CILAMCE
Ouro Preto/MG - Brazil
2003
XXIV IBERIAN LATIN-AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING

NEW LEARING SCHEME FOR MULTILAYER PERCEPTRON NEURAL NETWORK APPLIED TO METEOROLOGICAL DATA ASSIMILATION

Alexandre G. Nowosad
Centro de Previsão de Tempo e Estudos Climáticos (CPTEC)
Instituto Nacional de Pesquisas Espaciais (INPE)
alex@cptec.inpe.br
Haroldo F. de Campos Velho
Laboratório Associado de Computação e Matemática Aplicada (LAC)
Instituto Nacional de Pesquisas Espaciais (INPE)
haroldo@lac.inpe.br

Abstract. *The meteorological data assimilation process can be described as a procedure that uses observational data to improve the weather forecast produced by means of a mathematical model. Traditional methods include the Kalman filter. However, this method demands a heavy computational power. Recently, neural networks have been proposedas a new method for meteorological data assimilation by employing a multilayer perceptron network to emulate Kalman filtering at a lower computational cost. This paper presents a new scheme for learning process for the multilayer perceptron network, giving a more stable behavior for the assimilated data. Numerical results are shown for the one-dimensional shallow water meteorological model.*

Keywords: *Data assimilation, Artificial neural networks, Learning process, Numerical weather pretiction.*

1. INTRODUCTION

The data assimilation process can be described as a procedure that uses observational data to improve the prediction made by an inaccurate mathematical model. For example, suppose a computational model where many properties are only expressed approximately, like turbulent fluxes. Typically, the assimilation process can be outlined as a two step process (Yang & Cotton, 1998):

Forecast step: $\boldsymbol{w}_n^f = F[\boldsymbol{w}_{n-1}^a]$

Analysis step: $\boldsymbol{w}_n^a = \boldsymbol{w}_n^f + \boldsymbol{d}_n$

where w_n represents model state variable at time step n; F[.] is the mathematical (forecast) model, superscripts f and a denote forecast and analyzed values respectively, and $\boldsymbol{d}_n$ is the innovation of the observational data.

Several methods of data assimilation have been developed for air quality problems (Zannetti, 1990), numerical weather prediction (Daley, 1991), and numerical oceanic simulation (Bennet, 1992). In the case of atmospheric continuous data assimilation there are many deterministic and probabilistic methods (Daley, 1991). Deterministic approaches include dynamic relaxation, variational methods and Laplace transform, whereas probabilistic approaches include optimal interpolation and Kalman Filtering. Dynamic relaxation assumes the prediction model to be perfect, as does Laplace transform. Variational methods and optimal interpolation can be regarded as minimum-mean-square estimation of the atmosphere.

In the Kalman filtering, the analysis innovation $\boldsymbol{d}_n$ is computed as a linear function of the misfit between observation (superscript o) and forecast (superscript f):

$$\boldsymbol{d}_n = \boldsymbol{G}_n(\boldsymbol{w}_n^o - \boldsymbol{H}_n \boldsymbol{w}_n^f) \tag{1}$$

where $\boldsymbol{G}_n$ is the weight (gain) matrix, $\boldsymbol{w}_n^o$ is the observed value of $\boldsymbol{w}_n$ and $\boldsymbol{H}_n$ is the observation matrix. An adaptive extended Kalman filter has been tested in strongly nonlinear dynamical systems for assimilation procedure, such as the Lorenz chaotic system. Kalman filtering has the advantage of minimizing the error in the assimilation *plus* propagating this minimized error from one data insertion to the next. However, this process involves a heavy computational load, in particular for large meteorological systems. A strategy to alleviate this load is the use of neural networks to emulate the accuracy of the Kalman filtering (Jazwinski, 1970; Todling, 1997; Nowosad et al., 2000). Neural networks (Haykin, 1994) can be efficiently applied to map two data sets. Several architectures have been proposed for neural networks, in particular Multilayer Perceptron with backpropagation learning (Haykin, 1994) can be mentioned.

