

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,100

Open access books available

127,000

International authors and editors

145M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Data Assimilation by Artificial Neural Networks for an Atmospheric General Circulation Model

Rosangela Saher Cintra and
Haroldo F. de Campos Velho

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.70791>

Abstract

Numerical weather prediction (NWP) uses atmospheric general circulation models (AGCMs) to predict weather based on current weather conditions. The process of entering observation data into mathematical model to generate the accurate initial conditions is called data assimilation (DA). It combines observations, forecasting, and filtering step. This paper presents an approach for employing artificial neural networks (NNs) to emulate the local ensemble transform Kalman filter (LETKF) as a method of data assimilation. This assimilation experiment tests the Simplified Parameterizations Primitive-Equation Dynamics (SPEEDY) model, an atmospheric general circulation model (AGCM), using synthetic observational data simulating localizations of meteorological balloons. For the data assimilation scheme, the supervised NN, the multilayer perceptrons (MLPs) networks are applied. After the training process, the method, forehead-calling MLP-DA, is seen as a function of data assimilation. The NNs were trained with data from first 3 months of 1982, 1983, and 1984. The experiment is performed for January 1985, one data assimilation cycle using MLP-DA with synthetic observations. The numerical results demonstrate the effectiveness of the NN technique for atmospheric data assimilation. The results of the NN analyses are very close to the results from the LETKF analyses, the differences of the monthly average of absolute temperature analyses are of order 10⁻². The simulations show that the major advantage of using the MLP-DA is better computational performance, since the analyses have similar quality. The CPU-time cycle assimilation with MLP-DA analyses is 90 times faster than LETKF cycle assimilation with the mean analyses used to run the forecast experiment.

Keywords: artificial neural networks, data assimilation, numerical weather prediction, computer performance, ensemble Kalman filter

1. Introduction

For operating systems in weather forecasting, one of the challenges is to obtain the most appropriate initial conditions to ensure the best prediction from a physical mathematical

model that represents the evolution of the atmospheric dynamics. Performing a smooth melding of data from observations and model predictions, the assimilation process carries out a set of procedures to determine the best initial condition. Atmospheric observed data are used to create meteorological fields over some spatial and/or temporal domain.

The analysis, i.e., initial condition, for NWP is combination of measurements and model predictions to obtain a representation of the state of the modeled system as accurate as possible. The analysis is useful in itself as a description of the physical system, but it can be used as an initial state for the further time evolution of the system [22]. The research of data assimilation methods has been studied for atmospheric and oceanic prediction, besides other dynamics researches like ionosphere and hydrological. The different algorithms of data assimilation were applied varying in complexity, optimality, and formulation. The approach of Bayesian scheme [31] uses ensembles of integrations of prediction models, where added perturbations to initial conditions and model formulation; the mean of ensemble forecasts can be interpreted as a probabilistic prediction. The ensemble Kalman filter (EnKF) [11, 23] uses a probability density function associated with the initial condition, characterizing the Bayesian approaches [9], and represents the model errors by an ensemble of estimates in state space. The Kalman filter (KF) [27] is one good technique to estimate an initial condition to a linear dynamic system. A useful overview of most common data assimilation methods used in meteorology and oceanography and detailed mathematical formulations can be found in texts such as Daley [9] and Kalnay [29].

The modern DA techniques represent a computational challenge, even with the use of parallel computing with thousands of processors. Nowadays, the operational NWP is using a higher resolution model, and the amount of observations has an exponential growth because of launch of new satellite. There is a computational challenge to get the analysis (initial condition) to run models, and so we need to make a prediction on time. The computational challenge to the data assimilation techniques lies in millions of equations involved in NWP models.

The DA algorithms are constantly updated to improve their performance. The example is the version of the EnKF [11] restricted to small areas (local); the local ensemble Kalman filter (LEKF) [38] is a version of the EnKF. We propose the application of artificial neural networks (NNs) like a DA technique to get a quality analysis and to solve the computational challenge.

First, the application of NN was suggested as a possible technique for data assimilation by [24, 30, 43]. The researches with NN (for data assimilation method) were initiated at INPE (National institute for Space Researcher) with Nowosad [37], see also [44, 5]; they used an NN over all spatial domains. Later, this method was improved by [16, 17], where they introduced a modification on the NN application, in which the analysis was obtained at each grid point, instead of at all points of the domain. They also evaluated the performance of two feed-forward NN (multilayer perceptron and radial basis function) and two recurrent NN (Elman and Jordan, see description in [19, 20]) [17]. Ref. [13] applied NN to emulate the particle filter and the variational data assimilation (4D var) for the Lorenz chaotic system. In 2012, Furtado [40] used an ocean model to emulate a variational method called representer. The NN technique was successful for all experiments, but they use theoretical or low-dimensional models. In 2010, Refs. [6, 7] applied this approach of supervised NN to an atmospheric general

circulation model (AGCM) to emulate a LETKF method. This is the experiment described in this paper, this experiment is the first one to use the 3D global atmospheric model; but the NN methodology research continues, see [41, 42], where this method is applied to FSU (Florida State Model) AGCM to emulate the LETKF data assimilation method too.

In every experiment, NNs were applied to mimic other data assimilation methods to obtain the analyses to initiate the forecast models. They do not use an error model estimation or error observation estimation. The main advantage to using NN is the speed-up of the data assimilation process.

This paper presents the approach based on a set of NN *multilayer perceptron* (MLP) [Section 3] employed to emulate the LETKF. The LETKF technique was used as the reference analysis, see [29, 32], Section 2.3. More information about LETKF can be obtained from [2, 26, 35]. The initial conditions generated by NNs are applied to a nonlinear dynamical system; the AGCM is the Simplified Parameterizations Primitive-Equation Dynamics (SPEEDY). The DA method is tested with synthetic conventional data, simulating measurements from surface stations (data at each 6 hours on a day) and upper-air soundings (data at each 12 hours on a day). The application of NN produces a significant reduction for the computational effort compared to LETKF. The goal of using NN approach is to obtain a similar quality for analyses with better computational performance for prediction process.

Summarizing, the NN technique uses the function:

$$x^a = F_{NN}(y^o, x^f) \quad (1)$$

where F_{NN} is the data assimilation process, y^o represents the observations, x^f is a model forecast (simulated), and x^a is the analysis field.

The observations used in operational data assimilation are conventional and satellite data. The observations include surface and upper-air observations; here, we simulate observations of one type of measurement, meteorological balloons. The grid of synthetic observations seeks to reproduce the stations of World Meteorological Organization (WMO) of radiosonde observations.

The experiment was conducted using the SPEEDY model [3, 21], which is a 3D global atmospheric model, with simplified physics parameterization by [36]. The spatial resolution considered is T30 L7 for the spectral method explained in Section 2.2. This paper shows that the analysis computed by the NN has the similar quality as the analysis produced by LETKF with minor computational effort.

2. Methodology

2.1. Artificial neural network (NN)

An NN is composed of simple processing units that compute certain mathematical functions (usually nonlinear). An NN consists of interconnected artificial neurons or nodes, which are

inspired by biological neurons and their behavior. The neurons are connected to others to form a network, which is used to model relationships between artificial neurons. NN has the ability to learn and store experimental knowledge. It is a computational system with nodes that can be parallel processing.

Each artificial neuron is constituted by one or more inputs and one output. The neuron processing is nonlinear and adaptable. Each neuron has a function to define the output, associated with a learning rule. The neuron connection stores a nonlinear weighted sum called a weight. The inputs are multiplied by weights, and the results go through the activation function. This function activates or inhibits the next neuron.

