Atmospheric and Oceanographic Sciences Library 46

Vladimir M. Krasnopolsky

The Application of Neural Networks in the Earth System Sciences

Neural Networks Emulations for Complex Multidimensional Mappings

Springer

The Application of Neural Networks in the Earth System Sciences

ATMOSPHERIC AND OCEANOGRAPHIC SCIENCES LIBRARY
VOLUME 46

Vladimir M. Krasnopolsky

The Application of Neural Networks in the Earth System Sciences

Neural Networks Emulations for Complex Multidimensional Mappings

Springer

Vladimir M. Krasnopolsky
NOAA Center for Weather and Climate Prediction
Camp Spring, MD, USA

ISSN 1383-8601
ISBN 978-94-007-6072-1 ISBN 978-94-007-6073-8 (eBook)
DOI 10.1007/978-94-007-6073-8
Springer Dordrecht Heidelberg New York London

Library of Congress Control Number: 2013941400

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my daughter Katya and grandson Mark.

Preface

Scientific discovery consists in the interpretation for our own convenience of a system of existence which has been made with no eye to our convenience at all.

– Norbert Wiener, *The Human Use of Human Beings*

Science is triumphant with far-ranging success, but its triumph is somehow clouded by growing difficulties in providing for the simple necessities of human life on earth.

– Barry Commoner, *Science and Survival*

This book introduces some applications of Computational Intelligence (CI) to problems of Earth System Science (ESS). In my opinion, the meeting of CI and ESSs is not a coincidence. There is an affinity between these two fields of science at a very deep level. Both of them use a systems approach; they see their object as a complex system of partly autonomous, evolving, and adaptive subsystems intensively interacting with each other and with their environment, which also changes due to the interaction between subsystems and due to changes of the subsystems. This deep affinity between the two fields makes the approaches and tools developed in CI well suited for solving many problems in ESSs; therefore, CI can provide adequate models for modeling subsystems of the Earth System.

Such a system vision of objects of the study stimulates an understanding of similarity of many ESS problems from the mathematical point of view. In this book, I show that many subsystems of Earth System (ES) can be considered as complex multidimensional nonlinear mappings. CI provides a number of tools to approximate, emulate, or model such mappings; the particular tool considered in this book is the neural network (NN) technique. This book demonstrates many successful applications of NNs in ESSs. However, in addition to the use of the NN technique, I also attempt to demonstrate the advantages of using in ESS the CI vision of a subsystem (mapping) not as a static mapping but as an adaptive, evolving mapping interacting with the environment and adapting to it. The tremendous flexibility of the NN technique provides means for modeling such evolving adaptive mappings that function in a changing environment.

My goal in this book is to be tutorial in nature rather than to give a complete description of ESS NN applications. Thus, I selected some particular interesting applications and concentrated on a clear presentation of the methodological basis of these applications. Because both the ESS and CI fields are relatively new, in addition to presenting the NN background in Chap. 2, the book presents basic ESS background for each application that is introduced. For example, in Chap. 3, I include a detailed introduction into forward and inverse problems in remote sensing before discussing NN applications to satellite remote sensing; Chap. 4, which is devoted to NN applications in numerical climate and weather prediction, includes a brief introduction into numerical climate and weather modeling. This feature makes the book self-descriptive. The book presents a review of the field with the purpose of bringing the reader up-to-date on the state of the art. It can also serve as a convenient source book for researchers, teachers, and students who work in related fields.

College Park, MD, USA Vladimir M. Krasnopolsky

Acknowledgments

I would like to acknowledge the many people that have contributed to this book in different ways. I first became intrigued by NNs through discussions with Prof. S. Berkovich in 1989. Later, in 1992, my colleague and long-term collaborator, Dr. L.C. Breaker, suggested applying this technique to the problem of satellite retrievals. My long-term and fruitful collaboration with Drs. D. Chalikov, M. Fox-Rabinovitz, P. Rasch, and H. Tolman are gratefully acknowledged – they introduced me to fascinating world of climate, atmosphere, ocean, and wave physics and related numerical modeling. This, in turn, has led to many NN applications developed in collaboration with these researchers. I would also like to thank Dr. L.C. Breaker for his assistance in reading and commenting in the manuscript. My wife, Michelle Kray, edited and formatted the manuscript. I appreciate her help, patience, and support while I worked on the book. Finally, I am grateful for the continued support from the National Center for Environmental Prediction of National Weather Service at National Oceanic and Atmospheric Administration and of the Earth System Science Interdisciplinary Center at University of Maryland.

Contents

Abbreviations

AC	Anomaly correlation
BT	Brightness temperatures
CAM	Community atmosphere model (NCAR)
CAMRT	CAM radiation package
CFS	Climate Forecast System (NCEP NOAA)
CGCM	Coupled general circulation (or climate) model
CI	Computational Intelligence
CLD	Cloudiness
CMC	Canadian Meteorological Center
CMCGLB	Global Model from CMC
ConUS	Continental US
CP	Compound parameterization
CPC	Climate Prediction Center (NOAA)
CRM	Cloud-resolving model
CSRM	Cloud-system-resolving model
CTL	Control
DA	Dynamical adjustment
DAS	Data assimilation systems
DIA	Discrete interaction approximation
DJF	December-January-February
DWD	Deutscher Wetterdienst
ECMWF	European Centre for Medium-Range Weather Forecasts
ENM	Environmental numerical model
ENSO	El Niño-Southern Oscillation
EOF	Empirical orthogonal function
EPS	Ensemble prediction systems
ERS-2	European Remote Sensing scatterometer
ES	Earth System
ESS	Earth System Sciences
ETS	Equitable Threat Score
FM	Forward model

F2F	Field-to-field
F2P	Field-to-point
GCM	General circulation (or climate) model
GCRM	Global cloud-resolving model
GFS	Global Forecast System (NCEP NOAA)
GS	Goodberlet and Swift
GSW	Goodberlet, Swift, and Wilkerson
GSWP	Goodberlet, Swift, Wilkerson, and Petty
HEM	Hybrid environmental model
HGCM	Hybrid general circulation (or climate) model
HP	Hybrid parameterization
HPC	Hydro-meteorological Prediction Center (NOAA)
HYCOM	Hybrid Coordinate Ocean Model
iNN	Inverse NN
JJA	June-July-August
JMA	Japan Meteorological Agency
LW	Long wave
LWR	Long-wave radiation
MLP	Multilayer perceptron
MME	Multi-model ensemble
MMF	Multiscale modeling framework
NAM	North American Mesoscale Forecast System (NCEP NOAA)
NASA	National Aeronautics and Space Administration
NCAR	National Center for Atmospheric Research
NCEP	National Centers for Environmental Prediction
NN	Neural network
NNEM	NN ensemble mean
NNIA	Neural network interaction approximation
NNIAE	Neural network interaction approximation that uses the EOF basis
NOAA	National Oceanic and Atmospheric Administration
NSIPP	Natural Seasonal-to-Interannual Predictability Program (NASA)
NWP	Numerical weather prediction
OLR	Outgoing Longwave Radiation
PB	Physically based
PDE	Partial differential equations
PICE	Perturbed initial condition ensemble
PK	Petty and Katsaros
PPE	Perturbed physics ensemble
PRMSE	Profile RMSE
PSL	Pressure at the surface level
P2P	Point-to-point
QC	Quality control
RMSE	Root mean square error

RRTM	Rapid radiative transfer model
RS	Remote sensing
SD	Standard deviation
SLT	Statistical learning technique
SSH	Sea surface height
SSM/I	Special Sensor Microwave Imager
SST	Sea surface temperature
STPPE	Short-term perturbed physics ensemble
SWR	Short-wave radiation
TF	Transfer function
TOGA-COARE	Tropical Ocean Global Atmosphere Coupled Ocean–atmosphere Response Experiment
UKMO	United Kingdom Meteorological Office
WAVEWATCH III	NCEP wind wave model (NCEP NOAA)
WEM	Weighted ensemble mean

Chapter 1
Introduction

Life was simple before World War II. After that, we had systems.
– Grace Murray Hopper

There are no separate systems. The world is continuum. Where to draw a boundary around a system depends on the purpose of the discussion.
– Donella H. Meadows, *Thinking in Systems: A Primer*

Abstract In this chapter, a notion of Earth System (ES) as a complex dynamical system of interacting components (subsystems) is presented and discussed. Weather and climate systems are introduced as subsystems of the ES. It is shown that any subsystem of ES can be considered as a multidimensional relationship or mapping, which is usually complex and nonlinear. Evolution of approaches to ES and its subsystems is discussed, and the neural network (NN) technique as a powerful nonlinear tool for emulating subsystems of ES is introduced. Multiple NN applications, which have been developed in ES sciences, are categorized and briefly reviewed. The chapter contains an extensive list of references giving extended background and further detail to the interested reader on each examined topic.

We consider our planet as a complex, dynamical system of interacting components (subsystems), which is often simply referred to as the ES. ES contains the main components of planet Earth – the atmosphere, oceans, freshwater, soils, lithosphere, biosphere, and cryosphere (Lawton 2001) as its subsystems. To understand the major ES patterns and processes in their dynamics, we need to study not only the processes that go on within each component or subsystem of ES (traditionally the realms of atmospheric physics, oceanography, hydrology, geology, and ecology, to name some) but also the *interactions, relationships,* and *feedbacks between* them. The Earth, in fact, is only habitable because of these complex linkages and feedbacks between the atmosphere, oceans, land, biosphere, and cryosphere.

V.M. Krasnopolsky, *The Application of Neural Networks in the Earth System Sciences*,
Atmospheric and Oceanographic Sciences Library 46, DOI 10.1007/978-94-007-6073-8_1,

The interactions between subsystems condition, change, and manage many processes inside subsystems. It is the need to study the evolution of ES and understand these inter-component interactions, relationships, and the changes they cause in subsystem processes that defines ESS as a discipline in its own right. We still do not understand all of these feedbacks and cannot, as yet, build a model that reproduces all of the changes in ES, but these problems now hold center stage in ESS.

A large variety of highly nonlinear processes with tremendously wide spectrum of spatial and temporal scales contributes to ES, which adds to its extreme complexity. The temporal scales range from hundreds of millions of years (paleoclimatic phenomena) to several minutes (microscale weather events), and the spatial scales range from thousands of kilometers (global phenomena) to several millimeters (size of water droplets in the clouds).

Considering subsystems of ES formally, we can say that each subsystem in ES receives information (input) from other subsystems. This information comes as a set or a vector of input signals or parameters, which inform the subsystem about the status of the system as a whole and about the states of the related subsystems. Air and ocean water temperature and pressure, concentration of CO_2, radiation, and heat fluxes are just several examples of such parameters. The subsystem, in turn, communicates with the system and other subsystems, transmitting information to them concerning its state as a part of ES. This output information is transmitted as a set or vector of output parameters or signals. Thus, formally speaking, any subsystem of ES can be considered as a relationship, usually complex and nonlinear, between two vectors: a vector of input and a vector of output parameters. Such a relationship is called *mapping*.

Various mathematical methods are applied to describe, model, and emulate mappings that represent the subsystems of ES. Deterministic and statistical approaches are both employed. The deterministic approach is based on a more or less complete understanding of *first principles* or basic processes in the subsystem. This understanding is usually codified into a set of partial differential equations (PDE). Statistical approaches are based on working with data and extracting information directly from the data. They are also called statistical learning (a.k.a. machine learning, learning from data, predictive learning, data-driven) techniques because, in a sense, they learn relationships or mappings directly from the data. Such approaches are used when the understanding of processes in the subsystems is poor or incomplete or when deterministic approaches become too resource intensive (e.g., numerical solutions of PDE).

This book introduces a particular nonlinear CI or statistical learning technique (SLT), namely, the NN approach, and demonstrates how to apply it for modeling or emulation of important subsystems of ES. In this chapter in Sect. 1.1, a notion of ES as a complex dynamical system of interacting components (subsystems) is presented; the role of organization and structure of a system is discussed. Weather and climate systems are introduced as subsystems of the ES. It is shown that any subsystem of ES can be considered as a multidimensional relationship or mapping, which is usually complex and nonlinear. In Sect. 1.2, evolution of approaches to ES and its subsystems is discussed, and in Sect. 1.3 the neural network (NN) technique

as a powerful nonlinear tool for emulating subsystems of ES is introduced. Multiple NN applications, which have been developed in ES sciences, are categorized and briefly reviewed, and the structure of the book is outlined.

1.1 Systems, Subsystems, Organization, and Structure

Formally, a system can be defined as a set of *elements* or *parts* that is *coherently organized* and interconnected in a *pattern* or *structure* that produces a characteristic set of behaviors, often classified as its *function* or "purpose" (Meadows 2008). Thus, any system is composed of components or parts. In aggregations parts are added; in systems components or parts are arranged or organized; hence, each system has a well-defined structure. Systems are significant because of organization-positional values, because of their structure. If a system is properly structured or organized, then it is more than the total sum of its parts and the whole system may demonstrate behavior (quality) that cannot be predicted by the behavior of its parts. In such cases we are talking about a *synergy* of the parts in the system.

In ES and in many other systems, the constituent parts of the system are systems by themselves. For example, a complex climate and weather system (see Fig. 1.1) is a constituent of ES. The atmospheric constituent of the climate system is a complex

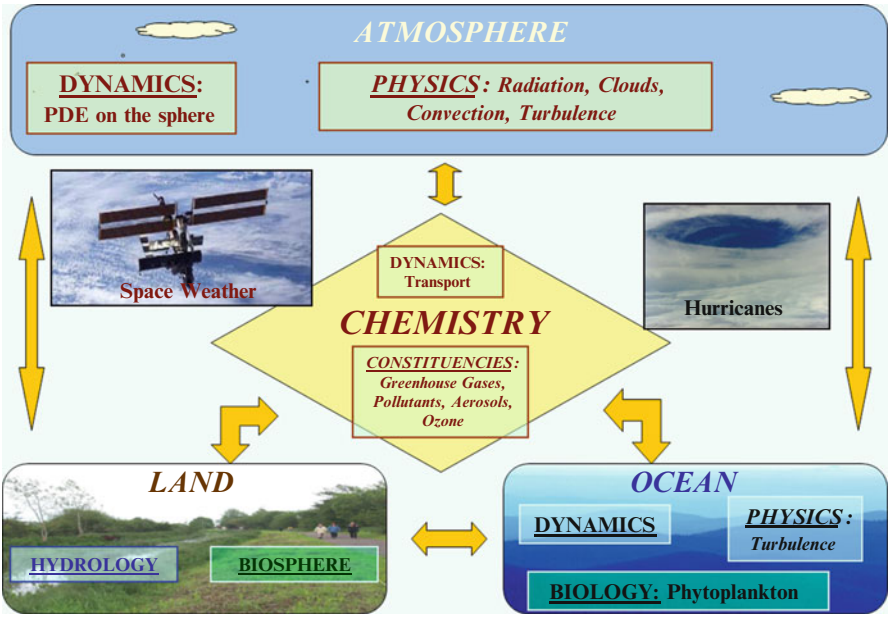

Fig. 1.1 Interdisciplinary complex climate and weather systems. Only several major interactions (feedbacks) between major subsystems are shown with *arrows*

system of interacting dynamical, physical (radiation, convection, etc.), and chemical processes (see Fig. 1.1). Such constituent parts of the whole system that themselves have structure (organization) are called subsystems. Systems arranged in such a way (nested systems in the system) are called *hierarchical systems* (Wilber 1995; Ahl and Allen 1996). A hierarchical system is an arrangement of subsystems, in which the subsystems are represented as being "above," "below," or "at the same level as" with respect to one another. In such a hierarchy, subsystems can interact either directly or indirectly and either vertically (between different levels of hierarchy) or horizontally (at the same level). The number of the vertical levels determines the depth or the vertical (hierarchical) complexity of hierarchical system (Salthe 1985).

Interactions and relationships at a higher level of hierarchical complexity organize and transform the lower-order interactions, produce organizations of lower-order relationships that are new and not arbitrary, and cannot be accomplished by those lower-order interactions alone. The higher-order relationship governs or coordinates the relationships of the next lower order; it embraces and transcends the lower orders (Wilber 1995). It is noteworthy that interactions in complex systems are better described by feedback loops than by one directional cause and effect type actions, which makes analysis of such systems even more difficult.

1.2 Evolution of Approaches to Earth System

Systematic study of various components of ES (climate, atmosphere, ocean, ice, etc.) started in the second half of the nineteenth century as separate fields: meteorology, oceanography, glaciology, etc. In many of these fields, the growing understanding of physical processes was codified into PDEs. Systematic data collection and archiving were established. In the first part of the twentieth century, the process of knowledge and data accumulating and developing mathematical tools was continued rather independently in these single disciplinary fields. In the 1920s, a pioneer of modern meteorology, Richardson (1922), attempted to integrate the full primitive equations of fluid dynamics, performing calculations by hand, which made a huge impact on and paved the way for the emerging science of numerical weather and climate prediction. The classical linear statistical framework was formulated by Fisher (1922), which allowed scientists to extract useful information directly from observed data and to make practical predictions even in the cases when the underlying processes were not sufficiently understood. At the same time, the weather observation network (primarily ground observation) continued to expand.

The middle of the twentieth century was marked by the growing understanding of the interrelatedness of the processes in the atmosphere, land, and oceans, through increased awareness of the multiple feedbacks between the abiotic nature and biosphere. The notion of Earth as a complex system was emerging. This new vision stimulated the development of and was supported by the emerging theory of complex systems (e.g., von Bertalanffy 1950).

With the advent of electronic computers in the 1950s, the work of Richardson, Fisher, and other pioneers came to fruition. The field of numerical weather prediction (NWP) was born. With the dawn of satellite era in the late 1960s, the quantity of observations started to increase rapidly. Regions that could not be observed systematically in details (e.g., the oceans and the polar regions) became accessible for satellite observations. The need to derive geophysical parameters from satellite data gave birth to the science of satellite remote sensing (RS). The wealth of global observations stimulated further development of NWP models and data assimilation systems (DAS) that allowed to properly merge or fuse the large amounts of data from different sources (e.g., ground observations and satellite retrievals) and provided more balanced and realistic initial conditions for NWP models. What is even more important, the global perspective provided by satellites facilitated the vision of Earth as a complex whole, as complex ES where classical linear one-way causality gives way to complex nonlinear feedback relationship.

ESS, which emerged in the last quarter of the twentieth century, was marked by transition from a single disciplinary, linear, and low-dimensional paradigm to a multidisciplinary, high-dimensional, and nonlinear paradigm. The corresponding evolution of deterministic methodology included the transition from a collection of separate atmospheric, oceanic, land, and more generally various geophysical numerical prediction models to very complex numerical prediction systems, which have been constructed by coupling atmospheric, oceanic, land, ice, etc., models and allowing them to interact through multiple feedback mechanisms in the process of their integration. Parallel evolution of statistical methodology includes the transition from classical linear, low-dimensional, and parametric approaches to nonlinear, high-dimensional, and nonparametric statistical framework. This new statistical framework can provide adequate statistical models for subsystems of ES that are acceptable for many applications. This new statistical paradigm is still under development and has many different appearances. The NN approach is one of the mostly popular and matured tools that emerged inside this paradigm and found a broad application in different fields of ESS.

1.3 Neural Networks in Earth System Sciences

The NN approach is a relatively new, diverse, and powerful CI approach or SLT that started developing rapidly in the mid-1980s after several major basic types of NNs were introduced in the works of Kohonen (1982), Hopfield (1982), Rumelhart et al. (1986), and Lippmann (1989). In the 1990s this technique evolved and matured rapidly; several well-written and fundamental textbooks have been published by Beale and Jackson (1990), Haykin (1994), Bishop (1995), Vapnik (1995), Ripley (1996), and Cherkassky and Mulier (1998) that introduced NNs as a new powerful statistical learning approach capable of providing a diverse family of flexible, multidimensional, nonlinear, data-driven models for various applications.

This approach became appealing to a broad community of professionals including scientists working in different fields of ESS such as satellite RS, meteorology, oceanography, hydrology, and climate and weather numerical modeling.

Since then a significant number of NN applications have been developed in these fields; the most important of them are summarized in Table 1.1. References presented there do not provide a complete list of the corresponding publications, or even the most important ones. Rather, this table gives representative examples of publications devoted to the topic. A number of these applications or groups of applications have already been reviewed in several review papers. Selected atmospheric, oceanic, and climate applications have been reviewed for the atmospheric and oceanic community by Gardner and Dorling (1998), Hsieh and Tang (1998), Krasnopolsky (2007), in the book edited by Haupt et al. (2009), and in the book by Hsieh (2009). For the NN professionals such applications have been reviewed by Krasnopolsky and Chevallier (2003) and Krasnopolsky and Fox-Rabinovitz (2006a). Selected RS applications have been reviewed for RS experts by Atkinson and Tatnall (1997) and for NN community by Krasnopolsky and Schiller (2003). Applications of the NN technique for developing nonlinear generalizations of multivariate statistical analysis have been recently reviewed by Hsieh (2004) (see also Hsieh (2009)). Finally, Solomatine and Ostfeld (2008) have reviewed hydrological NN applications.

As can be seen from Table 1.1, a wide variety of NN applications has been developed in different ESS fields. These applications utilize different types of NNs. The task of selecting a group of such applications for describing in a book may be approached from different directions. Our goal in this book is to be tutorial in nature rather than to give a complete description of ESS NN applications. We will select some particular interesting applications and concentrate on a clear presentation of the methodological basis of these applications.

In this venue, we have selected a group of ESS applications for review in this book, which, from the mathematical point of view, could be formulated as complex, multidimensional, nonlinear mappings and which, from the point of view of the NN technique, utilized a particular type of NN – the multilayer perceptron (MLP) (Rumelhart et al. 1986). This framework is broad and generic. It covers the majority of applications developed in ESS, namely, NN applications to a large variety of subsystems of ES that can be considered as complex, multidimensional, *nonlinear mappings* (see Table 1.1). To focus this book even more, we selected as particular examples mainly those applications which were developed with the author's participation. They are shown in italics in Table 1.1.

In the methodological Chap. 2, we introduce the concept and major properties of complex nonlinear mappings. We also introduce the MLP NN as a generic technique for the approximation of nonlinear continuous and almost continuous mappings. Our theoretical understanding of complex multidimensional nonlinear mappings and highly nonlinear approximation methods (like the MLP NN) is still quite fragmentary (DeVore 1998). This is why the material we put together in Chap. 2 is a collection of theoretical results and practical inferences based on numerical

Table 1.1 Some NN applications in climate, weather, and related fields

NN application		Publications
I. Satellite meteorology and oceanography		
I.1	Classification	Gallinari et al. (1991) and Bhattacharya and Solomatine (2006)
I.2	Pattern recognition, feature extraction	Bankert (1994) and Nabney (2002)
I.3	Change detection and feature tracking	Valdés and Bonham-Carter (2006)
I.4	*Fast forward models for variational retrievals*	Krasnopolsky (1996, 1997)
I.5	*Accurate transfer functions* (*retrieval algorithms*)	
	I.5.1 *Surface parameters*	Stogryn et al. (1994), Badran et al. (1995), and Krasnopolsky and Schiller (2003)
	I.5.2 Atmospheric profiles	Aires et al. (2002) and Mueller et al. (2003)
II. Predictions		
I.1	Geophysical time series	Elsner and Tsonis (1992) and Bollivier et al. (2000)
I.2	Regional and global climate	Pasini et al. (2006)
I.3	Time-dependent physical processes	Wu et al. (2006)
III. Hybrid climate and weather numerical models and data assimilation systems		
III.1	*Hybridparameterizations of physics*	Chevallier et al. (1998)
III.2	*Hybridclimate models*	Krasnopolsky and Fox-Rabinovitz (2006b)
III.3	*Fast emulations of model physics*	Krasnopolsky et al. (2002, 2010)
III.4	*NN-based parameterizations of model physics*	Krasnopolsky et al. (2011)
III.5	*Fast forward models for direct assimilation*	Krasnopolsky (1996, 1997)
III.6	*Observation operator for propagating a signal to different vertical levels and variables in data assimilation*	Krasnopolsky et al. (2006)
III.7	Hybrid coupled models	Tang and Hsieh (2003)
IV. Improving model forecast		
IV.I	Model output statistics	Marzban (2003)
IV.2	*NNmulti-model ensembles*	Krasnopolsky and Lin (2012)
V. Geophysical data fusion		Loyola and Ruppert (1998)
VI. Geophysical data mining		Brown and Mielke (2000)
VII. Interpolation and downscaling		Dibike and Coulibaly (2006) and Benestad et al. (2008)
VIII. Nonlinear multivariate statistical analysis		Hsieh (2004, 2009)
IX. Hydrology		Bhattacharya et al. (2005) and Solomatine and Ostfeld (2008)
X. Magnetosphere and ionosphere physics		Vörös and Jankovičová (2002) and Tulunay et al. (2004)

The list of applications included in the table is not intended to be exhaustive and is limited by the knowledge of the author. The sequence of applications does not reflect their importance and is more or less arbitrary. The references included in the *column* "Publications" are representative papers which deal with this particular application. When we had sufficient information, we included the first (to the best of our knowledge) publication of this application. Applications that are discussed in more details in this book are shown in italic in the table

experiments and experience obtained from the applications of the MLP technique to various problems. This tutorial material is valuable for understanding the entire topic and the following parts of this book. Chapter 2 contains some important details that formally and structurally belong there; however, during the first reading they may look like some interesting but abstract and secondary material. The importance and relevance of this material will certainly be appreciated during and after reading the rest of this book. Thus, we would recommend to readers interested in a deeper understanding of the technique to return to Chap. 2 during and after reading the rest of this book. Also, to emphasize the validity of the material presented in Chap. 2, we cross-reference between Chap. 2 and subsequent chapters to help readers to return to Chap. 2 to refresh their background information when reading the following chapters.

In Chap. 3, we discuss using NNs for RS applications: emulating solutions of forward and inverse problems in satellite RS. In Chap. 4, we describe the use of NNs for creating accurate and fast NN emulations of model physics parameterizations in atmospheric, oceanic, and ocean wave models and for developing hybrid models by combining these NN components with deterministic (based on first principles) model components. We also discuss the possibility of developing new NN-based parameterizations of model physics.

In Chap. 5, we introduce an NN application that allows the creation of NN emulations for the functions and mappings (between model state variables) often obscured in the numerical outputs of modern high resolution atmospheric and oceanic numerical models. These NN emulations should be helpful in modern DASs. At this point we also review NN ensemble approaches that allow enhancement of model predictions, improvement in the accuracy of NN emulations, and reduction of the uncertainties of NN Jacobians.

Chapter 6 contains conclusions. To the best of our knowledge, at this time, the NN is the only practical SLT tool for solving the majority of problems discussed in this book and more generally for emulating complex multidimensional mappings. However, for completeness, in Sect. 6.2 we briefly review some alternative CI techniques that have potentials to compete with NNs in the future and present some preliminary results with alternative approaches (Belochitski et al. 2011).

References

Ahl V, Allen TFH (1996) Hierarchy theory: a vision, vocabulary, and epistemology. Columbia University Press, New York

Aires F, Rossow WB, Scott NA, Chedin A (2002) Remote sensing from the infrared atmospheric sounding interferometer instrument: 2 Simultaneous retrieval of temperature, water vapor, and ozone atmospheric profiles. J Geophys Res. doi:10.1029/2001JD001591

Atkinson PM, Tatnall ARL (1997) Neural networks in remote sensing – introduction. Int J Remote Sens 18(699):709

Badran F, Mejia C, Thiria S, Crépon M (1995) Remote sensing operations. Int J Neural Syst 6:447–453

Bankert RL (1994) Cloud pattern identification as part of an automated image analysis. In: Proceedings of the seventh conference on satellite meteorology and oceanography, Monterey, CA, 6–10 June, pp 441–443

Beale R, Jackson T (1990) Neural computing: an introduction. Adam Hilger, Bristol/Philadelphia/New York

Belochitski AP, Binev P, DeVore R, Fox-Rabinovitz M, Krasnopolsky V, Lamby P (2011) Tree approximation of the long wave radiation parameterization in the NCAR CAM global climate model. J Comput Appl Math 236:447–460

Benestad RE, Hanssen-Bauer I, Chen D (2008) Empirical-statistical downscaling. World Scientific Publishing Company, Singapore

Bhattacharya B, Solomatine DP (2006) Machine learning in soil classification. Neural Netw 19:186–195

Bhattacharya B, Price RK, Solomatine DP (2005) Data-driven modelling in the context of sediment transport. Phys Chem Earth 30:297–302

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford

Bollivier M, Eifler W, Thiria S (2000) Sea surface temperature forecasts using on-line local learning algorithm in upwelling regions. Neurocomputing 30:59–63

Brown TJ, Mielke PW (2000) Statistical mining and data visualization in atmospheric sciences. Kluwer Academic Publishers, Boston

Cherkassky V, Mulier F (1998) Learning from data. Wiley, Hoboken

Chevallier F, Chéruy F, Scott NA, Chedin A (1998) A neural network approach for a fast and accurate computation of longwave radiative budget. J Appl Meteorol 37:1385–1397

DeVore RA (1998) Nonlinear approximation. Acta Numerica 8:51–150

Dibike YB, Coulibaly P (2006) Temporal neural networks for downscaling climate variability and extremes. Neural Netw 19:135–144

Elsner JB, Tsonis AA (1992) Nonlinear prediction, chaos, and noise. Bull Am Meteorol Soc 73:49–60

Fisher RA (1922) On the mathematical foundations of theoretical statistics. Philos Trans R Soc A222:309–368

Gallinari P, Thiria S, Badran F, Fogelman-Soulie F (1991) On the relations between discriminant analysis and multilayer perceptrons. Neural Netw 4:349–360

Gardner MW, Dorling SR (1998) Artificial neural networks (the multilayer perceptron) – a review of applications in the atmospheric sciences. Atmos Environ 32:2627–2636

Haupt SE, Pasini A, Marzban C (eds) (2009) Artificial intelligence methods in environmental sciences. Springer, New York

Haykin S (1994) Neural networks: a comprehensive foundation. Macmillan College Publishing Company, New York

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational ability. Proc Natl Acad Sci USA 79:2554–2558

Hsieh WW (2004) Nonlinear multivariate and time series analysis by neural network methods. Rev Geophys. doi:10.1029/2002RG000112

Hsieh WW (2009) Machine learning methods in the environmental sciences. Cambridge University Press, Cambridge

Hsieh WW, Tang B (1998) Applying neural network models to prediction and data analysis in meteorology and oceanography. Bull Am Meteorol Soc 79:1855–1870

Kohonen T (1982) Self-organizing formation of topologically correct feature maps. Biol Cybern 43:59–69

Krasnopolsky V (1996) A neural network forward model for direct assimilation of SSM/I brightness temperatures into atmospheric models. Working group on numerical experimentation blue book. 1.29–1.30. http://polar.ncep.noaa.gov/mmab/papers/tn134/OMB134.pdf

Krasnopolsky V (1997) A neural network based forward model for direct assimilation of SSM/I brightness temperatures. Tech note, OMB contribution No 140, NCEP/NOAA. http://polar.ncep.noaa.gov/mmab/papers/tn140/OMB140.pdf

Krasnopolsky VM (2007) Neural network emulations for complex multidimensional geophysical mappings: applications of neural network techniques to atmospheric and oceanic satellite retrievals and numerical modeling. Rev Geophys 45(3):RG3009. doi:10.1029/2006RG000200

Krasnopolsky VM, Chevallier F (2003) Some neural network applications in environmental sciences Part II: Advancing computational efficiency of environmental numerical models. Neural Netw 16:335–348

Krasnopolsky VM, Fox-Rabinovitz MS (2006a) Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. Neural Netw 19:122–134

Krasnopolsky VM, Fox-Rabinovitz MS (2006b) A new synergetic paradigm in environmental numerical modeling: hybrid models combining deterministic and machine learning components. Ecol Model 191:5–18

Krasnopolsky VM, Lin Y (2012) A neural network nonlinear multimodel ensemble to improve precipitation forecasts over continental US. Adv Meteorol 2012:11 pp. Article ID 649450, doi:10.1155/2012/649450. http://www.hindawi.com/journals/amet/2012/649450/

Krasnopolsky VM, Schiller H (2003) Some neural network applications in environmental sciences Part I: Forward and inverse problems in satellite remote sensing. Neural Netw 16:321–334

Krasnopolsky VM, Chalikov DV, Tolman HL (2002) A neural network technique to improve computational efficiency of numerical oceanic models. Ocean Model 4:363–383

Krasnopolsky VM, Lozano CJ, Spindler D, Rivin I, Rao DB (2006) A new NN approach to extract explicitly functional dependencies and mappings from numerical outputs of numerical environmental models. In: Proceedings of the IJCNN2006, Vancouver, BC, Canada, 16–21 July, pp 8732–8734

Krasnopolsky VM, Fox-Rabinovitz MS, Hou YT, Lord SJ, Belochitski A (2010) Accurate and fast neural network emulations of model radiation for the NCEP coupled climate forecast system: climate simulations and seasonal predictions. Mon Weather Rev 138:1822–1842. doi:10.1175/2009MWR3149.1

Krasnopolsky V, Fox-Rabinovitz M, Belochitski A, Rasch P, Blossey P, Kogan Y (2011) Development of neural network convection parameterizations for climate and NWP models using cloud resolving model simulations. NCEP office note 469. http://www.emc.ncep.noaa.gov/officenotes/newernotes/on469.pdf

Lawton J (2001) Earth system science. Science. doi:10.1126/science.292.5524.1965

Lippmann RP (1989) Pattern classification using neural networks. IEEE Commun Mag 27:47–64

Loyola D, Ruppert T (1998) A new PMD cloud-recognition algorithm for GOME. ESA Earth Obs Q 58:45–47

Marzban C (2003) Neural networks for postprocessing model output: ARPS. Mon Weather Rev 131:1103–1111

Meadows DH (2008) Thinking in systems: a primer. Chelsea Green Publishing Company, Vermont

Mueller MD et al (2003) Ozone profile retrieval from global ozone monitoring experiment data using a neural network approach (Neural Network Ozone Retrieval System (NNORSY)). J Geophys Res 108:4497. doi:10.1029/2002JD002784

Nabney IT (2002) Netlab: algorithms for pattern recognition. Springer, New York

Pasini A, Lorè M, Ameli F (2006) Neural network modelling for the analysis of forcings/temperatures relationships at different scales in the climate system. Ecol Model 191:58–67

Richardson LF (1922) Weather prediction by numerical process. Cambridge University Press, Cambridge

Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, Group PR (eds) Parallel distributed processing, vol 1. MIT Press, Cambridge, MA

Salthe SN (1985) Evolving hierarchical systems their structure and representation. Columbia University Press, New York

Solomatine D, Ostfeld A (2008) Data-driven modeling: some past experiences and new approaches. J Hydroinform 10(1):3–22

Stogryn AP, Butler CT, Bartolac TJ (1994) Ocean surface wind retrievals from special sensor microwave imager data with neural networks. J Geophys Res 90:981–984

Tang Y, Hsieh WW (2003) ENSO simulation and prediction in a hybrid coupled model with data assimilation. J Meteorol Soc Jpn 81:1–19

Tulunay Y, Tulunay E, Senalp ET (2004) The neural network technique – 2: an ionospheric example illustrating its application. Adv Space Res 33:988–992

Valdés JJ, Bonham-Carter G (2006) Time dependent neural network models for detecting changes of state in complex processes: applications in earth sciences and astronomy. Neural Netw 19:196–207

Vapnik VN (1995) The nature of statistical learning theory. Springer, New York

von Bertalanffy L (1950) An outline of general system theory. Br J Philos Sci 1:139–164

Vörös Z, Jankovičová D (2002) Neural network prediction of geomagnetic activity: a method using Hölder exponents. Nonlinear Processes Geophys 9:425–433

Wilber K (1995) Sex, ecology, spirituality: the spirit of evolution. Shambhala Publication, Boston

Wu A, Hsieh WW, Tang B (2006) Neural network forecasts of the tropical Pacific sea surface temperatures. Neural Netw 19:145–154

Chapter 2
Introduction to Mapping and Neural Networks

Science cannot exist without some small portion of metaphysics.
– Max Karl Ernst Planck, *The Universe in the Light of Modern Physics*

The aim of science is always to reduce complexity to simplicity.
– William James, *The Principles of Psychology*

Abstract In this chapter, the major properties of mappings and multilayer perceptron (MLP) neural networks (NNs) are formulated and discussed. Several examples of real-life problems (prediction of time series, interpolation of lookup tables, satellite retrievals, and fast emulations of model physics) that can be considered as complex, nonlinear, and multidimensional mappings are introduced. The power and flexibility of the NN emulation technique as well as its limitations are discussed; also, it is shown how various methods can be designed to bypass or reduce some of these limitations. The chapter contains an extensive list of references giving extended background and further detail to the interested reader on each examined topic. It can be used as a textbook and an introductory reading for students and beginning and advanced investigators interested in learning how to apply the NN technique to emulate various complex, nonlinear, and multidimensional mappings in different fields of science.

A mapping, M, between two vectors X (input vector) and Y (output vector) can be symbolically written as

$$Y = M(X); \quad X \in \Re^n, Y \in \Re^m \tag{2.1}$$

where n and m are the dimensionalities of the input and output spaces correspondingly. A large number of important practical ESS applications may be considered mathematically as a mapping (2.1). Keeping in mind that an NN technique will be

V.M. Krasnopolsky, *The Application of Neural Networks in the Earth System Sciences*, Atmospheric and Oceanographic Sciences Library 46, DOI 10.1007/978-94-007-6073-8_2,

used to approximate (or emulate) this mapping, we will call this mapping a *target mapping*, using a term taken from nonlinear approximation theory (DeVore 1998).

The target mapping may be defined explicitly or implicitly. It can be defined explicitly as a set of equations based on first principles and/or empirical dependencies (e.g., radiative transfer or heat transfer equations) or as a computer code. A collection of data records (e.g., observations, measurements, computer simulations) represents the target mapping implicitly. The target mapping is assumed to represent these data and to generate them as well.

In this chapter, the major properties of mappings and multilayer perceptron (MLP) neural networks (NNs) are formulated and discussed. In Sect. 2.1, several examples of real-life problems (prediction of time series, interpolation of lookup tables, satellite retrievals, and fast emulations of model physics) that can be considered as complex, nonlinear, and multidimensional mappings are introduced. The most important generic properties of mappings are discussed in Sect. 2.2. The MLP NN technique, which is used in this book as a generic tool for emulating nonlinear mappings, is introduced in Sect. 2.3. In Sect. 2.4, the power and flexibility of the NN technique as well as its limitations are discussed; also, it is shown how various methods can be designed to bypass or reduce some of these limitations. Section 2.5 discusses special features of NN emulations, and in Sect. 2.6 some final remarks are presented.

2.1 Mapping Examples

2.1.1 Prediction of Time Series

Prediction of time series may be considered as a mapping between the past and future (Elsner and Tsonis 1992; Bollivier et al. 2000; Maas et al. 2000). In this case, the vector $X = \{x_{t-k},\ x_{t-k+1},\ \ldots,\ x_t\}$ is a lag vector created from k past values of the time series for the variable x, and the vector $Y = \{x_{t+1},\ x_{t+2},\ \ldots,\ x_{t+p}\}$ contains p future or predicted values for the same variable x. The components of vectors X and Y in this case may be significantly correlated as sequential terms of the same time series that represents the sequential measurements of the same underlying process. Depending on the nature of the process represented by the time series, the target mapping M may be linear or nonlinear (e.g., in the case of a chaotic process) (Elsner and Tsonis 1992). Prediction of a time series as a mapping using NNs is discussed in detail in Weigend and Gershenfeld (1994).

2.1.2 Lookup Tables

Lookup tables are often used in various fields of science to accelerate calculations of complex mappings (2.1). Lookup tables allow one to replace a runtime computation with simpler operations with array indexing, i.e., with reading from memory.

The savings in terms of processing time can be significant, since retrieving a value from memory is often faster than undergoing a more time-consuming computation. To create a lookup table, the mapping (2.1) is pre-calculated on an n-dimensional grid obtained by a discretization of the input vector X.

For example, in some numerical climate and weather prediction models, the aerosol properties of the atmosphere are represented as time-dependent 3-D lookup tables, as expressed in the following equation:

$$y_i^{jklp} = M_i\left(\lambda_j, \mu_k, h_l, t_p\right) = \overline{M_i}\,(j, k, l, p)$$

where $\overline{M}$ is the lookup table and $i = 1, 2, 3$ correspond to the aerosol optical depth, the single scattering albedo, and asymmetry parameters. λ and μ are the longitude and the latitude, h is the height, and t is time – all placed on a 4-D grid, where $j = 1, \ldots, J$, $k = 1, \ldots, K$, $l = 1, \ldots, L$, and $p = 1, \ldots, P$ are the indices of the grid. The time index, p, for this example, changes from 1 to $P = 12$, because the aerosol parameters depend only on the month of the year in this model.

To calculate the value of y for a particular set of arguments $X = (\lambda, \mu, h, t)$, a set $\left(\lambda_j, \mu_k, h_l, t_p\right)$ closest to (λ, μ, h, t) has to be found, and a set of indexes (j, k, l, p) has to be determined, and then $\overline{M}_i(j, k, l, p)$ can be retrieved from the memory. If the grid is coarse (i.e., J, K, L, and P are small), the table is small and the calculation is fast; however, the accuracy of the calculated value for y may be low. If the grid is fine, the accuracy is improved; however, the time required for the retrieving and searching a large table, $\overline{M}_i$, may become comparable with (or even greater than) the time required for the direct computation of $\overline{M}_i$. Sometimes an intermediate solution that uses a coarse table in combination with an appropriate interpolation procedure is applied. Using interpolation can produce greater accuracy for values that fall between two precomputed values; however, the process of interpolation takes additional time. Thus, a compromise between the speed and the accuracy of the lookup table is always an issue when a lookup table approach is used.

There are two fundamental limitations on when it is possible to construct a lookup table for a required operation. The first is the amount of memory that is available: clearly, one cannot construct a lookup table that is larger than the space available for the table, although it is possible to construct disk-based lookup tables at the expense of increased search or lookup time. The other is the time required to search the table, which quickly increases with an increase in the number of dimensions involved in the mapping (2.1).

NNs can be considered and used as fast analytical lookup tables that does not require searching and provides a fast and accurate interpolation. For example, NNs have been used to retrieve atmospheric properties from satellite RS measurements (Vann and Hu 2002) as an alternative to the use of lookup table-based inversion algorithm. In this case NN is used to emulate and inverse the lookup table. NNs provide an impressive speedup that increases rapidly with an increase of dimensionality n of the input vector. For example, as it is shown by Vann and Hu (2002), for $n = 2$, the NN is $\sim$100 times faster, and for $n = 6$, the NN is already 10^4

times faster than the lookup table. It is noteworthy that such an impressive speedup at low dimensionalities n of the input vector has been demonstrated because NNs, in this work, not only emulate but also inverse the lookup table.

2.1.3 Satellite Remote Sensing

The third example of a generic application that can be formally considered as the mapping (2.1) is a retrieval algorithm (or a transfer function) in satellite RS that converts the input vector X of satellite measurements (calibrated or raw radiances, brightness temperature, backscatter coefficients, etc., at different frequencies) into the vector Y of geophysical parameters like wind speeds, atmospheric moisture parameters, and ocean and land surface characteristics. Here the components of vector X may (as in previous examples) be correlated because the frequency bands may not be completely independent and may overlap. The components of the output vector Y may be correlated because the corresponding geophysical parameters are physically related (Krasnopolsky et al. 1999, 2000). The target mapping in this example may also be a complicated nonlinear mapping. This application is discussed in Chap. 3.

2.1.4 Emulation of Subsystems of the Climate System

In Chap. 1 we mentioned that, formally speaking, any subsystem of ES can be considered as a relationship, usually complex and nonlinear, between two vectors: a vector of input and a vector of output parameters or as a *mapping* (2.1). The input and the output vectors allow the subsystem to exchange information with other subsystems of ES. The same is true with respect to numerical models that describe ES and its subsystems.

Figure 2.1 shows the hierarchical chart for a generic general circulation (or climate) model (GCM) which is used to produce climate projections and simulations. GCMs have a significant hierarchical complexity (see Sect. 1.1). The first five hierarchical levels are presented in the figure. The first level represents the numerical models for the major constituents (subsystems) of the GCM: ocean, atmospheric, land, and ice models. Only the structure of the atmospheric branch is presented for clarity in the figure, and, likewise, only atmospheric subsystems are shown. The subsystems that are shown in Fig. 2.1 (shown as boxes in the figure) at any hierarchical level are mappings that can be modeled or emulated by an appropriate NN. Boxes with red titles show subsystems that have been already modeled or emulated using an NN.

Tang and Hsieh (2003) studied the El Niño-Southern Oscillation (ENSO), a well-known climatic phenomenon characterized by quasiperiodic oscillations of sea surface temperature (SST) in the tropical Pacific Ocean. They used a climate

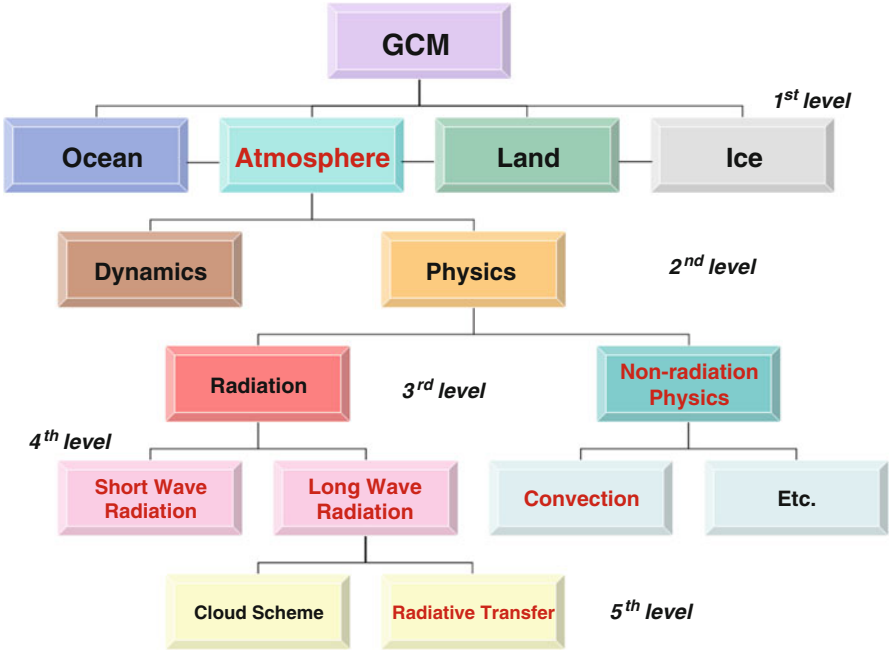

Fig. 2.1 The hierarchical chart of a generic GCM which provides climate projections and simulations. 2-D setup of this chart does not allow us to show all interactions between subsystems at the same hierarchical level and between levels; many of them are omitted for clarity

model, which includes two major subsystems of the climate system: atmospheric and oceanic subsystems. Running a coupled atmospheric oceanic model is computationally expensive, especially when an ensemble approach, requiring multiple model runs, is employed. To make the study computationally affordable, the authors considered the atmospheric subsystem of climate system as a mapping and used an NN to develop a fast nonlinear model of the atmospheric subsystem of climate system.

Other NN applications to subsystems of the atmospheric component of GCMs are shown in Fig. 2.1 at the third, fourth, and fifth hierarchical levels of GCM. They are discussed in detail in Chap. 4 together with several oceanic NN applications. For example, at the fourth level, the parameterizations of atmospheric physics are located in climate or weather prediction numerical models (see Fig. 2.1). The atmospheric long-wave radiation parameterization (see Chap. 4 for details and other examples) can be considered as a mapping. In this case, the input vector X is composed of several atmospheric state variables like temperature, humidity, and ozone concentration that are functions of height, and some surface characteristics. The output vector Y is composed of a function of height – long-wave heating rates, and heat fluxes. It becomes clear from this brief description that, in this case, we do not have a vector-to-vector mapping (2.1) but a functional mapping because

some of the components of the vectors X and Y are functions of a continuous variable – height. Nevertheless, by discretization of these functions on a vertical grid that transforms continuous functions into profiles (finite vectors), the problem can be converted to a vector-to-vector mapping (2.1). As in previous cases, here the components of the vectors X and Y may be significantly correlated because (1) they are physically related, and (2) they are related as the discretized values (elements of a profile) of the same continuous function where the elements are closely spaced in height. The target mapping is complicated and nonlinear in this case because the atmospheric radiation processes are complicated and nonlinear. The target mapping may be continuous or almost continuous, and, thus, it may contain a finite number of finite discontinuities (such as step functions) due to the impact of highly nonlinear atmospheric moisture processes.

2.2 Some Generic Properties of Mappings

Multidimensional, nonlinear mappings (2.1) are complicated mathematical objects that are not very well studied. These mappings have many different interesting properties. After the four previous examples, we now focus on certain generic properties of the mapping (2.1) that are typical and important for the applications presented in this book, keeping in mind that our goal is to develop an NN representation for the target mapping (2.1).

2.2.1 Mapping Dimensionalities, Domain, and Range

The first essential property of the target mapping is its *mapping dimensionalities*. A mapping is characterized by two dimensionalities: (1) the dimensionality n of the input space, $\Re^n$, and (2) the dimensionality m of the output space, $\Re^m$. In this book we will consider only mappings between vectors X and Y, which both consist of real numbers as their components, like mappings in the aforementioned examples. In this case, both input and output spaces are real vector spaces.

The second important property of the mapping (2.1) is the *mapping domain.* Only a part of the input space $\Re^n$ is spanned by the input vectors X. This part is called the mapping domain, D, and is determined by the particular application. Understanding the configuration of the mapping domain and its properties is essential for any application of the mapping (2.1) and for proper NN training and use (Bishop 1995). If all components of the input vector X are scaled to the range [−1.,1.], the volume of the input space $\Re^n$ is equal to 2^n and, therefore, grows exponentially with n. In computer, once the space is discretized on a grid by K values per dimension, then the problem grows even faster, as K^n. It means that in the input space we have K^n grid cells, and to represent our mapping, we need an exponentially large training set in

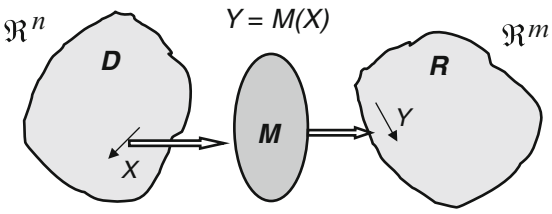

Fig. 2.2 The mapping (2.1), M, its input vector, X, output vector, Y, domain, D, and range, R

order to ensure that each grid cell contains at least one data point. This exponential growth in the amount of data with the increase of the input space dimensionality is often called the *curse of dimensionality* (Bishop 1995; Vapnik and Kotz 2006).

Fortunately, the components of the input vector, X, are usually interrelated or multi-collinear (Aires et al. 2004b) due to physical or statistical constraints that lead to both positive and negative consequences (see Sects. 2.3.4 and 2.4.2). On the positive side, these correlations effectively reduce the size, and sometimes dimensionality, of the part of the input space $\Re^n$ spanned by the input vectors X (the mapping domain, D). As a result, for a particular application, the mapping domain, D, may be significantly smaller than an n-dimensional cube in the input space $\Re^n$. On the negative side, it is often very difficult, if possible at all, to determine the actual shape and effective dimensionality of D, which makes it difficult to adequately sample the mapping domain D and to select the optimal architecture of the NN that emulates the target mapping.

The components of the output vector, Y, are also usually interrelated. As a result, the output vectors also span only a fraction of the output space $\Re^m$. This part of the output space is called the *range*, R. Understanding the properties of the range is very important for the proper testing and application of the NN approximations of a target mapping (2.1). Figure 2.2 illustrates the mapping (2.1), its domain, D, and range, R.

2.2.2 Mapping Complexity

Another property of mapping (2.1) that is important in the context of the applications reviewed in this book is *mapping complexity*. Mapping complexity is an intuitively clear notion. The mapping M performs a transformation of the input vector X to produce the output vector Y, and this transformation may be more or less complex. However, if we want to define complexity more precisely, we may encounter problems and ambiguities that are both qualitative and quantitative in nature (for a good introductory review of the topic and the related problems, see Reitsma (2001) and the references therein). Many different qualitative definitions (at least 30 according to Reitsma (2001)) of complexity have been introduced. For some of these definitions, corresponding quantitative definitions of complexity have likewise been proposed (Cilliers 2000; Gell-Mann and Lloyd 1996).

Keeping in mind the four examples of the mapping (2.1) we introduced earlier, we can suggest at least four different qualitative definitions of mapping complexity. In these examples, the target mapping, M, is a symbolic representation for a mathematical formalism based on first principles and describing a physical process or chain of interacting physical processes (e.g., atmospheric radiation). Therefore, we can talk about the *physical complexity* of the mapping (2.1) that corresponds to the complexity of the hierarchy of physical processes described by the mapping. We can also introduce quantitative or semiquantitative characteristics of physical complexity: the number of physical processes involved and the number of levels in the hierarchy of these processes involved (the hierarchical complexity; see Sect. 1.1).

The physical processes are represented mathematically by this mapping. Correspondingly, we can consider *mathematical complexity* of the mapping (2.1) and introduce quantitative or semiquantitative characteristics of mathematical complexity: the number of equations describing the physics, the type of equations (e.g., linear vs. nonlinear, ODE vs. PDE vs. integrodifferential equations), and the dimensionality of the equations. It is noteworthy that an ambiguity may exist for such measures of mathematical complexity because, for a particular physical process, alternative mathematical formalisms, based on first principles, often exist that lead to different types and numbers of equations for the description of the same physical system. As a result, several different estimates of physical and/or mathematical complexity may be obtained for the same target mapping (2.1). Euler vs. Lagrange formulations of the equations of geophysical fluid dynamics and the Schrödinger vs. Heisenberg formulations of quantum mechanics are examples.

The third type of complexity that can be introduced is mapping *numerical/computational complexity*. For this type of complexity, a quantitative measure, like the number of elementary numerical operations required for calculating Y given X, can be introduced. This measure is very important for it is closely related to the computation time. However, this measure is also ambiguous because, as we well know, different numerical schemes applied to the same set of equations (e.g., finite differences vs. variational methods for solving PDEs) may lead to dramatically differing counts of elementary numerical operations. Here again, several different estimates of numerical complexity may be obtained for the same mapping (2.1).

The fourth type of mapping complexity is called *functional complexity*. It describes the complexity of the functional dependency of the outputs, Y, versus the inputs, X, or the "smoothness" of this dependency. If the three previous definitions in some respects depend on, or are conditioned by, our knowledge of the internal structure of the target mapping (2.1), this fourth definition characterizes the complexity of the mapping as a whole, as a single, elementary object that transforms the input vector into the output vector. It is intuitively clear that the functional complexity of the mapping (2.1) can, in principle, be measured unambiguously. Unfortunately, it does not mean that there actually exist techniques that allow the introduction of a satisfactory measure of functional complexity for a multidimensional mapping (2.1). For example, for a function of one variable, an approximation procedure

can be used for measuring functional complexity. If n is the minimal order of a polynomial that approximates the function with the desired accuracy, the function may be considered to have polynomial complexity of order n. However, the direct generalization of this approach for the case of multidimensional mapping (2.1) is difficult, at best. A similar concept, however, can be applied using universal mapping approximators like MLP NNs. Thus, the complexity of the emulating NN can be used to measure the complexity of the target mapping (2.1) to be emulated by this NN. Appealing as this approach may appear, it still requires a clear definition of the accuracy of the NN emulation, which is discussed in Sect. 2.5. It also provides us with a measure of the mapping functional complexity post factum, after the NN approximation is performed; however, as we will show in the following chapters, we often need an estimate of the functional mapping complexity beforehand to develop an accurate NN approximation. It is noteworthy that the mapping dimensionalities n and m contribute to all of the types of mapping complexities considered above, although they cannot be used as unambiguous measures of these complexities.

2.2.3 Mappings Corresponding to Ill-Posed Problems

Among the applications considered in this book, we will find some problems that can be considered as continuous mappings (2.1); however, for these mappings small perturbations in X may cause large changes in Y; thus, the mapping is unstable. These problems belong to a wide class of ill-posed problems. In the case of the well-posed problem a solution usually exists, it is unique and stabile. Ill-posed problems do not satisfy one or several of these conditions (Vapnik 1995). Ill-posed problems usually arise when one attempts *to estimate an unknown cause from observed effects* (most of the geophysical inverse problems in ESS belong to this class, e.g., the satellite retrieval problem considered in Chap. 3), *to obtain the values of certain model parameters from the observed data*, or *to restore a whole object from its low dimensional projection* (e.g., estimating the NN Jacobian considered in Sect. 5.2). If X contains even a low level of noise, the uncertainties in Y may be very large. To solve ill-posed problems, additional a priori information about the solution (regularization) should be introduced into the approach to obtain solution (Vapnik and Kotz 2006).

2.2.4 Stochastic Mappings

In many practical applications, the mapping (2.1) contains an internal source of stochasticity. It may be due to several reasons: a stochastic process that the mapping describes, a stochastic method (e.g., Monte Carlo methods) implemented in mathematical formulation of the mapping, or uncertainties in the data that are

used to define the mapping. In this case we can modify the symbolic representation (2.1) introduced for the mapping above as

$$Y = M(X, \varepsilon) \tag{2.1a}$$

where ε is a vector stochastic variable reflecting explicitly a stochastic nature of the mapping – the mapping uncertainty. Assuming that stochastic part of the mapping is additive, representation (2.1a) can be simplified:

$$Y = M(X) + \varepsilon. \tag{2.1b}$$

It is noteworthy that the uncertainty ε is an inherent informative part of the stochastic mapping, which contains important statistical information about the mapping. *Actually, the stochastic mapping is a family of mappings distributed with a distribution function. The range and shape of the distribution function are determined by the uncertainty vector* ε. Examples of stochastic mappings are discussed in Sects. 4.3.6 and 5.2.

2.3 MLP NN: A Generic Tool for Modeling Nonlinear Mappings

2.3.1 NNs in Terms of Approximation Theory

The simplest MLP NN is a generic analytical nonlinear approximation or model for the target mapping (2.1). The MLP NN uses for the approximation a family of functions like

$$y_q = NN(X, a, b) = a_{q0} + \sum_{j=1}^{k} a_{qj} \cdot t_j; \quad q = 1, 2, \ldots, m \tag{2.2}$$

where

$$t_j = \phi\left(b_{j0} + \sum_{i=1}^{n} b_{ji} \cdot x_i\right) \tag{2.3}$$

and x_i and y_q are components of the input and output vectors, respectively; a and b are fitting parameters or NN weights; ϕ is a so-called activation or "squashing" function (a nonlinear function, often specified as the hyperbolic tangent); n and m are the numbers of inputs and outputs, respectively; and k is the number of the nonlinear basis function, t_j (2.3), in the expansion (2.2). The expansion (2.2) is a linear expansion (a linear combination of the basis function t_j (2.3)) and the

coefficients a_{qj} ($q = 1,\ldots,m$ and $j = 1,\ldots,k$) are the linear coefficients of this expansion. It is essential (see Sect. 2.4.1) that the basis functions t_j (2.3) are nonlinear with respect to inputs x_i ($i = 1,\ldots,n$) and to the fitting parameters or coefficients b_{ji} ($j = 1,\ldots,k$). As a result of the nonlinear dependence of the basis functions on multiple fitting parameters b_{ji}, the basis $\{t_j\}_{j=1,\ldots,k}$ becomes a very flexible set of nonorthogonal basis functions that have great potential to adjust to the functional complexity of the mapping (2.1) to be approximated. It has been shown by many authors in different contexts that the family of functions (2.2, and 2.3) can approximate any continuous or almost continuous (with a finite number of finite discontinuities, like step functions) mapping (2.1) (Cybenko 1989; Funahashi 1989; Hornik 1991; Chen and Chen 1995a, b). The accuracy of the NN approximation or the ability of the NN to resolve details of the target mapping (2.1) is proportional to the number of basis functions, k (Attali and Pagès 1997).

The MLP NN (2.2, and 2.3) itself is a particular type of mapping (2.1). In the case of the MLP NN, the *computational and functional complexity* of the NN mapping is closely related (they are especially close for NN emulations; see Sect. 2.5) and can be characterized by the number of fitting parameters a and b in (2.2, and 2.3). This number, the complexity of the MLP NN, is given by

$$N_c = k \cdot (n + m + 1) + m. \tag{2.4}$$

For a set of NNs approximating a particular target mapping (2.1) with a given number of inputs n and outputs m, a good measure of the NN complexity is the number of basis functions, k, that are used. The NN complexity grows linearly with the growth of the dimensionalities of the input space (the number of inputs), n, and the output space (the number of outputs), m.

The number of NN weights per NN output

$$n_c = N_c/m \tag{2.4a}$$

reflects the computational and functional complexity of the dependency of each NN output vs. NN inputs. This measure of the NN complexity is useful when NNs with different numbers of outputs are compared (see Sect. 4.3.3, subsection "Estimation of speedup").

It is noteworthy that, if for each mapping output y_q we construct a polynomial approximation, such a multidimensional polynomial of order P would have n^P unknown fitting parameters (Bishop 2006). Therefore, in the case of polynomial approximation, the computational complexity of the approximation for the entire mapping (2.1) is $m \cdot n^P$, which is a power law growth. The power law growth is slower than an exponential growth, but it is very fast and leads to the curse of dimensionality. Thus, the polynomial approximation is of limited practical utility for multidimensional function and mapping approximations. NNs manage to address the curse of dimensionality and, due to the aforementioned linear dependence on the dimensionality of the input space, remain a practical approximation (or model) even for high-dimensional mappings.

