Dealing with Inhomogeneous Outputs and High Dimensionality of Neural Network Emulations of Model Physics in Numerical Climate and Weather Prediction Models

Vladimir M. Krasnopolsky, Stephen J. Lord, Shrinivas Moorthi, and Todd Spindler

***Abstract* — In this paper we discuss our pilot study where the NN emulation technique developed previously for emulating model radiation parameterizations was applied to the part of the NCEP GFS model physics, GBPHYS, that is complimentary to the radiation parameterization. The results of the study showed that not all outputs of GBPHYS are emulated uniformly well with the original emulation approach. Significant differences between the radiation parameterizations and GBPHYS block and challenges for the NN emulation approach due to these differences are demonstrated and discussed. Several approaches that allowed us to deal with the challenges and that can be used to compliment the NN emulation approach for dealing with entire model physics are also introduced.**

I. INTRODUCTION

General Circulation Models (GCM) are used for numerical climate simulations and weather predictions. Modern GCMs are state of the art, complex systems of computer codes run on modern supercomputers (e.g., [1]). Because of the complexity of the physical processes in climate and weather systems, the calculation of model physics takes usually a very significant part of the total model computations. Evidently, this percentage is model dependent; however, for example, full model radiation is the most time-consuming component of any GCMs [2-4]; it takes usually more than 50% of the total model calculation time. In the National Centers for Environmental Prediction (NCEP) Global Forecast System (GFS), full model radiation takes about 50 – 60% of the total time required for calculation of model physics and the remaining 50 – 40% of time is consumed by the calculation of the rest of the model physics, which includes the moisture block, land surface model, ice model, etc. In this paper we will call this part of model physics, which is complimentary to the radiation block, GBPHYS, using the naming convention of the NCEP GFS source code.

This work was supported by the NOAA CDEP-CTB Grant NA06OAR4310047.

All authors are with the Environmental Modeling Center of the National Centers for Environmental Prediction, Camp Springs, MD 20746 USA; Vladimir Krasnopolsky is also with Earth System Science Interdisciplinary Center, University of Maryland, MD, USA; (the corresponding author: V. Krasnopolsky, phone: 301-763-8000 x 7262; fax: 301-763-8545; e-mail: Vladimir.Krasnopolsky@ noaa.gov).

In our previous studies [5,6] we demonstrated that the neural network (NN) emulation approach can be successfully used to significantly (one to two orders of magnitude) speed up the calculations of the model radiation, providing a sufficient accuracy in decadal (about 50 years) climate simulations. After speeding up the radiation calculations, GBPHYS becomes the computational bottleneck in GFS. Since the NN emulation approach that we developed is a rather generic one; it can be used not only for emulating of the radiation physics but also for emulating GBPHYS.

In this paper we investigate possibilities of using the NN emulation technique for speeding up calculations of the non-radiation part of model physics, GBPHYS. We discuss results of our pilot study that demonstrate differences between the radiation and GBPHYS (e.g., higher dimensionality, non-homogeneity of outputs) and challenges for the NN emulation approach due to these differences. We also propose and discuss some methods of meeting these challenges.

In section 2, we briefly review the GBPHYS structure and discuss some differences between GBPHYS and radiation parameterizations. In section 3 we describe a NN emulation of GBPHYS developed as a pilot study and illustrate the challenges encountered. In section 4, we propose some approaches to meet the challenges. Conclusions are given in section 5.

II. GBPHYS STRUCTURE AND DIFFERENCES BETWEEN GBPHYS AND RADIATION PARAMETERIZATIONS

The model physics that is a part of the GCM that calculates physical processes in the atmosphere (e.g., the long and short wave atmospheric radiation, turbulence, convection and large scale precipitation processes, clouds, interactions with land and ocean processes, etc.) structurally is separated in two blocks in NCEP GFS. The structure of GFS is schematically shown in Fig.1. The first, radiation block consists of two radiation parameterization: short and long wave radiations. Each radiation parameterization describes one physical process: the long or short wave atmospheric radiation. For the radiation parameterizations all inputs and outputs are homogeneous, they are defined at each model grid point. For example, over the night time part of the globe all outputs of the short wave radiation are zeros. These two radiation parameterizations do not interact with each other and are calculated independently. We emulated them with two different emulating NNs, one NN for the short wave and another one for the long wave radiation [6].

