Nonlinear Time Series Modeling and Prediction Using Functional Networks. Extracting Information Masked by Chaos

E. Castillo and J.M. Gutiérrez

Dpto. de Matemática Aplicada y Ciencias de la Computación, Universidad de Cantabria, E-39005, SPAIN

Abstract

Functional networks are a recently introduced extension of neural networks which deal with general functional models instead of sigmoidal-like ones. In this paper we show that functional network architectures provide simple and efficient techniques to model nonlinear time series. To this aim, the neural functions are approximated by finite combinations of known functions from a given family (polynomials, Fourier expansions, etc.) and the associated coefficients are estimated from data. In this paper we present two architectures from the same functional networks family, introducing efficient learning algorithms leading to error functions with a single global minimum that need not of an iterative process to be learned. We demonstrate the effectiveness of these models by applying them to several examples, including data from the Hénon, Holmes, Lozi and Burgers maps. Finally, we show that these models can also be used to extract information masked in chaotic time series.

1 Introduction

Time series analysis is an important technique used in many observational disciplines, such as physics, engineering and finance, to infer properties of a system from the analysis of a measured time record (data). This is done by fitting a representative model to the data with the aim of discovering the underlying structure. There are many standard methods associated with traditional time series analysis: linear regression, ARIMA models, etc. (see [1,2]). However, in recent years several innovative approaches have emerged (neural networks [3,4], time-delay embedding [5,6], wavelets [7], global dynamical models [8], etc.) to deal with problems associated with non-linear time series. These methods give new insight for this type of systems not available with the standard time series methods. For instance, Stern [4] shows that neural networks (a multilayer perceptron trained with the backpropagation

algorithm) outperforms standard auto-regressive models to approximate some chaotic time series. On the other hand, Farmer and Sidorowich [9] shows that some information about the actual dynamics of a nonlinear time series can be obtained by using a local approximation in a convenient state space of delay coordinates (using the dynamics resulting in the embedding space).

In this paper we present a new approach to this problem using functional networks. Functional networks are a natural extension of neural networks where the activation functions are unknown functions from given families to be estimated during the learning process (see [10,11] for an introduction to this subject and [12] for a comparison between functional and neural networks). Then, the threshold-like sigmoidal functions, which may be inappropriate to model the actual dynamics of non-linear time series, are replaced by functions from appropriate families for each specific problem (in this paper we shall consider the families of polynomial and trigonometric Fourier functions, but other families can also be considered). As we shall see, functional networks constitute an efficient and natural approach to time series analysis and will allow us to infer the deterministic structure of chaotic time series in a simple way. The topology of the network is a graphical representation of the properties we may know about the dynamics of the system such as, for example, the associativity property (see [11] for more details) and determines a functional expression involving the neural functions associated with the units of the network. We present efficient algorithms to learn this functional expression from data using least squares and demonstrate its effectiveness by applying it to some interesting time series obtained from several chaotic maps (Hénon, Holmes, Lozi, and Burgers maps). Note that as opposed to the method of Farmer and Sidorowich [9], functional networks are a global modeling technique, since neuron functions are approximated using least squares.

As an application of the models presented in this paper we show that they can also be used to extract the information masked in chaotic time series. The idea of encoding information in chaotic time series has been recently proposed as a promising tool in the field of secure communications (see [13] for a survey). Different approaches to implement this idea have been proposed and applied in experimental settings [14–16]. Most of them are based on synchronization of chaotic systems. In this case we have two chaotic systems (the transmitter and the receiver) that are able to synchronize. Then, the message is added to the broadband chaotic signal and transmitted to the receiver, which is synchronized to the chaotic component of the signal, thus, allowing to extract the information (see (a) in Figure 1). In this paper we show that, in some cases, the transmitter system can be replaced by some alternative mechanism that is also able to reproduce the deterministic component of the original chaotic system from the transmitted data (see (b) in Figure 1) and, therefore, extract the masked information. The examples shown in this paper prove that functional networks can be efficiently used for this task. The problem of extracting information masked by chaos has been already discussed from different points of view (see [17] and references therein). For instance Pérez and Cerdeira [18] show that it is possible to find some suitable return maps which allow the information to be extracted. Zhou and Chen [19] use a similar approach when the information is masked in symmetric maps. As we shall see, functional network provide a global method for information unmasking which is based on approximating the deterministic structure of the chaotic system used to encode the information.

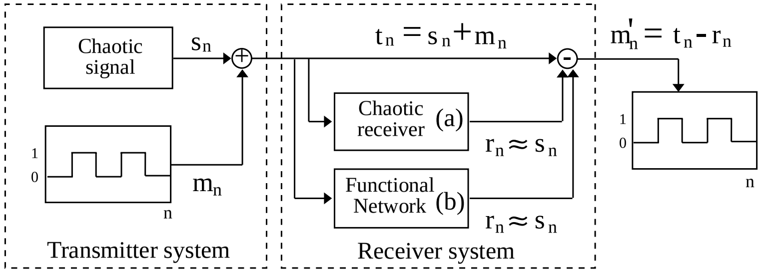

Fig. 1. Scheme for secure communications based on chaos synchronization.

This paper is structured as follows. In Section 2 we give a brief introduction to functional networks. In Section 3 we present the general functional network architecture to be used in this paper and we discuss two simple cases leading to straightforward learning algorithms. Section 4 analyzes the problem of model inference, describing some applications to modeling interesting chaotic time series obtained from several chaotic maps with a variety of structures. Section 5 presents an alternative model allowing interactions between couples of variables. In Section 6 the problem of secure communications using chaotic masking is discussed and, finally, Section 7 gives some conclusions.

