

## Predicción de series de tiempo aplicando redes neuronales artificiales

### Time series prediction using artificial neural networks

Miguel Angel Pérez-Chavarría <sup>1\*</sup>

Hugo Homero Hidalgo-Silva <sup>2</sup>

Francisco J. Ocampo-Torres <sup>3</sup>

<sup>1</sup> Instituto de Investigaciones Oceanológicas  
Universidad Autónoma de Baja California  
Apartado Postal 453, Ensenada, CP 22860, Baja California, México  
\* E-mail: mperez@faro.ens.uabc.mx

<sup>2</sup> Departamento de Ciencias de la Computación  
Centro de Investigación Científica y de Educación Superior de Ensenada  
Km. 107 carretera Tijuana-Ensenada  
Ensenada, CP 22860, Baja California, México

<sup>3</sup> División de Oceanología  
Centro de Investigación Científica y de Educación Superior de Ensenada  
Km. 107 carretera Tijuana-Ensenada  
Ensenada, CP 22860, Baja California, México

*Recibido en septiembre de 1999; aceptado en diciembre de 2001*

#### Resumen

En este trabajo se presentan algoritmos basados en redes neuronales artificiales (RNA) para la predicción de series temporales de las variables oceanográficas Índice de Oscilación del Sur (IOS) y anomalías de temperatura superficial del mar (ATSM). Se realizaron experimentos de predicción utilizando la arquitectura de redes neuronales conocida como redes con respuesta finita al impulso (RNRFI). Se hicieron experimentos variando los diferentes parámetros de la RNRFI, para determinar aquéllos que permitieran un mejor comportamiento. Se encontró que para series temporales del IOS y las ATSMs los mejores resultados se presentan al conformar una RNRFI con una neurona de entrada, filtros de orden 10 en la capa de entrada, dos capas posteriores de 8 neuronas con filtros de orden 5 para cada una de ellas, y una neurona de salida. Todas las redes fueron entrenadas con el algoritmo de aprendizaje retropropagación temporal, usando la sigmoide como función de activación en las capas ocultas y salida lineal. La razón de aprendizaje usada fue de 0.001.

En la mayoría de los experimentos realizados se obtuvo un error cuadrático medio normalizado de  $0.4 \pm 0.1$  y un coeficiente de correlación mayor que 0.8 entre la serie original y la predicha. Para el caso de las ATSMs, se observó que las RNA tuvieron un mejor comportamiento que otros métodos de predicción, considerando predicciones a corto plazo.

*Palabras clave:* predicción, serie-temporal, red-neuronal

#### Abstract

In this work, artificial neural network (ANN) algorithms were used to predict time series of the oceanographic variables Southern Oscillation Index (SOI) and sea surface temperature anomaly (SSTA). The finite impulse response neural network (FIRNN) was applied to data obtained from the NOAA. In order to determine the most efficient FIRNN architecture, several experiments were made varying different parameters. The best predictions were obtained for a network with one input neuron and 10th-order filters in the input layer, two 8-neuron 5th-order filter hidden layers and one output neuron. All the networks were trained with the temporal backpropagation learning algorithm, using the sigmoid transfer function at the hidden layers and a linear output. The learning rate was 0.001.

In most experiments a normalized mean square error of  $0.4 \pm 0.1$  and a correlation coefficient between the original and the predicted series greater than 0.8 were found. From a comparison with other SSTA prediction methods, the results obtained with the neural network were the best ones for the short term forecasting case.

*Keywords:* forecasting, time series, neural-network

## Introducción

Existen dos parámetros oceanográficos cuya predicción ha tomado gran importancia en la última década; el Índice de Oscilación del Sur y las anomalías en la temperatura superficial del mar. El primero representa la diferencia de presión al nivel del mar entre Darwin (Australia) y Tahití; y el segundo, la temperatura promedio en ciertas regiones del Pacífico tropical. Estos parámetros nos permiten estimar de alguna manera la existencia del fenómeno de El Niño Oscilación Sur (ENOS), que se presenta cuando una alta presión superficial sobre el poniente y una baja presión sobre el sureste del Pacífico tropical, coinciden con lluvia abundante, temperaturas superficiales tibias poco usuales y vientos relajados en el centro y oriente del Pacífico tropical (Philander, 1990). Dado que esta condición atmosférica se correlaciona fuertemente con la lluvia en diferentes lugares del mundo, con variaciones en la temperatura global que pueden afectar la pesca y la agricultura, es importante determinar dicha condición, pues esto permitiría prevenir desastres globales.

El objetivo de este trabajo consiste en aplicar las redes neuronales artificiales a la predicción del Índice de Oscilación del Sur (IOS) y de las anomalías de temperatura superficial del mar (ATSM).

## Redes neuronales artificiales

Una red neuronal artificial (RNA) es un sistema de procesamiento de información distribuida en paralelo que consta de elementos de procesamiento simples y adaptables, interconectados entre sí. La RNA funciona como un modelo estadístico que realiza una transformación entrada/salida mediante el ajuste de un conjunto de parámetros llamados pesos (Bishop, 1997).

El tipo de red neuronal más ampliamente usado es el perceptrón multicapa. Se puede considerar que esta clase de red modela las distribuciones condicionales de un vector de salida  $Y$ , dados los diferentes valores posibles de un vector de entrada  $X$ . La distribución de  $X$  no se modela, por esto, estas redes sólo son adecuadas para problemas de regresión y/o clasificación. Los perceptrones multicapa se han aplicado a una gran variedad de problemas. Los más típicos consisten en tomar información sensorial de algún tipo, en la entrada, y predecir alguna característica de lo que se está midiendo (Bishop, 1997).

Entre las aplicaciones más comunes de las RNA están: el reconocimiento de patrones, regresiones no lineales, procesamiento de señales, modelos financieros, aproximación de funciones, control, predicción de series de tiempo, etc. Una de las ventajas más relevantes que presentan las RNA sobre los modelos estadísticos tradicionales radica en que el tipo de restricciones que imponen las redes son menos rigurosas.

Esquemáticamente, las RNA se pueden representar por un grafo dirigido como se muestra en la figura 1. Los nodos representan la función de transferencia de una neurona y los arcos,

## Introduction

There are two oceanographic parameters of which prediction has accounted a great importance in the last decade, the Southern Oscillation Index and sea surface temperature anomalies. The first one represents the difference of pressure at sea level between Darwin (Australia) and Tahiti, and the second one, the average temperature in certain regions of the Tropical Pacific. These parameters allow us to estimate, in a way, the existence of the El Niño Southern Oscillation (ENSO), that is present when a surficial high pressure over the west and a low pressure over the southeast of the tropical Pacific, coincide with abundant rainfall, unusually warm surface temperatures and relaxed winds in the central and the east tropical Pacific (Philander, 1990). Given that this atmospheric condition is strongly correlated to rainfall in different places of the world and to global temperature variations that can affect fisheries as well as agriculture, its determination is important in order to allow preventing global disasters.

