A Self-Organizing NARX Network and its Application to Prediction of Chaotic Time Series

Guilherme de A. Barreto Aluizio F. R. Araújo
Departamento de Engenharia Elétrica (USP)
Av. Dr. Carlos Botelho, 1465
13560-970, São Carlos, SP, Brasil
{*gbarreto, aluizioa*}*@sel.eesc.sc.usp.br*

Abstract

This paper introduces the concept of *dynamic embedding manifold* (DEM), which allows the Kohonen self-organizing map (SOM) to learn dynamic, nonlinear input-ouput mappings. The combination of the DEM concept with the SOM results in a new modelling technique that we called *Vector-Quantized Temporal Associative Memory* (VQTAM). We use VQTAM to propose an unsupervised neural algorithm called *Self-Organizing NARX* (SONARX) network. The SONARX network is evaluated on the problem of modeling and prediction of three chaotic time series and compared with MLP, RBF and autoregressive (AR) models. Its is shown that SONARX exhibits similar performance when compared to MLP and RBF, while producing much better results than the AR model. The influence of the number of neurons, the memory order, the number of training epochs and the size of the training set in the final prediction error is also evaluated.

1 Introduction

The ability to handle spatiotemporal data is a requirement for an artificial device (model) trying to infer any dynamical properties of the system under study. The observed signals, generated by nonlinear dynamic systems, may be very complex. Examples of such a complexity are given by time series that possess a chaotic nature. Artificial neural networks, as computational models of learning and memory in living organisms, should be capable of dealing with such complex spatiotemporal data. In particular, the use of nonlinear neural models opens the door to study modeling and prediction of chaotic time series.

This task poses an important question: can prediction of a chaotic signal (here defined as a time series generated by a nonlinear dynamical system with a low dimension attractor) provide any knowledge about the system that produces the time series? It is known that the dynamical system is a map that relates the current state with the previous ones in state space. Therefore, when a model predicts the next point in state space, it aims to identify the map that characterizes the original dynamical system. The problem is that the map is not guaranteed to be unique when one uses a finite number of state space points. The other problem is that the map may be very complex and nonlinear, therefore there is no guarantee to identify the map with sufficient accuracy through prediction.

This paper shows how to use the Kohonen self-organizing map (SOM) [1] to learn nonlinear dynamic mappings such as those associated with chaotic time series. We propose the concept of *dynamic embedding manifold* (DEM). The combination of the DEM concept with the SOM results in a new modelling technique that we called *Vector-Quantized Temporal Associative Memory* (VQTAM). We use VQTAM technique to propose an unsupervised neural algorithm called *Self-Organizing NARX* (SONARX) network. With the SO-NARX network it is possible to identify the dynamics of nonlinear dynamic systems using the one-step prediction framework on a segment of the chaotic time series.

The remaining of the paper is organized as follows. In Section 2 we briefly describe how the Kohonen SOM can be used to learn static input-output mappings. In Section 3 we present several ways of introducing the temporal dimension into the SOM and introduce the concept of dynamic embedding manifold. In Section 4 we discuss the issue of modeling chaotic time series. In Section 5 several simulations are carried out using the SO-NARX network in one-step-ahead prediction tasks. The paper is concluded in Section 6.

0-7803-7044-9/01/$10.00 ©2001 IEEE

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 20,2022 at 13:42:35 UTC from IEEE Xplore. Restrictions apply.

2 The Self-Organizing Map

The neurons in the SOM are arranged in an output layer $\mathcal{A}$ with fixed topology, i.e., in one, two and even in three-dimensional configurations. Each neuron $i \in \mathcal{A}$ is associated with a weight vector $\mathbf{w}_i = [w_{i1} \quad w_{i2} \quad \ldots \quad w_{in}]^T \in \Re^n$ with the same dimension as the input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \Re^n$. Through an unsupervised learning process, the output neurons become tuned and organized after several presentations of the data. The learning algorithm that leads to self-organization can be summarized in two steps: (i) Search for the winning neuron, i^*, and (ii) weight adaptation. The first step is competitive, while the second one is cooperative.

