

Neural network for performance improvement in atmospheric prediction systems: Data Assimilation

Rosangela S. Cintra*, Haroldo F. de Campos Velho* and Helaine C. Furtado*

*Laboratory of Computing and Applied Mathematics

Brazilian National Institute of Space Research, São José dos Campos, SP, Brazil

Email:(rosangela.cintra,haroldo,helaine.furtado)@lac.inpe.br

Abstract—Predicting the weather is dependent on the initial states specified in the computer model used to make the prediction. The data assimilation (DA) schemes are state-estimation techniques to generate an appropriated initial states for numerical models. DA deals with observations and data from the nonlinear dynamical models, both data set are very large in use on operational weather centers. The output from the DA procedure is called *analysis*. Some DA techniques become computationally intensive. The artificial neural networks (NN) can be employed to improve the computational performance. Two DA schemes are analyzed here: the Local Ensemble Transform Kalman Filter [17], and a version of a variational assimilation method [2] named the *representer* method. The EnKF was applied to a 3D atmospheric global spectral model (SPEEDY model), while the representer scheme was applied to the 2D shallow-water model – for simulating the ocean circulation. These DA techniques were emulated by multilayer perceptron neural network (MLP-NN). The goal of this paper is to show the speed up for the DA computer performance in comparison to the methods emulated. The data assimilation process by NN preserves the analysis quality of the former DA techniques. In our experiments, the NN applied to DA on the SPEEDY model was 75 times faster than EnKF.

I. INTRODUCTION

Geophysical fluid dynamics models are comprised of fundamental concepts (laws) and parameterizations of physical, biological, and chemical components of the atmospheric and ocean systems. These concepts and parameterizations are expressed as mathematical equations, averaged over time and grid volumes in four dimensions (3 space dimensions, and one time dimension). The equations describe the evolution of many meteorological variables, and define the state of the physical system. These equations are then codified to a programming language, defining their possible interacting with other formulations, so that they can be solved on a computer and integrated forward in discrete time steps, attempting to solve a problem by repetitive calculations (iterations), before moving on to solving the next problem. These programs are called numerical models. All numerical models are based upon the set of governing equations, describing a number of known physical principles. Such formulation is mathematically described as an *initial value problem*.

For running any numerical model, it requires to compute the initial condition. Depending on the model type, it may also need of boundary conditions. In numerical weather prediction, the initial condition is obtained from a balanced analysis, incorporating observations describing the current state of the system. Data assimilation (DA) is the process where observa-

tions are embedded into models, and to adjust them in real time as new data becomes available. The results of DA process is a consistent model with the observed data and itself forecasting (also called *background*, which is initial condition to next model prediction period).

Global models work by calculating the system state at a number of discrete points on the Earth's atmosphere/ocean. These points are laid out as a mesh covering the surface of the Earth, and the vertical profiles of the atmosphere, dividing the domain into several little boxes. The number of boxes defines how finer is the model resolution. The computational effort is determined by the model resolution. The prediction period is calculated by time integration, starting from a set of initial conditions for the atmosphere and/or ocean, by using several discrete time-steps ahead. Large amount of data are employed during the DA cycle. In the operational weather prediction centers, usually state variables are of order $O(10^{7-9})$, and the number of observations of order $O(10^{5-7})$ for a period of 6 hours. The observations have great variability on time and space. A significant increase in the number of observations is expected to be as much as six orders of magnitude during the first decades of the 21 century. Typing DA as a big data issue, overwhelming data systems and computational resources. Therefore, it is important to develop new techniques for dealing with DA in near future [28].

Some advanced assimilation techniques require cycles of interaction between the model and statistical analysis, enhancing the computational effort [28].

There is an operation time-window to produce the forecasting. Many strategies can be adopted to fit the intensive computation with the operation period: the use of advanced computing, reduction of problem dimension to obtain a computer code feasible to run in real time. The computational complexity involved in data assimilation have been presented in the literature [22]. Modern DA techniques, like Ensemble Kalman filter (EnKF) [9], [18] and particle filter (PF) [20], are Bayesian techniques, represent a computational challenge, even with the use of parallel computing with thousands of processors. The algorithms are constantly updated and improved in its performance. One example is the version of the EnKF with small areas (local) for observation influence: the Local Ensemble Kalman filter (LEKF) [27].

