Neural Network Emulations for Complex Multidimensional Geophysical Mappings: Applications of Neural Network Techniques to Atmospheric and Oceanic Satellite Retrievals and Numerical Modeling

Vladimir M. Krasnopolsky[1,2]

[1]SAIC at EMC/NCEP/NOAA, MD, USA, vladimir.krasnopolsky@noaa.gov
[2]Earth System Science Interdisciplinary Center, University of Maryland, MD, USA

Corresponding author address:
Vladimir Krasnopolsky,
EMC/NCEP/NOAA
5200 Auth Rd.,
Camp Springs, MD 20746-4304;
Phone: 301-763-8000 x 7262;
Fax: 301-763-8545;
E-mail: Vladimir.Krasnopolsky@noaa.gov

Revised and submitted to Reviews of Geophysics
January 2007

Contents

Abstract....4
1 Introduction....5
2 Mapping and Neural Networks Background....7
2.1 Mapping Examples....8
2.2 Some Generic Properties of Mappings....10
2.2.1 Mapping dimensionalities, domain, and range....10
2.2.2 Mapping complexity....11
2.2.3 Mappings corresponding to ill-posed problems....14
2.3 MLP NN – A Generic Tool for Modeling Nonlinear Mappings....14
2.3.1 NNs in terms of approximation theory....14
2.3.2 NNs in traditional NN terms....16
2.3.3 Training set....16
2.3.4 Selection of the NN architecture....19
2.3.5 NN training....21
2.4 Advantages and Limitations of the NN Technique....22
2.4.1 Flexibility of the MLP NN....23
2.4.2 NN training as nonlinear optimization and multi-collinearity of inputs and outputs....24
2.4.3 NN generalization ability: interpolation and extrapolation. NN emulations....25
2.4.4 NN Jacobian....26
2.4.5 Multiple NN emulations for the same target mapping and NN ensemble approaches....28
2.4.6 NN uncertainties estimates....30
2.5 Discussion....31
3 Atmospheric and Oceanic Remote Sensing Applications....31
3.1 Deriving Geophysical Parameters from Satellite Measurements: Standard Retrievals and Variational Retrievals Obtained Through Direct Assimilation....33
3.1.1 Standard or conventional retrievals....34
3.1.2 Variational retrievals through the direct assimilation of satellite measurements....38
3.2 NNs for Emulating Forward Models....39
3.3 NNs for Solving Inverse Problems: NNs Emulating Retrieval Algorithms....40
3.4 Controlling the NN Generalization....40
3.5 Neural Network Emulations for SSM/I Data....42
3.5.1 NN emulation of the empirical FM for SSM/I....43
3.5.2 NN empirical SSM/I retrieval algorithms....44
3.6 Discussion....47
4 Applications of NNs to Developing Hybrid Atmospheric and Oceanic Numerical Models....48
4.1 Concepts of a Hybrid Model Component and a Hybrid Model....51
4.2 Hybrid parameterizations of physics....53
4.3 Hybrid Numerical Models: Accurate and Fast NN Emulations for Parameterizations of Model Physics 56
4.4 Atmospheric Applications: NN Emulation Components and HGCM....59
4.4.1 Models and Statistical Metrics....59
4.4.2 Emulating the Model Radiation....61
4.5 Ocean Application of the Hybrid Model Approach: Neural Network Emulation of Nonlinear Interactions in Wind Wave Models....67

4.6 Discussion 70
4.6.1 Summary and advantages of the hybrid modeling approach 70
4.6.2 Limitations of the current hybrid modeling framework 71
4.6.3 Current and future developments of the hybrid modeling approach 72
5 NN Emulations of Dependencies Between Model Variables 75
5.1 SSH Mapping and its NN Emulation 76
5.2 NN Ensembles for improving NN emulation accuracies and reducing NN Jacobian uncertainties 79
5.2.1 NN ensembles for improving the accuracy of a NN emulation; linear vs. nonlinear ensembles 80
5.2.2 NN ensembles for reducing the uncertainty of the NN Jacobian 82
5.3 Discussion 86
6 Conclusions 87
Acknowledgements 89
References: 90
Tables 106
Figure Captions 110
Figures 112

Abstract

A group of geophysical applications, which from the mathematical point of view can be formulated as complex, multidimensional, nonlinear mappings and which, in terms of the neural network (NN) technique, utilize a particular type of NN – the multilayer perceptron (MLP), is reviewed in this paper. This type of NN applications covers the majority of NN applications developed in geosciences like satellite remote sensing, meteorology, oceanography, numerical weather prediction and climate studies. The major properties of the mappings and MLP NNs are formulated and discussed. Three particular groups of NN applications are presented in this paper as illustrations: atmospheric and oceanic satellite remote sensing applications; NN emulations of model physics for developing atmospheric and oceanic hybrid numerical models; and NN emulations of the dependencies between model variables for application in data assimilation systems.

1 Introduction

The neural network (NN) approach is a relatively new, diverse, and powerful statistical learning technique (a.k.a. machine learning, learning from data, predictive learning, data driven approach) that started developing rapidly in the mid 80s 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 90s this technique matured; several well written and fundamental textbooks have been published [*Beale and Jackson,* 1990; *Bishop,* 1995; *Haykin,* 1994; *Ripley,* 1996; *Vapnik,* 1995; *Cherkassky and Mulier*, 1998] that introduced NNs as a new powerful statistical learning approach capable of providing a diverse family of flexible nonlinear data driven models for various applications. This approach became appealing to a broad community of professionals including scientists working in different fields of geosciences like satellite remote sensing, meteorology, oceanography, geophysical numerical modeling, etc. Since then a significant number of NN applications have been developed in these fields; the most important of them are summarized in Table 1. References presented there do not provide a complete list of the corresponding publications, or even the most important ones. Rather, they give 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 and oceanic applications have been reviewed for the atmospheric and oceanic community by *Gardner and Dorling* [1998] and by *Hsieh and Tang* [1998], and for the NN professionals by *Krasnopolsky and Chevallier* [2003] and *Krasnopolsky and Fox-Rabinovitz* [2006b]. Selected remote sensing applications have been reviewed for remote sensing 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]. *Solomatin* [2005] has reviewed hydrological NN applications.

Table 1

As can be seen from Table 1, a great variety of NN applications has been developed in different weather and climate related fields. These applications utilize different types of NNs. The task of selecting a group of such applications for reviewing may be approached from different directions. Our goal in this paper is to be a tutorial rather than to give a complete description of geophysical NN applications. We will leave out some particular interesting applications and try to concentrate on a clear presentation (where it is possible) of the methodological basis of the selected applications. In this venue, we selected a group of geophysical applications for review in this paper, which from the mathematical point of view, can be formulated as complex, multidimensional, nonlinear mappings and which utilized a particular type of NN from the point of view of the NN technique – the multilayer perceptron (MLP) [*Rumelhart* et al. 1986]. This framework is broad and generic. It covers the majority of applications developed in geosciences that can be considered as complex, multidimensional, nonlinear mappings and are partially presented in Table 1. To focus this paper even more, we selected as particular examples only those applications which were developed with the author's participation. They are shown in italics in Table 1.

In the methodological Section 2, we introduce the concept and major properties of complex nonlinear mappings. We also introduce the MLP NN as a generic technique for the nonlinear 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 Section 2 is a collection of theoretical results and practical inferences based on numerical experiments and experience from applications of MLP technique to various problems. This tutorial material is valuable for understanding the entire topic and the following parts of this paper. We would recommend to readers interested in a deeper understanding of the technique to return to Section 2 during and after reading the rest of this paper. In Section 3, we discuss using NNs for remote sensing applications: emulating solutions of forward and inverse problems in satellite remote sensing. In Section 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 (physical) principles) model components. In Section 5, we introduce a NN application that allows the creation of NN emulations for the functions and mappings between model state variables hidden in the numerical outputs of modern high resolution atmospheric and oceanic numerical models. There we also review NN ensemble approaches that allow improvement of the accuracy of NN emulations and reduce the uncertainties of NN Jacobians. Section 6 contains conclusions.

2 Mapping and Neural Networks Background

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

$$Y = M(X); \quad X \in \Re^n, Y \in \Re^m \tag{1}$$

A large number of important practical geophysical applications may be considered mathematically as a mapping like (1). Keeping in mind that a NN technique will be used to approximate this mapping, we will call it a **target mapping**, using a common term from

nonlinear approximation theory [*DeVore*, 1998]. The target mapping may be given to us explicitly or implicitly. It can be given 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. Observational records represent an implicit target mapping. In this case, the target mapping is assumed to be hidden in, or behind observed data and to generate these data.

2.1 Mapping Examples

Prediction of a geophysical time series may be considered as a mapping between the past and future [*Elsner and Tsonis*, 1992]. 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, predicted values of the same variable X. The components of vectors X and Y in this example may be significantly correlated as sequential terms of the same time series that represents the sequential measurements of the same physical process. Depending on the nature of the process represented by the time series, the target mapping M may be linear or nonlinear [*Elsner and Tsonis*, 1992]. Prediction of a time series as a mapping, and the use of NNs in this application, are discussed in great detail in Weigend and Gershenfeld [1994] and in other papers of this book.

The second example of a generic application that can be formally considered as the mapping (1), is a retrieval algorithm (or a transfer function) in satellite remote sensing 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, ocean and land surface characteristics, etc. Here the components of vector X may again 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 Section 3.

The third example of an important application that can be considered as the mapping (1) is the parameterizations of atmospheric physics in climate or weather prediction numerical models. The atmospheric long wave radiation (see Section 4 for details) can be considered as a mapping, where the input vector X is composed of several atmospheric state variables like temperature, humidity, ozone concentration, etc., that are functions of height, and some surface characteristics. The output vector Y is composed of one function of height – long wave heating rates, and several heat fluxes. It becomes clear from this brief description that in this case we do not have a vector to vector mapping (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 (1). Also, in this example the components of the vectors X and Y can be significantly correlated because: (i) they are physically related, and (ii) they are related as the discretized values (elements of a profile) of the same continuous function at close height values. The target mapping is also a complicated nonlinear one in this case because the atmospheric radiation processes are complicated nonlinear ones. The target mapping may be continuous or almost continuous, that is, it may contain a finite

number of finite discontinuities (like step functions) due to the impact of highly nonlinear atmospheric moisture processes. This application is discussed in Section 4.

2.2 Some Generic Properties of Mappings.

Multi-dimensional, nonlinear mappings (1) are complicated mathematical objects that are not very well studied. There are many different interesting properties of these mappings that could be discussed. However, after considering the three aforementioned examples, it will be easier for us to focus on some generic properties of the mapping (1) that are typical and important for the applications presented in this review, keeping in mind that our goal is to develop a NN emulation for the target mapping (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: (i) dimensionality n of the input space $\Re^n$, and (ii) dimensionality m of the output space $\Re^m$. The second property of the mapping (1) is the **mapping domain**. If all components of the input vector X are scaled to [-1.,1.], the volume of the input space $\Re^n$ is equal to 2^n and, therefore, grows exponentially with n. Ones the space is discretized, e.g., by K values per dimension then the problem size grows even faster, as K^n. This is usually called the *curse of dimensionality* [*Bishop*, 1995; *Vapnik and Kotz*, 2006]. Fortunately, the components of the input vector, X, are usually inter-related or multi-collinear [*Aires et al.* 2004b] due to the physical or statistical reasons that leads to both positive and negative consequences (see Sections 2.3.4 and 2.4.2). These correlations effectively reduce the size, and sometimes dimensionality, of the part of the input space $\Re^n$ spanned by the input vectors X [*Bishop*, 1995]. This part is called the mapping **domain, *D***, and is determined by a particular application. Understanding the

configuration of the mapping domain and its properties is very important for a proper NN training and application. The components of the output vector, Y, are usually also inter-related. 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 developed NN approximations of a target mapping (1).

2.2.2 Mapping complexity

Another property of the mapping (1) that is important in the context of the applications reviewed in this paper is the **mapping complexity**. Mapping complexity is an intuitively clear notion. The mapping M performs some 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 face a lot of problems and ambiguities on both the qualitative and quantitative levels (for a good introductory review of the topic and the related problems see *Reitsma* [2001] and the references therein). Many different qualitative definitions (actually, more than 30 according to *Reitsma* [2001]) of the complexity have been introduced. For some of these qualitative definitions, their measures (quantitative definitions of complexity) can be introduced [*Colliers*, 2000; *Gell-Mann and Lloyd*, 1996].

Keeping in mind those three examples of the mapping (1) we introduced above, we can suggest at least three different qualitative definitions of the mapping complexity. In these applications, the target mapping M is a symbolic representation of 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 (1) that corresponds to the complexity of the physical processes represented mathematically by this mapping. Correspondingly, we can introduce quantitative or semi-quantitative characteristics of physical complexity: the number of equations describing the physics, the type of these equations (e.g., linear vs. nonlinear, ODE vs. PDE vs. integro-differential equations), the dimensionality of the equations, etc. It is noteworthy that an ambiguity may exist for such measures of the physical complexity because 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 differing estimates of physical complexity may be obtained for the same target mapping (1). As an example, Euler vs. Lagrange formulations of the equations of geophysical fluid dynamics can be pointed out.

The second type of complexity that can be introduced is *mapping numerical or 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 the elementary numerical operations. Here again, several differing estimates of the numerical complexity may be obtained for the same mapping (1).

The third definition of mapping complexity is the *functional complexity*. It describes the complexity of the functional dependency of the outputs, Y, vs. inputs, X, or the "smoothness" of this dependency. If the two previous definitions in some respects depend

on, or are conditioned by, our knowledge of the internal structure of the target mapping (1), this third 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 (1) can, in principle, be measured unambiguously. Unfortunately, it does not mean that there actually exist technical tools that allow the introduction of a satisfactory measure of the functional complexity of a multidimensional mapping (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 the order n, etc. The direct generalization of this approach for the case of a multidimensional mapping (1) is hardly possible. However, a similar idea can be applied using universal mapping approximators like MLP NNs [*Cilliers*, 1998]. In other words, the complexity of the emulating NN can be used to measure the complexity of the target mapping (1) to be emulated by this NN. This approach looks attractive; however, it requires a clear definition of the accurate emulating NN, which we discuss in Subsection 2.4.3. It also provides us with a measure of the functional mapping complexity *post factum*, after the accurate NN approximation is performed; however, as we will show in the following sections, 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 three types of mapping complexity considered above; however, they can not be used as unambiguous measures of these complexities.

2.2.3 Mappings corresponding to ill-posed problems

Among the applications considered in this review, we will find some problems that can be considered as continuous unique mappings (1); however, for these mappings small perturbations in X may cause large changes in Y. The inverse mapping in this case may be discontinuous and the problem is then called ill-posed [*Vapnik*, 1995]. Ill-posed problems usually arise when one attempts *to estimate an unknown cause from observed effects* (most of the geophysical inverse problems belong to this class, e.g., the satellite retrieval problem considered in Section 3) or *to restore a whole object from its low dimensional projection* (e.g., estimating the NN Jacobian considered in Section 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 solution approach [*Vapnik and Kotz*, 2006].

2.3 MLP NN – A Generic Tool for Modeling Nonlinear Mappings

2.3.1 NNs in terms of approximation theory

The simplest MLP NN, which in traditional NN terms corresponds to a MLP NN with one hidden layer and a linear output layer, is a generic analytical nonlinear approximation or model for mapping, like the target mapping (1). The MLP NN uses for the approximation a family of functions like:

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

$$z_j = \phi(b_{j0} + \sum_{i=1}^{n} b_{ji} \cdot x_i) \tag{3}$$

where x_i and y_q are components of the input and output vectors respectively, a and b are fitting parameters or NN weights, and ϕ is a so called activation or "squashing" function (a

nonlinear function, often a hyperbolic tangent), n and m are the numbers of inputs and outputs respectively, and k is the number of the highly nonlinear basis function z_j (3) in the expansion (2). The expansion (2) is a linear expansion (a linear combination of the basis function z_j (3)) and the coefficients a_{qj} ($q = 1,\ldots,m$ and $j = 1,\ldots,k$) are linear coefficients in this expansion. It is essential (see Subsection 2.4.1) that the basis functions z_j (3) are nonlinear with respect to inputs x_i (i=1,…,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 $\{z_j\}_{j=1,\ldots,k}$ turns into a very flexible set of non-orthogonal basis functions that have a great potential to adjust to the functional complexity of the mapping (1) to be approximated. It has been shown by many authors in different contexts that the family of functions (2,3) can approximate any continuous or almost continuous (with a finite number of finite discontinuities, like a step function) mapping (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 (1) is proportional to the number of basis functions (hidden neurons) k [*Attali and Pages,* 1997].

The MLP NN (2,3) itself is a particular type of the mapping (1). In the case of the MLP NN, the **computational and functional complexities** of the NN mapping are closely related and can be characterized by the number of fitting parameters a and b in (2,3) (they are especially close for NN emulations, see Section 2.4.3). This number, the complexity of the MLP NN, is

$$N_c = k \cdot (n + m + 1) + m \tag{4}$$

For a set of NNs approximating a particular target mapping (1) with a given number of inputs n and outputs m, a good measure of the NN complexity is the number of hidden neurons k.

2.3.2 NNs in traditional NN terms

A pictographic language reminiscent of 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 of a neural network, a basis function z_j (3), or neuron was represented by a figure like Fig. 2. Then after *Rumelhart* et al. [1986] introduced the MLP NN, a pictographic representation of the entire NN was introduced (see Fig. 1). The neurons are situated into **layers** inside the MLP NN. The **connections** (arrows) in Fig. 1 correspond to the NN **weights**, the name used for fitting parameters a and b in NN jargon. For the simplest type of MLP NN that we consider here and that is sufficient for the approximation of any continuous and almost continuous mappings, there is a one to one correspondence between Eqs. (2,3) and Figs. 1 and 2. However, in general, the pictographic language (Figs. 1 and 2) 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. (2,3). The pictograms that represent the design of such NNs cannot be described by a closed set of equations; however, these pictograms can be easily translated into computer codes.

Fig. 1
Fig. 2

2.3.3 Training set

In a practical application, a target mapping (1) is usually represented and presented to the NN by a data set (training set) that consists of N pairs or records of input and output vectors X and Y,

$$C_{\boldsymbol{T}} = \{X_p, Y_p\}_{p=1,\ldots,N} \tag{5}$$

where $Y_p = M(X_p) + \xi_p$, ξ_p is an error of observation or calculation with the probability density function $\rho(\xi)$, and $X_p \in D$ and $Y_p \in R$. The training set is all that the NN knows about the target mapping that it is supposed to approximate. This is the reason why the MLP NN belongs to a class of data driven methods [*Cherkassky and Mulier*, 1998].

The training set represents the mapping (1) to the NN and, therefore, it has to be *representative*. It means that the training set has to have a sufficient complexity corresponding to the complexity of the target mapping, allowing the NN to obtain the desired accuracy for the approximation of the target mapping. The set should have a sufficient number N of properly distributed data records that adequately resolve the functional complexity of the target mapping (1). The set should have finer resolution where the target mapping is not very smooth and coarser resolution where it is smoother, namely, the domain D should be properly sampled. It may be over-sampled but not under-sampled. The paramount question remains, however, as to just how we should measure this target mapping smoothness in order to obtain desired results [*DeVore*, 1998]. As we discussed above, the inter-relations between inputs 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). *Kon and Plaskota* [2001] attempted to introduce a qualitative measure for the representativeness or necessary complexity of the training set. They introduced *informational complexity*, that is, the number of observations necessary and sufficient to construct a NN approximation under accepted assumptions. In an ideal situation *it should be a correspondence between the functional complexity of the target*

mapping (1), the complexity (4) of the approximating NN, and the informational complexity (number and distribution of data points) of the training set (5). Unfortunately, there are no general recipes for practical real life applications. The only practical, useful relationship that can be found in the literature is $N > N_c$. Actually, this relationship has a simple statistical meaning; the number of unknown parameters in the model (the number of weights in NN or the complexity of NN (4)) should not exceed the number of data points, N, in the training set (5).

