecedent flow ordinates as input.

Having demonstrated that a NN with a suitable choice of inputs is capable of reproducing, with some fidelity, the responses to storm events upon which it has not been trained, the question arises as to whether the approach offers any advantages over a conventional black-box rainfall-runoff model. The 4 verification events from the Cantley Estate were therefore modelled separately by means of a well-established, conceptual hydrological modelling package, RORB (Mein *et al.*, 1974).

The basic element of the RORB model is a single, conceptual, non-linear reservoir for which the relationship between storage, S, and discharge, Q, is given by:

$$S = K_c K_r Q^m \tag{4}$$

where K_c is a storage constant applicable to all sub-areas within the catchment and K_r is a relative delay time applicable to individual channel reaches within the network estimated from the expression:

$$K_r = f \frac{L_i}{L_{\text{av}}} \tag{5}$$

where L_i is the length of the reach represented by the storage element, L_{av} is the average flow distance of sub-catchment inflows within the channel network, and f is a factor depending upon the type of channel reach, i.e. natural, lined or unlined.

For this experiment, the power of the non-linear reservoir, m, was set to the default value of 0.8, and the initial loss and storage constant manipulated until the peak flow rate and total runoff volume were satisfactorily reproduced. In the case of the event of 3 July, 1957, which was double-peaked, the rainfall was separated into two bursts, thereby introducing the ratio between the runoff volumes caused by each burst as a third calibration parameter. The plots of the single-peaked storm of 26 August, 1956 and the double-peaked event of 3 July, 1957 are displayed in Figures 9.5 and 9.6 respectively.

The results are compared in Table 9.2 in terms of their coefficients of efficiency as defined by Nash and Sutcliffe (1970). The coefficients of efficiency (COE) of the two models on these four events are generally comparable in magnitude. However, in the case of the event of 3 July 1957 (that with a pronounced double peak), the performance of the NN is obviously superior.

Some important factors should be considered when evaluating these results. Firstly, the calibration parameters for the RORB model included an initial loss rate, while application of the NN did not involve any consideration of loss separation. In fact, the NN has no calibration parameters as such, but only the set of weights which it learns itself. It thus involves no operator intervention and no *a priori* knowledge of the catchment. Moreover, since the RORB model was calibrated for each event individually but the NN operated on all 4 events with the same set of weights determined from the training, this comparison is inherently unfavourable to the NN model. Although the training of the NN requires a substantial investment in computer time, the procedure is far more

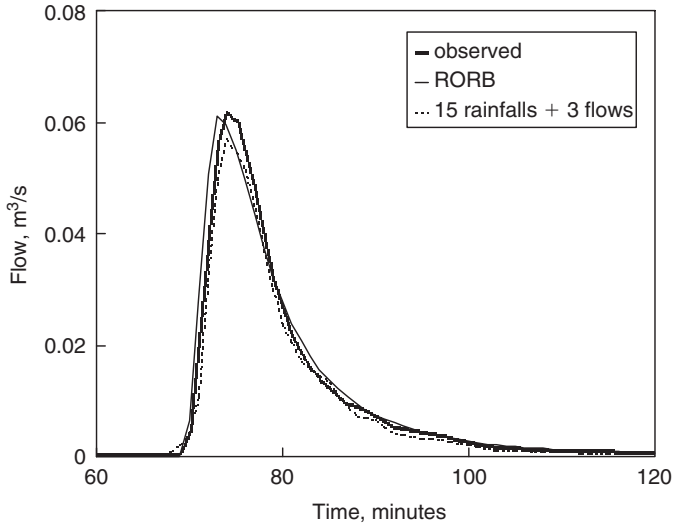


Fig. 9.5. Model comparison for storm of 26 August, 1956 (Cantley Estate).

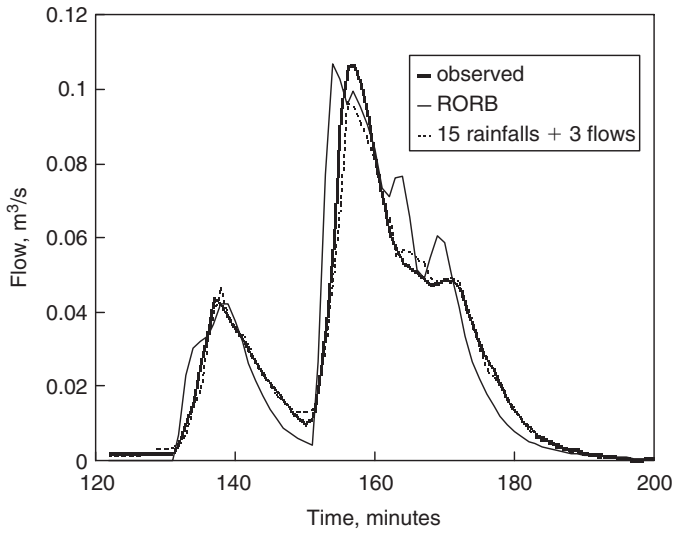


Fig. 9.6. Model comparison for storm of 3 July, 1957 (Cantley Estate).

straightforward than is the calibration of even a simple conceptual model, which must be undertaken on an event-by-event basis.

The results obtained are sufficient to demonstrate that, for situations involving rainfall-runoff modelling in which there are no extraneous influences such as land-use changes, a NN has the potential to perform in a comparable fashion, if not better than a conceptual hydrological model.

Table 9.2. Comparison between fit provided to four storm events by a NN model and the RORB conceptual model in terms of coefficients of efficiency (COE).

	Coefficients of efficiency	
Storm of	RORB	NN
23 August 1956	0.974	0.981
26 August 1956	0.983	0.982
3 July 1957	0.884	0.978
21 July 1957	0.990	0.951

4 CONCLUDING REMARKS

Since flood forecasting has emerged as one of the dominant applications of NN rainfall-runoff models, the problem of extrapolation beyond the confines of events in the training data set has to be carefully addressed. MLP having a sigmoidal activation function are not capable of departing very far outside the range of training events and this feature of their behaviour is not always well appreciated. The alternatives are either to choose a variable that is more constrained in its absolute values, such as change in flow or stage, or to adopt an alternative form of activation function, or even a different type of network. Definitive recommendations have yet to emerge on this critical issue. However, the further possibility of incorporating additional domain knowledge into the modelling process by adding a (synthetic) Estimated Maximum Flood to the training data set has been shown to hold considerable promise (see Hettiarachchi *et al.*, 2004).

The results of all of the numerical experiments reported to date indicate that suitably configured NN are capable of identifying usable relationships between runoff discharges and antecedent rainfall depths to an exceptional degree of accuracy. The relationships are obtained using only the raw, measured data and do not require the use of any derived or artificial calibration parameters. In particular, the NN model provides these exceptional results unhindered by constraints of volume continuity in the input and output data and, in fact, the units of the data are chosen simply for convenience of measurement and representation (e.g. rainfall depths in mm, discharges in m^3/s). Furthermore, simple, non-hydrological parameters like the percentage of impervious area may be easily incorporated into the model at the discretion of the modeller (see Minns & Hall, 1997). These types of parameters may be derived from simple measurements or may even be highly intuitive, and are likewise unrestricted in terms of conditions of dimension or hydrological-physical consistency.

The discussion above demonstrates that although a NN may be regarded as an ultimate form of 'black box' model (Minns & Hall, 1996), the potential user is not absolved from devoting some thought to the mode of presentation of data to the network. The principal question to be posed is: what exactly constitutes the pattern of inputs that produces the pattern of outputs? Moreover, do the selected input and output patterns contain additional information that is not strictly relevant to the purpose of the exercise? These questions are inevitably problem-dependent,

but in all cases the selection of inputs, whether data as recorded or variables derived from operations on recorded data, requires the application of hydrological insight as much as any conventional physical/conceptual rainfall-runoff model. Provided that such insight is applied, the performance of NN models can be undoubtedly superior to conventional hydrological models in situations that do not require more detailed knowledge of the hydrological system.

REFERENCES

Abbott, M.B., Bathurst, J.C., Cunge, J.A., O'Connell, P.E. & Rasmussen, J. 1986. An introduction to the European Hydrological System – Système Hydrologique Européen, 'SHE', 2: Structure of a physically-based, distributed modelling system. *Journal of Hydrology* 87: 61–77.

Abbott, M.B. & Refsgaard, J.C. 1996. *Distributed Hydrological Modelling*. Dordrecht, The Netherlands: Kluwer.

Abrahart, R.J. 1998. Neural networks and the problem of accumulated error: an embedded solution that offers new opportunities for modelling and testing. In *Proceedings Hydroinformatics '98*. 2: 725–731. Rotterdam: Balkema.

Abrahart, R.J. & Kneale, P.E. 1997. Exploring neural network rainfall-runoff modelling. In *Proceedings 6th National Hydrological Symposium*, British Hydrological Society, London, UK, 9.35–9.44.

Aleksander, I. & Morton, H. 1990. *An Introduction to Neural Computing*. London: Chapman and Hall.

Aly, A.H. & Peralta, R.C. 1999. Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm. *Water Resources Research* 35 (8): 2523–2532.

Amorocho, J. & Hart, W.E. 1964. A critique of current methods in hydrologic systems investigation. *Transactions of the American Geophysical Union* 45: 307–321.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology 2000. Artificial neural networks in hydrology. II: Hydrologic applications. *Journal of Hydrological Engineering* 5: 124–137.

Boogaard, H.F.P. van den, Gautam, D.K. & Mynett, A.E. 1998. Auto-regressive neural networks for the modelling of time series, In *Proceedings Hydroinformatics '98*. 2: 741–748. Rotterdam: Balkema.

Beale, R. & Jackson, T. 1990. *Neural Computing: An Introduction*. Bristol, UK: Institute of Physics.

Bras, R.L. 1990. *Hydrology. An Introduction to Hydrologic Science*. Massachusetts, USA: Addison-Wesley.

Campolo, M., Andreussi, P. & Soldati, A. 1999. River flood forecasting with a neural network. *Water Resources Research* 35(4): 1191–1197.

Carriere, P., Mohaghegh, S. & Gaskari, R. 1996. Performance of a virtual runoff hydrograph system. *Journal of Water Resources Planning and Management* 122(6): 421–427.

Dawson, C.W. & Wilby, R. 1998. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal* 43: 47–66.

Dawson, C.W. & Wilby, R. 2001. Hydrological modelling using artificial neural networks. *Progress in Physical Geography* 25: 80–108.

Dibike, Y.B., Solomatine, D, & Abbott, M.B. 1999. On the encapsulation of numerical-hydraulic models in artificial neural networks. *Journal of Hydraulics Research* 37(2): 147–161.

Elizondo, D., Hoogenboom, G. & McClendon, R.W. 1994. Development of a neural network model to predict daily solar radiation. *Agricultural and Forest Meteorology* 71, 115–132.

Fernando, D.A.K. & Jayawardena, A.W. 1998. Runoff forecasting using RBF networks with OLS algorithm. *Journal of Hydrological Engineering* 3(3): 203–209.

French, M.N., Krajewski, W.F. & Cuykendall, R.R. 1992. Rainfall forecasting in space and time using a neural network. *Journal of Hydrology* 137: 1–31.

Hall, M.J. & Minns, A.W. 1993. Rainfall-runoff modelling as a problem in artificial intelligence: experience with a neural network. In *Proceedings 4th National Hydrological Symposium.* British Hydrological Society, London, UK. 5.51–5.57.

Hall, M.J. & Minns, A.W. 1998. Regional flood frequency analysis using artificial neural networks. In *Proceedings Hydroinformatics '98.* 2: 759–763. Rotterdam: Balkema.

Hall, M.J. & Minns, A.W. 1999. The classification of hydrologically homogeneous regions*. Hydrological Sciences Journal* 44: 693–704.

Hall, M.J., Minns, A.W. & Ashrafuzzaman, A.K.M. 2000. Regional flood frequency analysis using artificial neural networks: a case study. In *Proceedings Hydroinformatics 4th International Conference, Iowa, USA 23–27 July 2000*.

Hall, M.J., Minns, A.W. & Ashrafuzzaman, A.K.M. 2002. The application of data mining techniques for the regionalisation of hydrological variables. *Hydrology and Earth System Sciences* 6: 685–694.

Hertz, J., Krogh, A. & Palmer, R.G. 1991. *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley.

Hettiarachchi, P., Hall, M.J. & Minns, A.W. 2004. The extrapolation of artificial neural networks for the modelling of rainfall-runoff relationships. Under review.

Hsu, K., Gupta, H.V. & Sorooshian, S. 1995. Artificial neural network modelling of the rainfall-runoff process. *Water Resources Research* 31(10): 2517–2530.

Imrie, C.E., Durucan, S. & Korre, A. 2000. River flow prediction using artificial neural networks: generalisation beyond the calibration range. *Journal of Hydrology* 233: 138–153.

Jayawardena, A.W. & Fernando, D.A.K. 1998. Use of radial basis function type artificial neural networks for runoff simulation. *Computer-aided Civil and Infrastructure Engineering* 13: 91–99.

Karunanithi, N., Grenney, W.J., Whitley, D. & Bovee, K. 1994. Neural networks for river flow prediction. *Journal Computing in Civil Engineering* 8(2): 201–220.

Lange, N.T.G. 1998. Advantages of unit hydrograph derivation by neural networks. In *Proceedings Hydroinformatics '98.* 2: 783–789. Rotterdam: Balkema.

Lorrai, M. & Sechi, G.M. 1995. Neural nets for modelling rainfall-runoff transformations. *Water Resources Management* 9: 299–313.

Luk, K.C., Ball, J.E. & Sharma, A. 1998. Rainfall forecasting through artificial neural networks. In *Proceedings Hydroinformatics '98*. 2: 797–804. Rotterdam: Balkema.

Maier, H.R. & Dandy, G.C. 1996. The use of artificial networks for the prediction of water quality parameters. *Water Resources Research* 32(4): 1013–1022.

Maier, H.R. & Dandy, G.C. 1999. Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Journal Environmental Modelling and Software* 15(1): 101–123.

Mason, J.C., Price, R.K. & Tem'me, A. 1996. A neural network model of rainfall-runoff using the radial basis function. *Journal of Hydraulics Research* 34(4): 537–548.

Mein, R.G., Laurenson, E.M. & McMahon, T.A. 1974. Simple nonlinear model for flood estimation. *Proceedings of the American Society of Civil Engineers, Journal of the Hydraulics Division*, No. 100 (HY11): 1507–1518.

Minns, A.W. 1996. Extended rainfall-runoff modelling using artificial neural networks. In *Proceedings Hydroinformatics '96*. 207–213. Balkema, Rotterdam.

Minns, A.W. 1998. *Artificial Neural Networks as Subsymbolic Process Descriptors*. Rotterdam: Balkema.

Minns, A.W. 2000. Subsymbolic methods for data mining in hydraulic engineering. *Journal of Hydroinformatics* 2(1): 3–13.

Minns, A.W. & Hall, M.J. 1996. Artificial neural networks as rainfall-runoff models. *Hydrological Sciences Journal* 41: 399–417.

Minns, A.W. & Hall, M.J. 1997. Living with the ultimate black box: more on artificial neural networks. In *Proceedings 6th National Hydrological Symposium*, British Hydrological Society, London, UK. 9.45–9.49.

Muttiah, R.S., Srinivasan, R. & Allen, P.M. 1997. Prediction of two-year peak stream discharges using neural networks. *Journal American Water Resources Association* 33: 625–630.

Nash, J.E. & Sutcliffe, J.V. 1970. River flow forecasting through conceptual models, I. A discussion of principles. *Journal of Hydrology* 10: 282–290.

Poff, N.L., Tokar, S. & Johnson, P. 1996. Stream hydrological and ecological responses to climate change assessed with an artificial neural network. *Limnology and Oceanography* 41(5): 857–863.

Proaño, C.O., Minns, A.W., Verwey, A. & Boogaard, H.F.P. van den 1998. Emulation of a sewerage system computational model for the statistical processing of large numbers of simulations. In *Proceedings Hydroinformatics '98*. 2: 1145–1152. Rotterdam: Balkema.

Raman, H. & Sunilkumar, N. 1995. Multivariate modelling of water resources time series using artificial neural networks. *Hydrological Sciences Journal* 40: 145–163.

Ranjithan, S., Eheart, J.W. & Garrett, J.H. 1993. Neural network-based screening for groundwater reclamation under uncertainty. *Water Resources Research* 29(3): 563–574.

Refsgaard, J.C., Seth, S.M., Bathurst, J.C., Erlich, M., Storm, B., Jørgensen, G.H. & Chandra, S. 1992. Application of SHE to catchments in India, Part 1. General results. *Journal of Hydrology* 140: 1–23.

Rizzo, D.M. & Dougherty, D.E. 1994. Characterisation of aquifer properties using artificial neural networks: neural kriging. *Water Resources Research* 30(2): 483–497.

Rogers, L.L. & Dowla, F.U. 1994. Optimisation of groundwater remediation using artificial neural networks with parallel solute transport modelling. *Water Resources Research* 30 (2): 457–481.

Rogers, L.L., Dowla, F.U. & Johnson, V.M. 1995. Optimal field-scale groundwater remediation using artificial neural networks and the genetic algorithm. *Environmental Science and Technology* 29: 1145–1155.

Schaap, M.G. & Bouten, W. 1996. Modelling water retention curves of sandy soils using neural networks. *Water Resources Research* 32(10): 3033–3040.

See, L. & Openshaw, S. 1998. Using soft computing techniques to enhance flood forecasting on the River Ouse. In *Proceedings Hydroinformatics '98.* 2: 725–731. Rotterdam: Balkema.

Shamseldin, A.Y. 1997. Application of a neural network technique to rainfall-runoff modelling. *Journal of Hydrology* 199: 272–294.

Smith, J. & Eli, R.N. 1995. Neural-network models of rainfall-runoff process. *Journal Water Resources Planning and Management* 121(6): 499–508.

Teegavarapu, R. 1998. Input structures for a neural network model used for streamflow forecasting. In *Hydrology in a Changing Environment.* Proceedings First British Hydrological Society International Conference, Wiley, Chichester, 3: 105–114.

Thirumalaiaah, K. & Deo, M.C. 1998. River stage forecasting using artificial neural networks. *Journal of Hydrological Engineering* 3(1): 26–32.

Watkins, L.H. 1962. The design of urban drainage systems. Research into the relation between the rate of rainfall and the rate of flow in sewers. *Road Research Technical Paper,* No. 55, HMSO, London.

Wen, C-G. & Lee, C-S. 1998. A neural network approach to multiobjective optimisation for water quality management in a river basin. *Water Resources Research* 34(3): 427–435.

Whitley, R. & Hromadka, T.V. 1999. Approximate confidence intervals for design floods for a single site using a neural network. *Water Resources Research* 35(1): 203–209.

Zealand, C.M., Burn, D.H. & Simonovic, S.P. 1999. Short term streamflow forecasting using artificial neural networks. *Journal of Hydrology* 214: 32–48.

Zhu, M.L. & Fujita, M. 1994. Comparisons between fuzzy reasoning and neural network methods to forecast runoff discharge. *Journal of Hydroscience & Hydraulic Engineering* 12(2): 131–141.

10

A Neural Network Approach to Rainfall Forecasting in Urban Environments

JAMES E. BALL AND KIN CHOI LUK
Water Research Laboratory, School of Civil and Environmental Engineering, The University of New South Wales, Sydney, Australia

ABSTRACT: An effective flood warning system in urban areas must provide the warning with sufficient lead time for an appropriate response by the relevant emergency services and the affected community. This requirement poses a critical problem as most urban catchments are characterised by a fast hydrologic response to storm events. The approach used here to forecast rainfall over the Upper Parramatta River Catchment in Sydney is based on the application of a pattern recognition technique using an artificial neural network. It assumes that the future rainfall is a function of a discrete number of past spatial and temporal rainfall records; an important task, therefore, is the determination of the number of spatial and temporal rainfall records necessary for accurate prediction of future rainfall. The rainfall prediction model performed best when an optimal amount of spatial and temporal rainfall information was provided to the network.

1 INTRODUCTION

Flash floods are a life-threatening phenomenon, which also result in economic losses and social disruption. Handmer *et al.* (1988), for example, estimated direct economic losses for residential property in the Toongabbie Creek catchment (a subcatchment of the Upper Parramatta River, in the western suburbs of Sydney, Australia) as being approximately $5 million (1986 Australian dollars) for the 1% Annual Exceedance Probability (AEP) event. Commercial and industrial activities within the catchment were also estimated to have suffered economic losses of a similar magnitude. In the decade since these estimates were made, significant additional urbanisation, which decreases the potential

response time of warning systems, together with inflation, make such potential losses an underestimate in present day terms.

Development of an effective flood warning system can be expected to mitigate these losses. For an effective flood warning system, there needs to be sufficient time between the recognition of a potential flood event and its occurrence, for dissemination of flood warning messages and the activation of appropriate emergency services. More accurate forecasts of a flood should help to increase the time between the recognition of a potential flood event and its subsequent occurrence. This need has prompted the development of an effective rainfall forecasting system, wherein it must be recognised that rainfall is a dynamic process, which varies in space and time. There is a need to consider the spatial variability in a point-based rainfall forecasting model and to transform the point data output into areal distributions using spatial tools within a Geographic Information System (GIS).

There are two basic approaches suitable for the development of a rainfall forecasting model. These can be categorised as (i) the process model approach, in which the physical processes influencing rainfall are analysed and process models are developed, although this approach may not be feasible because:

- rainfall is a complex dynamic system which varies both in space and time resulting in problems associated with the definition of solution space boundaries;
- even if the rainfall processes can be described concisely and completely, the volume of calculations involved may be prohibitive; and
- the data that are available to assist in the definition of control variables for the process models, such as pressure, evaporation, wind speed and direction are limited in both the spatial and temporal dimensions.

and (ii) the 'black box' model approach, in which pattern recognition technology is used to predict the most likely future pattern of rainfall in time and space. The aim is to extract from the historical rainfall records the essential patterns necessary for the prediction of future rainfall events. There are many alternative techniques for the extraction of the essential features from historical records; the technique used here is based on artificial neural networks (NN).

Rainfall forecasts at rain gauge locations provide only scattered data and forecasts of rainfall over a catchment. The rainfall forecasts may have a fine temporal resolution, but the true areal rainfall, which produces runoff, is not known. This highlights a critical problem in conventional rainfall-runoff modelling where simplified approaches, such as Thiessen Polygons, are used without taking either the spatial distribution or the dynamic properties of the rainfall into account. These simplified approaches can result in large errors in runoff estimation (Fontaine, 1991; Urbonas *et al.*, 1992). In response to this need, Ball and Luk (1998) developed a method to model the spatial variability of rainfall using point measurements of rainfall as the input information. NN rainfall forecasts can therefore be passed to this model for the determination of future rainfall patterns over a catchment. The integration of these tools provides a powerful forecasting solution.

2 THE UPPER PARRAMATTA RIVER CATCHMENT

2.1 *Catchment details*

The Upper Parramatta River Catchment is used as the study area for the development of an integrated rainfall forecasting system. This river is situated in the western suburbs of Sydney and drains into Sydney Harbour (Fig. 10.1 and 10.2). The tidal limit of the river is the Charles Street Weir. Immediately upstream of

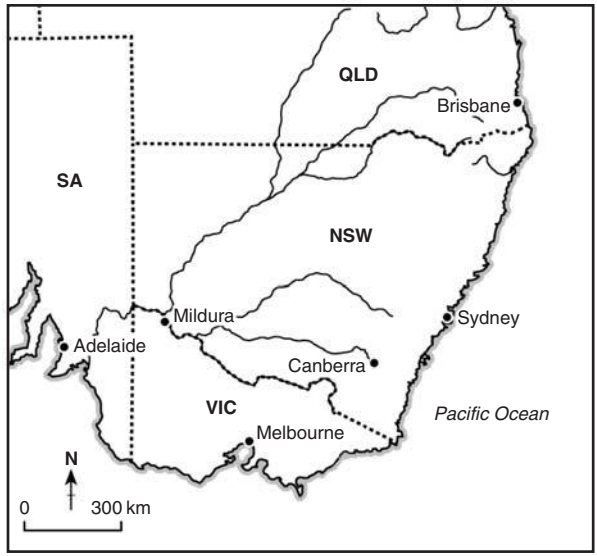


Fig. 10.1. Location of Sydney, Australia.

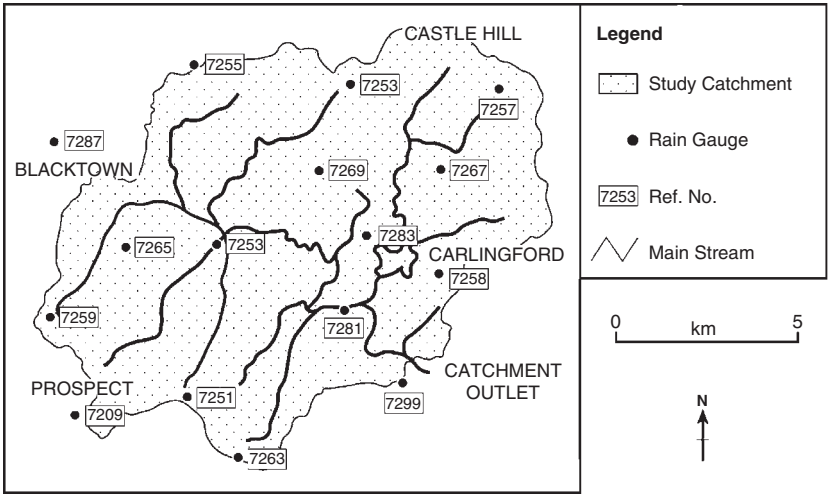


Fig. 10.2. Rain gauges within the Upper Parramatta River Catchment.

Table 10.1. Population and dwellings in Parramatta.

Year	Total population	% change over past five years	Total occupied dwellings	% change over past five years
1976	348398	–	100246	–
1981	374190	7.4	111064	10.8
1986	384601	2.8	119229	7.4
1991	435478	13.2	140900	18.2

the Weir, is the Parramatta Central Business District which has suffered considerable flood damage over a number of years. There are two main tributaries of the Parramatta river within the Upper Parramatta River Catchment: the Toongabbie Creek and Darling Mills Creek. The catchment is steep with an average slope of 1.2%.

The dominant land use is a mix of residential, commercial, industrial and open space (parkland) areas. Considerable development as a result of the rapid increase in population and dwellings has occurred within this catchment over the past two decades, as shown in Table 10.1 (Australian Bureau of Statistics, Census of Population and Housing, personal communication). The increase in the number of dwellings from 1986 to 1991 is 18%. A rough estimation of the increase in impervious area can be obtained by assuming an average dwelling size of 200 square metres and allowing 50% of this area for associated impervious area such as roads and footpaths. Using this approach, an estimate for the increase in impervious area is 6.5 km^2 or 6%. Both increases are significant from a hydrological and flood management viewpoint.

The effect of this urbanisation has been continuing increases in estimates of the peak level for all flood events. To mitigate the social and economic losses associated with these floods the Upper Parramatta River Catchment Trust (UPRCT) was instituted in 1989, with the task of managing flood mitigation measures within the catchment, among other duties.

2.2 *Rainfall records*

There are sixteen continuously recording rain gauges within the catchment (Fig. 10.2). The majority of these gauges have been installed by the UPRCT since its formation. Consequently, long-term records are not available, and there is, on average, one point rainfall sample for each 7 km^2 of catchment. Moreover, although this would represent a high density of rain gauge information for most catchments, Urbonas *et al.* (1992) suggest that an even higher density of spatial information is required if accurate predictions of catchment response are to be obtained for convective storm events (Table 10.2).

Records from the 16 rain gauges within the Upper Parramatta River Catchment were obtained from January 1991 to September 1996. During this period, 34 storm events occurred where the daily rainfall total exceeded 20 mm. More than 70% of such storms were convective and the rest were frontal. The convective storms occurred predominantly during the summer and autumn seasons,

Table 10.2. Accuracy of rainfall-runoff models (after Urbonas *et al.*, 1992).

Gauge density (km^2/gauge)	Range (%)	Mean deviation (%)
8.0	−100.0 to 150.0	−24.2
4.0	−75.3 to 94.5	0.5
2.7	−32.2 to 63.66	15.8
2.0	−32.2 to 18.8	−0.9
1.6	0.0 to 0.0	0.0

while the frontal storms were more evenly distributed through the year. The data series were extracted in 15-minute intervals. Missing rainfall values, due to malfunctioning gauges or errors in transmission of the data, were estimated from neighbouring rain gauges using the spatial rainfall model of Ball and Luk (1996, 1998).

3 RAINFALL FORECASTING – POINT PATTERNS

3.1 *Building a neural network solution*

To develop the proposed rainfall forecasting model, the continuous process of rainfall was represented using a discrete Markovian process, in which the rainfall value at a given location in space and time is a function of a finite set of previous realisations. With this assumption, a simple model structure can be expressed as:

$$X(t+1) = f[X(t), X(t-1), X(t-2), \ldots, X(t-k-1)] + e(t) \qquad (1)$$

where $\boldsymbol{X}(t) = [x_{1t}, x_{2t}, \ldots, x_{Nt}]T$ represents a vector of rainfall values $x_{1t}, x_{2t},..,$ x_{Nt} at N different gauge sites at time t, where T denotes the transpose operator, $f[\ ]$ is a non-linear mapping function, which shall be approximated using NN, $e(t)$ is a mapping error (to be minimised) and k is the (unknown) number of past realisations contributing to rainfall at the next time step, referred to as the model lag. If k is equal to 1, rainfall at the next time step is related only to the present rainfall, representing a lag-1 model.

Multi-layer feed-forward neural networks (MLFN) offer a straightforward approach to represent the above rainfall model. Further particulars about the workings of this network are provided in Chapter 2. The MLFN is presented with the current and past rainfall values as inputs, e.g. $X(t)$, …, $X(t - k + 1)$, and the next rainfall value $X(t + 1)$ is used as the network output. There are, however, several drawbacks associated with this approach. First, since the model lag k is unknown, a lengthy trial process is required to determine the optimal value of k. Second, for a network with high orders of lag, a large number of input nodes would be required. Consequently, the number of parameters will increase, making the network unnecessarily complex and with a higher risk of overfitting. Finally, the MLFN is a static model, which might struggle

or fail to model the dynamic nature of rainfall processes. Further details on MLFN are provided in Chapter 2.

Time delay neural networks (TDNN) offer an alternative solution, which can effectively model the rainfall process while keeping a minimum number of parameters. More details about this type of network are provided in Chapter 5. A distinctive feature of TDNN is the use of partial connections; this dramatically reduces the number of weights in the network compared with a fully connected MLFN architecture. In addition, TDNN have been developed for detecting local features within a larger pattern; this feature detection ability is very useful for the task of rainfall forecasting. Each TDNN, however, is still a static model. This static representation has several drawbacks. First, if the rainfall process has a long term memory, a large number of inputs nodes will be required, resulting in a network containing a large number of free parameters. Second, the number of past rainfall inputs has to be determined through the process of investigation, which often requires a lengthy series of trial and error experiments. A dynamic model may overcome this problem. Further details on TDNN are provided in Chapter 5.

NN with feedback connections that feed past states of the system back to the network can be used to build dynamic models. This is a recurrent network; recurrent networks exhibit full connection between each node and all other nodes in the network whereas partial recurrent networks contain a limited number of specific feedback loops. A recurrent network possesses the characteristic of dynamic memory. In addition, a recurrent network reduces the number of inputs and consequently the number of parameters, speeding up the calculations. For practical applications, the partial recurrent neural network (PRNN) is more appropriate because the training of such networks is similar to that of the MLFN and is therefore also much easier than a recurrent network. In partial recurrent networks the main network structure is feedforward and the feedforward connections are trainable. The feedback connections are formed through a set of 'context' units that are not trainable, which simplifies the training process. The function of the context units is to store information from the previous time steps. To achieve this, the context units make a copy of the activation of hidden nodes in the previous time step. Therefore, at time t the context units have some signals related to the state of the network at time $t - 1$. As a result, the rainfall at time $t + 1$ is a function of the rainfall at time t and the previous states of the system represented by the activation of the hidden nodes at time $t - 1$, expressed by

$$\boldsymbol{X}(t+1) = g(\boldsymbol{X}(t), O(t-1)) + e(t) \tag{2}$$

where $\boldsymbol{X}(t + 1)$ are rainfall at time $t + 1$, which are outputs of the network, $\boldsymbol{X}(t)$ are rainfall at time t, which are inputs of the network, $O(t - 1)$ are the activations of the hidden nodes at time $t - 1$ and copied back to the context units for input at time t, $g(\,)$ is a recurrent mapping function and $e(t)$ is the mapping error. Further details of PRNN are provided in Chapter 7.

3.2 *Neural network methodology*

MLFN, TDNN and PRNN models were developed for forecasting the storm events occurring over the Upper Parramatta River Catchment. The PRNN and TDNN were specifically developed to model the structures in time series, so they are considered to be the most suitable candidates for the current study. The MLFN, however, is the most popular model and has a relatively simple structure. The MLFN was included in this study to provide a base line for comparison. The networks were developed through: (i) data preparation, including data pre-processing, (ii) selection of training algorithm and performance indicators, and (iii) determination of the appropriate inputs and outputs.

In the data preparation step, two data sets were established for training and testing the network; the third data set was used for validating the training to ensure that the network learns the pertinent information and not the noise associated with the data used for training the network. To obtain unbiased samples for each of the data sets, the 34 storm events were divided randomly into:

- a training set – 16 storm events with a total of 748 rainfall periods. This data set was used to calibrate the connection weights of the various networks tested.
- a validation set – 8 storm events with a total of 376 rainfall periods. This data set was used to monitor the performance of the training and to provide an indication of when to cease training.
- a testing set – 10 storm events with a total of 625 rainfall periods. This final data set was used to evaluate the performance of the networks on data previously unseen by the network.

The selection of individual events for a particular data set was random; hence all events should have similar characteristics (Table 10.3).

Prior to training, the data were scaled to a smaller range [0, 1], which is associated with the choice of a sigmoid activation function. There are several alternative data transformation approaches that can be used, with each approach having its own advantages and disadvantages. In this case, a logarithmic algorithm was used to ensure that the recorded rainfall values were transformed into the desired range.

The next step in the implementation of the network is the selection of the training algorithm and performance indicator. The normalised mean squared error (NMSE) was chosen as the performance indicator for a comparison between the three alternative types of network. One problem with the use of sum squared error for the network comparison is that the rainfall series had different lengths, which introduces problems. This problem was overcome with a

Table 10.3. Summary of storm characteristics.

Characteristics	Training	Validation	Testing
Storm type	10 convective, 6 frontal	4 convective, 4 frontal	8 convective, 2 frontal
Storm duration (hours)	3 to 22	2 to 21	6 to 24
Time to max. rainfall	30 min to 10 hr	1 hr to 18 hr	45 min to 21 hr

normalised version of sum squared error, i.e. the normalised mean squared error (NMSE). Weigend *et al.* (1992) defined the NMSE as

$$\mathrm{NMSE} = \frac{\sum_N \sum_P (d_{np} - y_{np})^2}{\sum_N \sum_P (d_{np} - \bar{d}_{np})^2} \approx \frac{1}{NP\sigma^2} \sum_N \sum_P (d_{np} - y_{np})^2 \qquad (3)$$

where N is the total number of output nodes, P is the total number of data samples, d_{np} are the target outputs, y_{np} are the network outputs and σ^2 is the variance of the target outputs. NMSE is, in essence, the sum of squared errors normalised by the number of data samples over all output nodes and the estimated variance of the data.

The final step in the data preparation is the determination of the input and output data representation. For the Upper Parramatta River Catchment, there are a number of possible configurations for the input and output information. The three most feasible approaches are to:

- divide the catchment into grids (439 cells of 500 m × 500 m) and use rainfall at each cell as inputs to forecast rainfall at all cells simultaneously. The resulting outputs will be the rainfall at each cell of the catchment;
- use rainfall at the 16 gauges as inputs to forecast rainfall for the 16 gauges simultaneously. In this case, one network represents the rainfall for all 16 gauges, and from the rainfall forecasts for the 16 gauges, a spatial rainfall model is used to generate the rainfall at all points of the catchment; or
- use rainfall at the 16 gauges as inputs to forecast rainfall for a single gauge. This will result in 16 networks for the 16 gauges of the catchment. Again, after the rainfall forecasts for the 16 gauges are obtained, a spatial rainfall model is used to generate the rainfall at all points in the catchment.

Initial assessment of these three configurations suggests that the option of using information from the 16 rain gauges as input information and using the same locations as the output information (i.e. rainfall forecast) is the most desirable option. The main reasons for this are:

- information from all measurement points are used simultaneously to produce forecasts for each of the measurement points; and
- forecast results can be used readily in an existing spatial rainfall model.

The option of forecasting a single gauge was rejected because the same process was required once for each prediction location and, therefore, 16 networks were needed. In a similar manner, the option of using 439 cells as both input and output was rejected, since it involved the use of a large number of input and output nodes and because the consequent network would contain a large number of parameters (weights). For example, a one-hidden layer MLFN with 439 input nodes, 439 output nodes and 2 hidden nodes comprises 1756 connections ($439 \times 2 + 439 \times 2$, excluding the biases). The available rainfall data (max. 1749 data points) were not considered sufficient information to train a network of this size and nature.

Table 10.4. MLFN with different architectures and time lags.

	Normalised Mean Squared Error (NMSE)	
Network	Training	Testing
Lag-1 MLFN with 2 hidden nodes	0.53	0.71
Lag-1 MLFN with 128 hidden nodes	0.40	1.20
Lag-2 MLFN with 2 hidden nodes	0.51	0.73
Lag-2 MLFN with128 hidden nodes	0.36	0.96
Lag-3 MLFN with 2 hidden nodes	0.49	0.72
Lag-3 MLFN with 128 hidden nodes	0.32	1.26
Lag-4 MLFN with 2 hidden nodes	0.49	0.78
Lag-4 MLFN with 128 hidden nodes	0.27	2.33

Note: Each network was trained for 1000 epochs.

3.3 *Development of alternative networks*

The three alternative types of NN considered in this chapter were trained and tested with rainfall data collected from the catchment, i.e. a total of 34 storm events with depth recorded every 15-minutes. Various network configurations were explored to determine the effect of the key variables: lag of the network and the number of hidden nodes.

MLFN solutions comprised networks with 1, 2, 3 and 4 lags and 2, 4, 8, 16, 24, 32, 64, and 128 hidden nodes. We also explored MLFN with two hidden layers. For TDNN, the size of input windows was 2, 3 and 4, and for PRNN, the lag was fixed at 1, while the number of context units tried was 2, 4, 8, 16, 24, 32 and 64. The complete listing of all modelling results using these parameters is presented in Luk (1998).

Initial investigation of network performance with various numbers of hidden nodes indicated that the networks with a greater number of hidden nodes resulted in a lower training error, at a fixed stopping condition, which was set at 1000 epochs and where one epoch represented a complete sweep through the training patterns. Moreover, since the connection weights of the network were updated only at the end of each epoch, a maximum of 1000 epochs also means that the weights were updated at most 1000 times. This result is to be expected since more hidden nodes equates to more free parameters, which results in lower training errors. During testing, however, these solutions had poorer performance since the networks had over-learned the training data. This effect is illustrated in Table 10.4. Irrespective of the order of lag in the network, MLFN with 128 hidden nodes produced a smaller NMSE during training, but a much higher NMSE during testing. For example, the lag-4 MLFN with 128 hidden nodes had the smallest training error of 0.27 while the testing error for the same network was the highest at a value of 2.33.

The influence of the order of lag can also be assessed from Table 10.4. Perusal of the data shown in this table indicates that MLFN with higher order lags tended to learn the training data better. For validation, however, the reverse is the case. These results do not suggest that a network with higher order lag will give poorer results. The results in fact indicate that networks with higher order

lag, which contain more connection weights and hence more free parameters, like networks with more hidden nodes, tended to over-learn the training data.

From this analysis it was concluded that the performance of a network depended more on the complexities of the network than the inclusion of additional information through the use of data from previous time periods. The effect of using higher orders of lag was to increase such complexities without the provision of pertinent additional information. Consequently, it is suggested that the performance of a network depends not on the number of hidden nodes or the order of lag but rather on the combination of these two aspects.

3.4 *Comparison of alternative networks*

The network results of Luk (1998), in addition to being used for determination of network components and architectures, can also be used as the basis of a comparison between alternative types of network. The selection of a best network configuration based on NMSE scores computed using test data for eight networks with the lowest validation error are shown in Table 10.5. Each row of Table 10.5 represents a network with a specific lag. For example, the first row shows the results for a lag-1 MLFN with 24 hidden nodes, while the second row shows the results for a lag-2 MLFN with 8 hidden nodes.

In general, all three types of NN have comparable performance, which suggests that the functions being modelled are quite similar. NMSE test scores for all networks were in the range of 0.63 to 0.67. Such small differences are due to networks being developed on an optimal architecture defined through the interaction between the lag and the number of hidden nodes. For example,

Table 10.5. Comparison of alternative networks.

