PERGAMON

AVAILABLE AT
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Neural Networks 16 (2003) 335–348

Neural Networks

www.elsevier.com/locate/neunet

2003 Special Issue

Some neural network applications in environmental sciences. Part II: advancing computational efficiency of environmental numerical models

Vladimir M. Krasnopolsky[a,*], Frédéric Chevallier[b]

[a]*Science Applications International Corporation at National Centers for Environmental Prediction, 5200 Auth Road, Camp Spring, MD 20746, USA*
[b]*European Centre for Medium-Range Weather Forecasts, Reading, UK*

Abstract

A new generic neural network (NN) application—improving computational efficiency of certain processes in numerical environmental models—is considered. This approach can be used to accelerate the calculations and improve the accuracy of the parameterizations of several types of physical processes which generally require computations involving complex mathematical expressions, including differential and integral equations, rules, restrictions and highly nonlinear empirical relations based on physical or statistical models. It is shown that, from a mathematical point of view, such parameterizations can usually be considered as continuous mappings (continuous dependencies between two vectors) and, therefore, NNs can be used to replace primary parameterization algorithms. In addition to fast and accurate approximation of the primary parameterization, NN also provides the entire Jacobian for very little computation cost.

Four particular real-life applications of the NN approach are presented here: for oceanic numerical models, a NN approximation of the UNESCO equation of state of the sea water (NN for the density of the seawater) and an inversion of this equation (NN for the salinity of the seawater); for atmospheric numerical models, a NN approximation for long wave radiative transfer code; and for wave models, a NN approximation for the nonlinear wave–wave interaction. In all considered applications a significant acceleration of numerical computations has been achieved. The first two of these NN applications have already been implemented in the multi-scale ocean forecast system at NCEP.

The NN approach introduced in this paper can provide numerically efficient solutions to a wide range of problems in numerical models where lengthy, complicated calculations, which describe physical, chemical and/or biological processes, must be repeated frequently.

Keywords: Neural networks; Numerical modeling; Atmospheric modeling; Oceanic modeling; Wave modeling; Equation of state; Nonlinear interaction; Parameterization of physics

1. Introduction

In this paper, we discuss a recently emerged application of neural networks (NNs) for efficient (fast and accurate) calculations of the computationally expensive and complex mathematical formulations involved in environmental numerical models. Any atmospheric or oceanic circulation model is based on a set of prognostic and diagnostic differential equations together with additional equations required to obtain a mathematically closed system. Such a system, in principle, can then be solved to predict the evolution of the environment in time if the initial conditions and any required external boundary conditions are prescribed. Even though the forecast problem may now be considered solvable in a theoretical sense, in the real world of simulation, it is necessary to deal with practical aspects of available computational resources and minimize the computer time taken to produce a run, in particular in the context of operational forecast systems.

The numerical model contains terms that appear in the dynamical equations, representing the effects of unresolvable subgrid scale processes. These effects need to be parameterized in terms of the dependent variables. Also, implicitly contained in the system are processes that deal with model physics such as radiation, convection, nonlinear wave–wave interaction, etc. which need to be parameterized. Accurate treatments of such parameterizations generally require computations involving complex mathematical expressions, which may include differential and integral equations, rules, restrictions, highly nonlinear empirical expressions, etc. that are developed based on physical or statistical models. The complex mathematical formulations of these processes require considerable computational resource.

* Corresponding author. Tel.: +1-301-763-8000 ext. 7262; fax: +1-301-763-8545.
E-mail address: vladimir.krasnopolsky@noaa.gov (V.M. Krasnopolsky).

0893-6080/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0893-6080(03)00026-1

For example, a spectral atmospheric model with a well-developed description of physics and subgrid scale parameterizations may spend up to 70% of calculation time for simulating these processes (Estrade, Trémolet, & Sela, 2001). The infrared radiative heating/cooling code requires more than 10% of the computing time in the European Centre for Medium-Range Weather Forecast (ECMWF) general circulation model and in the National Centers for Environmental Prediction (NCEP) global model, even though the computations are not afforded at every grid point (at ECMWF) and at every time step.

In ocean models, the estimation of the full UNESCO equation of state to compute the seawater density, represented by an empirically derived highly nonlinear equation relating density to pressure, salinity, and temperature, takes a very significant amount (~40% in high resolution models) of the total computational effort. In addition, most forecast models include data assimilation procedures as an integral part of the forecast system to improve the initial conditions of the model. When dealing with ocean models, most often the data assimilation consists of assimilating surface and subsurface temperature observations to correct the model's thermal field. This temperature correction automatically makes it necessary to adjust the salinity field in the ocean model in order to avoid gravitational instabilities in the water column. This requires inverting the complicated oceanic equation of state, which makes the computational effort even more time-consuming than the forward problem of computing the density itself. Another example where intensive computations are needed in a forecast model is the calculation of the land surface temperature using a set of equations describing the atmospheric boundary layer and physical processes in the soil. Yet another example of intensive computational problem in forecast models is the wind wave forecasting problem in which an exact calculation of the nonlinear wave–wave interactions using the formulation of Hasselmann (1963) takes a prohibitively long time.

In view of the constraints imposed on the available computer resources, the calculation time allowed for each parameterization is strictly limited in most operational forecast models. Hence, very often it is necessary to use simplified forms of these complex representations in carrying out the time integrations in a forecast model, thereby sacrificing the accuracy of forecasts. For example, the nonlinear wave interactions in a wave forecast model are replaced by a simplified discrete interaction approximation (DIA) (Hasselmann et al., 1985). Similarly, simplified fast parameterizations of physics are used in many parts of atmospheric and oceanic models. In most of these cases, accurate physical models have been developed, but they cannot be used because they are computationally too expensive. Often simplified (even oversimplified due to computational efficiency requirements) parameterizations are obtained, for example, by neglecting higher order terms of perturbation theory, by using empirical approximations, or simply by neglecting the effects, which complicate the calculations. It is common in many parameterization schemes that the number of input and output variables is relatively small, whereas the volume of internal calculations is large. A typical example is the parameterization of the radiative fluxes in the atmosphere. Indeed, accurate treatment of cloud- and aerosol-radiation interactions involves elaborate and numerically onerous 3D scattering methods. Hence, most often the specific parameterization is a result of a compromise between accuracy and computational efficiency with an (sometimes) unpredictable effect on the forecast.

Improvements in forecast modeling can be achieved not only by improving the representation of such parameterizations as our understanding of the underlying physical processes increases but also by improving our ability to compute these parameterizations accurately within the constraints imposed by the available computer resources.

In this paper we present some of the problems dealing with physical parameterizations and their computations from a different (formal mathematical) point of view, namely that of improving the computational efficiency of available algorithms. We propose a generic approach, which is based on developing fast and accurate parameterizations of physics by approximating solutions of exact physical models using NNs. From this formal point of view an exact (best known) physical model representing a physical process performs a smooth conversion of an input vector of parameters, $X = \{x_1, x_2, ..., x_n\}$, $X \in \Re^n$ into an output vector of parameters, $Y = \{y_1, y_2, ..., y_m\}$, $Y \in \Re^m$. Thus, each output parameter y_i is a continuous function of multiple input variables $x_1, x_2, ..., x_n$ (input vector X). Symbolically this input–output dependence is depicted in Fig. 1.a and can be written as

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

If X and Y are related through a cause and effect principle, the forward parameterization, Eq. (1a), can be derived from first principles. It is usually a well-posed problem. If the inverse dependence

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

is required (Fig. 1b) in a numerical model, the inverse problem should be solved, which implies that Eq. (1a) should be inverted. A solution of the inverse problem (1b) or an inverse parameterization provides each output parameter x_i as a continuous function of multiple input variables $y_1, y_2, ..., y_m$ (vector Y is an input vector now). Often the inverse parameterization (1b) is an ill-posed problem, and sometimes multiple values of X can correspond to a single Y. Forward, Eq. (1a), and inverse, Eq. (1b), parameterizations represent the same mathematical object—a continuous mapping which is a continuous relationship between two vectors. Usually these input/output relationships are highly complex and nonlinear, but continuous or almost continuous (with a finite number of finite discontinuities), for physical