Artificial neural network was employed on data assimilation by Nowosad and co-authors [26], using all points in the domain as input for Multilayer Perceptron (MLP). This approach was improved by [14] and [15], where the perfor-

mance the MLP and Radial Basis Function, and two recurrent NN (Elman and Jordan) (see [16]) were analyzed. [15] use neural networks at each grid point. Such modification allowed to reduce the computational processing. Continuing the investigations, [11] evaluated the neural network performance to emulate the particle filter and the variational method for data assimilation applied to Lorenz chaotic system. In [5], this technique was applied to emulate Local Ensemble Transform Kalman Filter (LETKF) as a method for data assimilation to the SEEDY (*Simplified Parameterizations Primitive-Equation Dynamics*) model. Recently, [13] applied this technique to emulate the representer variational (R-Var) method, simulating ocean data using with a 2D shallow-water model.

The use of artificial neural networks preserves the quality of the analysis emulating different methods. This paper presents the results of data assimilation applying NN approach to improve the computational performance of forecast systems.

A. Data Assimilation Techniques

The purpose of data assimilation is to reconstruct as accurately as possible the state of the system using all available appropriate information: observations and the physical laws governing the system, available from the numerical models.

In present-day, two data assimilation techniques can be distinguished in numerical weather prediction. These are variational methods, of which the dominant one is the 4D-Var method [29], and the ensemble schemes, like Ensemble Kalman Filter [9] and Particle filters [21]. Variational assimilation searches for a minimum of a cost function [7], and its adoption involve by the need to develop and coding of adjoint model. In ensemble techniques, ensemble forecasts are used to evaluate the probability distribution.

The Kalman Filter equations are obtained from an analytical solution from setting the gradient of the cost function to zero, considering recursive least square and the assumption of the Gaussian probability density functions. A brief description for Kalman filter algorithm could expressed below:

1. Forecast model for state vector:

$$x_{n+1}^f = M_n(x_n^a) .$$

2. Up-date the covariance matrix:

$$P_{n+1}^f = M_n P_n^a M_n^T + W_n^b .$$

3. Compute Kalman gain:

$$K_{n+1} = P_{n+1}^f H_{n+1}^T [W_n^b + H_{n+1} P_n^f H_{n+1}^T]^{-1} .$$

4. Compute the analysis:

$$x_{n+1}^a = x_{n+1}^f + K_{n+1} [x_{n+1}^{\text{Obs}} - H_{n+1} x_{n+1}^f] .$$

5. Up-date the analysis covariance:

$$P_{n+1}^a = [I - G_{n+1} H_{n+1}^T] P_{n+1}^f .$$

The matrices M and H represent the dynamical system and observation system, respectively. The covariance matrix W^b identifies the modeling error.

One interesting approach to estimate the covariance matrix W^b was the development of the Ensemble Kalman Filter

(EnKF) by Evensen [8], [9]. The nonlinear evolution problem for the error covariance is calculated by sampling from the posterior pdf and propagating the samples, the model states, forward in time with the fully nonlinear model equations. At any time the samples can be used to calculate an approximate mean and error covariance. The best implementation is in localization, in which spurious correlations are eliminated by applying a cut-off radius of influence for each observation. This is the Local Ensemble Kalman Filter (LEnKF) ([27], [17]). The LEnKF algorithm capture the space of forecast uncertainties, formulated by ensemble-based Kalman filter scheme.

1) *Local Ensemble Transform Kalman Filter*: The LETKF algorithm is an EnKF-based scheme, in which the analysis ensemble members are constructed by a linear combination of the forecast ensemble member [23]. The ensemble transform matrix, composed of the weights of the linear combination, is computed for each local subset of the state vector independently. The local subset depends on the error covariance localization [24].

The idea of LETKF is to perform analysis at each grid point simultaneously using the state variables and all observations in the region centered at given grid point. Each member of the ensemble gets its forecast: $\{x_{n-1}^f\}^{(i)} : i = 1, 2, 3, \dots, k$, where k is the total members at time t_n , to estimate the state vector \bar{x}^f of the reference model. The ensemble is used to calculate the forecasting by the average $\bar{x}^f \equiv k^{-1} \sum_{i=1}^k \{x^f\}^{(i)}$ and the model error covariance matrix is given by:

$$P^f = (k-1)^{-1} \sum_{i=1}^k (\{x^f\}^{(i)} - \bar{x}^f)(\{x^f\}^{(i)} - \bar{x}^f)^T \quad (1)$$

The analysis must determine not only an state estimate \bar{x}^a and covariance P^a , but also an ensemble $\{x_{n-1}^a\}^{(i)} : i = 1, 2, 3, \dots, k$ with the appropriate sample mean $\bar{x}^a \equiv k^{-1} \sum_{i=1}^k \{x^a\}^{(i)}$ and covariance:

$$P^a = (k-1)^{-1} \sum_{i=1}^k (\{x^a\}^{(i)} - \bar{x}^a)(\{x^a\}^{(i)} - \bar{x}^a)^T \quad (2)$$

DA problems are often limited by the high dimensionality of states created by spacial discretization over large high-resolution grids and the extensive spacial structure of observations. LETKF are suitable for such problems, promising computational efficiency and accuracy in localization method.