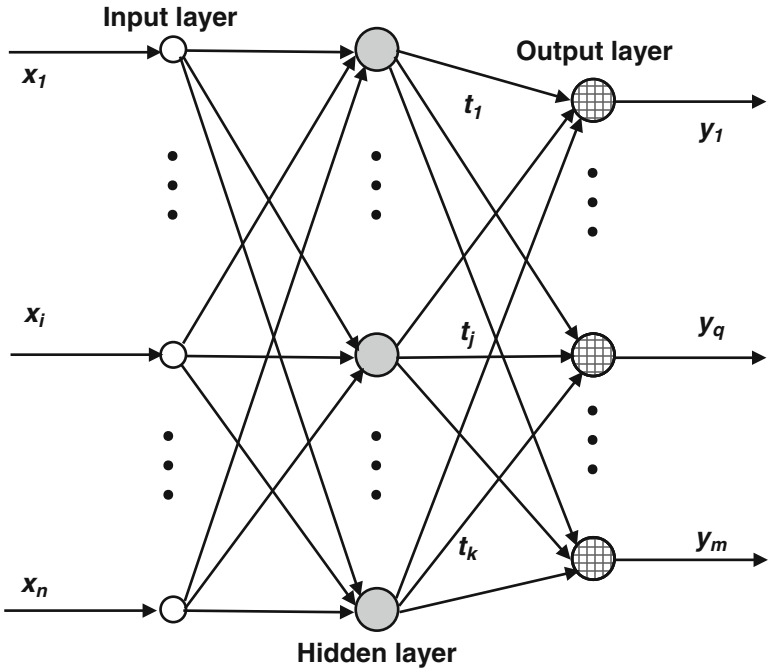

Fig. 2.3 The simplest MLP NN with one hidden layer and linear neurons in the output layer

2.3.2 NNs in Their Traditional Terms

A pictographic language reminiscent of a data flow chart is used traditionally in the NN field starting with the founding work by McCulloch and Pitts (1943). In this work devoted to the mathematical modeling of a *neuron*, a single cell in a neural network, a basis function t_j (2.3), or neuron was represented as shown in Fig. 2.4 (left). A step function was used in this work as the activation function, ϕ.

Then after Rumelhart et al. (1986) introduced the MLP NN, a pictographic representation of the entire NN was introduced (see Fig. 2.3). The neurons are situated into *layers* inside the MLP NN. The input layer is in a sense a symbolic layer. Input neurons do not perform any numerical function; they simply distribute inputs to neurons in the following hidden layer. The hidden layer (there can be several) is usually composed of nonlinear neurons (Eq. (2.3) and Fig. 2.4 left). The neurons in the output layer are usually linear (Eq. (2.2) and Fig. 2.4 to the right). The *connections* (arrows) in Fig. 2.3 correspond to the NN *weights*, the name used for the fitting parameters a and b in NN jargon.

Here we consider the simplest type of MLP NN that has one hidden layer and the output layer with linear neurons. Such architecture is sufficient for the

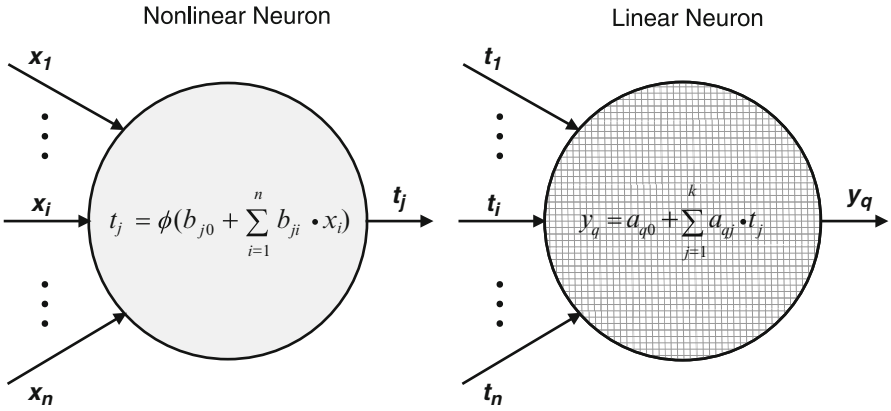

Fig. 2.4 The *right* figure shows linear (Eq. (2.2)) and the *left* – nonlinear (Eq. (2.3)) neurons

approximation of any continuous (or almost continuous) mapping (Cybenko 1989). More than one hidden layer and nonlinear neurons in the output layer may be introduced to solve specific problems (e.g., Hsieh 2004) or for practical convenience. For the simplest MLP NN considered here, there is a one-to-one correspondence between Eqs. (2.2, and 2.3) and Figs. 2.3 and 2.4. However, in general, the pictographic language (Figs. 2.3 and 2.4) is not redundant. This language can suggest NN topologies or architectures that probably cannot be represented analytically in terms of equations or that cannot evolve from Eqs. (2.2, and 2.3). The pictograms that represent the design of such NNs cannot be described by a closed set of equations; however, these pictograms can be translated into computer codes.

2.3.3 *Training Set*

In a practical application, a target mapping (2.1) is usually represented and presented to the NN by a data or training set that consists of N pairs of input and output vectors X and Y:

$$C_T = \{X_p, Y_p\}_{p=1,\ldots,N} \tag{2.5}$$

where $Y_p = M(X_p) + \xi_p$, $X_p \in D$ and $Y_p \in R$, and ξ_p represents any errors associated with the observations or calculation with a probability density function $\rho(\xi)$. The set C_T can also be considered as a combination of two rectangular matrixes,

$$C_T = \{C_X, C_Y\}, \tag{2.5a}$$

where C_X is a matrix of dimensionality $n \times N$ composed of all input vectors X and C_Y is a matrix of dimensionality $m \times N$ composed of all output vectors Y. The training set is all that the NN "knows" about the target mapping that it is expected to approximate. This is the reason why NN belongs to a class of *data-driven* or *learning from data* methods (Cherkassky and Mulier 2007).

The training set represents the mapping (2.1) for the NN, and, therefore, it has to be *representative*. It means that the training set has to have a sufficient complexity to represent the complexity of the target mapping, allowing the NN to achieve the desired accuracy of the approximation of the target mapping. The set should have a sufficient ample size, N, of properly distributed data points that adequately resolve the functional complexity of the target mapping (2.1). The training set should have finer resolution where the target mapping is not smooth and coarser resolution where it is smoother, namely, the domain D should be properly sampled. Certainly, it may be oversampled but not under-sampled. The fundamental question remains, however, as to just how we should measure this target mapping smoothness (or complexity) in order to obtain the desired results (DeVore 1998). The interrelations and correlations between inputs, as discussed earlier, simplify the sampling task for cases of high input dimensionality, reducing the size and the effective dimensionality of the domain.

The representativeness of the training set is a necessary condition for a good NN generalization (interpolation), for obtaining acceptable performance from an NN. Kon and Plaskota (2001) introduced a qualitative measure for the representativeness or necessary complexity of the training set. They introduced what they called *informational complexity*, which specifies the number of observations necessary and sufficient to construct an NN approximation under accepted assumptions. In an ideal situation, *there should be a correspondence between the functional complexity of the target mapping* (2.1)*, the complexity* (2.4) *of the approximating NN, and the informational complexity (number and distribution of data points) of the training set* (2.5). Unfortunately, there are no general recipes for most practical applications. The only useful relationship that can be found in the literature is $N > N_c$. Actually, this relationship has a simple statistical interpretation; the number of unknown parameters in the model (the number of weights in NN or the complexity of the NN (2.4)) should not exceed the number of data points, N, in the training set (2.5). It is noteworthy that NNs with $N_c \geq N$ can be considered if regularization procedures (see Sect. 2.3.7, subsection "Overfitting and regularization") are applied during the training.

As mentioned above and is discussed below, the training set should be somewhat redundant; however, an excessive redundancy with almost identical records in the training set does not improve the NN training and does obviously increase the NN training time. Various methods have been proposed to deal with such a redundancy of the training set. For example, Chevallier et al. (2000) introduce a sampling method that uses Euclidian or non-Euclidian distances in the input space $\Re^n$ to eliminate almost identical record from the training set. A dissimilarity index is introduced in this case,

$$D_n\left(X_i, X_j\right) = \left\| X_i - X_j \right\|$$

where $||\ldots||$ denotes a norm in the input space $\Re^n$. Only records with the dissimilarity indexes $D_n > d$ (a predefined distance in $\Re^n$) have been included in the training set.

Two distinct types of data (observed and simulated) are usually employed in the ES applications considered here. The first type of data is based on observations. These data usually contain a significant level of observational noise ξ. It is noteworthy that for an ill-posed problem (see Sect. 2.2.3 and Chap. 3), even a small level of noise in the data may lead to significant errors in the NN emulations. In the case of observed data, the sampling of the target mapping domain is controlled by the observation setup, technique, and conditions. Actually, in this case, because the target mapping is represented implicitly by the available observed data, the accuracy of the NN approximation and the ability of the NN approximation to resolve the target mapping are limited and determined by the observation setup, technique, and conditions. Two important problems when working with observed data are the data quality control and missing data handling (see Sect. 2.3.7, subsection "Missed Inputs and Outputs" for a brief discussion of the missing data problem). There is usually little we can do to improve or expand the data set in the case of observed data except to fuse it with simulated (model produced) data if such data can be produced.

If an explicit theoretical (based on first principles) or empirical model for the target mapping (2.1) is available, it can be used to simulate the data set (2.5). With simulated data, we have significantly more control over the sampling of the target mapping domain (the number and distribution of the data points) and as a result, on the NN accuracy and the ability of the emulating NN to resolve the target mapping. The level of noise in the simulated data is usually lower than that in the observed data. The simulated data do not have missing data per se; however, in some application a problem may arise, which is similar to the missing data problem (see Sect. 2.3.7, subsection "Missed Inputs and Outputs"). The simulated and observed data can, in principle, be merged or fused together to form an integrated data set using an appropriate technique that is able to account for the different error statistics and statistical properties of these two data types. One example of properly fused data is the analyzed data produced by a DAS (see Sect. 3.1.2).

2.3.4 Selection of the NN Architecture

To approximate a particular target mapping (2.1) with the MLP NN (2.2, and 2.3), we should first select the NN architecture or topology, the number of the inputs n, the outputs m, and the number of neurons k in the hidden layer. For each particular problem, n and m are determined by the input and output dimensionalities of the target mapping (the dimensions of the input and output vectors X and Y). Here we treat an entire mapping (2.1) as an elementary/single object and approximate its functionality (an input–output relationship) in its entirety. Practical implementation of this approach allows for multiple solutions in terms of the number of NN designs that can be used for an approximation. As a result, the MLP NN expressed in

Eqs. (2.2, and 2.3) can be implemented as a single NN with m outputs, m single-output NNs, or several multiple-output NNs with the total number of outputs equal to m.

Approximating the target mapping with a single NN is a convenient solution because of the simplicity of its design. It also has a great advantage in terms of speeding up the calculations when the outputs of the mapping and, therefore, the outputs of the approximating NN are significantly correlated. In the case of a single NN (2.2, and 2.3) with many outputs, all the outputs are different linear combinations of the same basis functions t_j or hidden neurons. Fewer neurons are required to approximate a particular number of correlated outputs than to approximate the same number of uncorrelated outputs. Thus, in the case of correlated outputs, one NN per approximation solution has a lower complexity N_c (2.4) and provides significantly higher performance at the same approximation accuracy than a battery (an array) of m single-output NNs (Krasnopolsky and Fox-Rabinovitz 2006). Also, a single emulating NN solution is less complicated in terms of NN training requirements. It leads to a lower dimensionality of the training space (see Sect. 4.3.4, subsection "Normalization of NN outputs, and an array of NNs Versus a single NN" for an example and discussion).

The number of hidden neurons k that determine the complexity (2.4) of the approximating NN in each particular case should be determined when taking into account the complexity of the target mapping to be approximated. The more complicated the mapping, the more hidden neurons k are required (Attali and Pagès 1997) (or the higher the required complexity N_c of the NN) to approximate this mapping with the desired accuracy or resolution. There is always a trade-off between the desired resolution of the target mapping and the complexity of the NN emulation. However, from our experience, the complexity k of the approximating NN should be carefully controlled and kept to the minimum level sufficient for the desired accuracy of the approximation to avoid overfitting and to allow for a smooth and accurate interpolation (see the discussion in Sect. 2.4). Unfortunately, in this regard, there are no universal rules or recommendations that can be given. Usually k is determined based on experience and experimentations.

The possibility of choosing among many topological solutions, from a single NN with m outputs to m single-output NNs, demonstrate the internal flexibility of the NN technique. This additional flexibility can be effectively used in many applications.

2.3.5 *Normalization of the NN Inputs and Outputs*

Another degree of flexibility is provided by the availability of different *normalizations* for NN inputs and outputs. NN inputs are usually normalized to an interval $[-a,a]$, using a simple equation,

$$\tilde{x}_i = a_i \cdot \frac{\left(2 \cdot x_i - x_i^{\max} - x_i^{\min}\right)}{\left(x_i^{\max} - x_i^{\min}\right)} \tag{2.6}$$

where x_i is the ith NN input before and $\tilde{x}_i$ after the normalization. If all a_i are equal to 1, then all inputs are normalized to the interval [−1.,1.]. By selecting different a_i for different inputs, we can change the sensitivity of NN to the variability of a particular input.

For NN with a single linear output, normalization of the output is relatively straightforward. Any traditional normalization, like normalizing over the interval [−a,a] via (2.6) or using the following normalization,

$$y' = \frac{y - \bar{y}}{\sigma} \tag{2.7}$$

where $\bar{y}$ is the mean value of y and σ is its standard deviation (SD), can be used and leads to similar approximation errors.

For a single NN with multiple outputs, the normalization of the outputs affects the approximation accuracy and NN performance more significantly than in the case of a single-output NN. Normalization similar to (2.7) for the case of multiple outputs can be written as

$$y'_q = \alpha \cdot \frac{y_q - \bar{y}_q}{\sigma_q} \tag{2.8}$$

where $\bar{y}_q$ and σ_q are the mean and SD of the qth output, y_q, and $\alpha \leq 1$ is introduced to accelerate the training of the linear weights in the output layer of the NN. This normalization improves approximation accuracies for small outputs; however, if these outputs are noisy, it propagates the noise to other outputs. Normalization (2.8) also reduces correlations that may exist between outputs.

An alternate normalization,

$$y'_q = \alpha \cdot \frac{y_q - \bar{y}_q}{\sigma} \tag{2.9}$$

where σ is the SD for all outputs, can also be employed in the case of NNs with several outputs. This normalization preserves correlations between outputs. Via taking into account these correlations, the normalization allows to reduce complexity and to improve performance of the emulating NN.

In the case of multiple outputs, the four different normalizations (2.6, 2.7, 2.8, and 2.9) lead to very different approximation errors and to different types of error distributions between outputs (see Sect. 4.3.4, subsection "Normalization of NN outputs, and an array of NNs Versus a single NN" for examples). For different NN applications, certain types of error distributions may be desirable; for example, smaller absolute or relative errors may be preferable. Different output normalizations in the case of a single emulating NN with multiple outputs may provide an additional tool for obtaining the desired result. This topic is also discussed by Krasnopolsky and Fox-Rabinovitz (2006).

2.3.6 Constant Inputs and Outputs

In practical applications some components of input and/or output vectors X and Y of the mapping (2.1) may have constant or almost constant values (their SDs are very small). In such cases, when the mapping is emulated with NN, constants should not be included in inputs or outputs because (1) they carry no information about input/output functional dependence and (2) if not removed they introduce additional noise in training and result in additional approximation errors. As for values that are almost constant, these small signals may be in some cases not a noise but very important signals; however, in such cases they should be specially treated to be used as NN inputs or outputs. For example, such inputs or outputs should be converted to their anomalies (i.e., their means should be subtracted); then the anomalies should be used as inputs or outputs for the emulating NN. However, depending on the level of uncertainty in the problem, information that these small signals may provide may be well below the level of uncertainty, and may be in many cases practically useless.

2.3.7 NN Training

After the NN architecture (topological parameters n, k, and m) is defined, the weights (a and b) can be found using the training set C_T (2.5) and the maximum likelihood method (Vapnik 1995) by maximizing the likelihood functional

$$L(a,b) = \sum_{i=1}^{N} \ln \rho\left(\xi_i\right) \tag{2.10}$$

with respect to the free parameters (i.e., the NN weights) a and b. Here, $\rho(\xi)$ is the probability density function for the approximation errors $\xi_i = Y_i - NN(X_i, a, b)$ (see Sect. 2.3.3) and the summation is performed over the N records in the training set. If the errors ξ_i are normally distributed, Eq. (2.10) leads to the minimization of the mean-square error function (also called the loss, or risk, or cost function) with respect to the NN weights given by W (a and b),

$$E(W) = \frac{1}{N}\sum_{i=1}^{N} \xi_i(W) = \frac{1}{N}\sum_{i=1}^{N} (Y_i - Z_i)^2 \tag{2.11}$$

where $Z_i = NN(X_i, W)$, E is the total error calculated over the entire training set (all N records of the training set are included), and $\xi_i = (Y_i - Z_i)^2$ is an error corresponding to the ith record in the training set. This procedure of minimization of the error function (2.11) is usually called *NN training*. Minimizing the error function is performed in the W-space (the space of NN weights or the training space), which has a dimensionality equal to the number of NN weights, N_C (2.4).

It is noteworthy that for a probability density function $\rho(\xi)$ other than Gaussian, the error function should be derived from the maximum likelihood functional (2.10). The error function may be significantly different than the mean-square error or loss function (2.11) (Liano 1996). However, in the majority of applications, the mean-square error function (2.11) is applied because it is simple and analytically differentiable.

Optimal values for the weights are obtained by minimizing the error function (2.11); this task is a nonlinear minimization problem. A number of methods have been developed for solving this problem (Bishop 1995; Haykin 2008). Here we briefly outline one of them, a simplified version of the steepest (or gradient) descent method known as the back-propagation training algorithm. It was introduced for practical applications by Werbos (1974) and as an NN training algorithm by Werbos (1982) and Rumelhart et al. (1986).

The back-propagation training algorithm is based on the simple idea that searching for a minimum of the error function (2.11) can be performed step by step iteratively, and that at each step we should increment or decrement the weights in such a way as to decrease the error function. This can be done, for example, using the following simple steepest descent rule,

$$W^{(n+1)} = W^{(n)} - \eta \frac{\partial E\left(W^{(n)}\right)}{\partial W} \tag{2.12}$$

where W is either one of two weights (a or b), $W^{(n+1)}$ is an adjusted or updated weight, $\eta > 0$ is a so-called learning constant, and $W^{(n)}$ is the weight at the previous nth iteration. The total error function E or a one record error function ξ_i can be used in (2.12) which leads to different training procedures, *batch* training, or *sequential* (or *online*) training (see next subsection).

Using (2.11), (2.2, and 2.3), and the chain rule of differentiation, the derivative in (2.12) can be expressed analytically through the derivative of the activation function ϕ, and through the weight values at the previous iteration step (Haykin 2008; Bishop 1995, 2006). At the first step when we do not have weights from a previous training iteration, a weight initialization problem arises that is familiar to those who use various kinds of iterative schemes. Many studies have been devoted to weight initialization (e.g., Nguyen and Widrow 1990; Wessels and Bernard 1992). Most of these procedures initialize the NN weights with small random numbers. Wessels and Bernard (1992), for example, generate random weights in the range $\left[-3\left/\sqrt{n}\,;3\left/\sqrt{n}\right.\right.\right]$, where n is the number of NN inputs.

The nonlinear error function (2.11) has multiple local minima. Moreover, due to the symmetry of MLP NN in the weight space, for each of these local minimum there exist $k!{\cdot}2^k$ ($k!$ is k-factorial) clone local minima with exactly the same error (Chen et al. 1993). The back-propagation algorithm converges to a local minimum, as does almost any algorithm available for solving the nonlinear optimization problem (NN training). Usually, multiple initializations (even multiple initialization procedures) are applied to avoid shallow local minima and to choose a local minimum with a sufficiently small error.

Batch Training and Sequential Training

If in Eq. (2.12) the total error function E is used to calculate the new adjusted weights; therefore, the entire training set has to be processed at each training iteration. Thus, at each training step, the weights are shifted in the direction of the greatest decrease in the total error function. This type of training that uses the entire data set to calculate each weight adjustment is called *batch* training.

Alternatively, we can use error function ξ_i in Eq. (2.12), which is one record (i.e., one pattern) error, for the weight adjustment; that is, the weights are updated after each data record from the training set is presented to the NN. This training approach is called *sequential* or *online* training. Sequential training works with the training set record by record until all patterns have been presented once to NN, which is called an *epoch*; then the process may be repeated many times (many epochs) cycling through the records of the training set in sequence or selecting patterns randomly. Thus, in the batch training, weights are updated once per epoch, while in the sequential training, weights are updated after each data pattern (record) or N times per epoch.

The batch approach is meaningful from the statistical perspective. It is often effective, and certain second-order optimization algorithms work better with batch training (Hsieh 2009); however, certain problems related to this algorithm can arise; they are discussed by Bishop (2006) and Hsieh (2009). The sequential approach has the obvious advantage in the case of long training sets: it works with one record at a time, which makes it independent of the number of N patterns in the training set. Thus, the sequential approach can be successfully used when dealing with data that arrive in real time. Each new data record can be used independently to update the NN weights online. Also, sequential training allows avoiding local minima of the total error function.

Missed Inputs and Outputs

In practical applications, some elements of the data matrixes C_X and/or C_Y, constituting the training set C_T (2.5a), may be missing or corrupted. Obviously, this problem often occurs when working with observed data. However, the problem of missing data or a similar problem also arises when using simulated data. In one of the applications discussed in Chap. 4, the mapping (2.1) describes the entire atmospheric physics; the output vector in this case includes, among others, two physical parameters: the land temperature at a given point and the ocean temperature at the same point. Obviously, at a particular location, only one of these two parameters is valid, and another one is missed. Thus, in this case, each record in the training set simulated by a numerical model contains at least one missing value.

Missing data is an important technical problem because the majority of multivariate data modeling and analysis techniques (including NNs) require complete data sets (all variables have to be represented for each data record). Rates of missing data less than 1 % are usually considered trivial, 1–5 % manageable, a rate of 5–15 % requires sophisticated methods to handle it, and more than 15 % may have a severe impact on the quality of the model (Luengo et al. 2010).

Several approaches have been proposed to treat missing data. The most economical way of dealing with the problem and to obtain complete data set is by deleting the records with missing data, i.e., working with a subset of the training set. This method, however, may become unacceptable in the case of small data sets, especially in the case of high-dimensional input and output vectors X and Y. Indeed, if only one element (one variable) of either vector is missing, the entire record (X_p, Y_p), which contains $n + m - 1$ acceptable variables, has to be removed. In the previous example that involves atmospheric physics, application of this method will lead to deleting the entire data set. Fortunately, a number of less economical but more sophisticated methods have been developed to deal with missing data (Richman et al. 2009). For example, a more sophisticated approach would be to use the maximum likelihood procedure, where the parameters of a model for the complete data are estimated. This model can be used to impute missing data. Finally, in the majority of cases, data set components are not independent from each other. Thus, through the identification of relationships between components, missing values can be determined and imputed.

The detailed discussion of the problem of missing data goes beyond the scope of this book. However, it is discussed by Richman et al. (2009) and Luengo et al. (2010); various methods of data imputation or replacement of missing data are considered in these works and in papers cited there. In this text, we mention one simple method of dealing with missing output data that is used in applications (Krasnopolsky et al. 2009) discussed in subsequent chapters. This method is equally effective if one or more components of the output vector Y_i are missing. In the application of this method, the error function (2.11) is modified by introducing in the error function a binary matrix α_{iq},

$$\bar{E}(W) = \frac{1}{N} \sum_{i=1}^{N} \sum_{q=1}^{m} \alpha_{iq} \cdot \left(y_q^i - z_q^i (X_i, W) \right)^2 \tag{2.11a}$$

where matrix, α, is defined in accordance with the following rule: for the training record number i,

$$\alpha_{iq} = \begin{cases} 1, \text{if the output number } q \text{ is defined} \\ 0, \text{if the output number } q \text{ is missing} \end{cases}.$$

Thus, the method avoids deleting the entire record number i from the training set. All components of the vector that are not missing can be used for training. Missing components are simply not included in the error function (they are included with zero weight).

Overfitting and Regularization

In this section we discuss the problem of overfitting for two different cases. First, when the level of noise in the training data is low and an NN or any other nonlinear

statistical model approximates the data well, it may occur that between the data points and/or at the ridges of the domain, the NN exhibits an unpredictable behavior (e.g., wild oscillations). Second, when dealing with noisy data containing outliers, the model, if the training is excessive, may fit the noise and outliers as well as the desired data. There are a number of reasons for overfitting. For example, the NN complexity, N_C, can be unreasonably high and approach or exceed the number of data points, N, in the training set. Also, the training set can be long (large N) but redundant; a simple target mapping can then be oversampled. In this case, NNs of high complexity can be selected based on the large value of N; however, because of the redundancy in the data, overfitting will occur.

One of the classic manifestations of overfitting is the convergence of the training process to a local minimum with large weights of alternating signs. To avoid converging to such local minima, *regularization* is often applied (Haykin 2008; Bishop 1995, 2006), which involves adding a penalty term to the error function (2.11). A modified (or regularized) error function can be written as

$$\tilde{E}(W) = E(W) + \lambda \cdot \sum_j W_j^2. \tag{2.11b}$$

The first term in Eq. (2.11b) is the standard error function (2.11), the second term is the regularization term, and λ is a coefficient that reflects the relative importance (or strength) of the regularization term. It is obvious that the regularization term in Eq. (2.11b) penalizes large weights providing guidance for the training procedure, based on the regularized error function (2.11b), to a solution that corresponds to a local minimum with smaller weights and thus preventing overfitting due to large weights.

It is noteworthy that in the case of complex NNs with a large number of weights, the dimensionality of the training space, N_C, is very high (hundreds of thousands in some applications discussed in the following chapters). As a result, the number of local minima in the error function can be very large. In such situations, NN training usually leads to one of the closest (to the initialization point) local minima. If initialization procedure is used that initializes NN weights with small random numbers (e.g., Nguyen and Widrow 1990), the training process will be attracted to a local minimum, which is close enough to the initialization point and where the weights will still be relatively small. Thus, application of an initialization procedure, like one just described, may make the use of regularization (2.11b) unnecessary for complex NNs.

Noisy Training Data and Stochastic Mappings

If NN is trained using the data that contain a significant level of noise or uncertainty, these data actually represent a stochastic mapping with an uncertainty ε (see Sect. 2.2.4). In this case, the NN emulation of the stochastic mapping (2.1b) can be written as

$$Y = M_{\mathrm{NN}}(X) + \varepsilon + \varepsilon_{\mathrm{app}}$$

where M_{NN} is an NN emulation of the mapping M (2.1b) and $\varepsilon_{\mathrm{app}}$ is an NN approximation error. Thus, in the case of the stochastic mapping (2.1b), the NN emulation task is different from that of emulating an exact mapping (2.1) which does not contain uncertainty ε.

This important difference should be taken into account when the NN approximation is trained, the approximation error statistics are analyzed and interpreted, and the NN architecture is selected. In the case of training, the usually used criterion of minimum of the error function (2.11) should be substituted by the requirement that the error should not exceed the uncertainty ε or

$$E(W) = \frac{1}{N}\sum_{i=1}^{N}[Y_i - M_{\mathrm{NN}}(X_i)]^2 \le \varepsilon^2. \tag{2.11c}$$

All NNs that satisfy the condition (2.11c) are valid emulations of the stochastic mapping (2.1b). It is clear that any estimate of the magnitude of the uncertainty ε is of paramount importance for the proper training of NN emulation of a stochastic mapping.

2.4 Advantages and Limitations of the NN Technique

Here we summarize the advantages and limitations of the MLP NN approach as applied to the emulation of complex multidimensional mappings (2.1). It is noteworthy that the majority of limitations we discuss here are not limitations of the MLP NN technique per se. These limitations are inherent with regard to nonlinear models, nonlinear approximation techniques, and nonlinear statistical approaches in general (Cheng and Titterington 1994). Also, the same feature of the NN technique that gives this technique a significant advantage under the normal circumstances is sometimes responsible for some of the limitations on the NN technique under special conditions. We will proceed with the discussion while keeping these two points in mind.

2.4.1 Flexibility of the MLP NN

The MLP NN is a universal and very flexible approximator. The great flexibility of the MLP NN is due to the fact that the basis functions (hidden layer neurons) t_j (2.3) are adjustable. They contain many internal nonlinear parameters b that can be adjusted during training process. Thus, the basis functions are not specified a priori; they are determined during the training process and "optimized" for a particular

mapping to be approximated. Barron (1993) showed that, for certain classes of mappings, a linear combination of such adjustable basis functions can provide an accurate approximation with far fewer functions than a linear combination of any fixed or rigid basis functions that contain no adjustable nonlinear parameters. Similar results were obtained by Krasnopolsky and Kukulin (1977). This is one way for the MLP NN to escape the curse of dimensionality.

Another way to look at the importance of the adjustable basis t_j is to demonstrate the independence of the approximation error with respect to the dimensionality n of the input space. When flexible basis functions (2.3) are used, the approximation error is $E \leq \frac{\alpha}{k^p}$, where $\alpha > 0$, $p > 0$, and p is independent of n. In contrast, for approximations using fixed basis functions, the approximation error is $E \leq \alpha / k^{\frac{1}{n}}$ for the same class of mapping (Barron 1993; Cheng and Titterington 1994). Therefore, for a fixed basis expansion, when the number of inputs increases, one needs more and more basis functions (the number of basis functions, k, has to increase) to achieve the same accuracy of approximation. Thus, for the MLP NN, it is the number of hidden neurons k, not the dimensionality of the input space, that determines the accuracy of the approximation.

The flexibility of the MLP NN may also lead to undesired consequences. The basis functions t_j are very flexible, nonorthogonal, and overlapping. These factors may lead to non-optimality or redundancy in the NN architecture. As a result, some of the hidden neurons may contribute very little to the approximation and could be removed by "pruning" without a significant impact on the approximation accuracy. Pruning and similar techniques (Bishop 1995; Haykin 2008) have been developed to optimize the NN architecture and complexity.

Thus, the flexibility of the MLP NN technique, if not properly implemented and controlled, may lead to unwanted consequences like overfitting (fitting the noise in the data), unstable interpolation, and uncertain derivatives. These limitations and the ways to control them will be discussed again below.

2.4.2 *NN Training, Nonlinear Optimization, and Multi-collinearity of Inputs and Outputs*

NN training, as described in Sect. 2.3.7, is an iterative procedure that does not involve any matrix inversion. It is robust, insensitive to multi-collinearities in input and output data, and always leads to a solution for the NN weights. On the other hand, as a nonlinear optimization, NN training always has multiple solutions that correspond to multiple local minima in the error or loss function (2.11). Multi-collinearities in the input and output data lead to an equalization of local minima, especially in the case of a higher input dimensionality. Therefore, multi-collinearities in input and output data partly alleviate the problem of seeking the local minimum with the smallest error among multiple local minima. From the point of view of the approximation problem, all of these local minima give similarly good solutions because the approximation errors for these minima are small and

approximately the same. On the other hand, these local minima, which are almost equivalent in terms of the approximation error, give different solutions in terms of the NN weights. These different NNs lead to different interpolations and different derivatives. Thus, because of the equalization of errors corresponding to different local minima, the approximation error may not be a sufficient criterion; hence, using additional criteria may be necessary for selecting solutions with acceptable interpolation properties and derivatives.

2.4.3 NN Generalization: Interpolation and Extrapolation

One of the vaguest terms in NN parlance is "generalization" or "generalization ability." This term came from the field of cognitive science and implies an acceptable performance of the trained NN for new inputs that were not included in the training set. However, it is clear that there are at least two different cases of generalization. In the first case, new inputs are located inside the domain, D, "between" the training data points. In the second case, new inputs are located beyond the area covered by the training set, namely, close to or outside of the boundary of the domain D. The first case corresponds to interpolation and the second to extrapolation.

It is well known that nonlinear extrapolation is an ill-posed problem, and its solution may require regularization (introducing additional information) (Vapnik 1995). We will not discuss nonlinear extrapolation here. However, even smooth interpolation is not guaranteed if the only criterion used for NN training is small approximation error (2.11). Moreover, multiple local minima with similarly small approximation errors may still lead to different interpolations. If the NN complexity is not controlled, overfitting may occur that may lead to poor interpolations, e.g., significant oscillations between training data points. As mentioned in Sect. 2.3.3, the representativeness of the training set is a necessary condition for acceptable interpolations. Additional measures for improving the interpolation ability of the NN approximation are discussed in Sect. 2.5.

2.4.4 NN Jacobian

The NN Jacobian, J, is an $m \times n$ matrix of the first derivatives of the NN outputs over the inputs,

$$J = \left[\frac{\partial y_q}{\partial x_i} \right]_{i=1,\ldots,n}^{q=1,\ldots m} \tag{2.13}$$

and may be useful in many cases. For example, in data assimilation applications (see Sect. 3.1.2), the Jacobian is used to create an adjoint (a tangent-linear

approximation) of the target mapping. The Jacobian is also instrumental in statistical sensitivity, robustness, and error propagation analyses of the target mapping and its NN emulation. An inexpensive, simple computation of the NN Jacobian by analytical differentiation of (2.2, and 2.3) is one of the advantages of the NN approach. However, Jacobian is not trained; it is simply calculated through a direct differentiation of a trained NN. In this case the statistical inference of a Jacobian represents an ill-posed problem, and it is not guaranteed that the derivatives will be sufficiently accurate. Moreover, the existence of multiple minima of the error function with similar approximation errors and different NN weights implies that there exist multiple solutions for emulating NNs that have similar approximation and interpolation errors but different Jacobians.

As mentioned in Sect. 2.4.3, if additional care is taken during the training process, NN emulations can demonstrate acceptable interpolation properties (Krasnopolsky and Fox-Rabinovitz 2006). Thus, on average, the derivatives of these emulations are sufficiently accurate to provide satisfactory interpolations. However, for certain applications, such accuracy of an NN Jacobian may not be sufficient. For those applications that require an explicit calculation of the NN Jacobian, several solutions have been offered and investigated:

1. The Jacobian (or the entire adjoint) can be trained as a separate NN (Krasnopolsky et al. 2002). Generation of a data set for training a Jacobian or adjoint is usually not a significant problem in those cases where simulated data are available.
2. An ensemble approach can be applied that uses an ensemble of NN emulations with the same architecture corresponding to different local minima of the error function or uses an ensemble of NN emulations with different numbers of hidden neurons (different complexities) to stabilize the NN Jacobian or to reduce the uncertainties of the NN Jacobian (Krasnopolsky 2007) (see also Sect. 5.2.2).
3. The mean Jacobian can be calculated over the entire data set (Chevallier and Mahfouf 2001) and used for further calculations, if necessary.
4. Regularization techniques like "weight smoothing" (Aires et al. 1999) or principal component decomposition (Aires et al. 2004b) can also be used to stabilize the Jacobians.
5. The Jacobian can be included in the NN architecture as additional outputs that can be trained.
6. The error or cost function $E(W)$ (2.11), which is minimized in the process of NN training, can be modified to accommodate the Jacobian; in other words, the Euclidian norm, which is usually used for calculating the error function, should be changed to the first-order Sobolev's norm error function, according to

$$E_J(W) = E(W) + \lambda \cdot \sum_{i=1}^{N} \| J\,(X_i) - J_{\mathrm{NN}}\,(X_i, W) \|^2$$

where J is the Jacobian matrix of mapping (2.1) (observed or simulated), J_{NN} is the Jacobian matrix of emulating NN, $||\ldots||$ denotes the matrix norm, and λ is a constant reflecting relative importance of the two terms constituting the modified error function $E_J(W)$. With this change from Euclidian to Sobolev's norm, the NN is trained to approximate not only the target mapping (as with the Euclidian norm) but also the mapping's first derivatives. This solution does not change the number of the NN outputs; however, it may require using more hidden neurons and may significantly complicate the minimization during the training since the complexity of the error function increases. Hornik et al. (1990) have shown that the function of Sobolev's space with all their derivatives can be approximated by an NN. This and other similar theoretical results are very important because they prove the existence of the approximation; however, they do not suggest explicit approaches. Some explicit approaches have been presented elsewhere (Cardaliaguet and Euvrard 1992; Lee and Oh 1997).

Solutions 5 and 6 require an extended training set that includes first derivatives. This requirement cannot usually be met when working with high-dimensional mappings represented by observed data. When working with data simulated by a physically based model, it is usually not difficult to additionally simulate the derivatives. Finally, it should be mentioned that Jacobian modeling for large NNs still remains an open issue.

2.4.5 Multiple NN Emulations for the Same Target Mapping and NN Ensemble Approaches

Nonlinear models and approximations have many nonlinear parameters that could change during the process of generating solutions (fitting the data), which makes the models very flexible and easily adjustable to a selected target mapping. Different combinations of these parameters may lead to multiple solutions with the same or almost the same values of approximation errors. The existence of multiple solutions is a generic property of nonlinear models. The multiple solutions may be almost identical or similar in terms of a particular criterion (e.g., error function) that is used to obtain the solutions. At the same time these models (e.g., NNs) may be different in terms of other criteria that provide complementary information about the target mapping. The availability of multiple solutions may at times be inconvenient and lead to uncertainties, e.g., the necessity of introducing an additional step to use additional criteria to select a single "optimal" model (solution). On the other hand, the availability of multiple models (e.g., NN emulations), providing complementary information about the target mapping, opens up opportunities to use an ensemble approach that allows integration of the complementary information contained in the ensemble members into an ensemble that collectively "knows" more about the target mapping than does any of the individual ensemble members (individual NN emulations).

The idea that an ensemble of learning models consisting of many members is capable of providing a better description of the system than any individual member model can be traced back to the late 1950s and mid-1960s (Selfridge 1958; Nilsson 1965). Since the early 1990s, many different algorithms based on similar ideas have been developed for NN ensembles (Hansen and Salamon 1990; Sharkey 1996; Naftaly et al. 1997; Opitz and Maclin 1999; Hsieh 2001).

An ensemble of NNs consists of a set of members, i.e., individually trained NNs. They are combined when applied to new data to improve the generalization (interpolation) ability because the previous research has shown that an ensemble is often more accurate than any single ensemble member. Various ways of combining NN ensemble members into the ensemble have been developed (Naftaly et al. 1997) (see also Sect. 5.3). Previous research also suggests that any mechanism that causes some randomness in the formation of the NN members can be used to form a more accurate NN ensemble (Opitz and Maclin 1999). For example, ensemble members can be created by training different members on different subsets from the training set (Opitz and Maclin 1999), by training different members on different subdomains of the training domain, by training different members using NNs with different architectures (different numbers of hidden neurons) (Hashem 1997), or by training different members using NNs with the same architecture but different initial conditions for the NN weights (Maclin and Shavlik 1995; Hsieh 2001).

Most previous work with NN ensembles has been done in the context of solving the classification problem (Hansen and Salamon 1990; Sharkey 1996; Opitz and Maclin 1999) or in time series prediction (Naftaly et al. 1997; Hsieh 2001). In the present context, in the approximation of a complex mapping (2.1), the members of the ensemble are separately trained approximating NNs, which provide different accuracies of approximation for the target mapping and different interpolations. Thus, we can expect that the ensemble average will provide higher-quality approximations and interpolations than the individual members (see Sect. 5.2.1). Krasnopolsky (2007) also applied the NN ensemble technique to reduce the uncertainty of the NN Jacobian (see Sect. 5.1.2).

2.4.6 NN Ensemble as Emulation of Stochastic Mappings

In subsection "Noisy training data and stochastic mappings" (Sect. 2.3.7) it was shown that when emulating stochastic mappings, multiple NNs that satisfy the criterion (2.11c) are valid emulations of the stochastic mapping (2.1b). Actually, we can assume that each of these NNs emulates a member of the family of mappings that together represent the stochastic mapping (2.1b). Therefore, all the NNs satisfying (2.11c) together – the entire ensemble of NNs – emulate the stochastic mapping (2.1b).

2.4.7 Estimates of NN Parameters' Uncertainty

The NN technique is a nonlinear statistical approach. As with any statistical approach, the NN technique is expected to provide not only an estimate of model parameters and outputs (through the minimization of an error or loss function) but also an estimate of the uncertainties in the NN weights and outputs. Because of the nonlinear nature of NNs, estimation of the NN uncertainties is a more complicated problem than that in the linear case. However, during the last decade progress has been made in this field for the cases of both the MLP NN with a single output (MacKay 1992; Bishop 1995; Neal 1996; Nabney 2002) and for multiple outputs (Aires et al. 2004a). Various Bayesian methods have been used in these studies for estimating the uncertainties of NN parameters (weights).

2.4.8 NNs Versus Physically Based Models: NN as a "Black Box"

Outputs of a physically based (PB) model may be related or correlated due to the physical laws (e.g., various conservation laws) and/or equations implemented in the model. NN emulations do not reproduce these relationships exactly. An accurate NN emulation approximates aforementioned relationships and/or correlations with an accuracy limited by the approximation error. If a higher accuracy is desired, a procedure similar to the regularization procedure (2.11b) can be used to include the error in the desired condition as a penalty function in the training process and to minimize this error. Also, the desired relationship may be superimposed on NN outputs exactly as a post-processing step (see Sect. 4.3.3, subsection "Balancing LWR and SWR heating rates").

One of the often-mentioned shortcomings of NNs is that they do not offer a straightforward physical interpretation (e.g., Zorita and von Storch 1999), as do PB models or simple linear statistical models, i.e., that NNs are not transparent "black boxes." In general this is true; it is difficult, if not impossible, to give a physical interpretation to the NN weights, but consider the following. NNs are admittedly more complex and less transparent than simple linear models and regressions. However, NNs are never used (or should never be used) for problems that can be solved using linear models. NNs are usually applied to model or emulate multidimensional, complex, nonlinear systems that, in principle, cannot be modeled or emulated by simple linear models. Thus, NNs should not be compared with linear approaches, which are not adequate in these cases. NNs should be compared with other nonparametric multidimensional statistical approaches or complex deterministic numerical models that are used for modeling multidimensional, complex, nonlinear systems. However, other nonparametric multidimensional statistical models are also not transparent.

As for the PB deterministic numerical models that are often applied to describe multidimensional, complex, nonlinear systems, comparison with NNs is also not so obvious under closer consideration. In our opinion, modern PB deterministic numerical models have also lost a significant part of their transparency. Here again, when talking about transparency and physical clarity of PB models, one usually keeps in mind first very simplified models developed at the dawn of numerical modeling (see Sect. 1.2). Those models evolved to become more adequate to the complexity of systems, which they model. They evolved into modern PB numerical models (see Sect. 2.1.4 and Chap. 4 for examples). Modern models are very complex because they model very complex systems like ES or its subsystems. They are built of multiple blocks developed by different people and different institutions. The physics implemented in these blocks is parameterized. The parameterized physics is not transparent; it lost direct connections with elementary physical processes and contains many approximations, simplifications, assumptions, and empirical parameters (see Chap. 4). To make these parameterizations working together coherently in the model, significant amount of tuning parameters are introduced. They do not have any physical meaning and are introduced to force different parts of the model working coherently.

Based on the above, direct comparison of NNs with PB models is clearly a difficult problem. Thus, in our view, it is not productive to oppose NNs or other machine learning statistical approaches and first principle models; they should be considered as complementary. In Chap. 4 we show that they can be synergistically combined within a hybrid modeling framework, which, if properly implemented, should combine the advantages of both approaches.

2.5 NN Emulations

In this book, we use the terms *emulating NN*, *NN emulation*, or NN *emulator* for an NN that provides the functional emulation of the target mapping (2.1), including a small approximation error (2.11) for the training set (2.5) and a smooth and accurate interpolation and a limited extrapolation of the training set data inside the domain D. These terms are introduced to distinguish between emulating NNs and approximating NNs. Approximating NNs are usually concerned with an analytic approximation of a data set with a small approximation error (2.11).

When an emulating NN is constructed, in addition to the criterion of a small approximation error (2.11), at least three other criteria mentioned in Sect. 2.4.3 are used:

1. The NN complexity (2.4) (the number k of hidden neurons) is controlled and restricted to a minimum number that yields a level of accuracy sufficient for a good approximation.
2. Independent validation and test data sets are used in the process of training to control overfitting (validation set) and after training to evaluate the interpolation accuracy (test set).

3. A limited and controlled (see Sect. 2.3.3) redundancy is introduced in the training set (additional data points added "in-between" training data points) for improving the NN's interpolation performance.

The correspondence between the emulating NN complexity (2.4) and target mapping complexity is usually better than that of an approximating NN with the same approximation error. The complexity, N_C, of an emulating NN is usually close to the minimum value possible; thus, the emulating NN is usually faster. Finally, it usually provides a better and smoother interpolation or generalization, better resolution of the target mapping for the same approximation accuracy, and smaller uncertainties in the NN Jacobian.

2.6 Final Remarks

In this chapter, we discussed the general properties of multidimensional complex mappings (2.1) and MLP NN (2.2, and 2.3) and also demonstrated relationships between their properties. Both fields are relatively new and so are growing rapidly in terms of the relevant theory and practical applications. In this discussion, we emphasized that a transition from linear statistical tools or models to nonlinear models (like NNs) requires some adjustment in our methodological framework, which may not, at times, be flexible enough to accommodate sophisticated nonlinear approaches.

Many of the advantages of nonlinear statistical techniques may become limitations under certain conditions and in the case of unskillful use. Some of the limitations of nonlinear models may become advantageous when more flexible approaches are employed, by combining different statistical approaches (e.g., NNs and the ensemble approach), and by using additional information. As mentioned earlier, at this time, the NN is probably the only practical SLT tool for solving the problems discussed in this book and more generally for emulating complex multidimensional mappings. For completeness, in Sect. 6.2 we briefly review some alternative techniques that have potentials to compete with NNs in the future and present some preliminary results with alternative approaches (Belochitski et al. 2011).

References

Aires F, Schmitt M, Chedin A, Scott N (1999) The "Weight Smoothing" regularization of MLP for Jacobian stabilization. IEEE Trans Neural Netw 10:1502–1510

Aires F, Prigent C, Rossow WB (2004a) Neural network uncertainty assessment using Bayesian statistics: a remote sensing application. Neural Comput 16:2415–2458

Aires F, Prigent C, Rossow WB (2004b) Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 3 Network Jacobians. J Geophys Res. doi:10.1029/2003JD004175

Attali J-G, Pagès G (1997) Approximations of functions by a multilayer perceptron: a new approach. Neural Netw 6:1069–1081
Barron AR (1993) Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans Inform Theory 39:930–945
Belochitski AP, Binev P, DeVore R, Fox-Rabinovitz M, Krasnopolsky V, Lamby P (2011) Tree approximation of the long wave radiation parameterization in the NCAR CAM global climate model. J Comput Appl Math 236:447–460
Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford
Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
Bollivier M, Eifler W, Thiria S (2000) Sea surface temperature forecasts using on-line local learning algorithm in upwelling regions. Neurocomputing 30:59–63
Cardaliaguet P, Euvrard G (1992) Approximation of a function and its derivatives with a neural network. Neural Netw 5:207–220
Chen T, Chen H (1995a) Approximation capability to functions of several variables, nonlinear functionals and operators by radial basis function neural networks. Neural Netw 6:904–910
Chen T, Chen H (1995b) Universal approximation to nonlinear operators by neural networks with arbitrary activation function and its application to dynamical systems. Neural Netw 6:911–917
Chen AM, Lu H, Hecht-Nielsen R (1993) On the geometry of feedforward neural network error surface. Neural Comput 5:91–927
Cheng B, Titterington DM (1994) Neural networks: a review from a statistical perspective. Stat Sci 9:2–54
Cherkassky V, Mulier F (2007) Learning from data, 2nd edn. Wiley, Hoboken
Chevallier F, Mahfouf J-F (2001) Evaluation of the Jacobians of infrared radiation models for variational data assimilation. J Appl Meteorol 40:1445–1461
Chevallier F, Morcrette J-J, Chéruy F, Scott NA (2000) Use of a neural-network-based longwave radiative transfer scheme in the EMCWF atmospheric model. Q J Roy Meteor Soc 126:761–776
Cilliers P (2000) What can we learn from a theory of complexity? Emergence 2:23–33. doi:10.1207/S15327000EM0201_03
Cybenko G (1989) Approximation by superposition of sigmoidal functions. Math Control Signal 2:303–314
DeVore RA (1998) Nonlinear approximation. Acta Numerica 8:51–150
Elsner JB, Tsonis AA (1992) Nonlinear prediction, chaos, and noise. Bull Am Meteorol Soc 73:49–60
Funahashi K (1989) On the approximate realization of continuous mappings by neural networks. Neural Netw 2:183–192
Gell-Mann M, Lloyd S (1996) Information measures, effective complexity, and total information. Complexity 2:44–52
Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal 12:993–1001
Hashem S (1997) Optimal linear combination of neural networks. Neural Netw 10:599–614
Haykin S (2008) Neural networks and learning machines. Pearson, New York
Hornik K (1991) Approximation capabilities of multilayer feedforward network. Neural Netw 4:251–257
Hornik K, Stinchcombe M, White H (1990) Universal approximation of an unknown mapping and its derivatives using multilayer feedforward network. Neural Netw 3:551–560
Hsieh WW (2001) Nonlinear principal component analysis by neural networks. Tellus 53A:599–615
Hsieh WW (2004) Nonlinear multivariate and time series analysis by neural network methods. Rev Geophys. doi:10.1029/2002RG000112
Hsieh WW (2009) Machine learning methods in the environmental sciences. Cambridge University Press, Cambridge
Kon M, Plaskota L (2001) Complexity of neural network approximation with limited information: a worst case approach. J Complex 17:345–365
Krasnopolsky VM (2007) Reducing uncertainties in neural network Jacobians and improving accuracy of neural network emulations with NN ensemble approaches. Neural Netw 20:454–461

Krasnopolsky VM, Fox-Rabinovitz MS (2006) Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. Neural Netw 19:122–134

Krasnopolsky VM, Kukulin VI (1977) A stochastic variational method for the few-body systems. J Phys G Nucl Partic Nucl Phys 3:795–807

Krasnopolsky VM, Gemmill WH, Breaker LC (1999) A multiparameter empirical ocean algorithm for SSM/I retrievals. Can J Remote Sens 25:486–503

Krasnopolsky VM, Gemmill WH, Breaker LC (2000) A neural network multi-parameter algorithm SSM/I ocean retrievals: comparisons and validations. Remote Sens Environ 73:133–142

Krasnopolsky VM, Chalikov DV, Tolman HL (2002) A neural network technique to improve computational efficiency of numerical oceanic models. Ocean Model 4:363–383

Krasnopolsky VM, Lord SJ, Moorthi S, Spindler T (2009) How to deal with inhomogeneous outputs and high dimensionality of neural network emulations of model physics in numerical climate and weather prediction models. In: Proceedings of international joint conference on neural networks, Atlanta, Georgia, USA, 14–19 June, pp 1668–1673

Lee JW, Oh J-H (1997) Hybrid learning of mapping and its Jacobian in multilayer neural networks. Neural Comput 9:937–958

Liano K (1996) Robust error measure for supervised neural network learning with outliers. IEEE Trans Neural Netw 7:246–250

Luengo J, Garcia S, Herrera F (2010) A study on the use of imputation methods for experimentations with radial basis function network classifier handling missing attribute values: he good synergy between RBFNs and event covering method. Neural Netw 23:406–418

Maas O, Boulanger J-P, Thiria S (2000) Use of neural networks for predictions using time series: illustration with the *El Niño Southern oscillation* phenomenon. Neurocomputing 30:53–58

MacKay DJC (1992) A practical Bayesian framework for back-propagation networks. Neural Comput 4:448–472

Maclin R, Shavlik J (1995) Combining the predictions of multiple classifiers: using competitive learning to initialize neural networks. In: Proceedings of the eleventh international conference on artificial intelligence, Detroit, MI, pp 775–780

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in neural nets. Bull Math Biophys 5:115–137

Nabney IT (2002) Netlab: algorithms for pattern recognition. Springer, New York

Naftaly U, Intrator N, Horn D (1997) Optimal ensemble averaging of neural networks. Comput Neural Syst 8:283–294

Neal RM (1996) Bayesian learning for neural networks. Springer, New York

Nguyen D, Widrow B (1990) Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In: Proceedings of international joint conference of neural networks, vol 3, San Diego, CA, USA, 17–21 June, pp 21–26

Nilsson NJ (1965) Learning machines: foundations of trainable pattern-classifying systems. McGraw Hill, New York

Opitz D, Maclin R (1999) Popular ensemble methods: an empirical study. J Artif Intell Res 11:169–198

Reitsma F (2001) Spatial complexity. Master's thesis, Auckland University, New Zealand

Richman MB, Trafalis TB, Adrianto I (2009) Missing data imputation through machine learning algorithm. In: Haupt SE, Pasini A, Marzban C (eds) Artificial intelligence methods in environmental sciences. Springer, New York

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, Group PR (eds) Parallel distributed processing, vol 1. MIT Press, Cambridge, MA

Selfridge OG (1958) Pandemonium: a paradigm for learning. In: Mechanization of thought processes. In: Proceedings of a symposium held at the National Physical Lab, HMSO, London, pp 513–526

Sharkey AJC (1996) On combining artificial neural nets. Connect Sci 8:299–313

Tang Y, Hsieh WW (2003) ENSO simulation and prediction in a hybrid coupled model with data assimilation. J Meteorol Soc Jpn 81:1–19

Vann L, Hu Y (2002) A neural network inversion system for atmospheric remote-sensing measurements. In: Proceedings of the IEEE instrumentation and measurement technology conference, vol 2, pp 1613–1615. doi:10.1109/IMTC.2002.1007201

Vapnik VN (1995) The nature of statistical learning theory. Springer, New York

Vapnik VN, Kotz S (2006) Estimation of dependences based on empirical data (information science and statistics). Springer, New York

Weigend AS, Gershenfeld NA (1994) The future of time series: learning and understanding. In: Weigend AS, Gershenfeld NA (eds) Time series prediction. Forecasting the future and understanding the past. Addison-Wesley Publishing Company, Reading, pp 1–70

Werbos P (1974) Beyond regression: new tools for prediction and analysis in the behavioral sciences. Ph.D. dissertation, Committee on Applied Mathematics, Harvard University, Cambridge, MA Reprinted in Werbos P (1994) The roots of backpropagation. Wiley, Hoboken

Werbos P (1982) Applications of advances in nonlinear sensitivity analysis, systems modeling and optimization. In: Drenick R, Kozin F (eds) Proceedings of the 70th IFIP, 1981. Springer, New York. Reprinted in Werbos P (1994) The roots of backpropagation. Wiley, Hoboken

Wessels LFA, Bernard E (1992) Avoiding false local minima by proper initialization of connections. IEEE Trans Neural Netw 3:899–905

Zorita E, von Storch H (1999) A survey of statistical downscaling techniques. J Climate 2:2474–2489

Chapter 3
Atmospheric and Oceanic Remote Sensing Applications

I can't help thinking that science would be more appealing if it had no practical use.
– Claude Levi-Strauss, *The Raw and the Cooked*

Abstract This chapter is devoted to atmospheric and oceanic satellite remote sensing (RS) NN applications. Two major RS problems, forward problem and inverse problem (or satellite retrievals), are introduced and discussed. Applications of forward models (FM) for solving the forward problem in the process of direct assimilation of satellite measurements and for variational retrievals, as well as applications of retrieval algorithms, solutions of the inverse problem, for assimilation of geophysical parameters in data assimilation systems, are discussed. Correspondingly, two neural network (NN) applications, NN FMs and NN retrieval algorithms, are introduced. An intelligent NN retrieval system, which incorporates an automatic quality control of satellite retrievals, is introduced. Previously developed RS NN applications are reviewed. Theoretical considerations are illustrated with real-life applications of the NN approach to the Special Sensor Microwave Imager (SSM/I). SSM/I NN FM and SSM/I NN retrieval algorithms are introduced, discussed, and compared with FMs and retrieval algorithms developed using other techniques elsewhere. Advantages and limitations of NN FMs and retrieval algorithms are discussed. An example of QuikSCAT wind vector retrievals is used to demonstrate great potential of using the NN technique to go beyond the standard point-wise retrieval paradigm. This chapter contains an extensive list of references giving extended background and further detail to the interested reader on each examined topic. It can serve as a textbook and an introductory reading for students and beginning and advanced investigators interested in learning how to apply the NN emulation technique in different RS applications.

V.M. Krasnopolsky, *The Application of Neural Networks in the Earth System Sciences*, Atmospheric and Oceanographic Sciences Library 46, DOI 10.1007/978-94-007-6073-8_3,

Estimating high-quality geophysical parameters, carrying information about the physical, chemical, and biological properties of the oceans, atmosphere, and land surface, from remote measurements is an important problem in ESS, in fields such as meteorology, oceanography, climatology, climate modeling, and weather prediction. Direct measurements of many parameters, such as land vegetation moisture, phytoplankton concentrations in the ocean, and aerosol concentrations in the atmosphere, are generally not available for the entire globe or at the required spatial and temporal resolution. Even when in situ measurements are available, they are usually sparse (especially over the oceans) and located mainly at ground level or at the ocean surface. Fortunately, such measurements can often be estimated indirectly because the geophysical parameters of interest influence the electromagnetic radiation measured by an RS instrument. Because remote measurements carry signatures of these geophysical parameters, they allow us to obtain spatially dense measurements globally at and above the ground and ocean surface. Remote sensors are deployed on several different platforms including satellites and aircraft.

Remote measurements of electromagnetic radiation at different wavelengths are highly accurate. However, the quality of the geophysical parameters derived from these measurements varies significantly and depends on the strength and uniqueness of the signal that represents or signifies a specific geophysical parameter and on the mathematical methods applied to extract the parameter, which may require the solution of forward and/or inverse RS problems. The NN technique is a useful mathematical tool for solving the forward and inverse problems in RS. The number of NN RS applications has been increasing steadily over the last two decades.

A broad class of NN applications has been developed for solving the forward and inverse problems in RS in order to infer geophysical parameters from satellite data, i.e., to produce so-called satellite retrievals. Examples of such applications are numerous (see also Table 1.1). The NN technique has been applied for the inversion of a multiple scattering model to estimate snow parameters from passive microwave measurements (Tsang et al. 1992). Smith (1993) used NNs for the inversion of a simple two-stream radiative transfer model to derive a leaf area index from the Moderate Resolution Imaging Spectrometer data. In other studies, NNs have been applied to simulate scatterometer measurements and to retrieve wind speed and direction from these measurements (Thiria et al. 1993; Cornford et al. 2001); to retrieve oceanic and atmospheric constituents from satellite measurements of ocean color (Brajard et al. 2006); to retrieve the sea surface salinity from the observed Soil Moisture and Ocean Salinity brightness temperatures (Ammar et al. 2008); to develop an inversion algorithm for radar scattering from vegetation canopies (Pierce et al. 1994); and to estimate atmospheric humidity (Cabrera-Mercader and Staelin 1995), temperature, moisture, and ozone profiles (Aires et al. 2002; Mueller et al. 2003). Stogryn et al. (1994) and Krasnopolsky et al. (1995) applied NNs to invert Special Sensor Microwave Imager (SSM/I) data and retrieve wind speed over the ocean surface. Davis et al. (1995) applied NNs to invert a forward model (FM) to estimate soil moisture, surface air temperature, and vegetation moisture from Scanning Multichannel Microwave Radiometer data. Using an NN technique, a fast SSM/I FM (Krasnopolsky 1996, 1997) and SSM/I multiparameter retrieval

algorithms (Krasnopolsky et al. 1999, 2000; Meng et al. 2007; Roberts et al. 2010) have been developed. Young (2009) used NNs to emulate an FM and retrieval algorithm and to derive wind speeds from the Synthetic Aperture Radar. Abdelgadir et al. (1998) applied NNs to the forward and inverse modeling of canopy directional reflectance. Schiller and Doerffer (1999) used an NN technique for inverting a radiative transfer FM to estimate the concentration of phytoplankton pigment from Medium Resolution Imaging Spectrometer data.

This chapter is devoted to atmospheric and oceanic satellite RS NN applications. In Sect. 3.1, two major RS problems, forward problem and inverse problem (or satellite retrievals), are introduced. Applications of FMs for solving the forward problem in the process of direct assimilation of satellite measurements and for variational retrievals, as well as applications of retrieval algorithms, solutions of the inverse problem, for assimilation of geophysical parameters in data assimilation systems, are discussed. Correspondingly, in Sects. 3.2 and 3.3, two NN applications, NN FMs and NN retrieval algorithms, are introduced. In Sect. 3.4, an intelligent NN retrieval system, which incorporates an automatic quality control of satellite retrievals, is introduced. In Sect. 3.5, theoretical considerations introduced in previous sections are illustrated with real-life applications of the NN approach to the Special Sensor Microwave Imager (SSM/I). SSM/I NN FM and SSM/I NN retrieval algorithms are introduced, discussed, and compared with FMs and retrieval algorithm developed using other techniques elsewhere. In Sect. 3.6, an example of QuikSCAT wind vector retrievals is used to demonstrate great potential of using the NN technique to go beyond the standard point-wise retrieval paradigm. Advantages and limitations of NN RS applications are discussed in Sect. 3.7.