Mathematically, we can describe the i th input with the following form:

$$\text{input summation: } u_i = \sum_{j=1}^p \rho_{w_{i \times j} \times x_j} \quad (2a)$$

$$\text{neuron output: } y_i = \phi(u_i) \quad (2b)$$

where x_1, x_2, \dots, x_p are the inputs; w_{i1}, \dots, w_{ip} are the synaptic weights; u_i is the output of linear combination; $\phi(\cdot)$ is the activation function, y_i is the i th neuron output, and p is number of neurons (**Figure 1(a)**).

A feed-forward network, which processes in one direction from input to output, has a layered structure. The input layer is the first layer of an NN, where the patterns are presented, the hidden layers are the intermediary layers, and the last layer is called the output layer, where the results are presented. The number of layers and the quantity of neurons in each are determined by the nature of the problem. In most applications, a feed-forward NN with a single layer of hidden units is used with a sigmoid activation function, such as the hyperbolic tangent function (Eq. 3)

$$\varphi(v) = \frac{1 - \exp(-av)}{1 + \exp(-av)} \quad (3)$$

The NN has two distinct phases: the training phase (learning process) and the run phase (activation or generalization). An iterative process for adjusting the weights is made for the training phase, where the NN establishes the mapping of input and target vector pairs for the best performance. This phase uses the learning algorithm, i.e., a set of procedures for adjusting the weights. "Epoch" is the name of a training set pass through the iterative network process, testing of the verification set each epoch; the iterative process continues or stops after defined criteria that can be the minimum error of mapping or a determined number of epochs. Once the processing is stopped, the weights are fixed, and the NN is ready to receive new inputs (different from training inputs) for which it calculates the corresponding outputs. The latter phase is called the generalization: each connection (after training) has an associated weight value that stores the knowledge represented in the experimental problem and considers the input received by each neuron of that NN.

Neural network designs or NN architectures are dependent on the learning strategy adopted, see Haykin [19]. The multilayer perceptron (MLP) (**Figure 1(b)**) is the NN architecture used in

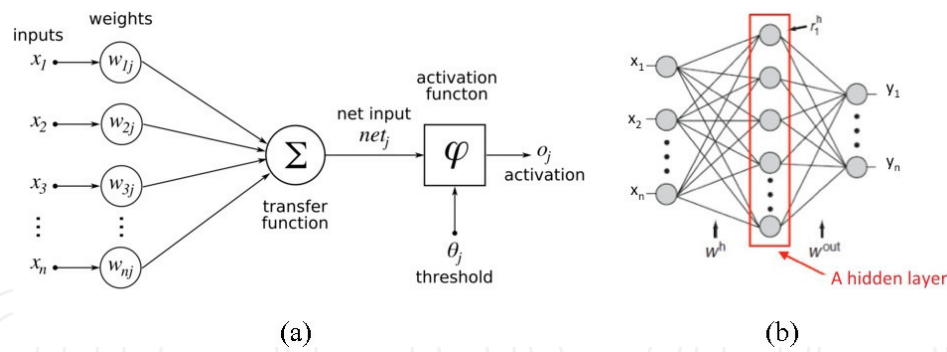


Figure 1. (a) Artificial neural network components and (b) multilayer perceptron.

this study, in which the interconnections between the inputs and the output layer have one intermediate layer of neurons, a hidden layer [14, 20]. NNs can solve nonlinear problems if nonlinear activation functions are used for the hidden and/or the output layers. In this work, during the training phase, the nonlinear activation functions employ the delta rule. Developed by [45], the delta rule is a version of the least mean square (LMS) method. The delta rule algorithm is summarized as follows:

1. Compute the error function $E(w_{ij})$, defining the distance between the target and the NN calculated output: $E(w_{ij}) \equiv [x_{ref}^a - x_{NN}^a]^2$
2. Compute the gradient of the error function $\partial E(w_{ij})/\partial w_{ij} = \delta_j y_i$, defining which direction should move in weight space to reduce the error, with $\delta_j \equiv (x_{ref}^a - x_{NN}^a) \phi'(v)$.
3. Select the learning rate η which specifies the step size taken in the weight space of updating equation;
4. Update the weight, k : $w_{ij}^k = w_{ij}^{k-1} + \Delta w_{ij}$, where $\Delta w_{ij} = -\eta E(w_{ij})/\partial w_{ij}$. One epoch or training step is a set of update weights for all training patterns, η is the learning rating.
5. Repeat Step 4 until the NN error function reaches the required precision. This precision is a defined parameter to stop the iterative process.

The supervised learning process, the functional to be minimized is treading as a function of the weights w_{ij} (Eq. 2) instead of the NN inputs. For a given input vector x , x_{NN}^a is compared to the target answer x_{ref}^a . If the difference is smaller than a required precision, no learning takes place; on the other hand, the weights are adjusted to reduce this difference. The goal is to minimize the error between the actual output y_i (or x_{NN}^a) and the target output (d_i) (or x_{ref}^a) of the training data. The set of procedures to adjust the weights is the learning algorithm *back propagation*, which is generally used for the MLP training. It performs the delta rule, considering a set of (input and target) pairs of vectors $\{(x_0, d_0), (x_1, d_1), \dots, (x_N, d_N)\}^T$, where N is the number of patterns (input elements) and one output vector $y = [y_0, y_1, y_2, \dots, y_N]^T$. The MLP performs a complex mapping $y = \varphi(w, x)$ parameterized by the synaptic weights w , and the functions $\varphi(\cdot)$ that provide the activation for the neuron. That is, for each (input/output) training pair, the delta rule determines the direction you need to be adjusted to reduce the error. In the

back-propagation supervised algorithm, the adjustments to the weights are conducted by back propagating of the error and the target output is considered the supervisor. Ref. [14] included brief introductions of MLP and the back-propagation algorithm.

The NN applications, generally, are on function approximation of modeling of nonlinear transfer functions and pattern classifications. Refs. [18, 25] reviewed applications of NN in environmental science including atmospheric sciences. They reviewed some NN concepts and some NN applications; these reviews were also for other estimation methods and its applications. Other reviews for NN applications in the atmospheric sciences, looking at prediction of air-quality, surface ozone concentration, dioxide concentrations, severe weather, etc., and pattern classifications applications in remote sensing data to obtain distinction between clouds and ice or snow were presented by [14]. Refs. [18, 25] also presented applications on classification of atmospheric circulation patterns, land cover and convergence lines from radar imagery, and classification of remote sensing data using NN. Data assimilation was not mentioned in such reviews.

2.2. SPEEDY model

The SPEEDY computer code is an AGCM developed to study global-scale dynamics and to test new approaches for numerical weather prediction (NWP). The dynamic variables for the primitive meteorological equations are integrated by the spectral method in the horizontal grid at each vertical level, more details in [3, 21]. The model has a simplified set of physical parameterization schemes that are similar to realistic weather forecasting numerical models. The goal of this model is to obtain computational efficiency while maintaining characteristics similar to the state-of-the-art AGCM with complex physics parameterization [32].

According to Ref. [36], the SPEEDY model simulates the general structure of global atmospheric circulation (**Figure 2**), and some aspects of the systematic errors are similar to many errors in the operational AGCMs. The package is based on the physical parameterizations adopted in more complex schemes of the AGCM, such as convection (simplified diagram of mass flow), large-scale condensation, clouds, short-wave radiation (two spectral bands), long-wave radiation (four spectral bands), surface fluxes of momentum, energy (aerodynamic formula), and vertical diffusion. Details of the simplified physical parameterization scheme can be found in Ref. [36].

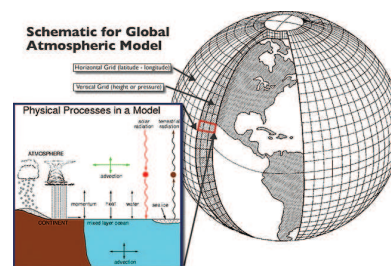


Figure 2. Schematic for global atmospheric model. **Source:** Center for Multiscale Modeling of Atmospheric Processes.