2 Functional Networks

Functional networks have been recently introduced as a powerful generalization of neural networks [10,11]. Figure 2(a) shows a typical architecture of a functional network illustrating its main components. Besides of the input and output layers (sets $\{x_1, x_2, x_3\}$ and $\{x_6\}$, respectively), a functional network consists of one or several layers of intermediate storing units (in Fig. 2(a) the only intermediate layer is $\{x_4, x_5\}$), which store information produced by neuron units, and one or several layers of neuron, or processing, units (the layers $\{f_1, f_2\}$ and $\{f_3\}$ in Fig. 2(a)). A neuron unit evaluates a set of input values, coming from the previous layer (of intermediate or input units) and delivers a set of output values to the next layer (of intermediate or output units). To this

end, each neuron has associated a neuron function which can be multivariate and can have as many arguments as inputs. This mathematical model of functional networks parallels printed circuit boards with electronic components, thus, giving an appealing interpretation to functional nets and an interesting and natural additional application (see Figure 2(b)).

An interesting characteristic of functional networks is that the output of several neuron units can coincide in an intermediate storing unit, indicating that they must be equal. Each of this intermediate storing units with multiple inputs represents a functional constraint in the model and leads to a system of functional equations (see [20,21] for an introduction to functional equations). By solving this system, the initial neuron functions can be simplified (for example, reducing the number of arguments) leading to an equivalent simpler functional network [11].

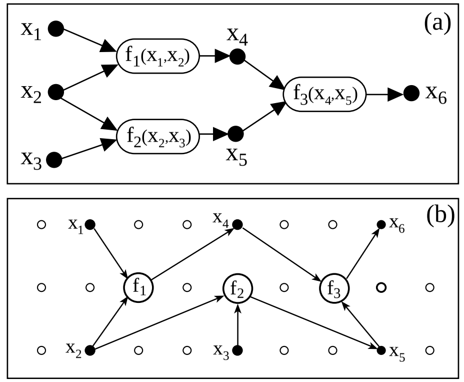

Fig. 2. (a) Functional network with three input, two intermediate, and one output storing units and three neuron, or processing, units; (b) equivalent parallel printed circuit board.

An important problem associated with functional networks is that of analyzing the uniqueness of representation, i.e., obtaining the more general sets of functions satisfying the functional constraints imposed by the network topology. We shall analyze these problems using the functional network architecture presented in the next section.

3 Separable Functional Networks

An interesting family of functional network architectures with many applications is the so called separable functional networks (see Fig. 3), which has associated a functional expression which combines the separate effects of input variables. For the case of two inputs, x and y, and one output, z, we have:

$$z = F(x,y) = \sum_{i=1}^{n} f_i(x) g_i(y). \quad (1)$$

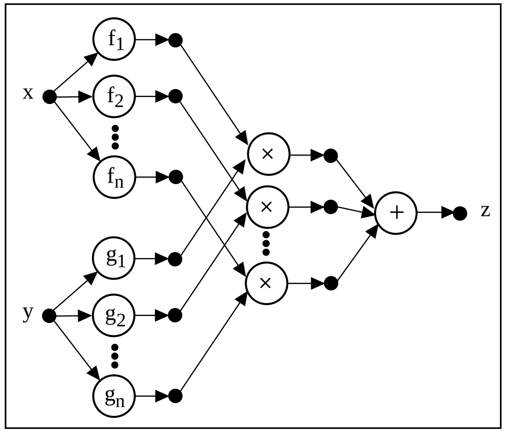

Fig. 3. Separable functional network architecture with two inputs and one output.

For illustrative purposes, we start by considering the simplest architecture from this family, which neglects double interactions by separating the contributions of each of the inputs in the form

$$z = F(x,y) = f(x) + g(y). \quad (2)$$

Note that this functional corresponds to (1) with $n = 2$ and $g_1 = f_2 = 1$.

In this case, the uniqueness of representation problem reduces to finding the relationships among the functions of two different representations of (2), say,

$$F(x,y) = f_1(x) + g_2(y) = f_1^*(x) + g_2^*(y). \quad (3)$$

The following theorem gives the general solution of this problem for all the functional network architectures of the form (1) (see [21]).

Theorem 1 *All solutions of equation $\sum_{i=1}^{n} f_i(x) g_i(y) = 0$ can be written in the form $f(x) = A\,\varphi(x)$, $g(y) = B\,\psi(y)$, where A and B are constant matrices (of dimensions $n \times r$ and $n \times n - r$, respectively) with $A^T B = 0$, and $\varphi(x) = (\varphi_1(x), ..., \varphi_r(x))$ and $\psi(y) = (\psi_{r+1}(y), ..., \psi_n(y))$ are two arbitrary systems of mutually linearly independent functions, and r is an integer between 0 and n.*

If the functions of one of the representations, say f_1 and g_2, are supposed to be known, then this theorem gives the structure of the functions associated with any other representation. Note that the general solution depends on arbitrary constants and, perhaps, on arbitrary functions. In the above case, from (3) we have the equation

$$f_1(x) + g_2(y) - f_1^*(x) - g_2^*(y) = (f_1(x) - f_1^*(x)) + (g_2(y) - g_2^*(y)) = 0.$$

Then, from Theorem 1 we obtain:

$$\begin{pmatrix} f_1(x) - f_1^*(x) \\ 1 \end{pmatrix} = \begin{pmatrix} c_1 \\ 1 \end{pmatrix} (1), \quad \begin{pmatrix} 1 \\ g_2(y) - g_2^*(y) \end{pmatrix} = \begin{pmatrix} 1 \\ c_2 \end{pmatrix} (1) \quad (4)$$

where

$$\begin{pmatrix} c_1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ c_2 \end{pmatrix} = 0 \quad \Leftrightarrow \quad c_1 = -c_2 = c. \quad (5)$$

Finally, from (4) and (5) we get the constraints:

$$f_1^*(x) = f_1(x) - c, \; g_2^*(y) = g_2(y) + c, \quad (6)$$

where c is an arbitrary constant. Therefore if we know a couple of functional neurons satisfying (2), any other couple of the form (6) will determine the same functional network (i.e., both give the same output for the same inputs). This is an important fact to be considered during the learning process, since some initial functional conditions have to be given in order to have a unique representation of the functional network. For instance, in the above example, it is necessary to give an initial value for one of the functions in order to eliminate the arbitrariness associated with constant c.

