Development of Neural Network Convection Parameterizations for Numerical Climate and Weather Prediction Models using Cloud Resolving Model Simulations

Vladimir M. Krasnopolsky, *Member, INNS,* Michael S. Fox-Rabinovitz, and Alexei A. Belochitski

***Abstract* — A novel approach based on the neural network (NN) technique is formulated and used for development of a NN ensemble stochastic convection parameterization for numerical climate and weather prediction models. This fast parameterization is built based on data from Cloud Resolving Model (CRM) simulations initialized with TOGA-COARE data. CRM emulated data are averaged and projected onto the General Circulation Model (GCM) space of atmospheric states to implicitly define a stochastic convection parameterization. This parameterization is comprised as an ensemble of neural networks. The developed NNs are trained and tested. The inherent uncertainty of the stochastic convection parameterization derived in such a way is estimated. The major challenges of development of stochastic NN parameterizations are discussed based on our initial results.**

I. Introduction

Clouds and convection are among the most important and complex phenomena of the Earth's physical climate system. The processes that control clouds, and through which they interact with other components of the Earth system involve slow and fast fluid motions carrying heat, moisture, momentum and trace constituents, and influence other important physical processes through phase changes of water substances, radiative transfer, chemistry, production and removal of trace constituents, and atmospheric electricity.

In spite of intense studies for centuries, clouds still provide an intellectual and computational challenge. Because of the vast range of time and space scales involved, researchers typically focus on a particular component of a cloud system, with a narrow range of time and space scales, and prescribe features of the cloud that operate outside of that range. For example, 'box' models treat a small air parcel as an approximately spatially homogeneous medium over length scales of order a meter to explore the evolution of a cloud drop spectrum. Motions of the air as a fluid are. prescribed, and the focus is on molecular scale motions (e.g vapor deposition) and drop scale motions (e.g. drop coagulation). At the other end of the spectrum of representations of clouds is their representation in large scale models, for example, in General Circulation Models (GCMs). GCMs treat convective clouds very simply.

Parameterizations must represent the effect of clouds on time and space scales that are well below the resolution of the scales explicitly treated in GCMs, which resolve atmospheric features with space scales of order 100km, and time scales of order 10 minutes. Numerical Weather Prediction (NWP) models typically operate at smaller spatial and temporal scales, but most cloud processes are still acting well below these scales. For this reason, models produce independent, but connected 'parameterizations' (representations) of each of these processes.

In this paper we focus on improving the representation (parameterization) of convective clouds. Convective clouds are loci for some of the most vigorous vertical motions in the atmosphere, transporting heat, momentum, moisture, and trace constituents from near the surface to 10s of km in altitude in minutes. They are also regions where much of the rain hitting the earth's surface is produced, and the local origin or source of water that appears in stratiform clouds like 'cirrus anvils'.

The representation of cloud processes has long been recognized as a challenge at any of the space and time scales mentioned above, and the problem is particularly difficult for global models [1, 2]. The scientific community realizes that many critical aspects of our ability to represent the atmosphere for climate and NWP problems are being hindered by the representation of clouds, and that we have reached an impasse in our ability to improve these processes. These issues are eloquently discussed in [3]. That paper makes the point that models that explicitly resolve processes at the smaller time and space scales that are relevant to many features of clouds systems (e.g. 10s of meters to less than 10 kilometers, and time scales of seconds to minutes), so called Cloud Resolving Models (CRM) also called Cloud System Resolving Models (CSRMs) or Large Eddy Simulation Models (LES), are usually (but not invariably) able to simulate component aspects and evolution of the cloud systems much more realistically than

Manuscript received January 29, 2010. This work was supported by the U.S. Department of Energy under the DOE ARM grant DEFG0208ER64606 and the computer resources were provided by DOE NERSC and NCAR CISL.

V. M. Krasnopolsky is with the National Centers for Environmental Prediction, NOAA & with the Earth System Sciences Interdisciplinary Center, University of Maryland, Camp Spring, MD 20746, USA (phone: 301-763-8000x7262; fax: 301-763-8545; e-mail: Vladimir.Krasnopolsky@noaa.gov).

M. S. Fox-Rabinovitz is with the Earth System Sciences Interdisciplinary Center, University of Maryland (e-mail: foxrab@essic.umd.edu and alexei@essic.umd.edu).

A. A. Belochitski is with the Earth System Sciences Interdisciplinary Center, University of Maryland (e-mail: foxrab@essic.umd.edu and alexei@essic.umd.edu).

large scale models. The improvements seen in representing clouds using CSRMs and LES have provided the motivation for the approaches described in section 2.

A. Current efforts by others to improve the treatment of clouds in large scale models

Because of the problems described above, the research community has been exploring alternate ways of making progress in representing clouds in global models. Among the alternate paradigms for treating clouds in global models are the following:

1. Cloud System Resolving Models (CSRMs). These models, first developed in the 1970s and 1980s [4], operate over a limited area (typically continental scale or smaller) at finer spatial and temporal time scales than a global model (although they still require many simplifications). The spatial and temporal average behavior of these models tells us what might be gained if we could resolve many of the phenomena that global models must ignore (e.g. higher resolution fluid dynamic motions that can resolve some updrafts and downdrafts, convective organization, meso-scale circulations, and stratiform and convective components that interact with each other, etc.). It is important to note that CSRMs do not simulate all processes important to cloud systems from first principles. There are still many sets of processes (for example microphysical and turbulent processes) that are still treated crudely, but they do resolve many more phenomena than today's global model parameterizations.

