Electronic System for Chaotic Time Series Prediction Associated to Human Disease

A.D. Pano-Azucena, E. Tlelo-Cuautle
Electronics Department
INAOE. Puebla, México.
Email: ana.dalia.p.a@gmail.com, etlelo@inaoep.mx

Sheldon X.-D. Tan
Electrical Engineering
University of California Riverside, Riverside CA.
Email: stan@ee.ucr.edu

***Abstract*—It is well-known that a large number of natural phenomena exhibit chaotic behavior, e.g. brain activity, mental illness, bioelectric signals, pancreatic beta cell, and so on. That way, researchers have the challenge to develop systems that guarantee the prediction of chaotic time series, so that it can be used to prevent the activation of an epileptic attack or other human disorder. In this paper we show the usefulness of the multilayer perceptron (MLP), which is in the family of artificial neural networks, to predict chaotic time series, which in this research paper were obtained from real chaotic systems based on saturated nonlinear function series and from the Rössler system. We highlight the hardware implementation of the prediction system that is verified by using a field-programmable gate array (FPGA). The root-mean-square error is provided to show the suitability of the proposed electronic system.**

I. Introduction

Many physical and biological phenomena have chaotic behavior. In engineering, the word chaos or chaotic is associated to a complex dynamical behavior that has special characteristics, such as extreme sensitivity to small variations in the initial conditions from which the system is turned on, and other characteristics related to physics and mathematics, namely: presence of at least of one positive Lyapunov exponent, a continuous power spectra, and the system is determinist because one knows its evolution and all its parameters are well-defined.

The development of chaotic systems is nowadays very popular and they can be implemented with either or both analog or digital electronics. Every chaotic system can be implemented with electronic devices to verify the behavior of the mathematical models that describe different chaotic phenomena in different areas of research, such as: physics, chemistry, hydrology, neurologic, human behavior and so on [1, 2, 3]. Further, if one is interested to know the future evolution of the chaotic time series provided by these dynamical systems, one can simulate large times, but if one do not know the mathematical model of the system, then the challenge is how to predict future data of the chaotic dynamics from experimental data.

In this manner, this paper describes the application of a multilayer perceptron (MLP) to predict chaotic time series that are taken from real chaotic systems based on saturated nonlinear function (SNLF) series [4], and from the Rössler system [5]. Those chaotic time series are saved and are taken as inputs to a MLP that has the capability of predict the future behavior of the data, so that the proposed electronic system can be used to predict future evolution of human disorders that have chaotic behavior.

Some similar works for time series prediction have been already developed by applying artificial neural networks and MLP, see for instance [6, 7, 8]. Also, other time series prediction techniques are available, but they are mostly suitable for software systems, such as: fuzzy systems [9, 10, 11], support vector machines [12, 13], hybrid systems [14, 15, 16], and so on. Since we are interested to provide an ambulatory system of low weight, then we introduce the idea on the development of an electronic system that is verified herein by using a field-programmable gate array (FPGA). The system is based on a MLP to take advantage of its suitability to perform parallel processing and the operations are simple mathematics that are very easy to describe into an FPGA.

Section II describes the two chaotic systems already implemented into an FPGA from which we take the chaotic time series. Section III details the design of the MLP, where three topologies are described. Section IV shows the electronic implementation of the three MLPs by using an FPGA. Also, the prediction of the three MLPs are shown by measuring the root-mean-square error (RMSE) to demonstrate their suitability for chaotic time series prediction. Finally, the conclusions are listed in Sect. V.

II. Generation of chaotic time series

This section shows the generation of chaotic time series by using two chaotic systems, namely: Rössler and the chaotic oscillator based on saturated nonlinear functions (SNLF) series. They are modeled by the dynamical equations of third order shown in (1) and (2), respectively. These chaotic systems have been widely studied and used generate chaotic behavior, and their associated mathematical descriptions consist of three state variables (x_1, x_2 and x_3), real coefficients, and functions that include the nonlinearity. Both chaotic oscillators are autonomous and their numerical simulation requires initial conditions. In the case of the Rössler chaotic system given by (1), the simulation is performed by setting the parameter values: $a = b = 0.2$ and $c = 5.7$ [5]. Figure 1 shows the state variable x_1 and this

2575-2634/18/$31.00 ©2018 IEEE
DOI 10.1109/ICHI.2018.00044

time series is used to test the electronic system for chaotic time series prediction, as shown in the following sections. The chaotic oscillator based on SNLF series described by (2), can be simulated by setting the parameter values to: $a = b = c = d = 0.7$, and the nonlinear function $f(x_1)$ has three linear segments as shown in [4], to generate a double-scroll attractor. The time series corresponding to the state variable x_1 is also shown in Fig. 1.

