Time Series Forecasting using Recurrent Neural Networks and Wavelet Reconstructed Signals

Angel Garcia-Pedrero and Pilar Gomez-Gil

***Abstract*— In this paper a novel neural network architecture for medium-term time series forecasting is presented. The proposed model, inspired on the Hybrid Complex Neural Network (HCNN) model, takes advantage of information obtained by wavelet decomposition and of the oscillatory abilities of recurrent neural networks (RNN). The prediction accuracy of the proposed architecture is evaluated using 11 economic time series of the NN5 Forecasting Competition for Artificial Neural Networks and Computational Intelligence, obtaining an average SMAPE of 27%. The proposed model shows a better mean performance in time series prediction of 56 values than a feed-forward network and a fully recurrent neural network with a similar number of nodes.** [1]

I. INTRODUCTION

Forecasting consists of the evaluation of future values of a time series, x_{t+h}, $h \geq 1$ based on the observations of its past values $x_1, x_2, \ldots, x_t$ [1]. This is a problem with a wide range of applications in control, signal processing, meteorology and economy. Although several methods have been developed to obtain accurate predictions, nowadays, forecasting is still an open problem. Neural networks are a popular method to model temporal processes, because of their ability to capture essential functional relationships among data. Particularly in some prediction applications [2], among the best performances have been obtained by RNNs.

In the last years, an important number of works related to the use of RNNs for time series prediction have been published. For example, Cai et al. [3] proposed a learning algorithm based on particle swarm optimization and evolutionary algorithms (PSO-EA). That algorithm is employed to train an Elman network to predict time series and tested with the CATS Benchmark, an artificial time series containing 5,000 values. CATS contains 100 missing values distributed in 5 sections. The forecasting of the missing values is made using 5 networks; each network predicts 20 values. The prediction of all missing values with respect to real values has a mean squared error (MSE) of 351. Beliave et al. [4] implemented a chaotic neural network based on a biological model called K_{III} set. This network, able to oscillate in a complex non-periodic way, is trained using a Hebbian learning rule. The results of the network are combined with nearest neighbors to predict missing values of CATS Benchmark. Training set is composed by 45 input samples, each containing 10 values of the time series. The obtained predictions have a MSE of 73. Hybrid complex neural network (HCNN) is a recurrent neural model for long-term time series prediction proposed by Gomez-Gil [5]. HCNN uses Fourier analysis to get information about time series behavior to train several harmonic generators. A harmonic generator is a fully connected neural network with the ability of generating sinusoidal series in a autonomous way. Each harmonic generator is connected to other recurrent neurons using feed-forward connections. This network is trained using backpropagation through time algorithm. The capacity of HCNN for time series forecasting is tested using 512 values of an electrocardiogram. The results point out a MSE of $2.5E^{-3}$

The main key points involved with neural networks for time series prediction include data preprocessing, learning algorithms, neural architecture design/modifications and parameter selection. In this paper, we propose a novel neural architecture, inspired on the previous work of the HCNN [1], that takes advantage of both the time-frequency information obtained by wavelet decomposition of the training signal and the oscillatory abilities of recurrent neural networks.

This paper is organized as follows: Section II introduces key concepts about the design of the proposed neural architecture. Section III presents a description of time series employed. Some results are presented in Section IV. Finally, Section V presents some conclusions.

II. FORECASTING MODEL

A. Recurrent Neural Networks (RNN)

A RNN may be defined as:

$$\begin{aligned} y_j(t+1) &= \varphi\Big(\sum_{i=1}^{m+n} w_{ji} z_i(t)\Big) \qquad (1) \\ z_i(t) &= \begin{cases} y_i(t), & i \leq n \\ u_{i-n}, & i > n \end{cases} \end{aligned}$$

where m is the number of inputs, n is the number of hidden and output neurons, $\varphi(.)$ is an arbitrary differential function, commonly a sigmoid, y_j denotes the output of the j-th neuron and w_{ji} the connection from i-th to j-th neuron. The external inputs u_i and recurrent inputs y_i are represented as z_i for convenience.