The aim of this work consists on applying the artificial neural networks in the prediction of the Southern Oscillation Index (SOI) and of sea surface temperature anomalies (SSTA).

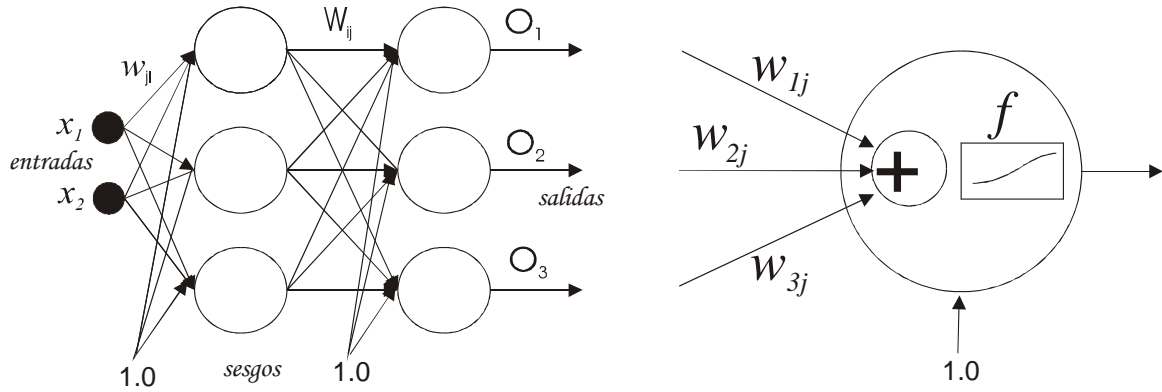
## Artificial neural networks

An artificial neural network (ANN) is a processing system of information distributed in parallel that consists of simple and adaptable processing elements, interconnected among them. The ANN works as a statistical model performing an input/output transformation by adjusting a set of parameters called weights (Bishop, 1997).

The most widely used kind of neural network is the multilayer perceptron. It can be considered that this kind of network models the conditional distributions of an output vector  $Y$ , given the different possible values of an input vector  $X$ . The marginal distribution of  $X$  is not modeled, therefore, these networks are only adequate for regression problems and/or classification. Multilayer perceptrons have been applied to a wide variety of problems. The most typical take some sensory information of input and predict some features of what is being sensed (Bishop, 1997).

Among the most common applications of the ANNs are: the pattern recognition, non-linear regressions, signal processing, financial models, function approximation, control, prediction of time series, etc. One of the most relevant advantages exhibited by the ANN over traditional statistical models, is that the constraints imposed by the networks are more flexible.

Graphically, the ANNs can be represented by a graph directed as shown in figure 1. The nodes represent the transfer function of a neuron and the arcs the weights of the connections among them. In the feedforward ANN, that will be described next, the connections are always forward, this is,



**Figura 1.** La figura de la izquierda muestra el esquema de una red neuronal artificial. Las  $x$  son las señales de entrada, las  $w_{ij}$  son las ponderaciones de los arcos entre las entradas y las unidades de la capa oculta, las  $W_{ij}$  son las ponderaciones entre las unidades de la capa oculta y las unidades de salida y las  $O$  son las salidas. La figura de la derecha muestra esquemáticamente la función de transferencia de una neurona artificial.

**Figure 1.** The figure to the left shows the scheme of an artificial neural network.  $x$  are the input signals,  $w_{ij}$  are the ponderates of the arcs among the inputs and the units of the hidden layer,  $W_{ij}$  are the ponderates between the units of the hidden layer and the output units and  $O$  are the outputs. The figure to the right shows schematically, the transference function of an artificial neuron.

los pesos de las conexiones entre ellas. En las RNA tipo avance, que son las que se describirán enseguida, las conexiones son siempre hacia adelante, es decir, no existen lazos en su arquitectura. Las unidades se ordenan para formar capas, de tal forma que las conexiones van de una capa hacia la siguiente, como se observa en la figura 1. A las unidades de las capas intermedias se les denomina “ocultas”, ya que no poseen conexiones directas hacia las entradas o las salidas. Cada unidad oculta, suma sus entradas y agrega un término constante (sesgo) para formar la entrada total  $h_j = \alpha_j + \sum_l w_{jl}x_l$ , a la que se le aplica una función  $f$ , para generar una salida  $V_j = f(h_j)$ . Así pues, la unidad de salida recibe  $h_i = \alpha_i + \sum_j W_{ij}V_j$  y produce la salida final  $O_i = f(h_i)$  (Hertz y Palmer, 1991).

Observe que el índice  $l$  se refiere a las entradas  $x$ ,  $j$  a las neuronas en la capa oculta e  $i$  a las unidades en la salida. Por lo tanto, una red como la de la figura 1 representa la función

$$O_i = f(h_i)$$

con:

$$h_j = \alpha_j + \sum_l w_{jl}x_l$$

y

$$h_j = \alpha_j + \sum_l w_{jl}x_l \quad (1)$$

there are no loops in their architecture. Units are ordered to form layers in such a way that connections always go from one layer to the next, as observed in figure 1. Units in the intermediate layers are called “hidden”, since they do not have direct connections towards the inputs or the outputs. Each hidden unit sums its inputs and adds a constant term (bias) in order to form the total input  $h_j = \alpha_j + \sum_l w_{jl}x_l$ . A function  $f$  is applied to this to generate the output  $V_j = f(h_j)$ . Thus, the output unit receives  $h_i = \alpha_i + \sum_j W_{ij}V_j$  and produces a final output  $O_i = f(h_i)$  (Hertz and Palmer, 1991).

Observe that the index  $l$  refers to the inputs  $x$ ,  $j$  to the neurons in the hidden layer and  $i$  to the output units. Therefore, a network as that in figure 1 represents the function

$$O_i = f(h_i)$$

with:

$$h_i = \alpha_i + \sum_j W_{ij}f(h_j)$$

and

$$h_j = \alpha_j + \sum_l w_{jl}x_l \quad (1)$$

Usually, the non-linear function  $f$  is a sigmoid of the form

$$f(h) = \frac{1}{e^{-h} + 1} \quad (2)$$

Generalmente la función no-lineal  $f$  es una sigmoide de la forma

$$f(h) = \frac{1}{e^{-h} + 1} \quad (2)$$

Las unidades de entrada solamente se usan para distribuir las entradas de manera tal que, para ellas,  $f(h) = h$ .

Esta aproximación posee una motivación biológica dada por McCulloch y Pitts (1943). Ésta es bastante burda con respecto al modelo biológico de una red neuronal en el cerebro, sin embargo, ha demostrado ser de gran utilidad en tareas como las descritas anteriormente.