3.1 Deriving Geophysical Parameters from Satellite Measurements: Conventional Retrievals and Variational Retrievals

Satellite RS data are used in a wide variety of applications and by a wide variety of users. Satellite sensors generate measurements like radiances, backscatter coefficients, and brightness temperatures (BTs). The applications usually deal with geophysical parameters such as pressure, temperature, wind speed and direction, and water vapor concentration derived from satellite data. Satellite FMs, which simulate satellite measurements from given geophysical parameters, and retrieval algorithms, which transform satellite measurements into geophysical parameters, play the role of "mediators" between the satellite sensors and the applications. There exists an entire spectrum of different approaches to extract geophysical information from the satellite measurements. At one end of this spectrum, there are "satellite only" approaches; we refer to them as conventional or traditional retrievals. They employ measurements acquired from one particular sensor, sometimes from different channels (frequencies, polarizations, viewing angle, etc.) of the same sensor, to estimate the geophysical parameters.

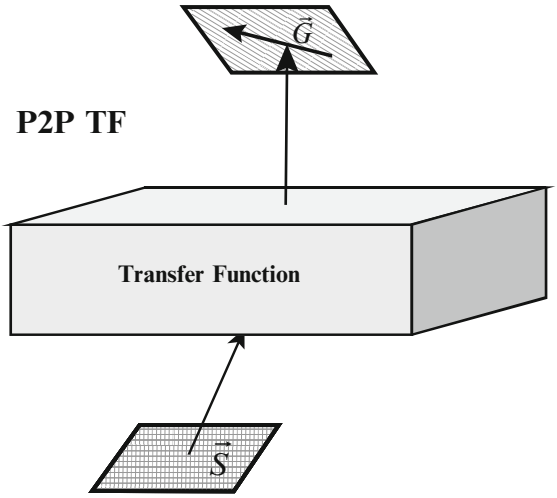

Fig. 3.1 The P2P retrieval paradigm. The P2P retrieval algorithm uses a vector of satellite measurements, S, collected from the footprint (represented by the parallelepiped *below*) and produces a vector of geophysical parameters, G, averaged over the same area (the parallelepiped *above*)

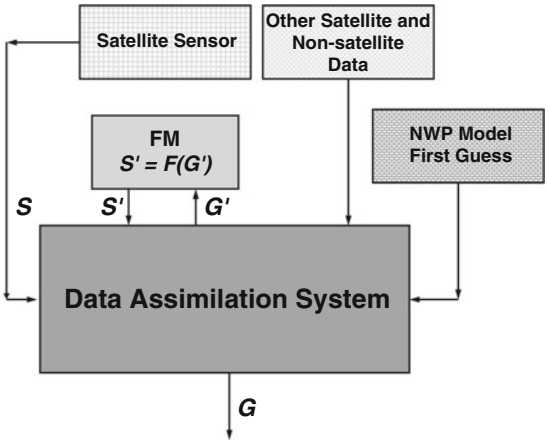

Fig. 3.2 Schematic representation of variational retrievals. An analysis state vector G' and a vector of simulated sensor measurements S' are related through the forward model F. G is the analyzed field – the final product of DAS

Variational retrieval techniques or direct assimilation techniques (see Sect. 3.1.2) lie at the other end of the spectrum. They use a DAS, together with an NWP model and analysis (Prigent et al. 1997), which in turn includes variety of meteorological measurements (from buoys, radiosondes, ships, aircrafts, etc.) as well as data from numerous satellite instruments. Figures 3.1 and 3.2 give schematic representations of these two approaches in performing satellite retrievals. Many approaches have been developed which belong in the intermediate portion of this spectrum. These approaches use measurements from several satellite sensors, combine satellite measurements with other kinds of measurements, and/or use background fields or profiles from NWP models for regularization of the inverse problem (retrievals) or for ambiguity removal (see Sect. 3.6). Thus, these approaches use some type of data fusion to regularize the solution of the inverse problem.

It is noteworthy that over the last few years, direct assimilation of some satellite measurements into modern DASs has been successfully developed and implemented. It has improved the quality of assimilated products and numerical forecasts that use some of these products in specifying initial conditions. Since direct assimilation uses the instrument measurements themselves (e.g., radiances), it replaces or eliminates the need for using retrievals of corresponding geophysical parameters in DASs. However, there are still many other geophysical parameters that have not yet been included, or it is not clear from theoretical and/or practical considerations how they could be included into DASs through direct assimilation. Since there are also other applications for the retrieved geophysical parameters, there remains an important need for conventional retrievals to represent these geophysical parameters and to develop the corresponding retrieval algorithms to which NNs could be efficiently applied.

3.1.1 Conventional P2P Retrievals

Conventional methods of using satellite data (conventional retrievals) involve solving an inverse or retrieval problem and deriving a transfer function (TF), f, which relates a geophysical parameter of interest G (e.g., surface wind speed over the ocean, atmospheric moisture concentration, SST) to a satellite measurement S (e.g., BTs, radiances, reflection coefficients),

$$G = f(S) \tag{3.1}$$

where both G and S are usually vectors and f is a mapping in this case. The TF, f (also called a retrieval algorithm), usually cannot be derived directly from first principles because the relationship (3.1) does not correspond directly to a cause and effect relationship and multiple values of G can sometimes correspond to a single S. For the case of FMs,

$$S = F(G) \tag{3.2}$$

where F is an FM, which relates a vector S to a vector G, FM can usually be derived from first principles and physical considerations (e.g., based on radiative transfer theory) in accordance with cause and effect principles because the geophysical parameters affect the satellite measurements (but not vice versa). Thus, the forward problem (3.2) is well posed in contrast to the inverse problem (3.1) which is often ill posed (Parker 1994), although, from a mathematical point of view, both FM (3.2) and TF (3.1) are continuous (or almost continuous) mappings between the two vectors S and G. Even in the cases where the mapping (3.1) is not unique, this multivalued mapping may be considered as a collection of single-valued continuous mappings.

FM and TF discussed above correspond to *the point-wise retrieval paradigm*. In this framework, TF (3.1) maps a vector of satellite measurements, S, for a particular location, to a vector of geophysical parameters, G (or vice versa in the case of FM (3.2)), at the same location. This location corresponds to the sensor footprint at the Earth's surface. The size of the footprint determines the sensor resolution. Roughly speaking, the sensor does not resolve any features smaller than the size of the footprint; thus, the footprint corresponds to one point or one pixel in the space of satellite measurements. The retrieved vector of geophysical parameters G corresponds to a point in the space of geophysical parameters, respectively, and it represents the average values of these parameters over the sensor footprint. Thus, the point-wise retrieval approach uses only locally averaged information for one particular footprint and retrieves only locally averaged information about the geophysical parameters at the same location. Figure 3.1 shows a schematic representation for such a retrieval algorithm. We will call this retrieval approach *point-to-point (P2P) retrievals* because it produces one vector G using measurements S from one footprint (or vice versa for FM).

In order to derive the TF (3.1), the FM (3.2) has to be inverted (an inverse problem has to be solved). The inversion technique usually applied searches for a vector G^0 which minimizes the following functional (Stoffelen and Anderson 1997),

$$\|\Delta S\| = \left\| S^0 - F(G) \right\| \tag{3.3}$$

where S^0 is a vector of satellite measurements. Since the FM, F, is usually a complicated nonlinear function, this approach leads to a full-scale nonlinear optimization problem, involving issues such as slow convergence and multiple solutions. This approach does not determine the TF explicitly; it assumes this function implicitly, and for each new measurement S^0, the entire process has to be repeated.

A simplified linearization method to minimize the functional (3.3) can be applied if a good approximation for the solution of the inverse problem is available, that is, an approximate vector of the geophysical parameters G^0. Then the difference vector ΔS is small, and there is a vector G in close proximity to G^0 ($|\Delta G| = |G - G^0|$ is small) where $\Delta S(G) = 0$. By expanding $F(G)$ in a Taylor series and keeping only those terms that are linear with respect to ΔG, we can obtain a system of linear equations to calculate the components of the vector ΔG (e.g., Wentz 1997),

$$\sum_{i=1}^{n} \frac{\partial F(G)}{\partial G_i}|_{G=G^0} \Delta G_i = S^0 - F\left(G^0\right) \tag{3.4}$$

where n is the dimension of vector G. The matrix of derivatives in (3.4) is the Jacobian, J, of FM, F, where

$$J = \left[\frac{\partial S_j}{\partial G_i} \right]_{i=1,\ldots,n}^{j=1,\ldots,m} = \left[\frac{\partial F_j(G)}{\partial G_i} \right]_{i=1,\ldots,n}^{j=1,\ldots,m} \tag{3.5}$$

After ΔG is calculated, the next iteration of (3.4) with $G^0 = G^0 + \Delta G$ is performed. The process generally converges quickly to the vector of retrievals G. Also in this case, the TF (3.1), f, is not determined explicitly but is only determined implicitly for the vector S^0 by the solution of (3.4). This type of retrieval can be called a "local" or "localized" linear inversion. These techniques (3.3, and 3.4) are usually referred to as PB *retrievals*. It is important to emphasize that the PB algorithms (3.3, and 3.4) are, by definition, *multiparameter algorithms* since they retrieve several geophysical parameters simultaneously (resulting in a complete vector G).

Empirical algorithms are based on an approach which, from the beginning, assumes the existence of an explicit analytical representation for a TF, f. A mathematical (statistical) model, f_{mod}, is chosen (usually a type of a regression), which relates a geophysical parameter g_k (e.g., wind speed) to a vector of satellite measurements S and contains a vector of empirical (or model, or free) parameters $a = \{a_1, a_2, \ldots\}$,

$$g_k = f_{\text{mod}}(S, a) \tag{3.6}$$

where g_k is a retrieved estimate of G_k (the same parameter retrieved by the multiparameter algorithm (3.3)), and the free parameters a are determined from an empirical (or simulated) matchup data set $\{G_k, S\}$ collocated in space and time. Statistical techniques such as the method of least squares can be used to calculate the parameters a. This type of retrieval is often called a "global" inversion since the obtained TF is not restricted to a given vector of satellite measurements at any given location. The subscript k in (3.6) emphasizes the fact that the majority of empirical retrieval algorithms are single-parameter algorithms. As we will show, g_k, in the case of single-parameter algorithms, is close but not equal to G_k. A set of single-parameter algorithms, for example, exists for SSM/I. They retrieve only a single parameter: the wind speed (Goodberlet et al. 1989), the water vapor (Alishouse et al. 1990; Petty 1993), or the cloud liquid water (Weng and Grody 1994).

Krasnopolsky et al. (1999, 2000) showed that single-parameter algorithms have additional (compared to multiparameter retrievals) systematic (bias) and random (unaccounted variance) errors in the single retrieved parameter g_k. Multiparameter algorithms *retrieve several geophysical parameters simultaneously* (the entire vector G), which, to a significant extent, determines the state of the atmosphere and/or the ocean surface in a given area at a given time. The single-parameter retrieval algorithm (3.6) falls short in this regard. The retrievals it produces, g_k, correspond to unknown "mean" atmospheric and surface states (other components of the vector G, G_i where $i \neq k$ are undetermined), which cannot be specified without additional information. For example, for the SSM/I instrument described in Sect. 3.5, single-parameter wind speed retrievals do not correspond to specific amounts of columnar water vapor, liquid water, or specific values of SST. Thus, single-parameter retrieval algorithms effectively average over an ensemble of atmospheric and surface states for all of the related geophysical parameters except for g_k which is retrieved. This averaging process gives rise to additional errors in the single retrieved parameter. Obviously these errors do not arise in the multiparameter approach.

If g_k is a geophysical parameter retrieved by the single-parameter algorithm (3.6) and G_k is the same geophysical parameter retrieved by the corresponding multiparameter algorithm (e.g., (3.4)), then the additional systematic error (bias) can be estimated as

$$\overline{(G_k - g_k)} = \sum_{i \neq k} \alpha_i \cdot b_i + \sum_{i \neq k} \beta_i \cdot \sigma_i^2 + \sum_{i,j} \gamma_{ij} \cdot c_{ij} + \cdots . \tag{3.7a}$$

The horizontal bar above the symbols on the left-hand side implies averaging over all G_i with $i \neq k$, which are not known for single-parameter algorithms; b_i and ${\sigma_i}^2$ are the biases and variances of these geophysical parameters; c_{ij} are the correlation coefficients between them; and α_i, β_i, and γ_{ij} are coefficients that are given in Krasnopolsky et al. (1999, 2000). Similar estimates can also be obtained for the additional variances or random error components:

$$\overline{(G_k - g_k)^2} = \sum_{i,j} \delta_{ij} \cdot \sigma_i \cdot \sigma_j + \cdots . \tag{3.7b}$$

It is clear from (3.7) that the multiparameter wind speed retrievals, G_k, compared with single-parameter retrievals, g_k, do not contain additional systematic (bias) or random errors due to unaccounted variability for all of the G_i with $i \neq k$. The absence of these additional errors provides a significant advantage for the multiparameter approach.

Thus, the obvious way to improve single-parameter retrievals (3.6) is to include other parameters in the retrieval process using an empirical multiparameter approach, which, as in the PB multiparameter approach (3.3, and 3.4), inverts the data in the complete space of the geophysical parameters (Krasnopolsky et al. 1999, 2000). Thereby, the complete vector of the related geophysical parameters is retrieved simultaneously from a given vector of satellite measurements S,

$$G = f_{\text{mod}}(S) \tag{3.8}$$

where $G = \{G_i\}_{i=1,\ldots,n}$ is now a vector containing the primary, physically related geophysical parameters, which contribute to the observed satellite measurements S. Because Eqs. (3.1), (3.2), (3.6), and (3.8) represent continuous mappings, the NN technique is well suited for emulating the FM, TF and empirical TF, f_{mod}. In fact, the NN technique is very well suited (much better than are classical statistical techniques) for developing empirical multiparameter retrieval algorithms (3.8).

The retrievals (3.6) and (3.8) are global in scope because the same explicit TF, if properly constructed, can be applied over the entire globe. The term "global" should not be confused here with the field-wise retrievals discussed in Sect. 3.6. TF (3.1), (3.6), and (3.8) still follow the aforementioned P2P point-wise retrieval paradigm (Fig. 3.1). The conventional retrievals derived using TF (3.1) have the same spatial resolution as the sensor measurements and produce instantaneous

values of geophysical parameters over the areas where the measurements are available. Geophysical parameters derived using conventional retrievals can be used for many applications, in particular, in the NWP DASs (see the next section).

3.1.2 *Variational Retrievals Through the Direct Assimilation of Satellite Measurements*

Because conventional retrievals are based on the solution of an inverse problem which is usually mathematically ill posed (Parker 1994), this approach has some rather subtle properties and error characteristics (Eyre and Lorenc 1989), which cause additional errors and problems in obtained retrievals (e.g., an amplification of the errors and ambiguities). As a result, high-quality sensor measurements can be converted into lower-quality geophysical parameters. This type of error can be avoided or reduced by using a variational retrieval technique or inversion through direct assimilation of the satellite measurements per se (Lorenc 1986; Parrish and Derber 1992; Phalippou 1996; Prigent et al. 1997; Derber and Wu 1998; McNally et al. 2000). The variational retrieval technique performs satellite retrievals inside DAS.

DAS prepares initial conditions for the NWP and climate systems. It fuses different types of observations in a product called analysis (Daley 1991). Data assimilation proceeds by analysis cycles. In each analysis cycle, observations of the current (and possibly past) state of a system are combined with the results from a numerical weather prediction model (the forecast or the first guess) to produce an analysis, which is considered as "the best" estimate of the current state of the system. This is called the analysis step. Essentially, the analysis step tries to balance the uncertainty in the data and in the forecast. The model is then advanced in time and its result becomes the first guess in the next analysis cycle.

The process of creating the analysis in a variational DAS involves minimization of a "cost function." A typical cost function, χ, would be the sum of the squared deviations of the analysis values from the observations weighted by the accuracy of the observations, plus the sum of the squared deviations of the first-guess fields and the analyzed fields weighted by the accuracy of the first guess as

$$\chi = (\xi - \xi_b)^T B^{-1} (\xi - \xi_b) + (\xi - \xi_O)^T Q^{-1} (\xi - \xi_O) . \tag{3.9}$$

A background term containing an NWP model first guess must be added as in (3.9) to regularize an otherwise ill-posed data assimilation problem (Parrish and Derber 1992); ξ is a field of the geophysical parameter being analyzed, ξ_b is the first-guess background field, and ξ_O represents observations. The accuracy of the observations is represented in (3.9) by the observational error covariance matrix, Q, and the accuracy of the first guess is represented by the background error covariance matrix, B (Daley 1991).

Simply speaking, minimizing (3.9) makes sure that the analysis does not drift too far away from observations and forecasts that are known to usually be reliable.

In the case of conventional P2P retrievals considered in the previous section, the contribution χ_G to the second term of the variational analysis cost function χ (3.9) from a particular retrieval, G_0, can be expressed as

$$\chi_G = \frac{1}{2}(G - G_O)^T (O + E)^{-1} (G - G_O) \tag{3.10a}$$

where $G_O = f(S_O)$ is a vector of the retrieved geophysical parameter, S_O is a vector of the sensor measurements, G is the vector of the geophysical parameters being analyzed, and the observational error covariance matrix Q is the sum of the expected error covariance of the observations, O, and the expected error covariance of the retrieval algorithm E.

Variational retrievals or direct assimilation of satellite data offer an alternative to the conventional P2P approach in deriving geophysical parameters from the satellite measurements and are shown in Fig. 3.2. They use the entire DAS as a retrieval algorithm for the inversion of FM.

In this case, a contribution χ_S to the second term of the analysis cost function (3.9) from a particular sensor measurement, S_O, is

$$\chi_S = \left(S' - S_O\right)^T (O + E)^{-1} \left(S' - S_O\right) \tag{3.10b}$$

where $S' = F(G')$ and F is the FM (3.2), which relates an analysis state vector G' (a vector of geophysical parameters in the analysis) to a vector of simulated sensor measurements S', O is the expected error covariance of the observations, and E is the expected error covariance of the FM. The forward problem (3.2) is well posed in contrast with the inverse problem (3.1). However, the entire data assimilation problem is ill posed and a background term containing an NWP model first guess has to be added to both (3.10a) and (3.10b) to regularize the problem (Parrish and Derber 1992).

The variational retrievals are inherently field-wise; that is, they produce an entire field (global, in the case of global DAS) for the geophysical parameter G. This field-wise retrieval approach takes a field of satellite measurements and produces a field of geophysical parameters; thus, it belongs to the field-wise field-to-field (F2F) retrieval paradigm (see Sect. 3.6). The field G has the same resolution as the numerical forecast model used in the DAS. This resolution may be lower or higher than the resolution of conventional retrievals.

The variational retrievals are also not instantaneous but usually averaged in time over the analysis cycle; however, the field is continuous and coherent (e.g., it should not have problems such as directional ambiguity that occurs in the scatterometer-derived winds described in Sect. 3.6). Variational retrievals are the result of fusing many different types of data (including satellite data, ground observations, and numerical model first-guess fields) within the DAS. Sparse conventional retrievals can be also converted into continuous fields using the data assimilation procedure (3.9).

It is important to emphasize a very significant difference between the use of the explicit TF for conventional retrievals and the use of FM for variational retrievals. In conventional retrievals, the explicit TF (3.1) is usually simple (e.g., a regression or an NN) and is applied once per sensor observation to produce a geophysical parameter. In variational retrievals, the FM, which is usually much more complicated than a simple explicit TF, and its Jacobian composed of partial derivatives (the number of derivatives is equal to $m \times n$, where m and n are the dimensions of the vectors G and S, respectively) have to be estimated for each of the k iterations performed in DAS during the process of minimizing the cost function (3.10b). Because DAS is very time consuming, the calculation time is a very important issue here. Thus, the requirements for simplicity of the FM used in the variational retrievals are restrictive, and variational retrievals often require some special, simplified, and fast versions of FMs.

3.2 NNs for Emulating Forward Models

FMs are usually complex due to the complexity of the physical processes which they describe and the complexity of the first principle formalism on which they are based (e.g., radiative transfer theory). The dependence of satellite measurements on the geophysical parameters, which FMs describe, is complicated and nonlinear. These dependencies may exhibit different types of nonlinear behavior. As we discussed earlier, FMs are usually employed in PB retrieval algorithms, where they are numerically inverted to retrieve geophysical parameters, and in DASs where they are used for the direct assimilation of satellite measurements (variational retrievals). Both numerical inversions and direct assimilation are iterative processes where FMs and their Jacobians (3.5) are calculated many times for each satellite measurement. Thus, the retrieval process becomes very time consuming and can be prohibitively expensive for operational (real-time) applications.

For such applications, it is essential to have fast and accurate versions of FMs. Because the functional complexity of FM mappings (see Sect. 2.2.2) is usually not as great as their physical and mathematical complexities, NNs can provide fast and accurate emulations of FMs. Moreover, an NN can also provide an entire Jacobian matrix with only a small additional computational effort. This is one of NN applications where not only the NN emulation itself but also the NN Jacobian should be carefully tested and controlled (see Sect. 2.4.4). An example of using an NN FM (aka an observation operator) in DAS is introduced in Sect. 5.1. In this section, an application of the NN ensemble technique to improve the accuracy of the NN Jacobian is also demonstrated.

To develop an NN emulation for an FM, a training set which consists of matched pairs of vectors of geophysical parameters and satellite measurements, $\{G,\ S\}_{i=1,\ldots,N}$, has to be created. If a PB FM exists, it can be used to simulate the training set. Otherwise, empirical data can be used to create a training set.

3.3 NNs for Solving Inverse Problems: NNs Emulating Retrieval Algorithms

NNs can be used in at least two different ways to serve as retrieval algorithms. First, as shown in the previous section, in PB retrieval algorithms, a fast NN, emulating the complex and slow PB FM and its Jacobian, can be used to speed up the local inversion process (3.4). Second, in many cases NNs can be used for a global inversion to explicitly invert an FM. In such cases, after an inversion the NN provides an explicit retrieval algorithm (or TF), which is a solution for the inverse problem and can be used for retrievals. To train an NN which emulates an explicit retrieval algorithm, a training set, $\{G, S\}_{i=1,\ldots,N}$, is required. As in the case of FMs, simulated or empirical data can be used to create the training set.

In addition to the complications related to FMs (complexity, nonlinearity, etc.), retrieval algorithms exhibit certain problems because they are solutions to the inverse problem, which is usually ill posed. This is why mathematical tools, which are used to develop retrieval algorithms, have to be accurate and robust in order to deal with these additional problems. NNs are fast, accurate, flexible, and generic tools for emulating nonlinear (continuous) mappings and can be effectively used for modeling multiparameter retrieval algorithms. An additional problem related to retrieval algorithms is regularizing the solution of the inverse problem. To regularize an ill-posed inverse problem, additional (regularization) information should be introduced (Vapnik and Kotz 2006). The NN technique is flexible enough to accommodate regularization information as additional inputs and/or outputs and as additional regularization terms in the error or loss function (2.11). For example, in their pioneering work in using NNs for the simultaneous retrieval of temperature, water vapor, and ozone atmospheric profiles from satellite measurements (Aires et al. 2002; Mueller et al. 2003), the authors made good use of this NN flexibility by introducing the first-guess solution for these profiles from an atmospheric model or DAS as additional regularizing inputs in their NN-based retrieval algorithms. Roberts et al. (2010) used a first guess of SST as an additional input to improve the accuracy of their NN multiparameter retrieval algorithm.

3.4 Controlling the NN Generalization and Quality Control of Retrievals

NNs are well suited to modeling complicated nonlinear relationships between multiple variables, as is the case in multispectral RS. Well-constructed NNs (NN emulations) have very good interpolation properties; however, they may produce unrealistic outputs when forced to extrapolate (see Sect. 2.4.3). The NN training data (simulated using an FM or constructed from empirical data) span a certain manifold D_T (a subdomain of D, $D_T \in D$) in the full domain D. During the retrieval process,

satellite measurements (i.e., actual data) that serve as inputs to the NN, f_{NN}, which emulates a TF (3.1), may not always lie in D_T. There are many sources that can cause such deviations of the data from a low-dimensional manifold D_T of training data, e.g., simplifications built into an FM design, neglecting the natural variability of parameters occurring in the model, and measurement errors in the satellite data not taken into account during the generation of the training data. For instance, when empirical data are used, extreme events (highest and lowest values for geophysical parameters) are usually not sufficiently represented in the training set because they occur so rarely. Thus, during the retrieval stage, real data, in some cases, may force the NN emulation f_{NN} to extrapolate. The error resulting from forced extrapolation will increase with the distance of the input point from D_T and will also depend on the orientation of the input point relative to D_T.

In order to recognize NN input patterns not foreseen in the NN training phase and, thus, beyond the scope of the inversion algorithm, a validity check (Krasnopolsky and Schiller 2003) can be used. This check works in the following manner. Let us assume that the FM $S = F(G)$ has an inverse $G = f(S)$; then, by definition $S = F(f(S))$. Further, let us assume that f_{NN} is the NN emulating the inverse model in the domain D_T. As was mentioned above, the result of $G_0 = f_{\mathrm{NN}}(S_0)$ for $S_0 \notin D_T$ may be arbitrary, and, in general, $F(f_{\mathrm{NN}}(S_0))$ will not be equal to S_0. The validity condition

$$S = F\left(f_{\mathrm{NN}}(S)\right) \tag{3.11}$$

is a necessary condition for $S \in D_T$. Now, if in the application of the NN f_{NN}, S is not in the domain D_T, the NN f_{NN} is forced to extrapolate. In such a situation, the validity condition may not be fulfilled, and the resulting G is, in general, meaningless.

There is another common situation when the validity condition (3.11) for retrievals may become instrumental. In some cases, the satellite measurements do not carry signatures of any geophysical parameters, or these signals are very weak and noisy. For example, in satellite retrievals of various surface parameters, such a situation may occur when the radiation from the surface is completely shielded by heavy cloudiness. In such cases, the retrieval $G_0 = f_{\mathrm{NN}}(S_0)$ is meaningless and the validity check will not be satisfied. Usually retrieval flags are introduced to solve the problem (e.g., Stogryn et al. 1994). Retrieval flags identify (empirically or theoretically) manifolds inside the domain D where the aforementioned situation occurs. However, in the case of multidimensional domains with complicated geometry, it may be difficult, or impossible, to introduce retrieval flags in advance. However, by using the validity condition, we may be able to identify questionable events during the retrieval process.

For operational applications, it is necessary to report such events to a higher evaluation level. The validity check may serve as the basis for developing quality control (QC) procedures. QC procedures are usually applied to the satellite retrievals online and in the DAS. In order to perform the validity test, the FM must be applied after each inversion. This requires speed and accuracy in FM. Such an FM can be

developed by training an NN that accurately emulates the original FM, $S = F_{\mathrm{NN}}(G)$. Thus, a fast validity check algorithm consists of a combination of inverse and forward NNs. In addition to the inversion, it computes a quality measure for the inversion:

$$\delta = \|S - F_{\mathrm{NN}}\left(f_{\mathrm{NN}}(S)\right)\| \tag{3.12}$$

Finally, the solution to the problem of a scope check is obtained by estimating the quality measure δ (3.12) where S results from the satellite measurements. This procedure (1) allows for the detection of situations where the FM and/or TF is inappropriate, (2) does an "in scope" check for the retrieved parameters even if the domain has a complicated geometry, (3) can be used for QC of the retrievals, for flagging questionable retrievals, and (4) can be used for expanding the training set for values of S (that are within the domain D) that are underrepresented in the training set. An application of this technique to the SSM/I wind speed retrievals is described in Sect. 3.5.3.

There also exists a more straightforward approach to the QC of satellite retrievals. An additional "error" NN can be trained to predict the errors of retrievals produced by the NN retrieval algorithm. The error NN can be trained using the same training set that was used for training the NN retrieval algorithm. The error NN has the same inputs as the NN retrieval algorithm and one output, which is an estimate of the retrieval error. The error NN works in parallel with the NN retrieval algorithm. The latter produces the retrieval and the former the error estimate for the retrieval. If the error estimate is larger than a predetermined threshold, the retrieval is flagged as unreliable. The error estimate is usually different from, but highly correlated with, the retrieval error itself. Thus, it cannot be used for the error correction but only for QC purposes. A QC procedure based on the error NN is described in Sect. 4.6.5 in the context of the compound parameterization.

3.5 Neural Network Emulations for SSM/I Data

In previous chapters, we discussed the theoretical possibilities and premises for using NNs for modeling TFs and FMs. In this chapter, we illustrate these theoretical considerations with real-life applications of the NN approach to the SSM/I forward and retrieval problems. SSM/I is a well-established instrument, flown since 1987. Several SSM/I instruments (F8, F10, F11, F13, etc.) have been functioning for significant period of time. The satellites that carry the SSM/I are polar orbiting with an orbital period of 102 min. Each satellite provides coverage over a particular ocean basin twice a day, once during the descending orbit and once during the ascending orbit. The SSM/I generates BTs in seven channels at four frequencies (19, 22, 37, and 85 GHz), each with vertical and horizontal polarization (the 22 GHz channel

senses only vertical polarization). The spatial resolution is about 50 km at 19 and 22 GHz, about 30 km at 37 GHz, and 15 km at 85 GHz. It has a swath width of approximately 1,400 km.

The SSM/I infers BTs passively, receiving microwave radiation emitted by the ocean surface and transmitted through the atmosphere. The emission is affected by the surface wind speed (which changes the roughness of the ocean surface), by the air temperature and humidity above the ocean surface, and by the SST. The propagation of microwave radiation through the atmosphere is influenced by the cumulative amounts of water vapor and liquid water in the atmospheric column (Wentz 1997). As a result, the BTs contain signatures from all of these geophysical parameters, and these parameters (surface wind speed, air temperature, air humidity, columnar water vapor, columnar liquid water, and SST) can be extracted from the BTs using retrieval algorithms.

Many different retrieval algorithms and several FMs have been developed for the SSM/I, and several different databases are available for algorithm development and validation. Various techniques have been applied in algorithm development. Thus, detailed comparison of different methods and approaches for this instrument can be made.

3.5.1 NN Emulations for the Empirical FM for the SSM/I

The empirical FM developed for the SSM/I represents the relationship between the vector of geophysical parameters G and the vector of satellite brightness BTs, S, where $S = \{T19V, T19H, T22V, T37V, T37H\}$. Here in the BT notation $TXXY$, XX represents the frequency in GHz and Y the polarization. Four geophysical parameters are included in G (surface wind speed W, columnar water vapor V, columnar liquid water L, and SST), or more compactly, $G = \{W, V, L, T_s$ (or $SST)\}$. These are the major parameters that influence BTs measured by satellite sensors, which were used as inputs in the PB FMs of Petty and Katsaros (1992, 1994) (referenced below as PK) and Wentz (1997) (see Table 3.1). The NN emulation (FM1 in Table 3.1) (Krasnopolsky 1996), which implements this SSM/I FM has 4 inputs $\{W, V, L, SST\}$, one hidden layer with 12 neurons, and 5 nonlinear BT outputs $\{T19V, T19H, T22V, T37V, T37H\}$. The derivatives of the outputs over the inputs, which can be easily calculated, constitute the Jacobian matrix (3.5). The Jacobian is required in the process of direct assimilation of the SSM/I BTs when the gradient of the SSM/I contribution to the cost function (3.10b) χ_s is calculated (Parrish and Derber 1992; Phalippou 1996). Estimating an NN emulation of the FM and its derivatives is a much simpler and faster task than calculating radiative transfer FMs.

A raw buoy-SSM/I matchup database created by the Navy Fleet Numerical Meteorology and Oceanography Center was used for the NN algorithm development,

Table 3.1 Comparison of PB radiative transfer and empirical NN FMs under clear and clear + cloudy (*in parentheses*) weather conditions

			BT RMS error (°K)	
Author	Type	Inputs	Vertical	Horizontal
Petty and Katsaros (1992)	PB	*W, V, L, SST, Theta, P0, HWV, ZCLD, Ta, G*	1.9 (2.3)	3.3 (4.3)
Wentz (1997)	PB	*W, V, L, SST, Theta*	2.3 (2.8)	3.4 (5.1)
Krasnopolsky (1996)	NN, FM1	*W, V, L, SST*	1.5 (1.7)	3.0 (3.4)

Theta is the incidence angle, *P0* – surface pressure, *HWV* – vapor scale height, *ZCLD* – cloud height, *Ta* – effective surface temperature, and *G* – the lapse rate. PB is for physically based

validation, and comparison. This database is quite representative, with the exception of high latitude and high wind speed events. In order to improve the representativeness of the database, the data sets were enriched by adding to the database matchup databases collected by the high latitude European ocean weather ships Mike and Lima. Various filters have been applied to remove errors and noisy data (for a detailed discussion, see Krasnopolsky (1996) and Krasnopolsky et al. (1996, 1999)).

The matchup databases for the F11 SSM/I have been used for training (∼3,500 matchups) and validation (∼3,500 matchups) for the FM. The NN emulation of FM (FM1) was trained using all matchups that corresponded to clear and cloudy weather conditions in accordance with the retrieval flags introduced by Stogryn et al. (1994). Only those cases where the microwave radiation could not penetrate the clouds were removed. Then, ∼6,000 matchups for the F10 instrument were used for testing and comparison of FM1 with the PB FM by Petty and Katsaros (1994) and Wentz (1997) FM. The root mean square errors (RMSEs) for FM1 were systematically better than those for the PK and Wentz FMs for all weather conditions and all of the channels considered. With the NN FM, the horizontally polarized channels 19H and 37H had the highest RMSE, ∼2.5°K under clear and ∼3°K under clear and cloudy conditions. For the vertically polarized channels, the RMSEs were lower, 1.5°K under clear and 1.7°K under partly clear and partly cloudy conditions. The same trend was observed for the PK and Wentz FMs. Table 3.1 presents bulk statistics (RMSEs) for the three FMs discussed here. In the table, the RMSEs are slightly different from those presented above because they are averaged over different frequencies separately for the vertical and horizontal polarizations.

Thus, the NN FM gives results that are comparable or better in terms of RMSEs than the results obtained with more sophisticated physically based models (shown in Table 3.1), and is much simpler than the PB FMs. The NN FM is not as general as a radiative transfer model; it was developed to be applied in a DAS for variational retrievals and direct assimilation of SSM/I BTs at particular frequencies and from a particular instrument. However, for this particular application (direct assimilation) and instrument, it is significantly simpler and faster, which is especially important in real-time (operational) applications. This is also one of the applications where the accuracy of the NN Jacobian is essential. The NN FM simultaneously calculates the BTs and Jacobian matrix. Krasnopolsky (1996) showed that for this particular

Table 3.2 Error budget (in m/s) for different SSM/I wind speed algorithms under clear and clear + cloudy (*in parentheses*) conditions

Algorithm	Method	Bias	Total RMSE	$W > 15$ m/s RMSE
GSW	Multiple linear regression	−0.2 (−0.5)	1.8 (2.1)	(2.7)
GSWP	Generalized linear regression	−0.1 (−0.3)	1.7 (1.9)	(2.6)
GS	Nonlinear regression	0.5 (0.7)	1.8 (2.5)	(2.7)
Wentz	Physically based	0.1 (−0.1)	1.7 (2.1)	(2.6)
NN0	Neural network	0.0 (0.0)	1.4 (1.6)	(3.5)
NN1	Neural network	−0.1 (−0.2)	1.5 (1.7)	(2.3)
NN2	Neural network	(−0.3)	(1.5)	–
NN3	Neural network	(−0.2)	(1.6)	–

GSW (Goodberlet et al. 1989), GSWP (Petty 1993), GS (Goodberlet and Swift 1992), Wentz (1997), NN0 (Krasnopolsky et al. 1995), NN1 (Krasnopolsky et al. 1996, 1999), NN2 (Meng et al. 2007), NN3 (Roberts et al. 2010)

application, the NN Jacobian is sufficiently smooth. In Sect. 5.1, a generic NN ensemble technique (Krasnopolsky 2007) is discussed that improves the stability and reduces uncertainties of the NN Jacobian.

3.5.2 NN Empirical SSM/I Retrieval Algorithms

The SSM/I wind speed retrieval problem is a perfect example that illustrates the general discussion presented in Sects. 3.1 and 3.3. The problems encountered in the case of the SSM/I wind speed retrievals are representative, and the methods used to solve them can easily be generalized for other geophysical parameters and sensors. About ten different SSM/I wind speed retrieval algorithms, both empirical and physically based, have been developed using a large variety of approaches and methods. Here these algorithms are compared in order to illustrate some of the properties of the different approaches mentioned earlier and some of the advantages and limitations of the NN approach.

Goodberlet et al. (1989) developed the first global empirical SSM/I wind speed retrieval algorithm. Statistics for this algorithm are shown in Table 3.2 under the abbreviation GSW. This algorithm is a single-parameter algorithm (it retrieves only wind speed) and is linear with respect to BTs, i.e., a linear multiple regression is used for f_{mod} in (3.6) to approximate the SSM/I TF, and is given by

$$W_{\text{GSW}} = 147.9 + 1.0969T19V - 0.4555T22V - 1.76T37V + 0.786T37H$$

This algorithm presents a linear approximation to a nonlinear (especially under cloudy sky conditions) SSM/I TF f (3.6). Under clear sky conditions (Table 3.2), it retrieves the wind speed with acceptable accuracy. However, under cloudy conditions where the amount of the water vapor and/or cloud liquid water in the atmosphere increases, errors in the retrieved wind speed increase significantly.

Goodberlet and Swift (1992) tried to improve the GSW algorithm performance under cloudy conditions, using nonlinear regression with a rational type of nonlinearity:

$$W_{\mathrm{GS}} = \frac{(W_{\mathrm{GSW}} - 18.56 \cdot \alpha)}{(1-\alpha)}, \text{ where } \alpha = \left(\frac{30.7}{\Delta_{37}}\right)^4 \text{ and}$$
$$\Delta_{37} = T37V - T37H.$$

Since the nature of the nonlinearity of the SSM/I TF under cloudy conditions is not known precisely, the application of such a nonlinear regression with an assumed form of nonlinearity was not sufficient to improve the algorithm performance (we refer to this algorithm as GS). In many cases, the GS algorithm generates false high wind speeds when the observed wind speeds are less than 15 m/s (Krasnopolsky et al. 1996).

A nonlinear (with respect to BTs) algorithm introduced by Petty (1993) and called the GSWP algorithm here is based on a generalized linear regression. It presents a case where nonlinearities introduced in the algorithm represent the nonlinear behavior of the TF more accurately. This algorithm introduces a nonlinear correction for the linear GSW algorithm when the amount of water vapor, V, in the atmosphere is greater than zero, as

$$W_{\mathrm{GSWP}} = W_{\mathrm{GSP}} - 2.13 + 0.2198 \cdot V - 0.4008 \cdot 10^{-2} \cdot V^2, \text{ where}$$
$$V = 174.1 + 4.638 \cdot \ln(300 - T19V) - 61.76 \cdot \ln(300 - T22V)$$
$$+ 19.58 \cdot \ln(300 - T37V)$$

Table 3.2 shows that the GSWP algorithm improves the accuracy of retrievals compared to the linear GSW algorithm under both clear and cloudy conditions. However, it does not improve the GSW algorithm performance at high wind speeds because most of the high wind speed events occur at mid- and high latitudes where the amount of water vapor in the atmosphere is not significant. Here, the cloud liquid water is the main source of the nonlinear behavior in the TF. However, it has not been taken into account in the GSWP algorithm.

NN algorithms have been introduced as an alternative to nonlinear and generalized linear regressions because NNs can adjust their nonlinear behavior to that of TF better than regression-based approaches. Stogryn et al. (1994) developed the first NN SSM/I wind speed retrieval algorithm, which consisted of two NNs, each with surface wind speed as the single output. Thus, this algorithm is a single-parameter algorithm (3.6). One NN performs retrievals under clear conditions and the other under cloudy conditions. Krasnopolsky et al. (1995) showed that a single NN (NN0 in Table 3.2) with the same architecture (a single output) can generate retrievals for surface winds under both clear and cloudy conditions with the same accuracy as the two NNs developed by Stogryn et al. (1994). This application of a single NN emulation led to a significant improvement in wind speed retrieval accuracy under

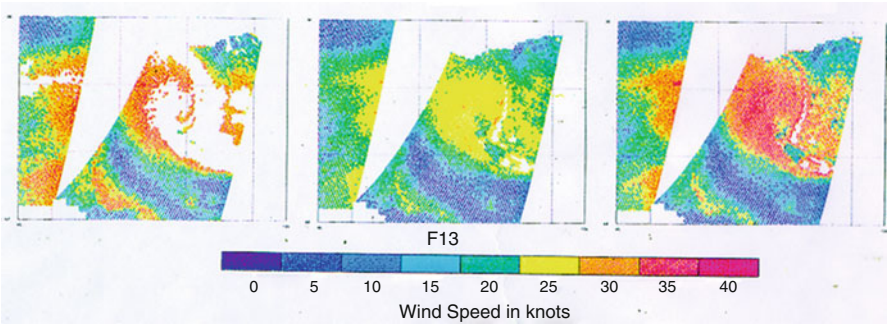

Fig. 3.3 Wind speed fields retrieved from the SSM/I measurements (satellite F13) for a mid-latitude storm located to the northeast of Australia. Two passes (one ascending and one descending) are shown in each panel. Each panel shows the wind speeds retrieved by the (*left to right*) GSW, NN0, and NN1 algorithms. The GSW algorithm does not produce reliable retrievals in the areas with high level of moisture (*white areas*). NN0 fills these gaps; however, it underestimates high wind speeds. NN1 produces accurate high winds under high levels of moisture. 1 knot ≈ 0.514 m/s

clear conditions compared to the GSW algorithm. Under higher moisture/cloudy conditions, the improvement was even greater (25–30 %) compared to the GSW algorithm. The increase in areal coverage due to the improvements in accuracy was about 15 % on average and higher in areas where there were significant weather events (and higher levels of atmospheric moisture) as can be seen in Fig. 3.3.

Both NN algorithms give very similar results because they had been developed using the same matchup database. This database, however, does not contain any matchups for wind speeds higher than about 20 m/s and contains very few matchups for wind speeds higher than 15 m/s. These algorithms are also single-parameter algorithms, i.e., they retrieve only one parameter – wind speed; therefore, they cannot account for the variability in all related atmospheric (e.g., water vapor and liquid water) and surface (e.g., SST) parameters, which takes on greater importance at higher wind speeds. This is why these NN algorithms pose the same problem; they cannot generate wind speeds at ranges higher than 18–19 m/s with acceptable accuracy (see Table 3.2 and Fig. 3.3).

The next-generation NN algorithm – a multiparameter NN algorithm (3.8) developed (NN1 in Table 3.2) by Krasnopolsky et al. (1996, 1999) – solved the high wind speed problem (see Table 3.2 and Fig. 3.3) through three main advances. First, a new buoy/SSM/I matchup database was used in the development of this algorithm. It contained an extensive matchup data set for the F8, F10, and F11 sensors, provided by Navy Research Laboratory, and augmented with additional data from the European Ocean Weather Ships Mike and Lima for high latitude, high wind speed events (up to 26 m/s). Second, the NN training method was improved by enhancing the learning process for the high wind speed range. Third, the variability of related atmospheric and surface parameters was taken into account; surface wind speed (W), columnar water vapor (V), columnar liquid water (L), and SST are all retrieved simultaneously. In this case, the output vector of geophysical parameters

is presented by $G = \{W, V, L, SST\}$. The NN1 algorithm uses five SSM/I channels, including 19 GHz and 37 GHz for horizontal and vertical polarization and 22 GHz for vertical polarization.

Figure 3.3 shows a comparison of the GSW, NN0, and NN1 algorithms for the case of a mid-latitude storm located northeast of Australia. The GSW algorithm does not produce reliable retrievals in the areas with high levels of moisture close to the center of the storm. White gaps (missed retrievals) can be seen there in the most active and energetic areas of the storm. Wind speeds from these areas, if they were available, would provide a major impact in DAS. NN0 fills these gaps; however, it does underestimate the high wind speeds. NN1 produces reliable and accurate high winds even for high levels of moisture.

Meng et al. (2007) used the NN multiparameter retrieval approach developed by Krasnopolsky et al. (1996, 1999) to design another NN multiparameter retrieval algorithm (NN2 in Table 3.2) for the SSM/I. They used all 7 SSM/I BTs as inputs. Their output vector also had four components $G = \{W, T_a, H, SST\}$ where surface wind speed (W), surface air temperature (T_a), humidity (H), and SST were retrieved simultaneously. In this case, the training database was limited to maximum wind speeds of about 20 m/s. Moreover, there were only a few higher speed events with $W > 15$–17 m/s in their database.

Roberts et al. (2010) developed an NN multiparameter retrieval algorithm (NN3 in Table 3.2) for the SSM/I, which significantly improves the error characteristics of the air temperature, specific humidity, and SST as compared to previous methods. Improvements are due to correct accounting for the effects of high cloud liquid water and due to the use of a first-guess SST as an additional NN input.

Table 3.2 shows a comparison of the performance of all of the aforementioned empirical algorithms in terms of the accuracy of the retrieved surface wind speeds. It also shows statistics for a PB algorithm developed by Wentz (1997), which is based on a linearized numerical inversion (3.4) of a PB FM. The statistics presented in Table 3.2 were calculated using independent data sets. Table 3.2 shows that the NN algorithms outperform all other algorithms. All algorithms except for the NN algorithms show a tendency to overestimate high wind speeds. This happens because high wind speed events are usually accompanied by a significant amount of cloud liquid water in the atmosphere. Under these conditions the TF f becomes a complicated nonlinear function, and simple one-parametric regression algorithms cannot provide an adequate representation of this function and confuse high concentrations of cloud liquid water with very high wind speeds. NN1 shows the best overall performance, in terms of bias, RMSE, and high wind speed performance.

As mentioned above, one of the significant advantages of the NN1 algorithm is its ability to retrieve simultaneously not only wind speed but also the three other atmospheric and ocean surface parameters columnar water vapor V, columnar liquid water L, and SST. Krasnopolsky et al. (1999) showed that the accuracy of the retrievals for other geophysical parameters is also high and close to those obtained by the algorithms of the Alishouse et al. (1990) (for V) and Weng and

Grody (1994) (for L). In addition, Krasnopolsky et al. (1999, 2000) have shown that the errors of the multiparameter NN algorithms have a weaker dependence on the related atmospheric and surface parameters than the errors of the single-parameter algorithms. The retrieved *SST* in this case is not accurate (the RMSEs about 4 °C, see Krasnopolsky et al. (1996)); however, including *SST* in the vector of retrieved parameters reduces errors in other retrievals that are correlated with the *SST*. Figure 3.4 illustrates the coherence of the retrieved fields for three parameters (W, L, and V) and shows good agreements of the wind speed retrieved from the SSM/I, with those retrieved from the European Remote Sensing scatterometer ERS-2, and also measured by buoys.

For the multiparameter NN algorithm NN2 (Meng et al. 2007), the choice of the additional outputs, surface air temperature (T_a) and humidity (H), that are physically related to, and correlated with, *SST*, makes the accuracy of retrieved SSTs higher (the bias is ∼0.1 °C and RMSE, 1.54 °C). In accordance with the classical, "linear" RS paradigm, the SSM/I instrument does not employ frequency that is sensitive to SST. However, due to the nonlinear nature of the NN emulation and the proper choice of output parameters, the multiparameter NN algorithm can use weak nonlinear dependencies between NN inputs and outputs, and between NN outputs, to retrieve SST with higher accuracy. For the multiparameter NN algorithm NN3 (Roberts et al. 2010), the use of an SST first guess as an additional input, which is possible due to flexibility of the NN technique, improves the accuracy of the retrieved SST to an even greater extent (almost zero bias and an RMSE of 0.6 °C).

3.5.3 Controlling the NN Generalization for the SSM/I

The NN1 retrieval algorithm has been used as the operational algorithm in the global DAS at National Centers for Environmental Prediction (NCEP) of National Oceanic and Atmospheric Administration (NOAA) since 1998. Given five BTs, it retrieves the four geophysical parameters ocean surface wind speed, water vapor and liquid water concentrations, and SST. At high levels of liquid water concentration, microwave radiation cannot penetrate clouds and surface wind speed retrievals become impossible. BTs on these occasions fall far outside the training domain D_T. The retrieval algorithm in these cases, if not flagged properly, produces wind speed retrievals which are physically meaningless (i.e., not related to actual surface wind speed). Usually a statistical retrieval flag, based on global statistics, is used to detect such occurrences. Under complicated local conditions, however, it can produce a significant number of false alarms or does not produce alarms when required.

The validity check shown in Fig. 3.5, if added to a standard retrieval flag, helps to detect such occurrences. The NN SSM/I FM, FM1 (see Sect. 3.5.1) is used in combination with the NN1 retrieval algorithm. For each satellite measurement S,

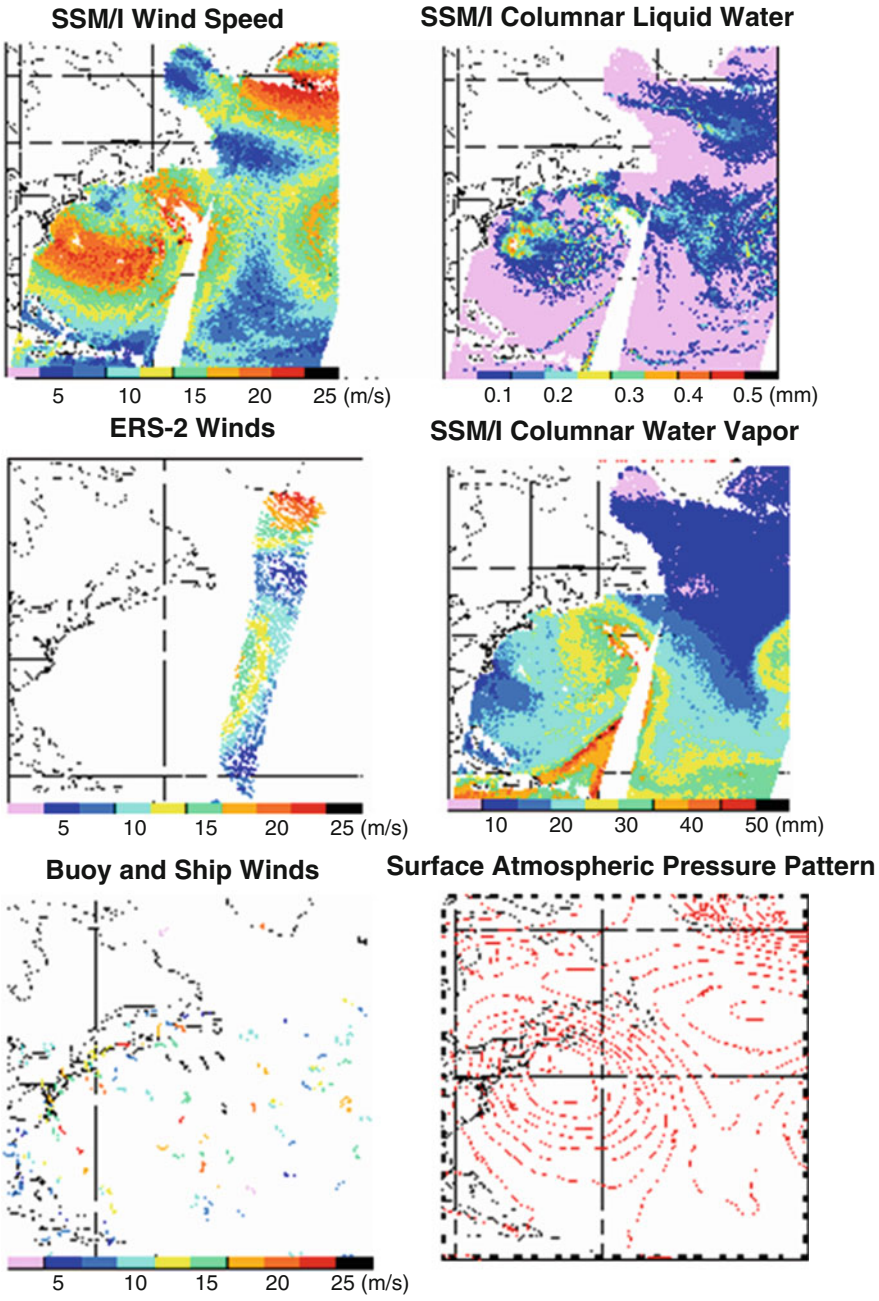

Fig. 3.4 Comparison of the wind speeds (*upper left*), columnar liquid water (*upper right*), and columnar water vapor (*middle right*) retrieved using the NN1 algorithm with the scatterometer (ERS-2) wind speeds (*middle left*), buoy wind speeds (*lower left*), and model pressure field (*lower right*) for the same time period

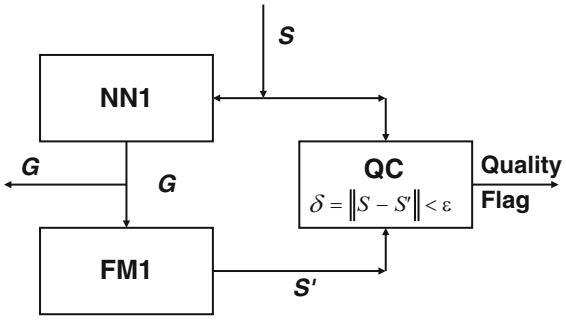

Fig. 3.5 An intelligent retrieval system consists of the SSM/I retrieval algorithm NN1, the FM, FM1, and a QC block implementing a validity check. NN1 emulating the inverse model retrieves vector G of four geophysical parameters: ocean surface wind speed (W), water vapor (V) and liquid water (L) concentrations, and sea surface temperature (SST) if given a vector of five BTs, S. This vector G is fed to FM1 emulating the FM to generate BTs S'. The difference $\delta = |S - S'|$ is monitored and raises a warning flag if it exceeds a suitably chosen threshold

the geophysical parameters retrieved from BTs, S, are fed into the NN SSM/I FM which produces another set of BTs, S'. For S within the training domain ($S \in D_T$), the difference, $\delta = |S - S'|$, is sufficiently small. For S outside the training domain, larger differences raise a warning flag, if it is above a suitably chosen threshold. It was shown by Krasnopolsky and Schiller (2003) that applying the control of generalization reduces the RMSE significantly, and the maximum error is reduced even more. Thus, this approach is very efficient at removing outliers.

3.6 Using NNs to Go Beyond the Standard Retrieval Paradigm

3.6.1 Point-Wise Retrievals

All retrieval algorithms discussed and presented up to this point correspond to the point-wise retrieval paradigm (see Sect. 3.1.1 and Fig. 3.1). In the framework of this paradigm, the retrieval algorithm (3.4), (3.6), and/or (3.8) maps a vector of satellite measurements, S, acquired at a particular location, to a vector of geophysical parameters, G, at the same location. As we can see, in the case of the SSM/I P2P approach, it provides satisfactory results. BTs utilized by these algorithms provide sufficient information to retrieve the wind speed and other geophysical parameters with acceptable accuracy. However, the P2P paradigm is not always successful. Sometimes, as in the case of wind vector retrievals from the QuikSCAT, this paradigm is not sufficient and should be augmented by methods capable of introducing additional information in the retrieval process.

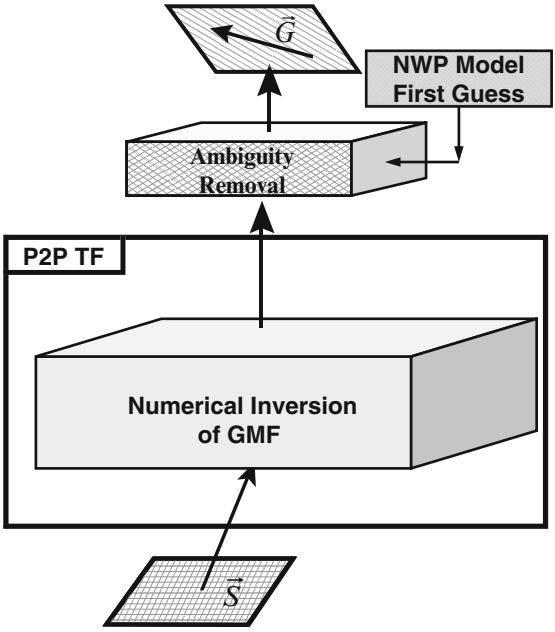

Fig. 3.6 The QuikSCAT retrieval algorithm consists of the P2P TF generated by numerical inversion of the geophysical model function (GMF) and augmented by a field-wise ambiguity removal procedure, which uses a first-guess field from an NWP model

3.6.2 Problems with Point-Wise Retrievals

QuikSCAT sensor is a spaceborn scatterometer that has been designed to provide ocean surface wind vector retrievals at high resolution (the size of a single resolved cell is about 25 km) over a swath width of 1,800 km (72 cells across the swath). The instrument has two beams with horizontal and vertical polarization and incident angles of 54 and 46 degrees, respectively, which provide four different backscatter measurements (2 fore and 2 aft) per target cell. For most of the cells along the swath, QuikSCAT provides four backscatter measurements and an azimuth look angle. This information, theoretically speaking, should be sufficient for accurate ocean surface wind vector retrievals.

The current retrieval procedure (Dunbar et al. 2006) is depicted in Fig. 3.6. It consists of two major steps:

1. Numerical inversion of the QuikSCAT empirical FM, or so-called geophysical model function, based on the maximum likelihood principle (2.10); this inversion produces from one to four wind vector solutions or ambiguities (this step actually determines implicitly a P2P TF).
2. An additional field-wise procedure – an ambiguity removal step, which uses a median filter and NCEP's global model first-guess wind field to select from four algorithm solutions one, so-called nudged, solution, which is closest to the first-guess wind pattern.

Thus, in this case, the P2P TF per se is not sufficient. The retrieved wind vectors do not have satisfactory accuracy (for wind direction, see below); also, the vectors retrieved from the neighboring cells often do not agree with each other. Thus, they do not create a smooth and continuous wind field. Consequently, the local, point-wise P2P retrieval procedure is not sufficient to retrieve wind vectors to produce wind patterns that are consistent and of reasonable accuracy.

The retrieval procedure requires additional nonlocal, field-wise information to produce smooth wind vector fields with consistent wind patterns. The ambiguity removal procedure provides such information that is taken from an NWP model 6-h-forecast first-guess wind field. The resulting nudged wind field is smooth and has good statistical properties (see the bias and RMSE in Table 3.4) when compared to the model or analysis wind vector field. This field is suitable for many applications (e.g., for marine meteorologists). However, if we want satellite retrievals to provide independent information about wind vectors (e.g., required for DAS), then problems arise (Krasnopolsky and Gemmill 2001).

From informational point of view, the nudged wind vector field, obtained after the application of the ambiguity removal procedure based on using a first-guess field, combines information from two different sources. It contains (1) local, point-wise satellite information about the wind vectors, which is presented by the maximum likelihood solutions, and (2) nonlocal, field-wise information about the same wind vector field, which was produced by the DAS and numerical forecast model and which is stored in the first-guess wind vector field. The ambiguity removal procedure actually performs a smooth fusion or integration of these two types of information: the satellite-derived information and the first-guess information.

The contribution of satellite information in the nudged solution varies from area to area; as a result, some parts of the wind vector field (such as areas between swathes and areas of heavy rain) that demonstrate a realistic continuous wind pattern may contain very little independent satellite information. To illustrate this possibility and to show that conventional error statistics are not reliable indicators of the quality of wind field in these cases, we consider an extreme case where there is no satellite information concerning wind direction in the maximum likelihood solutions, which are employed by the ambiguity removal procedure to select the nudged solution and using the first-guess wind vector field.

Now, let us assume that, for each QuikSCAT wind vector cell, we have, on average, M ambiguities ($1 \leq M \leq 4$ for QuikSCAT), and that we have one ambiguity vector per angle equal to $\beta = 360^\circ/M$ degrees (Krasnopolsky and Gemmill 2001). Let us also assume that all satellite-derived wind directions for these ambiguities are replaced by random numbers (i.e., corresponding to no satellite information about wind direction at all!) that are uniformly distributed over the interval $[0, \beta]$. In this case, the probability density function for wind direction is $P(x) = 1/\beta$. Let us apply the ambiguity removal procedure to these wind vectors and estimate the RMSE of the nudged solution, RMSE_N, obtained after the application of the ambiguity removal procedure. The RMSE_N is calculated vs. the first-guess wind vector field directions (these directions are also assumed to be uniformly distributed random numbers). The RMSE_N calculated for this hypothetical case provides us with the

Table 3.3 Zero skill RMSE for different number of ambiguities $M > 1$

M	2	3	4	6
RMSE_N	73°	49°	37°	25°

measure of zero skill or zero satellite information RMSE for the nudged solution for wind directions (in the case of M ambiguities). The RMSE_N can be calculated as

$$\text{RMSE}_N = \sqrt{\int_0^{\beta}\int_0^{\beta} (x_1 - x_2)^2 \cdot P\,(x_1, x_2) \cdot \mathrm{d}x_1 \mathrm{d}x_2} = \frac{\beta}{\sqrt{6}} = \frac{360^\circ}{\sqrt{6} \cdot M}$$

Table 3.3 summarizes the particular values of RMSE_N for different number of ambiguities, M. These values indicate essentially zero retrieval skill for particular numbers of ambiguities M. For example, if, in the case of four ambiguities ($M = 4$), comparison of the nudged solution wind vector field area with the analysis gives $\text{RMSE} \geq \text{RMSE}_N = 37^\circ$, then it indicates that, in this area, the nudged wind vector field is dominated by the first guess and contains very little independent information derived from the QuikSCAT sensor. Table 3.3 also shows that, in a hypothetical case where the retrieval algorithm produced six ambiguities (e.g., a scatterometer with three beams), the ambiguity removal procedure using the first-guess field could produce the nudged field with an acceptable RMSE_N of about 25°; however, this field would contain no independent satellite information concerning wind direction. All of the information concerning wind direction would come from the first-guess field in this case.