2. Recently attempts have been made to develop global models that resolve some cloud scale motions (down to horizontal resolutions of about 3km) (Global Cloud Resolving Models or GCRMs, e.g. [5,6]). They are incredibly expensive to run when compared with the cost of a typical general circulation model (see next item), and have generally been used in idealized settings for exploratory experiments (in an water covered planet scenario for example)

3. The term 'super-parameterization', also known as a 'Multiscale Modeling Framework' (MMF), was originally suggested in [7], and subsequently developed by a group at Colorado State University [8,9]. This concept refers to the embedding a simplified CSRM into each column of a global model. Because of the simplifications, such a model is substantially less expensive (by a factor of 10^3-10^4) than a GCRM, but it is still enormously more costly (10^2 -10^3) than models using a conventional convective parameterization [9]. A typical MMF parameterization might use a two-dimensional CSRM with 64 sub-columns within each GCM column, to produce sub-resolution of about 4km. The simplifications buy a substantial reduction in cost, with a consequent compromise in accuracy of physical representation. There are for example, significant changes in the storm evolution (e.g. vertical velocities) produced in 2-D CSRMs when compared with 3-D formulations, and 2D CSRM simulate too rapid a transition from shallow to deep convection and too much cloud cover [10].

In this paper we introduce an alternative approach based on the neural network (NN) technique. This approach allows us to develop a NN convection parameterization, which can be used as a parameterization in GCM (in our case, in NCAR CAM) and can effectively take into account major sub-grid scale effects taken into account by other approaches (see above) at a fraction of the computational cost. In Section 2 we introduce our approach, discuss sources of uncertainties in the convection parameterization and estimate these uncertainties. We also describe the NN training and perform an initial evaluation of the developed NN convection parameterization. Section 3 contains conclusions.

II. Formulation of our Approach: Development of NN ensemble convection parameterization from CRM data

Our goal is to develop NNs which emulate the behavior of a CSRM/CRM simulated data at larger scales (closer to GCM scales) for a variety of regimes and initial conditions. The resulting emulation can be used as a novel, and computationally viable parameterization. If successful, it will produce a parameterization of similar or better quality to the superparameterization or MMF, effectively taking into account sub-grid scale (in terms of GCM) effects at a fraction of the computational cost.

As we showed in our previous works (e.g., [11]) any parameterization of model physics can be emulated using NNs. NN is an analytical approximation that uses a family of functions like:

$$y_q = a_{q0} + \sum_{j=1}^{k} a_{qj} \cdot \phi(b_{j0} + \sum_{i=1}^{n} b_{ji} \cdot x_i); \quad q = 1,2,\ldots,m \qquad (1)$$

where xi and yq are components of the input and output vectors X and Y, respectively, a and b are fitting parameters, and $\phi(b_{j0} + \sum_{i=1}^{n} b_{ji} \cdot x_i)$ is a "neuron". The activation ϕ function is usually a hyperbolic tangent, n and m are the numbers of inputs and outputs respectively, and k is the number of neurons in (1).

A. Design and development of the NN parameterization and the training set

Fig. 1 summarizes the process of development of the NN parameterization. The CRM used in our work is the SAM (System for Atmospheric Modeling) developed by M. Khairoutdinov and D. Randall and their collaborators [12]. It has been used for our CRM simulations. SAM uses TOGA-COARE data (ARM or other observations) for initialization and forcing and has the horizontal resolution of about 1 km, 64 or 96 vertical layers, and time integration step of 5 s. We drive the CRM over a domain of 256x256km.

The development of NN parameterization is a multi-step process. These steps are:

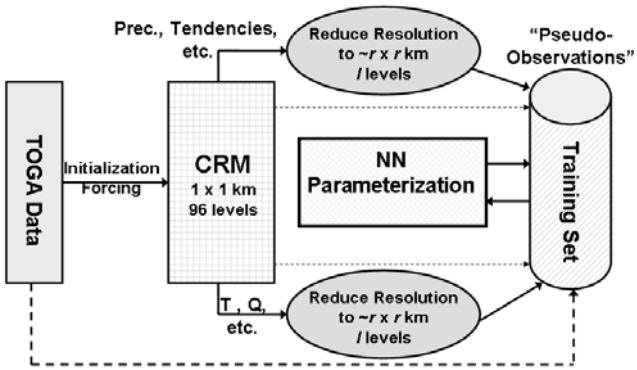

Fig. 1 Development of the NN Parameterization.