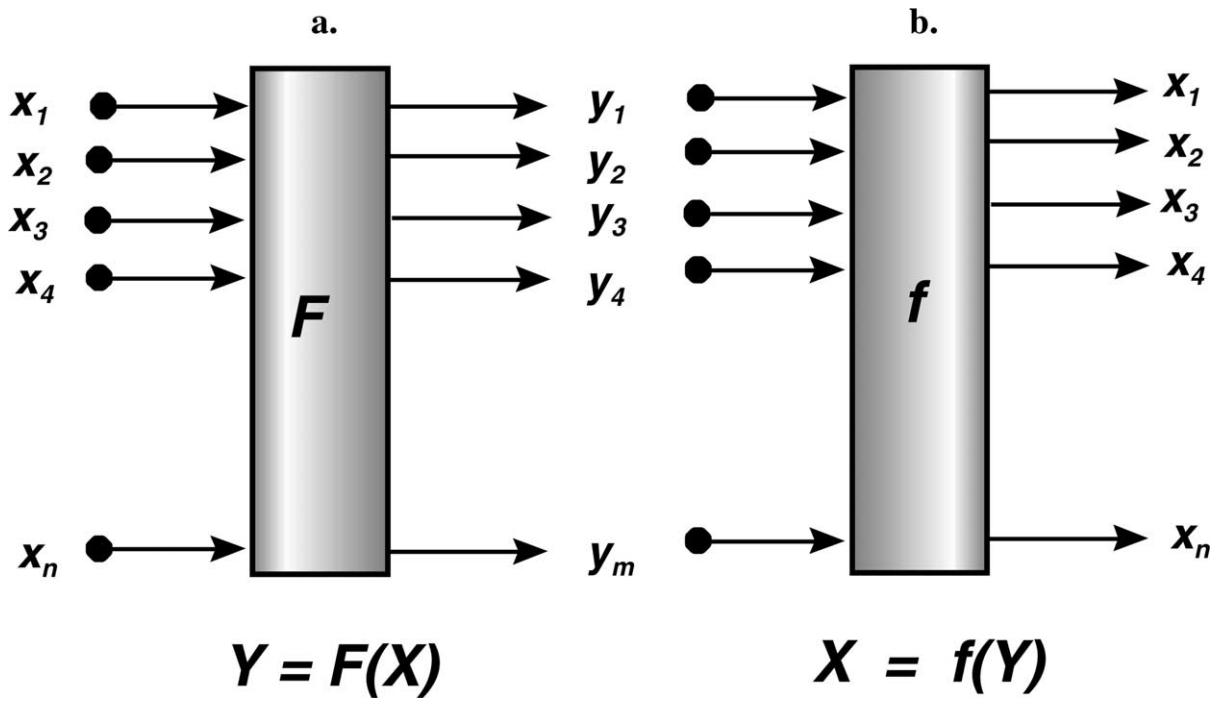

Fig. 1. Graphical representation of forward (a) and inverse (b) parameterizations.

processes taken into account in atmospheric, oceanic, and wave models. Hence, if exact solutions to these complex relationships are calculated, however expensive the computational efforts may be, these solutions can be used by the generic mathematical tool—that is, the NNs—to produce fast and accurate approximations for continuous and almost continuous mappings (Attali & Pagès, 1997; Chen & Chen 1995a,b; Cybenko, 1989; Funahashi, 1989). In this approach, the costly exact calculation of the physics needs to be performed only once and 'off line' to enable the development of the fast and accurate approximation. After that only this fast and accurate approximation will be used to calculate the physics (coefficients of differential equations) 'on line' in a numerical model.

We assume that the readers of this journal are well familiar with the NN technique. Therefore we do not describe this technique in the paper. We only present here a brief list of main properties of NNs, which make them a very suitable generic tool for our application.

- NNs are able to accurately approximate complicated nonlinear input/output relationships (any continuous and almost continuous nonlinear mappings).
- While training the NN is often time consuming, its application is not. After the training is finished (it is usually performed only once), each application of the trained NN is an estimation of a simple algebraic expression with known coefficients, which is practically instantaneous (several tens of floating point additions and multiplications).
- NNs are analytically differentiable, in a way that make the calculation of entire Jacobian matrix cheap.
- NN technique is flexible enough to accommodate various additional constraints, which may arise in this application.

In Section 2 of this paper we present two (forward and inverse) parameterizations for oceanic models, in Section 3 for atmospheric, and in Section 4 a parameterization for wind wave models developed using NNs. In Section 5, we discuss some important features of our approach and some generalizations of standard NN techniques, which are required to accommodate these features.

2. Oceanic applications: NNs for efficient calculation of sea water density or salinity from the UNESCO Equation of State

In this section, we apply a NN technique to two related problems in the fast calculation of physics in oceanic modeling and data assimilation. (i) In most ocean models, the UNESCO International Equation of State for Seawater (UNESCO, 1981) (UES) is used for the calculation of the seawater density at each point of a 3D grid using a relatively small time step. The frequency of updating the density depends on specifics of the model. For high-resolution models, the solution of this equation consumes a significant part of the overall computation time. (ii) In the data assimilation process, assimilation of temperature alone, without making corresponding adjustments to salinity, in ocean models, which employ the full equation of state, can lead to problems of gravitational instabilities (Chalikov et al. 1998; Woodgate, 1998). To adjust the salinity, we need to

calculate the salinity from UES as a function of temperature, density and depth (or pressure), i.e. solve an inverse problem in many points. Numerical inversion of the UES is an iterative procedure, which can consume several orders of magnitude more time than solving of the UES itself.

The UES for seawater gives the following expression for the density anomaly δ_ρ (kg/m^3) as described by Fofonoff and Millard (1983)

$$\delta_\rho(T,S,P) = \rho(T,S,P) - 1000 \tag{2}$$

$$\rho(T,S,P) = \frac{\rho(T,S,0)}{1 - \dfrac{P}{K(T,S,P)}}$$

where ρ is the density of seawater in kg/m^3, T the temperature in °C, S the salinity in practical salinity units (psu), P the pressure, and $K(T,S,P)$ is a bulk modulus (UNESCO, 1981).

The UES (2) is empirically based and given over a 3D domain $D = \{-2 < T < 40\,^\circ\text{C},\ 0 < S < 40\ \text{psu},\ \text{and}\ 0 < P < 10{,}000\ \text{decibars}\}$. This domain represents all possible combinations of T, S, and P, which are globally encountered. Mathematically, the functions $\rho(T,S,0)$ and $K(T,S,P)$ are represented by multi-dimensional high degree polynomials and, as a result, the density (2) is a ratio of two 3D polynomials which contain more than 40 parameters.

The UES has two major drawbacks when it is applied in the context of ocean modeling. The first is its cumbersome form. For high-resolution models, the solution of this equation at each point of a 3D grid for each time step consumes a significant part (up to 40%) of the overall computation time. Second, it is not a simple matter using the UES to obtain solutions for salinity of seawater, since this solution represents an inverse dependence.

The UES determines the density field from observed temperature, salinity, and pressure to within a standard error of approximately 0.009 kg m^{-3}; however, there are several natural processes (e.g. variations in the composition of dissolved salts) (Apel, 1987), contributing to the uncertainty in the density of natural seawater. The resulting natural uncertainty in the density is of the order of 0.1 kg m^{-3} (Krasnopolsky et al., 2000; Krasnopolsky, Chalikov, & Tolman, 2003). Taking these uncertainties into account, it does not make sense to use parameterization with higher accuracy in numerical ocean models if accuracy and computing time increase together. This is why the accuracy of 0.1 kg m^{-3} was selected as the expected accuracy for the NN parameterization. The accuracy of the NN parameterization for salinity expressed in terms of density was also expected to be better than 0.1 kg m^{-3}.

Since the depth Z is used in NCEP ocean model as a vertical coordinate, and there exist a simple pressure–depth relationship (Krasnopolsky et al., 2003), we use Z instead of P in our consideration. The UES defines two relationships (second relationship for salinity through inversion)

$$\rho = \rho(T,S,Z) \tag{3a}$$

$$S = S(T,\rho,Z) \tag{3b}$$

which are continuous mappings (degenerated mappings because 1D vectors are on the left). The NN technique was applied to approximate Eqs. (3a) and (3b). To create a training set for these NNs parameterizations in the 3D domain D (see earlier), 4000 points (T_i, S_i, Z_i) were generated on a grid. The UES was used to estimate the density of seawater, ρ_i, for each point. This simulated data set $\{\rho_i, T_i, S_i, Z_i\}$ was used in order to train the NNs to extract density and salinity. NNs with three nonlinear neurons in one hidden layer and one linear neuron in the output layers were selected. Two NN parameterizations were obtained (Fig. 2)

$$\rho = \rho_{\text{NN}}(T,S,Z) \tag{4a}$$

$$S = S_{\text{NN}}(T,\rho,Z) \tag{4b}$$

where both ρ_{NN} and S_{NN} are expressed by

$$\rho(S)_{\text{NN}} = \sum_{j=1}^{k} \omega_j^{\rho(S)} \left[\tanh\left(\sum_{i=1}^{n} \Omega_{ji}^{\rho(S)} x_i + B_j^{\rho(S)} \right) \right] + \beta^{\rho(S)} \tag{5}$$

where ω, Ω, β, and B are NN weights and biases. Derivatives (Jacobian matrix) shown in Fig. 2 as additional NN outputs are not actual outputs, which are trained during the NN training; they are calculated analytically through direct differentiating Eq. (5). The NN parameterization (4a) for the density is about two times faster then the UES. The calculation of the Jacobian matrix with the NN parameterization requires an additional time, which is about 70% of time required for the calculation of density. The NN parameterization (4b) for the salinity is several hundreds

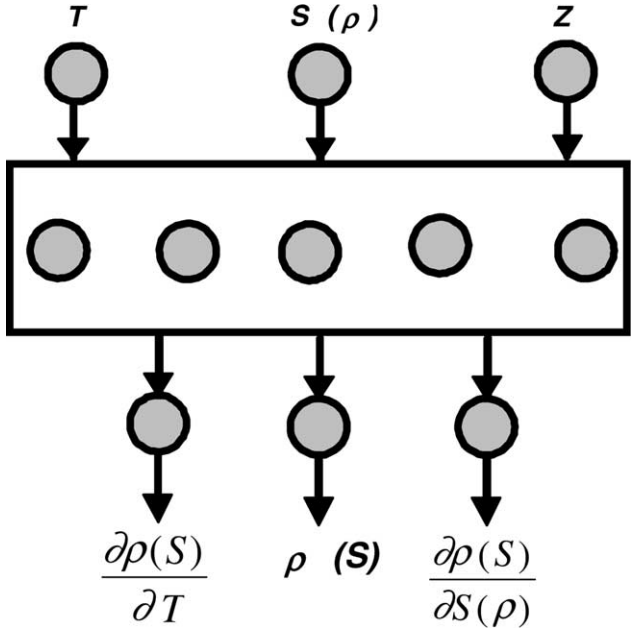

Fig. 2. Schematic representation of NN parameterizations for the density and salinity of seawater. Additional outputs for derivatives are not trained, they are calculated.

times faster than an iterative numerical inversion of the UES. In addition, the time required for the numerical inversion of the UES (rate of conversion of the iteration process) varies significantly. It strongly depends on inversion algorithm and on the choice of the initial approximation for the salinity.

