Neural Networks-Based Time Series Prediction Using Long and Short Term Dependence in the Learning Process

J. Pucheta, D. Patiño and B. Kuchen,

***Abstract*— In this work a feedforward neural networks-based nonlinear autoregression (NAR) model for forecasting time series is presented. The learning rule used to adjust the neural net weights is based on the Levenberg-Marquardt method. In function of the long and short stochastic dependence of the time series, we propose an on-line heuristic law to set the training process and to modify the neural net topology. The input patterns for the neural network-based model are the values of the time series after applying a time-delay operator. Hence, the neural-net output will tend to approximate the current value available from the series. The coefficients of the nonlinear filter are adjusted on-line in the learning process, by considering a criterion that modifies at each time-stage the number of patterns, the number of iterations, and the length of the tapped-delay line, in function of the Hurst's value *(H)* calculated for the time series. According to the stochastic behavior of each series, *H* can be greater or smaller than 0.5, which means that each series tends to present long or short term dependence, respectively. The algorithm is applied to the 11 time series to forecast the next 18 values given in the *NN3 Forecasting Competition* for Neural Networks and Computational Intelligence.**

I. INTRODUCTION

A. Overview of the NN Approach

This work presents a solution to the NN3 Forecasting Competition for the Neural Networks & Computational Intelligence, which is organized as special sessions of the International Symposium of Forecasting, ISF'07, International Joint Conference on Neural Networks, IJCNN'07, and International Conference on Data Mining, DMIN'07. The proposed solution is based on the classical nonlinear autoregression filter using *time lagged feedforward networks.* The innovation is made on the learning process, which employs the Levenberg-Marquardt rule and considers the long and short term stochastic dependence of passed values of the time series to adjust at each time-stage the number of patterns, the number of iterations, and the length of the tapped-delay line, in function of the Hurst's value *(H)* of the signal. According to the stochastic characteristics of each series, *H* can be greater or smaller than 0.5, which means that each series tends to present long or short term dependence, respectively.

In order to adjust the design parameters and see the performance of the proposed prediction model, sinusoidal and square signals are used. Then, the neural network-based nonlinear filter is applied to the 11 time series to forecast the next 18 values given in the NN3 Forecasting Competition.

B. Fractional Brownian Motion

In this work the Hurst's value is used in the learning process to modify on-line the number of patterns and number of iterations presented. The *H* parameter is useful for the definition of the Fractional Brownian Motion (**fBm**).

The **fBm** is defined in the pioneering work by Mandelbrot and van Ness [6], through its stochastic representation

$$B_H(t)=\frac{1}{\Gamma\left(H+\frac{1}{2}\right)}\left(\int_{-\infty}^{0}\left((t-s)^{H-\frac{1}{2}}-(-s)^{H-\frac{1}{2}}\right)dB(s)+\int_{0}^{t}(t-s)^{H-\frac{1}{2}}dB(s)\right) \quad (1.1)$$

where, $\Gamma(\cdot)$ represents the Gamma function

$$\Gamma(\alpha)=\int_0^\infty x^{\alpha-1}e^{-x}dx, \quad (1.2)$$

and 0<*H*<1 is called the Hurst parameter. The integrator *B* is a stochastic process, ordinary Brownian motion. Note, that *B* is recovered by taking *H*=1/2 in (1.1). Here, it is assumed that *B* is defined on some probability space (Ω, *F, P*), where *Ω, F* and *P* are the sample space, the sigma algebra (event space) and the probability measure, respectively. So, a **fBm** is a continuous-time Gaussian process depending on the so-called Hurst parameter 0<*H*<1. It generalizes the ordinary Brownian motion corresponding to *H*=0.5, and whose derivative is the white noise.

The **fBm** is self-similar in distribution and the variance of the increments is given by

$$\mathrm{Var}\left(B_H(t)-B_H(s)\right)=\nu\left|t-s\right|^{2H} \quad (1.3)$$

where, *v* is a positive constant.

This special form of the variance of the increments suggests various ways to estimate the parameter *H*. In fact, there are different methods for computing the parameter *H* associated to Brownian Motion [2] [3] [5]. In this work, the algorithm uses a wavelet-based method for estimating *H* from a trace of the **fBm** with parameter *H* [1] [3] [4]. The trace path from the **fBm** are shown in Fig. 1, where can be noted the difference in the velocity and the amount of its increments.