For the GBPHYS block the situation is different. This part of the model physics consists of sub-blocks that describe various physical processes (see Fig.1). Each of these processes is described by a complex multidimensional parameterization or model. These parameterizations and models interact with each other (exchange information) by virtue of multiple internal variables, which are not inputs or outputs of the GBPHYS block.

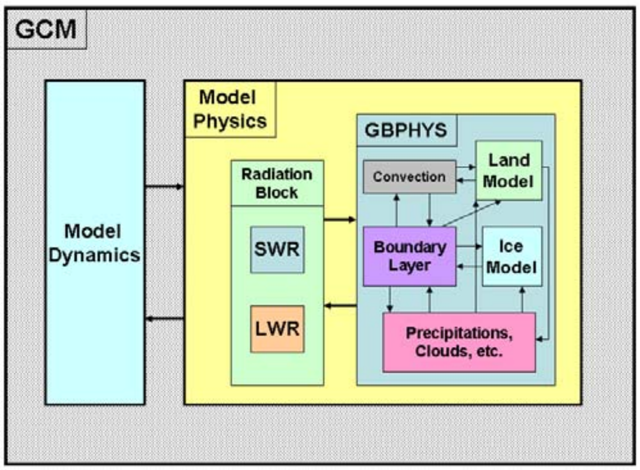

Fig. 1 The GCM structure.

The complex structure of GBPHYS leads to several significant differences as compared with the radiation block. Here we discuss a couple of the most important differences.

A. *Significantly Higher Dimensionality*

The GBPHYS block as well as the radiation parameterizations have two kinds of inputs and outputs (I/O). The first kind of I/O is composed of profiles. In the model physics, which is a 1-D parameterization of 3-D physical processes, profile represents a 1-D element of a 3-D field in a particular point of horizontal (e.g., latitude-longitude) grid. For example, in a GCM, atmospheric pressure, temperature, and humidity are 3-D fields, and in the model physics these variables are represented as vertical profiles (e.g., a dependence of the temperature on the vertical coordinate) at each latitude and longitude of horizontal grid. Each profile has many elements; the number of elements is equal to the number of the vertical layers (*nlay*) resolved in the model (*nlay* = 64 in GSF).

The second kind of I/O is composed of scalars. These scalars are elements of 2-D field in a particular point of horizontal (e.g., lat-lon) grid. The vegetation index, sea surface temperature (SST), and outgoing long wave upward flux at the top of the atmosphere are examples of such fields.

The dimensionality of the input and output vectors of radiation and GBPHYS blocks can be calculated as,

$$\mathbf{dim} = np \cdot nlay + ns \tag{1}$$

where **dim** is the dimensionality of the input or output vector, *np* is the number of input or output profiles, and *ns* is the number of scalar inputs or outputs. It is clear from (1) that profiles contribute to the dimensionality *nlay* (64 for GFS) times more than scalar variables.

One of the important differences between radiation parameterizations and the GBPHYS block is that the GBPHYS block has more inputs and outputs and, therefore, much higher dimensionality of input and output vectors. If for the long wave radiation the dimensionality of the input vector is up to 600 (8 profile plus about 10 scalars), for GBPHYS the dimensionality of the input vector is more than 1000 (17 profiles plus about 50 scalars). If for the long wave radiation the dimensionality of the output vector is 69 (one profile plus 5 scalars), for GBPHYS the dimensionality of the output vector is about 500 (7 profiles plus about 40 scalars).

As we will show in the next section, the dimensionality of the input and output vector of the part of the model physics that is emulated by NN is closely related to the number of NN emulation inputs and outputs, to the numerical complexity of emulating NN, and to the speedup of the model physics calculations provided by the NN emulation.

B. *Inhomogeneous Inputs and Outputs*

The inputs and outputs of radiation parameterizations are homogeneous; that is, all I/O at each particular location are defined simultaneously. Even for the short wave (solar) radiation that is calculated only over the day time part of the globe, the inputs and outputs over the night time of the globe are well defined and have physically meaningful values. All outputs have zero values over the night time part of the globe which is physically meaningful for the sun radiation. This value does not create a discontinuity in I?O at the day/night boundary.