Two major types of data are usually used in the geophysical applications considered here. The first type of data is the observed data. These data usually contain a significant level of observational noise ξ. In the case of the ill-posed problem (see Sections 2.2.3 and 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 the target mapping is represented implicitly by the data. The accuracy of the NN approximation and the ability of the NN approximation to resolve the target mapping are limited by the observation setup, technique, and conditions. There is usually little we can do to improve or expand the data set in this case 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 (1) is available, it can be used to simulate the data set (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, over 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 and observed data can, in principle, be fused together in 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 fused data is the analyzed data produced by a data assimilation system.

2.3.4 Selection of the NN architecture

To approximate a particular target mapping (1) with the MLP NN (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 (1) as an elementary/single object and approximate its functionality (an input-output relationship) as a whole. Practical implementation of this approach allows for multiple solutions in terms of the number of NN designs that can be used for an approximation. The MLP NN presented by Eqs. (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 approximating NN is a convenient solution because of the simplicity of its design. It also has a significant 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 (2,3) with many outputs, all the outputs are different linear combinations of the same basis functions z_j, or hidden neurons. Fewer neurons are required to approximate a particular number of correlated outputs than to approximate the same number of uncorrelated ones. Thus, in the

case of correlated outputs one NN per approximation solution has a lower complexity N_c (4), and provides significantly higher performance at the same approximation accuracy than a battery of m single-output NNs [*Krasnopolsky and Fox-Rabinovitz,* 2006b].

The possible choices among many topological solutions, from a single NN with m outputs to m single-output NNs, demonstrate an important flexibility of the NN technique that offers a speed vs. accuracy trade-off. This additional flexibility can be effectively used for various applications. Another degree of flexibility is provided by the availability of different *normalizations* for NN inputs and outputs. This topic is discussed in detail by *Krasnopolsky and Fox-Rabinovitz* [2006b].

The number of hidden neurons k, that determines the complexity (5) 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 approximating 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 below in Section 2.4). Unfortunately, there are no universal rules or recommendations to be given here. Usually k is determined using experience and experiments.

2.3.5 NN training

After these topological parameters are defined, the weights (a and b) can be found using the training set C_T(5) and the maximum likelihood method [*Vapnik*, 1995] by maximizing the functional

$$L(a,b) = \sum_{i=1}^{N} \ln \rho(Y_i - NN(X_i, a, b)) \tag{6}$$

with respect to free parameters (NN weights) a and b. Here, $\rho(\xi)$ is the probability density function for errors ξ_p (see Section 2.3.3). If the errors ξ_p are normally distributed, Eq. (6) leads to the minimization of the least-square error, or loss , or risk, or cost function with respect to the NN weights a and b,

$$E(a,b) = \sum_{i=1}^{N} (Y_i - NN(X_i, a, b))^2 \tag{7}$$

This procedure is usually called NN training. It is noteworthy that for a probability density function $\rho(\xi)$ other than the normal one, the error function should be derived from the maximum likelihood functional (6). The error function may be significantly different than the least-square error or loss function (7). However, in the majority of applications the least-square error function (7) is applied.

Optimal values for weights are obtained by minimizing the error function (6) or (7); this task is a nonlinear minimization problem. A number of methods have been developed for solving this problem [*Bishop*, 1995; *Haykin,* 1994]. Here we briefly outline one of them, a simplified version of the steepest (or gradient) descent method known as the back-propagation training algorithm [*Rumelhart* et al., 1986].

The back-propagation training algorithm is based on the simple idea that searching for a minimum of the error function (7) can be performed step by step in iterations, 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 using, for example, a simple steepest descent rule,

$$\Delta W = -\eta \frac{\partial E}{\partial W} \tag{8}$$

where η is a so-called learning constant and W is either one of two weights (a or b). Using (7) and (2,3), the derivative in (8) can be expressed analytically through the derivative of the activation function ϕ, and through the weight values at the previous iteration step. 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 publications have been devoted to weight initialization [e.g., *Nguyen and Widrow*, 1990; *Wessels and Bernard*, 1992].

A nonlinear error function (7) has multiple local minima. 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 initialization procedures are applied to avoid shallow local minima and to choose a local minimum with a sufficiently small error.

2.4 Advantages and Limitations of the NN Technique

Here we attempt to summarize the advantages and limitations of the MLP NN technique as applied to the emulation of complex multidimensional mappings (1). It is noteworthy that the majority of limitations we discuss here are not limitations on the MLP NN technique *per se*. These limitations are inherent limitations on nonlinear models, nonlinear approximation techniques, and nonlinear statistical approaches [*Cheng and Titterington*, 1994] in general. 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 function z_j (hidden layer neurons) are adjustable. They contain many internal nonlinear parameters b that can be adjusted to the target mapping during training. *Barron* [1993] showed that a linear combination of such adjustable basis functions can provide an accurate approximation with far fewer units than a linear combination of any fixed basis functions for certain classes of mappings where the number of inputs $n \geq 3$. Similar results were demonstrated in a different context by *Krasnopolsky and Kukulin* [1977]. This is a way for the MLP NN to escape the curse of dimensionality.

Another way to look at this important advantage of the adjustable basis z_j, is to demonstrate the independence of the approximation error with respect to the dimensionality n of the input space. In this case the approximation error, $E \leq \frac{\alpha}{k^p}$, where $\alpha > 0$, $p > 0$ and p is independent of n. In contrast, for approximations using fixed basis functions, the approximation error is $E \leq \frac{\alpha}{k^{\frac{1}{n}}}$ for the same class of mapping [*Barron*, 1993; *Cheng and Titterington*, 1994]. It means that for a fixed basis expansion when the number of inputs increases, one needs more and more basis functions (the number of basis functions or hidden neurons, k, has to increase) to achieve the same accuracy of approximation. In other words,

for the MLP NN, it is the number of hidden neurons k and not the number of inputs n that determines the accuracy of the approximation.

The basis functions z_j are very flexible, non-orthogonal, and overlapping. On the other hand, these useful properties may easily lead to non-optimality or redundancy in the NN architecture. It means that some hidden neurons may contribute very little to the approximation and could be removed by pruning without a significant impact on approximation accuracy. Pruning and other similar techniques [*Bishop,* 1995; *Haykin,* 1994] have been developed to optimize the NN architecture and complexity. The flexibility of the MLP NN technique, if not properly controlled, may lead to some unwanted consequences like overfitting (fitting noise in the data), unstable interpolation, and uncertain derivatives. These limitations and the ways to control them will be discussed in this section.

2.4.2 NN training as nonlinear optimization and multi-collinearity of inputs and outputs

NN training, as described in Section 2.3.5, is an iterative procedure that does not involve any matrix inversion. It is robust and 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 of the error or loss function (7). Multi-collinearities in 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 these local minima give almost equally good solutions because the approximation errors for these minima are almost equally small. On the other hand these local minima, which are almost equivalent in terms of the

approximation error, give different solutions in terms of NN weights. These different NNs provide different interpolations and different derivatives. Thus, because of the equalization of errors corresponding to different local minima the approximation error may be not instrumental without using additional criteria, for selecting solutions with good interpolation properties and derivatives.

2.4.3 NN generalization ability: interpolation and extrapolation. NN emulations.

One of the vaguest terms in NN jargon is "generalization" or "generalization ability". This term came from traditional cognitive science applications and means a good performance of the trained NN on new inputs that were not included in the training set. However, it is clear that there are at least two different cases of generalization. First, when new inputs are located inside the domain, *D*, "between" the training data points. Second, when new inputs are located beyond the area covered by the training set, namely close to or outside the boundary of the domain *D*. The first case is an interpolation situation; the second case is an extrapolation one.

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 a small approximation error (7). Moreover, multiple local minima with very close and small approximation errors may still lead to different interpolations. If the NN complexity is not controlled, overfitting may occur that may lead to poor interpolation, e.g., significant oscillations in-between training data points. As mentioned above in Subsection 2.3.3, the representativeness of the training set is a

necessary condition for good interpolation. Three additional measures are instrumental in improving the interpolation ability of the NN approximation:

1. the NN complexity (the number k of hidden neurons) has to be controlled and restricted to a minimum level sufficient for good approximation;
2. independent validation and test data sets should be used in the process of training to control overfitting (the validation set), and after the training to evaluate the interpolation accuracy (the test set); and
3. a redundant training set (additional data points added "in-between" the training data points) is helpful for improving the NN interpolation abilities.

In this paper we will call such an NN approximation – an **emulating NN, NN emulation,** or **NN emulator.** NN emulation provides the functional emulation of the target mapping (1), including a small approximation error (7) for the training set (5) and a smooth and accurate interpolation between training set data points inside the domain D. The correspondence between the emulating NN complexity and target mapping complexity is usually better than of an approximating NN with the same approximation error. The complexity of an emulating NN is usually close to the minimal one; thus, the emulating NN is usually faster. Usually it provides a better interpolation generalization, better resolution of the target mapping at the same approximation accuracy, and smaller uncertainties in the NN Jacobian.

2.4.4 NN Jacobian

The NN Jacobian, a matrix of the first derivatives of NN outputs over inputs, may be useful in many cases. For example, in data assimilation applications a Jacobian is used to create an adjoint (a tangent-linear approximation) of the target mapping. A Jacobian is also

instrumental in a statistical analysis (sensitivity, robustness, and error propagation analyses) and inversion of the target mapping and its NN emulation. An inexpensive computation of the NN Jacobian by the analytical differentiation of (2,3) is one of the advantages of the NN approach. However, the Jacobian is not trained; it is simply calculated through a direct differentiation of an emulating NN. In this case the statistical inference of a Jacobian is 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 very close approximation errors means that there exist multiple solutions for emulating NNs that have close approximation and interpolation errors but different Jacobians.

As mentioned above in Section 2.4.3, if additional care is taken during the training, NN emulations can demonstrate good interpolation properties [*Krasnopolsky and Fox-Rabinovitz*, 2006b]. This implies that, on average, the derivatives of these emulations are sufficiently accurate to provide a satisfactory interpolation. However, for other applications, such the accuracy of a NN Jacobian may be not sufficient. For those applications that require an explicit calculation of the NN Jacobian, several solutions have been offered and investigated: (i) the Jacobian (or the entire adjoint) can be trained as a separate additional 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); (ii) 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*, 2006], (see also Section 5.2.2 of this paper); (iii) the mean Jacobian can be

calculated over the data set and used [*Chevallier and Mahfouf*, 2001]; (iv) regularization techniques like "weight smoothing" [*Aires et al.,* 1999] or the technique based on a principle component decomposition [*Aries et al.,* 2004b] can be used to stabilize the Jacobians; or (v) the Jacobian that is included in the training data set and as actual additional outputs in the NN can be trained, and the error or cost function 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. Actually, *Hornik et al*., [1990] showed that the function of Sobolev's space can be approximated with all their derivatives. 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 developed by other authors [*Cardaliaguet and Euvrard*, 1992; *Lee and Oh*, 1997].

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 training since the complexity of the error function increases. This solution also requires the availability of an extended training set that includes first 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

The existence of multiple solutions is a property of nonlinear models and nonlinear approximations. These models have many nonlinear parameters that could change in the

process of generating solutions. These multiple solutions may be close in terms of a particular criterion used for obtaining the solutions. At the same time these models (NNs) may be different in terms of other criteria that provide complementary information about the target mapping. The availability of multiple solutions may lead to some inconveniences and uncertainties, e.g., the necessity of introducing an additional step to use additional criteria to select a single model. On the other hand, the availability of multiple models (NN emulations) providing complementary information about the target mapping opens the opportunity to use an ensemble approach that allows integration of the complementary information contained in the ensemble members into an ensemble that "knows" more about the target mapping than does any of the ensemble members (particular 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 particular member model can be traced back to as early as the late 1950's and early-mid 1960's [*Selfridge*, 1958; *Nilsson*, 1965]. Since the early 1990's, 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 showed that an ensemble is often more accurate than any single ensemble member. Different ways of combining NN ensemble members into the ensemble have been developed [*Naftaly et al.,* 1997] (see Section 5.2). Previous research also suggests that any mechanism that causes some randomness in the formation of NN members can be used to form a NN ensemble [*Opitz and Maclin*, 1999]. For example,

ensemble members can be created by training different members on different subsets of the training set [*Opitz and Maclin*, 1999], by training different members on different sub-domains 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].

In the context of our application, an approximation of a complex mapping (1), the members of the ensemble are separately trained approximating NNs which provide different accuracies of approximation for the target mapping and different interpolation accuracies. We can expect that the ensemble average will provide a better approximation and interpolation than the individual members (see Section 5.2.1). *Krasnopolsky* [2006] also applied the NN ensemble technique to reduce the uncertainty of the NN Jacobian (see Section 5.2.2). Most previous work with NN ensembles have been done in the context of solving the classification [*Hansen and Salamon*, 1990; *Sharkey*, 1996; *Opitz and Maclin*, 1999] or the prediction of time series problems [*Naftaly* et al., 1997, *Hsieh*, 2001].

2.4.6 NN uncertainties estimates

The NN technique is a nonlinear statistical approach. As with any statistical approach, the NN technique is expected to provide not only the minimization of an error or loss function but also an estimate of the uncertainties in NN weights and outputs. Because of the nonlinear nature of the NN technique, an estimation of the NN uncertainties is a more complicated problem than that in the case of linear statistical tools. However, during the last decade progress has been made in this field in the cases of both the MLP NN with a single output [*MacKay*, 1992; *Bishop*, 1995; *Neal*, 1996; *Nabney*, 2002] and with multiple outputs

[*Aires et al.*, 2004a]. Various Bayesian methods have been used in these studies for estimating the uncertainties of NN parameters (weights).

2.5 Discussion

In this section, we discussed general properties of the multidimensional complex mappings (1) and MLP NN (2,3) and also attempted to demonstrate relationships between their properties. Both fields are young and are fast growing in terms of a developed theory and practical applications. In this discussion, we tried to emphasize that a transition from linear statistical tools to nonlinear ones (like NNs) requires some adjustment in our methodological framework, which is sometimes not flexible enough to accommodate sophisticated nonlinear approaches. Many of the advantages of nonlinear statistical approaches may turn into limitations under some special conditions. Some of the limitations of nonlinear tools may be turned into advantages by using more flexible approaches, combining different statistical approaches (e.g., NNs and the ensemble approach), and using additional information.

3 Atmospheric and Oceanic Remote Sensing Applications

Estimating high quality geophysical parameters (information about the physical, chemical, and biological properties of the oceans, atmosphere, and land surface) from remote measurements (satellite, aircraft, etc.) is a very important problem in fields such as meteorology, oceanography, climatology and environmental modeling and prediction. Direct measurements of many parameters of interest, like vegetation moisture, phytoplankton concentrations in the ocean, and aerosol concentrations in the atmosphere are, in general, not available for the entire globe 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. Often such measurements can be estimated indirectly from the influence of these geophysical parameters on the electromagnetic radiation measured by a remote sensor. Remote measurements allow us to obtain spatially dense measurements all over the globe at and above the level of the ground and ocean surface.

The remote measurements themselves are usually very accurate. The quality of geophysical parameters derived from these measurements varies significantly depending on the strength and uniqueness of the signal from the geophysical parameter and the mathematical methods applied to extract the parameter, i.e., to solve forward and/or inverse remote sensing (RS) problems. The NN technique is a useful mathematical tool for solving the forward and inverse problems in RS accurately. The number of NN RS applications has been increasing steadily over the last decade. Examples of such applications follow.

The NN technique was 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 the leaf area index from Moderate Resolution Imaging Spectrometer data. In other studies, NNs were 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 ocean color sensor [*Brajard et al*., 2006]; to develop an inversion algorithm for radar scattering from vegetation canopies [*Pierce et al.,* 1994]; to estimate atmospheric humidity profiles [*Cabrera-Mercader and Staelin*, 1995], atmospheric temperature, moisture, and ozone profiles [*Aires et al*., 2002] and atmospheric ozone profiles [*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 surface wind speed. *Davis et al*., [1995] applied NNs to invert a forward model to estimate soil moisture, surface air temperature, and vegetation moisture from Scanning Multichannel Microwave Radiometer data. Using a NN technique, a fast SSM/I forward model [*Krasnopolsky*, 1997] and SSM/I multi-parameter retrieval algorithm [*Krasnopolsky, Breaker, and Gemmill*, 1999, 2000; *Meng et al.*, 2006] have been derived from empirical data (buoy SSM/I collocations). *Abdelgadir et al*., [1998] applied NNs to the forward and inverse modeling of canopy directional reflectance. *Schiller and Doerffer* [1999] used a NN technique for inverting a radiative transfer forward model to estimate the concentration of phytoplankton pigment from Medium Resolution Imaging Spectrometer (MERIS) data.

3.1 Deriving Geophysical Parameters from Satellite Measurements: Standard Retrievals and Variational Retrievals Obtained Through Direct Assimilation

Satellite sensors generate measurements like radiances, backscatter coefficients, brightness temperatures, etc. The applications utilize usually geophysical parameters such as pressure, temperature, wind speed and direction, water vapor concentration, etc. derived from satellite data. There exists an entire spectrum of different approaches in extracting geophysical information from the satellite measurements. At one end of this spectrum 'satellite only' approaches are located; we will call them standard or traditional retrievals. They use measurements performed by one particular sensor only, sometimes from different channels (frequencies, polarizations, etc.) of the same sensor to estimate geophysical parameters. Variational retrieval techniques or direct assimilation techniques are located at the other end of the spectrum. They use an entire data assimilation system (DAS), including a numerical

weather prediction (NWP) model and analysis [*Prigent et al.*, 1997], which in turn includes all kind of meteorological measurements (buoys, radiosondes, ships, aircrafts, etc.) as well as data from numerous satellite sensors. Many approaches have been developed which belong to the intermediate part 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, i.e., these approaches use some type of data fusion to regularize the solution of the inverse problem.

3.1.1 Standard or conventional retrievals

Conventional methods for using satellite data (standard 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, sea surface temperature (SST), etc.) to a satellite measurement S (e.g., brightness temperatures, radiances, reflection coefficients, etc.)

$$G = f(S) \tag{9}$$

where both G and S may be vectors. The TF f, (also called a retrieval algorithm) usually cannot be derived directly from first principles because the relationship (9) does not correspond to a cause and effect principle and multiple values of G can sometimes correspond to a single S. Forward models,

$$S = F(G) \tag{10}$$

where F is a forward model (FM), which relates a vector G to a vector S, can usually be derived from first principles and physical considerations (e.g., a radiative transfer theory) in accordance with cause and effect principles because geophysical parameters affect the

satellite measurements (but not vice versa). Thus, the forward problem (10) is a well-posed problem in contrast to the inverse problem (9) which is often an ill-posed one [*Parker*, 1994]; although, from a mathematical point of view, both FM (10) and TF (9) are continuous (or almost continuous) mappings between the two vectors S and G. Even in the cases where the mapping (9) is not unique, this multi-valued mapping may be considered as a collection of single-valued continuous mappings. In order to derive the TF (9), the FM (10) 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 functional [*Stoffelen and Anderson*, 1997]

$$\|\Delta S\| = \|S^0 - F(G)\| \tag{11}$$

where S^0 is an actual vector of satellite measurements. Since the FM F is usually a complicated nonlinear function, this approach leads to a full-scale nonlinear optimization with all its numerical problems, like a slow convergence, multiple solutions, etc. 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 (11) can be applied if there is a good approximation for the solution of the inverse problem, 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 $\Delta S(G) = 0$. Expanding $F(G)$ in a Taylor series and keeping only those terms which 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],

$$\sum_{i=1}^{n} \frac{\partial F(G)}{\partial G_i}\Big|_{G=G^0} \Delta G_i = S^0 - F(G^0) \tag{12}$$

where n is the dimension of vector G. After ΔG is calculated, the next iteration of (12) with $G^0 = G^0 + \Delta G$ is performed. The process is expected to converge quickly to the vector of retrievals G. Again, in this case the TF (9) f is not determined explicitly but is only determined implicitly for the vector S^0 by the solution of (12). This type of retrieval can be called a "local" or "localized" linear inversion. These techniques (11,12) are usually called physically based retrievals. It is important to emphasize that the physically based algorithms (11,12) are by definition multi-parameter algorithms since they retrieve several geophysical parameters simultaneously (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 usually chosen (usually some kind of a regression), which contains a vector of empirical (or model) parameters $a = \{a_1, a_2, \ldots\}$,

$$G_k = f_{mod}(S,a) \tag{13}$$

where these parameters are determined from an empirical (or simulated) matchup data set $\{G_k, S\}$ collocated in space and time and use, for example, statistical techniques such as the method of least-squares. This type of retrieval can be called a "global" inversion as it is not restricted to a given vector of satellite measurements. The subscript k in G_k stresses the fact that the majority of empirical retrieval algorithms are single-parameter algorithms. For example, for SSM/I there exist algorithms which retrieve only wind speed [*Goodberlet et al.*, 1989], water vapor [*Alishouse et al.*, 1990; *Petty*, 1993], or cloud liquid water [*Weng and Grody*, 1994]. *Krasnopolsky et al.*, [1999, 2000a] showed that single-parameter algorithms have additional (compared to multi-parameter retrievals) systematic (bias) and random (unaccounted variance) errors in a single retrieved parameter G_k.