2.1 Learning Static Input-Output Mappings

The SOM algorithm has beenmostly applied to pattern recognition tasks which involves clustering or categorization, classification and data visualization. More recently, the SOM has been extended to learn static input-output mappings. This is possible through the concept of *embedding manifold* [2]. Mathematically, the embedding manifold, X^{emb}, is the result of the Cartesian product of the input space X^{in} and the output space X^{out}:

$$X^{emb} \equiv X^{in} \times X^{out} \subset \Re^d \qquad (1)$$

where $d = \dim(X^{in}) + \dim(X^{out})$. The dimensions of the input and output spaces do not need to be the same. Generally speaking, the joint distribution of the input-output data belong to a nonlinear manifold which is embedded in the augmented input space $\Re^d$.

The practical effect of this definition is that the input vector $\mathbf{x}^{in}(t) \in X^{in}$ and the output vector $\mathbf{x}^{out}(t) \in X^{out}$ are put together to form a single, *augmented* input vector $\mathbf{x}^{emb}(t) \in X^{emb}$. In other words, the network input vector and the weight vector of a neuron i in the SOM has their dimensions increased to include de output data, being described now as follows:

$$\mathbf{x}^{emb} = \begin{pmatrix} \mathbf{x}^{in} \\ \mathbf{x}^{out} \end{pmatrix} \quad \text{and} \quad \mathbf{w}_i^{emb} = \begin{pmatrix} \mathbf{w}_i^{in} \\ \mathbf{w}_i^{out} \end{pmatrix}$$

For example, to learn the inverse kinematics of an industrial robot arm [2], the input vector $\mathbf{x}^{in}$ can be the Cartesian position of the end-effector, while the output vector $\mathbf{x}^{out}(t)$ can be the joint angles. For training purpose, the winning neurons are found using $\mathbf{x}^{emb}$ or, more commonly, using only the portion corresponding to $\mathbf{x}^{in}$. The input vector $\mathbf{x}^{in}$ is used to update the weights $\mathbf{w}^{in}$, while $\mathbf{x}^{out}$ updates the weights $\mathbf{w}^{out}$. As the training evolves, the SOM learns to associate a given output signal with the corresponding input, while performing vector quantization of the input and output spaces. For this reason, we refer to this technique as *Vector Quantized Associative Memory* (VQAM). It is worth noting that standard supervised neural networs such as MLP and RBF learn an input-output mapping through error correction schemes, while the SOM learns it via the VQAM memory scheme. The element responsible for the "connection" of the input space $\mathcal{X}^{in}$ with the output $\mathcal{X}^{out}$ is the winning neuron i^*.

3 Time in the SOM

A major characteristic of the standard SOM described in the previous section is that it can only learn an input-output mapping that is *static*. That is, those mappings that can be represented as follows:

$$\mathbf{y}(t) = \mathbf{f}(\mathbf{x}(t))$$

where $\mathbf{x}(t)$ and $\mathbf{y}(t)$ represent the system input and output, respectively. For a neural network to be *dynamic*, it must be given memory about past states of the system being modeled. Currently, four techniques have been used in order to enable the SOM to process temporal data. The first one adds temporal information to the input of the SOM through the use of *delay lines* and *leaky integrators* [3]. The second technique adds temporal information internally to the network in the activation and/or learning rules [4]. The third technique uses SOMs hierarchically, trying to capture spatiotemporal aspects of the input through successive refinements [5]. The fourth approach uses a feedback loop to insert temporal information into the SOM [6].

3.1 Learning Dynamic Input-Output Mappings

In this section we propose to extend the concept of embedding manifold in order to take into account temporal aspects. In a simple notation, we have:

$$X^{emb}(t) \equiv X^{in}(t) \times X^{out}(t)$$

where $X^{emb}(\cdot)$ is called *dynamic embedding manifold* (DEM). The main novelty is the inclusion of temporal dependence in each one of the involved spaces, $X^{in}(t)$ and $X^{out}(t)$. Thus, the augmented vector $\mathbf{x}^{emb}(t)$ $=[\mathbf{x}^{in}(t)\ \mathbf{x}^{out}(t)]^T$ contains now information about current and past states of the system. This memory mechanism can assume the form of time delays and/or leaky integrators.