2) *Representer variational method*: The representer variational system minimizes the effort required for 4D-Var. The minimization algorithm for the R-Var system is the iterated, indirect representer algorithm. First, the system solves an iteration that yields a sequence of linear systems of Euler-Lagrange (EL) equations, for the minimize the penalty. Secondly, it solves an indirect solution of each linear system [3].

Considering jointly the forward ocean circulation computer code and the data assimilation scheme based on the R-Var approach, this set expresses the Inverse Ocean Modeling (IOM) system [2]. In the R-Var system, a penalty functional to be minimized is formulated for the assimilation problem, defining a fixed-interval smoothing problem. The penalty is quadratic for the residuals. The tangent linear version to the model is

considered. There are tangent linearization of the constraints that render the penalty. The procedure starts by solving the Green function associated to the EL equations. The previous solution combined with the model covariance matrix error provides the computation of representers. The posterior error covariance matrix at the end of smoothing interval becomes the prior error covariance matrix at the beginning of the next time period [2].

The representer algorithm is employed to find in the data subspaces, calculating all the representers as a linear basis for the innovation vector for each assimilation cycle. Given a background field x^f and weighted residual λ , the linear EL to be solved is obtained by using the *Representer method*:

$$-\frac{\partial \lambda}{\partial t} = \left[\frac{\partial M(x^f)}{\partial x} \right]^T (\lambda) - H^T W_{oo} (d - H\hat{x}) \quad (3)$$

with the optimal linearized forward dynamical system given by:

$$\begin{aligned} \frac{\partial \hat{x}}{\partial t} &= M(x^f) + \frac{\partial M(x^f)}{\partial x} (\hat{x} - x^f) + C_{qq} \lambda \\ \hat{x}(0) &= x^f + C_{ii} \lambda(0) . \end{aligned} \quad (4)$$

In the above equations, \hat{x} is the optimal solution. The weights W_j ($j = oo, qq, ii$) are defined as the inverses of the respective covariance error matrices: C_{oo} , C_{qq} , C_{ii} for observations, modeling, and initial condition, respectively. At each step t , $M(x^f)$ is the model dynamics. The representer expansion to the innovation for uncoupling Eqs. (3)–(4) is

$$\hat{x}(t) = x^f(t) + \sum_{m=1}^M \beta_m r_m(t) . \quad (5)$$

Here the background x^f (i.e., the trajectory around which the model is linearized) is also taken as the first guess (the solution that the assimilation will correct). The representer functions r_m ($m = 1, \dots, M$) are computed from two steps. First:

$$\begin{aligned} -\frac{\partial \alpha_m}{\partial t} &= \left[\frac{\partial M(x^f)}{\partial x} \right]^T (\alpha_m) - H^T \delta(t - t_m) \\ \alpha_m(T) &= 0 . \end{aligned} \quad (6)$$

Secondly, combining the solution from the above equation weighted by the modeling error covariance matrix:

$$\begin{aligned} \frac{\partial r_m}{\partial t} &= \frac{\partial M(x^f)}{\partial x} (r_m) + C_{qq} \alpha_m \\ r_m(0) &= C_{ii} \alpha_m(0) . \end{aligned} \quad (7)$$

The representer expansion coefficients β_m ($m = 1, \dots, M$), in Eq. (5), are grouped in the vector $\beta = [\beta_1 \dots \beta_M]^T$, and this vector is the solution of the linear system [2]:

$$[R^e + W_{oo}^{-1}] \beta = d - Hx^f, \quad (8)$$

where R^e is the representer matrix:

$$R^e = \begin{bmatrix} | & | & & | \\ r_1 & r_2 & \dots & r_M \\ | & | & & | \end{bmatrix}$$

obtained by evaluating the representer functions at the measurements sites (i.e., the m -th column of R^e is Hr_m), and the observations are writing by $d = Hx + \epsilon$, where ϵ is the vector of measurement errors. More details about the method can be found in [2].