The obvious way to improve single-parameter retrievals (13) is to include other parameters in the retrieval process using an empirical multi-parameter approach, which as in the physically based multi-parameter approach (11,12), inverts the data in the complete space of the geophysical parameters [*Krasnopolsky et al*., 1999, 2000a]. Thus, the complete vector of the related geophysical parameters is retrieved simultaneously from a given vector of satellite measurements S,

$$G = f_{mod}(S) \tag{14}$$

where $G = \{G_i\}$ is now a vector containing the primary, physically-related geophysical parameters, which contribute to the observed satellite measurements S. These retrievals do not contain the additional systematic and random errors just described. Because Eqs. (9), (10), (13), and (14) represent continuous mappings, the NN technique is well suited for emulating the FM, TF and empirical TF, f_{mod}.

Geophysical parameters derived using standard retrievals can be used for many applications, such as the NWP DASs. In this case, a contribution to the variational analysis cost function χ_G from a particular retrieval, G^0, is:

$$\chi_G = \frac{1}{2}(G - G^O)^T (O + E)^{-1} (G - G^O) \tag{15}$$

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 a vector of the geophysical parameters being analyzed, O is the expected error covariance of the observations, and E is the expected error covariance of the retrieval algorithm.

3.1.2 Variational retrievals through the direct assimilation of satellite measurements.

Because standard 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 retrievals (e.g., an amplification of errors, ambiguities, etc.). As a result, high-quality sensor measurements might be converted into lower-quality geophysical parameters. This type of error can be avoided or reduced by using a variational retrieval technique (or an inversion) through direct assimilation of satellite measurements [*Lorenc*, 1986; *Parrish and Derber*, 1992; *Phalippou*, 1996; *Prigent* et al., 1997; *Derber and Wu*, 1998; *McNally* et al., 2000].

Variational retrievals or direct assimilation of satellite data offer an alternative to deriving geophysical parameters from the satellite measurements. They use the entire data assimilation system for the inversion (as a retrieval algorithm). In this case, a contribution to the analysis cost function χ_S from a particular sensor measurement, S^0, is:

$$\chi_S = \frac{1}{2}(S - S^O)^T (O + E)^{-1} (S - S^O) \tag{16}$$

where $S = F(G)$, and F is a FM (10) which relates an analysis state vector G (or 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 forward model. The forward problem (10) is a well-posed problem in contrast to the inverse problem (9). However, a background term has to be added to (16) to prevent the data assimilation problem from being ill-posed [*Parrish and Derber*, 1992].

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., a radiative transfer theory). Dependencies of satellite measurements on geophysical parameters, which FMs describe, are complicated and nonlinear. These dependencies may exhibit different types of nonlinear behavior. FMs are usually exploited in physically based retrieval algorithms where they are numerically inverted to retrieve geophysical parameters and in data assimilation systems 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 are calculated many times for each satellite measurement. Thus, the retrieval process becomes very time consuming, sometimes 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 Section 2.2.2) is usually not as high as their physical complexity, NNs can provide fast and accurate emulations of FMs. Moreover, a NN can also provide an entire Jacobian matrix with only a small additional computational effort. This is one NN application where the NN Jacobian should be carefully tested and controlled (see Section 2.4.4.).

To develop a NN emulation for the 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 physically based 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 several different ways for retrieval algorithms. In physically based retrieval algorithms a fast NN, emulating the complex and slow physically based FM and its Jacobian, can be used to speed up the local inversion process (12). NNs can be used in many cases for a global inversion to explicitly invert a FM. In such cases, after an inversion the NN provides an explicit retrieval algorithm (or TF), which is a solution of the inverse problem and can be used for retrievals. To train a 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.

A serious additional problem related to retrieval algorithms is the problem of 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. For example, in their pioneering work on using NNs for the simultaneous retrieval of temperature, water vapor, and ozone atmospheric profiles [*Aires., et al.*, 2002; *Mueller, et al.*, 2003] from satellite measurements, the authors made good use of this NN flexibility by introducing the first guess from the atmospheric model or DAS as additional regularizing inputs in their NN based retrieval algorithms.

3.4 Controlling the NN Generalization

Well-constructed NNs (NN emulations) have good interpolation properties; however, they may produce unpredictable outputs when forced to extrapolate (see Section 2.4.3). The NN training data (produced by a FM or constructed from empirical data collections) cover a

certain manifold D_T (a sub-domain $D_T \subseteq D$) in the full domain D. Real data to be fed into the NN f_{NN}, which emulates a TF (9), may not always lie in D_T. There are many sources for such deviations of real data from the low dimensional manifold D_T of training data. In order to recognize NN inputs not foreseen in the NN training phase and, thus, out of the scope of the inversion algorithm, a validity check [*Schiller and Krasnopolsky*, 2001] can be used. This check may serve as the basis for a quality control (QC) procedure. Some kind of QC procedure is usually applied to the satellite retrievals in DAS.

Let the model $S = F(G)$ have an inverse $G = f(S)$, then, by definition $S = F(f(S))$. Further, let f_{NN} be the NN emulating the inverse model in the domain D_T. The result of $G_0 = f_{NN}(S_0)$ for $S_0 \notin D_T$ may be arbitrary and, in general, $F(f_{NN}(S_0))$ will not be equal to S_0. The validity of $S = F(f_{NN}(S))$ is a necessary condition for $S \in D$. Now, if in the application stage of the NN f_{NN}, S is not in the domain S_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. For operational applications, it is necessary to report such events to the next higher evaluation level. In order to perform the validity test, the FM must be applied after each inversion. This requires a fast but accurate FM. Such a FM can be achieved by developing a NN that accurately emulates the original FM, $S = F_{NN}(G)$. Thus, the validity check algorithm consists of a combination of inverse and forward NNs that, in addition to the inversion, computes a quality measure for the inversion:

$$\delta = \| S - F_{NN}(f_{NN}(S)) \| \tag{17}$$

The solution to the problem of a scope check is obtained by estimating δ (17) where S is the result of the satellite measurement. This procedure (*i*) allows the detection of situations where the forward model or/and transfer function is inappropriate, (*ii*) does an "in scope"

check for the retrieved parameters even if the domain has a complicated geometry, and (*iii*) can be adapted to all cases where a NN is used to emulate the inverse of an existing forward model. For examples of applications of the scope check for quality control of retrievals and for development of intelligent NN based retrieval systems see [*Krasnopolsky and Schiller*, 2003].

3.5 Neural Network Emulations for SSM/I Data

In previous sections, we discussed the theoretical possibilities and premises for using NNs for modeling TFs and FMs. In this section, we illustrate these theoretical considerations using 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. Many different retrieval algorithms and several forward models have been developed for this sensor and several different databases are available for algorithm development and validation. Various different techniques have been applied to the algorithm development. Therefore, we can present an extensive comparison of different methods and approaches for this instrument. A raw buoy-SSM/I matchup database created by the Navy was used for the NN algorithm development, validation, and comparison. This database is quite representative, with the exception of high latitude and high wind speed events. In order to improve this situation the data sets were enriched by adding matchup databases collected by the high latitude European ocean weather ships Mike and Lima to the Navy database. Various filters have been applied to remove errors and noisy data (for a detailed discussion see *Krasnopolsky* [1997], and *Krasnopolsky et al.,* [1999]). The matchup databases for the F11 SSM/I have been used for training (about 3,500 matchups) and testing (about 3,500 matchups) our forward model and retrieval algorithm. Then, more than 6,000 matchups for the F10

instrument were used for the validation. The NN emulations were trained using all matchups that correspond 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 cannot penetrate the clouds were removed.

3.5.1 NN emulation of the empirical FM for SSM/I

The empirical SSM/I FM represents the relationship between the vector of geophysical parameters G and vector of satellite brightness temperatures (BTs) S, where $S = \{T19V, T19H, T22V, T37V, T37H\}$ ($TXXY$ means XX frequency in GHz and Y polarization), $G = \{W, V, L, T_s$ (or SST)$\}$. Four geophysical parameters are included in G (surface wind speed W, columnar water vapor V, columnar liquid water L, and SST). These are the main parameters influencing BTs measured by satellite sensors, which were used as inputs in the physically based FMs of *Petty and Katsaros* [1992, 1994] (referenced to below as PK) and *Wentz* [1997] (see Table 2). The NN emulation [*Krasnopolsky*, 1997], which implements this SSM/I FM has 4 inputs $\{W, V, L, SST\}$, one hidden layer with 12 neurons, and 5 nonlinear BT outputs $\{T19V, T19H, T22V, T37V, T37H\}$.

After NN FM was trained, it was validated and compared with PB forward models by *Petty and Katsaros* (PK) and *Wentz* [1997]. The RMS errors for NN FM are systematically better than those for the more sophisticated physically-based PK and Wentz FMs for all weather conditions and all channels considered (see Table 2).

Table 2

The NN FM is not as general as a radiative transfer model; it was developed to be applied to the data assimilation system for variational retrieval and direct assimilation of SSM/I BTs at particular frequencies from a particular instrument. However, for this particular application (direct assimilation) and particular instrument it has a significant

advantage (it is significantly simpler and faster), especially in an operational environment. The NN FM simultaneously calculates the BTs and Jacobian matrix, which is required in the process of direct assimilation [*Parrish and Derber*, 1992; *Phalippou*, 1996]. This is one of the applications where the accuracy of the NN Jacobian is essential. *Krasnopolsky* [1997] has demonstrated that for this particular application the NN Jacobian is sufficiently smooth and accurate. In Section 5.2 a generic NN ensemble technique [*Krasnopolsky*, 2006] is discussed that improves the stability and reduces uncertainties of the NN emulation Jacobian if desired.

3.5.2 NN empirical SSM/I retrieval algorithms

The SSM/I wind speed retrieval problem is a perfect example illustrating the general discussion presented in Sections 3.1 and 3.3. The problems encountered in the case of SSM/I wind speed retrievals are very 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 advantages of the NN approach (for detailed discussion and comparison of algorithms see [*Krasnopolsky et al.,* 1999] and [*Krasnopolsky and Schiller*, 2003]).

Goodberlet et al., [1989] developed the first global SSM/I wind speed retrieval algorithm. This algorithm is a single-parameter algorithm (it retrieves only wind speed) and is linear with respect to BTs (a linear multiple regression is used). Statistics for this algorithm are shown in Table 3 under the abbreviation GSW. *Goodberlet and Swift* [1992] (GS in Table 3) tried to improve the GSW algorithm performance under cloudy conditions,

using nonlinear regression with a rational type of nonlinearity. Another nonlinear (with respect to BTs) version of the GSW single-parameter algorithm (called the GSWP algorithm here) introduced by *Petty* [1993] is based on a generalized linear regression.

Single-parameter NN algorithms have been introduced as an alternative to nonlinear and generalized linear regressions because the NN can model the nonlinear behavior of a TF better than these regressions. *Stogryn et al*., [1994] developed the first NN SSM/I wind speed algorithm, which consists of two NNs, each with the surface wind speed as a single output. One NN performs retrievals under clear and the other under cloudy conditions. *Krasnopolsky et al*., [1995] showed that a single NN 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]. Application of a single NN emulation led to a significant improvement in wind speed retrieval accuracy under clear conditions. 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 (higher levels of atmospheric moisture). However, because this algorithm used for the training the data set that did not contain high wind speed events, it cannot generate acceptable wind speeds at ranges higher then 18 - 19 m/s.

Table 3

The next generation NN algorithm - a multi-parameter NN algorithm developed at NCEP (NN1 in Table 3) by *Krasnopolsky et al.,* [1996, 1999] solved the high wind speed problem through three main advances. First, a new buoy/SSM/I matchup database described above that contained wind speeds up to 26 m/s was used in the development of this algorithm. Second, the NN training method was improved by enhancing the learning 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.

Meng, et al., [2006] (NN2 in Table 3) use the NN multi-parameter retrieval approach developed by *Krasnopolsky et al.*, [1996, 1999] to design another NN multi-parameter retrieval algorithm for SSM/I. They use all 7 SSM/I BTs as inputs. Their output vector also has four components $G = \{W, T_a, H, SST\}$ where surface wind speed (W), surface air temperature (T_a), humidity (H), and SST are retrieved simultaneously. In this case, the training database was limited by maximum wind speeds of about 20 m/sec. Moreover, there were only a few higher speed events with $W > 15 – 17$ m/sec.

Table 3 shows a comparison of the performance of all the aforementioned empirical algorithms in terms of the accuracy of the surface wind speed retievals. It also shows statistics for a physically based algorithm developed by *Wentz* [1997], which is based on a linearized numerical inversion (12) of a physically based FM. The statistics presented in Table 3 were calculated using independent buoy-SSM/I matchups. Table 3 shows that the NN algorithms outperform all other algorithms. All algorithms except the NN1 algorithms show a tendency to overestimate high wind speeds (see [*Krasnopolsky et al.*, 1999, 2000a] for discussion). NN1 shows the best total performance, in terms of bias, RMSE, and high wind speed performance.

Krasnopolsky et al., [1999, 2000a] have shown that the errors of multi-parameter NN algorithms have a weaker dependence on the related atmospheric and surface parameters than the errors of the single-parameter algorithms considered. The retrieved *SST* in this case is not accurate (the RMS error is about 4°C, see *Krasnopolsky et al*., [1996]); however, including *SST* in the vector of retrieved parameters decreases the error in other retrievals correlated with the *SST*. For the multi-parameter NN algorithm NN2 [*Meng et al*., 2006], the choice of the additional outputs surface air temperature (T_a) and humidity (H), that are closely and physically related and correlated with *SST*, makes the accuracy of the retrieved SST signal higher (the bias is about 0.1°C and RMS error 1.54°C). In accordance with the classical, "linear" remote sensing paradigm, the SSM/I instrument does not have the frequency required to sense SST. However, due to the nonlinear nature of the NN emulation and the proper choice of output parameters the multi-parameter NN algorithm is able to use weak nonlinear dependencies between NN inputs and outputs and between NN outputs to retrieve SST with a good accuracy.

3.6 Discussion

In this section we discussed a broad class of NN applications dealing with the solution of the RS forward and inverse problems. Theoretical considerations presented here were illustrated using several real-life applications that exemplify a NN based intelligent integral approach where the entire retrieval system, including the quality control block [*Krasnopolsky and Schiller*, 2003], is designed from a combination of several specialized NNs. This approach offers significant advantages in real life operational applications. This intelligent retrieval system can produce not only accurate retrievals, but it also performs an

analysis and quality control of the retrievals and environmental conditions, rejecting any poor retrieval that occurs.

The NN applications presented in this section illustrate the strengths and limits of the NN technique for inferring geophysical parameters from remote sensing measurements. NNs successfully compete with other statistical methods and usually perform better because they are able to emulate the functional relationship between inputs and the outputs in an optimal way. NNs can successfully compete with even physically based approaches because, in many cases, explicit knowledge of very complicated physical processes in the environment is limited and a NN based empirical approach is more appropriate**.** It can take into account more physics implicitly than a physically based approach would include explicitly. However, the success of the NN approach strongly depends on the adequacy of the data set used for the NN training (see Section 2.3.3). The data availability, precision, quality, representativeness, and amount are crucial for success in this type of NN application.

4 Applications of NNs to Developing Hybrid Atmospheric and Oceanic Numerical Models

The past several decades revealed a well pronounced trend in geosciences. 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 nonlinear 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 these interdisciplinary

complex geophysical/environmental numerical models has increased tremendously during the last few decades, due to improvements in their quality via developments in numerical modeling and computing capabilities.

Traditional complex environmental numerical models (ENM) are deterministic models based on "first principle" equations. For example, general circulation models (GCM) a.k.a. global climate models are numerical atmospheric and oceanic models for climate simulation and weather prediction that are based on solving time-dependent 3-D geophysical fluid dynamics equations on a sphere. The governing equations of these models can be written symbolically as,

$$\frac{\partial \psi}{\partial t} + D(\psi, x) = P(\psi, x) \tag{18}$$

where ψ are 3-D prognostic or dependent variable or set of variables (e.g., temperature, wind, pressure, moisture); x is a 3-D independent variable (e.g., latitude, longitude, and pressure or height); D is the model dynamics (the set of 3-D partial differential equations of motion, thermodynamics, etc., approximated with a spectral or grid-point numerical scheme); and P is the model physics and chemistry (e.g., the long- and short-wave atmospheric radiation, turbulence, convection and large scale precipitation processes, clouds, interactions with land and ocean processes, etc., and the constituency transport, chemical reactions, etc., respectively). These environmental models 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).

Another example of a complex ENM is an ocean wind wave model developed for simulation and forecast purposes [*Tolman,* 2002]. It is based on a form of the spectral energy or action balance equation

$$\frac{DF}{Dt} = S_{in} + S_{nl} + S_{ds} + S_{sw} \tag{19}$$

where F is the 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.

It is important to emphasize that the subsystems of a complex climate (or weather) system, such as physical, chemical, and other processes, are so complicated that it is currently possible to include them into GCMs only as 1-D (in the vertical direction) simplified or parameterized versions (a.k.a. parameterizations). These parameterizations constitute the right hand side forcing for the dynamics equations (18,19). Some of these parameterizations are still the most time consuming components of ENMs (see examples in the next subsection). Thus the parameterizations have a very complicated internal structure, are formulated using relevant first principles and observational data, and are usually based on solving deterministic equations (like radiation equations) and some secondary empirical components based on traditional statistical techniques like regression. Accordingly, for widely used state-of-the-art GCMs all major model components (subsystems) are predominantly deterministic; namely, not only model dynamics but the model physics and chemistry are also based on solving deterministic first principle physical or chemical equations.

In this section, we discuss the concepts of hybrid parameterization (HP) and hybrid environmental models (HEM). HEMs are based on a synergetic combination of

deterministic numerical modeling (first principle equations) with NN emulations of some model physics components. We discuss the conceptual and practical possibilities of developing a hybrid GCM (HGCM) and HEM; namely, the possibility of combining accurate and fast NN emulations of model physics components with the deterministic model dynamics of a GCM or ENM, which are the types of complex environmental models used for modern atmospheric and ocean climate modeling and weather prediction.

4.1 Concepts of a Hybrid Model Component and a Hybrid Model

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 for model physics, keeping in mind that the approach is applicable to other model components (chemical, hydrological and other processes) as well. Parameterizations of model physics are approximate schemes, adjusted to model resolution and computer resources, and based on simplified physical process equations and empirical data and relationships. The parameterizations are still so time-consuming, even for the most powerful modern supercomputers, that some of the parameterizations have to be calculated less frequently than the model dynamics. Also, different physical parameterizations are calculated at different frequencies inversely proportional to their computational complexity. This may negatively affect the accuracy of climate and other environmental simulations and predictions.