The boundary conditions of the SPEEDY model include topographic height and land-sea mask, which are constant. Sea surface temperature (SST), sea ice fraction, surface temperature in the top soil layer, moisture in the top soil layer, the root-zone layer, snow depth, all of which are specified by monthly means. Annual-mean fields specify bare-surface albedos, and fraction of land-surface covered by vegetation. The lower boundary conditions such as SST are obtained from the ECMWF's reanalysis in the period 1981–1990. The incoming solar radiation flux and the boundary conditions are updated daily. The SPEEDY model is a hydrostatic model in sigma coordinates. Ref. [3] also describes the vorticity-divergence transformation scheme.

The SPEEDY model is global with spectral resolution T30L7 (horizontal truncation of 30 numbers of waves and 7 levels). The vertical coordinates are defined on sigma ($\sigma = p/p_0$, where p_0 is the surface pressure) surfaces, corresponding to 7 vertical pressures levels (100, 200, 300, 500, 700, 850, and 925 hPa). The horizontal coordinates are latitude and longitude on regular grid, corresponding to a regular grid with 96 zonal points (longitude) and 48 meridian points (latitude). The schematic for global model and its physical packages can be seen at **Figure 2**. The prognostic variables for the model input and output are the absolute temperature (T), surface pressure (p_s), zonal wind component (u), meridional wind component (v), and an additional variable and specific humidity (q).

2.3. Brief description on local ensemble transform Kalman filter

The analysis is the best estimate of the state of the system based on the optimizing criteria. The probabilistic state-space formulation and the requirement for updating information when new observations are encountered are ideally suited to the Bayesian approach. The Bayesian approach is a set of efficient and flexible Monte Carlo methods for solving the optimal filtering problem. Here, one attempts to construct the posterior probability density function (pdf) of the state using all available information, including the set of received observations. Since this pdf embodies all available statistical information, it may be considered as a complete solution to the estimation problem.

In the field of data assimilation, there are only few contributions in sequential estimation (EnKF or PF filters). The EnKF was first proposed by [11] and was developed by [4, 12]. It is related to particle filters [1, 10] in the context that a particle is identified as an ensemble member. EnKF is a sequential method, which means that the model is integrated forward in time and whenever observations are available; these EnKF results are used to reinitialize the model before the integration continues. The EnKF originated as a version of the Extended Kalman Filter (EKF) [28]. The classical KF method, see [27], is optimal in the sense of minimizing the variance only for linear systems and Gaussian statistics. Analysis perturbations are added to run the ensemble forecasts, the mean of ensemble forecasts is the estimation error for analysis. Ref. [35] added Gaussian white noise to run the same forecast for each member of the ensemble in LETKF. The EnKF is a Monte Carlo integration that governs the evolution of the pdf, which describes the *a priori* state, the forecast and error statistics. In the analysis step, each ensemble member is updated according to the KF scheme and replaces the covariance matrix by the sampled covariance computed from the ensemble forecasts. Ref. [23] applied the EnKF to an

atmospheric system. They applied a state model ensemble to represent the statistical model error. The scheme of analysis acts directly on the ensemble of state models, when observations are assimilated. The ensemble of analysis is obtained by assimilation for each member of the reference model. Several methods have been developed to represent the modeling error covariance matrix for the analysis applying the EnKF approach; the local ensemble transform Kalman filter (LETKF) is one of them. Ref. [26] proposed the LETKF scheme as an efficient upgrade of the local ensemble Kalman filter (LEKF). The LEKF algorithm creates a close relationship between local dimensionality, error growth, and skill of the ensemble to capture the space of forecast uncertainties formulated with the EnKF scheme (e.g., [45]). In addition, Ref. [29] describes the theoretical foundation of the operational practice of using small ensembles, for predicting the evolution of uncertainties in high-dimension operational NWP models.

The LETKF scheme is a model-independent algorithm to estimate the state of a large spatial temporal chaotic system [38]. The term “local” refers to an important feature of the scheme: it solves the Kalman filter equations locally in model grid space. A kind of ensemble square root filtering [32, 45], in which the analysis ensemble members are constructed by a linear combination of the forecast ensemble members. The ensemble transform matrix, composed of the weights of the linear combination, is computed for each local subset of the state vector independently, which allows essentially parallel computations. The local subset depends on the error covariance localization [33]. Typically, a local subset of the state vector contains all variables at a grid point. The LETKF scheme first separates a global grid vector into local patch vectors with observations. The basic 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. The local strategy separates groups of neighboring observations around a central point for a given region of the grid model. Each grid point has a local patch; the number of local vectors is the same as the number of global grid points [35].

The algorithm of EnKF follows the sequential assimilation steps of classical Kalman filter, but it calculates the error covariance matrices as described below:

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 tn , to estimate the state vector \bar{x}^f of the reference model. The ensemble is used to calculate the mean of forecasting (\bar{x}^f):

$$\bar{x}^f \equiv k^{-1} \sum_{i=1}^k \{x^f\}^{(i)}. \quad (4)$$

Therefore, the model error covariance matrix:

$$P^f = (k-1)^{-1} \sum_{i=1}^k \left(\{x^f\}^{(i)} - \bar{x}^f \right) \left(\{x^f\}^{(i)} - \bar{x}^f \right)^T. \quad (5)$$

The analysis step determines a state estimate to each ensemble member:

$$\{x^a\}^{(i)} = \{x^f\}^{(i)} + W_K \left[x^{obs} - H \left(\{x^f\}^{(i)} \right) \right] \quad (6)$$

$$W_K = P^f H^T [H P^f H^T + R]^{-1}. \quad (7)$$

The analysis $\{x^a\}^{(i)}$ $i = 1, 2, 3, \dots, k$, (Eq. 6) by solving (Eq. 7) for W_k to get the optimal weight (e.g., Kalman gain). The matrix H represents the observation operator. The covariance matrix R identifies the observation error. The analysis step also updates the covariance error matrix P^a (Eq. 8)

$$P^a = (k - 1)^{-1} \sum_{i=1}^k \left(\{x^a\}^{(i)} - \bar{x}^a \right) \left(\{x^a\}^{(i)} - \bar{x}^a \right)^T \quad (8)$$

with the appropriate ensemble analyses mean:

$$\bar{x}^a \equiv k^{-1} \sum_{i=1}^k \{x^a\}^{(i)}. \quad (9)$$

The LETKF scheme has been applied to a low-dimensional AGCM SPEEDY model [32], a realistic model according to [42]. The LETKF scheme was also employed in the following: the AGCM for the Earth Simulator by [35] and the Japan Meteorological Agency operational global and mesoscale models by [34]; the Regional Ocean Modeling System by [41]; the global ocean model known as the Geophysical Fluid Dynamics Laboratory (GFDL) by [39]; and GFDL Mars AGCM by [15].

3. MLP-DA in assimilation for SPEEDY model

The NN configuration for this experiment is a set of multilayer perceptron, hereafter, referred to as MLP-DA. On the present paper, the NN configuration (number of layers, nodes per layer, activation function, and learning rate parameter) was defined by empirical tests, and we found the following characteristics:

1. two input nodes, one node for the meteorological observation vector and the other for the 6-hour forecast model vector;
2. one output node for the analysis vector results;
3. one hidden layer with 11 neurons;
4. the hyperbolic tangent (Eq. 3) as the activation function (to guarantee the nonlinearity for results);
5. learning rate η is defined do each MLP; and
6. training stops when the error reaches 10^{-5} or after 5000 epochs, which criterion first occurs.