To evaluate the accuracy of the NN approximation (4a) and (4b), 16,000 points were generated within the domain D on a grid, which did not include the training set points. The density of seawater calculated from the UES (2) was compared to that calculated from the NN, $\rho_{NN}(T,S,Z)$, using Eq. (4a). Table 1 shows several statistical measures of the differences (or errors) between the UES and the NN estimates for density. In terms of the bias and the RMS differences, the NN results for density clearly satisfy the criterion mentioned earlier; both the bias and the RMS values do not exceed the uncertainties indicated there and are <0.1 kg m^{-3}.

To evaluate the errors in using the NN approach to estimate the salinity, we used the same 16,000 points (ρ_i, T_i, S_i, Z_i) which were used to estimate the density. Initially, the NN for S_{NN} Eq. (4b) was applied to calculate a new salinity, s_i, using the corresponding values (T_i, ρ_i, Z_i). Then the differences $(S_i - s_i)$ were utilized to estimate the accuracy of the NN-derived salinities (first row of Table 2). To further evaluate the quality of the NN-derived salinities, the UES was applied again, this time to the triad (T_i, s_i, Z_i) to recalculate the density of seawater, ρ_i'. If the NN-obtained values for salinity were perfect, then the density, ρ_i', would be equal to ρ_i. The differences between these two values, $(\rho_i - \rho_i')$, were then used to further estimate the accuracy of the salinity-trained NN in terms of the density (second row of Table 2).

Table 2 shows that the NN estimates of salinity (4b) have an RMS error of 0.1 psu. In terms of the related error in density, this accuracy corresponds to an RMS error of 0.08 kg m^{-3}, which again does not exceed the uncertainties discussed earlier.

A substantial additional acceleration of calculations may be achieved by the use of differential increments of density, temperature, and salinity. Hence, we extend our approach to estimate these quantities also. Additionally, an efficient way of substantially reducing the computational burden is to replace the calculations of density per se by calculations of its total differential

$$\Delta\rho = \frac{\partial\rho}{\partial T}\Delta T + \frac{\partial\rho}{\partial S}\Delta S \tag{6}$$

Table 1
Minimum, maximum, and mean (i.e. the bias) errors (ϵ) and the RMS error, all expressed in kg m^{-3} ($\epsilon = \rho_{UES} - \rho_{NN}$)

Min ϵ	Max ϵ	Bias	RMS
−0.12	0.15	0.00	0.04

Table 2
Accuracies of the salinities estimated by the NN in terms of salinity and density

Units	Min error	Max error	Mean error	RMS error
psu	−0.33	0.85	0.00	0.10
kg m^{-3}	−0.27	0.71	0.00	0.08

Minimum, maximum, and mean errors together with the RMS errors are presented.

where ΔT and ΔS are increments of T and S, $\partial\rho/\partial T$, and $\partial\rho/\partial S$ are functions of T, S, and z. In this approach, after the density and its derivatives are calculated, Eq. (6) is used during several (usually several tens) steps of integration to estimate the new density. Then the density and its derivatives are recalculated, using the UES or NN approximation of the UES, to update the estimated values obtained using Eq. (6).

The density and the derivatives $\partial\rho/\partial T$ and $\partial\rho/\partial S$ can be accurately calculated from the NN ρ_{NN}, Eq. (4a) (Fig. 2). Thus, Eq. (6) can be reduced to

$$\Delta\rho = \frac{\partial\rho_{NN}}{\partial T}\Delta T + \frac{\partial\rho_{NN}}{\partial S}\Delta S \tag{7}$$

The estimation of the density using Eq. (6) requires several calculations of the UES (in that case the Jacobians are computed by centered finite differences). The estimation of the density using Eq. (7) requires one estimate of the NN (4a) and its derivatives, which is faster than Eq. (6). If $\partial\rho_{NN}/\partial z$ is also used in Eq. (7) for vertical integration, the gain in the speed of calculations due to the use of the NN increases.

Table 3 illustrates the accuracy of Jacobian calculations using the NN parameterization (4a) with three neurons in one hidden layer. All errors are estimated with respect to the Jacobian calculated explicitly, using the UES (2). The RMSN error is a relative error in percents with respect to the norm of the corresponding derivative. Errors in derivatives do not exceed 6%, which is acceptable for calculation of density.

The NN approach (7) is used in the high resolution Multiscale Ocean Forecast System (Chalikov, Rao, Rivin, Krasnopolsky, & Grumbine, 2002). This use of NN and its

Table 3
Accuracies of the Jacobian estimated by the NN in kg m^{-3} K^{-1}, kg m^{-3} psu^{-1}, and kg m^{-3} m^{-1}

Derivative	Min error	Max error	Mean error	RMS error	RMSN error
$\partial\rho_{NN}/\partial T$	−0.04	0.06	0.01	0.02	6%
$\partial\rho_{NN}/\partial S$	−0.02	0.02	0.003	0.006	0.8%
$\partial\rho_{NN}/\partial z$	−0.0001	0.0001	0.0001	0.0001	2%

Minimum, maximum, and mean errors together with the RMS errors are presented. RMSN error is the RMS error relative to the norm of the corresponding derivative.

derivatives has been shown to accelerate the density calculations about 10 times with the error in the density calculations not exceeding the natural uncertainty 0.1 kg m^{-3}. Therefore, the computational expense of calculating density has decreased from 40% to about 4–5% of the total time of integration as a result of using the NN approximation in the model.

3. Atmospheric applications: NNs for efficient calculation of infrared radiative fluxes

The next application that we present tackles the problematic tradeoff between accuracy and speed in the atmospheric infrared radiation computations. Transfer of energy by infrared radiation significantly contributes to the variations of atmospheric temperature. The infrared atmospheric spectrum encompasses a wide range of variability, from the slow-varying Planck function to the very detailed structures of the individual absorption bands. As a consequence, accurate modeling of the atmospheric radiative processes requires a high spectral resolution, which consumes a lot of computing time, and therefore can hardly be used for simulation of the atmosphere. Radiation computations also involve integrals over solid angle and altitude, that cannot be analytically solved and therefore significantly add to the computational burden. Two strategies are used, often simultaneously, to make the models affordable. On the one hand, statistical approaches have been developed that simplify the calculations of the three integrals (over solid angle, altitude and wave number) and of the impact of clouds (Goody & Yung, 1989). On the other hand, the computations are not performed at every time step and at every grid point of the atmospheric model (Morcrette, 2000). As an example, in the NCEP forecast model, infrared radiation variables are updated every 3 h only.

The computational efficiency is even more an issue in elaborate 4D variational (4D-Var) analysis schemes. These powerful data assimilation systems have been developed in operational weather centers, like ECMWF or MétéoFrance, to correct the atmospheric forecasts at regular times with the observations that have been received since the previous analysis: 4D-Var determines a statistically optimum forecast, given the initial forecast and its assumed error characteristics on the one side, and the observations with their error specifications on the other (Courtier, Thépaut, & Hollingsworth, 1994). To do this, perturbations of the atmosphere need to be propagated in time at every step of the minimization process with a linearized physics. Only computationally efficient parameterizations can be used for this purpose. For instance, in the operational 4D-Var physics at ECMWF, radiation perturbations are created by temperature changes only. Neither water vapor nor cloud evolution is taken into account (Mahfouf, 1999).

NN-based radiative transfer models (Escobar-Munoz, Chédin, Chéruy, & Scott, 1993; Faure, Isaka, & Guillemet, 2001; Key & Schweiger 1998; Schwander, Kaifel, Ruggaber, & Koepke, 2001) may be able to address these issues. In particular, the NeuroFlux approach (Chéruy, Chevallier, Scott, & Chédin, 1996; Chevallier, Chéruy, Scott, & Chédin, 1998b) has been successively tested in the Laboratoire de Météorologie Dynamique climate model (Chevallier, Chéruy, Li, & Scott, 1998a), in the ECMWF forecast model (Chevallier, Morcrette, Chéruy, & Scott, 2000b) and is currently being tested in the ECMWF 4D-Var system. This section describes the method, updates the results and summarizes the experience gained by this approach on which considerable effort has been invested.

NeuroFlux is mainly a NN-based version of the broadband radiation model of Morcrette (1991), hereafter EC-OPE, even though other models could be used in the training. EC-OPE was the operational code at ECMWF in the 1990s. The integration over wave number is performed using a band emissivity method in six spectral regions covering the long-wave spectrum. The transmission functions for water vapor and carbon dioxide are fitted using Padé approximants. Multi-layer gray bodies represent the clouds (Washington & Williamson, 1977).