### Entrenamiento

La ecuación (1) representa una **parametrización** de la relación entrada-salida para datos multidimensionales. Esta representación se encuentra dentro de una clase general de funciones, capaces de aproximar cualquier función continua de manera uniforme, siempre que se posean suficientes unidades en las capas ocultas (Bishop, 1997). El entrenamiento o aprendizaje representa la tarea de encontrar el conjunto de pesos en las conexiones que permitan tener la mejor representación entrada-salida.

Suponiendo que se tiene un conjunto de datos, patrones de entrenamiento o ejemplos  $\{\mathbf{x}_p, \mathbf{d}_p\}$ , donde  $\mathbf{x}_p$  es el vector de entradas,  $\mathbf{d}_p$  es el vector de salidas deseadas,  $p$  indica el número de ejemplo o patrón, y que la salida de la red se puede expresar como  $\mathbf{y} = F(\mathbf{x}; \mathbf{w})$ , entonces, el entrenamiento se realiza escogiendo el vector  $\mathbf{w}$  que minimice

$$E(\mathbf{w}) = \sum_p \|\mathbf{d}_p - F(\mathbf{x}_p; \mathbf{w})\|^2 \quad (3)$$

Rumelhart *et al.* (1986) propusieron realizar el proceso de minimización usando una forma de descenso de gradiente, con regla de actualización

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial E}{\partial w_{ij}} \quad (4)$$

donde  $\eta$  es un parámetro que controla la velocidad de aprendizaje, esto es, la tasa de aprendizaje. Ellos observaron que las derivadas parciales se pueden calcular de manera eficiente por medio de un algoritmo conocido como retropropagación de errores (error backpropagation) (Rumelhart y Williams, 1986) que ha resultado ser el método estándar para realizar el entrenamiento en este tipo de redes. El algoritmo realiza la actualización (4), para cada peso de la red y cada patrón de entrenamiento. Al haberse realizado el proceso de entrenamiento para todos los patrones, se ha completado una época. Una vez concluido el entrenamiento, cuando el error (3) ha disminuído aceptablemente, se puede evaluar el

The input units are only used to distribute inputs, such that, for them,  $f(h) = h$ .

This approximation has a biological motivation given by McCulloch and Pitts (1943). This is very rough in respect to the biological model of a neural network in the brain, however, it has demonstrated to be of great utility in tasks such as those previously described.

### Training

Equation (1) represents a parametric input-output relationship for multidimensional data. This representation is included in a general class of functions, capable of uniformly approximating any continuous function, when sufficient units are possessed in the hidden layers (Bishop, 1997). The training or learning represents the task of finding the set of weights in the connections that allow to have the best input-output representation.

Assuming that there is a data set of training patterns or examples  $\{\mathbf{x}_p, \mathbf{d}_p\}$ , where  $\mathbf{x}_p$  is the input vector,  $\mathbf{d}_p$  is the desired output vector,  $p$  indicates the number of example or pattern, and that the network output can be expressed as  $\mathbf{y} = F(\mathbf{x}; \mathbf{w})$ , then the training is done by selecting the vector  $\mathbf{w}$  that minimizes

$$E(\mathbf{w}) = \sum_p \|\mathbf{d}_p - F(\mathbf{x}_p; \mathbf{w})\|^2 \quad (3)$$

Rumelhart *et al.* (1986), proposed performing the minimization process using gradient descent, with the updating rule

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial E}{\partial w_{ij}} \quad (4)$$

where  $\eta$  is a parameter that controls the speed of learning, thus called learning rate. They observed that the partial derivatives could be efficiently calculated by an algorithm known as error backpropagation (Rumelhart and Williams, 1986), which became the standard method for training in this kind of networks. The algorithm performs the updating (4), for each weight in the network, as well as for each training pattern. When the training process has been done for all of the patterns, an epoch has been completed. Once the training is finished, when the error (3) has decreased to an acceptable level, the behavior of the network can be evaluated considering patterns not included during the training. This evaluation is performed by calculating an adjust measure, as the normalized mean quadratic error (NMQE):

$$NMQE = \frac{\sum_{j,k} (d_{jk} - y_{jk})^2}{\sum_{j,k} (d_{jk} - \mu)^2} \quad (5)$$

comportamiento de la red ante patrones no incluidos durante el entrenamiento. Esta evaluación se realiza mediante el cálculo de una medida de ajuste, como el error cuadrático medio normalizado (*ECMN*)

$$ECMN = \frac{\sum_{j,k} (d_{jk} - y_{jk})^2}{\sum_{j,k} (d_{jk} - \mu)^2} \quad (5)$$

donde  $d_{jk}$  es el valor real de la salida  $j$  para el patrón  $k$ ,  $y_{jk}$  es la predicción y  $\mu$  es la media muestral de la salida real sobre todos los patrones de prueba. Las sumas son sobre todas las salidas, para todos los patrones de prueba.

### Predicción de señales temporales

Las RNA han probado ser una alternativa poderosa a las técnicas tradicionales de regresión y clasificación. Sin embargo, la predicción de patrones temporales es un problema particularmente difícil y requiere de modificar la topología de la red. Para este tipo de tareas, las arquitecturas más adecuadas han demostrado ser la recurrente (RNR) (Hertz y Palmer, 1991), la de retardo en el tiempo (RNRT) (Hertz y Palmer, 1991) y, más recientemente, la de respuesta finita al impulso (RNRFI) (Wan, 1993). El objetivo es entrenar una RNA usando el conjunto de variables  $x(t)$ ,  $x(t - \Delta)$ ,  $x(t - 2\Delta)$ , ...,  $x(t - (M-1)\Delta)$  como patrón de entrada, donde  $\Delta$  es un retardo en el tiempo.

Las RNRs se obtienen al modificar la topología de la red convencional en avance, permitiendo conexiones de retroalimentación, cuidadosamente seleccionadas. La recurrencia permite que la red almacene parte de la secuencia pasada, sin modificar mucho el entrenamiento. Existen varias arquitecturas propuestas que incorporan la recurrencia (Hertz y Palmer, 1991), con aplicaciones principalmente en el reconocimiento de secuencias.

La RNRT representa una forma muy simple de reconocer secuencias, que incorpora parte de la secuencia en un patrón espacial para la capa de entrada a la red. Se toma parte de la señal, y los valores  $x(t)$ ,  $x(t - \Delta)$ ,  $x(t - 2\Delta)$ , ...,  $x(t - (M-1)\Delta)$  se presentan simultáneamente en la entrada de una red en avance con  $M$  unidades en la capa de entrada y una unidad de salida. El nombre red de retardo en el tiempo proviene de que su implementación puede ser mediante el paso de la señal a través de una línea de retardo, como se observa en la figura 2.

