Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematical Modelling 32 (2008) 2621–2633
APPLIED MATHEMATICAL MODELLING
www.elsevier.com/locate/apm

New approach to applying neural network in nonlinear dynamic model

Fabrício P. Härter *, Haroldo Fraga de Campos Velho

Instituto Nacional de Pesquisas Espaciais, Laboratório Associado de Computação e Matemática Aplicada, São José dos Campos, SP, Brazil

Received 2 January 2007; received in revised form 31 July 2007; accepted 17 September 2007
Available online 30 October 2007

Abstract

In this work, radial basis function neural network (RBF-NN) is applied to emulate an extended Kalman filter (EKF) in a data assimilation scenario. The dynamical model studied here is based on the one-dimensional shallow water equation DYNAMO-1D. This code is simple when compared with an operational primitive equation models for numerical weather prediction. Although simple, the DYNAMO-1D is rich for representing some atmospheric motions, such as Rossby and gravity waves. It has been shown in the literature that the ability of the EKF to track nonlinear models depends on the frequency and accuracy of the observations and model errors. In some cases, just fourth-order moment EKF works well, but will be unwieldy when applied to high-dimensional state space. Artificial Neural Network (ANN) is an alternative solution for this computational complexity problem, once the ANN is trained offline with a high order Kalman filter, even though this Kalman filter has high computational cost (which is not a problem during ANN training phase). The results achieved in this work encourage us to apply this technique on operational model. However, it is not yet possible to assure convergence in high dimensional problems.

Keywords: Dynamo model; Data assimilation; Extended Kalman filter; Artificial neural network; Radial base function

1. Introduction

After 1950, the numerical weather prediction (NWP) became an operational procedure. Essentially, NWP consists of the time integration of the Navier–Stokes equation using numerical procedures. Therefore, after some time-steps, there is a disagreement due to small uncertainties in the specification of the initial conditions (IC) and model errors. In other words, the sensitive dependence on the IC causes the forecasting error to grow exponentially with time [1]. NWP is an initial value problem, and this implies that a better representation of the IC will produce a better prediction.

* Corresponding author. Tel.: +55 061 3343 1719; fax: +55 061 3343 1619.
E-mail address: fabricio.harter@inmet.gov.br (F.P. Härter).

0307-904X/$ - see front matter
doi:10.1016/j.apm.2007.09.006

The problem of estimating the IC is so complex and important that it becomes a science called *Data Assimilation* [2]. If the insertion of the data noise into an inaccurate computer model corrupted the prediction (see Fig. 1), then it is important to apply some data assimilation technique.

Many methods have been developed for data assimilation [2]. They have different strategies to combine the forecasting (*background*) and observations and differ in their numerical cost, their optimality, and their suitability for real-time applications. From a mathematical point of view, the assimilation process can be represented by

$$x^{a} = x^{f} + \theta p[y^{o} - H(x^{f})], \tag{1}$$

where x^{a} is the value of the analysis; x^{f} is the forecasting (from the mathematical model, also known as background field); θ is the weight matrix, generally computed from the covariance matrix of the prediction errors from forecasting and observation; y^{o} denotes the observation; H represents the observation system; $\{y^{o} - H(x^{f})\}$ is the innovation; and $p[.]$ is a discrepancy function.

Extended Kalman filter (EKF) and four-dimensional variational analysis (4D-Var) are the state of the art in data assimilation techniques. The EKF is a development of the least-squares analysis method in the framework of sequential data assimilation, in which each background is provided by a forecast that starts from a previous analysis (see Section 3). It is adapted to the real-time assimilation of observation distributed in time into a forecast model [3]. In cases where nonlinearities are important as in weather forecasting, it may be necessary to include empirical correction terms in the equation [4], or to use a more general stochastic prediction method such as an ensemble prediction (or Monte Carlo) method [5,6], which yields an algorithm known as the ensemble Kalman filter.

Lister et al. [7] quantified the computational complexity and parallel scalability of the Kalman filter at NASA's Global Modeling and Assimilation Office (GMAO). These authors show that for atmospheric assimilation, the gridded dynamical fields typically have more than 10^{6} variables; therefore, the full error covariance matrix may be in excess of a teraword. For the Kalman filter this problem will require petaflops s^{-1} of computing to achieve an effective throughput for scientific research. It means that the Kalman filter approach may become useless as the resolution of the numerical models is increased and/or the observational net is improved.

An alternative to solve this computational cost is to emulate an EKF through an artificial neural network (ANN), once the ANN could be trained offline with "the best EKF" (computational cost is not a problem during ANN training phase). However, it is not yet possible to assure convergence in high dimensional problems.

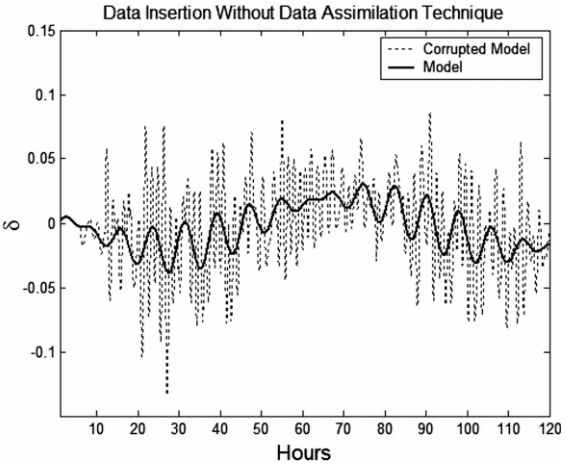

Fig. 1. Shock in the numerical model after data insertion, without data assimilation technique.