Manuscript received March 14, 2007. This work was supported in part by the National Agency for Scientific and Technological Promotion (ANPCyT) under grant PAV-TIC-076, PICT/04 N^0 25423.

J. A. Pucheta is with the Department of Electronic Engineering, Faculty of Exact, Physical and Natural Sciences, at National University of Córdoba. X5016GCA, Córdoba, Argentina. (julian.pucheta@gmail.com).

The authors H.D. Patiño and B. Kuchen are with the Institute of Automatics (INAUT), Faculty of Engineering at National University of San Juan, 5400 San Juan, Argentina. (dpatino@inaut.unsj.edu.ar).

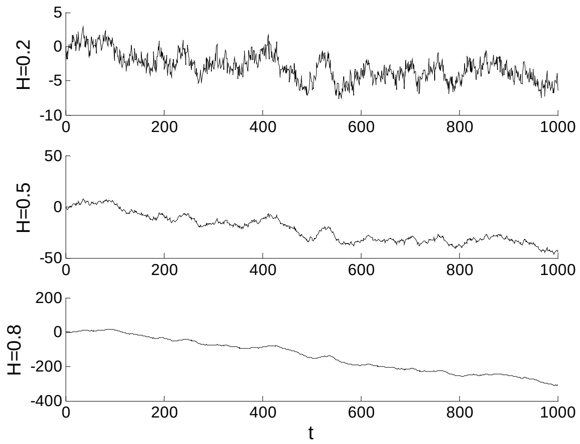

Fig. 1. Three sample path from fractional Brownian motion for three values of *H*.

C. Problem Statement

The classical prediction problem may be formulated as follow. Given past values of a process that are uniformly spaced in time, as shown by *x(n-T), x(n-2T), . . . , x(n-mT)*, where *T* is the sampling period and *m* is the prediction order, it is desired to predict the present value *x(n)* of such process. Therefore, we like to obtain the best prediction (in some sense) of the present values corresponding to a random or pseudo-random signal.

The predictor system may be implemented using either an autoregression model-based linear or a nonlinear adaptive filter, depending on whether the process is linear or nonlinear. In the second case, neural networks are used as a nonlinear model building, in the sense that smaller the prediction error is (in a statistical sense), the better the net serves as model of the underlying physical process responsible for generating the data. In this work, *time lagged feedforward networks* are used.

Thus, the present value of the signal is used as the desired response for the adaptive filter, and the past values of the signal supply as input of the adaptive filter. Then, the adaptive filter output will be the *one-step prediction signal*. In Fig. 2 is shown the block diagram of the nonlinear prediction scheme based on a neural network filter.

In this work, a prediction device is designed such that starting from a given sequence $\{x_n\}$ at time *n* corresponding to a time series, it can be obtained the best prediction $\{x_e\}$ for the following 18 values sequence.

Hence, it is proposed a predictor filter with an input vector l_x, which is obtained by applying the delay operator, Z^{-1}, to the sequence $\{x_n\}$. Then, the filter output will generate x_e as the next value, that will be equal to the present value x_n. So, the prediction error at time *k* can be evaluated as

$$e(k) = x_n(k) - x_e(k),$$

which is used for the learning rule to adjust the neural network weights.

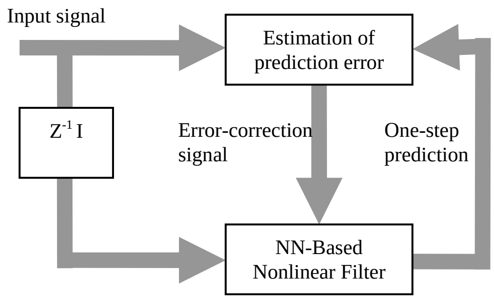

Fig. 2. Block diagram of the nonlinear prediction.

II. DESCRIPTION OF THE PREDICTION MODEL

A. NN-Based Nonlinear Autoregression Model

We propose a neural network-based nonlinear filter based on a nonlinear autoregression model [7] [8] [9]. The neural network used is a *time lagged feedforward networks* type. The neural net topology consists of l_x inputs, one hidden layer of H_o neurons, and one output neuron. The learning rule used in the learning process is based on the Levenberg-Marquardt method.