NeuroFlux has been derived from EC-OPE using the same cloud representation and the multi-layer perceptron. Consistently with the former, upward and downward fluxes are computed in NeuroFlux as

$$F(P_i) = \sum_k a_k(P_i) \cdot F_k(P_i) \tag{8}$$

where P_i is the pressure level, F_k the infrared flux in the presence of a single layered black cloud in atmospheric layer k or the clear-sky flux (with the convention $k = 0$ for clear sky), and a_k is a weight. The a_ks are computed with a simple parameterization as a function of the layered cloud characteristics (cloud cover, liquid and ice water contents, particle size, etc.) and depend on the way cloudy layers overlap. In NeuroFlux, the F_ks are computed with artificial NNs with single hidden layers, whereas EC-OPE uses the above-mentioned band-emissivity method.

To summarize, NeuroFlux is made of a battery of specialized NNs (one for each atmospheric layer k and for each type of flux, upward or downward), the inputs of which include the temperature and gas (water vapor and ozone) profiles, the surface characteristics and the mean carbon dioxide concentration, whereas the cloud characteristics are processed by a separate parameterization. This way of doing reduces the dimension of the individual NNs, compared to a system where all computations would be performed by a single NN. With that design, NeuroFlux is about eight times faster than EC-OPE.

The accuracy of NeuroFlux has been assessed through code-by-code comparisons, climate simulations, and 10-day forecasts. Figs. 3 and 4 illustrate the performance of the version that fits the 50-level vertical resolution that was used

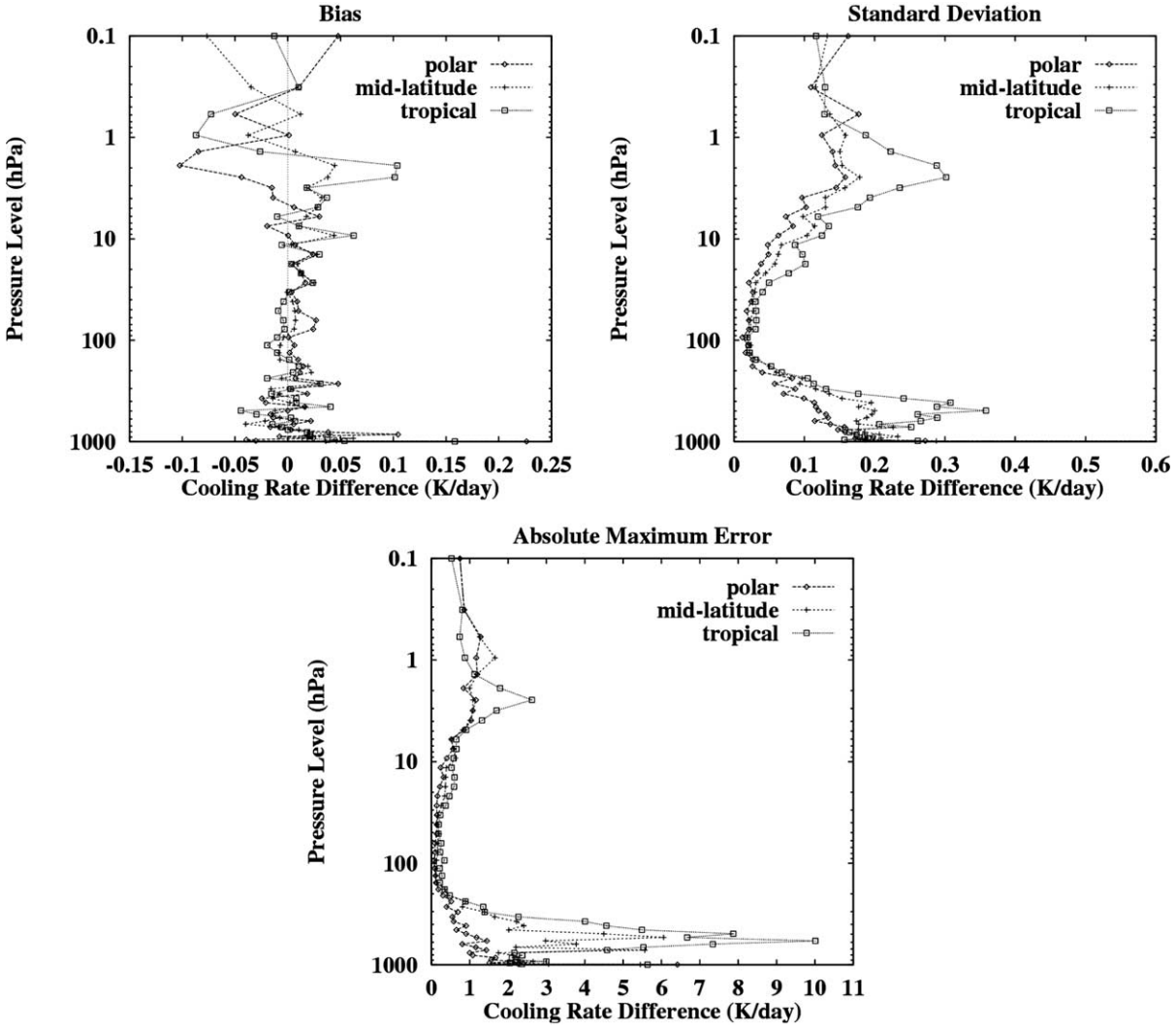

Fig. 3. Statistics of the differences between the computations of NeuroFlux and those of EC-OPE: cooling rates from NeuroFlux minus cooling rates from EC-OPE. ECMWF 6-hour forecasts, L50 T319 (about 60 km horizontal resolution). 1 February 1999, 00, 06, 12 and 18 UTC. The infrared cooling rates are the contribution of the infrared radiation to the variations of temperature over time. They are proportional to the derivative of the net fluxes with respect to pressure.

in the ECMWF operational forecast system in 1999. The accuracy of NeuroFlux is comparable to the accuracy of EC-OPE, with a neutral impact on the simulations. In particular the uncertainty introduced by NeuroFlux in the cloud cover simulations was shown to be much smaller than that one induced by the reduced temporal frequency of radiation computation in the ECMWF climate simulations (Chevallier et al., 2000b).

Even though the inputs of the NNs in NeuroFlux do not include the cloud profiles, some of the NNs reach sizes that are unusually large. For instance, the NN that compute the clear-sky upward fluxes in a 60-layer vertical grid contains about 200 inputs and 60 outputs. As a consequence of this huge variable space, the set up of the training datasets is particularly involving. It relies on the sampling of hundreds of thousands of atmospheric profiles with a simple topologic approach, as described by Chevallier, Chédin, Chéruy, and Morcrette (2000a). Similar work is done each time the vertical resolution is increased, because new levels are expected to provide original information about the profiles that cannot be obtained by a simple interpolation from the lower resolution datasets. Each version of the training datasets includes more than ten thousand profiles.

Despite the large number of synoptic weights in NeuroFlux, the partial derivatives of the computed fluxes with respect to atmospheric variables (i.e. the Jacobians) contain features that are considered not to be realistic (Chevallier & Mahfouf, 2001). The characteristics of the small noise of the computed fluxes (Fig. 3) show up in the flux Jacobians. This is different from the behavior of the smaller NNs presented in Section 2 (Table 3). Regularization techniques, such as that one proposed by Aires, Schmitt, Chédin, and Scott (1999), only partially improve the quality of the NN computation. Larger NNs may solve the problem, but would complicate the training even more and would slow down the code, making it less attractive. This issue of having correct sensitivities is very important for the application to 4D-Var where linearizations of