The use of ANN for data assimilation is a very recent issue. ANNs were suggested as a possible technique for data assimilation by Hsieh and Tang [8], but the first implementation of the ANN in order to emulate a KF in a data assimilation scenario was employed by Nowosad et al. [9] (see also Campos Velho et al. [10]). The ANN has also been used in the works of Liaqat et al. [11] and Tang and Hsieh [12] for data assimilation. The technique developed by Nowosad et al. [9] is quite different from the latter two works, where the ANN is used to represent an unknown equation in the mathematical system equations of the model. However, the Liaqat et al. [11] and Tang and Hsieh [12] approaches show the good skill of the ANN to emulate a dynamical system. Indeed, this ability is fully explored when ANNs are employed as a black-box as a prediction tool for time series and/or substituting a dynamical system.

In contrast to previous works using ANN for data assimilation, this paper deals with radial base function neural network (RBF-NN). In addition, cross validation is used as a learning process. The best improvement in this work compared with later works is that here the ANNs are implemented to decrease the dimension of the problem as discussed in Section 5. The dynamic model (DYNAMO-1D) is applied as a test model for assimilating noise data.

This paper is structured as follows: Section 2 describes the DYNAMO-1D model; Section 3 presents a basic introduction to EKF; in Section 4 there is a formulation for RBF-NN; numerical results are shown in Section 5; and finally Section 6 contains the concluding remarks.

2. The meteorological model DYNAMO

The dynamic model 1D was derived by Lynch and Huang [13] to test assimilation and initialization methods. Although the DYNAMO model is based on shallow-water approach, it represents important atmospheric motions, such as Rossby and gravity waves. The physical equations of the model are

$$\frac{\mathrm{d}u}{\mathrm{d}t} - fv + \frac{\partial \phi}{\partial x} = 0, \tag{2}$$

$$\frac{\mathrm{d}v}{\mathrm{d}t} + fu + \frac{\partial \phi}{\partial y} = 0, \tag{3}$$

$$\frac{\mathrm{d}\phi}{\mathrm{d}t} + \phi(u_x + v_y) = 0, \tag{4}$$

where u is zonal wind speed (x-coordinate), v is meridional wind speed (y-coordinate), t is time, $\phi = \int_{h_0}^{h_s} g\,\mathrm{d}z$ is the geopotential height, where $h_0 = 0$ and $h_s = 8$ km, and $f = f_0 + \beta y$ is the Coriolis force for β-plane approximation [14]. The space domain is a channel: $x \in [0, L]$, and periodic boundary conditions are assumed for all variables.

Assuming that the mean wind (zonal wind) is constant, and it is geostrophically balance with the geopotential ($f\bar{u} = -\partial\bar{\phi}/\partial y$), with the deviations (perturbation) U, V, Φ given by

$$u = \bar{u} + U(x,t); \quad v = V(x,t); \quad \phi = \bar{\phi}(y) + \Phi(x,t);$$

and using the perturbation method, these equations are manipulated to produce a one-dimensional version of the shallow-water equations, whose prognostic variables are vorticity $\varsigma = v_x$ and divergence $\delta = u_x$, given by the curl and by the divergence of the motion equations, respectively, and geopotential height. This manipulation reduces the dimension of the model, because the y dependence was eliminated and prognostic variables $(\varsigma, \delta, \Phi)$ have just x and t dependence, as shown below:

$$\varsigma_t + (u\varsigma)_x + \delta f + \beta V = 0, \tag{5}$$

$$\delta_t + (u\delta)_x - \varsigma f + \beta U + \Phi_{xx} = 0, \tag{6}$$

$$\Phi_t + (u\Phi)_x - f\bar{u}V + \bar{\phi}\delta = 0. \tag{7}$$

The Helmholtz theorem permits the splitting of the horizontal wind field $\vec{V} = U\vec{i} + V\vec{j}$ into non-divergent and non-rotational parts:

$$\vec{V}_H = \vec{V}_\chi + \vec{V}_\psi = \vec{k} \times \vec{\nabla}\psi + \vec{\nabla}\chi, \tag{8}$$

where ψ is the current function and χ is the potential velocity, thus

$$\nabla^2\psi = \varsigma, \tag{9}$$

$$\nabla^2\chi = \delta. \tag{10}$$

Consequently, the horizontal wind components are obtained from

$$V = \nabla\psi, \tag{11}$$

$$U = \nabla\chi. \tag{12}$$

Then, in each time-step, two Poisson equations are solved. In the next step the system Eqs. (5)–(7) are linearized and presented as a matrix prognostic equation:

$$\frac{\partial W}{\partial t} + AW + BX = -R_0 N_W(W, X), \tag{13}$$

where the state vectors are

$$W = [\varsigma \quad \delta \quad \Phi]^{\mathrm{T}}, \tag{14}$$

$$X = [V \quad U \quad \Phi]^{\mathrm{T}}, \tag{15}$$

and the matrix coefficients and nonlinear terms are given below:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & \frac{\partial^2}{\partial x^2} \\ 0 & R_F & 0 \end{bmatrix}, \tag{16}$$

$$B = \begin{bmatrix} R_\beta & 0 & 0 \\ 0 & R_\beta & 0 \\ -R_0 u_0 & 0 & 0 \end{bmatrix}, \tag{17}$$

$$N_W(W, X) = [N_W^\varsigma \quad N_W^\delta \quad N_W^\Phi]^T, \quad N_W^\alpha = (u\alpha)_x, \text{ with : } \alpha = \varsigma, \delta, \Phi. \tag{18}$$

The relation between the prognostic variables vector and the primitive variables vectors is given by

$$W = MX, \tag{19}$$

where M is a first order linear differential operator

$$M = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial X & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{20}$$

Substituting Eq. (19) in Eq. (13) gives the primitive formulation

$$M\frac{\partial X}{\partial t} + (AM + B)X = -R_0 N(X), \tag{21}$$

and the equivalent nonlinear operator is

$$N(X) = \frac{\partial(X_2 MX)}{\partial x}. \tag{22}$$

The numerical model is formulated with a two step method and periodic boundary condition. A complete formulation for this problem is found in [15].