With the inclusion of *time* into the embedding manifold, the SOM can now be used to approximate input-output mappings such as those described by:

$$\mathbf{y}(t) = \mathbf{f}(\mathbf{x}(t), \ldots, \mathbf{x}(t-q); \mathbf{y}(t-1), \ldots, \mathbf{y}(t-p)) \quad (2)$$

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 20,2022 at 13:42:35 UTC from IEEE Xplore. Restrictions apply.

where p and q are the orders of the input and output, respectively. The ideia underlying the concept of embedding manifold also applies here. The only difference is that the nature of the association between $X^{in}(t)$ and $X^{out}(t)$ is not only *spatial* but also *temporal*. For this reason, we refer to this technique as *vector quantized temporal associative memory* (VQTAM). As an illustrative example of the power of the VQTAM technique, we use the standard SOM to perform nonlinear modelling and prediction of chaotic time series.

4 The SONARX Network

Consider a scalar time series denoted by $\{y(t)\}$, which is described by a special case of Eq. (2), called *nonlinear regressive model* of *order* p as follows:

$$y(t) = \mathbf{f}\left(y(n-1), y(n-2), \ldots, y(n-p)\right) + \varepsilon(t) \quad (3)$$

where $\mathbf{f}$ is a *nonlinear function* of its arguments and ε is a *residual* which is included to take into account any modeling uncertainty and/or noise. It is assumed that $\varepsilon(t)$ is drawn from a *white Gaussian noise process*. The nonlinear function $\mathbf{f}$ is unknown, and the only thing that we have available is a set of *observables*: $y(1), y(2), \ldots, y(N)$, where N is the total length of the time series. The requirement is to construct a physical model of the time series, given this data set. To do so, we propose to use the SOM with the DEM concept as a *one-step predictor* of some order p. Since this network is able to approximate nonlinear input-ouput mappings such as those described by Eq. (2), we refer to it as the *Self-Organizing NARX* (SONARX) network, as being the unsupervised version of the NARX network [7].

For training purpose, the past p samples $y(n-1)$, $y(n-2), \ldots, y(n-p)$ form the SONARX network input vector $\mathbf{x}^{in}(t)$, while the output vector $\mathbf{x}^{out}$ is defined as the future sample $y(t)$. Then, we have:

$$\begin{aligned} \mathbf{x}^{in}(t) &= [y(t-1), y(t-2), \ldots, y(t-p)] \\ \mathbf{x}^{out}(t) &= y(t+T) \end{aligned}$$

where T defines the horizon of prediction. In this paper, we set $T = 1$. The winning neuron i^* is determined on the basis of the input vector $\mathbf{x}^{in}(t)$:

$$i^*(t) = \arg\min_i \{\|\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)\|\} \quad (4)$$

The weight vectors are adjusted as follows:

$$\begin{aligned} \mathbf{w}_i^{in}(t+1) &= \mathbf{w}_i^{in}(t) + \alpha(t)[\mathbf{x}^{in}(t) - \mathbf{w}_i^{in}(t)] \\ \mathbf{w}_i^{out}(t+1) &= \mathbf{w}_i^{out}(t) + \alpha'(t)[\mathbf{x}^{out}(t) - \mathbf{w}_i^{out}(t)] \end{aligned}$$

for all neurons i belonging to the neighborhood of the winning unit i^*. The parameters $0 < \alpha(t), \alpha'(t) \leq 1$ are the learning rates associated with $\mathbf{w}_i^{in}(t)$ and $\mathbf{w}_i^{out}(t)$, respectively. During the test phase, the input vector $\mathbf{x}^{in}(t)$ is formed with p past unseen samples, the winning unit is found according to Eq. (4), and the one-step-ahead estimate of the time series is recovered from $\mathbf{w}_{i^*}^{out}(t)$:

$$\hat{y}(t) \equiv \mathbf{w}_{i^*}^{out}(t)$$

This is an *feedforward prediction scheme* in the sense that the actual output of the network is *not* fed back to the input during prediction. The VQTAM approach used to train the SO-NARX network evolves so as to associate the current value of the time series with its predecessors. Due to the nature of the SOM learning, this association converges to a steady state that minimizes the squared value of the *prediction error*, $e(t) = y(t) - \hat{y}(t)$, $p+1 \leq t \leq N$ in a topology-preserving sense.