Network	Training (NMSE)	Validation (NMSE)	Testing (NMSE)	Stopping epoch	Training error at 1000 epoch (NMSE)
MLFN Lag 1 (16-24-16)	0.50	0.68	0.64	200	0.49
MLFN Lag 2 (32-8-16)	0.51	0.69	0.66	100	0.47
MLFN Lag 3 (48-4-16)	0.48	0.69	0.67	700	0.47
MLFN Lag 4 (64-2-16)	0.52	0.71	0.65	200	0.49
PRNN (16-4-16)	0.49	0.67	0.64	300	0.48
TDNN Lag 2 (32-16-16)	0.50	0.67	0.63	100	0.41
TDNN Lag 3 (48-32-16)	0.50	0.69	0.64	100	0.41
TDNN Lag 4 (64-32-16)	0.51	0.69	0.65	100	0.40

Notation: The network configuration is denoted by three figures (x-y-z), where x = no. of input nodes, y = no. of hidden nodes and z = no. of output nodes.

the lag-1 MLFN requires more hidden nodes to achieve an optimal solution while the more complicated lag-4 MLFN requires two hidden nodes due to the large number of parameters associated with high order of lag. This result is consistent with the concept of the existence of an optimal configuration for a network that was discussed earlier.

Figures 10.3 and 10.4 compare three one-step ahead forecasts of rainfall depth at a single gauge. The hyetograph in each figure shows the actual rainfall recorded at that gauge site and the forecast rainfall, and similar plots for other gauges and storm events presented in Luk (1998) and Luk *et al.* (2000).

Analysing the forecast errors for one storm with each of the three solutions, we conclude that: (i) the forecast error increases as the rate of change of rainfall intensity increases; (ii) the networks made better predictions after the peak of the storm event; and (iii) all the networks tended to under-predict the rainfall when the rate of change in the rainfall intensity was positive, and to over-predict the rainfall when the rate of change in the rainfall intensity was negative.

Summarising all the comparison tests, it was found that:

- the three alternative types of NN have comparable performance;
- MLFN with lower orders of lag have a marginally better performance than networks with higher orders of lag;
- MLFN with higher lags tended to over-learn the training data, resulting in smaller training errors, but larger validation errors;
- MLFN with lower lags require more hidden nodes, and vice versa, suggesting an optimal set of architectural components;
- PRNN showed comparable performance with lag-1 MLFN and outperformed MLFN with higher order lag;

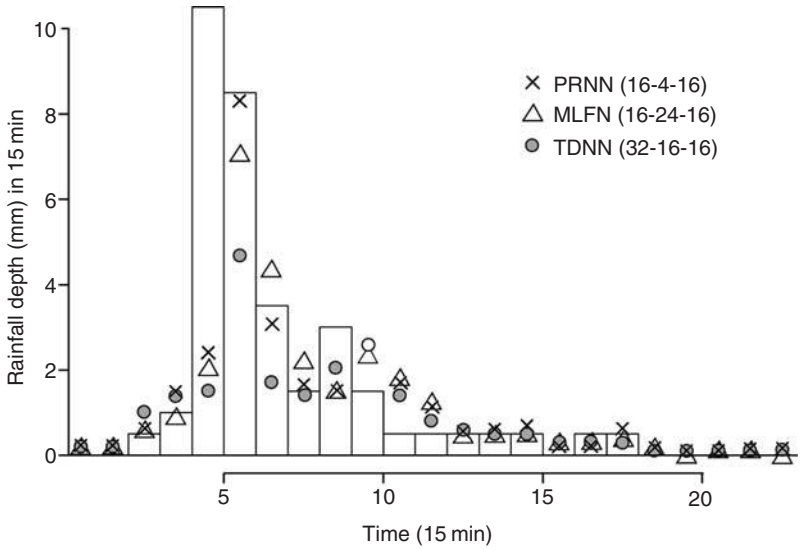


Fig. 10.3. Forecasting rainfall at gauge no. 7253 for the storm event on 2 January 1996.

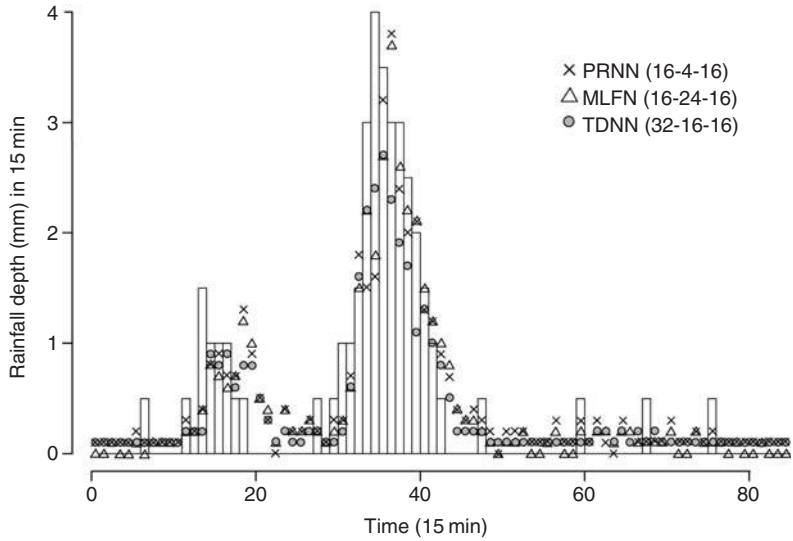


Fig. 10.4. Forecasting rainfall at gauge no. 7253 for the storm event on 6 January 1996.

- a lag-2 TDNN provided the lowest error for the test data set (0.63); and
- all forecast errors in all cases increased as the rate of change of rainfall intensity increased in either a positive or negative direction.

4 RAINFALL FORECASTING – AREAL DISTRIBUTIONS

4.1 *Building an integrated modelling solution*

The integration of GIS and NN approaches provides a powerful rainfall forecasting model that merges the merits of the two technologies together. For the purposes of developing an areal rainfall forecasting system, it was assumed that no rainfall data, other than that measured at the rainfall gauges, were available for this catchment. There are two alternative approaches that could be used to generate future rainfall forecasts; these two approaches are:

- Use of a GIS to estimate the rainfall for each cell within the catchment based on the measured rainfall at each gauge. This will generate a significant number of estimated rainfall surfaces. The NN could then be used to map these estimates to produce a direct rainfall forecast for each cell within the catchment.
- Use of an NN to forecast the rainfall at each of the rainfall gauges and then use the GIS to generate a distributed rainfall forecast for all cells within the catchment.

The first approach of using catchment cells as the location of both input and output information was not considered feasible due to the large number of input and output nodes that would be required, which for 0.25 km^2 cells in the Upper Parramatta River catchment is 439 per layer. This would result in 1756

connections for a 3-layer MLFN and training such a large network would be extremely time consuming. Moreover, a network with so many free parameters would need a large volume of detailed rainfall data that seldom exists.

4.2 *Integrated solution*

150 artificial storm events were used to build and test the integrated rainfall forecasting system. This need for synthetic data was due to an inadequate number of recorded storm events. The forecast rainfall at each cell was compared with the actual value of rainfall (from the artificial event) to ascertain the forecast error. Of these events, 100 storm events were used for training the NN, 25 storm events for validation of the training process, and 25 storm events for testing the rainfall forecasting system.

The artificial storm events were assumed to be a random process, with some degree of tracking, and were generated from a mixture of autoregressive and random techniques. The first step involved the random start of a storm centre at a point close to or within the study catchment. The location of this starting point was biased according to historical records of storms over the Upper Parramatta River Catchment. The storm then moved towards the centre of the catchment subject to deviations from the initial direction defined by the following autoregressive equation:

$$\text{direction}(t) = 0.8 * \text{direction}(0) + 0.2 * \text{direction}(t-1) + e(t) \tag{4}$$

where direction(t) is the direction of the storm centre at time t, direction(0) is the initial direction of storm movement and $e(t)$ is the random deviation of storm movement, which had a mean of 0° and a standard deviation of 15°. The intensity of the storm at its centre was an autoregressive process that had the following relationship:

$$P_{max} = 0.2 * P_{max}(t-1) + 0.8 * e(t) \tag{5}$$

where $P_{max}(t)$ is the rainfall intensity at the storm centre (mm/hr) at time t, and $e(t)$ is the random fluctuation. The movement of the storm centre was a random process with a mean speed of 12 km/hr, and a standard deviation of 2 km/hr, as illustrated in Figure 10.5. This storm moved across the catchment from the North East to South West over a period of 2.5 hours (15 min × 10 time steps).

4.3 *Test results and discussion*

In ascertaining the accuracy of the spatial rainfall forecasts, both visual and arithmetic comparisons were established. Validation of the spatial rainfall forecasting system was on the basis of

- replicating the areal rainfall patterns (visual inspection);
- tracking the movement of storm centres (visual inspection);
- predicted rainfall at individual cells; and
- predicted rainfall for subcatchments.

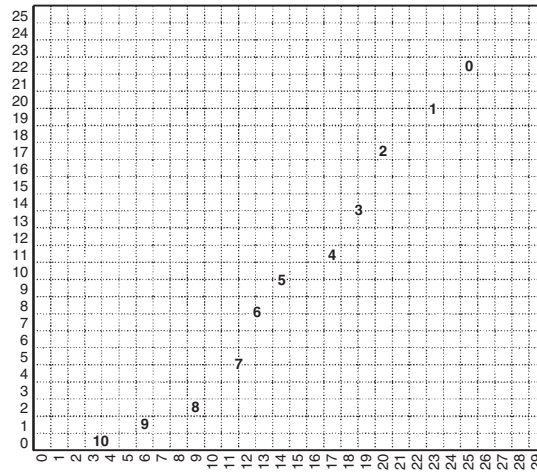


Fig. 10.5. Track of storm centres.

Table 10.6. Forecast errors for storm no. 138.

Time step	Intensity (mm/15-min)	Distance (km)	Angle (degree)
1	−4.0	0.11	−47.2
2	−4.6	0.64	12.3
3	−5.8	−0.61	28.3
4	−1.5	−0.66	16.4
5	−3.5	−0.56	−15.8
6	1.8	−0.22	25.4
7	−3.1	−1.05	−10.7
8	1.8	−1.00	−1.5
9	−2.0	−0.34	109.4
10	−3.6	−0.17	74.8

Table 10.7. Error statistics for storm no. 138.

Statistic	Intensity (mm/15-min)	Distance (km)	Angle (degree)
High	1.8	−0.22	28.3
Low	−5.8	−1.05	−15.8
Mean	−1.7	−0.68	7.0
Median	−2.3	−0.63	7.4
Standard deviation	3.1	0.31	18.9
Skewness	0.1	0.18	−0.1

The results for Storm No.138 are presented in Tables 10.6 and 10.7 and in Figures 10.6 and 10.7. Figure 10.6 shows the distribution of rainfall at one instant during this testing event. A complete history of this storm event, together with other events, is presented in Luk (1998). Figure 10.7 maps the

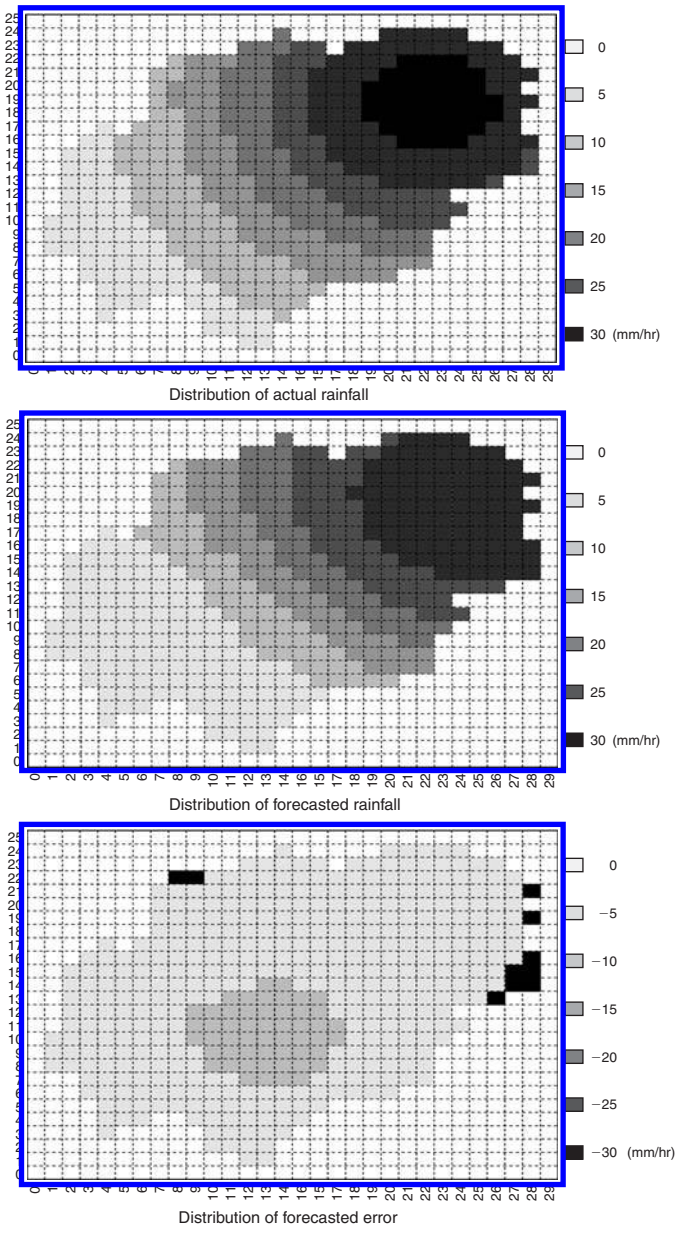


Fig. 10.6. Predicted and actual rainfall during a validation storm event.

actual storm centre track against the predicted storm centre track for the same storm event. The actual track of storm centres is represented by numbers, whereas the forecast track is shown by alphabetic characters. Both tracks had similar characteristics; the forecasting system did very well in tracking the

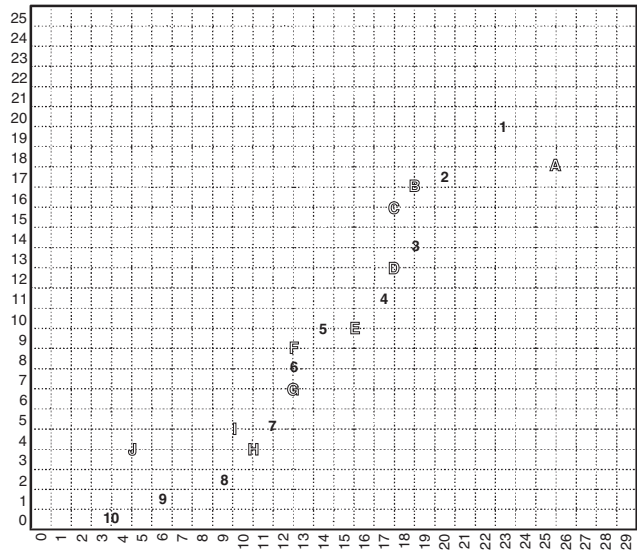


Fig. 10.7. Forecasting movement of storm centres.

Table 10.8. Error statistics for 25 test storm events.

Statistic	Intensity (mm/15-min)	Distance (km)	Angle (degree)
High	8.2	0.73	178.3
Low	−16.8	−1.68	−172.5
Mean	−2.8	−0.62	−0.9
Median	−2.9	−0.62	0.6
Standard deviation	4.3	0.49	53.0
Skewness	−0.1	0.27	−0.03

storm centres from time steps 2 to 8. There were, however, distance errors in forecasting the storm centres at time steps 7 and 8. These errors resulted in an overestimation of the spatial rainfall distribution.

Tables 10.6 and 10.7 also show large angle errors in the first and last two time steps. It is suspected that these errors are due to initiation and boundary conditions related to one or both parts of the combined modelling operation. Due to high uncertainties at such locations, the values at these time steps are excluded from the calculation of the error characteristics presented in Table 10.7. The average intensity of this storm event was 28.9 mm per 15-minute period (or approximately 76 mm/hr). The error in the forecast was small and the prediction of location was excellent. The range of error in location was only −1.05 to −0.22 km, with a mean error of −0.68 km, which means that, on average, the storm centre is predicted to be in a cell that is adjacent to the correct cell.

Similar characteristics were replicated for all 25 storm events and are summarised in Table 10.8. The integrated forecasting system is observed to have

Table 10.9. Normalised mean squared errors for all 25 test storm events.

	NMSE at each time step										
Event	1	2	3	4	5	6	7	8	9	10	Mean
126	0.444	0.245	0.227	0.429	0.015	0.028	0.072	0.162	—	—	0.203
127	0.063	0.875	0.506	0.482	0.042	0.083	0.101	0.027	0.055	—	0.248
128	0.246	0.227	0.156	0.232	0.327	0.436	0.312	0.046	0.097	0.193	0.227
129	0.362	0.046	0.177	0.161	0.341	0.156	0.254	0.617	0.587	0.163	0.286
130	0.328	0.336	0.173	0.056	0.150	0.177	0.554	0.116	0.196	0.193	0.237
131	0.160	0.512	0.305	0.010	0.015	0.031	0.185	0.066	0.098	—	0.154
132	0.469	0.264	0.321	0.078	0.070	0.239	0.454	0.348	0.215	0.143	0.295
133	0.703	0.123	0.267	0.199	0.206	0.074	0.161	0.299	0.124	—	0.240
134	0.175	0.328	0.328	0.074	0.175	0.168	0.040	0.055	0.194	—	0.171
135	0.145	0.092	0.266	1.643	0.167	0.307	0.241	0.132	0.176	—	0.352
136	0.884	0.350	0.120	0.503	0.202	0.236	0.140	0.144	0.137	0.140	0.286
137	0.070	0.443	0.213	0.224	0.615	0.375	0.063	0.221	—	—	0.278
138	0.130	0.086	0.286	0.139	0.127	0.081	0.026	0.176	0.199	0.396	0.165
139	0.150	0.245	0.301	0.427	0.062	0.054	0.082	0.080	—	—	0.175
140	1.561	0.165	0.388	0.244	0.419	0.210	0.237	0.410	0.101	0.059	0.344
141	4.422	0.026	0.826	0.147	0.217	0.199	0.278	0.053	0.048	0.113	0.587
142	0.039	0.178	0.056	0.134	0.083	0.167	0.286	0.024	0.299	1.755	0.302
143	0.416	0.028	0.383	0.070	0.809	0.441	0.224	0.117	0.165	0.147	0.280
144	0.756	0.181	0.305	0.222	0.191	0.204	0.347	0.207	0.261	—	0.297
145	0.703	0.112	0.132	0.651	0.103	2.333	0.211	0.226	0.115	0.094	0.430
146	0.434	0.196	0.13	0.128	0.257	0.056	0.045	—	—	—	0.178
147	0.093	0.655	0.104	0.402	0.355	0.069	0.071	0.009	0.059	—	0.202
148	0.149	0.107	0.073	0.093	0.207	0.073	0.150	0.170	0.185	—	0.134
149	0.063	0.186	0.09	0.102	0.476	0.120	0.285	0.041	0.051	—	0.157
150	1.075	0.432	0.391	0.097	0.566	0.296	0.251	0.080	0.073	0.156	0.342
Overall Mean NMSE											0.263

produced reliable predictions for rainfall intensities and location of storm centres, but this was not the case for prediction of storm movement.

The normalised mean squared error for each time step during all 25 test storm events is 0.263, which is considered to be a reasonable result for a reliable model (Table 10.9). There were, however, several abnormal figures, but all of these occurred at the first time step and the integrated solution provided accurate forecasts for the remainder of the storm movements and associated rainfall intensities. These errors are ascribed to problems in recognising the initial position of the storm centres.

5 CONCLUSIONS

Development of an integrated rainfall forecasting system using GIS and NN technologies has been the focus of this chapter. This development process involved investigations into models of the spatial distribution of rainfall, and the appropriate form of NN for rainfall forecasting as well as an assessment of the forecast accuracy of the proposed system.

During NN development, three alternative types were identified, developed and compared: a MLFN; a PRNN; and a TDNN. It was found that all three types were able to make reasonable 15-minute forecasts of rainfall for multiple locations within a catchment. From test results, the following points are noted:

- For each network, there was an optimal configuration, which was determined from a combination of the number of hidden nodes and the lag of the network;
- The three networks had comparable performance when developed and trained to reach their optimal forecasting power; and
- Networks with lower lags outperformed ones with higher lags due to the 15-min rainfall time series possessing no long-term relationships.

The integration of GIS and NN technologies enabled reasonable 15-minute forecasts of the spatial distribution of rainfall over the Upper Parramatta River Catchment. The system accurately preserved spatial rainfall patterns and produced good forecasts with strong agreement to actual rainfall values. In conclusion, the rainfall forecasting system developed for the Upper Parramatta River Catchment had the following characteristics:

- A spatial and temporal distributed architecture;
- The system has been developed for real-time operation. It receives present rainfall values at multiple gauge positions and produces rainfall forecasts for each cell in the catchment; and
- With collection of new rainfall data, the NN component can be re-trained to provide improved performance.

REFERENCES

Ball, J.E. & Luk, K.C. 1996. Determination of the Rainfall Distribution over a Catchment using Hydroinformatics Tools. *Proc. 2nd International Conference on Hydroinformatics*, Zurich, Switzerland: 369–376.

Ball, J.E. & Luk, K.C. 1998. Modelling the spatial variability of rainfall over a catchment. *ASCE, Journal of Hydrologic Engineering* 3(2):122–130.

Elman, J.L. 1990. Finding structure in time. *Cognitive Science* 14:179–211.

Fontaine, T.A. 1991. Predicting Measurement Error of Areal Mean Precipitation During Extreme Events. *Water Resources Bulletin* 27(3): 509–520.

Handmer, J.W., Smith, D.I. & Lustig, T.L. 1988. The Sydney Floods of 1986: Warnings, Damages, Policy and Future. *Proc. Hydrology and Water Resources Symposium, 1988, Australian National University, Canberra, Australia*, IEAust Nat Conf Pub 88/1: 206–210.

Luk, K.C. & Ball, J.E. 1996. *Application of GIS for Modelling of the Spatial Distribution of Rainfall.* Research Report 191, Water Research Laboratory, School of Civil and Environmental Engineering, University of New South Wales.

Luk, K.C. 1998. *An application of hydroinformatic tools for rainfall forecasting.* PhD Dissertation, School of Civil and Environmental Engineering, The University of New South Wales, Sydney, Australia.

Luk, K.C., Ball, J.E. & Sharma, A. 2000. A study of optimal lag and spatial inputs to artificial neural network for rainfall forecasting. *Journal of Hydrology* v.227: 56–65.

Luk, K.C., Ball, J.E. & Sharma, A. 2001. An application of artificial neural networks for rainfall forecasting. *Mathematical and Computer Modelling* 33: 683–693.

Rumelhart, McClelland, and the PDP Group 1986. *Parallel Distributed Processing*, Vols 1 & 2. Cambridge, MA: MIT Press.

Urbonas, B.R., Guo, J.C.Y. & Janesekok, M.P. 1992. Hyetograph density effects on urban runoff modelling, *Proc. Int. Conf. On Comp. Applications in Water Resources, Tamkang University, Tamsui*, Taiwan: pp. 32–37.

Weigend, A.S., Huberman, B.A. & Rumelhart, D.E. 1992. Predicting Sunspots and Exchange Rates with Connectionist Networks. In M. Casdagli, & S. Eubank (Eds.), *Non-linear Modeling and Forecasting*: (pp. 395–431). Addison Wesley.

11

Water Quality and Ecological Management in Freshwaters

PAULINE E. KNEALE
School of Geography, University of Leeds, UK

ABSTRACT: The sustainable management of watercourses involves an appreciation of the interactions between multiple ecological, hydrological, hydrochemical and anthropogenic factors. In these matters getting the water flow right is vital, adding the complex ecology adds to the challenge. Since many hydroecological interactions are difficult to model in a process manner and ecological data are often limited, neural networks are a valuable forecasting tool. This chapter presents a literature review of the use of artificial neural networks in water quality applications and considers the value of the various approaches adopted.

1 INTRODUCTION

The sustainable management of river water resources and modelling of freshwater ecological systems is challenged by the interaction of many heterogeneous processes. The complex dynamics are not easily coded into process models, and, when available, the required data are often sparse with variable sequencing or qualities. In addition to receiving natural, urban and industrial effluent inflows, river waters may be polluted by diffuse runoff from agricultural, industrial and household processes, and from buildings and road surfaces which impacts on the in-stream ecology (Beasley & Kneale, 2002; Ellis & Hvitved-Jacobsen, 1996; Lee & Bang, 2000; Moog & Chovanec, 2000). This water may be abstracted later for drinking or industrial use. Water quality management agendas involve identifying and preventing pollution, and identifying and restoring water courses, to enhance water quality and make water available for use and re-use downstream. Sensitive water bodies and ecosystems may require protection, mitigation or remediation.

Effective water quality management involves respecting the ecological integrity and diversity of the system. In planning water and related land

resources, the US Water Resources Council (1973) recommended consideration of national economic development and environmental quality. Therefore, modelling the quality of flow for management purposes may also require the coupling of hydrological, biogeochemical, ecological and socio-economic models at river basin and sub-basin scales. Colasanti (1991) discussed the similarities between the structure and behaviour of artificial neural networks (NN) and natural ecological systems, arguing that the parallels implied that NN techniques should be appropriate tools for ecological modelling. In this chapter the current role of NN in contributing to the management of river water quality is reviewed.

2 DATA ISSUES

Given the complexity of the biogeochemical processes involved, the changes that occur from point to point and the natural variability of environmental systems, the NN approach to identifying and managing pollution is extremely valuable. But as previous chapters have shown, NN solutions require appropriate data and evaluation. Modelling biological and chemical processes in water and associated ecological behaviour is a technically complex issue. Compared with river flow forecasting, where lengthy records of stage or discharge and rainfall may be available at multiple stations at 15 minute or hourly intervals, good water quality data are often very limited; it may be weekly or monthly data, for a reduced network of sites, and a limited number of parameters. Ecological observations are even less frequent, often comprising snapshot surveys made on one or two days each year.

Point data may be available for the chemistry of the water column and bed sediment, but the equations which govern the spatial and temporal chemical interactions between the column and the sediment are by no means definitive. Simple models using 1D equations, driven by average travel times, may be successful in forecasting the movement of pollution plumes downstream but miss the detail required to look at biological and chemical interactions. However, when attempting to account for a large number of factors, distributed mathematical modelling becomes infinitely more complex, and requires heavy computational investment.

The key questions for evaluating a NN application include asking:

- Does the approach provide an answer that is appropriate for a particular experimental investigation?
- Does it improve on current forecasting practice?
- Is the forecast accurate enough and achieved at an acceptable cost?

Bowden *et al*. (2002) using data for the River Murray catchment analyse issues that can arise as the result of the way in which subsets of data are selected for the training, testing, and validation of models and shows that the less than optimal selection will impair model performance. Cigizoglu (2003) shows that using longer training stage datasets improves the NN forecast and argues that generating synthetic series can be advantageous where the full data set is limited.

3 MODELLING STYLES

Because NN were designed to mimic certain biological functions of the brain, we might have expected widespread interest in applying this technology to biological applications of water quality. But, as Table 11.1 indicates, NN applications have really only mushroomed since 1996. The range of applications is very wide, from explorations of individual parameters (Cancilla & Fang, 1996; Moatar *et al.*, 1999), through fisheries management (Aurelle *et al.*, 1999; Olden & Jackson, 2002), to describing more complex patterns and interactions for management purposes (Schleiter *et al.*, 1999; Walley & Fontama, 1998).

The objective evaluation of NN outputs comes in one of two forms. Either NN forecasts are compared to field or laboratory time series as in the case of Maier *et al.* (1998) and Recknagel *et al.* (1997), or outputs are compared to results developed using alternative forecasting techniques. Table 11.1 indicates that in 40% of the applications, linear regression is the chosen alternative; comparisons with process models are infrequent.

The following statement made by Lek and Guegan (1999) when reviewing NN progress in a wide range of applications is true for these cases: 'Most of these examples showed that NN performed better than more classical modelling methods'. It can be argued, somewhat cynically, that the authors would not have published if their results had been worse. It should also be noted that multiple regression is a less sophisticated form of black-box input-output modelling, and that in a very complex and non-linear natural world, NN models should do better by definition. For optimum evaluation NN forecasts should be set against the most sophisticated modelling approach normally used for each application.

It is notable that where results are reported with reference to times series, the correlation coefficients are frequently above 0.9. Even where the ecological and hydrological processes are complex, the results are impressive as in Karul *et al.*'s (2000) study of modelling eutrophication in Turkish lakes and dams. The authors state that 'despite the very complex and peculiar nature of Keban Dam, a relatively good correlation (correlation coefficient between 0.60 and 0.75) was observed between the measured and calculated values. For Mogan and Eymir, which are much smaller and more homogenous lakes compared to Keban Dam Reservoir, correlation values as high as 0.95 were achieved between the measured and calculated values' (p. 145).

4 LABORATORY ANALYSIS

The increasing need to detect low levels of pollutants in water and sediment and to look at the reactions of combinations of herbicides, pesticides and chemical pollutants on organisms demands discriminating laboratory analytical techniques that are sensitive and fast. Water quality managers may first encounter NN in the laboratory where such tools are being incorporated into 'smart' and/or integrated sensing equipment to automate pollutant detection and to discriminate chemicals or identify organisms (Morris *et al.*, 2001) at very high speeds. Yatsenko (1996) shows how NN can assist in disentangling information from

Table 11.1. A selection of applications of NN in water quality modelling.

Topic/ authors	NN learning system (software if known)	Application	Model comparison (where made)
Water chemistry			
Cancilla and Fang (1996)	Backpropagation (Brainmaker Professional)	Water chemistry, three sites on the Niagara River, USA	PCA and UPM
Clair and Ehrman (1998)	Backpropagation (Neuroshell 2)	Changing climate impacts on discharge, dissolved organic carbon, and nitrogen from 14 eastern Canadian rivers	
Clair *et al.* (1999)	Backpropagation (Neuroshell 2)	DOC in runoff for climate modelling	Linear regression
Gong and Denœux (1996)	Epochwise backsweep	Transfer of solid waste in sewer pipes	
Lek *et al.* (1996)	Backpropagation	Phosphorous and orthophosphate runoff from 927 USA tributaries	
Lek *et al.* (1999)	Backpropagation	Nitrogen in stream water using land use variables. Data from 927 US watersheds	
Maier and Dandy (1996)	Backproagation	Salinity of River Murray	
Maier and Dandy (1998)	(NeuralWorks Professional II/Plus)		
Maier and Dandy (2001)			
Manescu *et al.* (1998)		DO and flow	
Moatar *et al.* (1999)	Backpropagation	Daily pH	1st and 2nd order polynomial regression
Poff *et al.* (1996)	Backpropagation	Temperature and flow changes in response to climate change scenarios	
Starrett *et al.* (1998)	Backpropagation	Pesticide leaching from golf courses	
Starrett *et al.* (2001)			
Yang *et al.* (1997)	Backpropagation	Daily soil pesticide concentrations	
Zhang and Stanley (1997)	Backpropagation	Water colour for treatment and drinking water management	
Cyanobacteria			
Maier and Dandy (1997)	Backpropagation (NeuralWorks Professional II/Plus)	Forecasting algal blooms in River Murray using *Anabaena* and river water chemistry, colour, and temperature	
Maier *et al.* (1998)	Backpropagation (NeuralWorks Professional II/Plus)	*Anabaena* in River Murray	

Maier *et al*. (2001)	Backpropagation	*Anabaena* in River Murray, four week forecasts	Associative memory networks (AMNN)
Recknagel (1997)	Backpropagation	Blue green algae species forecast using water chemistry information for lakes in Japan	
Recknagel *et al.* (1997)	Backpropagation (EXPLORER from Neural Ware Inc.)	Blue-green algae species abundance is forecast from water chemistry inputs for lakes in Finland and Japan, and River Murray, Australia	
Walter *et al.* (2001)	Backpropagation	Chlorophyll-α as a measure of eutrophication, Burrinjuck Reservoir, Australia	Deterministic model SALMO
Wei *et al.* (2001)	Backpropagation (MATLAB Neural Network Toolbox)	Algal densities – four genera	
Whitehead *et al.* (1997)	Backpropagation	Algal models for River Thames reaches	Time series, Dynamic process growth model
Wilson and Recknagel (2001)	Backpropagation	1 and 30 day forecasts of algal abundance, 6 lakes	
Fish			
Aurelle *et al.* (1999)	Backpropagation	Distinguishing hatchery and natural brown trout stocks	
Baran *et al.* (1996)	Backpropagation	Brown trout population density	Multiple regression
Brosse *et al.* (1999)	Backpropagation	Lake fish abundance, multiple species	Multiple regression
Brosse *et al.* (2001)	Kohonen self-organizing map	Fish assemblages, 15 species, Lake Pareloup, France	Principal component analysis
Chen and Ware (1999)	Backpropagation	Pacific herring stocks	Multiple regression, process based climate-stock recruitment model
Gozlan *et al.* (1999)	Backpropagation	River fish diversity and abundance	Multiple regression
Huse and Ottersen (2003)	Backpropagation	Recruitment and biomass development of Northeast Arctic cod.	
Ibarra *et al.* (2003)	Backpropagation	Fish guilds, Garonne	
Laë *et al.* (1999)	Backpropagation	Fish yields from 59 lakes in Africa	Descriptive statistics, Stepwise multiple regression
Lek and Baran (1997)	Backpropagation	Brown trout population density	
Mastrorillo *et al.* (1997a)	Backpropagation	Minnow abundance	Stepwise multiple regression

Table 11.1. *(Continued)*

Topic/ authors	NN learning system (software if known)	Application	Model comparison (where made)
Mastrorillo *et al.* (1997b)	Backpropagation	Presence/absence 3 small fish: minnow, gudgeon, stone loach	Discriminant factor analysis
Reyjol *et al*. (2001)	Backpropagation	Habitat modelling, brown trout, minnow, stone loach.	
Zhou (2003)	Backpropagation	Abundance of Pacific salmon Oncorhynchus spp.	Moving average
Ecological Interactions and Management			
Campolo *et al.* (1999)	Backpropagation (Stuttgart Neural Network Simulator)	Flow forecasting to manage water quality at low flows	
Chon *et al.* (1996)	Kohonen Network	Benthic macroinvertebrates, Suyong river, Korea	
Chon *et al.* (2000)	Adaptive Resonance Theory Kohonen Network	Monthly benthic macroinvertebrates in two streams	
Karul *et al.* (1999)	Backpropagation	Lake eutrophication	Multiple regression
Karul *et al.* (2000)	Backpropagation	Eutrophication in lakes and dams in Turkey	
Lek and Guegan (1999)	Backpropagation Kohonen Self-Organizing Map	A review paper discussing the use of NN for ecological modelling	
Obach *et al.* (2001)	Kohonen Self-Organizing Map	Ecosystem dynamics, 30 year record of macroinvertebrate and habitat data	
Olden (2003)	Backpropagation	Lake fish communities	
Paruelo and Tomasel (1997)	Backpropagation	Seasonal vegetation growth	Multiple regression
Schultz and Wieland (1997)	Backpropagation	Managing crop development and soil moisture using meteorological and crop data	
Schleiter *et al.* (1999)	Backpropagation with senso-nets	Multiple water chemistry parameters, habitat variables and benthic macro-invertebrates	
Walley and Fontama (1998)	Backpropagation initial experiments included trials with 7 types of NN before backpropagation was selected	Average score per taxon (ASPT) Number of families in unpolluted rivers	
Wen and Lee (1998)	Backpropagation	Catchment scale planning and management	

sensors as part of water analysis procedures. He uses a neural chip to separate and classify pollutant characteristics. Charef *et al.* (2000) use the NN approach to estimate concentrations of COD (Chemical Oxygen Demand) in urban waste waters. Detailed discussion on the place of NN technologies in laboratories is beyond the scope of this review, but these applied uses of NN solutions are typically tied to pollution monitoring at water and sewage treatment works (Adgar *et al.*, 2000; Baxter *et al.*, 2001; Brion *et al.*, 2001; Choi & Park, 2001; Delgrange-Vincent *et al.*, 2000; Joo *et al.*, 2000; Milot *et al.*, 2002; Pigram & Macdonald, 2001; Serodes *et al.*, 2001; Shetty *et al.*, 2003).

5 FLOW FORECASTING IN RIVERS

At the heart of a traditional distributed forecasting model for water quality or pollution monitoring there are routing models that forecast the in-channel movement of water. Corne *et al.* (1998) have shown that the travel times of flood waves can be forecast where the river data series is long enough to train the initial model. But pollution wave travel times vary from event to event and do not move at the same pace as the flood wave. At times of high flow a pollution incident may be mitigated by dilution, but the more rapid travel times may also be critical in planning downstream pollution prevention measures. The potential of NN solutions to act independently of a flow model by looking at event data individually is a potential strength over traditional models that forecast the chemistry of the runoff and then route the flows downstream as a separate operation. NN are particularly valuable where flow is not well related to precipitation because of abstraction for hydro power plants, reservoirs or irrigation (Golob *et al.*, 1998; Stokelj *et al.*, 2002). The application of NN in other flow routing applications can be found elsewhere in this book (Chapters 5, 6 and 10).

What is not clear is whether in the long-term eco-hydro-chemical models should be linked to flow forecasting models with good flood-wave travel-time forecasting performance. Or whether the NN treatment of flow and chemistry data independently but simultaneously will make robust models that solve this non-linear problem. TDNN algorithms may prove more accurate since they can search for appropriate lags in sequential data series (see Chapter 5). Developing research in the latter area is potentially more appealing.

6 WATER QUALITY MODELLING

Classical water quality modelling started in the 1920s with the Streeter-Phelps models for dissolved oxygen (DO) forecasts and some of the earliest NN investigations modelled the same variable. Partly this is driven by the importance of DO in any aquatic system and because this is a variable with reasonably detailed field records in many cases. Manescu *et al.* (1998) use a NN solution in conjunction with a Geographical Information System (GIS) to forecast DO and stream flow. The presented results are good but there is no independent forward

testing for accuracy and stability. They pick up the point that non-uniformity of sampling over 3 years forced the use of monthly averaged data. The model works well but any data averaged over a month hides individual incidents of high or low levels. This is fine for applications where monthly averages are the required output, but the NN can forecast at shorter time intervals, given appropriate data.

Daily data are available for the Middle Loire which permits Moatar *et al.* (1999) to forecast pH. The inputs are river flow and radiation, variables chosen as important in controlling the eutrophication processes during summer low-flow regimes. Testing was undertaken using 4 years of data and validated with 1 year of independent data. The reported pH forecast accuracy is 86%.

6.1 *Chemical pollution*

Nutrient loading is an issue in eutrophication where enriched runoff leads to increased algal growth and decreased water quality. Phosphorous (P) has been shown to be significantly related to enrichment characteristics, and so the possibility of forecasting P levels on a national basis was explored by Omernik (1997). Omernik took data for 927 tributary catchments from across the USA which did not have point source pollution inputs. Using multiple regression she found it necessary to divide the USA into three regions and produce different models for each region. Lek *et al.* (1996) have explored the forecasting potential of the same data set using backpropagation neural network (BPNN) modelling and compared the performance of the different approaches. Lek *et al.* used six independent variables to develop models for four dependent variables which characterise P in runoff (Table 11.2). In each model run the six independent variables were linked to one of the dependent variables (concentration or export of P). Experimental runs with up to 20 hidden nodes showed that a good predictive model could be created for each measure of P using only 5 hidden nodes and training for 500 iterations. The independent variables reflect land use and stocking density, but have no direct information about regional

Table 11.2. Variables used by Lek *et al.* (1996) to forecast four measures of P in runoff from USA catchments.

Independent variables		Dependent variables	
FOR	% of catchment under forest	CTP	Concentration of total phosphorous ($mg \cdot l^{-1}$)
AGR	% of catchment under agriculture	COP	Concentration of ortho-phosphorous ($mg \cdot l^{-1}$)
OTH	% other land uses	ETP	Export of total phosphorous ($mg \cdot l^{-1}$)
PRE	Average annual precipitation (cm)	EOP	Export of ortho-phosphorous ($mg \cdot l^{-1}$)
FLO	Discharge ($m^3 \cdot s^{-1}$)		
ANI	Animal density (animal/km^{-2})		

variability. The forecasting efficiency in terms of a correlation coefficient was in each case better than 0.7. NN performance also either equalled or improved upon the regression approach. Moreover, there is no need for multiple regional models, as a single network can cope with variation across the US. The advantage of the NN approach over regression modelling is that the non-linearity inherent in the systems is dealt with better. The performance of the BPNN in this case shows that a single model can outperform the multiple regression approach, and is more flexible.