The vectors values represent individual grid points for a single variable with a correspondent observation value on model point localization. The grid points is considered where observation value exists, see **Figure 3**. In the training algorithm, the MLP-DA computes the output and compared it with the “input” analysis vector of LETKF results (the target data), but it is not a node for the MLP generalization. The output vectors represent the analysis values for one grid point too. Care must be taken in specifying the number of neurons. Too many neurons can lead the NN to memorize the training data (over fitting), instead of extracting the general features that allow the generalization. Too few neurons may force the NN to spend too much time trying to find an optimal representation and thus wasting valuable computation time.

One strategy used to collect data and to accelerate the processing of the MLP-DA training was to divide the entire globe into six regions: for the Northern Hemisphere, 90° N and three longitudinal regions of 120° each; for the Southern Hemisphere, 90° S and three longitudinal regions of 120° each. This division provides the same size for each region, but the number of observations is distinct, as illustrated by **Figure 3**. This regional division is applied only for the MLP-DA; the LETKF procedures are not modified.

The MLP-DA scheme was developed with a set of 30 NN (six regions with five prognostic variables (p_s , u , v , T , and q)). Each grid point has all vertical layers values for the model. One MLP with characteristics described above was designed for each meteorological variable of the SPEEDY model and each region. Each MLP has two *inputs* (model and observation vectors), one output neuron which is the *analysis* vector, and the training scheme is the back-propagation algorithm.

The MLP-DA is designed to emulate the global LETKF analysis for SPEEDY initial condition. The LETKF analysis is the mean field of an ensemble of analyses. Fortran codes for SPEEDY and LETKF [32] were adapted to create the training data set for that period. The upper levels and the surface covariance error matrices to run the LETKF system, as well as the SPEEDY model boundary conditions data and physical parameterizations, are the same as those used for Miyoshi’s experiments.

The initial process is the implementation of the model, it assumes that it is perfect (initialization = 0); and the SPEEDY model T30 L7 was integrated for 1 year of spin-up, i.e., the period required for a model to reach steady state and obtain the simulated atmosphere. The model

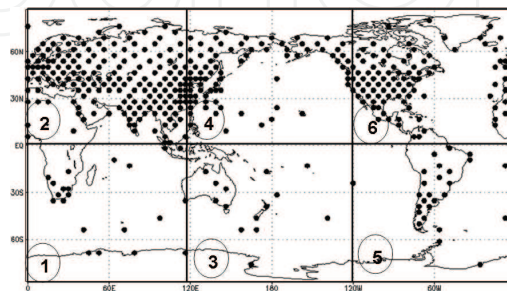


Figure 3. Observations localizations in global area. The dot points represent radiosonde stations (about 415) divided in six regions.

ran (without interruption) four times per day, from 01 January 1981 until 31 December 1981 and the last result is the initial condition for SPEEDY to 01 January 1982 0000 UTC. The model fields, so-called “true” (or control) model, are generated without data assimilation (each 6 hours forecast, is the initial condition for the next execution). The “true” model forecasts collected for executions without DA, considered four times per day (0000, 0600, 1200, 1800 UTC), the model run from 01 January 1982 through 31 December 1984 and collected analysis for each run.

The synthetic observations are generated, reading the “true” SPEEDY model fields, adding a random noise of meteorological variables: surface pressure (ps), zonal wind component (u), vertical wind component (v), absolute temperature (T), and specific humidity (q). The observation localization is on grid model point. An observation *mask* is designed, adding a positive flag to grid point, where the observation should be considered; the locations simulate the WMO data stations observations from radiosonde (**Figure 3**). Except for ps observations, the other observations are upper level with seven levels. Both assimilation schemes, LETKF and MLP-DA, use the same number of observations at the same grid point, i.e., the observation localization mask.

3.1. Training process

The training process for the experiment is conducted with data obtained from the SPEEDY model and the LETKF analyses. The LETKF analyses are executed with synthetic observations: upper levels wind, temperature and humidity, and surface pressure to 0000 and 1200 UTC and 0600 and 1800 UTC with surface observations only. The LETKF runs generate the analyses target vectors, the input observations vectors, and analyses field to run the SPEEDY model, which generates the input forecasts vectors for training the MLP-DA. The training is made with back-propagation algorithm.

Executions of the model with the LETKF data assimilation are made for the same period mentioned for the true model: from 01 January 1982 through 31 December 1984. The ensemble forecasts/analyses of LETKF have 30 members. The ensemble average of the forecasts and analyses fields, to this training process, is obtained by running SPEEDY model with the LETKF scheme.

These data are collected, initially, by dividing the globe into two regions (northern and southern hemispheres), but the computational cost was high because the training process took 1 day for the performance to converge. Next, the two regions were divided each into three regions, for a total of six regions. Then, we use this division strategy to collect the 30 input vectors (observations, mean forecasts, and mean analyses) at chosen grid points by the observation mask during LETKF process. The NN training process begin after collecting the input vectors for whole period (3 years), the training took about 15 min, for a set of 30 NN.

The MLP-DA data assimilation scheme has no error covariance matrices to spread the observation influence. Therefore, it is necessary to capture the influence of observations from the neighboring region around a grid point considered as a “new” observation. This calculation is based on the distance from the grid point related to observations inside a determined neighborhood (initially: $\gamma = 0$)

$$\hat{y}_{i\pm m, j\pm m, k\pm m}^o = \frac{y_{ijk}^o}{(6 - \gamma)r_{ijk}^2} + \sum_{l=1}^6 \alpha_l \frac{y_{i\pm m, j\pm m, k\pm m}^o}{r_{ijk}^2}$$

$$(m = 1, 2, \dots, M)$$
(10)

$$\alpha_l = \begin{cases} 0 & (\text{if there is no observation}) \\ 1 & (\text{if there is observation, and : } \gamma_{\text{new}} = \gamma_{\text{old}} + 1) \end{cases}$$

where \hat{y}^o is the weighted observation, M is the number of discrete layers considered around observation,

$r_{ijk}^2 = (x_p - y_i^o)^2 + (y_p - y_j^o)^2 + (z_p - y_k^o)^2$, where (x_p, y_p, z_p) is coordinate of the grid point, and the (y_i^o, y_j^o, y_k^o) is the coordinate of the observation, and γ is a counter of grid points with observations around that grid point (y_i^o, y_j^o, y_k^o) . If $\gamma=6$, there is no influence to be considered. Each observation's influence computed on a certain grid point is a new location, hereafter referred to as pseudo-observation, which adds values to the three input vectors to NN training process and also adds positive flags in the observation mask.

Then, the grid points to be considered in MLP-DA analysis are greater than grid points considered to LETKF analysis, although these calculations are made without interference on LETKF system. The back-propagation algorithm stops the training process using the criteria cited at item 6 above (Section 3), after obtaining the best set of weights; it is a function of *smallest error* between the MPL-NN analysis and the target analysis (e.g., when the root mean square error between the calculated output and the input target vector is less than 10^{-5}). The learning process is the same for each MLP of the set of 30 NN and takes about 15 min to get the fixed weights before the MLP-DA data assimilation cycle or generalization process of MLP-DA.

3.2. Generalization process

The training is performed with combined data from January, February, and March of 1982, 1983, and 1984, and in generalization process, MLP-DA is able to perform analyses similar to the LETKF analyses.

The generalization process is indeed the data assimilation process. The MLP-DA results a global analysis field. The MLP-DA activation is entering by input values (only 6 hours forecast and observations) at each grid point once, with no data used in the training process. The input vectors are done at grid model point, where it is marked (by positive flag mask) with observation or pseudo-observation (Eq. 10). The procedure is the same for all NN but one NN for each region, and each prognostic variable has own connection weights. All NNs have one hidden layer, with the same number of neurons for all regions. The regional grid points are put in the global domain to make the analysis field after generalization process of the MLP-DA, e.g., the activation of 30 NN results a global analysis. The regional division is only for inputting each NN activation.