4.1 Three Chaotic Time Series

Modeling and prediction of chaotic time series is still a relatively new research topic, dating back to 1987 [8]. The interesting point about this field is that, even under noise-free conditions, a chaotic system shows an apparently random behavior, but may still be modeled using techniques from nonlinear *deterministic* system identification. However, even with perfect modeling of the dynamical behavior of the system in the noise free case, only short-term predictions are possible due to the extreme sensitivity of chaotic systems to uncertainties in initial conditions. In this paper, three well-known chaotic time series widely used as benchmarks are considered: the Mackey-Glass time series [9], the Lorenz time series [10], and the chaotic laser time series [11]. For all time series a total of 3000 samples were used and scaled between $[-1, 1]$. Each series was further divided into two sets for cross-validation purpose: 2000 samples for training and 1000 samples for testing.

5 Simulations

An unidimensional SO-NARX with 400 neurons is trained for 250 epochs. Each epoch consists of one presentation of the entire sequence minus the τ first samples. The established criterion to compare one-step-ahead predictors is based on the normalized average prediction error, $pe = \frac{1}{M\sigma^2}\sum_{i=1}^{M}(\hat{y}(i) - y(i))^2$, where $\hat{x}(i)$ is the predicted value, $x(i)$ is the actual value of the sample, M is the number of samples and σ^2 is the variance of the time series. The first set of tests is concerned with the determination of the order p of the

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 20,2022 at 13:42:35 UTC from IEEE Xplore. Restrictions apply.

memory for the vector $\mathbf{x}^{in}(t)$. Figure 1 shows the result for the Mackey-Glass time series. The prediction errors were computed for 800 time steps, and the minimum value was found for $p = 2$. For the Lorenz and Laser time series the minimum value of the prediction error ocurred for $p = 3$.

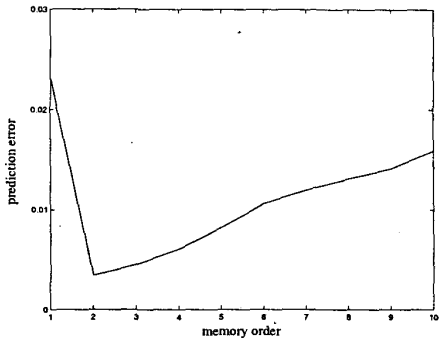

Figure 1: Determination of memory order of SO-NARX for the Mackey-Glass time series.

A sample of the Laser time series used for testing the SONARX together with the predicted values is shown in the Figures 2a and 2b for $p = 1$ and $p = 3$ (best value suggested by SONARX for this time series), respectively. The histograms of the prediction errors $pe(t)$ for the Mackey-Glass time series and the corresponding autocorrelation functions are shown in Figures 3a,b and 4a,b, respectively. The analysis of the residuals can be used to help choosing an adequate memory order. For example, the histogram for the SO-NARX network with $p = 2$ is more similar to a Gaussian distribution than that for $p = 1$. Furthermore, the autocorrelation function for $p = 2$ indicates that the prediction errors are less correlated than those for $p = 1$. These graphics confirms that the memory order to model correctly the Mackey-Glass time with the SO-NARX network should be $p = 2$.

The next simulations evaluate the sensitivity of the SO-NARX to several parameters such as the number of neurons, the length of the training sequence, and the number of training epochs. Figure 5 shows the evolution of the final prediction error for the Laser time series as the number of neurons increase from 1 to 1000. The prediction error decays exponentially fast and stabilizes around 500 neurons. From this value onwards, larger changes in the number of neurons result in very small changes in the final prediction error. The same behavior occurs for the other two time series.