Moving from a regional to a local scale, Cancilla and Fang (1996) sought to compare techniques which will discriminate water chemical data from three sites on the Niagara River. In this application the goal is classification, i.e. can the NN or other methods discriminate within and between samples to identify their source? The data comprised 24 hour composite water samples, taken weekly and analysed for 32 target compounds, including chlorinated pesticides and polynuclear aromatic hydrocarbons (PAHs). The NN was trained to see how well the program could determine the source of each sample. The authors compared the performance of the NN with results from a Universal Process Modelling (UPM) analysis (O'Sullivan, 1991; Teranet IA Inc, 1992) and Principal Components Analysis (PCA). All three approaches discriminated the sources of the water samples.

In their experiments models were built using 25% of the data and then validated against the unseen 75%. Table 11.3 shows the relative performance of the NN and UPM approaches, which are both successful, with results above the 90% level. The NN performs slightly better. Additionally the authors investigate the size of the minimum data set required to build an adequate model. Figure 11.1 shows the results of training with sample sizes ranging from 2 to 112. A predictive success of 65% was achieved from models developed on 2% of the samples, but when given 25% of the sampled data (35–40 records), prediction success rose above the 90% level. This is an example of NN being used to classify data, in this case to identify patterns and changes in patterns. Given appropriately accurate data sets for normal hydrological conditions, this technique has tremendous potential in natural and drinking water quality analyses to spot changes in trends and locate contamination sources.

The water supply for Adelaide, Australia, relies on piped transfers from adjacent basins. The piped supply contributes 10–80% of demand and may have a high salinity. Maier and Dandy (1996) sought to forecast salinity levels and recognised the potential appropriateness of developing NN solutions 'because longer-term forecasts are required, non-linear relationships are

Table 11.3. The percentage of times samples were correctly identified from chemical data for 3 sites on the Niagara River (from Cancilla & Fang, 1996).

Method	NN (%)	UPM (%)
Training data 25% (37 seen samples)	100	93.7
Validation data 75% (112 unseen samples)	94.4	91.7

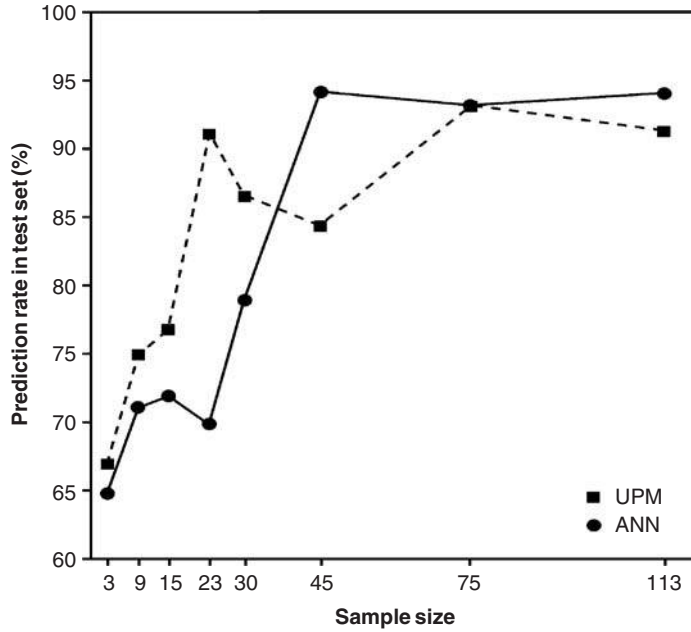


Fig. 11.1. The effect of the training data set size on the prediction rate (from Cancilla & Fang, 1996).

suspected and it is difficult to prescribe the exact mathematical relationship between the variables' (p. 1016).

Their initial model had 141 inputs: flows, lagged flows and salinity data for a series of sites, and the output was a 14-day ahead forecast of flow and salinity at Adelaide. The final model had a reduced number of inputs (51) which reduced training time and raised output efficiency. The validation 14-day forecasts had errors ranging from 5.3–7.0%. This is acceptable for planning purposes, but could probably be improved given further data, although Maier and Dandy (1998) show that it can also be improved by looking in detail at the parameters of the NN structure itself. Through careful adjustment of the learning rate, searching for appropriate local minima and paying attention to the length of the training period, the model can be further optimised. The authors make the point that the model will learn common patterns quickly and well but is slow to learn infrequent patterns and may model these very badly. Similarly, in flood modelling, Kneale and See (2000) deal with this issue by training networks for extreme events on the high stage hydrographs only, and ignoring the low flow discharge patterns. In looking at chemical and biological responses it may be a good managerial approach to develop NN solutions for typical and regular conditions and specialist models for rare events.

Where water is taken from rivers for drinking water, the intake quality determines the level of purification required. On the North Saskatchewan River at Edmonton, Alberta, the key variable is colour (Zhang & Stanley,

1997). Catchment land cover is forest and agriculture, the water quality is good but subject to high colour in spring and summer. In managing the treatment works, a one-day ahead forecast is appropriate, giving time for chemical additives to be adjusted for the next intake. The input parameters are based on river flow, change in flow rate, current and lagged colour data, precipitation and temperature. Data are lagged to take account of river water travel times and seasonality indices. The authors acknowledge that the models described in the paper may not be the optimum solutions for this particular site but the results show that NN can provide practical tools for forecasting water quality in real-time so that operational decisions can be taken. Potentially the results from this type of modelling can be linked to a computerised water treatment control process, with additives and settlement times being controlled through the model.

Amongst a host of other applications Starrett *et al*. (1998, 2001) have shown that their BPNN can be used to forecast the leaching of pesticides, in this case for the short turf on golf courses. Their KTURF model outperformed the comparative regression models to forecast the complex interaction between the pesticide, its solubility, leaching rates, and soil parameters. This model has potential practical use in many irrigation applications where the leaching of chemicals or pesticides are of concern.

6.2 *Climate change*

There have, to date been limited numbers of 'what if' climate and water chemistry modelling studies that used NN technologies, but Clair and Ehrman (1998) is a valuable exception. These authors take up the challenge of assessing climate change influences on forecasts of monthly discharge (Q, $m^3 \cdot s^{-1}$), dissolved organic carbon (DOC, $kg \cdot ha^{-1} \cdot mo^{-1}$), and dissolved organic nitrogen (DON, $kg \cdot ha^{-1} \cdot mo^{-1}$) for 14 basins in Atlantic Canada where wetlands influence the hydrological response. Modelling comprised a three-layer BPNN process and the inputs were: month number, basin area, basin slope and 6 climate inputs which are entered for the current and preceding month – maximum, minimum and mean monthly temperature, total rain, total snow and total precipitation. This modelling is at the seasonal scale, taking monthly values and summing them to generate seasonal figures. The model is trained and validated on historic data but can also be used with GCM (global circulation model) forecasts to look at the impact of future climates. Such forecasts are clearly limited by the accuracy of the GCM estimates of climate parameters and results cannot be compared with outcomes. However, logical results are found. A warmer winter scenario reduces storage in the snow pack and so produces more runoff, whereas summer warming increases evapotranspiration and the forecast flows are lower. The forecasts for Q and DOC which follow similar patterns are good. The DON forecasts are less good, because nitrogen runoff is a more complex process than climate and basin topography can be pattern matched against the results. These results have practical management implications where water is extracted for irrigation and power generation and in flood control, but as the authors point out, their reliability for future climate related forecasting is a function of the GCM data input quality.

An indirect investigation of climate change consequences for ecology is discussed by Poff *et al*. (1996). They use BPNN to forecast flows in the Little Pauxtent and Independence Rivers. The models are trained on historic flow data and then used to generate new forecasts given various scenarios for climate change. The discussion of the ecological implications of various scenarios is based on flow data only, and there is no ecological element in the modelling. However, the NN forecasts provide an interesting basis for future speculation.

Frakes and Yu (1999) compare NN with a GIS based hydrological model to forecast direct surface runoff, on a daily basis, from four sub-basins of the Susquehanna River. The results were similar, with the NN doing slightly better in the larger sub-basins and slightly less well in the smaller basins.

While Pearson *et al.* (2002) are concerned with forecasting plant species evolution their modelling approach couples the power of a NN approach with a process based climate–hydrological model, which identifies bioclimatic envelopes and predicts species distribution changes in response to various climate change scenarios. This coupling approach giving additional power to the modeller.

6.3 *Biological interactions*

Water quality may be characterised directly by its chemistry; but the abundance or absence of biota is also a significant indicator of the quality and health of a channel reach. Algae and indicator species such as *Cryptosporidia* present water managers with immediate problems (Maier & Dandy, 1997; Recknagel *et al*., 1997; Maier *et al.*, 1998). The abundance and diversity of fish species in a river reach are also an indication of water quality (Baran *et al.*, 1996; Lek & Baran, 1997). There are a number of investigations that have studied fish population diversity and the effectiveness of NN simulations to forecast biological population diversity (Mastrorillo *et al.*, 1997a, b; Walley & Fontama, 1998) or to establish general relationships between environmental variables and the ecological status of a river or lake.

Table 11.1 indicates that fish modellers are wedded to BPNN and generally compare the results to regression models. Baran *et al.* (1996) compare forecasts of population density and biomass in *Salmo trutta* L. (brown trout) for 220 channel reaches on 11 streams in the Pyrenees. The NN model was trained on data from 165 reaches and validated on 55 reaches. In both analyses, observed and estimated biomass and density are related, but the NN correlation coefficients are higher, and therefore the models are more accurate (Fig. 11.2). Taking these data further, NN are used to predict the density and biomass of trout from environmental variables (Lek & Baran, 1997). The independent variables are mean Froude number, mean depth, mean bottom velocity, mean surface velocity, % shelter, % deep water, % total cover and stream altitude. Again the results are very useful in an area where modelling ecological relationships from deterministic flow equations is fraught with difficulty.

Gozlan *et al.* (1999) use NN to look at the structure and diversity of young (0+ years) fish in the River Garonne. For thirty-eight sites on the river

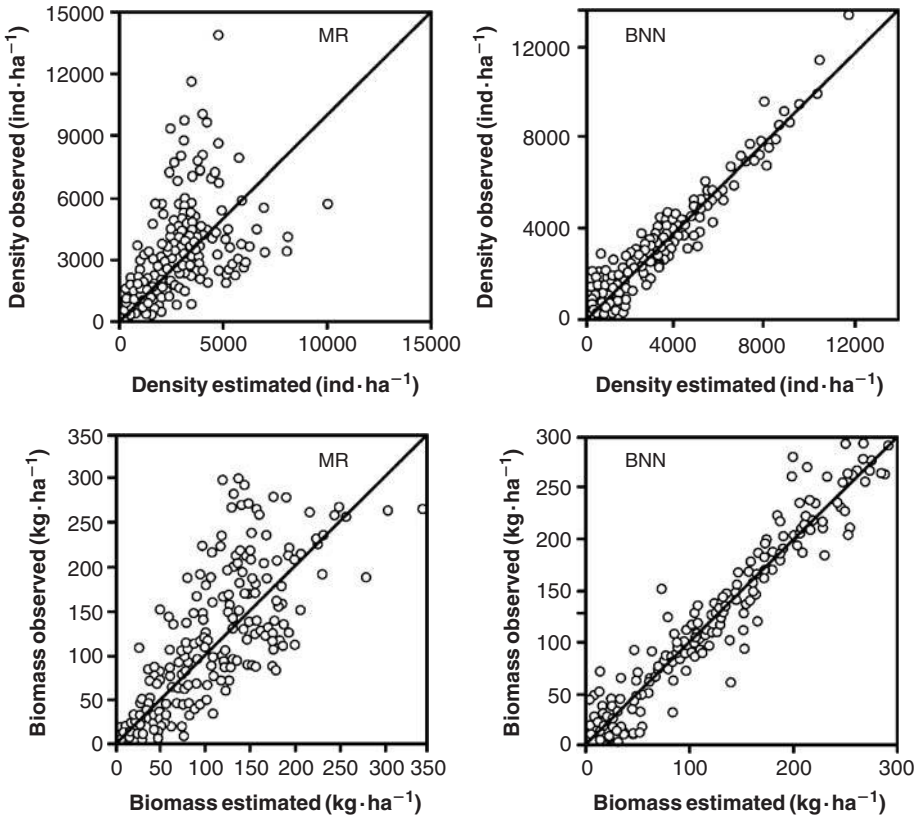


Fig. 11.2. Relationship between density or biomass, observed and estimated using multiple regression models (MLR) and backpropagation neural networks (BPNN). (Redrawn from Baran *et al.,* 1996, Figs. 11.2 and 11.3).

system, seventeen channel characteristics are used to forecast the abundance of six fish species and related variables (Table 11.3). The non-linearity of the relationships and the mix of data types suggest that NN should provide a more accurate forecast than a multiple regression approach and their results show this to be true. In this study 4250 fish were captured in 596 samples and of these 3911 were 0+. The specific species included in the model comprise 71% of the total. The NN forecasts the abundance of the species, which on this river is ecologically very helpful. The hydrology of the Garonne is influenced by dams along the main channel which have reduced microhabitat diversity. This has changed the overall diversity of the fish population. Robust species like Chubb were not affected by the changes but lentic species are reduced in numbers and rheophil and limnophil species are more dominant. The success of these NN models once again indicates that such technologies can provide useful planning tools for estimating the impact of physical change to a channel morphology on fish populations.

Table 11.3. The parameters used by Gozlan *et al.* (1999) for sites on the River Garonne.

Input parameters		Output parameters
Flowing channel (0/1)	% clay	Abundance of *Blicca bjoerkna*
Partially abandoned channel (0/1)	% silt	Abundance of *Leuciscus cephalus*
Abandoned channel (0/1)	% sand	Abundance of *Leponmis gibbosus*
Distance from bank (m)	% gravel	Abundance of *Barbus barbus*
Water depth (cm)	% pebbles	Abundance of *Gambusia affinis*
Slope of bank	% cobbles	Abundance of *Rutilis rutilis*
Temperature (°C)	Macrophytes (0/1)	Specific richness
Water velocity	Algae (0/1)	Number of 0+ fishes
% roots		Shannon diversity index

Mastrorillo *et al.* (1997a) use NN to explore the dynamics of *Phoxinus phoxinus* (minnow) populations because these fish are typical of the small species which are critical in the food chain dynamics of streams. The use of a BPNN shows minnow abundance increases as gravel and sand increases, but decreases with distance from the bank, stream velocity and pebbles. The BPNN successfully picks out the habitat preferences for this species as shown in other ecological studies (Mastrorillo *et al.*, 1996) provided there is a suitably large database. This work is further developed in Mastrorillo *et al.* (1997b) where BPNN forecast the presence or absence of three species (*Phoxinus phoxinus; Gobio gobio* – gudgeon; *Barbatula barbatula* – stone loach) in the same river. Comparison with a discriminant factor analysis shows that NN are able to assign a higher number of individuals correctly. Both methods can predict the presence or absence of the three species but the NN are more accurate.

Going beyond water quality, Aurelle *et al.* (1999) used NN to look at genetically differentiated forms of *Salmo trutta* (brown trout). From the river management perspective this study shows that the impact of stocking and the genetic differentiation of different populations of trout can be successfully undertaken.

Forecasting the biomass of *Clupea pallasi* (Pacific herring) stocks over a 41 year period (Chen & Ware, 1999) is an example of a longer-term ecosystem research application. NN performance of optimal models measured in terms of the coefficient of determination is between 0.6–0.7, whereas a multiple regression model achieved 0.29, and a process climate-stock recruitment model scored 0.42. Essentially the NN can forecast the biomass of the fish stock at a higher accuracy than alternative methods and in a way that positively assists in the management of fish resources.

6.4 *Macroinvertebrates*

The biological condition of rivers in the UK is monitored using the BMWP (Biological Monitoring Working Party) system where families of benthic macroinvertebrates are grouped according to their sensitivity to organic pollution. Families which have low scores are relatively insensitive and so are widespread whereas families with high scores are sensitive to pollution and generally only found at clean, rural streams (Chesters, 1980). BMWP scores are used in RIVPACS, a software model that combines macroinvertebrate information with catchment data to forecast the 'expected' species composition at clean sites (Wright *et al.*, 1995). This system uses the average score per taxon (ASPT) and the number of families present (NFAM) because they are less sensitive to seasonal and sampling variations than raw BMWP totals. Walley and Fontama (1998) report the results of a NN study to forecast ASPT and NFAM as a basis for biological classification of water quality. The thirteen input variables describe catchment location co-ordinates, distance from source, slope, alkalinity, discharge, altitude, boulders, pebbles, sand, silt, river width and depth. Two NN were developed and both solutions were observed to perform slightly better than the industry standard RIVPACS III. Where the NN appeared to be working less well the authors suggest the problem lies in the data quality rather than the forecasting mechanism. For example, high altitude sites gave relatively poor predictions, but of the five sites above 450 m, three had unusually high alkalinities so it is probable that these three sites lead to distortion in the results at non-alkaline locations. The input variables could be expanded to consider other geological or catchment characteristics, and it is probable that some of the observed error is due to the natural seasonal and spatial variation in species distributions.

6.5 *Algal blooms*

The issue of algal blooms in lakes and rivers is biologically significant and has implications for recreational users of the resource (Ferguson, 1997; Codd *et al.*, 1995). Bloom appearance is a complex function of nutrient loading, enriched runoff, BOD, air and water temperatures and physical characteristics of the river, lake or reservoir. Kneale and Howard (1997) showed the inadequate power of multiple regression to explain bloom behaviour in UK lakes and reservoirs, a problem that is relevant when data collection is restricted to monthly spot sampling. Statistics can describe these data, but forecasting potential is very limited. Recknagel *et al.* (1997) characterise the roles and limitations of different approaches to phytoplankton modelling (Table 11.4), an approach that could be usefully replicated in other areas of quality modelling.

Recknagel *et al.* (1997) applied NN to the problem of forecasting seasonal changes in the numbers of cells present for a range of algal species. These authors use different input and output layers, for each of the five lakes considered, but with a common architecture (Fig. 11.3). Their success in forecasting the number of cells present for a range of species from meteorological and

Table 11.4. Characteristics of phytoplankton modelling (from Recknagel *et al*., 1997).

	Empirical steady state models	Deterministic models	Time-series analysis models	Heuristic models	Fuzzy models	Neural network models
Data type	Cross-section	Cross-section/ time series	Time series	Cross-section/ time series	Cross-section/ time series	Time series
Time step for simulation		Minute/hour/day	Day		Month	Day
Time resolutions of predictions	Season/year	Day/week/month	Month	Month/season	Month/season	Day/week
Considered control factors for phytoplankton growth	Limiting factors	Limiting, physiological and multiple factors	Limiting factors	Limiting, physiological and multiple factors	Limiting and multiple factors	Limiting and multiple factors
Bases for predictions	Relationships	Trend/ seasonality/ serial dependency	Multivariate relationships	Seasonality/ serial dependency	Seasonality/ serial dependency	Connections
Predicted phytoplankton composition	Chlorophyll-a	Species/ functional groups	Chlorophyll-a	Species assemblages	Species assemblages	Species assemblages

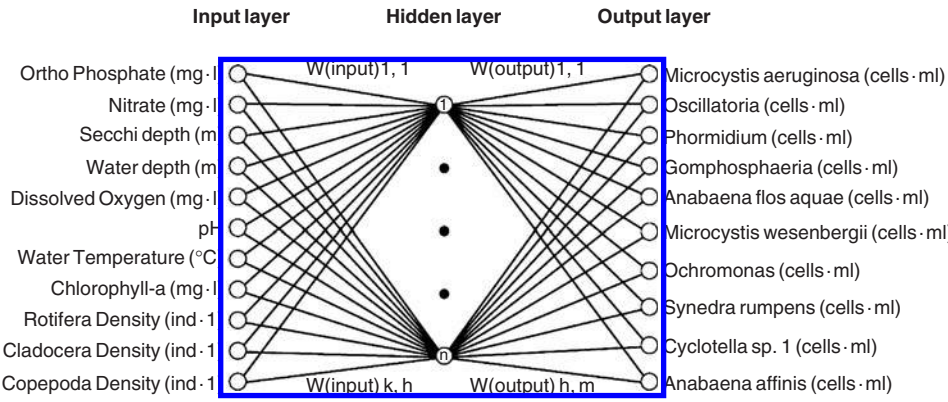


Fig. 11.3. NN structure used to forecast algal species at Lake Kasumigaura, Japan (from Recknagel *et al.*, 1997).

water quality information is indicative of the strength of this approach. These authors also looked in more detail at the internal weightings assigned by the different models and use sensitivity analysis to understand a little more about the processes driving the bloom forecasts. This allows them to determine that in the Lake Kasumigaura example, 'chlorophyll-a determines most of the dynamics of *Microcystis*' and that clumps of *Microcystis* are indirectly protected from grazing by zooplankton which preferentially feed on non-toxic algae. Effectively the NN models forecast the 'timing, magnitude and succesion of several algal species realistically, such as *Microcystis, Oscillatoria* and *Phormidium*' (Recknagel *et al.*, 1997).

When there are good data sets, as for the lower River Thames, where the UK Environment Agency collects flow, chemical and bio-indicator data regularly, algal forecasts should be more operationally useful. Whitehead *et al.* (1997) modelled the growth of algae and its transport downstream with the benefit of weekly data. Usefully this paper compares NN performance against that of a time series model and a dynamic mass balance and growth model. As a forecasting tool NN are seen to perform as well as the two alternative models, but as the authors state 'the advantage of NN is that no subjective information is required to determine the model structure or estimate parameters'. NN are therefore seen as the more pragmatic approach and one that can be implemented independently by forecasters who do not have detailed local catchment knowledge. Black box NN provide the forecast but do not give an insight into processes of growth and mass, for which the mass balance model is the more appropriate.

For the River Murray, Australia, Maier and Dandy (1997) and Maier *et al.* (1998) forecast *Anabena* cell counts using BPNN. Blooms occur over relatively short periods, 12–14 weeks in the summer months. This application shows that NN may be calibrated on data from the first seven weeks of the year

when colonies are developing but are not a management problem, and then used to make forecasts for the next four weeks when numbers multiply and the matter becomes more serious. The models provide useful forecasts of both the incidence and the magnitude of the peak in *Anabena* cell counts. As would be expected, temperature is the key variable, and forecasts based on temperature alone were very successful. Adding turbidity, soluble phosphorous and discharge to the inputs slightly improves the forecast, as does adding lagged data. For water management purposes a four week ahead forecast of cyanobacteria peaks is acceptable; it allows time for management intervention if wished and adequate warning if necessary. More recently Maier *et al*. (2001) looked at four-week ahead forecasting, comparing neurofuzzy associative memory neural networks (AMNN) with BPNN. This is an approach that others will no doubt want to explore because they show the AMNN to be more parsimonous and a slight improvement on the BPNN.

In searching for a management tool to deal with eutrophication in lakes Karul *et al.* (1999) develop a model to forecast chlorophyll-α. The inputs are PO_4 phosphorous, NO_3 nitrogen, alkalinity, pH, water temperature, suspended solids, conductivity, DOC, and Secchi depth. The comparable multiple regression model has a correlation coefficient of 0.71, which is very similar to the NN performance at 0.74. The authors comment that the NN win out where the inputs and outputs are not linearly correlated, but where the correlation is linear, the two approaches generate very similar results.

As in other areas the user must define the required output. If you want an estimate of chlorophyll-α at a given time interval, NN will do the job, but if you want process insights then look elsewhere.

6.6 *Urban runoff*

Urban storm water management presents a series of engineering challenges, not the least of which is limited data availability because continuous monitoring within pipes and sewers is expensive and difficult. There are some limited applications but the information is relatively sparse. Zaghloul and Abu Kiefa (2001) have reported on the use of NN approaches to perform sensitivity analysis in conjunction with the very widely used US Environmental Protection Agency Storm Water Management Model. Previdi *et al.* (1999) used NN to forecast flow in experimental urban networks and Gong and Denœux (1996) use PRNN within a model to forecast both flow and total suspended solids. The authors state that, in principle, their approach could be extended to forecast COD, nitrogen heavy metals and other variables but they are restricted to TSS by the availability of the data. The results are very good, improving on process modelling, but again data availability is a key limitation to further work. Ha *et al.* (2003) link a radial-basis-function neural network and a NN to classify land use and land cover and then link to a process based urban runoff and water quality model. They show that this coupling can be used to forecast changing water quality runoff in response to an actual or planned urban landscape change.

7 INTEGRATED MODELS FOR CATCHMENT MANAGEMENT

One can envisage a catchment forecasting model that integrates water resource demands with real-time flow and quality information, handles conflicting demand requirements including economic elements, makes decisions about flow regimes, extractions and releases, and thus provides a decision support system. Since these diverse demands are highly non-linear and data are likely to be recorded on different scales and time frames, NN should offer some advantages. The program described by Wen and Lee (1998) moves in this direction. These authors use data for the Tou-Chen River, Taiwan to seek management proposals which:

(a) find a reasonable allocation of waste loading for each pollution source against discharge to the river
(b) achieve higher standards of water quality for fish, and improve the environmental quality, and
(c) determine a basis for the total elimination of mass loading in a deteriorating river.

The flexibility of the NN approach allows economic and environmental variables to be linked in a manner that would not be clear in a process model. The authors conclude that their NN solutions can be used to manage the river basin and select feasible strategies to meet environmental quality goals.

The speed of execution of NN solutions and their general tendency to require fewer parameters than conventional models present the forecaster with tools to compile and compare multiple management scenarios. Yang *et al*. (1997), modelling accumulated pesticide concentrations in agricultural fields, showed that the model could be run in real time, while spraying was taking place, and allow field application rates to be varied. The output from this model, which has the potential to generate multiple scenarios, could provide the input to a hydrological drainage and runoff model.

There are a limited number of cases where there has been a comparative study of water quality and/or ecological model performance. In the ecological literature Paruelo and Tomasel (1997) test aspects of a simulated data series for a seasonal vegetation growth index and show that NN models outperform regression models as a forecasting tool. NN solutions handle the non-linearities in the system better. These authors generate their data series, which creates an interesting experiment, but field trials would be very valuable. This is the crux. The data must be available.

More recently these approaches have been used by Chen and Mynett (2003) to explore eutrophication, Shim *et al.* (2002) for flood control on the Han River Korea, Wang and Jamieson (2002) for wastewater treatment planning and Xia *et al.* (2001) to forecast lake basin water and salt balances.

8 ARE NN USEFUL FOR WATER QUALITY MODELLING?

Since Colasanti (1991) speculated on the value of NN for ecological forecasting there has been a wealth of hydrological water quality applications.

The similarities between the structure and behaviour of NN and natural ecological systems argues for the use of NN tools in ecological modelling. But do these NN applications improve on current forecasting practice and are the forecasts accurate enough and achieved at an acceptable cost?

Looking across the water quality modelling literature it is clear that in the last six years NN have been used in a diverse range of applications. Results from comparison studies with alternative statistical approaches suggests that the NN models give slightly better results. NN do demand large data sets but these solutions have a real edge in eco-biological studies in being able to handle data that are qualitative. Where model performance falls below expectation, errors are most frequently attributed to inadequacy in the data, as for example reported in Walley and Fontama (1998). When compared with statistical alternatives the evidence suggest that NN models can more accurately forecast non-linearity, handle complex inter-relationships and give a good indication of input-output responses. In water quality modelling hydroinformatics is thus seen to run in parallel with ecological informatics, hopefully to the benefit of both.

In many traditional examples of water quality modelling, the bio-chemical elements followed on as a second step after a hydrological model was created. The water quality elements were in some cases constrained by the structure of the original model and limited by the lower quality and consistency of data. There is no water quality equivalent of the 15-minute stage or rainfall record nor is the spatial distribution of these data comparable. The NN approach offers the opportunity for the modeller to treat all inputs independently and to link detailed and sparse data sets as inputs on an equal basis. The flexibility this affords is a major breakthrough. Their practical potential to combine diverse data, for example crop information with land use and weather data in agricultural modelling, has been highlighted by Schultz and Wieland (1997). Lek and Guegan (1999) stress the powerful potential of NN models to solve 'hard computational problems' including those where the underlying ecological relations are not understood. Gevrey *et al.* (2003) are confident enough to say 'Convinced by the predictive quality of artificial neural network (ANN) models in ecology, we have turned our interests to their explanatory capacities'.

NN software packages are available but clearly users need to understand the real world function well enough to make good judgements about NN model architecture, training and validation. Zhang and Stanley (1997) show that the variable which is most highly correlated with the output, yesterday's raw water colour with today's, does not give a good model. The strong autocorrelation in the data causes the model to forecast a repetition of today's values for tomorrow. 'The model lost its ability to distinguish the small individual difference between today's colour value and yesterday's' (p. 2343), but it is this crucial small scale difference that the water treatment manager requires. Hence the modeller's intuitive ideas about model structure were modified in the light of initial model forecasts. Starrett *et al.* (1988) describe the trial and error process involved in finding the optimum number of hidden layers and hidden nodes, and in comparing the effect of different transfer functions on the forecast. Indeed, as stated, 'tens of run were made with each new network structure to

determine the solution that had the lowest error' (p. 3103). The forecaster must take the time to investigate the effects of model architecture to ensure the validity of the results. However, this is one of the few papers to consider that the time taken to create and test the model was a constraint.

The major disadvantage in taking a NN approach is that it does not give the modeller further insights or explanations of catchment processes and dynamic change, although Recknagel *et al.* (1997) have shown that sensitivity analysis can produce useful insights about the processes. For the water quality manager this is a technique that is powerful, easily available and produces results that are directly understood by the user. Where this type of information is required then NN modelling offers a useful managerial tool. Working further with the approach and recognising its value in allowing the integration of various types of data, as in the Wen and Lee (1998) case where the approach is used to choose multiple options, is perhaps the next most useful step.

The increase in NN applications in the late 1990s has been followed by continued acceptance of the technique (Brion & Lingireddly, 2003; Lee *et al.*, 2003) and links to the development of fuzzy logic (Maier *et al.*, 2001), fusion modelling (Abrahart & See, 2002) and genetic algorithm rule-based models (Bowden *et al.*, 2003; Chen *et al.*, 2003; Wang & Jamieson, 2002). So far the dominance of backpropagation learning rules is clear. The availability of NN tools as shareware and the speed of model calculation make this an appealing set of technologies. Where the modelling alternative is a regression or correlation approach then the theoretical advantage of applying a non-linear modelling solution makes considerable sense.

REFERENCES

Abrahart, R.J. & See, L. 2002. Multi-model data fusion for river flow forecasting: an evaluation of six alternative methods based on two contrasting catchments. *Hydrology and Earth System Sciences* 6(4): 655–670.

Adgar, A., Cox C.S. & Böhme, T.J. 2000. Performance improvements at surface water treatment works using ANN-based automation schemes. *Chemical Engineering Research and Design* 78(7): 1026–1039.

Aurelle, D., Lek, S., Giraudel, J.L. & Berribi, P. 1999. Microsatellites and artificial neural networks: tools for the discrimination between natural and hatchery brown trout (*Salmo trutta* L.) in Atlantic populations. *Ecological Modelling* 120(2–3): 313–324.

Baran, P., Lek, S., Delacoste, M. & Belaud, A. 1996. Stochastic models that predict trout population density or biomass on a mesohabitat scale. *Hydrobiologia* 337: 1–9.

Baxter, C.W., Zhang, Q., Stanley, S.J., Shariff, R., Tupas, R.R. & Stark, H.L. 2001. Drinking water quality and treatment: the use of artificial neural networks. *Canadian Journal of Civil Engineering* 28: 26–35.

Beasley, G. & Kneale, P.E. 2002. Reviewing the impact of metals and PAHs on macroinvertebrates in urban watercourses. *Progress in Physical Geography* 26(2): 236–270.

Bowden, G.J., Maierm H.R. & Dandy, G.C. 2002. Optimal division of data for neural network models in water resources applications. *Water Resources Research* 38: 1010.

Brion, G.M. & Lingireddly, S. 2003 Artificial neural network modelling: a summary of successful applications relative to microbial water quality. *Water Science and Technology* 47: 235–240.

Brion, G.M., Neelakantan, T.R. & Lingireddy, S. 2001. Using neural networks to predict peak Cryptosporidium concentrations. *Journal of American Water Works Association* 93(1): 99–105.

Brosse, S., Giraudel, J.L. & Lek, S. 2001. Utilisation of non-supervised neural networks and principal component analysis to study fish assemblages. *Ecological Modelling* 146: 159–166.

Brosse, S., Guegan, J-F., Tourenq, J-N & Lek, S. 1999. The use of artificial neural networks to assess fish abundance and spatial occupancy in the littoral zone of a mesotrophic lake. *Ecological Modelling* 120: 299–311.

Campolo, M., Soldati, A. & Andreussi, P. 1999. Forecasting river flow rate during low-flow periods using neural networks. *Water Resources Research* 35(11): 3547–3552.

Cancilla, D.A. & Fang, X. 1996. Evaluation and quality control of environmental analytical data from the Niagara River using multiple chemometric methods. *International Association Great Lakes Research* 22(2): 241–253.

Charef, A., Ghauch, A., Baussand, P. & Martin-Bouyer, M. 2000. Water quality monitoring using a smart sensing system. *Measurement* 28: 219–224.

Chen, D.G. & Ware, D.M. 1999. A neural network model for forecasting fish stock recruitment. *Canadian Journal of Fisheries and Aquatic Sciences* 56(12): 2385–2396.

Chen, Q.W. & Mynett, A.E. 2003 Integration of data mining techniques and heuristic knowledge in fuzzy logic modelling of eutrophication in Taihu Lake. *Ecological Modelling* 162: 55–67.

Chen, W.C., Chang. N.B. & Chen, J.C. 2003 Rough set-based hybrid fuzzy-neural controller design for industrial wastewater treatment. *Water Research* 37: 95–107.

Chesters, R.K. 1980. *Biological Monitoring Working Party: The 1978 testing exercise.* TM No 19, Department of Environment: Water Data Unit.

Choi, D.J. & Park, H. 2001. A hybrid artificial neural network as a software sensor for optimal control of a wastewater treatment process. *Water Research* 3516: 3959–3967.

Chon, T.-S., Park, Y.S., Moon, K.H. & Cha, E.Y. 1996. Patternizing communities by using an artificial neural network. *Ecological Modelling* 90(1): 69–78.

Chon, T-S., Park, Y-S. & Park J.H. 2000. Determining temporal pattern of community dynamics by using unsupervised learning algorithms. *Ecological Modelling* 132(2): 151–166.

Cigizoglu, H.K. 2003. Incorporation of ARMA models into flow forecasting by artificial neural networks. *Environmetrics* 14: 417–427.

Clair, T.A. & Ehrman, J.M. 1998. Using neural networks to assess the influence of changing seasonal climates in modifying discharge, dissolved organic

carbon, and nitrogen export in eastern Canadian rivers. *Water Resources Research* 34(3): 447–455.

Clair, T.A., Ehrman, J.M. & Higuchi, K. 1999. Changes in freshwater carbon exports from Canadian terrestrial basins to lakes and estuaries under a 2 × CO_2 atmospheric scenario. *Global Biochemical Cycles* 13(4): 1091–1097.

Codd, G.A., Edwards, C., Beattie, K.A., Lawton, L.A., Campbell, D.L. & Bell, S.G. 1995. Toxins from cyanobacteria (blue-green algae). In W. Wiessner, E. Schnepf & R.C. Starr (Eds.), *Algae, Environment and Human Affairs.* (pp. 1–17) Bristol: Biopress Ltd.

Colasanti, R.L. 1991. Discussions of the possible use of neural network algorithms in ecological modelling. *Binary* 3: 13–15.

Corne, S., Kneale, P.E., Openshaw, S. & See, L. 1998. The use and evaluation of artificial networks in flood forecasting. 33rd MAFF Annual Conference.

Delgrange-Vincent, N., Cabassud, C., Cabassud, M., Durand-Bourlier, L. & Laine, J.M. 2000. Neural networks for long term prediction of fouling and backwash efficiency in ultrafiltration for drinking water production. *Desalination* 131(1–3): 353–362.

Ellis, J. B. & Hvitved-Jacobsen, T. 1996. Urban drainage impacts on receiving waters. *Journal of Hydraulic Research* 34(6): 771–783.

Ferguson, A.J.D. 1997. The role of modelling in the control of toxic blue-green algae. *Hydrobiologia* 349: 1–4.

Frakes, B. & Yu, Z.B. 1999. An evaluation of two hydrologic models for climate change scenarios. *Journal of the American Water Resources Association* 35: 1351–1363.

Gevrey, M., Dimopoulos, L. & Lek, S. 2003 Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling* 160: 249–264.

Golob, R., Stokelj, T. & Grgic, D. 1998. Neural-network-based water inflow forecasting. *Control Engineering Practice* 6(5): 593–600.

Gong, N. & Denœux, T. 1996. Urban stormwater pollution forecasting using recurrent neural networks. *Journal of Systems Engineering* 6: 137–147.

Gozlan, R.E., Mastrorillo, S., Copp, G.H. & Lek, S. 1999. Predicting the structure and diversity of young-of-the-year fish assemblages in large rivers. *Freshwater Biology* 41: 809–820.

Ha, S.R., Park S.Y. & Park, D.H. 2003. Estimation of urban runoff and water quality using remote sensing and artificial intelligence. *Water Science and Technology* 47: 319–325.

Huse, G. & Ottersen, G. 2003. Forecasting recruitment and stock biomass of Northeast Arctic cod using neural networks. *Scientia Marina* 67: 325–335.

Ibarra, A.A., Gevrey, M., Park, Y.S., Lim, P. & Lek, S. 2003. Modelling the factors that influence fish guilds composition using a back-propagation network: Assessment of metrics for indices of biotic integrity. *Ecological Modelling* 160: 281–290.

Joo, D S., Choi, D J. & Park, H. 2000. Determination of optimal coagulant dosing rate using an artificial neural network. *Journal Water Supply Research Technology Aqua* 49(1): 49–55.

Karul, C., Soyupak, S. & Yurteri, C. 1999. Neural network models as a management tool in lakes. *Hydrobiologia* 409: 139–144.

Karul, C., Soyupak, S., Cilesiz, A.A., Akbay, N. & German, E. 2000. Case studies on the use of neural networks in eutrophication modelling. *Ecological Modelling* 134: 145–152.

Kneale, P.E. & Howard, A. 1997. Statistical analysis of algal and water quality data. *Hydrobiologia* 349: 59–63.

Kneale, P.E., See, L. & Kerr, P. 2000. Developing a prototype Neural Net Flood Forecasting Model for Use by the Environment Agency. *Proceedings 35th MAFF Conference of River and Coastal Engineers*, 11.03.1–11.03.4

Laë, R., Lek, S. & Moreau, J. 1999. Predicting fish yield of African lakes using neural networks. *Ecological Modelling* 120(2–3): 325–335.

Lee, J.H. & Bang, K.W. 2000. Characterization of urban stormwater runoff. *Water Research* 34(6): 1773–1780.

Lee, J.H.W., Huang, Y., Dickman, M. & Jayawardena, A.W. 2003. Neural network modelling of coastal algal blooms. *Ecological Modelling* 159: 179–201.

Lek, S. & Baran, P. 1997. Estimations of trout density and biomass: a neural networks approach. *Nonlinear Analysis, Theory, Methods and Applications* 30(8): 4985–4990.

Lek, S. & Guegan, J.F. 1999. Artificial neural networks as a tool in ecological modelling, an introduction. *Ecological Modelling* 120: 65–73.

Lek, S., Dimopoulos, I. & Fabre, A. 1996. Predicting phosphorus concentration and phosphorus load from watershed characteristics using backpropagation neural networks. *Acta Œecologica* 17: 43–53.

Lek, S., Guiresse, M. & Giraudel, J-L. 1999. Predicting stream nitrogen concentration from watershed features using neural networks. *Water Research* 33: 3469–3478.

Maier, H.R. & Dandy, G.C. 1996. The use of artificial neural networks for the prediction of water quality parameters. *Water Resources Research* 32(4): 1013–1022.

Maier, H.R. & Dandy, G.C. 1997. Modelling cyanobacteria (blue-green algae) in the River Murray using artificial neural networks. *Mathematics and Computers in Simulation* 43: 377–386.

Maier, H.R. & Dandy, G.C. 1998. Understanding the behaviour and optimising the performance of back-propagation neural networks: an empirical study. *Environmental Modelling and Software* 13: 179–191.

Maier, H.R. & Dandy, G.C. 2001. Neural network based modelling of environmental variables: A systematic approach. *Mathematical and Computer Modelling* 33: 669–682.

Maier, H.R., Dandy, G.C. & Burch, M.D. 1998. Use of artificial neural networks for modelling cyanobacteria *Anabena* spp. In the River Murray, South Australia. *Ecological Modelling* 105: 257–272.

Maier, H.R., Sayed, T. & Lence, B.J. 2001. Forecasting cyanobacterium *Anabaena* spp. in the River Murray, South Australia, using B-spline neurofuzzy models. *Ecological Modelling* 146: 85–96.

Manescu, A., Ionescu, F., Marinescu, V., Marinescu, M. & Barbu, C. 1998. Informatic system for water quality protection in a river basin area. In V. Babovic & L.C. Larsen (Eds.), *Hydroinfomatics '98*. 1: 319–323. Rotterdam: Balkema.