3.1 Learning: Estimating the Neuron Functions from Data

The problem of learning the functional network associated with (3) reduces to estimating the neuron functions f and g from the available data. This problem can be generally stated by considering a standard functional form for each neuron function, such as a mth-order polynomial, or a m-order Fourier series, and fitting the coefficients to the data set using least squares.

When the data is given in the form of a time series $\{x_k\}$ consisting of n points, we use an embedding of the time series in an appropriate delayed-coordinates space to train the functional network. Note that the deterministic structure of the underlying system can always be reproduced by choosing an appropriate dimension d for the delayed space (see [6]). Then, for the case $d = 2$ we use the training data consisting of triplets $\{(x_{0i}, x_{1i}, x_{2i}); i = 3, \ldots, n\}$, where each of the triplets is obtained from three consecutive terms of the time series: $x_{0i} = x_i$, $x_{1i} = x_{i-1}$, and $x_{2i} = x_{i-2}$.

Then we can approximate the functions f and g in (2) by considering a linear combination of known functions from a given family (in this paper we shall consider polynomials or Fourier expansions):

$$\hat{f}(x) = \sum_{j=1}^{m_1} a_{1j}\phi_{1j}(x), \quad \hat{g}(x) = \sum_{j=1}^{m_2} a_{2j}\phi_{2j}(x),$$

where the coefficients a_{kj} are the parameters of the functional network, i.e., they play the role of the weights on a neural network. Then, the error can be measured by

$$e_i = x_{0i} - \hat{f}(x_{1i}) - \hat{g}(x_{2i}); \; i = 1, \ldots, n. \tag{7}$$

Thus, to find the optimum coefficients we minimize the sum of square errors

$$Q = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(x_{0i} - \sum_{k=1}^{2} \sum_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki}) \right)^2 . \tag{8}$$

As we have shown before, an initial functional condition has to be given in order to have a unique representation. In this case we consider the initial functional condition

$$\hat{f}(u_0) \equiv \sum_{j=1}^{m_1} a_{1j}\phi_{1j}(u_0) = v_0, \tag{9}$$

where u_0 and v_0 are given constants.

Using the Lagrange multipliers we define the auxiliary function

$$Q_\lambda = \sum_{i=1}^{n} \left(x_{0i} - \sum_{k=1}^{2} \sum_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki}) \right)^2 + \lambda \left(\sum_{j=1}^{m_1} a_{1j}\phi_{1j}(u_0) - v_0 \right).$$

Then, the minimum can be obtained by solving the following system of linear

equations, where the unknowns are the coefficients a_{kj} and the multiplier λ:

$$\begin{cases} \dfrac{\partial Q_\lambda}{\partial a_{1r}} = -2\sum_{i=1}^{n} \left(x_{0i} - \sum_{k=1}^{2}\sum_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki}) \right) \phi_{1r}(x_{1i}) + \lambda\phi_{1r}(u_0) = 0, \\ \qquad\qquad r = 1, \ldots, m_1, \\ \dfrac{\partial Q_\lambda}{\partial a_{2r}} = -2\sum_{i=1}^{n} \left(x_{0i} - \sum_{k=1}^{2}\sum_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki}) \right) \phi_{2r}(x_{2i}) = 0, r = 1, \ldots, m_2, \\ \dfrac{\partial Q_\lambda}{\partial \lambda} = \sum_{j=1}^{m_1} a_{1j}\phi_{1j}(u_0) - v_0 = 0. \end{cases} \tag{10}$$

4 Inferring Nonlinear Models From Time Series

In this section we describe some applications of the above functional network architecture to infer the models associated with some chaotic time series. As a first example, we consider the Hénon map, one of the most popular and simplest dynamical system defined as [22]:

$$x_n = 1 + y_{n-1} - ax_{n-1}^2, y_n = 0.3x_{n-1}, \tag{11}$$

or, using delayed coordinates, as

$$x_n = 1 - ax_{n-1}^2 + 0.3x_{n-2}. \tag{12}$$

It can be shown that, depending on the values of the parameter a, the system exhibits a very rich behavior, including the apparison of deterministic chaos (see [23] for a complete analysis of the dynamics of this system). In particular we consider a chaotic time series consisting of 300 points generated from the initial conditions $x_0 = 0.5, x_1 = 0.5$ for the parameter value $a = 1.4$ (see Figure 4(a)).

Despite of the seemingly stochastic dynamics of the time series, the first time delay embedding in Figure 4(b) shows the deterministic quadratic relationship between the variables x_n and x_{n-1}. However, it is not straightforward obtaining the optimal embedding dimension from the time series. Then, we can use a generalized functional network architecture based in (2) with as many inputs as delayed coordinates to infer the structure underlying the chaotic time series (see Figure 5). Then, the learning algorithm described in Sec. 3.1 can automatically calculate the optimal embedding space and the resulting approximation of the system (note that the generalization of the learning algorithm for more than two inputs is a trivial task).

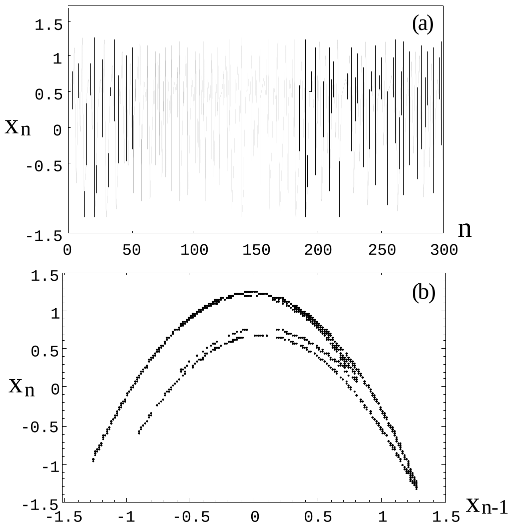

Fig. 4. (a) Time series of a chaotic orbit of the Hénon map. (b) Phase space of the first embedding of the system showing the quadratic relationship between the variables.