Figure 6 shows the evolution of the final prediction error for the Laser time series as the number of training

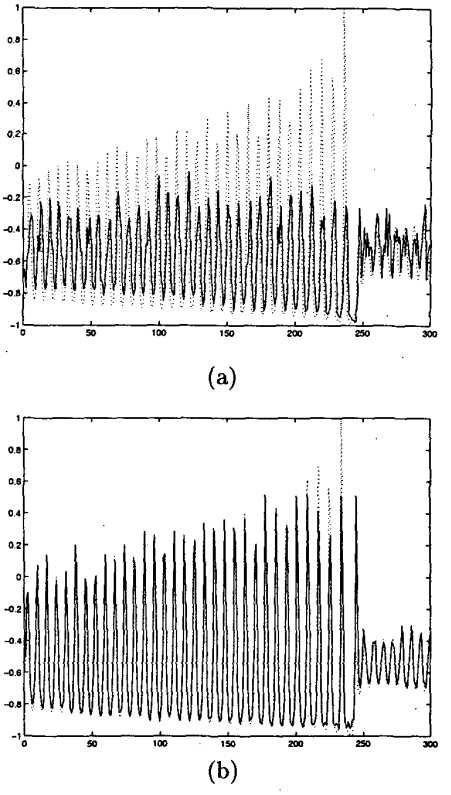

Figure 2: Predicted (solid line) and actual (dotted line) test samples of the Laser time series for (a) $p = 1$ and (b) $p = 3$.

epochs increase from 1 to 500. The prediction error decays exponentially very fast and stabilizes after 150 epochs. From this quantity onwards, larger increments in the number of epochs result in very small changes in the final prediction error. In other words, this means that 150 epoch is sufficient for the SO-NARX to learn in a statistical sense the dynamics of the chaotic system. The same behavior is observed for the other two series.

Figure 7 shows the evolution of the final prediction error for the Mackey-Glass and Lorenz time series as the number of training samples from 100 to 10000. For the Mackey-Glass time series the prediction error decays exponentially very fast and stabilizes after 5500 samples. From this quantity onwards, larger increments in the number of samples result in very small changes in the final prediction error. For the Lorenz time series the prediction error reaches a minimum around 300 samples only. From this value onwards, the prediction error increases in a very small rate until it stabilizes around 5500 samples. The interpretation of the results shown

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 20,2022 at 13:42:35 UTC from IEEE Xplore. Restrictions apply.

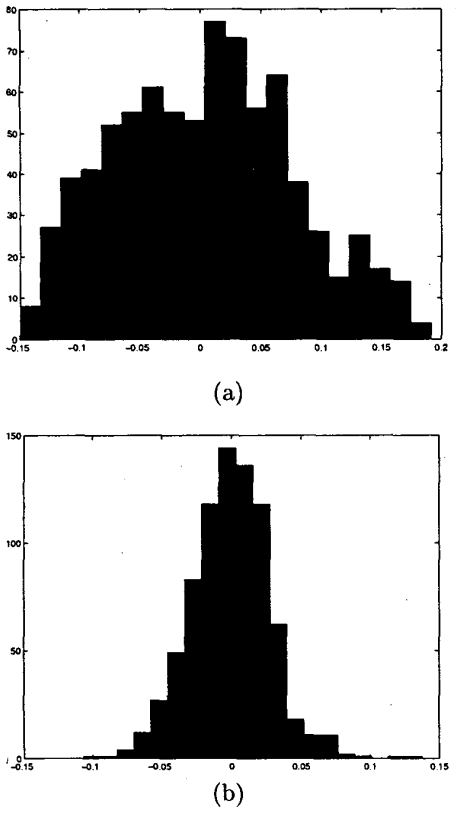

Figure 3: Histograms of the prediction errors $e(t)$, $t = 1, \ldots, 800$, for (a) $p = 1$, and (b) $p = 2$.