The MLP-DA data assimilation is performed for 1-month cycle. It starts at 0000 UTC 01 January 1985, generating the initial condition to SPEEDY model. Running the SPEEDY model starting at 31 December 1984 1800 UTC carried out the previous model prediction. There were observations available at 0000 UTC 01 January 1985. Therefore, an analysis was computed for the SPEEDY model at 0000 UTC 01 January 1985. The SPEEDY model is re-executed with the former analysis, producing a new 6 hours forecast. The process is repeated at each 6 hours.

In this experiment, the MLP-DA begins the activation at 01 January 1985 0000 UTC and generates analyses and 6 hours forecasts up through 31 January 1985 1800 UTC.

4. Results

The input and output values of prognostic variables (p_s , u , v , T , and q) are processed on grid model points for time integrations to an intermittent forecasting and analysis cycle. Taking into account the pseudo-observation (Eq. 10), two grid layers ($M = 2$) around a given observation are considered.

The results show the comparison of analysis fields, generated by the MLP-DA, the LETKF, and the true model fields. The global surface pressure fields (at 11 January 1985 1800 UTC) and differences between the analyses are shown in **Figure 4**. The analysis fields and the differences between both assimilation, for 11 January 1985 at 1800 UTC at 950 and 500 hPa are also shown, for at 18 UTC at levels 950 hPa (near surface) and 500 hPa are also shown, for T , u , v , and q meteorological global fields, in **Figures 5–10**. These results show that the application of MLP-DA, as an assimilation system, generates analyses similar to those calculated by the LETKF system. Sub-figure (d) from **Figures 5–10** shows very small differences between the MLP-DA and LETKF analyses. The difference field of absolute temperature (K) at 500 hPa is about 3 degrees; and the difference field of humidity at 950 hPa is about 0.002 kg/kg.

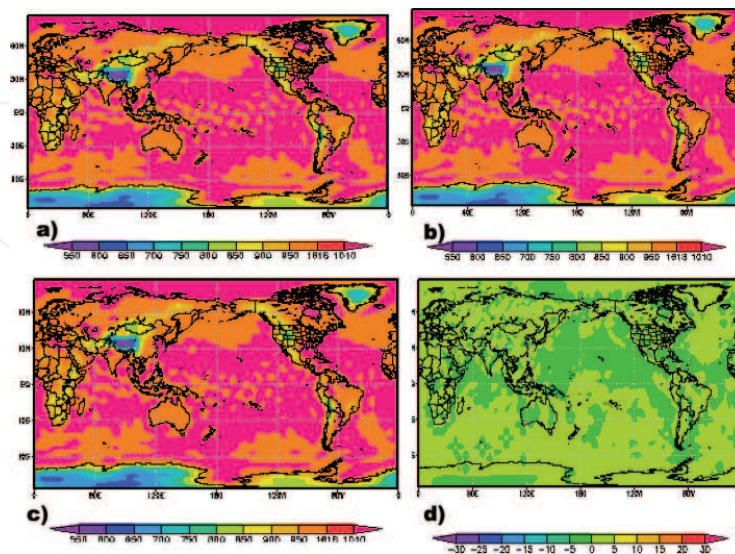


Figure 4. Surface pressure (PS) [Pa]—Jan/11/1985 at 18 UTC (a) LETKF analysis, (b) ANN analysis, (c) true model, and (d) differences analysis.

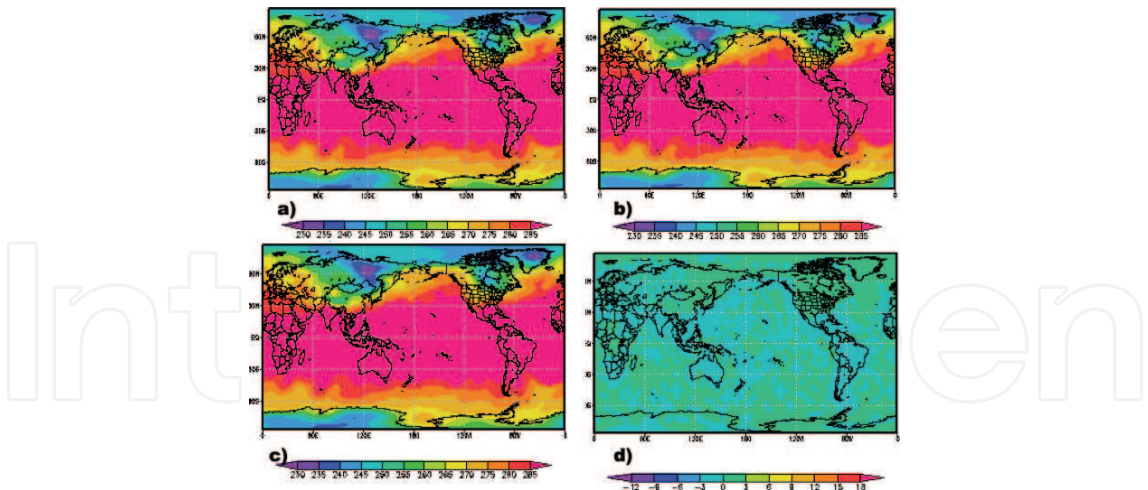


Figure 5. Absolute temperature (T) [K] Fi-950 hPa, Jan/11/1985 at 18 UTC. (a) LETKF analysis, (b) ANN analysis, (c) true model, and (d) differences analysis.

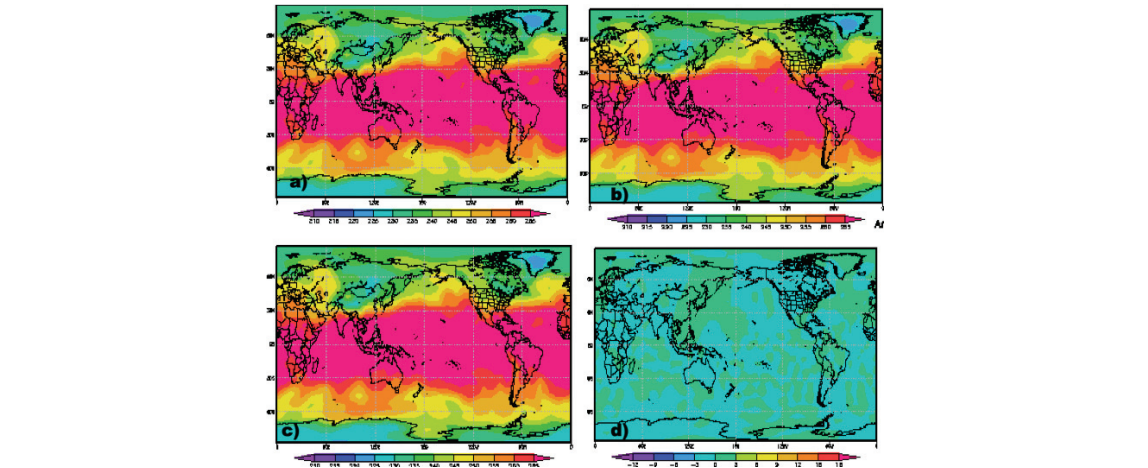


Figure 6. Absolute temperature (T) [K] at 500 hPa—Jan/11/1985 at 18 UTC (a) LETKF analysis, (b) ANN analysis, (c) true model, and (d) differences analysis.