Mastrorillo, S., Dauba, F. & Belaud, A. 1996. Utilisation des microhabitats par le Vairon, le Gougon et la Loche franche dans trois rivières du sud-ouest de la France. *Annals Limnology* 32: 75–85.

Mastrorillo, S., Lek, S. & Dauba, F. 1997a. Predicting the abundance of minnow *Phoxinus phoxinus* (Cyprinidae) in the River Ariège (France) using artificial neural networks. *Aquatic Living Resources* 10: 169–176.

Mastrorillo, S., Lek, S., Dauba, F. & Belaud, A. 1997b. The use of artificial neural networks to predict the presence of small-bodied fish in a river. *Freshwater Biology* 38: 237–246.

Milot, J., Rodriguez, M.J. & Serodes, J.B. 2002. Contribution of neural networks for modeling trihalomethanes occurrence in drinking water. *Journal of Water Resources Planning and Management-ASCE* 128: 370–376.

Moatar, F., Fessant, F. & Poirel, A. 1999. pH modelling by neural networks. Application of control and validation data series in the Middle Loire river. *Ecological Modelling* 120(1–2): 141–156.

Moog, O. & Chovanec, A. 2000. Assessing the ecological integrity of rivers: walking the line among ecological, political and administrative interests. *Hydrobiologia* 422: 99–109.

Morris, C.W., Autretm A. & Boddy, L. 2001. Support vector machines for identifying organisms – a comparison with strongly partitioned radial basis function networks. *Ecological Modelling* 146: 57–67.

O'Sullivan, P.J. 1991. *Application of a new technique for modeling system behavior*. Nanaimo, B.C.: Teramet IA Inc.

Obach, M., Wagner, R., Werner, H. & Schmidt, H-H. 2001. Modelling the population dynamics of aquatic insects with artificial neural networks. *Ecological Modelling* 146: 207–217.

Olden, J.D. 2003. A species-specific approach to modeling biological communities and its potential for conservation. *Conservation Biology* 17: 854–863.

Olden, J.D. & Jackson, D.A. 2003. A comparison of statistical approaches for modelling fish species distributions. *Freshwater Biology* 47: 1976–1995.

Omernik, J.M. 1997. Non-point sources stream nutrient level relationships: A nationwide study. EPA-600/3-77-105, September 1977. Corvallis Environmental Research Laboratory, Office of Research and Development, US EPA, Corvallis, Oregon, 151.

Paruelo, J.M. & Tomasel, F. 1997. Prediction of functional characteristics of ecosystems: A comparison of artificial neural networks and regression models. *Ecological Modelling* 98(2–3): 173–186.

Pearson, R.G., Dawson, T.P., Berry, P.M. & Harrison, P.A. 2002. SPECIES: A Spatial Evaluation of Climate Impact on the Envelope of Species. *Ecological Modelling* 154: 289–300.

Pigram, G.M. & Macdonald, T.R. 2001. Use of neural network models to predict industrial bioreactor effluent quality. *Environmental Science and Technology* 35(1): 157–162.

Poff, N., Tokar, S. & Johnson, P. 1996. Stream hydrological and ecological responses to climate change assessed with an artificial neural network. *Limnological Oceanography* 41(5): 857–863.

Previdi, F., Lovera, R. & Mambretti, S. 1999. Identification of the rainfall-runoff relationship in urban drainage networks. *Control Engineering Practice* 12: 1489–1504.

Recknagel, F. 1997. ANNA – Artificial Neural Network model for predicting species abundance and succession of blue-green algae. *Hydrobiologia* 349: 47–57.

Recknagel, F., French, M., Harkonen, P. & Yabunaka, K. 1997. Artificial neural network approach for modelling and prediction of algal blooms. *Ecological Modelling* 96(1): 11–28.

Reyjol, Y., Lim, P., Belaud, A. & Lek, S. 2001. Modelling of microhabitat used by fish in natural and regulated flows in the river Garonne (France). *Ecological Modelling* 146: 131–142.

Schleiter, I., Borchardt, D., Wagner, R., Dapper, T., Schmidt, K-D, Schmidt, H-H. & Werner, H. 1999. Modelling water quality, bioindication and population dynamics in lotic ecosystems using neural networks. *Ecological Modelling* 120: 271–286.

Schultz, A. & Wieland, R. 1997. The use of neural networks in agroecological modelling. *Computers and Electronics in Agriculture* 18: 73–90.

Serodes, J.B., Rodriguez, M.J. & Ponton, A. 2001. Chlorcast(C): a methodology for developing decision-making tools for chlorine disinfection control. *Environmental Modelling and Software with Environment Data News* 16(1): 53–62.

Shetty, G.R., Malki, H. & Chellam, S. 2003. Predicting contaminant removal during municipal drinking water nanofiltration using artificial neural networks. *Journal of Membrane Science* 212: 99–112.

Shim, K.C., Fontane, D.G. & Labadie, J.W. 2002. Spatial decision support system for integrated river basin flood control. *Journal of Water Resources Planning and Management-ASCE* 128: 190–201.

Starret, S.K., Adams, G., Hill, J. & Armatys, T. 2001. KTURF: a pesticide and leaching model, Online http://www.eece.ksu.edu/~starret/KTURF/ (Accessed 29 April 2003).

Starrett, S.K., Najjar, Y., Adams, G. & Hill, J. 1998. Modeling pesticide leaching from golf courses using artificial neural networks. *Communication Soil Science Plant Anal* 29(19–20): 3093–3106.

Stokelj, T., Paravan, D. & Golob, R. 2002. Enhanced artificial neural network inflow forecasting algorithm for run-of-river hydropower plants. *Journal of Water Resources Planning and Management-ASCE* 128: 415–423.

Teranet IA Inc. 1992. *ModelWare™ and ModelWare Professional User's Manual.* Nanaimo, B.C.:Teranet IA Inc.

Walley, W.J. & Fontama, V.N. 1998. Neural network predictors of average score per taxon and number of families at unpolluted river sites in Great Britain. *Water Research* 32(3): 613–622.

Walter, M., Recknagel, F., Carpenter, C. & Bormans, M. 2001. Predicting eutrophication effects in the Burrinjuck Reservoir (Australia) by means of the deterministic model SALMO and the recurrent neural network model ANNA. *Ecological Modelling* 146: 97–113.

Wang, C.G. & Jamieson, D.G. 2002. An objective approach to regional wastewater treatment planning. *Water Resources Research* 38: 1022.

Water Resources Council. 1973. Principles and standards for planning water and related land. *Federal Register* 38(3): 174.

Wei, B., Sugiura, N. & Maekawa, T. 2001. Use of artificial neural network in the prediction of algal blooms. *Water Research* 35(8): 2022–2028.

Wen, C-G. & Lee, C-S. 1998. A neural network approach to multiobjective optimization for water quality management in a river basin. *Water Resources Research* 34(3): 427–436.

Whitehead, P.G., Howard, A. & Arulmani, C. 1997. Modelling algal growth and transport in rivers: a comparison of time series analysis, dynamic mass balance and neural network techniques. *Hydrobiologia* 349: 39–46.

Wilson, H-H & Recknagel, F. 2001. Towards a generic artificial neural network model for dynamic predictions of algal abundance in freshwater lakes. *Ecological Modelling* 146: 69–84.

Wright, J.F., Furse, M.T., Clarke, R.T., Moss, D., Gunn, R.J.M., Blackburn, J.H., Symes, K.L., Winder, J.M., Grieve, N.J. & Bass, J.A.B. 1995. Testing and further development of RIVPACS, National Rivers Authority, R&D Note 453.

Xia, J., Chen, Z., Huang, G.H. & Rong, X. 2001An integrated hydro-ecological modeling approach applied to the Lake Bositeng Basin in China. *Water International*, 26: 105–118.

Yang, C.C., Prasher, S.O., Sreekanth, S., Patni, N.K. & Masse, L. 1997. An artificial neural network model for simulating pesticide concentrations in soil. *Transactions of the ASAE* 40(5): 1285–1294.

Yatsenko, V. 1996. Determining the characteristics of water pollutants by neural sensors and pattern recognition methods. *Journal of Chromatography A* 722: 233–243.

Zaghloul, N.A. & Abu Kiefa, M.A. 2001. Neural network solution of inverse parameters used in the sensitivity-calibration analyses of the SWMM model simulations. *Advances in Engineering Software* 32: 587–595.

Zhang, Q. & Stanley, S.J. 1997. Forecasting the raw-water quality parameters for the North Saskatchewan River by neural network modeling. *Water Research* 31(9): 2340–2350.

Zhou, S.J. 2003. Application of artificial neural networks for forecasting salmon escapement. *North American Journal of Fisheries Management* 23(1): 48–59.

12

Neural Network Modelling of Sediment Supply and Transfer

SUSAN M. WHITE
Institute of Water and Environment, Cranfield University, Silsoe, Bedfordshire, UK

ABSTRACT: The sediment supply-transport-deposition cycle, which controls sediment yield, consists of a complex and highly non-linear set of processes. All stages of the cycle are affected by a range of natural and anthropogenic influences. It is thus difficult to characterise and quantify all the important factors, making the effective measurement or prediction of sediment movement extremely difficult. Neural networks offer a way of including much complexity via both quantitative and qualitative inputs. Preliminary studies on soil loss from small catchments under varying land use, and sediment transport in several rivers in the north of England, have shown the potential benefits of neural network technologies to improve estimation of different phases of the sediment movement cycle.

1 INTRODUCTION

The supply and transfer of sediment to and through the river system consists of a number of complex phenomena. At all stages of the process, from initial detachment of sediment particles to arrival of sediment at a point of interest, a combination of highly non-linear and interacting processes contribute to this complexity. There are problems in the estimation of any individual process component, and therefore the estimation of a final sediment flux resulting from the combin-ation of all components is extremely difficult and prone to error. In contrast, many of the techniques available for estimation of various parts of the sediment supply and transfer system are very simple, and often empirically based. This combination of complex process and simplistic estimation occurs in several areas of science. For many projects in engineering, agriculture or hydrology, an assessment of erosion rates or sediment movement has to be made rapidly and without recourse to detailed and expensive field monitoring programmes.

As the contributory processes are so complex, their full explanation in mathematical terms is not possible. Even if it were possible to derive physically-based equations that reflect the full complexity of the situation, it is certain that for any given study or project, sufficient data would not be available to apply such techniques. This is particularly so in cases where one is trying to predict what may happen over the course of, for example, the next 100 years, during which time many of the controlling parameters may change. However, in many situations, an estimate of erosion rates or sediment yields must be obtained so recourse is made to the simpler and empirical models.

One may argue that where estimates are made for long time scales the potential change in short term sediment supply or transfer rates, due for example to changes in land use or management, are so great that complex modelling methods are not justified. Unforeseen changes in the system may alter estimates by a factor greater than the inherent error in an over-simplistic estimation method. There is merit in such a point of view; however, some estimation procedures result in extremely high errors, even for today's well monitored sites, which may also determine the success or failure of a project. Thus it seems logical to attempt to improve the estimation methodologies available. The development of artificial neural networks (NN) over recent years offers good potential to develop more robust and accurate sediment modelling solutions (Li & Gu, 2003; Nagy *et al.*, 2002; Tayfur, 2002).

In this chapter, an overview is given of the processes involved in sediment supply and transfer. The difficulties of measurement are discussed and a summary of widely used prediction techniques given. At each stage of this discussion the non-linearities and interactions between processes and stages of the supply-transfer chain are highlighted. The potential advantages offered by NN are then explored. This is followed by reports of some preliminary studies carried out using NN for the estimation of sediment yield from small catchments and for larger river sites. Finally, an assessment of results to date and future research directions is given.

2 THE SEDIMENT SUPPLY AND TRANSFER SYSTEM

The sediment supply and transfer system starts with the detachment of soil particles from their parent material or current location. Eroded material is transported overland, through the air, or through the river system and at some later stage is deposited. This sequence of processes occurs many times as a sediment particle moves from its original detachment site through the catchment (Fig. 12.1).

All three components of the sediment cycle vary, both spatially and temporally, and are affected by a range of controlling parameters. Sediment supply sources include geological weathering, freeze-thaw processes, sheet, rill and gully erosion, wind erosion, mass movements, construction sites, mining, logging, roads and river bank collapse. The rates of supply from these various sources are in turn affected by land use and management practices, climate, topography, geology and soil type. Thus both the potential for a particular site to supply sediment and the actual rate at which sediment is supplied to some

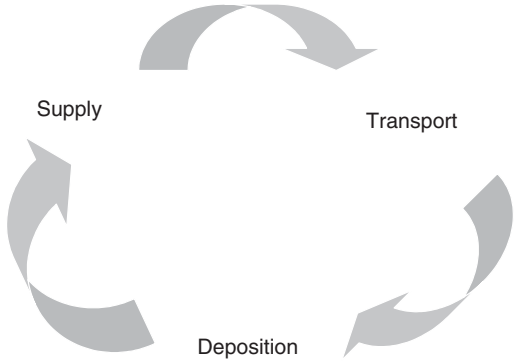


Fig. 12.1. The sediment cycle.

point down-slope or down-stream show extreme variability in both time and space. In areas where the regolith is disturbed, for example in mining and agriculture, the sediment load is likely to be enhanced and enriched with pollutants whose transport, deposition and re-erosion may also be of interest (Doyle *et al*., 2003; Erskine *et al*., 2002; Martin & Calvert, 2003; Melieres *et al.*, 2003).

Sediment transport is controlled by gravity and erosion agents such as water and wind. In mass movement the energy to move sediment down-slope is gravitational in nature, although water may well have played an important initial role in destabilising the slope. Where water flows over land or through a drainage network, the ability of the water to move particles is related to the depth and velocity of the water as well as the characteristics of individual sediment particles (size, shape, density). A similar combination of factors comes into play for sediment transport by wind. A sediment particle may take seconds or thousands of years to move from its initial point of detachment across a catchment and through a river network. Once in a river system, sediment moves in three main modes:

1. **Bedload** consists of large particles which are moved along in continuous contact with the river bed, i.e. shear velocities of flow are never sufficient to entrain this material into the body of the flow.
2. **Suspended bed load** consists of material that is intermittently in suspension in the flow body, and at other times on the river bed (either at rest or forming part of the bedload). Material is entrained from the river bed as the shear velocity of the flow increases, and is re-deposited as velocities decrease again. The result is that this material is found in varying concentrations throughout the river section. Generally a logarithmic profile is found through depth, with a normal distribution across the section. Maximum transport rates are generally in mid-river and in the mid-depth range, coincident with the maximum velocity zones in straight river sections. The entrainment and deposition process is complicated by the burial and shading of soil particles by larger particles. Thus material of a size that could, theoretically, be entrained at a certain velocity may be trapped on the river bed under larger particles and thus unavailable for transport. In extreme cases river beds may effectively be arrowed by large particles.

3. **Washload** consists of material that is constantly in suspension, and is found at constant concentrations throughout the river depth, and across the profile. For estimation methods and measurement programmes this is often defined as material finer than 63 μm in diameter, although intuitively this criterion will be different for different flow rates.

Normally measurements of bedload and suspended material (including washload) are made separately. Bedload is often not measured directly but is estimated from the bed grain size data.

At any point from the time of initial detachment sediment can be picked up and transported by a body of water with sufficient energy. Once that energy is dissipated in some way, such as a reduction in flow rate, water entering a reservoir, a wider section of river, or shallow flow over floodplains, sediment will settle out of the flow and be deposited. This happens in both overland and river flow. Thus a particular soil particle can be moved in several steps from its original location to a river, with a time delay of days to centuries, depending on the obstacles that are in its way. Once in a river, sediment can only be transported when there is sufficient flow to do so. For a particular river, flow will never have sufficient energy to move some particles; some will only be able to move along the bed and some will be intermittently in suspension in the flow, whilst the very finest particles may be permanently in suspension. Individual particles can be buried under other particles and so trapped within the river system for considerable amounts of time before re-entrainment.

Thus transport of sediment through the river system is dependent on both the supply of sediment to the river channel and the ability of the flow in the river to move the supplied sediment. Rivers are often described as supply or capacity limited transport systems, although in reality there is not a discernible transport capacity limit for the finest sediment particles. In practice a river can move from one state to the other dependent on the temporal distribution of both sediment supply (amount of sediment and sediment particle size) and river flow. In the majority of rivers the ability of the river to transport sediment is not the limiting factor; it is rather the supply of sediment from the range of sources in the catchment which restricts the amount of sediment moved. This means that for most rivers the relationship between sediment concentration and water discharge exhibits marked hysteresis. This can be in either a clockwise or anticlockwise direction dependent on the proximity of active sediment sources to the monitoring point. In practice, data from an individual site will exhibit both leading and lagging sediment peaks relative to the flow peak. Figure 12.2 demonstrates this for a site on the River Tees in north-east England for a series of events before and after a large storm (on 28 October 2000) which activated new, local sediment sources. Post-flood events all had leading sediment peaks. Hysteresis for events with leading and lagging sediment peaks (above and below zero time difference on the graph) is in opposite directions.

White (1996) assessed the contribution to overall sediment yield of wet and dry season flows, thunderstorms and tropical cyclones for a reservoir catchment in the Philippines. It was generally held that the cyclone events, which generate

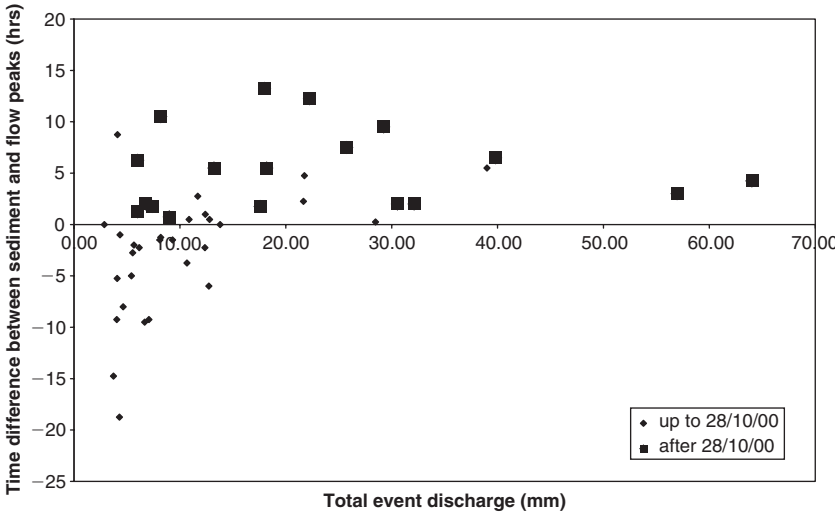


Fig. 12.2. The impact of new, local sediment sources on the time difference between sediment concentration and flow peaks. Sediment peaks in advance of flow peaks are positive.

large widespread flooding, were the main reason for the high sedimentation rates of reservoirs, but it was also clear that much of the sediment supply work was done by smaller frequent and localised thunderstorm events. When a cyclone occurred there was already a large amount of detached sediment moving through the river system, which was available for transport when the river discharges were raised.

It is also clear from a number of comparative and paired catchment studies, that sediment supply can be dramatically reduced (or increased) by human activity. For example, in a study in Malawi, White *et al.* (1988) found that the difference in sediment yield between a fully managed and a traditionally farmed catchment was of the order of 100 times. These characteristics of sediment movement, which are temporally erratic, spatially variable and impossible to monitor on an appropriate scale to permit process modelling, suggests that NN should theoretically present a practical option for the forecaster.

Large catchments do, however, behave as integrators of the individual parts of the catchment, and in the same way as hydrological modelling began with 'lumped' and 'empirical' approaches and developed to 'distributed' and 'physically-based' tools, so sediment yield modelling has followed a similar route.

3 EROSION AND SEDIMENT YIELD ASSESSMENT

For the NN modeller the potential to integrate hillslope and river processes is attractive. It has been shown elsewhere in this book that river flow, which is a main controlling factor for sediment transport, may be forecast reasonably

well with NN. But there is no absolute requirement to model river flow as a pre-requisite to modelling sediment transport, as early empirical models show. Sediment supply and transport rate forecasts have historically developed from two directions, agricultural engineering and civil engineering. These two sub-disciplines developed distinctively different approaches because of the differing objectives of their studies. Agricultural engineers were primarily interested in field scale erosion of soil, i.e. the loss of soil from fields, and measures to prevent it. Civil engineers were interested in the transport of sediment by rivers, and particularly in the long-term yield of sediment to reservoir impoundments. Over the years, work by specialists with other interests and disciplines have resulted in new approaches to erosion and sediment transport estimation. The following section gives an overview of the techniques that are available for assessment of field-scale erosion, in-stream sediment transport and long-term sediment yield. The potential use of NN at each of these scales is highlighted.

3.1 *Field-scale erosion rates*

Work to assess field-scale erosion rates developed rapidly early in the 20th century as a result of severe erosion in the USA. A series of monitoring sites were developed to measure the rate of sediment loss for different soils and topographic conditions under different land uses, land management regimes and rainfall. Measurements were carried out on standard rectangular plots. Such measurements have been replicated in many countries throughout the world. In the USA, a total of 10,000 plot years of data was collated and used to develop an empirical relationship between soil loss and its controlling factors (Wischmeier & Smith, 1978). This relationship, the Universal Soil Loss Equation (USLE), forms the basis for many erosion and sediment yield estimation techniques.

The basic model is:

$$E = R \cdot K \cdot LS \cdot C \cdot P \tag{1}$$

where:

E = Rate of soil loss
R = Rainfall erosivity parameter
K = Soil erodibility parameter
LS = A topographic factor, accounting for slope length and steepness
C = A crop cover factor
P = A management factor

The equation was modified for situations where runoff or rainfall dominates the sediment detachment process, to produce RUSLE (SWCS, 1995) and MUSLE (Williams & Berndt, 1977) respectively. In Southern Africa, a similar technique, the Soil Loss Estimator for Southern Africa (SLEMSA), was developed (Elwell, 1980).

Alternative attempts were made to develop a more physically-based model to include field-scale processes such as rills, under the auspices of the Water Erosion Prediction Project (WEPP) (Flanagan & Nearing, 1995; Renschler, 2003).

In fact what is actually estimated by these techniques is a very small-scale land-based sediment yield figure. This can give an idea of loss of soil from an agricultural area, or the potential relative contribution of a certain area of a river catchment to the overall sediment load carried by the river. These techniques allow comparison of different potential land uses or management techniques in terms of their effectiveness at maintaining soil in place and can, therefore, be used in planning, design and assessment of catchment management plans or as a means of evaluating soil conservation programmes.

The USLE and its derivatives are a common method that is used to provide the soil loss estimation component in several catchment scale hydrological models. The widely used Soil Water Assessment Tool (SWAT) (Arnold *et al*., 1998; Chanasyk *et al*., 2003) has an erosion model based on MUSLE. In common with other USLE based models this effectively restricts its use to areas where sheet and rill erosion dominate the sediment supply system.

3.2 *Sediment transport rates*

In order to design sediment extraction devices, to plan dredging programmes, to understand sediment supply and movement through a catchment, and to assess the influence of sediment on river ecology and habitat (particularly for fisheries; see Chapter 11), we need to estimate sediment transport rates at a point.

This can be done in two ways, although in reality monitored data are required to validate any modelling approach:

- Measurement of sediment concentrations across a river cross-section. This is labour intensive and expensive. In order to define the whole sediment transport regime, measurements must be carried out at frequent intervals over a long period. In addition, there is often hysteresis in the sediment concentration-discharge curve, and this may change with varying antecedent conditions and with changes in the catchment (on a seasonal or longer term). The result is a high degree of uncertainty, with the sediment rating curve often consisting of a cloud of points around a regression line. It may also be necessary to carry out repeat surveys of the river channel to identify deposition and erosion zones.
- A number of sediment transport formulae are available (for a review of sediment transport in rivers, see Fisher, 1993). These provide an estimate of the capacity of the river to transport sediment of a given size, and thus theoretically provide an upper limit to the sediment transport rate. There are many equations available, most of which have been developed with flume data. Such equations may be unrealistic for natural river conditions. Most of the equations require a representative bed grain size as input. This is very difficult to obtain, as bed material clearly changes with flow condition. The available equations normally give a wide range of answers for a given river site.

- Other investigators use a 'stream power' approach, first suggested by Bagnold in 1966, which considers a body of moving water in a river to be analogous to an engine able to exert power in order to carry out the work of sediment transport. This approach has been revisited and improved recently (Finlayson & Montgomery, 2003; Martin & Church, 2000).

3.3 *Sediment yield*

Sediment yield can be defined as the total volume or mass of sediment passing a point of interest over a specified time. For example, the long term average annual input of sediment to a reservoir may be needed for the practical management of siltation problems or be required in reservoir design modelling.

The need to estimate sediment yield has mainly and historically been related to the construction of large dams. As well as estimating how long a reservoir will take to fill with water, there is also a need to estimate how fast it will fill with sediment. This defines the viable life of the reservoir. In reality sediment yield estimates are subject to so much uncertainty that it is extremely rare for a proposed dam project to be abandoned because of predicted high sedimentation rates. At best, the design may be adapted to include some sediment exclusion or extraction device such as low level sluices, but these are often ineffective.

Sediment yield estimates for dam projects often constitute 1 or 2 pages in an overall feasibility study running to hundreds if not thousands of pages. And yet one of the major problems encountered in many reservoirs around the world is that of excessive sedimentation. Even where sedimentation is not a problem in volume terms, water quality problems may be associated with input of contaminants attached to the sediments. Sediment inputs are almost without exception in excess of those predicted at the feasibility stage, often by an order of magnitude (see for example: Brabben, 1982; Fish, 1983; Patnaik, 1975; Dickinson *et al*., 1990). This poor record of pre-construction estimates of sediment yield rates has many contributory causes. In order to understand the shortcomings in currently available techniques, it is necessary to look in more detail at the options that are available for sediment yield estimation, and the factors that contribute to error.

There are four basic options for sediment yield estimation:

- Measurement of sediment concentration and stage (river flow) in order to develop a sediment rating curve. This is then used in conjunction with a long-term river discharge record to estimate long-term sediment yield. This is subject to the same problems of measurement as discussed above.
- The use of hydrological models of various types (conceptual, black-box, stochastic, physically-based). Even the most sophisticated hydrological models use empirical relationships for sediment components, often based on USLE. Although some make an attempt to impose a more rigorous physically based approach, even these resort to empirical 'parameters'. In order to get the supply side of the equation correct, predictions of overland flow have to be correct. There is much debate in the literature over whether any model (even the most complex) can realistically model internal catchment

processes. Although models may make a good prediction of river flow at the catchment outlet, the various components of that flow (overland, shallow soil, groundwater interaction, etc.) may not be well estimated in time or space and both are necessary pre-requisites for accurate modelling of sediment supply and transport.

- Sediment yield models (e.g. Al Kadhimi, 1980; Fournier, 1960) are normally highly empirical. They are based on simple bulk catchment characteristics such as catchment area, main channel slope and vegetation type.
- GIS based models of erosion and sediment transfer, which are based on erosion risk, are themselves often based on one of the empirical soil loss models in an attempt to represent spatial variability in sediment supply. The link between the estimation of movement of eroded sediments from the hillslope to and within rivers is often weak. Some work has been based on distance and slope to main river channel, i.e. the further the original erosion is from a river and the shallower the slope between it and the river, the longer it will take for eroded material to move into the river. Some attempts have also been made to integrate this sort of approach with more sophisticated distributed hydrological models (e.g. Burton & Bathurst, 1998).

In estimating at any scale (field-scale erosion, sediment transport and sediment yield) we are interested in three phenomena and their interactions - supply, transport and deposition of sediment. As described above, all of these processes are extremely complex and non-linear, and show high temporal and spatial variability, with in-built and variable lags in the system and continue to need field assessments to judge reliability (Walling *et al.*, 2003).

4 WHY USE NN FOR SEDIMENT MODELLING?

A consideration of the factors affecting supply and transport of sediment and the inadequacy of the techniques available for estimation allows us to list a number of possible advantages of NN sediment modelling:

1. Non-linear behaviour can be learnt – e.g. variation due to antecedent conditions, location of rainfall events, seasons, thresholds, etc.
2. There is no need to explicitly define all contributory processes as they can be represented by proxy variables (e.g. date). This is also true of statistical models.
3. Input variables can include those that use, for example, varying upstream inputs to represent spatial variability (e.g. flow rates for upstream tributaries or upstream raingauge data). Causal links do not have to be explicitly explained.
4. NN have been shown to work for river flow prediction, which is one of the main controlling factors for sediment transport.
5. There is no requirement with NN to model river flow as a pre-requisite to modelling sediment transport, thus removing one major source of potential error.

6. Where data are scarce there may be a possibility to transfer models between hydrologically similar rivers with geographically similar catchments - although 'similar' remains to be defined.
7. NN are mathematically and computationally less demanding than most sediment related modelling alternatives.
8. Non-contributing variables can be identified and removed or ignored. This is also true for statistical models.
9. Some understanding of the physical system functioning may be gained by an analysis of which events are affected by the removal of input variables (Abrahart *et al.*, 2001) or hidden neurons, which represent specific hydrological processes, e.g. overland flow, flow recessions, flow peaks (Wilby *et al.*, 2003).
10. Potentially, more general models can be developed.

5 PREVIOUS NN STUDIES IN SEDIMENT MODELLING

There are a limited number of studies that have used NN to model the sediment supply and transfer system. Abrahart and White (2001) applied NN to a series of four small experimental catchments in Malawi. The catchments experienced differing land use and management regimes and had been monitored over a period of 3–5 years. A small data set with 117 records containing rainfall, runoff and sediment yield were available covering the 1981 to 1985 rainy seasons. Models for predicting sediment output were developed using multiple linear regression (MLR) (Fig. 12.3) as a benchmark and a fully connected 8:8:1 NN trained with backpropagation and jitter, which is the addition of artificial noise (Fig. 12.4). In both cases, proxy variables were used to represent different catchment characteristics. Due to the small size of the data set, the networks

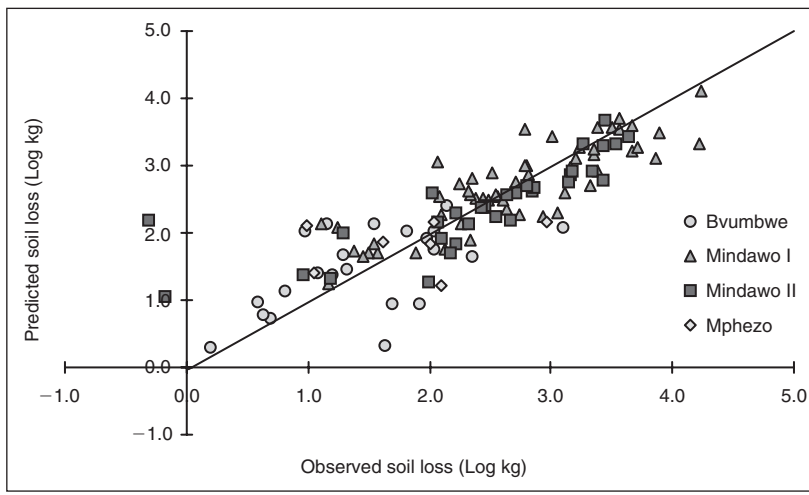


Fig. 12.3. Multiple linear regression prediction with superimposed line of perfect agreement.

were trained with increasing amounts of jitter, which was added to the input patterns during training in order to facilitate a generic solution and to avoid over-fitting.

The results showed that the NN significantly outperformed the MLR approach. The NN solution provided a tighter fit to the data and created a marked reduction in the number of pronounced outliers. The NN approach also had the added advantage of producing a solution with a similar pattern of error for each catchment unlike the MLR, which exhibited bias in favour of the catchment with the largest number of recorded observations. The NN technique produced a more flexible response to changing circumstances and the pattern of error between the individual catchments was equalised so that the model could be transferred to unknown scenarios with reasonable confidence.

Jain (2001) used NN to model the sediment-rating curve at two sites on the Mississippi River. This work built on his earlier research to model discharge rating curves, which also used NN (Jain & Chalisgaonkar, 2000). A sediment rating curve expresses a non-linear relationship between the sediment and river discharges and is usually modelled using regression analysis and curve fitting. NN provide an alternative method for modelling this non-linear relationship. Different combinations of input variables were used including current and previous levels, discharge and sediment concentration inputs while the output layer had two nodes, one for discharge, and one for sediment concentration. The number of hidden layer nodes was determined through trial and error. Conventional rating curves were also developed for the two gauging stations concerned to provide a comparison. The results showed that the NN provided a better estimate of sediment discharge for the validation period at both sites, relative to a conventional curve fitting approach, which encountered problems with peak prediction. NN were also successful in modelling the hysteresis concentration using 161 observations from 4 streams. Half of the observations

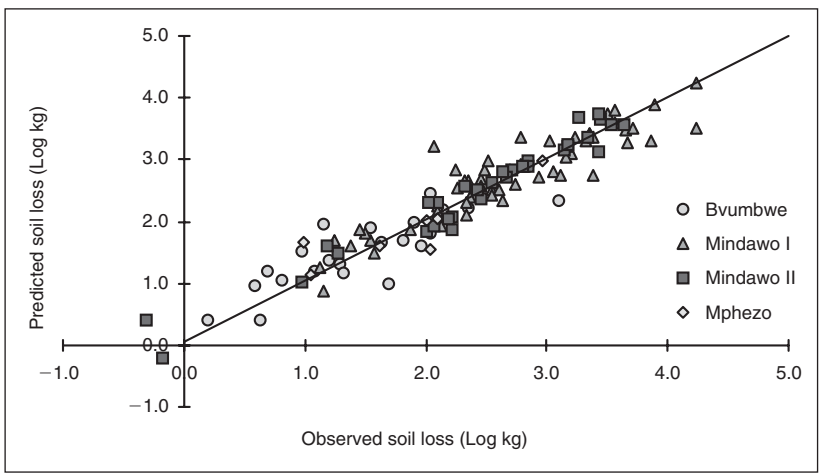


Fig. 12.4. NN prediction using standard inputs and jitter set at ±0.0%.

were used for training and the remaining half for verification. The input variables included 6 parameters covering tractive shear stress, suspension, water depth ratio and Froude effect.

Nagy *et al.* (2002) used NN trained with backpropagation to estimate total sediment concentration. Testing was undertaken to determine the most suitable NN discharge, Reynolds numbers and stream width ratio, which were used to predict the sediment transport. The final architecture contained 12 hidden nodes. The model was verified with a large number of data points from several rivers. The results showed that the NN approach estimated sediment concentrations well compared to conventional methods. The authors also compared the NN results with seven other formulae presented in the literature, and the NN model gave the best results overall.

NN have also been used to model sheet sediment transport (Tayfur, 2002). Three-layer feedforward networks were trained and tested using experimental slope and rainfall intensity input data to predict sediment discharge as an output. The NN were compared to results from the most commonly used physical models including flow velocity, shear stress, stream power and unit stream power. The results showed that the NN performed as well as the physically based models at predicting nonsteady-state sediment loads at varying slopes. The various approaches were also used to predict the mean sediment discharge from experimental runs. The results indicated that the NN performed better than the physically based models at steep and very steep slopes under very high rainfall intensities.

6 NN EXPERIMENTS

This section discusses initial results from an investigation into the possibility of modelling sediment concentration directly from flow rate without the need for a sediment-rating curve. Data from the Land Ocean Interaction Study (LOIS) (Wass & Leeks, 1999) and the sustainability in managed barrages study (SIMBA: www.silsoe.cranfield.ac.uk/iwe/projects/simba.htm) were used in these experiments. Flow and sediment concentration data for two stations were used: the Low Moor station on the River Tees, and the Thornton Manor station on the River Swale, both in north east England. The catchments providing sediment to the Swale River and the Tees River have similar characteristics, which justified the use of the data for both rivers during the training and testing stages.

The backpropagation method was employed to train the NN. A series of trials were undertaken to determine optimum values for the number of nodes in each layer, the number of hidden layers, the learning rate, the momentum rate and the connection weights. Fixed stopping was used and all data were scaled between 0 and 1.

6.1 *Spatial transfer based on flow*

The first set of experiments was designed to train NN for a station on the River Swale and for the solution to be tested and applied to a station on the River Tees. NN were trained using 15-minute flow values as inputs for the time

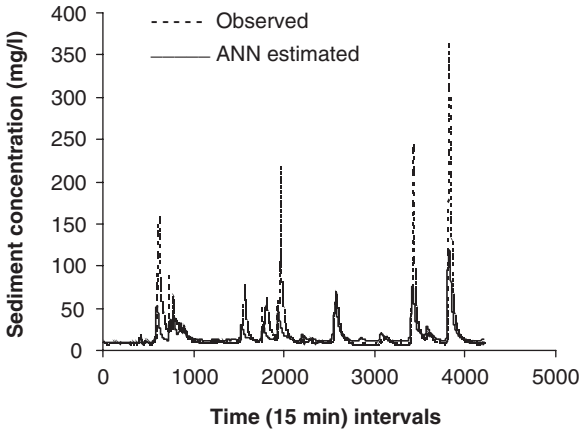


Fig. 12.5. The estimated and actual suspended sediment concentrations at Low Moor for the period 23/01/2000 to 07/03/2000 at time t. The six network inputs were flow t to $t - 5$.

period 10/11/1994 to 25/12/1994 for the River Swale. Six flow values (t to $t - 5$) were used to estimate the unique suspended sediment concentration at time t. The testing of the trained NN was accomplished with 15-minute flow and suspended sediment data from the Low Moor station on the River Tees for the time period 23/01/2000 to 07/03/2000. The results are shown in Figure 12.5.

The results indicate that the suspended sediment concentration predicted by the NN generally approximated the observed value but underestimated the peaks. The MSE and the estimated total sediment yield at the end of the testing period were 472 and 3496 t respectively. The difference between the estimated and observed sediment yield (5664 t) was 38%.

The experiment was repeated, replacing the training data with the testing data and vice versa, i.e. the Tees River data were used for training and the Swale River data for testing. The time periods of the data considered during both stages remained unchanged. The results showed that the NN predicted the general behaviour of the observed Swale series but again there was a general underestimation of the peaks. The NN estimated a sediment yield of 7739 t, which was just 4.4% higher than the observed value of 7412 t, with an MSE of 632 t. The small difference in yield estimates arose because the underestimation of peaks was balanced by an overestimation of low flows.

The above experiment was then repeated with the input and output data rescaled to between 0.2 to 0.8 and an input time interval from $t - 8$ to $t - 13$. The results showed that changing the input time interval and the scaling limits provided closer approximations to the observed values. The MSE decreased to 533 t and the total sediment yield estimated was 2.3% higher than that observed.

These results show that if river flow and suspended sediment data from a nearby river with similar catchment characteristics are available, then the general behaviour of the suspended sediment time series can be estimated using the available flow series for the corresponding time interval. The results

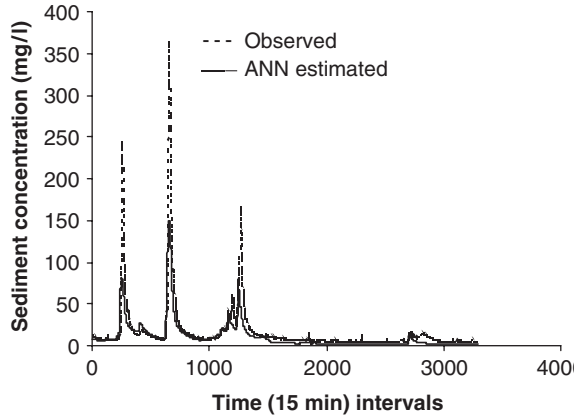


Fig. 12.6. The estimated and actual suspended sediment concentrations at Low Moor for the period 25/02/2000 to 30/03/2000 at time t. The six network inputs were flow t to $t - 5$.

obtained are especially significant considering the expense of installing sediment measurement equipment and the importance of providing realistic future estimates for a river's potential sediment yield.

6.2 *Temporal transfer based on flow*

In the second set of experiments, NN were trained on 15-minute data from the Low Moor station on the River Tees for the time period 21/01/2000 to 25/02/2000 and tested on a later period at the same location. The input data consisted of six flow values from t to $t - 5$ and the output was the unique suspended sediment concentration at time t. The testing stage covered 25/02/2000 to 30/03/2000. The NN estimated time series was again close to the observed record but the peaks were underestimated (Fig. 12.6). The total estimated sediment yield, 3980 t, was 5.2% less than the observed value of 4206 t.

In this experiment it was shown that even in the absence of observed sediment data, it was possible to obtain reliable corresponding estimates by training with sediment and flow data from previous events at the same location.