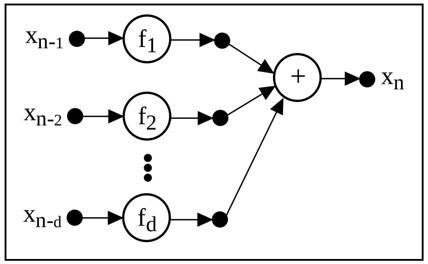

Fig. 5. Separable functional network for a d-dimensional delayed space.

As a first example, we choose a polynomial functional family, say $\phi = \{1, x, x^2\}$, for all the neuron functions and use the learning algorithm considering the functional condition $f(1) = 0$ and starting with $d = 1$. We obtained the root mean square (RMS) error 0.695 (for $d = 1$) and 0 (for $d > 1$). Then the learning algorithm obtained a poor approximation of the data using only one delayed coordinate x_{n-1} and the exact Hénon map given in (12) using two, or more, delayed variables $x_{n-1}, x_{n-2}, \ldots$. In the case of using two variables we get the model:

$$f(x_{n-1}) = 1.4 - 1.4\,x_{n-1}^2;\; g(x_{n-2}) = -0.4 + 0.3\,x_{n-2} \tag{13}$$

which gives the exact Hénon map $F(x_{n-1}, x_{n-2}) = f(x_{n-1}) + g(x_{n-2}) = 1 - 1.4\,x_{n-1}^2 + 0.3\,x_{n-2}$. Note that any other initial functional condition will lead to the same final model, but a different representation of the neuron functions would be obtained. For example, for $f_1(1) = 1$ we get:

$$f(x_{n-1}) = 2.4 - 1.4\,x_{n-1}^2,\; g(x_{n-2}) = -1.4 + 0.3\,x_{n-2}$$

which also gives the exact Hénon map when adding the obtained functions.

In the previous example we have obtained the exact model because we have used some information about the structure of the underlying model, i.e., the most convenient family of functions for the polynomial dynamics of the time series. However, if we use a different family for the neuron functions, such as a Fourier expansion given by $\phi = \{sin(x), cos(x), \ldots, sin(m\,x), cos(m\,x)\}$, then we obtain an approximate model.

Proceeding as in the previous example, the RMSE obtained when considering $m = 6$ for $d = 1$, 2, 3 and 4 are 0.16, $4.5\,10^{-5}$, $8.5\,10^{-6}$ and $5.5\,10^{-6}$, respectively. Note the resulting substantial reduction of the error when the number of delayed variables increases from one to two. However, the error decreases slowly when adding more delayed coordinates. This indicates that a good choice for the embedding space is $d = 2$ (the actual embedding of the system). Table 1 illustrates the performance of the approximate model showing the RMSE for different values of m in this case. Note that the errors obtained for both the training and a test data (the next 1000 points of the orbit) are very similar, indicating that no overfitting is produced during the learning process.

Note that the size of the available data is not important for these models, since once the sample size has reached a threshold value, the functional network has already captured the underlying deterministic dynamics and the improvement associated with an increase of the sample size is very small. In particular, if we consider orbits with $300, 1000$ and 3000 points of the Hénon map, given in (12), and use a Fourier model with $m = 6$ we obtain the RMSEs $4.5 \times 10^{-5}, 4.8 \times 10^{-5}$ and 4.65×10^{-5}, respectively. This shows that the errors stabilize after reaching a threshold sample size.

With the aim of illustrating the performance of the above functional network in different situations, Table 1 also shows the results obtained for two additional chaotic time series corresponding to the Holmes and Lozi maps.

The Holmes map is a cubic 2D map that can be written as [24]:

$$x_n = 2.76\,x_{n-1} - x_{n-1}^3 - 0.2\,x_{n-2}, \tag{14}$$

Network	Par.	RMSE Training Data			RMSE Test Data		
		Henon	Holmes	Lozi	Henon	Holmes	Lozi
m=4	16	0.0058	0.0099	0.038	0.0064	0.026	0.039
m=5	20	$7.8\ 10^{-4}$	0.0023	0.028	$7.9\ 10^{-4}$	0.0061	0.028
m=6	24	$4.5\ 10^{-5}$	$3.8\ 10^{-4}$	0.021	$9.3\ 10^{-5}$	$9.9\ 10^{-4}$	0.025
m=7	28	$6.7\ 10^{-6}$	$8.3\ 10^{-5}$	0.016	$2.2\ 10^{-5}$	$4.1\ 10^{-4}$	0.02

Table 1
Performance of several Fourier functional networks for the Hénon, Holmes and Lozi time series. The number of parameters and the RMS errors obtained in each case are shown.

whereas the Lozi map involves non-differentiable functions which make difficult the modeling of the associated time series. This map is given by [25]:

$$x_n = 1 - 1.7\,|x_{n-1}| + 0.5\,x_{n-2}. \tag{15}$$

The time series and the associated first embedding spaces for the Holmes and Lozi maps are shown in Figures 6 and 7, respectively.

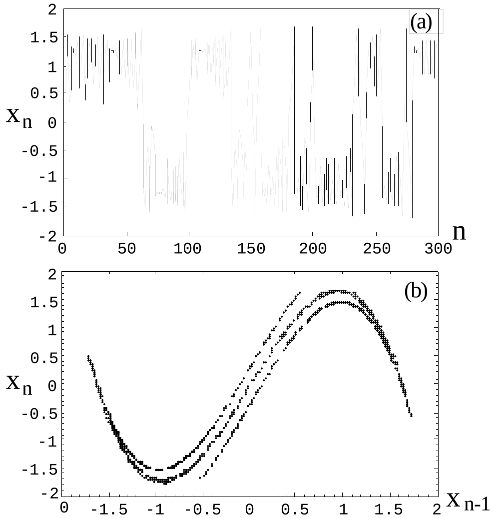

Fig. 6. (a) Time series of a chaotic orbit of the Holmes map. (b) Phase space of the first embedding of the system showing the cubic relationship between the variables. The initial conditions are $x_0 = 0.1$ and $x_1 = 0.3$.

