Neurocomputing 71 (2008) 3335–3343

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Long-term time series prediction with the NARX network: An empirical evaluation

José Maria P. Menezes Jr., Guilherme A. Barreto *

*Department of Teleinformatics Engineering, Center of Technology, Campus of Pici, Federal University of Ceará, Av. Mister Hull, S/N, CP 6005, CEP 60455-760, Fortaleza, CE, Brazil*

ARTICLE INFO

Available online 15 July 2008

*Keywords:*
NARX neural network
Long-term prediction
Nonlinear traffic modeling
Chaotic time series
Recurrence plot

ABSTRACT

The NARX network is a dynamical neural architecture commonly used for input–output modeling of nonlinear dynamical systems. When applied to time series prediction, the NARX network is designed as a feedforward time delay neural network (TDNN), i.e., without the feedback loop of delayed outputs, reducing substantially its predictive performance. In this paper, we show that the original architecture of the NARX network can be easily and efficiently applied to long-term (multi-step-ahead) prediction of univariate time series. We evaluate the proposed approach using two real-world data sets, namely the well-known chaotic laser time series and a variable bit rate (VBR) video traffic time series. All the results show that the proposed approach consistently outperforms standard neural network based predictors, such as the TDNN and Elman architectures.

## 1. Introduction

Artificial neural networks (ANNs) have been successfully applied to a number of time series prediction and modeling tasks, including financial time series prediction [12], river flow forecasting [3], biomedical time series modeling [11], communication network traffic prediction [2,6,13], chaotic time series prediction [42], among several others (see [34], for a recent survey). In particular, when the time series is noisy and the underlying dynamical system is nonlinear, ANN models frequently outperform standard linear techniques, such as the well-known Box–Jenkins models [7]. In such cases, the inherent nonlinearity of ANN models and a higher robustness to noise seem to explain their better prediction performance.

In one-step-ahead prediction tasks, ANN models are required to estimate the next sample value of a time series, without feeding back it to the model's input regressor. In other words, the input regressor contains only actual sample points of the time series. If the user is interested in a longer prediction horizon, a procedure known as multi-step-ahead or long-term prediction, the model's output should be fed back to the input regressor for a fixed but finite number of time steps [39]. In this case, the components of the input regressor, previously composed of actual sample points of the time series, are gradually replaced by previously predicted values.

If the prediction horizon tends to infinity, from some time in the future the input regressor will start to be composed only of estimated values of the time series. In this case, the multi-step-ahead prediction task becomes a *dynamic modeling* task, in which the ANN model acts as an autonomous system, trying to recursively emulate the dynamic behavior of the system that generated the nonlinear time series [17,18]. Multi-step-ahead prediction and dynamic modeling are much more complex to deal with than one-step-ahead prediction, and it is believed that these are complex tasks in which ANN models play an important role, in particular recurrent neural architectures [36].

Simple recurrent networks (SRNs) comprise a class of recurrent neural models that are essentially feedforward in the signal-flow structure, but also contain a small number of local and/or global feedback loops in their architectures. Even though feedforward multilayer perceptron (MLP)-like networks can be easily adapted to process time series through an input tapped-delay line, giving rise to the well-known time delay neural network (TDNN) [36], they can also be easily converted to SRNs by feeding back the neuronal outputs of the hidden or output layers, giving rise to Elman and Jordan networks, respectively [23]. It is worth pointing out that, when applied to long-term prediction, a feedforward TDNN model will eventually behave as a kind of SRN architecture, since a global loop is needed to feed back the current estimated value into the input regressor.

The aforementioned recurrent architectures are usually trained by means of temporal gradient-based variants of the backpropagation algorithm [35]. However, learning to perform tasks in which the temporal dependencies present in the input–output

* Corresponding author. Tel./fax: +55 85 3366 9467.
*E-mail addresses:* josemenezesjr@gmail.com (J.M.P. Menezes Jr.), guilherme@deti.ufc.br (G.A. Barreto).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.01.030

signals span long time intervals can be quite difficult using gradient-based learning algorithms [4]. In [27], the authors report that learning such long-term temporal dependencies with gradient-descent techniques is more effective in a class of SRN model called *Nonlinear Autoregressive with eXogenous input* (NARX) [28] than in simple MLP-based recurrent models. This occurs in part because the NARX model's input vector is cleverly built through two tapped-delay lines: one sliding over the input signal together and the other sliding over the network's output.

Despite the aforementioned advantages of the NARX network, its feasibility as a nonlinear tool for univariate time series modeling and prediction has not been fully explored yet. For example, in [29], the NARX model is indeed reduced to the TDNN model in order to be applied to time series prediction. Bearing this under-utilization of the NARX network in mind, we propose a simple strategy based on Taken's embedding theorem that allows the original architecture of the NARX network to be easily and efficiently applied to long-term prediction of univariate nonlinear time series.

Potential fields of application of our approach are communication network traffic characterization [14,16,45] and chaotic time series prediction [22], since it has been shown that these kinds of data present long-range dependence (LRD) due to their self-similar nature. Thus, for the sake of illustration, we evaluate the proposed approach using two real-world data sets obtained from these domains, namely the well known chaotic laser time series and a variable bit rate (VBR) video traffic time series.

The remainder of the paper is organized as follows. In Section 2, we describe the NARX network model and its main characteristics. In Section 3 we introduce the basics of the nonlinear time series prediction problem and present our approach. The simulations and discussion of results are presented in Section 4. The paper is concluded in Section 5.

## 2. The NARX network

NARX [26,30,33] is an important class of discrete-time nonlinear systems that can be mathematically represented as

$$y(n+1) = f[y(n), \ldots, y(n-d_y+1); u(n-k), u(n-k+1), \ldots, u(n-d_u-k+1)], \quad (1)$$

where $u(n) \in \mathbb{R}$ and $y(n) \in \mathbb{R}$ denote, respectively, the input and output of the model at discrete time step $n$, while $d_u \geqslant 1$ and $d_y \geqslant 1$, $d_u \leqslant d_y$, are the input-memory and output-memory orders, respectively. The parameter $k$ ($k \geqslant 0$) is a delay term, known as the process dead-time.

Without lack of generality, we always assume $k = 0$ in this paper, thus obtaining the following NARX model:

$$y(n+1) = f[y(n), \ldots, y(n-d_y+1); u(n), u(n-1), \ldots, u(n-d_u+1)], \quad (2)$$

which may be written in vector form as

$$y(n+1) = f[\mathbf{y}(n); \mathbf{u}(n)], \quad (3)$$

where the vectors $\mathbf{y}(n)$ and $\mathbf{u}(n)$ denote the output and input regressors, respectively.

The nonlinear mapping $f(\cdot)$ is generally unknown and can be approximated, for example, by a standard MLP network. The resulting connectionist architecture is then called a *NARX network* [10,32], a powerful class of dynamical models which has been shown to be computationally equivalent to Turing machines [38]. Fig. 1 shows the topology of a two-hidden-layer NARX network.

