



# 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<sup>1,2</sup>

Received 19 April 2006; revised 23 January 2007; accepted 19 March 2007; published 25 September 2007.

[1] 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 application 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.

**Citation:** Krasnopolsky, V. M. (2007), Neural network emulations for complex multidimensional geophysical mappings: Applications of neural network techniques to atmospheric and oceanic satellite retrievals and numerical modeling, *Rev. Geophys.*, 45, RG3009, doi:10.1029/2006RG000200.

## 1. INTRODUCTION

[2] The neural network (NN) approach is a relatively new, diverse, and powerful statistical learning technique (also known as machine learning, learning from data, predictive learning, or the data-driven approach) that started developing rapidly in the mid-1980s after several major basic types of NNs were introduced in the works of *Kohonen* [1982], *Hopfield* [1982], *Rumelhart et al.* [1986], and *Lippmann* [1989]. In the 1990s this technique 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, and geophysical numerical modeling. Since then a significant number of NN applications have been developed in these fields; the most important of them are summarized in Table 1. 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 *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 the NN community by *Krasnopolsky and Schiller* [2003]. Applica-

<sup>1</sup>Science Applications International Corporation, Environmental Modeling Center, National Centers Environmental Prediction, NOAA, Camp Springs, Maryland, USA.

<sup>2</sup>Earth System Science Interdisciplinary Center, University of Maryland, College Park, Maryland, USA.

TABLE 1. Some NN Applications in Climate- and Weather-Related Fields<sup>a</sup>

NN Application	Studies
Satellite meteorology and oceanography	
Classification	<i>Bhattacharya and Solomatine</i> [2006]
Pattern recognition, feature extraction	<i>Bankert</i> [1994]
Change detection and feature tracking	<i>Valdés and Bonham-Carter</i> [2006]
Fast forward models for variational retrievals <sup>b</sup>	<i>Krasnopolsky</i> [1997]
Accurate transfer functions (retrieval algorithms) <sup>b</sup>	
Surface parameters	<i>Stogryn et al.</i> [1994], <i>Krasnopolsky and Schiller</i> [2003]
Atmospheric profiles	<i>Aires et al.</i> [2002], <i>Müller et al.</i> [2003]
Predictions	
Geophysical time series	<i>Elsner and Tsonis</i> [1992]
Regional and global climate	<i>Pasini et al.</i> [2006]
Time-dependent physical processes	<i>Wu et al.</i> [2006]
Hybrid climate and weather numerical models and data assimilation systems	
New hybrid parameterizations of physics	<i>Chevallier et al.</i> [1998]
Fast emulations of model physics <sup>b</sup>	<i>Krasnopolsky et al.</i> [2002, 2005a]
Fast forward models for direct assimilation <sup>b</sup>	<i>Krasnopolsky</i> [1997]
Forward models for propagating a signal to different vertical levels and variables <sup>b</sup>	<i>Krasnopolsky et al.</i> [2006]
Hybrid coupled models	<i>Tang and Hsieh</i> [2003]
Geophysical data fusion	<i>Loyola and Ruppert</i> [1998]
Geophysical data mining	<i>Brown and Mielke</i> [2000]
Interpolation and downscaling	<i>Dibike and Coulibaly</i> [2006]
Nonlinear multivariate statistical analysis	<i>Hsieh</i> [2004]
Hydrology	<i>Bhattacharya et al.</i> [2005]

<sup>a</sup>The list of applications included in Table 1 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 Studies column are representative papers which deal with that particular application. When we had sufficient information, we included the first (to the best of our knowledge) publication on that application.

<sup>b</sup>These applications are discussed in more detail in this paper and were developed with the author's participation.

tions of the NN technique for developing nonlinear generalizations of multivariate statistical analysis have been recently reviewed by *Hsieh* [2004]. *Solomatine* [2005] has reviewed hydrological NN applications.

[3] 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 utilizes 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. These are noted in Table 1.

[4] In the methodological section 2 we introduce the concept and major properties of complex nonlinear map-

pings. 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 sections 3–5. 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

[5] A mapping  $M$  between two vectors  $\mathbf{X}$  (input vector) and  $\mathbf{Y}$  (output vector) can be symbolically written as

$$\mathbf{Y} = M(\mathbf{X}); \quad \mathbf{X} \in \mathbb{R}^n, \mathbf{Y} \in \mathbb{R}^m. \quad (1)$$

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

[6] The 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  $\mathbf{X} = \{x_{t-k}, x_{t-k+1}, \dots, x_t\}$  is a lag vector created from  $k$  past values of the time series for the variable  $\mathbf{X}$  and the vector  $\mathbf{Y} = \{x_{t+1}, x_{t+2}, \dots, x_{t+p}\}$  contains  $p$  future, predicted values of the same variable  $\mathbf{X}$ . The components of vectors  $\mathbf{X}$  and  $\mathbf{Y}$  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]. The prediction of a time series as a mapping and the use of NNs in this application are discussed in great detail by Weigend and Gershenfeld [1994].

[7] 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  $\mathbf{X}$  of satellite measurements (calibrated or raw radiances, brightness temperature, backscatter coefficients, etc., at different frequencies) into the vector  $\mathbf{Y}$  of geophysical parameters like wind speeds, atmospheric moisture parameters, and ocean and land surface characteristics. Here the components of vector  $\mathbf{X}$  may again be correlated because the frequency bands may not be completely independent and may overlap. The components of the output vector  $\mathbf{Y}$  may be correlated because the corresponding geophysical parameters are physically related [Krasnopolsky et al., 1999, 2000a]. The target mapping in this example may also be a complicated nonlinear mapping. This application is discussed in section 3.

[8] 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 (LWR) (see section 4 for details) can be considered as a mapping, where the input vector  $\mathbf{X}$  is composed of several atmospheric state variables, like temperature, humidity, and ozone concentration, that are functions of height and some surface characteristics. The output vector  $\mathbf{Y}$  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  $\mathbf{X}$  and  $\mathbf{Y}$  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  $\mathbf{X}$  and  $\mathbf{Y}$  can be significantly correlated because (1) they are physically related and (2) they are related as the discretized values (elements of a profile) of the same continuous function 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) because of the impact of highly nonlinear atmospheric moisture processes. This application is discussed in section 4.

### 2.2. Some Generic Properties of Mappings

[9] Multidimensional, 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

[10] The first essential property of the target mapping is its mapping dimensionalities. A mapping is characterized by two dimensionalities: (1) dimensionality  $n$  of the input space  $\mathbb{R}^n$  and (2) dimensionality  $m$  of the output space  $\mathbb{R}^m$ . The second property of the mapping (1) is the mapping domain. If all components of the input vector  $\mathbf{X}$  are scaled to  $[-1, 1]$ , the volume of the input space  $\mathbb{R}^n$  is equal to  $2^n$  and therefore grows exponentially with  $n$ . Once 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, 2006]. Fortunately, the components of the input vector  $\mathbf{X}$  are usually interrelated or multicollinear [Aires et al., 2004b] because of the physical or statistical reasons, which 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  $\mathcal{R}^n$  spanned by the input vectors  $\mathbf{X}$  [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  $\mathbf{Y}$  are usually also interrelated. As a result, the output vectors also span only a fraction of the output space  $\mathcal{R}^m$ . 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

[11] 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  $\mathbf{X}$  to produce the output vector  $\mathbf{Y}$ , 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 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].

[12] Keeping in mind those three examples of the mapping (1) we introduced in section 2.1, 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 semiquantitative characteristics of physical complexity, the number of equations describing the physics, the type of these equations (e.g., linear versus nonlinear, ordinary differential equation versus partial differential equation versus integrodifferential 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 versus Lagrange formulations of the equations of geophysical fluid dynamics can be pointed out.

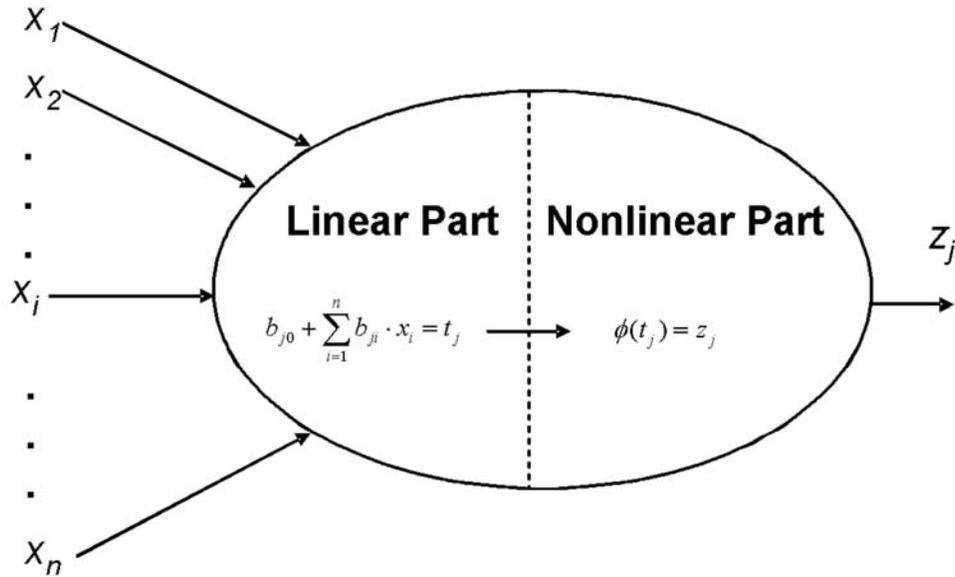
[13] The second type of complexity that can be introduced is mapping numerical or computational complexity. For this type of complexity a quantitative measure, like the number of elementary numerical operations required for

calculating  $\mathbf{Y}$  given  $\mathbf{X}$ , 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 versus 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).

[14] The third definition of mapping complexity is the functional complexity. It describes the complexity of the functional dependency of the outputs  $\mathbf{Y}$  versus inputs  $\mathbf{X}$  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  $n$ , and so on. 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 [Colliers, 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 section 2.4.3. It also provides us with a measure of the functional mapping complexity after the fact, after the accurate NN approximation is performed; however, as we will show in sections 3–5, 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 in section 2.2.2; however, they cannot be used as unambiguous measures of these complexities.

### 2.2.3. Mappings Corresponding to Ill-Posed Problems

[15] 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  $\mathbf{X}$  may cause large changes in  $\mathbf{Y}$ . 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



**Figure 1.** A generic neuron (equation (3)). In general, it consists of a linear and a nonlinear part. In our case the output neurons  $y_q$  in equation (2) have only a linear part.

restore a whole object from its low-dimensional projection (e.g., estimating the NN Jacobian considered in section 5.2). If  $\mathbf{X}$  contains even a low level of noise, the uncertainties in  $\mathbf{Y}$  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, 2006].