1. Simulating CRM data. The model is run for a number of days (120 days in our current experiments) and the high resolution output of the model is archived.
2. Reducing the resolution of simulated data. The high resolution CRM simulated data are averaged in space and time. The data are averaged to a reduced horizontal resolution of 1 km < $r \leq R$, where R is the GCM resolution, and are interpolated/averaged to the number of vertical layers $l = L$, where L is the number of vertical layers in GCM.
3. Projecting a CRM space of atmospheric states to a GCM space of atmospheric states. From the reduced resolution CRM simulated data created at the previous step, the subset of variables is selected and this subset constitutes the NN development set. Only variables that can be identified with corresponding GCM variables or can be calculated from or converted to prognostic or diagnostic variables available in GCM, are included in the development set (called "pseudo-observations" in Fig.1). Only these variables are used as inputs and outputs of our NN parameterization. The development set implicitly represents a stochastic convection parameterization with an uncertainty, which is an inherent feature of such a parameterization (see Section 2.2). The pseudo-observations are separated into the training and test/validation sets. The dashed lines in Fig. 1 show that, if it is found to be desirable, the high resolution CRM simulated data and/or even observed data can be added to the development set to enrich sub-grid scale variability in the development data.
4. The NN parameterization is trained using the training set. Due to the inherent uncertainty of pseudo-observations, it is implemented as an ensemble of NNs.

B. Parameterization and its Uncertainties

The first three steps of the development process formulated in the previous sections introduce uncertainties in the data set of "pseudo-observations". The uncertainties are introduced at each of these steps and their sources can be traced step by step. First, the CRM may be formally considered as a mapping $\widetilde{M}_{CRM}$, that defines the relationship between two vectors: the input vector (x) and the output vector (y) that are composed of CRM variables. At each time step the mapping $\widetilde{M}_{CRM}$ given vector x produces vector y or,

$$y = \widetilde{M}_{CRM}(x) \tag{2}$$

The first step in our developmental process consists of **applying the mapping** (2) at each time step of the CRM simulation and continuing with the simulation for a period of time T. The simulation is initialized and at each time step forced by observed data; however, because the CRM physics is partially parameterized and contains a number of simplifications, the simulated will deviate from the observed data. This difference between the observed reality and the "CRM reality" is the first contribution into the uncertainty of the NN parameterization.

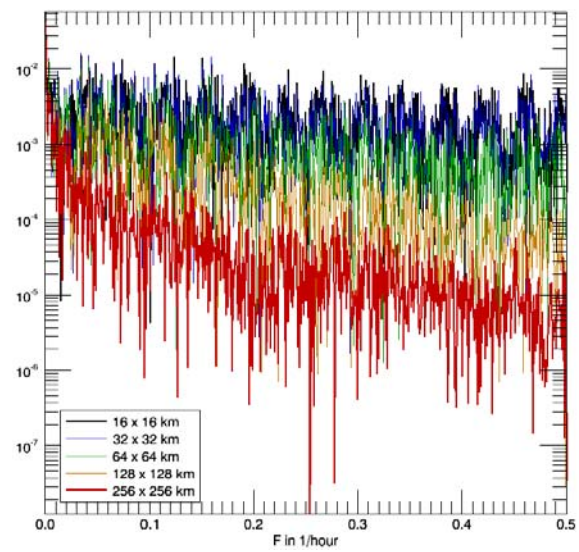

Fig.2 Spectra of horizontally averaged data for the precipitation rate for different averaging areas ($r \times r$) with r = 16, 32, 64, 128, and 256 km

The second step in our development process consists of **averaging** high resolution simulated data (x, y) over some area r (1 km < r < R) and over a time interval t ($\tau < t < T$), where τ = 5s is the CRM integration time step. As a result, averaged vectors of simulated data are produced. Fig. 2 shows the evolution of the spectrum of the precipitation rate with the changing the resolution of the field we are sampling. We display the sensitivity in precipitation to resolution changes over r from 16 to 256 km. As expected, Fig. 2 clearly shows the reduction of high frequency variability in averaged simulated data as compared with the original high resolution simulated data x and y.

The third step of our developmental process – **projecting the** CRM space of atmospheric states onto the GCM space of atmospheric states – consists of transition from averaged CRM variables to a subset of these variables, X and Y, with the subsequent averaging over the removed variables. The **new variables** X and Y **include only variables that can be identified with the corresponding GCM variables or can be calculated from or converted to prognostic or diagnostic variables available in the GCM**; all other CRM

simulated variables can not be used as not existing in GCM and therefore are omitted or projected out.

Both of the above operations – averaging and projecting – lead to a significant uncertainty in X and Y. Taking into account the uncertainty, and assuming the mapping between the new vectors of projected variables X and Y exists, it can be written as,

$$Y = M(X) + \varepsilon \tag{3}$$

This new mapping (3) is obtained from the mapping (2) by averaging and projecting out all omitted CRM variables. The mapping (3) is an approximate one, which is reflected by introducing the uncertainty vector ε. This uncertainty emerges due to the averaging of CRM data and due to an unaccounted variability of omitted parameters. After projection of these vectors onto the GCM space, *the projected vectors X and Y do not correspond to any particular values of omitted parameters; these values are uncertain.* Actually when we learn mapping M from data, the projected vectors X and Y correspond to the mean values of omitted parameters calculated over the training set.