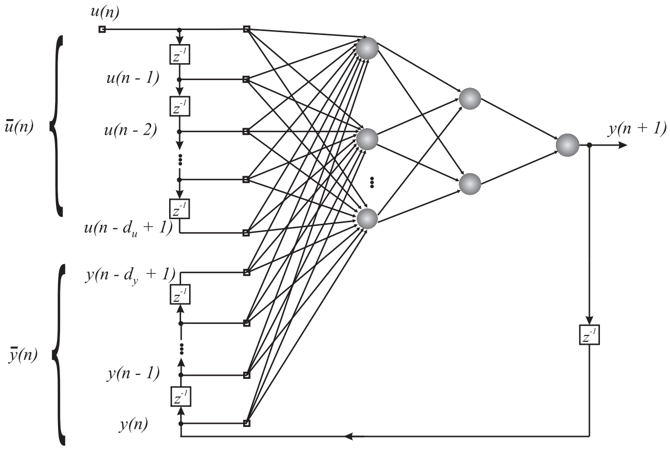

**Fig. 1.** NARX network with $d_u$ delayed inputs and $d_y$ delayed outputs ($z^{-1}$ = unit time delay).

In what concern training the NARX network, it can be carried out in one out of two modes:

- *Series-parallel* (*SP*) *mode*. In this case, the output's regressor is formed only by actual values of the system's output:

$$\hat{y}(n+1) = \hat{f}[\mathbf{y}_{\text{sp}}(n); \mathbf{u}(n)], \\ = \hat{f}[y(n), \ldots, y(n-d_y+1); u(n), u(n-1), \ldots, u(n-d_u+1)], \quad (4)$$

where the hat symbol ($\wedge$) is used to denote estimated values (or functions).
- *Parallel* (*P*) *mode*. In this case, estimated outputs are fed back and included in the output's regressor[1]:

$$\hat{y}(n+1) = \hat{f}[\mathbf{y}_{\text{p}}(n); \mathbf{u}(n)], \\ = \hat{f}[\hat{y}(n), \ldots, \hat{y}(n-d_y+1); u(n), u(n-1), \ldots, u(n-d_u+1)]. \quad (5)$$

As a tool for nonlinear system identification, the NARX network has been successfully applied to a number of real-world input–output modeling problems, such as heat exchangers, waste water treatment plants, catalytic reforming systems in a petroleum refinery and nonlinear time series prediction (see [29] and references therein).

As mentioned in the Introduction, the particular topic of this paper is the issue of nonlinear univariate time series prediction with the NARX network. In this type of application, the output-memory order is usually set $d_y = 0$, thus reducing the NARX network to the TDNN architecture [29], i.e.,

$$y(n+1) = f[\mathbf{u}(n)], \\ = f[u(n), u(n-1), \ldots, u(n-d_u+1)], \quad (6)$$

where $\mathbf{u}(n) \in \mathbb{R}^{d_u}$ is the input regressor. This simplified formulation of the NARX network eliminates a considerable portion of its representational capabilities as a dynamic network; that is, all the dynamic information that could be learned from the past memories of the output (feedback) path is discarded.

For many practical applications, however, such as self-similar traffic modeling [16], the network must be able to robustly store information for a long period of time in the presence of noise. In gradient-based training algorithms, the fraction of the gradient due to information $n$ time steps in the past approaches zero as $n$ becomes large. This effect is called the *problem of vanishing*

[1] The NARX model in P mode is also known as *output-error model* [30].

*gradient* and has been pointed out as the main cause for the poor performance of standard dynamical ANN models when dealing with LRDs.

The original formulation of the NARX network does not circumvent the problem of vanishing gradient, but it has been demonstrated that it often performs much better than standard dynamical ANNs in such a class of problems, achieving much faster convergence and better generalization performance [28]. As pointed out in [27], an intuitive explanation for this improvement in performance is that the output memories of a NARX neural network are represented as jump-ahead connections in the time-unfolded network that is often encountered in learning algorithms such as the backpropagation through time (BPTT). Such jump-ahead connections provide shorter paths for propagating gradient information, reducing the sensitivity of the network to long-term dependencies.

Hence, if the output memory is discarded, as shown in Eq. (6), performance improvement may no longer be observed. Bearing this in mind as a motivation, we propose a simple strategy to allow the computational resources of the NARX network to be fully explored in nonlinear time series prediction tasks.

## 3. Nonlinear time series prediction with NARX network

In this section we provide a short introduction of the theory of embedding and state-space reconstruction. Interested readers are referred to [1] for further details.

The state of a deterministic dynamical system is the information necessary to determine the evolution of the system in time. In discrete time, this evolution can be described by the following system of difference equations:

$$\mathbf{x}(n+1) = \mathbf{F}[\mathbf{x}(n)], \tag{7}$$

where $\mathbf{x}(n) \in \mathbb{R}^d$ is the state of the system at time step $n$, and $\mathbf{F}[\cdot]$ is a nonlinear vector valued function. A time series is a time-ordered set of measures $\{x(n)\}, n = 1, \ldots, N$, of a scalar quantity observed at the output of the system. This observable quantity is defined in terms of the state $\mathbf{x}(n)$ of the underlying system as follows:

$$x(n) = h[\mathbf{x}(n)] + \varepsilon(t), \tag{8}$$

where $h(\cdot)$ is a nonlinear scalar-valued function, $\varepsilon$ is a random variable which accounts for modeling uncertainties and/or measurement noise. It is commonly assumed that $\varepsilon(t)$ is drawn from a Gaussian white noise process. It can be inferred immediately from Eq. (8) that the observations $\{x(n)\}$ can be seen as a projection of the multivariate state space of the system onto the one-dimensional space. Eqs. (7) and (8) describe together the state-space behavior of the dynamical system.

In order to perform prediction, one needs to reconstruct (estimate) as well as possible the state space of the system using the information provided by $\{x(n)\}$ only. In [40], Takens has shown that, under very general conditions, the state of a deterministic dynamic system can be accurately reconstructed by a time window of finite length sliding over the observed time series as follows:

$$\mathbf{x}_1(n) \triangleq [x(n), x(n-\tau), \ldots, x(n-(d_\mathrm{E}-1)\tau)], \tag{9}$$

where $x(n)$ is the sample value of the time series at time $n$, $d_\mathrm{E}$ is the embedding dimension and $\tau$ is the embedding delay. Eq. (9) implements the delay embedding theorem [22]. According to this theorem, a collection of time-lagged values in a $d_\mathrm{E}$-dimensional vector space should provide sufficient information to reconstruct the states of an observable dynamical system. By doing this, we are indeed trying to unfold the projection back to a multivariate state space whose topological properties are equivalent to those of the state space that actually generated the observable time series, provided the embedding dimension $d_\mathrm{E}$ is large enough.

The embedding theorem also provides a theoretical framework for nonlinear time series prediction, where the predictive relationship between the current state $\mathbf{x}_1(t)$ and the next value of the time series is given by the following equation:

$$x(n+1) = g[\mathbf{x}_1(n)]. \tag{10}$$

Once the embedding dimension $d_\mathrm{E}$ and delay $\tau$ are chosen, one remaining task is to approximate the mapping function $g(\cdot)$. It has been shown that a feedforward neural network with enough neurons is capable of approximating any nonlinear function to an arbitrary degree of accuracy. Thus, it can provide a good approximation to the function $g(\cdot)$ by implementing the following mapping:

$$\hat{x}(n+1) = \hat{g}[\mathbf{x}_1(n)], \tag{11}$$

where $\hat{x}(n+1)$ is an estimate of $x(n+1)$ and $\hat{g}(\cdot)$ is the corresponding approximation of $g(\cdot)$. The estimation error, $e(n+1) = x(n+1) - \hat{x}(n+1)$, is commonly used to evaluate the quality of the approximation.