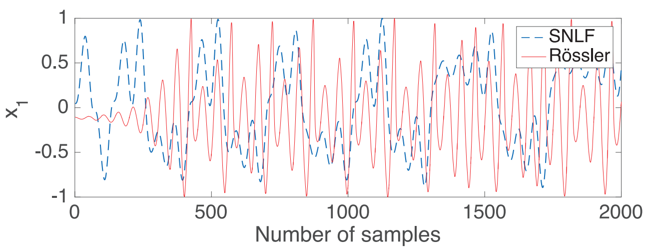

Figure 1. Generation of chaotic time series from the Rössler chaotic oscillator and the one based on SNLF series.

$$\begin{aligned} \dot{x_1} &= -x_2 - x_3 \\ \dot{x_2} &= x_1 + a x_2 \\ \dot{x_3} &= b + x_3(x_1 - c) \end{aligned} \quad (1)$$

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= x_3 \\ \dot{x_3} &= -a x_1 - b x_2 - c x_3 + d f(x_1) \end{aligned} \quad (2)$$

III. Design of the MLP for chaotic time series prediction

Artificial neural networks are mathematical tools originally inspired by the way in which the human brain processes information. It consists of a set of elementary processing units called neurons or nodes whose processing capability is stored in the connections by synaptic weights, and whose adaptation depends on learning [17]. There are three kinds of neurons: input (allocate input values), hidden (perform operations and consists of one or more layers), and output ones (perform operations and compare the values with target or reference ones). Figure 2 shows the basic neuron structure where x_j represents the input signals, w represents the synaptic weights, b is the bias and $f(.)$ denotes the activation function [17, 18]. The final state is evaluated by (3), while in this work the lineal (l) and hyperbolic tangent (h) are used as function f. Figure 3 shows an example of a MLP with 6 layers, and the description of each layer.

$$y = f(u) = f\left(\sum_{j=1} x_j w_j + b\right) \quad (3)$$

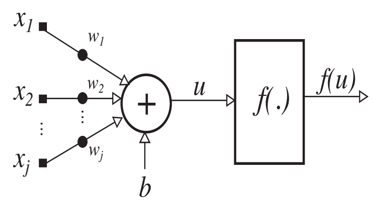

Figure 2. Functional structure of a neuron.

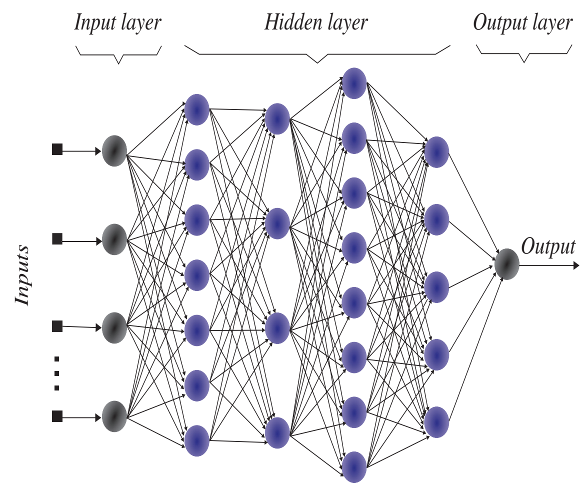

Figure 3. MLP consisting of five layers taken from [19].