For example, in the case of a complex GCM, calculation of a physics package (including the atmospheric and land physics) at typical (a few degrees) resolution as in the National Center for Atmospheric Research (NCAR) Community Atmospheric Model (CAM) takes about 70% of the total model computations. This is despite the fact that while

the model dynamics is calculated every 20 minutes, some computationally expensive parts of the model physics (e.g., short wave radiation) are calculated every hour. The most time consuming calculations of the model atmospheric physics, full long wave radiation including calculation of optical properties, are done only once every 12 hours while the heating rates and radiative fluxes are calculated every hour. More frequent model physics calculations, 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 (19), the calculation of the source term, S_{nl} requires roughly 10^3 to 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 for the remainder of the wave model.

This situation is a generic and important motivation in looking for alternative, faster, and most importantly very 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 (e. g., *Krasnopolsky,* [1997] and *Chevallier at al*., [1998]) and for emulations of model physics parameterizations in ocean and atmospheric numerical models [*Krasnopolsky et al.*, 2000b, 2002, 2005]. In these works, the calculation of model physics components has been accelerated by 10 to 10^5 times as compared to the time needed for calculating the corresponding original parameterizations of the model physics.

Approaches formulated by *Chevallier at al*., [1998, 2000] and *Krasnopolsky et al.,* [2000b, 2002, 2005] represent two different ways of introducing a hybridization of first principle and NN components in the physics parameterizations as well as in complex ENMs. 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 (ENM). These two approaches lead to the concepts of a hybrid parameterization (HP) [*Chevallier at al*., 1998, 2000] and a hybrid environmental model (HEM) or hybrid GCM (HGCM) [*Krasnopolsky et al.*, 2000b, 2002, 2005a; *Krasnopolsky and Fox-Rabinovitz*, 2006a,b]. These two concepts have been debated by *Chevallier* [2005] and *Krasnopolsky et al.* [2005c] and are discussed in the following subsections. Another type of hybrid models – 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 Hybrid parameterizations of physics

Chevallier et al., [1998, 2000] considered a component of the complex GCM (the ECMWF global atmospheric model) – the LWR parameterization. Putting it in terms of the system levels, this single parameterization is considered to be the system and its constituents, with the blocks calculating fluxes, the blocks calculating cloudiness, etc., as the 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 (1),

$$Y = M(X) \tag{20}$$

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, $Y = (Q,f)$. Here Q is a profile of cooling rates $Q = (C_r^1, C_r^2, \ldots, C_r^L)$, where C_r^j is the cooling rate at the j-th vertical model level, and f is a vector of auxiliary fluxes computed by the LWR parameterization. Because of the presence of the cloud variable C, the mapping (20) 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 to separate cloud variables C. In this parameterization, level fluxes are calculated as,

$$F(S,T,V,C) = \sum_i \alpha_i(C) F_i(S,T,V) \tag{21}$$

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 hybrid parameterization "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 at each level (21) is a linear combination of approximating NNs and cloud physics coefficients $\alpha_i(C)$. As the result, the "NeuroFlux" hybrid LWR parameterization developed by *Chevallier et al.,* [1998, 2000] is a battery of about 40 NNs. To calculate "NeuroFlux" outputs, namely the cooling rates C_rs, linear combinations of the individual approximating NNs F (eq. 21) are differentiated at each vertical level,

$$C_r(P) = \frac{\partial F(P)}{\partial P} \quad , \tag{22}$$

where P is atmospheric pressure.

The "NeuroFlux" has a very good accuracy; its bias is about 0.05 K/day and RMS error is about 0.1 K/day compared to the LWR parameterization by *Washington and Williamson* [1977]. It is eight times faster than the parameterization by *Washington and Williamson* [1977]. This HP approach has already led to the successful operational implementation of "NeuroFlux" in the ECMWF 4-DVar data assimilation system.

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 the possibility of separating the dependence on the cloudiness (see eq. 21). Many other LWR parameterizations, like the NCAR CAM LWR parameterization [*Collins,* 2001; *Collins et al.,* 2002] or the LWR parameterization developed by *Chou et al.,* [2001], do not allow for such separation of variables. Thus, for these LWR parameterizations as well as the short wave radiation (SWR) and the moisture model physics block parameterizations, the HP approach developed by *Chevallier et al.,* [1998, 2000] cannot be applied directly; it should be significantly modified or redesigned for each particular new parameterization.

4.3 Hybrid Numerical Models: Accurate and Fast NN Emulations for Parameterizations of Model Physics

A new concept of a complex hybrid environmental model (HEM) has been formulated and developed by *Krasnopolsky et al.,* [2000b, 2002, 2005a] and by *Krasnopolsky and Fox-Rabinovitz* [2006a,b]. The hybrid modeling approach considers the whole GCM or ENM as a system. Dynamics and parameterizations of physics, chemistry, etc., are considered to be the components of the system. Hybridization in this case is introduced at the level of components inside the system (ENM). For example, the entire LWR (or SWR) parameterization is emulated by a single NN as a single/elementary object or block. The NN emulation approach is based on the general fact that any parameterization of model physics can be considered as a continuous or almost continuous mapping (1, 20) (see Sections 2 and Subsection 4.2).

Here we use the NCAR CAM (see *J. Climate* [1998] for the description of the model), a widely recognized state-of-the-art GCM used by a large modeling community for climate predictions, and the state-of-the-art NCEP wind wave model [*Tolman*, 2002] as examples of a complex GCM and ENM. After applying the hybridization approach to the first principle based components of these models by developing NN *emulations* of model physics parameterizations, these models become the examples of an HGCM and HEM, correspondingly.

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

1. Problem analysis or analysis of the model component (target mapping (1); i.e., the original parameterization) to be approximated to determine the optimal structure and configuration of the NN emulations - 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 (2,3) (see Sections 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 using data for NN training that are simulated by running an original GCM, i.e., a GCM with the original parameterization. When creating a representative data set, the original GCM must be run long enough to produce all possible atmospheric model simulated states, phenomena, etc. Here, due to the use of simulated data, it is not a problem to generate the sufficiently representative (and even redundant) data sets required to create high quality NN emulations (see Section 2.4.3). 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, poor representation of extreme events, etc.).
3. Training the NN. Several different versions of NNs with different architectures, 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 a sufficient emulation accuracy to create the high quality NN emulations required (see Section 2.4.3).

Testing the HGCM that uses the trained NN emulation consists of two major steps. The *first step* is testing the accuracy of the NN approximation against the original parameterization using the 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 complicated 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 Section 4.4). The *second* test *step* consists of a comprehensive comparison and analysis of parallel HGCM and GCM runs. For the parallel model simulations all relevant model prognostic (i.e., time-dependent model variables) and diagnostic fields 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 a HGCM with NN emulation (see Section 4.4). This test step involving model simulations is crucially important. 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.

4.4 Atmospheric Applications: NN Emulation Components and HGCM

4.4.1 Models and Statistical Metrics

The NCAR CAM and NASA NSIPP (Natural Seasonal-to-Interannual Predictability Program) GCM are used in this section as examples of GCMs. The NCAR CAM is a spectral model that has 42 spectral components (or approximately 3° – 3.5° horizontal resolution) and 26 vertical levels. The NSIPP model is a grid point GCM that has 2° × 2.5° latitude × longitude horizontal resolution and 40 vertical levels. NN emulations were developed for the two most time consuming components of model physics, the long wave radiation (LWR) and short wave radiation (SWR). The NCAR and NSIPP models have different LWR and SWR parameterizations. The complete description of the NCAR CAM atmospheric LWR is presented by *Collins* [2001] and *Collins et al.,* [2002], and the NSIPP LWR by *Chou et al.,* [2001]. The full model radiation (or total LWR and SWR) calculations take ~70% of the total model physics calculations.

The NN emulations developed were tested against the original NCAR CAM LWR and SWR parameterizations. The following statistics and statistical cross-sections were calculated 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:

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

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{L} [Y(i,j) - Y_{NN}(i,j)]^2}{N \times L}} \tag{23}$$

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

These two error characteristics (Eqs. (23)) describe the accuracy of the NN emulation integrated over the entire 4-D (latitude, longitude, height, and time) data set. Using a minor modification of Eqs. (23), the bias and RMSE for the mth vertical level of the model can be calculated:

$$B_m = \frac{1}{N}\sum_{i=1}^{N}[Y(i,m) - Y_{NN}(i,m)]$$
$$RMSE_m = \sqrt{\frac{\sum_{i=1}^{N}[Y(i,m) - Y_{NN}(i,m)]^2}{N}} \tag{24}$$

The root mean square error can also be calculated for each ith profile:

$$prmse(i) = \sqrt{\frac{1}{L}\sum_{j=1}^{L}[Y(i,j) - Y_{NN}(i,j)]^2} \tag{25}$$

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 standard deviation σ_{PRMSE} which are location independent:

$$PRMSE = \frac{1}{N}\sum_{i=1}^{N} prmse(i)$$
$$\sigma_{PRMSE} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}[prmse(i) - PRMSE]^2} \tag{26}$$

The statistics (26) and (23) 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 calculated as:

$$rmsep(j) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[Y(i,j) - Y_{NN}(i,j)]^2} \tag{27}$$

4.4.2 Emulating the Model Radiation

The function of the LWR parameterization in atmospheric GCMs is to calculate the heating fluxes and rates produced by LWR processes. As was already mentioned, the entire LWR parameterization can be represented as an almost continuous mapping (Eq. 20). Here a very general and schematic outline of the internal structure of this parameterization is given in order to illustrate the complexity that makes it a computational "bottleneck" in the NCAR CAM physics. This information about the internal structure of the LWR parameterization was not used when creating the LWR NN emulation.

The method for calculating LWR in the NCAR CAM is based on LW radiative transfer equations in an absorptivity/emissivity formulation (see *Collins* [2001] and references there),

$$\begin{aligned} F^{\downarrow}(p) &= B(p_t)\cdot\varepsilon(p_t,p) + \int_{p_t}^{p}\alpha(p,p')\cdot dB(p') \\ F^{\uparrow}(p) &= B(p_s) - \int_{p}^{p_s}\alpha(p,p')\cdot dB(p') \end{aligned} \tag{28}$$

where $F^{\uparrow}(p)$ and $F^{\downarrow}(p)$ are the upward and the 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. To solve the integral equations (28), the absorptivity and emissivity have to be calculated by solving the following integro-differential equations,

$$a(p,p') = \frac{\int_0^\infty \{dB_\nu(p')/dT(p')\} \cdot [1-\tau_\nu(p,p')] \cdot d\nu}{dB(p)/dT(p)}$$
$$\varepsilon(p_t,p) = \frac{\int_0^\infty B_\nu(p_t) \cdot [1-\tau_\nu(p_t,p)] \cdot d\nu}{B(p_t)} \tag{29}$$

where the integration is over wave number ν, and $B(p_t)$ is the Planck function. To solve Eqs. (29) for the absorptivity and emissivity, additional calculations have to be performed and the atmospheric transmission τ_ν has to be calculated. This calculation involves a time consuming integration over the entire spectral range of gas absorption. Eqs. (28,29) illustrate the complexity of the LWR internal structure and explain the poor computational performance of the original NCAR CAM LWR parameterization, which in this case is determined by the mathematical complexity (see Section 2.2.2) of the original LWR parameterization.

The input vectors for the NCAR CAM LWR parameterization include ten vertical profiles (atmospheric temperature, humidity, ozone, CO_2, N_2O, CH_4, two CFC mixing ratios (the annual mean atmospheric mole fractions for halocarbons), pressure, and cloudiness) and one relevant surface characteristic (upward LWR flux at the surface). The CAM LWR parameterization output vectors consist of the vertical profile of heating rates (HRs) and several radiation fluxes, including the outgoing LWR flux from the top layer of the model atmosphere (the outgoing LWR or OLR). 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.

NCAR CAM was run for two years to generate representative data sets. The first year of the model simulation was divided into two independent parts, each containing

input/output vector combinations. The first part was used for training and the second for validation (control of overfitting, control of a NN architecture, etc.). The second year of the simulation was used to create a test data set completely independent from both the training and validation sets. This data set was used for testing only. All approximation statistics presented in this section were calculated using this independent test data set.

Several NNs have been developed that all have one hidden layer with 20 to 300 neurons. Varying the number of hidden neurons allows one to demonstrate the dependence of the accuracy of NN emulation on this parameter, which is actually the complexity of the NN emulation, as well as selecting an optimal NN emulation [*Krasnopolsky et al.,* 2005a] with the minimal complexity (see Section 2.4.3) that still provides an emulation accuracy sufficient for a successful multi-decadal climate model integration.

All NN emulations [*Krasnopolsky et al.,* 2005a, *Krasnopolsky and Fox-Rabinovitz*, 2006a,b] developed for the NCAR CAM LWR and SWR have almost zero or negligible systematic errors (biases). Fig. 3 illustrates convergences of root mean square errors (23, 24, and 26) that 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 integrations using the NN emulations within HGCM.

Fig. 3

The NN emulation with $k = 50$ (NN50) is the simplest NN emulation that could be integrated into the model for decadal (40 years or longer) climate simulations without any visible (significant) accumulations of errors in climate simulations, compared to the control

run with the original LWR parameterization. This is the main indicator (in the framework of the NN application) that the accuracy of the NN emulation is sufficient for this application.

Fig. 4

Fig. 4 shows the vertical error profile (27) *rmsep(j)* for the "optimal" NN emulation with 50 hidden neurons, NN50 (solid line). It shows that the errors are very small; at the top 10 levels the error does not exceed 0.2 K/day, at the top 20 levels it does not exceed 0.3 K/day and reaches just about 0.6 - 0.8 K/day at the lowest level, which does not lead to significant errors in the 40 year climate simulations with HGCM. In addition to having sufficient emulation accuracy, the NN50 NN emulation performs about 150 times faster than the original NCAR CAM LWR parameterization in a code by code comparison.

For assessing the impact of using an NN emulation of the LWR parameterization in the HGCM, parallel climate simulation runs were performed with the original GCM (NCAR CAM including the original LWR parameterization) as the control run and with the HGCM (NCAR CAM including the NN emulations of LWR described above). The climate simulations were run for 50 years. As is usually done in climate simulations the simulated fields for the first 10 years, that potentially include the climate model spin-up effects, are not used for the analysis of the simulation results, leaving the remaining 40 year period to be used for that purpose[*Krasnopolsky et al.,* 2005, *Krasnopolsky and Fox-Rabinovitz*, 2006a,b]. Comparisons between the control and NN emulation runs are presented in Table 4 (rows 2, 3, and 4). They are done by analyzing the time (40-year) and global mean differences between the results of the parallel runs, as is routinely done in climate modeling. In the climate simulations performed with the original GCM and with HGCM, the time and global mean mass or mean surface pressure are precisely preserved, which is the most important preservation property for climate simulations. For the NN50 run, there is no

difference in mean sea surface pressure between the NN and control runs (see Table 4). Other time global means, some of which are also presented in Table 4, show a profound similarity between the parallel simulations for these terms. These very small differences indicate the very close results from the parallel climate simulations. Other simulations (with NN90, NN150, NN200, etc.) also show that the HGCM results are profoundly similar to those of the original GCM [*Krasnopolsky et al.,* 2005a, *Krasnopolsky and Fox-Rabinovitz*, 2006a,b]. It is noteworthy that the differences between these parallel runs (HGCM and GCM) do not exceed the differences seen in two identical GCM runs performed on different supercomputers.

Table 4

The robustness of the NN emulation approach was investigated using another GCM. The NASA NSIPP GCM (with a different LWR parameterization and other different model components compared to the NCAR CAM and its LWR parameterization) was used for this purpose. The NN emulation accuracy and complexity results in this case [*Krasnopolsky et al.,*, 2005b; *Krasnopolsky and Fox-Rabinovitz*, 2006a,b] are very similar to the ones presented above for NCAR CAM. This illustrates the robustness of the NN emulation approach.

The second component of atmospheric radiation is short wave radiation (SWR). LWR and SWR together comprise the total atmospheric radiation. The function of the SWR parameterization in atmospheric GCMs is to calculate the heating fluxes and rates produced by SWR processes. A description of the NCAR CAM atmospheric SWR parameterization is presented in a special issue of *Journal of Climate* [1998]. The input vectors for the NCAR CAM SWR parameterization include twenty-one vertical profiles (specific humidity, ozone concentration, pressure, cloudiness, aerosol mass mixing ratios, etc.) and several

relevant surface characteristics. NN emulations for the CAM-2 and CAM-3 versions of NCAR CAM SWR parameterizations have been developed [*Krasnopolsky and Fox-Rabinovitz*, 2006a,b]. 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 as compared with CAM-2. The CAM SWR parameterization output vectors consist of a vertical profile of heating rates (HRs) and several radiation fluxes.

The data sets for training, validating, and testing SWR emulating NNs were generated in the same way as those for the LWR NN emulations described above. SWR NN emulations were tested against the original NCAR CAM SWR parameterizations using the independent test set.

The NN emulations of NCAR CAM-2 and CAM-3 SWR parameterizations have 173 and 451 inputs, respectively, and 33 outputs, which are the same numbers as the inputs and outputs for the original NCAR CAM-2 and CAM-3 SWR parameterizations. As in the case of the LWR parameterizations, several NNs were developed that all have one hidden layer with 20 to 300 neurons. The NN emulation with $k = 55$ (NN55) is the simplest NN emulation that satisfies the sufficient accuracy criterion; it could be integrated in the HGCM for multi-decadal simulations without visible (significant) accumulations of errors in climate simulations as compared to the control run with the original SWR parameterization. Fig. 4 shows the vertical error profile (27) *rmsep(j)* for the "optimal" NN emulation NN55 (dashed line). It shows that the errors are very small, even smaller than for the LWR NN emulation. In addition to sufficient emulation accuracy, the NN55 SWR NN emulation performs about

20 times faster than the original NCAR CAM SWR parameterization in a code by code comparison.

Comparisons between the control and NN emulation SWR runs are presented in rows 5 and 6 of Table 4 (see the Table 4 explanation in the text above). The HGCM results are profoundly similar to those of the original GCM [*Krasnopolsky and Fox-Rabinovitz*, 2006a,b].

The next logical step is to combine two NN emulations (LWR and SWR) to emulate the total model radiation. The NN50 LWR emulation and NN55 SWR emulation described above were combined together in one HGCM. This HGCM with the NN emulations of the total model radiation was integrated for 40 years and the results of the climate simulation were compared with those of the NCAR CAM-2 GCM simulation control run with the original NCAR CAM LWR and SWR parameterizations. Comparisons are presented in rows 7 and 8 of Table 4 (see the Table 4 explanation in the text above). For the total radiation, the differences between HGCM and GCM runs are as small as for two previous cases (see Table 4). This demonstrates that using several NN emulations with negligible systematic errors (biases) in one HGCM does not lead to an increase of the differences between this HGCM and the original GCM.

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

The ocean wind wave model used for simulation and forecast purposes is another example of an ENM. It is based on a form of the spectral energy or action balance equation (19) 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*)

the calculation of the S_{nl} interactions requires the integration of a six-dimensional Boltzmann integral:

$$
\begin{aligned}
& S_{nl}(k_4) = T \otimes F(k) = \\
& = \omega_4 \int G(k_1, k_2, k_3, k_4) \cdot \delta(k_1 + k_2 - k_3 - k_4) \cdot \delta(\omega_1 + \omega_2 - \omega_3 - \omega_4) \\
& \times [n_1 \cdot n_3 \cdot (n_4 - n_2) + n_2 \cdot n_4 \cdot (n_3 - n_1)] dk_1\, dk_2\, dk_3 \\
& n(k) = \frac{F(k)}{\omega}; \qquad \omega^2 = g \cdot k \cdot \tanh(kh)
\end{aligned}
\tag{30}
$$

where the mapping is symbolically represented by T and the complicated coupling coefficient G contains moving singularities. This integration requires roughly 10^3 to 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]). Two decades of experience with the DIA in wave models has identified significant shortcomings in the DIA [*Tolman et al.*, 2005].