The learning rule modifies the number of patterns and the number of iterations at each time-stage according to the parameter *H*, which gives short and long term dependence of the sequence $\{x_n\}$, or from a practical point of view it gives the *ruggedness* of the time series.

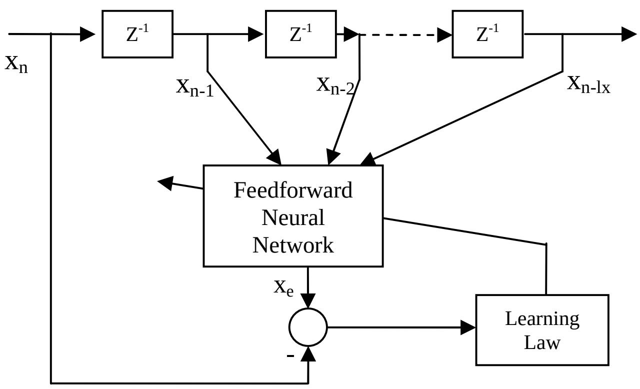

Fig. 3. Neural Network-based nonlinear predictor filter.

In order to predict the sequence $\{x_e\}$ one-step ahead, the first delay taken off from the tapped-line x_n is used as input. Therefore, the output prediction can be denoted by

$$x_e(n+1) = F_p\left(Z^{-1}I(\{x_n\})\right) \tag{1.4}$$

where, F_p is the nonlinear predictor filter operator, and $x_e(n+1)$ the output prediction at $n+1$.

B. The Proposed Learning Process

The weight of the net are adjusted based on the Levenberg-Marquardt rule, which considers the long and short term stochastic dependence of the time series measured by the Hurst's parameter H.

The *proposed learning process* consists on changing both the number of patterns and the number of iterations in function of the parameter H for each corresponding time series. The learning process is performed using a batch model. In this case the weight updating is performed after the presentation of all training examples, constituting an epoch. The pairs of the used input-output patterns are

$$(x_i, y_i) \quad i = 1,2,....,N_p \tag{1.5}$$

where, x_i and y_i are the corresponding input and output pattern respectively, and N_p is the number of input-output patterns presented at each epoch.

Here, the input vector is define as

$$X_i = Z^{-1} I\left(\{x_i\}\right), \tag{1.6}$$

and its corresponding output vector as

$$Y_i = x_i. \tag{1.7}$$

Furthermore, the index i is within the range of N_p given by

$$H_o \le N_p \le 3 \cdot l_x$$

where, H_o is the number of the hidden neurons and l_x is the dimension of the input vector.

In addition, through each epoch the number of iterations performed i_t is given by

$$1 \le i_t \le 2(H_o - 1).$$

The proposed criterion to modify the pair (i_t , N_p) is given by the statistical dependence of the time series $\{x_n\}$, supposing that is a **fBm**. The dependence is evaluated by the Hurst's parameter H, which is computed using a wavelet-based method [1] [4].

Then, a *heuristic adjustment* for the pair (i_t ,N_p) in function of H according to the membership functions shown in Fig. 4 is proposed.

Finally, the number of inputs of the nonlinear filter is tuned —that is the length of tapped-delay line, according to the following heuristic criterion: when the training process is completed, both sequences, $\{x_n\}$ and $\{\{x_n\},\{x_e\}\}$, should have the same H parameter. If the error between $H(\{x_n\})$ and $H(\{\{x_n\},\{x_e\}\})$ is grater than a threshold parameter θ the value of l_x is increased (or decreased), according to $l_x \pm 1$. Explicitly,

$$l_x = l_x + 1 \cdot \mathrm{sign}(\theta).$$

Here, the threshold θ was set about 5%.

III. MAIN RESULTS

A. Set-up of Model and Learning Process

The initial conditions for the filter and learning algorithm are shown in Table 1. The initial number of hidden neurons and iteration are set in function of the input number.

Table 1 shows the initial conditions of the learning algorithm used for forecasting the 11 time series, which sizes have a variable length, between 120 and 170 values.