In a recent paper, Gardner and Dorling (1998) did a survey on applications of artificial neural networks (ANN) in meteorology, where a brief introduction about ANN and the backpropagation algorithm are shown. It also cites applications in the atmospheric sciences aiming at: (i) prediction (air-quality: surface ozone concentration, sulfur dioxide concentrations; severe weather; Indian monsoon, Brazilian rainfall anomalies, solar radiation); (ii) function approximation (air-quality, modeling of non-linear transfer functions), and, (iii) pattern classification (cloud classification; distinction between clouds and ice or snow; classification of atmospheric circulation patterns; land cover classification; classification of convergence lines from radar imagery; etc.). Although, the use of ANN for data assimilation

can be understood as a case of function approximation, this application was not mentioned in Gardner and Dorling's paper.

The current work is based on an application of neural network with *backpropagation* learning for data assimilation. This paper shows the strategies used to optimize the training phase. A new scheme for learning process for the multilayer perceptron network is based on the changing the bias update, producing a more stable behavior for the assimilated data. The training was applied on the Shallow Water model (Lynch, 1984). The next section provides a brief introduction to neural networks. However, it is not the aim of this paper to present an overview on ANN. Instead, a brief description of the ANN used is focused: the Multilayer Perceptron with backpropagation learning (Haykin, 1994). A further section discusses the neural network architecture used for the data assimilation application. The final section adds some comments and remarks.

2. MULTILAYER PERCEPTRON NEURAL NETWORKS WITH BACK-PROPAGATION LEARNING

An ANN is an arrangement of units characterized by:

- a large number of very simple neuron-like processing units;
- a large number of weight-biased connections between the units, where the knowledge of the network is stored;
- a highly parallel, distributed control.

The processing element (unit) in an ANN combines linearly multiple weighted inputs that are forwarded to an activation function. There are several different architectures of ANN, most of which depend on the learning strategy adopted.

The multilayer perceptron with backpropagation learning, or backpropagation neural network, is a feed-forward network composed of an input layer, an output layer, and a number of hidden layers for extracting high order statistics from the input data. Each of these layers that may contain one or more neurons. This is typically known as supervised learning as both the input and the expected output are fed with data to train the network.

Figure 1 shows a backpropagation neural network with one hidden layer. Functions g and F_{ANN} provide the activation for the hidden layer and the output layer, respectively. In order to make the network more flexible to solve nonlinear problems, the activation functions for the hidden layer are sigmoid functions.

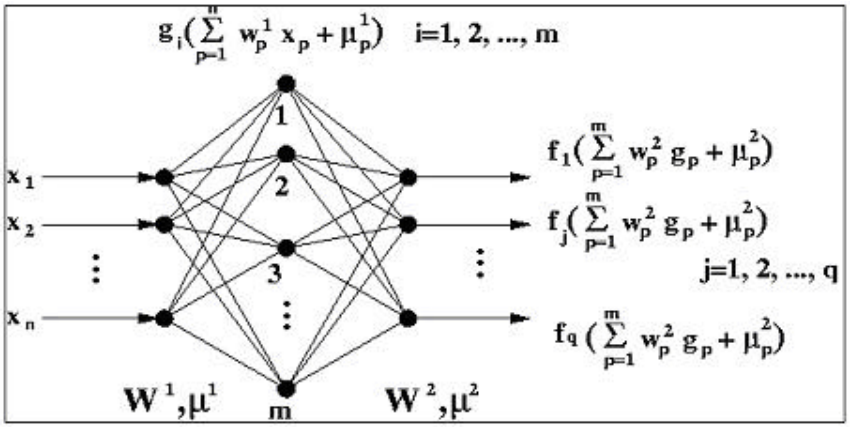


Figure 1. Multilayer Perceptron with one hidden layer with *m* neurons.