3. Kalman filter

The Kalman filter is the best linear unbiased estimator, and the adaptation of the Kalman filter technique to a nonlinear system is called the extended Kalman filter (EKF). This technique is frequently applied in estima-

tion and control problems. Nowadays this filter has been applied in hydrology, oceanography and meteorology.

In this work the ANN emulates the EKF. A brief description of this filter will be outlined here, as in [16]. Let the prediction nonlinear model be

$$X^{\mathrm{f}}_{n+1} = f_n(X^{\mathrm{f}}_n) + \mu_n, \tag{23}$$

where f_n is a mathematical description of the nonlinear system, μ_n is a stochastic forcing (dynamic modeling noise), and the observation model is given by

$$Z^{\mathrm{f}}_n = h_n(X^{\mathrm{f}}_n) + v_n, \tag{24}$$

where v_n is the observation noise. The typically Gaussian, zero-mean and orthogonality hypotheses for the noise are adopted. $f_n(X^{\mathrm{f}}_n)$ and $h_n(X^{\mathrm{f}}_n)$ are nonlinear functions of X_n. The following approximation is used:

$$f_n(X_n) = f_n(X^{\mathrm{a}}_n) + F_n(X_n - X^{\mathrm{a}}_n), \tag{25}$$

$$h_n(X_n) = h_n(X^{\mathrm{a}}_n) + H_n(X_n - X^{\mathrm{f}}_n), \tag{26}$$

where F_n and H_n are given by

$$F_n = \left.\frac{\partial f_n}{\partial X}\right|_{X=X^{\mathrm{a}}_n}, \tag{27}$$

$$H_n = \left.\frac{\partial h_n}{\partial X}\right|_{X=X^{\mathrm{f}}_n}. \tag{28}$$

Thus (23) and (24) could be rewritten as

$$X^{\mathrm{f}}_{n+1} = F_n X^{\mathrm{f}}_n + \mu_n, \tag{29}$$

$$Z_n = H X^{\mathrm{f}}_n + v_n. \tag{30}$$

The term X_{n+1} is estimated through the recursion

$$X^{\mathrm{a}}_{n+1} = (I - G_{n+1}H_{n+1})F_n X^{\mathrm{a}}_n + G_{n+1}Z^{\mathrm{f}}_{n+1}, \tag{31}$$

where X^{a}_{n+1} is the estimator and G_n is the matrix that minimizes the trace of the prediction error covariance matrix, that is, the sum of the squares of the prediction errors in each component of X_{n+1}:

$$J_{n+1} = E\{(X^{\mathrm{a}}_{n+1} - X^{\mathrm{f}}_{n+1})^T (X^{\mathrm{a}}_{n+1} - X^{\mathrm{f}}_{n+1})\}. \tag{32}$$

The algorithm of the Kalman filter is shown in Fig. 2, where Q_n is the covariance of μ and R_n is the covariance of v.

The assimilation is done through the sampling:

$$r_{n+1} \equiv Z_{n+1} - Z^{\mathrm{f}}_{n+1} = Z_{n+1} - H_n X^{\mathrm{f}}_{n+1}. \tag{33}$$

4. Neural networks

Gardner and Dorling [17] did a survey on applications of the ANN in meteorology, where a brief introduction about ANN and the back-propagation algorithm are shown. It also cited applications in the atmospheric sciences looking at prediction (air-quality: surface ozone concentration, sulphur dioxide concentrations; severe weather; Indian monsoon, Brazilian rainfall anomalies, solar radiation), function approximation (air-quality, modeling of nonlinear transfer functions), and pattern classification (cloud classification; distinction between clouds and ice or snow; classification of atmospheric circulation patterns; land cover classification; classification of convergence lines from radar imagery; etc). Although, the use of ANN for data assimilation can be understood as function approximation, this application was not mentioned in Gardner and Dorling's paper [17].

An artificial neural network (ANN) is an arrangement of units characterized by

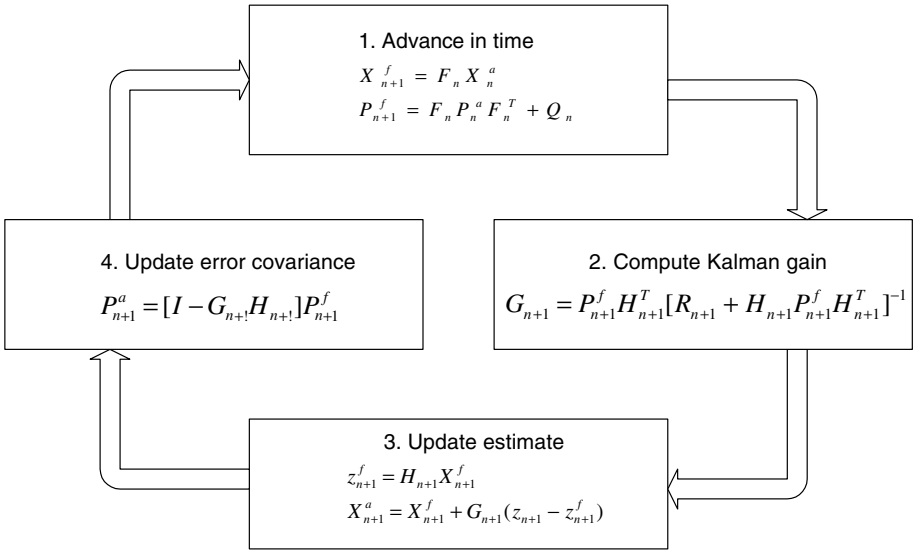

Fig. 2. Schematic diagram of the linear Kalman filter.