If we set $\mathbf{u}(n) = \mathbf{x}_1(n)$ and $y(n+1) = x(n+1)$ in Eq. (6), then it leads to an intuitive interpretation of the nonlinear state-space reconstruction procedure as equivalent to the time series prediction problem whose goal is to compute an estimate of $x(n+1)$. Thus, the only thing we have to do is to train a TDNN model [36]. Once training is completed, the TDNN can be used for predicting the next samples of the time series.

Despite the correctness of the TDNN approach, recall that it is derived from a simplified version of the NARX network by eliminating the output memory. In order to use the full computational abilities of the NARX network for nonlinear time series prediction, we propose novel definitions for its input and output regressors. Firstly, the input signal regressor, denoted by $\mathbf{u}(n)$, is defined by the delay embedding coordinates of Eq. (9):

$$\mathbf{u}(n) = \mathbf{x}_1(n) = [x(n), x(n-\tau), \ldots, x(n-(d_\mathrm{E}-1)\tau)], \tag{12}$$

where we set $d_u = d_\mathrm{E}$. In words, the input signal regressor $\mathbf{u}(n)$ is composed of $d_\mathrm{E}$ actual values of the observed time series, separated from each other of $\tau$ time steps.

Secondly, since the NARX network can be trained in two different modes, the output signal regressor $\mathbf{y}(n)$ can be written accordingly as

$$\mathbf{y}_\mathrm{sp}(n) = [x(n), \ldots, x(n-d_y+1)] \tag{13}$$

or

$$\mathbf{y}_\mathrm{p}(n) = [\hat{x}(n), \ldots, \hat{x}(n-d_y+1)]. \tag{14}$$

Note that the output regressor for the SP mode shown in Eq. (13) contains $d_y$ past values of the actual time series, while the output regressor for the P mode shown in Eq. (14) contains $d_y$ past values of the estimated time series. For a suitably trained network, no matter under which training mode, these outputs are estimates of previous values of $x(n+1)$. Henceforth, NARX networks trained using the regression pairs $\{\mathbf{y}_\mathrm{sp}(n), \mathbf{x}_1(n)\}$ and $\{\mathbf{y}_\mathrm{p}(n), \mathbf{x}_1(n)\}$ are denoted by NARX-SP and NARX-P networks, respectively. These NARX networks implement following predictive mappings, which can be visualized in Figs. 2 and 3:

$$\hat{x}(n+1) = \hat{f}[\mathbf{y}_\mathrm{sp}(n), \mathbf{u}(n)] = \hat{f}[\mathbf{y}_\mathrm{sp}(n), \mathbf{x}_1(n)], \tag{15}$$

$$\hat{x}(n+1) = \hat{f}[\mathbf{y}_\mathrm{p}(n), \mathbf{u}(n)] = \hat{f}[\mathbf{y}_\mathrm{p}(n), \mathbf{x}_1(n)], \tag{16}$$

where the nonlinear function $\hat{f}(\cdot)$ is readily implemented through an MLP trained with plain backpropagation algorithm.

It is worth noting that Figs. 2 and 3 correspond to the different ways the NARX network can be trained; i.e., in SP mode or in

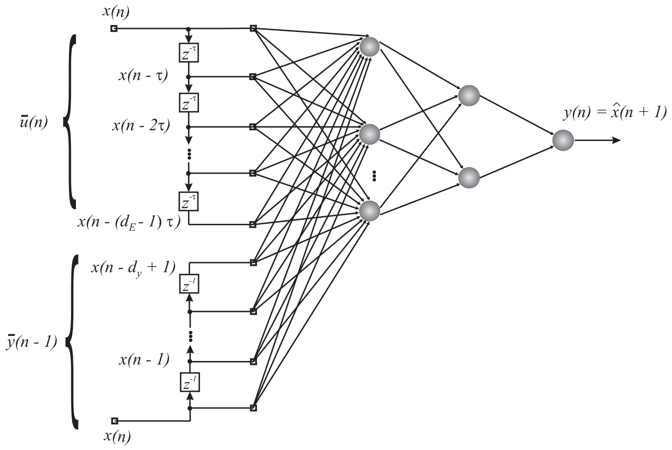

**Fig. 2.** Architecture of the NARX network during training in the SP mode ($z^{-\tau} = \tau$ unit time delays).

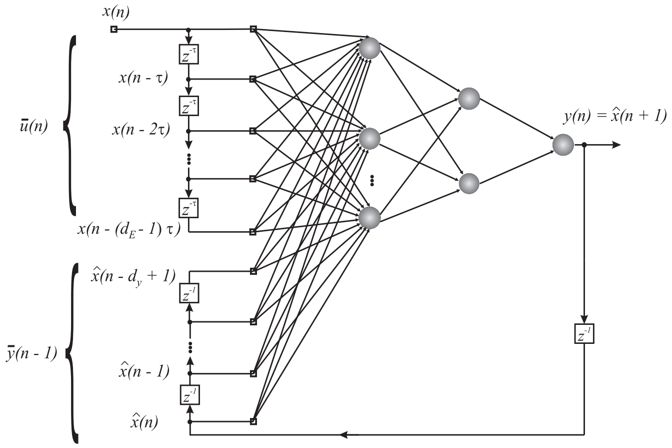

**Fig. 3.** Architecture of the NARX network during training in the P mode ($z^{-\tau} = \tau$ unit time delays).

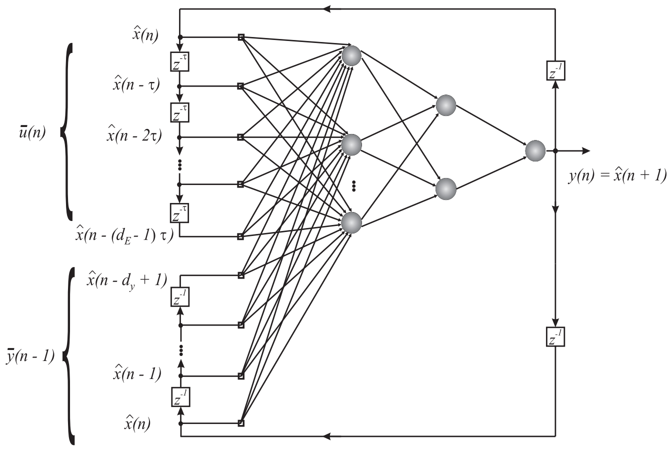

**Fig. 4.** Common architecture for the NARX-P and NARX-SP networks during the testing (recursive prediction) phase.

P mode, respectively. During the testing phase, however, since long-term predictions are required, the predicted values should be fed back to both, the input regressor $\mathbf{u}(n)$ and the output regressor $\mathbf{y}_{\mathrm{sp}}(n)$ (or $\mathbf{y}_{\mathrm{p}}(n)$), simultaneously. Thus, the resulting predictive model has two feedback loops, one for the input regressor and the other for the output regressor, as illustrated in Fig. 4.