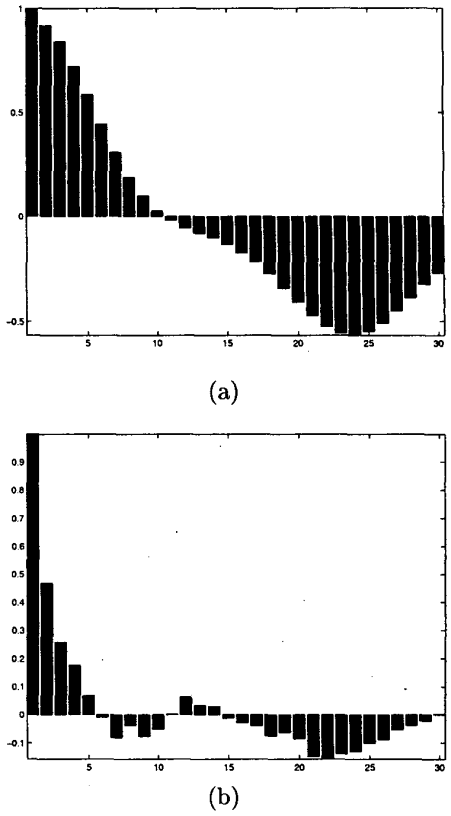

Figure 4: Autocorrelation functions of the prediction errors $e(t)$, $t = 1, \ldots, 800$, for (a) $p = 1$, and (b) $p = 2$.

in Figures 6 and 7 is as follows. Since the three time series are chaotic, the "randomness" that they exhibited does not go away by gathering more information after a certain number of training samples or epochs.

The last simulation compares the SONARX with MLP and RBF networks and a linear AR model. The orders $p = 49, 20, 33$ of the AR model for the three time series were found by the AIC procedure [12] and the coefficients were computed using a robust Bayesian method proposed in [13]. The MLP network has 7 units in the hidden layer and 1 unit in the output layer. The transfer function for all neurons in the hidden and output layers is the hyperbolic tangent function. The MLP was trained with the backpropagation learning algorithm with adaptive learning rate and momentum. The RBF network has 17 gaussian basis functions whose the centers were computed by the K-means algorithm, and 1 output neuron whose weights were computed via the pseudoinverse method [14]. Number of input neurons of the MLP and RBF networks is equal to the dimension of $\mathbf{x}^{in}$. The results in Table 1 shows that it is possible for the SONARX network, using a simpler formulation of the prediction task, to produce prediction errors of the same of order of magnitude of those produce by MLP and RBF networks. Furthermore, the SONARX network is faster to train than the MLP and RBF networks for the same final prediction error.

Table 1: SONARX versus MLP, RBF and AR models.

	Mackey-Glass	**Lorenz**	**Laser**
SONARX	0.0034431	0.0028182	0.040541
MLP	0.0011379	0.0011939	0.026615
RBF	0.0077794	0.0015367	0.035016
AR(49,20,33)	0.64220	0.020582	0.81695

6 Conclusion

In this paper, we proposed a simple mathematical concept, called *dynamic embedding manifold* (DEM), which enables the Kohonen map to learn nonlinear, dynamic input-output mappings. The combined use

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 20,2022 at 13:42:35 UTC from IEEE Xplore. Restrictions apply.

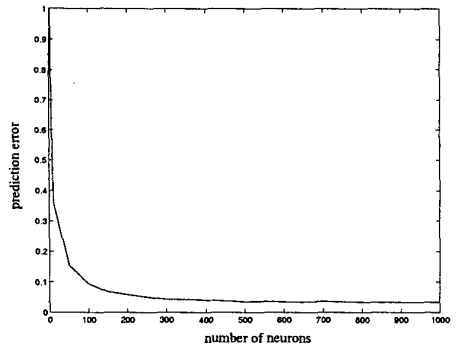

Figure 5: Evolution of the prediction error for the Laser series by increasing the number of neurons in the SONARX network.