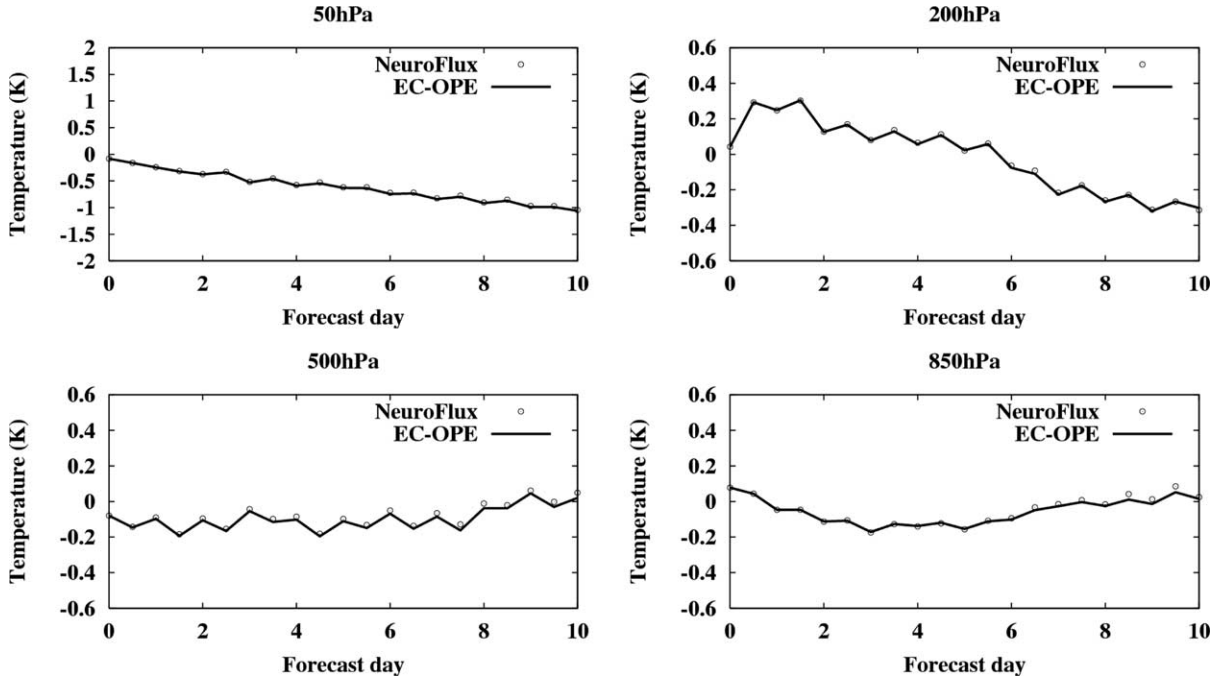

Fig. 4. L50 T319 (about 60 km horizontal resolution) simulations: forecast verification for temperature expressed in terms of mean temperature bias over the Northern Hemisphere (12 cases) The forecast system uses either EC-OPE (full lines) or NeuroFlux (circles).

the parameterizations are used. However, the approach of NeuroFlux allows for an elegant solution. Eq. (8) is differentiated as

$$\mathrm{d}F(P_i) = \sum_k a_k(P_i)\cdot \mathrm{d}F_k(P_i) + F_k(P_i)\cdot \mathrm{d}a_k(P_i) \tag{9}$$

Eq. (9) is used to calculate the flux perturbations. Its terms are determined as follows. The a_ks and the F_ks are obtained from NeuroFlux. A pre-computed mean Jacobian matrix, instead of the unsatisfactory NeuroFlux Jacobians, allows for a reasonably accurate estimation of the $\mathrm{d}F_k$s using a first-order Taylor approximation, since those partial fluxes are weakly nonlinear. Finally, the $\mathrm{d}a_k$s are obtained analytically from the tangent–linear version of the a_k model.

This approach is faster than the full tangent–linear model of NeuroFlux by almost a twofold factor, which makes it even more attractive. It is being evaluated as part of an improved version of the 4D-Var linearized physics at ECMWF, which introduces the treatment of the interaction between cloud and radiation (Janisková, Mahfouf, Morcrette, & Chevallier, 2003). This package includes NeuroFlux, a shortwave radiation model and a diagnostic cloud scheme, on top of the currently operational simplified and linearized physics. It is evaluated by studying the linearized time evolution of analysis increments with reference to the nonlinear computation using the full physics of the forecast model. An improvement (respectively a degradation) of the tangent–linear trajectory makes it closer to (respectively further away from) the nonlinear one. Fig. 5 illustrates the positive impact of the new physics on the wind increments after a 12 h integration, with a global improvement of 3.3%. Therefore the tangent–linear wind increments are more realistic when the new physical package is used. Preliminary results show a subsequent improvement of the forecast quality (Janisková et al., 2002).

4. Wave application: a NN approximation for nonlinear interactions in wind wave models

Ocean wind wave modeling for hindcast and forecast purposes has been at the center of interest of wave forecasters for many decades. Numerical prediction models are generally based on a form of the spectral energy or action balance equation

$$\frac{\mathrm{D}F}{\mathrm{D}t} = S_{\mathrm{in}} + S_{\mathrm{nl}} + S_{\mathrm{ds}} + S_{\mathrm{sw}} \tag{10}$$

where F is the spectrum, S_{in} the input source term, S_{nl} the nonlinear interaction source term, S_{ds} the dissipation or 'whitecapping' source term, and S_{sw} represents additional shallow water source terms. The JONSWAP study (Hasselmann et al., 1973) identified the active role of the nonlinear interactions in wave growth. The SWAMP study (SWAMP Group, 1985) then identified the need for explicit modeling of S_{nl} in wave models. State-of-the-art or so-called third generation wave models therefore explicitly model this source term.

In its full form (Hasselmann & Hasselmann, 1985), the calculation of the interactions S_{nl} requires the integration of

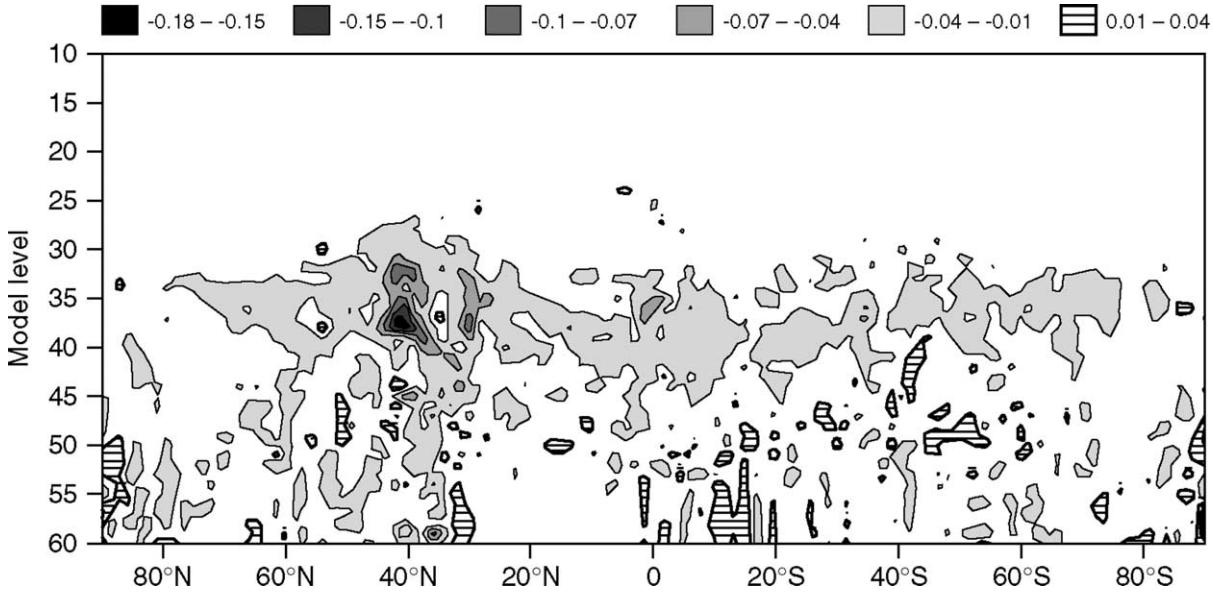

Fig. 5. L60 T159 (about 125 km horizontal resolution). Zonal mean impact of the new linearized physics (with NeuroFlux) on the zonal component of the wind, in m s^{-1}, in a 12 h simulation starting on 15 March 2001 00 UTC. Negative (respectively positive) values indicate an improvement (respectively a degradation) compared to the previous linearized physics. Both linearized physics are evaluated with reference to the nonlinear run.

a 6D Bolzmann integral

$$S_{\mathrm{nl}}(\vec{k}_4) = T \otimes F(\vec{k}) = \omega_4 \int G(\vec{k}_1,\vec{k}_2,\vec{k}_3,\vec{k}_4)\cdot\delta(\vec{k}_1+\vec{k}_2-\vec{k}_3 - \vec{k}_4)\cdot\delta(\omega_1+\omega_2-\omega_3-\omega_4)\times[n_1\cdot n_3\cdot(n_4 - n_2) + n_2\cdot n_4\cdot(n_3-n_1)]\mathrm{d}\vec{k}_1\,\mathrm{d}\vec{k}_2\,\mathrm{d}\vec{k}_3 \quad (11)$$

$$n(\vec{k}) = \frac{F(\vec{k})}{\omega};\quad \omega^2 = g\cdot k\cdot\tanh(kh)$$

where the complicated coupling coefficient G contains moving singularities (Hasselmann, 1963). This integration requires roughly 10^3–10^4 times more computational effort than all other aspects of the wave model combined. Present operational constraints require that the computational effort for the estimation of S_{nl} should be of the same order of magnitude as the remainder of the wave model. This requirement was met with the development of the DIA (Hasselmann et al., 1985). The development of the DIA allowed for the successful development of the first third-generation wave model WAM (Komen et al., 1994; WAMDI Group, 1988). More than a decade of experience with the WAM model and its derivatives has identified shortcomings of the DIA. The DIA tends to unrealistically increase the directional width of spectra, has a systematic spurious impact on the shape of the spectrum near the spectral peak frequency, and has a much too strong signature at high frequencies. In present third generation wave models, these deficiencies can be countered at least in part by the dissipation source term S_{ds}, which is generally used for tuning the energy balance in the Eq. (10). Although this approach often gives good results, it is counterproductive, because it prohibits development of dissipation source terms based on solid physical considerations. With our increased understanding in the physics of wave generation and dissipation, this becomes an even bigger obstacle impeding further development of third-generation wave models.