6.3 *Temporal transfer based on sediment data*

If some sediment data are available then preceding values of sediment concentration can be used as training data. This was carried out for the River Ouse at Skelton for the time period 07/01/1995 to 18/01/1995 using the 6 previous sediment concentration values as inputs ($t - 1$ to $t - 6$). The results are shown in Figure 12.7, which indicates that the availability of the preceding sediment concentration data would improve the ability of the NN to predict future sediment concentration values (the MSE in this experiment was just 32.5 t). The sediment concentrations predicted by the NN do not reflect the full variability of the monitored data, although it is not certain that the high variability seen in the monitored data is a true representation of sediment movement in the river. Sediment monitoring is a complex procedure with high levels of uncertainty

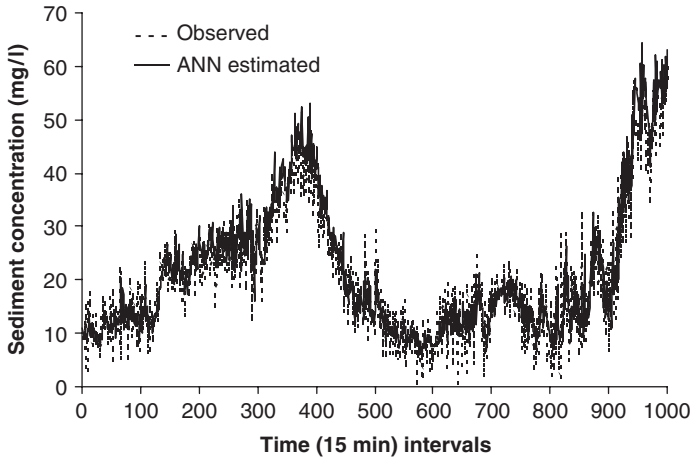


Fig. 12.7. The estimated and actual suspended sediment concentrations at Skelton for the period 07/01/1995 to 18/01/1995 at time t. The six network inputs were suspended sediment concentration at $t - 1$ to $t - 6$.

attached to the data. Even so, the total sediment yield estimated from field data and from the NN are very close, and in terms of estimating sedimentation rates in reservoirs, this would be an entirely acceptable estimation procedure. However, this example must be considered as illustrative only, as the existence of immediately preceding sediment concentration data would mean that this is a situation where one may not, in practice, need to use NN.

In these examples the data sets used were small and the training times for the NN were limited. It is clear that with the availability of a longer continuous data set the NN would be trained with more input and output patterns. This would increase the accuracy of the models by providing better validation results.

No technique can be expected to model processes that are outside the range of those considered in its development, and yet this is often what happens with the modelling of sediment yield. Data may be available for a series of flow conditions, but are usually not monitored during extreme flows. This is in part because extreme flows are by definition rare and will probably not occur during any monitoring period, and in part because of the difficulties of obtaining a true measurement of sediment transport rate in high flow conditions. Normally, very few data are available at all. In such circumstances it is important to include process understanding in any modelling approach. This can potentially be done with NN by including information on upstream conditions without explicitly defining the controlling processes.

7 CONCLUSIONS

The modelling of any part of the sediment supply-transport-deposition cycle is highly complex, involving many non-linearities and a diverse mix of controlling

factors. Existing techniques are limited by the ability of models to fully represent this complexity. In particular the often extreme spatial and temporal variability of all phases of the cycle presents difficulties for both data input and model construction. NN offer an alternative estimation procedure with the capability to learn complex and non-linear behaviour. A major advantage is the possibility of using proxy variables to represent different climatic or hydrological situations or land use management combinations within a catchment. Such an approach means that important influences can be included in the model without explicit parameterisation. Preliminary tests at the small catchment and river basin scale have shown promising results. Work is ongoing to combine such a modelling approach with recent developments in field data collection in order to better represent variability in the system.

There are relatively fewer applications for NN modelling in sediment transport forecasting than there are, for example, in the water quality and fisheries areas (Table 11.1) but the potential for use is just as extensive. Given that bedload sediments are critical for spawning, or as habitat sites, and that pollutants readily attach themselves to sediments on slopes in the stream bed, and when suspended in the water column, the scope for applied forecasting is considerable. There is also, for example, little research on coupling the hillslope-stream interface in this context. The USLE model has held its premier position for nearly a century. Its straightforward approach is appealing in simplifying processes that are distributed and highly complex. NN would appear to have an advantage in being mathematically and computationally less demanding than process-based hydrological modelling approaches and have the potential to generate new options in both hillslope runoff modelling and stream sediment transport forecasting.

REFERENCES

Abrahart, R.J. & White, S.M. 2001. Modelling sediment transfer in Malawi: Comparing backpropagation neural network solutions against a multiple linear regression benchmark using small data sets. *Phys. Chem. Earth (B)* 26: 19–24.

Abrahart, R.J., See, L. & Kneale, P. 2001. Applying saliency analysis to neural network rainfall-runoff modelling. *Computers and Geosciences* 27: 921–928.

Al Kadhimi, A.M.H. 1980. *Land Use, Water Yield and Soil Erosion: Simulation of Cause and Effect*. Unpublished PhD Thesis, Dept of Civ Eng, Univ of Strathclyde, Glasgow, UK.

Arnold, J.G., Srinivasan, R., Muttiah, R.S. & Williams, J.R. 1998. Large area hydrologic modeling and assessment part I: model development. *Journal of American Water Resources Association* 34(1): 73–89.

Bagnold, R.A. 1966. *An Approach to the Sediment Transport Problem from General Physics*. United States Geological Survey Professional Paper 4221.

Brabben, T.E. 1982. *Sedimentation Survey, Karangkates Reservoir, East Java, Indonesia. Field Data and Reservoir Survey, June 1982*. Hydraulics Research Report No. OD 50, Wallingford, UK.

Burton, A. & Bathurst, J.C. 1998. Physically based modelling of shallow landslide sediment yield at a catchment scale. *Environmental Geology* 35(2–3): 89–99.

Chanasyk, D.S., Mapfumo, E. & Willms, W. 2003. Quantification and simulation of surface runoff from fescue grassland watersheds. *Agricultural Water Management* 59: 137–153.

Dickinson, A. Amphlett, M.B. & Bolton, P. 1990. *Sediment Discharge Measurements Magat Catchment: Summary Report 1986–1988*. Hydraulics Research Report No. OD 122, Hydraulics Research, Wallingford, UK.

Doyle, M.W., Stanley, E.H. & Harbor, J.M. 2003. Channel adjustments following two dam removals in Wisconsin. *Water Resources Research* 39: 1011.

Elwell, H.A. 1980. *A Soil Loss Estimation Technique for Southern Africa*. Proceedings of Conservation '80, Silsoe, Bedford, UK.

Erskine, W.D., Mahmoudzadeh, A. & Myers, C. 2002. Land use effects on sediment yields and soil loss rates in small basins of Triassic sandstone near Sydney, NSW, Australia. *Catena* 49: 271–287.

Finlayson, D.P. & Montgomery, D.R. 2003. Modeling large-scale fluvial erosion in geographic information systems.*Geomorphology* 53: 147–164.

Fish, I.L. 1983. *Reservoir Sedimentation Study, Wlingi, East Java, Indonesia*. Hydraulics Research, Report No. OD 59, Wallingford, UK.

Fisher, K.R. 1993. *Manual of Sediment Transport in Rivers*. Report SR 359, Wallingford, UK.

Flanagan, D.C. & Nearing, M.A. (Eds.) 1995. *USDA Water Erosion Prediction Project. Hillslope Profile and Watershed Model Documentation*. USDA-ARS-NSERL Report # 10. West Lafayette, IN: NSERL.

Fournier, F. 1960. *Climat et erosion, la relation entre l'erosion du sol par l'eau et les precipitations atmospheriques.* Paris: Presses Universitaires de France.

Jain, S.K. 2001. Development of integrated sediment rating curves using ANNs. *Journal of Hydraulic Engineering* 127: 30–37.

Jain, S.K. & Chalisgaonkar, D. 2000. On setting up stage-discharge relations using ANN. *Journal of Hydrological Engineering* ACSE 5: 428–433.

Li, Y.T. & Gu, R.R. 2003 Modeling flow and sediment transport in a river system using an artificial neural network. *Environmental Management* 31: 122–134.

Martin, A.J. & Calvert, S.E. 2003. Hydrological and geochemical controls governing the distribution of trace metals in a mine-impacted lake. *Environmental Geology* 43: 408–418.

Martin, Y. & Church, M. 2000. Re-examination of Bagnold's empirical bedload formulae. *Earth Surf. Process. Landforms* 25: 1011–1024.

Melieres, M.A., Pourchet, M., Spadini, L. & Charlet, L. 2003. The large impact of gold-mining in total-Hg content in sediment of the dam lake of Petit Saut (French Guiana). *Journal de Physique IV* 107: 863–865.

Nagy, H.M., Watanabe, K. & Hirano, M. 2002. Prediction of sediment load concentration in rivers using artificial neural network model. *Journal of Hydraulic Engineering* 128: 588–595.

Patnaik, 1975. Soil erosion: A menace to the nation. *Indian Farming* 24(11), 7–10.

Renschler, C.S. 2003. Designing geo-spatial interfaces to scale process models: the GeoWEPP approach. *Hydrological Processes* 17: 1005–1017.

SWCS, 1995. *User Guide: Revised Universal Soil Loss Equation. Version 1.04*. Soil and Water Conservation Society, p. 145.

Tayfur, G. 2002. Artificial neural networks for sheet sediment transport. *Hydrological Sciences Journal* 47: 879–892.

Walling, D.E., He, Q. & Whelan, P.A. 2003. Using (CS)-C-137 measurements to validate the application of the AGNPS and ANSWERS erosion and sediment yield models in two small Devon catchments. *Soil & Tillage Research* 69: 27–43.

Wass, P.D. & Leeks, G.J.L. 1999. Suspended sediment fluxes in the Humber catchment, UK. *Hydrological Processes* 13(7): 935–953.

White, S.M. 1996. Comparison of runoff and sediment yield for two neighbouring catchments in the Philippines. *Cuaternario y Geomorfología* 10(2): 59–77.

White, S., Coe, R. & Nice, S. 1988. *Multiple Regression in Runoff and Erosion Studies: A Case Study from Malawi.* Hydraulics Research Report No. OD/TN 37, Wallingford, UK.

Wilby, R.L., Abrahart, R.J. & Dawson, C.W. 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrological Sciences Journal* 48(2): 163–181.

Williams, J.R. & Berndt, H.D. 1977. Sediment yield prediction based on watershed hydrology. *Transactions of ASAE* 20(6): 1100–1104.

Wischmeier, W.H. & Smith, D.D. 1978. Predicting rainfall erosion loss: a guide to conservation planning. In: *U.S. Department Agric. Washington D.C. Handb. No. 537*.

13

Nowcasting Products from Meteorological Satellite Imagery

GEORGE S. PANKIEWICZ
Satellite Applications Section, The Meteorological Office, Exeter, UK

ABSTRACT: Nowcasting and Numerical Weather Prediction (NWP) form the backbone of quantitative weather forecasting at the UK Meteorological Office (Met Office), covering very short to medium range forecasts (up to about 5 days ahead), and for horizontal scales from about 1 km to global. Due to the way in which observations are assimilated into NWP, very short range nowcasting methods have a significant impact on forecasts such as cloud and precipitation, and indeed are used for mesoscale model initialisation.

Meteorological satellites using visible, infrared and microwave radiometers provide a valuable source of information in support of both nowcasting and NWP. This chapter describes the use of meteorological satellite image data in nowcasting, and why neural networks have started to become an important technique for dealing with the complex patterns present in such data in near-real time operational forecasting applications.

Two such applications researched and developed at the Met Office, which use visible and infrared geostationary meteorological satellite imagery, are presented. The first concerns the discrimination of cloud type for determining which of two nowcasting techniques to use in a flood forecasting system. The second application is concerned with efforts to determine a probability of precipitation from visible and infrared imagery to improve precipitation forecasts from the British Isles precipitation radar network for use in the Met Office's Nimrod nowcasting system.

1 NOWCASTING

Nowcasting concerns very short range weather forecasting over horizontal scales of around 1 to 100 km. It tends to focus on the extrapolation of observed trends

in cloud, precipitation and similar meteorological variables, as opposed to the approach of NWP, which determines the atmospheric state and uses quasi-geostrophic dynamical equations to forecast future states from which distributions of cloud and precipitation can be inferred (Golding, 1998).

1.1 *Nowcasting techniques*

Many nowcasting techniques are available, such as extrapolation, advection, conceptual models, statistical techniques, expert systems and decisions trees, as well as the use of 1-dimensional and NWP models (Conway, 1992; Conway, 1999).

The original nowcasting method is simple linear extrapolation based on the assumption that the rate of change is constant. Persistence, for example, is a special example of linear extrapolation in which the rate of change is zero. Advection is an alternative to linear extrapolation, which can allow for curved motion to be forecast. Typically, the components of advection are derived from NWP forecasts.

Conceptual models embody some qualitative or semi-quantitative description of meteorological phenomena, such as their structure, life-cycles and mechanisms. Examples include the development of mid-latitude depressions (cyclogenesis), and the concept of a warm conveyor belt of moist air in so-called 'Spanish plume' events.

Statistical techniques are often used in situations where the physical basis of a relationship is poorly understood, or there is some complex nonlinear relationship, for example between observations and atmospheric variables, and where sufficient data exists to help define that relationship. Although regression techniques have been used for many years, it is no surprise that artificial neural networks (NN) started to take their place amongst this set of tools in the early 1990s, precisely for those reasons mentioned, but not least because nowcasting products are often required in near-real time.

If a forecasting task can be reduced to a set of rules, then decision trees, which represent the rules in the form of a flow-chart, can be used in the process. 'Inference engines' can operate on sets of rules to reach conclusions, and these are the basis of expert systems.

Perhaps the dominant forecast method in the medium-range since the 1960s has been the NWP model, in which a 3-dimensional representation of the atmosphere is contained, either in a grid-point or in a spectral space. Such numerical models can be global in nature, or can represent a limited area with finer horizontal and vertical resolution. Some numerical modelling centres, such as the Met Office, run both global and local models, so that boundary conditions from the global model can be used to initialize values at the edges of local models. The models integrate the so-called primitive equations which represent Newton's second law of motion, the ideal gas equation, the first law of thermodynamics, the conservation of mass and a pressure tendency equation, using the best 'analyzed' state of the atmosphere in terms of motion (winds), mass (pressure and temperature) and moisture (humidity).

Although NN have been suggested as a forecasting alternative to NWP models, it is very unlikely that they would be used, because the physical basis of these models at synoptic scales is well understood. The main problem with NWP models is one of gathering sufficient observational data, assimilating it, and using the finest horizontal and vertical resolutions possible, the last two factors being largely dependent on the computing power available. (The Met Office currently uses a Cray T3E supercomputer to run its Unified Model (Cullen, 1993), which caters for medium-range forecasting, climate and ensemble prediction applications at global- and meso- scales, all in one operational suite.)

1.2 *Nowcasting versus NWP?*

Nowcasting systems rely heavily on remotely sensed observations of atmospheric variables such as cloud and precipitation, and usually contain relatively simple assumptions for extrapolating them. NWP also relies on remotely sensed observations to determine the state of the atmosphere, but because of the relative lack of observations (10^5 for example in a global NWP model) compared to the number of grids points used (typically 10^7 in such models), there are many more degrees of freedom than the observational information can provide, and the problem is said to be 'under-determined'.

NWP therefore relies heavily on 'first-guess' fields: on forecasts produced from a previous forecast run. Data assimilation techniques are then used to weigh up the information from observations against the information in the first-guess to produce the best idea of the current state of the atmosphere, the analysis. Assimilation requires a large number of observations (often from a 6 hour period centred on the analysis, because of the asynoptic nature of some data such as weather radar, satellite retrievals and lightning detection), and together with the effects of limited resolution, a significant amount of time is required for the observations to affect the model forecast. In other words, the information input to NWP models is relatively slow. This means that NWP models tend to produce poor forecasts of variables such as cloud and precipitation at short forecast lead times, but because they have a good representation of the atmospheric dynamics, produce better forecasts in the medium range (3 to 48 hours).

In addition to the resolution problems, current NWP models suffer from timeliness and quality of very short range forecasts. So does this mean that nowcasting should rely on observational data alone and forecasting techniques such as advection, statistical analysis, conceptual models and expert systems?

1.3 *Sources of data*

Nowcasting is inextricably linked to remotely sensed observations and near-real time products that can be obtained from platforms such as weather radar and geostationary meteorological satellites. In answer to the last question, however, the truth is that more and more nowcasting systems are making use of first-guess and forecast data from NWP models, so we should not refer to 'Nowcasting versus NWP', but rather to 'Nowcasting using NWP'. Indeed,

modern nowcasting systems use as many techniques that are required, and as much observational and model data to provide significant information for the required forecast variable.

Consider the case of the Met Office's Nimrod nowcasting system (Golding, 1998), in which precipitation forecasts use:

1. radar and satellite observations to generate frequent rain rate analyses
2. linear extrapolation, or advection from NWP forecast winds, depending on recent quality levels, to advect the precipitation
3. relative use of observational data and NWP precipitation forecasts, where the latter dominates towards the end of the nowcast, as the relevance of the observational data declines.

Nowcasting data sources therefore not only include weather radar network data, geostationary satellite imagery, synoptic reports, but indirectly, all the observations that are used in NWP models: data from radiosondes, drifting and stationary buoys, aircraft reports, as well as top-of-the-atmosphere infrared and microwave radiances, scatterometer winds and atmospheric motion vectors derived from meteorological satellites.

2 THE USE OF METEOROLOGICAL SATELLITE IMAGERY IN NOWCASTING

At the Met Office, meteorological satellite observations have been used to improve weather forecasts in a number of ways. The classic approach was to provide imagery products for bench forecasters, such as visible and infrared imagery, and information on cloud top height and the presence of fog. These products were used to aid decision making for a range of tasks, from local forecasting to the validation of NWP models.

In recent years, the increase in speed and memory of computers has helped to process satellite data in near-real time, so that a much larger array of meteorological products can be considered. These include temperature and humidity retrievals from infrared and microwave sounding instruments and atmospheric motion vectors obtained from geostationary visible, infrared and water vapour imagery. A large number of additional meteorological products are expected to be obtained from new satellite instruments. Meteosat Second Generation produces images every 15 minutes with 3 km sub-satellite resolution for 12 visible and infrared channels, and products include high-resolution winds, convective rainfall rate, stability analysis imagery and information on rapidly developing thunderstorms.

2.1 *Visible, infrared and microwave imagery*

Satellite sensor measurements can be broadly classified according to their use in nowcasting or NWP. The domains of nowcasting and NWP have already been discussed, and it is no surprise, therefore, that nowcasting requires near-real time

mesoscale products, usually related to specific meteorological variables such as cloud and precipitation, whilst NWP requires global data, usually related to the atmospheric state such as temperature and humidity soundings and wind-speed. An important difference is the platform used to get the meteorological variables of interest. On balance, polar orbiters tend to provide data for NWP, and geostationaries the data for nowcasting.

Geostationary meteorological satellites nearly always have one visible and one infrared channel (at about 0.7 μm and 11.0 μm respectively), and occasionally a channel centred on the 6.7 μm water vapour band. From the radiance observed in the visible channel, a measure of the visible reflectivity is obtained, which is high for opaque cloud, and low for warm land and sea surfaces. Infrared channel radiances are used to determine an infrared brightness temperature of a given pixel. For example, thick cirrus cloud at high altitudes can result in a cloud-top brightness temperature of around 220 K, whilst land surfaces in summer can produce brightness temperatures in excess of 300 K. Figure 13.1 shows visible and infrared images of the UK on Christmas day 1995, together with the corresponding values for selected pixels, in a 2-dimensional 'feature space'. Note that different cloud and surface classes occupy different regions of the space: thick cirrus clouds (diamonds) have reflectivities of 55–90% and brightness temperatures of 215–230 K, whilst clear sea (crosses) occupies a smaller cluster with reflectivities of 10–20% and brightness temperatures of 275–285 K.

As well as images at visible and infrared wavelengths, some radiometers have been flown which operate in the microwave (for example, the Special Sensor Microwave Imager on the US Defense Meteorological Satellite Program satellites). These passive microwave imagers operate at wavelengths of several millimetres, although channels are usually expressed in terms of frequency, with SSM/I for example having channels at 19, 22, 37 and 85 GHz.

2.2 *Cloud*

Figure 13.1 illustrates how visible and infrared satellite imagery can be used as a source of information on cloud amount and type. Kidder and Vonder Haar (1995) provide an excellent review of cloud products from meteorological satellite imagery, and Pankiewicz (1995) discusses the use of pattern recognition techniques for such products.

The simplest and oldest technique for discriminating cloud from cloud-free areas is to use a threshold in both visible and infrared channels (Arking, 1964), such that if a pixel is brighter or colder than some threshold value, it is assumed to be cloud covered. Two fundamental problems with this technique, however, are the choice of threshold and the effects of sub-pixel cloud. An alternative is to use histogram techniques, in which histogram contours are plotted in a space similar to that shown in Figure 13.1. In one approach (Desbois *et al.*, 1982), clusters of classes are determined by first identifying local maxima, then assigning samples to the nearest maxima to form growing clusters.

Spatial coherence (Coakley & Bretherton, 1982) relies on parts of the infrared brightness temperature image being either completely clear or completely cloudy.

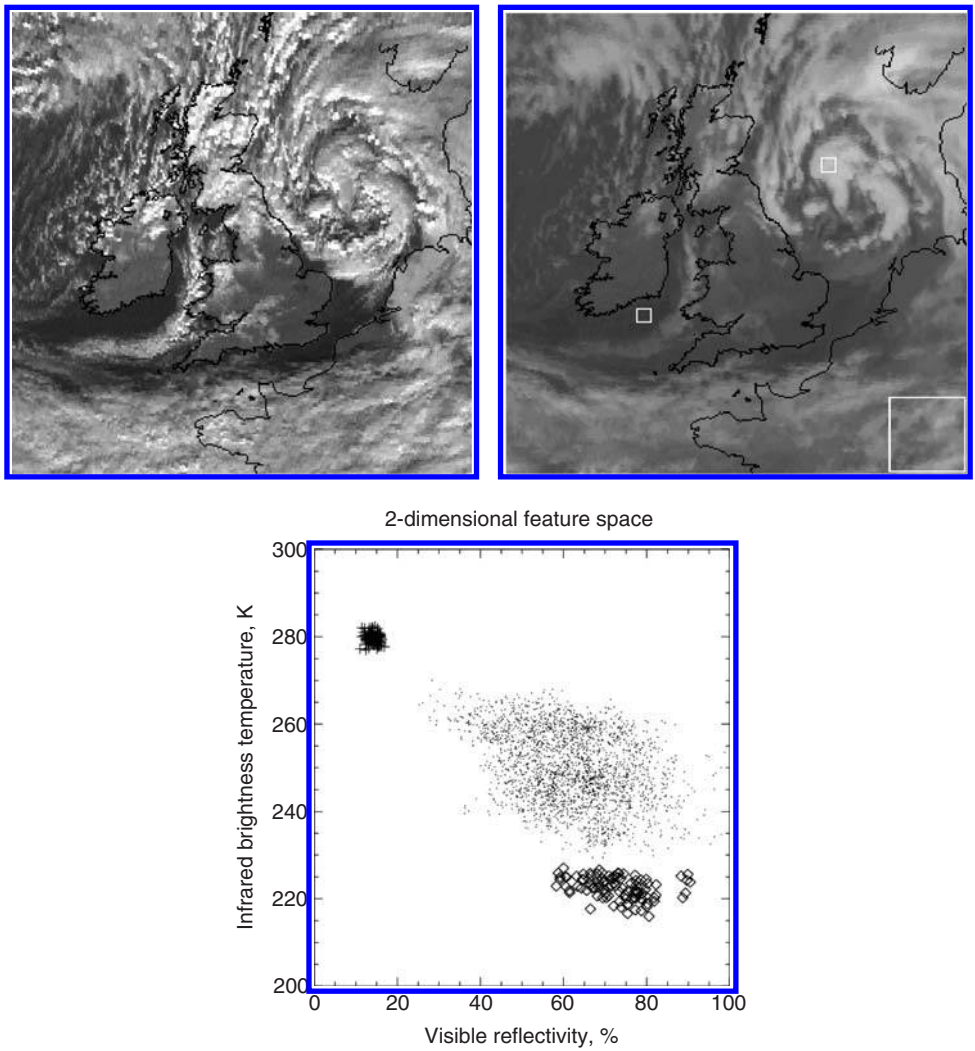


Fig. 13.1. Meteosat visible and infrared images taken at 11:30UTC on 25th December 1995, processed at the Met Office to produce visible reflectivity in the range 0–100%, and infrared brightness temperature in the range 193–303 K, both represented here in grey-levels from 0–255. A 2-dimensional feature space shows samples of data taken from 2500 pixels in the bottom right hand corner (mid-level cloud, dots), 100 pixels in the centre of the North Sea depression (thick cirrus, diamonds), and 100 pixels just south of Ireland (clear sea, crosses).

The mean and standard deviations of infrared radiances are calculated and plotted against each other. Areas free of cloud have low standard deviation and low mean radiance, whilst areas filled with cloud also have low standard deviation but high mean radiance. Partly cloudy areas have high standard deviation and an intermediate mean radiance. The radiance values for clear and cloudy areas

are then determined, and knowing the actual radiance, the effective cloud amount can be calculated for each pixel.

To some extent, pattern recognition techniques take the concept of thresholding, histogramming and spatial coherence, and use all of this information together to determine cloud properties in imagery. A multi-dimensional space is constructed, the most basic example being where each radiance measurement relates to one axis (as in Fig. 13.1). However, other features such as standard deviation and textural measures (Haralick, 1986) of a small group of pixels can be used to provide additional information. As in histogramming, samples are used to define clusters of classes. However, the idea is then to discriminate these classes so that any set of features values can be identified with that class again. Over the past 20 years or more, a number of cloud type classifiers have been constructed using the elements of pattern recognition (Parikh, 1997 provided an early review), using parallelepipeds in a 7-dimensional feature space (Karlsson, 1994) and Gaussian maximum likelihood classification (Ebert, 1987; Garand, 1988).

2.3 *Precipitation*

The estimation of precipitation using satellite imagery is 'one of the most difficult and unsolved problems facing the science and technology of satellite remote sensing' (Levizzani, 1999). Reviews of visible, infrared and microwave techniques have been produced by Barrett and Martin (1981) and Kidder and Vonder Haar (1995), who broadly classify visible and infrared techniques into cloud indexing, bi-spectral techniques, life history methods and the use of cloud models. With any visible or infrared technique, precipitation rates are inferred indirectly: the precipitation falls from the cloud base, but visible and infrared radiances are observed at the top of the atmosphere. Hence no cloud means there is no precipitation, but cloudy regions may also produce no precipitation, and clouds in the tropics produce precipitation in quite a different way to mid-latitude frontal bands. In addition, verification of precipitation is difficult. Data from radar networks are used, but come with their own sources of uncertainty. The representativity of rain gauges makes their use with satellite data a problem. Since radars sample volumes comparable to satellite pixels, radar networks are probably the best source of verification data.

The oldest technique of cloud indexing aims to identify cloud types in satellite imagery, after which rain rates can be assigned to the different cloud types, e.g. the widely used Global Precipitation Index method, (Arkin & Meisner, 1987).

Bi-spectral methods (Lovejoy & Austin, 1979) attempt to combine the rules of probabilities of precipitation associated with both visible and infrared channels, so that clouds which have the best chance of raining must be both cold and bright. Currently, the Met Office's Nimrod nowcasting system uses this type of method to determine a probability of precipitation from Meteosat (Cheng & Brown, 1995).

For convective clouds, life-history techniques are particularly relevant, since there is a clear relationship between the rate of change of a cirrus anvil

associated with cumulonimbus clouds and the resulting precipitation. One of the most widely used techniques is known as the Griffith-Woodley technique (Griffith *et al.*, 1978), which identifies cloud colder than 253 K in a series of images. The maximum areal extent is obtained, and the inferred rain rates are determined according to the stage of cloud growth, the cold cloud area and the cloud-top temperature within the cold cloud area.

In principle, pattern recognition techniques can combine cloud type, bi-spectral and life history information, if sufficiently accurate and comprehensive training sets are available. Wu *et al.* (1985) used the Gaussian maximum likelihood method to classify samples measuring 400 km^2 taken from GOES imagery. To train the classifier, rain rates were inferred from collocated National Weather Service radar data. More recently, Grassotti and Garand (1994) used cloud-top pressure, albedo and cloud fraction as input features to an unsupervised classifier, as well as 6-hourly estimated NWP model rain rates.

2.4 *Neural network approaches to cloud and precipitation estimation*

Visible and infrared cloud and precipitation nowcasting products are ideal candidates for retrieval by pattern recognition techniques, and in particular, multilayer perceptron neural networks (MLP) using the backpropagation learning algorithm (Rumelhart *et al.*, 1986). They both require two aspects of pattern recognition: automatic recognition of the structures and shapes observed in cloud (machine vision), and automatic recognition of rain rates or cloud types in terms of spectral and textural data, as well as ancillary data, such as latitude and hour of the day (statistical pattern recognition). The reasons that MLP have been considered useful for this work is that:

1. Apparently unrelated input features (such as average values, grey-level difference vectors, hour of the day and latitude) can all be combined.
2. Complex nonlinear relationships between meteorological variables (such as cloud structure and rain rate for example) can be learned.
3. Preconceived ideas of class distributions (such as Gaussians) are not assumed – NN training is 'data driven'.
4. Operational use of trained NN is very fast, important in near-real time applications where large datasets (such as images) are processed.
5. In classification mode (with two or more output nodes), estimates are Bayesian in the sense that a mixture of information is provided to the trained NN, and the outputs represent the likelihood of those classes existing. (This means that for all output values less than 0.1, for example, an extra 'don't know' class can be ascribed.)
6. Adaptability to loss of data ('graceful degradation') is possible, which can be of particular use in meteorological satellite imagery applications, where visible channel data effectively disappears at night.

There are naturally a number of drawbacks to these techniques, including the amount and quality of data used in training (robustness), the correct representation of all likely situations in the training set (extreme events), and the fact

that to some extent, the trained network is a black box (although the strengths of connections between input and output nodes can indicate the relative importance of certain relationships). However, given a sufficiently good training set, these problems can be largely overcome, resulting in good candidates for operational applications.

3 CLOUD CLASSIFICATION FOR A FLOOD FORECASTING SYSTEM

One key problem related to very short range forecasting in mid-latitudes concerns the different types of weather that can result from conditions controlled by synoptic-scale advection as opposed to conditions controlled by mesoscale convection. In advective situations, frontal and stratiform cloud types are more dominant, and cloud and rain tend to be advected according to steering level winds.

On the other hand, convection is often governed by small, local instabilities in the lowest few kilometres of the atmosphere, which are able to trigger the release of large quantities of Convectively Available Potential Energy (CAPE) over a short period of time. Once this CAPE has been released, development of convective cells and storms can be extremely fast: a mature cumulonimbus storm can develop from a cumulus cloud in as little as 30 minutes. The weather associated with severe convection can include a number of violent phenomena: heavy precipitation (rain rates of more than $30\,\mathrm{mmh}^{-1}$ are not uncommon over a period of a few minutes), strong gusts, downdraughts (important to aviation), not to mention lightning, hail and tornadoes.

Forecasting in such convective situations can be improved with an understanding of the life-cycle of convective cells, as well as knowing the recent history of those cells. This is because convective cells tend to undergo stages of evolution, dissipation, splitting and merging, even to the extent of producing daughter cells. It would therefore appear to be of interest to not just describe cloud according to the well-known classes such as altocumulus, cirrostratus and stratocumulus, but into more physically related classes including stratiform cloud, and shallow and deep cumuliform (convective) cloud.

This section discusses the use a NN cloud classifier trained to distinguish just these classes, for use in a flood forecasting system which is required to use advective or convective forecasting techniques, appropriately.

3.1 *Overview of the GANDOLF system*

GANDOLF (an acronym for Generating Advanced Nowcasts for Deployment in Operational Land-based Flood forecasts) is a system designed to provide automated precipitation forecasts to the Environment Agency, and to issue warnings of likely accumulations in regions of interest (Collier *et al.*, 1995; Pierce *et al.*, 1995). Many of the sources of error in flood forecasting often occur during periods of heavy convective precipitation, because the exact timing, location and intensity of convective activity is difficult to assess, and many

nowcasting systems currently rely on extrapolative and advective techniques. However, a promising new technique which uses an object-oriented conceptual model (OOM) of convective cells (Hand & Conway, 1995) began a series of tests at the Met Office in 1994, and was more recently trialled in a pre-operational version of GANDOLF during the summer months of 1996 to 1998 in collaboration with the Environment Agency.

There are three levels of operation of GANDOLF:

- Level 1 (Monitor) co-ordinates the gathering of remote data, primarily from the Nimrod nowcasting system. Processed Meteosat data are used at this stage to determine the overall synoptic conditions in terms of advection or convection, and a decision is made to use Nimrod nowcasts or to start the OOM. If precipitation is detected within the GANDOLF domain, and specifically within the Chenies radar domain (an area of about 420 km × 420 km centred to the north west of London), the second level of operation is started.
- Level 2 (Action) is concerned with running and validating the OOM. If the OOM forecasts convective precipitation in the Thames Domain (NG 380–580 km E; 120–280 km N), GANDOLF initiates the third and highest level of operation.
- Level 3 (Alert) is the third and highest level of operation. The latest OOM precipitation forecast is sent to the Flood Warning Centre. Verification statistics are also sent when observed 10 minute instantaneous rain rate and 15 minute accumulations are available. Level 3 issues heavy precipitation warning messages if such precipitation is forecast.

3.2 *Cloud classifier training and testing*

Some automatic technique for distinguishing stratiform cloud from convective cloud was therefore required for level 1 of the GANDOLF system. A good candidate method seemed to be the use of cloud classification from satellite imagery, available in near-real time, for the whole of the GANDOLF domain.

3.2.1 *Early results*

In 1996, a MLP was trained using Meteosat visible and infrared imagery, for use in the pre-operational version of GANDOLF (Pankiewicz, 1997). A total of 365 Meteosat visible and infrared images was restored from a satellite image archive from 1994, one image pair being selected for every day of the year, at some randomly chosen time between dawn and dusk. The area covered by the imagery included northwest Europe, the North Sea and part of the Atlantic Ocean east of 20°W, roughly from 45°N to 65°N.

For each image pair, a total of six samples of 17 × 17 pixels were selected by an experienced meteorologist at the Met Office, using surface analyses to verify airmasses and cloud types. This resulted in 2190 samples (over 6×10^5 pixels), which were labelled into one of: clear land or sea, stratiform cloud (including frontal cirrus, thin cirrus, altostratus, nimbostratus, stratus, fog and haar), shallow convective cloud (including stratocumulus and cumulus), and

deep convection (cumulonimbus and mesoscale convective systems). The number of samples in each class was not the same, but was a measure of the *a priori* distributions of the classes encountered in the imagery, which can be an important factor in terms of classification accuracy for such NN (Foody *et al.*, 1995). The samples were finally split randomly into a training set of two-thirds and a test set of one-third of the samples.

Feature selection at this stage was performed by a literature search of proven cloud classification features (Gu *et al.*, 1991; MacLaren *et al.*, 1994; Bankert 1994). A more objective method is discussed in Section 4.2. Simple features were used including the minimum, maximum, standard deviation and mean Robert's gradient in the visible, the maximum, ratio of minimum to maximum and standard deviation in the infrared, and the month of the year and hour of the day of the sample (9 inputs plus a threshold node). All features were normalized linearly in the range 0 to 1, especially as few feature values were found to occur near the limits. Output class values were set to 0 or 1 for each sample with the true class being represented by 1.

One hidden layer was deemed sufficient for the network, and after numerous trials, the fewest misclassifications were obtained for a 10:12:4 network architecture. The network used backpropagation with a momentum term added. The number of samples used in the training set was found to improve classification accuracy if the ratio was approximately 1:4:3:1 for clear, stratiform, shallow and deep convective cloud, respectively. A deliberately large learning rate of 0.9 was set to provide a gradient descent that would cover a large portion of the error surface. Changes in the mean square error were checked every five epochs, and training was stopped if the error was less than 10^{-5} for 10 consecutive epochs, or when a maximum number of 200 epochs had passed. If an apparently good solution was obtained, classification accuracies were calculated 10 times on randomly chosen subsets of the test set, since a large standard deviation in classification accuracy implies that the network became stuck in a local minimum (MacLaren *et al.*, 1994). The best classification accuracy obtained in this way was $80.8 \pm 2.2\%$.

Table 13.1 below shows a typical confusion matrix for this network, in which the number of true samples used are compared against the number estimated by

Table 13.1. A typical confusion matrix for the 1996 version of the GANDOLF cloud classifier, with a classification accuracy of 81.0%. True class numbers are given in the rows (50 per class), and NN estimates in the columns.

	Estimated clear	Estimated stratiform	Estimated shallow convection	Estimated deep convection
True clear	49	1	0	0
True stratiform	7	30	8	5
True shallow convection	0	2	42	6
True deep convection	1	4	4	41

the NN. Ideally, the matrix should be diagonal, but in practice, non-zero off-diagonal elements exist, which indicate where the worst misclassifications occur. The main source of error was in identifying stratiform cloud. For example, some cases of thin cirrus were misclassified as clear and some cases of broken stratus were misclassified as shallow convection.

3.2.2 *Night-time, dawn and dusk*

One of the problems with the 1996 version of the network was that it relied on visible imagery, and could not be used at night. To improve the classifier, 100 cases of Meteosat visible and infrared imagery were restored from archive (a new training set), spread over the period July 1995 to December 1997. Half of the cases were chosen randomly, whilst the remainder were chosen to contain examples of particular meteorological situations that might be missed in the randomly chosen set (such as polar Arctic and polar maritime showers, Spanish plume events, deep convection in a polar maritime returning flow, anticyclonic clear situations and so on).

Some 3000 samples (nearly 9×10^5 pixels) at 5 km resolution were selected and were once again labelled according to clear, stratiform cloud, or shallow or deep convective cloud. Met Office analyses were used to ensure the synoptic situation warranted the class chosen, and deep convective cloud was discriminated from shallow convective cloud if cloud top temperatures were less than $-15\,^{\circ}\mathrm{C}$, warmer than the corresponding threshold used during the initial GANDOLF classifier training in 1996. The final distribution of samples is given in Table 13.2.

Initially, the plan was to provide two cloud classifiers, one for visible and infrared data, the other for infrared only. However, some experiments were performed with a single cloud classifier, that could adapt to loss of visible data at night. It became obvious after some time that two classifiers tended to show significant mismatches at dawn and dusk when GANDOLF switched from one to the other. NN are, however, claimed to be good at coping with the loss of input data (graceful degradation), so an experiment was tried in which missing visible channel input values were set to 256/255 (1.0039; the normal range is 0.0 to 1.0). This appeared to work very well indeed, providing a network with an overall (day and night) classification accuracy of 92%. Nine input features were used for this network, the same as those used for the 1996 GANDOLF classifier.

Table 13.2. Distribution of training and testing samples by class for the 1999 version of the GANDOLF cloud classifier.

	Clear	Stratiform	Shallow convection	Deep convection
Visible and infrared	189	789	99	298
Infrared only	274	1190	14	147
Total	463	1979	113	445

The few remaining problems were now concerned with the classification of deep convective cloud at night, and the changeover from the 23:30 image to the 00:00 image (due to the training data being non-cyclic). To avoid these problems, some modifications were made, including the addition of the mean Robert's gradient as an input feature in the infrared, which showed an ability to discriminate deep convective cloud and stratiform cloud, and a change from using the minimum visible and infrared values to the mean visible and infrared values, which was found to better discriminate shallow convection. Additionally, the hour of the day and month of the year features in the training set were randomly split into two, with one set having 24 hours (or 1 year) added if the value was less than 12:00 (or before July), or subtracted if it was greater than 11:59 (or after June), to provide a training set covering 48 hours or 2 years for each feature. This enabled the network to perform the same fit to the data at the start and end of these features, thereby improving the 23:30 to 00:00 and December to January matches. Table 13.3 below shows an example of the network classification accuracy for the NN delivered for operational use within GANDOLF, for a typical distribution of samples.

Figure 13.2 gives an example of the 1999 GANDOLF NN cloud classifier, at dusk at 19:00 on the 6th June 1999, when visible data were starting to be lost. Stratiform cloud is correctly identified in the infrared section, but convective cloud remains near the coast of southern France. This would have resulted in the object oriented forecast model of convective cells being used, since most the south of the UK is covered with deep convective cloud.

In practical use, the classifier, although trained on samples of 17×17 pixels, was used to calculate features in regions of 7×7 pixels, and was used to classify tiles of 3×3 pixels (15×15 km). This loses little information in terms of classification accuracy, except that the centres of deep convective cells tend to become classified as stratiform cloud. This is strictly true given that they consist of cirrus anvils. The advantages of this procedure are that edge effects are improved (tiles of 17×17 pixels may classify everything as stratiform cloud if the region contains clear sea and stratiform cloud), the image is not too noisy

Table 13.3. A confusion matrix for the 1999 version of the GANDOLF cloud classifier, with a classification accuracy of 92.7% for a typical distribution of samples. True class numbers are given in the rows, and NN estimates are given in the columns.