Mathematically, a perceptron network simply maps input vectors of real values into output vectors of real values. The connections in the figure have associated weights that are adjusted during learning process, thus changing the performance of the network.

There are two distinct phases in the usage of an ANN: the training phase (learning process) and the running phase (activation of the network). In the training phase, the weights are adjusted for the best performance of the network in mapping the many input-output vector pairs. In the activation phase, once the weights are established, new inputs are presented to the network in order to compute the corresponding outputs, based on what it has learned in the previous phase.

The training phase of a multilayer perceptron is controlled by a supervised learning algorithm. The main difference between supervised and unsupervised learning is that the latter uses only information contained in the input data, whereas the former requires both input and output (desired) data, which allows the calculation of the network error as the difference between the calculated output and the desired vector. The *Backpropagation Algorithm* consists of the adjustment of the network weights by backpropagating such error through the network. The weight change rule is a development of the perceptron learning rule: weights are changed by an amount proportional to the error at that neuron unit times the output of the unit feeding into the weight. This is the essence of the so-called *delta rule*. The training phase can make use of two modes: *batch mode* and *sequential mode* (Haykin, 1994). The former deals with the whole input data whereas the latter carries out the training based on each input pattern. The scope of this paper is restricted to batch mode in which *all* the input examples are taken at once and the learning procedure searches a set of weights q and biases m that minimizes the total squared error:

$$e_m = \sum_{k=1}^{N} \left\| F_{ANN}(X_k, q, m, m) - F(X_k) \right\|_2 \qquad (2)$$

where N is the number of examples in the training set, X_k is the input vector of example k, q and m are the weights and biases of the network, F_{ANN} is the approximation and F is the desired output value.

3. NEW LEARNING STRATEGY

There are two distinct phases for using an ANN: the training phase (learning process) and the running phase (activation of the network). In the training phase, the weights are adjusted for the best performance of the network in establishing the mapping of many input-output vector pairs. Once trained, the weights are fixed and new inputs can be presented to the network for it to compute corresponding outputs, based on what it has learned. One idea to improve the performance of the neural network is to look for better strategies for the learning process. The main idea of this strategy lies in reducing the learning rate of the bias.

In the following algorithm y_{ik} is the input to neuron j for each example k and Δ_{ik} is the gradient of the error at neuron i for example k. An adaptive version of the backpropagation algorithm in batch mode proceeds like this (Demuth, 1994):

[1.] Start with learning rate h_0, momentum constant, a =0.9, and momentum a_0 =0;

[2.] Calculate outputs of network and total e_m;

[3.] If e_m is acceptable stop;

[4.] At iteration m the backpropagation algorithm calculates:

[4.1] The error gradient of each neuron i for each example k: Δ_{ik};

[4.2] New weights and biases using, for each neuron i, its inputs j and each example k

[4.2.1] $\Delta(q_y)_{ij} = \Sigma_k \Delta_{ik} y_{kj}$

[4.2.2] $\Delta q_{ij}(m) = a_m \Delta q_{ij}(m-1) + (1-a_m) h_m \Delta(q_y)_{ij}$

[4.2.3] $\Delta m_i'(m) = a_m \Delta m_i(m-1) + (1-a_m) h_m \Sigma_k \Delta_{ik}$

[5.] If: $e_m > 1.04 e_{m-1}$ then $h_{m+1} = 0.7 h_m$;

$a_{m+1} = 0$, go to 1;

Else:

[5.1] if $e_m > e_{m-1}$ then $h_{m+1} = 1.05 h_m$ and $a_{m+1} = a$;

[5.2] Accept weights/biases: $q_{ij} = q_{ij}'$ and $m_i = m_i'$;

[5.3] go to 2.