B. Artificial Neural Networks

Gardner [10] did a survey on applications of the Artificial Neural Networks (ANN) in meteorology and oceanography. It also cites the applications in the atmospheric sciences looking at prediction process for air-quality, surface ozone concentration, dioxide concentrations, severe weather, Indian monsoon, Brazilian rainfall anomalies, solar radiation. The survey also mentioned other applications exploring the other properties from the ANN, as function approximation and pattern classification.

ANN is a computational system with parallel and distributed processing. It has the ability to learn and store the experimental knowledge. This arrangement is composed of simple processing units (nodes or neurons) that computes certain mathematical functions (usually nonlinear). Each artificial neuron is constituted by one or more inputs and one output. They have a principle of nonlinear, parallel, local processing, and adaptation. ANN can be devised in two phases: the training phase (learning process) and the run phase (activation or generalization). The training phase is an iterative process for adjusting the weights for the best performance of the network associated with a learning rule. Mathematically, the i^{th} neuron can be described as the following form:

$$\text{Input summation: } u_i = \sum_{j=1}^p w_{ij} x_j \quad (9)$$

$$\text{neuron output: } y_i = \varphi(u_i) \quad (10)$$

where x_0, x_1, \dots, x_p are the inputs; w_{i1}, \dots, w_{ip} are the synaptic weights; u_i is the output of linear combination; $\varphi(\cdot)$ is the activation function, and y_i is the neuron output. The use of units with nonlinear activation functions generalizes the delta rule, to solve nonlinear problems. The overall idea is to treat the neural network as a function of the weights w_{ij} , Eq. (9), instead of the inputs. The goal is to minimize the error between the actual outputs y_i and the target outputs in training data.

The set of procedures to adjust the weights is the learning algorithm. A scheme called *back-propagation*, performs the delta rule, Eq. (9). The back-propagation training is a supervised or teaching learning. The adjustments on the weights are conducted by back propagating algorithm, considering the difference between the ANN calculated output and the target output (the expected response to the input vector). Once trained, the weights are fixed and new inputs can be presented to the network, that calculates the corresponding outputs based on training, this phase is called the generalization.

Multilayer Perceptron (MLP) is an ANN topology where the interconnections of the inputs to the output layer have at least one intermediate layer of neurons: a hidden layer ([10], [16]). As mentioned before, the *back-propagation* algorithm can be used for training the MLP-NN. ANN are able to solve nonlinear problems, if nonlinear activation functions are used for the hidden and/or the output layers.

The experiments for data assimilation were performed using multilayer perceptron neural network (MLP-NN), with back propagation algorithm. The MLP-NN configuration (number of layers, nodes per layer, activation function, and learning rate parameter) was defined by empirical tests.

C. Models

1) *Linear 2D shallow-water*: The shallow water equations provide the fundamental description of free surface hydrodynamics in water or other incompressible fluids. This modeling is applied when the aspect ratio between the vertical length scale and the horizontal length scale is much smaller than one. Shallow water equations can also be used to model wave propagation. In such case, the underlying assumption is that the depth of the basin, where the fluid is moving, is smaller than to the wave length of the disturbance.

The equations are derived from the principles of conservation of mass and momentum for fluids, the hyperbolic Navier-Stokes equations, from depth-integrating. The independent variables are time (t), and two space coordinates (x, y). The dependent variables are the fluid height or depth H , and the two-dimensional fluid velocity components (u, v). The force acting on the fluid is gravity, represented by g .

Considering the linear shallow-water equations on a rotating planet, the equations are expressed as (where: $(x, y) \in \Gamma$):

$$\frac{\partial \mathbf{u}}{\partial t} + f \hat{\mathbf{k}} \times \mathbf{u} = -g \nabla (q - \bar{q}) - r_{\mathbf{u}} \mathbf{u} / H, \quad (11)$$

$$\frac{\partial q}{\partial t} + \nabla \cdot (H \mathbf{u}) = 0. \quad (12)$$

where f is the local value of Coriolis parameter, $\hat{\mathbf{k}}$ is a unit vector in the direction perpendicular to the velocity vector, ∇ is the two-dimensional nabla operator, \mathbf{u} is the two-dimensional velocity vector, $q = q(x, y, t)$ is the sea-level disturbance, $H = H(x, y)$ is the mean depth of the ocean, r is the bottom drag coefficient, and $\bar{q} = \bar{q}(x, y, t)$ is the unperturbed height of the shallow layer of fluid.