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

The processing element (unit) in an ANN is a linear combiner with multiple weighted inputs, followed by an activation function. There are several different architectures of ANN, most of which directly depend on the learning strategy adopted. Two distinct phases can be devised while using an ANN: the training phase (learning process) and the run phase (activation of the network). The training phase consists of adjusting the weights for the best performance of the network in establishing the mapping of many input/output vector pairs. Once trained, the weights are fixed and the network can be presented to new inputs for which it calculates the corresponding outputs, based on what it has learned. The back-propagation algorithm is used as the learning process for the ANN topology studied here [18].

The kth neuron can be described by the two coupled equations

$$w_k = \sum_{j=1}^{m} \theta_{kj} \widehat{X}_j, \tag{34}$$

$$X_{\mathrm{NN},k} = \varphi(w_k + b_k), \tag{35}$$

where $\widehat{X}_1, \ldots, \widehat{X}_m$ are the inputs; $\theta_{kl}, \ldots, \theta_{km}$ are the connection weights for the neuron-k, w_k is the linear output of the linear combination among weighted inputs, b_k is the bias; $\varphi(\cdot)$ is the activation function, and $X_{\mathrm{NN},k}$ is the neuron output. For the present application $\widehat{X} = \left[\widehat{X}_1 \quad \cdots \quad \widehat{X}_m\right]^{\mathrm{T}}$ with $\widehat{X}_j = \left[X^{\mathrm{o}}_j \quad X^{\mathrm{f}}_j\right]^{\mathrm{T}}$, being $X^{\omega}_j = \left[V^{\omega}_j \quad U^{\omega}_j \quad \phi^{\omega}_j\right]^{\mathrm{T}}$ with the superscript ω denoting observation (o) or forecasting (f), and m the number points in the discrete space grid. Finally, $X_{\mathrm{NN}} = \left[X^{\mathrm{a}}_{\mathrm{NN},1} \quad \cdots \quad X^{\mathrm{a}}_{\mathrm{NN},m}\right]^{\mathrm{T}}$ is the analysis vector, with $\widehat{X}^{\mathrm{a}}_{\mathrm{NN},j} = \left[V^{\mathrm{a}}_j \quad U^{\mathrm{a}}_j \quad \phi^{\mathrm{a}}_j\right]^{\mathrm{T}}$ and the superscript 'a' denoting analysis.

For driving us to design the appropriate topology for the ANN, we try to use the same standard applied for data assimilation in Lorenz's system by Hartër and Campos Velho [18]. In reference [18], the authors investigated different types and topologies for data assimilation: multi-layer perceptron, radial base function, Elman neural network, and Jordan neural network; considering one and two hidden layers with several number of neurons. For that experiment, convergence was noted for all ANN and some topologies studied, where good results were obtained with one hidden layer, and with two hidden layers the results were not improved.

ANN has been applied in several applications: function approximation, time series prediction, and control [19]. An ANN is a good strategy for interpolation procedures, but they can produce wrong predictions when used for extrapolation. In meteorology some patterns can be identified, due to the diurnal cycle, the cycle of the seasons, allowing to recognize some periodic patterns for climate and the weather. However, this is not a permanent role, and for this reason, the weather prediction remains as a challenge, coming from the stochastic components in the dynamics and phenomena not yet well understood or studied or modeled. In general, it is hard to prove that an ANN can fully emulate a nonlinear dynamical system: one needs to verify on the specific case.

4.1. Radial base function neural networks

Radial basis function (RBF) are employed as a procedure as a function approximation or interpolation, and it is a real-value depending only on the distance from point $\mathbf{c}$: $\psi(\mathbf{x},\mathbf{c}) = \psi(\|\mathbf{x}-\mathbf{c}\|)$. A function satisfying such property is a radial function, and the norm is usually the Euclidean distance. The interpolation equation using a RBF approximation is given by

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \psi(\|\mathbf{x}-\mathbf{c}_i\|), \tag{36}$$

where w_i are weights for the interpolation. The sum can also represent an artificial neural network: a radial base function neural network (RBF-NN), and the RBF $\psi(\|\mathbf{x}-\mathbf{c}_i\|)$ is the *activation function* in the jargon of the ANN. The RBF-NNs are feedforward networks with typically one hidden layer only, whose aim is to extract high order statistics from the input data [19]. They have been developed for data interpolation in multidimensional space [20]. RBF-NN can also learn arbitrary mappings. RBF-NN hidden layer units have a receptive field, which has a center, that is, a particular input value at which they have a maximal output. Their output tails off as the input moves away from this point. Fig. 3 depicts an RBF-NN, with one hidden layer.