There are many RNN models; some examples include the proposed by Elman, Hopfield and the fully-connected recurrent neural networks. RNNs are able to create a rich

Both authors are with the National Institute of Astrophysics, Optics and Electronics, Computational Science Department, Luis Enrique Erro 1, Tonantzintla, Puebla, Mexico. {pgomez, agarciapedrero}@ccc.inaoep.mx

[1]This paper will be Published at: IEEE Proceedings of the 20th. International Conference on Electronics, Communications and Computers. CONIELECOMP 2010. Indexed at IEEE Explore, see www.ieee.org

representation about past information [6], so they are selected for this work.

B. Discrete Wavelet Decomposition

Discrete Wavelet Transform (DWT) can break down a signal into many lower resolution components using a wavelet function. Decomposition consists of filtering and decimating operations. During decomposition, low-pass (L) and high-pass (H) filters are applied over the signal. L and H filters are known as decomposition filters and their outputs are commonly called approximation (cA) and detail (cD) coefficients, respectively. This process is also known as wavelet decomposition tree.

A wavelet decomposition tree produces signals that may contain important information about the behavior of the original signal. The decomposition process may be applied iteratively; the level of decomposition applied to a signal depends on the specific problem to be tackled. Fig. 1 shows a wavelet decomposition tree with three levels.

Fig. 1. Wavelet decomposition tree with 3 levels

This reconstruction process, also known as Inverse Discrete Wavelet Transform (iDWT), is performed using undecimating and filtering. Undecimating consists of inserting zeros between values of the input signal; the filtering operation is carried out using synthesis filters based on decomposition filters. The reconstructed signal is obtained by adding the outputs of synthesis filters. Fig. 2, shows the reconstruction process; inputs to L' and H' synthesis filters are the approximation and detail coefficients, respectively.

Fig. 2. Reconstruction process of a signal.

The original signal can be reconstructed from approximation and detail coefficients without losing information. Also, it is possible to reconstruct approximation and detail signals using their coefficients. For example, an approximation signal can be reconstructed using cA_1 (see Fig. 1) and replacing cD_1 coefficients by zeros. Approximation signals contain low-frequency information and so more information about long-term behavior. On the other hand, detail signals have high-frequency information that denotes short-term changes.

C. Proposed Architecture

The proposed architecture in this work contains one input layer, two hidden layers and one output layer (see Fig. 3). The input layer is composed by 5 neurons and is directly connected to both hidden layers. The first hidden layer is formed by three fully RNNs. Each RNN receives 5 inputs from the input layer and produces one output. The input values correspond to five values coming from the reconstructed embedded system of the original signal. The second hidden layer is fully recurrent; it has 10 neurons and receives inputs coming from the output neurons of each RNN located at first hidden layer and from the input layer. The output layer has only one neuron and receives inputs only from the second hidden layer. This output neuron represents the prediction of one point ahead of the reconstructed embedded system of the signal. Hyperbolic tangent is used in the network as activation function.

Fig. 3. Proposed Architecture.

The aim of the fully recurrent sub-networks located at the first hidden layer is to represent time-frequency relationships that compose the original signal. To do this, each sub-network is trained to predict one point ahead of reconstructed signals obtained from wavelet decomposition and reconstruction. Daubechies (db) wavelet function set is employed to decompose and to reconstruct these signals. Reconstructed signals were obtained using coefficients cA_3, cD_3, cD_2 and cD_1 (see Fig. 1) labeled A, B, C and D respectively. To reduce the total number of neurons in the proposed architecture and consequently training time, only three reconstructed signals out of the four are used to train three sub-networks. All possible sets with combinations of three reconstructed signals are created, and the best combination is selected as follows: for each set, their signals are added and this result is subtracted from the original signal to obtain the Mean Square Error (MSE) of the combination. The combination with the smallest MSE is selected. The number of nodes of each sub-network is experimentally determined evaluating its capacity to forecast 56 values ahead of a reconstructed signal. Real time real learning based on extended Kalman filter (RTRL-EKF) algorithm [7] is employed to train the sub-networks. The training is made around 500 epochs.