### 2.3. MLP NN: A Generic Tool for Modeling Nonlinear Mappings

#### 2.3.1. NNs in Terms of Approximation Theory

[16] 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

$$y_q = \text{NN}(\mathbf{X}, a, b) = a_{q0} + \sum_{j=1}^k a_{qj} z_j, \quad q = 1, 2, \dots, m, \quad (2)$$

$$z_j = \phi \left( b_{j0} + \sum_{i=1}^n b_{ji} x_i \right), \quad (3)$$

where  $x_i$  and  $y_q$  are components of the input and output vectors, respectively,  $a$  and  $b$  are fitting parameters or NN weights,  $\phi$  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  $z_j$  (equation (3)) in the expansion (2). The expansion (2) is a linear expansion (a linear combination of the basis function  $z_j$  (equation (3))), and the coefficients  $a_{qj}$  ( $q = 1, \dots, m$  and  $j = 1, \dots, k$ ) are linear coefficients in this expansion. It is essential

(see section 2.4.1) that the basis functions  $z_j$  (equation (3)) are nonlinear with respect to inputs  $x_i$  ( $i = 1, \dots, n$ ) and to the fitting parameters or coefficients  $b_{ji}$  ( $j = 1, \dots, k$ ). As a result of the nonlinear dependence of the basis functions on multiple fitting parameters  $b_{ji}$  the basis  $\{z_j\}_{j=1, \dots, k}$  turns into a very flexible set of nonorthogonal 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) and (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, 1995b]. 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 Pagès, 1997].

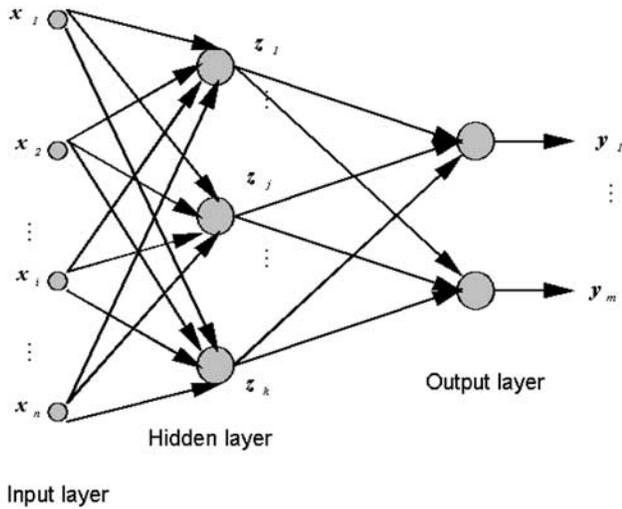
[17] The MLP NN ((2) and (3)) itself is a particular type of the mapping (1). In the case of the MLP NN the computational and functional complexities of the NN mapping are closely related and can be characterized by the number of fitting parameters  $a$  and  $b$  in (2) and (3) (they are especially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

$$N_c = k(n + m + 1) + m. \quad (4)$$

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

[18] A pictographic language reminiscent of a data flow-chart is used traditionally in the NN field, starting with the founding work by McCulloch and Pitts [1943]. In this work devoted to the mathematical modeling of a neuron a single



**Figure 2.** 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) multilayer perceptron (MLP) neural network (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  $y_q$ .

cell of a neural network, a basis function  $z_j$  (equation (3)), or neuron was represented by a figure like Figure 1. Then after *Rumelhart et al.* [1986] introduced the MLP NN, a pictographic representation of the entire NN was introduced (see Figure 2). The neurons are situated into layers inside the MLP NN. The connections (arrows) in Figure 2 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 equations (2) and (3) and Figures 1 and 2. However, in general, the pictographic language (Figures 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 equations (2) and (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.

### 2.3.3. Training Set

[19] 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  $\mathbf{X}$  and  $\mathbf{Y}$ ,

$$C_T = \{\mathbf{X}_p, \mathbf{Y}_p\}_{p=1, \dots, N}, \quad (5)$$

where  $\mathbf{Y}_p = M(\mathbf{X}_p) + \xi_p$ ,  $\xi_p$  is an error of observation or calculation with the probability density function  $\rho(\xi)$ , and  $\mathbf{X}_p \in D$  and  $\mathbf{Y}_p \in R$ . 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*].

[20] 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 approximate the target mapping with the desired accuracy. 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 oversampled 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 in section 2.2.1, the interrelations between inputs simplify the sampling task for cases of high input dimensionality, reducing the size and the effective dimensionality of the domain.

[21] 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).

[22] 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  $\xi$ . 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.

[23] 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 (equation (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

[24] To approximate a particular target mapping (1) with the MLP NN (2) and (3), we should first select the NN architecture or topology, the number of the inputs  $n$ , the number of 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  $\mathbf{X}$  and  $\mathbf{Y}$ ). 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 equations (2) and (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  $m$ .

[25] 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) and (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  $N_c$  (equation (4)) and provides significantly higher performance at the same approximation accuracy than a battery of  $m$  single-output NNs [Krasnopolsky and Fox-Rabinovitz, 2006b].

[26] 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 versus 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].

[27] 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 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

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

$$L(a, b) = \sum_{i=1}^N \ln \rho(\mathbf{Y}_i - \text{NN}(\mathbf{X}_i, a, b)) \quad (6)$$

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

$$E(a, b) = \sum_{i=1}^N (\mathbf{Y}_i - \text{NN}(\mathbf{X}_i, a, b))^2. \quad (7)$$

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 squares error or loss function (7). However, in the majority of applications the least squares error function (7) is applied.

[29] 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].

[30] 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

$$\Delta W = -\eta \left( \frac{\partial E}{\partial W} \right), \quad (8)$$

where  $\eta$  is a so-called learning constant and  $W$  is either one of two weights ( $a$  or  $b$ ). Using (7), (2), and (3), the derivative in (8) can be expressed analytically through the derivative of the activation function  $\phi$  and through the weight values at the previous iteration step. At the first step where 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*].

[31] 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

[32] 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 of nonlinear models, nonlinear approximation techniques, and nonlinear statistical approaches [*Cheng and Titterton, 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

[33] 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.

[34] 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  $E \leq (\alpha/k^p)$ , where  $\alpha > 0$ ,  $p > 0$ , and  $p$  is independent of  $n$ . In contrast, for approximations using fixed basis functions the approximation error is  $E \leq (\alpha/k^{1/n})$  for the same class of mapping [*Barron, 1993; Cheng and Titterton, 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.

[35] The basis functions  $z_j$  are very flexible, nonorthogonal, and overlapping. On the other hand, these useful properties may easily lead to nonoptimality 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 Multicollinearity of Inputs and Outputs

[36] 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 multicollinearities 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). Multicollinearities in input and output data lead to an equalization of local minima, especially in the case of a higher input dimensionality. Therefore multicollinearities 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, Extrapolation, and NN Emulations

[37] 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 and, 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.

[38] 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 (equation (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 in section 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.

[39] In this paper we will call such an improved 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 (equation (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

[40] 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 equations (2) and (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 emulat-

ing NNs that have close approximation and interpolation errors but different Jacobians.

[41] As mentioned 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 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: (1) 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). (2) An ensemble approach can be applied that uses an ensemble of NN emulations with the same architecture corresponding to different local minima of the error function or uses an ensemble of NN emulations with different numbers of hidden neurons (different complexities) to stabilize the NN Jacobian or to reduce the uncertainties of the NN Jacobian [Krasnopolsky, 2007] (see also section 5.2.2). (3) The mean Jacobian can be calculated over the data set and used [Chevallier and Mahfouf, 2001]. (4) Regularization techniques, like “weight smoothing” [Aires et al., 1999] or the technique based on a principle component decomposition [Aires et al., 2004b], can be used to stabilize the Jacobians. Or (5) 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 norm. Actually, Hornik et al. [1990] showed that the function of Sobolev 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].

[42] 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

[43] 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).

[44] 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 1950s and early to middle 1960s [Selfridge, 1958; Nilsson, 1965]. Since the early 1990s, many different algorithms based on similar ideas have been developed for NN ensembles [Hansen and Salamon, 1990; Sharkey, 1996; Naftaly et al., 1997; Opitz and Maclin, 1999; Hsieh, 2001].

[45] 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 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 subdomains of the training domain, by training different members using NNs with different architectures (different numbers of hidden neurons) [Hashem, 1997], or by training different members using NNs with the same architecture but different initial conditions for the NN weights [Maclin and Shavlik, 1995; Hsieh, 2001].

[46] 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 [2007] also applied the NN ensemble technique to reduce the uncertainty of the NN Jacobian (see section 5.2.2). Most of the previous work with NN ensembles has 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

[47] 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

[48] In section 2 we discussed general properties of the multidimensional complex mappings (1) and MLP NN (2) and (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

[49] 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 parameters 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.

[50] 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.

[51] 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], and to estimate atmospheric humidity profiles [Cabrera-Mercader and Staelin, 1995], atmospheric temperature, moisture, ozone profiles [Aires *et al.*, 2002], and atmospheric ozone profiles [Müller *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 multiparameter retrieval algorithm [Krasnopolsky *et al.*, 1999, 2000a; Meng *et al.*, 2007] 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 data.

### 3.1. Deriving Geophysical Parameters From Satellite Measurements: Standard Retrievals and Variational Retrievals Obtained Through Direct Assimilation

[52] Satellite sensors generate measurements like radiances, backscatter coefficients, and brightness temperatures. 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; that is, these approaches use some type of data fusion to regularize the solution of the inverse problem.

#### 3.1.1. Standard or Conventional Retrievals

[53] 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  $\mathbf{G}$  (e.g., surface wind speed over the ocean, atmospheric moisture concentration, sea surface temperature (SST), etc.) to a satellite measurement  $\mathbf{S}$  (e.g., brightness temperatures, radiances, reflection coefficients, etc.),

$$\mathbf{G} = f(\mathbf{S}), \quad (9)$$

where both  $\mathbf{G}$  and  $\mathbf{S}$  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  $\mathbf{G}$  can sometimes correspond to a single  $\mathbf{S}$ . Forward models

$$\mathbf{S} = F(\mathbf{G}), \quad (10)$$

where  $F$  is a forward model (FM), which relates a vector  $\mathbf{G}$  to a vector  $\mathbf{S}$ , 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  $\mathbf{S}$  and  $\mathbf{G}$ . Even in the cases where the mapping (9) is not unique, this multivalued 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  $\mathbf{G}^0$  which minimizes the functional [Stoffelen and Anderson, 1997]

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

where  $\mathbf{S}^0$  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 and multiple solutions. This approach does not determine the TF explicitly; it assumes this function implicitly, and for each new measurement  $\mathbf{S}^0$  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  $\mathbf{G}^0$ . Then the difference vector  $\Delta\mathbf{S}$  is small, and there is a vector  $\mathbf{G}$  in close proximity to  $\mathbf{G}^0$  ( $|\Delta\mathbf{G}| = |\mathbf{G} - \mathbf{G}^0|$  is small) where  $\Delta\mathbf{S}(\mathbf{G}) = 0$ . Expanding  $F(\mathbf{G})$  in a Taylor series and keeping only those terms which are linear with respect to  $\Delta\mathbf{G}$ , we can obtain a system of linear equations to calculate the components of the vector  $\Delta\mathbf{G}$  [e.g., *Wentz, 1997*],

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

where  $n$  is the dimension of vector  $\mathbf{G}$ . After  $\Delta\mathbf{G}$  is calculated, the next iteration of (12) with  $\mathbf{G}^0 = \mathbf{G}^0 + \Delta\mathbf{G}$  is performed. The process is expected to converge quickly to the vector of retrievals  $\mathbf{G}$ . Again, in this case the TF  $f$  (equation (9)) is not determined explicitly but is only determined implicitly for the vector  $\mathbf{S}^0$  by the solution of (12). This type of retrieval can be called a “local” or “localized” linear inversion. These techniques (11) and (12) are usually called physically based retrievals. It is important to emphasize that the physically based algorithms (11) and (12) are by definition multiparameter algorithms since they retrieve several geophysical parameters simultaneously (a complete vector  $\mathbf{G}$ ).