The proposed modification in the backpropagation algorithm affects the updating of $\Delta m_i'(m)$ in step 4.2.3:

$$\Delta m_i'(m) = a_m \Delta m_i(m-1) + (1-a_m) h_m^2 \sum_k \Delta_{ik} \tag{3}$$

This modification tends to make the rate of change in biases m slower than the rate of change in weights q, because usually $h < 1$. However no proof is presented of convergence. The only contribution of this work is to show a new algorithm that resulted in better function approximation in one specific experiment.

4. NUMERICAL RESULTS

Two neural network topologies were tested in this work.

The topology for the first neural network consists of 120 neurons in the input layer, 80 neurons in the hidden layer, and 60 neurons in the output layer. The input layer neurons correspond to forecast and observation data of a given parameter whereas the output layer neurons correspond to the assimilated data. The network was trained using 100 example s.

The topology for the second neural network consists of 120 neurons in the input layer, 50 neurons in each of 2 hidden layers and 60 neurons in the output layer. The rest of the setup is the same. The network was trained using 500 examples.

Tests for the data assimilation process were performed on the Shallow Water physical model equation (Lynch, 1984). Dynamical equations for this model are:

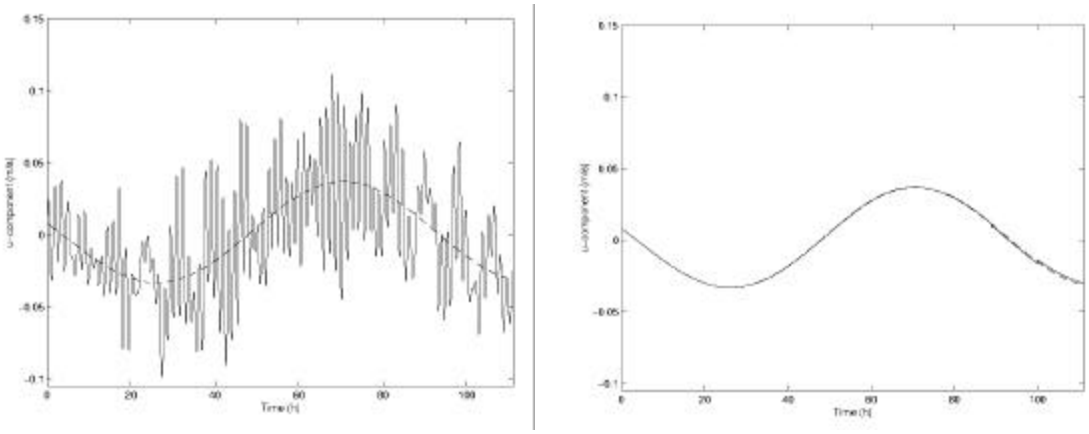
$$\frac{\partial z}{\partial t} + R_o \frac{\partial (uz)}{\partial x} + d + R_b v = 0 \tag{4}$$

$$\frac{\partial d}{\partial t} + R_o \frac{\partial (ud)}{\partial x} - z + R_b u + \frac{\partial^2 f}{\partial x^2} = 0 \tag{5}$$

$$\frac{\partial f}{\partial t} + R_o \frac{\partial (uf)}{\partial x} - R_o u_0 v + R_F d = 0 \tag{6}$$

where u, v are zonal and meridional wind components; f is the geopotential; $d = \partial u / \partial x$ is the divengence; $z = \partial v / \partial x$ is the vorticity; R_o=0.10, R_F=0.16, R_b=10, are dimensionless

numbers: Rossby, Froude, and a number that gives the importance of b-effect (Lynch, 1984). Hereafter prognostic variables will be grouped into a vector $w = [z \quad d \quad f]^T$. The system is discretized using forward and central finite difference method for time and space integration, where $N_x \Delta x = L$=10000 Km, being L the total length of the channel; N_x=32 the number of grid points; and Δt=100 seconds.



(a) (b)

Figure 2. Shallow water equation model - Assimilation with 80 neurons in the hidden layer (zonal wind component): (a) without assimilation; (b) neural network assimilation.