For the GBPHYS block, the situation is different; not all I/O at each particular location are defined simultaneously. For example, land temperature is defined only over the land and set to zero over the ocean that creates discontinuity in I/O over the land/ocean boundary; similar problem exists for SST in that it is set to zero over the land.

III. Neural Network Emulation of GBPHYS: First Results and Challenges

The NN emulation technique [8] is based on the fact that the entire model physics as well as a single parameterizations of model physics may be considered mathematically as a continuous or almost continuous (like a step function) mapping between two vectors *X* (input vector) and *Y* (output vector) and symbolically can be written as:

$$Y = M(X); \quad X \in \Re^n, Y \in \Re^m \tag{2}$$

where *M* denotes the mapping, *n* is the dimensionality of the input space (the number of NN inputs), and *m* is the dimensionality of the output space (the number of NN outputs). Such a mapping can be approximated by NN (multilayer perceptrons, in our case) [7].

The simplest multi-layer perceptron (MLP) is a vector valued NN. It is composed of nonlinear neurons z_j and is an analytical approximation that uses a family of functions like:

$$y_q = a_{q0} + \sum_{j=1}^{k} a_{qj} \cdot z_j; \quad q = 1,2,\ldots,m \tag{3}$$

where $z_j = \phi(b_{j0} + \sum_{i=1}^{n} b_{ji} \cdot x_i)$ is a "neuron", x_i and y_q are components of the input and output vectors X and Y, respectively, and a and b are fitting parameters (or NN weights). The activation function ϕ is usually a hyperbolic tangent, n and m are the numbers of inputs and outputs, respectively, and k is the number of neurons in the hidden layer. The numerical complexity of the MLP, can be well approximated by a number of NN weights [8]:

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

For NN approximating a particular part of the model physics (e.g., radiation parameterization or GBPHYS), which can be considered as a mapping (2) with a given number of inputs n and outputs m, the number of hidden neurons k depends on the intricacy of the internal structure of the mapping. The NN numerical complexity N_C determines the time required for estimating NN (3). This time is directly proportional to N_C with the coefficient of proportionality depending mainly on hardware properties of the computer used. The numerical complexity of a NN emulation, N_C, can also be used as a measure of the complexity of the approximated mapping (1) or the emulated part of the model physics.

As can be concluded from the above, the I/O dimensionality **dim** (1) and the intricacy of the internal structure shown in Fig. 1 determines the complexity of different parts of the model physics and of NNs emulating these parts. Table 1 compares these complexities for radiation parameterizations and GBPHYS.

Table 1 Complexities of different parts of GFS model physics

	LWR	SWR	GBPHYS
N_C	33,294	45,173	237,719

Table 1 shows that the complexity of GBPHYS is almost an order of magnitude higher than the complexities of the radiation parameterizations. The reasons for such a high complexity of GBPHYS are higher I/O dimensionalities specified in section *2.A* and higher intricacy of the internal structure shown in Fig. 1.

The higher complexity of GBPHYS causes difficulties in the training of the emulating NN and limits the speedup of the model physics calculations achieved by the introduction of NN emulations. Despite these difficulties, in this pilot study we have used a straightforward approach used in our previous works with the radiation parameterizations [5,6] and developed a NN emulation for GBPHYS. A NN emulation was trained on a data set collected during one day of integration of GFS and validated on an independent data set collected during the another day of GFS integration.