[54] 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  $f_{\text{mod}}$  is usually chosen (usually some kind of a regression) which contains a vector of empirical (or model) parameters  $a = \{a_1, a_2, \dots\}$ ,

$$\mathbf{G}_k = f_{\text{mod}}(\mathbf{S}, a), \quad (13)$$

where these parameters are determined from an empirical (or simulated) matchup data set  $\{\mathbf{G}_k, \mathbf{S}\}$  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  $\mathbf{G}_k$  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 multiparameter retrievals) systematic (bias) and random (unaccounted variance) errors in a single retrieved parameter  $\mathbf{G}_k$ .

[55] The obvious way to improve single-parameter retrievals (13) is to include other parameters in the retrieval process using an empirical multiparameter approach, which as in the physically based multiparameter approach (11) and (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  $\mathbf{S}$ ,

$$\mathbf{G} = f_{\text{mod}}(\mathbf{S}), \quad (14)$$

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

[56] Geophysical parameters derived using standard retrievals can be used for many applications, such as the NWP DASSs. In this case a contribution to the variational analysis cost function  $\chi_{\mathbf{G}}$  from a particular retrieval  $\mathbf{G}^0$  is

$$\chi_{\mathbf{G}} = \frac{1}{2} (\mathbf{G} - \mathbf{G}^0)^T (O + E)^{-1} (\mathbf{G} - \mathbf{G}^0), \quad (15)$$

where  $\mathbf{G}^0 = f(\mathbf{S}^0)$  is a vector of the retrieved geophysical parameter,  $\mathbf{S}^0$  is a vector of the sensor measurements,  $\mathbf{G}$  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

[57] 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*].

[58] 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  $\chi_{\mathbf{S}}$  from a particular sensor measurement  $\mathbf{S}^0$  is

$$\chi_{\mathbf{S}} = \frac{1}{2} (\mathbf{S} - \mathbf{S}^0)^T (O + E)^{-1} (\mathbf{S} - \mathbf{S}^0), \quad (16)$$

where  $\mathbf{S} = F(\mathbf{G})$ ,  $F$  is a FM (equation (10)) which relates an analysis state vector  $\mathbf{G}$  (or a vector of geophysical

parameters in the analysis) to a vector of simulated sensor measurements  $\mathbf{S}$ ,  $O$  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

[59] FMs are usually complex because of 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.

[60] 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.).

[61] To develop a NN emulation for the FM, a training set, which consists of matched pairs of vectors of geophysical parameters and satellite measurements  $\{\mathbf{G}, \mathbf{S}\}_{i=1, \dots, N}$ , 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

[62] 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  $\{\mathbf{G}, \mathbf{S}\}_{i=1, \dots, N}$  is required. As in the case of FMs, simulated or empirical data can be used to create the training set.

[63] 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, 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 from satellite measurements, Aires *et al.* [2002] and Müller *et al.* [2003] 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

[64] 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 subdomain  $D_T \subseteq D$ ) in the full domain  $D$ . Real data to be fed into the NN  $f_{NN}$ , which emulates a TF (equation (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.

[65] Let the model  $\mathbf{S} = F(\mathbf{G})$  have an inverse  $\mathbf{G} = f(\mathbf{S})$ ; then by definition,  $\mathbf{S} = F(f(\mathbf{S}))$ . Further, let  $f_{NN}$  be the NN emulating the inverse model in the domain  $D_T$ . The result of  $\mathbf{G}_0 = f_{NN}(\mathbf{S}_0)$  for  $\mathbf{S}_0 \notin D_T$  may be arbitrary, and in general,  $F(f_{NN}(\mathbf{S}_0))$  will not be equal to  $\mathbf{S}_0$ . The validity of  $\mathbf{S} = F(f_{NN}(\mathbf{S}))$  is a necessary condition for  $\mathbf{S} \in D$ . Now, if in the application stage of the NN  $f_{NN}$ ,  $\mathbf{S}$  is not in the domain  $D_T$ , the NN  $f_{NN}$  is forced to extrapolate. In such a situation the validity condition may not be fulfilled, and the resulting  $\mathbf{G}$  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,  $\mathbf{S} = F_{NN}(\mathbf{G})$ . 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

$$\delta = \|\mathbf{S} - F_{NN}(f_{NN}(\mathbf{S}))\|. \quad (17)$$

The solution to the problem of a scope check is obtained by estimating  $\delta$  (equation (17)), where  $\mathbf{S}$  is the result of the satellite measurement. This procedure (1) detects situations where the forward model or/and transfer function is

**TABLE 2. Comparison of Physically Based Radiative Transfer and Empirical NN Forward Models Under Clear and Clear Plus Cloudy Weather Conditions<sup>a</sup>**

Study	Type	BT RMS Error, °K	
		Vertical	Horizontal
<i>Petty and Katsaros</i> [1992]	PB	1.9 (2.3)	3.3 (4.3)
<i>Wentz</i> [1997]	PB	2.3 (2.8)	3.4 (5.1)
<i>Krasnopolsky</i> [1997]	NN, empirical	1.5 (1.7)	3.0 (3.4)

<sup>a</sup>The numbers in parentheses are the measurements from clear plus cloudy weather conditions. PB, physically based.

inappropriate, (2) does an “in-scope” check for the retrieved parameters even if the domain has a complicated geometry, and (3) 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

[66] In sections 3.1–3.4 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 3500 matchups) and testing (about 3500 matchups) our forward model and retrieval algorithm. Then, more than 6000 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

[67] The empirical SSM/I FM represents the relationship between the vector of geophysical parameters  $\mathbf{G}$  and vector

of satellite brightness temperatures (BTs)  $\mathbf{S}$ , where  $\mathbf{S} = \{\text{T19V, T19H, T22V, T37V, T37H}\}$  (TXXY means XX frequency in GHz and Y polarization) and  $\mathbf{G} = \{W, V, L, \text{SST}\}$ . Four geophysical parameters are included in  $\mathbf{G}$  (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] and *Wentz* [1997] (see Table 2). The NN emulation [*Krasnopolsky*, 1997], which implements this SSM/I FM, has 4 inputs  $\{W, V, L, \text{SST}\}$ , 1 hidden layer with 12 neurons, and 5 nonlinear BT outputs  $\{\text{T19V, T19H, T22V, T37V, T37H}\}$ .

[68] After NN FM was trained, it was validated and compared with physically based forward models by *Petty and Katsaros* [1992, 1994] and *Wentz* [1997]. The RMS errors for NN FM are systematically better than those for the more sophisticated physically based *Petty and Katsaros* and *Wentz* FMs for all weather conditions and all channels considered (see Table 2).

[69] The NN FM is not as general as a radiative transfer model; it was developed to be applied in 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*, 2007] 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

[70] 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 10 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]).

[71] *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 as GSW. *Goodberlet and Swift* [1992] (GS in

**TABLE 3. Error Budget for Different SSM/I Wind Speed Algorithms Under Clear and Clear Plus Cloudy Conditions<sup>a</sup>**

Algorithm	Method	Bias, m/s	Total RMSE, m/s	RMSE for $W > 15$ m/s, m/s	Source
GSW	multiple linear regression	-0.2 (-0.5)	1.8 (2.1)	(2.7)	<i>Goodberlet et al.</i> [1989]
GSWP	generalized linear regression	-0.1 (-0.3)	1.7 (1.9)	(2.6)	<i>Petty</i> [1993]
GS	nonlinear regression	0.5 (0.7)	1.8 (2.5)	(2.7)	<i>Goodberlet and Swift</i> [1992]
Wentz	physically based	0.1 (-0.1)	1.7 (2.1)	(2.6)	<i>Wentz</i> [1997]
NN1	neural network	-0.1 (-0.2)	1.5 (1.7)	(2.3)	<i>Krasnopolsky et al.</i> [1996, 1999]
NN2	neural network	(-0.3)	(1.5)	...	<i>Meng et al.</i> [2007]

<sup>a</sup>The numbers in parentheses are the measurements from clear plus cloudy weather conditions.

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 (GSWP) algorithm introduced by *Petty* [1993] is based on a generalized linear regression.

[72] 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 conditions, and the other performs retrievals 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 was higher in areas where there were significant weather events (higher levels of atmospheric moisture). However, because this algorithm used the data set that did not contain high wind speed events for the training, it cannot generate acceptable wind speeds at ranges higher than 18–19 m/s.

[73] The next generation NN algorithm, a multiparameter NN algorithm developed at National Centers for Environmental Prediction (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 in section 3.5 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, and 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  $\mathbf{G} = \{W, V, L, SST\}$ . 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.

[74] *Meng et al.* [2007] (NN2 in Table 3) use the NN multiparameter retrieval approach developed by *Krasnopolsky et al.* [1996, 1999] to design another NN multiparameter retrieval algorithm for SSM/I. They use all seven SSM/I BTs as inputs. Their output vector also has four components  $\mathbf{G} = \{W, T_a, H, SST\}$ , 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/s. Moreover, there were only a few higher speed events with  $W > 15$ –17 m/s.

[75] Table 3 shows a comparison of the performance of all the aforementioned empirical algorithms in terms of the accuracy of the surface wind speed retrievals. It also shows statistics for a physically based algorithm developed by *Wentz* [1997], which is based on a linearized numerical inversion (equation (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, root-mean-square error (RMSE), and high wind speed performance.

[76] *Krasnopolsky et al.* [1999, 2000a] have shown that the errors of multiparameter 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 RMSE 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 multiparameter NN algorithm NN2 [*Meng et al.*, 2007] the choice of the additional outputs surface air temperature  $T_a$  and humidity  $H$ , which 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 RMSE 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, because of the nonlinear nature of the NN emulation and the proper choice of output parameters the multiparameter 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

[77] Throughout section 3 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 not only can produce accurate retrievals but also performs an analysis and quality control of the retrievals and environmental conditions, rejecting any poor retrieval that occurs.

[78] 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

[79] 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 interdisciplinary complex geophysical/environmental numerical models has increased tremendously during the last few decades because of improvements in their quality via developments in numerical modeling and computing capabilities.

[80] Traditional complex environmental numerical models (ENM) are deterministic models based on “first principle” equations. For example, general circulation models (GCM),

also known as global climate models, are numerical atmospheric and oceanic models for climate simulation and weather prediction that are based on solving time-dependent three-dimensional (3-D) geophysical fluid dynamics equations on a sphere. The governing equations of these models can be written symbolically as

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

where  $\psi$  is a 3-D prognostic or dependent variable or set of variables (e.g., temperature, wind, pressure, and 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 (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 chemistry (the constituency transport, chemical reactions, etc.). 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).

[81] 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

$$\frac{DF}{Dt} = S_{in} + S_{nl} + S_{ds} + S_{sw}, \quad (19)$$

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.

[82] 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 (also known as parameterizations). These parameterizations constitute the right-hand side forcing for the dynamics equations (18) and (19). Some of these parameterizations are still the most time-consuming components of ENMs (see examples in section 4.1). 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.

[83] 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

[84] 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 are 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.

[85] 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 min, some computationally expensive parts of the model physics (e.g., short-wave radiation (SWR)) 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.

[86] In the wind wave model (19) the calculation of the source term  $S_{nl}$  requires roughly  $10^3$ – $10^4$  times more computational effort than all other aspects of the wave model 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.