The numerical experiment was made inserting *observations* every 11.1 hours. The observational data were the same as forecast data added to a Gaussian deviations with zero-mean. For the data assimilation procedure tests were performed with the two neural networks. In the case of the first one the weights and bias were generated after 179519 iterations and the program took several days to finish its execution. Figure 2 shows the assimilation for the zonal wind component.

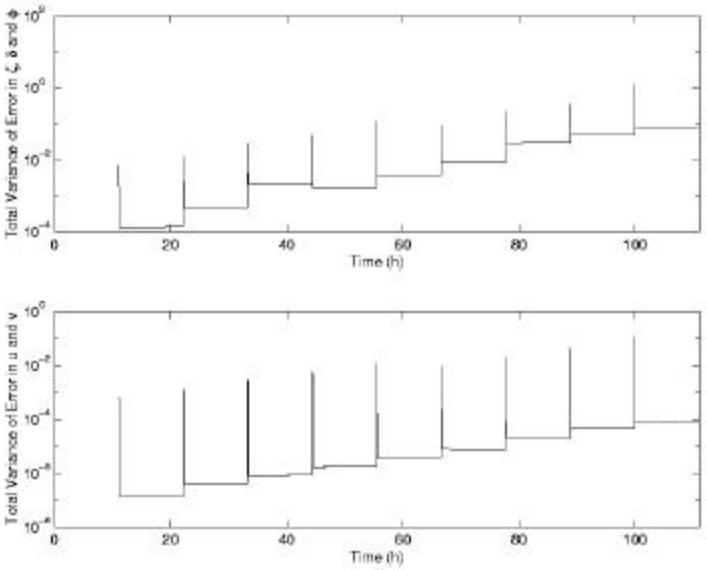


Figure 3. Error variances for standard scheme for updating the bias in the learning phase.

From figure 2, it can be noted that the neural network approach was effective as a data assimilation method, using the standard back-propagation scheme for training the network. However, the error in the assimilation process is increasing with time, as shown in figure 3. Experiments using more hidden layers failed.

The assimilation is also effective for data assimilation with the second neural network trained using the proposed modification. But, unlike the standard back-propagation scheme, the error variance becomes more stable, as shown in figure 4.

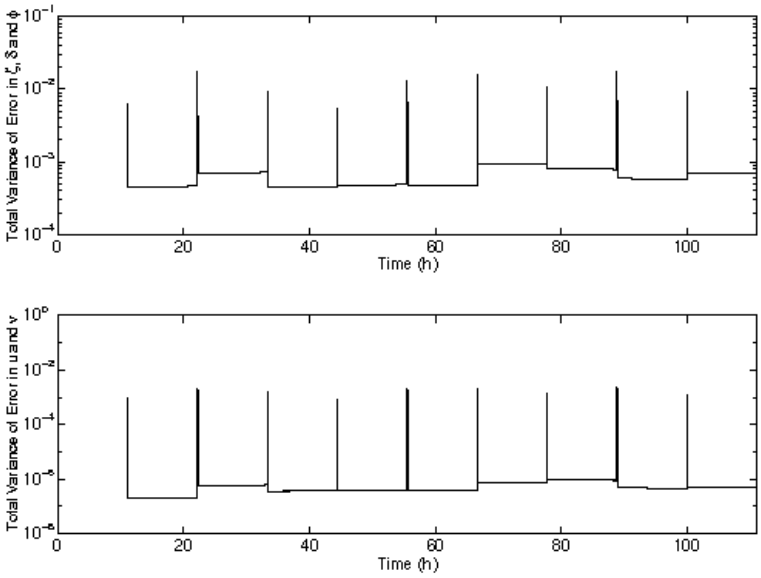


Figure 4. Error variances for new scheme for updating the bias in the learning phase.