When considering the above, it is crucially important for the development of third 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 *and Tolman et al.,* 2005]. NNs can be applied here because the nonlinear interaction (30) is essentially a nonlinear mapping, symbolically represented by T, which relates two vectors F and S_{nl} (2-D fields in this case). Discretization of S and F (as is necessary in any numerical approach) reduces (30) to a continuous mapping of two vectors of finite dimensions. Modern high resolution wind wave models use descretization on a two

dimensional grid which leads to S and F vector dimensions on the order of $N \sim 1000$. It seems unreasonable to develop a NN emulation of such a high dimensionality (about 1000 inputs and outputs). Moreover, such a NN will be grid dependent.

In order to reduce the dimensionality of the NN and convert the mapping (30) to a continuous mapping of two finite vectors that are less dependent on the actual spectral discretization, the spectrum F and source function S_{nl} are expanded using systems of two-dimensional functions, each of which (Φ_i and Ψ_q) creates a complete and orthogonal two-dimensional basis

$$F \approx \sum_{i=1}^{n} x_i \Phi_i, \quad S_{nl} \approx \sum_{q=1}^{m} y_q \Psi_q, \tag{31}$$

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

$$x_i = \iint F \Phi_i, \quad y_q = \iint S_{nl} \Psi_q, \tag{32}$$

where the double integral identifies integration over the spectral space. Now, the developed NN emulation relates vectors of coefficients $\boldsymbol{X}$ and $\boldsymbol{Y}$: $\boldsymbol{Y} = \boldsymbol{T_{NN}(X)}$. Typically, $n = 20 \div 50$ and $m = 100 \div 150$ in eq. (31). Thus, the reduction of the dimensionality of the NN emulation is very significant.

Fig. 5

Different approaches to the basis functions choice has been investigated in [*Krasnopolsky et al.,* 2002]. Empirical Orthogonal Functions (EOFs) or principal components [*Lorenz,* 1956; *Jolliffe,* 2002] have been selected [*Tolman et al.,* 2005]. EOFs compose 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}. The main advantage of EOFs is the fast convergence of the decomposition (see

[*Krasnopolsky et al.,* 2002, *Tolman et al.,* 2005] for more detailes). A graphical representation of the NNIA algorithm is shown in Fig. 5.

The NNIA is nearly ten times more accurate than DIA. It is about 10^5 times faster than the original parameterization and only seven times slower than DIA. As in the case of the atmospheric radiation, a careful investigation of the parallel runs with the original ENM (the wave model with the original wave-wave interaction) and the HEM run with the NN emulation should be performed for the final test of the NN emulation [*Tolman et al*., 2005].

4.6 Discussion

4.6.1 Summary and advantages of the hybrid modeling approach

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

(i) the conceptual and practical possibility of developing HEMs with accurate NN emulations of model components, which preserve the integrity and all the detailed features of the original ENM;

(ii) NN emulations of model physics parameterizations developed by *Krasnopolsky et al.,* [2000, 2002, 2005] are practically identical to the original physical parameterizations, due to the capability of NN techniques to very accurately emulate complex systems like the model physics. This fact allows the integrity and level of complexity of the state-of-the-art parameterizations of model physics

to be preserved. As a result, for example, a HGCM using these NN emulations produces climate simulations that are practically identical to those of the original GCM. It is noteworthy that the NN emulation developed has the same inputs and outputs as the original parameterization and is used precisely as its functional substitute within the model.

(iii) that accurate NN emulations are robust and very fast (10 to 10^5 times faster than the original parameterization) so the significant speed-up of HEM calculations can be achieved without compromising accuracy;

(iv) that 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; and

(v) that 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 cloud resolving model, that are extremely time consuming or even computationally prohibitive if used in their original form will become computationally "affordable" in HENMs when using their accurate and computationally much more efficient NN emulations.

4.6.2 Limitations of the current hybrid modeling framework

The development of NN emulations, 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 Section 2.3.3). Because of high dimensionality of the input domain that is on the order of several hundreds 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 the NN training. Another related problem is that NN emulations are supposed to be developed for an environmental or climate system that changes in time. This means that the domain configuration for a climate simulation may evolve over time, for example, when using a future climate change scenario. In both situations described the emulating NN may be forced to extrapolate beyond its generalization ability and may lead to errors in NN outputs and result in simulation errors in the corresponding HEM. The next subsection is devoted to addressing these and other issues.

4.6.3 Current and future developments of the hybrid modeling approach

Two new techniques are being developed to take care of the kind of problems outlined in the previous section and to make the NN emulation approach suitable for long-term climate change simulations and other applications - a *compound parameterization* (CP) and a NN *dynamical adjustment* (DA) [*Krasnopolsky and Fox-Rabinovitz*, 2006a,b]. Here they are only briefly outlined.

CP consists of the following three elements: the original parameterization, its NN emulation, and a quality control (QC) block. During a routine HEM simulation with CP, QC block determines (at each time step of integration 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 produced by a short retraining using the accumulated input/output records. Thus, the adapted NN emulation becomes dynamically adjusted to the changes and/or new events/states produced by the complex environmental or climate system.

There were different possible designs considered for QC [*Krasnopolsky and Fox-Rabinovitz*, 2006a,b]. The first and simplest QC design is based on a set of regular physical and statistical tests that are used to check the consistency of the NN outputs. This is the simplest, mostly generic but not sufficiently focused approach.

The second more sophisticated and effective QC design is based on training, for each NN emulation, additional NNs to specifically predict the errors in the NN emulation outputs from a particular input. If these errors do not exceed a predefined threshold the NN emulation is used; otherwise, the original parameterization is used instead. A CP of this design was successfully tested for the NCAR CAM SWR. For the SWR NN55 (see Section 4.4.2) an error NN was trained which estimated a NN55 output error *prmse(i)* (25) for each particular input vector X_i. Fig. 6 shows the comparison of two error probability density functions. One curve (solid line) corresponds to the emulation errors of NN55, another (dashed line) corresponds to the emulation errors of the CP (both errors are calculated vs. the original parameterization on the independent test set; vertical axes are logarithmic). Fig. 6 demonstrates the effectiveness of CP; the application of CP reduces medium and large errors by about an order of magnitude. It is noteworthy that for this CP less than 2% of the SWR NN emulation outputs are rejected by QC and calculated using the original SWR parameterization. Further refinement of the criteria used in the QC may result in a reduction in the already small percentage of outliers.

Fig. 6

The third QC design is based on the domain check technique proposed in the context of NN applications to satellite remote sensing (see Section 3.4). In this case, QC is based on a combination of forward and inverse NNs. This design has already been successfully applied, as a preliminary study, to the ocean wave model (Section 4.5) [*Tolman and Krasnopolsky,* 2004].

The parameterization Jacobian, a matrix of the first derivatives of parameterization outputs over inputs, may be useful in many cases. For example, in data assimilation applications (an optimal blending of observational and simulated data to produce the best possible fields) a Jacobian is used to create an adjoint (a tangent-linear approximation). A Jacobian is also instrumental for a statistical analysis of the original parameterization and its NN emulation. An inexpensive computation of the Jacobian when using a NN emulation is one of the advantages of the NN approach. Using this Jacobian in combination with the tangent-linear approximation can additionally accelerate the calculations [*Krasnopolsky et al.,* 2002]. However since the Jacobian is not trained, it is simply calculated through the direct differentiation of an emulating NN. In this case the statistical inference of a Jacobian is an ill-posed problem and it is not guaranteed that the derivatives will be sufficiently accurate (see Section 2.4.4.).

It is noteworthy that for the type of NN applications considered in this section, the NN emulation approach that treats a parameterization of model physics as a single object offers a simple and straightforward solution that alleviates the need for calculating the NN Jacobian explicitly. The adjoint tangent-linear approximation of a parameterization (e.g., of a radiation parameterization) may be considered as an independent/new parameterization, the NN emulation approach can be applied to such a parameterization, and a separate NN

emulation can be trained to emulate the adjoint. For other applications that require an explicit calculation of the NN Jacobian, several solutions have been offered and investigated (see Sections 2.4.4 and 5.2.2).

5 NN Emulations of Dependencies Between Model Variables

The output of any complex geophysical numerical model, such as models for atmospheric and ocean climate simulations or for numerical weather prediction, contains a great deal of data in the form of 2- and 3-D high resolution numerical fields of prognostic and diagnostic variables. This output contains, in an implicit form, the highly complex functional dependencies and mappings (1) between the state variables of the model. These relationships are governed by the physics and dynamics of the numerical model that was used for the simulations.

For example, when 2-D observations like surface wind, surface currents, or sea surface elevation are assimilated into an atmospheric or oceanic data assimilation system (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, 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 this kind of simplified linear dependencies from model simulations [*Mellor and Ezer*, 1991] or observed data [*Guinehut et al*, 2004] for use in an ocean DAS. However, these simplified and generalized linear dependencies that are often derived from local data sets do not properly represent the complicated nonlinear relationship between the model variables. If we were able to extract/emulate these dependencies in a simple, but not overly

simplified yet 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. These analytical functions and mappings could also be used for efficient model output compression, archiving, and dissemination, and for sensitivity studies and error analysis. The existence of a generic technique that would allow 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].

In Section 5.1 we introduce this generic NN application by using a particular application, a NN emulation for sea surface height developed by [*Krasnopolsky et al.,* 2006]. This application, namely developing NN emulations to extract functions and mappings from model outputs for their later use in DAS and for sensitivity and error analysis, requires that we obtain a NN emulation Jacobian of reasonable quality. The Jacobian analysis and ensemble approaches that improve the quality of the NN emulation and NN Jacobian [*Krasnopolsky*, 2006] are presented in Section 5.2.

5.1 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 study is the HYbrid Coordinate Ocean Model (HYCOM). This model is a primitive equation model that uses a generalized hybrid coordinate (isopycnal/terrain following (σ)/z-level) in the vertical (see [*Bleck*, 2002] for details). The hybrid coordinate extends the geographic range of applicability of the traditional isopycnal (levels of constant water density) coordinate circulation models toward

shallow coastal seas and unstratified parts of the ocean. The vertical coordinate evaluation for HYCOM is discussed in [*Chassignet et al.,* 2003]. The particular version of HYCOM used in this study has a domain that covers the entire Atlantic Ocean with an average horizontal resolution of ⅓°; and 25 vertical levels.

Since the reduced model physics has a mainly 1-D vertical structure, it was assumed that SSH η at a particular model grid point (i.e., at a particular horizontal lat/lon) depends only on the vector of state variables X at the same horizontal location and the same time. Therefore, this dependence (a target mapping) can be written as

$$\eta = \phi(X) \tag{33}$$

where ϕ is a nonlinear continuous function and X is a vector that represents a complete set of state variables, which determine SSH. In this particular case the vector X was selected as $X = \{I, \theta, z_{mix}\}$, where I is the vector of the interfaces (the vertical coordinates used in HYCOM), θ is the potential temperature vector, 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 (33) with this particular selection of components for the vector X will not be applicable in coastal areas (where the depth is less then 250 – 500 m). In the coastal areas a different set of state variables should be selected. All the statistics presented later in this section were calculated using a test set where coastal areas were excluded.

The NN technique is applied to derive an analytical NN emulation for the relationship between model state variables, X, and SSH η,

$$\eta_{NN} = \phi_{NN}(X) \tag{34}$$

using the simulated model fields which are treated as error free data. A simulation that covers almost two years (from Julian dates 303, 2002 to 291, 2004) was used to create training, validation and test data sets. Training and validation data sets cover the period from 300, 2002 to 53, 2004 and an independent test set – the period from 53, 2004 to 291, 2004. Each data set consists of $N = 536{,}259$ 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. The improvement in the accuracy of the NN emulation (RMSE) with the increase in the complexity (the number of hidden neurons) of the emulating NN was evaluated. All trained NNs have 50 inputs and one output in accordance with the dimensionalities of the target mapping (33). The number of hidden neurons k varied from 3 to 100. No significant and consistent improvement in the RMSE after the number of hidden neurons k reaches the values of 5 – 10 was found; the further improvement did not exceed 0.25 cm. Therefore, to limit NN complexity and improve its interpolation abilities (see Section 2.4.3), 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 by a time interval of about eight months from the last day of the simulation used for training and validation (52, 2004). Fields generated by the model at 00Z were used to create inputs, X, for the NN emulation (34). Then the NN emulation (34) was applied over the entire domain to generate the 2-D field of η_{NN}. This field was compared with the corresponding field of SSH η generated by the model. With the exception of several spots (most of them still close to the coastal areas) the

difference between two one-day fields does not exceed ± 10 cm. The accuracy of the NN emulation over the entire domain is considered to be satisfactory; the bias is about 1cm and RMSE is about 4.7 cm. This conclusion about the accuracy of the NN emulation (34) is based on the fact that the NN emulation will be used in the DAS together with satellite measurements of SSH that have an accuracy on the order of 5 cm or less.

The accuracy of the NN emulation may be improved using a NN ensemble approach (see the next section). 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.

5.2 NN Ensembles for improving NN emulation accuracies and reducing NN Jacobian uncertainties

As mentioned in Section 2.4.3, it is desirable to keep the NN emulation complexity (the number of hidden neurons) at the minimum in order to improve NN generalization (interpolation) ability and the stability of the NN Jacobian; however, minimization of the NN complexity reduces the approximation accuracy of NN emulations. Using an NN ensemble approach (see Section 2.4.5) is a way around this dilemma.

In the context of the problem described in the previous subsection, the NN ensemble approach leads to the following solution. The complexity of the NN emulation (34) was limited; only three hidden neurons were allowed. Then ten NN emulations (34) with the same number of hidden neurons were trained using differently perturbed initial conditions for the NN weights. As a result, a NN ensemble that consists of ten members, ten NN emulations of identical architecture (50 inputs, 1 output, and 3 neurons in 1 hidden layer) but with different weights, different approximation accuracies, and different Jacobians has been

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.

5.2.1 NN ensembles for improving the accuracy of a NN emulation; linear vs. nonlinear ensembles

After the NN ensemble was created, each NN member (a particular realization of the NN emulation (34)) was applied to the test set and the error statistics for each NN member were calculated and plotted in Fig. 7 (left). The vertical axis of the figure shows the random part of the approximation error (the standard deviation of the error) and the horizontal axis the value of the systematic error (bias). Both errors are normalized to the corresponding maximum member error (member bias or error standard deviation). Each ensemble member is represented in this figure by a star. The spread of 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 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 regular statistical sense) of these ten numbers was calculated and compared to the exact output to calculate the ensemble statistics represented by the cross in Fig. 7 (left). 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. Fig. 7 (left) also illustrates the known fact that ensemble approaches are very effective in reducing random errors; it shows that the ensemble random error is less than the random error of any of the ensemble members. The

reduction in systematic (~15%) and random (~9%) errors with respect to the maximum single member errors is moderate but still significant.

Fig. 7

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 these nonlinear correlations and to use them to improve ensemble averaging we developed a nonlinear ensemble that uses an additional averaging NN to calculate the ensemble average. The inputs to the averaging NN are composed of the outputs from the ensemble member NNs. 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 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. 7 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. 7 (left) 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.

Fig. 7 (right) shows 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 exists 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. 7 (right) 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 ensemble in Fig. 7 (right).

Fig. 7 (right) shows that the NN ensemble approach is an effective tool in reducing extreme outliers (by ~25%). However, a careful analysis of the figure reveals also other interesting features of the statistics presented in this figure. The distribution of stars shows a significant spread, and also demonstrates a significant clustering and correlation between the extreme outliers produced by ensemble members. These facts, and the position of the conservative ensemble (cross) in the figure, suggest that the members of the ensemble are correlated nonlinearly. The significant improvement introduced by the nonlinear ensemble (diamond) supports this conclusion. This technique was also applied with similar results [*Fox-Rabinovitz et al.*, 2006] to NN emulations developed for the LWR parameterization of NCAR CAM (see Section 4.4).

5.2.2 NN ensembles for reducing the uncertainty of the NN Jacobian

The NN emulation (34) can be used in the ocean DAS to enhance assimilated 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

$$\left\{\frac{\partial \phi_{NN}}{\partial X_i}\right\}_{i=1,\ldots,n},$$

$$\Delta\eta_{NN} = \sum_{i=1}^{n} \frac{\partial \phi_{NN}}{\partial X_i}\bigg|_{X=X_0} \cdot \Delta X_i \tag{35}$$

where ΔX_i are increments of state variables, X_0 is an initial value of state variables and n is the dimensionality of the vector X (the number of inputs of the NN emulation (34)). Then the calculated $\Delta\eta_{NN}$ is compared with the observed $\Delta\eta_{obs}$ and the difference is used to adjust ΔX.

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 subsection was used to create an ensemble of ten NN Jacobians, $\left\{\frac{\partial \phi_{NN}^{j}}{\partial X_i}\right\}_{i=1,..n}^{j=1,..ens}$, where ens = 10 is the number of ensemble members. Then the ensemble mean Jacobian was calculated,

$$\overline{\frac{\partial \phi_{NN}}{\partial X_i}} = \frac{1}{ens}\sum_{j=1}^{ens} \frac{\partial \phi_{NN}^{j}}{\partial X_i}, \quad i = 1,\ldots,n \tag{36}$$

Now Eq. (35) was used to calculate $\Delta\eta_{NN}$ using each ensemble member's Jacobian and the ensemble average Jacobian (36). These values of $\Delta\eta_{NN}$ 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 NNs training and validation. The fields generated by the model at 00Z were used to create inputs X for the NN emulation Jacobians. Then the NN emulation Jacobian ensemble members were applied over the entire domain (excluding coastal areas) to generate an ensemble of 2-D fields of $\Delta\eta_{NN}^{j}$ using Eq. (35). Also, $\overline{\Delta\eta_{NN}}$ was calculated

using the ensemble average Jacobian (36) in (35). A non-dimensional distance in the model state space between the vectors X_0 and $X = X_0 + \Delta X$ was also introduced,

$$S = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\Delta X_i}{X_0^i}\right)^2} \tag{37}$$

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 such case study are presented in Figs. 8 – 10.

Fig. 8

Fig. 8 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 position we moved to the left and right grid point by grid point, using X values at these grid points to calculate ΔX and the non-dimensional distance in the model state space, S (37). These values of ΔX were used in (35) to calculate $\Delta\eta$.

Fig. 9

Fig. 9 shows $\Delta\eta$ calculated using (35) and the NN ensemble member Jacobians (an envelope of thin solid lines illustrates the Jacobian uncertainties), the exact $\Delta\eta$ calculated from the model (thick solid line), and $\Delta\eta$ calculated using the ensemble average Jacobian (36) (thick dashed line). $\Delta\eta$ is shown vs. the distance in the model state space, S. This figure demonstrates how significantly the NN Jacobian can be improved by using the ensemble average. The larger the distance S, the more significant the reduction of the Jacobian uncertainties becomes.

Fig. 10 (left) shows the bias and random error for $\Delta\eta$ calculated along the path shown in Fig. 8 using (35). The asterisks correspond to the errors when ensemble member Jacobians were used in (35), and the cross to the ensemble average Jacobian (36). The

ensemble bias is equal to the mean bias of the members as expected when using this simple method to calculate the ensemble average. This figure also shows that in the case of Jacobian the ensemble approach very effectively reduces random errors or error SD; the ensemble random error (~1.1 cm) is less than the random error of any ensemble member. The reduction in bias (~90%) and random error (~65%) with respect to the maximum single member errors is very significant.

Fig. 10

Fig. 10 (right) shows minimum and maximum errors along the path or the statistics for extreme outliers. When each ensemble member NN Jacobian is applied to (35), an output is produced with an error. 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 behavior that we can expect from this particular NN emulation. These two extreme outliers (negative and positive) for each NN member are represented in the figure by stars. When used in (35), the ensemble average Jacobian (36) also generates two extreme outliers that are represented in the figure by the cross. This figure shows that the NN ensemble approach is also an effective tool for reducing large errors (by ~4 times) in NN Jacobians.