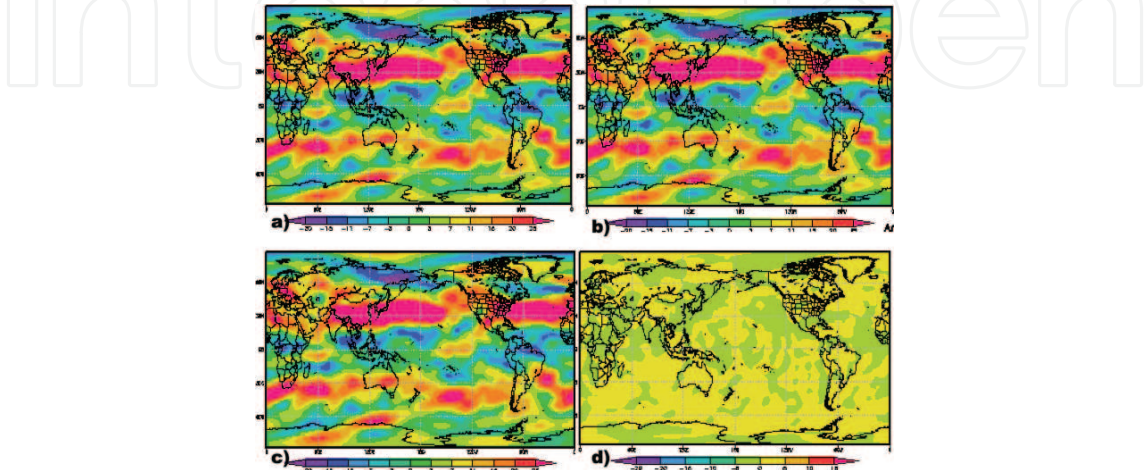


Figure 7. Zonal wind component (u) [m/s] at 500 hPa—Jan/11/1985 at 18 UTC. (a) LETKF analysis, (b) ANN analysis, (c) true model, and (d) differences analysis.

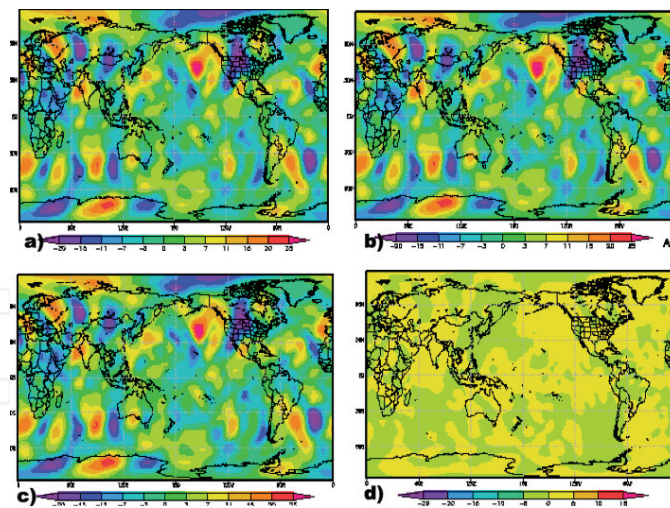


Figure 8. Meridional wind component (v) [m/s] fields at 500 hPa—Jan/11/1985 at 18 UTC (a) LETKF analysis, (b) ANN analysis, (c) true model, and (d) differences analysis.

Monthly average of absolute temperature analyses fields was obtained. The field of differences between the analyses (LETKF and MLP-DA) for data assimilation cycles is shown in **Figure 11**. The differences are slightly larger in some regions, such as the northeast regions of North America and South America.

The root mean square error (RMSE) of the absolute temperature analyses related to true model is calculated by fixing a point at longitude (87 W) for all latitude points. **Figure 12** shows the temperature RMSEs for the entire period of the assimilation cycle (January 1985). Subfigure (a) for **Figure 12** shows the RMSE of the LETKF analysis by line and the RMSE of the MLP-DA analysis by circles; and subfigure (b) for **Figure 12** shows the differences between LETKF and MLP-DA analyses RMSE. The differences are less than 10^{-3} .

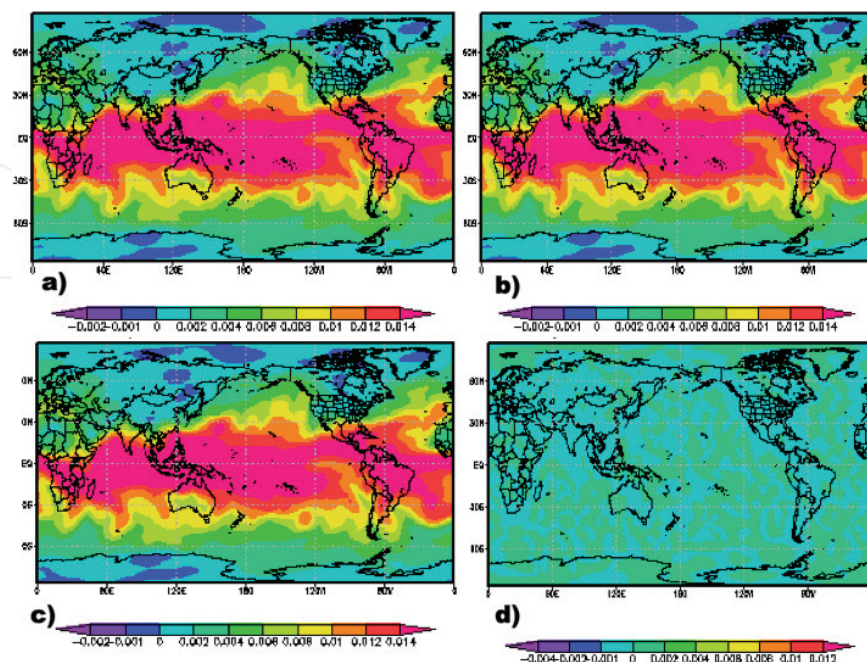


Figure 9. Specific humidity (q) at 950 hPa—Jan/11/1985 at 18 UTC (a) LETKF analysis, (b) ANN analysis, (c) true model, and (d) differences analysis.

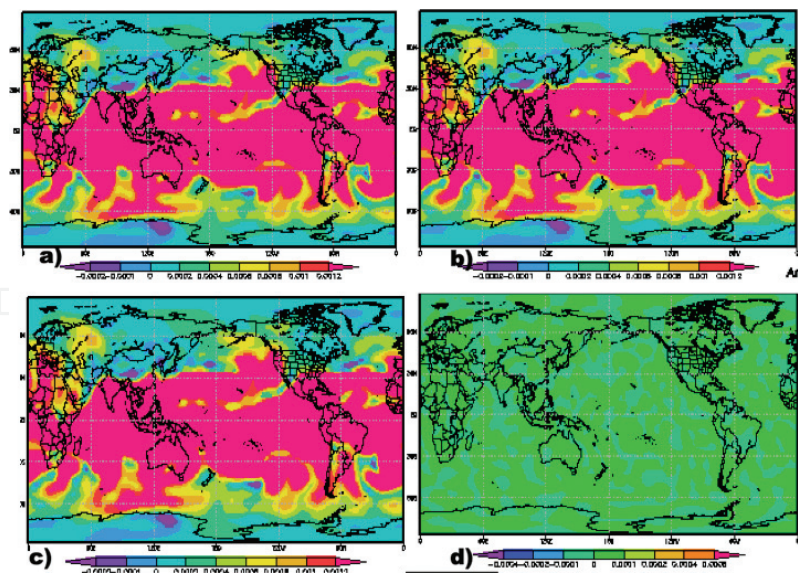


Figure 10. Specific humidity (q) at 950 hPa—Jan/11/1985 at 18 UTC (a) LETKF analysis, (b) ANN analysis, (c) true model, and (d) differences analysis.