[87] 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; *Chevallier et al.*, 1998] and for emulations of model physics parameterizations in ocean and atmospheric numerical models [*Krasnopolsky et al.*, 2000b, 2002, 2005a]. In these works the calculation of model physics components has been accelerated by  $10$ – $10^5$  times as compared to the time needed for calculating the corresponding original parameterizations of the model physics.

[88] Approaches formulated by *Chevallier et al.* [1998, 2000] and *Krasnopolsky et al.* [2000b, 2002, 2005a, 2005c] 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 et al.*, 1998, 2000] and a HEM or HGCM [*Krasnopolsky et al.*, 2000b, 2002, 2005a; *Krasnopolsky and Fox-Rabinovitz*, 2006a, 2006b]. These two concepts have been debated by *Chevallier* [2005] and *Krasnopolsky et al.* [2005c] and are discussed in sections 4.2 and 4.3. Another type of hybrid model (hybrid coupled model, where a simplified atmosphere is described by a neural network model and the ocean is described by a dynamical model) was introduced and described by *Tang and Hsieh* [2003] and *Li et al.* [2005].

#### 4.2. Hybrid Parameterizations of Physics

[89] *Chevallier et al.* [1998, 2000] considered a component of the complex GCM (the European Centre for Medium-Range Weather Forecasts (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, 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),

$$\mathbf{Y} = M(\mathbf{X}); \quad (20)$$

in this particular case the input vector  $\mathbf{X} = (\mathbf{S}, \mathbf{T}, \mathbf{V}, C)$ , where the vector  $\mathbf{S}$  represents surface variables,  $\mathbf{T}$  is a vector (profile) of atmospheric temperatures,  $C$  is a profile of cloud variables, and the vector  $\mathbf{V}$  includes all other variables (humidity profile, different gas mixing ratio profiles, etc.). The output of the LWR parameterization, vector  $\mathbf{Y}$ , is composed of two vectors  $\mathbf{Q}$  and  $\mathbf{f}$ ,  $\mathbf{Y} = (\mathbf{Q}, \mathbf{f})$ . Here  $\mathbf{Q}$  is a profile of cooling rates  $\mathbf{Q} = (C_r^1, C_r^2, \dots, C_r^L)$ , where  $C_r^j$  is the cooling rate at the  $j$ th vertical model level and  $\mathbf{f}$  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 separating cloud variables  $C$ . In this parameterization, level fluxes are calculated as

$$\mathbf{F}(\mathbf{S}, \mathbf{T}, \mathbf{V}, C) = \sum_i \alpha_i(C) \mathbf{F}_i(\mathbf{S}, \mathbf{T}, \mathbf{V}), \quad (21)$$

where  $i$  is an index for the vertical level, each partial or individual flux  $\mathbf{F}_i(\mathbf{S}, \mathbf{T}, \mathbf{V})$  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  $\mathbf{F}_i(\mathbf{S}, \mathbf{T}, \mathbf{V})$ . Thus the flux at each level (equation (21)) is a linear combination of approximating NNs and cloud physics coefficients  $\alpha_i(C)$ . As a 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  $C_r$ , linear combinations of the individual approximating NNs  $\mathbf{F}$  (equation (21)) are differentiated at each vertical level,

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

where  $P$  is atmospheric pressure.

[90] NeuroFlux has a very good accuracy; its bias is about 0.05 K/d, and RMS error is about 0.1 K/d compared to the LWR parameterization by *Washington and Williamson* [1977]. It is 8 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-D variational data assimilation system.

[91] 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 equation (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 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

[92] A new concept of a complex HEM has been formulated and developed by *Krasnopolsky et al.* [2000b, 2002, 2005a] and *Krasnopolsky and Fox-Rabinovitz* [2006a, 2006b]. 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) and (20) (see sections 2 and 4.2).

[93] Here we use the NCAR CAM (see the special issue on the National Center for Atmospheric Research Community Climate Model in *Journal of Climate*, 11(6), 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, respectively.

[94] *Krasnopolsky and Fox-Rabinovitz* [2006a, 2006b] 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.

[95] 1. The first step is 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 (3) in one hidden layer of (2) and (3) (see sections 2.2.2, 2.3.1, and 2.3.4).

[96] 2. The second step is 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, because of 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, and poor representation of extreme events).

[97] 3. The third step is 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).

[98] 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 are 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

[99] The NCAR CAM and NASA Natural Seasonal-to-Interannual Predictability Program (NSIPP) GCM are used in this section as examples of GCMs. The NCAR CAM is a spectral model that has 42 spectral components (or approximately  $3^\circ$ – $3.5^\circ$  horizontal resolution) and 26 vertical levels. The NSIPP model is a grid point GCM that has  $2^\circ \times 2.5^\circ$  (latitude  $\times$  longitude) horizontal resolution and 40 vertical levels. NN emulations were developed for the two most time-consuming components of model physics, the LWR and 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 is presented by Chou *et al.* [2001]. The full model radiation (or total LWR and SWR) calculations take  $\sim 70\%$  of the total model physics calculations.

[100] 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:

$$B = \frac{1}{NL} \sum_{i=1}^N \sum_{j=1}^L [\mathbf{Y}(i,j) - \mathbf{Y}_{\text{NN}}(i,j)]$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^L [\mathbf{Y}(i,j) - \mathbf{Y}_{\text{NN}}(i,j)]^2}{NL}}, \quad (23)$$

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

[101] These two error characteristics (equation (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 equation (23), the bias and RMSE for the  $m$ th vertical level of the model can be calculated as

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

$$\text{RMSE}_m = \sqrt{\frac{\sum_{i=1}^N [\mathbf{Y}(i, m) - \mathbf{Y}_{\text{NN}}(i, m)]^2}{N}}. \quad (24)$$

[102] The root-mean-square error can also be calculated for each  $i$ th profile,

$$\text{PRMSE}(i) = \sqrt{\frac{1}{L} \sum_{j=1}^L [\mathbf{Y}(i,j) - \mathbf{Y}_{\text{NN}}(i,j)]^2}. \quad (25)$$

[103] 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  $\sigma_{\text{PRMSE}}$ , which are location-independent,

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

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

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 RMSEP can be calculated as

$$\text{RMSEP}(j) = \sqrt{\frac{1}{N} \sum_{i=1}^N [\mathbf{Y}(i,j) - \mathbf{Y}_{\text{NN}}(i,j)]^2}. \quad (27)$$

#### 4.4.2. Emulating the Model Radiation

[104] 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 (equation (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.

[105] The method for calculating LWR in the NCAR CAM is based on long-wave radiative transfer equations in an absorptivity/emissivity formulation [see *Collins*, 2001, and references therein],

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

where  $F^\uparrow(p)$  and  $F^\downarrow(p)$  are the upward and the downward heat fluxes,  $B(p) = \sigma \times T^4(p)$  is the Stefan-Boltzmann relation, pressures  $p_s$  and  $p_t$  refer to the top and surface atmospheric pressures, and  $\alpha$  and  $\varepsilon$  are the atmospheric absorptivity and emissivity. To solve the integral equation (28), the absorptivity and emissivity have to be calculated by solving the following integrodifferential equations:

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

where the integration is over wave number  $\nu$  and  $B(p_t)$  is the Planck function. To solve equation (29) for the absorptivity and emissivity, additional calculations have to be performed, and the atmospheric transmission  $\tau_\nu$  has to be calculated. This calculation involves a time-consuming integration over the entire spectral range of gas absorption.

Equations (28) and (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.

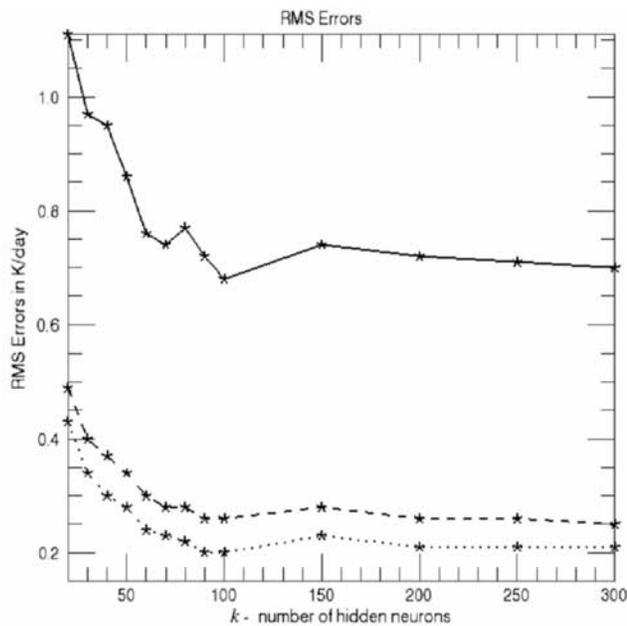
[106] The input vectors for the NCAR CAM LWR parameterization include 10 vertical profiles (atmospheric temperature, humidity, ozone, CO<sub>2</sub>, N<sub>2</sub>O, CH<sub>4</sub>, two CFC mixing ratios (the annual mean atmospheric mole fractions for halocarbons), pressure, and cloudiness) and one relevant 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 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.

[107] NCAR CAM was run for 2 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 part was used 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.

[108] Several NNs have been developed that all have one hidden layer with 20–300 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 multidecadal climate model integration.

[109] All NN emulations [*Krasnopolsky et al.*, 2005a; *Krasnopolsky and Fox-Rabinovitz*, 2006a, 2006b] developed for the NCAR CAM LWR and SWR have almost zero or negligible systematic errors (biases). Figure 3 illustrates convergences of root-mean-square errors ((23), (24), and (26)) that are random errors in the case of negligible biases. Figure 3 shows that an error convergence has been reached when the number of hidden neurons  $k \approx 100$ . However, the convergence becomes slow and nonmonotonic 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.

[110] 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



**Figure 3.** Convergence of root-mean-square errors ((23), (24), and (26)). The solid line represents  $RMSE_m$  (equation (24)) for  $m = 26$ , the dashed line represents  $RMSE$  (equation (23)), and the dotted line represents  $PRMSE$  (equation (26)).

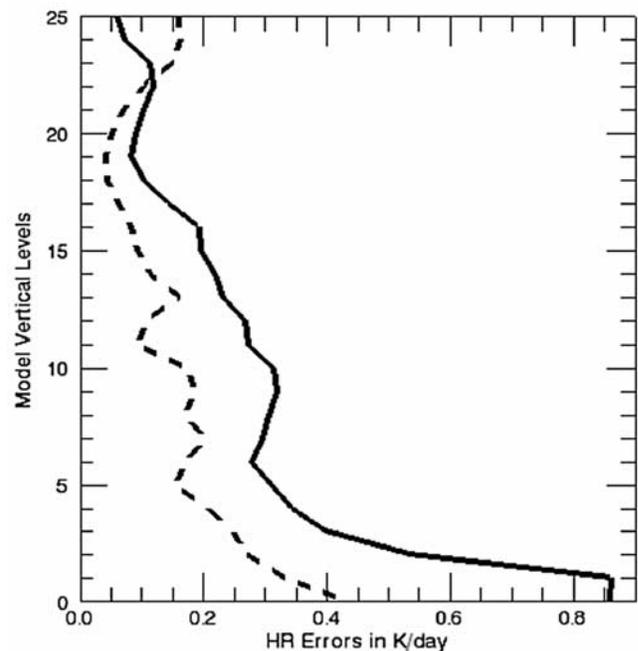
(in the framework of the NN application) that the accuracy of the NN emulation is sufficient for this application. Figure 4 shows the vertical error profile  $RMSEP(j)$  (equation (27)) for the “optimal” NN emulation with 50 hidden neurons, NN50 (solid line). It shows that the errors are very small; at the top 10 levels the error does not exceed 0.2 K/d, at the top 20 levels it does not exceed 0.3 K/d, and it reaches just about 0.6–0.8 K/d 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.