The weights for the interpolation (36) can be computed when a set of points of the function f is known. The collocation procedure can be applied to determine the interpolation weights, by solving the linear algebraic system:

$$\boldsymbol{\Psi}\mathbf{w} = \mathbf{f} \tag{37}$$

being $\boldsymbol{\Psi}_{ji} = \psi(\|\mathbf{x}_j - \mathbf{c}_i\|)$, $\mathbf{w} = [\,w_1 \quad w_2 \quad \cdots \quad w_N\,]^{\mathrm{T}}$, and $\mathbf{f} = [\,f(\mathbf{x}_1) \quad f(\mathbf{x}_2) \quad \cdots \quad f(\mathbf{x}_M)\,]^{\mathrm{T}}$. Micchelli [21] has proved the following theorem:

Theorem 1. *for a set of distinct points* $\{\mathbf{x}\}_{i=1}^{N}$, *the N-by-N interpolation matrix* Ψ, *with jith entry* $\Psi_{ji} = \psi(\|\mathbf{x}_j - \mathbf{c}_i\|)$, *is a nonsingular matrix.*

A large class of functions follows the Micchelli's theorem, among them:

1. Gaussian functions:$\psi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right)$, for some $\sigma > 0$ and $r \in \Re$;
2. Multiquadrics:$\psi(r) = (r^2 + c^2)^{\alpha}$, for some $c > 0, \alpha > 0$, and $r \in \Re$;
3. Inverse multiquadrics:$\psi(r) = \frac{1}{(r^2+c^2)^{\beta}}$, for some $c > 0$, $0 < \beta < 1$, and $r \in \Re$;
4. Linear:$\psi(r) = r$, for some $r \in \Re$.

The activation function used here in the RBF-NN is a Gaussian function. In deed, Poggio and Girosi [22] have shown that kernel estimation methods, such as Parzen windows [23], can be regarded as special cases of the regularization networks.

4.1.1. The learning process

The learning process is the procedure to identify the weights of the approximation (36). A supervised learning is an ill-posed hypersurface reconstruction problem [19,22]. A general scheme for solving an ill-posed problem is the regularization theory [23]. Poggio and Girosi [22] have shown that the determination of the

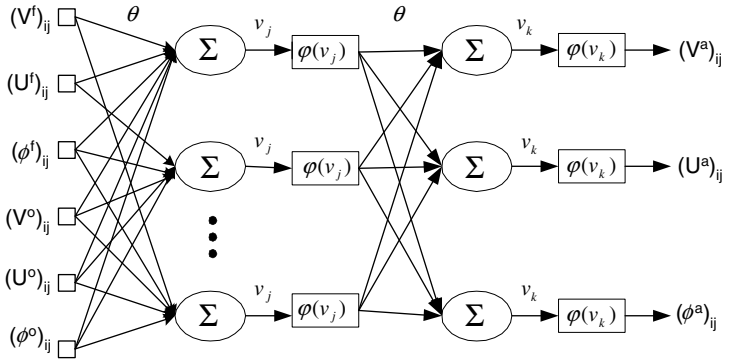

Fig. 3. Outline for neural network.

expansion coefficients in Eq. (36) is theoretically understood as an optimum solution of a regularized functional:

$$J(\mathbf{w}) = \sum_{j=1}^{N_x} [f^{\mathrm{Obs}}(\mathbf{x}_j) - f(\mathbf{x}_j, \mathbf{w})]^2 + \lambda \Omega(\mathbf{w}), \tag{38}$$

where the first term of the right hand side is the square difference between the observed value (target) and the outputs from the RBF-NN, λ is the regularization parameter, and $\Omega(\mathbf{w})$ the regularization operator. The Tikhonov regularization is $\Omega(\mathbf{w}) = \|D\mathbf{w}\|^2$, where D is a linear differential operator. The Fréchet derivative is applied for computing the minimum of the functional (38), and the Green's functions with centers $\mathbf{c}_i$ (multivariate Gaussian function [19]) satisfying the Euler–Lagrange equation for the Tikhonov functional $J(\mathbf{w})$. For details of this derivation the reader can see references [19,22]. This procedure leads to a class of the *regularization networks* [22], that it includes the RBF-NN as a special case. Regularization networks are closely related to the several neural networks algorithms, as back-propagation [22].

The strategy of this paper for learning process is to use RBF-NN with Gaussian function, employing the back-propagation algorithm [19] to compute the weight of the connections.

5. Numerical experiments

The experiments are performed using the following settings for the Kalman filter:$H_n = I$, $Q_n = (0.1 rand)^2 I$, $P_0^{\mathrm{f}} = 0$ and $F_n = I$. $R_n = r_{\max} I$, where $r_{\max} = \max_i[r_i^2(n)]$ and $r_n = Z_n - Z_n^{\mathrm{f}}$. The DYNAMO-1D is integrated with $\Delta t = 60$ s. The data insertion is done every 3 h (360 time-steps). For the training data set, 600 data are considered (from -3700 h up to -100 h), and 6000 data are used for cross validation (from -100 up to 0 h). After the training, the model is integrated for 180 days (from 0 up to 4320 h) as shown Fig. 4. The observation was obtained by adding Gaussian random noise in the numerical model.

In order to test the RBF-NN for emulating an EKF in data assimilation scenario, we used the following learning rates in the hidden and output layers: $\eta_{\widehat{X}} = 0.1$ (the same ratio used for all input parameters: $\widehat{X}_j^{\omega}$). The number of neurons in a hidden layer, number of layers and the learning rate as well was obtained by numerical experimentation.