There is not a general rule to choose the number of layers and the number of neurons in each layer, so that one must explore all combinations of connections and also all possibilities of values of each parameter and activation function. Some authors recommend to use one hidden layer with low number of neurons. However, that is a good option just for simple problems, but for complex ones like in this research paper, one must explore different topologies and parameter values of the neurons. For instance, Table I lists three MLP topologies, two of them designed by applying the geometric pyramid rule [20], and the other one taken from [19]. These three MLPs are compared according to the root-mean-square error (RMSE) to see the prediction capability for the chaotic time series of the two chaotic oscillators described above. That way, the prediction of the chaotic time series from the simulation of (1) and (2), is performed only for the state variable $x_1(t)$, and then we predict 6-steps ahead $x_1(t-6)$, 12-steps ahead $x_1(t-12)$, and 18-steps ahead $x_1(t-18)$.

Among the training algorithms available into MATLAB^{TM}, the levenberg-marquardt algorithm (trainlm) is applied herein. The training is executed using 3 subsets of data. The first one (training) computes the gradient, weights and bias updating. The second subset (validation) monitors the error during the training. The third subset

Table I
CHARACTERISTICS OF THREE MLPS: 5 AND 6 LAYERS DESIGNED BY APPLYING THE GEOMETRIC PYRAMID RULE [20], AND THE 6 LAYERS TAKEN FROM [19].

Characteristics	**Geometric pyramid rule**		**MLP [19]**
Number of layers	5	6	6
Number of neurons	16-8-4-2-1	32-16-8-4-2-1	4-7-4-8-5-1
Activation function	*h-h-h-h-l*	*h-h-h-h-h-l*	*h-h-h-h-h-l*
Learning algorithm	Levenberg-Marquardt algorithm		

(test) adjust the error during the validation process. Such processes are executed into Matlab.

From the results provided by Matlab, one can choose the best MLP topology and then one must adjust the weights and biases of the neurons, so that the training of the MLP is initialized n times with different weights and biases assigned randomly. Figure 9 shows the summary of the three MLPs when they are compared with respect to the RMSE obtained during the stage of validation. As one sees, both MLPs consisting of 6 layers have low RMSE. In this manner, once the MLP topology has been trained, one choose the set of weights and biases to proceed to the hardware implementation.

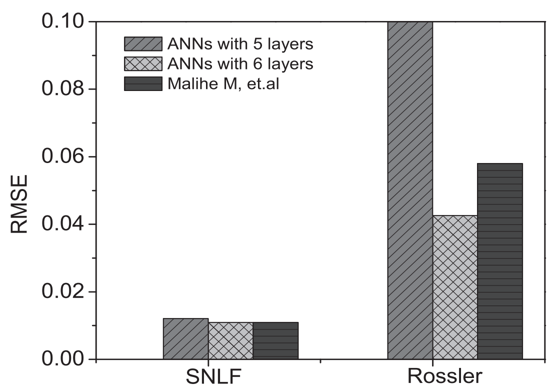

Figure 4. Performance of the three MLPs listed in Table I in terms of RMSE for predicting the two chaotic time series from Fig. 1.

Figures 5 and 6 highlight the prediction obtained during the validation process for each chaotic oscillator by using the MLP topology provided in [19].

IV. HARDWARE IMPLEMENTATION OF THE MLP FOR CHAOTIC TIME SERIES PREDICTION

In computer arithmetic, the performance of a system depends on the representation of the numerical values, and also on the mathematical operations that are algebraic for the case of implementing a MLP topology [21]. In this research paper we use the numerical representation known

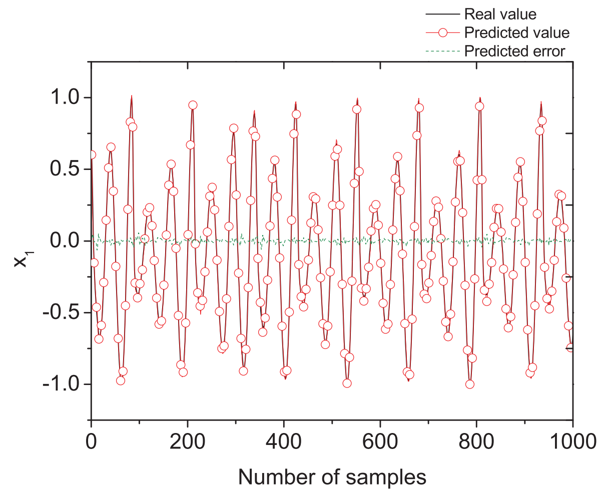

Figure 5. Prediction of the time series generated by Rössler system.