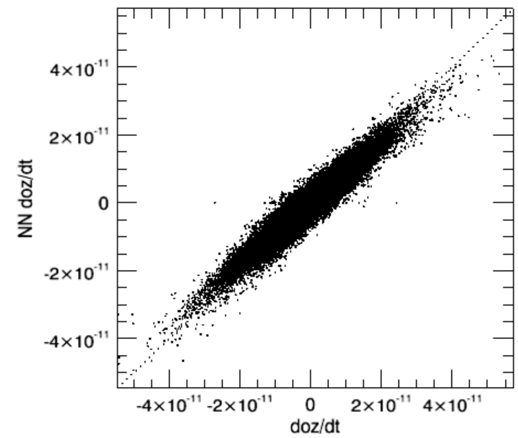

Fig. 2 Scatter plot (NN outputs vs. independent validation data) for ozone tendency (doz/dt); correlation coefficient CC = 0.95

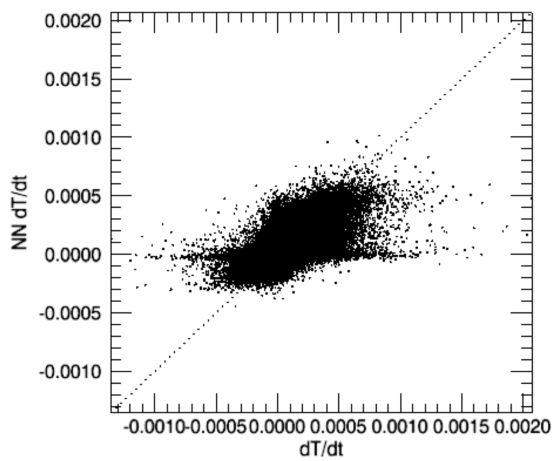

Fig. 3 Scatter plot (NN outputs vs. independent validation data) for temperature tendency (dT/dt); correlation coefficient CC = 0.75

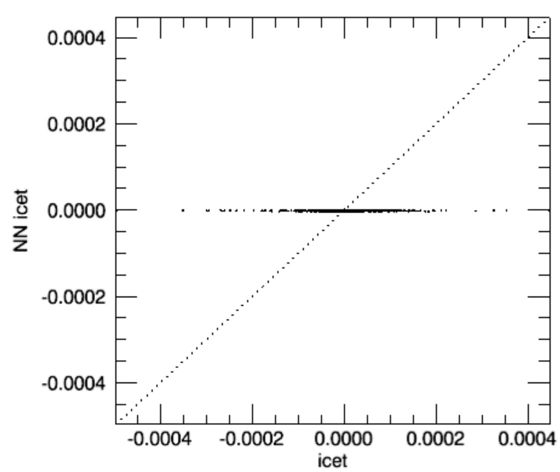

Fig. 4 Scatter plot (a NN output vs. independent validation data) for sea ice temperature

Figs. 2 to 4 show scatter plots for several outputs of NN emulating GBPHYS. Fig. 2 shows results for the tendency of the ozone concentration. These values are reproduced by the emulating NN with the high accuracy, the correlation between NN outputs and the validation data is very high. For the air temperature tendencies (Fig. 3) the accuracy is significantly lower, and for the sea ice temperatures (Fig. 4) there is no correlation between the NN outputs and validation data. The major reason for these striking differences between outputs is their non-homogeneity described earlier in this section. Many outputs related to surface parameters have discontinuities that negatively affect the NN training. Because the ozone is concentrated mainly in the upper layers of the atmosphere, the ozone outputs (Fig. 2) are affected by discontinuities at the surface in a minimal way. The air temperature at lower levels is affected by discontinuities in SST and in the land surface temperature more significantly. Multiple outliers in Fig. 3 are due to the lover level air temperature outputs. Finally, the sea ice temperature outputs that are defined over a small portion of the globe are completely confused by zero values presented to them over the rest of the globe during the NN training (Fig.4).

The aforementioned results show that high complexity and dimensionality of GBPHYS together with inhomogeneous outputs present challenges for the NN emulation approach and require significant modifications to the NN emulation approach. These modifications are considered in the next section.

IV. Modifications of NN emulation approach to emulate GBPHYS

In this section, we discuss possible modifications of the NN emulation approach to meet the challenges described in the previous section. We consider a couple of techniques that we developed to reduce complexity and dimensionality and to deal with the inhomogeneous outputs of NN emulations.