[111] 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., 2005a; Krasnopolsky and Fox-Rabinovitz, 2006a, 2006b]. Comparisons between the control and NN emulation runs are presented in Table 4 (rows 1–3). They are done by analyzing the time (40 years) 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, 2006b]. 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.

[112] 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, 2006b] are very similar to the ones presented above for NCAR CAM. This illustrates the robustness of the NN emulation approach.

[113] The second component of atmospheric radiation is short-wave radiation. 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



**Figure 4.** Vertical error profiles (equation (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).

**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 and SWR Parameterizations (in GCM), Their NN Emulations (in HGCM), and Their Differences<sup>a</sup>**

GCM Versions	Fields				
	Mean Sea Level Pressure, hPa	Surface Temperature, °K	Total Precipitation, mm/d	Total Cloudiness, fractions in %	Wind at 12 km, m/s
GCM with the original LWR and 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, %	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, %	0.001	0.02	0.	0.3	0.06
HGCM with LWR and SWR NN emulations	1011.50	288.92	2.89	61.12	16.29
Difference, %	0.002	0.03	1.04	0.6	0.5

<sup>a</sup>Row 1 shows the original LWR and SWR parameterizations. Rows 2 and 3 show results for HGCM with LWR NN emulation only and differences versus the original GCM. Rows 4 and 5 show results for HGCM with SWR NN emulation only and differences versus the original GCM. Rows 6 and 7 show results for HGCM with both NN emulations and differences versus the original GCM. NN50 (LWR) and NN55 (SWR) are used.

description of the NCAR CAM atmospheric SWR parameterization is presented in the special issue “The National Center for Atmospheric Research Community Climate Model” in *Journal of Climate*, 11(6), 1998. The input vectors for the NCAR CAM SWR parameterization include 21 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, 2006b]. 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 HRs and several radiation fluxes.

[114] 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.

[115] 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 20–300 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 multidecadal simulations without visible (significant) accumulations of errors in climate simulations as compared to the control run with the original SWR parameterization. Figure 4 shows the vertical error profile RMSEP( $j$ ) (equation (27)) 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.

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

[117] 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 6 and 7 of Table 4 (see explanation 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

[118] 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  $\mathbf{F}$  and has the nonlinear wave-wave interaction source term  $\mathbf{S}_{nl}$  as a part of the model physics. In its full form [e.g., Hasselmann and Hasselmann, 1985] the calculation of the  $\mathbf{S}_{nl}$  interac-

tions requires the integration of a six-dimensional Boltzmann integral,

$$\begin{aligned} \mathbf{S}_{nl}(\mathbf{k}_4) &= \mathbf{T} \otimes \mathbf{F}(\mathbf{k}) \\ &= \omega_4 \int \mathbf{G}(\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4) \delta(\mathbf{k}_1 + \mathbf{k}_2 - \mathbf{k}_3 - \mathbf{k}_4) \\ &\quad \cdot \delta(\omega_1 + \omega_2 - \omega_3 - \omega_4) [n_1 n_3 (n_4 - n_2) \\ &\quad + n_2 n_4 (n_3 - n_1)] d\mathbf{k}_1 d\mathbf{k}_2 d\mathbf{k}_3 \\ n(\mathbf{k}) &= \frac{\mathbf{F}(\mathbf{k})}{\omega}; \quad \omega^2 = g k \tanh(k h), \end{aligned} \quad (30)$$

where the mapping is symbolically represented by  $\mathbf{T}$  and the complicated coupling coefficient  $\mathbf{G}$  contains moving singularities. This integration requires roughly  $10^3$ – $10^4$  times more computational effort than all other aspects of the wave model combined. Present operational constraints require that the computational effort for the estimation of  $\mathbf{S}_{nl}$  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].

[119] Thus it is crucially important for the development of third-generation wave models to develop an economical yet accurate approximation for  $\mathbf{S}_{nl}$ . A neural network interaction approximation (NNIA) was explored to achieve this goal [Krasnopolsky *et al.*, 2002; Tolman *et al.*, 2005]. NNs can be applied here because the nonlinear interaction (30) is essentially a nonlinear mapping, symbolically represented by  $\mathbf{T}$ , which relates two vectors  $\mathbf{F}$  and  $\mathbf{S}_{nl}$  (2-D fields in this case). Discretization of  $\mathbf{S}$  and  $\mathbf{F}$  (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 discretization on a two-dimensional grid, which leads to  $\mathbf{S}$  and  $\mathbf{F}$  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.

[120] 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  $\mathbf{F}$  and source function  $\mathbf{S}_{nl}$  are expanded using systems of two-dimensional functions, each of which ( $\Phi_i$  and  $\Psi_q$ ) creates a complete and orthogonal two-dimensional basis

$$\mathbf{F} \approx \sum_{i=1}^n \xi_i \Phi_i, \quad \mathbf{S}_{nl} \approx \sum_{q=1}^m y_q \Psi_q, \quad (31)$$

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

$$\xi_i = \int \int \mathbf{F} \Phi_i, \quad y_q = \int \int \mathbf{S}_{nl} \Psi_q, \quad (32)$$

where the double integral identifies integration over the spectral space. Now the developed NN emulation relates

vectors of coefficients  $\mathbf{X}$  and  $\mathbf{Y}$ :  $\mathbf{Y} = \mathbf{T}_{NN}(\mathbf{X})$ . Typically,  $n = 20$ – $50$  and  $m = 100$ – $150$  in equation (31). Thus the reduction of the dimensionality of the NN emulation is very significant.

[121] Different approaches to the basis functions choice have been investigated by 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  $\Phi_i$  and  $\Psi_q$  are functions of two variables  $f$  and  $\theta$ . The set of spectra  $\mathbf{F}$  and source terms  $\mathbf{S}_{nl}$ , which are used for the training of the NN, are also used to generate the EOFs for decomposing  $\mathbf{F}$  and  $\mathbf{S}_{nl}$ . The main advantage of EOFs is the fast convergence of the decomposition (see Krasnopolsky *et al.* [2002] and Tolman *et al.* [2005] for more details). A graphical representation of the NNIA algorithm is shown in Figure 5.

[122] The NNIA is nearly 10 times more accurate than DIA. It is about  $10^5$  times faster than the original parameterization and only 7 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

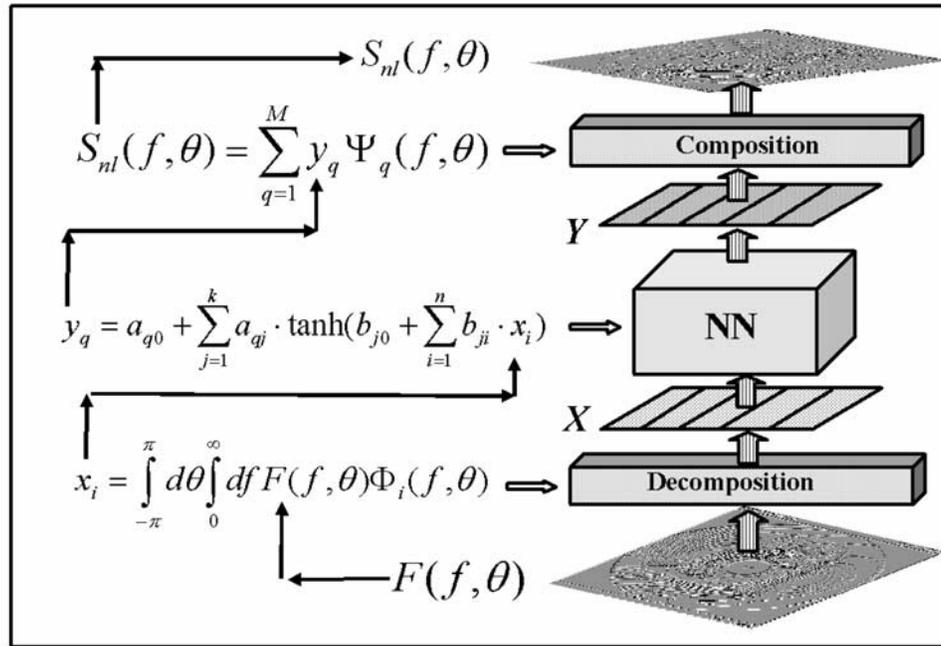
### 4.6.1. Summary and Advantages of the Hybrid Modeling Approach

[123] Throughout section 4 we reviewed a new hybrid paradigm in environmental numerical modeling. Within the framework of this paradigm a new type of ENM, a 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 the following.

[124] 1. 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, is presented.

[125] 2. NN emulations of model physics parameterizations developed by Krasnopolsky *et al.* [2000b, 2002, 2005a] are practically identical to the original physical parameterizations, because of 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.

[126] 3. Accurate NN emulations are robust and very fast ( $10$ – $10^5$  times faster than the original parameterization), so



**Figure 5.** Graphical representation of the neural network interaction approximation (NNIA) algorithm.

the significant speedup of HEM calculations can be achieved without compromising accuracy.

[127] 4. 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.

[128] 5. 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 and 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

[129] 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

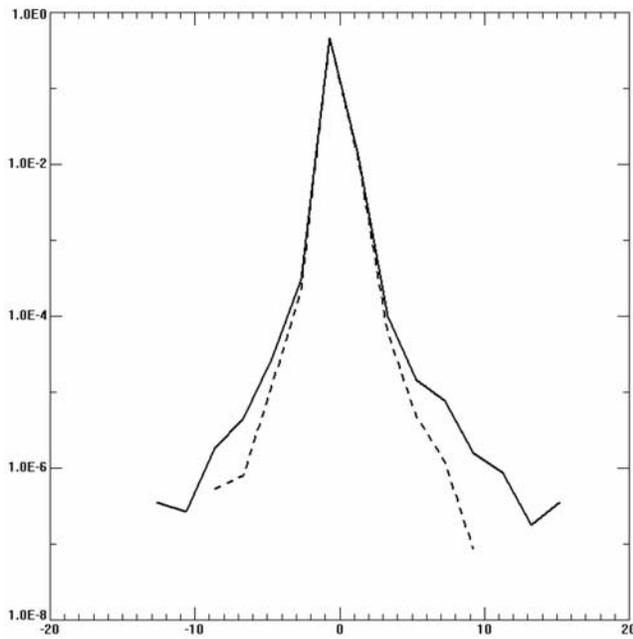
scenario. In both situations described the emulating NN may be forced to extrapolate beyond its generalization ability and may lead to errors in NN outputs and result in simulation errors in the corresponding HEM. Section 4.6.3 is devoted to addressing these and other issues.

#### 4.6.3. Current and Future Developments of the Hybrid Modeling Approach

[130] Two new techniques are being developed to take care of the kind of problems outlined in section 4.6.2 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, 2006b]. Here they are only briefly outlined.

[131] 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), on the basis of 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.