Thus, *mapping (3) is thus not an exact mapping as is the mapping (2)*; it is a **stochastic mapping** between two random vector variables X and Y. Stating this we'd like to emphasize that *stochastic elements emerge in this problem before introducing the NN emulation technique*. Actually, **(3) is a stochastic parameterization**, which inherently contains the uncertainty ε**.** It is *implicitly defined by the training set (X, Y)***. The uncertainty in this case is not a destructive noise; it is an inherent informative part of the stochastic parameterization, which contains important statistical information about sub-grid scale** (in terms of GCM) **effects. Actually, the stochastic parameterization is a family of mappings distributed with a distribution function inside the range and with a shape of the distribution function determined by the uncertainty vector** ε **.**

C. NN Emulation of the Parameterization and Estimation of its Uncertainties

1) Data

A limited data set was simulated for the development due to the lack of longer observational data needed for driving/forcing SAM simulations. SAM/CRM using the TOGA-COARE forcing was ran for the 256 x 256 km domain with 1 km resolution and 96 vertical layers (0 – 28 km) for 120 days. Then it was averaged at every hour of model integration to produce a simulation data set with an effective resolution of 256 km. Finally, only variables that are available in GCM (NCAR CAM) or can be calculated there have been selected. The final data set consists of 2,800 records (every hour). The simulation dataset was partitioned into two parts: a training set – 2,240 records or 80% of data and a test set – 560 records or 20% of data. Namely, first 2,240 records are included in the training set and the last 560 records – in the test set.

These two data sets have been used for the NN training and test/validation. As was noticed in the previous section, these data implicitly represent a stochastic parameterization that inherently contains an uncertainty ε, which is not a useless noise. However, from the point of view of a single NN that is trained using the data, the data (both component X and Y of the data) contain a significant level of noise.

Symbolically, the NN emulation of the stochastic parameterization (3) can be written as,

$$Y = M_{NN}(X) + \varepsilon + \varepsilon_{app} \tag{4}$$

where M_{NN} is a NN emulation of the mapping M (3) and ε_{app} is a NN approximation error. Thus, in the case of the stochastic parameterization, the NN emulation task is different from that of emulating a original *deterministic* parameterization in GCM (e. g. [13, 14]). The simulated data that represent a deterministic parameterization do not contain a noise (here we are talking about the noise of the magnitude significantly higher than round off errors).

This important difference should be taken into account when the NN approximation is trained, the approximation error statistics are analyzed and interpreted, and the NN architecture is selected. For example, **in the case of training the usually used criterion of minimum of the root mean square error should be substituted by the requirement that the root mean square error should not exceed the uncertainty** ε **or,**

$$\frac{1}{N}\sum_{i=1}^{N}[Y_i - M_{NN}(X_i)]^2 < \varepsilon^2 \tag{5}$$

All NNs that satisfy the condition (5) are equally valid emulations of the stochastic parameterization (4). Actually, each of these NNs emulates a member of the family of mappings that together represent the stochastic parameterization (3). **Therefore, all the NNs satisfying (5) together – the entire ensemble of NNs – represent the stochastic parameterization (3).** It is clear now that any estimate of the magnitude of ε is of a paramount importance for our approach. We will attempt to derive such an estimate in the next sections.

2) NN architectures and NN training

Selecting of an emulating NN architecture includes two different aspects and types of decisions: (i) the selection of inputs and outputs and their numbers (n and m in (1)), which, as we have already mentioned, are determined by the availability of the variables in the GCM, and (ii) the selection of the number of hidden neurons (k in (1)) in the emulating NN, which is determined by many factors (the length of the training set, the level of noise in the data, the character of conversions of the training and test errors, etc.).

Table 1 shows seven different (in terms of inputs and outputs) architectures we have experimented with here. The

major inputs for different NN architectures are the vertical profiles of the following model prognostic and diagnostic

TABLE I

DIFFERENT INVESTIGATED NN ARCHITECTURES (COMBINATIONS OF INPUTS AND OUTPUTS) . T IS TEMPERATURE, Q – ATMOSPHERIC MOISTURE OR VAPOR MIXING RATIO, W – VERTICAL VELOCITY, U AND V ARE HORIZONTAL WIND COMPONENTS, RELH IS RELATIVE HUMIDITY, RAD –RADIATIVE HEATING/COOLING RATES, Q1C – THE "APPARENT HEAT SOURCE", Q2 – THE "APPARENT MOIST SINK", PREC – THE PRECIPITATION RATE, AND CLD –CLOUDINESS.