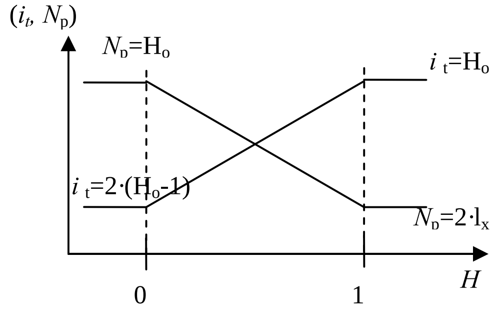

Fig. 4. Heuristic adjustment of (i_t, N_p) in terms of H.

Variable	Initial Condition
l_x	16
H_o	l_x/3.2
i_t	H_o-1
H	0.5

Table 1. Initial condition of the learning algorithm.

B. Preliminary Results Using Other Ttime Series

In order to test the proposed design procedure of the neural network-based nonlinear predictor, an experiment with sinusoidal and square signals was performed. The performance of the filter is evaluated using the mean Symmteric Mean Absolute Percent Error (SMAPE) proposed in the NN3 evaluation:

$$SMAPE_S = \frac{1}{n}\sum_{t=1}^{n} \frac{|X_t - F_t|}{(X_t + F_t)/2} \cdot 100 \tag{1.8}$$

where, t is the time observation, n is the test set size, s each time series, X_t and F_t are the actual and the forecast time series values at time t respectively.

The SMAPE of each series s calculates the symmetric absolute error in percent between the actual X_t and its corresponding forecast F_t value, across all observations t of the test set of size n for each time series s.

Fig. 5 shows the predictor nonlinear-filter response, giving the 16 future values for a sinusoidal time series. The used sine time series has a period T=0.48 s, and it is sampled at T_0=0.05 s. The initial length of the tapped-delay line was set-up at 16 taps, and at the end of the learning process got be equal to 16.

For a square time series, Fig. 6 presents the forecasted 18 values. Here the value of H, across for the complete time series $\{x_n\}$ and $\{x_e\}$, differs at a 5%. To improve the forecasting performance of the neural network filter, it is used as initial condition of l_x= 17, in order to increase H of the $\{x_e\}$. The new results are shown in Fig. 7, where the percentage is declined in the order of 2%.

The filter structure and learning parameters are adjusted at each time- sample in function of the H value. The evolution of those parameters is shown in Fig. 8, where can be note the variation of the number of used pattern in the learning process. Given that the number of iterations at each epoch is small at the beginning, there are not changes at the parameters. Then, for more learning time the computation of H can be evaluated more accurately, which can got to set the values of (N_p, i_t), as is shown in Fig. 8.

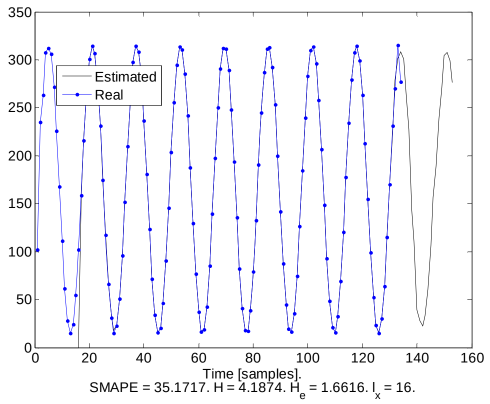

Fig. 5. Prediction of a sinusoidal time series.

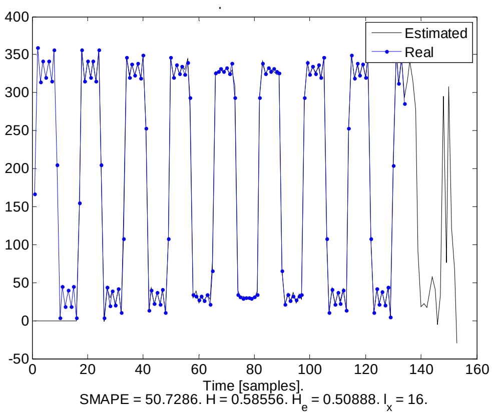

Fig. 6. Prediction of the next 18 values of a square time series.

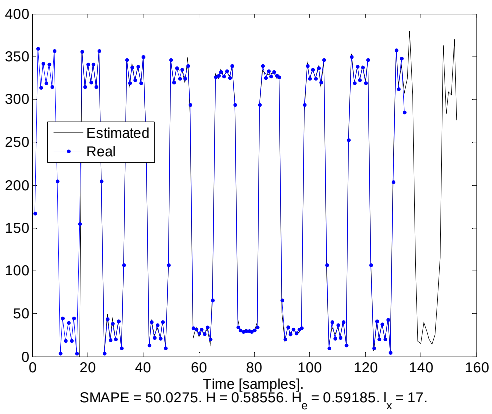

Fig. 7. Final prediction after adjusting the length of the tapped-delay line of neural network in function of the *H error*.