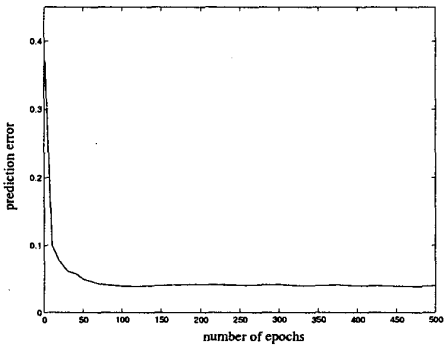

Figure 6: Evolution of the prediction error for the Laser series as the number of epochs for training the SONARX network increases.

of the SOM with the DEM concept results in a technique that we called *vector quantized temporal associative memory* (VQTAM). This technique was succesfully applied to the problem of modeling and prediction of chaotic time series. The VQTAM technique is quite general and can be used to learn input-output mappings occurring in problems in which the temporal dimension plays an essential role. Further studies are being carried out in order to apply the concepts of DEM and VQTAM to sensorimotor learning and control.

Acknowledgments: The authors thank FAPESP for its financial support under the grant #98/12699-7.

References

[1] T. Kohonen, *Self-organizing maps*, Springer Verlag, 2nd extended edition, 1997.

[2] J. Walter and H. Ritter, "Rapid learning with parametrized self-organizing maps," *Neurocomputing*, vol. 12, pp. 131–153, 1996.

[3] G. de A. Barreto and A. F. R. Araújo, "Time in self-organizing maps: An overview of models," *Intl. J. Comput. Research*, vol. 10, no. 2, pp. 235–259, 2000.

[4] T. Koskela, M. Varsta, J. Heikkonen, and K. Kaski, "Time series prediction using recurrent SOM with local linear models," *Intl. J. Knowledge-based Intelligent Engineering Syst.*, vol. 2, no. 1, pp. 60–68, 1998.

[5] J. Kangas, "Time-delayed self-organizing maps," in *Proc. IJCNN'90*, 1991, vol. II, pp. 331–336.

[6] T. Voegtlin, "Context quantization and contextual self-organizing maps," in *Proc. IJCNN'2000*, 2000, vol. 6, pp. 20–25.

[7] T. Lin, B.G. Horne, P. Tino, and C.L. Giles, "Learning long–term dependencies in NARX recurrent neural networks," *IEEE Trans. Neural Net.*, vol. 7, no. 6, pp. 1329–1338, 1996.

[8] D. Kugiumtzis, B. Lillekjendlie, and N. Christophersen, "Chaotic time series Part II: System identification and prediction," *Model., Ident. & Contr.*, vol. 15, no. 4, pp. 225–243, 1994.

[9] M. C. Mackey and L. Glass, "Oscillations and chaos in physiological control systems," *Science*, vol. 197, pp. 287–289, 1977.

[10] E. N. Lorenz, "Deterministic nonperiodic flow," *J. Atmospheric Sci.*, vol. 357, pp. 130–141, 1963.

[11] A. Weigend and N. Gershefeld (editors), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, 1993.

[12] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Contr.*, vol. 19, no. 6, pp. 716–723, 1974.

[13] G. de A. Barreto and M. G. Andrade, "Bayesian inference and markov chain monte carlo methods applied to streamflow forecasting," in *Proc. 6th PMAPS*, Funchal, Ilha da Madeira, 2000, pp. FOR–34.

[14] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan Publishing Company, New york, 1994.

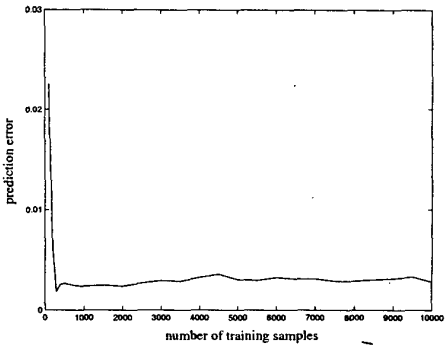

Figure 7: Evolution of the prediction error for the Lorenz time series as the number of training samples increases.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DO CEARA. Downloaded on December 20,2022 at 13:42:35 UTC from IEEE Xplore. Restrictions apply.