	Estimated clear	Estimated stratiform	Estimated shallow convection	Estimated deep convection
True clear	60	0	0	0
True stratiform	1	256	0	0
True shallow convection	1	6	30	1
True deep convection	0	9	15	73

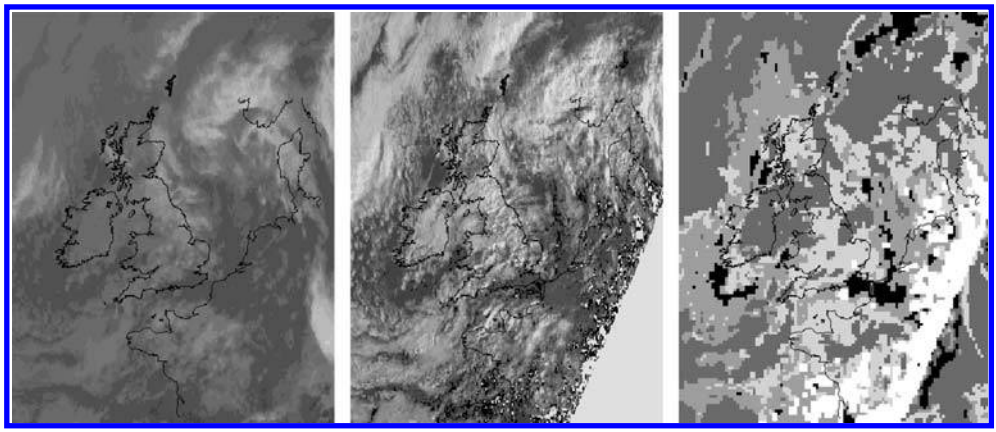


Fig. 13.2. GANDOLF NN cloud classification at dusk, 19:00UTC on 6th June 1999. The left panel shows the Meteosat parallax corrected infrared image for the Nimrod area at 5 km resolution, the centre panel the Meteosat visible image (note the effect of dusk), and the right panel shows the classified image (black is clear, dark grey is stratiform cloud, mid-grey is shallow convective cloud, light grey is deep convective cloud and white is unclassified).

(which occurs if every pixel is classified), and the time required to perform the classification is about 32 times quicker.

4 PRECIPITATION ESTIMATION FOR THE NIMROD NOWCASTING SYSTEM

Spurious echoes in the form of anomalous propagation (anaprop) and clutter can provide fairly common, yet unwanted signals in weather radar data, resulting in false observations of precipitation. The Nimrod nowcasting system produces a variety of cloud and precipitation fields which are also used in initialising the mesoscale model. The quality of radar data therefore has an important consequence for both short-range forecast guidance and mesoscale model initialisation.

Currently, the Nimrod nowcasting system attempts to detect spurious radar echoes automatically using a scheme that includes a Probability of Precipitation (PoP) field determined from two independent sources: ground-based synoptic data and Meteosat visible and infrared imagery. The Meteosat data are analysed on a pixel-by-pixel basis to help form the PoP field by using a simple thresholding technique, described by Cheng *et al.* (1993), and based on the method of Lovejoy and Austin (1979). Essentially, cloud-free areas result in a low PoP, whilst in cloudy areas, PoP is estimated according to the climatological occurrence obtained by Cheng *et al.* (1993). The scheme is often referred to as the Nimrod anaprop removal scheme, although the aim is to remove all spurious echoes. This Meteosat PoP is then combined with the ground-based synoptic reports and forecast PoP using Bayes' theorem (Pamment & Conway, 1998), and results in a value of a parameter known as alpha, which can range

from 0 (definitely no rain) through 1 (the climatological probability of rain) to large values (definitely rain). A threshold is set within the field of alphas to remove radar echoes where there is a sufficiently low probability of rain.

Although the current Meteosat thresholding scheme provides a Probability of Detection of about 60% for cold frontal precipitation where rain rates are greater than $1/32\,\mathrm{mmh}^{-1}$, it is about 50% for warm fronts, and only 30% for cold-air convection.

A study was undertaken to estimate precipitation rate in 4 classes from NOAA AVHRR data using a NN classifier, which provided a new capability of incorporating spectral and textural image characteristics. The results were encouraging, with an average PoP of 72% at a threshold of $1/8\,\mathrm{mmh}^{-1}$, taken in various synoptic conditions, compared to similar mesoscale model PoP values of 69% at T + 0 and 63% at T + 6. False Alarm Rates (FAR) were 37% from the NN, compared to 51% from the mesoscale model at T + 0 and 56% at T + 6.

The combination of different discriminatory inputs such as infrared brightness temperature and visible reflectivity texture meant that local neighbourhood information around the pixel of interest could be used to improve the estimate of precipitation, over and above that of a threshold technique (for example in cases of cold-air convection, by recognizing convective cells). The NN approach used in this case had further advantages in that its output values provided Bayesian estimates of the PoP directly, and that it could operate at high speed, because of the statistical nature of the trained NN.

4.1 *Training and validation data*

At the Met Office, Meteosat image data are processed through the Autosat system, which reprojects images into polar stereographic coordinates, and derives a range of products, including visible reflectivity and infrared brightness temperatures, from the instrument counts. Infrared grey-scales range from 4 to 251, and represent brightness temperatures of 198 K to 308 K respectively. The Nimrod nowcasting system ingests these two Meteosat products, corrects them for parallax and sun angle and maps them onto a 5 km grid within the Nimrod domain (Golding, 1998).

Each image underwent pre-processing including identification and removal of corrupt images and was then composited into the Nimrod domain at 5 km resolution.

For the development of the PoP classifier, we used 48 sets of Meteosat and radar composite images from July 1995 to June 1997 (including visible reflectivity data when available). Two sets were restored for each month at random times, coinciding with half hourly Meteosat imagery.

For each radar composite, two additional composites were chosen before and after the time of interest, to form a radar movieloop with 5 half-hourly frames. These movieloops were examined for anaprop, often seen best in this way as patches of shimmering radar echoes bearing little relation to real rain systems and often related to orography. Areas suspected of being contaminated with

anaprop were marked on a copy of each image, to avoid selection for the NN training set.

A total of 3200 samples of 17 × 17 pixels (85 × 85 km) were selected from the collocated infrared brightness temperature and radar rain rate fields from uncontaminated regions of the images. Samples were labelled as no rain ($\leqslant 1/32\,\mathrm{mmh}^{-1}$) or as rain ($> 1/32\,\mathrm{mmh}^{-1}$), depending on the central radar rain rate (centre labelling) or the average radar rain rate over the sample (average labelling).

Average labelling is less sensitive, but provides a better correlation between cloud brightness temperature statistics and rain or no rain. Any subsample could therefore be chosen, with its own centre or average label, if required. Of the 3200 samples, 2097 had no rain as the central label and 1103 had rain, giving a dry to wet ratio of 1:1.9. This compares to the dry to wet ratio used by Cheng *et al.* (1993) of 1:2.8.

From this set of 3200 samples, 1142 were selected for which uncorrupted visible reflectivity data were known to be available. These data were restored from archive, and added to the training and validation sets. Of the 1142 samples, 749 had no rain as the central label and 393 had rain, giving the same dry to wet ratio as for the infrared only cases.

4.2 *Feature selection for the PoP classifier*

Given samples of infrared brightness temperature or visible reflectivity pixels which we wish to correlate with a central or average rain rate label, it is possible to calculate numerous statistics or features over different sized regions. The most obvious features are the infrared brightness temperature or visible reflectivity values at the central pixel; given this information, a NN classifier should be able to classify as well as, or better than the threshold classifier discussed in Cheng *et al.* (1993), for example. However, it is possible to extract some simple local features, such as the mean, standard deviation, minimum, maximum, and the range and ratio of the maximum and minimum, providing a total of 7 features per spectral channel, including the central value.

It is also possible to extract textural features from regions of at least 3 × 3 pixels, such as grey-level difference vectors (Weszka *et al.*, 1976). A grey-level difference vector $\Delta g(\phi)$ is the absolute difference between two grey-levels with a fixed spatial relationship ϕ of angle, and distance (in pixels). In this work, four relationships were used: (0°,1.00), (45°,1.41), (90°,1.00) and (135°, 1.41). Grey-level differences were calculated for every pair of pixels in the sample governed by these relationships, to produce a series of histograms $h_\phi(\Delta g)$. These histograms were then used to construct five statistics of interest: the mean, contrast, angular second moment, entropy and homogeneity, as detailed in Pankiewicz (1994).

To determine the ability of these 17 candidate features per spectral channel to discriminate rain from no rain, multi-dimensional Bhattacharyya distances were calculated (see Gu *et al.* (1991), for example). Bhattacharyya distances were calculated for sample sizes of of 7 × 7 pixels, using centre labelling for

cases where both visible reflectivity and infrared brightness temperature data were available. In this case, the largest Bhattacharyya distances were found to continue increasing at a feature vector size of 9 dimensions, with textural features starting to be used at 9 dimensions.

The largest Bhattacharyya distance of a feature vector without textural features was found to consist of the minimum, maximum, range and ratio of the visible reflectivity, plus the minimum, maximum, range and ratio of the the infrared brightness temperature (8 dimensions). At 9 dimensions, the largest Bhattacharyya distance included the same features as at 8, together with the maximum visible reflectivity entropy measure of texture.

4.3 *PoP training and testing*

Feature selection showed that 5 or 8 features depending on availability of visible reflectivity data are most useful at discriminating 2 rain classes (no rain and rain with a threshold of 1/32 mmh^{-1}). The 3200 training samples were split randomly into a training set and a validation set, containing two-thirds and one-third of the data respectively. A MLP was trained with the input features described above, together with a bias node. Weight values were recorded every 5 epochs, and the validation set was used to obtain the average Probabilities of Detection (POD) and False Alarm Rates (FAR). The aim was to increase the PoP and decrease the FAR. Note however that to get the same ratio of wet to dry pixels, we require FAR = 1-PoP. If FAR > 1-PoP, then the scheme overestimates the number of wet pixels. In tests, a 1:2.8 wet to dry distribution of samples was used in order to compare the results with the work of Cheng *et al.* (1993).

After a large number of experiments in which the number of hidden nodes was changed, as well as values of the learning rate and momentum factor, the best network was found to produce a PoP of 66 ± 9%, a FAR of 57 ± 5% and an Equitable Threat Score (ETS) of 17 ± 6%. These are average scores in the sense that the validation set was constructed out of samples taken during various synoptic conditions. An idea of the variances was obtained by calculating PoP, FAR and ETS for subgroups of samples from the total validation set. A total of 8 hidden nodes provided this solution, and a learning rate of values of 0.5 and momentum factor of 0.05 were used. The input features were normalized according to zero mean, 4σ variance, and all samples were 7×7 pixels with centre class labelling.

The results for the infrared only network can be compared with the best PoP of 61 ± 24% and FAR of 47 ± 12% for cold frontal cases obtained by Cheng *et al.* (1993), and their values of 28 ± 21% and 74 ± 11% respectively for cold-air convection, and 50 ± 19% and 50 ± 16% respectively for mesoscale convective systems.

An example of the resulting PoP where echoes were recorded in the Nimrod radar composite is shown in Figure 13.3 for 20:30UT on 15th May 1997, together with the Meteosat infrared image and the radar composite. The radar composite was obtained after processing with the Meteosat threshold classifier, and a large number of spurious echoes still remain over Ireland, Strathclyde,

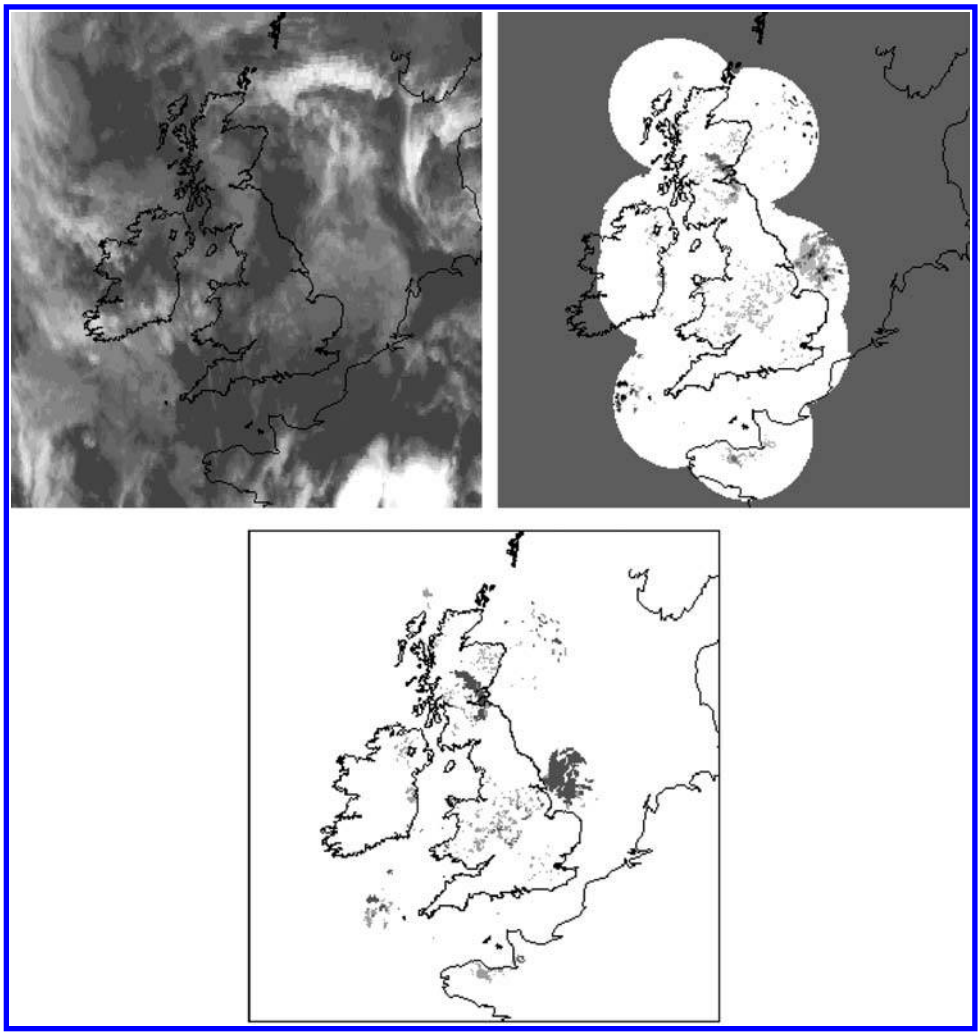


Fig. 13.3. An example of PoP values from the infrared feature NN: The top left panel shows the Meteosat infrared image for the Nimrod area at 5 km resolution, at 20:30UTC on 15th May 1997. The top right panel shows rain rates within the Nimrod radar composite area (white is no rain, light grey $>1/32\,\mathrm{mmh}^{-1}$, dark grey $>1/2\,\mathrm{mmh}^{-1}$ and black $2\,\mathrm{mmh}^{-1}$), and the lower panel shows the NN PoP field where radar echoes were recorded (dark grey is PoP >0.4, light grey is PoP $\leqslant 0.4$).

Grampian region, the Midlands, the southeast and the Normandy and Brittany coast. The NN PoP classifier has assigned PoP values of around 0.02–0.3 over Northern Ireland, Strathclyde and the Grampian region, 0.02–0.5 over the Midlands, 0.01–0.3 over the southeast and 0.02–0.3 over the Normandy and Brittany coast. However, PoP values of around 0.5 are found for the band of rain over the Firth of Forth, and 0.5–0.7 for the rain off the Lincolnshire coast.

For the visible and infrared feature network, the 1142 available samples were again split into a training set from two-thirds of the data, and a validation set from the remaining third. The same procedure was adopted to search for the best classifier, and in this case, the best network was found to produce a PoP of 69 ± 10%, a FAR of 41 ± 3%, and an ETS of 33 ± 7%, all superior to the infrared only PoP classifier. A total of 12 hidden nodes were required, together with a learning rate of 0.2 and a momentum factor of 0.01. All other NN aspects were the same as for the network using infrared features alone.

4.4 *Results of trialling the PoP classifier in Nimrod*

In the Nimrod anaprop removal scheme, PoP values are not used in the calculation of a final probability map. Alpha values are used instead, where in general:

$$\alpha = \frac{P(W)}{P(D)} \tag{1}$$

where P(W) is the probability of a pixel being wet and P(D) is the probability of a pixel being dry. An alpha value can be said to be the 'odds' that a pixel is actually wet and ranges from 0 (definitely no rain) through to 1 (the climatological probability of precipitation) to large values up to 100 (definitely rain). The use of alpha values in the Nimrod anaprop removal scheme is described by Pamment and Conway (1998). The NN PoP product is converted into a field of alphas before it is implemented into the current anaprop removal scheme.

The NN PoP field must be shown to provide a better diagnosis of precipitating and non-precipitating cloud than the current field (described by Cheng *et al.*, 1993), if it is to be used in the Nimrod anaprop removal scheme. The current PoP field diagnoses shallow and small-scale convection poorly and this is particularly evident when only infrared data are available: the NN classifier would be of particular value if it can show improvements in such conditions.

The classifier must identify areas of non-precipitating cloud in order that spurious echoes in radar images can be effectively removed. However, an overriding requirement of the anaprop removal scheme is that precipitation echoes must not be removed; therefore, the classifier must also accurately identify areas of precipitation.

The performance of the new classifier was evaluated during two periods, between 28th May and 8th September 1998, and 28th October and 18th December 1998.

During the initial period, the assessment was designed to measure the performance of the NN PoP classifier, without surface reports or short period forecasts. The PoP from the classifier was converted into a field of alphas (NN alphas) which could be directly compared to the Nimrod Meteosat field of alphas (Nimrod alphas) of the current anaprop removal scheme. In the current scheme, an alpha value of 1 is the threshold with values below 1 assumed to indicate dry conditions and any echoes falling in these regions being deleted. Alpha values of 1 or greater are assumed to indicate a climatological probability of

rain and any echoes falling in these regions are retained. A set of linear transforms were used to convert the NN PoP into alpha values. These transforms were derived by examining a number of radar composites and finding a transformation which produced a balance of alpha values for wet and dry radar pixels in a similar way to the current Nimrod alphas field. The threshold at which echoes would be deleted or retained was derived by finding a value which produced a balance of anaprop removal and rain retention similar to the current Nimrod alphas. The same linear transforms were applied to the NN PoP field, regardless of whether infrared only or visible and infrared data were used. In this period, the transforms used meant that, for the NN PoP, the threshold probability below which echoes are removed was 0.143 (i.e. a NN PoP of 0.143 is equal to an alpha of 1).

The NN scheme performed better than the Nimrod scheme in terms of the accuracy of anaprop diagnosis. Some 97% of the cases examined showed the NN scheme to have either diagnosed all cases of anaprop correctly, or retained only some light anaprop or clutter. This compares favourably with 84% for the Nimrod scheme. In terms of the accuracy of rain diagnosis, the current Nimrod scheme performed better than the NN scheme. A total of 77% of the cases examined showed the Nimrod scheme to have either diagnosed all cases of rain correctly or removed only an insignificant amount of light rain (e.g. from the edges of rain clouds). The NN scheme diagnosed 60% of cases to this level of accuracy, with the remainder having rain deletion likely to give a misleading analysis and forecast, or deletion of significant amounts of rain.

Over the winter assessment period (28th October–18th December 1998), 144 cases were examined. The anaprop assessment scores show that the current Nimrod scheme removed more anaprop and clutter than the NN scheme in 47% of cases examined. In 76% of cases examined, the NN scheme retained a significant amount of anaprop or an amount likely to cause problems for TV applications. The NN scheme performed better than the Nimrod scheme in terms of the amount of rain retained. In the rain assessment, the NN scheme deleted significant amounts of rain in only 1% of cases, compared with 15% of cases for the Nimrod scheme. On the basis of the assessment results, the NN PoP classifier has not been implemented operationally within Nimrod.

The NN scheme showed early promise, particularly for improving PoP diagnosis based on infrared data only. Although a number of problems emerged during the trial period, it became apparent that nearly all of these could be remedied. Given that few alternatives exist to improve the analysis of spurious echoes from independent data, and that the NN technique can be refined with the use of surface temperature, snow cover and a surface climatological albedo field (all used in the current scheme), an improved NN PoP classifier is currently being developed.

5 CONCLUSIONS

The role of meteorological satellite imagery as a data source for nowcasting products has been described, emphasising the need for the ability of such products to

incorporate ancillary information, such as time, geographical location and NWP analysis data. From a list of nowcasting techniques currently available, it appears that NN, in the role of pattern recognition systems, can provide the flexibility and speed for building the type of nowcasting products that are being required for near-real time operational use.

Two examples of the application of NN used in this way have been presented, one for use in discriminating stratiform cloud and convective cloud in the GANDOLF flood forecasting system, and the other for determining a probability of precipitation from satellite imagery in order to assess rain rates from the British Isles radar network in the Nimrod nowcasting system.

The cloud discriminator undertook some 3 years of testing in a pre-operational environment before its limitations could be properly assessed. Modification of NN for additional sources of information and new training data is a relatively simple matter, and an improved version of the cloud classifier became part of the operational GANDOLF system on 1st June 1999.

The probability of precipitation estimator for Nimrod underwent trialling during 1998, and although it showed early promise with an ability to detect light rain from shallow convection in summer, also showed a number of problems, particularly during winter, when there is small contrast between land surface temperature and low cloud. However, it is expected that the problems can be remedied, and further development is planned in the near future.

The applications of NN described in this chapter have been shown to provide useful near-real time operational techniques. From the wealth of data expected from new and upcoming meteorological satellites, and from the increasing complexity of nowcasting systems, it is expected that the requirement for such techniques will continue well into the foreseeable future.

REFERENCES

Arkin, P.A. & Meisner, B. 1987. The relationship between large-scale convective rainfall and cold cloud over the Western Hemisphere during 1982–1984. *Monthly Weather Review* 115: 51–74.

Arking, A. 1964. Latitudinal distribution of cloud cover from TIROS III photographs. *Science* 143: 569–572.

Bankert, R.L. 1994. Cloud classification of AVHRR imagery in maritime regions using a probabilistic neural network. *Journal of Applied Meteorology* 33: 909–918.

Barrett, E.C. & Martin, D.W. 1981. *The use of satellite data in rainfall monitoring*. London: Academic Press.

Cheng, M., Brown, R. & Collier, C.G. 1993. Delineation of precipitation areas using Meteosat infrared and visible data in the region of the United Kingdom. *Journal of Applied Meteorology* 32: 884–898.

Cheng, M. & Brown, R. 1995. Delineation of precipitation areas by correlation of Meteosat visible and infrared data with radar data. *Monthly Weather Review* 123: 2743–2757.

Coakley, J.A. & Bretherton, F.P. 1982. Cloud cover from high resolution scanner data: detecting and allowing for partially filled fields of view. *Journal of Geophysical Research* 87: 4917–4932.

Collier, C.G., Hardaker, P.J. & Pierce, C.E. 1995. The GANDOLF system: An automatic procedure for forecasting convection. *Proc. 27th conference on radar meteorology, Vail, Colorado,* 1995: 606–608. Boston: AMS.

Conway, B.J. 1992. Current developments in very-short range forecasting. In *Report of the CAS group of rapporteurs on short-range weather prediction research, TD 479*. Geneva: WMO.

Conway, B.J. 1999. An overview of nowcasting techniques. *SAF training workshop: Nowcasting and very short range forecasting, EUM P25*: 34–43. Darmstadt: EUMETSAT.

Cullen, M.J.P. 1993. The unified forecast/climate model. *Meteorological Magazine* 122: 81–94.

Desbois, M., Seze, G. & Szejwach, G. 1982. Automatic classification of clouds on Meteosat imagery: application to high-level clouds. *Journal of Applied Meteorology* 21: 401–412.

Ebert, E. 1987. A pattern recognition technique for distinguishing surface and cloud types in the polar regions. *Journal of Climate and Applied Meteorology* 26: 1412–1427.

Foody, G.M., McCulloch, M.B. & Yates, W.B. 1995. The effect of training set size and composition on artificial neural network classification. *International Journal of Remote Sensing* 16: 1707–1723.

Garand, L. 1988. Automated recognition of oceanic cloud patterns. Part I: Methodology and application to cloud climatology. *Journal of Climate* 1: 20–39.

Golding, B.W. 1998. Nimrod, a system for generating automated very short range forecasts. *Meteorological Applications* 5: 1–16.

Grassotti, C. & Garand, L. 1994. Classification-based rainfall estimation using satellite data and numerical forecast model fields. *Journal of Applied Meteorology* 33: 159–178.

Griffith, C.G., Woodley, W.L., Browner, S., Teijeiro, J., Maier, M., Martin, D.W., Stout, J. & Sikdar, D.N. 1978. Rain estimation from geosyncrhonous satellite imagery – visible and infrared studies. *Monthly Weather Review* 106: 1153–1171.

Gu, Z.Q., Duncan, C.N., Grant, P.M., Cowan, C.F.N., Renshaw, E. & Mugglestone, M.A. 1991. Textural and spectral features as an aid to cloud classification. *International Journal of Remote Sensing* 12: 953–968.

Hand, W.H. & Conway, B.C., 1995. An object-oriented approach to nowcasting showers. *Weather & Forecasting* 10: 327–341.

Haralick, R.M. 1986. Statistical image texture analysis. In T. Young & K. Fu (Eds.), *Handbook of pattern recognition and image processing* (pp. 247–279). New York: Academic Press.

Karlsson, K.G. 1994. Satellite-estimated cloudiness from NOAA AVHRR data in the Nordic area during 1993. *Meteorology & climatology report RMK 66*. Norrköping: Swedish Meteorological and Hydrological Institute.

Kidder, S.Q. & Vonder Haar, T.H. 1995. *Satellite meteorology: an introduction*. London: Academic Press.

Levizzani, V. 1999. Convective rain from a satellite perspective: achievements and challenges. *SAF training workshop: Nowcasting and very short range forecasting, EUM P25*: 75–84. Darmstadt: EUMETSAT.

Lovejoy, S. & Austin, G.L. 1979. The delineation of rain areas from visible and IR satellite data for GATE at mid-latitudes. *Atmosphere-Ocean* 17: 77–92.

MacLaren, I., Walder, P. & Reid, C. 1994. Neural networks and cloud classification. *Proc. 10th EUMETSAT data users' conference, Cascais, Portugal, 5–9 September 1994, EUM P15*: 541–547. Darmstadt: EUMETSAT.

Olson, W.S., Kummerow, C.D., Viltard, N.F. & Hong Y. 1998. Retrieval of precipitation and latent heating profiles from combined TRMM TMI/PR/VIRS observations. *9th conference on satellite meteorology and oceanography, Paris, France, 25–29th May 199, EUM P22*: 505–508. Boston: AMS.

Pamment, J.A. & Conway, B.J. 1998. Objective identification of echoes due to anomalous propagation in weather radar data. *Journal of Atmospheric and Oceanographic Technology* 15: 98–113.

Pankiewicz, G.S. 1994. The Application of Pattern Recognition Techniques to Cloud Detection. *Forecasting Research Division Technical Report no.129*. Bracknell: UK Met Office.

Pankiewicz, G.S. 1995. Pattern recognition techniques for the identification of cloud and cloud systems. *Meteorological Applications* 2: 257–271.

Pankiewicz, G.S. 1997. Neural network classification of convective airmasses for a flood forecasting system. *International Journal of Remote Sensing* 18: 887–898.

Parikh, J. 1997. A comparative study of cloud classification techniques. *Remote Sensing of the Environment* 6: 67–81.

Pierce, C.E., Collier, C.G. & Hardaker, P.J. 1995. Forecasts of heavy convective rain for use within a flash flood warning system. *Proc. 5th British Hydrological Society Symposium, Edinburgh, UK*: 4.17–4.23. London: British Hydrological Society.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. 1986. Learning internal representation by error propagation. In Rumelhart, D.E., McClelland, J.L. (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol.1 (pp. 318–362.) MIT Press: Cambridge MA.

Weszka, J.S., Dyer, C.R. & Rosenfeld, A. 1976. A comparative study of texture measures for terrain classification. *IEEE Transactions on System, Man and Cybernetics* 6: 269–285.

Wu, R., Weinman, J.A. & Chin, R.T. 1985. Determination of rainfall rates from GOES satellite images by a pattern recognition technique. *Journal of Atmospheric and Oceanic Technology* 2: 314–330.

14

Mapping Land Cover from Remotely Sensed Imagery for Input to Hydrological Models

GILES M. FOODY
School of Geography, University of Southampton, Highfield, UK

ABSTRACT: Remote sensing has been widely used as an input to hydrological models. Often the remotely sensed data are used indirectly, especially through the derivation of land cover data that may be incorporated into spatially distributed models. Although land cover mapping is one of the most common applications of remote sensing, there are numerous problems with the techniques used, which limit map accuracy and thus the associated value of the map to hydrological modellers. This chapter focuses on the potential of feedforward neural networks as a tool for land cover mapping using supervised digital image classification. Neural networks are considered relative to conventional methods and examples of comparative performance in different situations are provided.

1 INTRODUCTION

Remote sensing has many applications in hydrology but is particularly attractive as a source of data for hydrological models. Modelling work has progressed from a period characterised as being data sparse and computationally constrained to one that is data rich and computationally powerful (DeCoursey, 1988; Melone *et al*., 1998; Storck *et al*., 1998; Jakeman *et al*., 1999). Many hydrological models have been produced but selecting a model and obtaining appropriate data to use in it can, however, be difficult (Melone *et al*., 1998). Initially, and to some extent as a function of past data availability and computing facilities, emphasis was placed on lumped models. Although valuable, lumped models are spatially constrained. Remote sensing has the potential to provide complete data coverage of large areas. As a consequence, remote sensing may be used to parameterise spatially distributed models (Harvey & Solomon, 1984). These distributed models offer the potential to refine our understanding of important

hydrological issues, particularly those for which the spatial dimension is important (Dunn *et al*., 1998; Frankenberger *et al*., 1999).

Artificial neural networks (NN) are a form of computational intelligence that imitates functions of the human brain (Aleksander & Morton, 1990; Fischer, 1998). NN are general purpose computing tools that can be used to solve complex non-linear problems (Simpson & Li, 1993; Fischer, 1996) and have been used in a wide range of applications in remote sensing. These tools have been used for unsupervised classification (Baraldi & Parmiggiani, 1995), geometric correction (Smith *et al.*, 1995), image compression (Walker *et al.*, 1994), model inversion or variable estimation (Pierce *et al.*, 1994; Schweiger & Key, 1997; Wang & Dong, 1997), but especially for supervised classification (Benediktsson *et al.*, 1990; Kanellopoulos *et al.*, 1992; Mannan & Ray, 2003). A broad range of different network types have been used including radial basis function networks (RBFN) and binary diamond networks (BDNN) (Salu & Tilton, 1993; Bishop, 1995; Murnion, 1996) but by far the most widely used is the feedforward multi-layer perceptron (MLP) network (see Chapter 2). For brevity, attention here is focused only on the use of feedforward NN for supervised digital image classification. This technique converts remotely sensed imagery, sometimes in association with ancillary data sets such as topography, into a thematic map.

Remote sensing has developed into a major source of data for hydrological models (Klein & Barnett, 2003; Lacaze *et al.,* 2003; Hillard *et al.*, 2003; Mackaya *et al.,* 2003). The image data are acquired by a sensor in a consistent manner that provides a complete and near-simultaneous coverage of large areas. Furthermore, remotely sensed imagery are available at a range of spatial and temporal scales and so capable of providing information on a vast range of environmental issues (Foody & Curran, 1994). In essence, the imagery acquired by remote sensors provide a spatial representation of the manner in which electromagnetic radiation interacts with the Earth's surface. The nature of the interactions of radiation with the Earth's surface is controlled by a set of state variables (Verstraete *et al*., 1996; Curran *et al*., 1998). As these state variables, typically basic physical, chemical and biological properties of the surface, control the observed remotely sensed response they can be estimated directly from remotely sensed imagery. Fortunately some hydrological variables of interest are, or are very highly correlated with, state variables and so can also be estimated directly. For example, evapotranspiration, a fundamental component of the hydrological cycle, may be estimated from remotely sensed imagery acquired in the thermal infrared part of the electromagnetic spectrum (Hoshi *et al*., 1989; Engman & Gurney, 1991). The estimation of hydrological variables from remotely sensed data can, however, be extremely difficult, especially if the variable of interest is only indirectly related to the observed remotely sensed response or requires highly accurate radiometric calibration of the data. It is, therefore, common for remote sensing to be used indirectly as a source of data for hydrological models. Thus, for example, measures of vegetation amount such as leaf area index (LAI), which is effectively a state variable (Curran *et al*., 1998; Foody & Boyd, 2002), or biomass (Foody *et al*., 2001, 2003; Held *et al*., 2003) can be derived remotely and may be used to derive estimates of vegetation parameters for hydrological models (Schultz, 1996; Watson *et al*., 1999). Similarly, land cover

is strongly linked to a range of hydrological variables of interest and state variables controlling the remotely sensed response. Land cover has a strong influence over key hydrological properties such as interception, infiltration and evaporation. Thus land cover maps may be used to estimate a range of hydrological variables or used to parameterise models (Hoschi *et al*., 1989; Neumann & Schultz, 1989; Sharma & Singh, 1992; Garatuza-Payan *et al*., 1998; Storck *et al*., 1998). Land cover maps derived from remotely sensed images have, therefore, been widely used as an input to hydrological models including those focused on the estimation of evapotranspiration (Jensen & Chery, 1980; Uchida & Hoshi, 1988), and stream flow and sediment yield (Harvey & Solomon, 1984). Furthermore, as land cover controls many basic properties, changes in land cover have important hydrological implications. The effects of major changes in land cover and use such as those associated with urbanisation (Olsin *et al*., 1988; Bellal *et al*., 1996; Kang *et al*., 1998; Tsihrintzis & Hamid, 1998), forest clearance (Schultz, 1996; Brooks & Spencer, 1997; Crockford & Richardson, 1998; Woube, 1999), agricultural expansion and management practices (Boardman, 1995; Barjracharya & Lal, 1999) have, amongst others (Walling, 1981), attracted considerable interest. These land cover changes may be monitored as a consequence of the temporal dimension of remote sensing.

Remote sensing may be used to map land cover at a range of spatial and temporal scales. Classification techniques are generally used in the mapping of land cover from remotely sensed imagery. This applies to both visual and digital based approaches to mapping. Visual interpretation proceeds in a similar manner to aerial photograph interpretation, with the analyst using image tonal, textural and contextual features to identify land cover, often with the aid of a discrimination or interpretation key (Lillesand & Kiefer, 2000). Although visual classification can be highly accurate, digital analyses are often preferred. Digital classifications can be undertaken more objectively than visual analyses and can more easily handle the voluminous digital images acquired by current and proposed sensing systems. Moreover, the output of the digital classification is typically in a format conducive for integration with other digital data sets, particularly those held within a geographical information system (GIS), which is attractive for use with hydrological models, especially distributed models.

Many digital image classification approaches have been used to map land cover with varying degrees of success. Two broad categories of classification technique have been used, unsupervised and supervised (Mather, 1999; Tso & Mather, 2001). Unsupervised classifications seek to identify natural classes within the imagery. They are essentially clustering algorithms that identify a set of spectrally dissimilar classes present in the imagery. Once derived the analyst must then attempt to label the classes in some meaningful way. If, as is more typical, the classes of interest are known in advance it may be preferable to use a supervised classification. With a supervised classification the analyst specifies the classes of interest and provides the classification with information on which to characterise each class in the imagery. This enables the analyst to focus on the classes of interest to the particular study, which can be a major advantage as the classes of hydrological significance may not be depicted on available thematic maps (Knox & Weatherfield, 1999). On the basis of the class

characterisations derived and the classifier's decision rule, the class membership of every image pixel may be predicted. If desired, different spatial units (e.g. fields or land parcels such as hillslopes rather than individual pixels) that match more closely the hydrological issues under study (Bronstert, 1999) could be used as the basic spatial unit in a particular classification; for simplicity the standard per-pixel approach to classification will be assumed throughout but the discussion has applicability to other approaches.

Despite the considerable developments made recently, the accuracy with which land cover may be mapped from remotely sensed imagery with a supervised classification is often perceived to be inadequate for operational applications (Wilkinson, 1996). This limits the value of land cover maps derived from remotely sensed imagery for input to hydrological models as error will propagate into later analyses based upon them. Consequently, the considerable potential of remote sensing as a source of land cover data for use in hydrological models is not being realised. Many reasons may be cited for this situation. These include issues such as the nature of the land cover classes, the spectral, spatial and radiometric resolutions of the remotely sensed imagery and the methods used in mapping (Meiner, 1996; Takara & Kojima, 1996; Campbell, 2002). Here, attention is focused only on the latter issue with particular emphasis on the potential use of NN as tools for mapping land cover from remotely sensed imagery.

2 SUPERVISED DIGITAL IMAGE CLASSIFICATION

Supervised image classification is one of the most common digital image analyses undertaken in remote sensing. It aims essentially to convert the remotely sensed imagery, sometimes in association with ancillary data sets (e.g., topography), into a thematic map. Prior to a classification a series of pre-processing analyses may be undertaken to prepare the imagery. Typically pre-processing involves the removal or reduction of distortions or degradations in the imagery (Mather, 1999). This may include making corrections for atmospheric attenuation effects (Chavez, 1996), variations in the sensor's viewing geometry due to both sensor and terrain variables (Meyer *et al*., 1993; Ekstrand, 1996) and image geometry (Mather, 1999). If, for example, a time series of imagery is to be used in mapping it may also be necessary to radiometrically correct or calibrate the imagery to absolute units. Further pre-processing of the imagery may be undertaken to meet the requirements of the particular mapping investigation. If the imagery are noisy (e.g. synthetic aperture radar imagery) the analyst may seek to reduce noise prior to the classification through the application of a smoothing or low-pass filter. Perhaps, more commonly, it may be desirable to use only a sub-set of the remotely sensed data in the classification. Although multispectral imagery are usually required for an accurate land cover classification, many sensors, especially hyperspectral instruments, provide an unnecessarily large amount of data for classification applications. Typically the imagery acquired in the different wavebands of a sensor are highly inter-correlated and, therefore, the useful discriminatory information could be conveyed by a subset of the wavebands recorded. A feature selection analysis may be undertaken to define a

sub-set of the acquired imagery that reduces the size of the data set while maintaining the useful discriminatory information. This is often based on inter-waveband comparisons of class separ-ability or data compression techniques such as principal components analysis (Mather, 1999). The feature selection is helpful not only in reducing the size of the data set but also in terms of reducing ground data requirements and maximising classification accuracy. It can, however, be difficult and sometimes the data discarded may contain highly significant discriminatory information.

When all pre-processing operations have been completed the imagery may be classified. A supervised digital image classification can be broken down into three basic stages. First is the training stage which involves the analyst identifying sites of known class membership in the imagery. These sites are used to derive training statistics that describe the classes in the remotely sensed data. In the second, class allocation, stage of the classification these training statistics are used with the classification decision rule to allocate each pixel in the remotely sensed image to one of the defined classes. For example, the maximum likelihood classification allocates each pixel to the class with which it has the highest posterior probability of membership. The third and final stage of the classification is the testing stage in which the accuracy of the classification is assessed (Campbell, 2002). Many factors influence the quality of a supervised digital image classification. Ground data on class membership are, for example, required in the training and testing stages and the acquisition and quality of these data can have a major impact on the resulting classification (Campbell, 2002). The classification algorithm must also be selected with care. Classification algorithms differ enormously in terms of the assumptions made and sensitivity to deviations from the assumed conditions. The maximum likelihood classification, for example, is a conventional statistical classifier that assumes, amongst other things, that the spectral data for each class are normally distributed. This is often not the case and also places a demand for a large ground data set on which to derive accurate estimates of the class distributions (Mather, 1999). Nonetheless, the conventional statistical classification approaches are the most widely used techniques for land cover mapping from remotely sensed imagery, although a range of, often non-parametric, alternatives have been advocated (e.g. Dymond, 1993; Peddle, 1993; Tso & Mather, 2001).

3 ADVANTAGES OF NEURAL OVER STATISTICAL CLASSIFICATION

Conventional statistical image classification techniques are not always appropriate for mapping land cover from remotely sensed imagery. For example, the requirements and assumptions of the maximum likelihood classification, one of the most widely used techniques, are often unsatisfiable. Of particular concern are four inter-related problems:

- In a conventional parametric classifier, the data are assumed to be normally distributed. This may often not be the case with remotely sensed imagery. Furthermore, there may be significant inter-class differences in the

distributions which cannot be normalised prior to the classification as class membership, the desired output of the classification, is unknown.