Taking into account the above considerations, it is reasonable to introduce a complementary characteristic for the QuikSCAT nudged solution – an amount (percentage) of independent satellite information, α, which estimates the contribution of the QuikSCAT data to the nudged wind vector solution. This parameter will vary spatially; it will be different for different locations, and, considering informational content, the best way to calculate the parameter α is to use a Bayesian approach (Bishop 2006); however, in the interest of simplicity, we introduce the following linear approximation for α:

$$\alpha = \left(1 - \frac{\text{RMSE}}{\text{RMSE}_N}\right) \cdot 100\ \%$$

Table 3.4 shows selected wind speed and direction statistics (biases and RMSEs) for the nudged solution and for the maximum likelihood (ML) first ambiguity, which corresponds closely to the P2P retrieval paradigm. It also shows the contribution from independent satellite information via α. The results of an NN P2P algorithm are also included.

As can be seen from Table 3.4, the P2P NN algorithm provides a high-quality wind speed estimate, with essentially zero bias and an RMSE of the order of

Table 3.4 Selected wind speed and direction statistics for different QuikSCAT wind retrieval algorithms

	Wind speed ($m \cdot s^{-1}$)		Wind direction (deg)		
Algorithm/solution	Bias	RMSE	Bias	RMSE	α (%)
Nudged solution	−0.5	1.6	2°	18°	∼65
P2P ML – 1st ambiguity	−0.5	1.7	2°	59°	100
P2P NN	0.1	1.7	3°	51°	100

The P2P ML is for the P2P maximum likelihood case

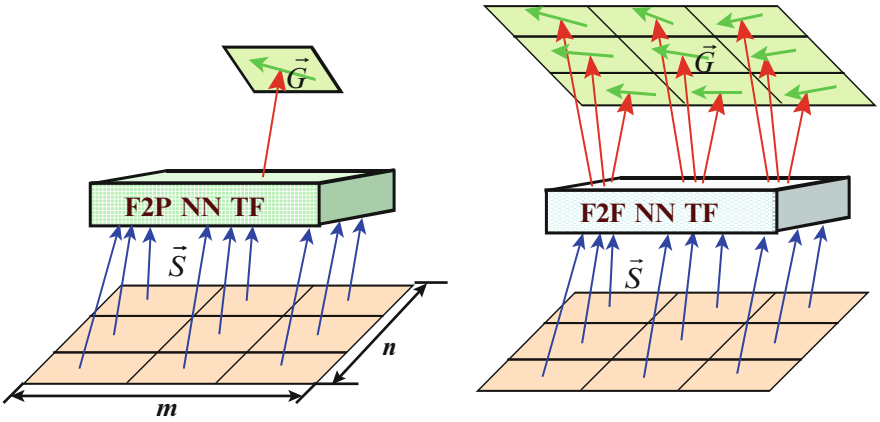

Fig. 3.7 Schematic representation of two NN-based field-wise approaches: F2P (*left figure*) and F2F (*right figure*). TF represents the transfer function. $m \times n$ is the base size, i.e., the number of cells that serve as NN inputs and/or outputs. The case for $n = m = 3$ is shown

1.7 $m \cdot s^{-1}$. For wind direction, the P2P NN algorithm produces a small but significant improvement over the first ambiguity obtained from the P2P algorithm (51° vs. ∼59°); however, even for the NN algorithm, which is based on the point-wise P2P paradigm, the accuracy in wind direction is not sufficient for most practical applications.

3.6.3 Field-Wise Retrieval Paradigms

The flexibility of the NN technique can be used to formulate an NN-based field-wise retrieval paradigm, *which allows us to introduce nonlocal, field-wise information in the retrieval process on the level of satellite measurements without using additional non-satellite information for regularization*. Figure 3.7 portrays two such alternative field-wise retrieval frameworks. In these approaches nonlocal, field-wise information is introduced in the retrieval process using satellite measurements themselves.

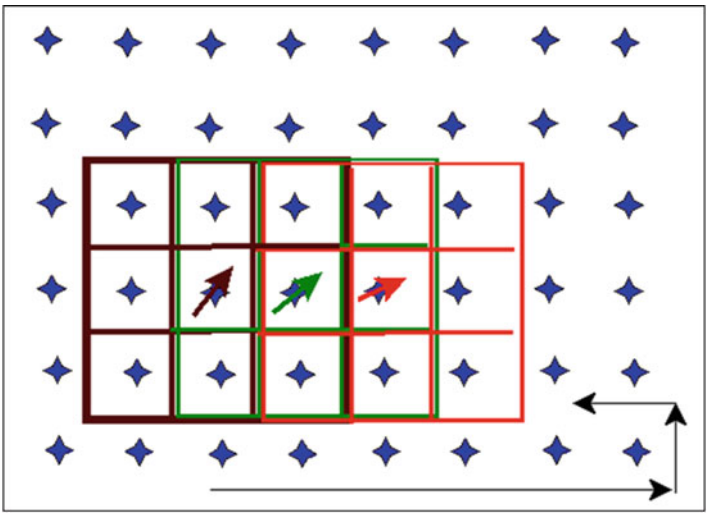

Fig. 3.8 Training and retrieving paths for F2P and F2F algorithms. *Colored frames* show the sequential positions of the base. *Colored arrows* show the wind vectors in the central cell of the base at each base position. *Black arrows* show the path of the center of the base across and along the swath

Instead of using the measurement vector S from one wind vector cell, alternative approaches use a set of the measurement vectors S from a section of the QuikSCAT swath (a base, or stencil, or template) with a size of $n \times m$ wind vector cells. The field-to-point (F2P) approach (see Fig. 3.7 (left), where the case $n \times m = 3 \times 3$ is shown) retrieves only one wind vector, which corresponds to the central wind vector cell of the section. However, in the process of training and then in the process of retrieval, the transition to each new (training or retrieval) step involves shifting template (see Fig. 3.8) one cell across the swath or one cell along the swath (e.g., at the edges). As the result, the sequential wind vector retrievals will be produced with a use of swath sections which significantly overlap; therefore, they will be significantly correlated and coherent.

The second, F2F, alternative approach (see Fig. 3.7 (right), where the case $n \times m = 3 \times 3$ is shown) brings into the retrieval process even more nonlocal information. In addition to field-wise information about satellite measurements over neighboring swath cells, it also involves, during the NN training, field-wise information about neighboring wind vectors, i.e., about acceptable wind vector patterns. However, in contrast to standard ambiguity removal procedures, the information about neighboring wind vectors is required only during the training of the algorithm. After training, in the retrieval mode, the F2F algorithm retrieves the entire $n \times m$ segment of the wind vector field in contrast to the P2P and F2P algorithms, which produce only one wind vector. Because the algorithm is assumed to be trained using physically coherent wind fields, we expect the retrieved wind vector field to be coherent and physically meaningful without additional smoothing.

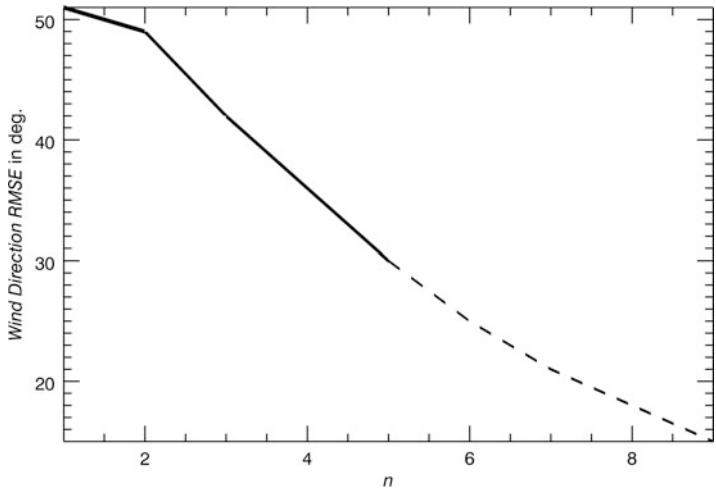

Fig. 3.9 RMSE of the wind directions retrieved by a F2F NN algorithm vs. n ($n \times n$ is the base size, i.e., the number of cells in the segments which have been used as inputs and outputs by the algorithm)

Development of the F2P and F2F retrieval algorithms for QuikSCAT requires a new type of collocation matchup data sets to be created. For example, traditional satellite/buoy collocations are suitable only for the P2P paradigm. Both the F2P and F2F frameworks require field-wise ground truth data – a continuous wind vector field. Analyzed wind vector fields produced by DAS can be used to create training sets for the F2P and F2F NN retrieval algorithms. Creating training sets for the F2P and F2F algorithms is a more difficult process than in the case of the P2P algorithm. The training set is larger and its structure is more complex.

A limited training set was created for QuikSCAT to perform experiments with NN field-wise retrieval algorithms. NCEP analysis wind fields have been collocated with QuikSCAT data. Our experiments showed that implementation of the F2P configuration for the NN algorithm leads to a significant improvement over the P2P configuration; however, the F2F configuration results in an additional improvement over the F2P configuration. The results of our experiments with the field-wise NN configurations are presented in Fig. 3.9. This figure shows that the accuracy of the wind direction retrieved by a F2F NN algorithm improves with increasing n, where $n \times n$ is the base size (the number of cells in the segments, which are used as inputs and outputs) of the algorithm. A smooth extrapolation, shown in the figure by the dashed line, suggests that, at $n = 7$ or 8, the accuracy of the NN F2F algorithm becomes comparable with the accuracy of algorithms using background fields for the ambiguity removal.

Figure 3.10 shows the wind direction binned scatter plot for the case of an NN with 50 hidden neurons and a 4×4 base size. It is noteworthy that F2F NN with the 8×8 base has about two hundred inputs and outputs. To train such an NN, a nonlinear optimization problem with a dimensionality of about 20,000 must be

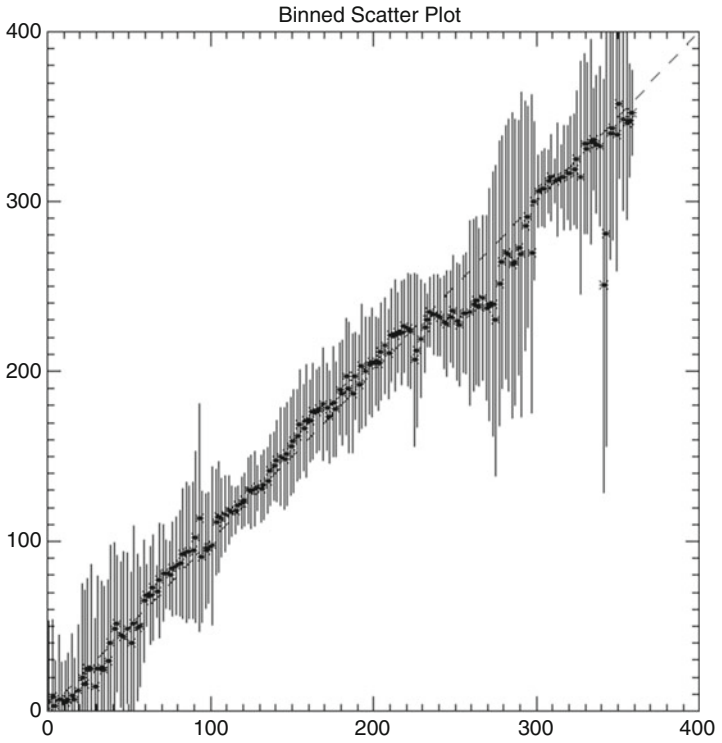

Fig. 3.10 Wind direction binned scatter plot. The *horizontal axis* shows the analyzed wind directions in degrees, and the *vertical* represents the F2F NN (with 50 hidden neurons and 4×4 base size) wind directions in degrees. *Asterisks* show binned values, and the *bars* are proportional to the scatter inside the bin

solved. In Chap. 4, we demonstrate that NN techniques have been developed to deal with increasingly large NNs, and the training of such and even larger NNs is possible using modern computers.

In the point-wise P2P framework, NNs are one of many tools (together with regression-based approaches) for constructing empirical TFs. *For alternative field-wise (F2P and F2F) retrieval paradigms, the NN technique becomes a unique tool for implementing them.* NNs are uniquely suited for developing F2P and F2F retrieval algorithms.

3.7 Discussion

In this chapter we discussed a broad class of NN applications dealing with the solution of the RS forward and inverse problems. These applications are closely related to the conventional and variational retrievals, which estimate geophysical

parameters from remote satellite measurements. Both conventional and variational techniques require a mechanism to convert satellite measurements into geophysical parameters or vice versa. Conventional retrievals use a TF (a solution to the inverse problem) and variational retrievals use an FM (a solution to the forward problem) for this purpose. From a mathematical point of view, the TF and the FM can be considered as continuous nonlinear mappings. For the NN technique is a generic technique for continuous nonlinear mapping, it can be used for modeling both TFs and FMs.

Theoretical considerations presented in this chapter were illustrated using several real-life applications that exemplify the NN-based intelligent approach (e.g., the approach and design presented in Fig. 3.5) where the entire retrieval system, including the QC block, is designed by combining several specialized NNs. This approach offers significant advantages for a number of practical applications. The intelligent retrieval system produces not only accurate retrievals but also performs an analysis and QC of the retrievals and environmental conditions, rejecting poor retrievals in the process.

The applications discussed in this chapter illustrate possibilities for successfully applying NNs in the framework of a conventional P2P point-wise retrieval paradigm. In this case, NNs can effectively compete with other statistical and PB methods and provide faster and more accurate retrievals. However, as we showed in Sect. 3.6, NNs also allow us to formulate new field-wise F2P and F2F retrieval paradigms, which can be successfully implemented only using NNs.

The NN applications presented in this chapter illustrate strengths and weaknesses of the NN technique for inferring geophysical parameters from RS measurements. NNs successfully compete with other statistical methods and usually perform better because they are able to emulate complex nonlinear functional relationships between the inputs and the outputs in an optimal way. NNs can also successfully compete with PB approaches because, in many cases, explicit knowledge of complicated physical processes in ES is limited, and an NN-based empirical approach is well suited to address such problems. It can encompass more physics implicitly (by learning from the data) than PB approaches can include explicitly. NN-based field-wise algorithms can even explicitly take into account nonlocal field-wise information that is difficult, or impossible, with other approaches.

However, the success of the NN approach strongly depends on the representativeness of the data sets that are used for training (see Sect. 2.3.3). Data availability, accuracy, quality, quantity, and representativeness are essential for successful development of NN applications.

References

Abdelgadir A et al (1998) Forward and inverse modeling of canopy directional reflectance using a neural network. Int J Remote Sens 19:453–471

Aires F, Rossow WB, Scott NA, Chedin A (2002) Remote sensing from the infrared atmospheric sounding interferometer instrument: 2 Simultaneous retrieval of temperature, water vapor, and ozone atmospheric profiles. J Geophys Res. doi:10.1029/2001JD001591

Alishouse JC et al (1990) Determination of oceanic total precipitable water from the SSM/I. IEEE Trans Geosci Remote GE-23:811–816
Ammar A, Labroue S, Obligis E, Mejia C, Crépon M, Thiria S (2008) Sea surface salinity retrieval for the SMOS mission using neural networks. IEEE Trans Geosci Remote 46:754–764
Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
Brajard J, Jamet C, Moulin C, Thiria S (2006) Use of a neuro-variational inversion for retrieving oceanic and atmospheric constituents from satellite ocean color sensor: application to absorbing aerosols. Neural Netw 19:178–185
Cabrera-Mercader CR, Staelin DH (1995) Passive microwave relative humidity retrievals using feedforward neural networks. IEEE Trans Geosci Remote 33:1324–1328
Cornford D, Nabney IT, Ramage G (2001) Improved neural network scatterometer forward models. J Geophys Res 106:22331–22338
Daley R (1991) Atmospheric data analysis. Cambridge University Press, New York
Davis DT et al (1995) Solving inverse problems by Bayesian iterative inversion of a forward model with application to parameter mapping using SMMR remote sensing data. IEEE Trans Geosci Remote 33:1182–1193
Derber JC, Wu W-S (1998) The use of TOVS cloud-cleared radiances in the NCEP SSI analysis system. Mon Weather Rev 126:2287–2299
Dunbar RS et al (2006) QuikSCAT science data product user manual, version 3.0. Jet Propulsion Laboratory, Doc. D-18053-Rev. A. http://podaac.jpl.nasa.gov/allData/quikscat/L2B12/docs/QSUG_v3.pdf
Eyre JR, Lorenc AC (1989) Direct use of satellite sounding radiances in numerical weather prediction. Meteorol Mag 118:13–16
Goodberlet MA, Swift CT (1992) Improved retrievals from the DMSP wind speed algorithm under adverse weather conditions. IEEE Trans Geosci Remote 30:1076–1077
Goodberlet MA, Swift CT, Wilkerson JC (1989) Remote sensing of ocean surface winds with the special sensor microwave imager. J Geophys Res 94:14547–14555
Krasnopolsky V (1996) A neural network forward model for direct assimilation of SSM/I brightness temperatures into atmospheric models. Working group on numerical experimentation blue book. 1.29–1.30. Tech note, OMB contribution No 134, NCEP/NOAA Camp Springs, MD. http://polar.ncep.noaa.gov/mmab/papers/tn134/OMB134.pdf
Krasnopolsky V (1997) A neural network based forward model for direct assimilation of SSM/I brightness temperatures. Tech note, OMB contribution No 140, NCEP/NOAA Camp Springs, MD. http://polar.ncep.noaa.gov/mmab/papers/tn140/OMB140.pdf
Krasnopolsky VM (2007) Reducing uncertainties in neural network Jacobians and improving accuracy of neural network emulations with NN ensemble approaches. Neural Netw 20:454–461
Krasnopolsky VM, Gemmill WH (2001) Using QuikSCAT wind vectors in data assimilation system. Tech note, OMB Contribution No 209, NOAA/NCEP/EMC Camp Springs, MD. http://polar.ncep.noaa.gov/mmab/papers/tn209/omb209.pdf
Krasnopolsky VM, Schiller H (2003) Some neural network applications in environmental sciences part I: Forward and inverse problems in satellite remote sensing. Neural Netw 16:321–334
Krasnopolsky V, Breaker LC, Gemmill WH (1995) A neural network as a nonlinear transfer function model for retrieving surface wind speeds from the special sensor microwave imager. J Geophys Res 100:11033–11045
Krasnopolsky V, Gemmill WH, Breaker, LC (1996) A new transfer function for SSM/I based on an expanded neural network architecture. Tech note, OMB contribution No 137, NCEP/NOAA Camp Springs, MD. http://polar.ncep.noaa.gov/mmab/papers/tn137/omb137.pdf
Krasnopolsky VM, Gemmill WH, Breaker LC (1999) A multiparameter empirical ocean algorithm for SSM/I retrievals. Can J Remote Sens 25:486–503
Krasnopolsky VM, Gemmill WH, Breaker LC (2000) A neural network multi-parameter algorithm SSM/I ocean retrievals: comparisons and validations. Remote Sens Environ 73:133–142
Lorenc AC (1986) Analysis methods for numerical weather prediction. Q J Roy Meteor Soc 122:1177–1194

McNally AP, Derber JC, Wu W-S, Katz BB (2000) The use of TOVS level 1B radiances in the NCEP SSI analysis system. Q J Roy Meteor Soc 126:689–724

Meng L et al (2007) Neural network retrieval of ocean surface parameters from SSM/I data. Mon Weather Rev 135:586–597

Mueller MD et al (2003) Ozone profile retrieval from global ozone monitoring experiment data using a neural network approach (Neural Network Ozone Retrieval System (NNORSY)). J Geophys Res 108:4497. doi:10.1029/2002JD002784

Parker RL (1994) Geophysical inverse theory. Princeton University Press, Princeton

Parrish DF, Derber JC (1992) The National meteorological center's spectral statistical-interpolation analysis system. Mon Weather Rev 120:1747–1763

Petty GW (1993) A comparison of SSM/I algorithms for the estimation of surface wind. In: Proceedings of the shared processing network DMSP SSM/I algorithm symposium, Monterrey, 8–10 June

Petty GW, Katsaros KB (1992) The response of the special sensor microwave/imager to the marine environment Part I: An analytic model for the atmospheric component of observed brightness temperature. J Atmos Ocean Tech 9:746–761

Petty GW, Katsaros KB (1994) The response of the SSM/I to the marine environment part II: a parameterization of the effect of the sea surface slope distribution on emission and reflection. J Atmos Ocean Tech 11:617–628

Phalippou L (1996) Variational retrieval of humidity profile, wind speed and cloud liquid-water path with the SSM/I: potential for numerical weather prediction. Q J Roy Meteor Soc 122:327–355

Pierce L, Sarabandi K, Ulaby FT (1994) Application of an artificial neural network in canopy scattering inversion. Int J Remote Sens 15:3263–3270

Prigent C, Phalippou L, English S (1997) Variational inversion of the SSM/I observations during the ASTEX campaign. J Appl Meteorol 36:493–508

Roberts B, Clayson CA, Robertson FR, Jackson D (2010) Predicting near-surface atmospheric variables from SSM/I using neural networks with a first guess approach. J Geophys Res. doi:10.1029/2009JD013099

Schiller H, Doerffer R (1999) Neural network for emulation of an inverse model-operational derivation of case II water properties from MERIS data. Int J Remote Sens 20:1735–1746

Smith JA (1993) LAI inversion using a back-propagation neural network trained with a multiple scattering model. IEEE Trans Geosci Remote GE-31:1102–1106

Stoffelen A, Anderson D (1997) Scatterometer data interpretation: estimation and validation of the transfer function CMOD4. J Geophys Res 102:5767–5780

Stogryn AP, Butler CT, Bartolac TJ (1994) Ocean surface wind retrievals from special sensor microwave imager data with neural networks. J Geophys Res 90:981–984

Thiria S, Mejia C, Badran F, Crepon M (1993) A neural network approach for modeling nonlinear transfer functions: application for wind retrieval from spaceborn scatterometer data. J Geophys Res 98:22827–22841

Tsang L et al (1992) Inversion of snow parameters from passive microwave remote sensing measurements by a neural network trained with a multiple scattering model. IEEE Trans Geosci Remote GE-30:1015–1024

Vapnik VN, Kotz S (2006) Estimation of dependences based on empirical data (information science and statistics). Springer, New York

Weng F, Grody NG (1994) Retrieval of cloud liquid water using the special sensor microwave imager (SSM/I). J Geophys Res 99:25535–25551

Wentz FJ (1997) A well-calibrated ocean algorithm for special sensor microwave/imager. J Geophys Res 102:8703–8718

Young GS (2009) Implementing a neural network emulation of a satellite retrieval algorithm. In: Haupt SE, Pasini A, Marzban C (eds) Artificial intelligence methods in environmental sciences. Springer, New York

Chapter 4
Applications of NNs to Developing Hybrid Earth System Numerical Models for Climate and Weather

- *Everything we think we know about the world is a model*
- *Our models do have a strong congruence with the world*
- *Our models fall far short of representing the real world fully*

– Donella H. Meadows, *Thinking in Systems: A Primer*

Abstract In this chapter, numerical modeling background is introduced and a number of neural network (NN) applications developed for numerical weather prediction (NWP) models and climate simulation systems are presented. The hierarchy of numerical models describing weather and climate processes of different scales is introduced and discussed. The notion of hybrid models that combine deterministic physically based parts with statistical blocks is introduced. Several atmospheric and oceanic applications of the NN technique to produce statistical blocks for hybrid numerical models are introduced and discussed in detail. These applications include fast and accurate NN emulations of atmospheric radiation parameterizations and new NN-based convection parameterization for atmospheric models, and fast and accurate NN emulations of nonlinear wave-wave interaction parameterization for ocean wind wave models. The chapter contains an extensive list of references giving extended background and further detail to the interested reader on each examined topic. It can serve as a textbook and an introductory reading for students and beginner and advanced investigators interested in learning how to apply the NN emulation technique to different numerical modeling problems.

Increasing demand for accurate weather prediction and climate projections leads to tremendous complexity of modern NWP models and climate simulation systems. The required growth of model complexity leaves behind not only our understanding of first principles but also the computational capabilities of modern supercomputers. Future trends in climate and weather modeling will continue to increase computational and storage requirements. Scientific advances will require increases in the physical resolution of the models; an increase in the number of ensemble

V.M. Krasnopolsky, *The Application of Neural Networks in the Earth System Sciences*, Atmospheric and Oceanographic Sciences Library 46, DOI 10.1007/978-94-007-6073-8_4,

runs; enhanced "quality" in terms of clouds, aerosols, biogeochemical cycles, and other processes; and a broadening of the overall scope that will include the upper-atmosphere regions.

The climate system is a subsystem of the ES (Schellnhuber 1999). According to the modern concept, the climate system consists of the abiotic world, the geosphere, which is sometimes called the physical climate system, and the living world, called the biosphere (see Fig. 1.1). The geosphere is further divided into subsystems, namely, the atmosphere, the hydrosphere (mainly the oceans but also lakes and rivers), the cryosphere (inland ice, sea ice, permafrost, and snow cover), the pedosphere (the soils), and the lithosphere (the Earth's crust and the more flexible upper Earth's mantle) (Peixioto and Oort 1992).

All components or subsystems of the climate systems are also complex systems. For example, the atmosphere is a nonlinear system encompassing a large variety of physical and chemical processes of very different spatial and temporal scales and actively interacting with other subsystems of the climate system (ocean, land, ice, etc.). The ocean is also a nonlinear system of interacting physical, chemical, and biological processes of very different scales; it interacts in many different ways (with multiple feedbacks) with the atmosphere, ice, and land.

The past several decades have revealed a well-pronounced trend in ESS. This trend marks a transition from investigating simpler linear or weakly nonlinear single-disciplinary systems like simplified atmospheric or oceanic systems that include a limited description of the physical processes, to studying complex non-linear multidisciplinary systems like coupled atmospheric-oceanic climate systems that take into account atmospheric physics, chemistry, land-surface interactions, etc. The most important property of a complex interdisciplinary system is that it consists of subsystems that, by themselves, are complex systems. Accordingly, the scientific and practical significance of interdisciplinary complex ES/environmental numerical models and prediction systems consisting of multiple coupled models has increased tremendously during the last few decades, due to improvements in their quality via better understanding of the basic ES processes and their relationships, developments in numerical modeling, and growing computing capabilities. Thus, global ES modeling activities are going to consume a tremendous amount of computing resources.

NNs are increasingly being applied to reduce demands for computing resources. They are used to accelerate calculations in NWP and climate simulation systems. When our understanding of first principles is not complete, NNs can be used to introduce in these systems new physics learned from observed or simulated data. NNs are also applied for data mining to automatically extract useful information (relationships, correlations, etc.) from massive and fast growing amounts of observed and simulated data.

In the following sections of this chapter, a numerical modeling background is introduced and a number of NN applications developed for NWP models and climate simulation systems are presented. In Sect. 4.1, the hierarchy of numerical models describing weather and climate processes of different scales is introduced and discussed. Parameterizations of model physics as a method of representing

physical processes in numerical models are discussed. The notion of hybrid models that combine deterministic physically based parts with statistical blocks is introduced in Sect. 4.2. Several atmospheric applications of the NN technique dedicated to produce statistical blocks for hybrid numerical models are introduced and discussed in details in Sect. 4.3. These applications include fast and accurate NN emulations of atmospheric radiation parameterizations and new NN-based convection parameterization for atmospheric models. Section 4.4 describes an oceanic application of the NN technique, an accurate NN emulations of nonlinear wave-wave interaction parameterization for ocean wind wave models. In Sect. 4.5, advantages and limitations of NN techniques presented in the chapter are discussed.

4.1 Numerical Modeling Background

As mentioned in Chap. 1, our understanding of "first principles" governing climate and weather systems and their subsystems is codified into systems of PDEs. These PDEs, approximated with a spectral or grid-point numerical scheme, constitute numerical climate model which can be integrated using modern computers. A large variety of highly nonlinear processes with broad spectra of spatial and temporal scales contribute to ES. The temporal scales range from several minutes (some weather events) to hundreds of millions of years (paleoclimate phenomena), and the spatial scales range from tens of thousands of kilometers (global phenomena) to millimeters (size of water droplets in the cloud). In such situations, a single numerical model cannot encompass the entire complexity of ES or even of a single but also tremendously complex, subsystem of ES, such as the climate system. This is why a wide range of numerical models with different resolutions that cover different spatial domains and incorporate processes of different spatial and temporal scales have been developed (Claussen 2001). Within this range, there is a large variety of models from GCMs, describing global, large-scale climate and weather patterns (see Sect. 2.1.4 and Fig. 2.1), to large eddy simulation models that describe the dynamics and evolution of single atmospheric and/or oceanic vortices.

Each of these numerical models has two basic characteristics: (1) spatial resolution, λ (actually two different resolutions, a horizontal and a vertical), and (2) an integration time step, τ. By definition, the numerical model cannot directly take into account processes or resolve features of size $r \leq \lambda$ and of duration $t \leq \tau$. Symbolically, a numerical model over a domain D can be written as

$$\begin{aligned} &\frac{\partial \psi}{\partial t} + \Omega(\psi, x) = P(\psi, x) + F(x, t); \quad x \in D \\ &\psi_{t=0} = \psi_0; \quad \psi_{x=B} = \psi_B \end{aligned} \tag{4.1}$$

where ψ represents the 3-D prognostic or dependent variables (e.g., temperature, wind, moisture); ψ_0, the initial conditions usually produced by a DAS (see Sect. 3.1.2), and ψ_B are the boundary conditions for ψ; x is a vector of independent

variables (e.g., latitude, longitude, and pressure, height, or depth); Ω represents the model dynamics (the set of 3-D PDEs of motion, thermodynamics, etc., approximated with a spectral, grid-point, or finite-element numerical scheme); F is the external forcing; and P is the term (called model physics) that effectively takes into account subgrid scale processes. Usually model physics is implemented as a superposition of several parameterizations, $p_k(\psi, x)$, each of which effectively describes a physical process, i.e., $P(\psi, x) = \sum_k p_k(\psi, x)$. Subgrid scale processes are those with spatial scales $r \leq \lambda$ and/or duration $t \leq \tau$. The model does not resolve such processes explicitly, but they are effectively taken into account through the appropriate parameterizations. Numerical models (4.1) based on "first principle" equations are also called deterministic models.

4.1.1 Climate- and Weather-Related Numerical Models and Prediction Systems

Global Models

Marked progress has been achieved during the past decades in modeling the separate elements of the climate system (e.g., Grassl 2000). This stimulated effort to put all of the separate pieces together, first in the form of comprehensive coupled models of atmospheric and oceanic circulation or coupled general circulation models (CGCMs or simply GCMs) and eventually in the form of climate system models which also include biological and geochemical processes (e.g., Foley et al. 1998; Cox et al. 2000). Comprehensive models of global atmospheric and oceanic circulation describe many details of the flow, such as individual weather systems and regional currents in the ocean.

Modern GCMs are either fully coupled atmosphere-ocean-land/biosphere-chemistry models or partially coupled models (e.g., with the chemistry component calculated off-line, driven by the flow simulated by an atmosphere-ocean-land model). CGCMs for climate simulation and weather prediction are based on solving time-dependent 3-D geophysical fluid dynamics equations (4.1) on a sphere because the domain, D, includes the entire globe in this case. P in (4.1) includes the parameterized model physics and chemistry (e.g., the long- and short-wave atmospheric radiation, turbulence, convection and large-scale precipitation processes, clouds, and interactions with land and ocean processes and the constituency transport and chemical reactions, respectively). GCMs treat physics in a simplified way using parameterizations to effectively represent the processes involved (see Sect. 4.2.2).

While scientific problems that use these models are among the most computationally intensive applications in the history of scientific exploration, the models employ drastic simplifications in their treatment of many processes important in climate

and weather. For example, parameterizations must effectively represent the effect of clouds over time and space scales that are well below the resolution of the scales explicitly treated in GCMs, which resolve atmospheric features with space scales of order 100 km and time scales of order 10 min (Randall et al. 2003). NWP models typically operate at smaller spatial and temporal scales, but most cloud processes are still acting well below these scales.

Another example of a complex global model is an ocean wind wave model developed for ocean wave simulation and forecasting (Tolman 2002). It is based on a form of the spectral energy or action balance equation

$$\frac{DF}{Dt} = S_{\text{in}} + S_{\text{nl}} + S_{\text{ds}} + S_{\text{sw}} \tag{4.2}$$

where F is the 2-D wave spectrum, S_{in} is the input source term, S_{nl} is the nonlinear wave-wave interaction source term, S_{ds} is the dissipation or "whitecapping" source term, and S_{sw} represents additional shallow water source terms. In Sect. 4.4, an NN emulation of the S_{nl} source term is introduced and discussed.

The major limitations of GCMs are limited resolution and simplified model physics. Even within these limitations, the application of GCMs and CGCMs to long-term climate studies leads to high computational cost. High resolution is required to resolve meso- and fine-scale weather and climate features related to convection processes and different types of clouds and precipitations in the atmosphere and to different types of circulation processes in the ocean. State-of-the-art CGCMs have resolutions of several tens of kilometers. Even using the most powerful computers, only a very limited number of multi-decadal experiments can be performed with such high-resolution models. GCMs today satisfactorily reproduce large-scale weather and climate features. However, because of simplified parameterized physics that GCMs use, they cannot simulate accurately many important fine-scale processes like cloudiness and convective precipitations (e.g., Rasch et al. 2000).

Regional Models

Regional models operate over a limited domain (typically continental scale or smaller). As a result, they can be run at finer spatial and temporal resolutions than a global model (although they still require many simplifications) at the same computational cost. Symbolically they can be represented by Eq. (4.1) and usually use the same parameterized physics as GCMs. They receive initial and boundary conditions and large-scale forcing from GCMs. Because of their higher resolution, regional models resolve mesoscale features significantly better than GCMs. The quality of forecasts from regional models, however, can be limited by the boundary conditions that the models receive, particularly for cases where the regional model is not nested within a larger model.

Cloud-Resolving Models

Cloud-system-resolving models (CSRMs) or simply cloud-resolving models (CRMs), first developed in the 1970s and 1980s (e.g., Krueger 1988), operate over a limited area (typically several hundred by several hundred kilometers) at finer spatial and temporal scales than global and regional models (although they still require many simplifications). They resolve many of the phenomena that global and regional models must ignore (e.g., higher-resolution fluid dynamical motions that can resolve some updrafts and downdrafts, convective organization, mesoscale circulations, and stratiform and convective components that interact with each other). These are models with sufficient spatial and temporal resolution to represent individual cloud elements and cover a wide enough range of time and space scales to permit statistical analysis of simulated cloud systems.

Despite their high computational cost, CSRMs do not simulate cloud systems from first principles. Although the cloud-scale and mesoscale *dynamical* processes, which must be parameterized in atmospheric GCMs, are explicitly simulated in CSRMs on scales down to a kilometer or so in the horizontal and 100 m or so in the vertical, the important microphysical, turbulent, and radiative processes must still be parameterized. These processes are still treated albeit crudely, but CRMs do resolve many more phenomena than today's global and regional model physics parameterizations.

Recently, attempts have been made to develop global models that resolve certain cloud-scale motions (down to horizontal resolutions of about 3 km) (global cloud-resolving models (GCRMs), e.g., Miura et al. 2005; Satoh et al. 2005). They are extremely expensive to run (by a factor of 10^5–10^6) when compared with the cost of a typical GCM and have generally been used in idealized settings for exploratory experiments (e.g., in a water-covered planet scenario).

Multiscale Modeling Framework or Superparameterization

The term "superparameterization," also known as a multiscale modeling framework (MMF), was originally suggested by Grabowski (2001) and subsequently developed by a group at Colorado State University (e.g., Khairoutdinov and Randall 2001; Randall et al. 2003). This concept refers to the embedding of a simplified two-dimensional CRM into each column of a global model (see Fig. 4.1).

For example, the MMF community atmosphere model (CAM) described by Randall et al. (2003) and Khairoutdinov et al. (2005) consists of a global climate model with a simplified 2-D CRM nested into each GCM (CAM) grid square. Because of the simplifications, such a model is substantially less expensive (by a factor of 10^3-10^4) than a GCRM but still far more costly (10^2-10^3) than a GCM using a conventional convective parameterization (Randall et al. 2003). The simplifications of CRM result in a substantial reduction in cost, with a consequent compromise in the accuracy of the physical representation. These issues are discussed more thoroughly in the references.

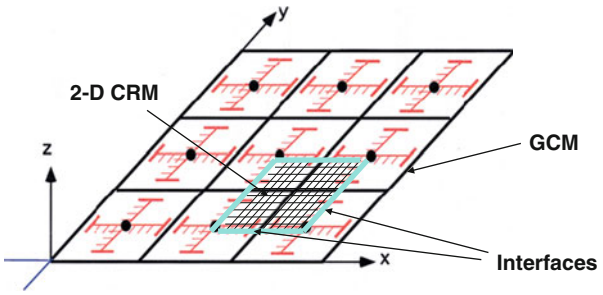

Fig. 4.1 Schematic representation of MMF

4.1.2 Representation of Physics in Global and Regional Models: Parameterizations of Physics

The subsystems of a complex climate (or weather or ocean wave) system, such as physical, chemical, and other processes, are so complicated that it is currently possible to include these processes in GCMs only in simplified parameterized form (a.k.a. parameterizations). These parameterizations constitute forcing, P, on the right-hand side for the dynamical equations (4.1) or S for (4.2). Parameterizations of model physics are approximate schemes, adjusted to model resolution and computer resources, and based on simplified equations for physical process and empirical data and dependencies. They are based on solving deterministic equations (like the radiative transfer equations); however, they also may include secondary empirical components based on traditional statistical techniques like regression. Accordingly, for widely used state-of-the-art GCMs, all major model components (i.e., subsystems) are predominantly deterministic. Not only the model dynamics but the model physics and chemistry are based on solving deterministic first principle physical or chemical equations.

In parameterizations, 3-D physics is reduced to 1-D parameterized model physics; however, even after such tremendous simplifications, the parameterizations so obtained are very complex and computationally time consuming. The percentage of total calculation time required for calculating model physics is model dependent. In all cases, however, the complete calculation of model radiation physics is the most time-consuming task for all GCMs (e.g., Morcrette et al. 2007, 2008; Manners et al. 2009). In both climate modeling and NWP, the calculation of radiative transfer is necessarily a trade-off between accuracy and computational efficiency. There exist very accurate methods such as line-by-line procedures that could be employed to very accurately calculate the radiative fluxes for every grid point at each time step. However, if the radiative transfer were to be computed for every grid point and at each time steps, it would generally require more CPU time than the rest of the model components combined, i.e., model dynamics and other physical parameterizations (Morcrette et al. 2008). Therefore, a number of additional simplifications are usually made to reduce the computational burden to manageable levels (Lacis

and Oinas 1991). However, even after aforementioned simplifications, radiation parameterizations cannot be made sufficiently computationally efficient to permit calculations for every grid point at each time step.

4.1.3 An Example: Parameterization of Long-Wave Radiation Physics

As a typical example, the method for calculating long-wave radiation (LWR) employed in National Center for Atmospheric Research (NCAR) CAM is presented here. The LWR radiation parameterization, a portion of the NCAR CAM radiation package *CAMRT*, is based on the long-wave (LW) radiative transfer equations in an absorptivity/emissivity formulation (see Collins (2001) and references there),

$$F^{\downarrow}(p) = B\left(p_t\right) \cdot \varepsilon\left(p_t, p\right) + \int_{p_t}^{p} \alpha\left(p, p'\right) \cdot \mathrm{d}B\left(p'\right)$$

$$F^{\uparrow}(p) = B\left(p_s\right) - \int_{p}^{p_s} \alpha\left(p, p'\right) \cdot \mathrm{d}B\left(p'\right) \tag{4.3}$$

where $F^{\uparrow}(p)$ and $F^{\downarrow}(p)$ are the upward and downward heat fluxes, $B(p) = \sigma \cdot T^4(p)$ is the Stefan-Boltzmann relation, pressures p_s and p_t refer to the top and surface atmospheric pressures, and α and ε are the atmospheric absorptivity and emissivity, respectively. To solve the integral equations (4.3), the absorptivity and emissivity have to be calculated by solving the following integrodifferential equations,

$$\alpha\left(p, p'\right) = \frac{\int_0^{\infty} \left\{\mathrm{d}B_{\nu}\left(p'\right)/\mathrm{d}T\left(p'\right)\right\} \cdot \left[1 - \tau_{\nu}\left(p, p'\right)\right] \cdot \mathrm{d}\nu}{\mathrm{d}B(p)/\mathrm{d}T(p)}$$

$$\varepsilon\left(p_t, p\right) = \frac{\int_0^{\infty} B_{\nu}\left(p_t\right) \cdot \left[1 - \tau_{\nu}\left(p_t, p\right)\right] \cdot \mathrm{d}\nu}{B\left(p_t\right)} \tag{4.4}$$

where the integration is over wave number ν and $B(p_t)$ is the Planck function. To solve Eq. (4.4) for the absorptivity and emissivity, additional calculations have to be performed, and the atmospheric transmission τ_{ν} also has to be calculated. This calculation involves time-consuming integration over the entire spectral range of gas absorption. The solution of this "simplified" 1-D problem (4.3, and 4.4) is so time consuming that it takes about 50 % of the total time required for the model integration even when the radiation is calculated with the reduced frequency (once per several integration time steps).

4.1.4 *Methods Currently Used to Reduce Computational Burden*

In the case of a complex GCM, calculation of the atmospheric radiation at spatial resolutions of a few degrees, as in the NCAR CAM, takes about 70 % of the total model computation time. Similar numbers could be presented for other GCMs. To reduce the cost of these calculations, they are usually made at lower temporal and/or spatial resolutions. Rather drastic reductions in temporal resolution are often made (e.g., radiation calculations are made every 1 or 3 h for the climate and global forecast models at NCEP and UKMO (Manners et al. 2009)). Between radiative transfer calculations, major changes may occur in the radiative profiles (caused primarily by two factors: changes in clouds and changes in the angle of incident solar radiation) that are not represented. A reduced horizontal resolution approach (the radiative calculations are performed on a coarser grid with a following interpolation of the results to an original finer grid) is used to speed up the radiation calculations at the European Centre for Medium-Range Weather Forecasts (ECMWF) (Morcrette et al. 2007, 2008). A reduced vertical resolution approach (the full radiation is calculated at every other vertical level and interpolated on the intermediate levels) is used in the Canadian operational Global Environmental Multiscale Model (e.g., Côté et al. 1998a, b).

Such approaches reduce horizontal, or vertical, or the temporal variability of the radiation fields. Thus, these approaches may reduce the accuracy of a model's radiation calculation and its spatial or/and temporal consistency with other parts of the model physics and with model dynamics, which may, in turn, degrade the accuracy of climate simulations and weather predictions. More frequent calculations of the model physics, which is desirable for temporal consistency with model dynamics, and the future introduction of more sophisticated model physics parameterizations will result in a further increase in the computational time spent calculating model physics.

In the wind wave model (4.2) discussed earlier, the calculation of the source term S_{nl} requires roughly 10^3–10^4 times more computational effort than all other aspects of the wave model calculations. Present operational constraints require that the computational effort to estimate S_{nl} should be of the same order of magnitude as for the rest of the wave model.

4.2 Hybrid Model Component and a Hybrid Model

As discussed above, one of the main problems in the development and implementation of modern high-quality high-resolution environmental models is the complexity of the physical, chemical, and other processes involved. Here we will discuss NN emulations as a tool for speeding up calculations of model physics

and for developing new model physics parameterizations, keeping in mind that the approach is applicable to other model components (chemical, hydrological, and other processes) as well.

The situation presented in previous sections provides the motivation to look for alternative, faster, and accurate ways of calculating model physics, chemistry, hydrology, and other processes. During the last decade, a new statistical learning approach based on NN approximations or emulations was applied for the accurate and fast calculation of atmospheric radiative processes (Krasnopolsky 1996, 1997; Chevallier et al. 1998) and for emulations of model physics parameterizations in ocean and atmospheric numerical models (Krasnopolsky et al. 2002, 2005a, 2008b, 2010). In these works, calculation of the model physics components has been accelerated by factors of 10–10^5 compared to the time needed for calculating the corresponding original parameterizations of the model physics.

Approaches formulated by Chevallier et al. (1998, 2000), Krasnopolsky (1996), and Krasnopolsky et al. (2002, 2005a, 2010) represent two different ways of a hybridization of first principle and NN components in the physics parameterizations as well as in complex climate and NWP models. These approaches introduce hybridization at two different system levels, at the level of the subsystem (a single parameterization) and at the level of the entire system, a numerical model. These two approaches lead to the concepts of a hybrid parameterization (HP) (Chevallier et al. 1998, 2000) and a hybrid model or hybrid GCM (HGCM) (Krasnopolsky et al. 2002, 2005a; Krasnopolsky and Fox-Rabinovitz 2006a, b). These two concepts have been debated by Chevallier (2005) and Krasnopolsky et al. (2005b) and are discussed in the following sections. Another type of hybrid model – the hybrid coupled model, where a simplified atmosphere is described by a neural network model and the ocean – by a dynamical model, was introduced and described by Tang and Hsieh (2003) and Li et al. (2005).

4.2.1 Hybrid Parameterizations of Physics

Chevallier et al. (1998, 2000) considered the LWR parameterization – a component of the complex GCM (the ECMWF global atmospheric model). Putting it in terms of the system levels, this single parameterization is considered to be the system and its constituents, the blocks calculating fluxes, the blocks calculating cloudiness, etc., as subsystems. The hybridization of first principle components with NN emulations is introduced on the level of these constituents and inside the system, which, in this case, is the LWR parameterization. A generic LWR parameterization can be represented as a mapping (2.1),

$$Y = M(X) \tag{4.5}$$

in this particular case, the input vector $X = \{S,T,V,C\}$, where the vector S represents surface variables, T is a vector (profile) of atmospheric temperatures, C is a profile

of cloud variables, and the vector V includes all other variables (humidity profile, different gas mixing ratio profiles, etc.). The output of the LWR parameterization, vector Y, is composed of two vectors Q and f, where $Y = \{Q,f\}$. Here Q is a profile of cooling rates $Q = \left\{C_r^1, C_r^2, \ldots, C_r^L\right\}$, where C_r^j is the cooling (or heating) rate of the jth vertical model level ($j = 1, \ldots, L$), and f is a vector of auxiliary fluxes computed by the LWR parameterization. Because of the presence of the input cloud variable C, the mapping (4.5) may have some finite discontinuities, that is, *it is almost continuous.*

The ECMWF LWR parameterization considered by Chevallier et al. (1998, 2000) is based on the Washington and Williamson (1977) approach, which allows separating the cloud variables, C. In this parameterization, level fluxes are calculated as

$$F\,(S,T,V,C) = \sum_i \alpha_i(C) F_i\,(S,T,V) \tag{4.6}$$

where i is an index for the vertical level. Each partial or individual flux $F_i(S,T,V)$ is a continuous mapping, and all discontinuities related to the cloudiness are included in $\alpha_i(C)$. In their HP, which they refer to as the "NeuroFlux," Chevallier et al. (1998, 2000) combined calculations of cloudiness functions $\alpha_i(C)$ based on first principle equations with NN approximations for a partial or individual flux $F_i(S,T,V)$. Thus, the flux, F (4.6), at each level is a linear combination of approximating NNs for fluxes F_i and cloud physics coefficients $\alpha_i(C)$. As a result, the "NeuroFlux" hybrid LWR parameterization developed by Chevallier et al. (1998, 2000) is an array or battery of about 40 NNs. To calculate "NeuroFlux" outputs, namely, the cooling rates, C_rs, linear combinations of the individual approximating NNs for F_i (Eq. 4.6) are differentiated at each vertical level, as

$$C_r(P) = \frac{\partial F(P)}{\partial P}, \tag{4.7}$$

where P is atmospheric pressure.

At moderate vertical resolutions of less than 50–60 vertical layers, the "NeuroFlux" has high accuracy; its bias is about 0.05 K/day, and the *RMSE* is about 0.1 K/day compared to the LWR parameterization by Washington and Williamson (1977). It is also eight times faster than the parameterization by Washington and Williamson (1977). However, because of NeuroFlux's suboptimal numerical design (see detailed discussion in Krasnopolsky et al. 2005b), at a vertical resolution of 60 layers or more, both accuracy and speed of NeuroFlux cannot be achieved simultaneously (Morcrette et al. 2008). Consequently, the NeuroFlux was used only in DAS for the 4D-Var linearized physics for which the accuracy requirements are less stringent.

As for limitations of the HP approach, the main one stems from a basic feature of the HP approach; it is based on the analysis of the internal structure of a particular parameterization. The final design of HP is based on and follows this internal structure. Because all parameterizations have different internal structures, the

approach and design of a HP developed for one parameterization usually cannot be used, without significant modifications, for another parameterization. For example, the approach used by Chevallier et al. (1998, 2000) and the design of the HP "NeuroFlux" is completely based on separating the dependence on the cloudiness (see Eq. 4.6). Many other LWR parameterizations, like the NCAR CAM radiation parameterizations (Collins 2001; Collins et al. 2002) or the parameterizations developed by Chou et al. (2001), do not allow for such separation of variables. Thus, for these radiation parameterizations as well as for the moisture model physics block parameterizations, the HP approach developed by Chevallier et al. (1998, 2000) cannot be applied directly; it must be significantly modified or redesigned for each particular new parameterization.

4.2.2 Hybrid Numerical Models

A new concept of a complex hybrid numerical model has been formulated and developed by Krasnopolsky et al. (2002, 2005a) (see also Krasnopolsky and Fox-Rabinovitz 2006a, b). The hybrid modeling approach considers the entire GCM as a system. Dynamics and parameterizations of physics, chemistry, etc., are considered to be the subsystems or components of the system. Hybridization in this case is introduced at the level of components inside the system. For example, the entire LWR, or short-wave radiation (SWR), or convection parameterization is represented by a single emulating NN as a single/elementary object or block. This NN approach is based on the general fact that any parameterization of model physics can be considered as a continuous or almost continuous mapping (2.1 or 4.5).

Accurate and Fast NN Emulations for Parameterizations of Model Physics

Krasnopolsky and Fox-Rabinovitz (2006a, b) formulated a developmental framework and test criteria that can be recommended for developing and testing the statistical components of the HGCM, e.g., NN emulations of model physics components. The developmental process consists of three major steps:

1. Problem analysis or analysis of the model component (the original parameterization or the target mapping (4.5 or 2.1)) to be approximated to determine the optimal structure and configuration of the NN emulations. That is to estimate the number of inputs and outputs and the first guess of the functional complexity of the original parameterization that determines an initial number of hidden neurons in one hidden layer of (2.2 and 2.3) (see Sects. 2.2.2, 2.3.1, and 2.3.4).
2. Generation of representative data sets for training, validation, and testing. This is achieved by running an original GCM, i.e., a GCM with the original parameterization, and by saving the simulated data to be used later for the NN training. When creating a representative data set, the original GCM must be run

long enough to produce all possible atmospheric states, phenomena, etc. Here, due to the use of simulated data, it is not a problem to generate sufficiently representative (and even redundant) data sets required to create high-quality NN emulations (see Sect. 2.5). Using model-simulated data for NN training allows a high accuracy of emulation to be achieved because simulated data are almost free of the problems typical in empirical data (like a high level of observational noise, sparse spatial and temporal coverage, and poor representation of extreme events).

3. Training the NN. Several different versions of NNs with different architectures (various numbers of hidden neurons), initialization, and training algorithms should be trained and validated. As for the NN architecture, the number of hidden neurons k should be kept to the minimum number that provides the required accuracy of the NN emulation (see Eqs. 4.8, 4.9, 4.10, 4.11, and 4.12).

Testing the trained NN emulation and the HGCM that uses this NN consists of two major steps:

1. The accuracy of the NN approximation is tested against the original parameterization using an independent test data set. In the context of the hybrid approach, the accuracy and improved computational performance of NN emulations, and eventually the HGCM, is always measured against the corresponding controls, namely, the original parameterization and its original GCM. Both the original parameterization and its NN emulation are complex multidimensional mappings. Many different statistical metrics of the emulation accuracy should be calculated to assure that a sufficiently complete evaluation of the emulation accuracy is obtained. For example, total, level, and profile statistics have to be evaluated (see Eqs. (4.8, 4.9, 4.10, 4.11, and 4.12)).
2. A comprehensive comparison and analysis of parallel HGCM and GCM runs is performed. For the parallel model simulations, all relevant model prognostic and diagnostic fields and their statistics should be analyzed and carefully compared to assure that the integrity of the original GCM and its parameterization, with all its details and characteristic features, is precisely preserved when using an HGCM with NN emulation (see Sects. 4.3 and 4.4). This test step involving model simulations is vital. GCMs are essentially nonlinear complex systems; in such systems, small systematic, and even random, approximation errors can accumulate over time and produce a significant impact on the quality of the model results. Therefore, the development and application framework of the new hybrid approach should be focused on obtaining a high accuracy in both NN emulations and HGCM simulations.

As was mentioned above, both the original parameterization and its NN emulation are complex multidimensional mappings. Because of their complexity, many different statistics and statistical cross sections should be calculated to obtain a satisfactory comparison between these two objects and to evaluate the accuracies of the NN emulations. The mean difference B (bias or systematic error

of approximation) and the root mean square difference *RMSE* (a root mean square error of approximation) between the original parameterization and its NN emulation are calculated as follows:

$$B = \frac{1}{N \times L} \sum_{i=1}^{N} \sum_{j=1}^{L} [Y(i,j) - Y_{\mathrm{NN}}(i,j)]$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{L} [Y(i,j) - Y_{\mathrm{NN}}(i,j)]^2}{N \times L}} \tag{4.8}$$

where $Y(i, j)$ and $Y_{\mathrm{NN}}(i, j)$ are outputs from the original parameterization and its NN emulation, respectively; the index $i = (latitude, longitude)$, $i = 1, \ldots, N$ determines the horizontal location (grid point) of a vertical profile; N is the number of the model horizontal grid points; and $j = 1, \ldots, L$ is the vertical index where L is the number of the model vertical levels.

The two bulk error characteristics presented in Eq. (4.8) describe the accuracy of the NN emulation integrated over the entire 4D (latitude, longitude, height, and time) data set. Using a minor modification of Eq. (4.8), the bias and *RMSE* for the mth vertical level of the model can be calculated as

$$B_m = \frac{1}{N} \sum_{i=1}^{N} [Y(i,m) - Y_{\mathrm{NN}}(i,m)]$$

$$RMSE_m = \sqrt{\frac{\sum_{i=1}^{N} [Y(i,m) - Y_{\mathrm{NN}}(i,m)]^2}{N}}. \tag{4.9}$$

The root mean square error can also be calculated for each ith profile (ith horizontal grid point):

$$prmse(i) = \sqrt{\frac{1}{L} \sum_{j=1}^{L} [Y(i,j) - Y_{\mathrm{NN}}(i,j)]^2} \ldots, i = 1, \ldots, N. \tag{4.10}$$

This error is a function of the horizontal location of the profile. It can be used to calculate a mean profile root mean square error *PRMSE* and its SD, σ_{PRMSE}, which characterize the entire data set and are location independent:

$$PRMSE = \frac{1}{N} \sum_{i=1}^{N} prmse(i)$$

$$\sigma_{PRMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} [prmse(i) - PRMSE]^2}. \tag{4.11}$$

The statistics (4.11) and (4.8) both describe the accuracy of the NN emulation integrated over the entire 4-D data set. However, because of a different order of integration, it reveals different and complementary information about the accuracy of the NN emulations. The root mean square error profile can be also calculated:

$$rmsep(j) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[Y\,(i,j) - Y_{\mathrm{NN}}\,(i,j)]^2}, \quad j = 1,\ldots,L. \tag{4.12}$$

The principal, direct benefit of the NN emulation is a significantly enhanced computational performance (speed up) as compared with the current original parameterization. This speedup can be used in several different ways: (1) to speed up the model runs, (2) to calculate parameterizations of model physics more often (e.g., at each integration time step), and (3) to improve model physics by introducing in the model more sophisticated and realistic parameterizations of physics that currently cannot be used because of their computational cost; however, after emulating with NNs, they become computationally affordable.

Compound and Adjustable Parameterizations, Quality Control of NN Emulations

The accuracy of NN emulations of model physics depends significantly on our ability to generate a representative training set and to avoid using NNs for extrapolation beyond the domain covered by the training set. Because of high dimensionality of the input domain (i.e., dimensionality of the NN input vector X), which is of the order of several hundred or more, it is difficult if not impossible to cover the entire domain, which may have a very complex shape, even when we use model-simulated data for the NN training. Also, the domain may change with the evolution of the system during a simulation period. In such situations, the emulating NN may be forced to extrapolate beyond its generalization ability, which may lead to larger errors in NN outputs and correspondingly in the numerical model simulations in which NN emulations are used.

For example, the developed NN emulations of model radiation are very accurate. Larger errors and outliers (a few extreme errors) in NN emulation outputs occur only when NN emulations are exposed to inputs not represented sufficiently in the training set. These errors have a very low probability (see Fig. 4.15) and are distributed randomly in space and time. However, when long multi-decadal climate simulations are performed, and NN emulations are used in a very complex and essentially nonlinear climate model over long integration times, the probability of larger errors and their undesirable impact on the model simulations increase. As we learned from our experiments with GCMs (e.g., Fig. 4.16 in Sect. 4.3.5), the model was robust enough to overcome such randomly distributed errors without their accumulation over time. However, for these few cases of larger errors, it is still essential to develop an internal QC procedure capable of controlling these infrequent larger errors (Krasnopolsky et al. 2008a).

In another application of an NN approximation for nonlinear interactions in a wave model, the model did not prove sufficiently robust to retain stability for time integrations of even a few hours (e.g., Fig. 4.23 in Sect. 4.4.2). Thus, in this case, introducing an internal QC method for identifying and controlling larger NN emulation errors is particularly important for the successful application of NN emulation of the model physics (Tolman and Krasnopolsky 2004).

Therefore, for many applications, it is essential to introduce a QC procedure, which can predict and eliminate larger errors from NN emulations during the integration of highly nonlinear numerical models, and not just relying on the robustness of the model that can vary significantly for different models. In Sects. 4.3.5 and 4.4.2, the concept of a compound parameterization (CP) that incorporates an emulating NN, an original parameterization, and a QC procedure is introduced. CP makes the NN emulation approach even more reliable, robust, and generic. It also provides a tool for developing NN emulations adjustable to changes in the model environment and in the climate system.

Using NNs for Developing New NN-Based Parameterizations

The NN technique can be used either to emulate or to improve model physics. It was mentioned that because of the simplified parameterized physics that GCMs use, they cannot simulate accurately many important fine-scale processes like cloudiness and convective precipitations (e.g., Rasch et al. 2000). In section "Cloud-resolving models," it was mentioned that CRMs resolve many of the phenomena that lower-resolution global and regional models do not resolve (e.g., higher-resolution fluid dynamic motions supporting updrafts and downdrafts, convective organization, mesoscale circulations, and stratiform and convective components that interact with each other). NN techniques can be used to build a bridge or interface between CRMs and GCMs, for example, to develop an NN convection parameterization, which can be used as a parameterization in GCMs and can effectively account for major subgrid scale effects taken into account by other approaches (like MMF; see subsection "Multiscale modeling framework or superparameterization" of Sect. 4.1.1). The idea is to develop NNs which emulate the behavior of a CRM and can be run at larger scales (closer to GCM scales) in a variety of regimes and initial conditions. The resulting emulation can be used as a novel and computationally viable parameterization in a GCM. It may produce a parameterization of similar or better quality compared to the superparameterization or the MMF, effectively taking into account subgrid scale (in terms of a GCM) effects at a fraction of the computational cost (Krasnopolsky et al. 2011).

Figure 4.2 summarizes the process of development of such an NN parameterization. The CRM uses data, Tropical Ocean Global Atmosphere Coupled Ocean-atmosphere Response Experiment (TOGA-COARE), Atmospheric Radiation Measurement, or other observations, for initialization and forcing and has the horizontal resolution of about 1 km, 64 or 96 vertical layers, and a time step of 5 s.