Thus, unlike the TDNN-based approach for the nonlinear time series prediction problem, the proposed approach makes full use of the output feedback loop. Eqs. (12) and (13) are valid only for one-step-ahead prediction tasks. Again, if one is interested in multi-step-ahead or recursive prediction tasks, the estimates $\hat{x}$ should also be inserted into both regressors in a recursive fashion.

One may argue that, in addition to the parameters $d_{\mathrm{E}}$ and $\tau$, the proposed approach introduces one more to be determined, namely, $d_y$. However, this parameter can be eliminated if we recall that, as pointed out in [18], the delay embedding of Eq. (9) has an alternative form given by

$$\mathbf{x}_2(n) \triangleq [x(n), x(n-1), \ldots, x(n-m+1)], \tag{17}$$

where $m$ is an integer defined as $m \geqslant \tau \cdot d_{\mathrm{E}}$. By comparing Eqs. (13) and (17), we find that a suitable choice is given by $d_y \geqslant \tau \cdot d_{\mathrm{E}}$, which also satisfies the necessary condition $d_y > d_u$. However, we have found by experimentation that a value chosen from the interval $d_{\mathrm{E}} < d_y \leqslant \tau \cdot d_{\mathrm{E}}$ is sufficient for achieving a predictive performance better than those achieved by conventional neural based time series predictors, such as the TDNN and Elman architectures.

Finally, the proposed approach is summarized as follows. A NARX network is defined so that its input regressor $\mathbf{u}(n)$ contains samples of the measured variable $x(n)$ separated $\tau$ ($\tau > 0$) time steps from each other, while the output regressor $\mathbf{y}(n)$ contains actual or estimated values of the same variable, but sampled at consecutive time steps. As training proceeds, these estimates should become more and more similar to the actual values of the time series, indicating convergence of the training process. Thus, it is interesting to note that the input regressor supplies medium- to long-term information about the dynamical behavior of the time series, since the delay $\tau$ is usually larger than unity, while the output regressor, once the network has converged, supplies short-term information about the same time series.

## 4. Simulations and discussion

In this paper, our aim is to evaluate, in qualitative and quantitative terms, the predictive ability of the NARX-P and NARX-SP networks using two real-world data sets, namely the chaotic laser and the VBR video traffic time series. For the sake of completeness, a performance comparison with the TDNN and Elman recurrent networks is also carried out.

It is worth emphasizing that our goal in the experiments is to evaluate if the output regressor $\mathbf{y}_{\mathrm{sp}}$ (or $\mathbf{y}_{\mathrm{p}}$) in the input layer of the NARX network improves its predictive performance. Thus, to facilitate the performance comparison, all the networks we simulate have two hidden layers and one output neuron.

All neurons in both hidden layers and the output neuron use hyperbolic tangent activation functions. The standard backpropagation algorithm is used to train the networks with learning rate equal to 0.001 (selected heuristically). No momentum term is used. In what concerns the Elman network, only the neuronal outputs of the first hidden layer are fed back to the input layer.

The number of neurons, $N_{\mathrm{h},1}$ and $N_{\mathrm{h},2}$, in the first and second hidden layers, respectively, are equal for all simulated networks. These values are chosen according to the following heuristic rules [31]:

$$N_{\mathrm{h},1} = 2d_{\mathrm{E}} + 1 \quad \text{and} \quad N_{\mathrm{h},2} = \sqrt{N_{\mathrm{h},1}}, \tag{18}$$

where $N_{\mathrm{h},2}$ is rounded up towards the next integer number. The first rule is motivated by Kolmogorov's theorem on function approximation [19]. The second rule simply states that the number of neurons in the second hidden layer is the square root

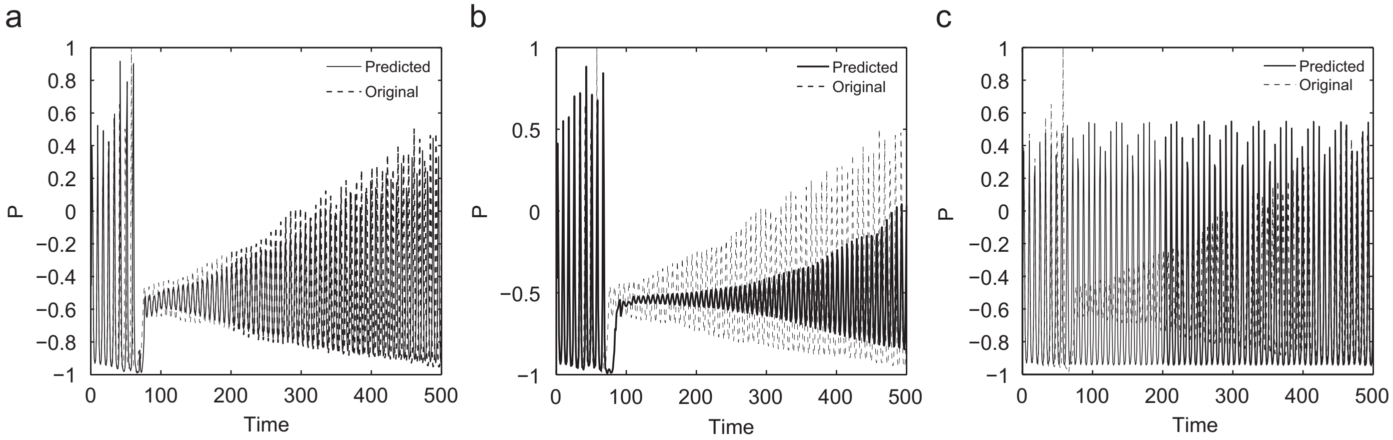

**Fig. 5.** Results for the laser series: (a) NARX-SP, (b) Elman, and (c) TDNN.

of the product of the dimension of the first hidden layer and the dimension of the output layer. Finally, we set $d_y = 2\tau d_E$, where $\tau$ is selected as the value occurring at the first minimum of the mutual information function of the time series [15].

The total number $M$ of adjustable parameters (weights and thresholds) for each of the simulated networks are given by

$$\begin{aligned}
M &= (d_E + 1) \cdot N_{h,1} + (N_{h,1} + 2) \cdot N_{h,2} + 1 \quad \text{(TDNN)},\\
M &= (N_{h,1} + d_E + 1) \cdot N_{h,1} + (N_{h,1} + 2) \cdot N_{h,2} + 1 \quad \text{(ELMAN)},\\
M &= (d_E + d_y + 1) \cdot N_{h,1} + (N_{h,1} + 2) \cdot N_{h,2} + 1 \quad \text{(NARX)}. \qquad (19)
\end{aligned}$$

Once a given network has been trained, it is required to provide estimates of the future sample values of a given time series for a certain prediction horizon $N$. The predictions are executed in a recursive fashion until desired prediction horizon is reached, i.e., during $N$ time steps the predicted values are fed back in order to take part in the composition of the regressors. The networks are evaluated in terms of the *normalized mean squared error* (NMSE),

$$NMSE(N) = \frac{1}{N \cdot \sigma_x^2} \sum_{n=1}^{N} (x(n+1) - \hat{x}(n+1))^2, \qquad (20)$$

where $x(n+1)$ is the actual value of the time series, $\hat{x}(n+1)$ is the predicted value, $N$ is the horizon prediction (i.e., how many steps into the future a given network has to predict), and $\hat{\sigma}_x^2$ is the sample variance of the actual time series. The NMSE values are averaged over 10 training/testing runs.