Then this same procedure was applied at all grid points in the model domain. The errors have been calculated along numerous paths (both horizontal and vertical) all over the model domain. The RMS error in $\Delta\eta$ was calculated as a function (averaged in each bin) of the non-dimensional distance S over the entire domain. The ensemble significantly improves statistics at all distances S considered. In this case, the ensemble is always better than the best ensemble member.

To better understand the magnitudes of errors presented in this and the previous section, these magnitudes should be compared with the errors in the observed satellite data $\Delta\eta_{obs}$ assimilated in the oceanic DAS (see the previous subsection). The accuracy of the observed data is about 5 cm. That means our NN emulation (33) and the ensemble techniques allow a reduction in the Jacobian uncertainties and produce an ensemble Jacobian (36) that is sufficiently accurate to be used in ocean DASs.

5.3 Discussion

The NN application described in Section 5.1 is an important application *per se*. The output of any complex geophysical numerical model contains, in an implicit form, the highly complex functional dependencies and mappings between the model variables. The functions and mappings provided by the NN emulation approach in a simple analytical form (34) could be used to improve understanding of the underlying nonlinear dependencies, which is a matter of a great scientific interest and practical importance. These NN emulations can also be used for efficient model output compression, archiving, and dissemination, and for sensitivity studies and error analysis.

On the other hand, a transition from the raw model output data to analytical equations like (34) can be considered as a first step in a new field of science dealing with knowledge generalization, structuring, and compression. This field is very new and very broad; the example in Section 5.1 is one particular example in one particular subfield; however, it definitely points in the direction of knowledge generalization. This example shows that NNs and similar techniques also have a realistic potential to become a generic tool in the new field of knowledge generalization, structuring, and compression.

It is also clear that the ensemble approaches introduced in Section 5.2 within the context of ocean sea surface elevation mapping are generic approaches that can be used and have been already used [*Fox-Rabinovitz et al.*, 2006] in other applications. In particular, a new ensemble approach introduced in Section 5.2.2 to reduce uncertainties in the NN emulation Jacobian has a wide area of application where a calculation of the NN Jacobian in a NN application is also desirable.

6 Conclusions

During the last several decades an objective generic trend surfaced in geophysical studies, from 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 followed by a trend in geophysical modeling (including statistical modeling); a transition from simple, low-dimensional, linear or weakly nonlinear models to complex, multidimensional, nonlinear models and from simple, linear statistical tools like linear models and linear regressions to sophisticated, nonlinear statistical tools like nonlinear regressions, NNs, support vector machines, etc.

Transitioning to nonlinear models and statistical tools showed their generally greater adequacy in addressing the problems that are considered by modern geophysics. This transition and the following intensive use of nonlinear statistical tools and models also revealed their complexity and flexibility that, if not controlled properly, may lead in some cases to undesirable results and erroneous predictions. What should the strategy be in this situation? Should we return to simple linear tools and models? Unfortunately, we cannot do that because the objects of our study have become essentially complex and nonlinear. In our 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 adequate, and to learn how to control possible undesirable side effects while maximizing the advantages that the complexity and flexibility of these nonlinear models and tools offer to our studies.

In discussions included in this review, we tried to emphasize that the transition from linear statistical tools to nonlinear ones (like NNs) requires, to some extent, an adjustment of our thinking process or our philosophy. For example, when we deal with relatively simple linear systems and use linear statistical tools (such as a simple linear regression) to model these systems, we can assume in some cases that parameters of our statistical model have a physical meaning, that they are directly related to characteristics of physical processes 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 (like NNs) we probably should, from the beginning, focus on getting good predictions of the behavior of the system under consideration but not on giving physical interpretations to the multiple parameters of our nonlinear model [*Vapnik and Kotz*, 2006].

In this review, we presented and discussed one particular type of NN technique – the MLP NN – and one particular application of this NN, the NN Emulations for Complex Multidimensional Geophysical Mappings. It was shown that even this one generic NN application and type of NN technique cover a great 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 (like the 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*., 2003]. There are also other types of NNs that provide

solutions to other generic applications (like the pattern recognition problems [*Ripley*, 1996; *Nabney*, 2002]), but this review does not deal directly with these types of NNs and NN 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 the NN that provides a solution, and NN generalization capabilities. These generic issues and their discussions are applicable to other types of NNs and other NN applications.

Acknowledgements

The author would like to thank Dr. M. Fox-Rabinovitz for fruitful collaboration, multiple and stimulating discussions of topics presented in this paper and for careful reading and commenting the manuscript. The author also thanks Drs. Chalikov and Tolman for long-term fruitful collaboration.

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., M. Schmitt, A. Chedin, and N. Scott (1999), The "Weight Smoothing" Regularization of MLP for Jacobian Stabilization, *IEEE Trans. Neural Networks*, *10*, 1502-1510.

Aires, F., C. Prigent, and W.B. Rossow (2004a), Neural network uncertainty assessment using Bayesian statistics: A remote sensing application, *Neural Computation, 16*, 2415-2458.

Aires, F., C. Prigent, and W.B. Rossow (2004b), Neural network uncertainty assessment using Bayesian statistics with application to remote sensing: 3. Network Jacobians, *J. Geophys. Res., 109*, D10305

Aires, F., W.B. Rossow, N.A. Scott, and A. Chedin (2002), Remote sensing from the infrared atmospheric sounding interferometer instrument. 2. Simultaneous retrieval of temperature, water vapor, and ozone atmospheric profiles. *J. Geophys. Res., 107*, 4620

Alishouse, J. C., et al. (1990), Determination of oceanic total precipitable water from the SSM/I, *IEEE Transactions on Geoscience and Remote Sensing*, *GE, 23*, 811–816.

Atkinson, P.M., and A.R.L. Tatnall (1997), Neural networks in remote sensing – introduction, *Int. J. Remote Sens.*, *18:* (4), 699-709.

Attali, J-G., and G. Pagès (1997), Approximations of Functions by a Multilayer Perceptron: A New Approach, *Neural Networks*, *6*, 1069-1081.

Bankert, R.L. (1994), Cloud pattern identification as part of an automated image analysis, in *Proc. Seventh Conference on Satellite Meteorology and Oceanography, Monterey, CA,*

June 6-10, 1994, 441-443

Barai, S.V., and Y. Reich (1999), Ensemble modeling or selecting the best model: Many could be better than one, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 13*, pp. 377-386

Barron, A.R. (1993), Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Trans. Inform. Theory*, *39*, 930-945

Beale, R., and T. Jackson (1990), *Neural Computing: An Introduction*, 240 pp., Adam Hilger, Bristol, Philadelphia and New York.

Bhattacharya, B., C. Buraimo, D.P. Solomatine (2005), Data-driven Modelling in Context to Sediment Transport. *Journal of Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and AtmospheresParts A/B/C*, *30* (4-5), 297-302.

Bhattacharya, B., and D.P. Solomatine (2006), Machine learning in soil classification, *Neural Network, 19*, 186-195

Bishop, C.M. (1995), *Neural Networks for Pattern Recognition*, 482 pp., Oxford University Press, Oxford, U.K.

Bleck, R. (2002), An oceanic general circulation model framed in hybrid isopycnic-cartesian coordinates, *Ocean Modell., 4***,** 55–88.

Brajard, J., C. Jamet, C. Moulin, and S. Thiria (2006), Use of a neuro-variational inversion for retrieving oceanic and atmospheric constituents from satellite ocean color sensor: Application to absorbing aerosols, *Neural Networks*, *19*, 178-185.

Brown, T.J. and P.W. Mielke (2000), *Statistical mining and data visualization in atmospheric sciences*, Boston: Kluwer Academic Publishers. 80 pp.

Cabrera-Mercader, C. R., and D. H. Staelin (1995), Passive microwave relative humidity

retrievals using feedforward neural networks, *IEEE Transactions on Geoscience and Remote Sensing, 33*, 1324–1328.

Cardaliaguet, P., and G. Euvrard (1992), Approximation of a function and its derivatives with a neural network, *Neural Networks, 5*, 207-220.

Chassignet, E. P., L. T. Smith Jr., G. R. Halliwell Jr., and R. Bleck (2003), North Atlantic simulations with the Hybrid CoordinateOcean Model (HYCOM): Impact of the vertical coordinate choice, reference pressure, and thermobaricity, *J. Phys.Oceanogr., 33*, 2504–2526.

Chen, T., and H. Chen (1995a), Approximation Capability to Functions of Several Variables, Nonlinear Functionals and Operators by Radial Basis Function Neural Networks, *Neural Networks*, *6*, 904-910.

Chen, T., and H. Chen (1995b) Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Function and Its Application to Dynamical Systems, *Neural Networks*, *6*, 911-917.

Cheng, B., and D.M. Titterington (1994), Neural Networks: A Review from a Statistical Perspective, *Statistical Science, 9*, 2-54.

Cherkassky, V., and F. Mulier (1998), *Learning from data*, 441 pp., John Wiley, Hoboken, N.J.

Chevallier, F., F. Chéruy, N. A. Scott, and A. Chedin (1998), A Neural Network Approach for a Fast and Accurate Computation of Longwave Radiative Budget, *J. Appl. Meteor., 37,* 1385-1397.

Chevallier, F., J.-J. Morcrette, F. Chéruy, and N. A. Scott (2000), Use of a neural-network-based longwave radiative transfer scheme in the EMCWF atmospheric model. *Quarterly*

Journal of Royal Meteorological Society, 126, 761-776

Chevallier, F., and J-F Mahfouf (2001), Evaluation of the Jacobians of Infrared Radiation Models for Variational Data Assimilation, *J. Appl. Meteor.*, *40*, 1445-1461.

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, *Monthly Weather Review*, *133*, 3721-3723.

Chou, M.-D., M. J. Suarez, X.-Z. Liang, and M. M.-H. Yan (2001), A Thermal Infrared Radiation Parameterization for Atmospheric Studies, Technical Report Series on Global Modeling and Data Assimilation, Editor Max J. Suarez, NASA/TM-2001-104606, Vol. 19.

Cilliers, P. (1998), *Complexity and Postmodernism: Understanding Complex Systems*, 157 pp., Routledge, London, UK.

Colliers, P. (2000), What Can We Learn From a Theory of Complexity?, *Emergence, 2*, No. 1, 23-33,(doi: 10.1207/S15327000EM0201_03)

Collins, W.D. (2001), Parameterization of generalized cloud overlap for radiative calculations in general circulation models, *J. Atmos. Sci., 58*, 3224-3242

Collins, W.D, J.K. Hackney, and D.P. Edwards (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 (D22)*, 1-20

Cornford, D., I. T. Nabney, and G. Ramage (2001), Improved neural network scatterometer forward models, *J. Geophys. Res., 106*, 22331–22338.

Cybenko, G. (1989), Approximation by Superposition of Sigmoidal Functions, *Math. Control Signals and Systems*, *2*, 303-314.

Davis, D. T., 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. and Remote Sens., 33*, 1182–1193.

Derber, J. C., and W.-S. Wu (1998), The use of TOVS cloud-cleared radiances in the NCEP SSI analysis system, *Monthly Weather Reviews, 126*, 2287–2299.

DeVore, R.A. (1998), Nonlinear approximation, *Acta Numerica*, *8*, 51-150.

Dibike, Y. B., and P. Coulibaly (2006), Temporal neural networks for downscaling climate variability and extremes, *Neural Networks, 19*, 135-144

Elsner, J.B., and A.A. Tsonis (1992), Nonlinear prediction, chaos, and noise, *Bull. Americ. Metoer. Soc., 73*, 49-60

Eyre, J. R., and A. C. Lorenc (1989), Direct use of satellite sounding radiances in numerical weather prediction, *Meteorology Magazine, 118*, 13–16.

Fox-Rabinovitz, M. S., V. M. Krasnopolsky, and A. Belochitski (2006), Ensemble of Neural Network Emulations for Climate Model Physics: The Impact on Climate Simulations, in *Proceedings of the IJCNN2006, July 16-21, 2006, Vancouver, BC, Canada,* pp. 9321-9326, CD-ROM.

Funahashi, K. (1989), On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks*, *2,* 183-192.

Gardner, M.W., and S.R. Dorling (1998), Artificial neural networks (the multilayer perceptron) – A review of applications in the atmospheric sciences, *Atmos. Environ., 32*, 2627-2636.

Gell-Mann, M., and S. Lloyd (1996), Information Measures, Effective Complexity, and Total Information, *Complexity, 2*, 44-52.

Goodberlet, M. A., C. T. Swift, and J. C. Wilkerson (1989), Remote sensing of ocean surface winds with the special sensor microwave imager, *J. Geophys. Res., 94*, 14547–14555.

Goodberlet, M. A., and C. T. Swift (1992), Improved retrievals from the DMSP wind speed algorithm under adverse weather conditions. *IEEE Transactions on Geoscience and Remote Sensing, 30,* 1076–1077.

Guinehut, S., P.Y. Le Traon, G. Larnicol, S. Philipps (2004), Combining Argo and remote-sensing data to estimate the ocean three-dimensional temperature fields – a first approach based on simulated data, *J. Marine Systems*, *46*, 85-98

Hansen, L. K., and P. Salamon (1990), Neural network ensembles, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *12*(10), 993-1001.

Hashem, S. (1997), Optimal linear combination of neural networks, *Neural Networks, 10*, 599-614.

Hasselmann, S., and K. Hasselmann (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, *Journal of Physical Oceanography, 15*, 1369-77.

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, *Journal of Physical Oceanography, 15*, 1378-1391.

Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*, 696 pp., Macmillan College Publishing Company, New York, U.S.A.

Hennon, C. C., C. Marzban, and J. S. Hobgood (2005), Improving tropical cyclogenesis statistical model forecasts through the application of a neural network classifier, *Wea. Forecasting*, *20*, 1073-1083,doi: 10.1175/WAF890.1.

Hopfield, J.J. (1982), Neural networks and physical systems with emergent collective computational ability, *Proceedings of the National Academy of Science of the U.S.A., 79,* 2554-2558

Hornik, K., M. Stinchcombe, and H. White (1990), Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *Neural Networks, 3*, 551-560

Hornik, K. (1991), Approximation Capabilities of Multilayer Feedforward Network. *Neural Networks, 4*, pp. 251-257.

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

Hsieh, W.W. (2001), Nonlinear principle component analysis by neural networks, *Tellus, 53A*, 599-615

Hsieh, W.W. (2004), Nonlinear Multivariate and Time Series Analysis by Neural Network Methods, *Rev. Geophys., 42*, RG1003

Jolliffe, I.T. (2002), *Principal Component Analysis*, 502 pp., Springer-Verlag, New-York.

Journal of Climate (1998), *11*, No. 6 (the special issue).

Kohonen, T. (1982), Self-organizing formation of topologically correct feature maps, *Biol. Cybernetics, 43,* 59-69

Kon, M., and L. Plaskota (2001), Complexity of Neural Network Approximation with limited information: a worst case approach, *J. Complexity*, *17*, 345-365

Krasnopolsky, V., L. C. Breaker, and W. H. Gemmill (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., W. H. Gemmill, and L. C. Breaker (1996), A new transfer function for SSM/I based on an expanded neural network architecture. Technical note, OMB contribution No. 137, NCEP/NOAA.

Krasnopolsky, V (1997), A neural network-based forward model for direct assimilation of SSM/I brightness temperatures, Technical note, OMB contribution No. 140, NCEP/NOAA.

Krasnopolsky, V. M., W. H. Gemmill, and L. C. Breaker (1999), A multiparameter empirical ocean algorithm for SSM/I retrievals, *Canadian Journal of Remote Sensing*, *25*, 486–503.

Krasnopolsky, V. M., W. H. Gemmill, and L. C. Breaker (2000a), A neural network multi-parameter algorithm SSM/I ocean retrievals: comparisons and validations, *Remote Sensing of Environment, 73*, 133–142.

Krasnopolsky, V.M. et al. (2000b), Application of neural networks for efficient calculation of sea water density or salinity from the UNESCO equation of state, in *Proceedings, Second Conference on Artificial Intelligence, AMS, Long Beach, CA, 9–14 January, 2000*, pp. 27–30.

Krasnopolsky,V.M, D.V. Chalikov, and H.L. Tolman (2002), A neural network technique to improve computational efficiency of numerical oceanic models, *Ocean Modelling, 4*, 363-383.

Krasnopolsky, V.M., and F. Chevallier (2003), Some neural network applications in environmental sciences. Part II: Advancing computational efficiency of environmental numerical models, *Neural Networks, 16*, 335-348.

Krasnopolsky, V.M., M.S. Fox-Rabinovitz, and D.V. Chalikov (2005a), New Approach to Calculation of Atmospheric Model Physics: Accurate and Fast Neural Network Emulation of Long Wave Radiation in a Climate Model, *Monthly Weather Review*, *133*, 1370-1383.

Krasnopolsky,V.M, M.S. Fox-Rabinovitz, and Ming-Dah Chou (2005b), Robustness of the NN Approach to Emulating Atmospheric Long Wave Radiation in Complex Climate Models, in *Proc. of International Joint Conference on Neural Networks*, July 31-August 4, 2005, Montréal, Québec, Canada, pp. 2661-2665.

Krasnopolsky, V.M., M.S. Fox-Rabinovitz, and D.V. Chalikov (2005c), Reply, *Monthly Weather Review*, *133*, 3724-3729.

Krasnopolsky, V.M., and M.S. Fox-Rabinovitz (2006a), A New Synergetic Paradigm in Environmental Numerical Modeling: Hybrid Models Combining Deterministic and Machine Learning Components, *Ecological Modelling, v. 191*, 5-18.

Krasnopolsky, V.M., and M.S. Fox-Rabinovitz (2006b), Complex Hybrid Models Combining Deterministic and Machine Learning Components for Numerical Climate Modeling and Weather Prediction, *Neural Networks, 19*, 122-134

Krasnopolsky, V.M., and V.I. Kukulin (1977), A stochastic variational method for the few-body systems, *J. Phys. G: Nucl. Phys.*, *3*, 795-807.

Krasnopolsky, V.M., C. J. Lozano, D. Spindler, I. Rivin, and D.B. Rao (2006), A new NN approach to extract explicitly functional dependencies and mappings from numerical

outputs of numerical environmental models, in *Proceedings of the IJCNN2006, July 16-21, 2006, Vancouver, BC, Canada*, pp. 8732-8734, CD-ROM.

Krasnopolsky, V.M. (2006), Reducing Uncertainties in Neural Network Jacobians and Improving Accuracy of Neural Network Emulations with NN Ensemble Approaches, in *Proceedings of the IJCNN2006, July 16-21, 2006, Vancouver, BC, Canada*, pp. 9337-9344, CD-ROM.

Krasnopolsky, V.M., and H. Schiller (2003), Some neural network applications in environmental sciences. Part I: Forward and inverse problems in satellite remote sensing, *Neural Networks, 16*, 321-334.

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

Li, S., W. W. Hsieh, and A. Wu (2005), Hybrid coupled modeling of the tropical Pacific using neural networks, *J. Geophys. Res.*, *110*, doi: 10.1029/2004JC002595.

Lippmann, R.P. (1989), Pattern classification using neural networks, *IEEE Commun. Mag.*, *27*, 47-64.

Lorenc, A. C. (1986), Analysis methods for numerical weather prediction, *Quarterly Journal of Royal Meteorology Society, 122*, 1177–1194.

Lorenz, E.N. (1956), *Empirical Orthogonal Functions and Statistical Weather Prediction*, Sci. Rep., No. 1, Statistical Forecasting Project, M.I.T., Cambridge, MA, 48 pp.

Loyola D., T. Ruppert (1998), A new PMD cloud-recognition algorithm for GOME, *ESA Earth Observation Quarterly*, *58*, 45-47.