El entrenamiento es mediante retropropagación de los errores. Su aplicación ha tenido gran éxito en la predicción de series de tiempo (Mozer, 1993; Wan, 1993).

### Redes neuronales de respuesta finita al impulso (RNRFI)

En el modelo tradicional, tal como se describió anteriormente, los pesos de las conexiones son variables reales y el punto de unión, corresponde a una simple suma ponderada

where  $d_{jk}$  is the real output value  $j$  for the pattern  $k$ ,  $y_{jk}$  is the prediction and  $\mu$  is the sample mean of the real output over all the test patterns. The sums are over all the outputs, for all the test patterns.

### Prediction of temporal signals

The ANNs are a powerful alternative to the traditional regression and classification techniques. However, the prediction of temporal patterns is a particularly difficult problem and requires modifying the network topology. For this kind of tasks, the most adequate architectures are the recurrent neural network (RNN) (Hertz and Palmer, 1991), the time-delay neural network (TDNN) (Hertz and Palmer, 1991) and, more recently, the finite impulse response neural network (FIRNN) (Wan, 1993). The objective is to train an ANN using the set  $x(t)$ ,  $x(t - \Delta)$ ,  $x(t - 2\Delta)$ , ...,  $x(t - (M - 1)\Delta)$  as input pattern, where  $\Delta$  is a time delay.

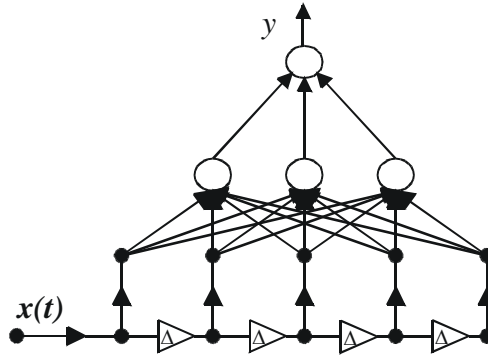
Recurrent networks are obtained by modifying the topology of the conventional feedforward network allowing feed-back connections carefully selected. Recurrence allows the network to store part of the past sequence without modifying much of the training. There are several architectures proposed that incorporate recurrence (Hertz and Palmer, 1991), with applications mainly in the recognition of sequences.

The TDNN represents a simple way of recognizing sequences, which incorporates part of the sequence in a spatial pattern for the network input layer. Part of the signal is taken, and values  $x(t)$ ,  $x(t - \Delta)$ ,  $x(t - 2\Delta)$ , ...,  $x(t - (M - 1)\Delta)$  are presented simultaneously in the input of a feedforward network with  $M$  units in the input layer and one output unit. The name of time-delay network is given since its implementation can be made by passing the signal through a delay line, as observed in figure 2.

The training is done by error backpropagation. Its application has been very successful in the time series prediction (Mozer, 1993; Wan, 1993).

### Finite impulse response neural network (FIRNN).

In the traditional model, as described previously, the weights of the connections are real variables and their junction corresponds to a single static weighted sum, giving only the spatial representation of the signals that arrive from different neurons. In the FIRNN architecture, the model extends since now the weights of the connections correspond to finite impulse response filters (Wan, 1990). Finite impulse response (FIR) filters are equivalent to self-regressive models, or moving averages, in which the response or output is guaranteed to be zero in a certain number of time-steps, when the input is zero (Gershenfeld and Weigend, 1994).



**Figura 2.** Esquema de una red temporal. La  $x$  es la señal de entrada y las  $\Delta$  son los retardos que se le aplican a la entrada.  
**Figure 2.** Temporal network diagram.  $x$  is the input signal and  $\Delta$  are the delays that are applied in the input.

estática, proporcionando sólo la representación espacial de las señales que llegan de diferentes neuronas. En la arquitectura de las RNRFI se extiende el modelo, ya que ahora los pesos de las conexiones corresponden a filtros de respuesta finita al impulso (Wan, 1990). Los filtros de respuesta finita al impulso son equivalentes a modelos autoregresivos, o promedios móviles, en los cuales se garantiza que, cuando la entrada es cero, la respuesta o salida es cero en un cierto número de pasos de tiempo (Gershenfeld y Weigend, 1994).

Al agregar el filtrado, se está incluyendo la dimensión temporal fundamental, pues cada filtro corresponde a una línea de retardo. El peso entre la neurona  $i$  y la  $j$ , que anteriormente era un escalar, ahora corresponde al vector  $\mathbf{w}_{ij} = [w_{ij}(0), w_{ij}(1), \dots, w_{ij}(T)]$ , donde  $T$  corresponde al número de retardos en el tiempo para la capa correspondiente al peso  $w_{ij}$ .

Para adaptar los filtros a la red, se supone que cada uno de ellos se comporta como una respuesta al impulso finita y discreta, es decir, ahora la entrada total de una unidad, depende de las entradas retardadas

$$s(k) = \sum_{i=0}^N \sum_{n=0}^T w_i(n)x_i(k-n) = \sum_i \mathbf{w}_i \cdot \mathbf{x}_i(k) \quad (6)$$

donde  $k$  es el índice discreto de tiempo para alguna razón de muestreo  $\Delta$  y  $w_i(n)$  corresponde a los coeficientes de los filtros RFI, y el estado retardado es  $\mathbf{x}_i(k) = [x_i(k), x_i(k-1), \dots, x_i(k-M)]$ , donde  $M$  corresponde al orden del filtro.

El filtrado se realiza mediante el producto punto  $\mathbf{w}_i \cdot \mathbf{x}_i(k)$  donde la relación temporal está implícita en la notación. La figura 3 muestra una representación típica de un filtro RFI. Como se observa, la representación corresponde a una red tradicional en avance, en la que los pesos de los arcos corresponden a los filtros RFI.

Para completar el modelo, tenemos que la salida es:

$$y(k) = f(s(k)) \quad (7)$$

donde sólo se ha agregado el índice temporal  $k$ .

By adding the filtering, the fundamental temporal dimension is being included since every filter corresponds to a delay line. The weight between neurons  $i$  and  $j$ , previously a scalar, now is a vector  $\mathbf{w}_{ij} = [w_{ij}(0), w_{ij}(1), \dots, w_{ij}(T)]$ , where  $T$  corresponds to the number of time delays for the layer corresponding to the weight  $w_{ij}$ .

To adapt the filters to the network, each one of them is supposed to behave as a finite and discrete response to the impulse; e.g., now the total input of a unit depends on the delayed inputs

$$s(k) = \sum_{i=0}^N \sum_{n=0}^T w_i(n)x_i(k-n) = \sum_i \mathbf{w}_i \cdot \mathbf{x}_i(k) \quad (6)$$

where  $k$  is the discrete index of time for any sampling rate  $\Delta$  and  $w_i(n)$  corresponds to the coefficients of the FIR filters and the delayed state is  $\mathbf{x}_i(k) = [x_i(k), x_i(k-1), \dots, x_i(k-M)]$ , where  $M$  belongs to the filter order.