From the results shown in Table 1 we can see that both the Hénon and Holmes models can be accurately described with an approximated functional network.

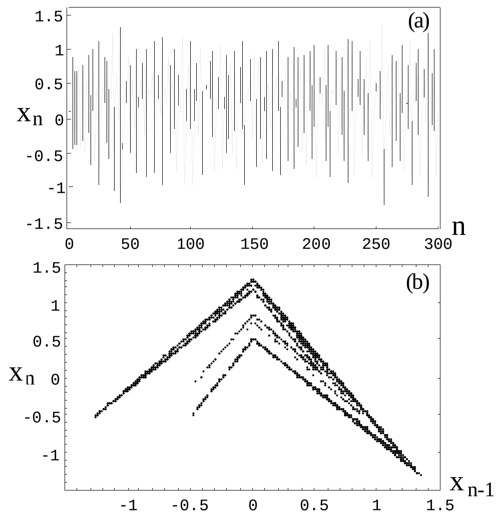

Fig. 7. (a) Time series of a chaotic orbit of the Lozi map. (b) Phase space of the first embedding of the system showing the non-differentiable relationship between the variables. The initial conditions are $x_0 = 0.5$ and $x_1 = 0.7$.

In Section 6 we shall see that these approximated models can be used to extract information masked in the time series. On the other hand the approximation achieved for the Lozi map is poorer than in the previous cases. Figure 8 shows the Lozi time series and the error obtained when using the functional network obtained with a Fourier functions family with $m = 7$ terms. In this situation, an improved approximate model has to be used in order to unmask the information that may be encoded in the time series.

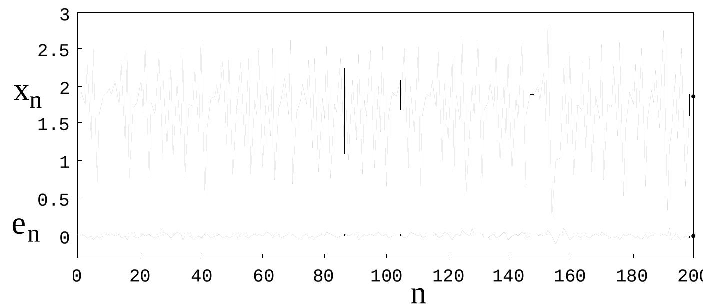

Fig. 8. Time series $1.5{+}x_n$ and errors e_n obtained for the Lozi map when considering $m = 7$.

5 A Double Interaction Neural Network

In this section we extend the above functional network architecture to consider interactions between couples of input variables. We consider the model given in Figure 9 (the inputs will be associated with time-delayed coordinates of the time series). Note that this is also a network of the family (1) and defines the following model:

$$F(x,y) = f_1(x) + g_2(y) + f_3(x)g_3(y). \quad (16)$$

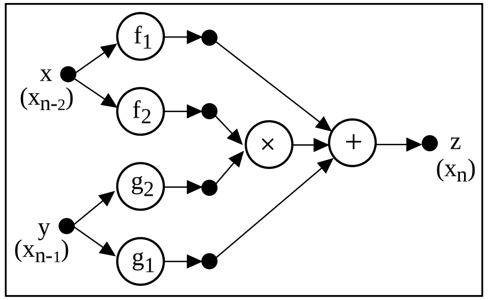

Fig. 9. Functional network with two inputs and one output.

In this case the uniqueness of the representation problem reduces to finding the relationships among the functions of two different representations:

$$\begin{aligned} &f_1(x) + g_2(y) + f_3(x)g_3(y) - f_1^*(x) - g_2^*(y) - f_3^*(x)g_3^*(y) = \\ &(f_1(x) - f_1^*(x)) + (g_2(y) - g_2^*(y)) + f_3(x)g_3(y) - f_3^*(x)g_3^*(y) = 0 \end{aligned} \quad (17)$$

Theorem 1 also gives the general solution of this problem. We have:

$$\begin{aligned} &\begin{pmatrix} f_1(x) - f_1^*(x) \\ 1 \\ f_3(x) \\ f_3^*(x) \end{pmatrix} = \begin{pmatrix} c_1 & c_2 \\ 1 & 0 \\ 0 & 1 \\ c_3 & c_4 \end{pmatrix} \begin{pmatrix} 1 \\ f_3(x) \end{pmatrix}, \\ &\begin{pmatrix} 1 \\ g_2(y) - g_2^*(y) \\ g_3(y) \\ -g_3^*(y) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ c_5 & c_6 \\ c_7 & c_8 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ g_3^*(y) \end{pmatrix} \end{aligned} \quad (18)$$

In this case, solving the system $A^T B = 0$ we obtain $c_5 = -c_1$, $c_6 = c_3$, $c_7 = -c_2$, and $c_8 = c_4$. Therefore, we finally get the following relationships

between both representations:

$$\begin{cases} f_1^*(x) = f_1(x) - c_2\, f_3(x) - c_1 \\ f_3^*(x) = c_4\, f_3(x) + c_3 \\ g_2^*(x) = g_2(y) - \dfrac{c_3}{c_4}\, g_3(y) - \dfrac{c_2\, c_3}{c_4} + c_1 \\ g_3^*(x) = \dfrac{1}{c_4} g_3(y) + \dfrac{c_2}{c_4}, \end{cases} \quad (19)$$

where c_1, c_2, c_3, and c_4 are arbitrary constants. Therefore, four functional conditions have to be given in order to define a unique representation for this functional network.