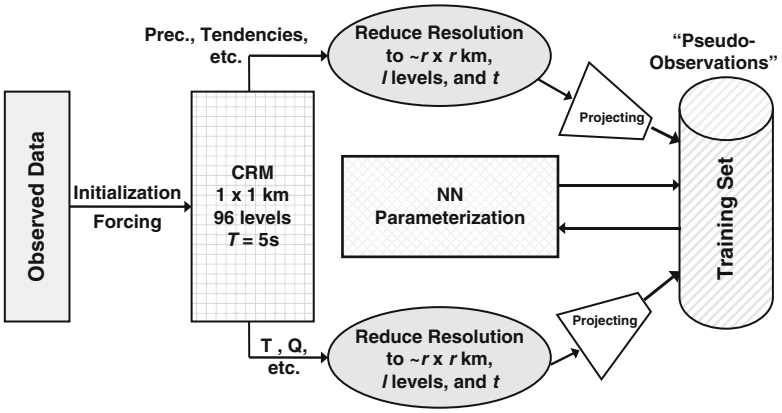

Fig. 4.2 The process of development of an NN convection parameterization

The CRM is usually integrated over a domain of 256×256 km. The development of an NN parameterization is a multistep process. These steps are (Krasnopolsky et al. 2013) the following:

1. *Simulating CRM data*. The model is run for some time, which is limited by the data available for initialization and forcing, and the high-resolution output of the model is archived.
2. *Reducing the resolution of simulated data*. The high-resolution CRM-simulated data are averaged in space and time. The data are averaged to a reduced horizontal resolution of $\rho < r \leq R$, where ρ and R are the CRM and GCM resolutions correspondingly, and are interpolated/averaged to the number of vertical layers $l = L$, where L is the number of vertical layers in the GCM.
3. *Projecting a CRM vector of atmospheric states onto a GCM vector of atmospheric states*. From the reduced resolution CRM-simulated data created at the previous steps, the subset of variables is selected and this subset constitutes the NN development set. Only variables that can be identified with corresponding GCM variables or can be calculated from or converted to prognostic or diagnostic variables available in the GCM are included in the development set (called "pseudo-observations" in Fig. 4.2). Only these variables are used as inputs and outputs in the NN parameterization. The choice of proper "inputs" and "outputs" for an NN convection parameterization is very important. For example, a simple convective parameterization might define "temperature," "water vapor," and the convergence of temperature and water vapor to be "inputs" and produce Q1C and Q2, the apparent heat and moisture tendencies as the "outputs." The outputs Q1C and Q2 clearly depend upon other variables (e.g., the condensed water in each CRM column) that are not necessarily considered to be part of either the inputs or outputs of the NN. These variables cannot be included as NN inputs and/or outputs simply because they are not available in the GCM. From the

point of view of GCM "model reality," these variables are "hidden" variables responsible for subgrid scale variability. The acknowledgement of this challenge requires the introduction of the concept of uncertainty and "stochasticity." The development set of pseudo-observations implicitly represents a stochastic convection parameterization (i.e., a stochastic mapping) with an uncertainty, which is an inherent feature of such a parameterization (see Sect. 4.3.6).

4. *Adjusting the differences*. The pseudo-observations that are used for the development of the NN parameterization are not real observations. They represent the virtual reality of the averaged CRM-simulated data. We use them for the development of the parameterization that is further introduced in CAM. CAM has its own virtual reality, which may not be in complete agreement with the averaged CRM-simulated data, and therefore, with the NN parameterization trained on pseudo-observations derived from the averaged CRM-simulated data. Thus, special effort may be required to synchronize or make consistent the virtual realities of CAM and the averaged CRM-simulated data. CRM vs. GCM mean differences for all variables selected as the NN parameterization inputs and outputs have to be determined and compensated for (Krasnopolsky et al. 2011). These differences are a result of CAM and CRM being two different models with different temporal and spatial scales and resolutions, with different dynamics and physics; they also have different boundary and initial conditions and different forcing.
5. *Creating data sets*. The development set of pseudo-observations is separated into the independent training and test/validation sets. Then the NN parameterization is trained using the training set. Due to the inherent uncertainty of pseudo-observations, the NN parameterization derived from these data is a stochastic parameterization and is implemented as an ensemble of NNs (see Sect. 4.3.6).

The validation procedure for the NN parameterization consists of two steps. First, the trained NN ensemble stochastic parameterization is applied to the test set and error statistics are calculated. Second, the tested NN parameterization is introduced into the GCM or in a single-column GCM to validate its behavior in the model simulations. This last step is the most important step of the validation process.

NN Emulation for Superparameterizations

The concept of superparameterization and its particular implementation was introduced in subsection "Multiscale modeling framework or superparameterization" of Sect. 4.1.1. As can be seen from this description, the superparameterization is similar conceptually to a regular parameterization of model physics. At each time step, the superparameterization (the embedded CRM) receives a vector of input parameters X, which describes a state of the atmosphere in this column in terms of the GCM variables. After integration of the CRM in the column of the GCM, it returns back to the GCM a vector of output parameters Y, which describes the physical forcing for this column in terms/variables of the GCM. As a result, the entire

superparameterization, from a mathematical point of view, can be considered as a mapping. Taking into account the physical and mathematical properties of the CRM, this mapping is continuous or almost continuous (may contain finite discontinuities like step functions) and can be emulated by an NN with a specified accuracy.

However, as in the previous section, which considered developing a new NN parameterization based on CRM-simulated data, in the case of emulating superparameterization, a stochasticity (random behavior) emerges in the problem because of unaccounted variability related to the initial conditions, ξ, that are remembered by the embedded CRM between the CAM time steps. The initial conditions are hidden from the CAM environment. Thus, the SP should be considered as a stochastic mapping (2.1a), $Y = F(X, \xi)$. As a result, uncertainty will emerge in the simulated data, which cannot be accounted for by a single emulating NN. Thus, an ensemble of NNs is better suited for emulating the superparameterization than a single emulating NN.

The principal direct benefit of the NN superparameterization would be significantly enhanced computational performance over the current MMF. We can imagine using a faster model in many ways: (1) Increase the resolution of the CRM at 2-D or perhaps go to a 3-D CRM in the MMF. This is computationally intensive but is doable if we only need to create a training set. (2) Increase the vertical resolution of the MMF (both outer grid and CRM), which would have a significant positive impact on model performance. (3) Investigate the performance of the MMF as a global climate model. This includes an entire range of potential experiments, such as adding an ocean model and/or running the model over a range of increased greenhouse gas concentrations.

4.3 Atmospheric NN Applications

In this section, several complex weather- and climate-related NN applications are introduced to illustrate the approaches presented in the previous sections of Chap. 4. Four different GCMs are used in these applications:

- The NCAR CAM (see *J. Climate* 1998, for a detailed description of this model) is a spectral model that has 42 spectral components (or approximately 3°–3.5° horizontal resolution) and 26 vertical levels (T42L26). The results presented below were obtained using earlier versions of NCAR CAM, CAM-2, and CAM-3. The atmospheric component of the model has been run using the climatological SST.
- The NASA NSIPP (Natural Seasonal-to-Interannual Predictability Program) model is a grid-point GCM that has $2° \times 2.5°$ latitude $\times$ longitude horizontal resolution and 40 vertical levels.
- The NCEP Climate Forecast System (CFS) (Saha et al. 2010), a state-of-the-art GCMs, is used for climate predictions. The atmospheric component of the CFS version used here has 126 spectral components ($\sim 1°$ horizontal resolution) and

64 vertical levels (T126L64). In CFS, the atmospheric model is coupled with the 40-level interactive MOM4 ocean model, the interactive Noah land model with four soil levels with improved treatment of snow and frozen soil, an interactive sea ice model with fractional ice cover and depth permitting a subgrid scale mountain blocking, a new seasonal climatological aerosol treatment, a historical CO_2 database from global observations collected by the World Meteorological Organization, a variable solar constant database, and historical stratospheric volcanic aerosol distributions are also included.

- The NCEP Global Forecast System (GFS) is identical to the atmospheric component of the CFS. However, the GFS has significantly higher spectral resolution with 574 spectral components ($\sim$0.2° horizontal resolution) and 64 vertical levels (T574L64) and is used for the NWP.

After applying the hybridization approach to the first-principle-based components of these models by developing NN *emulations* for existing parameterizations of model physics or new NN-based parameterizations, these models become HGCMs.

4.3.1 NN Emulation of Model Physics Components

Here we discuss the development of NN emulations for the atmospheric model physics components, using model radiation, both LWR and SWR parameterizations, as examples. A similar approach can be applied to other parts of the atmospheric physics.

LWR and SWR together comprise the total atmospheric radiation. The function of the LWR and SWR parameterizations in atmospheric GCMs is to calculate the heat fluxes and heating rates produced by LWR and SWR processes. As mentioned, the entire LWR or SWR parameterizations can be represented as almost continuous mappings (Eq. 4.5). In this section, we describe applications of the NN emulation approach to three LWR and two SWR parameterizations in four different models (NCAR CAM, NASA NSIPP, and NCEP CFS and GFS) – all models that have been described in the previous sections.

An outline of the internal structure of the *CAMRT* LWR parameterization, which was used in NCAR CAM (see Eqs. 4.3 and 4.4), is given in Sect. 4.1.3. A detailed description of *CAMRT* LWR and SWR parameterizations can be found in (Collins 2001; Collins et al. 2002). The input vectors for the NCAR CAM LWR parameterization include ten vertical profiles of atmospheric temperature, humidity, concentrations of ozone, CO_2, N_2O, CH_4, two CFC mixing ratios, the annual mean atmospheric mole fractions for halocarbons, pressure, and cloudiness, and the upward LWR flux at the surface, for a total of 220 inputs. Thirty-three outputs include a profile of the heating rates (26 outputs) and seven heat fluxes.

The input vectors for the NCAR CAM *CAMRT* SWR parameterization include 21 vertical profiles (specific humidity, ozone concentration, pressure, cloudiness,

aerosol mass mixing ratios, etc.) and several surface variables, for a total of 173 and 451 inputs for CAM-2 and CAM-3, respectively. The SWR parameterization has 33 outputs. The major difference between the CAM-2 and CAM-3 SWR versions is that CAM-3 uses significantly more information about aerosols. This extended aerosol information is responsible for a substantial increase in the number of inputs into the CAM-3 SWR parameterization compared with CAM-2.

The NASA NSIPP model uses the LWR parameterizations developed by Chou et al. (2001). The input vector for the NSIPP LWR contains surface temperature and five vertical profiles that include cloud fraction, pressure, temperature, specific humidity, and ozone mixing rate, for a total of 202 inputs and 41 outputs.

The NCEP CFS and GFS contain a GCM version (v2.3) of the rapid radiative transfer model (RRTM) for LWR (hereafter referred to as RRTMG-LW) (e.g., Mlawer et al. 1997; Iacono et al. 2000). For the CFS and GFS RRTM LWR, the input parameters include nine profiles: atmospheric pressure, temperature, specific humidity, ozone mixing ratio, total cloud fraction, cloud liquid water path, mean effective radius for liquid cloud, cloud ice water path, and mean effective radius for ice cloud, for a total of 598 inputs. RRTMG-LW has 69 outputs. CO_2 is time dependent, and so its global mean is specified as a function of time.

The SWR parameterization used in the CFS and GFS is a modified version of RRTMG-SW (v2.3) (Clough et al. 2005). RRTMG-SW uses a fast two-stream radiative transfer scheme and supports detailed specification of absorption and scattering processes by clouds, aerosols, and absorbing gases (H_2O, O_3, CO_2, CH_4, N_2O, O_2). Thus, in the current version of the SWR parameterization, the level of atmospheric CO_2 concentration and its time dependence is presented by the entire 3-D CO_2 field that changes with time in accordance with the change of the mean CO_2 level that increased from 350 to 380 ppmv during the period of model integration presented here (1990–2006). The NN emulations of the SWR parameterization have for a total of 562 inputs and 73 outputs.

It is noteworthy that in the case of the NN emulation, the number of NN inputs is less than the number of inputs of the original parameterization, which is the number of input profiles multiplied by the number of vertical layers plus the number of relevant single-level variables. Many input variables (e.g., almost all gases) have zero or constant values for the upper vertical layers, and for some gases, the entire volume mixing ratio profile is a constant (obtained from climatological data). To improve the accuracy of the approximation, these constant inputs were not used for NN training. Constant inputs (zero or nonzero) do not contribute to the functional input/output relationship and should not be used for developing NN emulations (see Sect. 2.3.6). Moreover, if they were used, they would introduce an additional source of noise (an approximation error).

In addition, for SWR, 2,688 inputs describing the optical depth, the single scattering albedo, and the asymmetry parameters for 14 aerosol species were substituted by five inputs: $\cos(\tau)$, $\sin(\tau)$, $\cos(lon)$, $\sin(lon)$, and lat, where lon is the longitude, lat is the latitude, and $\tau = 2 \cdot \pi / T \cdot \mu$, where μ is the month of the year and $T = 12$. Such a substitution is possible because in NCEP CFS and GFS aerosols are calculated using the specific humidity profiles and 3-D lookup

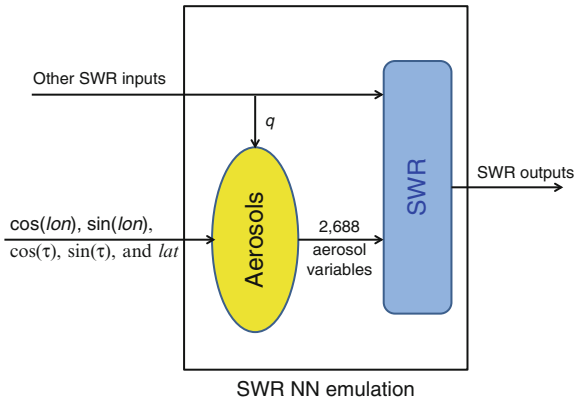

Fig. 4.3 The NN emulation for NCEP SWR parameterization. The NN emulates both the SWR parameterization and the aerosol model in this case

tables composed of global climatological monthly data, which are different for different months of the year. Thus, the aerosol inputs to the SWR parameterization are actually highly correlated, and, in terms of functional input/output dependences, the aerosol characteristics are functions of *lat*, *lon*, τ, and the profile of specific humidity, q, only. Since the profile of the specific humidity is already included in NN SWR inputs, only the five aforementioned additional variables have to be included to allow the NN to completely emulate the contribution of aerosols into SWR. Actually, our SWR NN emulates both the aerosol model and the SWR parameterization in this case, as shown in Fig. 4.3.

Although both RRTMG-LW and RRTMG-SW are built with fast computation schemes designed for GCM applications, they still represent the most time-consuming physics in the NCEP CFS and GFS models. The percentage of the total model computation time used by model physics and radiation (LWR and SWR) varies depending largely on the horizontal and vertical resolutions, the time step, the frequency of radiative calculations, and the computing environment (e.g., the number of processors and threads). For example, in a CFS configuration at the T126L64 resolution, with the RRTMG-LW and RRTMG-SW both called every hour, the portion of the radiation computation time is about 57 % of the total atmospheric GCM model computation time.

For all of these radiation schemes, the LWR and SWR output vectors consist of the vertical profiles of heating rates (HRs) and radiation fluxes, including the outgoing LWR (or OLR) flux from the top layer of the model atmosphere.

4.3.2 Generating the Training Set

The entire coupled model (for NCEP CFS) or the atmospheric component of the model (for NCAR CAM and NCEP GFS) was run for several years. The NCEP CFS model was run continuously for 10 years and the NCAR CAM for two years.

All inputs and outputs of the original LWR and SWR parameterization have been saved over the globe for 2 days per month, i.e., for 1 day at the beginning and 1 day in the middle of the month, every 3 h (eight times per day) to cover the annual and diurnal cycles, which constituted 1,920 global data sets for NCEP CFS and 384 global data sets for NCAR CAM. For the NCEP GFS NWP model, during 1 year, a 10-day forecast was run each first and fifteenth day of the month. All inputs and outputs of the original LWR and SWR parameterizations have been saved over the globe eight times per day during each day of the 10-day forecast. Thus, 1,920 global data sets have been generated for NCEP GFS. About 300 data records for NCEP CFS and GFS and 1,500 data records for NCAR CAM have been randomly selected from each global data set. Each record consists of a combination of the radiation inputs and outputs at a particular horizontal location for all vertical levels. A total collection of about 600,000 radiation inputs and outputs was divided into three independent parts, each containing about 200,000 input/output vector combinations (records). The first part was used for training, the second for validation (control of overfitting and control of NN architecture), and the third part for testing the approximation quality. All approximation statistics presented in this chapter were calculated using the independent test data set. The accuracy of the NN emulation, i.e., mean errors (or biases) and *rmse*, is calculated against the control (the original parameterization).

4.3.3 NN Emulations for the Model Radiation

In Sect. 4.3.1, the selection of inputs and outputs of NN emulations for model LWR and SWR parameterizations was discussed. Here the selection of the number of hidden neurons for the emulating NNs is discussed. Also the bulk approximation statistics for different models are compared, and the NN emulation performances vs. the original parameterization are evaluated.

Selecting the Number of Hidden Neurons

The NN emulation of the NCAR CAM LWR parameterization has the same number of inputs (220 total) and outputs (33 total) as the original NCAR CAM LWR parameterization. Several emulating NNs have been developed that all have one hidden layer with 20–300 neurons. Varying the number of hidden neurons, k, allows us to demonstrate the dependence of the accuracy of the NN emulation on this parameter, which is actually a measure of the complexity of the NN emulation (2.4), as well as to select an "optimal" NN emulation with the minimal complexity (see Sect. 2.5). This is an emulation that has the accuracy sufficient for successful multi-decadal climate model integration.

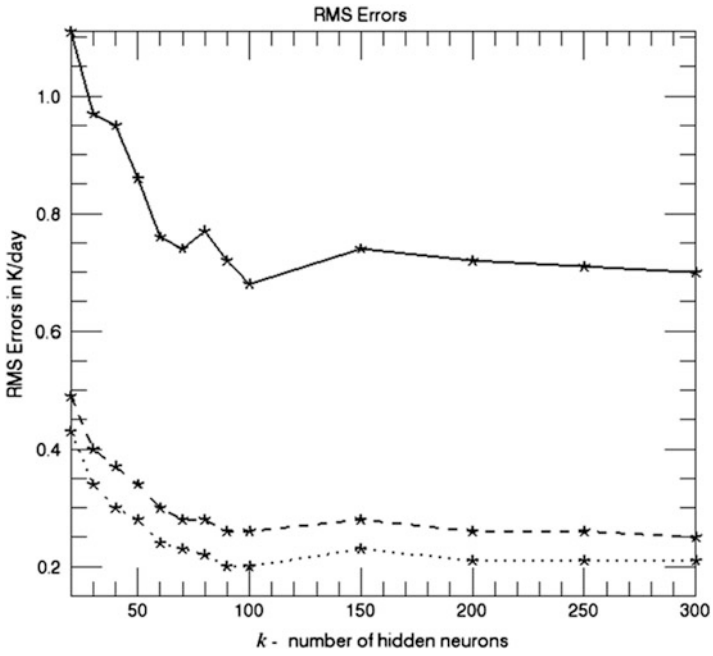

Fig. 4.4 The convergence of root mean square errors (4.8, 4.9, and 4.11) for NNs with different number of hidden neurons, k, emulating the NCAR CAM LWR parameterization. *Solid line*, $RMSE_m$ (4.9) for $m = 26$; *dashed line*, *RMSE* (4.8); and *dotted line*, *PRMSE* (4.11)

All NN emulations developed for the NCAR CAM LWR have negligible systematic errors (biases). Figure 4.4 illustrates convergences of root mean square errors (4.8, 4.9, and 4.11) calculated using an independent test data set. These errors are random errors in the case of negligible biases. The figure shows that an error convergence has been reached when the number of hidden neurons $k \approx 100$. However, the convergence becomes slow and non-monotonic at $k \approx 50$.

The final decision about the optimal NN emulation (in terms of sufficient accuracy and minimal complexity) to be implemented into the model is based on decadal (50 year) integrations using the NN emulations within HGCM (see Sect. 4.3.4) (Krasnopolsky et al. 2005a; Krasnopolsky and Fox-Rabinovitz 2006a, b). The NN emulation with $k = 50$ is the simplest NN emulation that could be integrated into the model for decadal (50 years or longer) climate simulations without any visible (significant) accumulations of errors in the climate simulations, compared to the control run using the original LWR parameterization. This is the primary indicator that the accuracy of the NN emulation is sufficient for this application.

Figure 4.5 shows the vertical error profile (4.12) $prmse(j)$ (right panel, solid line) for the "optimal" NN emulation of NCAR CAM LWR with 50 hidden neurons ($k = 50$). It shows that the errors are very small; at the top 10 levels, the error does

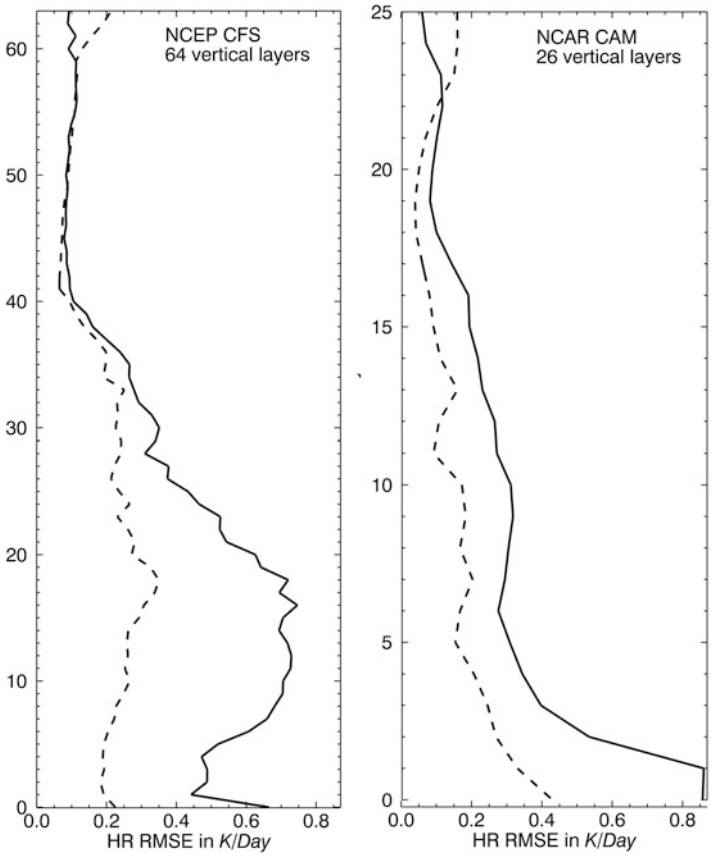

Fig. 4.5 Vertical distributions of NN emulation errors *rmsep* (4.12) for two models: the *right panel*, NCAR CAM (26 vertical layers), and the *left panel*, NCEP CFS (64 vertical layers). *Solid line* corresponds to LWR and *dashed line* to SWR. The obvious difference between NCAR CAM and NCEP CFS errors at the lower levels is due to the fact that the majority of additional (as compared to NCAR CAM) vertical levels in NCEP CFS are introduced in and near the surface boundary layer; thus, the errors in the *lower* part of the *left panel* are significantly magnified in the *right panel*. Actually, in terms of the height, the errors and their vertical distributions are similar for both models (Krasnopolsky et al. 2010 (©) American Meteorological Society. Used with permission)

not exceed 0.2 K/day; at the top 20 levels, it does not exceed 0.3 K/day and reaches 0.6–0.8 K/day at the lowest level, which does not lead to significant errors in the 50-year climate simulations with HGCM. In addition to having sufficient accuracy, the NN emulation performs about 150 times faster than the original NCAR CAM LWR parameterization in a code-by-code comparison.

The same approach has been used to select the optimal number of hidden neurons, k, for all other emulations presented here. Thus, $k = 55$ was selected

Table 4.1 Statistics for estimating the accuracy of the HR calculations (in K/day) and the computational performance of the NN LWR emulation vs. the original parameterization for NCAR CAM (T42L26), NSIPP model (40 vertical levels), and NCEP CFS (T126L64) LWR

				NCEP CFS (L = 64)		
	Statistics	NCAR CAM (L = 26)	NASA NSIPP (L = 40)	RRTMG	Change due to balancing	RRTMF
Total error statistics	*Bias*	$3.\cdot 10^{-4}$	$3.\cdot 10^{-4}$	$2.\cdot 10^{-3}$	$6.\cdot 10^{-4}$	$7.\cdot 10^{-4}$
	RMSE	0.34	0.22	0.49	$1.\cdot 10^{-4}$	0.42
	PRMSE	0.28	0.14	0.39	$3.\cdot 10^{-4}$	0.30
	σ_{PRMSE}	0.2	0.16	0.31	$1.\cdot 10^{-4}$	0.30
Bottom layer error statistics	*Bias*	$-2.\cdot 10^{-3}$	$3.\cdot 10^{-3}$	$-1.\cdot 10^{-2}$	$-6.\cdot 10^{-4}$	$6.\cdot 10^{-3}$
	RMSE	0.86	0.41	0.64	$1.\cdot 10^{-5}$	0.67
Top layer error statistics	*Bias*	$-1.\cdot 10^{-3}$	$-5.\cdot 10^{-3}$	$-9.\cdot 10^{-3}$	$6.\cdot 10^{-4}$	$2.\cdot 10^{-3}$
	RMSE	0.06	0.1	0.1782	$4.\cdot 10^{-3}$	0.09
NN complexity	n_C	490	397	520	–	1,468
Speedup, η	*Times*	150	–	16 (20)	–	21

Total statistics show the bias, *RMSE* (4.8), *PRMSE*, and σ_{PRMSE} (4.10) for the entire 3-D HR fields. Layer (for the *top* and *bottom* layers) statistics show the bias and *RMSE* (4.9) for one *horizontal* layer (the *top* or *bottom* layer). Also, the changes in statistics due to the balancing procedure (see section "Balancing LWR and SWR heating rate") are shown for RRTWG (RRTMG and RRTMF are different versions of the radiation code developed by AER Inc. (see Sect. 4.3.1 and references there)) LWR and SWR NN emulations. The NN complexity n_C (2.4a) and average speedup η are also shown. (Here η shows an averaged (over a global data set) speedup or how many times NN emulation is faster than the original parameterization in a single processor code-by-code comparison; the number in parentheses shows the speedup in multiprocessor environment.)

for the NN emulation of NCAR CAM SWR parameterization and $k = 75$ for both NCEP LWR and SWR parameterizations to obtain comparable accuracies of approximation.

Bulk Approximation Statistics

The NN emulations have been validated against the original LWR and SWR parameterizations. To calculate the error statistics presented in Tables 4.1 and 4.2 and in Fig. 4.5, the original parameterizations and their NN emulations have been applied to the validation data set. Two sets of corresponding HR profiles have been generated for both LWR and SWR. Total and level mean differences (biases or mean errors), *RMSEs* (4.8 and 4.9), profile *RMSE* or *PRMSE* (4.11), and *rmsep* (4.12) have been calculated.

As can be concluded from Tables 4.1 and 4.2 and from Fig. 4.5, NN emulations for both LWR and SWR successfully handle the nonlinearity at the top of the atmosphere where mean differences and *RMSEs* are very small with *RMSEs* being even smaller than the total *RMSE*. In the bottom layer, the nonlinearity

Table 4.2 Statistics for estimating the accuracy of the HR calculations (in K/day) and the computational performance of the NN SWR emulation vs. the original parameterization for NCAR CAM (T42L26) and NCEP CFS (T126L64)

			NCEP CFS ($L = 64$)	
	Statistics	NCAR CAM ($L = 26$)	RRTMG	Change due to balancing
Total error statistics	*Bias*	$-4. \cdot 10^{-3}$	$5. \cdot 10^{-3}$	$-3 \cdot 10^{-3}$
	RMSE	0.19	0.20	$-5. \cdot 10^{-3}$
	PRMSE	0.15	0.16	$-5. \cdot 10^{-3}$
	σ_{PRMSE}	0.12	0.12	$1. \cdot 10^{-3}$
Bottom layer error statistics	*Bias*	$-5. \cdot 10^{-3}$	$9. \cdot 10^{-3}$	$-8. \cdot 10^{-3}$
	RMSE	0.43	0.22	−0.01
Top layer error statistics	*Bias*	$2. \cdot 10^{-3}$	$1.3 \cdot 10^{-2}$	$4. \cdot 10^{-3}$
	RMSE	0.17	0.21	$1. \cdot 10^{-3}$
NN complexity	n_C	439	706	–
Speedup, η	*Times*	20	60 (88)	–

Total statistics show the bias, *RMSE* (4.8), *PRMSE*, and σ_{PRMSE} (4.10) for the entire 3-D HR fields. Layer (for the *top* and *bottom* layers) statistics show the bias and *RMSE* (4.9) for one horizontal layer (the *top* or *bottom* layer). Also, the changes in statistics due to the balancing procedure (see section "Balancing LWR and SWR heating rate") are shown for RRTWG SWR NN emulations. The NN complexity n_C (2.4a) and average speedup η are also shown

does not cause significant increases in mean differences; the *RMSEs* increase by approximately a factor of two but they remain sufficiently small.

It is noteworthy that the approximation errors are identified as being "sufficiently small" if they are of such a small magnitude that they have almost negligible impacts on model behavior as demonstrated below and by Krasnopolsky et al. (2008a, 2010) for NCAR CAM and NCEP CFS and GFS. Only validations of NN emulations in parallel model runs allow us to finally conclude that the approximation errors are sufficiently small.

In terms of the accuracy statistics presented, there are practically no differences between NCAR CAM with 26 vertical layers, NASA NSIPP with 40 vertical layers, and NCEP CFS with 64 vertical layers. This fact illustrates the robustness of the NN emulation approach with respect to the changes in the model vertical resolution. As shown in Fig. 4.5, the entire vertical distributions of errors (for both LWR and SWR) are similar for NCEP CFS and NCAR CAM. Thus, the accuracy of our NN emulation approach does not depend significantly on vertical resolution of the model. However, it does depend on the vertical location of the atmospheric layer. Finally, we note that the layer *RMSE* increases near the surface for all models.

Estimation of Speedup

The NN complexity n_C (2.4a) and average speedup η (how many times NN emulations are faster than the original parameterization) are also shown in Tables 4.1

and 4.2. The NN complexity per output, n_C, is used because NNs with different number of outputs are compared here. In this case, n_C provides a more adequate metric for comparisons. For the LWR parameterization, we see a significant decrease in the speedup for NCEP CFS with 64 vertical layers vs. NCAR CAM with 26 vertical layers, although the LWR NN emulation for NCEP CFS is still 16 times faster than the original parameterization. For the SWR parameterization, the opposite tendency is observed; that is, the NCEP CFS SWR NN is more than three times faster than the NCAR CAM SWR NN.

These seemingly contradictory speedups for LWR and SWR emulations can be explained (for detailed discussion of this topic, see Krasnopolsky et al. 2010) by the interplay of the two main contributing factors: the physical and mathematical complexities (see Sect. 2.2.2) of the radiation calculation itself (the number of treated species, spectral bands, parameterization schemes, etc.) and the dependence of the particular numerical scheme implemented in the radiative transfer on the number of vertical layers in the model. The results presented in Tables 4.1 and 4.2 illustrate the fact that the numerical scheme implemented in the NCEP CFS RRTMG-LW parameterization is significantly more efficient (linear with respect to the number of vertical levels L) than that of the original NCAR CAM LWR parameterization (quadratic with respect to L). Thus, a smaller speedup factor is produced by the NN emulation for NCEP CFS LWR than that of NCAR CAM LWR.

The NCEP CFS's RRTMG-SW includes more spectral bands and uses more complex treatment for a larger variety of absorbing/scattering species; thus, NN emulation shows a larger speedup value η than that of NCAR CAM SWR. In any case, the NN emulation approach is significantly less dependent (in terms of both the accuracy and speedup) on the increase of vertical resolution than the NN-based hybrid LWR parameterization NeuroFlux (Sect. 4.2.1), where with a vertical resolution of 60 layers and more, both accuracy and speedup could not be achieved simultaneously (Morcrette et al. 2008). For the NN emulation approach, for the model with 64 vertical layers, the desired accuracy of the NN emulation could be achieved simultaneously with a significant speedup of $\sim$16 times for the LWR and of $\sim$60 times for the SWR parameterizations.

The radiative transfer calculations take different times under different cloud conditions because of the varying complexity of cloud-radiation interaction. More detailed estimations of speedup have been separately performed for three different types of cloudiness: clear sky, three cloud layers, and a more complex cloud condition where deep convection occurs. Three thousand profiles have been used for each test. The results for the calculation time and speedup are presented in Table 4.3. For a more complex cloud-radiation interaction (deep convection), the calculation of the original LWR and SWR parameterizations takes $\sim$22 and $\sim$57 % more time, respectively, than for clear sky conditions. Obviously, the time required for the NN radiation calculations does not depend on the cloud conditions. Thus, the speedup is significantly higher for more complex cloud-radiation interactions.

Table 4.3 Comparison of calculation time and speedups η for LWR and SWR RRTMG original parameterizations and NN emulations (time is shown per 3,000 profiles) under different cloud conditions

Parameterization	LWR RRTMG			SWR RRTMG		
Type of cloudiness	Clear Sky	3-layer clouds	Deep convection	Clear sky	3-layer clouds	Deep convection
Original parameterization (time in sec)	9.6	10.1	11.7	33.8	42.8	52.9
NN (time in sec)	0.6	0.6	0.6	0.6	0.6	0.6
η (times)	16	16.8	19.5	56	71	88

The calculations were performed using a single processor of an IBM Power 6 supercomputer

As Table 4.3 shows, the average speedup presented in Tables 4.1 and 4.2 is closer to the minimal speedup obtained under clear sky conditions. The results presented in Tables 4.1, 4.2, and 4.3 were obtained in a code-by-code comparison and represent adequately the situation when the model is run on a single processor. However, if we compare the control model run using the original parameterizations with the NN run when both runs use multiple processors and threads, the actual speedup will be significantly higher and closer to the maximum value shown in Table 4.3 because it will be determined by the slowest calculation, which is the deep convection condition. Radiation in the control run for all other cloud conditions is calculated faster, but the next integration time step will not start before the radiation calculations for the deep convection condition are completed; at the same time, the time for the radiation calculations in the NN run does not depend on the cloud conditions. Thus, in the case of parallel calculations utilizing multiple processors and threads, in addition to a significant speedup, the use of the NN emulations in the model provides an additional advantage, namely, it helps to achieve a significantly better load balance.

Using NN emulations simultaneously for LWR and SWR or for the full model radiation, results in an overall speedup of about 20–25 % for the NCEP CFS climate simulations and seasonal predictions when both LWR and SWR are calculated every hour. The speedup η provided by NN emulations (see Tables 4.1 and 4.2) can be also used for more frequent calculations of model radiation.

Balancing LWR and SWR Heating Rates

For both LWR and SWR parameterizations, the parameterization output vectors consist of the vertical profile of cooling or heating rates (HRs) and several radiation fluxes at the top and the bottom of the atmosphere. There exists an integral relationship between pressure, heating rates, and fluxes. For example, this relationship for the imbalance, ε, can be written as

$$\varepsilon = \frac{\sum_{k=1}^{L} \alpha_k \cdot h_k}{\sum_{k=1}^{L} \alpha_k} + \frac{\Phi}{\sum_{k=1}^{L} \alpha_k} = 0.$$

$$\Phi = \begin{cases} F_{\text{tup}} - F_{\text{sup}} + F_{\text{sdn}} & \text{for LWR} \\ F_{\text{tup}} - F_{\text{tdn}} - F_{\text{sup}} + F_{\text{sdn}} & \text{for SWR} \end{cases} \tag{4.13}$$

where $\alpha_k = (p_k - p_{k-1})/G$, p_k and h_k are pressure and heating rate at a vertical level k, G is a constant, F_{tup} is the total sky outgoing LWR or SWR flux at the top of the atmosphere, F_{tdn} is the total sky downward SWR flux at the top of the atmosphere, F_{sup} is the total sky upward LWR or SWR flux at the surface, and F_{sdn} is the total sky downward LWR or SWR flux at the surface.

The outputs of the original radiation parameterizations satisfy the relationship (4.13) with high accuracy because these relationships are explicitly (or implicitly) included into the parameterizations. Obviously, the outputs of the NN emulations will satisfy (4.13) only approximately, i.e., in this case, the imbalance $\varepsilon \neq 0$; however, ε is small. For example, for the RRTMG LWR NN emulation presented in Table 4.1, the mean value for ε is $6.5 \cdot 10^{-4}$ K/day. A correction can be introduced for the heating rates, given by $\tilde{h}_k = h_k + \varepsilon$. The correction makes the balanced heating rates $\tilde{h}_k$ satisfy the relationship (4.13). This correction is very small and, as the results presented in Tables 4.1 and 4.2 show, this balancing procedure does not significantly affect the overall accuracy of LWR NN and marginally improves the overall accuracy of SWR NN. However, it is noteworthy that the heating rates produced by the NN emulation can be easily balanced by using (4.13).

4.3.4 *Validation of NN Emulations in Parallel Decadal Climate Simulations and Weather Forecasts*

In this section, the ultimate validation of NN emulations – comparisons between two parallel runs of the model – is presented. One run, which is called "control run," uses the original LWR and SWR parameterizations and the other, the NN run, uses their NN emulations. Both spatial and temporal characteristics of prognostic and diagnostic fields are compared for the parallel runs. Such validations have been performed using decadal climate simulation runs for NCAR CAM and NCEP CFS and 8-day forecasts runs for NCEP GFS weather forecast. Only the validation of NN emulations in the model can ultimately determine if the approximation accuracy evaluated in the previous sections is sufficient for NN emulations to be used in the model in place of the original parameterizations.

Measures of Impact of Using NN Emulations for the Full Model Radiation

To evaluate the NN-induced changes, the differences between the runs have been compared with such commonly used measures as observation errors or uncertainties of the reanalysis. We show that the differences are smaller than these conventional measures.

Intuition suggests that the differences might be made even smaller if the accuracy of the NN emulations were better (the approximation errors presented above were smaller) and, in the limit, the differences would run to zero with increasing NN emulation accuracy. However, in this case, as in many cases when we deal with complex nonlinear systems, our intuition does not provide us with reliable guidance. Because a GCM (NCAR CAM or NCEP CFS or GFS) is an essentially nonlinear system, it may produce something akin to the "butterfly effect," that is a significant reaction/response even to small perturbations in the model or in the model computational environment (e.g., routine changes in computer hardware, operational system, compilers, libraries). Any, even infinitesimal change in model formulation, initial conditions or computational environment may make two model integrations diverge, with the effect that after the deterministic predictability is lost (which takes just weeks for the atmosphere, although longer for the ocean), the timing and location of weather patterns becomes essentially independent for the two integrations. Hence, the two control model runs produced with the aforementioned small changes provide, in essence, two independent samples of the model's climatology, and their difference represents the model's internal variability. Thus, the internal variability of the model provides an estimate of the uncertainty of climate simulations and plays for climate simulations the same role that observational noise does for observations. Therefore, from a practical point of view, if changes introduced in the model lead to the changes in climate simulations of the order of the model internal variability, the impact of the changes can be considered negligible.

In order to emphasize how small the changes introduced by the use of NN emulations are, for the NCEP CFS, the model's internal variability was estimated and used for comparison. If the differences/changes introduced in the model results by using the NN emulation are of the same order of magnitude as the aforementioned model's internal variability, it demonstrates that, from the practical point of view, the approximation error of the NN emulation is negligible and, therefore, the NN's accuracy is sufficient for the use in the model. A very important conclusion for the NN emulation training is that, for NN emulations to be used in a complex nonlinear environment, the accuracy of the NN emulation should not exceed the natural internal variability of the environment; otherwise, problems as overfitting (fitting the noise as well as the signal) may occur.

To estimate the model's internal variability, we produced two control runs with the original NCEP CFS model configuration, i.e., without NNs. The first run was performed before and the second run after the routine changes (introduced quasi-regularly by system administrators) of the version of the FORTRAN compiler and FORTRAN libraries. Small differences between these two runs (which are similar to those due to changes in a computer operation system and/or in hardware) are

shown below for NCEP CFS together with the differences between the parallel NN and control runs for comparison, as an additional measure of the NN emulation accuracy.

Comparison of Parallel Runs

Two forty-year (1961–2001) parallel climate simulation runs (control and NN runs) have been performed for NCAR CAM. For NCEP CFS, three 17-year (1990–2006) parallel climate simulations have been performed: one NN run and two control runs (see previous section). A series of 8 day weather forecasts have been performed (both control and NN runs) for NCEP GFS. For the climate runs, the differences between the parallel runs have been analyzed in terms of spatial (global) means as well as temporal characteristics. For weather forecasts, statistical metrics routinely used in NWP (like anomaly correlation) have been employed. Here only selected results are presented to illustrate the NN emulation performance; for detailed analysis and discussion, see Krasnopolsky et al. (2008b, 2010, 2012).

First, the differences between the parallel simulations in terms of spatial and temporal radiation characteristics are presented. Figure 4.6 shows zonal and time-mean SWR heating rates (in K/day) for the NN run (the upper left panel), the control run (the upper right panel), and their difference (the bottom panel) for NCAR CAM 40-year runs. The zonal mean is a 2-D field obtained from the 3-D field of HRs, depending on lat, lon, and a vertical coordinate, i.e., pressure or height, by integrating over the longitude. The HR patterns (the upper panels) are very similar and the differences (the bottom panels) are small; they do not exceed 0.1 K/day. It is noteworthy that the HR differences in SWR are a bit larger near the surface because the HRs are larger there (Fig. 4.6).

The differences between the NN and control parallel runs and the differences between two control parallel runs for zonal and time-mean LWR and SWR fluxes are presented in Fig. 4.7 for NCEP CFS 17-year runs. The upper row of Fig. 4.7 shows the differences for zonal and time-mean top of atmosphere upward long (left panel)- and short (right panel)-wave fluxes (in W/m^2) for winter. The lower row of Fig. 4.7 shows the differences for zonal and time-mean downward (left panel) and upward (right panel) surface LW fluxes (in W/m^2). For the fluxes presented in Fig. 4.7, both the differences between the NN radiation and control runs (solid lines) and the differences between two control (dashed lines) runs (the model internal variability) are small and similar in magnitude. They do not exceed 2–3 W/m^2, i.e., they are within observational errors and uncertainties of the reanalysis (e.g., Kalnay et al. 1996; Kistler et al. 2001). The similarity of the differences in magnitude indicates that the differences between the NN radiation and control runs are comparable with the model's internal variability; thus, from the practical point of view, they are negligible.

Next, Fig. 4.8 shows the NCAR CAM global temperature field, a prognostic variable, at height of 850 mb. The horizontal fields presented in the upper and middle panels are very similar. From the difference field (the bottom panel),

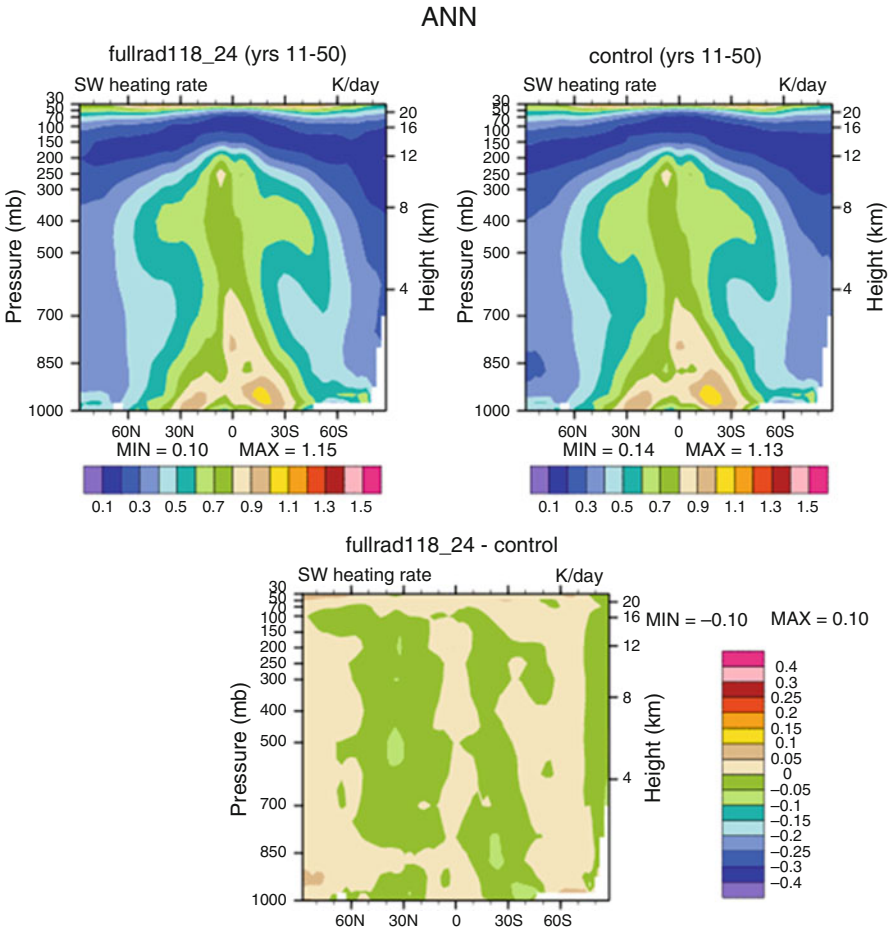

Fig. 4.6 The NCAR CAM zonal and time (1961–2001) mean SWR heating rates, in K/day, for the NN run (the *upper left panel*), the control run (the *upper right panel*), and their difference (the *bottom panel*). Zonal mean is a 2-D field obtained from the 3-D field of HRs, depending on *lat, lon*, and a vertical coordinate, by integrating over longitude. Thus, the horizontal axis represents latitude

the bias is very small (−0.06 K), *RMSE* is small (0.34 K), and minimum and maximum values (∼−1.6 K and ∼0.9 K) are well within observational or reanalysis errors/uncertainties.

Figures 4.9 and 4.10 show total precipitation and total cloud fields for NCEP CFS. These fields are very sensitive to any changes in the model and, therefore, provide a sensitive indicator of the accuracy of NN emulations. Highly similar results have been obtained for these parallel runs in terms of the time-mean spatial fields presented in Figs. 4.9 and 4.10. The figures contain two rows: The upper row shows fields produced by the control (left) and full NN radiation (right) runs, and the

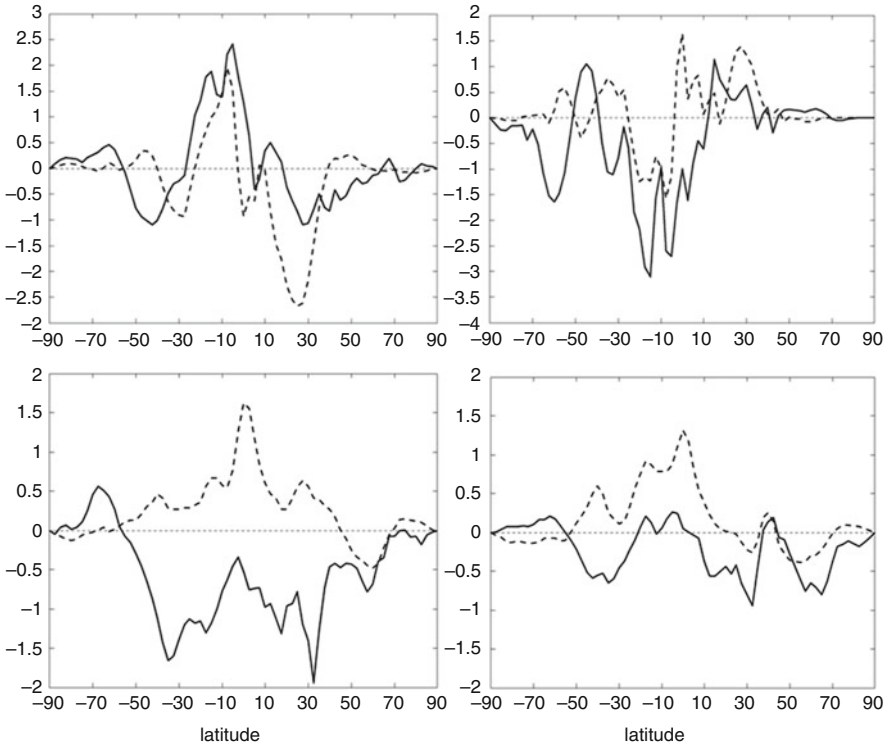

Fig. 4.7 The *upper row*: the differences for NCEP CFS zonal and time (1990–2006) mean of the top of atmosphere upward long (*left panel*)- and short (*right panel*)-wave fluxes (in W/m^2) for the winter. The *solid line* is the difference of the full radiation NN run and the control (CTL), and the *dashed line* is the difference between two control runs (the model internal variability) presented for comparison. The *lower row*: the differences for zonal and time annual mean downward (*left panel*) and upward (*right panel*) surface long-wave flux (in W/m^2). The zonal mean shows a curve obtained from the 2-D field of radiation flux, depending on *lat* and *lon*, by integrating over the longitude. The flux differences are multiplied by *cos*(*lat*) to equalize the areas (Krasnopolsky et al. 2010) (©) American Meteorological Society. Used with permission

lower row shows the difference fields, i.e., the differences between the full radiation NN run and the control run (left) and the differences between two control runs (i.e., model's internal variability) presented for comparison (right).

Figure 4.9 shows the 17-year (1990–2006) time-mean distributions and differences for total precipitation rate for summer, and Fig. 4.10 presents the 17-year (1990–2006) time-mean distributions and differences of total clouds for winter for the parallel full radiation NN and control runs. The precipitations and cloud patterns for parallel total radiation NN and control runs are very close. The precipitation rates and cloud mean differences are quite limited and occur mostly in the tropics; they are also very close in magnitude (as well as *RMSEs*) and pattern to the model's internal variability. The results for other seasons are similar.

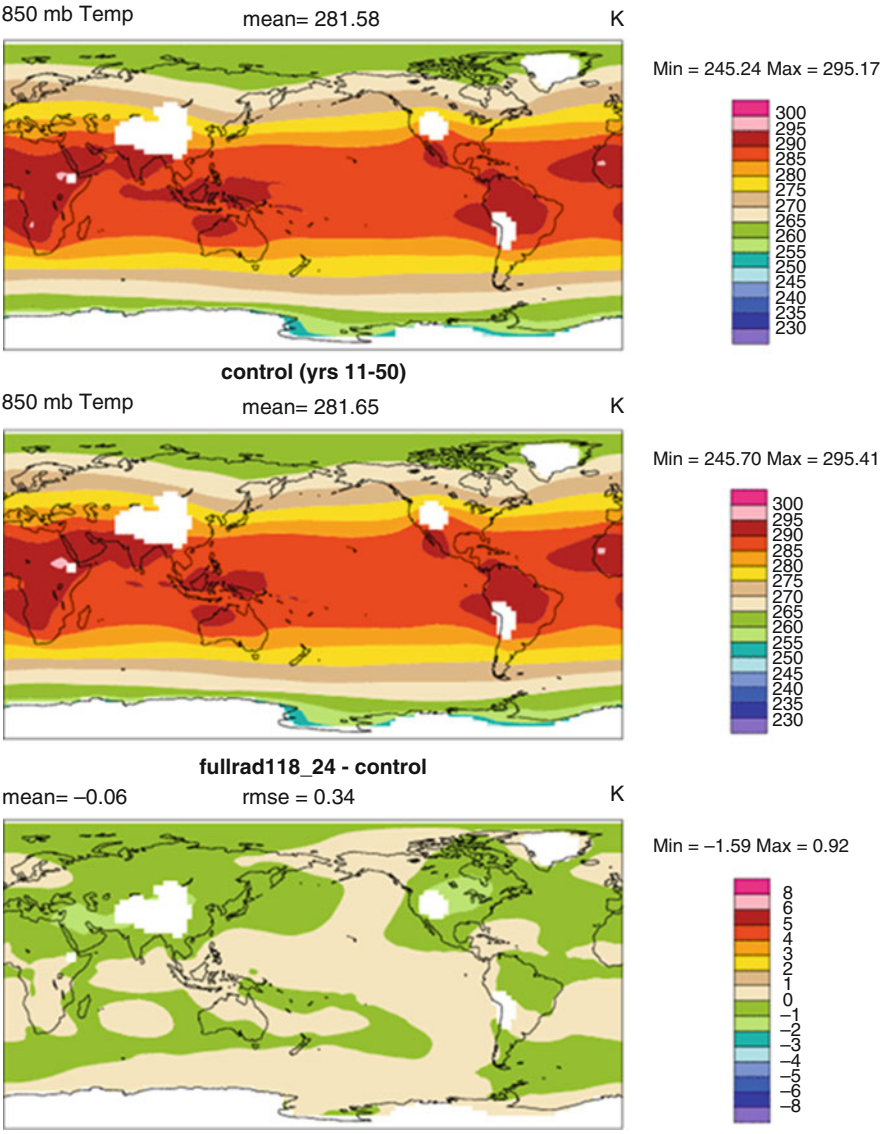

Fig. 4.8 The NCAR CAM time-mean (1961–2001) temperature at 850 hPa, in K, for the full radiation NN run (the *upper panel*), the control run (the *middle panel*), and their difference (the *bottom panel*)

After validating the NN radiation at the climate time scales and spatial resolutions in decadal climate simulations, the NN radiation was introduced in a NWP model, the NCEP GFS, and tested at weather time scales and spatial resolutions in a series of 8-day forecasts. The LWR and SWR emulations with 100 hidden neurons have

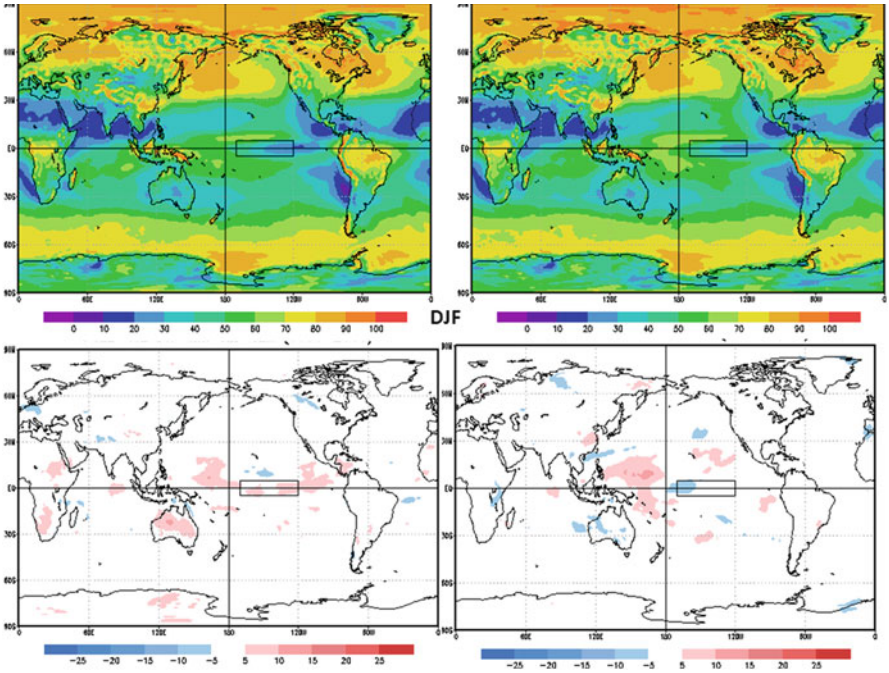

Fig. 4.9 The NCEP CFS time-mean (1990–2006) total precipitation rates for summer (JJA) for the parallel full radiation NN and control runs. The *upper row panels*: *left*, the control (CTL), and *right*, full radiation NN run. The *bottom row panels*: *left*, the difference (full radiation NN run – CTL), and *right*, the model internal variability for comparison. The contour intervals for the fields are 1 mm/day for the 0–6 mm/day range and 2 mm/day for the 6 mm/day and higher. The contour intervals for the differences (the *bottom panels*) are 1 mm/day

been selected for an initial validation because they appear to be acceptable in terms of both accuracy and minimal complexity. A series of 8-day forecasts has been run using the GSF model. The comparisons of anomaly correlations (AC), biases, and *RMSEs* have been performed for instantaneous model prognostic and diagnostic fields produced at each day of the 8-day forecasts. The NN radiation and control runs are very close in terms of statistics that were calculated. For example, Fig. 4.11 shows the AC, which is a statistical measure of correctness of the forecast. AC can be calculated using the following equation (Krishnamurti et al. 2003):

$$\mathrm{AC} = \frac{\sum \left\{ \left[(T_f - T_c) - \overline{(T_f - T_c)} \right] \cdot \left[(T_v - T_c) - \overline{(T_v - T_c)} \right] \right\}}{\sqrt{\sum \left[(T_f - T_c) - \overline{(T_f - T_c)} \right]^2 \cdot \sum \left[(T_v - T_c) - \overline{(T_v - T_c)} \right]^2}} \tag{4.14}$$

where the suffix f denotes the forecast, suffix c denotes climatology, and suffix v stands for verifying analysis. The overbar is the global (area) mean and T is the global temperature. The summation is performed over all grid points of the

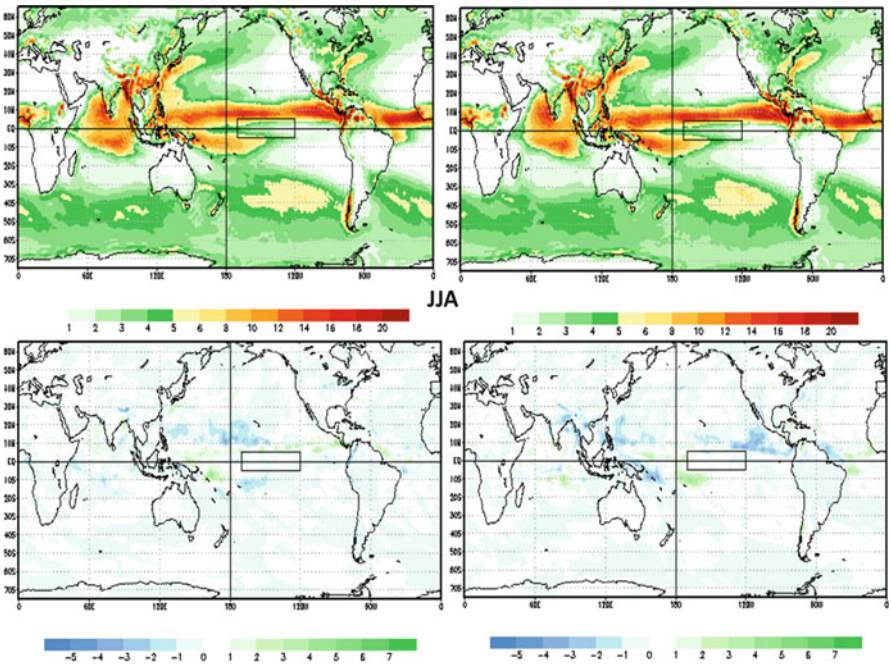

Fig. 4.10 The NCEP CFS time-mean (1990–2006) total clouds for winter (DJF) for the parallel full radiation NN and control runs. The *upper row panels*: *left*, the control (CTL), and *right*, full radiation NN run. The *bottom row panels*: left, the difference (full radiation NN run – CTL), and right, the model internal variability for comparison. The contour intervals for the cloud fields are 10 % and for the differences are 5 %

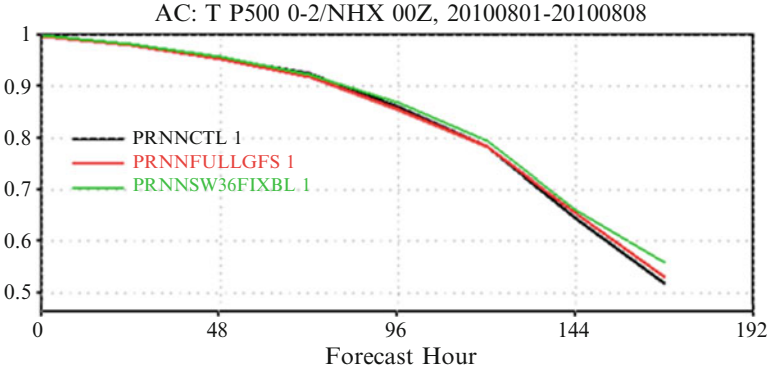

Fig. 4.11 Anomaly correlation at 500 mb for the global temperature field. *Black line*, control GFS run with the original LWR and SWR parameterizations; *green line*, GFS run with NN SWR and the original LWR parameterization; and the *red line*, GFS run with NN SWR and NN LWR

horizontal global grid. In general, the closer AC is to 1, the better the forecast; AC usually decreases with the increasing the forecast time.

Figure 4.11 shows AC for the temperature at a height of 500 hPa. It demonstrates that NN LWR, which is about 20 times faster, and NN SWR, which is about 60 times faster than the original RRTM LW and SW radiation codes, respectively, are very accurate and do not degrade the accuracy of the GFS 8-day 500 hPa temperature forecast for this case. Similar results have been shown for the entire series of 8-day forecasts (Krasnopolsky et al. 2012).

Normalization of NN Outputs, and an Array of NNs Versus a Single NN

In Sects. 2.3.4 and 2.3.5, a choice of the NN emulation architecture (an array or a battery of NNs vs. a single NN) and different normalizations of NN outputs have been discussed. Here these topics are illustrated using an example of NN emulations of the NCAR CAM LWR parameterization (see also Krasnopolsky and Fox-Rabinovitz 2006b).

For a single NN with multiple outputs, the normalization of outputs affects the approximation accuracy more significantly than for an NN with a single output. Figure 4.12 illustrates the dependence of approximation errors, at the different vertical model levels, on the type of output normalization. The right panel shows the absolute approximation *rmsep* (Eq. 4.12) for LWR heating rates in K/day; the left panel shows the relative approximation *rmsep* normalized at each vertical level using the standard deviation (σ_q) of the heating rate at this level, q. The dotted curve corresponds to the [−1.,1.] output normalization (Eq. 2.6). It is clear that this normalization deemphasizes in the error function the vertical levels with small σ_q (levels 13–18) and large σ_q (levels 0–3), leading to larger and vertically nonuniform errors, both absolute and relative. The normalization (Eq. 2.8) for the case of multiple outputs is introduced to accelerate the training of linear weights in the output layer of the NN. In the case of multiple outputs, this normalization leads to very different approximation errors (dashed curves) as compared with the normalization (Eq. 2.6). This normalization leads to a more uniform vertical distribution of relative errors (left panel) and significantly reduces relative and absolute errors at vertical levels with small σ_q (levels 13–18); however, it significantly increases the errors at vertical levels with large σ_q (levels 0–3).