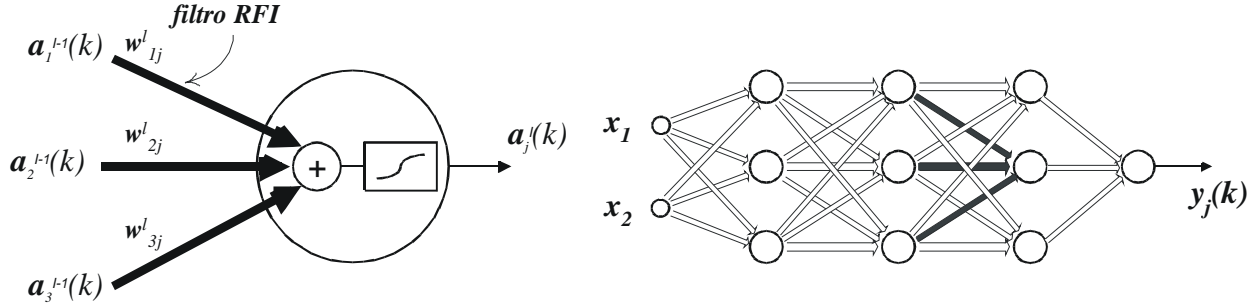
Filtering is performed by the dot product  $\mathbf{w}_i \cdot \mathbf{x}_i(k)$  where the temporal relation is implicit in the notation. Figure 3 shows a typical representation of a FIR filter. As it is observed, the representation corresponds to a traditional feedforward network in which the weights of the arcs correspond to FIR filters.

To complete the model, the output is:

$$y(k) = f(s(k)) \quad (7)$$

where the temporal index  $k$  has been included.

The learning can be done by building a static equivalent for the network, this is, specifying each filter with its corresponding temporal delay (what is known as unfolding) and backpropagating the structure obtained. Wan (1993), proposed a new algorithm that he called temporal backpropagation, that allows to perform the training more efficiently, without the need of unfolding.



**Figura 3.** Arquitectura tipo de respuesta finita al impulso (RFI). En el modelo espacio-temporal de la neurona, las señales de entrada  $a(k)$  se pasan a través de filtros sinápticos  $w$ . Todas las conexiones se modelan con filtros de orden  $k$  tipo RFI.

**Figure 3.** Finite impulse response (FIR) architecture. In the space-temporal model of the neuron, input signals  $a(k)$  pass through synaptic filters  $w$ . All connections are modeled with  $k$ -order filters of the kind FIR.

El aprendizaje se puede realizar construyendo un equivalente estático de la red, esto es, especificando cada filtro con su correspondiente retardo temporal, (lo que se conoce como desdoblamiento) y aplicando retropropagación a la estructura resultante. Wan (1993) propuso un nuevo algoritmo, al que llamó retropropagación temporal, que permite realizar el entrenamiento de manera más eficiente, sin necesidad de realizar el desdoblamiento.

#### Pre-procesamiento de señales temporales

Antes de realizar el modelado de la serie temporal, se consideró adecuado hacer un análisis previo de la misma usando una herramienta conocida como prueba Gamma (Durrant, 2001). Dicha prueba es una técnica de análisis que estima la varianza del ruido en un conjunto continuo de datos y nos ayuda a conocer el error cuadrático medio que debemos permitir durante el entrenamiento para evitar un posible sobreentrenamiento. También es posible estimar el número necesario de entradas a la red neuronal a través de la dimensión de inmersión (*embedding dimension*), así como el número de datos suficientes para construir el modelo (Stefánsson y Jones, 1996). Lo anterior nos permite en forma aparente, determinar la configuración de la red y depender menos de la técnica de prueba y error.

Basados en los resultados obtenidos al aplicar la prueba Gamma a datos del IOS, se construyeron modelos de RNRTs con arquitecturas  $n_i - 30 - 24 - 1$ , donde  $n_i$  representa el número de unidades de entrada, el cual depende de la máscara utilizada. La configuración se encontró mediante prueba y error, se realizaron pruebas con menos unidades pero no se obtuvo convergencia. El entrenamiento se realizó hasta que el error  $E$  se redujo a 0.001, con una tasa de aprendizaje  $\eta = 0.01$  y un momentum  $\alpha = 0.8$ . Los valores de  $\eta$  y  $\alpha$  también se obtuvieron mediante prueba y error. La red captura la dinámica del proceso, esto es, puede realizar predicciones de un paso cuando se le alimenta con patrones que pertenecen a la serie, incluso con los de prueba. Sin embargo, al momento de realizar la predicción iterada, su comportamiento diverge