After sub-networks are trained, they are integrated into the whole architecture and their weights are set fixed. Then, the complete architecture is trained about 50 epochs using RTRL-EKF algorithm.

During the prediction phase, the inputs to the architecture at each time are the outputs of the architecture at previous time (known as recurrent prediction). Therefore, external inputs are not necessary during this process, except for a set the initial conditions defined during training phase. Also, it must be noticed that in this phase, sub-networks work as autonomous generators being fed each time with their own outputs at previous times.

III. DATA PREPROCESSING

The forecast ability of this architecture was tested using a prediction problem proposed at the website of the NN5 Forecasting Competition [8]. This dataset contains 111 daily valued time series. Fig. 4 shows the first time series of NN5 dataset. Each time series has 791 values and may present missing values, outliers and/or seasonality. For the experiments presented here, missing values were substituted by the mean of the two nearest neighbors, whose values were known.

Due to the fact that the activation function of this neural network is defined for $(-1, 1)$, training data is scaled into this interval to improve learning. As suggested by Crone [9] for this class of data, linear scaling was employed, which is defined as:

$$Z_t = lb + \frac{x_t - \min(X)}{\max(X) - \min(X)}(ub - lb) \qquad (2)$$

where lb and ub are lower and upper bounds of interval in the data to be scaled and x_t represents a point of time series X to be scaled.

IV. RESULTS

Several experiments were done using a subset of 11 series taken from NN5 dataset [8]. Such subset was chosen according with the rules given by the NN5 competition, and it is used to compare the performance of the proposed model with results published by the competition website and with other two neural net models. For each time series and each neural model, 12 experiments were executed, each with different random initial weights, but the same corresponding neural model configuration. The first 635 values of each series were employed to train all models. The next 56 values were used as a testing set to compare the performance of the proposed architecture with respect to other two models: a fully connected recurrent neural network and a three-layer feed-forward neural network. The last 56 values of the series were used as a validation set to compare the performance of this architecture with respect to the competition results.

Fig. 4. The first time series of NN5 dataset.

Prediction performance is evaluated using two error measures: MSE and Symmetric Mean Absolute Percentage Error (SMAPE). SMAPE is employed in NN5 time series forecasting competition as the criteria to determine the winner. SMAPE is defined as:

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|x_t - \hat{x}_t|}{(x_t + \hat{x}_t)/2}\right)(100\%) \qquad (3)$$

where x_t is the real value, $\hat{x}_t$ is the forecasted value and n is the number of predicted values.

Both the fully-connected and the feed-forward networks used to compare results have 5 neurons in the input layer, 26 neurons in the hidden layer and one neuron as output layer. One network for each model and each time series is trained and tested. The performance of each model for all time series is calculated using the mean of MSE and SMAPE for all the 12 prediction experiments obtained by the three models. It must be pointed out that, before the measurements are calculated, the predictions are linearly re-scaled to the original signal interval.

The results obtained by the proposed and the other two models are analyzed with respect to the best case, worst case and average behavior. Best and worst cases are selected in terms of the lowest and highest values respectively, of the mean SMAPE gotten by the proposed network. For best and worst cases, the wavelet db10 function was used to build reconstructed signals A, B and C, that were employed to train sub-networks with 11, 8 and 4 neurons in their hidden layers, respectively.

Table I shows the mean performances of the three network models obtained using the time series that best predicted the proposed architecture. Table II shows the mean performances obtained using the time series that was predicted worst by the proposed architecture. Table III shows the average of all experiments using all time series. Also, Fig. 5 shows a plot of the results of the experiment with best predictions and compared to real values of the testing set. In the other hand, Fig. 6 shows the results of the experiment with the worst predictions.

Fig. 5. Predictions of the three models for the best case.

It may be observed that, in the worst case, the predictions of the proposed architecture are better than feed-forward and fully recurrent neural network models.

Fig. 6. Predictions of the three models for the worst case.