A. Reducing Dimensionality and Complexity of Emulating NNs

It was shown in the previous section that the complexity of the emulating NN (see Eq. (4)) is determined by dimensionalities of NN I/O and by dimensionality of the NN hidden layer (the number of hidden neurons, k). Thus, reducing any of these dimensionalities leads to a reducing of the NN complexity. Actually, the dimensionality of NN outputs cannot be reduced without reducing the effective vertical resolution of the model physics. The number of hidden neurons, k, is also defined more or less tightly by the required accuracy of approximation (the complexity of the I/O relationship). Thus, the input dimensionality is the only one that can be reduced using at least two different approaches.

1) Reducing dimensionality by sampling smooth profiles

Profile variables like pressure, temperature, humidity, etc. contribute mostly in the dimensionality of NN input because each profile variable adds up to 64 inputs to NN emulation. On the other hand, input profiles contain a lot of redundancy that, if properly identified, can be use to reduce the input dimensionality.

First, many input variables (e.g., pressure and all gases) have zero or constant values for upper vertical layers, for some gases the entire volume mixing ratio profile is constant (climatological values). These constant inputs were not used for NN training to improve the accuracy of the approximation. Constant inputs (zero or nonzero) do not contribute to the functional input/output relationship and should not be used for the development of NN emulations. Moreover, if they were used, they would introduce an additional noise (an approximation error). We have removed such inputs from the GBPHYS emulating NN; that reduced the input dimensionality by about 7%.

Second, some profiles depend on the vertical coordinate very smoothly. A profile of a variable is the dependence of the variable on the vertical coordinate discretized on a grid (64 vertical grid points in NCEP GFS). Thus, a profile is a vector composed of 64 components. Autocorrelation functions (ACF) of vertical profiles of some model variables are shown in Fig. 5.

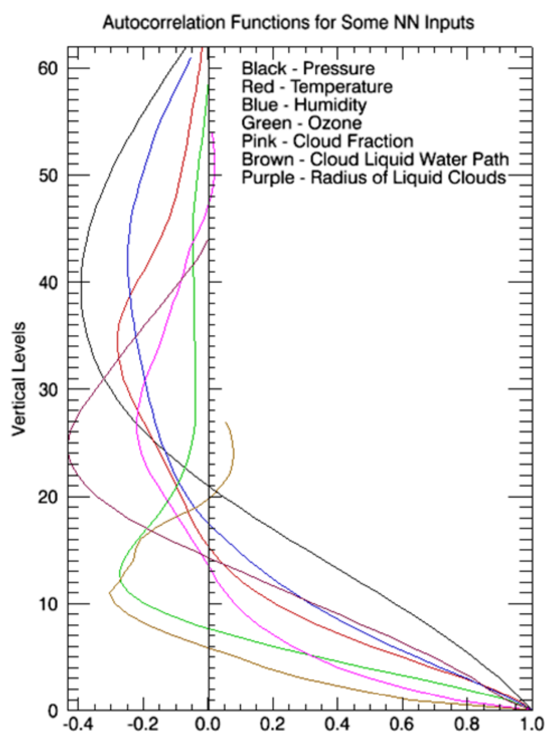

Fig. 5 Autocorrelation function for several NN input profiles

ACF of a profile shows the correlation between adjacent components of the profile (between values of the corresponding variables at the adjacent model levels). Slowly decreasing ACF (like those for pressure, temperature and humidity in Fig. 5) shows that the adjacent components of the profile are highly correlated and that redundant information is introduced if all of them are used as inputs for the emulating NN. For such profiles a sampling can be applied to reduce the redundancy and dimensionality of the NN inputs. For these profiles every other or even every third level can be selected as NN input. For some other profiles (e.g., cloud fraction and cloud liquid path; shown in pink and brown in Fig. 5) the corresponding ACFs decrease very quickly, which means that the redundancy for these variables is insignificant and the sampling should not be applied. We checked the sampling procedure using radiation parameterization as a test bed. This procedure allowed us to reduce the input dimensionality by an additional 7 – 10% without any significant reduction in the approximation accuracy.