The problem of learning the neuron functions from data now involves estimating the functions f_1, f_3, g_2 and g_3 in (16) by considering linear combinations of known functions from a given family:

$$\begin{aligned} \hat{f}_1(x) &= \sum_{j=1}^{m_1} a_{1j}\phi_{1j}(x); \quad \hat{g}_2(y) = \sum_{j=1}^{m_2} a_{2j}\phi_{2j}(y), \\ \hat{f}_3(x) &= \sum_{j=1}^{m_3} a_{3j}\phi_{3j}(x); \quad \hat{g}_3(y) = \sum_{j=1}^{m_4} a_{4j}\phi_{4j}(y), \end{aligned} \quad (20)$$

where the coefficients a_{kj} are the parameters of the functional network. In this case the sum of square errors becomes a nonlinear function on the coefficients a_{kj}:

$$Q = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(x_{0i} - \sum_{k=1}^{2} \sum_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki}) - \sum_{j=1}^{m_3} \sum_{l=1}^{m_4} a_{3j}a_{4l}\phi_{3j}(x_{1i})\phi_{4l}(x_{2i}) \right)^2 \quad (21)$$

Then, we can use any standard minimization method, such as the Powell's multidimensional method (see Press et al. [26]) to obtain the optimal coefficients. A simplification of this model which leads to a linear system of equations can be obtained by considering one of the functions $f_3(x)$ or $g_3(y)$, say $g_3(y)$, in (16) to be known. In this case (19) becomes:

$$\begin{cases} f_1^*(x) = f_1(x) - c_1 \\ f_3^*(x) = f_3(x) + c_3 \\ g_2^*(x) = g_2(y) - c_3\, g_3(y) + c_1 \\ g_3^*(x) = g_3(y), \end{cases} \quad (22)$$

which implies that we only need to specify two functional conditions to have a unique representation of the model. Similarly the sum of square errors (21) becomes the linear function

$$Q = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(x_{0i} - \sum_{k=1}^{2}\sum_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki}) - \sum_{j=1}^{m_3} a_{3j}\phi_{3j}(x_{1i})g_3(x_{2i}) \right)^2 . \quad (23)$$

If we consider the initial functional conditions

$$\hat{f}_1(u_1) \equiv \sum_{j=1}^{m_1} a_{1j}\phi_{1j}(u_1) = v_1; \;\; \hat{g}_2(u_2) \equiv \sum_{j=1}^{m_2} a_{2j}\phi_{2j}(u_2) = v_2, \quad (24)$$

where u_1, v_1, u_2 and v_2 are given constants, then the Lagrange multipliers technique leads to the auxiliary function

$$\begin{aligned} Q_\lambda = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \left(x_{0i} - \sum_{k=1}^{2}\sum_{j=1}^{m_k} a_{kj}\phi_{kj}(x_{ki}) - \sum_{j=1}^{m_3} a_{3j}\phi_{3j}(x_{1i})g_3(x_{2i}) \right)^2 \\ - \sum_{k=1}^{2} \lambda_k \sum_{j=1}^{m_k} (a_{kj}\phi_{kj}(u_k) - v_k). \end{aligned} \quad (25)$$

Then, the minimum can be obtained by solving the following system of linear equations, where the unknowns are the coefficients a_{kj} and the multipliers λ_1 and λ_2:

$$\begin{cases} \dfrac{\partial Q_\lambda}{\partial a_{sr}} = -2\sum_{i=1}^{n} e_i \phi_{sr}(x_{si}) + \lambda_s \phi_{sr}(u_s) = 0, s = 1,2, \; r = 1,\ldots,m_s, \\ \dfrac{\partial Q_\lambda}{\partial a_{3r}} = -2\sum_{i=1}^{n} e_i \phi_{3r}(x_{1i}) g_3(x_{2i}) = 0, r = 1,\ldots,m_3, \\ \dfrac{\partial Q_\lambda}{\partial \lambda_k} = \sum_{j=1}^{m_k} a_{kj}\phi_{kj}(u_k) - v_k = 0; \;\; k = 1,2, \end{cases} \quad (26)$$

Then, solving this linear system we get the optimal neuron functions for a given problem.

As an illustrative example, consider the Burgers map given by [27]:

$$x_n = (1.8 - x_{n-2}^2)x_{n-1}, \quad (27)$$

which includes double interactions between the variables. Figure 10(a) shows a time series consisting of 500 points obtained for the initial conditions $x_0 = 0.1$ and $x_1 = 0.3$.

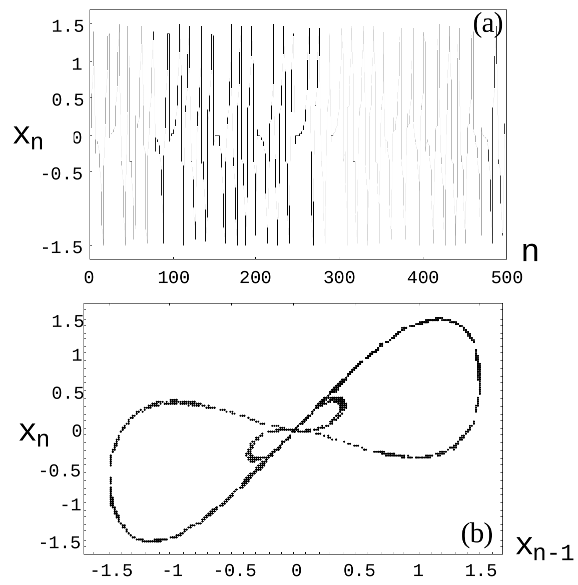

Fig. 10. (a) Time series of a chaotic orbit of the Burgers map. (b) Phase space of the first embedding.

Then, we can use the functional network in (16) to infer a model for the time series. For example, if we consider the family $\phi = \{1, x, x^2\}$ for the functions f_1, f_3, and g_2 and choose the function g_3 to be $g_3(x_{n-1}) = x_{n-1}$, then we obtain

$$f_1(x_{n-2}) = 1, \;\; g_2(x_{n-1}) = -1 + 3x_{n-1}, \;\; f_3(x_{n-2}) = -1.2 - x_{n-2}^2,$$

which gives the exact Burgers model in (27). Note that if we consider the nonlinear estimation model with the base family $\phi = \{1, x, x^2\}$ for all the neuron functions and minimize the error function in (21), we also get the exact Burgers map after a few iterations.