MacKay, D.J.C. (1992), A practical Bayesian framework for back-propagation networks, *Neural Comput., 4(3)*, 448-472

Maclin, R., and J. Shavlik (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

Marzban, C., and G. J. Stumpf (1996), A neural network for tornado prediction based on doppler radar-derived attributes, *J. Appl. Met.*, *35*, 617-626.

McCulloch, W.S., and W. Pitts (1943), A logical calculus of the ideas immanent in neural nets, *Bull. Math. Biophys., 5*, 115-137.

McNally, A. P., J. C. Derber, W.-S. Wu, and B. B. Katz (2000), The use of TOVS level 1B radiances in the NCEP SSI analysis system, *Quarterly Journal of Royal Meteorological Society, 126*, 689–724.

Mellor, G. L., and T. Ezer (1991), A Gulf Stream model and an altimetry assimilation scheme, *J. Geophys. Res.*, *96*, 8779-8795.

Meng, L., et al. (2006), Neural network retrieval of ocean surface parameters from SSM/I data, *Month. Wea. Rev.*, in press.

Mueller, M. D. 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.

Nabney, I.T. (2002), *Netlab: Algorithms for pattern recognition*, Springer-Verlag, New York.

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

Neal, R.M. (1996), *Bayesian learning for neural networks*, Springer-Verlag, New York.

Nguyen, D., and B. Widrow (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, , 3*, pp. 21-26.

Nilsson, N. J. (1965), *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, 137 pp., NY: McGraw Hill, New York.

Opitz D., and R. Maclin (1999), Popular ensemble methods: an empirical study, *J. Artificial Intelligence Res., 11*, 169-198.

Parker, R. L. (1994), *Geophysical inverse theory*, 400 pp, Princeton: Princeton University Press.

Parrish, D. F., and J. C. Derber (1992), The national meteorological center's spectral statistical-interpolation analysis system, *Month. Weather Rev., 120*, 1747–1763.

Pasini, A, M. Lor`e, and F. Ameli (2006), Neural network modelling for the analysis of forcings/temperatures relationships at different scales in the climate system, *Ecological Modelling, 191,* 58–67

Petty, G. W. (1993), A comparison of SSM/I algorithms for the estimation of surface wind, in *Proceedings Shared Processing Network DMSP SSM/I Algorithm Symposium*, 8–10(June), 1993.

Petty, G. W., and K. B. Katsaros (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, *Journal of Atmospheric Oceanic Technology*, *9*, 746–761.

Petty, G. W., and K. B. Katsaros (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, *Journal of Atmospheric Oceanic Technology, 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, *Quarterly Journal of Royal Meteorological Society*, *122,* 327–355.

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

Prigent, C., L. Phalippou, and S. English (1997), Variational inversion of the SSM/I observations during the ASTEX campaign, *J. Applied Meteor*., *36*, 493–508.

Reitsma F. (2001), *Spatial Complexity*, 149 pp, Masters Thesis, Auckland University, New Zealand.

Ripley, B.D. (1996), *Pattern Recognition and Neural Networks*, 403 pp., Cambridge University Press, Cambridge, U.K.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams (1986), Learning internal representations by error propagation, in *Parallel Distributed Processing*, vol. 1, edited by D.E. Rumelhart, J.L. McClelland, and P.R. Group, pp. 318-362, MIT Press, Cambridge, Mass.

Schiller, H., and R. Doerffer (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.

Schiller, H., and V. M. Krasnopolsky (2001), Domain check for input to NN emulating an inverse model, in *Proceedings of International Joint Conference on Neural Networks*, pp. 2150–2152, July 15–19, Washington, DC.

Selfridge, O. G. (1958), Pandemonium: a paradigm for learning. In Mechanisation of Thought Processes, in *Proc. of a Symp. Held at the National Physical Lab.*, HMSO, London, pp. 513-526.

Sharkey, A. J. C. (1996), On combining artificial neural nets, *Connection Science*, *8*, 299-313.

Smith, J. A. (1993), LAI inversion using a back-propagation neural network trained with a multiple scattering model, *IEEE Transactions on Geoscience and Remote Sensing*, *GE-31*, 1102–1106.

Solomatin, D. P. (2005), Data-driven Modeling and Computational Intelligence Methods in Hydrology, *Encyclopedia of Hydrological Sciences*, Wiley.

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

Stogryn, A. P., C. T. Butler, and T. J. Bartolac (1994), Ocean surface wind retrievals from special sensor microwave imager data with neural networks, *J. Geophys. Res., 90*, 981–984.

Tang, Y., and W. W. Hsieh (2003), ENSO simulation and prediction in a hybrid coupled model with data assimilation., *J. Meteorol. Soc. Japan*, *81*, 1-19.

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

Tolman, H.L. (2002), User manual and system documentation of WAVEWATCH III version 2.22. Technical note 222, NOAA/NWS/NCEP/MMAB, 133 pp.

Tolman, H.L. and V.M. Krasnopolsky (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, H.L., V.M. Krasnopolsky, and D.V. Chalikov (2005), Neural network approximations for nonlinear interactions in wind wave spectra: direct mapping for wind seas in deep water, *Ocean Modelling, 8*, 253-278

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 Transactions on Geoscience and Remote Sensing*, *GE-30*, 1015–1024.

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

Vapnik, V.N. (1995), *The Nature of Statistical Learning Theory*, 189 pp., Springer, New York.

Vapnik, V.N., and S. Kotz (2006), *Estimation of Dependences Based on Empirical Data (Information Science and Statistics)*, 495 pp., Springer, New York.

Washington, W.M. and D.L. Williamson (1977), A Description of NCAR GCM's in General Circulation Models of the Atmospheres. *Methods in Computational Physics*, J. Chang Ed., 17, Academic Press, 111-172.

Weigend, A.S., and N.A. Gershenfeld (1994), The future of time series: learning and understanding, pp. 1-70, In *Time series prediction. Forecasting the future and understanding the past*, eds. A.S. Weigend,, and N.A. Gershenfeld, pp.643, Addison-Wesley Publishing Company, Reading, MA

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

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

Wessels L. F. A., and E. Bernard (1992), Avoiding false local minima by proper initialization of connections, *IEEE Trans. Neural Networks, 3*, pp. 899-905.

Wu, A., W. W. Hsieh, and B. Tang (2006), Neural network forecasts of the tropical Pacific sea surface temperatures, *Neural Networks, 19*, 145-154

Tables

Table 1. Some NN Applications in Climate and Weather Related Fields.

The list of applications included in the table is not 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 paper are shown in italic in the table.

NN APPLICATION		PUBLICATIONS
I. Satellite Meteorology and Oceanography		
I.1	Classification	*Bhattacharya and Solomatine* [2006]
I.2	Pattern Recognition, Feature Extraction	*Bankert* [1994]
I.3	Change Detection & Feature Tracking	*Valdés and Bonham-Carter* [2006]
I.4	*Fast Forward Models for Variational Retrievals*	*Krasnopolsky* [1997]
I.5	*Accurate Transfer Functions (Retrieval Algorithms)*	
	I.5.1 Surface Parameters	*Stogryn et al*. [1994]*; Krasnopolsky and Schiller* [2003]
	I.5.2 Atmospheric Profiles	*Aires et al.* [2002]; *Mueller et al.* [2003]
II. Predictions		
I.1	Geophysical Time Series	*Elsner and Tsonis* [1992]
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	New Hybrid Parameterizations of Physics	*Chevallier et al.* [1998]
III.2	*Fast Emulations of Model Physics*	*Krasnopolsky et al.* [2002, 2005a]
III.3	*Fast Forward Models for Direct Assimilation*	*Krasnopolsky* [1997]
III.4	*Forward Models for Propagating a Signal to Different Vertical Levels and Variables*	*Krasnopolsky* [2006]
III.5	Hybrid Coupled Models	*Tang and Hsieh* [2003]
IV. Geophysical Data Fusion		*Loyola and Ruppert* [1998]
V. Geophysical Data Mining		*Brown and Mielke* [2000]
VI. Interpolation and Downscaling		*Dibike and Coulibaly* [2006]
VII. Nonlinear Multivariate Statistical Analysis		*Hsieh* [2004]
VIII. Hydrology		*Bhattacharya et al*., [2005]

Table 2. Comparison of physically based (PB) radiative transfer and empirical NN forward models under clear and clear+cloudy (in parentheses) weather conditions.

Author	**Type**	**BT RMS Error (°K)**	
		Vertical	**Horizontal**
P & K (1992)	PB	1.9 (2.3)	3.3 (4.3)
Wentz (1997)	PB	2.3 (2.8)	3.4 (5.1)
Krasnopolsky (1997)	NN, emp.	1.5 (1.7)	3.0 (3.4)

Table 3. Error budget (in m/s) for different SSM/I wind speed algorithms under clear and clear+cloudy (in parentheses) conditions.

Algorithm	**Method**	**Bias in m/s**	**Total RMSE in m/s**	**RMSE in m/s for W > 15 m/s**
GSW[1]	Multiple linear regression	-0.2 (-0.5)	1.8 (2.1)	(2.7)
GSWP[2]	Generalized Linear regression	-0.1 (-0.3)	1.7 (1.9)	(2.6)
GS[3]	Nonlinear regression	0.5 (0.7)	1.8 (2.5)	(2.7)
Wentz[4]	Physically-based	0.1 (-0.1)	1.7 (2.1)	(2.6)
NN1[5]	Neural network	-0.1 (-0.2)	1.5 (1.7)	(2.3)
NN2[6]	Neural network	(-0.3)	(1.5)	–

[1]*Goodberlet et al.,* 1989 [2]*Petty*, 1993 [3]*Goodberlet and Swift*, 1992 [4]*Wentz*, 1997

[5]*Krasnopolsky et al.,* 1996,1999 [6]*Meng et al.*, 2006

Table 4. Time (40-years) and global means for mass (mean sea level pressure) and other model diagnostics for the NCAR CAM-2 climate simulations with the original LWR & SWR parameterizations (in GCM) – row 1, their NN emulations (in HGCM), and their differences (in %). NN50 (LWR) and NN55 (SWR) are used. Rows 3 and 4 show results for HGCM with LWR NN emulation only and differences vs. the original GCM. Rows 5 and 6 show results for HGCM with SWR NN emulation only and differences vs. the original GCM. Rows 7 and 8 show results for HGCM with both NN emulations and differences vs. the original GCM.

Fields / **GCM versions**	**Mean Sea Level Pressure (*hPa*)**	**Surface Temperature (°K)**	**Total Precipitation (*mm/day*)**	**Total Cloudiness (*fractions, %*)**	**Wind at 12 km *(m/s)***
GCM with the original LWR & SWR Parameterizations	1011.48	289.02	2.86	60.71	16.21
HGCM with LWR NN Emulation	1011.48	288.97	2.89	61.26	16.16
Difference in %	0.	0.02	1.04	0.9	0.3
HGCM with SWR NN Emulation	1011.49	288.97	2.86	60.89	16.20
Difference in %	0.001	0.02	0.	0.3	0.06
HGCM with LWR & SWR NN Emulations	1011.50	288.92	2.89	61.12	16.29
Difference in %	0.002	0.03	1.04	0.6	0.5

Figure Captions

Fig. 1. Simplest feed-forward (data propagate forward from input to output), fully connected (each neuron in the previous layer is connected to each neuron in the following layer) MLP NN. The input layer consists of the input neurons x_i, the hidden layer consists of the hidden neurons z_j, and the output layer consists of the output neurons y_q.

Fig. 2. A generic neuron (3). In general, it consists of a linear and a nonlinear part. In our case, the output neurons, y_q, in Eq. (2) have only linear part.

Fig. 3. The convergence of root mean square errors (23, 24, and 26). Solid line – $RMSE_m$ (24) for $m = 26$, dashed line – RMSE (23), and dotted line – PRMSE (26).

Fig. 4. The vertical error profiles (27), *rmsep(j),* for the "optimal" LWR NN emulation with 50 hidden neurons (NN50) – solid line and for the "optimal" SWR NN emulation with 55 hidden neurons (NN55) – dashed line.

Fig. 5. Graphical representation of the NNIA algorithm.

Fig. 6. Probability density distributions of emulation errors for the SWR NN emulation NN55 (solid line) and for the compound SWR parameterization. Both errors are calculated vs. the original SWR parameterization. Compound parameterization reduces the probability of medium and large errors by an order of magnitude. Horizontal axis is RMSE in K/day; vertical axis is logarithmic.

Fig. 7. The left panel: 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 the diamond at the figure. The right panel shows extreme outliers' statistics: 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 ensemble – by the diamond at the figure.

Fig. 8. The location of the cross section (white horizontal line) inside the model domain; white dot show the position of X_0. The picture is plotted in the internal model coordinates.

Fig. 9. $\Delta\eta$ calculated using (35) and the NN ensemble member Jacobians (an envelope of thin solid lines that illustrates the Jacobian uncertainties), exact $\Delta\eta$ calculated from the model (thick solid line), and $\Delta\eta$ calculated using the ensemble average Jacobian (36) (thick dashed line). $\Delta\eta$ is shown vs. the distance in the model state space, S (37).

Fig. 10 The left panel: the systematic error (bias) and the random error (error standard deviation) for $\Delta\eta$ calculated along the path shown in the upper right panel using (35); the stars correspond to errors when the ensemble member Jacobians were used in (35), the cross corresponds to the case when the ensemble average Jacobian (36) was used. The right panel shows the minimum and maximum errors along the path; the stars correspond to errors when the ensemble member Jacobians were used in (35) and the cross to when the ensemble average Jacobian (36) was used in (35).

Figures

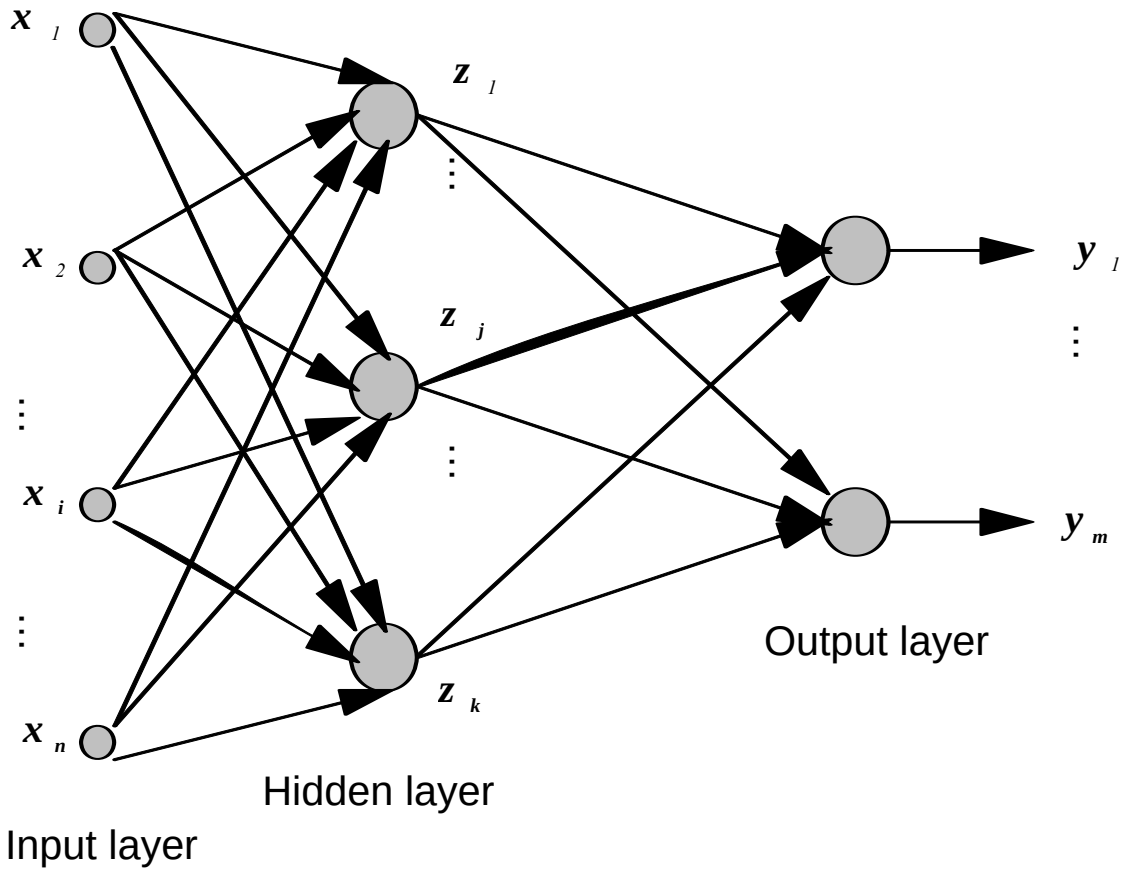

Fig. 1. Simplest feed-forward (data propagate forward from input to output), fully connected (each neuron in the previous layer is connected to each neuron in the following layer) MLP NN. The input layer consists of the input neurons x_i, the hidden layer consists of the hidden neurons z_j, and the output layer consists of the output neurons y_q.

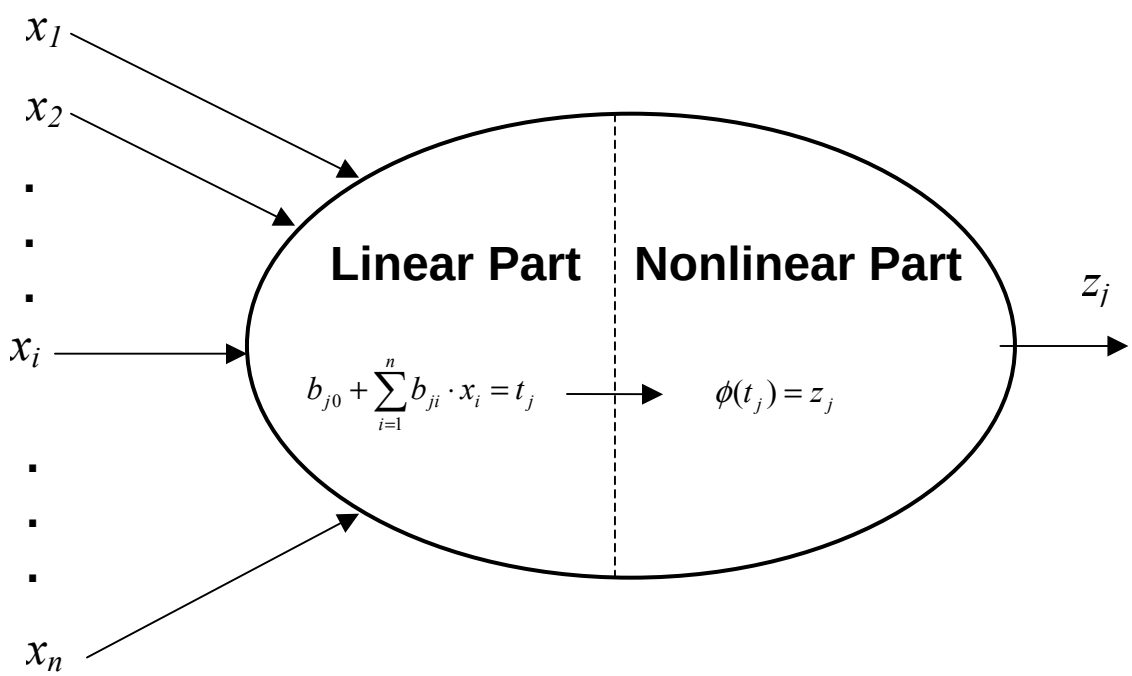

Fig. 2. A generic neuron (3). In general, it consists of a linear and a nonlinear part. In our case, the output neurons, y_q, in Eq. (2) have only linear part.