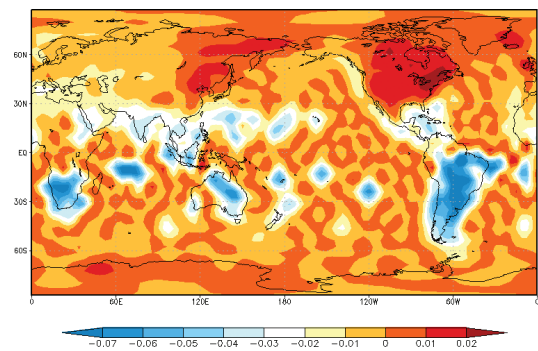


Figure 11. Differences field of the average of absolute temperature MLP-DA analysis and LETKF analysis for the assimilation cycle.

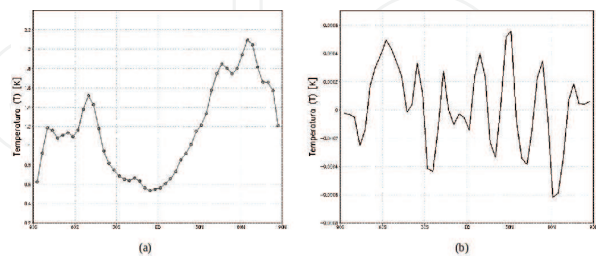


Figure 12. Meridional root mean square error for entire period of the assimilation cycles. RMSE analyses to the “true” state to (a) the errors of the LETKF analysis (line) and the errors of MLP-DA analysis (circles) to the absolute temperature at 500 hPa. (b) Differences of RMSE analyses.

4.1. Computer performance

Several aspects of modeling stress computational systems and push the capability requirements. These aspects include increased grid resolution, the inclusion of improved physics

processes and concurrent execution of earth-system components, and coupled models (ocean circulation and environmental prediction, for example).

Often, real-time necessities define capability requirements. In data assimilation, the computational requirements become much more challenging. Observations from Earth-orbiting satellites in operational numerical prediction models are used to improve weather forecasts. However, using the amount of satellite data increases the computational effort. As a result, there is a need for an assimilation method able to compute the initial field for the numerical model in the operational window time to make a prediction. At present, most of the NWP centers find it difficult to assimilate all the available data because of computational costs and the cost of transferring huge amounts of data from the storage system to the main computer memory.

The data assimilation cycle has a recent forecast and the observations as the inputs for assimilation system. The described MLP-DA system produced an analysis to initiate the actual cycle. This time simulation experiment is for January 1985 (28 days). There were 2075 observations inserted at runs of 0600 and 1800 UTC for surface variables, and 12,035 observations inserted at runs of 0000 and 1200 UTC for all upper layer variables.

The LETKF data assimilation cycle initiates running the ensemble forecasts with the SPEEDY model, and each analysis produced to each member at the latter LETKF cycle to result 30 (members) 6-hour forecasts; the second step is to compute the average of those forecasts. After, with a set of observations and the mean forecast, the LETKF system is performed. The LETKF cycle results one analysis to each member for the ensemble and one average field of the ensemble analyses. The MLP-DA data assimilation cycle is composed by the reading of 6-hour forecast of SPEEDY model from latter cycle and reading the set of observations to the cycle time, the division of input vectors, the activation of MLP-DA, and the assembly of output vectors to a global analysis field.

The MLP-DA runtime measurement initiates after reading the 6-hour forecast of SPEEDY model from latter cycle and the set of observations. The time of generalization includes the division of observation and prediction fields into regions, and the execution of the various trained networks by gathering all regions in a global analysis. It initiates after reading the mean 6-hours forecast of SPEEDY model and the set of observations. The LETKF time includes the results of 30 analyses and one mean ensemble analysis. The comparison in **Table 1** is the data assimilation cycles for the same observations points and the same model resolution to the same time simulations. LETKF and MLP-DA executions are performed independently. Considering the total execution time of those 112 cycles simulated, the computational performance of the MLP-DA data assimilation is better than that obtained with the LETKF approach. These results show that the computational efficiency of the NN for data assimilation to the SPEEDY model, for the adopted resolution, is 90 times faster and produces analyses of the same quality (**Table 1**). Considering only the analyses execution time of those 112 data assimilation processes simulated, the computational efficiency of MLP-DA is 421 times faster than LETKF process. **Table 2** shows the mean execution time of each element to one cycle of the LETKF data assimilation method (ensemble forecast and analysis) and the MLP-DA method (model forecast and analysis). The computational efficiency of one MLP-DA execution keeps the relationship about speed-up, comparing with one LETKF execution (421 times faster). Details for this experiment can be found in Ref. [6].

Execution of 112 cycles	MLP-DA (hour:min:sec)	LETKF (hour:min:sec)
Analysis time	00:00:25	03:14:55
Ensemble model time	00:00:00	01:05:44
Single model time	00:02:28	00:00:00
Total time	00:02:53	04:20:39

Table 1. Total running time of 112 cycles of complete data assimilation (analysis and forecasting).

Cycle element	hour:min:sec:lll
Ensemble (model)	00:00:35:214
Single model	00:00:01:174
LETKF analysis	00:01:37:463
NN analysis	00:00:00:231

Table 2. Mean time of one execution (hour:min:sec:lll).

5. Conclusions

In this study, we evaluated the efficiency of the MLP-DA in an atmospheric data assimilation context with a 3D global model. The MLP-DA is able to emulate systems for known data assimilation scheme. For the present investigation, the MLP-DA approach is used to emulate the LETKF method, which is designed to improve the computational performance. The another experiments with the same methodology can be found in [7, 8].

The NN learned the whole process of the LETKF scheme of data assimilation through training process. The results for the MLP-DA analyses are very close to the results obtained from the LETKF data assimilation for initializing the SPEEDY model forecast, i.e., the analyses obtained with MLP-DA are similar to analyses computed by the LETKF. The difference between MLP-DA and LETKF analyses to surface pressure fields belongs to interval $[-5, 5]$ hPa. However, the computational performance of the set of 30 NN is better than LETKF scheme. The MLP-DA accelerates the LETKF data assimilation computation.

The application of the present NN data assimilation methodology is under investigation at the Center for Weather Prediction and Climate Studies (Centro de Previsão de Tempo e Estudos Climáticos-CPTEC/INPE) with operational numerical global model and real observations. After investigation with Florida State University model made in 2014, the results are found in [41, 42].

Acknowledgements

The authors thank Dr. Takemasa Miyoshi and Prof. Dr. Eugenia Kalnay for providing computer routines for the SPEEDY model and the LETKF system. This paper is a contribution of

the Brazilian National Institute of Science and Technology (INCT) for Climate Change funded by CNPq Grant Number 573797/2008-0 e FAPESP Grant Number 2008/57719-9. Author HFCV also thanks to the CNPq (Grant number 311147/2010-0).