Architecture	Inputs	Outputs
{2} – 47:40	T, Q	Q1C, Q2, PREC
{3} – 47:59	T, Q	Q1C, Q2, PREC, CLD
{4} – 87:66	T, Q, U, V	Q1C, Q2, PREC, CLD
{5} – 58:66	T, Q, W	Q1C, Q2, PREC, CLD
{6} – 81:66	T, Q, W, RelH	Q1C, Q2, PREC, CLD
{7} – 66:66	T, W, RelH	Q1C, Q2, PREC, CLD
{9} – 84:66	T, Q, W, RAD	Q1C, Q2, PREC, CLD

fields: Tabs - temperature, QV - water vapor, RelH – relative humidity, U and V – horizontal wind components, W - vertical velocity, and Rad – radiation heating/cooling rates. The major outputs for all NN architectures (except for the NN architecture {2}), are the vertical profiles (or vectors) of the following model prognostic and diagnostic fields: Q1C (the "apparent heat source"), Q2 (the "apparent moist sink"), PREC (the precipitation rate, a scalar), and CLD (cloudiness). For the NN architecture {2} no CLD vertical profile was used.

Next, the number of hidden neurons (HID) has to be selected. NNs with this number of hidden neurons will be used for comparisons of different architectures. We selected several architectures, varied HID, trained a corresponding NN, and tested it. Fig. 3 shows results of these experiments for the output parameter Q1C. The figure shows NN errors on training and test sets for HID changing from 1 to 20 for the architecture {3} (see Table 1).

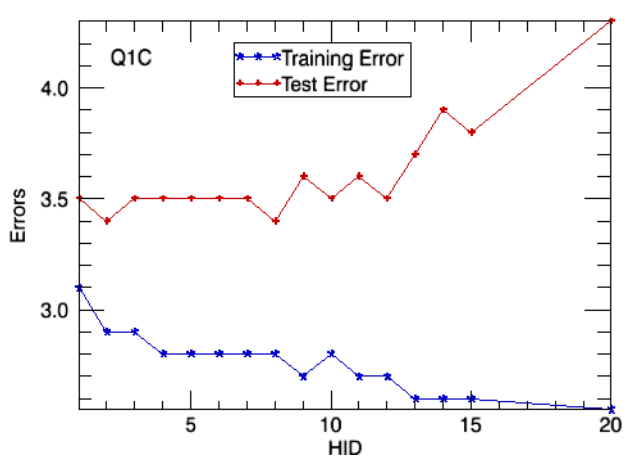

Fig. 3 NN approximation error on training and test sets for Q1C.

It is important to understand that NN training (a least square minimization) attempts to minimize the total $(\varepsilon + \varepsilon_{app})$*, that is the approximation error and the uncertainty (noise).* ***Because of very different statistical properties of these components, they can be approximately separated and roughly estimated using detailed information about the training and test statistics.*** *This is discussed in more detail below.*

Fig. 3 demonstrates a classic situation that is usually takes place when NN is trained using the data with a significant level of noise. The training error, after a sharp initial drop, decreases very slowly. The test error, after an initial drop, stabilizes and then increases. The interpretation of this behavior is well known. After the initial improvement of the approximation of the data due to an increasing flexibility of an approximating NN, a short interval of stability is reached (at HID ~ 3 to 7) when NN fits the signal but filters out the noise. Here the training error keeps decreasing; however, the test error is almost constant. Then with the increase of the flexibility of the approximating NN, it starts fitting the noise (i.e., the overfitting occurs). The training error keeps slowly decreasing; however, the test error quickly increases. The number of fitting parameters (NN weights) in NNs with HID >10 exceeds 1129. Taking into account that the training set contains 2240 records, it is not surprising that clearly pronounced overfitting is observed at HID > 10.

The above results are presented for the NN architecture {3} (see Table 1); however, other architecture produce similar results. Thus, we can conclude that, for a particular simulation (data set) used, HID = 5 would be a good approximation for the number of hidden neurons in emulating NN. This value is inside of the interval of stability of the test error when the NN emulation fits the mapping (3) but does not fit the noise in the data. For that reason, comparisons of different architectures presented in the following subsection are performed for HID = 5.

D. Comparison of different NN architectures and interpretation of NN training and test results

Here we present the comparison of different NN architectures defined in Table 1. NNs presented here were trained using the training set and tested using the independent test set; both sets are described above in Section 2.C.1. For each NN and for each NN output variable error statistics were calculated (bias, RMSE, and correlation coefficient) by comparison of NN generated output variables with the corresponding ones in the training or test set. All NNs have the number of hidden neurons HID = 5. It means that they all are inside the stability interval where NN fits well enough the signal in the training set and is not significantly responsive to noise in the data (see Fig. 3). The *training errors* for all output parameters are significantly less sensitive to the selection of the NN architecture than the test errors and to the selection of HID inside the interval of stability (see Fig. 3). Thus**, the training errors can be considered as a rough estimate of the noise in the data that is the inherent uncertainty of the stochastic parameterization (3)**.