On the other hand, if we consider a Fourier expansion of the neuron functions of the form

$$\phi = \{sin(x), \ldots, sin(7x), cos(x), \ldots, cos(7x)\}$$

and choose the function g_3 to be $g_3(y) = sin(x) + \ldots + sin(7x) + cos(x) + \ldots + cos(7x)$, then we get an approximate model with RMSE 0.00029. In this case, a similar error, 0.00018, is obtained with the nonlinear estimation method. This shows that we can easily get an accurate approximate model for a given time series using the above double interaction functional network.

6 Extracting Information Masked by Chaos

In this section we present an application of the above functional network models to extract information masked in chaotic time series. We consider the situation in which we have a transmitter-receiver couple of chaotic systems able to synchronize (see Fig. 1). For example, if we consider two identical replicas of the Hénon map given in (12) starting from different initial conditions, then it is possible to synchronize both systems by injecting the x variable of the transmitter into the receiver system. It has also been shown that synchronization is robust when some noise (or information) is added to the driving signal. Therefore, the message can be added (at small magnitude) to the chaotic orbit without spoiling synchronization (a detailed study of the influence of signal and message powers on the masking process is given in [28]). For example, consider the binary message shown in Figure 11, where each bit is transcribed as 20 consecutive values of the time series m_n. The value -1 is used to denote the bit 0 and the value 1 is used for the bit 1. Note that the series m_n is scaled a factor 10^{-3} before it is added to the chaotic orbit obtained from the Hénon map.

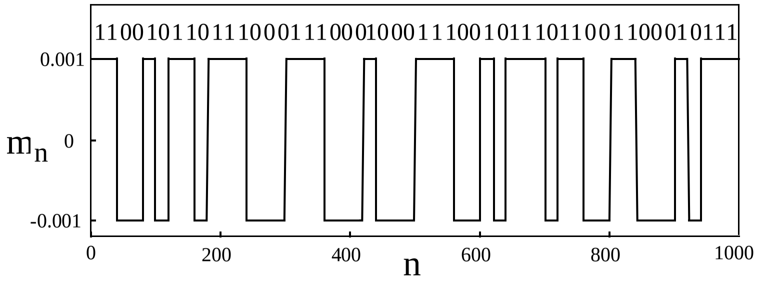

Fig. 11. Binary message "110010110111..." where each bit is represented by 20 consecutive sequence points.

Figure 12(b) shows the transmitted chaotic signal encoding the message. Then, we can use the exact or one of the approximated functional network models obtained in Sec. 4 to reproduce the deterministic component of the original chaotic system from the transmitted data. For example, we can use a polynomial family of functions and train the functional network in (13) using the transmitted signal. In this case, the functional network reproduces the exact Hénon map and the error obtained when comparing the actual and predicted values for each point of the time series gives the value of the encoded bit. Figure 12(c) shows the error obtained when using the functional network. Then, after a low-pass filtering (see Fig. 12(d)) we can recover the original message as shown in Figure 12(e).

Similar results can also be obtained when using an approximate functional network to learn the time series. For example, a functional network with Fourier approximation neuron functions with $m = 7$ (see Table 1) can also unmask the encoded message, since the error of the approximation is lower than 10^{-3} (the value of the message). The message reconstructed in this case can be seen in Figure 13. This is a surprising result, since it implies that no knowledge about the system is needed to use an automatic functional network for obtaining a good approximation of the time series and, therefore, discover the underlying message.

We have obtained similar results when applying this method to the other models presented in this paper. Therefore, functional networks appear to be promising mechanisms to replace the receiver system in secure communications based on chaos synchronization (see Figure 1) and, therefore, to extract the masked information.

7 Conclusions and Further Remarks

In this paper we have presented functional networks as promising tools for modeling and predicting nonlinear time series. One of the main properties of these models is that they can be learned from data by just solving a linear system of equations. Thus, the problems associated with multiple optimal values are avoided in these models. In general, several models with a reasonable number of degrees of freedom (small number compared to the sample size) must be used for a given time series. The selection of the adequate model can be based on the Mallows' C_p statistic (see Mallows [29]), or the minimum description length principle (see Rissanen, [30,31]). These methods penalize both the number of parameters and the errors, leading to a compromise model. This problem will be the scope of a future paper. We have also shown that functional networks can be used to extract the information masked in the time series. The results presented in the paper are promising but they will be extended and analyzed in more detail in a future paper.

References

[1] D.R. Brillinger, Time series. Data analysis and theory (McGraw-Hill, 1981). New York).