During the learning process, the cost function (the square difference between the ANN output and the target data) is decreasing (it is not shown here), but this does not mean that the ANN will have an effective generalization. On the other hand, the minimization can fall in a *local minimum* of the error surface. Appropriate momentum constant and learning ratios can avoid these local minima, and the cross validation is an alternative to achieve the best set of weights, corresponding to the best generalization.

Cross validation consists of splitting the target data set into two sub-sets, one for training and another one for validation. For each iteration, the connection weights are tested with the second data set to evaluate the

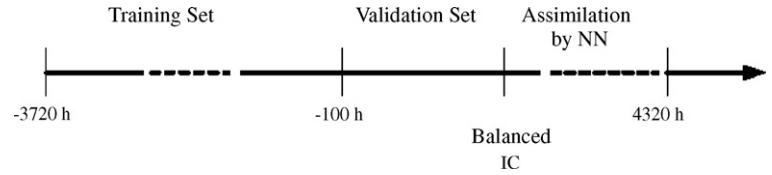

Fig. 4. Schematic diagram of the assimilation system.

ANN skill for generalization. In our experiments, the number of iterations was fixed at 6000, and cross validation was used to keep the best weight set for the generalization.

The training, cross validation, and assimilation errors are computed for each epoch n as follows at the middle point of the space domain ($L/2$):

$$\mathrm{TE}_n = \frac{1}{600}\sum_{i=1}^{600}\frac{1}{2}(C_i^{\mathrm{KF}} - C_i^{\mathrm{NN}})^2; \tag{39}$$

$$\mathrm{VE}_n = \frac{1}{6000}\sum_{k=1}^{6000}\sqrt{(C_k^{\mathrm{KF}} - C_k^{\mathrm{NN}})^2}; \tag{40}$$

$$\mathrm{AE}_n = \frac{1}{10^3}\sum_{k=1}^{10^5}\sqrt{(C_k^{\mathrm{Obs}} - C_k^{\mathrm{NN}})^2}; \tag{41}$$

where ($C = U, V, \Phi, \varsigma, \delta$). Results shown below are obtained with only one hidden layer. However, some experiments were also performed considering two hidden layers looking for a better performance. However, the results are quite similar to those obtained with one hidden layer. Therefore, results with two hidden layers will not be discussed.

The best architecture of this ANN is obtained with five neurons in the hidden layer and the smallest activation error was obtained at the 612th epoch, although the training errors decrease abruptly in the second epoch.

RBF-NN presents a good performance with $\mathrm{AE}_U = 0.0037$, $\mathrm{AE}_V = 0.0027$, $\mathrm{AE}_\phi = 0.0136$, $\mathrm{AE}_\varsigma = 0.0050$, $\mathrm{AE}_\delta = 0.00888$ and average error over all variables $\overline{\mathrm{AE}} = 0.0068$ for components U, V, Φ, ς, δ, and the average error, respectively. Fig. 4 shows the performance as ς varies.

Fig. 5 depicts the last 5 days of the total integration (180 days), where the data assimilation is performed by reference model (letter a) and by the RBF-NN with one hidden layer (letter b). Although there is some noise dumping caused by the dynamic relaxing the RBF-NN is effective in carrying out the assimilation. Fig. 6 shows the divergence (more noise among tested variables) field for the first day of DYNAMO integration, reference model (upper panel) and RBF-NN (lower panel). This figure shows some short waves in the domain of integration. It is pointed out that the flow was split in two parts, first one just divergent and another just rotational. Rossby waves are responsible for the rotational part of the flow and gravity waves are responsible for the divergent part of the flow. These short waves in the divergent field should be removed by an initialization technique such as digital filter, although the temporal evolution of divergent kinetic energy depicted in Fig. 7, shows that the filter works well. Fig. 8 shows the temporal evolution of the assimilation error for EKF and RBF-NN. It is shown that although the RBF-NN approach avoids the instability depicted in Fig. 1, the EKF is more precise.

In addition the cross-validation approach, the reduction of the dimension is the best novelty in this work. This new feature became the ANN competitive with other methods in data assimilation for an operational environment. In [9], where the ANN was applied in the same sense as in this work, the ANN architecture has 120 inputs (forecasting and observation of the U, V and Φ for 20 grid points), 80 neurons in the hidden layer and 60 neurons in output layer (U, V, and Φ estimated by EKF for 20 grid points). It means that 14800 weights must be found during the training phase. In the version explored in this paper, there are (for the same DYNAMO-1D model) 6 inputs (forecasting and observation of U, V, and Φ for just 1 grid point), 15 neurons

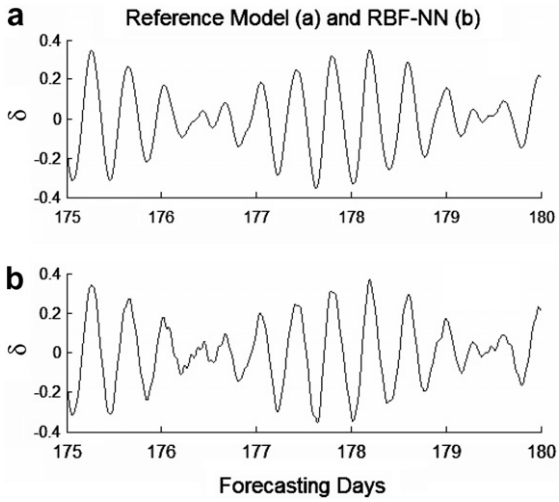

Fig. 5. Data assimilation for the DYNAMO-1D model using RBF-NN: component-δ. Temporal evolution in a grid central point. (a) Reference model and (b) RBF-NN.