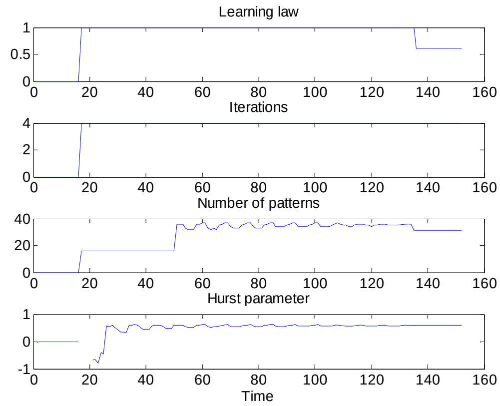

Fig. 8. Evolution of the learning parameters.

C. Main Prediction Results for the NN3 Time Series

In the following figures are shown the forecast for the time series number 2 and 4 of the 11 ones proposed in the NN3 competition.

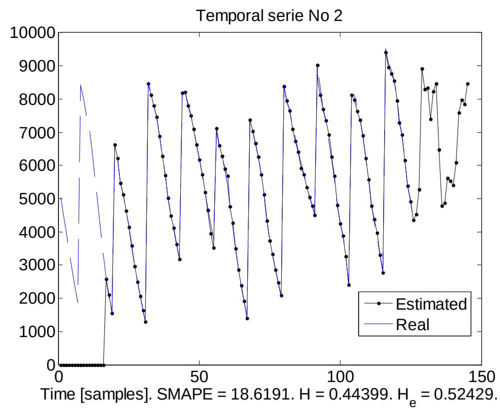

Fig. 9. Time forecast for the time series No 2.

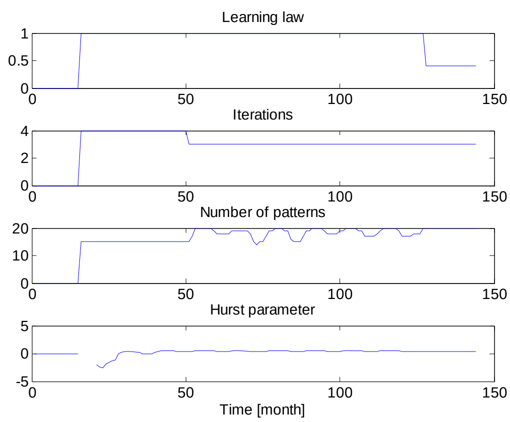

Fig. 10. Evolution of the learning algorithm's parameters.

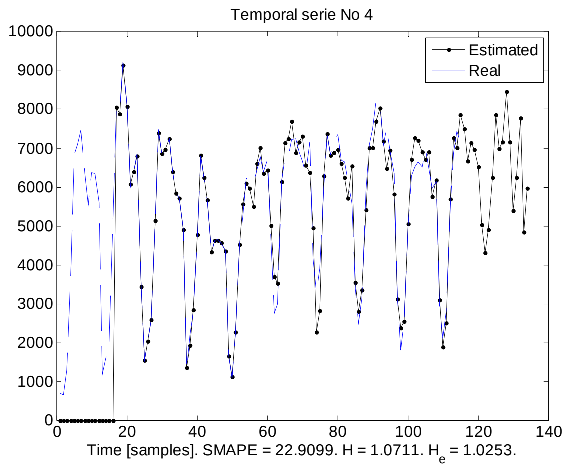

Fig. 11. Time series number 4 with *H* nearly 1.

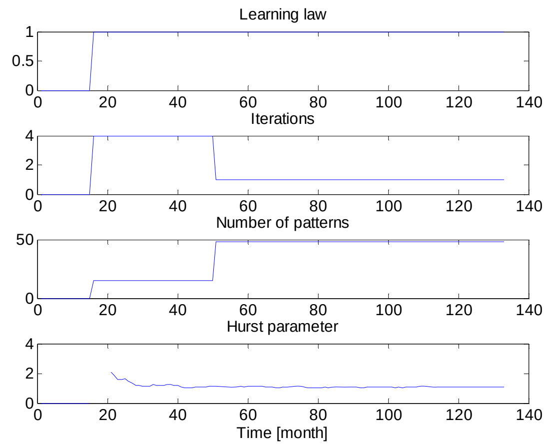

Fig. 12. Evolution of the learning algorithm's parameters.