Coordinates (x, y) are the Cartesian ones, where x and y are directed to eastward and northward, respectively. The velocity vector $\mathbf{u} = (u, v)$ has the eastward direction (u) and the northward direction (v) components. The Coriolis parameter is defined by the f -plane: $f = 2\Omega \sin \phi_0$, where ϕ_0 is a mean latitude, Ω is the Earth rotation speed. If $q \equiv \bar{q}$, then the ocean is in hydrostatic balance [2]: this is the *equilibrium* tide of Newton. The scalar form of these vectorial equations, discretization procedure, initial and boundary conditions, forcings, and model parameters applied to this linear shallow-water model can be found in [13] and [2, page 197].

2) *SPEEDY model*: The *Simplified Parameterizations Primitive-Equation Dynamics* (SPEEDY) is an atmospheric general circulation model (AGCM) of intermediate complexity, based on a spectral primitive-equation dynamical core, and a set of simplified physical parameterization schemes. The computer code was developed to study global-scale dynamics and testing new approaches for numerical weather prediction. The dynamic variables on a set of nonlinear differential equations that are used to approximate global atmospheric flow, i.e., consisting of a form of the Navier-Stokes equations (conservation of momentum), thermal energy equation, continuity equation (conservation of mass), consisting into the primitive equations. These equations are integrated by spectral method in the horizontal, at each vertical level [4]. The SPEEDY model has a simplified set of physical parameterization schemes, but they are similar to realistic weather forecasting numerical models. The goal of this model is to obtain computational

efficiency, while maintaining characteristics to the state-of-the-art AGCM with complex physics parametrization [25].

The model is global with spectral resolution T30L7 (horizontal truncation with 30 wave members, and seven vertical levels), corresponding to regular grid with 96 zonal points (longitude), 48 meridian points (latitude), and 7 vertical pressures levels (100, 200, 300, 500, 700, 850, 925 hPa). The prognostic variables of input and output model are the absolute temperature (T), surface pressure (p_s), component of zonal wind (u), component of meridional wind (v), and an additional variable and specific humidity (q).

The error of upper levels and surface covariance matrices to run LETKF system, as well as the SPEEDY model's boundary conditions data and physical parametrizations, are the same as used by [23].

II. METHODOLOGY

Similar quality for the analysis (initial condition for NWP) procured for LETKF and R-Var was found by Cintra (2010) and Furtado (2012) applying the NN to emulate these techniques. Furtado [13] worked with a 2D ocean model Eq. (11), originally codified using R-Var. Cintra [5] employed NN as a method for data assimilation to emulate the LETKF to the SPEEDY model. This NN approach uses the supervised back propagation algorithm. Data set for learning phase was generated by using synthetic observations under R-Var and LETKF assimilation schemes.

This paper presents the computational performance of MLP-NN to emulate LETKF scheme with SPEEDY model for two experiments: Conv, and Sate. MLP-NN to emulate R-Var method was evaluated by the SW2D experiment. A brief description on the experiments is given below.

- Conv: Conventional observational experiment, where the observation grid has five meteorological variables (u, v, T, q, ps), simulating a world rawinsonde network, and the 3D grid model has $(48 \times 96 \times 7)$ grid points, for latitude, longitude, and vertical directions, respectively;
- Sate: Satellite observational experiment, with temperature retrievals, where a dense network (one point at each two model grid) for the observation points was considered – the model grid points was the same as above;
- SW2D: Two-dimensional linear shallow-water model, where three state variables (u, v, q) were applied at one grid point. The grid model was (20×11) for x and y coordinates. The model had a prediction interval with 20 time-steps.

All experiments use synthetic observations. They were generated from control model fields, adding Gaussian random noise (white noise). The control model fields to the assimilation method is obtained from the integration of models without noise.

The configuration MLP-NN of experiments are described in table (I). All neural networks has one hidden layer, and the activation function is the hyperbolic tangent:

$$\varphi(v) = \frac{1 - \exp(-av)}{1 + \exp(-av)},$$

for hidden and output layers. The input/output vectors have individual values for a grid point for each time-step.

TABLE I. MLP-NN PARAMETERS: LETKF–SPEEDY FOR CONV AND SATE EXPERIMENTS, R-VAR TO THE LINEAR SHALLOW-WATER (SW2D) MODEL.