*Chaotic laser time series.* The first data sequence to be used to evaluate the NARX-P and NARX-SP models is the chaotic laser time series [42]. This time series comprises measurements of the intensity pulsations of a single-mode Far–Infrared–Laser $NH_3$ in a chaotic state [21]. It was made available worldwide during a time series prediction competition organized by the Santa Fe Institute and, since then, has been used in benchmarking studies.

The laser time series has 1500 points which have been rescaled to the range $[-1, 1]$. The rescaled time series was further split into two sets for the purpose of performing one-fold cross-validation, so that the first 1000 samples were used for training and the remaining 500 samples for testing. The embedding dimension was estimated as $d_E = 7$ by applying Cao's method [8], which is a variant of the well-known false nearest neighbors method.[2] The embedding delay was estimated as $\tau = 2$. For the chosen parameters, the total number of modifiable weights and biases for the three simulated neural architectures are the following: $M = 189$ (TDNN), $M = 414$ (Elman) and $M = 609$ (NARX).

The results are shown in Figs. 5(a)–(c), for the NARX-SP, Elman and TDNN networks, respectively.[3] A visual inspection illustrates clearly that the NARX-SP model performed better than the other two architectures. It is important to point out that a critical situation occursaround time step 60, where the laser intensity collapses suddenly from its highest value to its lowest one; then, it starts recovering the intensity gradually. The NARX-SP model is able to emulate the laser dynamics very closely. The Elman's network was doing well until the critical point. From this point onwards, it was unable to emulate the laser dynamics faithfully, i.e., the predicted laser intensities have much lower amplitudes than the actual ones. The TDNN network had a very poor predictive performance. From a dynamical point of view the output of the TDNN seems to be stuck in a limit cycle, since it only oscillates endlessly.

It is worth mentioning that the previous results did not mean that the TDNN and Elman networks cannot learn the dynamics of the chaotic laser. Indeed, it was shown to be possible in [18] using sophisticated training algorithms, such as BPTT [43] or real-time recurrent learning (RTRL) [44]. In what concern the TDNN network, our results confirms the observations reported by Eric Wan [41, p. 62] in his Ph.D. Thesis. There, he states that the standard MLP, using the input regressor $\mathbf{x}_1(t)$ only and trained with the instantaneous gradient descent rule, has been unable to accurately predict the laser time series. In his own words, "the downward intensity collapse went completely undetected", as in our case.

In sum, our results show that under the same conditions, i.e., with the same number of hidden neurons, using the standard gradient-based backpropagation algorithm, a short time series for training, and the same number of training epochs, the NARX-SP network performs better than the TDNN and Elman networks. It seems that the presence of the output regressor $\mathbf{y}_{sp}$ improves indeed the predictive power of the NARX network.

For the sake of comparison, under similar training and network evaluation methodologies, the FIR-MLP model proposed by Eric Wan [41] achieved very good long-term prediction results on the laser time series, which are equivalent to those obtained by the NARX-SP network. However, the FIR-MLP required $M = 1105$ adjustable parameters to achieve such a good performance, while the NARX-SP model required roughly half the number of parameters (i.e., $M = 609$).

The long-term predictive performances of all simulated networks can be assessed in more quantitative terms by means of

[2] A recent technique for the estimation of $d_E$ can be found in [25].

[3] The results for the NARX-P network are not shown since they are equivalent to those shown for the NARX-SP network.

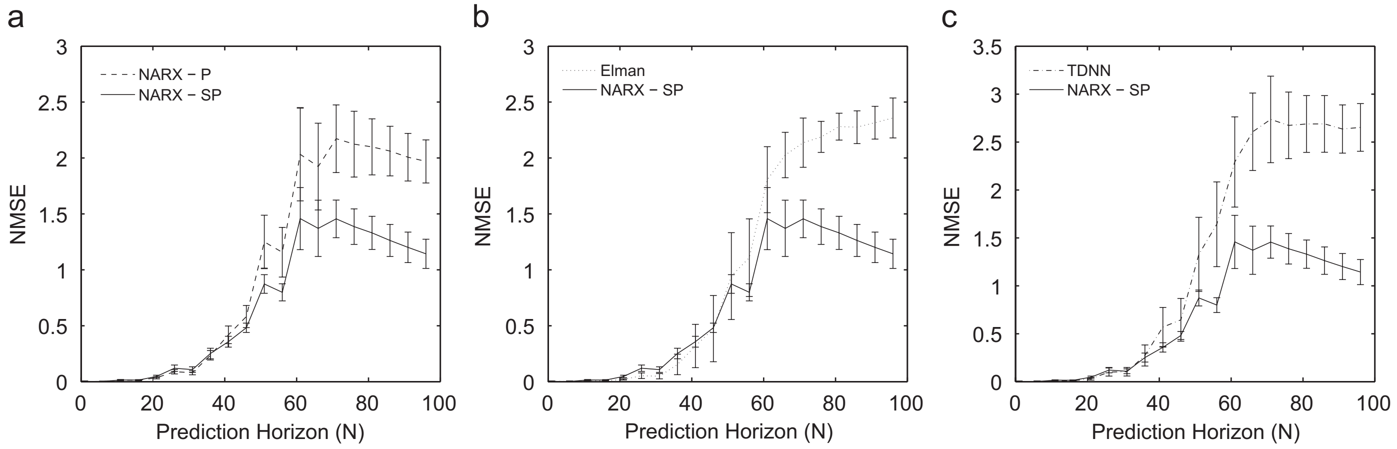

**Fig. 6.** Multi-step-ahead NMSE values and the corresponding confidence intervals: (a) NARX-P × NARX-SP models, (b) Elman × NARX-SP models, and (c) TDNN × NARX-SP models.

NMSE curves as a function of the prediction horizon ($N$). Figs. 6(a)–(c) show the NMSE curve of the NARX-SP model (best performance) in comparison to the curves obtained by the NARX-P, Elman and TDNN models, respectively. It is worth emphasizing two types of behavior in these figures. Below the critical time step (i.e., $N<60$), the reported NMSE values are approximately the same for all models, with a small advantage to the Elman network. This means that, while the critical point is not reached, all networks predict well the time series. For $N>60$, the NARX-P and NARX-SP models reveal their superior performances. For all figures, the confidence intervals of the NSME values, for a significance level of $\alpha = 0.05$, are also provided. These intervals were generated for a sample size of $n = 10$, assuming a Gaussian distribution for the NSME values. Thus, the critical value $t_{\alpha/2,n-1} = 2.093$ used for building the corresponding 95% confidence intervals are obtained from a $t$-student distribution with $n-1$ degrees of freedom.

A useful way to qualitatively evaluate the performance of the NARX-SP network for the laser series is through recurrence plots [9]. These diagrams describe how a reconstructed state-space trajectory recurs or repeats itself, being useful for characterization of a system as random, periodic or chaotic. For example, random signals tend to occupy the whole area of the plot, indicating that no value tends to repeat itself. Any structured information embedded in a periodic or chaotic signal is reflected in a certain visual pattern in the recurrence plot.