Following this assumption, **the errors on the test set**

should be considered as a combination of the uncertainty (an estimate for it is provided by the training error) and an approximation error. For example, for Q1C, for the architecture {6} the training error is 2.3 K/day and the test error is 2.9 K/day. Thus, assuming that the uncertainty and the approximation error are independent, i.e., that in (4),

$$(\varepsilon + \varepsilon_{app})^2 = \varepsilon^2 + \varepsilon_{app}^2$$

only 1.8 K/day can be attributed to the NN approximation error. If we perform such a correction for all statistics, we find that, as in the aforementioned example, after the separation of the uncertainty (the training error) the NN approximation errors on the test set, in most cases, do not exceed significantly the uncertainty. In our case of NN emulation of a stochastic parameterization, the major criterion for evaluation of the NN emulation (5) is the similarity of the approximation error and the uncertainty. Thus, all trained NNs can be considered as equally valid emulations of the parameterization (3). **These NNs can be considered together as an NN ensemble emulation of the stochastic parameterization (3).**

Figs. 4 - 7 show the performance of NNs with different architectures on the independent test set. Fig. 4 demonstrates predictions of precipitation time series produced by different NNs in comparison with "pseudo-observations". These NNs produce an envelope (with a rather measurable spread) which on average gives a very good prediction of the precipitation on the test set. The spread of the envelope shows that there is still a measurable difference between NNs with different architectures and some of the members of the envelope (e.g. {9}) give results that are closer to the "pseudo-observations". The magnitude of the spread reflects the uncertainty of the parameterization (3).

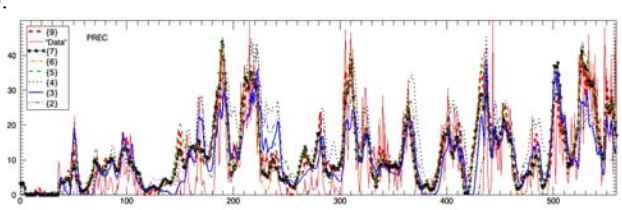

Fig. 4 NN simulations of precipitation (in mm/day) on the test set. The legend describes different curves presented in the figure; the number in parentheses indicates the NN architectures described in Table 1.

Figures 5 and 6 depict the mean profiles for outputs of the NN parameterization: Q1C and CLD. As in the case of precipitations, different NNs create envelopes with significant spreads for the mean profiles.

In general, the results show that the statistical structure of "pseudo-observations" and therefore of the parameterization (3) is well represented by envelopes created by different NNs. The spread of these envelopes reflects the magnitude of the uncertainty in the parameterization (3). The differences between the members of the ensemble inside the envelope are small as compared with the uncertainty; however, these differences are significant. They give estimates of the differences between members of the family of parameterizations determined by (3) and implicitly available in "pseudo-observations".

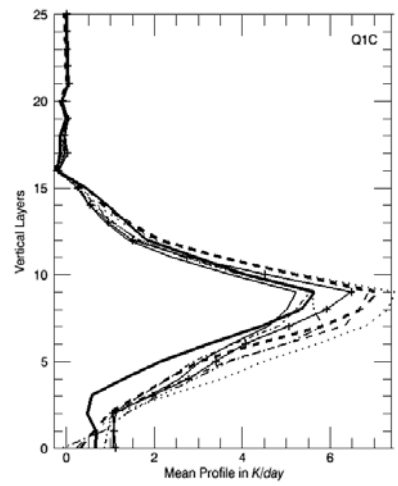

Fig. 5. Q1C (the apparent heat source from convection) mean profile on the test set produced by different NNs. The legend (see Fig. 6) describes different curves presented in the figure; the number in parentheses indicates the NN architecture described in the Table. The vertical levels are the same as in NCAR CAM.

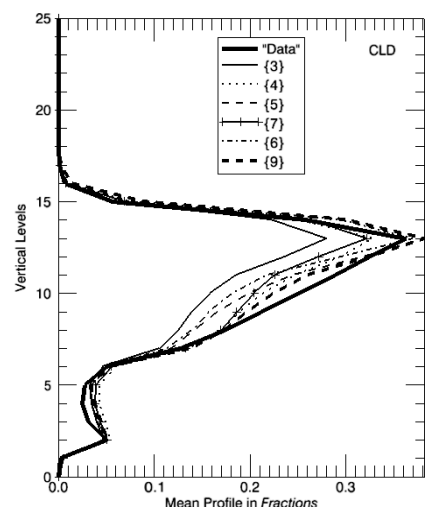

Fig. 6 Same as in Fig. 5 but for CLD

Fig. 7 shows the Hovmöller diagrams for the time series of cloudiness (CLD) profiles for "pseudo-observation" (the upper panel) and for a single member of the NN ensemble with the architecture {9} (the bottom panel). The patterns generated by NN are a bit smoothed, diffused; they are less sharp than the "observed" ones but very well recognizable. NN represents the sequence of patterns very well and without shifts.

As a result, we can conclude that the errors calculated on the test set cannot be considered as an estimate of the NN accuracy only; they contain a contribution of the parameterization uncertainty ε and should be adjusted as in the example above. Another conclusion that can be made is that the errors on the test set are more sensitive to the NN

architecture; however, the variations of errors for different NN architectures (the spread of the envelope created by different NNs) is representative of the level of noise in the