[132] There were different possible designs considered for QC [Krasnopolsky and Fox-Rabinovitz, 2006a, 2006b]. The first and simplest QC design is based on a set of regular physical and statistical tests that are used to check the



**Figure 6.** Probability density distributions of emulation errors for the SWR NN emulation NN55 (solid line) and for the compound SWR parameterization (dashed line). Both errors are calculated versus the original SWR parameterization. Compound parameterization reduces the probability of medium and large errors by an order of magnitude. The horizontal axis is RMSE in K/d; the vertical axis is logarithmic.

consistency of the NN outputs. This is the simplest, mostly generic but not sufficiently focused approach.

[133] 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  $PRMSE(i)$  (equation (25)) for each particular input vector  $X_i$ . Figure 6 shows the comparison of two error probability density functions. The solid line corresponds to the emulation errors of NN55, and the dashed line corresponds to the emulation errors of the CP (both errors are calculated versus the original parameterization on the independent test set; vertical axes are logarithmic). Figure 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.

[134] 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].

[135] 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).

[136] 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

[137] 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.

[138] 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 dependency 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, dissemination, and 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].

[139] 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, 2007] are presented in section 5.2.

### 5.1. SSH Mapping and Its NN Emulation

[140] Sea surface height (SSH)  $\eta$  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  $\sigma/z$  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 by 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  $1/3^\circ$  and 25 vertical levels.

[141] Since the reduced model physics has a mainly 1-D vertical structure, it was assumed that SSH  $\eta$  at a particular model grid point (i.e., at a particular horizontal latitude/longitude) depends only on the vector of state variables  $\mathbf{X}$  at the same horizontal location and the same time. Therefore this dependence (a target mapping) can be written as

$$\eta = \phi(\mathbf{X}), \quad (33)$$

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

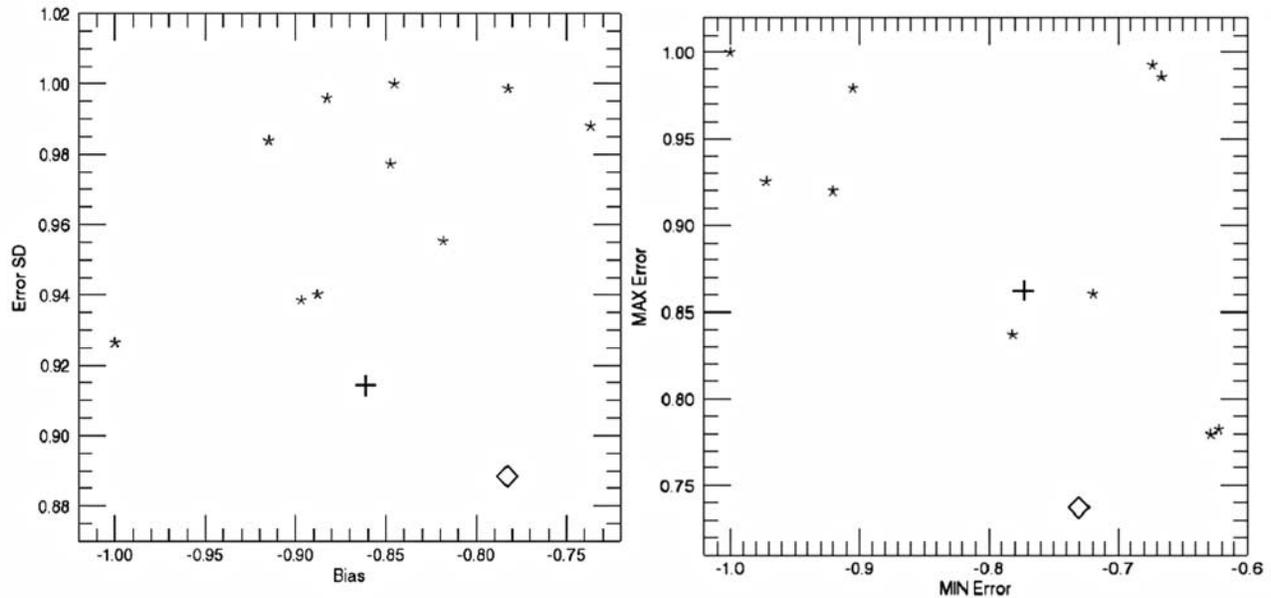
[142] The NN technique is applied to derive an analytical NN emulation for the relationship between model state variables  $\mathbf{X}$  and SSH  $\eta$ ,

$$\eta_{\text{NN}} = \phi_{\text{NN}}(\mathbf{X}), \quad (34)$$

using the simulated model fields which are treated as error free data. A simulation that covers almost 2 years (from Julian day 303, 2002, to day 291, 2004) was used to create training, validation, and test data sets. Training and validation data sets cover the period from day 300, 2002, to day 53, 2004, and an independent test set, the period from day 53 to day 291, 2004. Each data set consists of  $N = 536,259$  records ( $\{\eta_i, \mathbf{X}_i\}_{i=1, \dots, N}$ ) collocated in space and time and uniformly distributed over the model domain.

[143] 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 1 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.

[144] In the next test applied to the NN emulation the last day of the entire simulation (day 291, 2004) was selected. This day is separated by a time interval of about 8 months from the last day of the simulation used for training and validation (day 52, 2004). Fields generated by the model at 00Z were used to create inputs  $\mathbf{X}$  for the NN emulation (34). Then the NN emulation (34) was applied over the entire domain to generate the 2-D field of  $\eta_{\text{NN}}$ . This field was compared with the corresponding field of SSH  $\eta$  generated by the model. With the exception of several spots (most of them still close to the coastal areas) the difference between two 1 day fields does not exceed  $\pm 10$  cm. The accuracy of the NN emulation over the entire domain is considered to be



**Figure 7.** (left) Random part of the emulation error (the standard deviation 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 is represented by the cross, and the nonlinear ensemble using the averaging NN is represented by the diamond. (right) Extreme outliers' statistics. The vertical axis shows the largest positive (or maximum) and the horizontal axis shows the largest negative (or minimum) emulation error over the entire test set; each ensemble member is represented by a star, the conservative ensemble is represented by the cross, and the nonlinear ensemble is represented by the diamond.

satisfactory; the bias is about 1 cm, 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.

[145] The accuracy of the NN emulation may be improved using a NN ensemble approach (see section 5.2). The use of the NN emulation in DAS is conditioned by the quality of the NN Jacobian. The accuracy of the NN Jacobian and the possibility of improving this accuracy by using NN ensembles are also discussed in section 5.2.

## 5.2. NN Ensembles for Improving NN Emulation Accuracies and Reducing NN Jacobian Uncertainties

[146] 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.

[147] In the context of the problem described in section 5.1 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 10 NN emulations (equation (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 10 members, 10 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 in sections 5.2.1 and 5.2.2.

### 5.2.1. NN Ensembles for Improving the Accuracy of a NN Emulation; Linear Versus Nonlinear Ensembles

[148] 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 Figure 7 (left). The vertical axis of Figure 7 (left) shows the random part of the approximation error (the standard deviation of the error), and the horizontal axis shows 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 Figure 7 (left) by stars. The spread of the ensemble members is significant in Figure 7 (left). The systematic error changes about 25%, and the random error changes about 10% for different members.

[149] 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 10 NN

ensemble members was applied to the test set record by record. Thus, for each record (set of inputs), 10 NN outputs were produced. Then the mean value (in a regular statistical sense) of these 10 numbers was calculated and compared to the exact output to calculate the ensemble statistics represented by the cross in Figure 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. Figure 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 ( $\sim 15\%$ ) and random ( $\sim 9\%$ ) errors with respect to the maximum single-member errors is moderate but still significant.

[150] 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 (1 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.

[151] The result for the nonlinear ensemble using the averaging NN is shown in Figure 7 by a diamond. Figure 7 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 Figure 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.

[152] Figure 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 Figure 7 (right) by stars. Each ensemble also generates these two extreme outliers, shown by the cross for the conservative ensemble and the diamond for the nonlinear ensemble in Figure 7 (right).

[153] Figure 7 (right) shows that the NN ensemble approach is an effective tool in reducing extreme outliers

(by  $\sim 25\%$ ). However, a careful analysis of Figure 7 (right) also reveals other interesting features of the statistics presented. 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 Figure 7 (right), 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

[154] 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  $\{\partial\phi_{\text{NN}}/\partial\mathbf{X}_i\}_{i=1,\dots,n}$ ,

$$\Delta\eta_{\text{NN}} = \sum_{i=1}^n \left. \frac{\partial\phi_{\text{NN}}}{\partial\mathbf{X}_i} \right|_{\mathbf{X}=\mathbf{X}_0} \Delta\mathbf{X}_i, \quad (35)$$

where  $\Delta\mathbf{X}_i$  are increments of state variables,  $\mathbf{X}_0$  is an initial value of state variables, and  $n$  is the dimensionality of the vector  $\mathbf{X}$  (the number of inputs of the NN emulation (34)). Then the calculated  $\Delta\eta_{\text{NN}}$  is compared with the observed  $\Delta\eta_{\text{obs}}$ , and the difference is used to adjust  $\Delta\mathbf{X}$ .

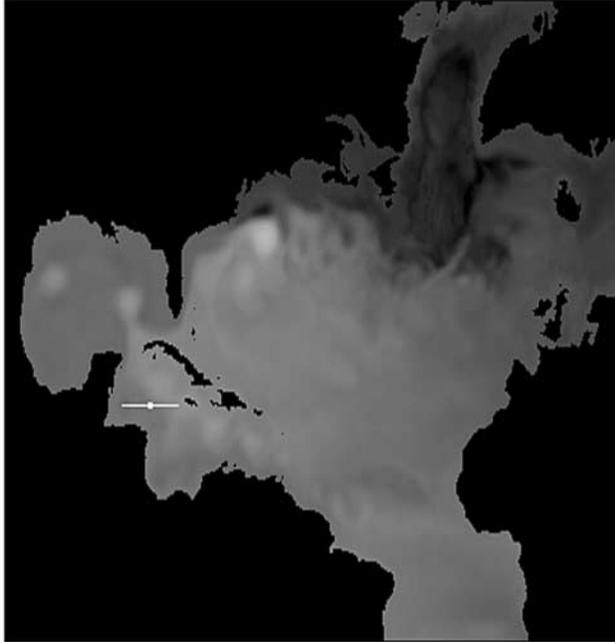
[155] The quality of the single NN Jacobian may not be sufficient to allow its use in DAS applications. However, an ensemble approach can be used to improve the NN Jacobian calculations. The NN ensemble described in section 5.2.1 was used to create an ensemble of 10 NN Jacobians  $\{\partial\phi_{\text{NN}}^j/\partial\mathbf{X}_i\}_{i=1,\dots,n}^{j=1,\dots,n_{em}}$ , where  $n_{em} = 10$  is the number of ensemble members. Then the ensemble mean Jacobian was calculated as

$$\overline{\frac{\partial\phi_{\text{NN}}}{\partial\mathbf{X}_i}} = \frac{1}{n_{em}} \sum_{j=1}^{n_{em}} \frac{\partial\phi_{\text{NN}}^j}{\partial\mathbf{X}_i}, \quad i = 1, \dots, n. \quad (36)$$

[156] Then equation (35) was used to calculate  $\Delta\eta_{\text{NN}}$  using each ensemble member's Jacobian and the ensemble average Jacobian (36). These values of  $\Delta\eta_{\text{NN}}$  were compared with the exact  $\Delta\eta$  known from the model simulation.

[157] 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  $\mathbf{X}$  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  $\Delta\eta_{\text{NN}}^j$  using equation (35). Also,  $\overline{\Delta\eta_{\text{NN}}}$  was calculated using the ensemble average Jacobian (36) in (35).