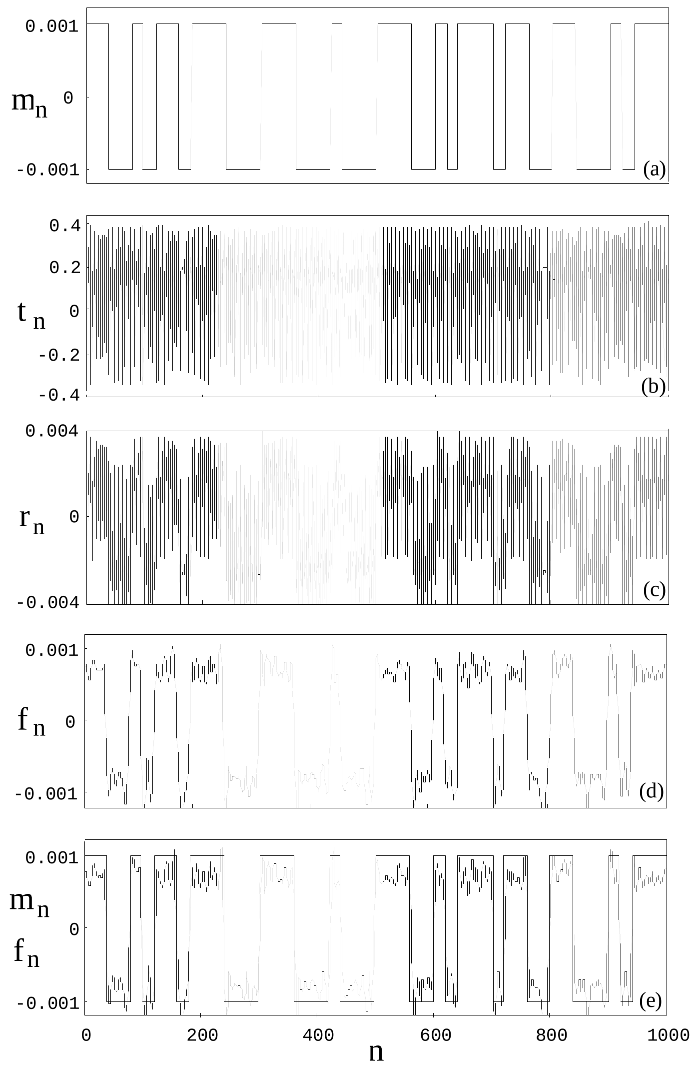

Fig. 12. Unmasking a digital signal transmission: (a) the original binary m_n-message where each bit is represented by 20 points of the series m_n; (b) actual chaotic transmitted signal (message + chaotic signals); (c) reconstructed message obtained from the exact polynomial functional network in (1); (d) low-pass filtered signal of r_n. Finally, (e) shows both the original and recovered messages.

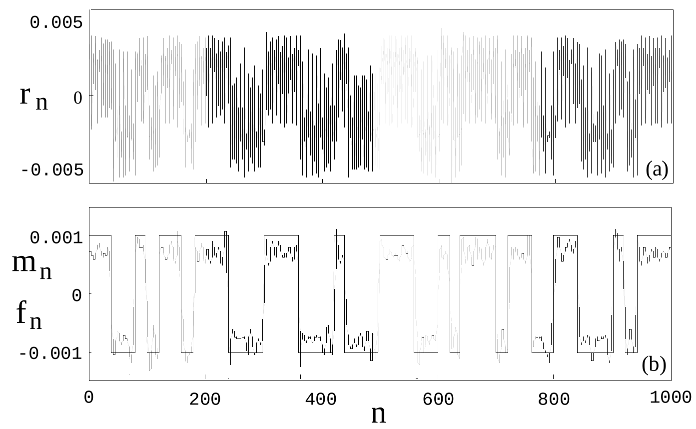

Fig. 13. Unmasking a digital signal transmission: (a) reconstructed message obtained from the Fourier approximate functional network with $m = 7$. (b) Low-pass filtered received and original messages.

[2] S.M. Pandit and S.M. Yu, Time series and system analysis with applications, (Wiley, New York, 1983).

[3] J.A. Attali and G. Pages, Neural Networks 10(6) (1997) 1069.

[4] H.S. Stern, Technometrics 38(3) (1996) 205.

[5] N.H. Packard,J.P. Crutchfield,J.D. Farmer, and R.S. Shaw, Phys. Rev. Lett. 45 (1980) 712.

[6] F. Takens, in Dynamical Systems and Turbulence, eds. D.A. Rand and L.S. Young (Springer Verlag, Berlin, 1980).

[7] Y. Meyer, and R.D. Ryan, Wavelets. Algorithms and Applications (SIAM, Philadephia, 1991).

[8] Y.A. Kravtsov, and J.B. Kadtke, editors, Predictability of Complex Dynamical Systems (Springer, 1997).

[9] J.D. Farmer and J.J. Sidorowich, Phys. Rev. Lett. 59 (1987) 845.

[10] E. Castillo, A. Cobo, and R. Gómez-Nesterkín, Technical Report, Dept. of Applied Mathematics, University of Cantabria (1997) 97-1.

[11] E. Castillo, Functional Networks, Neural Proc. Lett. in press.

[12] E. Castillo and J. M. Gutiérrez, in Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing (Cancún, 1998), to appear.

[13] T. Carroll and L. M. Pecora, Nonlinear Dynamics in Circuits (World Scientific, 1997).

[14] K. Murali and M. Lakshmanan, Phys. Rev. A. 48 (1993) 1624.

[15] K.M. Cuomo and A.V. Oppenheim, Phys. Rev. Lett. 71 (1993) 65.

[16] J.M. Gutiérrez and A. Iglesias, Phys. Lett. A 239 (1998) 174.

[17] L. M. Pecora, editor, Chaos in Communications (SPIE Proceedings Vol. 2038, 1993).

[18] Pérez G, and Cerdeira, H.A., Phys. Rev. Lett. 74:11 (1995) 1970.

[19] Zhou, C-S. and Chen, T-L., Phys. Lett. A. 234 (1997) 429.

[20] J. Aczél, Lectures on Functional Equations and Their Applications (Academic Press, 1966).

[21] E. Castillo and R. Ruiz-Cobo, Functional Equations in Science and Engineering (Marcel Dekker, 1992).

[22] M. Hénon, Comm. Math. Phys. 50 (1976) 69.

[23] C. Simò, J. of Stat. Phys. 21 (1979) 465.

[24] P.J. Holmes, Philos. T. Roy. Soc. 292 (1979) 419.

[25] R. Lozi, J. Phys. (Paris) C5 39 (1978) 9.

[26] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, Numerical Recipes (2nd edition) (Cambridge U.P., 1992).

[27] R.R. Whitehead and N. MacDonald, Physica D 13 (1984) 401.

[28] K. M. Short, Int. J. Bifurcation and Chaos, 4 (1994) 959.

[29] C. L. Mallows , Technometrics 15 (1973) 661.

[30] J. Rissanen, The Annals of Statistics 11 (1983) 416.

[31] J. Rissanen, Stochastic Complexity in Statistical Inquiry (World Scientific, 1989).