MLP-NN	Conv	Sate	SW2D
input nodes	2	2	*
output nodes	1	1	*
hidden neurons	11	6	*
learning rate	0.001	0.005	0.005

Those parameters with (*) in the table I are different to each variable of MLP-NN: different MPL-NNs were necessary to emulate the analysis from the representer variational scheme, because there are several properties to be estimated (initial condition – for next time-integration interval, boundary condition, and model forcing). For each property to be estimated with the R-Var technique, one MLP-NN was designed [13].

A. Training/Generalization process

The training procedure was made with the back-propagation algorithm. The best set of weights was obtained when the error between the analysis calculated by the network and the target analysis was smaller than a fixed small value.

1) *Experiments Conv and Sate*: The training process was conducted with the forecast from the SPEEDY model, with LETKF as the data assimilation method.

Different NNs were applied to different regions of the world. The globe was divided into six regions of $90^\circ \times 120^\circ$. In the Conv experiment, a set of 30 NN (see Table I), where one MLP-NN was designed to each meteorological variable: 5 variables \times 6 regions. This division has homogeneous size regions, but the number of observations is distinct. The Sate experiment were designed by six MLP-NN, to assimilate temperature for each region. The number of observations is equal for all regions.

Three first months were collected from the years 1982, 1983, and 1984 for the training process. The NN topology is the same for all regions, but with different connections weights. The MLP-NN analysis is calculated to each grid point.

The MLP-NN generalization was performed during one month of time integration, with four data assimilation cycles per day. The experiment was carried out in the period: Jan/01/1985 (at 00 UTC) up to Jan/31/1985. The Fig. 1 shows an example of MLP-NN analysis close to LETKF analysis to SPEEDY model for p_s field at 09/01/2013; and Fig. 2 shows an example of (v) Zonal Wind field MLP-NN analysis similar to R-var analysis to SW2D model at time step= 10.

2) *Experiment SW2D*: The differential equations are discretized with Arakawa C-grid, and a *forward-backward* scheme for time stepping [1]. The model was integrated with 40 time-steps, for each prediction interval. For the first interval, the boundary conditions, $(x, y) \in \partial\Gamma$, were north and south rigid walls ($v(x, 0, t) = v(x, Y, t) = 0$), while all fields were periodic on the x -direction. The model forcings were $F_u = -C_d \rho_a u_a^2 / (H \rho_w)$, and $F_v = 0$, with initial condition equal zero for all variables in the internal domain, $(x, y) \in \Gamma$.

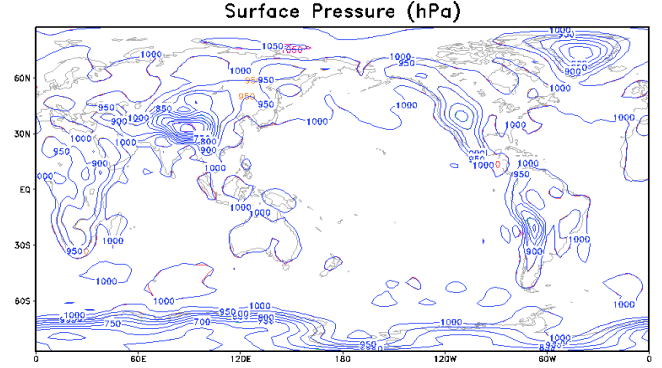


Fig. 1. LETKF (red line) and MLP-NN (blue line) analyses of Surface Pressure (hpa) at 09/01/1985 at 12UTC, for SPEEDY model.

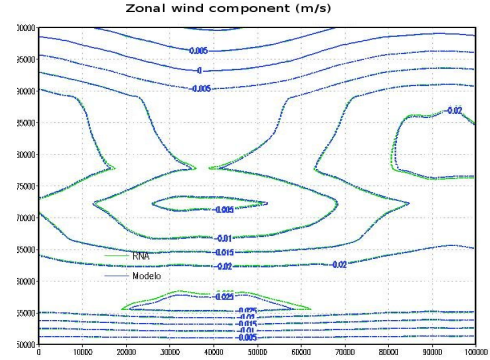


Fig. 2. Representer (blue line) and MLP-NN (green line) analyses of Zonal Wind Component, at time step 10 for SW2D model.

The training process uses 20 prediction intervals, where each interval is executed with 40 time-steps, performing 800 time-steps for learning data set. In this data assimilation procedure/experiment, several properties are estimated: forcings (f_u, f_v, f_q), initial condition for u, v, q , boundary condition: $v(x, 0, t)$ and $v(x, Y, t)$. Eight MLP-NNs were defined (3 variables + 3 forcings + 2 boundaries). For initial condition and forcing terms, the input is the observations and model variables ($u^{Obs, Mod}, v^{Obs, Mod}, q^{Obs, Mod}$). For the boundary condition identification, observations and model variable $v^{Obs, Mod}$ are used as the input.