Author details

Rosangela Saher Cintra* and Haroldo F. de Campos Velho

*Address all correspondence to: rosangela.cintra@inpe.br

National Institute for Space Research, São José dos Campos, SP, Brazil

References

- [1] Andrieu C, Doucet A. Particle filtering for partially observed Gaussian state space models. *Journal of the Royal Statistical Society*. 2002;**64**(4):827-836
- [2] Bishop HC, Etherton BJ, Majumdar SJ. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly Weather Review*. 2001;**129**:420-436
- [3] Bourke W. A multilevel spectral model: I. Formulation and hemispheric integrations. *Monthly Weather Review*. 1974;**102**:687-701
- [4] Burgers G, van Leeuwen P, Evensen G. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*. 1998;**126**:1719-1724
- [5] Campos Velho HF, Vijaykumar NL, Stephany S, Preto AJ, Nowosad AG. A neural network implementation for data assimilation using MPI. In: Brebia CA, Melli P, Zanasi A, editors. *Application of High Performance Computing in Engineering*. Vol. Section 5. Southampton: WIT Press; 2002. p. 211-220
- [6] Cintra RS. Assimilação de dados por redes neurais artificiais em um modelo global de circulação atmosférica. D.Sc. dissertation on Applied Computing. São José dos Campos, Brazil: National Institute for Space Research; 2010
- [7] Cintra RS, Campos Velho HF. Data assimilation for satellite temperature by artificial neural network in an atmospheric general circulation model. XVII Congresso Brasileiro de Meteorologia, Gramado, RGS, Brasil, September 23rd–28th, 2012.
- [8] Cintra RS, Campos Velho HF, Furtado HC. Neural network for performance improvement in atmospheric prediction systems: Data assimilation. 1st BRICS Countries & 11th CBIC Brazilian Congress on Computational Intelligence. Recife, Brasil, “Porto de Galinhas” Beach, September 8th–11th, 2013.
- [9] Daley R. *Atmospheric Data Analysis*. New York, USA: Cambridge University Press; 1991. 471 pp

- [10] Doucet A, Godsill S, Andrieu C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*. 2002;**10**(3):197-208
- [11] Evensen G. Sequential data assimilation with a nonlinear quasigeostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysics Research*. 1994;**99**:10143-10162
- [12] Evensen G. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*. 2003;**53**:343-367
- [13] Furtado HC. Redes neurais e diferentes metodos de assimilação de dados em dinâmica não linear. São José dos Campos, Brazil: Brazilian National Institute for Space Research master thesis in Applied Computation program; 2008
- [14] Gardner M, Dorling S. Artificial neural networks, the multilayer perceptron. *Atmospheric Environment*. 1998;**32**(114/15):2627-2636
- [15] Greybush S. Mars Weather and Predictability: Modeling and Ensemble Data Assimilation of Space Craft Observations [Ph.D. thesis]. College Park, Maryland, USA: University of Maryland; 2005
- [16] Harter F, Campos Velho HF. Recurrent and feedforward neural networks trained with cross correlation applied to the data assimilation in chaotic dynamic. *Revista Brasileira de Meteorologia*. 2005;**20**:411-420
- [17] Harter FP, Campos Velho HF. New approach to applying neural network in nonlinear dynamic model. *Applied Mathematical Modeling*. 2008;**32**(12):2621-2633. DOI: 10.1016/j.apm.2007.09.006
- [18] Haupt SE, Pasini A, Marzban C. *Artificial Intelligence Methods in the Environmental Sciences*. 1st ed. Springer: Berlin, Heidelberg; 2009. 424 pp
- [19] Haykin S. *Redes neurais princípios prática*. Vol. 2. Porto Alegre: Editora Bookman; 2001
- [20] Haykin S. *Adaptive Filter Theory*. New Delhi, India: Dorling Kinderley; 2007
- [21] Held I, Suarez M. A proposal for the intercomparison of dynamical cores of atmospheric general circulation models. *Bulletin of American Meteorological Society*. 1994;**75**:1825-1830
- [22] Ho'lm EV. *Lecture Notes on Assimilation Algorithms*. Reading, UK: European Centre for Medium-Range Weather Forecasts; 2008. 30 pp
- [23] Houtekamer PL, Mitchell HL. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*. 1998;**126**:796-811
- [24] Hsieh W, Tang B. Applying neural network models top prediction and data analysis in meteorology and oceanography. *Bulletin of the American Meteorological Society*. 1998;**79**(9): 1855-1870
- [25] Hsieh WW. *Machine Learning Methods in the Environmental Sciences: Neural Networks and Kernels*. Cambridge University Press: New York, NY, USA; 2009. 349 pp

- [26] Hunt B, Kostelich EJ, Szunyogh I. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D*. 2007;**230**:112-126
- [27] Kalman RE. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*. 1960;**82**(Series D):35-45
- [28] Kalman RE, Bucy RS. New results in linear filtering and prediction theory. *Transactions of the ASME Journal of Basic Engineering*. 1961;**83**(Series D):95-108
- [29] Kalnay E. *Atmospheric Modeling, Data Assimilation and Predictability*. 2nd ed. New York: Cambridge University Press; 2003
- [30] Liaqat A, Fukuhara M, Takeda T. Applying a neural collocation method to an incompletely known dynamical system via weak constraint data assimilation. *Monthly Weather Review*. 2003;**131**(8):1697-1714
- [31] Lorenc AC. Analysis methods for numerical weather prediction. *Quarterly Journal of Royal Meteorological Society*. 1986;**112**:1177-1194
- [32] Miyoshi T. *Ensemble Kalman Filter Experiments with a Primitive Equation Global Model [PhD thesis]*. College Park, Maryland, USA: University of Maryland; 2005. 197 p
- [33] Miyoshi T, Kunii M. The local ensemble transform Kalman filter with the weather research and forecasting model: Experiments with real observations. *Pure and Applied Geophysics*. 2012;**169**:321-333
- [34] Miyoshi T, Sato Y, Kadowaki T. Ensemble Kalman filter and 4D-var intercomparison with the Japanese operational global analysis and prediction system. *Monthly Weather Review*. 2010;**138**:2846-2866
- [35] Miyoshi T, Yamane S. Local ensemble transform Kalman filtering with an AGCM at a T159/L48 resolution. *Monthly Weather Review*. 2007;**135**:3841-3861
- [36] Molteni F. Atmospheric simulations using a GCM with simplified physical parametrizations. I: Model climatology and variability in multi-decadal experiments. *Climate Dynamics*. 2003;**20**:175-191
- [37] Nowosad A, Neto AR, Campos Velho H. Data assimilation in chaotic dynamics using neural networks. *International Conference on Nonlinear Dynamics, Chaos, Control and Their Applications in Engineering Sciences*. 2000:212-221
- [38] Ott E, Hunt BR, Szyniogh I, Zimin AV, Kostelich EJ, Corazza M, Kalnay E, Patil DJ, York J. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A*. 2004;**56**:415-428
- [39] Penny S. *Data Assimilation of the Global Ocean using the 4D Local Ensemble Transform Kalman Filter (4D-LETKF) and the Modular Ocean Model (MOM2) [PhD]*.
- [40] Furtado HCM, Cintra RSC, Campos Velho HF. Artificial neural networks emulating representer method at shallow water model 2D. In: *DINCON Conferência Brasileira de Dinâmica*. 2015

- [41] Cintra RS, Campos Velho HF, Furtado HC. Neural network for performance improvement in atmospheric prediction systems: Data assimilation. 1st BRICS Countries & 11th CBIC Brazilian Congress on Computational Intelligence. Recife, Brasil, Porto de Galinhas Beach, September 8th–11th, 2013.
- [42] Cintra RSC, Campos Velho HF, Anochi J, Cocke S. Data assimilation by artificial neural networks for the global FSU atmospheric model: Surface pressure. In: 2nd Latin-American Congress on Computational Intelligence (LA-CCI). 2015. Curitiba. CBIC and LA-CCI 2015.
- [43] Cintra RSC, Cocke S. A local ensemble transform Kalman filter data assimilation system for the global FSU atmospheric model. *Journal of Mechanics Engineering and Automation*. 2015;5:185-196
- [44] Vijaykumar NL, Campos Velho HF, Stephany S, Preto AJ, Nowosad AG. Optimized neural network code for data assimilation. *Brazilian Congress on Meteorology*, 12, 38413849. 2002
- [45] Widrow B, Hoff M. Adaptive switching circuits. *IRE WESCON Conv. Record*, Pt.4.94, 95, 96-104. 1960