Recurrence plots are built by calculating the distance between two points in the state space at times $i$ (horizontal axis) and $j$ (vertical axis):

$$\delta_{ij} = \|\mathbf{D}(i) - \mathbf{D}(j)\|, \tag{21}$$

where $\|\cdot\|$ is the Euclidean norm. The state vectors $\mathbf{D}(n) = [\hat{x}(n), \hat{x}(n-\tau), \ldots, \hat{x}(n-(d_E-1)\tau)]$ are built using the points of the predicted time series. Then, a dot is placed at the coordinate $(i,j)$ if $\delta_{ij}<r$. In this paper, we set $r = 0.4$ and the prediction horizon to $N = 200$.

The results are shown in Fig. 7. It can be easily visualized that the recurrence plots shown in Figs. 7(a) and (b) are more similar with one another, indicating that NARX-SP network reproduced the original state-space trajectory more faithfully.

*VBR video traffic time series*. Due to the widespread use of Internet and other packet/cell switching broad-band networks, VBR video traffic will certainly be a major part of the traffic produced by multimedia sources. Hence, many researches have focused on VBR video traffic prediction to devise network management strategies that satisfy QoS requirements. From the point of view of modeling, a particular challenging issue on network traffic prediction comes from the important discovery of self-similarity and LRD in broad-band network traffic [24]. Researchers have also observed that VBR video traffic typically exhibits burstiness over multiple time scales (see [5,20], for example).

In this section, we evaluate the predictive abilities of the NARX-P and NARX-SP networks using VBR video traffic time series (trace), extracted from Jurassic Park, as described in [37]. This video traffic trace was encoded at University of Würzburg with MPEG-I. The frame rates of video sequence that coded Jurassic Park have been used. The MPEG algorithm uses three different types of frames: intraframe (I), predictive (P) and bidirectionally-predictive (B). These three types of frames are organized as a group (group of picture, GoP) defined by the distance $L$ between I frames and the distance $M$ between P frames. If the cyclic frame pattern is {IBBPBBPBBPBBI}, then $L = 12$ and $M = 3$. These values for $L$ and $M$ are used in this paper.

The resulting time series has 2000 points which have been rescaled to the range $[-1, 1]$. The rescaled time series was further split into two sets for cross-validation purposes: 1500 samples for training and 500 samples for testing.

Evaluation of the long-term predictive performances of all networks can also help assessing the sensitivity of the neural models to important training parameters, such as the number of training epochs and the size of the embedding dimension, as shown in Fig. 8.

Fig. 8(a) shows the NMSE curves for all neural networks versus the value of the embedding dimension, $d_E$, which varies from 3 to 24. For this simulation we trained all the networks for 300 epochs, $\tau = 1$ and $d_y = 24$. One can easily note that the NARX-P and NARX-SP performed better than the TDNN and Elman networks. In particular, the performance of the NARX-SP was rather impressive, in the sense that it remains constant throughout the studied range. From $d_E \geqslant 12$ onwards, the performances of the NARX-P and NARX-SP are practically the same. It is worth noting that the performances of the TDNN and Elman networks approaches those of the NARX-P and NARX-SP networks when $d_E$ is of the same order of magnitude of $d_y$. This suggests that, for NARX-SP (or NARX-P) networks, we can select a small value for $d_E$ and still have a very good performance.

Fig. 8(b) shows the NMSE curves obtained from the simulated neural networks versus the number of training epochs, ranging from 90 to 600. For this simulation we trained all the networks

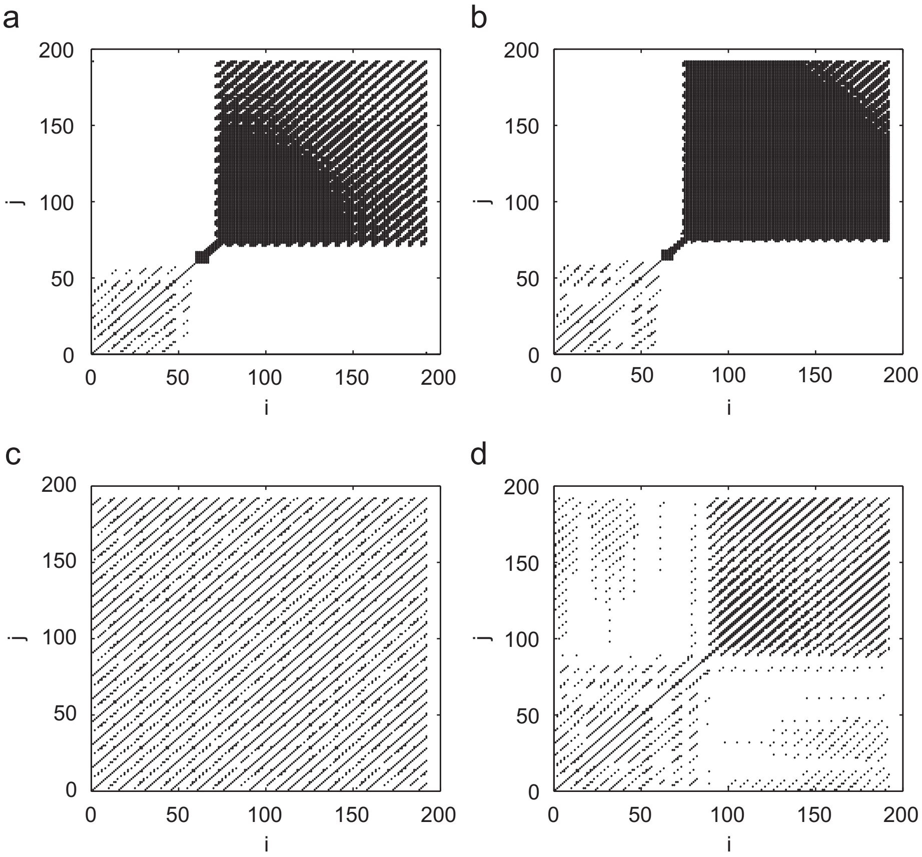

**Fig. 7.** Recurrence plot of (a) the original laser time series and the ones produced by (b) NARX-SP, (c) TDNN, and (d) Elman networks.