- To define a representative sample from which the descriptive statistics (e.g. mean and variance) are generated and upon which the analysis is based may require a large amount of data. Typically, it is recommended that the minimum training set size is some 10–30 times the number of discriminating variables (e.g. wavebands) per-class (Mather, 1999). Clearly a very large training set is required for mapping from some multi-spectral imagery and this runs contrary to a major goal of remote sensing, which involves extrapolation over large areas from limited ground data. With high dimensional data sets, such as those acquired by hyperspectral sensors operating in hundreds of wavebands, the training set requirement for correct application of such a classification may be exorbitantly high. Related to this issue, the 'Hughes Phenomenon', whereby classification accuracy may decline with an increase in the number of discriminating variables, may be observed with algorithms such as the maximum likelihood classification. To avoid this problem it may be necessary to perform some kind of feature selection to focus only on part of the data set which conveys useful discriminatory information.
- The classification can only make use of data acquired at a high level of numerical precision (e.g., ratio level) and cannot accommodate directly data with a directional component. Unfortunately, there may often be useful discriminatory information that is available at a low level of measurement precision (e.g., a nominal level soil type map) or has a directional component (e.g., terrain slope aspect). As re-scaling the data is often difficult the only practical means of integrating the ancillary data is by stratifying the imagery by the low level ancillary data and classifying each stratum independently. This will, however, also compound the training data requirement problem as each stratum will require its own training set.
- The maximum likelihood classification is computationally demanding and, therefore, relatively slow. The significance of this problem may become increasingly evident in the near future given the large data volumes anticipated from proposed remote sensing systems (Gershon & Miller, 1993) and increasing use of hyperspectral sensors.

Although the analyst may proceed with a maximum likelihood classification when its assumptions are not satisfied, for example without correcting for non-normal distributions or with disregard to the training set size requirements for a particular data set, it is likely that the full information content of the remotely sensed imagery would not be utilized. Consequently, alternative classification approaches have been sought and recently much attention has focused on feedforward NN.

The feedforward NN approach to supervised image classification is less sensitive to some of the problems associated with conventional statistical classifications. This is evident in relation to the four problems with the maximum likelihood classification discussed above. First, NN make no assumptions about the nature and distribution of the data. As a consequence, a large sample may

not required to estimate the properties of the distribution, although a representative training set is still required to provide an adequate description of the classes with training set properties, such as size and composition, requiring careful selection in relation to the properties of the classes and the network itself (Baum & Haussler, 1989; Foody *et al.*, 1995; Staufer & Fischer, 1997). Second, as NN learn the underlying relationships in the data and effectively weight the importance of the discriminating variables, such tools may, therefore, have no limitations placed on data dimensionality which reduces the need for feature selection to identify, for instance, an optimal band combination for a classification. Pre-processing operations including feature selection can, however, still be beneficial, particularly in the reduction of network complexity and thereby training requirements in addition to the provision of potential increases in classification accuracy (Chang & Lippmann, 1991; Benediktsson & Sveinsson, 1997; Yu & Weigl, 1997; Kavzoglu & Mather, 2002). Third, NN are able to use directly data acquired at any level of measurement precision and accommodate directional data when appropriately scaled. These factors combined also enable NN to be used as black box tools which may be attractive when there is little or no prior knowledge about the particular problem. Lastly, classification by trained NN are extremely rapid (Peddle *et al*., 1994).

4 SUPERVISED IMAGE CLASSIFICATION WITH A NEURAL NETWORK

Feedforward NN are particularly attractive for supervised image classification as a consequence of their ability to learn by example and generalise (Schalkoff, 1992). Each network may be envisaged as comprising a set of simple processing units arranged in layers, with each unit in a layer connected by a weighted channel to every unit in the adjacent layer(s). Combined these elements transform the remotely sensed image (which essentially depicts the measured multispectral radiance) into a thematic map (depicting the spatial distribution of the land cover classes of interest) (Fig. 14.1). The NN architecture is determined by a range of factors which relate, in part, to the nature of the remotely sensed imagery and the desired land cover classification. There is, for example, usually an input unit for every discriminating variable and an output unit associated with each class in the classification. The number of hidden units and layers is defined subjectively, often on the basis of a series of trial runs. Alternatively, the number of hidden units and layers could be defined with the aid of various published ‘rules of thumb’ or optimised with the use of methods that allow the network to add or delete units until a satisfactory structure is produced (Chauvin, 1989; Jiang *et al*., 1994; Bishop, 1995; Bischof & Leonardis, 1997; Kavzoglu & Mather, 2003). In general, the larger the number of hidden units and layers used, the more able the network will be to learn the training data but this may be achieved at the expense of an undesirable reduction in the overall capacity for generalisation and an increase in computer processing time.

Each NN is initially constructed with the inter-unit weights set at randomly defined values. The magnitude of each of these weighted connections is then

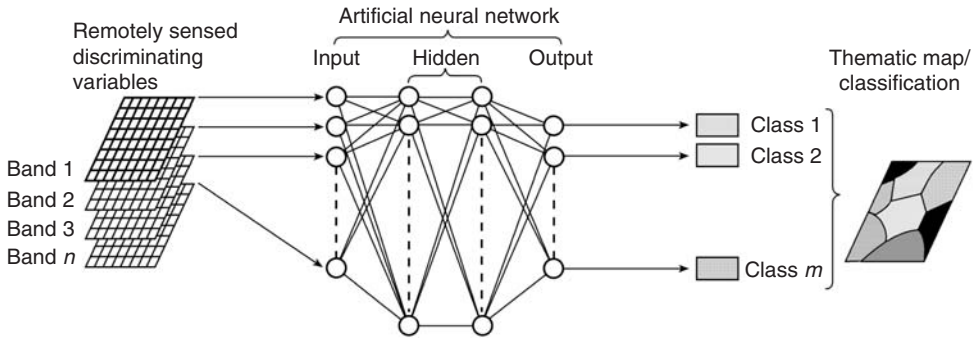


Fig. 14.1. Classification of remotely sensed imagery with an artificial neural network. Typically a bias unit (not shown for clarity) is connected to the processing units by a weighted connection and included in the training process.

adjusted during an iterative training process which aims to minimise output errors (Schalkoff, 1992). Once the overall output error has declined to an acceptable level, which is typically determined subjectively, training ceases and the trained network is then ready for the classification of pixels of unknown land cover class membership. For this, the remotely sensed data for each pixel of unknown class membership are input to the trained network and the pixel allocated to the class associated with the most highly activated unit in the output layer. This type of approach has been widely used for the classification of remotely sensed imagery and generally found to be more accurate than alternative classification approaches (Bendiktsson *et al*., 1990; Kanellopoulos *et al*., 1992; Peddle *et al.*, 1994; Foody *et al*., 1995). Before briefly illustrating its potential with an example, some disadvantages of the approach will be considered and possible solutions to some problems raised indicated by a further example.

5 PROBLEMS WITH NEURAL NETWORKS FOR SUPERVISED CLASSIFICATION

Although emulating some aspects of the human eye-brain system, which is very effective for pattern recognition, NN generally used in mapping land cover from remotely sensed imagery have the computing power of lower life-forms such as earthworms (Simpson & Li, 1993). The relative simplicity of NN typically used for land cover mapping can be illustrated with reference to the number of weighted connections in the network which is a function of the number of units and their arrangement. The human brain, for instance, contains some 10^{11} units or neurons (Aleksander & Morton, 1990) while NN used in remote sensing are much smaller, with typically less than 10^3 units. There are many other problems associated with the use of NN for supervised image classification. Here, some are briefly discussed and possible solutions indicated.

While NN may generally be used to classify imagery at least as accurately as other classification approaches there are a range of factors that limit their use (Wilkinson, 1997). Feedforward NN are, for example, semantically poor. Thus,

while NN may be able to accurately map land cover from remotely sensed imagery it is difficult understand how the result was achieved. It is, for example, difficult to identify the relative contribution of different wavebands for inter-class discrimination which would be useful in the design of new sensors. Some information may be gleaned from an analysis of the weighted connections but if the analyst wanted to understand how, for example, a particular class allocation was achieved it may be preferable to adopt an alternative technique and those based on genetic programming and fuzzy logic may be more appropriate (Corne *et al.*, 1996). The accuracy of a classification is also not always the only concern of the analyst. Other criteria of classification performance may be important. In many of the comparative studies undertaken, performance criteria other than accuracy, notably training time, have been evaluated and revealed that NN may be less attractive than other classification approaches. This problem may, however, be resolved through developments in computing, particularly in parallel hardware or through other approaches that seek to accelerate training (Dawson *et al.*, 1993; Manry *et al.*, 1994). Alternatively, if training time is a major constraint, different network types, such as the 'one-shot' BDNN (Salu & Tilton, 1993) and RBFN (Bishop, 1995), or fast learning algorithms, could be used.

Perhaps one of the most important problems is that classification, by whatever method, is highly subjective (Johnston, 1968). Despite apparent objectivity, the analyst has control over a range of NN parameters that strongly influence network performance, especially in terms of speed and accuracy (Foody, 1999a). Issues such as the selection of a suitable network architecture and properties, together with the avoidance of problems such as overfitting to the training data while deriving a sufficient generalisation capacity are important (Fischer *et al.*, 1997), but largely based on subjective decisions. Even if the various network parameters are selected judiciously there is still no guarantee that NN will provide an acceptable let alone optimal land cover classification. While the adoption of NN avoids problems with the assumptions made by other classification techniques it does not free the analyst from a range of basic problems that are common to all supervised image classifications. Many factors will influence classification performance. For instance, the accuracy of the classification may be constrained largely by the quality of the training data. Issues such as the size and composition of the training sets have a considerable effect on the accuracy of NN classifications (Zhuang *et al.*, 1994; Foody *et al.*, 1995; Blamire, 1996; Staufer & Fischer, 1997) as they do on other classification approaches. The analyst must also accept that the use of NN does not guarantee that the classification will be sufficiently accurate. Although generally more accurate than other classifications, the accuracies reported in the literature, like those of other classifiers, often fall short of an operationally acceptable level (Wilkinson, 1996). Moreover, NN classifications do not always provide the highest classification accuracy, on a per-class and/or overall basis and some form of consensual or hybrid approach may be desirable (Wilkinson *et al*., 1995; Wilkinson, 1996). Perhaps more importantly, the appropriateness of classification as a tool for mapping is debatable.

The classification process outlined and its resulting outputs are 'hard', with each image pixel allocated to a single land cover class. This type of classification

is only appropriate for the mapping of classes that are discrete and mutually exclusive. On many occasions this will not be the case. Many land cover classes are continuous and so inter-grade (Foody, 1996; Foody & Boyd, 1999; Kent *et al*., 1997). This is in part a consequence of class definition. For example, forest classes are often defined on the basis of relatively arbitrary thresholds of tree canopy cover. Furthermore, 'hard' classification is only appropriate if the basic spatial unit used, typically the pixel, is pure (i.e., represents an area of homogeneous cover of a single class). This is rarely the case, with many pixels of mixed land cover composition contained within remotely sensed imagery (Campbell, 2002). These, mixed pixels, may occur whatever the nature of the classes. For instance, with continuous classes, mixed pixels will occur in the inter-class transition zones where the classes co-exist spatially. Whereas for discrete classes, the area represented by a pixel will often enclose or straddle class boundaries. The exact proportion of mixed pixels in an image will vary with a range of factors, notably the land cover mosaic on the ground and the sensor's spatial resolution, but is often very large (Campbell, 2002). Mixed pixels may, for instance, vastly dominate coarse spatial resolution imagery used in mapping land cover at regional to global scales (Foody *et al.*, 1997). With fine spatial resolution imagery mixing also occurs but here the concern is generally about the extent of sub-class components, such as soil, leaves and shadow of an individual vegetation class. Mixed pixels will, therefore, be evident in fine spatial resolution data sets, particularly for heterogeneous classes such as urban areas (Townshend, 1981).

For the full potential of remote sensing as a source of land cover data to be realised alternative approaches to conventional 'hard' classifications may be required. Most attention has focused on fuzzy or soft classifications that allow for partial and multiple class membership (Wang, 1990; Tso & Mather, 2001). In such a classification, the full class membership of each pixel is partitioned between all classes and so a pixel can display any possible membership scenario, from full membership to one class through to having its membership divided, in any permutation, between all classes.

The conventional NN approach to classification outlined above is 'hard' but can be softened to provide a fuzzy land cover classification (Foody, 1996). For this, the magnitude of the activation level of an output unit may be taken as a measure of the strength of membership to the class associated with the unit that reflects the fractional coverage of the class in the area represented by the pixel (Foody, 1996, 1997) in a manner similar to mapping probabilities from the maximum likelihood classification (Foody, 1996). Thus, rather than deriving only the code of the class associated with the most activated network output unit the magnitude of the activation level of each output unit could be derived and mapped. This makes fuller use of the information content of the remotely sensed imagery and may enable a more accurate and appropriate representation of land cover to be derived. This applies to both relatively discrete (Foody *et al*., 1997) and continuous land cover classes (Foody & Boyd, 1999). The magnitude of the output units has also been utilised in various measures for different hydrological and meteorological applications (e.g. Chapter 13).

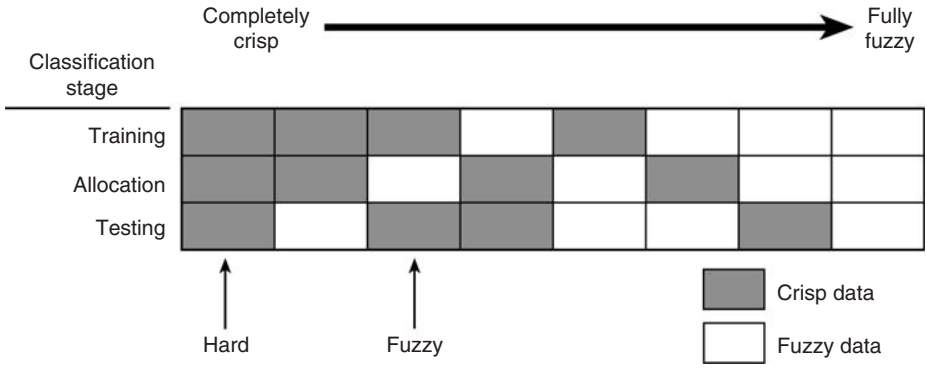


Fig. 14.2. The continuum of classification fuzziness.

In addition to being able to derive a fuzzy class allocation, it is possible to use mixed pixels or fuzzy data directly in the training stage of NN classification, as the analyst must specify the target vector for the training data set (Foody, 1997; Foody *et al*., 1997). As a consequence, NN may be able to produce a land cover classification at any point along the continuum of classification fuzziness (Foody, 1999b). At the hard end of this continuum are the conventional 'hard' or completely-crisp classifications (Fig. 14.2). These are based on the standard approach to classification in which a pixel is associated with a single class at each stage of the classification and so the data may be considered to belong to crisp rather than fuzzy sets. Most supervised image classifications adopt this approach but its application may be inappropriate due, for instance, to the presence of mixed pixels. At the other extremity of the continuum are fully-fuzzy classifications (Fig. 14.2). In these, fuzziness is accommodated in all three stages of the classification. This type of approach may provide a more realistic and accurate representation of the land cover of a site and use more fully the information content of the remotely sensed imagery. Between these extremes lie classifications of varying fuzziness, including those generally referred to in the literature as fuzzy classifications, in which only the class allocation stage actually accommodates fuzziness (Foody, 1999b). Recognising the existence of the continuum and designing a classification to fit the appropriate point along it for a particular study may aid land cover mapping and thereby hydrological modelling. The potential of feedforward NN for classification along the continuum can be illustrated by comparative evaluation against conventional classification methods.

6 EVALUATION RELATIVE TO OTHER CLASSIFICATION APPROACHES

Numerous comparative studies have been undertaken to assess the accuracy of NN based classifications relative to those derived from more conventional classification approaches such as maximum likelihood classification, linear

discriminant analysis and evidential reasoning (e.g. Benediktsson *et al*., 1990; Peddle *et al*., 1994; Paola & Schowengerdt, 1995). As a generalisation, these studies have revealed that a neural network may be used to map land cover at least as accurately, but commonly more accurately than conventional classification approaches. For instance, Peddle *et al*. (1994) show that classifications derived from NN were generally, but not always, more accurate than those derived using maximum likelihood and evidential reasoning approaches. Moreover, the incorporation of use of additional discriminatory information (e.g., image texture) to the analyses increased the accuracy of NN classifications while that of the conventional maximum likelihood classification declined. NN, therefore, have considerable potential for accurate land cover mapping and in the realisation of the potential of remote sensing as a source of land cover data for input to hydrological models. This can be illustrated, here with reference to classifications undertaken at either end of the continuum of classification fuzziness.

6.1 *Hard*

In many studies of agricultural regions land cover maps depicting crop types have been used to parameterise hydrological models. Remote sensing is particularly attractive for crop mapping as the temporal repeat cycle of many sensing systems enables the observation of crop-specific cycles which greatly enhances class separability (Middlekoop & Janssen, 1991). The relative advantage of the NN approach to conventional classification can be illustrated with a simple example. Using fine spatial resolution SAR imagery, in which mixing of classes within pixels could be avoided, that were acquired on four dates through a growing season Foody *et al*., (1995) compared the accuracy of mapping seven crop types using NN and discriminant analysis. The discriminant analysis used was relatively similar to the maximum likelihood classification and allocated each case to the class with which it has the highest posterior probability of membership (Tom & Miller, 1994). The NN comprised 4 input (one for each SAR image), 3 hidden and 7 output units (one for each class) and used the quickprop learning algorithm (Fahlman, 1988). Using the same data sets, the accuracy of the classifications were calculated as 65.5% and 81.9% for the discriminant analysis and NN, respectively. In this instance, the NN was therefore able to derive a significantly more accurate classification (at the 95% level of confidence) than the conventional statistical approach.

6.2 *Fuzzy*

In many instances the land cover to be mapped may be considered to be fuzzy. This is particularly the situation when the spatial resolution of the sensor used is comparable to or coarser than the typical size of the features being mapped (e.g. fields). In such circumstances a fuzzy classification may be undertaken and NN are particularly attractive as they may be used at any point along the continuum of classification fuzziness. This can be illustrated with an example

of mapping urban land cover. Urban land cover is highly heterogeneous and can significantly influence hydrological processes. Often of key concern is the amount and distribution of impervious surfaces such as roads, pavements and buildings. These are typically of sub-pixel size in satellite sensor data and so cannot be mapped accurately by conventional digital image classifications. However, NN may be trained with mixed data and used to derive a fuzzy classification. As an example, Foody (1997) used a BPNN with a single hidden layer to derive a fuzzy classification of an urban area. The NN predictions of sub-pixel class composition were found to be closely correlated with reference data ($r > 0.88$) indicating the potential to derive sub-pixel level land cover data. This is very advantageous when mixing is a problem, which is often the case for heterogeneous land covers and/or the remotely sensed data with a coarse spatial resolution (Foody, 2000; Zhang & Foody, 2001).

7 SUMMARY AND CONCLUSIONS

Remote sensing is a major source of information for hydrological models. Frequently the remotely sensed imagery are used indirectly through the provision of land cover data. NN are particularly attractive for the production of land cover maps from remotely sensed imagery via a supervised digital image classification analysis. The independence of restrictive assumptions, ability to integrate diverse data sets and, ultimately, the derivation of very accurate land cover classifications are key advantages of the NN approach. Although NN have many advantages over conventional classification approaches and have often been noted to provide more accurate classifications they are not without their problems. Indeed there is an argument that NN are the second best way of performing a task. Thus, if, for example, the data set to be classified does satisfy the requirements of the maximum likelihood classification then that approach, rather than a NN, may be the most appropriate. In such circumstances, the fundamental model underlying the maximum likelihood classification is a major advantage over the distribution-free black-box approach of the NN. By recognising that different classification approaches vary in their ability to separate the classes in an image, however, it may be appropriate to adopt a multi-classifier approach to make the best use of each method of classification (Wilkinson *et al.*, 1995; Roli *et al*., 1997; Warrender & Augusteijn, 1999).

A major advantage of NN for land cover mapping lies in their flexibility, particularly in relation to their applicability at any point along the continuum of classification fuzziness. This helps reduce the problems associated with the often-inappropriate dependence on conventional 'hard' classification techniques for land cover mapping when there is generally significant fuzziness to be accommodated. Since NN may be used to derive a classification at any point along the continuum of classification fuzziness they, therefore, have considerable potential for land cover mapping applications and play a significant role in the realisation of the potential of remote sensing as a source of information for hydrological modelling applications.

REFERENCES

Aleksander, I. & Morton, H. 1990. *An Introduction to Neural Computing.* London: Chapman and Hall.

Bajracharya, R.M. & Lal, R. 1999. Land use effects on soil crusting and hydraulic response on surface crusts on a tropical Alfisol. *Hydrological Processes* 13: 597–609.

Baraldi, A. & Parmiggiani, F. 1995. A neural network for unsupervised categorisation of multivalued input patterns: an application to satellite image clustering. *IEEE Transactions on Geoscience and Remote Sensing* 33: 305–316.

Baum, E.B. & Haussler, D. 1989. What size net gives valid generalizations. In D.S. Touretzky (Ed.), *Advances in Neural Information Processing Systems I*, 81–90. San Mateo, CA: Morgan Kaufmann.

Bellal, M., Sillen, X. & Zech, Y. 1996. Coupling GIS with a distributed hydrological model for studying the effect of various urban planning options on rainfall runoff relationship in urbanized watersheds. In K. Kovar & H.P. Nachtnebel (Eds.), *Application of Geographic Information Systems in Hydrology and Water Resource Management* (pp. 99–106). IAHS Publication 235, International Association of Hydrological Science, Wallingford.

Benediktsson, J.A. & Sveinsson, J.R. 1997. Feature extraction for multisource data classification with artificial neural networks. *International Journal of Remote Sensing* 18: 727–740.

Benediktsson, J.A., Swain, P.H. & Ersoy, O.K. 1990. Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing* 28: 540–551.

Bischof, H. & Leonardis, A. 1998. Finding optimal neural networks for land use classification. *IEEE Transactions on Geoscience and Remote Sensing* 36: 337–341.

Bishop, C.M. 1995. *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press.

Blamire, P.A. 1996. The influence of relative sample size in training artificial neural networks. *International Journal of Remote Sensing* 17: 223–230.

Boardman, J. 1995. Damage to property by run-off from agricultural land, south Downs, southern England. *The Geographical Journal*, 161: 177–191.

Brooks, S.M. & Spencer, T. 1997. Changing soil hydrology due to rain forest logging: an example from Sabah Malaysia. *Journal of Environmental Management* 49: 297–310.

Bronstert, A. 1999. Capabilities and limitations of detailed hillslope modelling. *Hydrological Processes* 13: 21–48.

Campbell, J.B. 2002. *Introduction to Remote Sensing* (3rd ed.) London: Taylor and Francis.

Chang, E.I. & Lippmann, R.P. 1991. Using genetic algorithms to improve pattern classification performance. In R.P. Lippmann, J.E. Moody & D.S. Touretzky (Eds.), *Advances in Neural Information Processing Systems 3* (pp. 797–803). San Mateo, CA: Morgan Kaufmann.

Chauvin, Y. 1989. A back-propagation algorithm with optimal use of hidden units. In D.S. Touretzky (Ed.), *Advances in Neural Information Processing Systems I* (pp. 519–526). San Mateo, CA: Morgan Kaufmann.

Chavez, P.S. 1996. Image-based atmospheric corrections – revised and revisted. *Photogrammetric Engineering and Remote Sensing* 62: 1025–1036.

Corne, S., Murray, T., Openshaw, S., See, L. & Turton, I. 1996. Using artificial intelligence techniques to model subglacial water systems. *Proceedings 1st International Conference on Geocomputation* 1: 135–155.

Cote, S. & Tatnall, A.R.L. 1995. A neural network-based method for tracking features from satellite sensor images. *International Journal of Remote Sensing* 16: 3695–3701.

Crockford, R.H. & Richardson, D.P. 1998. Litterfall, litter and associated chemistry in a dry sclerophyll eucalypt forest and a pine plantation in south-eastern Australia: 2. Nutrient recycling by litter, throughfall and stemflow. *Hydrological Processes* 12: 385–400.

Curran, P.J., Milton, E.J., Atkinson, P.M. & Foody, G.M. 1998. Remote sensing: from data to understanding. In P.A. Longley, S.M. Brooks, R. McDonnell & W. MacMillan (Eds.), *Geocomputation: A Primer* (pp. 33–59) Chichester: John Wiley.

Dawson, M.S., Fung, A.K. & Manry, M.T. 1993. Surface parameter retrieval using fast learning neural networks. *Remote Sensing Reviews* 7: 1–18.

DeCoursey, D.G. 1988. A critical assessment of hydrological modelling. *Modelling Agricultural, Forestry and Rangeland Hydrology.* ASAE Publication 07-88 American Society of Agricultural Engineers, St Joseph Michigan, 478–493.

Dunn, S.M., McAlister, E. & Ferrier, R.C. 1998. Development and application of a distributed catchment-scale hydrological model for the River Ythan N.E. Scotland. *Hydrological Processes* 12: 401–416.

Dymond, J.R. 1993. An improved Skidmore/Turner classifier. *Photogrammetric Engineering and Remote Sensing* 59: 623–626.

Ekstrand, S. 1996. Landsat TM-based forest damage assessment: correction for topographic effects. *Photogrammetric Engineering and Remote Sensing* 62: 151–161.

Engman, E.T. & Gurney, R.J. 1991. *Remote Sensing in Hydrology.* London: Chapman and Hall.

Fahlman, S.E. 1988. Faster-learning variations of backpropagation: an empirical study. *Proceedings of the 1988 Connectionist Models Summer School* (pp. 38–51). San Mateo, CA: Morgan Kaufmann.

Fischer, M.M. 1998. Computational neural networks a new paradigm for spatial analysis. *Environment and Planning A* 30: 1873–1891.

Fischer, M.M., Gopal, S., Staufer, P. & Steinnocher, K. 1997. Evaluation of neural pattern classifiers for a remote sensing application. *Geographical Systems* 4: 195–225.

Foody, G.M. 1996. Approaches for the production and evaluation of fuzzy land cover classifications from remotely-sensed data. *International Journal of Remote Sensing* 17: 1317–1340.

Foody, G.M. 1997. Fully fuzzy supervised classification of land cover from remotely sensed imagery with an artificial neural network. *Neural Computing and Applications* 5: 238–247.

Foody, G.M. 1999a. Image classification with a neural network: from completely-crisp to fully-fuzzy situations. In P.M. Atkinson & N. J. Tate (Eds.), *Advances in Remote Sensing and GIS.* Chichester: John Wiley.

Foody, G.M. 1999b. The continuum of classification fuzziness in thematic mapping. *Photogrammetric Engineering and Remote Sensing* 65: 443–451.

Foody, G.M. 2000. Mapping land cover from remotely sensed data with a softened feedforward neural network classification. *Journal of Intelligent and Robotic Systems* 29: 433–449.

Foody, G.M. & Boyd, D.S. 1999. Fuzzy mapping of tropical land cover along an environmental gradient from remotely sensed data with an artificial neural network. *Journal of Geographical Systems* 1: 23–35.

Foody, G.M. & Boyd, D.S. 2002. Sharpened mapping of tropical forest biophysical properties from coarse spatial resolution satellite sensor data. *Neural Computing and Applications* 11: 62–70.

Foody, G.M., Boyd, D.S., & Cutler, M.E.J. 2003. Predictive relations of tropical forest biomass from Landsat TM data and their transferability between regions. *Remote Sensing of Environment* 85: 463–474.

Foody, G.M. & Curran, P.J. 1994. Scale and environmental remote sensing. In G.M. Foody & P.J. Curran (Eds.), *Environmental Remote Sensing at Regional to Global Scales* (pp. 223–232). Chichester: John Wiley.

Foody, G.M. & Cutler, M.E.J. 2003. Tree biodiversity in protected and logged Bornean tropical rain forests and its measurement by satellite remote sensing. *Journal of Biogeography* 30: 1053–1066.

Foody, G.M., Cutler, M.E., McMorrow, J., Pelz, D., Tangki, H., Boyd, D.S. & Douglas, I. 2001. Mapping the biomass of Bornean tropical rain forest from remotely sensed data. *Global Ecology and Biogeography* 10: 379–387.

Foody, G.M., McCulloch, M.B. & Yates, W.B. 1995. Classification of remotely sensed data by an artificial neural network: issues related to training data characteristics. *Photogrammetric Engineering and Remote Sensing* 61: 391–401.

Foody, G.M., Lucas, R.M., Curran, P.J. & Honzak, M. 1997. Non-linear mixture modelling without end-members using an artificial neural network. *International Journal of Remote Sensing* 18: 937–953.

Frankenberger, J.R., Brooks, E.S., Walter, M.T., Walter, M.F. & Steenhuis, T.S. 1999. A GIS-based variable source area hydrology model. *Hydrological Processes* 13: 689–700.

Garatuza-Payan, J., Shuttleworth, W. J., Encinas, D., McNeil, D. D., Stewart, J. B., deBruin, H. & Watts, C. 1998. Measurement and modelling of evaporation for irrigated crops in north-west Mexico. *Hydrological Processes* 12: 1397–1418.

Gershon, N.D. & Miller, C.G. 1993. Dealing with the data deluge. *IEEE Spectrum* 30(7): 28–32.

Harvey, K.D. & Solomon, S.I. 1984. Satellite remotely-sensed land-use land-cover data for hydrological modelling. *Canadian Journal of Remote Sensing* 10: 68–91.

Held, A., Ticehurst, C., Lymburner, L. & Williams, N. 2003. High resolution mapping of tropical mangrove ecosystems using hyperspectral and radar remote sensing. *International Journal of Remote Sensing* 24: 2739–2759.

Hillard, U., Sridhar, V., Lettenmaier, D.P. & McDonald, K.C. 2003. Assessing snowmelt dynamics with NASA scatterometer (NSCAT) data and a hydrologic process model. *Remote Sensing of Environment* 86: 52–69.

Hoshi, T., Uchida, S. & Kotoda, K. 1989. Development of a system to estimate evapotranspiration over complex terrain using Landsat MSS, elevation and meteorological data. *Hydrological Sciences Journal* 4: 635–649.

Jakeman, A.J., Green, T.R., Beavis, S.G., Zhang, L., Dietrich, C.R. & Crapper, P.F. 1999. Modelling upland and instream erosion, sediment and phosphorus transport in a large catchment. *Hydrological Processes* 13: 745–752.

Jensen, J.R. & Chery, D.L. 1980. Landsat crop identification for watershed balance determinations. *International Journal of Remote Sensing* 4: 345–359.

Jiang, X., Chen, M.-S., Manry, M.T., Dawson, M.S. & Fung, A.K. 1994. Analysis and optimization of neural networks for remote sensing. *Remote Sensing Reviews* 9: 97–114.

Johnston, R.J. 1968. Choice in classification: the subjectivity of objective methods. *Annals of the Association of American Geographers* 58: 575–589.

Kang, S., Park, J.I. & Singh, V.P. 1998. Effect of urbanization on runoff characteristics of the on-Cheon Stream watershed in Pusan, Korea. *Hydrological Processes* 12: 351–363.

Kanellopoulos, I., Varfis, A., Wilkinson, G.G. & Megier, J. 1992. Land-cover discrimination in SPOT HRV imagery using an artificial neural network – a 20-class experiment. *International Journal of Remote Sensing* 13: 917–924.

Kavzoglu, T. & Mather, P.M. 2002. The role of feature selection in artificial neural network applications. *International Journal of Remote Sensing*, 23, 2919–2937.

Kavzoglu, T. & Mather, P.M. 2003. The use of backpropagating artificial neural networks in land and classification. *International Journal of Remote Sensing*, 24, 4907–4938.

Kent, M.K., Gill, W.J., Weaver, R.E. & Armitage, R.P. 1997. Landscape and plant community boundaries in biogeography. *Progress in Physical Geography* 21: 315–353.

Kent, J.T. & Mardia, K.V. 1988. Spatial classification using fuzzy membership models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10: 659–671.

Klein, A.G. & Barnett, A.C. 2003. Validation of daily MODIS snow cover maps of the Upper Rio Grande River Basin for the 2000–2001 snow year. *Remote Sensing of Environment* 86: 162–176.

Knox, J.W. & Weatherfield, E.K. 1999. The application of GIS to irrigation water resource management in England and Wales. *The Geographical Journal* 165: 90–98.

Lacaze, R., Donier, S., Lacarrere, P. & Roujean, J.L. 2003. AVHRR-derived land surface conditions for flux simulations with a mesoscale model over the HAPEX-Sahel study area. *Journal of Applied Meteorology* 42: 686–700.

Lillesand, T.M. & Kiefer, R.W. 2000. *Remote Sensing and Image Interpretation* (4th ed.). New York: Wiley.

Mackaya, D.S., Samanta, S., Nemani, R.R. & Band L.E. 2003. Multi-objective parameter estimation for simulating canopy transpiration in forested watersheds. *Journal of Hydrology* 277: 230–247.

Mannan, B. & Ray, A.K. 2003. Crisp and fuzzy competitive learning networks for supervised classification of multispectral IRS scenes. *International Journal of Remote Sensing* 24: 3491–3502.

Manry, M.T., Dawson, M.S., Fung, A.K., Apollo, S.J., Allen, L.S. & Lyle, W.D. 1994. Fast training of neural networks for remote sensing. *Remote Sensing Reviews* 9: 77–96.

Mather, P.M. 1999. *Computer Processing of Remotely-Sensed Images* (2nd ed.). Chichester: Wiley.

Meiner, A. 1996. Evaluation of AVHRR-based land cover data as input for regional modelling of nutrient load to the Baltic Sea. In K. Kovar & H.P. Nachtnebel (Eds.), *Application of Geographic Information Systems in Hydrology and Water Resource Management* (pp. 629–635). IAHS Publication 235, International Association of Hydrological Science: Wallingford.

Melone, F., Corradini, C. & Singh, V.P. 1998. Simulation of the direct runoff hydrograph at basin outlet. *Hydrological Processes* 12: 769–779.

Meyer, P., Itten, K.I., Kellenberger, T., Sandmeir, S. & Sanmier, R. 1993. Radiometric corrections of topographically induced effects on Landsat TM data in Alpine environments. *ISPRS Journal of Photogrammetry and Remote Sensing* 48: 17–28.

Middlekoop, H. & Janssen, L.F. 1991. Implementation of temporal relationships in knowledge based classification of satellite images. *Photogrammetric Engineering and Remote Sensing* 57: 937–945.

Murnion, S.D. 1996. Comparison of back propagation and binary diamond neural networks in the classification of a Landsat TM image. *Computers and Geosciences* 22: 995–1001.

Neumann, P. & Schultz, G.A. 1989. Hydrological effects of catchment characteristics and land use changes determined by satellite imagery and GIS. In A. Rango (Ed.), *Remote Sensing and Large-Scale Global Processes* (pp. 169–176). IAHS Publication 186, International Association of Hydrological Sciences: Wallingford.

Oslin, A.J., Westsmith, R.A. & Morgan, D.S. 1988. STREAMS: A basin and soil erosion model using CADD, remote sensing and GIS to facilitate watershed management. *Modelling Agricultural, Forestry and Rangeland Hydrology*, ASAE Publication 07-88 American Society of Agricultural Engineers: St Joseph Michigan, 470–477.

Paola, J.D. & Schowengerdt, R.A. 1995. A detailed comparison of backpropagation neural network and maximum likelihood classification for urban

land use classification. *IEEE Transactions on Geoscience and Remote Sensing* 33: 981–996.

Peddle, D.R. 1993. An empirical comparison of evidential reasoning, linear discriminant analysis, and maximum likelihood algorithms for alpine land cover classification. *Canadian Journal of Remote Sensing* 19(1): 31–44.

Peddle, D.R., Foody, G.M., Zhang, A., Franklin, S.E. & LeDrew, E.F. 1994. Multisource image classification II: An empirical comparison of evidential reasoning and neural network approaches. *Canadian Journal of Remote Sensing* 20(4): 397–408.

Pierce, L.E., Sarabandi, K. & Ulaby, F.T. 1994. Application of an artificial neural network in canopy scattering inversion. *International Journal of Remote Sensing* 15: 3263–3270.

Roli, F., Giacinto, G. & Vernazza, G. 1997. Comparison and combination of statistical and neural network algorithms for remote-sensing image classification. In I. Kanellopoulos, G.G. Wilkinson, F. Roli, & J. Austin (Eds.), *Neuro-computation in Remote Sensing Data Analysis.* (pp. 117–124). Berlin: Springer.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. 1986 Learning internal representation by error propagation. In D.E. Rumelhart & J.L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (pp. 318–362). Cambridge, MA: MIT Press.

Salu, Y. & Tilton, J. 1993. Classification of multispectral image data by the binary diamond neural network and by non-parametric, pixel-by-pixel methods. *IEEE Transactions on Geoscience and Remote Sensing* 31: 606–617.

Schalkoff, R.J. 1992. *Pattern Recognition: Statistical, Structural and Neural Approaches*. New York: Wiley.

Schultz, G.A. 1996. Remote sensing and GIS from the perspective of hydrological systems and process dynamics. In K. Kovar & H.P. Nachtnebel (Eds.), *Application of Geographic Information Systems in Hydrology and Water Resource Management* (pp. 637–647). IAHS Publication 235, International Association of Hydrological Science: Wallingford.

Schweiger, A.J. & Key, J.R. 1997. Estimating surface radiation fluxes in the Arctic from TOVS brightness temperatures. *International Journal of Remote Sensing* 18: 955–970.

Sharma, K.D. & Singh, S. 1992. Runoff estimation using Landsat Thematic Mapper data and the SCS model. *Hydrological Sciences Journal* 37: 39–52.

Simpson, G. & Li, K. 1993. *Artificial Neural Networks: Solutions to Problems in Remote Sensing*. Report to DRA Farnborough on contract FRN1b/150, Earth Observation Sciences: Farnham.

Smith, M.J., Moore, T. & Dumville, M. 1995. The performance of neural nets in image rectification. *RSS'95 Remote Sensing in Action*. Remote Sensing Society, Nottingham, 11–18.

Staufer, P. & Fischer, M.M. 1997. Spectral pattern recognition by a two-layer perceptron: effects of training set size. In I. Kanellopoulos, G.G. Wilkinson, F. Roli & J. Austin (Eds.), *Neuro-computation in Remote Sensing Data Analysis* (pp. 105–116) Berlin: Springer.

Storck, P., Bowling, L., Wetherbee, P. & Lettenmaier, D. 1998. Application of a GIS-based distributed hydrology model for prediction of forest harvest effects on peak stream flow in the Pacific Northwest. *Hydrological Processes* 12: 889–904.

Takara, K. & Kojima, T. 1996. GIS-aided land cover classification assessment based on remote sensing images with different spatial resolution. In K. Kovar & H.P. Nachtnebel (Eds.), *Application of Geographic Information Systems in Hydrology and Water Resource Management* (pp. 659–665). IAHS Publication 235, International Association of Hydrological Science: Wallingford.

Tom, C.H. & Miller, L.D. 1984. An automated land-use mapping comparison of the Bayesian maximum likelihood and linear discriminant analysis algorithms. *Photogrammetric Engineering and Remote Sensing, 50*, 193–207.

Townshend, J.R.G. 1981. The spatial resolving power of earth resources satellites. *Progress in Physical Geography* 5: 32–55.

Tsihrintzis, V.A. & Hamid, R. 1998. Runoff quality prediction for small urban catchments using SWMM. *Hydrological Processes* 12: 311–329.

Tso, B. & Mather, P.M. 2001. *Classification Methods for Remotely-sensed Data*, London: Taylor and Francis.

Uchida, S. & Hoshi, T. 1988. On the regional characteristics of actual evapotranspiration derived from Landsat MSS and elevation data. *Geocarto International* 2: 57–66.

Verstraete, M.M., Pinty, B. & Myeni, R.B. 1996. Potential and limitations of information extraction on the terrestrial biosphere from satellite remote sensing. *Remote Sensing of Environment* 58: 201–214.

Visa, A. & Peura, M. 1997. Generalisation of neural network based segmentation results for classification purposes. In I. Kanellopoulos, G.G. Wilkinson, F. Roli & J. Austin (Eds.), *Neuro-computation in Remote Sensing Data Analysis* (pp. 255–261). Berlin: Springer.

Walker, N.P., Eglen, S.J. & Lawrence, B.A. 1994. Image compression using neural networks. *GEC Journal of Research* 11: 66–75.

Walling, D.E. 1981. Hydrological processes. In K.J. Gregory & D.E. Walling (Eds.), *Man and Environmental Processes* (pp. 57–81). London: Butterworths.

Wang, F. 1990. Fuzzy supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 28: 194–201.

Wang, Y. & Dong, D. 1997. Retrieving forest stand parameters from SAR backscatter data using a neural network trained by a canopy backscatter model. *International Journal of Remote Sensing* 18: 981–989.

Warrender, C.E. & Augusteijn, M.F. 1999. Fusion of image classifications using Bayesian techniques with Markov random fields. *International Journal of Remote Sensing* 20: 1987–2002.

Watson, F.G.R., Vertessy, R.A. & Grayson, R.B. 1999. Large scale modelling of forest hydrological processes and their long-term effect on water yield. *Hydrological Processes* 13: 689–700.

Wilkinson, G.G. 1996. Classification algorithms – where next? In E. Binaghi, P.A. Brivio & A. Rampini (Eds.), *Soft Computing in Remote Sensing Data Analysis* (pp. 93–99). Singapore: World Scientific.