Considering the above, it is of crucial importance for the development of third generation wave models to develop *an economical yet accurate approximation* for S_{nl}. Here, we explore a neural network interaction approximation (NNIA) to achieve this goal (Krasnopolsky et al., 2001; Krasnopolsky, Chalikov, & Tolman, 2003). NNs can be applied here because the nonlinear interaction (11) is essentially a nonlinear mapping (symbolically represented in Eq. (11) by T) which relates two vectors (2D fields in this case). Thus, the nonlinear interaction source term can be considered as a nonlinear mapping between a spectrum F and a source term S_{nl}

$$S_{\mathrm{nl}} = T(F), \quad (12)$$

where T is the exact nonlinear operator given by the full Bolzmann interaction integral (11) (Hasselmann & Hasselmann, 1985; Resio & Perrie, 1991). Discretization of S and F (as is necessary in any numerical approach) reduces Eq. (12) to continuous mapping of two vectors of finite dimensions. Modern high resolution wind wave models use discretization on a 2D grid which leads to dimensions of S and F vectors of order of $N > 600$ (Tolman, 1999). It seems unreasonable to develop a NN approximation of such a high dimensionality (more than

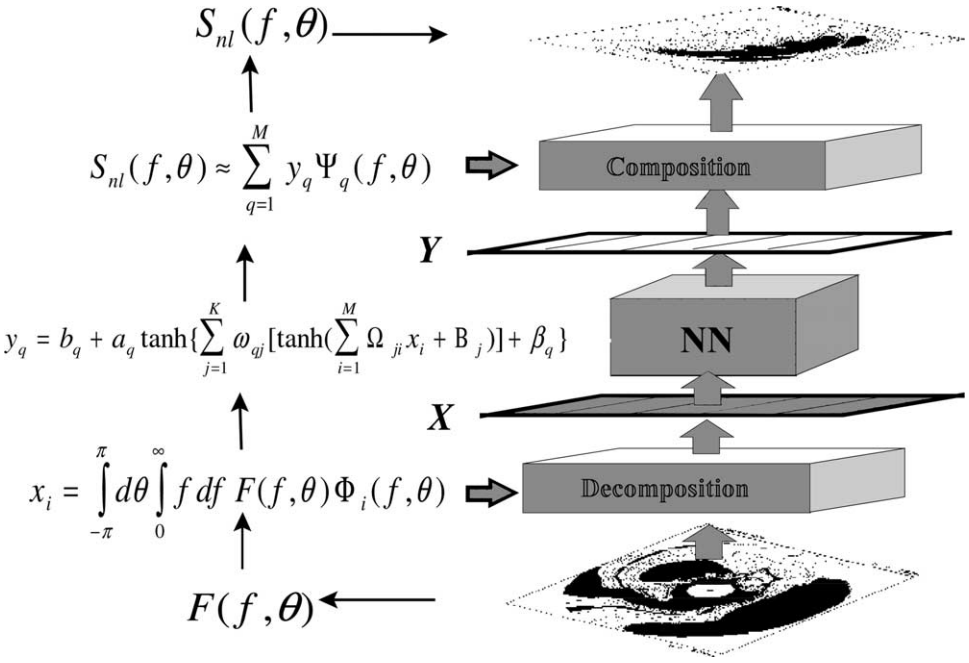

Fig. 6. Graphical representation of the NNIA algorithm.

600 inputs and outputs). Moreover, such a NN will be grid dependent.

In order to reduce the dimensionality of the NN and convert the mapping (12) to a continuous mapping of two finite vectors independent on the actual spectral discretization, the spectrum F and source function S_{nl} are expanded using systems of 2D functions each of which (Φ_i and Ψ_q) creates a complete and orthogonal 2D basis

$$F \approx \sum_{i=1}^{n} x_i \Phi_i, \qquad S_{\mathrm{nl}} \approx \sum_{q=1}^{m} y_q \Psi_q, \tag{13}$$

where for x_i and y_q we have

$$x_i = \int\int F \Phi_i, \qquad y_q = \int\int S_{nl} \Psi_q, \tag{14}$$

where the double integral identifies integration over the spectral space. Because both sets of basis functions $\{\Phi_i\}_{i=1,\ldots,n}$ and $\{\Psi_q\}_{q=1,\ldots,m}$ are complete, increasing n and m in Eq. (13) improves the accuracy of approximation, and any spectrum F and source function S_{nl} can be approximated by Eq. (13) with a required accuracy. Substituting Eq. (13) into Eq. (12) we can get

$$Y = T(X), \tag{15}$$

which represents a continuous mapping of the finite vectors $X \in \Re^n$ and $Y \in \Re^m$, and where T still represents the full nonlinear interaction operator. This operator can be approximated with a NN with n inputs and m outputs and k neurons in the hidden layer

$$Y = T_{\mathrm{NN}}(X). \tag{16}$$

The accuracy of this approximation (T_{NN}) is determined by k, and can generally be improved by increasing k.

To train the NN approximation T_{NN} of T, a training set has to be created that consists of pairs of vectors X and Y. To create this training set, a representative set of spectra F_p has to be generated with corresponding (exact) interactions $S_{\mathrm{nl},p}$ using Eq. (11). For each pair $(F, S_{\mathrm{nl}})_p$, the corresponding vectors $(X, Y)_p$ are determined using Eq. (14). These pairs of vectors are then used to train the NN to obtain T_{NN}.

After T_{NN} has been obtained by training, the resulting NN interaction approximation (NNIA) algorithm consists of three steps: (1) decompose the input spectrum, F, by applying Eq. (14) to calculate X; (2) estimate Y from X using Eq. (16); and compose the output source function, S_{nl}, from Y using Eq. (13).

A graphical representation of the NNIA algorithm is shown in Fig. 6. Development of an actual NNIA requires the following steps: (1) select basis functions Φ_i and Ψ_q and the number of each (n, m); (2) design a NN topology (number of neurons k); (3) construct a representative training set; and (4) select training strategies.

Table 4
RMSE statistics for 10,000 S_{nl}

	Mean RMSE	σ_{RMSE}	Max RMSE
DIA	0.0133	0.0111	0.104
NNIA	0.0068	0.0063	0.065

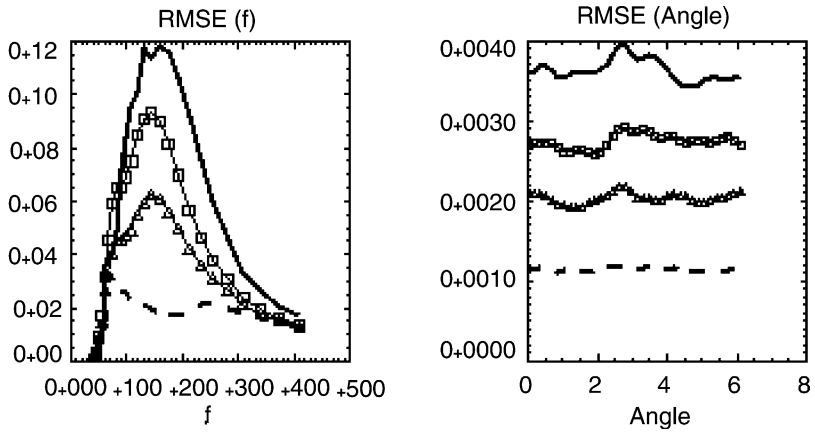

Fig. 7. RMSE as functions of frequency f and angle (averaged over entire test set). Dashed line-error of approximation (lower bound for all other errors). Solid line-DIA, line with squares-NNIA ($n = 51 : k = 20 : m = 64$), and line with triangles-NNIA (51:30:64).

The first three points all have a significant impact on both accuracy and economy of a NNIA. Unfortunately, there is no pre-defined way to tackle these issues as mentioned in Section 2. It is therefore unavoidable that the development of a NNIA involves multiple iterations. The first requirement for a NNIA to be potentially useful in operational wave modeling is that the exact interactions S_{nl} are closely reproduced for computational costs comparable to that of the DIA. The following shows the potential of this approach with the design of a simple ad hoc NNIA.

To address the basic feasibility of a NNIA, we have considered a NNIA to estimate the nonlinear interactions $S_{\mathrm{nl}}(f, \theta)$ as a function of frequency f and direction θ from the corresponding spectrum $F(f, \theta)$. Here we present the major results of this study to illustrate our approach (Krasnopolsky et al., 2001a). To train and test this NNIA, we used a set of about 20,000 simulated realistic spectra for $F(f, \theta)$, and the corresponding exact estimates of $S_{\mathrm{nl}}(f, \theta)$ (Van Veldder et al., 2000). Comparison of simulated spectra with spectra generated by the WAVEWATCH model (Tolman, 1999; Tolman & Chalikov, 1996) shows that the approach, which we use for simulating spectra, allowed us to simulate reasonably realistic and complicated spectra describing a broad range of wave systems. Separate data sets have been generated for training and validation.

As is common in parametric spectral descriptions, we choose separable basis functions where frequency and angular dependencies are separated. For Φ_i this implies:

$$\Phi_i(f, \theta) \Rightarrow \Phi_{ij} = \phi_{f,i}(f)\phi_{\theta,j}(\theta) \tag{17}$$

A similar separation is used for Ψ_q. Considering the strongly suppressed behavior of F and S_{nl} for $f \to 0$, and the quickly decreasing asymptotic behavior for $f \to \infty$, generalized Laguerre's polynomials (Abramowitz & Stegun, 1964) are used to define ϕ_f and ψ_f. Considering that no directional preferences exist in F and S_{nl}, Fourier decomposition is used for ϕ_θ and ψ_θ. The number of base functions is chosen to be $n = 51$ and $m = 64$ to keep the accuracy of approximation for F on average better than 2% and for S_{nl}—better than 5–6%. The number of hidden neurons was $k = 30$, which allows a satisfactory NN approximation of the mapping (15) using Eq. (16).