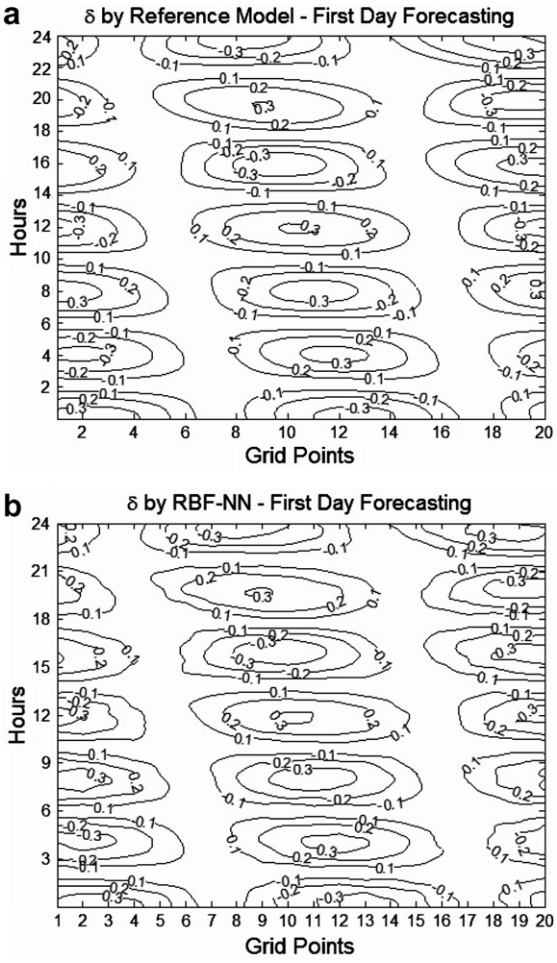

Fig. 6. Diverge field for the DYNAMO-1D model. (a) Reference model and (b) RBF-NN.

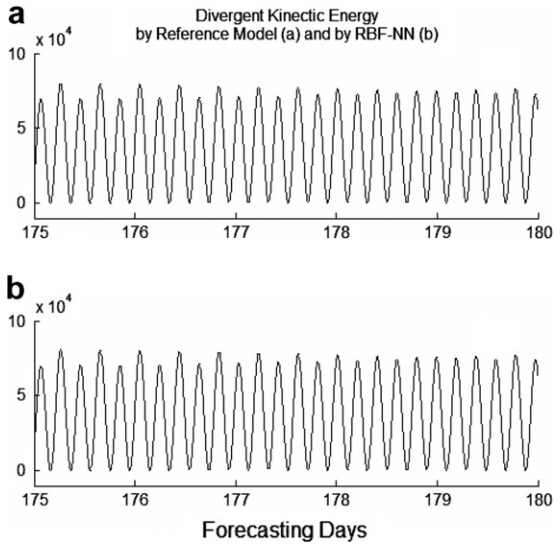

Fig. 7. Data assimilation for the DYNAMO-1D model using RBF-NN: component-δ. Temporal evolution in a grid central point of divergent kinetic energy. Reference model (upper panel) and RBF-NN (lower panel).

in the hidden layer in RBF-NN and 3 neurons in output layer (U, V, and Φ estimated by EKF for each grid point), which means that 135 connection weights must be found during training phase.

Global operational numerical models deal about 10^6 grid points. In an optimal hypothesis it would be necessary to have 10^{15} neurons in a hidden layer of RBF-NN (it is approximately the same input order), and then the number of connections to be found would be

$$2 \times 10^6 \times 10^5 + 10^6 \times 10^5 = 3 \times 10^{11},$$

where 2×10^6 is the number of variables in grid point in input of the net, 10^5 is number of neurons, 10^6 in second part of sum is the number of variables in grid point in output layer of the net.

In actual version of the algorithm, the RBF-NN should have

$$6 \times 15 + 3 \times 15 = 135$$

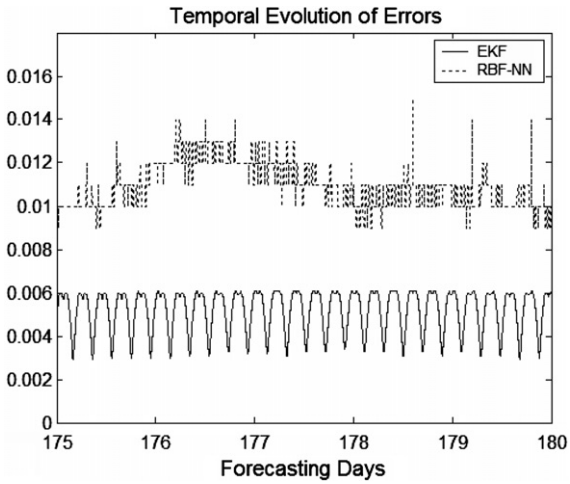

Fig. 8. Total error for EKF and RBF-NN.

connections, where 6 is the number of variables in grid point on the input of the net, 15 is the number of the neurons in the hidden layer and 3, in second part of sum, is the number of variables in grid point in the output layer. As shown, the number of connections decreased in the order of 10^9 in this research comparing with Nowosad et al. [9].

Fig. 8 shows the temporal evolutions of assimilation errors calculated by RBF-NN (dotted line) and by the EKF (continuous line). In this figure, the average assimilation error (AE_n – Eq. (41)) is considered: $\langle AE\rangle_n = (U + V + \Phi + \varsigma + \delta)/5$. The figure shows that both, RBF-NN and EKF, can carried out the data assimilation, but the EKF is more accurate.