Wilkinson, G.G. 1997. Open questions in neurocomputing for Earth observation, In I. Kanellopoulos, G.G. Wilkinson, F. Roli & J. Austin (Eds.), *Neuro-computation in Remote Sensing Data Analysis* (pp. 3–13). Berlin: Springer.

Wilkinson, G.G., Fierens, F. & Kanellopoulos, I. 1995. Integration of neural and statistical approaches in spatial data classification. *Geographical Systems* 2: 1–20.

Woodcock, C.E. & Gopal, S. 1995. Remote sensing of forests: new data layers for GIS. *Proceedings 1995 ACSM/ASPRS Annual Convention and Exposition*, (ACSM/ASPRS, Bethesda MD), 2: 420–428.

Woube, M. 1999. Flooding and sustainable land-water management in the lower Baro-Akobo river basin, Ethiopia. *Applied Geography* 19: 235–251.

Yu, S. & Weigl, K. 1997. A hybrid method for preprocessing and classification of SPOT images, In I. Kanellopoulos, G.G. Wilkinson, F. Roli & J. Austin (Eds.), *Neuro-computation in Remote Sensing Data Analysis* (pp. 141–143). Berlin: Springer.

Zhang, J. & Foody, G.M. 2001. Fully-fuzzy supervised classification of sub-urban land cover from remotely sensed imagery: statistical and artificial neural network approaches. *International Journal of Remote Sensing* 22: 615–628.

Zhang, M. & Scofield, R.A. 1994. Artificial neural network techniques for estimating heavy convective rainfall and recognising cloud mergers. *International Journal of Remote Sensing* 15: 3241–3261.

Zhuang, X., Engel, B.A., Lozano-Garcia, D.F., Fernandez, R.N. & Johannsen, C.J. 1994. Optimisation of training data required for neuro-classification. *International Journal of Remote Sensing* 15: 3271–3277.

15

Towards a Hydrological Research Agenda

ROBERT J. ABRAHART
School of Geography, University of Nottingham, UK

PAULINE E. KNEALE AND LINDA M. SEE
School of Geography, University of Leeds, UK

ABSTRACT: The use of artificial neural networks in hydrology is on the increase. This data-driven technology, along with other alternative methods, can be used to develop models that are capable of delivering superior forecasts and predictions in comparison to traditional hydrological models. However, we are a long way from understanding the optimal models that relate to specific circumstances and pertinent model development issues. In this chapter a research agenda for the next decade is discussed including the need to: continue improving existing neural network models, further compare neural network models with operational and process-based solutions, develop and assess meaningful evaluation criteria, improve model understanding and build dedicated hydrological neural network software packages.

1 A RESEARCH AGENDA

Artificial neural networks (NN) are now commonly used in many fields, ranging from social sciences to engineering, where in comparison to more conventional statistical or theoretical approaches, such tools are providing better and/or faster solutions to different types of problem. Despite this overall success, the uptake of NN technologies in the hydrological sciences has been much slower although we are now witnessing an increasing momentum in the reported use of NN and other AI technologies. Throughout this book there are clear illustrations of areas in which hydrological science could benefit from the application of NN including: rainfall-runoff modelling (Chapters 3 to 9); rainfall forecasting (Chapter 10); water quality prediction (Chapter 11); sediment modelling (Chapter 12); and applied remote sensing (Chapters 13 and 14). The examples presented in

these chapters all demonstrate that neurocomputing can produce models of similar or superior performance, but to date there is little evidence that the available technologies are being transferred into an operational environment, which is the next crucial step forward, if the widespread potential of these tools is to be successful. To facilitate a process of technological transfer, neurohydrologists will need to pursue a research agenda that continues to tackle questions about improvements in modelling mechanics, and also address the concerns of traditional hydrologists who prefer to use process-based models and to exclude data-driven approaches. This chapter suggests five general directions in which research in the next decade could be pursued:

- improvement to existing NN hydrological models including the investigation of current NN hydrological problems;
- more emphasis on their comparison with operational and process-based models;
- the development and construction of more powerful and more efficacious modelling evaluation criteria;
- further research into understanding their internal workings and the real-world meaning of each component; and
- the construction of integrated and hybrid NN software for hydrological research.

Each of these items is discussed in the sections that follow.

2 CONTINUE TO IMPROVE EXISTING NN MODELS

This is a massive area of research that ranges from experimentation with different NN algorithms and architecture to the development of hybrid models and dynamic solutions.

2.1 *Algorithms and architectures*

The multi-layer perceptron trained using backpropagation is the default situation and the most common tool that is used in all areas of application. Some initial research into alternative algorithms and architectures is however occurring in the area of rainfall-runoff modelling: see for example Chapter 5 on Time Delay Neural Networks (TDNN), Chapter 6 on Cascading Correlation Neural Networks (CCNN) and Chapter 7 on Partial Recurrent Neural Networks (PRNN). There is also a vast, theoretical literature on other types of network which remains an area for further research, e.g. second order NN methods (Shepherd, 1997).

Another important area for research is the fact that there are no fixed rules for the design of an individual solution and that it is not possible to establish an appropriate model *a priori*. It is likewise impossible to obtain subsequent confirmation that the final architecture is the optimum one (Kanellopoulos *et al.*, 1992). The problem of no theoretical support persists throughout all areas of design; even the most basic matters must be determined through experimentation. Most practitioners use intuition or trial and error assessment procedures. Even sophisticated

automated search procedures will suffer from comparable problems since different algorithms can produce different answers (for more details see Maier and Dandy (2000, pp. 106–108)). Empirical testing is thus paramount and some first steps in this direction, that relate internal functions to conceptual model processes in which the number of required hidden nodes is matched against specific mechanical 'anchors', is illustrated in Wilby *et al.* (2003).

2.2 *Learning regimes*

Training is the process of optimising connection weights in the search for an appropriate level of fit. It is not possible to determine the best course of action *a priori*. There are no fixed rules about: the selection and implementation of an appropriate learning algorithm; the number, distribution, or format of the training patterns; or the manner in which the data are presented. The training process must also be considered in relation to the selection of an appropriate architecture and such decisions are further complicated with the problem that it is possible to overfit the training data. Trial and error, plus regular monitoring with alternative data sets, is again the best advice to date (Sarle, 2000).

Most NN hydrological forecasting and prediction has involved modelling combinations of traditional rainfall-runoff variables in a supervised learning environment. However, various problems can arise from skewed data distributions or weak deterministic hydrological relationships, and the supervised learning model thus puts an arduous burden on the learning mechanism since it requires a complete set of all possible input-output situations. It is therefore suggested that the information content, on either the input or the output side of the modelling function, must be increased if these tools are to achieve improved performance. It is in this respect that alternative relationships derived from the original data could also perhaps be used to help build superior modelling solutions. Reed and Marks (1999) have classified potential addition and replacement strategies for improvements to the supervised training model into 'hints' and 'distal learners':

- Supervised learning with hints is where extra output nodes are added to the network, which is equivalent to training the network to learn additional functions. These hint functions should be related to the main function of interest and should be easier to learn. Their use could create non-zero derivatives in regions where the original function has plateaued and thus help to speed-up convergence. The provision of additional constraints could also control the final solution under problematic circumstances or help overcome difficulties associated with local minima in the underlying relationship.
- Supervised learning with a distal teacher, is where the outputs from a network act as inputs to another system, which then transforms the network output into the final output. The overall modelling procedure is thus split into a two-stage operation. The first part is used to predict simple intermediate concepts and the second part then transforms these output values into the required outputs. The potential to achieve significant gains exists when the intermediate function is easier to learn and the final transformation is based on an exact statistical, mathematical or trigonometric function.

Some hydrological examples of addition and replacement strategies that have been suggested in the past include: the addition of binary inputs to mark the start and end of rainfall (Hall & Minns, 1993); the introduction of additional outputs to help preserve or monitor global features in the overall structure or particular aspects of the data set (French *et al*., 1992); and conversion of the predicted data into a set of alternative representations that are better items to model such as the use of Fourier Series parameters instead of actual hydrological variables (Smith & Eli, 1995).

Yet another problem for investigation is the fact that it is not possible to establish confidence intervals for each coefficient; t-values for example are meaningless in the case of a non-linear solution. The recommended method for testing a neural solution is empirical and involves validation against one or more sets of unseen test data (for more details see Maier and Dandy (2000, p. 112)). This process demands unbiased datasets that are difficult to construct, and provides a test for overfitting not a measure of trustworthiness. Other assessment mechanisms include: an examination of residuals (Hsu *et al.*, 1995; Lek *et al*., 1996); and the plotting of error bounds (Hwang & Ding, 1997; Whitley & Hromadka, 1999).

2.3 *Multi-modelling and hybrid modelling solutions*

Most NN hydrological modelling applications have to date involved the construction of a single standalone NN solution. Yet there exists a growing realisation that more significant applications can be facilitated through the development of multi-network solutions, in the form of serial and parallel combinations, as illustrated in Chapter 4. Sharkey (1999) has also argued that multi-neural applications would in most cases provide improved performance over single-neural modelling solutions. Indeed, compound solutions could be developed to perform tasks that cannot be modelled with a single solution, and a modular combination of neural components could be used to resolve difficult problems in a more effective manner. Moreover, better overall performance and higher levels of skill, can often be achieved through an efficacious mix of unstable predictors which are organised in the form of a 'redundant combination'. There are two main approaches that could be investigated: ensemble combinations and modular combinations.

The term 'ensemble' is used to describe a combination of redundant networks; other terms for this form of combination include 'committee' and 'committee machine'. In an ensemble combination the component networks are redundant in that each network provides a solution to the same task, or problem, or task component. The final result is then amalgamated or averaged using a suitable method of numerical combination. The motivation behind this approach is to obtain improved generalisation capabilities and thus guard against the failure of each individual component solution. Individual components could for instance be developed using variation in: initial random weights, architectures, training algorithm, or data sampling. Empirical investigation into the effectiveness of different methods of ensemble creation suggests that variation in the training

data has the greatest potential for producing networks that make different errors (Sharkey & Sharkey, 1995; Sharkey *et al.*, 1996; Tumer & Ghosh, 1996; Cigizoglu, 2003).

In the modular approach the task or problem is decomposed into a number of sub-tasks and the complete task solution requires the contribution of several modules. There are various motivations for taking a modular approach, some of the most important are to:

- use decomposition to improve performance i.e. no monolithic solutions; use of more appropriate modules or blend of modules; a mixture of experts approach;
- enable switching of control from one solution to another;
- use sub-problems to extend the capabilities of a single solution;
- permit recombination of data from different sources and partial representations;
- build less complex solutions that are easier to understand;
- develop modular combinations that can be modified and extended with minimal effort;
- use individual components that are quicker to train;
- create modular solutions that will permit the incorporation of prior knowledge that is implemented in terms of an appropriate decomposition.

Further development and testing is required of hybrid models that are derived from an integration of neurocomputing tools with other 'smart technologies' or conventional and traditional techniques. There is a growing realisation in the field of intelligent systems (Teodorescu, 2000) that complex problems will require hybrid solutions, derived from an amalgamation of diverse tools and methods, originating from traditional mathematics (e.g. dynamical systems; linear algebra; symbolic computation) and computer science (e.g. logic and theorem proving; cellular automata; graph algorithms). There is also a marked trend towards combining genetic algorithms, fuzzy logic, NN and expert systems. Each individual technique has specific computational properties that are well suited to a particular problem but not to others. Two or more techniques are thus often combined in such a manner as to overcome the limitations of an individual technique. Thus hybrid systems are also important when considering the complex and varied nature of application domains wherein different component problems, or associated sub-problems, might require different types of processing or different forms of solution.

Hybrid formulations could range from the provision of large-scale multi-model unions to the simple insertion of direct replacement parts. Two examples will be provided in support of the general case. Shamseldin and O'Connor (2001) have demonstrated that a neural solution can be coupled to a conceptual model in the manner of a hydrological model-output updating procedure, with proven benefits, for use in real time forecasting applications. Dssanayake and Phan-Thien (1994) have described a neural method for solving partial differential equations that could be adopted in process-based distributed modelling applications. The mathematical solution of problems associated with continuum mechanics, are well developed and well understood, but the numerical implementation of such methods is seldom straightforward. Yet it is often the case that a quick and

(within-reason) accurate solution can be achieved from a neural approximation, that can deliver an acceptable set of independent outputs, with minimal user effort under either linear or non-linear conditions.

NN can also be used to perform data fusion operations on the modelling outputs that are produced from an independent set of individual neural forecasters, e.g. from an ensemble combination of or mixed neural and conventional hydrological forecasters (Abrahart & See, 2002). Data fusion in this context describes the process of combining or amalgamating information from multiple sensors, data sources or modelling outputs and can involve serial, parallel, or mixed strategies of combination. The principal objective of data fusion is to provide a solution that is either more accurate in some form or which allows one to make additional inferences above and beyond those which could be achieved through the use of single source data alone. Data fusion can also operate at more complicated feature-based or decision-based levels, using any combination of input types, to produce either a numerical output, a feature output, or a higher level decision. Decisions from individual sensors or sources can also be fused into a higher-level decision in the manner of an expert system; see Dasarathy (1994, 1997).

2.4 *Dynamic modelling*

Further development and testing is required of dynamic tools that are able to model complex chaotic systems. Most hydrological modelling has assumed that it is possible to describe the rainfall-runoff transformation using a finite number of basic operators rather than attempting to explain the total process as a complex whole. It is likewise assumed that: (i) the process is deterministic or stochastic; (ii) small distortions in the data can have little or no effect on the power of the model to predict; and (iii) problems associated with modelling and calibration issues will disappear with improved access to greater volumes of more and better catchment data derived from automated recording devices. However, such beliefs are not consistent with the concept of chaos, or the idea that simple nonlinear deterministic systems can at times behave in what appears to be an unpredictable and chaotic manner (Gleick, 1987; Stewart, 1997). Extreme or irregular behaviour can indeed arise from a simple deterministic system, which contains a small number of non-linear inter-dependent variables, wherein small initial distortions will under certain circumstances develop in an exponential manner.

The term ‘chaos’ is used in a formal sense to describe the complicated behaviour of a non-linear dynamic system, that is a self-organised process, in which the next state of the system can be expressed as a non-linear function of previous states. The phase space of a chaotic system is the feature space in which the process involved is traced over time, such that a chaotic process goes around areas or points that are situated in phase space, termed chaotic attractors. There is no repetition of the same trajectories and the system can evolve in continuous or discrete time. It can embrace stretches and contractions of the phase space and it is sensitive to initial conditions such that different initial states will produce different trends and outcomes. Sivakumar (2000) discusses the hydrological implications of chaotic behaviour that is observed in rainfall (e.g. Sivakumar *et al.*,

1999) and runoff (e.g. Krasovskaia *et al*., 1999) time series data. These findings suggest that a similar situation will exist for the rainfall-runoff transformation process and should therefore be incorporated within the model building process (albeit that a recent attempt to determine the existence of this chaotic process was inconclusive (Sivakumar *et al*., 2001)). Kasobov (1996, pp. 486–493) lists several different types of NN that could be used to model nonlinear dynamical systems: (i) recurrent networks, with feedback connections, can be used to learn short-term time-dependencies e.g. using input neurons that could feed their output back to themselves – thus producing a response to a given input that did not occur the first time that input was presented; (ii) connectionist models, based on chaotic neurons, in which the behaviour of a neuron depends on its external stimulus, or connectionist models with oscillators, in which functional units comprise two more neurons that act as excitors or inhibitors and model frequencies, phases or amplitudes; and (iii) spatio-temporal solutions in which time is treated as a variable. For other possibilities see Zaldivar *et al*. (2000) and Giustolisi (2000).

3 FURTHER COMPARISON WITH SOLUTIONS OPERATIONAL AND PROCESS-BASED SOLUTIONS

Extensive exploration and reporting is required to establish the exact relationship between neural solutions and traditional or conventional hydrological models. Such studies must provide a detailed account of each reported investigation and contain all model-related facts and decisions. The main focus of recent comparisons has been with statistical time series predictors and conceptual models, for example, Tingsanchali and Gautam (2000) compare two lumped conceptual models and a statistical solution with a NN forecaster. However, the range and scope of neural applications must be extended, to include a consideration of more complex problems and undertakings and to make comparisons against state-of-the-art distributed process modelling operations.

Further studies should also be directed towards the emulation of existing hydrological mechanisms, either in part or in full, since there are clear benefits to be had in terms of a speed-up (see Abebe *et al*., 2000; Gautam, 1998; Khindker *et al*., 1998; Liu & Lin, 1998). This would be of immense value with respect to real-time forecasting, large scale processing of detailed satellite data or for long term simulation runs e.g. MEDRUSH (Abrahart *et al*., 1996) and SIBERIA (Willgoose & Riley, 1998; Willgoose, 2000). Most neural applications involve less than 100 neurons and require a modest amount of training; software simulation in most situations will therefore be sufficient. However, more powerful mechanisms will be required to meet anticipated future demands, which must at some point proliferate to include useful things that can be done with 1000s of processing elements and 10000's of weighted connections. This will in turn herald a need for high performance hardware, in the form of dedicated neurocomputing platforms and hard-wired machines, which is a specialist area that will require substantial research into the technologies involved; such equipment is at present

restricted to a limited number of specialist areas that demand high performance computing e.g. high energy physics.

Sorooshian (1991) states that most hydrological models suffer from (i) an assumption that the dynamics of a watershed can be represented using a lumped approach and point-based data and that (ii) great inaccuracies occur when such models are applied to situations that possess significant differences in space-time scales. NN solutions should therefore be developed at different scales of spatial, temporal or process-based investigation; to provide cross-scale models that can be used to perform macro-unions for up-scaling or down-scaling operations. NN, based on multiple catchments, could also be used to provide evidence that improved levels of modelling can be achieved at different levels of process-based or spatio-temporal generalisation. Moreover, important environmental associations that suffer from poor conventional modelling or an insufficient theoretical foundation, might perhaps be modelled in a more accurate or more efficacious manner than was hitherto possible. Schaap and associates (Schaap & Bouten, 1996; Schaap & Leij, 1998; Schaap *et al*., 1998; Schaap *et al*., 1999; Lebron *et al.*, 1999) provide several classic examples on the prediction of soil water properties. This field of research is characterised with small data sets, widespread disparities, and problematic hysterisis loops; for NN modelling of loop-rating hysteresis curves, see Jain and Chalisgaonkar (2000). The use of fuzzification methods and neural modelling concepts to overcome sediment transfer prediction problems associated with small data sets has also been demonstrated; see Abrahart and White (2001).

4 DEVELOPMENT OF MORE MEANINGFUL EVALUATION CRITERIA AND ASSOCIATED HYDROLOGICAL MODELLING BENCHMARKS

The calculation of dimensionless indices to assess model skill is still the norm: descriptive statistics based on sum-squared error or similar measures are computed to provide an assessment of global or seasonal goodness-of-fit. However, conventional indices will tend to emphasise a limited number of features in the data, such that specific items of interest with respect to the proposed application will often be masked. Hall (2001) investigated volume, bias and timing issues. He commented that the preferred solution would be to use a series of different measurement criteria (ten were suggested) that focused on the more important aspects of model behaviour – as opposed to placing total reliance on a single index. Gupta *et al*. (1998) (see also Yapo *et al*., 1998) have also proposed a multi-criteria calibration procedure in which several different objectives could be satisfied and this approach might be adopted in a similar manner for model evaluation, as opposed to calibration, purposes. Model output could thus be assessed on the basis of multiple measures and different features, or different periods, using a dedicated interactive analytical toolbox that still awaits development. The hydrographic record can for example be divided into periods with or without rainfall. The rain-free periods might then be further divided into periods dominated by either

throughflow or baseflow processes, and performance assessed for each division e.g. Wagener *et al*. (2000).

The selection and weighting of individual components within a multi-assessment procedure must be application specific and related to fitness-for-purpose; the assessment of a flood model should perhaps emphasise peaks, whilst the assessment of a resources model might perhaps be more directed towards an evalu-ation of low flow sequences. Improved measures and metrics are also needed to extend the scope of the assessment exercise: ranging from hydrological indices related to operational forecasting (e.g. See *et al*. (2001) have suggested the use of alarm levels and the rising limb of the hydrograph) to geometrical analysis based on constructs that are devised to perform an evaluation of line simplification algorithms (e.g. differences in lengths, angles, curvatures, or displacements: for further details see McMaster (1987, 1989)). The list of additional items to be evalu-ated might also be extended to include a consideration of various non-skill-based issues such as: model construction time; data input requirements; error handling capabilities; or temporal adaptation characteristics.

True comparison of reported strategies is often difficult because the reported solutions are developed in different environments. Thus local variations can have a significant impact on model performance. To address this problem the development of better measures and metrics will also require the provision of associated benchmarks and standards. These must be based on shared data sets, which can be used to demonstrate the difference between strong and weak solutions, and thus facilitate subsequent multi-model comparison exercises.

5 IMPROVEMENTS TO MODEL UNDERSTANDING

It is difficult to detect or to understand the internal processes that are occurring within each model, since the information that has been extracted from the original training data, is distributed throughout the NN architecture. Neural solutions provide no justification or explanation for their answers; there are no facilities to match the 'how and why' querying procedures of an expert system. Each user must therefore have confidence in the construction of the NN and its associated modelling outputs. It is nevertheless desirable that each solution should be capable of imparting some form of an explanation, even if it is a partial explanation, as an integral part of its function. Minns (2000) has argued that neural methods can discover usable relations in measured and experimental data with little or no *a priori* knowledge of the governing physical process characteristics. The neural solution cannot perform an explicit identification of the form of a model but such form is nevertheless implicit in the neural structure, being encoded within the distribution of nodes and weights. Each NN solution is thus an electronic knowledge encapsulator, that stores encoded information at the sub-symbolic level, which is difficult to extract in mathematical terms due to the high degree of non-linear complexities that are involved. Direct translation of weighted coefficients into a mathematical equation does not help; even with small architectures such equations are too complicated for direct human comprehension (c.f. genetic programming).

Flood and Kartam (1993, p. 136) demonstrated that the NN output response surface of a NN is the amalgamation of the output response surfaces from each of the hidden neurons, and that analysis of the output from each hidden unit, will thus provide a representation of the function that each unit is performing. Wilby *et al.* (2003) have also shown that the inner workings of a river-level forecaster can be matched against the inner workings of a conceptual model that was cloned with a neural solution. NN internal functions can therefore be extracted and inspected for real-world meaning, in terms of a one-to-one mapping against recognised hydrological processes, along the same lines as those reported for data-based mechanistic modelling of rainfall-runoff in Young (1993, 1998a,b, 1999a,b), Young and Beven (1991, 1994), Young *et al*. (1997) and Lees (2000).

Two other preferred methods of extraction are based on global diagnostics. To perform a meaningful examination of individual weights and connections is problematic because the mutual interaction between each neural component is of critical importance. It would therefore seem prudent to develop further methods of exploration and analysis along similar lines to holistic testing procedures such as:

- sensitivity analysis: which investigates 'the rate of change in one factor with respect to change in another' (McCuen, 1973), and
- saliency analysis: which involves zeroing neural input vectors to determine the effect that a particular input has on the overall modelling process (Abrahart *et al*., 2001).

Other methods include the inspection of connection weights (Maier & Dandy, 1997), and the construction of stereotypical inputs or Hinton Diagrams, i.e. visualization of the weight matrix (Silverman & Dracup, 2000).

Further exploration and analysis of internal components and relationships is required, using both quantitative (i.e. measurement) and qualitative (i.e. visualization) techniques, to assess the influence and possible meaning of specific nodes, weights and topologies. Future research must also aim to extract relevant material and to discover useable relations that will foster better understanding and thus assist future modelling efforts at the development stage:

- to build parsimonious NN modelling solutions.
- and facilitate rapid prototyping of more complex mechanisms.

It is also recommended that the end product should be visualized and interpreted, in contrast to the provision of a detailed numerical description, that is based on the use of global statistics. Laffan (1998) reports that large-scale real time geographic visualisation of nodal output during training, for dynamic assessment purposes, is not a realistic option. However, post-model building visualization can still provide a powerful means of analysis, since it is easier for people to interpret diagrams vis-à-vis all other forms of digital data presentation. The end product does not remove the black box tag but does enable some insight into the mechanism and thus makes each solution a little more transparent to both practitioners and scientists.

6 DEVELOPMENT OF DEDICATED HYDROLOGICAL NN SOFTWARE PACKAGES

Most NN hydrological modelling applications have been developed using commercial packages or in-house programs, that adopt a static approach to modelling, based on the use of standard architectures and algorithms: (i) the data are organised; (ii) the data are imported; and (iii) the model is then developed. Each alternative solution that is to be tested thus requires a fresh start. No neural software package or program at present supplies a customisable hydrological programming interface that could be used to (i) ease the development process or (ii) provide a suitable test bed for rapid application development or for modelling exploration purposes. The present generation of software tools and products is thus unable to provide a high level of interaction commensurate with the demands of a modern data-driven paradigm. It is therefore argued that a bespoke NN software package should be developed, to perform detailed interactive exploration and analysis operations, and tailored to the purposes of hydrological science. This program should be in the form of a dedicated toolbox that would permit practitioners to assemble and test a range of solutions, or to perform basic modifications to a specific application, based on libraries of functions in an integrated and controlled user-empowered environment (cf. GeoVISTA Studio (Gahegan *et al*., 2000)).

Some basic examples of the different sorts of interactive experiment that model builders might wish to perform might include the adoption of alternative data pre-processing transformations. Sarle (2002), for example, explains that in addition to simple linear conversions there are several other recognised procedures which include (i) log transformations that can be used to reduce large gaps between upper outliers e.g. peak flow situations and (ii) logistical transformations that could be used to expand distances within a cluster of similar values e.g. in a parallel hybrid error-updating mechanism based on the prediction of residual data from a traditional model. The other major item that would be useful is an option to construct and incorporate dedicated transfer functions, that can be run or changed at will, without the need for a higher degree in software engineering. Imrie *et al*. (2000) have investigated the use of different output activation functions and the implication is that some functions might be more appropriate than others, under certain circumstances, according to the inherent hydrological properties of a catchment and its data e.g. use of cubic polynomial activation function for river flow prediction. The development of a dedicated neural-hydrological toolbox would also permit the incorporation of several automated mechanisms that would (i) perform a detailed exploration and analysis of internal relationships and (ii) could implement a broad range of appropriate hydrological evaluations.

7 FINAL WORD

NN have much to offer. The opportunities for exploring and forecasting are both numerous and challenging. Hydrological problems are appropriate, the

possibilities are exciting, the software is available, and we encourage hydrologists to continue their involvement.

REFERENCES

Abebe, A.J., Solomatine, D.P. & Venneker, R.G.W. 2000. Application of adaptive fuzzy rule-based models for reconstruction of missing precipitation events. *Hydrological Sciences Journal* 45(3): 425–436.

Abrahart, R.J. & White, S. 2001. Modelling sediment transfer in Malawi: comparing backpropagation neural network solutions against a multiple linear regression benchmark using small data sets. *Physics and Chemistry of the Earth (B)* 26(1): 19–24.

Abrahart, R.J., Kirkby, M.J., McMahon, M.L., Bathurst, J.C., Ewen, J., Kilsby, C.G., White, S.M., Diamond, S., Woodward, I., Hawkes, J.C., Shao, J. & Thornes, J.B. 1996. MEDRUSH – spatial and temporal river-basin modelling at scales commensurate with global environmental change. In K. Kovar & H.P. Nachtnebel (Eds.), *HydroGIS'96: Application of Geographic Information Systems in Hydrology and Water Resources Management.* IAHS Publication No. 235, 47–54.

Abrahart, R.J., See, L. & Kneale, P.E. 2001. Investigating the role of saliency analysis with a neural network rainfall-runoff model. *Computers and Geosciences* 27(8): 921–928.

Cigizoglu, H.K. 2003. Incorporation of ARMA models into flow forecasting by artificial neural networks, *Environmetrics* 14: 417–427.

Dasarathy, B.V. 1994. *Decision Fusion*. Los Alamitos, CA: IEEE Computer Society.

Dasarathy, B.V. 1997. Sensor fusion potential exploitation – Innovative architectures and illustrative applications. *Proceedings of the IEEE* 85: 24–38.

Dssanayake, M.W.M.G. & Phan-Thien, N. 1994. Neural-network based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering* 10: 195–201.

Flood, I. & Kartam, N. 1993. Neural networks in civil engineering. 1: principles and understanding. *Journal of Computing in Civil Engineering* 8(2): 131–148.

French, M.N., Krajewski, W.F. & Cuykendall, R.R. 1992. Rainfall forecasting in space and time using a neural network. *Journal of Hydrology* 137: 1–31.

Gahegan, M., Takatsuka, M. Wheeler, M. & Hardisty, F. 2000. GeoVISTA Studio: a geocomputational workbench. In: *GeoComputation 2000: Proceedings of the Fifth International Conference on GeoComputation, University of Greenwich, 23–25 August 2000.* Manchester: GeoComputation CD-ROM.

Gautam, D.K. 1998. Regressive neural networks for the modelling of time series. Unpublished MSc Thesis, IHE Delft, The Netherlands.

Gleick, J. 1987. *Chaos: Making a New Science.* New York: Viking Penguin.

Giustolisi, O. 2000. Input-output dynamic neural networks simulating inflow-outflow phenomena in an urban hydrological basin. *Journal of Hydroinformatics* 2(4): 269–279.

Gupta, H.V., Sorooshian, S. & Yapo, P.O. 1998. Towards improved calibration of hydrologic models: multiple and noncommensurable measures of information. *Water Resources Research* 34: 751–763.

Hall, M. 2001. How well does your model fit the data? *Journal of Hydroinformatics* 1(3): 49–55.

Hall, M.J. & Minns, A.W. 1993. Rainfall-runoff modelling as a problem in artificial intelligence: experience with a neural network. In *Proceedings Fourth National Hydrological Symposium, Cardiff, 1993*. London: British Hydrological Society, 5.51–5.57.

Hsu, K-L, Gupta, H.V. & Sorooshian, S. 1995. Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research* 31: 2517–2530.

Hwang, J.T.G. & Ding, A.A. 1997. Prediction intervals for artificial neural networks. *Journal of the American Statistical Association* 92: 748–757.

Imrie, C.E., Durucan S. & Korre A. 2000. River flow prediction using artificial neural networks: generalisation beyond the calibration range. *Journal of Hydrology* 233: 138–153.

Jain, S.K. & Chalisgaonkar, D. 2000. Setting up stage-discharge relations using ANN. *Journal of Hydrological Engineering* 5(4): 428–433.

Kanellopoulos, I., Varfis, A., Wilkinson, G.G. & Mégier. 1992. Land-cover discrimination in SPOT HRV imagery using an artificial neural network. *International Journal of Remote Sensing* 13(5): 917–924.

Kasobov, N.K. 1996. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. Cambridge, MA: MIT Press.

Khindker, M.U.L-H., Wilson, G. & Klinting, A. 1998. Application of neural nets in real time flash flood forecasting. In V. Babovic & C.L. Larsen (Eds.), *Hydroinformatics 98: Proceedings Third International Conference on Hydroinformatics, Copenhagen, Denmark, 24–26 August 1998*. Vol. 2, 777–782. Rotterdam: A.A.Balkema.

Krasovskaia, I., Gottschalk, L. & Kundzewicz, Z.W. 1999. Dimensionality of Scandinavian river flow regimes. *Hydrological Sciences Journal* 44(5): 705–723.

Laffan, S.W. 1998. Visualising neural network training in geographic space. In *Proceedings of the Third International Conference on GeoComputation*, University of Bristol, 17–19 September 1998. Manchester: GeoComputation CD-ROM.

Lebron, I., Schaap, M.G. & Suarez, D.L. 1999. Saturated hydraulic conductivity prediction from microscopic pore geometry measurements and neural network analysis. *Water Resources Research* 35(10): 3149–3158.

Lees, M.J. 2000. Data-based mechanistic modelling and forecasting of hydrological systems. *Journal of Hydroinformatics* 2(1): 15–34.

Lek, S., Delacoste, M., Baran, P., Dimopoulos, I., Lauga, J. & Aulagnier, S. 1996. Application of neural networks to modelling nonlinear relationships in ecology. *Ecological Modelling* 90: 39–52.

Liu, C.L. & Lin, S.C. 1998. Artificial neural network forecasting for chaotic dynamical processes. In V. Babovic & C.L. Larsen (Eds.), *Hydroinfomatics*

98: Proceedings of the Third International Conference on Hydroinformatics, Copenhagen, Denmark, 24–26 August 1998. Vol. 2, 791–796. Rotterdam: A.A. Balkema.

Maier, H.R. & Dandy, G.C. 1997. Modelling cyanobacteria (blue-green algae) in the River Murray using artificial neural networks. *Mathematics and Computers in Simulation* 43(3–6): 377–386.

Maier, H.R. & Dandy, G.C. 2000. Neural networks for the prediction of water resources variables: a review of modelling issues and applications. *Environmental Modelling and Software* 15: 101–124.

McCuen, R.H. 1973. The role of sensitivity analysis in hydrologic modelling. *Journal of Hydrology*, 18: 37–53.

McMaster, R.B. 1987. Automated line generalization. *Cartographica* 24(2): 74–111.

McMaster, R.B. 1989. The integration of simplification and smoothing algorithms in line generalization. *Cartographica* 26(1): 101–121.

Minns, A.W. 2000. Subsymbolic methods for data mining in hydraulic engineering. *Journal of Hydroinformatics* 2(1): 3–14.

Reed, R.D. & Marks, R.J. 1999. *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA: MIT Press.

Sarle, W.S. 2002. *FAQ posting for Usenet newsgroup comp.ai.neural-nets*. Last modified: 17/5/2002. URL = ftp://ftp.sas.com/pub/neural/FAQ.html (August 2003).

Schaap, M.G. & Bouten, W. 1996. Modelling water retention curves of sandy soils using neural networks. *Water Resources Research* 32(10): 3033–3040.

Schaap, M.G. & Leij, F.J. 1998. Using neural networks to predict soil water retention and soil hydraulic conductivity. *Soil and Tillage Research* 47: 37–42.

Schaap, M.G., Leij F.J. & van Genuchten, M.Th. 1998. Neural network analysis for hierarchical prediction of soil water retention and saturated hydraulic conductivity. *Soil Science Society of America Journal* 62: 847–855.

Schaap, M.G., Leij, F.J. & van Genuchten, M.Th. 1999. A bootstrap-neural network approach to predict soil hydraulic parameters. In M.Th. van Genuchten, F.J. Leij, L. Wu (Eds.), *Proceeding of the International Workshop on Characterization and Measurements of the Hydraulic Properties of Unsaturated Porous Media* (pp. 1237–1250.) Riverside, CA: University of California.

See, L. & Abrahart, R.J. 2001. Multi-model data fusion for hydrological forecasting. *Computers and Geosciences* 27(8): 987–994.

See, L., Kneale, P. & Smith, A. 2001. Developing evaluation measures for neural network operational forecasting. *Geophysical Research Abstracts* 3. (CD-ROM).

Shamseldin, A.Y. & O'Connor, K.M. 2001. A non-linear neural network technique for updating of river flow forecasts. *Hydrology and Earth System Science* 5(4): 577–597.

Sharkey, A.J.C. (Ed.) 1999. *Combining Artificial Neural Networks: Ensemble and Modular Multi-Net Systems*. London: Springer-Verlag.

Sharkey, A.J.C. & Sharkey, N.E. 1995. How to improve the reliability of artificial neural networks. *Technical Report CS-95-11*, Department of Computer Science, University of Sheffield.

Sharkey, A.J.C., Sharkey, N.E. & Chandroth, G.O. 1996. Neural nets and diversity. *Neural Computing and Applications* 4: 218–227.

Shepherd, A.J. 1997. *Second-Order Methods for Neural Networks*. London: Springer-Verlag.

Silverman, D. & Dracup, J.A. 2000. Long range precipitation in California; a look inside the 'black box' of a trained network. Chapter 14 in: R.S. Govindaraju & R. Rao (Eds.), *Artificial Neural Networks in Hydrology* (pp. 311–329). Dordrecht: Kluwer.

Sivakumar, B. 2000. Chaos theory in hydrology: important issues and interpretations. *Journal of Hydrology* 227: 1–20.

Sivakumar, B., Berndtsson, R., Olsson, J. & Jinno, K. 2001. Evidence of chaos in the rainfall-runoff process. *Hydrological Sciences Journal* 46: 131–145.

Sivakumar, B., Liong, S.Y., Liaw, C.Y. & Phoon, K.K. 1999. Singapore rainfall behaviour: Chaotic? *ASCE Journal of Hydraulic Engineering* 4(1): 38–48.

Smith, J. & Eli, R.N. 1995. Neural-network models of rainfall-runoff process. *Journal of Water Resources Planning and Management* 121: 499–509.

Sorooshian, S. 1991. Parameter estimation, model identification, and model validation: conceptual-type models. In D.S. Bowles & P.E. O'Connell (Eds.), *Recent Advances in the Modelling of Hydrologic Systems* (pp. 443–467). Rotterdam: Kluwer Academic.

Stewart, S. 1997. *Does God Play Dice? The New Mathematics of Chaos*. London: Penguin Books.

Teodorescu, H.-N., Mlynek, D., Kandel, A. & Zimmerman, H.-J. 2000. *Intelligent Systems and Interfaces*. Boston: Kluwer Academic Publishers.

Tingsanchali, T. & Gautam, M.R. 2000. Application of tank, NAM, ARMA and neural network models to flood forecasting. *Hydrological Processes* 14: 2473–2487.

Tumer, K. & Ghosh, J. 1996. Error correlation and error reduction in ensemble classifiers. *Connection Science* 8: 385–404.

USGS. 2003. *Real-Time Data for the Nation.* URL = http://waterdata.usgs.gov/nwis/rt (August).

Van den Boogaard, H.F.P & Kruisbrink, A.C.H. 1996. Hybrid modelling by integrating neural networks with numerical models. In A. Müller (Ed.), *Proceedings of the 2nd International Conference on Hydroinformatics, Zurich, Switzerland, 9-13 September 1996.* Vol. 2, 471–478. Rotterdam: A.A. Balkema.

Wagener, T., Boyle, D.P., Lees, M.J., Wheater, H.S., Gupta, H.V. & Sorooshian, S. 2000. A framework for development and application of hydrological models. In *Proceedings of the Seventh National Hydrology Symposium*, *Newcastle-upon-Tyne, September*. London: British Hydrological Society. 3.75–3.81.

Whitley, R. & Hromadka, T.V. 1999. Approximate confidence intervals for design floods for a single site using a neural network. *Water Resources Research* 35(1): 203–209.

Wilby, R.L., Abrahart, R.J. & Dawson, C.W. 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrological Sciences Journal* 48(2): 163–181.

Willgoose, G.R. & Riley, S.J. 1998. An assessment of the long-term erosional stability of a proposed mine rehabilitation. *Earth Surface Processes and Landforms* 23: 237–259.

Willgoose, G.R. 2000. *User Manual for SIBERIA (Version 8.10)*. Department of Civil, Surveying and Environmental Engineering, University of Newcastle, Callagham, Australia. http://rambler.newcastle.edu.au/~cegrw/SiberiaDownloads/SIBERIA_8.10_Manual.pdf (December).

Yapo, P.O., Gupta, H.V. & Sorooshian, S. 1998. Multi-objective global optimisation for hydrological models. *Journal of Hydrology* 204: 83–97.

Young, P.C. 1993. Time variable and state dependent parameter modelling of non-stationary and nonlinear time series. In Subba Rao, T. (Ed.), *Developments in Time Series*. London: Chapman and Hall.

Young, P.C. 1998a. Data-based mechanistic modelling of engineering systems. *Journal of Vibration and Control* 4: 5–28.

Young, P.C. 1998b. Data-based mechanistic modelling of environmental, ecological, economic and engineering systems. *Environmental Modelling and Software* 13: 105–122.

Young, P.C. 1999a. Data-based mechanistic modelling of rainfall-flow processes. In M.J. Anderson (Ed.), *Model Validation in Hydrological Science*. Chichester: John Wiley & Sons.

Young, P.C. 1999b. Data-based mechanistic modelling, generalised sensitivity and dominant mode analysis. *Computer Physics Communications* 117: 113–129.

Young, P.C. & Beven, K.J. 1991. Computation of the instantaneous unit hydrograph and identifiable component flows with application to small upland catchments. Comment. *Journal of Hydrology* 126: 389–396.

Young, P.C. & Beven, K.J. 1994. Data-based mechanistic modelling and the rainfall-flow non-linearity. *Environmetrics* 5(3): 335–363.

Young, P.C., Jakeman, A.J. & Post, D.A. 1997. Recent advances in the data-based modelling and analysis of hydrological systems. *Water Science Technology* 36(5): 99–116.

Zaldivar, J.M., Gutierrez, E., Galvan, I.M., Strozzi, F. & Tomasin, A. 2000. Forecasting high waters at Venice Lagoon using chaotic time series analysis and nonlinear neural networks. *Journal of Hydroinformatics* 2(1): 61–84.