Table 4 compares three important statistics for the source function RMS errors (with respect to exact solution) calculated using DIA and NNIA for 10,000 spectra (independent validation set). The NNIA is nearly twice as accurate as DIA.

Fig. 7 shows mean RMSE as functions of the frequency f (left) and the angle θ (right). It also illustrates the improvement of the NNIA accuracy by increasing the number of neurons, k, in the hidden layer from 20 to 30. Numbers in Table 4 correspond to a NNIA with 30 neurons in the hidden layer (51:30:64).

And finally, Fig. 8 compares the DIA, NNIA, and exact algorithms in terms of the accuracy and computational efficiency. The current preliminary version of the NNIA algorithm is twice as accurate and only about five times slower than the DIA algorithm. In the current version of

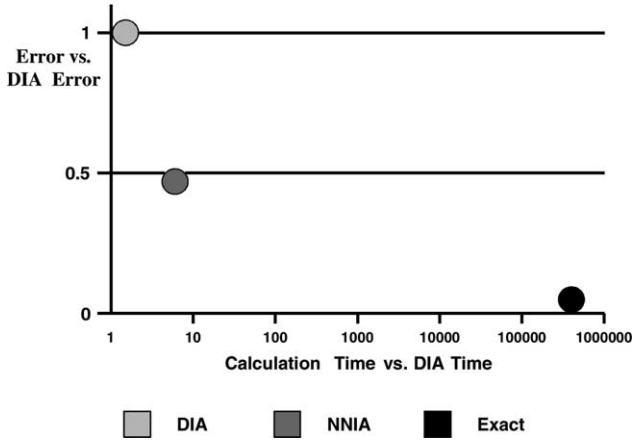

Fig. 8. Comparison of the accuracy and computational efficiency of the DIA, NNIA, and exact algorithms. The horizontal time scale is logarithmic.

the wind wave models, an algorithm that is up to 20 times slower than DIA can be accommodated; therefore, we still have enough room for further improving the accuracy of the NNIA. Considering that no optimization has yet been applied in the development of the NNIA composition and decomposition procedures, it appears reasonable to expect a final NNIA algorithm with computation requirements similar to DIA but with significantly higher accuracy.

5. Summary and discussion

In this paper we presented a recently emerged NN application developed by the authors for simplifying and accelerating time-consuming calculations in environmental numerical models using NN techniques. Parameterizations of physical, chemical, and biological processes, which occur at different scales, constitute an important class of such calculations. It is shown that, from a mathematical point of view, descriptions of such processes can usually be considered as continuous or almost continuous mappings (dependencies between two vectors). It is known that NNs are a generic tool for approximation of such mappings and, therefore, can be used for fast and accurate approximation of parameterizations of such processes. NNs can also easily provide analytical Jacobians. Because the NN Jacobian is computationally cheap, this approach is expected to be also very beneficial when used in 3D and especially in 4D variational data assimilation systems. We applied this approach to four specific problems associated with oceanic, atmospheric, and wave modeling.

The first and second applications considered in the paper deal with the oceanic equation of state, which is used for estimating the density and salinity of seawater in ocean circulation models. Separate NNs for density and salinity were developed using the UES as a basis. Although the estimation of density represents a forward problem, estimating salinity from the UES represents a complicated inverse problem, which has been very efficiently solved using the NN approach. The accuracy of the NN-generated densities and salinities were of the same order as those obtained directly from the UES itself. However, the time required to perform the calculations of density using the NN and the neural network Jacobian is several times less than that for UES. The time required for calculating salinity using the NN is several hundred times less than that required for the numerical inversion of the USE. Consequently, this approach has direct application to numerical ocean models where the equation of state must be estimated repeatedly. At NCEP, a NN equation for seawater density is currently used in an oceanic model.

In the third application, the NN approach was shown to successively handle the parameterization of infrared radiation in atmospheric models and to improve the tradeoff between speed and accuracy of such computation. In particular, the scheme described allows speeding up the computational time by a factor of eight compared to the reference model, while not affecting the quality of the atmosphere simulations. Further improvements of the method include the use of a more accurate reference model in the training phase, such as that currently operational at ECMWF (Morcrette, Mlawer, Iacono, & Clough, 2001). The high number of variables involved in this NN application made it necessary to develop original approaches, such as for the set-up of the training dataset. Also, the model computations are split into several modules, each one of them being parameterized by a specific NN. Despite this strategy, the NNs used are still very large, which affects the quality of the Jacobians, because it makes the training more difficult. Further increase of the vertical resolution, or the introduction of additional input or output variables, might further reduce the robustness of the model. This is an obvious limitation in the framework of the forecast models, which complexity constantly increases. However, the approach appears to be suitable for the variational assimilation, where other parts of the physics are very simplified and where the speed factor is crucial. Also, some other aspects of the radiation computation, such as the cloud horizontal heterogeneity, are crudely handled by forecast models and may benefit from NN parameterizations, as illustrated by Faure et al. (2001).

The fourth application deals with the nonlinear wave–wave interactions in wind wave models. A prototype of the NN approximation for this interaction is presented in this work. The NNIA calculations of S_{nl} are about five orders of magnitude faster than the exact computation. The NNIA calculations are twice more accurate than those from DIA (oversimplified approximation, which is currently used in the wind wave models) and require only 4–5 times more computational effort than the DIA calculations with less than 5% of this time spent in the actual NN part of the algorithm. Decomposition of the input spectra F and composing the source function S_{nl} from the NN output accounts for the rest. This decomposition (i) significantly reduces the size of the NN and (ii) makes the approach practically independent on the model grid and resolution.

These four applications illustrate the strengths and limitations of the NNs for the application to the fast simulation of environmental processes. In the case of retrievals, as discussed by Krasnopolsky (1997) and Krasnopolsky and Schiller (2003), NNs compete with other statistical methods, and usually perform better than those. This is because NNs are able to optimize the statistical link between the inputs and the outputs, provided that a proper (i.e. diverse and regularly spread) training dataset is gathered. In the present case, a sufficiently accurate physically based direct model is usually available. The NNs are likely to be faster, but significant speed gains can also be obtained with more powerful computers, making other parameterizations affordable. Also, the purely statistical formulation of the NNs often makes the refinements of their simulations, like additional inputs and outputs, or

more complex physics, more difficult than with an explicit physics.

The simulation of environmental processes may involve a large number of inputs (i.e. several hundreds), which make the NN too complex and complicates the training. For this complexity problem, two possible solutions were developed and illustrated: the input and output vectors may be projected on a basis (e.g. the NNIA application) or a battery of smaller NNs may be used (e.g. infrared radiation application).

Finally, a cheap computation of Jacobian is one of the advantages of the NN approach. Using this Jacobian in a combination with the tangent–linear approximation can additionally accelerate calculations (e.g. the seawater density application). However, since the Jacobian is not trained, it is simply calculated through direct differentiation of a trained NN. For large NNs (e.g. infrared radiation application) the accuracy of the NN Jacobian may not be sufficient for using with the tangent–linear approximation. Several solutions can be offered for this problem. First, the mean Jacobian can be calculated and used (e.g. infrared radiation application). Second, the Jacobian can be trained if included as actual additional outputs in the NN or if trained as a separate additional NN. This solution, however, leads to an increase of the NN complexity or to additional time for the Jacobian calculations. Third, regularization techniques can be used to stabilize the Jacobians (Aires et al., 1999). For instance, the error (or cost) function, which is minimized in the process of the NN training, can be modified to accommodate the Jacobian. Mathematically speaking, the Euclidian norm, which is usually used for calculating the error function, should be changed to the first order Sobolev norm. With such a change the NN is trained to approximate not only the function (as with the Euclidian norm) but also the function's first derivatives. Therefore, a new error function E can be expressed as a superposition of a standard error function E_0 and a Jacobian error function E_J

$$E_J = \sum_i^N \left(\frac{\partial F_{\mathrm{NN}}}{\partial x_i} - \frac{\partial F_t}{\partial x_i} \right)^2$$

where $\partial F_t/\partial x_i$ is the first derivative's training value calculated from the exact function to be approximated. This solution does not change the number of the NN outputs; however, it may require more hidden neurons and may significantly complicate the minimization during the training since the complexity of the error function increases. As a consequence, the Jacobian modeling for large NNs remains an open issue. This reflects the fact that one way of improving the computational efficiency of a direct model is to degrade the quality of the derivatives, while keeping the direct model within a given noise.

NNs obviously provide powerful solutions for the simulation of environmental processes, but like any other parameterization, their relevance needs to be regularly re-evaluated with respect to the particular computational and scientific contexts where they are developed and used.

Acknowledgements

We thank M. Janisková, at ECMWF for providing the results of Fig. 5. We also thank D. Chalikov, D.B. Rao, and H. Tolman, at NCEP for their help and support in developing oceanic and wave NN applications.

References

Abramowitz, M., & Stegun, I. A. (Eds.), (1964). *Handbook of mathematical functions with formulas, graphs and mathematical tables*. National Bureau of Standards.