6. Conclusions

In this work the ANN was tested as technique to solve a data assimilation problem in nonlinear dynamics. The RBF-NN was trained using cross validation scheme, so that a complete knowledge of the error surface is known, allowing the best weight sets be reached.

Neural networks are an attractive technique for data assimilation procedures, because after the training phase, the ANNs present a lower computational complexity than the Kalman filter and the variational approach. There are also other additional advantages in the ANN approach, such as the algorithm could be parallel, and a hardware implementation (neuro-computers) are possible too.

The algorithm implemented in this work, permits a decrease in the dimension problem. In the opinion of the authors, this is the best result obtained here. The RBF-NN was effective for data assimilation and the cross validation is a good strategy to choose the best weight set.

In order to improve the results obtained here, future works will test recurrent neural networks or feedback neural network (network of neurons with feedback connections). Other improvements for future work will involve the exploration of the observational spatial and temporal distributions of the noise data.

Acknowledgements

Authors want to thank to the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Brazilian agencies for research support. Thanks also to Dr. Tony Anderson for proof-reading this paper. The authors acknowledge for the anonymous referee, with suggestions help us to improve the paper.

References

[1] C. Grebogi, E. Ott, J. Yorke, Chaos, strange attractors, and fractal basin boundaries in nonlinear dynamics, Science 238 (1987) 585–718.
[2] E. Kalnay, Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, Cambridge, United Kingdom, 2004, pp. 341.
[3] F. Boutier, P. Courtier, Data assimilation concepts and methods, Lecture Note, European Centre for Medium-Range Weather Forecasts (ECMWF), <http://www.ecmwf.int/newsevents/training/lecture_notes/LN_DA.html>.
[4] D. Dee, A. Da Silva, Data Assimilation in the presence of forecast bias, Quart. J. Roy. Meteorol. Soc. 124 (1998) 269–295.
[5] R. Miller, M. Guil, F. Gauthiez, Advanced Data Assimilation in strongly nonlinear dynamical systems, J. Atmos. Sci. 51 (1994) 1037–1056.
[6] J. Anderson, An ensemble adjustment Kalman Filter for data assimilation, Mon. Weather Rev. 129 (2001) 2884–2903.
[7] P. Lister, J. Guo, T. Clune, W. Larson, The computational complexity and parallel scalability of Atmospheric Data Assimilation Algorithms, J. Atmos. Ocean. Technol. 21 (2004) 1689–1700.
[8] W. Hsieh, B. Tang, Applying Neural Network Models to Prediction and Data Analysis in Meteorology and Oceanography, Bull. Am. Meteorol. Soc. 79 (1998) 1855–1870.
[9] A. Nowosad, H. Campos Velho, A. Rios Neto, Data assimilation in chaotic dynamics using neural networks, in: Proceeding of the Third International Conference on Nonlinear Dynamics, Chaos, Control and Their Applications in Engineering Sciences, vol. 6, Associação Brasileira de Ciências Mecânicas, Rio de Janeiro, 2000, p. 212–221.
[10] H. Campos Velho, N. Vijaykumar, S. Stephany, A. Preto, A. Nowosad, A Neural Network Implementation for Data Assimilation using MPI, Applications of High-Performance Computing in Engineering, WIT Press, Southampton, United Kingdom, 2000, 280 p., Section 5, p. 211–220, ISBN 1-85312-924-0.

[11] A. Liaqat, M. Fukuhara, T. Takeda, Applying a neural collocation method to an incompletely known dynamical system via weak constraint data assimilation, Month. Weather Rev. 131 (2003) 1696–1714.
[12] B. Tang, W. Hsieh, Coupling neural network to dynamical systems via variational data assimilation, Month. Weather Rev. 131 (2001) 1855–1870.
[13] P. Lynch, X. Huang, Initialization of HIRLAM model using a digital filter, Month. Weather Rev. 14 (1992) 1019–1034.
[14] J. Holton, An Introduction to Dynamic Meteorology, Elsevier Academic Press, San Diego, USA, 1992, pp.511.
[15] H. Campos Velho, J. Claeyssen, A Comprehensive analysis of a barotrophic limited area model using the nonmodal matrix technique, Braz. J. Meteorol. 12 (1997) 41–50.
[16] A. Jazwinski, Stochastic processes and filtering theory, Academic Press, New York, USA, 1970.
[17] M.W. Gardner, S.R. Dorling, Artificial neural networks (the multilayer perpectron) – a review of applications in the atmospheric sciences, Atmos. Environ. 22 (1998) 2627–2636.
[18] F.P. Hartër, H.F. Campos Velho, Recurrent and Feedforward Neural Networks Trained with Cross Correlation Applied to the Data Assimilation in Chaotic Dynamic, Braz. J. Meteorol. 20 (2005) 411–420.
[19] S. Haykin, Neural Networks: A comprehensive foundation, McMillan College Publishing Company, New York, NY, USA, 1994.
[20] D.S. Broomhead, D. Lowe, Multivariate functional interpolation and adaptive networks, Complex Syst. 2 (1988) 321–355.
[21] C.A. Micchelli, Interpolation of scattered data: Distance matrices and conditionally positive definite functions, Construct. Approx. 2 (1986) 11–22.
[22] T. Poggio, F. Girosi, Networks for approximation and learning, Proc. IEEE 78 (1990) 1481–1497.
[23] A.N. Tikhonov, V.Y. Arsenin, Solutions of Ill-posed Problems, Winston & Sons, 1977, pp. 349.