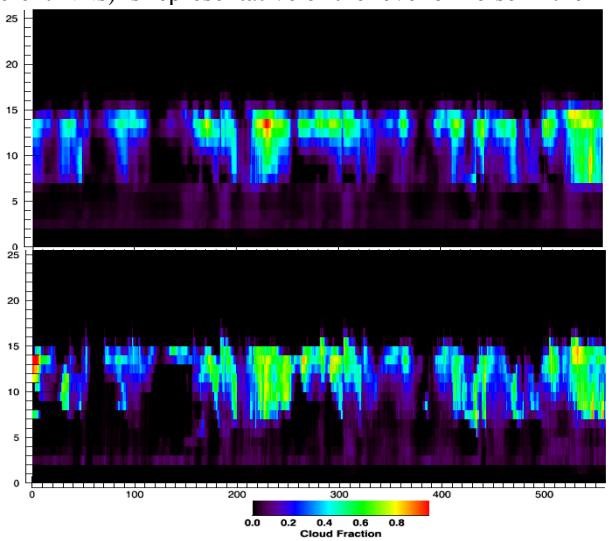

Fig. 7 Hovmöller diagrams for CLD profile time series: SAM "Data" – upper panel, NN {9} – lower panel.

data or the uncertainty of the stochastic parameterization (3). It means that, in the context of the current application (development of NN emulation for a stochastic parameterization (3)), selecting the best architecture for the emulating NN followed by the subsequent use of this "optimal" NN parameterization in GCM, is not the best approach. All NNs presented here (as well as other that could be developed by, for example, training NNs with the same architecture and the same number of neurons but with different initializations) can be considered as equally valid emulations of the parameterization (3). They **should be rather considered as members of a particular NN ensemble realization of the stochastic parameterization (3) represented by a particular data set. Thus, every single NN emulation can be considered as a particular realization of this stochastic parameterization.** The NN ensemble parameterization can be used in GCM in several different modes [15, 16].

We considered above an NN ensemble created by NNs with various architectures. Different methods of creating NN ensembles exist [15] and can be applied also in this case.

An initial validation of NN convection in NCAR CAM has been done in an off-line/diagnostic mode. Actually, CAM inputs have been used, at every time step of the control/original CAM integration, for parallel (off-line) calculations of the NN convection parameterization (CAM-NN) to produce its outputs as a diagnostic byproduct.

The CLD patterns shown in Fig. 3 are similar; their further in-depth analysis is needed. Note that the time series of precipitation and CLD for the CAM-NN- and control CAM runs (not shown) are similar in terms of both magnitude and frequency.

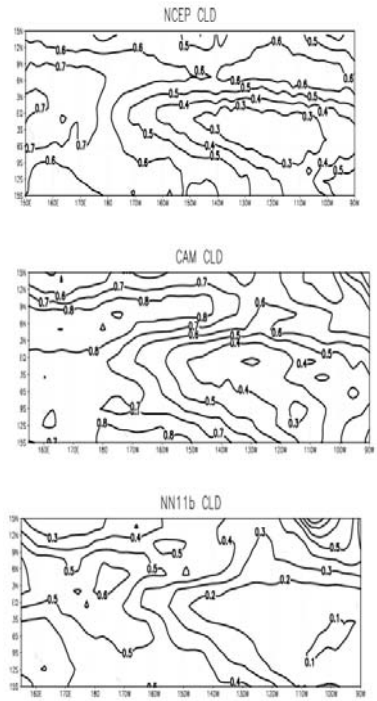

Fig. 8 Total CLD (in fractions, the contour interval is 0.1) for the Eastern Tropical Pacific Ocean (15 S to 15 N, 150 E to 90 W) for the 4-month TOGA-COARE period (Nov.-92 – Feb.-93) for the: NCEP reanalysis (upper), control CAM (middle), and CAM-NN (lower).

III. CONCLUSIONS

In this paper we introduce a novel approach to development of NN convection parameterizations based on applying the NN technique. This approach is used for development of a NN ensemble stochastic parameterization for climate models. This fast parameterization is built based on data from CRM simulations initialized and forced/driven with TOGA-COARE data. The SAM/CRM [12] has been used for CRM simulations. SAM emulated data were averaged and projected onto the GCM space of atmospheric states to implicitly define a stochastic convection parameterization. Next, the data were used to emulate the stochastic convection parameterization using an ensemble of neural networks defined by different choices of input and output variables. An ensemble of NNs with different architectures has been trained and tested. The inherent uncertainty of the stochastic convection parameterization derived in such a way is estimated.

The alternative approaches discussed in introduction require very large increases in the computational cost of current models. Their advantage is transparency, but we believe the computational cost is so high that it makes them impractical for most problems with present day computers. We believe that an ensemble of NNs, trained on the output from more realistic and comprehensive representations of convective processes over a variety of regimes could serve as a practical convective parameterization for global models. The NN ensemble can be used in global models in several

different ways [16].

We consider the results presented here as promising ones and expect that we will be able to produce a useful parameterization for a variety of application in weather and climate science. The NN convection parameterization provides an opportunity to capture many aspects of the more realistic representation at a fraction of the cost of the alternatives.

Our future plans include:

1. running SAM simulations initialized and forced by GCM (specifically, by NCAR CAM) to allow broader geographical coverage and to cover longer time period and more diverse weather conditions, and to develop a more representative training set
2. testing the NN convection parameterization trained using these new data in GCM in diagnostic and prognostic modes.