A distribution (not shown in Fig. 4.12), very similar to the dashed distribution of errors, is produced by the array of 33 single-output NNs, each of which uses normalization (Eq. 2.6). In both cases, information about correlations between outputs is lost and not used during the NN training (see Sect. 2.3.5).

A compromise between the normalization (Eq. 2.6) and normalization (2.8) can be reached using the normalization (Eq. 2.9). The errors for this normalization are also shown in Fig. 4.12 (solid curves). For different applications of the NN emulations, different types of error distribution may be desirable; smaller absolute or relative errors may be preferable. Different output normalizations in the case of a single emulating NN with multiple outputs may provide a tool for managing this type of problems.

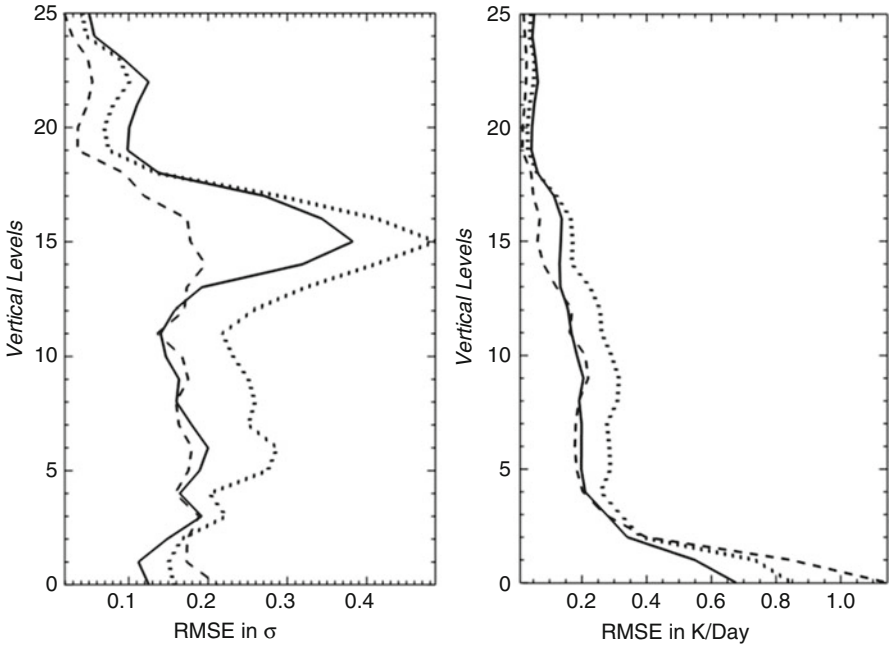

Fig. 4.12 LWR NN emulation errors for NCAR CAM. Vertical profiles of *rmsep* (Eq. 4.12) of LWR HRs at each of 26 model vertical levels for different architectures of the LWR NN emulation and for different types of output normalizations. The *left panel* shows relative *rmsep* in units of the standard deviations of LWR HRs calculated at the same vertical level. The *right panel* shows the absolute *rmsep* in K/day. *Solid*, *dashed*, and *dotted lines* show *rmsep* for three single NNs with 33 outputs and 150 hidden neurons each. *Dotted lines* correspond to the output normalization (Eq. 2.6), and *dashed* and *solid curves* correspond to the normalizations Eqs. (2.8) and (2.9), respectively

If single-output NNs are used, a battery of 33 NNs should be trained to emulate the LWR parameterization. An array of NNs with approximation accuracy close to that of a single NN with 150 neurons (e.g., Fig. 4.12, solid line) has a total of about 450 hidden neurons; therefore, taking into account the correlation of outputs by using a single emulating NN with multiple outputs allows us to obtain a performance gain (a calculation speedup) of about three times (for the same approximation accuracy) when using a single NN with multiple outputs.

4.3.5 Compound Parameterization for NCAR CAM Short-Wave Radiation

In subsection "Compound and adjustable parameterizations, quality control of NN emulations" of Sect. 4.2.2, the basic concepts associated with QC are discussed.

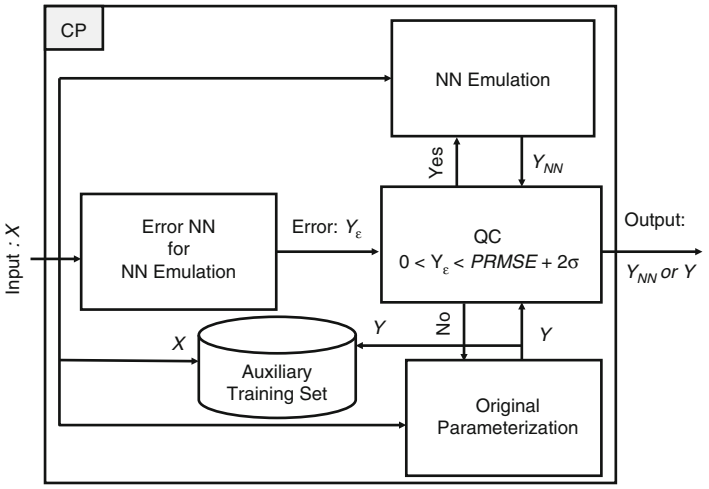

Fig. 4.13 Compound parameterization design for the NCAR CAM SWR. For a SWR NN emulation, additional NN (Error NN) is trained specifically for predicting the errors, Y_ε, in the NN emulation output Y_{NN} for a particular input, X. If these errors do not exceed a predefined threshold (in this case, the mean value plus two standard deviations), the SWR NN emulation is used; otherwise, the original SWR parameterization is used instead of the NN emulation. The auxiliary training set (ATS) is updated each time when QC requires using the original parameterization instead of the NN emulation. ATS is used for the following adjustments of the NN emulation

Here an example of a practical solution to this problem for NN emulations of the NCAR CAM SWR parameterization (see also Krasnopolsky and Fox-Rabinovitz 2006b; Krasnopolsky et al. 2008a) is presented.

An effective QC design is based on training an additional NN to specifically predict the errors of the NN emulation outputs for a particular input. The "error" NN has the same inputs as the NN emulation and one or several outputs, which predict errors of outputs generated by the NN emulation for these inputs. In the example presented here, an error metric that produces one integrated error for all outputs, *prmse* (Eq. 4.10), is used; thus, the error NN has one output.

The original parameterization, its NN emulation, the error NN, and the QC block constitute the CP, whose design is shown in Fig. 4.13. During the model integration, CP works in the following way: If the error predicted by the error NN does not exceed a predefined threshold, the NN emulation is used; otherwise, the original parameterization is used.

For the SWR NN emulation described in the Sect. 4.3.3 (using an NN with one hidden layer that contains 55 neurons and a linear output layer), an error NN was trained which predicted the SWR NN output error *prmse*(i) (Eq. 4.10) for each particular input vector X_i. The mean value of *prmse*, *PRMSE*, and its SD, σ (Eq. 4.11), are used in the QC block for calculating the threshold value.

Errors predicted by the error NN are close to the actual errors of the NN emulation calculated for the same input vector. Figure 4.14 shows the results of the calculations performed with the test data set containing more than 100,000 input

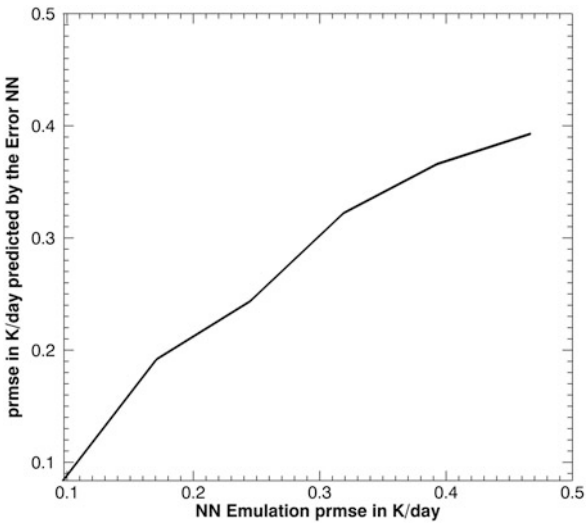

Fig. 4.14 The correlation (a binned scatter plot) between the actual error (*prmse* of the SWR NN emulation) and the error predicted by the error NN. Both errors are calculated vs. the original parameterization on an independent test data set. The correlation coefficient between the two errors is 0.87

records. For each of the inputs, the error predicted by the error NN and the actual error of the NN emulation have been calculated. The actual errors of the SWR NN emulation were binned, and for each bin, a corresponding mean error predicted by the error NN was calculated and plotted as a curve vs. the actual errors of the SWR NN emulation.

Figure 4.14 shows a very strong correlation between the error predicted by the error NN and the actual error of the SWR NN emulation calculated vs. the SWR original parameterization on an independent test data set. The dependence, linear for small errors, becomes nonlinear for larger errors. A high correlation coefficient (0.87) is obtained between these two errors calculated on the entire test data set.

Figure 4.15 shows the comparison of two error probability density functions. One curve (solid line) corresponds to the SWR NN emulation errors; another (dashed line) corresponds to the CP errors (both errors are calculated vs. the original parameterization on the independent test set; the vertical axis is logarithmic). Figure 4.15 demonstrates the effectiveness of CP; the error reduction is shown by the differences between the solid and dashed lines. The application of CP reduces medium and large errors by about an order of magnitude and completely eliminates errors exceeding 10 K/day (see also Table 4.4).

Figure 4.15 demonstrates the effectiveness of CP in removing outliers, and Table 4.4 supports this conclusion and shows improvements in other statistical measures as well. The use of CP (a) does not increase the systematic error (bias) which is almost zero and (b) significantly reduces the random errors. Especially significant is the reduction of extreme errors or outliers. It is noteworthy that for this CP and for this validation data set, less than 1 % of the SWR NN emulation outputs are rejected by QC and, therefore, are calculated using the original SWR parameterization. Further refinement of the criteria used in the QC may result in a further significant reduction in the already small percentage of outliers.

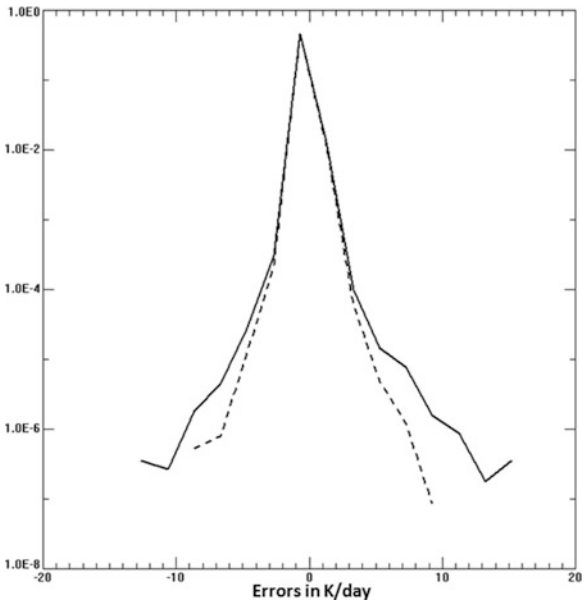

Fig. 4.15 Probability density distributions of emulation errors for the SWR NN emulation (*solid line*) and for the compound SWR parameterization shown in Fig. 4.13 (*dashed line*). The *vertical axis* is logarithmic and shows the error probability; the *horizontal axis* shows the NN emulation errors in K/day. In both cases, errors are calculated vs. the original SWR parameterization. The CP reduces the probability of medium and large errors by an order of magnitude and completely eliminates errors exceeding 10 K/day

Table 4.4 Error statistics for the SWR NN emulation and the SWR compound parameterization: bias and total *RMSE*, $RMSE_{26}$ at the lower model level, and extreme outliers (min and max errors)

	Bias	RMSE	$RMSE_{26}$	Min error	Max error
SWR	$4.\ 10^{-3}$	0.19	0.43	−46.1	13.6
SWR CP	$4.\ 10^{-3}$	0.17	0.30	−9.2	9.5

These statistics have been calculated on an independent one-year-long test set. All errors are in K/day

The CP design outlined above has been implemented in NCAR CAM using the SWR NN emulation. A number of 40-year model simulations have been performed with this QC procedure using different thresholds. An appropriate threshold of 0.5 K/day has been determined experimentally. In this context, choosing an appropriate threshold means that the selected threshold (which is approximately equal to $\mu + 2\sigma$) does not allow for even limited accumulation of errors (see the red line in Fig. 4.16) during the CAM simulation and, at the same time, does not reduce the computational speedup gained by using the fast NN emulation. Thus, at each integration time step and at each grid point in the model with CP, the error NN that predicts the error of the NN emulation was estimated, and if the predicted error did not exceed 0.5 K/day, the NN emulation outputs were calculated and used in the model; otherwise, the original parameterization was calculated and its outputs were used in the model.

The example shown in Fig. 4.16 illustrates the effectiveness of CP in eliminating any accumulation of errors in the course of the model integration. When the model is integrated without QC, the SWR NN emulation produces moderately increased

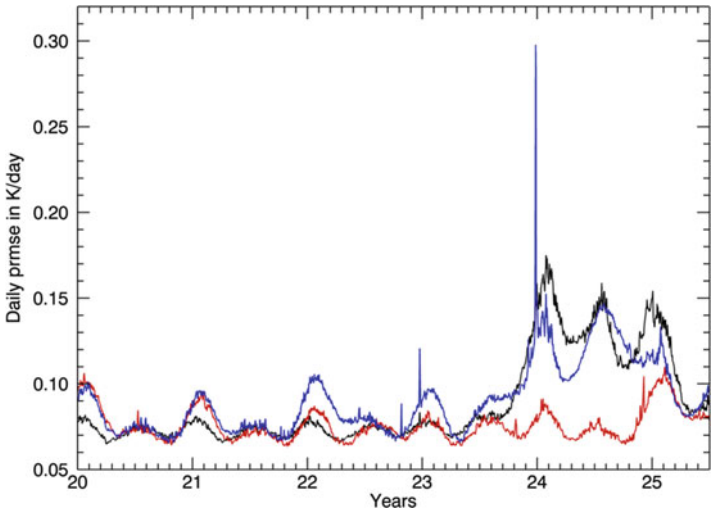

Fig. 4.16 Errors (vs. the original SWR parameterization) produced by the SWR NN emulation during the model run (*blue line*), errors predicted by the error NN (*black line*), and errors produced after introducing CP instead of the SWR NN emulation (*red line*) (Reprinted from Krasnopolsky et al. (2008a) with permission from Elsevier)

errors (errors increase from ∼0.07 K/day to ∼0.14 K/day) during the period between 24 and 25 years of the integration (the blue curve in Fig. 4.16). The error NN predicts this increase of the errors very well (the black curve in Fig. 4.16). After the QC was applied, that is, the model was integrated with CP, the level of errors dropped significantly, in general, and, even more important, the spikes and the bump between 24 and 25 years disappeared completely (the red curve in Fig. 4.16).

Using CP provides a stable and reduced error environment for model simulations compared to the model simulations performed without QC. It is noteworthy that the NN emulation outputs were rejected by the QC and the original parameterization was used on average for only 0.05–0.1 % (with the maximum below 0.4–0.6 %) of model grid points, throughout the entire 50-year model simulation. Therefore, the computational performance of the model with NN emulation was not reduced significantly, and CP is still about 20 times faster than the original SWR parameterization.

4.3.6 NN-Based Convection Parameterization for NCAR CAM Derived from CRM-Simulated Data

Subsection "Using NNs for developing new NN-based parameterizations" of Sect. 4.2.2 and Fig. 4.2 introduce the methodology for developing the NN-based convection parameterization. Here we introduce and discuss certain practical details and the results for this development (for more details, see Krasnopolsky et al. 2011, 2013).

Parameterization and Its Uncertainties

In this section, the sources of uncertainties, which emerge in the process of data preparation for the development of an NN convection parameterization for a GCM based on the data simulated by a CRM, are outlined (for a detailed discussion, see Krasnopolsky et al. (2011)). The major properties of the uncertainties are discussed. It is shown that the uncertainty is an inherent part of the data and, therefore, of the parameterization derived from these data and that this parameterization is essentially stochastic parameterization in nature.

The first three steps of the development process formulated in subsection "Using NNs for developing new NN-based parameterizations" of Sect. 4.2.2 introduce uncertainties in the training data set or "pseudo-observations." Uncertainties are introduced at each of these steps and their sources can be traced step by step. First, the CRM may be formally considered as a mapping, μ, which defines the relationship between two vectors: the input vector (x) and the output vector (y) that are composed of CRM variables. At each time step, given a vector x, the mapping μ produces vector y or

$$y = \mu(x). \tag{4.15}$$

Here (x, y) are high-resolution CRM variables (produced with spatial resolution of 1 km and temporal resolution of 5 s) or CRM-simulated data; they are related by the CRM and this fact is expressed in Eq. (4.15). The mapping μ is an exact (or deterministic, or PB) mapping, which means that it is explicitly represented by a complete set of CRM equations and that one particular y corresponds to each particular x. The first step in the developmental process consists of simulating the CRM or applying (4.15) at each time step and continuing with the simulation for a period of time T. The CRM simulation is forced at each time step by large-scale observational data. However, because the CRM physics (e.g., the microphysics) is partially parameterized and contains a number of simplifications, the CRM is not perfect and the CRM-simulated data will deviate from the observational data. This difference between the observed reality and the "CRM reality" is the first contribution to the uncertainty of the pseudo-observations and the parameterization derived from these data.

The second step in the development consists of averaging high-resolution simulated data (x, y) over a certain area r ($\rho < r < R$) and over a time interval t ($\tau < t < T$), where $\rho = 1$ km and $\tau = 5$ s are the CRM resolution and integration time step, respectively, and R and T are the GCM resolution and integration time step, respectively. As a result, the averaged vectors of simulated data $\underline{x}$ and $\underline{y}$ are produced. Here the bar below the symbol means averaging over r and t.

By changing r and t, we can regulate the amount of subgrid information (high-frequency variability) in pseudo-observations that we want our parameterization (derived from the pseudo-observations) to introduce in CAM. Thus, moving from $r = 1$ km and $t = 5$ s (high-resolution CRM data) to lower resolution, ρ, and larger t, we can gradually reduce the subgrid signal introduced in CAM. Determining the

optimal values for r and t is one of important topics to be investigated through the validation of the performance of the developed NN convection parameterizations in CAM.

It is important to emphasize that the new variables $\underline{x}$ and $\underline{y}$ are stochastic variables that are distributed around their mean values with a certain probability density functions.

The next step in the development process is *projecting the* CRM space of atmospheric states onto the GCM space of atmospheric states. It starts from a transition from averaged CRM variables $\underline{x}$ and $\underline{y}$ to a subset of these variables, X and Y. Let us write $\underline{x}$ and $\underline{y}$ as $\underline{x} = \{X, x'\}$ and $\underline{y} = \{Y, y'\}$. Here we split each vector $\underline{x}$ and $\underline{y}$ into two parts. The *new variables* X and Y *include only variables that can be identified with corresponding GCM variables or can be calculated from or converted to prognostic or diagnostic variables available in the GCM;* all other variables x' and y' are projected out (removed or averaged out). It can be shown (Krasnopolsky et al. 2011) that the new variables X and Y contain additional uncertainty due to *the unaccounted variability of the omitted parameters* x' and y'.

Finally, we assume that there exists a mapping between the new vectors of projected variables X and Y, which can be written as

$$Y = M(X) + \varepsilon. \tag{4.16}$$

The mapping M is a complex *stochastic mapping* between two *stochastic vector variables* X and Y (see Sect. 2.2.4). The stochastic mapping (4.16) is not exact. For each particular value of X, it may generate many different values of Y with different probabilities determined by their joint probability density function $\rho(X,Y)$ conditioned by the uncertainty ε. Also one value of Y can be generated by a stochastic mapping from different values of X with different probabilities determined by their joint probability density function.

The projected vectors X and Y do not correspond to any particular values of omitted parameters x' and y'*; these values are uncertain.* The uncertainty vector ε may contain not only random contribution but also a systematic component as well (Krasnopolsky et al. 2011).

The mapping (4.16) determines *a stochastic parameterization*, which inherently contains the uncertainty ε. The parameterization is *implicitly defined by the training set (X, Y)*. The uncertainty ε in this case is not a noise; it is an inherent informative part of the stochastic parameterization, which contains important statistical information about subgrid scale effects (in terms of the GCM).

NN Emulation of the Parameterization and Estimation of Its Uncertainties

A data set was simulated for the development of the NN convection parameterization. The set is limited due to the lack of longer observational data needed to provide the forcing for the CRM simulations. The CRM using TOGA-COARE forcing was run for the 256×256 km domain with 1 km resolution and 96 vertical

layers (0–28 km) for 120 days. Then the results were averaged every hour of the model integration to produce a simulation data set with an effective resolution of 256 km. Finally, only variables that are available in the GCM (NCAR CAM) or can be calculated from the GCM variables have been selected. The final data set consists of 2,800 records (one record per hour). The simulated data set was partitioned into two parts: a training set, with 2,240 records or 80 % of the data, and a test set, with 560 records or 20 % of the data. Namely, the first 2,240 records are included in the training set and the last 560 records in the test set.

These two data sets have been used for the NN training and test/validation. As indicated in the previous section, these data implicitly represent a stochastic parameterization that inherently contains an uncertainty, ε. In subsection "Noisy training data and stochastic mappings" of Sect. 2.3.7, the training criterion (2.11c) has been formulated for the case when the data contain a significant level of noise or uncertainty. Thus, in the case of the stochastic parameterization, the NN approximation task is different from that of emulating the original *deterministic* parameterization in the GCM that was described in the previous sections. The simulated data that represent a deterministic parameterization do not contain noise of magnitude significantly higher than round-off errors.

This important difference should be taken into account when the NN emulation is trained, the approximation error statistics are analyzed and interpreted, and the NN architecture is selected. In this case, all NNs that satisfy the condition (2.11c) are valid emulations of the stochastic parameterization (4.16). Actually, each of these NNs emulates a member of the family of mappings that together represent the stochastic parameterization (4.16). Therefore, all NNs that satisfy (2.11c), or the entire ensemble of NNs, represent the stochastic parameterization (4.16). Thus, estimates of the magnitude of the uncertainty, ε, are, indeed, important for this approach. Such an estimate is derived in the following sections.

NN Architectures and NN Training

Selecting an emulating NN architecture has two aspects: (i) selecting inputs and outputs and their numbers (n and m in (2.2)), which, as we have already indicated, are determined by the availability of the variables in the GCM, and (ii) selecting the number of hidden neurons (k in (2.2)) in the emulating NN, which is determined by many factors (the length of the training set, the level of noise in the data, the characteristics of conversions of the training and test errors, etc.).

Different NN architectures (combinations of inputs and outputs) have been investigated by Krasnopolsky et al. (2011). The architecture that has finally been selected is shown in Table 4.5. The values in Table 4.5 show how many vertical levels in the corresponding profile have been included as inputs in the NN. Many profiles have elements that are zeros, or constants, or values that are almost constant (their SDs are very small) for the entire data set. Such profile elements should not be included as NN inputs or outputs. If such variables are important from a physical point of view, they should be normalized or weighted (see Sect. 2.3.6) appropriately.

Table 4.5 NN architecture used for the development of convection parameterization

	NN inputs		NN outputs			
NN architecture in:out	T	QV	Q1C	Q2	PREC	CLD
36:55	18	18	18	18	1	18

T is the atmospheric temperature, *QV* is the atmospheric moisture, vapor mixing ratio, *Q1C* the "apparent heat source", *Q2* the "apparent moisture sink", *PREC* the precipitation rate; and *CLD* the cloudiness. *Numbers* in the table show the dimensionality of the corresponding input and output parameters. In:out stands for NN inputs and outputs and shows their corresponding numbers

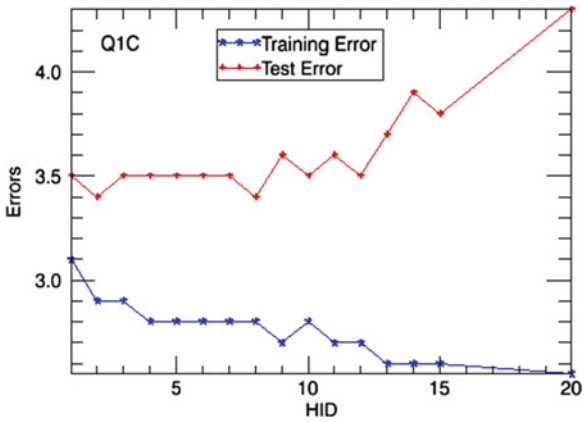

Fig. 4.17 NN approximation error for the training (*blue*) and test (*red*) sets for Q1C

A set of different NNs with different number of hidden neurons (HID) from 1 to 20 has been selected, and the corresponding NNs have been trained and tested. Figure 4.17 shows the results of these experiments, which include the approximation errors for an output parameter, Q1C. The figure shows NN errors for the training and test sets for HID increasing from 1 to 20 for the NN architecture shown in Table 4.5.

It is important to understand that NN training (based on least square minimization) attempts to minimize the total $(\varepsilon + \varepsilon_{\text{app}})$, that is, the sum of the approximation error and the uncertainty or noise (see subsection "Noisy training data and stochastic mappings" of Sect. 2.3.7). Because of the very different statistical properties of these components, they can be approximately separated and approximately estimated using detailed information about the training and test statistics.

Figure 4.17 demonstrates a classic situation that is usually observed when an NN is trained using data with a significant level of noise. The training error, after a sharp initial drop, decreases very slowly. The test error, after an initial drop, stabilizes and then increases. The interpretation of this behavior is well known. After the

Table 4.6 The number N_C of fitting parameters (NN weights) for different values of HID $= k$ (see Eq. (2.2))

HID	1	2	5	10	15	20
N_C	166	273	594	1,129	1,667	2,199

initial improvement in approximating the data due to the increasing flexibility of the approximating NN, a short interval of stability is reached (at HID $\sim$ 3–7) where the NN fits the signal but filters out the noise. Here the training error keeps decreasing but the test error is almost constant. Then with the increase in the flexibility of the approximating NN, it starts to fit the noise as well (overfitting occurs). The training error keeps slowly decreasing; however, the test error increases.

Table 4.6 shows the number of parameters to be fitted (NN weights) in the NNs with different HID, which were used for plotting Fig. 4.17. Taking into account that the training set contains 2,240 records, it is not surprising that pronounced overfitting occurs for HID > 10.

Thus, we conclude that for a particular simulation (data set) used, HID $= 5$ would be a good approximation for the number of hidden neurons in the emulating NN. This value falls within the interval of stability for the test error when the NN emulation fits the mapping (4.16) but does not fit the noise in the data.

The training errors for all output parameters are also not significantly sensitive to the selection of the NN architecture, i.e., to the selection of HID inside the interval of stability (see Fig. 4.17) than the test errors (Ts). Thus, *the training errors can be considered as a rough estimate of the noise in the data that reflects the inherent uncertainty of the stochastic parameterization* (4.16).

Following this conclusion, *the errors on the test set are then a combination of the uncertainty (an estimate for it is provided by the training error) and an approximation error.* For example, for Q1C, the training error is 2.8 K/day and the test error is 3.5 K/day. Thus, assuming that the uncertainty and the approximation error are independent, i.e.,

$$\left\langle (\varepsilon + \varepsilon_{\text{app}})^2 \right\rangle = \left\langle \varepsilon^2 \right\rangle + \left\langle \varepsilon_{\text{app}}^2 \right\rangle$$

for the test error of 3.5 K/day, only 2.1 K/day can be attributed to the NN approximation error, and 2.8 K/day should be attributed to the uncertainty, ε, of the stochastic mapping (4.16). Thus, after the separation of the uncertainty (the training error), the NN approximation errors on the test set, in most of the cases, do not exceed the uncertainty.

An ensemble of ten NNs with the same architecture with different initializations for the NN weights has been trained. The coefficients (weights) for these NNs have been initialized, using the initialization of Nguyen and Widrow (1990) (see also Sect. 2.3.7) with different sets of small random numbers before the training. As a result, all of these NNs have different weights because the training process converged to different local minima. For these NNs, the errors on training and test

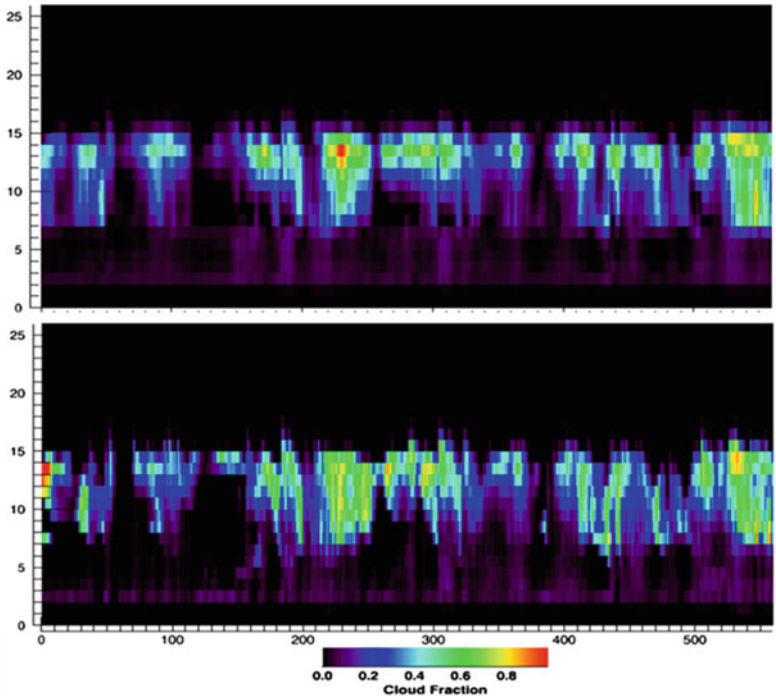

Fig. 4.18 Hovmöller diagrams for CLD profile time series: "pseudo-observations," in the *upper panel*, and the NN ensemble, in the *lower panel*

sets are similar. Hence, all trained NNs can be considered as valid emulations of the parameterization (4.16). *Thus, these NNs, taken together, can be considered as an NN ensemble emulation of the stochastic parameterization* (4.16).

Figure 4.18 illustrates performance of the trained NN ensemble on the independent test set. Figure 4.18 shows Hovmöller diagrams for the time series of cloudiness (CLD) profiles generated by the NN ensemble. The upper panels show "pseudo-observation" profiles and the lower panels show the time series of profiles generated by the NN ensemble. The patterns generated by the NNs are a bit smoothed or diffused; they are less sharp than the "observed" ones but still easily recognizable. The NN ensemble represents the sequence of patterns well and without significant shifts.

Validation of the NN Convection Parameterization in NCAR CAM

In this section, the NN stochastic parameterization is introduced into NCAR CAM. An ensemble of the ten NNs described above, which represents the stochastic parameterization (4.16), was used for testing in CAM. All input and output variables

for this NN architecture (Table 4.5) are directly available in CAM for use and comparison. Here the goal is to verify whether the NN ensemble, emulating the stochastic convection parameterization (4.16), provides meaningful and realistic outputs when using CAM inputs. The validation of our NN parameterization was performed in the following two experiments:

1. Over the TOGA-COARE location, the grid point (−2° S, 155° E) for the time period for which the TOGA-COARE experiment was conducted, where data are available for the TOGA-COARE winter, November 1992–February 1993. The data at this grid point and during this time interval have been used to develop the NN convection. The TOGA-COARE location, which is actually a small area in the Equatorial Pacific, is marked by a star in the middle panels of Fig. 4.20.
2. Over the large Tropical Pacific region (with an area size of $120° \times 30°$ and the following coordinates: 150° E < lon < 90° W, 15° S < lat < 15° N) during the decade (1990–2001, with the TOGA-COARE winter of 1992–1993 excluded). This is a challenging test for the NN's ability to generalize.

The parallel runs have been performed with the standard CAM and with the NN ensemble convection parameterization (CAM-NN run) for the decadal (1990–2001) winter climate simulations. Below we present some comparisons of the parallel decadal CAM-NN and CAM simulations and validate them against the NCEP reanalysis. For more systematic comparisons, see Krasnopolsky et al. (2011).

First, we will analyze the decadal simulations over the TOGA-COARE location only. The decadal mean CLD profiles for the TOGA-COARE location for CAM-NN and CAM shown in Fig. 4.19 are similar. The decadal mean profiles are consistent with those for the TOGA-COARE period (Krasnopolsky et al. 2011).

The time series of the decadal mean total CLD shown in Fig. 4.20 for the CAM run demonstrate measurably higher magnitudes, with a mean of 0.78, compared to that of the time series for the CAM-NN run, with a mean of 0.61. The time series of the recent NCEP reanalysis (Saha et al. 2010) shows lower magnitudes, with a mean of 0.54, which are significantly closer to those of the time series for CAM-NN.

The horizontal distributions of total precipitation for the large region of the Tropical Pacific Ocean (15 S to 15 N, 155 E to 90 W) for the CAM-NN run vs. the CAM control and the recent NCEP reanalysis have been produced and analyzed. The major result is that the regional precipitation distributions for the decadal parallel runs presented in Fig. 4.21 show a consistency and similarity, in terms of both pattern and magnitude, between the CAM-NN and CAM runs with the NCEP reanalysis used for validation.

Precipitation magnitudes for the CAM-NN run, in the area just around the Equator and extending to the southeastern part of the region, are overestimated compared to those from the CAM run and the NCEP reanalysis (Fig. 4.21). However, outside this area, i.e., the northern and southwestern parts of the region, precipitation magnitudes for the CAM run are underestimated compared to those of the CAM-NN run and the NCEP reanalysis (Fig. 4.21).

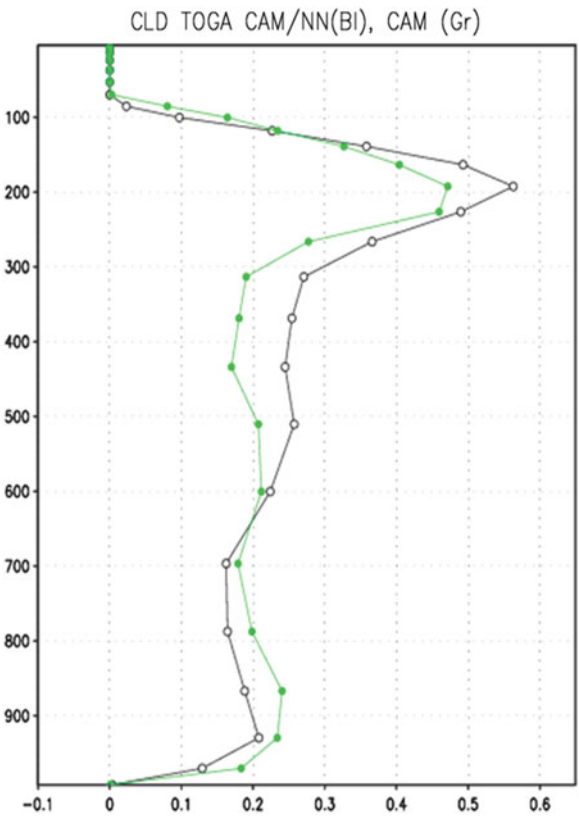

Fig. 4.19 Vertical profiles of decadal winter mean CLD for the TOGA-COARE location, in fractions, for CAM-NN (*black*) and CAM (*green*)

NCEP reanalysis (Kalnay et al. 1996), in the aforementioned area near the Equator and extending to the southeastern part of the region, precipitation magnitudes are higher than in the recent NCEP CFSRR reanalysis (Saha et al. 2010) and closer to those of the CAM-NN run. At this initial stage of the development of the stochastic NN convection parameterizations, it is reasonable to compare the CAM and CAM-NN runs in terms of their general consistency between themselves and with the NCEP reanalysis. Detailed climatological analysis of regional and global simulations for all seasons will be done at the next stage of the development. It will be based on using CRM simulations driven by CAM forcing. Thus, a representative global data set will be simulated for NN training, and global stochastic NN convection parameterizations will be developed for CAM.

Summarizing the results presented in this section, they illustrate the uncertainty of cloud and precipitation simulations for both the parallel CAM and CAM-NN runs. They also highlight the complexity of the analysis and validation of climate model simulations and the limitations of the data/information used in the tropics for validation.

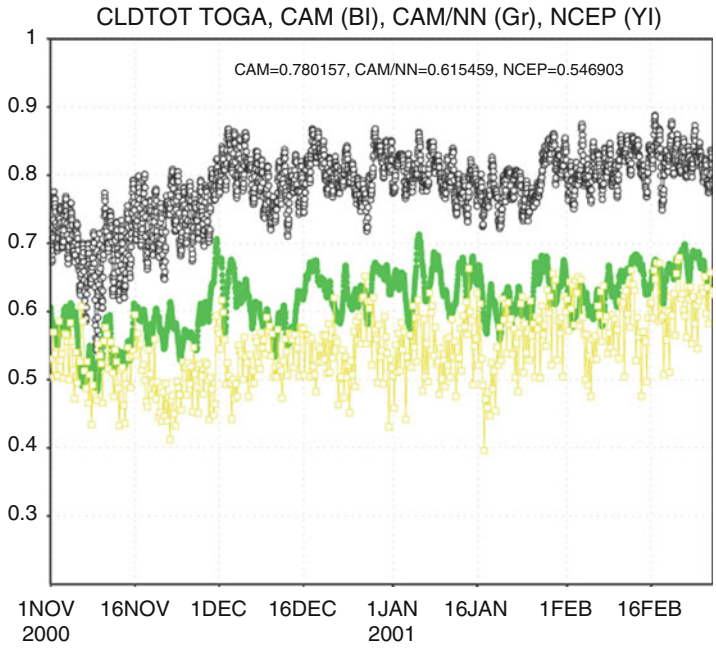

Fig. 4.20 Time series of decadal winter mean total cloudiness (in fractions) for the TOGA-COARE location for the CAM run (*black*) and CAM-NN (*green*) runs and for the NCEP reanalysis (*yellow*)

The results obtained for the decadal CAM-NN simulation are encouraging. They support the validity/soundness of the NN approach for developing NN convection parameterizations for climate models. However, it is clear that future work is needed for the practical implementation of NN convection parameterizations in climate models.

4.4 An Ocean Application of the Hybrid Model Approach: Neural Network Emulation of Nonlinear Interactions in Wind Wave Models

A state-of-the-art NCEP wind wave model called WAVEWATCH III (Tolman 2002) is an example of a complex environmental numerical model (ENM) (see also Sect. 4.1.1). After applying the NN emulation approach to a first-principle-based component of this model, the nonlinear wave-wave interaction source term, S_{nl} (Eq. 4.2), this model becomes a hybrid environmental model (HEM).

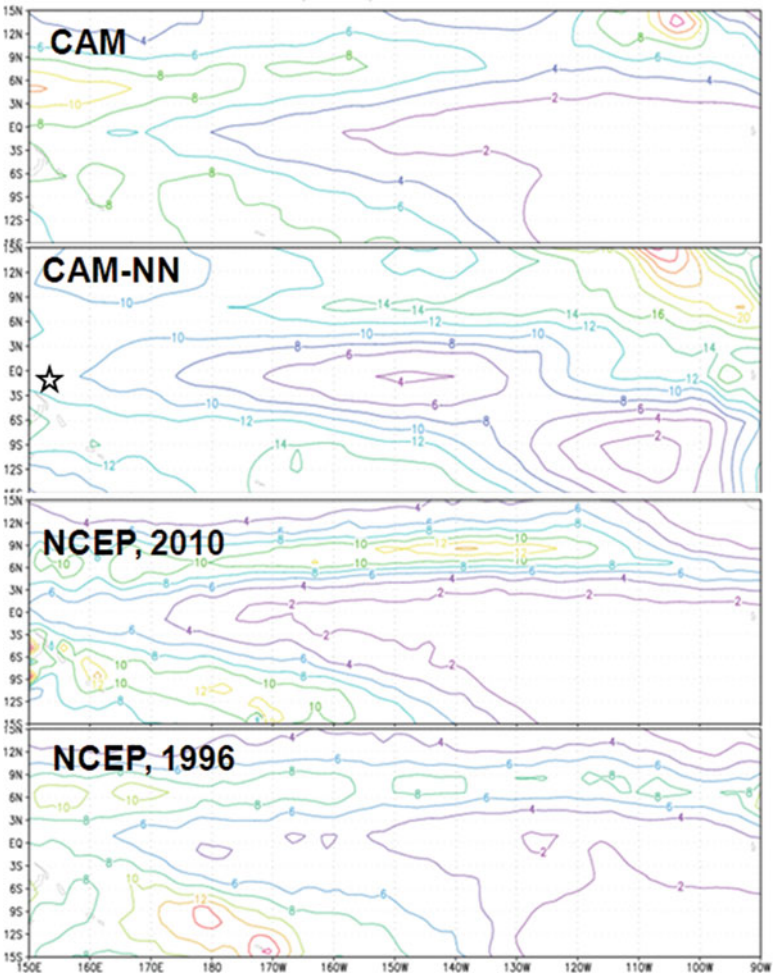

Fig. 4.21 Decadal winter mean precipitation distribution for the CAM (*upper panel*) and CAM-NN (*second panel*) runs over Tropical Pacific region (with the area size of 120° × 30° and the following coordinates: 150° E < *lon* < 90° W; 15° S < *lat* < 15° N). The *third* from the *top panel* shows the corresponding NCEP reanalysis decadal mean distribution. The TOGA-COARE location is shown by a *star* in the *second panel*. The *lowest panel* shows the previous NCEP reanalysis (Kalnay et al. 1996). The contour interval is 2 mm/day

The model is based on a form of the spectral energy or action balance equation (4.2) for the two-dimensional spectrum, F, and has the nonlinear wave-wave interaction source term S_{nl} as a part of the model physics. In its full form (e.g., Hasselmann and Hasselmann 1985; Hasselmann et al. 1985), the calculation of the

S_{nl} interactions requires the integration of a six-dimensional Boltzmann integral, which can be treated as a mapping:

$$\begin{aligned} S_{nl}\left(\vec{k}_4\right) &= T \otimes F\left(\vec{k}\right) \\ &= \omega_4 \int G\left(\vec{k}_1, \vec{k}_2, \vec{k}_3, \vec{k}_4\right) \cdot \delta\left(\vec{k}_1 + \vec{k}_2 - \vec{k}_3 - \vec{k}_4\right) \cdot \delta\left(\omega_1 + \omega_2 - \omega_3 - \omega_4\right) \\ &\quad \times \left[n_1 \cdot n_3 \cdot (n_4 - n_2) + n_2 \cdot n_4 \cdot (n_3 - n_1)\right] \mathrm{d}\vec{k}_1 \, \mathrm{d}\vec{k}_2 \, \mathrm{d}\vec{k}_3 \\ n(\vec{k}) &= \frac{F(\vec{k})}{\omega}; \quad \omega^2 = g \cdot k \cdot \tanh(kh) \end{aligned} \tag{4.17}$$

where the mapping is symbolically represented by T and the complicated coupling coefficient G contains moving singularities (for a detailed explanation, see Tolman et al. 2005). This integration requires roughly 10^3–10^4 times more computational effort than all other aspects of the wave model combined. Present operational constraints require that the computational effort for the estimation of S_{nl} should be of the same order of magnitude as the remainder of the wave model. This requirement was met with the development of the discrete interaction approximation (DIA, Hasselmann et al. 1985) for S_{nl}. Two decades of experience with the DIA in wave models has identified DIA's significant shortcomings (Tolman et al. 2005). Thus, it is crucially important for the development of new generation wave models to develop an economical yet accurate approximation for S_{nl}. A neural network interaction approximation (NNIA) was explored to achieve this goal (Krasnopolsky et al. 2002; Tolman et al. 2005).

4.4.1 NN Emulation for S_{nl}

A NN can be applied to emulate S_{nl} because the nonlinear interaction term (4.17) is essentially a nonlinear mapping, symbolically represented by T, which relates two vectors F and S (2-D fields in this case). Discretization of S and F (as is necessary in any numerical approach) reduces (4.17) to a continuous mapping of two vectors of finite dimensions. Modern high-resolution wind wave models use discretization on a two-dimensional grid which leads to S and F vector dimensions on the order of $N \sim 1{,}000$. It seems unreasonable to develop an NN emulation with such a high dimensionality ($\sim$1,000 inputs and outputs).

In order to reduce the dimensionality of the NN and convert the mapping (4.17) to a continuous mapping of two finite vectors, two systems of two-dimensional functions (Φ_i and Ψ_q) are used. Each set of functions creates a complete and orthogonal two-dimensional basis. The spectrum F and source function S_{nl} are expanded using these basis functions:

$$F \approx \sum_{t=1}^{n} x_i \Phi_i, \quad S_{\mathrm{nl}} \approx \sum_{q=1}^{m} y_q \Psi_q, \tag{4.18}$$

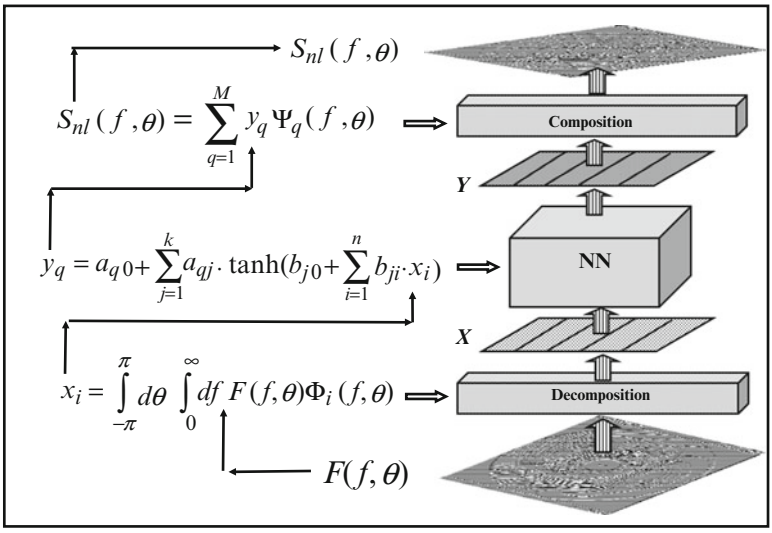

Fig. 4.22 Graphical representation of the NNIA and NNIAE algorithms (Reprinted from Krasnopolsky et al. (2002) with permission from Elsevier)

where for the coefficients of decomposition/composition x_i and y_q are

$$x_i = \iint F\,\Phi_i, \quad y_q = \iint S_{nl}\,\Psi_q, \tag{4.19}$$

where the double integral identifies integration over the spectral space. Now, the developed NN emulation relates vectors of coefficients $\boldsymbol{X}$ and $\boldsymbol{Y}$: $\boldsymbol{Y} = \boldsymbol{T}_{\mathrm{NN}}(\boldsymbol{X})$. Typically, $n = 20$–50 and $m = 100$–150 in Eq. (4.18). Thus, the reduction in dimensionality of the NN emulation is very significant.

To train the NN emulation T_{NN}, a training set has to be created that consists of pairs of the vectors $\boldsymbol{X}$ and $\boldsymbol{Y}$. To produce this training set, a representative set of spectra F has to be generated, and corresponding set of exact interactions S_{nl} has to be calculated using Eq. (4.17). For each pair $(F,\ S_{\mathrm{nl}})_p$ $(p = 1, \ldots, P$ where P is the number of spectra), the corresponding vectors $(\boldsymbol{X},\boldsymbol{Y})_p$ are determined using Eq. (4.19). These pairs of vectors are then used to train the NN T_{NN}. After T_{NN} has been trained, the resulting NNIA algorithm requires three steps: (1) Decompose the input spectrum F by applying Eq. (4.19) to calculate $\boldsymbol{X}$, (2) estimate $\boldsymbol{Y}$ from $\boldsymbol{X}$ using T_{NN}, and (3) compose the output source function S_{nl} from $\boldsymbol{Y}$ using Eq. (4.18). A graphical representation of the NNIA algorithm is shown in Fig. 4.22.

Two approaches have been used for the basis functions. The first (NNIA) uses a mathematical basis (Krasnopolsky et al. 2002). As is usually done in the parametric spectral description of wind waves, separable basis functions have been chosen where the frequency and angular dependencies are separated. The advantage of this choice of basis functions is the simplicity of the basis generation. The disadvantage

Table 4.7 Approximation *RMSE*s (in nondimensional units) and performance (see units in text) for DIA, NNIA, NNIAE, and exact S_{nl} calculation

Algorithm	RMSE	Performance
DIA	0.312	1
NNIA	0.088	4
NNIAE	0.035	7
Original parameterization	0.	$\sim 8. \times 10^5$

is the slow convergence of the decompositions. As an alternative, a second approach to the basis functions choice has been investigated. In this approach, empirical orthogonal functions (EOFs) or principal components (Lorenz 1956; Jolliffe 2002) are used (Tolman et al. 2005).

EOFs constitute a statistically optimal basis. In the case considered, the basis functions Φ_i and Ψ_q are functions of two variables f and θ. The set of spectra F and source terms S_{nl}, which are used for the training of the NN, are also used to generate the EOFs for decomposing F and S_{nl}. When using EOFs, the basis generation procedure is computationally time consuming, with the effort increasing as the resolution of the model increases. However, the basis generation needs to be performed only once for the NN training. Stored results can be used without the need for recalculations in the final NNIA algorithm. The main advantage of EOFs is the fast convergence of the decomposition.

To distinguish between NN algorithms using different basis functions for the decomposition, we use the abbreviation NNIAE for the NN algorithm that used the EOF basis. Table 4.7 shows comparisons of the accuracy and performance of DIA with the two NN emulations NNIA and NNIAE, versus the exact calculation of S_{nl} based on the original parameterization. Approximation errors (*RMSEs*) are calculated in nondimensional units, and performance is measured in DIA calculation times (taken as a unit). The NNIAE is nearly ten times more accurate than DIA. It is also about 10^5 times faster than the original parameterization. As in the case of atmospheric radiation, a careful investigation of parallel runs with the original ENM (the wave model with the exact wave-wave interaction) and the HEM run with the NN emulation has been performed for the final test of the NN emulation (Tolman et al. 2005).

4.4.2 *Validation of NNIAE in the Model and Compound Parameterization for S_{nl}*

The most critical test for any approximation to the nonlinear interactions is the capability of a model using the approximation to produce wave growth under strongly forced conditions (high winds). Under such conditions, large-scale features (in spectral space and in time) of the nonlinear interactions are essential to allow waves to grow higher and longer simultaneously, whereas small-scale features are essential to (locally) stabilize the shape of the spectrum. The NNIAE is therefore

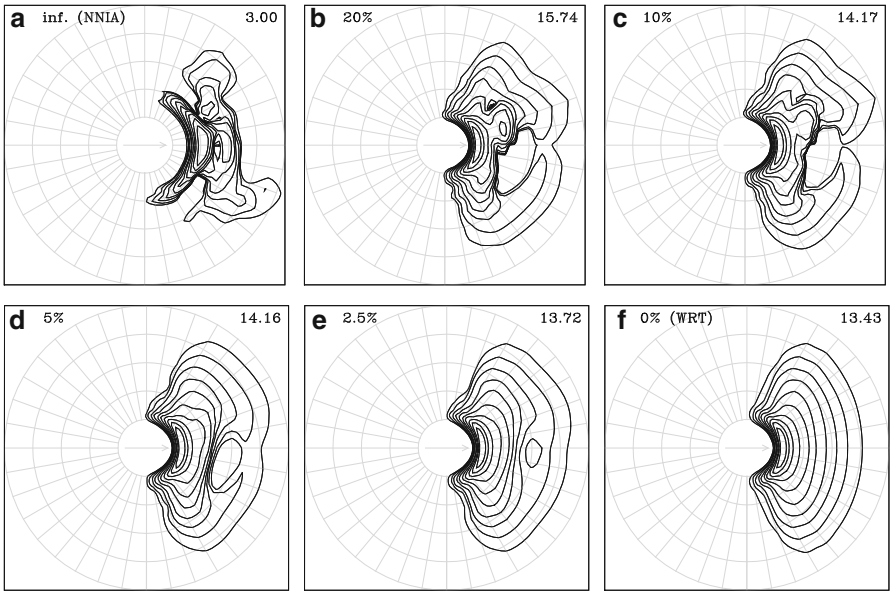

Fig. 4.23 Wave energy spectrum after 24 h of wave growth in WAVEWATCH III. (**a**) The NNIAE approximation to nonlinear interactions and (**b–e**) results obtained with increasingly strict QC in the CP approach. (**f**) Results with a full original nonlinear interaction parameterization WRT. Corresponding wave heights in meters are shown in the *upper right corner* of each panel and the error allowed in the QC in the *upper left corner* of each panel. The frequency increases along the radial direction (Reprinted from Krasnopolsky et al. (2008a) with permission from Elsevier)

trained with and applied to a simple case of wind wave growth, assuming spatially homogeneous conditions. For the training, a limited training set consisting of about 5,000 pairs of input spectra and output exact nonlinear interactions has been used. This training set samples a limited subdomain over the entire input space (space of all possible spectra).

NN interaction approximation (NNIAE) was developed by Tolman et al. (2005) using the limited data set described above. If, however, this approximation is applied in a full wave model, errors in the NNIAE accumulate, and the wave spectrum becomes unrepresentative for the training data set used for the development of the NNIAE. Thus, a balance between the source terms becomes unrealistic. The waves do not grow, and the spectral shape does not resemble the proper solution. The results for this case are presented in Fig. 4.23a.

Figure 4.23 (b–e) shows results of integration of CP in the WAVEWATCH III ocean wind wave model. Panel (f) shows the results of the model with the full exact (original) parameterization for the nonlinear interactions, consisting of a six-dimensional Boltzmann integral (4.17). Contours represent energy levels in polar coordinates in spectral space, with a logarithmic spacing of the contours at intervals that increase by a factor of 2. The consistent and axisymmetric shape of the spectrum is typical for gravity waves actively forced by winds (the so-called wind seas).

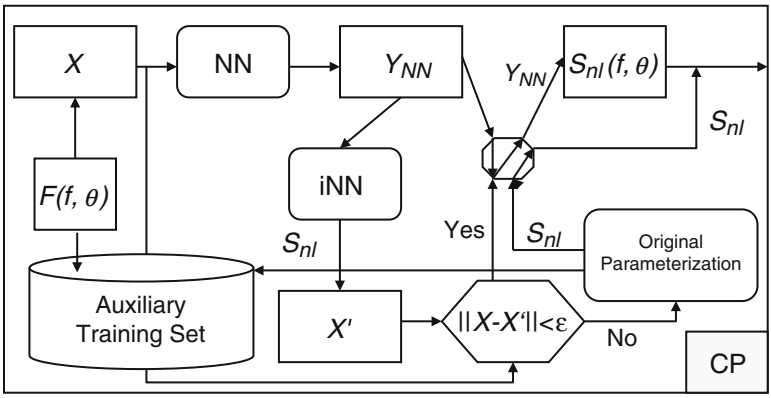

Fig. 4.24 Compound parameterization design for the NNIAE algorithms described in the text. Due to the use of the EOF decomposition and composition procedures, the inverse NN (iNN) and QC block are implemented for composition coefficients X and X'. The auxiliary training set is updated each time when QC requires employing the original parameterization and is used for the follow-up dynamical adjustment of the NN emulation (Reprinted from Krasnopolsky et al. (2008a) with permission from Elsevier)

An experimental CP for the nonlinear wave-wave interaction in the WAVEWATCH III ocean wind wave model was developed. It is illustrated in Fig. 4.24. This CP uses the domain check approach (see Sect. 3.4) with an inverse NN (iNN in the figure). Subsequently, when the CP presented in Fig. 4.24 was developed using the same data set and implemented, integration is sufficiently stabilized to allow for realistic wave growth (Fig. 4.23b, compare wave heights in upper right corners of each panel; see also Tolman and Krasnopolsky (2004)). Following the reduction of the errors allowed in the QC part of the CP (a more restrictive QC), the accuracy of the model clearly increases (Fig. 4.23c, d, and e).

An approach to describe the nonlinear wave-wave interactions most effectively in terms of computational efficiency and accuracy may well require a more complex CP than the CP approaches that have been discussed so far. The initial data decompositions using EOFs introduce a truncation error in the corresponding description of the wave spectrum. By definition, such truncations tend to filter out small-scale fluctuations, which in many processes can be considered as noise. For the wave growth process, however, these scales are essential to stabilize the spectral shape during model integration. It remains to be seen if this part of the solution can ever be described effectively using the NN approach. Figure 4.23 shows that a simple CP can circumvent this issue. Also, the small-scale processes in the nonlinear interactions could be modeled explicitly as a local diffusion process; however, the computational effort is orders of magnitude less than direct computation of nonlinear interactions at all spectral scales, because the latter involves a six-dimensional integration over the entire spectral space. Tentatively, a more complex CP approach for nonlinear wave-wave interactions could therefore be based on an NN approach for larger spectral scales and local diffusion to describe the smaller scales (to be trained simultaneously), combined with explicit QC to add robustness.

4.5 Discussion

4.5.1 Summary and Advantages of the Hybrid Modeling Approach

In this chapter, we introduced a new hybrid paradigm in environmental numerical modeling. Within the framework of this paradigm, a new type of ENM – an HEM – based on a synergetic combination of deterministic modeling and statistical learning within an HEM (using the NN technique) is introduced. This approach uses NNs to develop new fast model components and highly accurate and fast emulations of existing model physics. The results show that:

1. There exists the conceptual and practical possibility of developing HEMs with accurate and fast NN model components, which preserve the integrity and all the detailed features in the original ENM.
2. NN emulations of existing model physics parameterizations (Krasnopolsky et al. 2002, 2005a, 2008b, 2010, 2012) are functionally identical to the original physical parameterizations, due to the capability of NN techniques to accurately emulate complex systems (mappings) like parameterizations of the model physics. This capability allows the integrity and level of functional complexity of the parameterizations of the model physics to be preserved. As a result, an HGCM, using these NN emulations, produces climate simulations and weather forecasts that are practically identical to those of the original GCM. It is noteworthy that the NN emulations that were developed have the same inputs and outputs as the original parameterizations and are used precisely as their functional substitutes within the model.
3. Accurate NN emulations are robust and very fast (10–10^5 times faster than the original parameterization), so the significant speedup of HEM calculations can be achieved without compromising accuracy.
4. New computationally efficient NN model components (parameterizations) can be developed based on learning from observed data or data simulated by higher-resolution models (Krasnopolsky et al. 2011).
5. Statistical (NN) components can be successfully combined with deterministic model components within the HEM, so their synergy can be efficiently used for environmental and climate modeling without any negative impacts on simulation quality.
6. This productive synergy or new combination of state-of-the-art deterministic and NN emulation approaches leads to new opportunities in using HEMs for environmental and climate simulations and prediction. For example, new more sophisticated parameterizations, or even "superparameterizations" such as a CRM, that are extremely time consuming or even computationally prohibitive if used in their original form will become computationally "affordable" in ENMs when using their accurate and computationally much more efficient NN emulations in HEMs.

7. The stochastic nature of some of the components in the model physics can be adequately represented using ensemble of NNs to represent or emulate the stochastic component of the model physics.

4.5.2 *Limitations of the Current Hybrid Modeling Framework and Possible Solutions*

The development of NN emulations and parameterizations, the core of the hybrid modeling approach, depends significantly on our ability to generate a representative training set to avoid using NNs for extrapolation far beyond the domain covered by the training set (see Sect. 2.3.3). Because of the high dimensionality of the input domain that is often several hundred or more, it is rather difficult to cover the entire domain, especially the "far corners" associated with rare events, even when we use simulated data for NN training. Another related problem emerges if NN emulations should be developed for a nonstationary environmental or climate system that changes with time. This means that the domain configuration for a climate simulation may evolve when using a future climate change scenario. In both situations, the emulating NN may be forced to extrapolate beyond its generalization ability leading to errors in NN outputs and result in simulation errors in the corresponding HEM.

Two new techniques are being developed to accommodate the limitations outlined above and to supplement the NN emulation approach for long-term climate change simulations and other applications: (1) a *compound parameterization* and (2) an NN *dynamical adjustment* (DA) (Krasnopolsky and Fox-Rabinovitz 2006a, b). CP has been described in detail in Sects. 4.2.2, subsection "Compound and adjustable parameterizations, quality control of NN emulations", 4.3.5, and 4.4.2. In the following, the DA approach is briefly outlined.

During a routine HEM simulation with CP, the QC block determines (at each time step of integration and at each grid point based on some criteria) whether the NN emulation or the original parameterization has to be used to generate physical parameters (parameterization outputs). When the original parameterization is used instead of the NN emulation, its inputs and outputs are saved to further adjust the NN emulation. After accumulating a sufficient number of these records, a DA of the NN emulation is performed by a short NN retraining using the accumulated input/output records. Thus, the upgraded NN emulation becomes dynamically adjusted to the changes and/or new events/states produced by the complex environmental or climate system. Obviously, DA can also solve the problem of extreme or rare events.

Actually, DA can be performed after each new training record is generated in real time (or online) during the model run. This online DA can be implemented using sequential or the online NN training method described in Sect. 2.3.7, subsection "Batch training and sequential training."