B. Computer Performance Results

The *analysis* quality of LETKF and R-Var were evaluated by experts and the results are published in the literature [3], [19]. The similar quality analysis, to the former data assimilation methods cited, were obtained with MLP-NN [5], [6], [13].

In this paper, the comparative results for the computer performance among different methodologies is presented, by using the CPU-time. The same observations and time prediction period are run for all methods.

1) *Experiments Conv and Sate* : The SPEEDY model were run with LETKF initial condition for same period of MLP-NN generalization for both experiments. The experiment consists of one month for time integration, with four assimilation cycles

per day (at 00, 06, 12, 18 UTC). The experiment starts at 01/Jan/1985. The results are shown in Table II.

TABLE II. ONE MONTH EXPERIMENT FOR DATA ASSIMILATION (ANALYSIS AND FORECASTING) – CPU-TIME: HOUR:MIN:SEC.

Experiment	cycles	MLP-NN	LETKF
Conv	112	00:02:53	04:20:39
Sate	124	00:05:00	05:00:00

2) *SW2D Experiment*: The 2D Shallow-water model were run with the R-Var scheme to estimate initial condition, forcing, and boundary conditions values. During the 40 time-steps (the prediction period), there are observations on the the time-step 10 up to time-step 39. The analysis is computed by R-Var and MLP-NN using the collected observation on these 20 time-steps. This correspond the SW2D experiment, and results are shown in Table (III)

TABLE III. DATA ASSIMILATION SW2D EXPERIMENT (ANALYSIS) – CPU-TIME: MIN:SEC.

Experiment	DA	MLP-NN	R-Var
2swe	3 obs	04:00	00:02

III. CONCLUSION

These applications of MLP-NN produced significant reduction for the computational effort compared to LETKF and R-Var data assimilation schemes.

In the SPEEDY model experiment, with conventional observations, the MLP-NN scheme presented a speed-up 75 times related to the performance of the LETKF. For the SPEEDY experiment with satellite observations, the MLP-NN for the data assimilation was 87 times faster than the LETKF. The computer performance for the 2D shallow water experiment by using the MLP-NN had a speed up 113 times more efficient than the analysis computed by R-Var method.

The MLP-NN with the supervised learning approach achieved a better computational performance with similar quality for the analysis obtained with other formulations, i.e., from the computational point of view, the ANN is an efficient data assimilation process.

REFERENCES

- [1] F. Mesinger, A. Arakawa Numerical methods used in atmospheric models. GARP Publication, *WMO/ICSU Joint Organizing Committee*, Geneva, Switzerland, 64 pp, 2002.
- [2] A. Bennett *Inverse Modeling of The Ocean and Atmosphere*, 1st ed. Cambridge, U.K.: University Press, 2002.
- [3] A. Bennett, B.S. Chua, B.L. Pflaum, M. Erwig, Z. Fu, R.D. Loft, J.C. Muccino. Inverse Ocean Modeling System.Part I:Implementation. *Journal of Atmospheric and Oceanic Technology*, **25**, 1608-22, 2008.
- [4] W. Bourke. A multilevel spectral model: formulation and hemispheric integrations. *Mon. Wea. Rev.*, **102**, 687701, 1974.
- [5] R.S. Cintra *Assimilação de dados com redes neurais artificiais em modelo de circulação geral da atmosfera*. Thesis (D.Sc. on Applied Computing), National Institute for Space Research, 200 p., São José dos Campos. 2010. Available on-line: <<http://urlib.net/8JMKD3MGP7W/389UKPS>>.
- [6] R.S. Cintra, H.F. Campos Velho Global temperature assimilation by artificial neural networks for an atmospheric general circulation model. *Brazilian Congress of Computational Intelligence*, **10**, Fortaleza,CE, Brazil,2011