There are also some consequences to the approach. We are aware of the following issues:

1. We will have abstracted the physical description of the processes driving the system by another conceptual layer. A simple example of this situation can be seen when considering the possible influence of aerosol/cloud interactions. Unless we train the NN using aerosol distribution as 'inputs', and vary those inputs during the training process, we can not build a parameterization that is sensitive to those processes. Note that many convective parameterizations today are also unable to handle these processes.
2. We must of course have trained the NN parameterization on the 'right' inputs, and have sampled the phenomena of interest over the whole range of parameter space that the NN is expected to perform realistically over. The NN parameterization also needs to predict situations where no convection will occur, and adapt to different convective regimes (like shallow and deep maritime convection, midlatitude frontal convection, continent midlatitude summertime convection, continental deep convection). The parameterization must be able to seamlessly transition from one regime to another, or at the very least recognize where it should not be operating. Again, most current parameterizations have this same constraint.
3. It may be difficult to interpret the model response to variations in the important processes. For example it may be difficult to attribute increased precipitation to a particular process (e.g. a change in accretion processes in rain or snow falling from anvils) because those processes are not explicitly available within the parameterization to monitor. It should be possible to design an NN to output this information.

ACKNOWLEDGMENT

The authors would like to thank Prof. Marat Khairoutdinov (SUNY) for providing SAM and consultations on SAM and our collaborators Drs. Phill Rasch and Yefim Kogan for useful consultations and stimulating discussions.

REFERENCES

[1] Donner, L. J., and P. J. Rasch, 1989: "Cumulus initialization in a global model for numerical weather prediction", Mon. Wea. Rev., 117, 2654-2671.

[2] Rasch, P. J., M. J. Stevens, L. Ricciardulli, A. Dai, A. Negri, R. Wood, B. A. Boville, B. Eaton, and J. J. Hack, 2006: A Characterization of Tropical Transient Activity in the CAM3 Atmospheric Hydrologic Cycle, J. Climate 19, 2222–2242.

[3] Randall, D., S. Krueger, C. Bretherton, J. Curry, Duynkerke, M. Moncireff, B. Ryan, D. Starr, M. Miller, W. Rossow, G. Tselioudis, and B. Wielicki, 2003: Confronting models with data: The GEWEX Cloud Systems Study, Bull. Amer. Meteor. Soc. 84, 455-469, doi:10.1175/BAMS-84-4-455.

[4] Krueger, S. K., 1988: Numerical simulation of tropical clouds and their interaction with the subcloud layer, J. Atmos. Sci., 45, 2221–2250.

[5] Miura, H., H. Tomita, T. Casino, S. Riga, M. Satoh, and T. Matsuno, 2005: A climate sensitivity test using a global cloud resolving model under an aqua planet condition, Geophys. Res. Lett. 32, L19717, doi:10.1029/2005GL023672.

[6] Satoh, M., H. Tomita, H. Miura, S. Iga, and T. Nauno, 2005: Development of a global cloud resolving model – a multi-scale structure of tropical convection, J. of the Earth Simulator 3, 11-19.

[7] Grabowski, W. W., 2001: Coupling cloud processes with the large-scale dynamics using the Cloud-Resolving Convection Parameterization (CRCP), J. Atmos. Sci. 58, 978-997.

[8] Khairoutdinov, M. F., and D. A. Randall, 2001: A cloud resolving model as a cloud parameterization in the NCAR Community Climate System Model: Preliminary results. Geophys. Res. Let. 28, 3617-3620

[9] Randall, D., M. Khairoutdinov, A. Arakawa, and W. Grabowski, 2003: Breaking the cloud-parameterization deadlock, Bull. Amer. Meteor. Soc., 84, 1547-1564.

[10] Grabowski, W. W., Bechtold, P., Cheng, A., Forbes, R., Halliwell, C., Khairoutdinov, M., Lang, S., Nasuno, T., Petch, J., Tao, W., Wong, R., Wu, X., Xu, K., 2006: Daytime convective development over land: a model intercomparison based on LBA observations, Quarterly Journal of the Royal Meteorological Society.

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

[12] Khairoutdinov, M. F., and D. A. Randall, 2003: Cloud resolving modeling of the ARM summer 1997 IOP: Model formulation, results, uncertainties, and sensitivities. J. Atmos. Sci., 60, 607–625.

[13] Krasnopolsky, V. M., 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, Mon. Wea. Rev., vol. 133, No. 5, pp. 1370-1383.

[14] Krasnopolsky, V. M., 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.

[15] Krasnopolsky, V.M., 2007: "Reducing Uncertainties in Neural Network Jacobians and Improving Accuracy of Neural Network Emulations with NN Ensemble Approaches", Neural Networks, 20, pp. 454–461

[16] Krasnopolsky, V. M., M.S. Fox-Rabinovitz, and A. A. Belochitski, 2008: "Ensembles of Numerical Climate and Weather Prediction Models Using Neural Network Emulations of Model Physics", Proc. of the 2008 IEEE World Congress on Computational Intelligence, Hong Kong, June 1-6, 2008, CD-ROM, paper NN0498, pp. 1524-1531