2) Reducing dimensionality by removing dependent inputs

As was mentioned in the previous section the model physics and any of its parts can be considered as a mapping (2). The NN emulation approach uses the entire vector $X = \{x_1, x_2, \dots, x_n\}$ (2) as an input into the emulating NN. However, after closer consideration, we have found that some constituents of the input vector X are not independent, in a sense that they are functions of one or several other components of the input vector X. For example, if $x_3 = f(x_1, x_2, a)$, where a is a vector of constant parameters (e.g., a lookup table), then equation (2) can be transformed in such a way,

$$\begin{aligned} Y &= M(x_1, x_2, f(x_1, x_2), x_4, \dots, x_n) = \\ &Q(x_1, x_2, x_4, \dots, x_n) = Q(X'); \quad X' \in \Re^{n-1} \end{aligned} \tag{5}$$

where Q is the new mapping, and X' is the new input vector of reduced dimensionality (the component x_3 is omitted). Some cloud parameters and aerosols are examples of such dependent variables used for calculating model physics. Using radiation parameterizations as a test bed, we demonstrated that using such an approach the input dimensionality of the LWR NN emulation can be reduced by almost 50%. For the SWR NN emulation the efficiency of the approach is even higher; the input dimensionality could be reduced from about 3,200 to almost 500, by excluding 2,688 inputs that are calculated using humidity (which has been already included as an input) and time dependent 3-D look up tables. In the place of the removed variables, only three additional variables, time and horizontal coordinates, have been included in the reduced input vector X'.

B. Dealing with Inhomogeneous Outputs

The problem with inhomogeneous outputs emerges when, for example, a parameter that is defined over the land only (like land surface temperature) continues to be trained outside of the land. NN training is a nonlinear optimization process that minimizes an error function [8], using a training set composed of pairs of input and output vectors $\{\tilde{X}_i, \tilde{Y}_i\}_{i=1,\dots,N}$, where N is the number of input/output records in the training set. A regular error function can be written as,

$$E(w) = \sum_{p=1}^{N} \sum_{q=1}^{m} (\tilde{y}_q^i - y_q^i(\tilde{X}_i, w))^2 \tag{6}$$

where $\tilde{y}_q^i$ is a training value for the q-th NN output in the i-th training record, $\tilde{X}_i$ is the input vector in the i-th training record, y_q^i is the q-th NN output calculated for the i-th training input $\tilde{X}_i$ using eq. (3), and w is the vector of the NN weights that are trained.

In order to solve the problem with inhomogeneous outputs, we introduce a modified error function,

$$\overline{E}(w) = \sum_{p=1}^{N} \sum_{q=1}^{m} \alpha_{iq} \cdot (\tilde{y}_q^i - y_q^i(\tilde{X}_i, w))^2 \tag{7}$$

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

$$\alpha_{iq} = \begin{cases} 1, & \text{if the output number } q \text{ is defined} \\ 0, & \text{if not} \end{cases}$$

Using the modified error function (7) allows us to train each particular NN output only where and when it is defined and exclude it from the training when and where it is not defined.

V. Conclusions and Discussion

In our pilot study we applied the NN emulation technique developed previously for emulating model radiation parameterizations to the part of the NCEP GFS model physics, GBPHYS, that is complimentary to the radiation parameterization, The results of the study showed that the original emulation approach does not work uniformly well in emulating all the outputs of GBPHYS. Moreover, it demonstrated significant differences between the radiation parameterizations and the GBPHYS block.

In this paper we discussed two major differences: (1) GBPHYS has significantly higher complexity and dimensionality than radiation parameterizations and (2) it has inhomogeneous inputs and outputs. We also discussed challenges to the NN emulation approach due to these differences and introduced several approaches that allowed us to deal with the challenges and that can be used to compliment the NN emulation approach for dealing with the entire suite of model physics.

We introduced several approaches that help to reduce dimensionality of the NN input vector and, therefore, the complexity of the NN emulation. These approaches include a sampling of smooth input profiles and removing redundant (related or dependent) inputs. We also introduced an approach that allowed us to deal with inhomogeneous outputs. This approach is based on using a modified error function that lets us train a particular NN output only on those records of the training set where this output is defined.

These complimenting approaches have been tested using radiation parameterizations as a test beds. They demonstrated their efficiency, and our next step will be to apply the modified NN emulation approach to GBPHYS.