References

Chevallier F (2005) Comments on new approach to calculation of atmospheric model physics: accurate and fast neural network emulation of longwave radiation in a climate model. Mon Weather Rev 133:3721–3723

Chevallier F, Chéruy F, Scott NA, Chedin A (1998) A neural network approach for a fast and accurate computation of longwave radiative budget. J Appl Meteorol 37:1385–1397

Chevallier F, Morcrette J-J, Chéruy F, Scott NA (2000) Use of a neural-network-based longwave radiative transfer scheme in the EMCWF atmospheric model. Q J Roy Meteorol Soc 126:761–776

Chou M-D, Suarez MJ, Liang X-Z, Yan MM-H (2001) A thermal infrared radiation parameterization for atmospheric studies. In: Suarez MJ (ed) Tech report series on global modeling and data assimilation, vol 19, NASA/TM-2001-104606. National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt

Claussen M (2001) Earth system models. In: Ehlers E, Kraft T (eds) Understanding the Earth system: compartments, processes and interactions. Springer, Heidelberg/Berlin/New York

J Climate (1998) 11, No 6 (the special issue). http://journals.ametsoc.org/toc/clim/11/6

Clough SA, Shephard MW, Mlawer EJ, Delamere JS, Iacono MJ, Cady-Pereira K, Boukabara S, Brown PD (2005) Atmospheric radiative transfer modeling: a summary of the AER codes. J Quant Spectrosc Radiat 91:233–244. doi:10.1016/j.jqsrt.2004.05.058

Collins WD (2001) Parameterization of generalized cloud overlap for radiative calculations in general circulation models. J Atmos Sci 58:3224–3242

Collins WD, Hackney JK, Edwards DP (2002) A new parameterization for infrared emission and absorption by water vapor in the National Center for Atmospheric Research Community Atmosphere Model. J Geophys Res 107:1–20

Côté J, Desmarais J-G, Gravel S, Méthot A, Patoine A, Roch M, Staniforth A (1998a) The operational CMC-MRB global environmental multiscale (GEM) model Part II: Mesoscale results. Mon Weather Rev 126:1397–1418

Côté J, Gravel S, Méthot A, Patoine A, Roch M, Staniforth A (1998b) The operational CMC-MRB global environmental multiscale (GEM) model Part I: design considerations and formulation. Mon Weather Rev 126:1373–1395

Cox PM, Betts RA, Jones CD, Spall SA, Totterdell IJ (2000) Will carbon cycle feedbacks amplify global warming in the 21st century. Nature 408:184–187

Foley JA, Levis S, Prentice IC, Pollard D, Thompson SL (1998) Coupling dynamic models of climate and vegetation. Glob Change Biol 4:561–580

Grabowski WW (2001) Coupling cloud processes with the large-scale dynamics using the Cloud-Resolving Convection Parameterization (CRCP). J Atmos Sci 58:978–997

Grassl H (2000) Status and improvements of coupled general circulation models. Science 288:1991–1997

Hasselmann S, Hasselmann K (1985) Computations and parameterizations of the nonlinear energy transfer in a gravity wave spectrum Part I: A new method for efficient computations of the exact nonlinear transfer integral. J Phys Oceanogr 15:1369–1377

Hasselmann S et al (1985) Computations and parameterizations of the nonlinear energy transfer in a gravity wave spectrum Part II: Parameterization of the nonlinear transfer for application in wave models. J Phys Oceanogr 15:1378–1391

Iacono MJ, Mlawer EJ, Clough SA, Morcrette J-J (2000) Impact of an improved longwave radiation model, RRTM, on the energy budget and thermodynamic properties of the NCAR community climate model, CCM3. J Geophys Res 105(D11):14873–14890

Jolliffe IT (2002) Principal component analysis. Springer, New York

Kalnay E et al (1996) The NCEP/NCAR 40-year reanalysis project. Bull Am Meteorol Soc 77:437–471

Khairoutdinov MF, Randall DA (2001) A cloud resolving model as a cloud parameterization in the NCAR Community Climate System Model: preliminary results. Geophys Res Lett 28:3617–3620

Khairoutdinov MF, Randall DA, DeMotte C (2005) Simulations of the atmospheric general circulation using a cloud-resolving model as a super-parameterization of physical processes. J Atmos Sci 60:607–625

Kistler R et al (2001) The NCEP-NCAR 50-year reanalysis. Bull Am Meteorol Soc 82:247–268

Krasnopolsky V (1996) A neural network forward model for direct assimilation of SSM/I brightness temperatures into atmospheric models. Working group on numerical experimentation blue book. 1.29–1.30, Camp Spring, MD. http://polar.ncep.noaa.gov/mmab/papers/tn134/OMB134.pdf

Krasnopolsky V (1997) A neural network based forward model for direct assimilation of SSM/I brightness temperatures. Tech note, OMB contribution No 140, NCEP/NOAA Camp Spring, MD. http://polar.ncep.noaa.gov/mmab/papers/tn140/OMB140.pdf

Krasnopolsky VM, Fox-Rabinovitz MS (2006a) Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. Neural Netw 19:122–134

Krasnopolsky VM, Fox-Rabinovitz MS (2006b) A new synergetic paradigm in environmental numerical modeling: hybrid models combining deterministic and machine learning components. Ecol Model 191:5–18

Krasnopolsky VM, Chalikov DV, Tolman HL (2002) A neural network technique to improve computational efficiency of numerical oceanic models. Ocean Model 4:363–383

Krasnopolsky VM, Fox-Rabinovitz MS, Chalikov DV (2005a) New approach to calculation of atmospheric model physics: accurate and fast neural network emulation of long wave radiation in a climate model. Mon Weather Rev 133:1370–1383

Krasnopolsky VM, Fox-Rabinovitz MS, Chalikov DV (2005b) Reply. Mon Weather Rev 133:3724–3729

Krasnopolsky VM, Fox-Rabinovitz MS, Tolman HL, Belochitski A (2008a) Neural network approach for robust and fast calculation of physical processes in numerical environmental models: compound parameterization with a quality control of larger errors. Neural Netw 21:535–543

Krasnopolsky VM, Fox-Rabinovitz MS, Belochitski A (2008b) Decadal climate simulations using accurate and fast neural network emulation of full, long- and short wave, radiation. Mon Weather Rev 136:3683–3695. doi:10.1175/2008MWR2385.1

Krasnopolsky VM, Fox-Rabinovitz MS, Hou YT, Lord SJ, Belochitski A (2010) Accurate and fast neural network emulations of model radiation for the NCEP coupled climate forecast system: climate simulations and seasonal predictions. Mon Weather Rev 138:1822–1842. doi:10.1175/2009MWR3149.1

Krasnopolsky V, Fox-Rabinovitz M, Belochitski A, Rasch P, Blossey P, Kogan Y (2011) Development of neural network convection parameterizations for climate and NWP models using Cloud Resolving Model simulations. NCEP office note 469 Camp Spring, MD. http://www.emc.ncep.noaa.gov/officenotes/newernotes/on469.pdf

Krasnopolsky V, Belochitski A, Hou YT, Lord SJ, Yang F (2012) Accurate and fast neural network emulations of long and short wave radiation for the NCEP global forecast system model. NCEP office note 471. http://www.emc.ncep.noaa.gov/officenotes/newernotes/on471.pdf

Krasnopolsky VM, Fox-Rabinovitz MS, Belochitski AA (2013) Using ensemble of neural networks to learn stochastic convection parameterizations for climate and numerical weather prediction models from data simulated by a cloud resolving model. Adv Artif Neural Syst 2013:13 pp. Article ID 485913, doi:10.1155/2013/485913. http://dx.doi.org/10.1155/2013/485913

Krishnamurti TN et al (2003) Improved skill for the anomaly correlation of geopotential heights at 500 hPa. Mon Weather Rev 131:1082–1102

Krueger SK (1988) Numerical simulation of tropical clouds and their interaction with the subcloud layer. J Atmos Sci 45:2221–2250

Lacis AA, Oinas V (1991) A description of the correlated k-distribution method for modeling nongray gaseous absorption, thermal emission and multiple scattering in vertically inhomogeneous atmospheres. J Geophys Res 96:9027–9063

Li S, Hsieh WW, Wu A (2005) Hybrid coupled modeling of the tropical Pacific using neural networks. J Geophys Res. doi:10.1029/2004JC002595

Lorenz EN (1956) Empirical orthogonal functions and statistical weather prediction. Science reports, No 1, Statistical Forecasting Project. M.I.T., Cambridge, MA

Manners J, Thelen J-C, Petch J, Hill P, Edwards JM (2009) Two fast radiative transfer methods to improve the temporal sampling of clouds in NWP and climate models. Q J Roy Meteorol Soc 135:457–468. doi:10.1002/qj.385

Miura H, Tomita H, Casino T, Riga S, Satoh M, Matsuno T (2005) A climate sensitivity test using a global cloud resolving model under an aqua planet condition. Geophys Res Lett. doi:10.1029/2005GL023672

Mlawer EJ, Taubman SJ, Brown PD, Iacono MJ, Clough SA (1997) Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave. J Geophys Res 102(D14):16663–16682

Morcrette J-J, Bechtold P, Beljaars A, Benedetti A, Bonet A, Doblas-Reyes F, Hague J, Hamrud M, Haseler J, Kaiser JW, Leutbecher M, Mozdzynski G, Razinger M, Salmond D, Serrar S, Suttie M, Tompkins A, Untch A, Weisheimer A (2007) Recent advances in radiation transfer parameterizations. ECMWF Tech Memorandum No 539, October 18, 2007

Morcrette J-J, Mozdzynski G, Leutbecher M (2008) A reduced radiation grid for the ECMWF Integrated Forecasting System. Mon Weather Rev 136:4760–4772. doi:10.1175/2008MWR2590.1

Nguyen D, Widrow B (1990) Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In: Proceedings of the international joint conference of neural networks, vol 3, San Diego, CA, USA, 17–21 June, pp 21–26

Peixioto JP, Oort AH (1992) Physics of climate. Springer, New York

Randall D, Khairoutdinov M, Arakawa A, Grabowski W (2003) Breaking the cloud-parameterization deadlock. Bull Am Meteorol Soc 84:1547–1564

Rasch PJ, Feichter J, Law K, Mahowald N, Penner J, Benkovitz C, Genthon C, Giannakopoulos C, Kasibhatla P, Koch D, Levy H, Maki T, Prather M, Roberts DL, Roelofs G-J, Stevenson D, Stockwell Z, Taguchi S, Kritz M, Chipperfield M, Baldocchi D, McMurry P, Barrie L, Balkanski Y, Chatfield R, Kjellstrom E, Lawrence M, Lee HN, Lelieveld J, Noone KJ, Seinfeld J, Stenchikov G, Schwartz S, Walcek C, Williamson D (2000) A comparison of scavenging and deposition processes in global models: results from the WCRP Cambridge workshop of 1995. Tellus 52:1025–1056

Saha S et al (2010) The NCEP climate forecast system reanalysis. Bull Am Meteorol Soc 91:1015–1057

Satoh M, Tomita H, Miura H, Iga S, Nauno T (2005) Development of a global cloud resolving model – a multi-scale structure of tropical convection. J Earth Simul 3:11–19

Schellnhuber HJ (1999) "Earth system" analysis and the second Copernican revolution. Nature 402:C19–C28

Tang Y, Hsieh WW (2003) ENSO simulation and prediction in a hybrid coupled model with data assimilation. J Meteorol Soc Jpn 81:1–19

Tolman HL (2002) User manual and system documentation of WAVEWATCH III version 2.22. Tech note 222, NOAA/NWS/NCEP/MMAB Camp Spring, MD. http://polar.ncep.noaa.gov/mmab/papers/tn222/MMAB_222.pdf

Tolman HL, Krasnopolsky VM (2004) Nonlinear interactions in practical wind wave models. In: Proceedings of 8th international workshop on wave hindcasting and forecasting, Turtle Bay, Hawaii, 2004, CD-ROM, E.1

Tolman HL, Krasnopolsky VM, Chalikov DV (2005) Neural network approximations for nonlinear interactions in wind wave spectra: direct mapping for wind seas in deep water. Ocean Model 8:253–278

Washington WM, Williamson DL (1977) A description of NCAR GCM's in general circulation models of the atmospheres methods in computational physics. J Chang Ed 17:111–172

Chapter 5
NN Ensembles and Their Applications

Science has, as its whole purpose, the rendering of the physical world understandable and beautiful. Without this you have only tables and statistics.

– Julius Robert Oppenheimer

It is better to be roughly right than precisely wrong.

– John Maynard Keynes

Abstract In this chapter, various neural network (NN) ensemble applications including applications in data assimilation systems, nonlinear multi-model ensembles, ensembles with perturbed model physics, and others are introduced and discussed. It is shown that in many cases, NN ensemble approaches provide a better and more adequate emulation of the complex nonlinear mappings than a single NN. In this chapter, an NN ensemble approach is applied to introduce analytic approximations for highly complex functional dependencies and mappings between the model variables in an oceanic data assimilation system which enables 3-D assimilation of surface 2-D variables like the surface elevation. An NN ensemble approach is applied to derive a nonlinear multi-model ensemble for improving 24-h precipitation forecasts over the continental US (ConUS). Different possibilities for using the NN emulation technique in combination with the NN ensemble approach for generating stochastic or perturbed model physics and ensembles with perturbed model physics are considered. The chapter contains an extensive list of references giving extended background and further detail to the interested reader on each examined topic. It can serve as a textbook and an introductory reading for students and beginning and advanced investigators interested in learning how to apply the NN ensemble technique to various problems.

In Sect. 2.4.5 a specific capability of nonlinear statistical models (e.g., NNs) to provide multiple solutions for a particular problem was discussed. Also, an NN ensemble approach was introduced. This approach allows us to turn what is

V.M. Krasnopolsky, *The Application of Neural Networks in the Earth System Sciences*, Atmospheric and Oceanographic Sciences Library 46, DOI 10.1007/978-94-007-6073-8_5,

sometimes considered a disadvantage of nonlinear models (existence of multiple solutions) into an advantage. The NN ensemble approach in many cases provides a better and more adequate emulation of the mapping (2.1) than a single NN. For example, in Sect. 2.4.6, it was shown that the NN ensemble approach provides an adequate emulation tool in the case when the mapping (2.1) is a stochastic mapping (see Sect. 2.2.4).

In Sect. 4.3.6 an NN ensemble approach was used to develop a stochastic convection parameterization for GCMs. In this chapter several other applications of the NN ensemble approach in atmospheric and oceanic models are introduced and discussed. In Sect. 5.1, an NN approach is applied to introduce analytic approximations for highly complex functional dependencies and mappings between the model variables. This generic NN application is introduced in the context of a particular application, NN emulations for the SSH observation operator (Krasnopolsky et al. 2006; Krasnopolsky 2007a). This NN application, designed to work in DAS (see Sect. 3.1.2) and in sensitivity and error analysis, uses the NN emulation and its Jacobian. Thus, it requires the NN emulation Jacobian of satisfactory quality. The NN ensemble approach is introduced here to improve the quality of the NN emulation and NN Jacobian.

In Sect. 5.2, an NN ensemble approach is applied to derive a nonlinear multimodel ensemble (MME) for improving 24-h precipitation forecasts over the continental US. The nonlinear ensemble approach presented there allowed us to account for nonlinear correlations between ensemble members and produce an "optimal" forecast represented by a nonlinear NN ensemble mean.

In Sect. 5.3, different possibilities for using the NN emulation technique in combination with the NN ensemble approach for generating stochastic model physics or perturbed model physics are discussed. Several applications of NN-based perturbed model physics in ensemble prediction systems (EPS) are also discussed.

5.1 Using NN Emulations of Dependencies Between Model Variables in DAS

The output of any complex ES numerical model, such as models for climate simulations or NWP, contains a great deal of data in the form of 2- and 3-D high-resolution numerical fields of prognostic and diagnostic atmospheric and ocean state variables. This output contains, in an implicit form, highly complex physical relationships and statistical correlations between the state variables of the model, which mathematically can be represented as functional dependencies and mappings (2.1). These relationships are governed by the physics and dynamics of the numerical model that are used for the simulations. A clear understanding of these underlying nonlinear dependencies is a matter of a great scientific interest and practical importance. For example, these dependencies are essential for assimilating variables in DASs (see Sect. 3.1.2). These functions and mappings, if they were derived in analytical form, could also be used for efficient model output compression, archiving, and dissemination and for sensitivity studies and error analysis.

When 2-D observations like surface wind, surface currents, or sea surface elevation are assimilated into an atmospheric or oceanic DAS, the impact of these data in the DAS is localized at the vertical level where they are assimilated because there is usually no explicit mechanism in the DAS to propagate the impact of these data to other vertical levels and to other variables. Usually, this propagation occurs later, with a delay, during the integration of the model, in accordance with dependencies determined by the model physics and dynamics. Recently, several attempts have been made to extract these dependencies from model simulations (Mellor and Ezer 1991) or observed data (Guinehut et al. 2004) in a simplified linear form for use in an ocean DAS to allow for 3-D assimilation of the 2-D surface data. However, these simplified and generalized linear dependencies that are often derived from local data sets do not properly represent the complicated nonlinear relationships (mappings) between the model variables. If we were able to extract or emulate these mappings in a simple, but not overly simplified and yet in an adequately nonlinear analytical form, they could be used in the DAS to facilitate a more effective 3-D assimilation of the 2-D surface data.

The existence of a generic technique that allows the extraction of these nonlinear functions and mappings in a compact analytical form would also greatly facilitate the use of model output in qualitative and quantitative studies. It is only recently that initial steps have been taken to use the NN technique to accomplish this objective (Tang and Hsieh 2003; Krasnopolsky et al. 2006).

5.1.1 SSH Mapping and Its NN Emulation

Sea surface height (SSH), η, is one of the prognostic variables in ocean circulation models. The particular ocean model used in this example is the Hybrid Coordinate Ocean Model (HYCOM). This model is a primitive equation model that uses generalized hybrid coordinates (isopycnal/terrain following (σ)/z-level) in the vertical (see Bleck 2002 for details). The hybrid coordinate extends the applicability of the traditional isopycnal (levels of constant water density) coordinate circulation models to shallow coastal seas and to unstratified areas of the ocean. The vertical coordinate utilized in HYCOM is discussed in Chassignet et al. (2003). The particular version of HYCOM used here has a domain that covers the entire Atlantic Ocean with an average resolution of $1/3^\circ \times 1/3^\circ$ in the horizontal and 25 levels in the vertical.

Since the reduced model physics has a 1-D vertical structure, it was assumed that SSH, η, at a particular model grid point (i.e., at a particular horizontal location) depends only on the vector of state variables, X, at the same horizontal location and the same time. Therefore, this dependence (a target mapping, or so-called observation operator) can be written as

$$\eta = \phi(X), \tag{5.1}$$

where ϕ is a nonlinear continuous function and X is a vector that represents a complete set of state variables, which determine η. In this particular case the vector

Table 5.1 Periods covered by training, validation, and test data sets and their sizes

Set	Beginning date (Julian day, year)	End date (Julian day, year)	Size, N(number of profiles)
Training	303, 2002	52, 2004	563,259
Validation	303, 2002	52, 2004	563,259
Test	53, 2004	291, 2004	563,259

X was selected as $X = \{I, \theta, z_{\text{mix}}\}$, where I is the vector (a vertical profile) of the interfaces (the vertical coordinates used in HYCOM), θ is the profile of potential temperature, and z_{mix} is the depth of the ocean mixed layer, for a total of 50 variables. This set of variables represents or is used as a proxy for the physics of the deep ocean. Therefore, the mapping (5.1) with this particular selection of components for the vector X will not be applicable in coastal areas (where the depth is less than 250–500 m). In the coastal areas a different set of state variables should be selected. All the statistics presented later in this chapter were calculated using a test set where coastal areas were excluded.

The NN technique has been applied to derive an analytical NN emulation for the relationship between model state variables, X, and η, or the NN observation operator,

$$\eta_{\text{NN}} = \phi_{\text{NN}}(X). \tag{5.2}$$

NN weights have been trained using the simulated model fields, which are treated as error-free data. A simulation that covers almost 2 years, from Julian dates 303, 2002 to 291, 2004, was used to create training, validation, and test data sets. The periods covered by these data sets and their sizes are shown in Table 5.1. Each data set consists of records $\{\eta_i, X_i\}_{i=1,\ldots,N}$ collocated in space and time and uniformly distributed over the model domain.

As mentioned earlier, the accuracy of the NN emulation is evaluated over the model domain (excluding coastal areas) using the test set described in Table 5.1. All trained NNs have 50 inputs and 1 output in accordance with the dimensionalities of the target mapping (5.1). The number of hidden neurons k was varied from 3 to 30. There is no significant and consistent improvement in the approximation RMSE after k reaches values of 5–10; any further improvement does not exceed 0.25 cm. Therefore, to limit NN complexity and improve its interpolation abilities (see Sects. 2.4.3 and 2.5), only NNs with $k \leq 10$ were used in the following investigation.

In the next test applied to the NN emulation, the last day of the entire simulation (291, 2004) was selected. This day is separated in time by about 8 months from the last day of the simulation used for training and validation (52, 2004). The NN emulation (5.2) was applied over the entire domain to generate the 2-D field of η_{NN}. This field was compared with the corresponding field of η generated by the model. The difference between two fields is shown in Fig. 5.1.

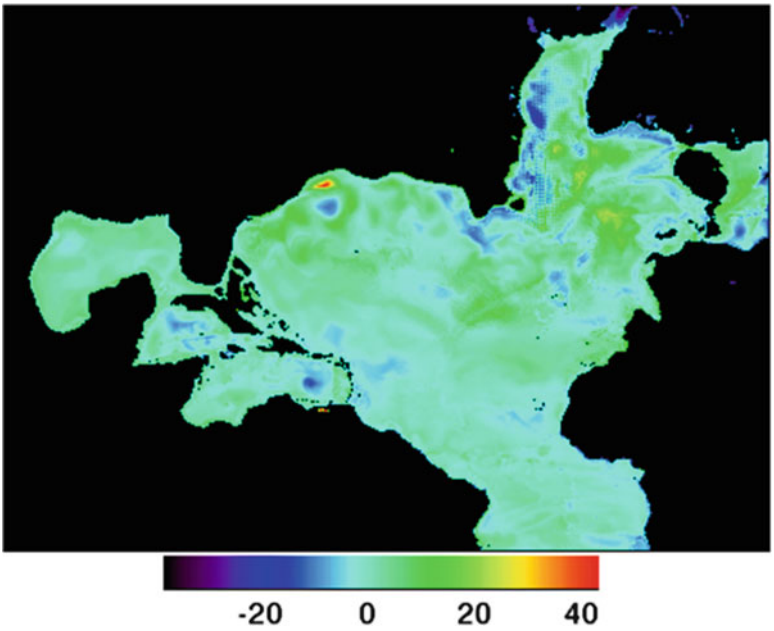

Fig. 5.1 The difference (in cm) between SSH field, η_{NN}, emulated by NN (5.2) with five hidden neurons, and exact SSH field, η. The entire domain of the Atlantic model is shown. The figure is plotted in the model internal x-y coordinates

With the exception of several spots (most of them still close to coastal areas), the differences do not exceed ± 10 cm. The accuracy of the NN emulation over the entire domain shown in Fig. 5.1 is satisfactory based on the intended application; the bias is about 1 cm and the RMSE is about 4.7 cm. The accuracy of the NN observation operator (5.2) is considered to be satisfactory because the NN emulation will be used in the DAS together with satellite measurements of SSH that have accuracy on the order of 5 cm or less.

The accuracy of the NN observation operator may be improved using an NN ensemble approach (see Sect. 5.1.2). The use of the NN emulation in DAS is conditioned by the quality of the NN Jacobian. The accuracy of the NN Jacobian and the possibility of improving this accuracy by using NN ensembles are also discussed in the next section.

5.1.2 NN Ensembles for Improving NN Observation Operator Accuracies and Reducing NN Jacobian Uncertainties

As mentioned in Sects. 2.4.3 and 2.5, it is desirable to keep the NN emulation complexity (the number of hidden neurons) at a minimum in order to improve

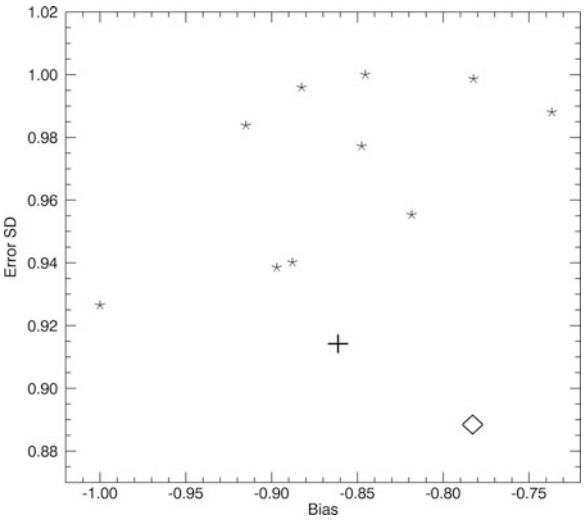

Fig. 5.2 The random part of the emulation error (the standard deviation, SD, of the error) normalized to the maximum member error (*the vertical axis*) and the systematic error (bias) also normalized to the maximum member error (*the horizontal axis*). Each ensemble member is represented by a star, the conservative ensemble average by the cross, and the nonlinear ensemble using the averaging NN by *diamond* at the figure

NN generalization (interpolation) ability and the stability of the NN Jacobian (see also Sect. 2.4.4); however, minimization of the NN complexity reduces the approximation accuracy of NN emulations. Using an NN ensemble approach (see Sect. 2.4.5) is a way to obtain a reasonable balance.

In the context of the problem described in the previous section, the NN ensemble approach leads to the following solution. The complexity of the NN emulation (5.2) was limited; only three hidden neurons were allowed. Then ten NN observation operators (5.2) with the same number of hidden neurons were trained using different initializations for the NN weights. As a result, an NN ensemble that consists of ten members, 10 NN observation operators of identical architecture (50 inputs, 1 output, and 3 neurons in 1 hidden layer) but with different weights, different approximation accuracies, and different Jacobians were created. When four or five neurons in one hidden layer were selected for the NN architecture, the obtained results were similar to those presented below.

NN Ensembles for Improving the Accuracy of an NN Emulation: Linear Versus Nonlinear Ensembles

After the NN ensemble was created, each NN member (a particular realization of the NN observation operator (5.2)) was applied to the test set, and the error statistics for each NN member were calculated and are plotted in Fig. 5.2. The vertical axis of the figure shows the random part of the approximation error (the SD of the error), and the horizontal axis is the value of the systematic error (bias). Both errors are normalized to the corresponding maximum member error (member bias or error SD). Each ensemble member is represented in this figure by a star. The spread of

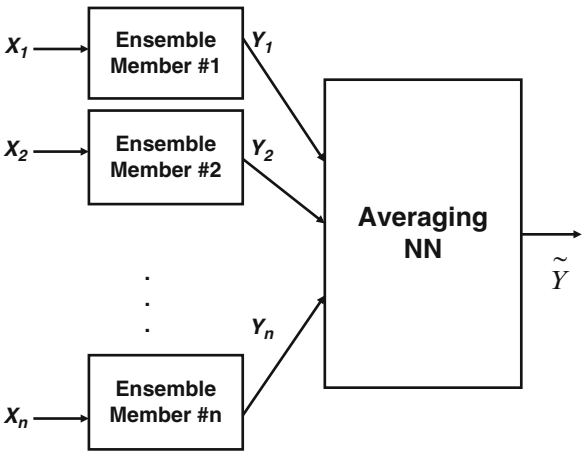

Fig. 5.3 Schematic representation of a nonlinear ensemble that uses an averaging NN. A tilde over the averaging NN output $\tilde{Y}$ emphasizes that a nonlinear ensemble average is produced. X is an input vector of the emulation (5.2)

the ensemble members is significant in this figure. The systematic error changes about 25 % and the random error about 10 % for different members.

The next step was to produce the ensemble average, which can be produced in different ways (Barai and Reich 1999). The first averaging approach used here is the simplest linear method of ensemble averaging – a conservative ensemble (Barai and Reich 1999). Each of the ten NN ensemble members was applied to the test set record by record. Thus, for each record (set of inputs), ten NN outputs were produced. Then the mean value (in a statistical sense) of these ten values was calculated and compared to the exact output to calculate the ensemble statistics represented by the cross in Fig. 5.2. For the conservative ensemble, the ensemble bias is equal to the mean bias of the members as expected when using this simple linear method of calculating the ensemble average. Figure 5.2 also illustrates the fact that ensemble approaches are effective in reducing random errors; it shows that in this case the ensemble random error is less than the random error of any of the ensemble members. The reduction in systematic ($\sim$15 %) and random ($\sim$9 %) errors with respect to the maximum single member errors is moderate but still significant.

The conservative ensemble is simple; however, it is linear and therefore completely neglects nonlinear correlations and dependencies between ensemble members. To estimate the contribution of aforementioned nonlinear correlations and to use them for improving ensemble averaging, we developed a nonlinear ensemble that uses an additional averaging NN to calculate the ensemble average. This approach is illustrated schematically in Fig. 5.3.

The inputs to the averaging NN are composed of the same outputs from the same ensemble member NNs that are used by the conservative ensemble. The number of inputs to the averaging NN is equal to the number of ensemble members (10 in this case) multiplied by the number of outputs in a single ensemble member NN (one, in our case). It has the same single output as a single ensemble member NN in this

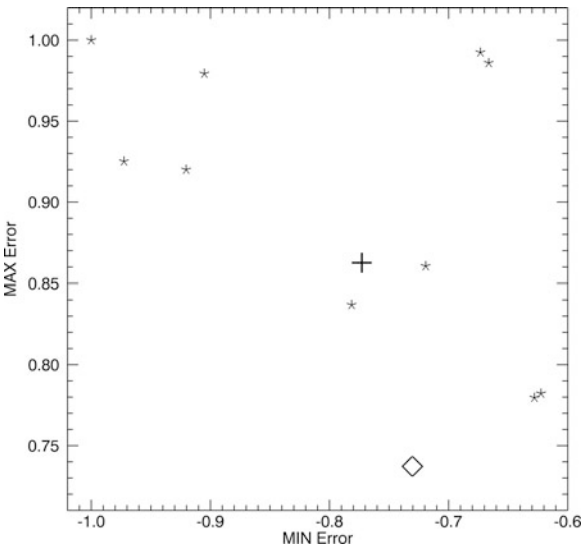

Fig. 5.4 Extreme outliers' statistics. The *vertical axis* shows the largest positive (or maximum) and the *horizontal axis* the largest negative (or minimum) emulation error over the entire test set. Each ensemble member is represented by a *star*, the conservative ensemble by the *cross*, and the nonlinear NN ensemble by the *diamond* at the figure

particular case. The averaging NN was trained using training and validation sets prepared on the same basis as the training and validation sets used for training the ensemble member NNs. The test statistics presented here were calculated using the test set.

The result for the nonlinear ensemble using the averaging NN is shown in Fig. 5.2 by a diamond. It shows that the magnitude of the nonlinear correlations between ensemble members is significant and can be successfully used to improve ensemble accuracy. A comparison of the positions of the cross and the diamond in Fig. 5.2 shows that, compared to the conservative ensemble, the nonlinear ensemble gives an additional improvement in bias on the order of 10 %. The nonlinear ensemble bias is close to the minimum ensemble member bias. An additional improvement in the random error is a bit smaller (about 5 %) but still significant.

Figure 5.4 shows the statistics for extreme outliers. When each ensemble member NN is applied to the test set, the NN produces an output with an error for each record. Among all these errors, there exist one largest negative (or minimum) error and one largest positive (or maximum) error, or two extreme outliers that demonstrate the worst-case scenarios that can be expected from this particular NN emulation. These two extreme outliers for each NN member are represented in Fig. 5.4 by a star. Each ensemble also generates these two extreme outliers, shown by the cross for the conservative ensemble and the diamond for the nonlinear NN ensemble in Fig. 5.4.

Figure 5.4 shows that the NN ensemble approach is an effective tool in reducing extreme outliers (by ∼25 %). The significant improvement introduced by the nonlinear ensemble (diamond) supports this conclusion. This technique was also applied with similar results to ensemble of NN emulations developed for the radiation parameterizations (Krasnopolsky 2007b) and the NN multi-model ensemble (see Sect. 5.2).

NN Ensembles for Reducing the Uncertainty of the NN Jacobian

The NN emulation (5.2) can be used in the ocean DAS to enhance assimilating SSH and to improve the propagation of the surface SSH signal to other vertical levels and other variables. In the ocean DAS, the increment of SSH, $\Delta\eta$, is calculated using the NN Jacobian $\left\{\frac{\partial\phi_{\mathrm{NN}}}{\partial X_i}\right\}_{i=1,\ldots,n}$,

$$\Delta\eta_{\mathrm{NN}} = \sum_{i=1}^{n} \left.\frac{\partial\phi_{\mathrm{NN}}}{\partial X_i}\right|_{X=X^0} \cdot \Delta X_i, \tag{5.3}$$

where ΔX_i are increments of the state variables, X^0 is a vector of initial values of state variables, and n is the dimensionality of the vector X, i.e., the number of inputs in the NN emulation (5.2). Then the calculated $\Delta\eta_{\mathrm{NN}}$ is compared with the observed $\Delta\eta_{\mathrm{obs}}$ and the difference is used to calculate ΔX and to adjust X. Conceptually this technique is very close to the inversion of an FM considered in Sect. 3.1.1 (see Eq. (3.4)) and to the variational retrievals considered in Sect. 3.1.2.

As discussed in Sect. 2.4.4, the quality of the single NN Jacobian may not be sufficient to allow its use in DAS applications; however, an ensemble approach can be used to improve the NN Jacobian calculations. The NN ensemble described in the previous section of this chapter was used to create an ensemble of ten NN Jacobians, $\left\{\left\{\frac{\partial\phi^j_{\mathrm{NN}}}{\partial X_i}\right\}_{i=1,\ldots,n}\right\}^{j=1,\ldots,p}$, where $p = 10$ is the number of ensemble members. Then the ensemble mean Jacobian was calculated:

$$\overline{\frac{\partial\phi_{\mathrm{NN}}}{\partial X_i}} = \frac{1}{p}\sum_{j=1}^{p}\frac{\partial\phi^j_{\mathrm{NN}}}{\partial X_i}, \quad i = 1,\ldots,n. \tag{5.4}$$

Next, Eq. (5.3) was used to calculate $\Delta\eta_{\mathrm{NN}}$ using each ensemble member's Jacobian and the ensemble average Jacobian (5.4). These values of $\Delta\eta_{\mathrm{NN}}$ were compared with the exact $\Delta\eta$ known from the model simulation.

This comparison technique was applied to the last day of the entire model simulation. This date is separated by about 8 months from the last day of the simulation used for the NN training and validation. The fields generated by the model were used to create inputs X for the Jacobians of the NN observation operator (5.2). Then the NN Jacobian ensemble members were applied over the entire domain (excluding coastal areas) to generate an ensemble of 2-D fields of $\Delta\eta^j_{\mathrm{NN}}$ using Eq. (5.3). Also, $\overline{\Delta\eta_{\mathrm{NN}}}$ was calculated using the ensemble average Jacobian (5.4) in (5.3). A nondimensional distance in the model state space between the vectors X^0 and $X = X^0 + \Delta X$ was also introduced:

$$S = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\Delta X_i}{X_i^0}\right)^2}. \tag{5.5}$$

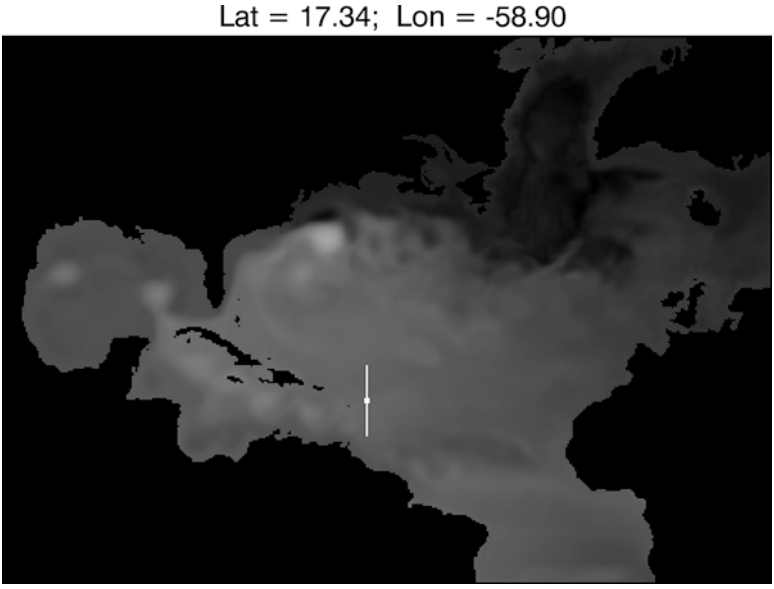

Fig. 5.5 The location of the cross-section (*white horizontal line*) inside the model domain; white dot shows the position of X^0

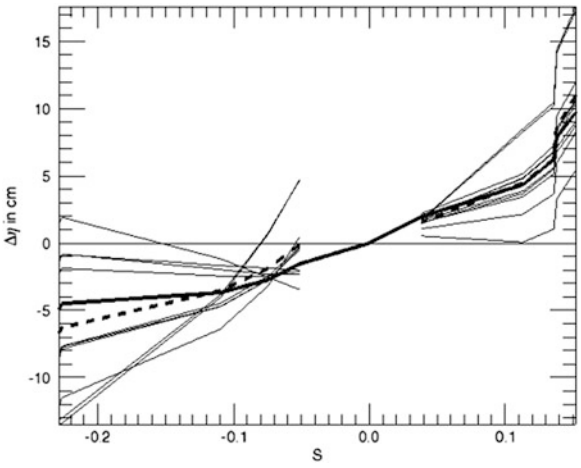

Fig. 5.6 $\Delta\eta$ calculated using (5.3) with the NN ensemble member Jacobians (an envelope of *thin solid lines* that illustrates the Jacobian uncertainties), the exact $\Delta\eta$ calculated from the model (*thick solid line*), and $\Delta\eta$ calculated using the NN ensemble average Jacobian (5.4) (*thick dashed line*). $\Delta\eta$ is shown vs. the distance S (5.5) in the model state space (Krasnopolsky 2007a)

Fields calculated in this way were compared with the corresponding field SSH, η, generated by the model. Multiple case studies were also performed at particular locations inside the model domain. The results of one case study are presented in Figs. 5.5, 5.6, 5.7, and 5.8.

Figure 5.5 shows the location of the cross-section (a white horizontal line) inside the model domain; the white dot shows the position of X^0. Starting from this

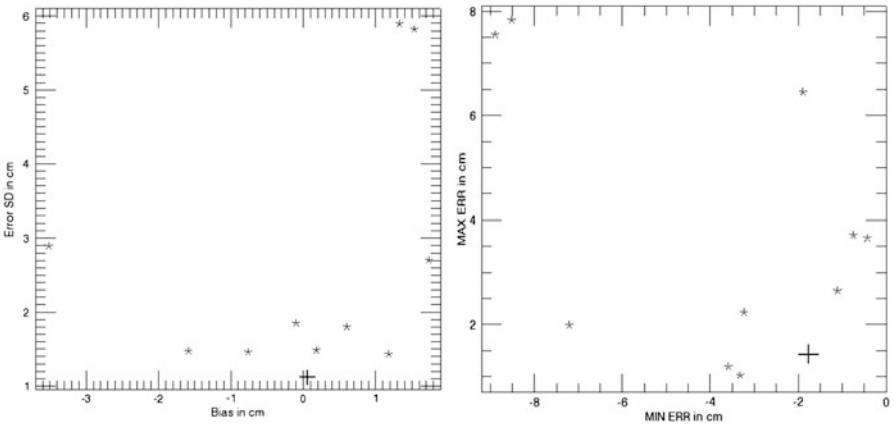

Fig. 5.7 The *left panel* shows the systematic error (bias) and the random error (error standard deviation) for $\Delta\eta$ calculated along the path shown in Fig. 5.5 using Eq. (5.3). The *right panel* shows the minimum and maximum errors along the path. The *asterisks* correspond to errors when the ensemble member Jacobians were used in (5.3), the cross corresponds to the case when the ensemble average Jacobian (5.4) was used in (5.3) (Krasnopolsky 2007a)

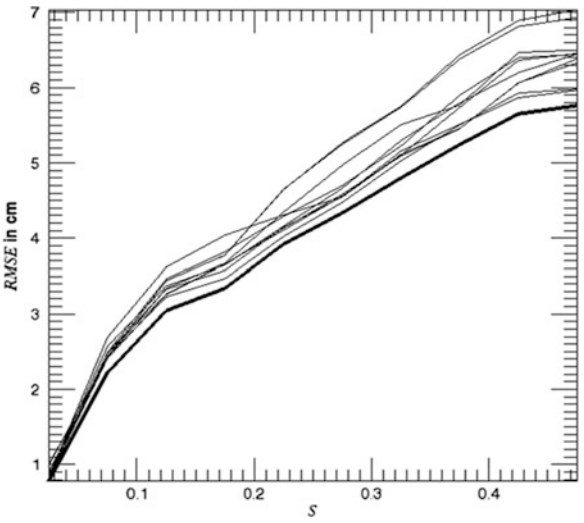

Fig. 5.8 Errors (RMSEs) of $\Delta\eta$ in cm as functions of nondimensional distance S (binned and averaged in each bin) over the entire model domain. *Thin lines* correspond to the ensemble members and the *thick line* shows the ensemble result (Krasnopolsky 2007a)

position, we moved up and down grid point by grid point, using X values at these grid points to calculate ΔX and the nondimensional distance in the model state space, S (5.5). These values of ΔX were used in (5.3) to calculate $\Delta\eta$.

Figure 5.6 shows $\Delta\eta$ calculated using (5.3) with the NN ensemble member Jacobians (an envelope of thin solid lines; their spread illustrates the Jacobian uncertainties), the exact $\Delta\eta$ calculated from the model (thick solid line), and $\Delta\eta$

calculated using the NN ensemble average Jacobian (5.4) (thick dashed line). $\Delta\eta$ is shown vs. the distance in the model state space, S (5.5). This figure demonstrates that the NN Jacobian can be improved significantly by using the ensemble average. The larger the distance S, the more significant the reduction in the Jacobian uncertainties is.

Figure 5.7 (left) shows the bias and random error for $\Delta\eta$ calculated along the path shown in Fig. 5.5 using (5.3). The asterisks correspond to the errors when ensemble member Jacobians were used in (5.3) and the cross to the ensemble average Jacobian (5.4). The ensemble bias is equal to the mean bias of the members as expected when using the simple conservative method to calculate the ensemble average. This figure also shows that in the case of the Jacobian, the ensemble approach very effectively reduces random errors or error SD; the ensemble random error ($\sim$1.1 cm) is less than the random error of any ensemble member. The reduction in bias ($\sim$90 %) and random error ($\sim$65 %) with respect to the maximum single member errors is very significant.

Figure 5.7 (right) shows minimum and maximum errors along the path or the statistics for extreme outliers. The same procedure, which was used to calculate errors presented in Fig. 5.4, was applied here. This figure shows that the NN ensemble approach is also an effective tool for reducing larger errors (by $\sim$4 times) in NN Jacobians.

Then the same procedure was applied at all grid points in the model domain. The errors have been calculated along numerous paths (both horizontal and vertical) over the entire model domain. Figure 5.8 shows the RMSE of $\Delta\eta$ as a function of the nondimensional distance S (averaged in each bin) over the entire domain. Thin lines correspond to the ensemble members while the envelope of thin solid lines illustrates the Jacobian uncertainties, and the thick line shows the NN ensemble result. The ensemble significantly improves the statistics for all values of S. Over the entire domain, the ensemble is always better than the best ensemble member.

5.1.3 Discussion

To better understand the magnitudes of errors presented in this section of Chap. 5, these magnitudes should be compared with the errors in the observed satellite data $\Delta\eta_{\text{obs}}$ assimilated in the oceanic DAS. The accuracy of the observed data is about 5 cm. Thus, the NN emulation (5.2) and the ensemble techniques allow a reduction in the Jacobian uncertainties and produce an ensemble Jacobian (5.4) that is sufficiently accurate to be used in the ocean DASs. These results are also important for conceptually similar problems described in Sect. 3.1: for retrieval algorithms based on the numerical inversion of FMs and for the variational retrievals.

The NN observation operator has been implemented in this example in the simplest manner; NNs were supplied with the information at a particular grid point and at a particular time, i.e., a point-wise approach was used. The flexibility of the NN approach allows us to introduce more sophisticated NN approaches.

For example, a field-wise approach, taking inputs from several neighborhood grid points, similar to the F2P or F2F approaches described in Sect. 3.6.3 of Chap. 3 can be applied and/or inputs from the previous time steps can be presented to NNs.

5.2 NN Nonlinear Multi-model Ensembles

In this section an approach based on the NN technique is introduced and applied to calculate a nonlinear average of an MME for improving 24-h precipitation forecasts over the ConUS. This approach allows us to account for nonlinear correlations between ensemble members and to produce an "optimal" forecast represented by a nonlinear NN ensemble mean. The NN approach is compared with the conservative MME, with multiple linear regression ensemble approaches, and with results obtained by a human forecaster.

For NWP models, rainfall is one of the most difficult fields to predict accurately. Detailed knowledge of the atmospheric moisture and vertical motion fields is critical for predicting the location and amount of rainfall, but these are difficult quantities to predict and observe accurately. NWP models must resort to parameterizations that treat convective clouds in a very simplified way to effectively take into account subgrid processes (cloud related processes), which determine very important parameters such as the amount of precipitation (see also discussions in Sect. 4.1.2 and Sect. 4.3.6, subsection "Parameterization and its uncertainties"). Thus, the errors in quantitative precipitation forecasts (QPFs) can arise as a result of limitations of the forecast model and errors in the observations.

To compensate for shortcomings in observing systems and model physics, there has been a trend in recent years toward ensemble forecasting, the realization of a number of model integrations using perturbed initial conditions. EPSs have been extensively tested and used in operations at the ECMWF and the US NCEP (Buizza et al. 2005; Palmer et al. 2007). The ensemble average has repeatedly been shown to give a more accurate forecast than a single realization of the forecast model (Zhang and Krishnamurti 1997; Du et al. 1997; Buizza and Palmer 1998). Drawbacks of the single-model EPSs are the following: (1) the technique is computationally expensive and lower-resolution versions of the models are generally employed that reduce the quality of the forecasts, and (2) assuming that errors result primarily from uncertainties in the initial conditions, any biases present in the model itself will also be present in the ensemble and may require calibration. The recent introduction of "stochastic" or "perturbed" physics attempts to account for uncertainties in the model subgrid scale processes (Buizza et al. 1999, 2005; Krasnopolsky et al. 2008) (see also Sect. 5.3), using ensembles with perturbed physics.

MME (aka poor man's ensemble) is another approach that has been taken to address the aforementioned issues. It combines forecasts from more than one NWP model. Ebert (2001) exhaustively investigated the advantages and disadvantages of the MME approach using an MME composed of seven operational NWP global and regional models. In the case of MME, the ensemble is composed of output from

different models and/or initial times, rather than a single model with perturbed initial conditions. Unlike EPSs that use singular vectors or breeding modes to generate optimal perturbations to the initial conditions, MME samples the uncertainty in the initial conditions via the different observational data, DASs, and initialization methods used by operational centers. MME also samples the uncertainty in model formulation due to the differences in model dynamics, the variety of model physics parameterizations, numerics, and resolutions. As a result, MME can be considered an approach, in which all components of the NWP system are perturbed, not only the initial conditions or model physics. Many authors (e.g., Speer and Leslie 1997; Du et al. 1997; Ebert 2001) have demonstrated the superior performance of MME.

5.2.1 Calculation of the Ensemble Average

In MME, as well as in EPS based on a single model, the final product is a combination of the ensemble members. At a particular time and location for an ensemble with N ensemble members, N predictors, P_i, $i = 1, \ldots, N$ are available for a particular variable P. To produce an ensemble prediction, ensemble members are combined in a predictand. The simplest and most common combination of the ensemble members is an ensemble mean (EM), which is calculated as a simple average of ensemble members and is what we have called a conservative ensemble,

$$\mathrm{EM} = \frac{1}{N}\sum_{i=1}^{N} P_i, \tag{5.6}$$

where N is the total number of ensemble members and P_i is the ith ensemble member generated by the model number i. This approach to combining ensemble members has two major advantages: it (1) does not require any additional information; therefore, (2) the unique result (5.6), EM, can always be calculated. The major disadvantage of the conservative ensemble (5.6) is that it does not make the best use of the information contained in the set of predictors.

More sophisticated approaches use weighted ensemble means (WEM),

$$\mathrm{WEM} = \frac{\sum_{i=1}^{N} W_i \cdot P_i}{\sum_{i=1}^{N} W_i}, \tag{5.7}$$

where ensemble members are subscribed with weights, W_i, that are usually based on ad hoc considerations. For example, if, from past experience, it is known that some models give better predictions than others, they can be assigned higher weights in (5.7).

A multiple linear regression technique was used by Krishnamurti et al. (1999, 2000) to determine the optimal weights, W_i, for combining the ensemble members. This approach can be used only if a training dataset is available to learn regression coefficients from data; a significant improvement was demonstrated over the conservative ensemble when using the weighted ensemble mean. If training data are available, Eq. (5.7) can be generalized and other predictors, x_i, $i = 1, \ldots, m$, can be included in the linear regression:

$$\mathrm{WEM} = \sum_{i=0}^{m} a_i \cdot x_i + \sum_{i=m+1}^{N+m} a_i \cdot P_{i-m}. \tag{5.8}$$

The aforementioned approaches (both simple and weighted means) implicitly assume a linear dependence between ensemble members and the best predicted value (the amount of precipitation in our case). However, in many cases, predictors are significantly correlated. In the case considered here, it happens because QPFs produced by different NWP models for the same time and location are similar and correlated. Linear regression becomes numerically ill conditioned when dealing with correlated predictors. Also, in some cases the assumption of linear dependence may be incorrect per se. For example, for longer forecast horizons when bifurcation of the ensemble forecasts may occur, this assumption can lead to misleading results. Also, for fields with high gradients and sharp, localized features (e.g., precipitation), the assumption of linearity may lead to significant problems in MME predictions (see a more detailed discussion in the following sections). In such cases the dependence between the ensemble members and the best predicted value may be complex and nonlinear.

Here, we relaxed the linearity assumption and allowed for an arbitrary nonlinear dependence between the MME members and the best predicted value, MME, as

$$\mathrm{MME} = f(X), \tag{5.9}$$

where the vector $X = \{x, P\}$, $P = \{P_i\}_{i=1,\ldots,N}$, is a vector of the ensemble members and $x = \{x_i\}_{i=1,\ldots,m}$, is a vector of additional predictors, which may accommodate time and location dependencies as well.

An NN technique is used to approximate this arbitrary nonlinear dependence (5.9) using a training set composed of past data to learn NN weights from the data. The nonlinear NN ensemble mean (NNEM), which we introduce here, is defined following (2.2 and 2.3) as an analytical MLP that can be written as

$$\mathrm{NNEM} = a_0 + \sum_{j=1}^{k} a_j \cdot \phi\left(b_{j0} + \sum_{i=1}^{n} b_{ji} \cdot X_i\right), \tag{5.10}$$

where the X_i are components of the input vector X (the same as in (5.9)) composed of the same N inputs (ensemble members) as those used in the EM and WEM equations (5.6) and (5.7) plus optional additional input parameters (see (5.9)), n is the number of inputs ($n \geq N$), and k is the number of neurons in (5.10).

It is worth repeating that expression (5.10) is capable of approximating any nonlinear relationship between non-stochastic variables. However, the training set that is used for training (5.10) is composed of inputs and outputs that contain uncertainties. The inputs, X, contain vectors of QPFs predicted by NWP models, and the outputs contain the observed QPFs. Both inputs and outputs contain significant uncertainties (see Sect. 5.2.1) and are stochastic variables. Thus, the nonlinear function f is also a stochastic function (or a degenerated stochastic mapping (2.1a, 2.1b)) because it describes a relationship between two stochastic variables.

Actually, the stochastic function is a family of functions, each of which describes a relationship between two variables inside a corridor determined by the uncertainties of these variables with a probability determined by their joint probability density function. As a result, a single NN (5.10) cannot provide an adequate approximation for such a stochastic function (5.9). However, the NN technique is rich and flexible enough to solve this problem. It was suggested (see Sect. 2.4.6, and Krasnopolsky et al. 2011) that an ensemble of NNs can be used to approximate these stochastic functions (and mappings). Thus, multiple NNs (an NN ensemble) have been produced to approximate the stochastic function (5.9), f, where each NN ensemble member is represented by (5.10). Finally, the QPF is calculated as a conservative ensemble of the NN ensemble members NNEM_i,

$$\mathrm{MNNEM} = \frac{1}{q}\sum_{i=1}^{q} \mathrm{NNEM}_i, \tag{5.11}$$

where q is the number of NNs (5.10) in the NN ensemble and each NNEM_i is one of q NNEMs (5.10). There are many different methods of creating NN ensembles. Here, we used an ensemble of NNs (5.10) that have different weights a and b corresponding to different local minima of the error function minimized during the NN training.

Using an ensemble of NN MME means (NNEMs) has an additional advantage. It allows us to calculate the uncertainty of the MME forecast as the SD of the NNEMs:

$$\sigma = \sqrt{\frac{1}{q-1}\sum_{i=1}^{q}(\mathrm{NNEM}_i - \mathrm{MNNEM})^2}.$$

Forecast and Verification Data

Several aforementioned MME techniques were applied for calculating 24-h precipitation forecasts over the ConUS (Lin and Krasnopolsky 2011). Twenty-four-hour precipitation forecasts over ConUS are available from eight operational models, including NCEP's mesoscale and global models (NAM and GFS), the regional and global models from the Canadian Meteorological Center (CMC and CMCGLB), global models from the Deutscher Wetterdienst (DWD), the ECMWF, the Japan

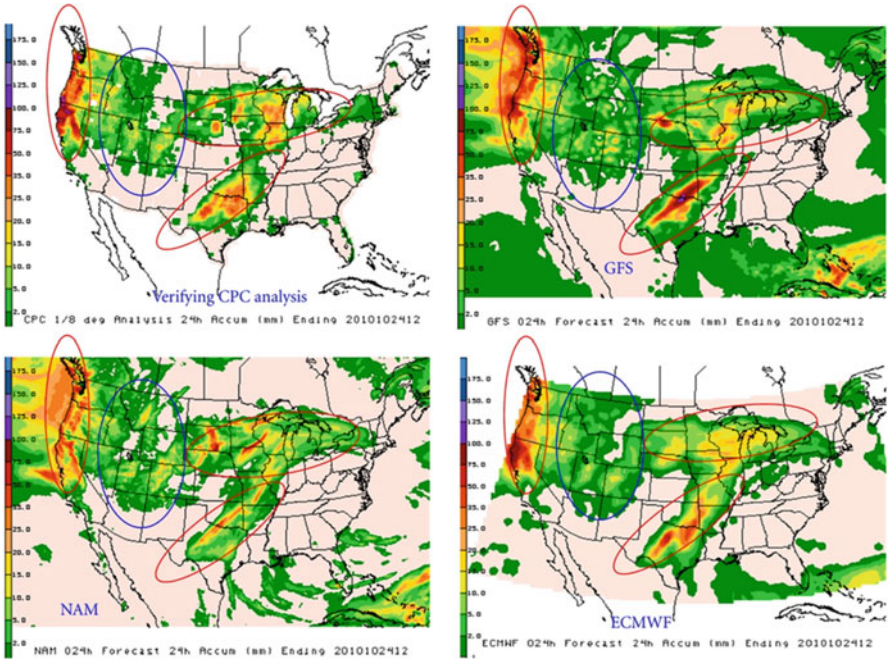

Fig. 5.9 The model results (24 h forecast) for three models (NAM, GFS, and ECMWF) together with the CPC verification analysis for October 24, 2010. *Red* and *blue ellipses* show high and low precipitation areas, respectively. The figure illustrates the uncertainties in model forecasts, especially for high and low precipitation (Krasnopolsky and Lin 2012)

Meteorological Agency (JMA), and the UK Met Office (UKMO). Also the NCEP Climate Prediction Center (CPC) precipitation analysis is available. The CPC's 1/8 degree daily gauge analysis is used in the training of NNs and for the verification of model predictions. All gridded data fields where interpolated to the same grid, the 40-km Lambert-conformal Advanced Weather Interactive Processing System Grid 212 that encompasses ConUS.

Data indicate that all models demonstrate similar deviations from the analysis: at lower levels of precipitation they are slightly wetter than the CPC analysis and at the higher levels (>50–60 mm/day) they are dryer than the CPC analysis (for detailed discussion, see Lin and Krasnopolsky 2011). In addition, locations of the highs and lows and details of the precipitation features are different in the precipitation fields produced by different models. The model results (24 h forecast) for three models (NAM, GFS, and ECMWF) together with the CPC verification analysis are shown in Fig. 5.9. The figure illustrates the uncertainties in model forecasts, especially for high and low precipitation.

Figure 5.10 shows a scatter plot, which presents all eight model predictions over the first 6 months of 2010 plotted vs. the CPC analysis. It demonstrates a tremendous spread in the model results, i.e., in MME members.

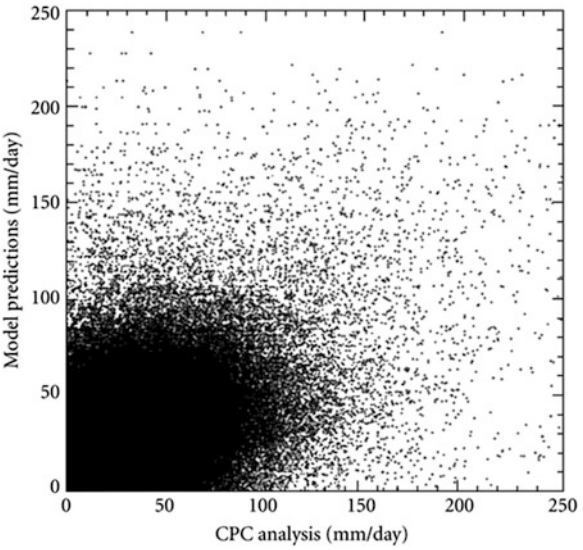

Fig. 5.10 Scatter plot showing 24 h precipitation forecasts obtained by eight models over the first 6 months of 2010 vs. corresponding CPC analysis (Krasnopolsky and Lin 2012)

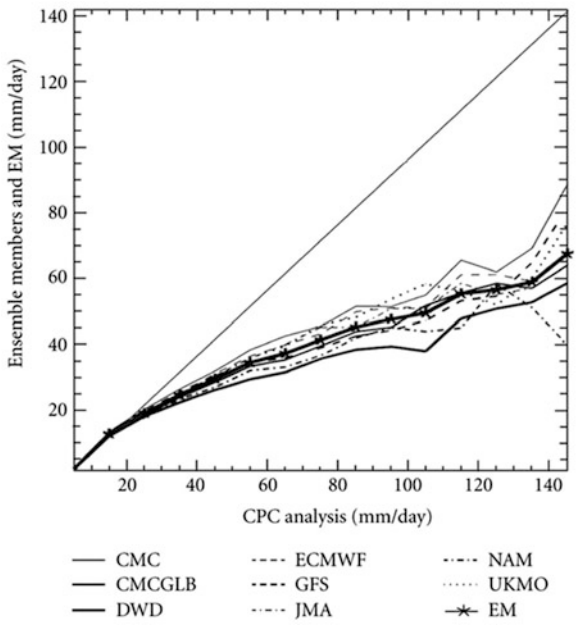

Fig. 5.11 Binned scatter plot for eight models (ensemble members) and EM (5.6) (Krasnopolsky and Lin 2012)

The uncertainty of the forecast is especially large at higher levels of precipitation. The binned scatter plots (the variable is divided in small bins, bin means are calculated, plotted, and connected with a line) for all eight model 24 h predictions vs. the CPC verification analysis are shown in Fig. 5.11 together with the conservative ensemble EM (5.6). The models create an envelope with the spread increasing with

an increase in the precipitation rate. All models have increasingly low bias at high levels of precipitations. Figures 5.10 and 5.11 demonstrate the stochastic nature of the system under consideration.

Ensemble Approaches to Improve Prediction of Precipitations

Here we compare approaches that apply the MME technique for 24 h precipitation forecasts using multiple linear regression and nonlinear (e.g., NN) techniques to improve upon the conservative linear ensemble (5.6). Because of the model problems described above, the research community has been exploring various ways of making better precipitation forecasts. Among the approaches investigated in this section, we consider an eight-member MME, which is averaged in three different ways calculating: (1) the conservative EM (5.6), (2) WEM (5.8) based on multiple linear regression, and (3) a nonlinear NN ensemble mean, NNEM (5.10).

As can be seen in Fig. 5.11, the conservative EM (5.6) runs through the middle of the envelope created by the models. EM provides a better placement of precipitation areas; however, in other respects it does not improve the situation significantly. Moreover, as illustrated in Figs. 5.13 and 5.14, EM (5.6) smoothes, diffuses features, reducing the spatial gradients; it has a high bias for low levels of precipitation (and produces large areas of false low precipitation) and low bias at high levels of precipitation (highs are smoothed out and reduced). These problems motivated the search for improved techniques, including nonlinear NN ensembles.