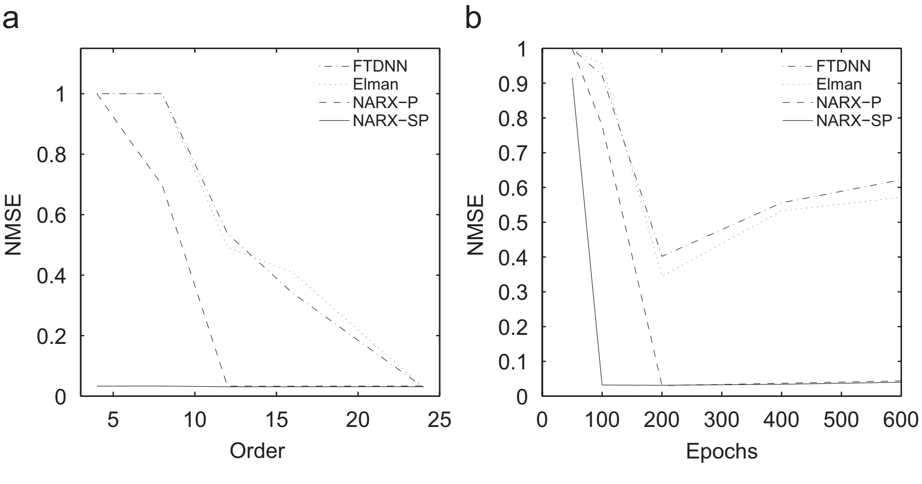

**Fig. 8.** Evaluation of the sensitivity of the neural networks with respect to (a) the embedding dimension and (b) the number of training epochs.

with $\tau = 1, d_E = 12$ and $d_y = 2\tau d_E = 24$. Again, better performances were achieved by the NARX-P and NARX-SP. The performance of the NARX-SP is practically the same from 100 epochs on. The same behavior is observed for the NARX-P network from 200 epochs on. This can be explained by recalling that the NARX-P uses estimated values to compose the output regressor $\mathbf{y}_p(n)$ and, because of that, it learns slower than the NARX-SP network.

Another important behavior can be observed for the TDNN and Elman networks. From 200 epochs onwards, these networks increase their NMSE values instead of decreasing them. We hypothesize that this behavior can be an evidence of overfitting, a phenomenon observed when powerful nonlinear models, with excessive degrees of freedom (too much adjustable parameters), are trained for a long period with a finite size data set. In this sense, the results of Fig. 8(b) strongly suggest that the NARX-SP and NARX-P networks are much more robust than the TDNN and Elman networks. In other words, the presence of an output regressor in the NARX-SP and NARX-P networks seems to turn them less prone to overfitting than the Elman and TDNN models, even when the number of free parameters in the NARX networks are higher than that in the Elman and TDNN models.

Finally, we show in Figs. 9(a)–(c), typically estimated VBR video traffic traces generated by the TDNN, Elman and NARX-SP

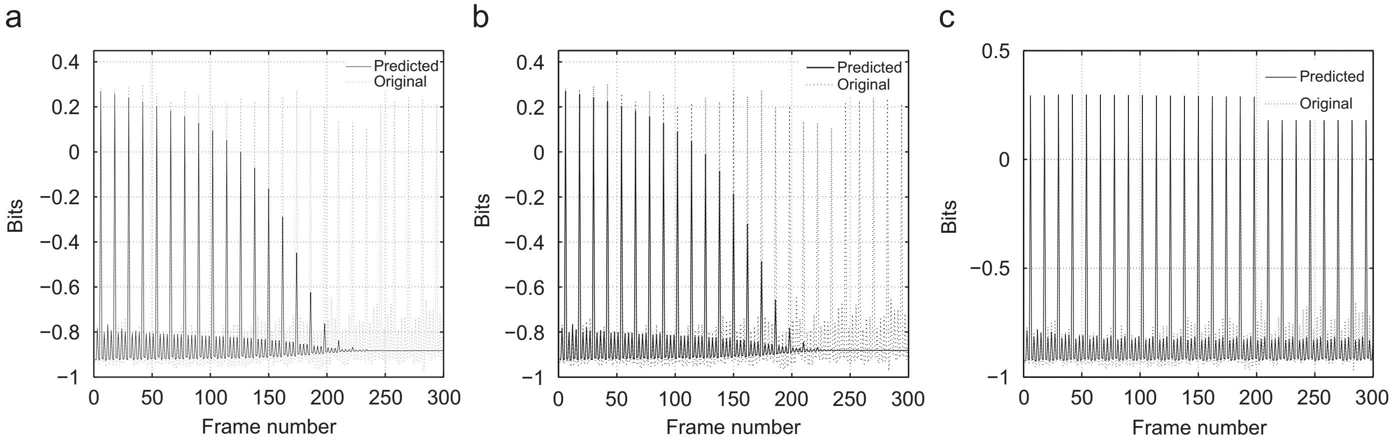

**Fig. 9.** Recursive predictions obtained by (a) TDNN, (b) Elman, and (c) NARX-SP networks.

networks, respectively. For this simulation, all the neural networks are required to predict recursively the sample values of the VBR video traffic trace for 300 steps ahead in time. For all networks, we have set $d_E = 12, \tau = 1, d_y = 24$ and trained the neural models for 300 epochs. For these training parameters, the NARX-SP predicted the video traffic trace much better than the TDNN and Elman networks.

As we did for the laser time series, we again emphasize that the results reported in Fig. 9 did not mean to say that the TDNN and Elman networks cannot ever predict the video traffic trace as well as the NARX-SP. They only mean that, for the same training and configuration parameters, the NARX-SP has greater computational power provided by the output regressor. Recall that the MLP is a universal function approximation; and so, any MLP-based neural model, such as the TDNN and Elman networks, are in principle able to approximate complex function with arbitrary accuracy, once enough training epochs and data are provided.

## 5. Conclusions and further work

In this paper, we have shown that the NARX neural network can successfully use its output feedback loop to improve its predictive performance in complex time series prediction tasks. We used the well-known chaotic laser and real-world VBR video traffic time series to evaluate empirically the proposed approach in long-term prediction tasks. The results have shown that the proposed approach consistently outperforms standard neural network based predictors, such as the TDNN and Elman architectures.

Currently, we are evaluating the proposed approach on several other applications that require long-term predictions, such as electric load forecasting and financial time series prediction. Applications to signal processing tasks, such as communication channel equalization, are also being planned.

## Acknowledgements

The authors would like to thank CNPq (Grant #506979/2004-0), CAPES/PRODOC and FUNCAP for their financial support.

## References

[1] H.D. Abarbanel, T.W. Frison, L. Tsimring, Obtaining order in a world of chaos, IEEE Signal Process. Mag. 15 (3) (1998) 49–65.

[2] A.F. Atiya, M.A. Aly, A.G. Parlos, Sparse basis selection: new results and application to adaptive prediction of video source traffic, IEEE Trans. Neural Networks 16 (5) (2005) 1136–1146.

[3] A.F. Atiya, S.M. El-Shoura, S.I. Shaheen, M.S. El-Sherif, A comparison between neural-network forecasting techniques-case study: river flow forecasting, IEEE Trans. Neural Networks 10 (2) (1999) 402–409.

[4] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Networks 5 (2) (1994) 157–166.

[5] J. Beran, R. Sherman, M.S. Taqqu, W. Willinger, Long-range dependence in variable-bit-rate video traffic, IEEE Trans. Commun. 43 (234) (1995) 1566–1579.

[6] A. Bhattacharya, A.G. Parlos, A.F. Atiya, Prediction of MPEG-coded video source traffic using recurrent neural networks, IEEE Trans. Neural Networks 51 (8) (2003) 2177–2190.

[7] G. Box, G.M. Jenkins, G. Reinsel, Time Series Analysis: Forecasting & Control, third ed., Prentice-Hall, Englewood Cliffs, NJ, 1994.

[8] L. Cao, Practical method for determining the minimum embedding dimension of a scalar time series, Physica D 110 (1–2) (1997) 43–50.

[9] M.C. Casdagli, Recurrence plots revisited, Physica D 108 (1) (1997) 12–44.