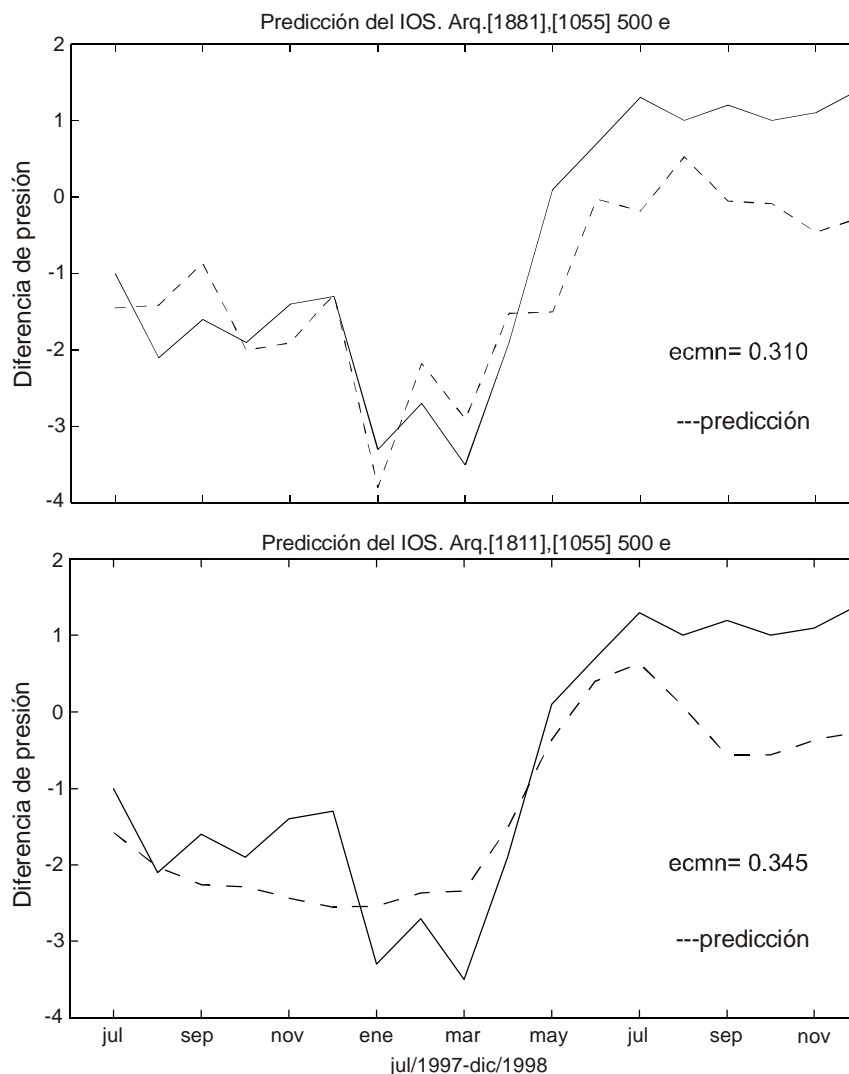
#### Pre-processing of temporal signals

Prior to modeling the temporal series, previous analysis was considered adequate using a tool known as Gamma test (Durrant, 2001). This test is an analysis technique that estimates the variance of the noise in a continuous data set and that helps to estimate the quadratic mean error that we can allow during the training, in order to avoid any possible over-training. It is also possible to estimate the number of inputs needed by the neural network by means of the embedding dimension, as well as the sufficient number of data to build the model (Stefánsson and Jones, 1996). This apparently allows us to determine the network configuration and to depend less on the trial and error technique.

Based on the results obtained when applying the Gamma test to the SOI data, TDNN models were built with architectures  $n_i - 30 - 24 - 1$ , where  $n_i$  represents the number of input units, depending on the mask used. The configuration was found by means of trial and error, tests were done with less units but no convergence was obtained. The training was done until the error  $E$  was reduced to 0.001, with a learning rate  $\eta = 0.01$  and a momentum  $\alpha = 0.8$ . Values  $\eta$  and  $\alpha$  were also obtained by trial and error. The network captures the dynamics of the process, that is, it can perform one-step predictions, when fed with patterns belonging to the series, even with those of the test set. However, at the moment of performing iterated prediction, its behavior diverge completely, as is observed from the  $MNQE = 7.5$  obtained after applying iterated prediction for the next 12 points of the series. For the SSTA data, the behavior is similar with networks trained with  $n_i - 20 - 10 - 1$  and  $n_i - 30 - 24 - 1$  architectures.

From the results obtained when applying the Gamma test to the studied time series, it can be concluded that the adequate configuration to perform iterated predictions cannot be precisely determined, thus more research is necessary (Durrant, 2001).

It is convenient to clear out that training a model to reproduce a time series does not imply that this will be able to



**Figura 4.** Predicción de una serie de tiempo del Índice de Oscilación del Sur (IOS) con datos de 1951 a junio de 1997 inclusive. Se predice la segunda mitad de 1997 y 1998. La predicción es la línea punteada. En la parte inferior de la figura, se muestra el efecto de disminuir la cantidad de neuronas en la segunda capa oculta.

**Figure 4.** Time series prediction of the Southern Oscillation Index (SOI) with data from 1951 to June 1997 inclusive. The second half of 1997 and all 1998 are predicted. The prediction is the dashed line. In the bottom of the figure, the effect of lowering the quantity of neurons in the second hidden layer is shown.

completamente, como se observa del *ECMN* de 7.5 obtenido después de aplicar la predicción iterada para los siguientes 12 puntos en la serie. Para los datos de ATSMs el comportamiento es similar, lográndose entrenar redes con arquitectura  $n_i - 20 - 10 - 1$  y con  $n_i - 30 - 24 - 1$ .

De los resultados obtenidos al aplicar la prueba Gamma a las series de tiempo estudiadas, se puede concluir que todavía no es posible determinar con precisión la configuración adecuada para realizar predicciones iteradas, lo cual requiere de más investigación (Durrant, 2001).

Es conveniente aclarar que entrenar un modelo para reproducir una serie de tiempo no implica que éste sea capaz de realizar predicciones iteradas a largo plazo, incluso aunque se obtenga un error de entrenamiento pequeño.

perform iterated long-term predictions, even when a small training error is obtained.

## Experiments and results

### Southern Oscillation Index prediction

For the prediction of the SOI, data published in the web site [www.nmic.noaa.gov](http://www.nmic.noaa.gov) were used. The series used comprises from 1951 to 1998, with a one-month sampling intervals. These data were normalized scaling to zero average and unitary variance. The first 546 data were taken from the series and were used to train a FIRNN. The next 18 data were used to test the prediction capacity.



## Experimentos y resultados

### Predicción del Índice de la Oscilación del Sur

Para la predicción del IOS, se utilizaron datos publicados en el sitio [www.nmic.noaa.gov](http://www.nmic.noaa.gov). La serie que se usó para la predicción comprende desde 1951 hasta 1998, con un intervalo de muestreo de un mes. Estos datos fueron normalizados escalando a cero de promedio y varianza unitaria. Se tomaron los primeros 546 datos de la serie y se usaron para entrenar una RNRFI. Los 18 datos siguientes se usaron para probar la capacidad de predicción.

Con el fin de encontrar la configuración adecuada al problema, se realizaron varios experimentos, partiendo de la configuración base construida en función de los parámetros estimados con la prueba Gamma. Ésta determinó que el error de entrenamiento debe ser del orden  $O(10^{-4})$ , el número de entradas debe ser alrededor de 36, es decir, los vectores de entrenamiento de la red deberán ser retardados 36 pasos en el tiempo y, con respecto al número mínimo de datos necesarios para el aprendizaje, se aplicó la prueba “M” y se encontró que todos los datos son necesarios, dado que el número de inmersión es muy grande (36) (Durrant, 2001). Finalmente, después de muchas pruebas se encontró que la mejor configuración utiliza la unidad de entrada, dos capas posteriores, y la unidad de salida. Esta arquitectura posee 8 unidades en cada capa, con filtros de orden 10 para los pesos de entrada a la primera capa y de orden 5, para los restantes. El valor de la razón de aprendizaje  $\eta = 0.001$  también fue estimado por prueba y error.

En la figura 4 se muestran los resultados obtenidos al entrenar la red con una serie de 1951 a junio de 1997 inclusive (546 datos) y se predice el IOS para la segunda mitad de 1997 y todo 1998 (18 datos). Para este parámetro, no es posible comparar los resultados con otras predicciones, ya que en el fenómeno del ENOS, la variable que más comúnmente se correlaciona, es la ATSM y no el IOS.