First, an improved linear technique was introduced and investigated. To make comparisons with the NN ensemble, we introduce the WEM (5.8) as a multiple linear regression using the same inputs as the NN ensemble (5.10). The multiple linear regression ensemble mean (WEM) was created in the following way (Krasnopolsky and Lin 2012):

$$\mathrm{WEM} = a_1 \cdot cjd + a_2 \cdot sjd + a_3 \cdot lat + a_4 \cdot lon + \sum_{i=1}^{8} a_{i+4} \cdot P_i, \qquad (5.12)$$

where $\{a_i\}_{i=1,\ldots,12}$ are regression parameters, $cjd = \cos\left(\frac{\pi}{183} \cdot jday\right)$, $sjd = \sin\left(\frac{\pi}{183} \cdot jday\right)$, $jday$ is the Julian day, lat is the latitude, lon is the longitude, and P_i are the ensemble members at a particular grid point in the ConUS grid. The first two parameters account for the annual cycle. Thus, the multiple linear regression (5.12) has a total of 12 input parameters.

The NN ensemble mean (NNEM) is defined as in (5.10) where the input vector X is composed of the same $n = 12$ inputs as those used for WEM (5.12). $k = 7$ was selected after multiple trials to avoid over-fitting (Krasnopolsky and Lin 2012). Both WEM and NNEM have one output – a 24 h precipitation forecast. The CPC analysis corresponding to the time of the forecast was used to train the outputs in both cases. We note that the regression parameters for WEM and the NN weights for NNEM do not change from grid point to grid point and do not depend on time. After WEM

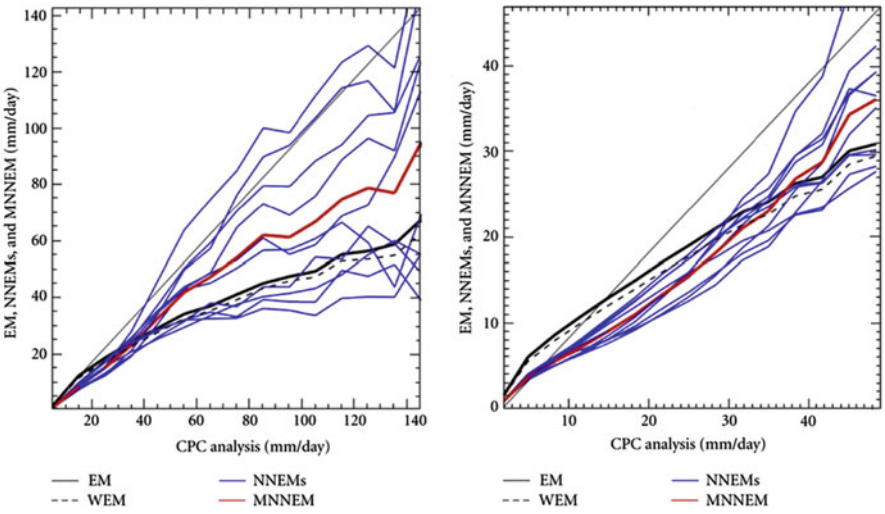

Fig. 5.12 Binned scatter plots for *EM* (5.6) (*black solid*), WEM (5.12) (*black dashed*), ten NNEMs (5.10) (NNEM$_i$, $i = 1, \ldots, q$ and $q = 10$, all *blue*), and MNNEM (*red*) that is defined by Eq. (5.11). The *right panel* shows the lower precipitation area magnified (Krasnopolsky and Lin 2012)

and NNEM are trained, they are used with the same set of regression coefficients (or weights for the NN) for any grid point in the ConUS grid at any time. Thus, the results depend on time and location only through their input parameters.

5.2.2 Results

The WEM and NNEM have been developed using 2009 data, more than 310,000 input/output records. They have been validated on independent data for the first half of 2010, e.g., the results shown in Figs. 5.10, 5.11, and 5.12 have been calculated using these validation data. Figure 5.11 shows the binned scatter plot for the amount of precipitation over the ConUS territory during the first 6 months of 2010. It shows the results from eight models together with the EM (5.6) results vs. the CPC analysis. Our validation showed that, for precipitation fields, WEM (5.12) does not significantly improve upon the conservative MME EM (5.6).

In Fig. 5.12 these two ensemble means, EM and WEM, are shown by thick solid and dashed black lines correspondingly. As can be seen from Figs. 5.11 and 5.12, all models, EM, and WEM are slightly wetter than the CPC analysis at lower precipitation amounts and significantly dryer than the CPC analysis at higher precipitation amounts. The linear ensembles, EM and WEM, do not change the situation significantly (see both panels of Fig. 5.12). Also the multiple linear regression ensemble, WEM, does not significant improve EM.

As we mentioned earlier, there is a significant difference between the linear ensemble averaging techniques (5.6 and 5.7) and the nonlinear (5.10). EM (5.6)

is always unique. WEM (5.7) always provides a unique solution for a given training set. Nonlinear ensemble averaging, and the NN ensemble mean NNEM (5.10), in particular, provide multiple solutions for a given training set. For accurate training data (described by the non-stochastic function (5.9) with no uncertainty), different solutions have different approximation errors, and the best solution with the smallest approximation error can be selected. For training data with a high level of uncertainty (noise), similar to the data shown in Figs. 5.10 and 5.11, the function (5.9) is a stochastic function, and multiple solutions may have almost the same approximation accuracy as the uncertainty in the data. Thus, all of these solutions provide valid representations of the stochastic function (5.9) and valid nonlinear averaging of the MME.

In terms of the NN approach, we trained an ensemble of ten NNs (5.10) with the same architecture ($n = 12$ inputs, one output, and $k = 7$ hidden neurons) but different initialization values for weights a and b (see Eq. (5.10)). The training of these NNs, which is a nonlinear minimization of an error function, leads to ten different local minima of the error function with approximately the same value of the approximation error. However, because these ten NNs have different weights a and b (see Eq. (5.10)), they produce very different results in the areas where the uncertainty of the data is higher (higher levels of precipitation).

The results of the application of different MME averaging NNs (NN ensemble members) to the validation data set are shown in Fig. 5.12. The figure shows binned scatter plots for EM (5.6), WEM (5.12), and ten NNEMs (5.10) (NNEM$_i$, $i = 1, \ldots, q$ and $q = 10$). The left panel displays the entire range of precipitation values from 0 to 145 mm/day, and the right panel magnifies the lower precipitation area from 0 to 50 mm/day.

All ten NNEMs are in a good agreement at the lower levels of precipitation, but they diverge significantly at the higher levels. The large spread at higher levels of precipitation reflects the uncertainty in the data and the differences in predicting higher levels of precipitation by the different members of the MME (see Fig. 5.10), i.e., the uncertainty of MME. Also, it is noteworthy that in the training and validation, data sets less than 0.5 % of the data records correspond to precipitation values greater than 50 mm/day and only a few records with values greater than 100 mm/day.

To improve the statistical significance of the nonlinear NN MME averaging (especially at higher precipitation values), we consider the ten aforementioned NNs as an ensemble of averaging NNs and calculate the NN ensemble mean MNNEM using Eq. (5.11). This is shown in Fig. 5.12 by a red solid line. MNNEM produces a significant improvement over the *EM* and WEM results at higher levels of precipitations (Fig. 5.12, left panel); it significantly reduces the low bias at higher precipitation levels (35 mm/day and higher). It also improves results at low precipitation levels, significantly reducing the high bias at lower precipitation levels (from 0 to 10 mm/day). However, at medium precipitation levels from ∼12 to 30 mm/day, MNNEM and the majority of the NN ensemble members have a lower bias than EM and WEM, as shown in Fig. 5.12 (right panel). Thus, the nonlinear NN ensemble averaging approach is flexible enough to balance wetness at lower amounts of precipitations with dryness at the higher amounts.

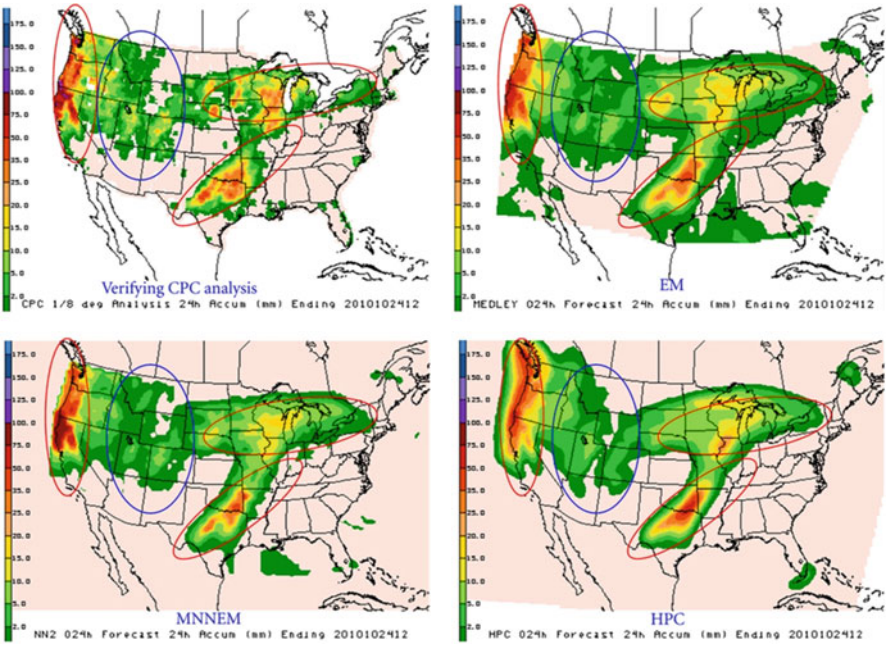

Fig. 5.13 Comparison of three 24 h forecasts: EM (*upper right*), MNNEM (*lower left*), and HPC (*lower right*) vs. CPC analysis for October 24, 2010. The *red ellipses* show high precipitation areas and the *blue ellipses* show low. HPC is for the Hydrometeorological Prediction Center (Krasnopolsky and Lin 2012)

Figures 5.13 and 5.14 demonstrate two case studies that show advantages of the nonlinear NN ensemble forecast, MNNEM, as compared with the conservative ensemble forecast, EM. Here we do not show the WEM (5.12) results because it is difficult to distinguish them from the EM results. The CPC analysis for the time corresponding to the forecast is used for verification. Also, a manual 24 h forecast produced at the Hydrometeorological Prediction Center (HPC) is also presented for comparison. To produce the HPC forecast, a forecaster uses the model forecasts as well as all available observations and satellite data (including satellite imagery) (Novak et al. 2011).

As Figs. 5.13 and 5.14 demonstrate, the nonlinear NN averaging of MME improves the positioning of the precipitation features inside the precipitation fields. It removes significant areas of false low-level precipitation produced by the conservative EM (5.6) technique. It sharpens the features and enhances precipitation fronts and the maxima in precipitation. The MNNEM technique provides a forecast that is comparable with the human HPC forecast while using fewer resources and less time.

The statistical results that characterize the accuracy of positioning the precipitation features are shown in Fig. 5.15. The statistics covers a period of 8 months, from November 15, 2010 to July 15, 2011 (the NNs have been trained on 2009 data).

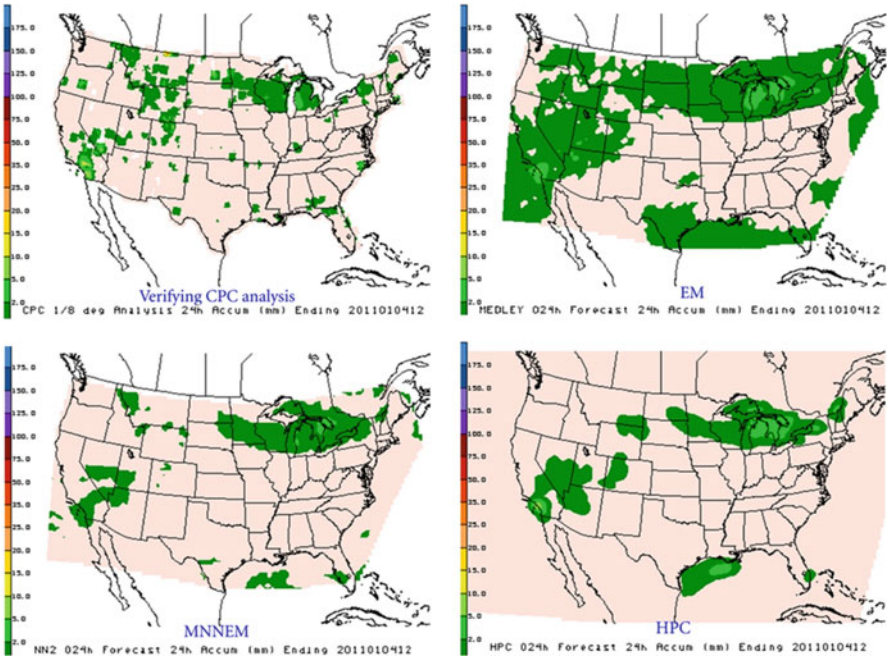

Fig. 5.14 The same as in Fig. 5.13 but for January 4, 2011 (Krasnopolsky and Lin 2012)

The Equitable Threat Score (ETS) (Wilks 2011) measures that fraction of observed events that are correctly predicted, adjusted for correct predictions that are due to random chance. Possible ETS ranges from −1/3 to 1 (a perfect forecast would have a score of 1 for every precipitation threshold). A bias score is defined simply as the ratio of the areal coverage of a forecast vs. the observed precipitation exceeding a given threshold. An ideal forecast would have a bias score of 1 at every threshold.

Summarizing, the MNNEM forecast is comparable with the HPC forecast and significantly better than *EM* at the threshold values of less than 0.1 in./day and of more than 1.0 in./day, which is in good agreement with the statistics presented in Fig. 5.12.

5.2.3 Discussion

A nonlinear NN ensemble approach to improve 24-h MME precipitation forecast was introduced in this section. It was shown that the NN MME improves upon the conservative linear ensemble; it:

1. Significantly reduces high bias at low precipitation levels
2. Significantly reduces low bias at high precipitation levels
3. Sharpens features making them closer to the observations

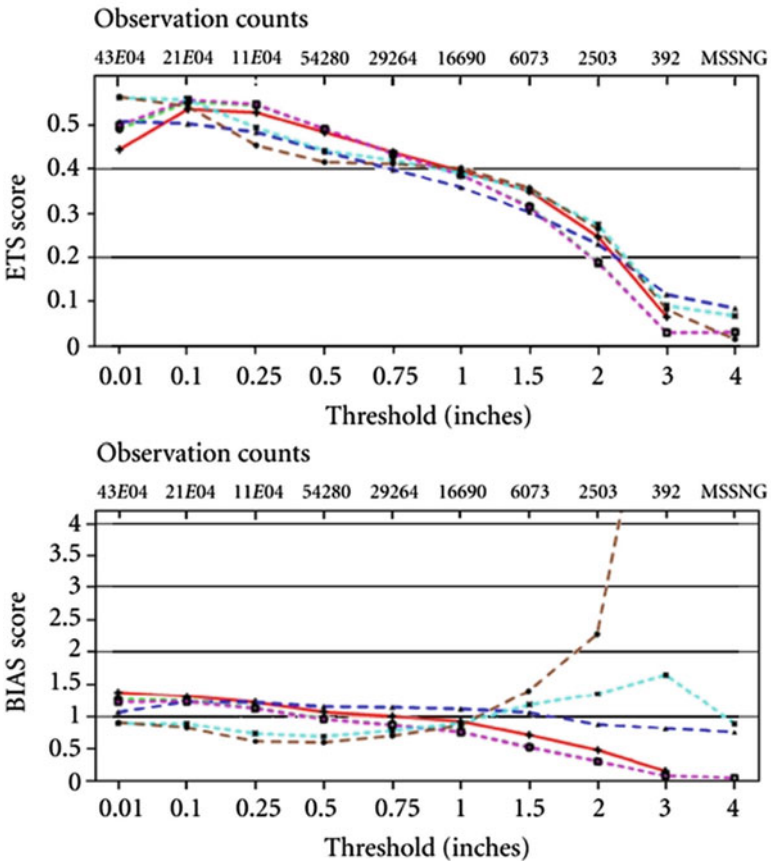

Fig. 5.15 ETS (*upper panel*) and bias score (*lower panel*) for a period of 8 months from November 15, 2010 to July 15, 2011. Five different 24 h MME forecasts are presented: EM (5.6) – *solid red*, WEM (5.9) – *dashed pink*, HPC forecast – *dashed blue*, MNNEM (5.10) – *dashed light blue*, and one of NNEM (one member of the NN ensemble (5.10)) forecasts – *dashed brown* (Krasnopolsky and Lin 2012)

It is noteworthy that the NN MME forecast works at least as well as that produced by a human analyst HPC forecast without the benefit of additional information that is available to the analyst, and it is less time and resource intensive.

It is also noteworthy that the NN technique is flexible enough to accommodate the time and space dependence of the environment in which NN works through additional time- and location-dependent inputs. The NN technique also allowed us to take into account the stochastic nature of the problem. We successfully used NN ensemble technique to approximate a stochastic function (mapping), which performs an averaging of MME taking into account significant uncertainty in the data used for NN training.

Any ensemble averaging technique (linear or nonlinear) based on using past data, including the NN approach introduced in this paper, requires additional maintenance efforts as compared with the simplest conservative ensemble (5.6). The MME ensemble members, the NWP models, are evolving complex systems. Their prediction quality changes with time. Accordingly, the function f (5.9) changes with time. Therefore, the quality of its approximation (5.8), or (5.10), or any other, should be permanently monitored, and, if necessary, the approximations should be periodically adjusted (retrained). For the approach introduced in this chapter, the NN ensemble trained on 2009 data still worked well in 2011. Hence, if the NN ensemble requires a retraining, it should be performed once per M ($M > 2$) years.

Here, the NN approach has been implemented in a simple way; NNs were provided with the same information that the linear MME used. The flexibility of the NN approach allows us to introduce more sophisticated NN approaches. For example, to introduce information available to a human analyst (and HPC analyses per se) as additional inputs to our NN or to implement a field-wise approach taking inputs from several neighborhood grid points similar to the F2P or F2F approaches described in Sect. 3.6.3 of Chap. 3. The nonlinear NN averaging approach presented here is generic. Although it was applied to precipitation fields here, it is clear that it can also be applied to other fields. Also, it can be applied to calculate nonlinear ensemble means in a single-model EPS as well.

5.3 Perturbed Physics and Ensembles with Perturbed Physics

In this section several applications of the NN emulation technique for perturbing model physics and calculating ensembles with perturbed model physics are introduced. Two types of perturbed physics ensembles are discussed: a regular perturbed physics ensemble (PPE), which follows the scenario of a perturbed initial condition ensemble (PICE) (see Fig. 5.16), a type of ensembles traditionally used in EPS, and a short-term perturbed physics ensemble (STPPE). The NN emulation technique can be efficiently used to create both the PPE and the STPPE. However, all three aforementioned types of ensembles (PICE, PPE, and STPPE) can significantly benefit, in terms of their numerical performance, by using NN emulations of model physics.

During the last decade, ensemble techniques have demonstrated significant successes in NWP (Palmer et al. 2007; Buizza et al. 2005) and in numerical climate simulations (Broccoli et al. 2003; Murphy et al. 2004; Stainforth et al. 2005; Yoshimori et al. 2005). The traditional ensemble approach, PICE is widely used in NWP EPSs and consists of introducing perturbations into the initial conditions because NWP problems (specifically, for short- to medium-range weather predictions) correspond to the initial condition problems for Eq. (4.1).

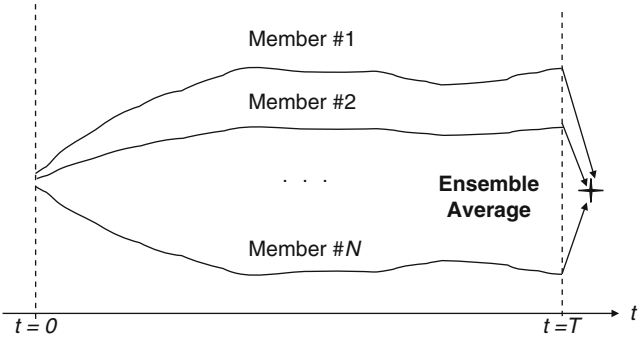

Fig. 5.16 The PICE and PPE scenario showing individual member trajectories

It was also found that for both NWP and climate applications, the spread of the PICE forecasts is insufficient to systematically capture reality, and perturbing the model physics has been introduced in some ensemble forecast systems (Buizza et al. 1999; 2005). Climate simulation problems are rather boundary condition and right-hand side (rhs) forcing problems in Eq. (4.1) than initial condition problems as it is for the NWP. For this kind of problem, an ensemble approach based on perturbation of the model physics (or perturbation of the forcing) seems to be appropriate. Thus, the perturbed physics ensembles are expected to be more effective for climate simulations and projections (Stainforth et al. 2005).

5.3.1 Ensemble Approaches in NWP and Climate Simulations

GCMs used for climate simulations and numerical NWP are complex nonlinear systems composed of many elements (see Eq. (4.1)): the initial conditions, ψ_0; boundary conditions, ψ_B; model dynamics, $\Omega(\Psi, t)$; and model physics, $P(\Psi, t) = \sum_k p_k(\Psi, t)$ (p_k are parameterizations that represent various physical processes). Here, Ψ is the atmospheric state vector, incorporating both ψ and x, and t is time. Each of these elements, as well as the initial conditions, can be considered as a specific component that has its own internal (natural) uncertainty. The major sources of uncertainty in the initial conditions are observation errors and subgrid variability in the data. For the model physics, the major source of uncertainty are subgrid scale physical processes (see also Sect. 4.3.6, subsection "Parameterization and its uncertainties" for more information). Thus, the initial conditions as well as the model physics may be perturbed within its natural range of uncertainty to produce an ensemble of model realizations. Each of these ensemble realizations produces a prediction which constitutes an ensemble member.

Formally, an EPS system may be represented as a set of numerical integrations,

$$\Psi_j(T) = \Psi_j(0) + \int_0^T \left[P_j\left(\Psi_j, t\right) + \Omega\left(\Psi_j, t\right)\right] dt, \tag{5.13}$$

where $j = 1, \ldots, N$ is the number of the ensemble member. All ensemble members are similar but yet slightly different. The ensemble approach allows for integrating the specific information contained in the individual ensemble members into an ensemble that "knows" more, or has more information about, or represents the predicted climate or weather better, than any of the individual ensemble members.

Ensembles with Perturbed Initial Conditions

The traditional ensemble approach, PICE, widely used in NWP consists of introducing perturbations into the initial conditions (Buizza 1997); the model physics is not perturbed and the P_j are the same in Eq. (5.13) for all ensemble members. Based on this approach, each ensemble member run starts from a uniquely perturbed initial condition $\Psi_j(0)$. After running independently for some prescribed time T, the results of the ensemble member runs are compared with each other and with observations and then averaged (see Fig. 5.16).

Usually, the ensemble average describes an actual weather or climate better at the moment $t = T$ than a single ensemble member. Using PICEs allows us to observe how small uncertainties in the initial conditions develop over the model integration time into significant/measurable differences in predicted atmospheric states. For PICEs, the initial time step is the only time step for which an uncertainty is taken into account, i.e., the only time when perturbations are introduced in a deterministic NWP model. PICE has proved to be an effective tool for NWP; however, the spread of the PICE forecasts is often insufficient for providing systematic improvements in NWP (Buizza et al. 2005).

Ensembles with Perturbed Physics

The perturbed physics ensembles (PPEs) are shown to be very effective for climate simulations and projections (Kharin and Zwiers 2000; Stensrud et al. 2000; Broccoli et al. 2003; Murphy et al. 2004; Stainforth et al. 2005; Yoshimori et al. 2005). Within this approach, each ensemble member uses a uniquely perturbed version of the model physics P_j. PPE can also be used in combination with PICE (Stainforth et al. 2005) as shown in Eq. (5.13).

Several different approaches have been used for perturbing model physics:

- Model random errors associated with physical parametrizations are simulated by multiplying the total parametrized physics P by a random number r_j sampled from a uniform distribution between 0.5 and 1.5 ($P_j = r_j \times P$) (Buizza et al. 1999, 2005).
- One or several of the model physics parameters that control key physical characteristics of the subgrid scale atmospheric and surface processes can be perturbed (one or several at a time) within the scope of their natural range of uncertainty (Murphy et al. 2004; Stainforth et al. 2005).

- Different model physical process parameterization schemes can be used to create various versions of perturbed model physics; the different versions are used in different ensemble members (Stensrud et al. 2000).
- NN emulations of model physics can also be used as a tool to create different realizations of the model physics for generating ensembles of the perturbed physics (Krasnopolsky et al. 2008).

Usually the same scenario, as that depicted in Fig. 5.16 for the PICE with perturbed initial conditions, is followed for creating PPE. A particular GCM ensemble member uses a particular version of the perturbed physics, P_j, throughout the entire GCM run over time T. Thus, in PPE, different versions of the perturbed physics (different realizations of the subgrid scale physics) are used for different ensemble members, and each ensemble member exists and evolves over the entire GCM integration, a period that is much longer than the characteristic time scale of the subgrid physical processes involved.

Short-Term Ensembles with Perturbed Physics

Using the perturbed physics approach for generating ensembles offers an opportunity to introduce an alternative ensemble approach, namely, a new type of ensemble – an STPPE (Krasnopolsky et al. 2008) that is not possible in the framework of the traditional PICE approach. In the STPPE mode, the ensemble of different realizations (perturbed versions) of the model physics is introduced for a time interval comparable with the time scales of the subgrid processes involved, namely, during one time step (or for some parameterizations, for several time steps) of the model integration. Symbolically the STPPE can be written as

$$\Psi(T) = \Psi(0) + \int_0^T \left[\frac{1}{N} \sum_{j=1}^{N} P_j\left(\Psi, t\right) + \Omega\left(\Psi, t\right) \right] \mathrm{d}t. \tag{5.14}$$

At each time step, an ensemble of different realizations of the model physics is generated and averaged. The ensemble average is used to integrate the model for producing the next time step. The STPPE averaging scenario is shown in Fig. 5.17.

The major differences between the PICE or PPE approaches (Fig. 5.16) and STPPE (Fig. 5.17) are:

- PICE and PPE consist of N independent model runs; STPPE consists of a single model run.
- In the PICE and PPE approaches, the ensemble averages for climate or weather characteristics are calculated at the end of all N model integrations combining climate or weather characteristics for all single ensemble member runs; within STPPE, the ensemble average is calculated at each integration time step, t_i, for the outputs of the ensemble members composed of perturbed components of the

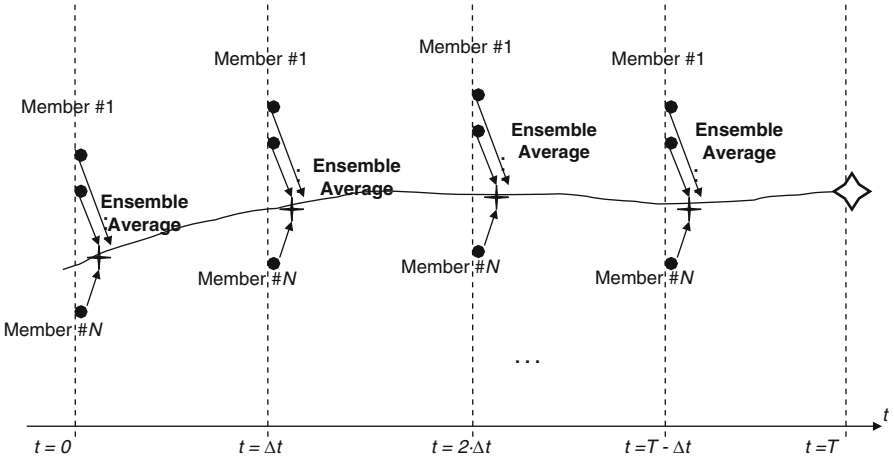

Fig. 5.17 The STPPE averaging scenario

model physics. The weather or climate characteristics obtained at the end are the results of this single STPPE run. There is no additional averaging of the weather or climate characteristics using this approach.

- SLPPE may be significantly faster than PICE or PPE.

In terms of computation time, the calculations of a perturbed version (or component) of the model physics take about $1/mT$, where $1/m < 1$ is a fraction of T required for calculation of the model physics (or a particular component/parameterization of the model physics that is perturbed) and T is the total time required for the integration of one PICE member; then the time required for an STPPE run is

$$T_{\text{STPPE}} = \left[\left(1 - \frac{1}{m}\right) + \frac{N}{m}\right] \cdot T, \tag{5.15}$$

whereas PICE or PPE runs take a longer time

$$T_{\text{PIEC}} = N \cdot T = N \cdot \left[\left(1 - \frac{1}{m}\right) + \frac{1}{m}\right] \cdot T. \tag{5.16}$$

A major limitation of all three ensemble approaches (PICE, PPE, and STPPE) is the time required for their execution. Both PICE and PPE require N (N – is the number of ensemble members) times more than a single model run; that is, $N \cdot T$, where T is the time required for one GCM run. STPPE requires significantly less time because only the model physics is calculated N times. For example, if the calculation of the model physics takes 50 % of the total model execution time (i.e., $m = 2$), the STPPE will be about two times faster than PICE or PPE runs, assuming that the number of ensemble members, N, is the same. If the model

physics calculation time is reduced, the STPPE becomes even more computationally efficient. In the next section, we show that STPPE becomes very efficient (orders of magnitude faster than PICE and PPE) when the NN technique is used to produce the ensemble of perturbed realizations of the model physics.

5.3.2 *Neural Network Ensembles with Perturbed Physics*

NN emulations can serve not only as a tool for introducing fast versions of model physics (see Chap. 4) but also as a promising approach for perturbing model physics. The NN emulation technique allows us to conveniently and naturally introduce perturbations in the model physics (or a component of the model physics) and to develop fast versions of perturbed model physics (or fast perturbed components of model physics). Also, using NN-based perturbed physics can make the computation time for the entire ensemble (if STPPE is used) comparable with the computation time for a single model run.

The NN emulation technique can be used to introduce an ensemble of perturbed model physics in the following way. The *j*th perturbed version of the unperturbed model physics, P, can be written as

$$P_j = P_j^{\mathrm{NN}} = P + \varepsilon_j, \tag{5.17}$$

where P_j^{NN} is an NN emulation number j of the original model physics, P, and ε_j is the emulation error for the NN emulation number j. As has been shown in Chaps. 2 and 4, ε_j can be controlled and changed significantly by varying k (the number of hidden neurons) in Eq. (2.2). Not only the value but also the statistical properties of ε_j can be controlled. For example, the systematic components of the emulation errors (biases) can be made negligible (therefore, ε_j are purely random in this case). Thus, ε_j can be made of the order of the magnitude of the natural uncertainty of the model physics (or of a particular parameterization). This uncertainty emerges due to unaccounted variability in the subgrid processes (see also discussion in Sect. 4.3.6).

Using NN emulations will speed up the calculations of all three types of ensembles (PICE, PPE, and STPPE). One PICE or PPE run with N ensemble members using N different NN emulations, each of which is n times faster than the original model physics, as perturbed versions of model physics will take time,

$$T_{\mathrm{PPE}}^{\mathrm{NN}} = N \cdot \left[\left(1 - \frac{1}{m} \right) + \frac{1}{m \cdot n} \right] \cdot T, \tag{5.18}$$

where $1/m < 1$ is the fraction of T required for calculation of the model physics. Thus, in the case of NCAR CAM, where $m \approx 3/2$ to 2 and $n \approx 10$–100, using NNs for PICE or PPE will speed up the calculations about a factor of two to three.

Acceleration of the calculations of PICE and PPE due to the use of NN emulations of the model physics is significant. However, the speedup will be much

more significant in the case of STPPE. When we use N NN emulations, each of which is n times faster than the original model physics, the STPPE run takes time

$$T_{\text{STPPE}}^{\text{NN}} = \left[\left(1 - \frac{1}{m}\right) + \frac{N}{m \cdot n}\right] \cdot T. \tag{5.19}$$

Thus, STPPE with $N = n$ ensemble members (N different NN emulations of the model physics taken as ensemble members) can be run as fast as a single ensemble member of PICE or PPE (see Eq. (5.16)).

5.3.3 *Comparisons of Different Ensembles with Perturbed NCAR CAM LWR*

In this section, the efficiency of STPPE is investigated. We seek to determine if this approach improves the accuracy of a climate simulation to a degree at least comparable with improvements provided by the PICE and PPE approaches. For the following experiments, the NCAR CAM was run using the original model physics and the original NCAR CAM initial conditions as the control against which all ensemble members for all three types of the ensembles and ensemble results are evaluated. Namely, the climate simulations obtained from the 15-year run of NCAR CAM with the original model physics (including original LWR parameterization) and the original initial conditions are used below as the control climate (the synthetic "observations") to establish a basis for comparison. All ensemble members and ensemble averages for the different ensembles (PICE, PPE, and STPPE) are compared with these synthetic "observations."

Next, to create an ensemble of perturbed physics, we emulated the original LWR parameterization (Collins et al. 2002) with six different NNs which approximate the original LWR parameterization with different approximation errors (Krasnopolsky et al. 2005; Krasnopolsky and Fox-Rabinovitz 2006).

The perturbed LWR parameterizations can be written as

$$\text{LWR}_j^{\text{NN}} = \text{LWR} + \varepsilon_j, \tag{5.20}$$

where LWR is the original NCAR CAM LWR, $\text{LWR}_j{}^{\text{NN}}$ is NN emulation number j of the original NCAR CAM LWR, and ε_j is the emulation error for NN emulation number j. Thus, the model physics that includes the LWR NN emulation, $\text{LWR}_j{}^{\text{NN}}$, can be considered as jth perturbed versions of the model physics, P_j.

A sufficiently diverse group of six NN emulations have been selected by mixing two different approaches to create an NN emulation ensemble. Five of these six ensemble members (NN emulations of LWR) have the same architecture, that is, the same number of neurons ($k = 150$ in Eq. (2.2)); however, these NNs are different because different initializations for the NN weights have been used. The sixth NN

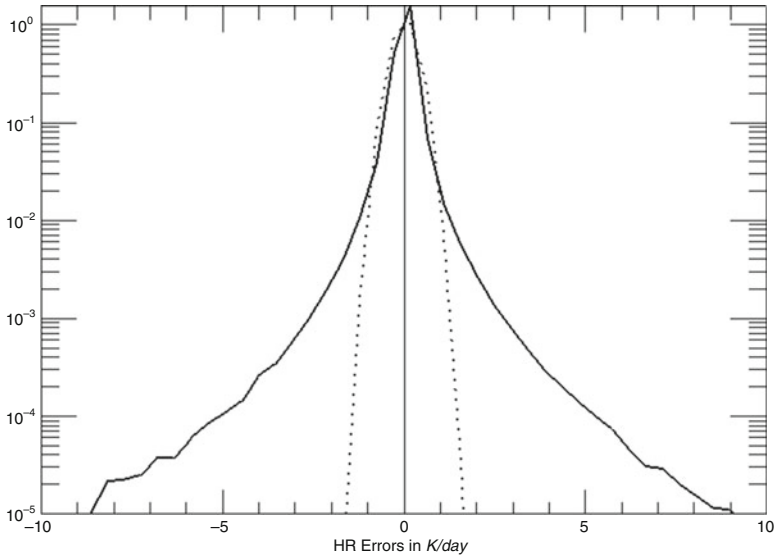

Fig. 5.18 Probability density function for ε_j. The mean $\varepsilon_j = 3.\ 10^{-4}$ K/day and the standard deviation is 0.35 K/day. The *dashed line* shows a normal distribution with the same mean and standard deviation for comparison

emulation ensemble member has a different architecture ($k = 90$ neurons). In terms of the accuracy of the approximation, there is a significant spread between the ensemble members. The approximation RMSE varies from 0.28 to 0.40 K/day for the ensemble member NNs. Thus, by using NN emulations instead of the original LWR parameterization, on average, such a perturbation level is introduced into the LWR model physics.

The distribution of approximation errors (perturbations) is shown in Fig. 5.18. It is obviously not normal. For the normal distribution with the same mean value and SD, the perturbation values would be very limited; however, because the distribution of ε_j is not normal, there is a small but finite probability of obtaining larger perturbations. If compared with the mean value, μ, and SD, σ, of the LWR itself ($\mu = -1.4$ K/day and $\sigma = 1.9$ K/day), the majority of perturbations belong to the interval $\mu \pm \sigma$; however, a very small number of perturbations reach the magnitude of about $\mu \pm 3\sigma$. Such a distribution is in a good agreement with the fact that the parameterizations of the model physics, on average, describe the parameterized processes well and the level of errors introduced due to subgrid effects is moderate; however, in some cases (e.g., rare or extreme events), the errors may be very significant.

In the case of CAM LWR, the NN emulations are about $n = 100$ times faster than the original LWR parameterization. Since calculation of the original CAM LWR takes about 30 % of the model integration time T ($m = 3$ in Eqs. (5.15, 5.16, 5.18, and (5.19)), using LWR NN emulations in PICE and PPE speeds up the calculations

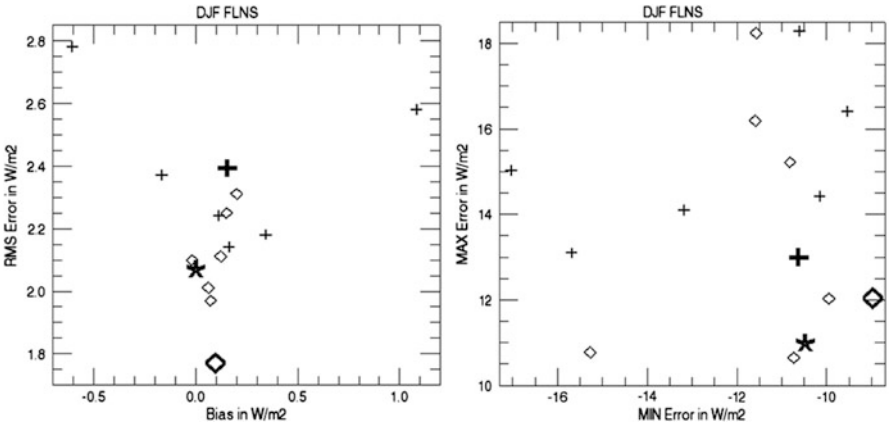

Fig. 5.19 Bias and RMSE (*left panel*) and min and max errors (*right panel*) for the winter DJF (December through February) surface net LWR flux (FLNS). *Diamonds* show the PICE members, *thick large diamond* – the PICE average; *crosses* show the PPE members, *thick large cross* – the PPE average; the *thick large star* shows the STPPE value

by about 30 %. For SHPPE, the use of NN emulations provides a much more significant speedup in the calculations. For this case, an STPPE run with $N = 6$ ensemble members and $m = 3$ takes about 70 % (Eq. 5.18) of the time required for a single ensemble member of PICE (Eq. 5.16).

To run a PICE that is used for comparison purposes, six perturbed initial conditions members were created by randomly perturbing the original initial conditions for the temperature field used in the control run. Then a PICE run (see Fig. 5.16) was performed, consisting of six climate simulations that have been run with NCAR CAM for 15 years, each with one of these six perturbed initial conditions. Next, we performed a PPE run (see Fig. 5.16); six climate simulations have been run with NCAR CAM for 15 years, each with one of the aforementioned six NN emulations. The same NNs were used as the NN ensemble members for STPPE. The results that include climate fields and diagnostics of each simulation (ensemble member) were compared with the control climate run of NCAR CAM performed with the original LWR and original initial conditions. The climate simulation errors – systematic (bias), RMSE, maximum (an extreme positive outlier), and minimum (an extreme negative outlier) – have been calculated for prognostic and diagnostic fields for each ensemble member vs. the control climate. For some of the fields, the errors are shown by diamonds (for PICE members) and crosses (for PPE members) in Figs. 5.19 and 5.20. Then the PICE and PPE averages were calculated (shown by large thick diamonds and crosses, respectively, in Figs. 5.19 and 5.20).

Next, an STPPE climate run has been performed. For this run, the six aforementioned NN emulations were applied and the LWR outputs calculated as the mean of the six NN emulation outputs at each time step and at each grid point throughout the entire model integration. The results of this simulation are compared with those of the control run and shown by large thick star in Figs. 5.19 and 5.20.

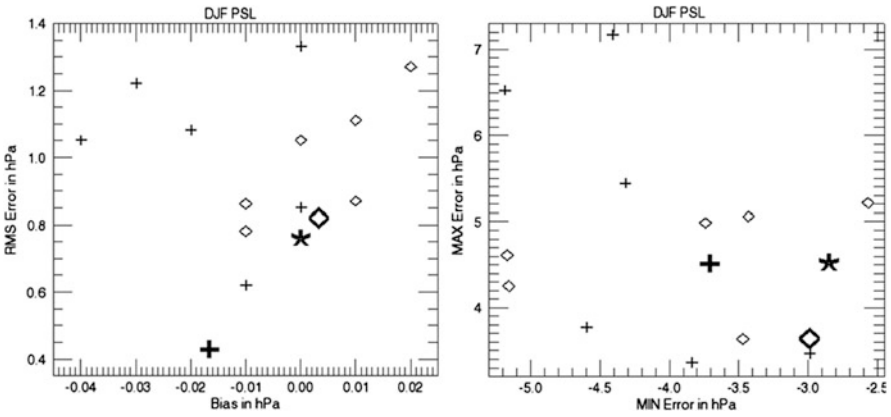

Fig. 5.20 Bias and RMSE (*left panel*) and min and max errors (*right panel*) for DJF pressure at the surface level (*PSL*). See the Fig. 5.19 caption for notations

Figure 5.19 shows the winter DJF (December through February) surface net LWR flux (FLNS) errors in W/m^2, as deviations from the control climate. It is noteworthy that the min and max errors shown in the right panel for this and the following figures are the extreme outliers obtained for the entire 15 years of model integration.

Figure 5.20 shows the errors for the DJF pressure at the surface level (PSL) as deviations from the control (in hPa). The results presented in Figs. 5.19 and 5.20 clearly demonstrate that all three ensemble approaches give similar results in terms of improvements in the accuracy of the climate simulations. Also they show that the PPE generates a significantly larger spread of the ensemble members than does the PICE with a random perturbation of the initial conditions, which is a plus. Similar results have been obtained for the other variables.

5.3.4 Discussion

In this section, an NN emulation technique was introduced as a tool for creating perturbed model physics for use in ensembles with perturbed physics. Also an STPPE approach was introduced. It is shown that the neural network emulation technique allows us to (1) introduce fast versions of model physics (or components of model physics) that can speed up the calculations for any type of ensemble up to 2–3 times, (2) conveniently and naturally introduce perturbations in the model physics (or a component of model physics) and to develop fast versions of perturbed model physics (or fast perturbed components of the model physics), and (3) make the computation time for the entire ensemble (in the case of the STPPE introduced here) comparable with the computation time for one single model run.

All three ensemble approaches – PICE, PPE, and STPPE – demonstrate similar improvements in the climate simulation accuracy. The use of any of these ensembles in the climate simulation significantly reduces the systematic error (bias); it also reduces the random error by making it close to that of the best individual ensemble member. The same is true for the extreme (min and max) errors. Using NN emulations of the model physics significantly improves the computational performance of any of the ensemble techniques that were investigated. However, it is important to emphasize that the STPPE is significantly faster than PICE or PPE.

References

Barai SV, Reich Y (1999) Ensemble modeling or selecting the best model: many could be better than one. AI EDAM 13:377–386

Bleck R (2002) An oceanic general circulation model framed in hybrid isopycnic-cartesian coordinates. Ocean Model 4:55–88

Broccoli AJ, Dixon KW, Delworth TL, Knutson TR, Stouffer RJ (2003) Twentieth-century temperature and precipitation trends in ensemble climate simulations including natural and anthropogenic forcing. J Geophys Res 108(D24):4798–4811. doi:10.1029/2003JD003812

Buizza R (1997) Potential forecast skill of ensemble prediction and spread and skill distributions of the ECMWF Ensemble Prediction System. Mon Weather Rev 125:99–119

Buizza R, Palmer TN (1998) Impact of ensemble size on ensemble prediction. Mon Weather Rev 126:2503–2518

Buizza R, Miller M, Palmer TN (1999) Stochastic representation of model uncertainties in the ECMWF Ensemble Prediction System. Q J R Meteorol Soc 125:2887–2908

Buizza R, Houtekamer PL, Toth Z, Pellerin G, Wei M, Zhu Y (2005) A comparison of the ECMWF, MSC, and NCEP Global Ensemble Prediction Systems. Mon Weather Rev 133:1076–1097

Chassignet EP, Smith LT Jr, Halliwell GR Jr, Bleck R (2003) North Atlantic simulations with the hybrid coordinate ocean model (HYCOM): impact of the vertical coordinate choice, reference pressure, and thermobaricity. J Phys Oceanogr 33:2504–2526

Collins WD, Hackney JK, Edwards DP (2002) A new parameterization for infrared emission and absorption by water vapor in the National Center for Atmospheric Research Community Atmosphere Model. J Geophys Res 107:1–20

Du J, Mullen SL, Sanders F (1997) Short-range ensemble forecasting of quantitative precipitation. Mon Weather Rev 125:2427–2459

Ebert EE (2001) Ability of a poor man's ensemble to predict the probability and distribution of precipitation. Mon Weather Rev 129:2461–2480

Guinehut S, Le Traon PY, Larnicol G, Philipps S (2004) Combining argo and remote-sensing data to estimate the ocean three-dimensional temperature fields – a first approach based on simulated data. J Mar Syst 46:85–98

Kharin VV, Zwiers FW (2000) Changes in the extremes in an ensemble of transient climate simulations with a coupled atmosphere–ocean GCM. J Climate 13:3760–3788

Krasnopolsky VM (2007a) Neural network emulations for complex multidimensional geophysical mappings: applications of neural network techniques to atmospheric and oceanic satellite retrievals and numerical modeling. Rev Geophys. doi:10.1029/2006RG000200

Krasnopolsky VM (2007b) Reducing uncertainties in neural network Jacobians and improving accuracy of neural network emulations with NN ensemble approaches. Neural Netw 20:454–461

Krasnopolsky VM, Fox-Rabinovitz MS (2006) Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. Neural Netw 19:122–134

Krasnopolsky VM, Lin Y (2012) A neural network nonlinear multimodel ensemble to improve precipitation forecasts over continental US. Adv Meteorol, 2012:11pp. Article ID 649450, doi:10.1155/2012/649450. http://www.hindawi.com/journals/amet/2012/649450/

Krasnopolsky VM, Fox-Rabinovitz MS, Chalikov DV (2005) New approach to calculation of atmospheric model physics: accurate and fast neural network emulation of long wave radiation in a climate model. Mon Weather Rev 133:1370–1383

Krasnopolsky VM, Lozano CJ, Spindler D, Rivin I, Rao DB (2006) A new NN approach to extract explicitly functional dependencies and mappings from numerical outputs of numerical environmental models. In: Proceedings of the IJCNN2006, Vancouver, BC, Canada, 16–21 July, pp 8732–8734

Krasnopolsky VM, Fox-Rabinovitz MS, Belochitski A (2008) Ensembles of numerical climate and weather prediction models using neural network emulations of model physics. In: Proceedings of the 2008 IEEE World congress on computational intelligence, Hong Kong, 1–6 June, paper NN0498, pp 1524–1531

Krasnopolsky V, Fox-Rabinovitz M, Belochitski A, Rasch P, Blossey P, Kogan Y (2011) Development of neural network convection parameterizations for climate and NWP models using cloud resolving model simulations. NCEP office note 469. http://www.emc.ncep.noaa.gov/officenotes/newernotes/on469.pdf

Krishnamurti TN, Kishtawal CM, LaRow TE, Bachiochi DR, Zhang Z, Williford CE, Gadgil S, Surendran S (1999) Improved weather and seasonal climate forecasts from multimodel superensemble. Science 285:1548–1550

Krishnamurti TN, Kishtawal CM, Zhang Z, LaRow T, Bachiochi D, Williford E, Gadgil S, Surendran S (2000) Multimodel ensemble forecasts for weather and seasonal climate. J Climate 13:4196–4216

Lin Y, Krasnopolsky VM (2011) Simple- and modified-poor man's QPF ensembles, and a neural network approach. In: 91th annual AMS meeting, AMS 24th conference on weather and forecasting/20th conference on numerical weather prediction, Paper 6A.2

Mellor GL, Ezer T (1991) A Sulf Stream model and an altimetry assimilation scheme. J Geophys Res 96:8779–8795

Murphy JM, Sexton DM, Barnett DN, Jones GS, Webb MJ, Collins M, Stainforth DA (2004) Quantification of modelling uncertainties in a large ensemble of climate change simulations. Nature 430:768–772

Novak DR, Bailey C, Brill K, Eckert M, Petersen D, Rausch R, Schichtel M (2011) Human improvement to numerical weather prediction at the Hydrometeorological Prediction Center. In: 91th annual AMS meeting, AMS 24th conference on weather and forecasting/20th conference on numerical weather prediction, Paper No 440

Palmer TN et al (2007) The ensemble prediction system – recent and ongoing developments. ECMWF Tech Memorandum No 540

Speer MS, Leslie LM (1997) An example of the utility of ensemble rainfall forecasting. Aust Meteorol Mag 46:75–78

Stainforth DA et al (2005) Uncertainty in predictions of the climate responses to rising levels of greenhouse gases. Nature 433:403–406

Stensrud DJ, Bao J-W, Warner TT (2000) Using initial condition and model physics perturbations in short-range ensemble simulations of mesoscale convective systems. Mon Weather Rev 128:2077–2107

Tang Y, Hsieh WW (2003) ENSO simulation and prediction in a hybrid coupled model with data assimilation. J Meteorol Soc Jpn 81:1–19

Wilks DS (2011) Statistical methods in the atmospheric sciences, 3rd edn. Academic, San Diego

Yoshimori M, Stocker TF, Raible CC, Renold M (2005) Externally forced and internal variability in ensemble climate simulations of the maunder minimum. J Climate 18:4253–4270

Zhang Z, Krishnamurti TN (1997) Ensemble forecasting of hurricane tracks. Bull Am Meteorol Soc 78:2785–2795

Chapter 6
Conclusions

Science is facts. Just as houses are made of stones, so science is made of facts. But a pile of stones is not a house and a collection of facts is not necessarily a science.

– Jules Henri Poincare, *Science and Hypothesis*

Abstract In this chapter, a summary of the material introduced in the book is presented. Advantages and limitations of neural network (NN) techniques are discussed. Some other statistical learning techniques, such as the nearest neighbor approximation, the regression tree, and the Random Forest approximation, are briefly discussed. Their performances are compared with the performance of the NN emulation for the case when these techniques are applied to emulate a long wave radiation parameterization in an atmospheric model. The chapter contains a list of references giving background and further detail to the interested reader on each examined topic.

During the last several decades, an objective generic trend emerged in ESS, a transition from studying simple, low-dimensional, single-disciplinary linear, or weakly nonlinear geophysical processes and systems to those that are complex, multidimensional, interdisciplinary, and nonlinear. This trend is closely matched by a trend in ESS modeling (including statistical modeling), a transition from simple, low-dimensional, linear, or weakly nonlinear models to complex, multidimensional, coupled, nonlinear models, and from simple, linear statistical tools like linear models and linear regressions to sophisticated, nonlinear statistical tools like nonlinear regressions, NNs, and support vector machines.

Transitioning to nonlinear models and statistical tools has shown their generally greater adequacy in addressing the problems that are an integral part of modern ESSs. This transition and the following intensive use of nonlinear statistical tools and models have also revealed that their complexity and flexibility, if not properly managed, may lead in some cases to undesirable results and erroneous predictions.

V.M. Krasnopolsky, *The Application of Neural Networks in the Earth System Sciences*, Atmospheric and Oceanographic Sciences Library 46, DOI 10.1007/978-94-007-6073-8_6,

What should the strategy be in this situation? To return to simple linear tools and models? Unfortunately, we cannot do that because the objects of our study are essentially complex and nonlinear. In my opinion, the only productive approach in this situation is to apply to the objects of our study extensively nonlinear statistical tools and models that are up to the task and to learn, through experience, how to minimize or eliminate possible undesirable side effects while maximizing the advantages that the complexity and flexibility of these nonlinear models and tools offer.

In the discussions included in each chapter of this book, I have tried to emphasize that the transition from linear statistical tools to nonlinear ones (such as NNs) requires, to some extent, an adjustment in our thinking and our philosophy. For example, when we deal with relatively simple linear systems and use linear statistical tools (such as simple linear regression) to model these systems, we can assume, in some cases, that the parameters of our statistical model have physical meaning, that they are directly related to the characteristics of the physical processes involved and/or tell us something about the physical structure of the system under consideration. When we deal with complex nonlinear systems and apply nonlinear statistical tools, we probably should focus primarily on obtaining the best predictions possible of the behavior of the system under consideration, but not on attempting to extract physical meaning from the multiple parameters of our nonlinear model (Vapnik and Kotz 2006).

6.1 Comments About NN Technique

In this book, I presented and discussed one particular type of NN technique – the MLP NN – and one generic application of this NN, NN emulations for complex multidimensional mappings. It was shown that even this one generic NN application covers a wide variety of important problems in atmospheric and oceanic sciences and can provide us with flexible, accurate, and fast nonlinear solutions for these problems. There are other generic applications, for example, classification problems (Hansen and Salamon 1990; Sharkey 1996; Opitz and Maclin 1999), that can be successfully solved using MLP NNs (Lippmann 1989; Marzban and Stumpf 1996; Hennon et al. 2005). There are also other types of NNs that provide solutions for other generic applications, like pattern recognition problems (Ripley 1996; Nabney 2002) or time series prediction (Weigend and Gershenfeld 1994), but this book does not deal directly with these types of NNs and their applications. However, many generic issues are discussed here, like NN building blocks, the complexity and dimensionality of the problem and corresponding complexity and dimensionality of nonlinear models (e.g., NNs) that provide solutions of the problem, and NN generalization capabilities. These basic issues are important and are applicable to other types of NNs and other NN applications.

While artificial neural networks can be considered a generic, state-of-the-art methodology for a wide range of high-dimensional approximation problems, they may not necessarily be the best solution for these applications. Although it is known that neural networks are universal approximators, i.e., they can approximate any continuous function to any predetermined accuracy (DeVore et al. 1997), this is achieved only by allowing the number of neurons to increase arbitrarily. Also, the learning of the network parameters (weights) requires the solution of a large, nonlinear optimization problem, which can deliver suboptimal solutions.

While the application of neural networks to the problem of emulating complex multidimensional mappings considered in this book has produced excellent results, it is not without its limitations. Foremost among these is that the neurons that are represented by sigmoid or hyperbolic tangent functions are not localized and their superposition is a complex nonorthogonal expansion. Thus, local features are reflected in many or all terms of the function expansions. This is analogous to Fourier methods. One of the great advances of the last few decades in image processing has been to replace Fourier methods by the more local wavelet methods. The aforementioned nonlocality makes the capturing of local or multiscale phenomena difficult. In our use of NN emulations, we also notice that NNs can sometimes exhibit relatively large errors (however, with small probability). Thus, although both the systematic and the random errors are very small, there exists a very small probability of larger errors, which needs to be avoided (an approach for solving this problem using NNs is described in Sect. 4.3.5).

Additionally, the approximation of the target mapping is trained using a data set that consists of observed or simulated evaluations of the original mapping gained during observations or numerical simulations. The inputs of this training data set, therefore, cover, in terms of the physics represented by the mapping, the physical states observed during a certain time period. However, the domain in which the mapping is to be evaluated may change with time as in the case of a changing climate. In such situations the approximation may be forced to extrapolate beyond its generalization ability, which may lead to larger errors. In this case it could become necessary to retrain the emulation in order to adapt it to the new environment.

This brings into question whether NNs are the ultimate SLT solution for the task of numerically emulating the mappings considered in this book. Indeed, since our goal is often to capture subtle multiscale phenomena, a more application-oriented, responsive, and adaptive learning method could be useful. However, talking about adaptive techniques, we should keep in mind that the success of such approaches depends very much on what is known a priori about the particular problem under consideration. In this respect, adaptive techniques are similar to inverse problems, which require additional information for their regularization. In both cases, solution of the problem is very sensitive to even small changes of regularization (a priori) information and/or additional assumptions that are introduced to make the problem resolvable (Novak 1996). Some of alternative CI approaches are reviewed in the next section.

6.2 Comments About Other Statistical Learning Techniques

The major SLTs are kernel methods (Vapnik and Kotz 2006; Hsieh 2009), NNs, and nearest neighbor algorithms (see Shakhnarovich et al. 2006 for general references). Each approach has tradeoffs. The advantage of NNs and kernel methods is that they can be implemented in high dimensions without meshing or splitting the domain of the approximated mapping into subdomains. Their disadvantage, as mentioned above, is twofold. Firstly, they are not local, by which we mean the building blocks are not locally supported functions and may require a long series of these building blocks to approximate a simple but localized function. However, it does not apply to radial basis function NNs that use Gaussian basis function. A second disadvantage is that they are not explicitly adaptive to changes in the underlying function or mapping.

Nearest neighbor methods have a local flavor but they are also typically not implemented adaptively. In other words, the rules for identifying nearest neighbors do not take into account the variability in the underlying function (which will be reflected in the data). Nevertheless this approach is closer to the methods we would like to employ. The severe obstructions encountered by this method when dealing with problems in higher spatial dimensions is the curse of dimensionality (see Sects. 2.2.1, 2.3.1, and 2.4.1).

To the best of our knowledge, the first attempt to apply methods other than NNs to emulate some of the complex mappings discussed in this book was performed by Belochitski et al. (2011). In this work, we used several methods within the class of nonparametric approximation methods as alternatives to neural networks. Nonparametric learning methods typically attempt to partition the input space and then use simple local models like piecewise constants to approximate the data. Two common statistical learning paradigms – (1) (approximate) nearest neighbors and (2) regression trees – were considered. Three methods belonging to these paradigms – (1) the approximate nearest neighbor approximation, (2) the regression tree, and (3) the Random Forest approximation – were considered (Belochitski et al. 2011). They have been applied to emulate the NCAR CAM LWR parameterization and have been compared with the NN emulation developed for this parameterization in Sect. 4.3.3.

The major results obtained in this study are (for a more detailed discussion, see Belochitski et al. 2011):

1. Both nearest neighbor type methods and regression trees are, in principle, able to achieve statistical approximation quality comparable with NNs; the Random Forest approximation has even better RMSE scores than NN emulations.
2. The NCAR CAM with a tree-based LWR emulation gave results in close agreement with calculations using the original parameterization; however, even with the best Random Forest approximation, the results are not as good as with a NN emulation. This seeming contradiction with the issues 1 points to the major advantage of the NN emulation technique as compared with tree-based methods.

For the case of multiple correlated outputs (LWR heating rate profile) considered here, NN is the only method capable of utilizing the correlation information. Namely, the NN emulation works with the LWR heating rate profile as with a single entity, preserving strong correlations between neighboring vertical levels of the profile. The tree-based methods considered here perform approximation level by level, independently for each element of the profile, completely missing the correlation information. As a result, despite comparable approximation accuracy, in the model, where vertical correlations in the heating rate profile are important, tree-based methods do not perform as well as NN LWR.

3. From a practical point of view, these methods cannot compete with NN emulations in terms of speeding up the calculations; also, nonparametric approximation methods are memory based, i.e., they need to store all the training data permanently. This makes their use in a parallel environment much more difficult than is the case with the relatively compact NN emulations.

Therefore, the ideas and results presented in Belochitski et al. (2011) can only be considered as preliminary steps toward a new emulation SLT, which can be an alternative to NN for emulating complex mappings. Thus, the results of this study demonstrate that, at this time, the NN is probably the only practical SLT tool for solving the problems discussed in this book and more generally for emulating complex multidimensional mappings.

References

Belochitski AP, Binev P, DeVore R, Fox-Rabinovitz M, Krasnopolsky V, Lamby P (2011) Tree approximation of the long wave radiation parameterization in the NCAR CAM global climate model. J Comput Appl Math 236:447–460

DeVore R, Oskolkov K, Petrushev P (1997) Approximation by feed-forward neural networks. Ann Numer Math 4:261–287

Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal 12:993–1001

Hennon CC, Marzban C, Hobgood JS (2005) Improving tropical cyclogenesis statistical model forecasts through the application of a neural network classifier. Weather Forecast 20:1073–1083. doi:10.1175/WAF890.1

Hsieh WW (2009) Machine learning methods in the environmental sciences. Cambridge University Press, Cambridge

Lippmann RP (1989) Pattern classification using neural networks. IEEE Commun Mag 27:47–64

Marzban C, Stumpf GJ (1996) A neural network for tornado prediction based on doppler radar-derived attributes. J Appl Meteorol 35:617–626

Nabney IT (2002) Netlab: algorithms for pattern recognition. Springer, New York

Novak E (1996) On the power of adoption. J Complex 12:199–237

Opitz D, Maclin R (1999) Popular ensemble methods: an empirical study. J Artif Intell Res 11:169–198

Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge

Shakhnarovich G, Darrell T, Indyk P (2006) Nearest-neighbor methods in learning and vision: theory and practice. MIT Press, Cambridge, MA

Sharkey AJC (1996) On combining artificial neural nets. Connect Sci 8:299–313
Vapnik VN, Kotz S (2006) Estimation of dependences based on empirical data (information science and statistics). Springer, New York
Weigend AS, Gershenfeld NA (1994) The future of time series: learning and understanding. In: Weigend AS, Gershenfeld NA (eds) Time series prediction. Forecasting the future and understanding the past. Addison-Wesley Publishing Company, Reading, pp 1–70

Index

V.M. Krasnopolsky, *The Application of Neural Networks in the Earth System Sciences*, Atmospheric and Oceanographic Sciences Library 46, DOI 10.1007/978-94-007-6073-8,