In the three tables may be observed that the proposed architecture obtains better MSE and SMAPE that the feed-forward and fully recurrent models. In average, the proposed network obtained a MSE of 34.0 ± 20.1 compared to the MSE of 198.7 ± 131.1 obtained by the fully recurrent and the MSE of 250.1 ± 226.0 obtained by the feed-forward model. In order to compare results with the NN5 forecasting competition, the following was done: for each time series, the proposed architecture getting the lowest SMAPE over the testing set was selected; using that model, a new SMAPE using the validation test was obtained; after that, the average SMAPE of the 11 time series was calculated getting 28.5%. This value locates the proposed model between the 25th. and 26th. place in the NN5 forecasting competition, with respect to the rank of statistical and neural networks methods. This result is located between the 16th and 17th place in the rank of neural networks and computational intelligence methods [8].

V. CONCLUSIONS

A new neural network architecture that is able to predict future values of time series, using a recurrent prediction fashion is presented. The proposed neural network is based on recurrent neural sub-networks that are able to learn reconstructed signals built by wavelet coefficients, and then combine them using a recurrent hidden layer and a feed-forward output layer. The performance of the proposed architecture was compared with other two neural models: a feed-forward and a fully recurrent neural network using 11 series of NN5 competition. The obtained results point out that the proposed method is better than the three other analyzed models in the prediction of 56 values of the time series previously mentioned.

TABLE I

PERFORMANCE OF THE THREE NEURAL MODELS FOR THE BEST PREDICTION CASE

Network	MSE	SMAPE
Feed-forward	240.7 ± 493.7	$32.6 \pm 25.0\%$
Fully recurrent	224.5 ± 198.0	$46.7 \pm 23.9\%$
Proposed model	31.5 ± 3.5	$20.6 \pm 23.9\%$

TABLE II

PERFORMANCE OF THREE NEURAL MODELS FOR THE WORST PREDICTION CASE

Network	MSE	SMAPE
Feed-forward	187.6 ± 114.1	$62.2 \pm 21.7\%$
Fully recurrent	107.9 ± 64.3	$53.9 \pm 14.1\%$
Proposed model	60.3 ± 91.8	$40.5 \pm 25.5\%$

TABLE III

AVERAGE PERFORMANCE OF THE THREE NEURAL MODELS

Predictors	MSE	SMAPE
Feed-forward	250.1 ± 226.0	$49.3 \pm 12.4\%$
Fully recurrent	198.7 ± 131.1	$60.8 \pm 13.0\%$
Proposed model	34.0 ± 20.1	$27.2 \pm 8.3\%$

REFERENCES

[1] P. Gomez-Gil, "Long term prediction, chaos and artificial neural networks. Where is the meeting point?" *Engineering Letters*, vol. 15, no. 1, August 2007.

[2] A. Lendasse, E. Oja, O. Simula, and M. Verleysen, "Time series prediction competition: The CATS benchmark," *Neurocomputing*, vol. 70, no. 13-15, pp. 2325–2329, 2007.

[3] X. Cai, N. Zhang, G. K. Venayagamoorthy, and D. C. W. II, "Time series prediction with recurrent neural networks trained by a hybrid pso-ea algorithm," *Neurocomputing*, vol. 70, no. 13-15, pp. 2342–2353, 2007.

[4] I. Beliaev and R. Kozma, "Time series prediction using chaotic neural networks on the cats benchmark," *Neurocomputing*, vol. 70, no. 13-15, pp. 2426–2439, 2007.

[5] P. Gomez-Gil and M. Ramirez-Cortes, "Experiments with a hybrid-complex neural networks for long term prediction of electrocardiograms." Proceedings of the IEEE 2006 International World Congress of Computational Intelligence, IJCNN 2006. Vancouver. Canada. pp. 4078-4083, 2006.

[6] D. P. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms,Architectures and Stability*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[7] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, July 1998.

[8] S. F. Crone, "NN5 forecasting competition for artificial neural networks & computational intelligence," 2008, avaible in http://www.neural-forecasting-competition.com/. Last review on march 2009.

[9] S. F. Crone, J. Guajardo, and R. Weber, "The impact of preprocessing on support vector regression and neural networks in time series prediction," in *DMIN*, 2006, pp. 37–44.