### Predicción de anomalías de la temperatura superficial del mar

A continuación se comparan tres métodos de predicción del parámetro ATSM. Se consideró una serie de tiempo publicada por la *National Oceanic and Atmospheric Administration* de los Estados Unidos en el sitio [www.nmic.noaa.gov](http://www.nmic.noaa.gov) de 1950 a 1998. El primero es un modelo acoplado océano-atmósfera con circulación global (GCM) que incorpora mediciones subsuperficiales en el océano en las condiciones iniciales (Huang *et al.*, 1999). Este modelo se aplica en el COLA (*Center for Ocean-Land-Atmosphere*). El segundo es un modelo dinámico acoplado océano-atmósfera (CMF) de bajo orden (Kleenman, 1995), que se usa en el BMRC (*Bureau of Meteorology Research Center*). El tercero, en el que se aplican las RNA, utiliza una configuración muy semejante a la que se usó con el

Aiming to find the adequate configuration to the problem, several experiments were done, starting with the base configuration built using the parameters estimated with the Gamma test. This test determined a training error of order  $O(10^{-4})$  and an input number around 36, this is, the training vectors of the network would be delayed 36 steps in time to obtain the minimum number of data needed for the learning, a test “M” was applied, finding that all data are required since the number of immersions is great (36) (Durrant, 2001). Finally, after a lot of tests, the best configuration was found to be the one that uses the input unit, two posterior layers and the output unit. This architecture has 8 units in each layer, with 10th-order filters for the input weights in the first layer and of 5th order in the rest. The learning rate value  $\eta = 0.001$  was also estimated by trial and error.

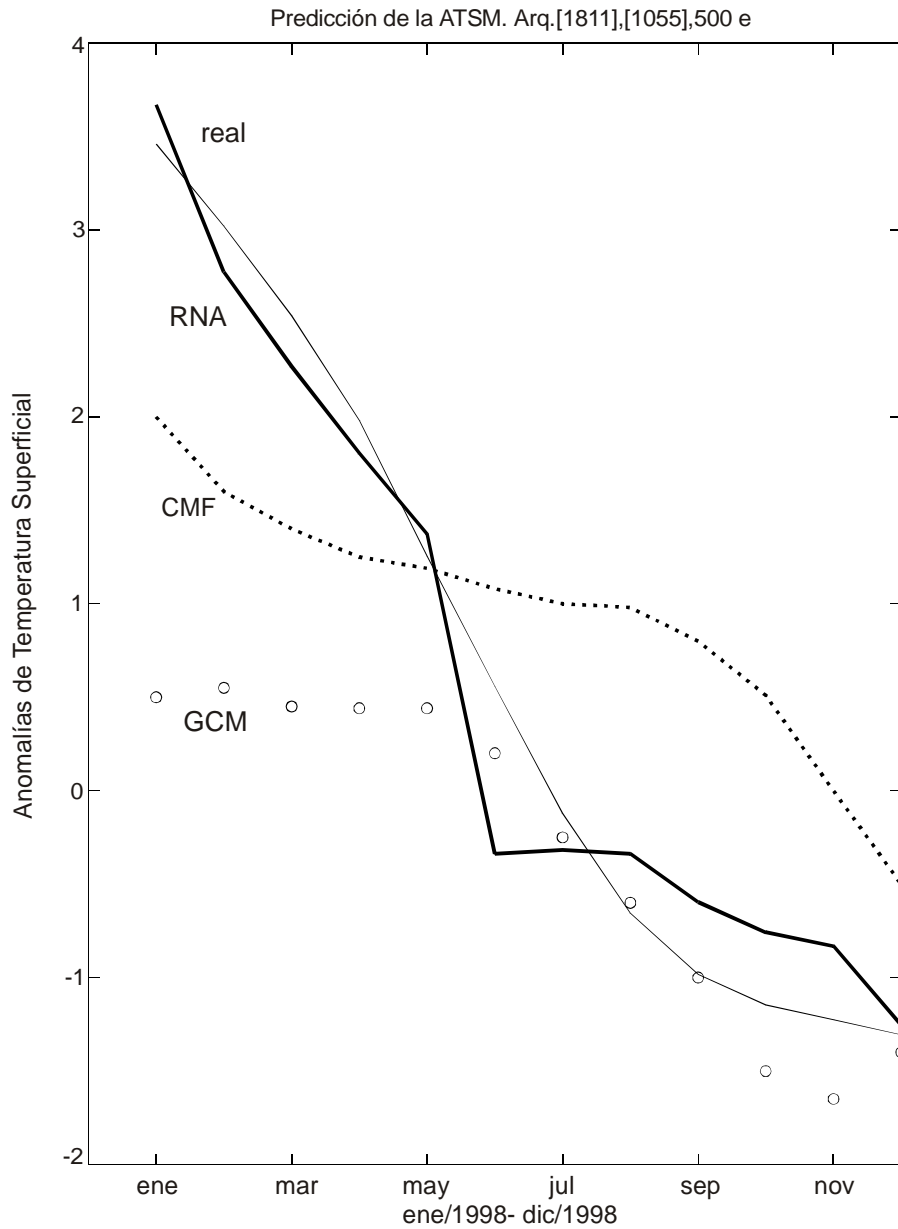
In figure 4 the results of training the network with a series from 1951 to June 1997 inclusive (546 data) are shown, and the second half of 1997 and all 1998 are predicted (18 data). For this parameter, it is not possible to compare the results with other predictions since, in the ENSO phenomenon, the variable commonly correlated is the SSTA and not the SOI.

### Prediction of the sea surface temperature anomalies

Three prediction methods for the SSTA parameter are compared next. A time series published by the U. S. National Oceanic and Atmospheric Administration, in the web site [www.nmic.noaa.gov](http://www.nmic.noaa.gov), from 1950 to 1998 was considered. The first one is a coupled ocean-atmosphere model with global circulation (GCM) that incorporates subsurface measurements in the ocean at initial conditions (Huang *et al.*, 1999). This model is applied in the Center for Ocean-Land-Atmosphere (COLA). The second one is a dynamic ocean-atmosphere model (CMF) of low order (Kleeman, 1995) that is used in the Bureau of Meteorology Research Center (BMRC). The third one, in which ANNs are applied, uses a configuration similar to that used with the SOI and estimated in the same way, an input neuron with a 10th-order filter, 8 neurons in the first layer with a 5th-order filter, one neuron in the second layer with a 5th-order filter and a neuron in the output layer. The training consisted of 500 epochs with the data set from 1950 to 1997 (576 data) and predicted the SSTAs corresponding to 1998. Correlation coefficients (*CCOR*) were calculated for each case. For the GCM model, the estimated *CCOR* was 0.832, for the CMF it was 0.848 and for the ANN it was 0.977. Figure 5 shows the actual curve and the three predictions. In it, the ANN can be observed to generate the best prediction which, in addition, features the highest correlation coefficient.