- [7] P. Courtier Dual Formulation of four-dimensional variational assimilation. *Q. J. Royal Meteor. Soc.*, **123(B)**, 2449-61, 1997.
- [8] G. Evensen Inverse Methods and Data Assimilation in nonlinear ocean models. *Physica D*, **77**, 108-109, 1994.
- [9] A. Evensen *Data assimilation: the ensemble Kalman filter*. Berlin: Springer, 2006.
- [10] T.W. Gardner, R.A. Wolf, R.W. Spiro, M.F. Thomsen First attempt at assimilation data to constrain a magnetospheric model. *Journal of Geophysical Research*, **104**, 25145–25152, 1999.
- [11] H.C. Furtado *Redes neurais e diferentes métodos de assimilação de dados em dinâmica não linear*. M.Sc. thesis (Applied Computing). National Institute for Space Research, 125 p. São José dos Campos, 2008.
- [12] H. C. M. Furtado, H. F. Campos Velho and E.E. Macau Data assimilation by neural network emulating representer method applied to the wave equation. *Proceedings:International Symposium on Uncertainty Quantification and Stochastic Modeling*, **1**, Maresias, SP, Brazil, 2012.
- [13] H.C. Furtado *Redes neurais para assimilação de dados em um modelo de circulação oceânica*. 173 p. Thesis (D.Sc. on Applied Computing), National Institute for Space Research. São José dos Campos, 2012. Available on-line:;<<http://urlib.net/8JMKD3MGP7W/3D3J5QS>>.
- [14] F.P. Härter and H.F. Campos Velho Recurrent and feedforward neural networks trained with cross correlation applied to the data assimilation in chaotic dynamic. *Brazilian Journal of Meteorology* **20**, 411-420, 2005.
- [15] F.P. Härter and H.F. Campos Velho New approach to applying neural network in nonlinear dynamic model. *Applied Mathematical Modeling*, **32**, 2621-2633, 2008.
- [16] S. Haykin *Neural Networks - a comprehensive foundation* 2nd ed. Prentice-Hall, 2001.
- [17] B.R. Hunt, E.J. Kostelich, and I. Szunyogh: Efficient data assimilation for spatiotemporal chaos: a. Local Ensemble Transform Kalman Filter. *Physica D*, **230**, 112-126, 2007.
- [18] P.L. Houtekamer and H.L. Mitchell: a sequential ensemble Kalman filter for atmospheric data assimilation. *Mon. Wea. Rev.*, **129**, 123-137, 2001.
- [19] E. Kalnay: *Atmospheric modeling, data assimilation and predictability*, Cambridge Univ. Press. Cambridge, UK, 341 p., 2003.
- [20] P.J. van Leeuwen Efficient Nonlinear Data-assimilation in Geophysical fluids dynamics. *Computer and Fluids*, **46**, 52-28, 2011.
- [21] P.J. van Leeuwen Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Q. J. Royal Meteorol. Soc.*, **136(B)**, 1991-1999, 2010.
- [22] P.M. Lyster, J. Guo, T. Clune, and J.W. Larson The Computational Complexity and Parallel Scalability of Atmospheric Data Assimilation Algorithms, *Journal of Atmospheric and Oceanic Technology*, **21(11)**, 1689-1700, 2004.
- [23] T. Miyoshi *Ensemble Kalman Filter experiments with a primitive equation global model*. PhD thesis (on Atmospheric Science). University of Maryland. College Park, Maryland, 2005.
- [24] T. Miyoshi and Yamane, S.: Local Ensemble Transform Kalman Filtering with an AGCM at a T159/L48 resolution, *Mon. Weather Rev.*, **135**, 3841-3861, 2007.
- [25] F. Molteni Atmospheric simulations using AGCM with simplified physical parameterizations. I: model climatology and variability in multi-decadal experiments. *Clim. Dyn.*, **20**, 175-191, 2003.
- [26] A.G. Nowosad, A. Rios Neto, H.F. Campos Velho Data assimilation in chaotic dynamics using neural networks. In: *International Conference on Nonlinear Dynamics, Chaos, Control and Their Applications in Engineering Sciences*, **6**, 212-221, 2000.
- [27] E. Ott, B.R. Hunt, I. Szunyogh, A.V. Zimin, E. Kostelich, M. Corazza, E. Kalnay, D.J. Patil and J. A. York:: A local ensemble Kalman filter for atmospheric data assimilation, *Tellus A*, **56**, 415-428, 2004.
- [28] R.B. Rood Fundamentals of modeling, data assimilation, and high-performance computing. Lecture Notes. *International Summer School of Atmospheric and Oceanic Sciences*. L'Aquila, Italy, 2004.
- [29] O. Talagrand and P. Courtier Variational assimilation of meteorological observations with the adjoint vorticity equation. I: Theory. *Quart. J. Roy. Meteor. Soc.*, **113**,1311-1328, 1987.
- [30] O. Talagrand *Data Assimilation in Meteorology and Oceanography*. Academic Press, 2008.