[10] S. Chen, S.A. Billings, P.M. Grant, Nonlinear system identification using neural networks, Int. J. Control 11 (6) (1990) 1191–1214.

[11] D. Coyle, G. Prasad, T.M. McGinnity, A time-series prediction approach for feature extraction in a brain–computer interface, IEEE Trans. Neural Syst. Rehabil. Eng. 13 (4) (2005) 461–467.

[12] S. Dablemont, G. Simon, A. Lendasse, A. Ruttiens, F. Blayo, M. Verleysen, Time series forecasting with SOM and local non-linear models—application to the DAX30 index prediction, Proceedings of the Fourth Workshop on Self-Organizing Maps, (WSOM) 03 (2003).

[13] A.D. Doulamis, N.D. Doulamis, S.D. Kollias, An adaptable neural network model for recursive nonlinear traffic prediction and modelling of MPEG video sources, IEEE Trans. Neural Networks 14 (1) (2003) 150–166.

[14] A. Erramilli, M.R.D. Veitch, W. Willinger, Self-similar traffic and network dynamics, Proc. IEEE 9 (5) (2002) 800–819.

[15] A.M. Fraser, H.L. Swinney, Independent coordinates for strange attractors from mutual information, Phys. Rev. A 33 (1986) 1134–1140.

[16] M. Grossglauser, J.C. Bolot, On the relevance of long-range dependence in network traffic, IEEE/ACM Trans. Networking 7 (4) (1998) 329–640.

[17] S. Haykin, X.B. Li, Detection of signals in chaos, Proc. IEEE 83 (1) (1995) 95–122.

[18] S. Haykin, J.C. Principe, Making sense of a complex world, IEEE Signal Process. Mag. 15 (3) (1998) 66–81.

[19] R. Hecht-Nielsen, Kolmogorov's mapping neural network existence theorem, in: Proceedings of the IEEE International Conference on Neural Networks, vol. 2, 1987.

[20] D. Heyman, T. Lakshman, What are the implications of long-range dependence for VBR video traffic engineering, IEEE/ACM Transactions on Networking 4 (1996) 301–317.

[21] U. Huebner, N.B. Abraham, C.O. Weiss, Dimensions and entropies of chaotic intensity pulsations in a single-mode far-infrared NH3 laser, Phys. Rev. A 40 (11) (1989) 6354–6365.

[22] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, second ed., Cambridge University Press, Cambridge, 2006.

[23] J.F. Kolen, S.C. Kremer, A Field Guide to Dynamical Recurrent Networks, Wiley, IEEE Press, New York, 2001.

[24] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, On the self-similar nature of ethernet traffic (extended version), IEEE/ACM Trans. Network. 2 (1) (1994) 1–15.

[25] A. Lendasse, J. Lee, V. Wertz, M. Verleysen, Forecasting electricity consumption using nonlinear projection and self-organizing maps, Neurocomputing 48 (1–4) (2002) 299–311.

[26] I.J. Leontaritis, S.A. Billings, Input–output parametric models for nonlinear systems—part I: deterministic nonlinear systems, Int. J. Control 41 (2) (1985) 303–328.

[27] T. Lin, B.G. Horne, C.L. Giles, How embedded memory in recurrent neural network architectures helps learning long-term temporal dependencies, Neural Networks 11 (5) (1998) 861–868.

[28] T. Lin, B.G. Horne, P. Tino, C.L. Giles, Learning long-term dependencies in NARX recurrent neural networks, IEEE Trans. Neural Networks 7 (6) (1996) 1424–1438.

[29] T. Lin, B.G. Horne, P. Tino, C.L. Giles, A delay damage model selection algorithm for NARX neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2719–2730.

[30] L. Ljung, System Identification: Theory for the User, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.

[31] T. Masters, Practical Neural Network Recipes in C++, Academic Press, New York, 1993.

[32] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Trans. Neural Networks 1 (1) (1990) 4–27.

[33] M. Norgaard, O. Ravn, N.K. Poulsen, L.K. Hansen, Neural Networks for Modelling and Control of Dynamic Systems, Springer, Berlin, 2000.

[34] A.K. Palit, D. Popovic, Computational Intelligence in Time Series Forecasting, first ed., Springer, Berlin, 2005.

[35] B.A. Pearlmutter, Gradient calculations for dynamic recurrent neural networks: a survey, IEEE Trans. Neural Networks 6 (5) (1995) 1212–1228.

[36] J.C. Principe, N.R. Euliano, W.C. Lefebvre, Neural Adaptive Systems: Fundamentals Through Simulations, Wiley, New York, 2000.

[37] O. Rose, Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems, in: Proceedings of the 20th Annual IEEE Conference on Local Computer Networks (LCN'95), IEEE Computer Society, 1995.

[38] H.T. Siegelmann, B.G. Horne, C.L. Giles, Computational capabilities of recurrent NARX neural networks, IEEE Trans. Syst. Man Cybern. B-27 (2) (1997) 208–215.

[39] A. Sorjamaa, J.H.N. Reyhani, Y. Ji, A. Lendasse, Methodology for long-term prediction of time series, Neurocomputing 70 (16–18) (2007) 2861–2869.

[40] F. Takens, Detecting strange attractors in turbulence, in: D.A. Rand, L.-S. Young (Eds.), Dynamical Systems and Turbulence, Lecture Notes in Mathematics, vol. 898, Springer, Berlin, 1981.

[41] E.A. Wan, Finite impulse response neural networks with applications in time series prediction, Ph.D. Thesis, Department of Electrical Engineering, Stanford University, USA, 1993.

[42] A. Weigend, N. Gershefeld, Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley, Reading, 1994.

[43] P. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (10) (1990) 1550–1560.

[44] R.J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Comput. 1 (2) (1989) 270–280.

[45] H. Yousefi'zadeh, E.A. Jonckheere, Dynamic neural-based buffer management for queueing systems with self-similar characteristics, IEEE Trans. Neural Networks 16 (5) (2005) 1163–1173.

**Jose Maria P. Menezes Jr.** was born in Teresina, Piaui, Brazil, in 1980. He received the B.S. degree in Electrical Engineering from the Federal University of Ceara in 2003 and his M.Sc. degree in Teleinformatics Engineering from the same university in 2006, working on recurrent neural architectures for time series prediction. Currently, he is pursuing the Ph.D. degree in Teleinformatics. His main research interests are in the areas of neural networks, time series prediction, communication network traffic modelling and nonlinear dynamical systems.

**Guilherme A. Barreto** was born in Fortaleza, Ceara, Brazil, in 1973. He received his B.S. degree in Electrical Engineering from the Federal University of Ceara in 1995, and both the M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Sao Paulo in 1998 and 2003, respectively. In 2000, he developed part of his Ph.D. studies at the Neuroinformatics Group of the University of Bielefeld, Germany. He is in the organizing committee of the Brazilian Symposium on Neural Networks (SBRN) and has been serving as reviewer for several neural-network related journals (IEEE TNN, IEEE TSMC, IEEE TKDE, IEEE TSP, Neurocomputing and IJCIA), and conferences (IJCNN, ICASSP, SBRN, among others). Currently, he is with the Department of Teleinformatics Engineering, Federal University of Ceara. His main research interests are self-organizing neural networks for signal and image processing, time series prediction, pattern recognition, and robotics.