5. FINAL REMARKS

The use of the proposed perceptron multilayer artificial neural network for meteorological data assimilation proved to be a good alternative as the results confirmed its feasibility. For the shallow water model, convergence was possible only after using 80 neurons in the hidden layer for the first neural network, and 50 neurons in the intermediate layers for the second one.

The focus of the present paper is to test a new scheme for training the neural network used in the data assimilation. The proposed modification in the standard learning process leads to a better network performance, with respect to the error stability for the analyzed case. A future task is to investigate this learning scheme for other models, such as pollutant dispersion model (Zannetti, 1990), and strongly non-linear systems in chaotic regime (Lorenz, 1960; Miller et al., 1994; Nowosad et al., 2000a; 2000b; 2003).

Even though the proposed modification in the training process was applied to the data assimilation, it is expected that similar performance would be achieved in other applications, mainly for function approximation situations.

Acknowledgements

This work was partially supported by the CNPq, Brazilian agency for research support.

REFERENCES

Bennet A. F., 1992, Inverse Methods in Physical Oceanography, Cambridge University Press, Cambridge, EUA.

Daley R., 1991, Atmospheric Data Analysis, Cambridge University Press, Cambridge, EUA.

Demuth H. & Beale, M., 1994, Neural Network Toolbox User's Guide (For Use with MATLAB), The MathWorks, Inc., MA, USA.

Gardner, M. W. & Dorling, S. R., 1998, Artificial Neural Networks (The Multilayer Perpectron) - A Review of Applications in the Atmospheric Sciences, Atmospheric Environment, vol. 32, n. 14/15, pp. 2627-2636.

Haykin, S., 1994, Neural Networks: A Comprehensive Foundation, Macmillan, New York.

Jazwinski, A. H., 1970, Stochastic Processes and Filtering Theory, Academic Press, New York, EUA.

Lorenz, E. N., 1963, Deterministic nonperiodic flow, Journal of Atmospheric Science, vol. 20, n. 2, pp. 130-141.

Lynch, P., 1984, DYNAMO: A One-Dimensional Primitive Equation Model, Irish Meteorological Service, Technical Note 44, Ireland.

Miller, R. N., Ghil, M. and Gauthiez, F., 1994, Advanced Data Assimilation in Strongly Nonlinear Dynamical Systems, Journal of the Atmospheric Sciences, vol. 51, n. 8, pp.1037-1056.

Nowosad, A. G., Rios Neto, A. and Campos Velho, H. F., 2000a, Data Assimilation in Chaotic Dynamics Using Neural Networks, Proceedings of Third International Conference on Nonlinear Dynamics, Chaos, Control and Their Applications in Engineering Sciences, July 31 - August 4, Campos do Jordão (SP), Brasil, Vol. 6, Chapter 6 - Control, pp. 212-221.

Nowosad, A. G., Campos Velho, H. F. and Rios Neto, A., 2000b, Neural Network as a New Approach for Data Assimilation, *Proceedings of the XI Brazilian Congress on Meteorology*, Rio de Janeiro, Brazil, 2000b, pp. 3078-3086.

Nowosad, A. G., Rios Neto, A. and Campos Velho, H. F., 2000c, Data Assimilation Using an Adaptative Kalman Filter and Laplace Transform, Hybrid Methods in Engineering, vol. 2, n. 3, pp. 291-310.

Nowosad, A. G., Rios Neto, A. and Campos Velho, H. F., 2003, Neural Network as a New Approach for Data Assimilation, Monthly Weather Review – submitted.

Todling, R., 1997, Estimation Theory and Foundations of Data Assimilation – Course Notes, Laboratório Nacional de Computação Científica, Rio de Janeiro, Brazil.

Yang, S. & Cotton, W. R., 1998, A Method of Continuous Data Assimilation using Short-range 4D-Var Analysis, Department of Atmospheric Science, Colorado State University, Technical Report - paper No. 653, Fort Collins (CO), USA.

Zannetti, P., 1990, Air Pollution Modeling, Computational Mechanics Publications, UK.