Aires, F., Schmitt, M., Chédin, A., & Scott, N. A. (1999). The weight smoothing regularization of MLP for Jacobian stabilization. *IEEE Transactions on Neural Networks*, *10*, 1502–1510.

Apel, J. R. (1987). *Principles of ocean physics*. New York: Academic Press, p. 145.

Attali, J.-G., & Pagès, G. (1997). Approximations of functions by a multilayer perceptron: a new approach. *Neural Networks*, *6*, 1069–1081.

Chalikov, D., et al. (1998). Revisiting the question of assimilating temperature alone into a full equation of state ocean model. *Ocean Modeling*, *116*, 13–14.

Chalikov, D., Rao, D. B., Rivin, I., Krasnopolsky, V., & Grumbine, R. (2002). *A multi-scale ocean forecast system for applications to global and regional domains. NCEP/NOAA Technical Note, OMB contribution No. 194*, Washington DC, 130 pp.

Chen, T., & Chen, H. (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., & Chen, H. (1995b). Universal approximation to nonlinear operators by neural networks with arbitrary activation function and its application to dynamical systems. *Neural Networks*, *6*, 911–917.

Chéruy, F., Chevallier, F., Scott, N. A., & Chédin, A. (1996). A fast method using neural networks for computing the vertical distribution of the thermal component of the Earth radiative budget. *Comptes Rendus de l'Académie des Sciences de Paris 322*(S. IIb), 665–672. in French.

Chevallier, F., Chédin, A., Chéruy, F., & Morcrette, J.-J. (2000a). TIGR-like atmospheric profile databases for accurate radiative flux computation. *Quarterly Journal of the Royal Meteorological Society*, *126*, 761–776.

Chevallier, F., Chéruy, F., Li, Z.X., & Scott, N.A (1998a). A fast and accurate neural network-based computation of longwave radiative budget application in a GCM. *Proceedings of the American Meteorological Society Conference*, Paris, France, 25–29 Mai 1998 (pp. 690–693).

Chevallier, F., Chéruy, F., Scott, N. A., & Chédin, A. (1998b). A neural network approach for a fast and accurate computation of longwave radiative budget. *Journal of Applied Meteorology*, *37*, 1385–1397.

Chevallier, F., & Mahfouf, J.-F. (2001). Evaluation of the Jacobians of infrared radiation models for variational data assimilation. *Journal of Applied Meteorology*, *40*, 1445–1461.

Chevallier, F., Morcrette, J.-J., Chéruy, F., & Scott, N. A. (2000b). Use of a neural-network-based longwave radiative transfer scheme in the EMMWF atmospheric model. *Quarterly Journal of the Royal Meteorological Society*, *126*, 761–776.

Courtier, P., Thépaut, J.-N., & Hollingsworth, A. (1994). A strategy for operational implementation of 4D-Var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, *120*, 1367–1388.

Cybenko, G. (1989). Approximation by superposition of sigmoidal functions, in mathematics of control. *Signals and Systems*, *2*, 303–314.

Escobar-Munoz, J., Chédin, A., Chéruy, F., & Scott, N. A. (1993). Réseaux de neurones multi-couches pour la restitution de variables thermodynamiques atmosphériques è l'aide de sondeurs verticaux satellitaires. *Comptes Rendus de l'Académié des Sciences de Paris*, *317*, 911–918. in French.

Estrade, J.-F., Trémolet, Y., & Sela, J. (2001). Experiments with NCEP's Spectral Model. In W. Zwieflhofer, & N. Kreitz (Eds.), *Developments in Teracomputing—Proceedings of the Ninth ECMWF Workshop on the Use of High Performance Computing in Meteorology, Reading, UK, 13–17 November 2000* (pp. 92–99). World Scientific Publishing Co.

Faure, T., Isaka, H., & Guillemet, B. (2001). Mapping neural network computation of high-resolution radiant fluxes of inhomogeneous clouds. *Journal of Geophysical Research*, *106*(D14), 14961–14974.

Fofonoff, N. P., & Millard, R. C., Jr. (1983). *Algorithms for computation of fundamental properties of seawater*. *UNESCO technical paper in marine science 44*, UNESCO.

Funahashi, K. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, *2*, 183–192.

Goody, R. M., & Yung, Y. L. (1989). *Atmospheric radiation*. New York: Oxford University Press, 519 pp.

Hasselmann, K. (1963). On the non-linear energy transfer in a gravity-wave spectrum. Part 3: evaluation of the energy flux and swell–sea interaction for a Neumann spectrum. *Journal of Fluid Mechanics*, *15*, 385–399.

Hasselmann, S., & Hasselmann, K. (1985). Computations and parametrizations of the nonlinear energy transfer in a gravity wave spectrum. Part I: a new method for efficient computations of the exact nonlinear transfer integral. *Journal of Physical Oceanography*, 1369–1377.

Hasselmann, K., et al. (1973). Measurements of wind–wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP). *Ergänzungshelf zur Deutschen Hydrographischen Zeitschrift, Reihe A (8)*, *12*, 95.

Hasselmann, S., et al. (1985). Computations and parametrizations of the nonlinear energy transfer in a gravity wave spectrum. Part II: parametrization of the nonlinear transfer for application in wave models. *Journal of Physical Oceanography*, *15*, 1378–1391.

Janisková, M., Mahfouf, J.-F., Morcrette, J.-J., & Chevallier, F. (2002). Linearized radiation and cloud schemes in the ECMWF model: development and evaluation. *Quarterly Journal of the Royal Meteorological Society*, *128*, 1505–1528.

Key, J., & Schweiger, A. J. (1998). Tools for atmospheric radiative transfer: streamer and fluxnet. *Computer and Geosciences*, *24*, 443–451.

Komen, G. J., et al. (1994). *Dynamics and modelling of ocean wave*, Cambridge: Cambridge University Press, 532 pp.

Krasnopolsky, V. M. (1997). *Neural networks for standard and variational satellite retrievals*. *NCEP/NOAA Technical Note, OMB contribution No. 148*, Washington DC, 43 pp.

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

Krasnopolsky, V., Chalikov, D. V., & Tolman, H. L (2001). Using neural network for parameterization of nonlinear interactions in wind wave models, *International Joint Conference on Neural Networks*, July 15–19, 2001, Washington, DC (pp. 1421–1425).

Krasnopolsky, V. M., Chalikov, D. V., & Tolman, H. L. (2003). A neural network technique to improve computational efficiency of numerical oceanic models. *Ocean Modelling*, *4*, 363–383.

Krasnopolsky, V. M., & Schiller, H (2002). Some neural network applications in environmental sciences. Part I: forward and inverse problems in satellite remote sensing. *Neural Networks*, *16*, 321–334.

Mahfouf, J.-F. (1999). Influence of physical processes on the tangent–linear approximation. *Tellus*, *51A*, 147–166.

Morcrette, J.-J. (1991). Radiation and cloud radiative properties in the ECMWF operational weather forecast model. *Journal of Geophysical Research*, *96D*, 9121–9132.

Morcrette, J.-J. (2000). On the effects of the temporal and spatial sampling of radiation fields on the ECMWF forecasts and analyses. *Monthly Weather Review*, *128*, 876–887.

Morcrette, J.-J., Mlawer, E. J., Iacono, M. J., & Clough, S. A. (2001). Impact of the radiation-transfer scheme RRTM in the ECMWF forecasting system. *ECMWF Newsletter*, *91*, 2–9.

Resio, D. T., & Perrie, W. (1991). A numerical study of nonlinear energy fluxes due to wave–wave interactions, 1, methodology and basic results. *Journal of Fluid Mechechanics*, *223*, 603–629.

Schwander, H., Kaifel, A., Ruggaber, A., & Koepke, P. (2001). Spectral radiative transfer modeling with minimized computation time by use of neural-network technique. *Applied Optics*, *40*, 331–335.

SWAMP Group (1985). *Ocean wave modeling*. New York: Plenum Press, 256 pp.

Tolman, H. L (1999). *User manual and system documentation of WAVEWATCH-III version 1.18, NOAA/NWS/NCEP/OMB technical note 166*, 110 pp.

Tolman, H. L., & Chalikov, D. V. (1996). Source terms in a third-generation wind wave model. *Journal of Physical Oceanography*, *26*, 2497–2518.

UNESCO (1981). *The practical salinity scale 1978 and the International Equation of State for Seawater 1980. Tenth Report of the Joint Panel on Oceanographic Tables and Standards, UNESCO Technical Papers in Marine Science No. 36*, Paris, France: UNESCO.

Van Veldder, et al (2000). Modelling of nonlinear quadruplet wave–wave interactions in operational coastal wave models, Abstract, accepted for presentation at ICCE 2000, Sydney.

WAMDI Group (1988). The WAM model—a third generation ocean wave prediction model. *Journal of Physical Oceanography*, *18*, 1775–1810.

Washington, W. M., & Williamson, D. L. (1977). A description of the NCAR GCMs. In J. Chang (Ed.), *General Circulation Models of the Atmosphere* (*Vol. 17*) (pp. 111–172). *Methods in Computational Physics*, New York: Academic Press.

Woodgate, R. A. (1998). Can we assimilate temperature data alone into a full equation of state model? *Ocean Modelling*, *114*, 4–5.