D. Main Results

The performance of the neural network-based predictor filter is evaluated through the SMAPE index, Eq. (1.8), across the 11 time series given in the NN3 competition. Fig. 13 shows the evolution of the SMAPE index for a *traditional neural network filter*, which uses a learning algorithm with fixed parameters. And another named *modified neural network filter*, which is proposed in this work and use the H parameter to adjust heuristically either structure of the net or parameters of the learning rule.

IV. Concussion

In this work a feedforward neural networks-based nonlinear autoregression (NAR) filter for forecasting time series has been presented. The learning rule proposed to adjust the neural net weights is based on the Levenberg-Marquardt method. And in function of the long and short term stochastic dependence of the time series, evaluated by the Hurst parameter H, an on-line heuristic adaptive law is proposed to update the neural net topology, number of input taps, and the number of patterns and iterations at each time-stage. The main results shows a good performance of the predictor system applied to the 11 time series, proposed in the NN3 competition, due to we obtained similar roughness for both the original and the forecast time series, evaluated by H and H_e respectively. These results encourage one to go on working with this new learning algorithm, applying to other neural network models, duo to the time series generated by humans interaction presents short and long term stochastic dependence.

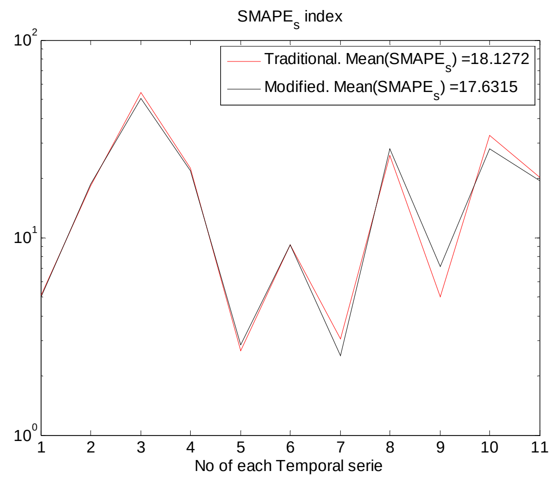

Fig. 13. The SMAPE index applied over the 11 time series.

References

[1] Abry, P.; P. Flandrin, M.S. Taqqu, D. Veitch (2003), "Self-similarity and long-range dependence through the wavelet lens," *Theory and applications of long-range dependence*, Birkhäuser, pp. 527-556.

[2] Bardet, J.-M.; G. Lang, G. Oppenheim, A. Philippe, S. Stoev, M.S. Taqqu (2003), "Semi-parametric estimation of the long-range dependence parameter: a survey," *Theory and applications of long-range dependence*, Birkhäuser, pp. 557-577.

[3] Dieker, T., 2004. Simulation of fractional Brownian motion. MSc theses, University of Twente, Amsterdam, The Netherlands.

[4] Flandrin, P. (1992), "Wavelet analysis and synthesis of fractional Brownian motion," IEEE Trans. on Information Theory, 38, pp. 910-917.

[5] Istas, J.; G. Lang (1994), "Quadratic variations and estimation of the local Hölder index of a Gaussian process," Ann. Inst. Poincaré, 33, pp. 407-436.

[6] Mandelbrot, B. B., (1983). The Fractal Geometry of Nature, Freeman, San Francisco, CA.

[7] Haykin, S (1999), Neural Networks: A comprehensive Foudation, 2nd Edition, Prentice Hall.

[8] Mozer, M. C., (1994). "Neural Net Architectures for Temporal Sequence Processing." A. S. Weigend and N. A. Gershenfeld, eds., Time Series Predictions: Forecasting the Future and Understanding the Past, pp. 243-264. Reading, M.A.: Addison-Wesley.

[9] Zhang, G.; B.E. Patuwo, and M. Y. Hu, (1998). "Forecasting with artificial neural networks: The state of art," J. Int. Forecasting, vol. 14, pp. 35-62.