Lat = 17.53; Lon = -82.96



**Figure 8.** Location of the cross section (white horizontal line) inside the model domain. The white dot shows the position of  $\mathbf{X}_0$ . The picture is plotted in the internal model coordinates.

A nondimensional distance in the model state space between the vectors  $\mathbf{X}_0$  and  $\mathbf{X} = \mathbf{X}_0 + \Delta\mathbf{X}$  was also introduced as

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^n \left( \frac{\Delta\mathbf{X}_i}{\mathbf{X}_0^i} \right)^2}. \quad (37)$$

[158] Fields calculated in this way were compared with the corresponding field SSH  $\eta$  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 Figures 8–10.

[159] Figure 8 shows the location of the cross section (a white horizontal line) inside the model domain; the white dot shows the position of  $\mathbf{X}_0$ . Starting from this position, we moved to the left and right grid point by grid point, using  $\mathbf{X}$  values at these grid points to calculate  $\Delta\mathbf{X}$  and the nondimensional distance in the model state space  $S$  (equation (37)). These values of  $\Delta\mathbf{X}$  were used in (35) to calculate  $\Delta\eta$ .

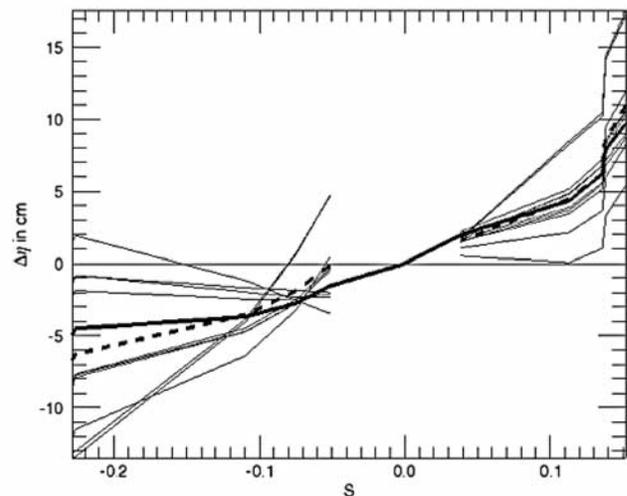
[160] Figure 9 shows  $\Delta\eta$  calculated using (35) and the NN ensemble member Jacobians (an envelope of thin solid lines illustrates the Jacobian uncertainties), the exact  $\Delta\eta$  calculated from the model (thick solid line), and  $\Delta\eta$  calculated using the ensemble average Jacobian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

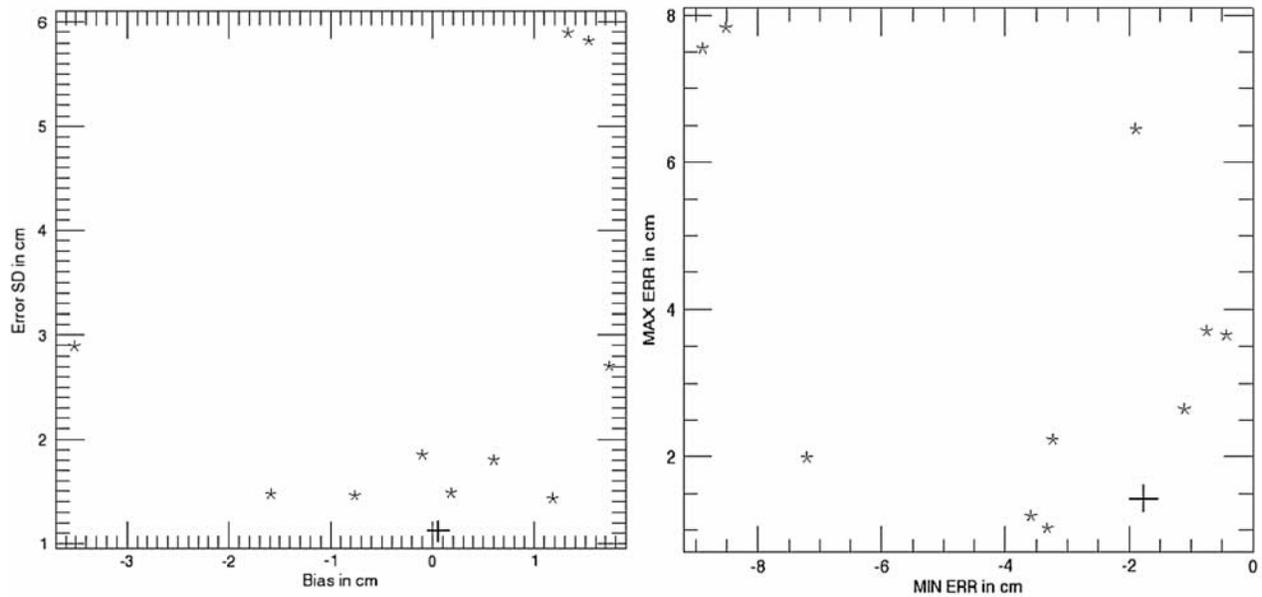
[161] Figure 10 (left) shows the bias and random error for  $\Delta\eta$  calculated along the path shown in Figure 8 using (35). The stars correspond to the errors when ensemble member Jacobians were used in (35), and the cross corresponds to the ensemble average Jacobian (36). The ensemble bias is equal to the mean bias of the members as expected when using this simple method to calculate the ensemble average. Figure 10 (left) also shows that in the case of Jacobian the ensemble approach very effectively reduces random errors or error standard deviation; the ensemble random error ( $\sim 1.1$  cm) is less than the random error of any ensemble member. The reduction in bias ( $\sim 90\%$ ) and random error ( $\sim 65\%$ ) with respect to the maximum single-member errors is very significant.

[162] Figure 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 exists 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 Figure 10 (right) by stars. When used in (35), the ensemble average Jacobian (36) also generates two extreme outliers that are represented in Figure 10 (right) by the cross. Figure 10 (right) shows that the NN ensemble approach is also an effective tool for reducing large errors (by  $\sim 4$  times) in NN Jacobians.

[163] Then this same procedure was applied at all grid points in the model domain. The errors have been calculated



**Figure 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 versus the distance in the model state space  $S$  (equation (37)).



**Figure 10.** (left) Systematic error (bias) and the random error (error standard deviation) for  $\Delta\eta$  calculated along the path shown in Figure 8 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. (right) Minimum and maximum errors along the path. The stars correspond to errors when the ensemble member Jacobians were used in equation (35), and the cross corresponds to when the ensemble average Jacobian (36) was used in (35).

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 nondimensional distance  $\mathbf{S}$  over the entire domain. The ensemble significantly improves statistics at all distances  $\mathbf{S}$  considered. In this case the ensemble is always better than the best ensemble member.

[164] To better understand the magnitudes of errors presented in this section and section 5.2.1, these magnitudes should be compared with the errors in the observed satellite data  $\Delta\eta_{\text{obs}}$  assimilated in the oceanic DAS (see section 5.2.1). 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

[165] 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.

[166] 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.

[167] 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

[168] 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, and support vector machines.

[169] 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.

[170] 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, 2006].

[171] 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 one 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., 2005]. 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 dimen-

sionality 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.

[172] **ACKNOWLEDGMENTS.** The author would like to thank M. Fox-Rabinovitz for fruitful collaboration, multiple and stimulating discussions of topics presented in this paper, and careful reading and commenting on the manuscript. The author also thanks D. V. Chalikov and H. L. Tolman for long-term fruitful collaboration.

[173] The Editor responsible for this paper was Gerald North. He thanks one anonymous technical reviewer and one anonymous cross-disciplinary reviewer.

## 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. Chédin, and N. Scott (1999), The “weight smoothing” regularization of MLP for Jacobian stabilization, *IEEE Trans. Neural Networks*, *10*, 1502–1510.
- Aires, F., W. B. Rossow, N. A. Scott, and A. Chédin (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*(D22), 4620, doi:10.1029/2001JD001591.
- Aires, F., C. Prigent, and W. B. Rossow (2004a), Neural network uncertainty assessment using Bayesian statistics: A remote sensing application, *Neural Comput.*, *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, doi:10.1029/2003JD004175.
- Alishouse, J. C., et al. (1990), Determination of oceanic total precipitable water from the SSM/I, *IEEE Trans. Geosci. Remote Sens.*, *23*, 811–816.
- Atkinson, P. M., and A. R. L. Tatnall (1997), Neural networks in remote sensing—Introduction, *Int. J. Remote Sens.*, *18*, 699–709.
- Attali, J.-G., and G. Pagès (1997), Approximations of functions by a multilayer perception: A new approach, *Neural Networks*, *6*, 1069–1081.
- Bankert, R. L. (1994), Cloud pattern identification as part of an automated image analysis, paper presented at Seventh Conference on Satellite Meteorology and Oceanography, Am. Meteorol. Soc., Monterey, Calif., 6–10 June.
- Barai, S. V., and Y. Reich (1999), Ensemble modeling or selecting the best model: Many could be better than one, *Artif. Intel. Eng. Design Anal. Manuf.*, *13*, 377–386.
- Barron, A. R. (1993), Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Trans. Inf. Theory*, *39*, 930–945.
- Beale, R., and T. Jackson (1990), *Neural Computing: An Introduction*, 240 pp., Adam Hilger, Bristol, U. K.
- Bhattacharya, B., and D. P. Solomatine (2006), Machine learning in soil classification, *Neural Networks*, *19*, 186–195.
- Bhattacharya, B., C. Buraimo, and D. P. Solomatine (2005), Data-driven modelling in context to sediment transport, *J. Phys. Chem. Earth B*, *30*, 297–302.
- Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, 482 pp., Oxford Univ. 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 atmo-