It is noteworthy that the modified NN emulation approach, which we introduced in this paper, preserves the major feature of our original emulation approach: the entire GBPHYS block is emulated by a single NN. Alternative approach that uses multiple NNs (a battery of NNs), each of which emulates a constituent of GBPHYS, such as land model, boundary layer, convection, etc. (see Fig.1), is also possible. In such an approach the complexity (size) of each constituent NN will be lower than the complexity of a single GSPHYS NN emulation. This simplifies the development of each constituent NN. However, such an approach requires training and validation of multiple NNs, including creating multiple training and validation data sets. Also, there are a lot of internal variables of the GBPHYS block linking the constituents (see Fig.1). In our approach, using a single emulation NN, they are not a part of I/O of the single emulating NN. In the alternative approach, using a battery of NNs, all these variables become I/O of the constituent NNs. As a result, the total complexity of the battery of NNs, especially at higher vertical resolutions, may become significantly higher than the complexity of the single emulating NN that we use [9]; thus, making the battery significantly slower that the single emulating NN. This discussion does not suggest that we completely exclude the multiple NN approach from consideration; however, we will consider it as a backup approach only if the single emulating NN approach cannot provide a sufficient accuracy for some outputs.

REFERENCES

[1] S. Saha, S. Nadiga, C. Thiaw, J. Wang, W. Wang, Q. Zhang, H. M. van den Dool, H.-L. Pan, S. Moorthi, D. Behringer, D. Stokes, M. Pena, S. Lord, G. White, W. Ebisuzaki, P. Peng, P. Xie, 2006: The NCEP Climate Forecast System. *Journal of Climate*, Vol. 19, No. 15, pages 3483.3517.

[2] J.-J. Morcrette, G. Mozdzynski and M. Leutbecher, 2008: A reduced radiation grid for the ECMWF Integrated Forecasting System, *Monthly Weather Review*, doi: 10.1175/2008MWR2590.1

[3] J.-J. Morcrette, P. Bechtold, A. Beljaars, A. Benedetti, A. Bonet, F. Doblas-Reyes, J. Hague, M. Hamrud, J. Haseler, J.W. Kaiser, M. Leutbecher, G. Mozdzynski, M. Razinger, D. Salmond, S. Serrar, M. Suttie, A. Tompkins, A. Untch and A. Weisheimer, 2007: Recent advances in radiation transfer parameterizations, ECMWF Technical Memorandum No. 539, October 18, 2007

[4] J. Manners, J.-C. Thelen, J. Petch, P. Hill & J.M. Edwards, 2008: Two fast radiative transfer methods to improve the temporal sampling of clouds in NWP and climate models, *Q. J. R. Meteorol. Soc.* 11 pp., in press.

[5] V.M. Krasnopolsky, M.S. Fox-Rabinovitz, and D.V. Chalikov, 2005: "New Approach to Calculation of Atmospheric Model Physics: Accurate and Fast Neural Network Emulation of Long Wave Radiation in a Climate Model", *Monthly Weather Review*, v. 133, No. 5, pp. 1370-1383.

[6] V. M. Krasnopolsky, M.S. Fox-Rabinovitz, and A. A. Belochitski, 2008: "Decadal Climate Simulations Using Accurate and Fast Neural Network Emulation of Full, Long- and Short Wave, Radiation.", *Monthly Weather Review,* 136*,* 3683–3695, *doi: 10.1175/2008MWR2385.1*.

[7] K. Funahashi, 1989: “On the Approximate Realization of Continuous Mappings by Neural Networks”. *Neural Networks*, 2, pp. 183-192

[8] V.M. Krasnopolsky, 2007: “Neural Network Emulations for Complex Multidimensional Geophysical Mappings: Applications of Neural Network Techniques to Atmospheric and Oceanic Satellite Retrievals and Numerical Modeling”, *Reviews of Geophysics*, 45, RG3009, doi:10.1029/2006RG000200.

[9] V.M. Krasnopolsky and M.S. Fox-Rabinovitz, 2006: “Complex Hybrid Models Combining Deterministic and Machine Learning Components for Numerical Climate Modeling and Weather Prediction”, *Neural Networks,* 19, pp. 122-134

MMAB Contribution No. 222