## Conclusions

Artificial neural networks algorithms were applied to the prediction of oceanographic time series data. Southern Oscillation Index and sea surface temperature anomalies were



**Figura 5.** Comparación de la predicción de anomalías de la temperatura superficial del mar (ATSM) para 1998 a partir de una serie de tiempo, por tres métodos diferentes. Para las redes neuronales artificiales (RNA) el coeficiente de correlación resulta de 0.979, para el modelo GCM de 0.832 y para el modelo CMF de 0.841.

**Figure 5.** Comparison of the prediction of the sea surface temperatures (SSTA) for 1998 from a time series, by three different methods. For the artificial neuron network (ANN) the correlation coefficient is 0.979, for the GCM model it is 0.832 and for the CMF model it is 0.841.

IOS, estimada de igual manera: una neurona de entrada, con un filtro de orden 10, ocho neuronas en la primera capa con un filtro de orden 5, una neurona en la segunda capa con un filtro de orden 5, y una neurona en la capa de salida. El entrenamiento consistió en 500 épocas con el conjunto de datos del periodo de 1950 a 1997 (576 datos) y se predijo la ATSM correspondiente a 1998. Se calcularon los coeficientes de correlación (*CCOR*) para cada caso. Para el modelo GCM el *CCOR* estimado fue de 0.832, para el CMF fue de 0.848 y para las RNA fue de 0.977. La figura 5 muestra la curva real y las

predicted using only historical time series data. That is, no complicated model was used that had to be initialized with boundary conditions or any other variable related to the physical phenomenon of which the time series is being predicted.

Different configurations of finite impulse response neuronal networks were tested, varying the different parameters of the network, from a base configuration that was estimated by applying the Gamma test (Durrant, 2001) to the time series studied. The final configuration was obtained by trial and error, as was the learning rate  $\eta$ . For the case of SOI and SSTA time

tres predicciones. En ella se puede observar que las RNA generan la mejor predicción que, además, posee el mayor coeficiente de correlación.

## Conclusiones

Se aplicaron algoritmos basados en redes neuronales artificiales para la predicción de series de tiempo de las variables oceanográficas Índice de la Oscilación del Sur y anomalías de la temperatura superficial del mar únicamente a partir de las series de tiempo históricas. Esto es, no se utilizó ningún modelo complicado que requiera ser inicializado con condiciones de frontera o alguna otra variable relacionada con el fenómeno físico cuya serie de tiempo se pretende predecir.

Se probaron diferentes configuraciones de redes neuronales tipo RFI, variando los diversos parámetros de la red, y partiendo de una configuración base que se estimó aplicando la prueba Gamma (Durrant, 2001) a las series de tiempo estudiadas. La configuración final se obtuvo mediante prueba y error, de igual manera que el valor de la tasa de aprendizaje  $\eta$ . Se encontró que, para el caso de series temporales del IOS y las ATSMs, la red más adecuada se construye con una capa de entrada de una neurona y un filtro de orden 10, dos capas ocultas con 8 neuronas en cada una de ellas y filtros de orden 5, y una capa de salida con una neurona. Todas las redes fueron entrenadas con el algoritmo de aprendizaje de retropropagación temporal, usando la sigmoide como función de activación y una razón de aprendizaje de 0.001.

Para los experimentos realizados, en la mayoría de los casos se obtuvo un error cuadrático medio normalizado de  $0.4 \pm 0.1$  y un coeficiente de correlación mayor que 0.8, entre la serie original y la predicha. Estos resultados son válidos sólo en la predicción a corto plazo.

## Agradecimientos

Se agradece a Erick Wan, por haber proporcionado los programas fuente del simulador de redes neuronales, que fueron la base para el desarrollo de este trabajo y a Julio Sheinbaum, por habernos proporcionado la serie de tiempo del IOS.

## Referencias

- Bishop, C. (1997). *Neural Networks for Pattern Recognition*. Oxford University Press.
- Durrant, P. (2001). *WinGamma: A non linear data analysis and modeling tool for the investigation of non-linear and chaotic systems with applied techniques for a flood prediction system*. PhD thesis. Department of Computer Science, Cardiff University.
- Gershenfeld, N. y Weigend, A. (1994). The future of time series: Learning and understanding. En A.S. Weigend and N.A. Gershenfeld (eds.), *Time Series Prediction; Forecasting the Future and Understanding the Past*. SFI Studies in the Science of Complexity, Proc. Vol. XV, Addison-Wesley.
- Hertz, J.; Krogh, A. y Palmer, G. (1991). *Introduction to the Theory of Neural Computation*, Addison-Wesley.
- Huang, B., Zhengxin, A., David, G., DeWitt, J. y Edwin, D. (1999). SST predictions with a global coupled GCM, Technical report, Center for Ocean-Land-Atmosphere Studies (COLA), Calverton, Maryland.
- Kleenman, R. (1995). Using a low order coupled ocean-atmosphere dynamical model, *Mon. Wea. Rev.* (123):3103–3113.
- Mozer, M. (1993). Neural net architectures for temporal sequence processing. En: A.S. Weigend y N.A. Gershenfeld (eds.), *Time Series Prediction; Forecasting the Future and Understanding the Past*. SFI Studies in the Science of Complexity, Proc. Vol. XV, Addison-Wesley.
- Philander, S. (1990). *El Niño, La Niña and the Southern Oscillation*, Academic Press.
- Rumelhart, D.E., Hinton, G.E. y Williams, R.J. (1986). Learning internal representation by error propagation. En: D.E. Rumelhart, J.L. McClelland y el grupo PDP (eds.), *Parallel Distributed Processing; Explorations in the Microstructure of Cognition*, Vol. 1 Foundations. MIT Press, Cambridge, Massachusetts, pp. 318–362.
- Stefánsson, A; Koncâr, N. y Jones, A. (1996). A note on the Gamma test, *Neural Computing and Applications* 5(3):131–133.
- Wan, E. (1990). Temporal backpropagation for FIR neural networks, en *Proceedings of the International Joint Conference on Neural Networks*, San Diego, CA, pp 575–580.
- Wan, E. (1993). Time series prediction by using a connectionist network with internal delay lines. En: A.S. Weigend y N.A. Gershenfeld (eds.), *Time Series Prediction; Forecasting the Future and Understanding the Past*. SFI Studies in the Science of Complexity, Proc. Vol. XV, Addison-Wesley.

## Acknowledgements

We thank Erick Wan for the source programs for the neural networks simulator, base of the development of this work. Thanks to Julio Sheinbaum for providing the time series of the Southern Oscillation Index.

English translation by Adrián R. López