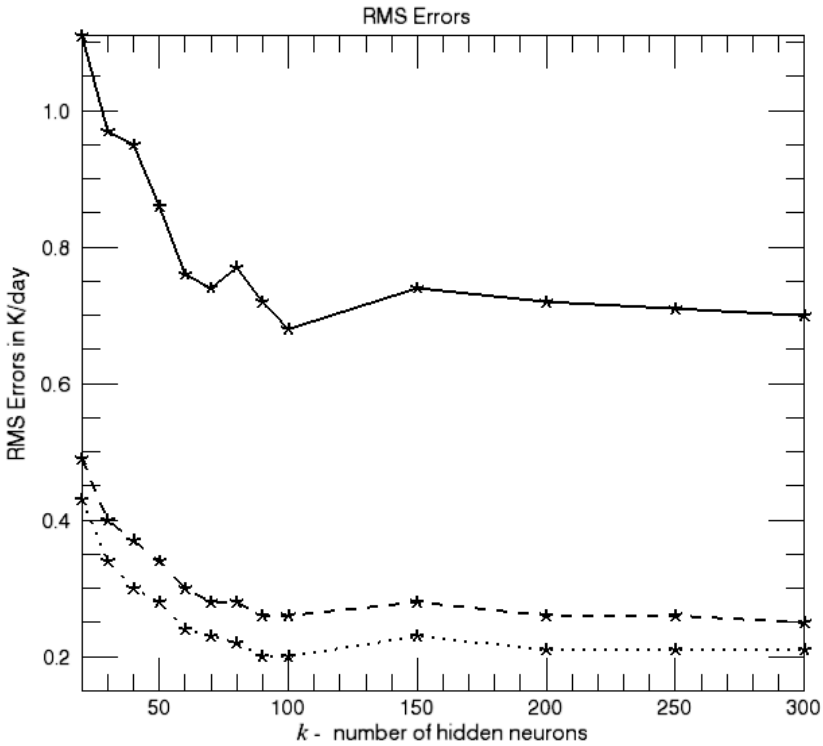

Fig. 3. The convergence of root mean square errors (23, 24, and 26). Solid line – $RMSE_m$ (24) for $m = 26$, dashed line – RMSE (23), and dotted line – PRMSE (26).

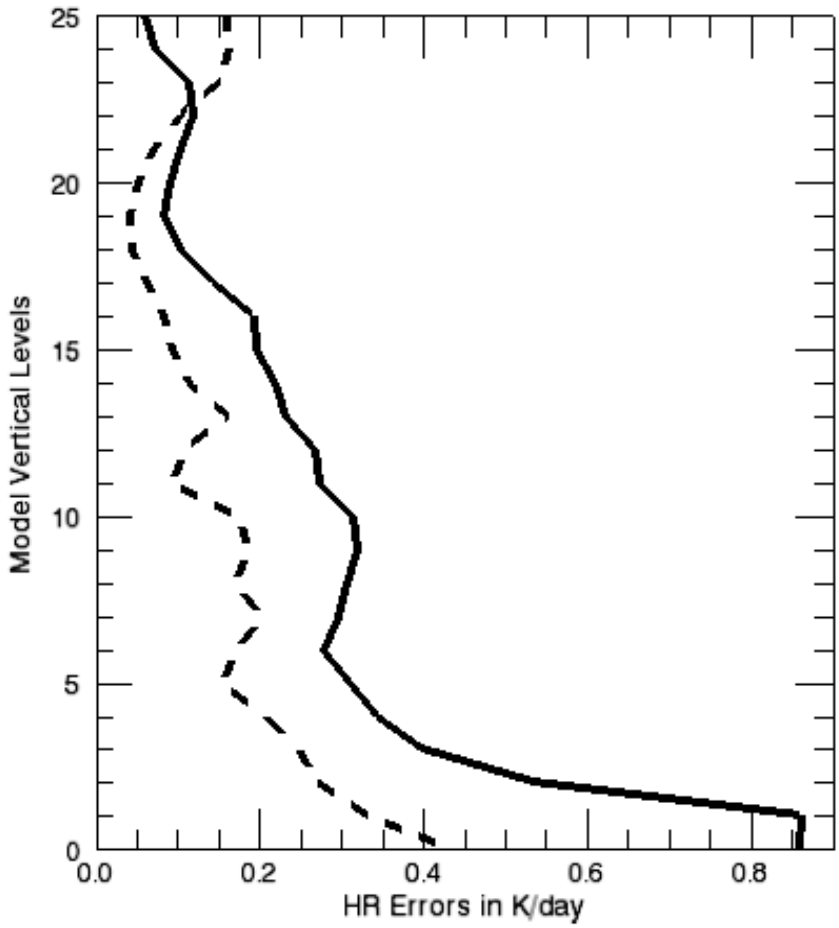

Fig. 4. The vertical error profiles (27), *rmsep(j),* for the "optimal" LWR NN emulation with 50 hidden neurons (NN50) – solid line and for the "optimal" SWR NN emulation with 55 hidden neurons (NN55) – dashed line.

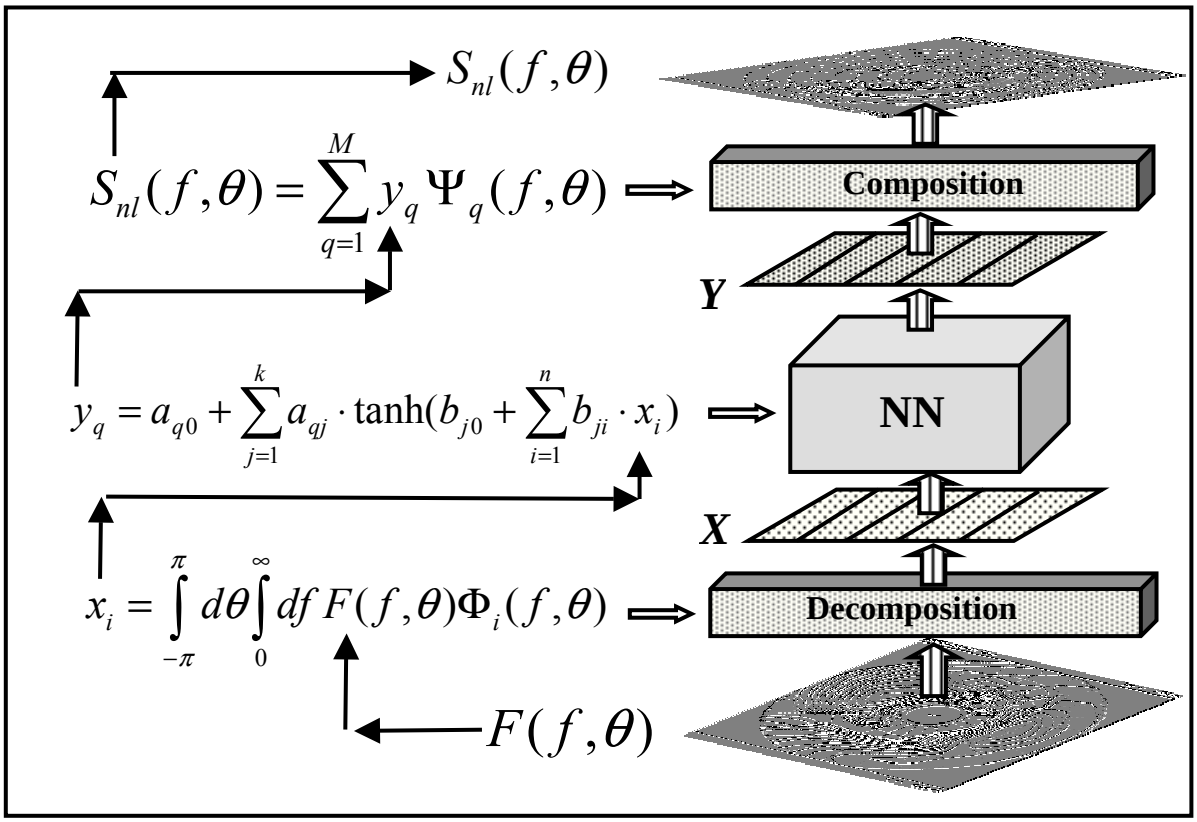

Fig. 5. Graphical representation of the NNIA algorithm.

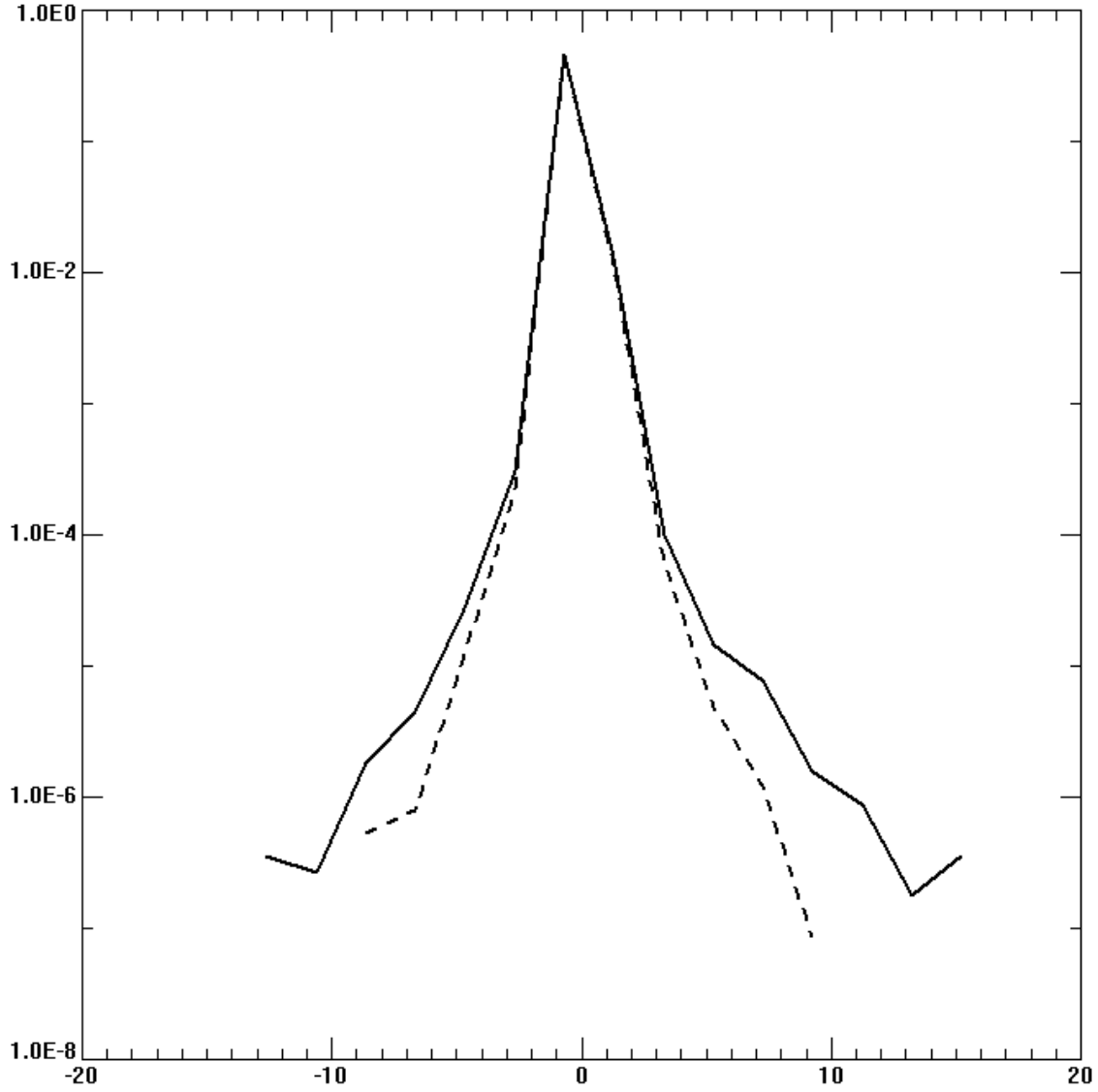

Fig. 6. Probability density distributions of emulation errors for the SWR NN emulation NN55 (solid line) and for the compound SWR parameterization. Both errors are calculated vs. the original SWR parameterization. Compound parameterization reduces the probability of medium and large errors by an order of magnitude. Horizontal axis is RMSE in K/day; vertical axis is logarithmic.

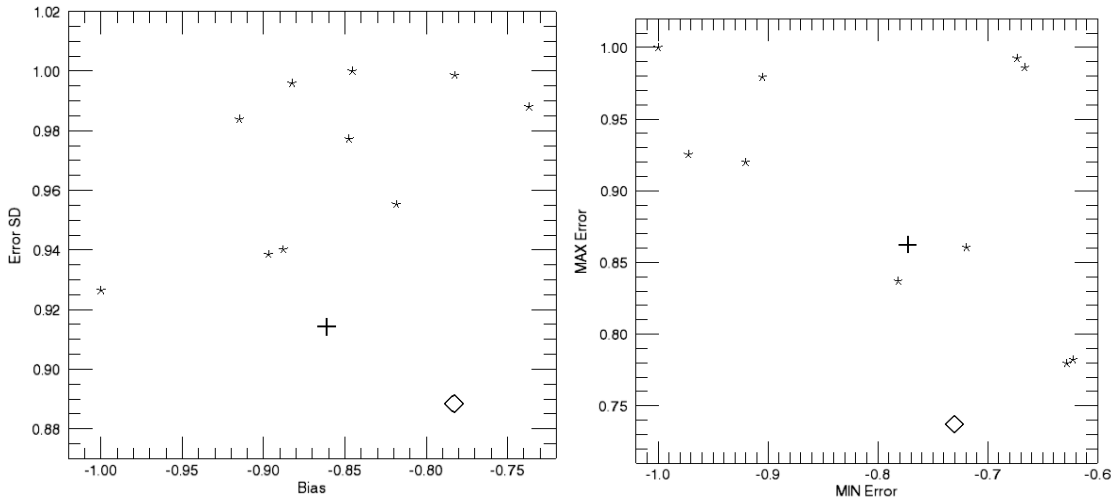

Fig. 7. The left panel: the random part of the emulation error (the standard deviation, SD, of the error) normalized to the maximum member error (the vertical axis) and the systematic error (bias) also normalized to the maximum member error (the horizontal axis); each ensemble member is represented by a star, the conservative ensemble average – by the cross, and the nonlinear ensemble using the averaging NN by the diamond at the figure. The right panel shows extreme outliers' statistics: 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 ensemble – by the diamond at the figure.

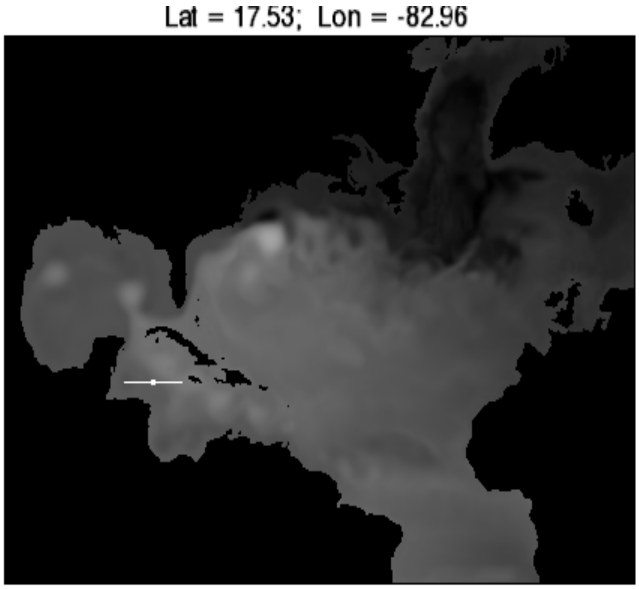

Fig. 8. The location of the cross section (white horizontal line) inside the model domain; white dot show the position of X_0. The picture is plotted in the internal model coordinates.

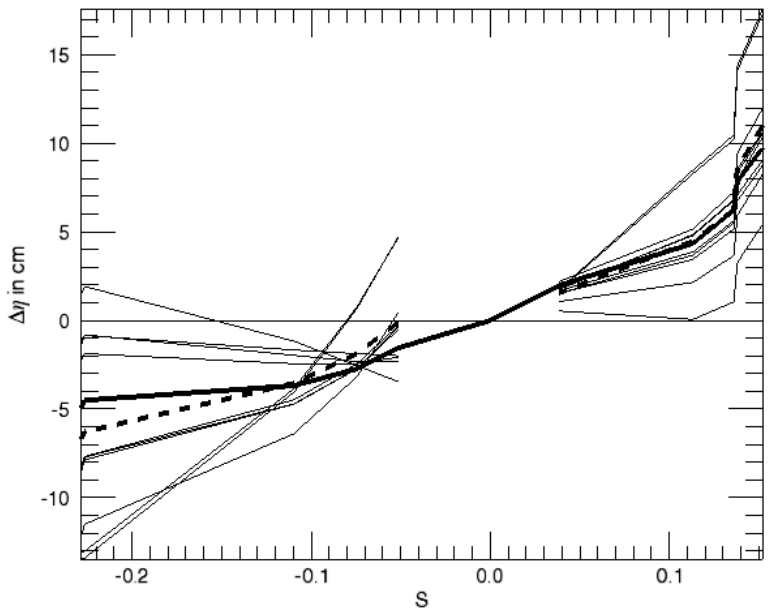

Fig. 9. $\Delta\eta$ calculated using (35) and the NN ensemble member Jacobians (an envelope of thin solid lines that illustrates the Jacobian uncertainties), exact $\Delta\eta$ calculated from the model (thick solid line), and $\Delta\eta$ calculated using the ensemble average Jacobian (36) (thick dashed line). $\Delta\eta$ is shown vs. the distance in the model state space, S (37).

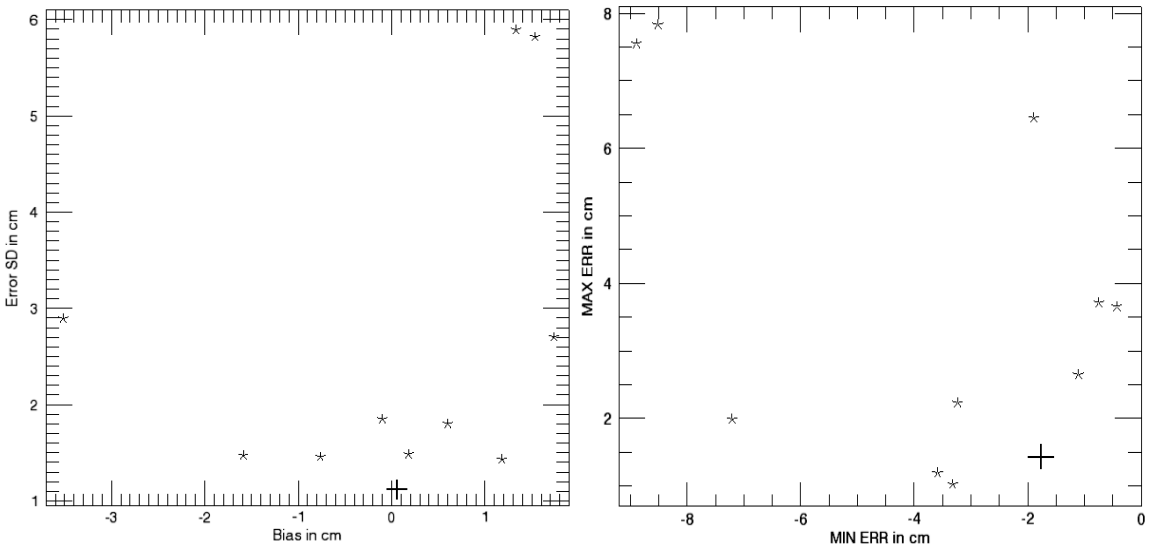

Fig. 10 The left panel: the systematic error (bias) and the random error (error standard deviation) for $\Delta\eta$ calculated along the path shown in the upper right panel using (35); the stars correspond to errors when the ensemble member Jacobians were used in (35), the cross corresponds to the case when the ensemble average Jacobian (36) was used. The right panel shows the minimum and maximum errors along the path; the stars correspond to errors when the ensemble member Jacobians were used in (35) and the cross to when the ensemble average Jacobian (36) was used in (35).