- spheric 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*, 80 pp., Kluwer Acad., Boston, Mass.
- Cabrera-Mercader, C. R., and D. H. Staelin (1995), Passive microwave relative humidity retrievals using feedforward neural networks, *IEEE Trans. Geosci. Remote Sens.*, 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 coordinate ocean 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. Titterton (1994), Neural networks: A review from a statistical perspective, *Stat. Sci.*, 9, 2–54.
- Cherkassky, V., and F. Mulier (1998), *Learning From Data*, 441 pp., John Wiley, Hoboken, N. J.
- Chevallier, F. (2005), Comments on “New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model,” *Mon. Weather Rev.*, 133, 3721–3723.
- Chevallier, F., and J.-F. Mahfouf (2001), Evaluation of the Jacobians of infrared radiation models for variational data assimilation, *J. Appl. Meteorol.*, 40, 1445–1461.
- Chevallier, F., F. Ch eruy, N. A. Scott, and A. Ch edın (1998), A neural network approach for a fast and accurate computation of longwave radiative budget, *J. Appl. Meteorol.*, 37, 1385–1397.
- Chevallier, F., J.-J. Morcrette, F. Ch eruy, and N. A. Scott (2000), Use of a neural-network-based longwave radiative transfer scheme in the EMCWF atmospheric model, *Q. J. R. Meteorol. Soc.*, 126, 761–776.
- Chou, M.-D., M. J. Suarez, X.-Z. Liang, and M. M.-H. Yan (2001), A thermal infrared radiation parameterization for atmospheric studies, in *Technical Report Series on Global Modeling and Data Assimilation*, edited by J. Max Suarez, *NASA Tech. Memo.*, NASA/TM-2001-104606, 37–45.
- Cilliers, P. (1998), *Complexity and Postmodernism: Understanding Complex Systems*, 157 pp., Routledge, London.
- Colliers, P. (2000), What can we learn from a theory of complexity?, *Emergence*, 2, 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), An updated parameterization for infrared emission and absorption by water vapor in the National Center for Atmospheric Research Community Atmosphere Model, *J. Geophys. Res.*, 107(D22), 4664, doi:10.1029/2001JD001365.
- Cornford, D., I. T. Nabney, and G. Ramage (2001), Improved neural network scatterometer forward models, *J. Geophys. Res.*, 106, 22,331–22,338.
- Cybenko, G. (1989), Approximation by superposition of sigmoidal functions, *Math. Control Signals Syst.*, 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. 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, *Mon. Weather Rev.*, 126, 2287–2299.
- DeVore, R. A. (1998), Nonlinear approximation, *Acta Numer.*, 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. Am. Meteorol. Soc.*, 73, 49–60.
- Eyre, J. R., and A. C. Lorenc (1989), Direct use of satellite sounding radiances in numerical weather prediction, *Meteorol. Mag.*, 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* [CD-ROM], pp. 9321–9326, IEEE Press, Piscataway, N. J.
- 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., and C. T. Swift (1992), Improved retrievals from the DMSP wind speed algorithm under adverse weather conditions, *IEEE Trans. Geosci. Remote Sens.*, 30, 1076–1077.
- 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, 14,547–14,555.
- Guinehut, S., P. Y. Le Traon, G. Larnicol, and 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. Mar. Syst.*, 46, 85–98.
- Hansen, L. K., and P. Salamon (1990), Neural network ensembles, *IEEE Trans. Pattern Anal. Mach. Intel.*, 12, 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, *J. Phys. Oceanogr.*, 15, 1369–1377.
- Hasselmann, S., et al. (1985), Computations and parameterizations of the nonlinear energy transfer in a gravity wave spectrum. Part II: Parameterization of the nonlinear transfer for application in wave models, *J. Phys. Oceanogr.*, 15, 1378–1391.
- Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*, 696 pp., Macmillan, New York.
- Hennon, C. C., C. Marzban, and J. S. Hobgood (2005), Improving tropical cyclogenesis statistical model forecasts through the application of a neural network classifier, *Weather Forecast.*, 20, 1073–1083.
- Hopfield, J. J. (1982), Neural networks and physical systems with emergent collective computational ability, *Proc. Natl. Acad. Sci. U. S. A.*, 79, 2554–2558.
- Hornik, K. (1991), Approximation capabilities of multilayer feedforward network, *Neural Networks*, 4, 251–257.
- 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.
- Hsieh, W. W. (2001), Nonlinear principle component analysis by neural networks, *Tellus, Ser. A*, 53, 599–615.
- Hsieh, W. W. (2004), Nonlinear multivariate and time series analysis by neural network methods, *Rev. Geophys.*, 42, RG1003, doi:10.1029/2002RG000112.
- 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.
- Jolliffe, I. T. (2002), *Principal Component Analysis*, 502 pp., Springer, New York.

- Kohonen, T. (1982), Self-organizing formation of topologically correct feature maps, *Biol. Cybern.*, 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. (1997), A neural network-based forward model for direct assimilation of SSM/I brightness temperatures, *OMB Contrib. 140*, Natl. Cent. for Environ. Predict., NOAA, Camp Springs, Md.
- Krasnopolsky, V. M. (2007), Reducing uncertainties in neural network Jacobians and improving accuracy of neural network emulations with NN ensemble approaches, *Neural Networks*, 20, 454–461.
- 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., and M. S. Fox-Rabinovitz (2006a), A new synergetic paradigm in environmental numerical modeling: Hybrid models combining deterministic and machine learning components, *Ecol. Modell.*, 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. Part. Phys.*, 3, 795–807.
- 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.
- 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, 11,033–11,045.
- Krasnopolsky, V., W. H. Gemmill, and L. C. Breaker (1996), A new transfer function for SSM/I based on an expanded neural network architecture, *OMB Contrib. 137*, Natl. Cent. for Environ. Predict., NOAA, Camp Springs, Md.
- Krasnopolsky, V. M., W. H. Gemmill, and L. C. Breaker (1999), A multiparameter empirical ocean algorithm for SSM/I retrievals, *Can. J. Remote Sens.*, 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 Sens. Environ.*, 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, paper presented at Second Conference on Artificial Intelligence, Am. Meteorol. Soc., Long Beach, Calif., 9–14 Jan.
- Krasnopolsky, V. M., D. V. Chalikov, and H. L. Tolman (2002), A neural network technique to improve computational efficiency of numerical oceanic models, *Ocean Modell.*, 4, 363–383.
- 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, *Mon. Weather Rev.*, 133, 1370–1383.
- Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and M.-D. Chou (2005b), Robustness of the NN approach to emulating atmospheric long wave radiation in complex climate models, in *Proceedings of 2005 IEEE International Joint Conference on Neural Networks: IJCNN05*, pp. 2661–2665, IEEE Press, Piscataway, N. J.
- Krasnopolsky, V. M., M. S. Fox-Rabinovitz, and D. V. Chalikov (2005c), Reply, *Mon. Weather Rev.*, 133, 3724–3729.
- 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* [CD-ROM], pp. 8732–8734, IEEE Press, Piscataway, N. J.
- 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, C09024, 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, *Q. J. R. Meteorol. Soc.*, 122, 1177–1194.
- Lorenz, E. N. (1956), Empirical orthogonal functions and statistical weather prediction, *Sci. Rep. I*, 48 pp., Stat. Forecast. Proj., Mass. Inst. of Technol., Cambridge.
- Loyola, D., and T. Ruppert (1998), A new PMD cloud-recognition algorithm for GOME, *Earth Obs. Q.*, 58, 45–47.
- MacKay, D. J. C. (1992), A practical Bayesian framework for back-propagation networks, *Neural Comput.*, 4, 448–472.
- Maclin, R., and J. Shavlik (1995), Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks, paper presented at 11th International Conference on Artificial Intelligence, Am. Assoc. on Artif. Intel., Detroit, Mich.
- Marzban, C., and G. J. Stumpf (1996), A neural network for tornado prediction based on Doppler radar-derived attributes, *J. Appl. Meteorol.*, 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, *Q. J. R. Meteorol. Soc.*, 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. (2007), Neural network retrieval of ocean surface parameters from SSM/I data, *Mon. Weather Rev.*, 135, 586–597.
- Müller, M. D., A. K. Kaifel, M. Weber, S. Tellmann, J. P. Burrows, and D. Loyola (2003), Ozone profile retrieval from Global Ozone Monitoring Experiment (GOME) data using a neural network approach (Neural Network Ozone Retrieval System (NNORSY)), *J. Geophys. Res.*, 108(D16), 4497, doi:10.1029/2002JD002784.
- Nabney, I. T. (2002), *Netlab: Algorithms for Pattern Recognition*, Springer, 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, 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 on Neural Networks, 1990*, vol. 3, pp. 21–26, IEEE Press, Piscataway, N. J.
- Nilsson, N. J. (1965), *Learning Machines: Foundations of Trainable Pattern-Classifying Systems*, 137 pp., McGraw-Hill, New York.
- Opitz, D., and R. Maclin (1999), Popular ensemble methods: An empirical study, *J. Artif. Intel. Res.*, 11, 169–198.
- Parker, R. L. (1994), *Geophysical Inverse Theory*, 400 pp., Princeton Univ. Press, Princeton, N. J.
- Parrish, D. F., and J. C. Derber (1992), The National Meteorological Center's spectral statistical-interpolation analysis system, *Mon. Weather Rev.*, 120, 1747–1763.
- Pasini, A., M. Lorè, and F. Ameli (2006), Neural network modeling for the analysis of forcings/temperatures relationships at different scales in the climate system, *Ecol. Modell.*, 191, 58–67.
- Petty, G. W. (1993), A comparison of SSM/I algorithms for the estimation of surface wind, paper presented at Shared Processing

- Network DMSP SSM/I Algorithm Symposium, Fleet Numer. Meteorol. and Oceanogr. Cent., Monterey, Calif., 8–10 June.
- 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, *J. Atmos. Oceanic Technol.*, *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, *J. Atmos. Oceanic Technol.*, *11*, 617–628.
- Phalippou, L. (1996), Variational retrieval of humidity profile, wind speed and cloud liquid–water path with the SSM/I: Potential for numerical weather prediction, *Q. J. R. Meteorol. Soc.*, *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. Appl. Meteorol.*, *36*, 493–508.
- Reitsma, F. (2001), Spatial complexity, M. S. thesis, 149 pp., Auckland Univ., Auckland, New Zealand.
- Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, 403 pp., Cambridge Univ. 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 the International Joint Conference on Neural Networks, 2001: IJCNN01*, vol. 3, pp. 2150–2152, IEEE Press, Piscataway, N. J.
- Selfridge, O. G. (1958), Pandemonium: A paradigm for learning, in *Mechanisation of Thought Processes*, pp. 513–526, Her Majesty's Stn. Off., London.
- Sharkey, A. J. C. (1996), On combining artificial neural nets, *Connection Sci.*, *8*, 299–313.
- Smith, J. A. (1993), LAI inversion using a back-propagation neural network trained with a multiple scattering model, *IEEE Trans. Geosci. Remote Sens.*, *31*, 1102–1106.
- Solomatini, D. P. (2005), Data-driven modeling and computational intelligence methods in hydrology, in *Encyclopedia of Hydrological Sciences*, pp. 293–306, John Wiley, Hoboken, N. J.
- 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.*, *99*, 981–984.
- Tang, Y., and W. W. Hsieh (2003), ENSO simulation and prediction in a hybrid coupled model with data assimilation, *J. Meteorol. Soc. Jpn.*, *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*, 22,827–22,841.
- Tolman, H. L. (2002), User manual and system documentation of WAVEWATCH III version 2.22, *Tech. Note 222*, 133 pp., Environ. Model. Cent., NOAA, Camp Springs, Md.
- Tolman, H. L., and V. M. Krasnopolsky (2004), Nonlinear interactions in practical wind wave models, paper presented at 8th International Workshop on Wave Hindcasting and Forecasting, U. S. Army Eng. Res. and Dev. Cent. Coastal and Hydraulics Lab., Turtle Bay, Hawaii.
- 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 Modell.*, *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 Trans. Geosci. Remote Sens.*, *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. (2006), *Estimation of Dependences Based on Empirical Data*, translated from Russian by S. Kotz, 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 Atmosphere, Methods Comput. Phys.*, vol. 17, edited by J. Chang, pp. 111–172, Academic, New York.
- Weigend, A. S., and N. A. Gershenfeld (1994), The future of time series: Learning and understanding, in *Time Series Prediction: Forecasting the Future and Understanding the Past*, edited by A. S. Weigend and N. A. Gershenfeld, pp. 1–70, Addison-Wesley, Reading, Mass.
- Weng, F., and N. G. Grody (1994), Retrieval of cloud liquid water using the Special Sensor Microwave Imager (SSM/I), *J. Geophys. Res.*, *99*, 25,535–25,551.
- 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*, 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.

---

V. Krasnopolsky, EMC, NCEP, NOAA, 5200 Auth Road, Camp Springs, MD 20746-4304, USA. (vladimir.krasnopolsky@noaa.gov)