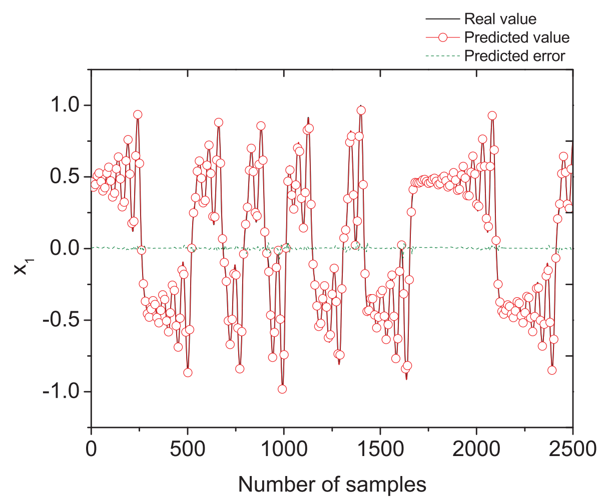

Figure 6. Prediction of the time series generated by the chaotic system based on SNLF series.

as fixed-point and the corresponding format is in the form of 4.28. It means that we use 1 bit to represent the sign of the numerical value, 3 bits to represent the integer part of a numeric value, and 28 bits to represent the fractional part to process a resolution of the numbers that generate a low rounding error. In electronics, and from Fig. 2, one can see the kind of digital hardware that is required to implement the topology. In this case, the neuron consists of a multiplication block, and adder block, and a block that is designed to perform the transfer function of type hyperbolic tangent h. This is sketched in Fig. 7. The transfer function known as hyperbolic tangent can be approached by piecewise-linear (PWL) functions, as shown in Fig. 8, as described in [22]. In this Figure one can see a very low error between the original hyperbolic tangent function and its PWL approximation, so that we proceed to implement this function with digital hardware, i.e. FPGA.

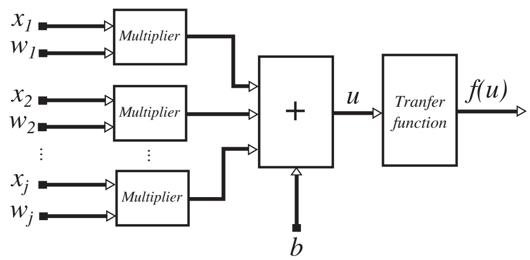

Figure 7. Synthesis of the functional structure of the artificial neuron shown in Fig. 2.

Once the MLP is implemented in hardware, and synthesized into an FPGA, the chaotic time series prediction can be performed as follows: In the case of the Rössler chaotic system given by (1), the first 1000 samples are used for training the MLP and the following 1000 samples are used for testing the prediction. For the chaotic oscillator based on SNLF series described by (2), the first 2500 samples are used for training and the following 2500 samples are used for testing. In both cases, the time series from the state variable x_1 is considered for prediction. Table II shows a comparison of the time series prediction by using the activation function implemented with the original hyperbolic tangent function, and the prediction obtained by using its PWL approximation shown in Fig. 8.

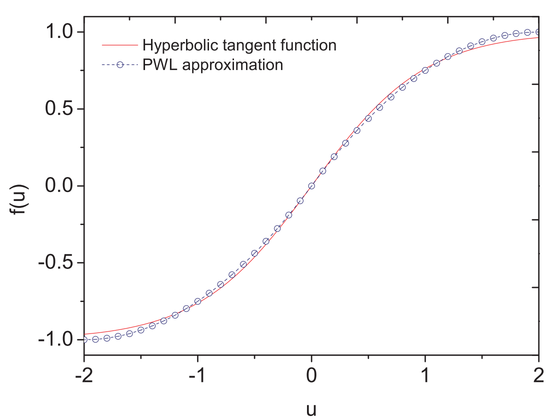

Figure 8. Approximation of the hyperbolic tangent function.

The MLP is connected to a personal computer (PC) to process the chaotic time series, as shown in Fig. 9, where one can observe the block descriptions of our proposed electronic system for time series prediction. The PC sends data of the chaotic time series to the FPGA-based prediction system,

Table II
RMSE OBTAINED DURING THE VALIDATION STAGE.

	SNLF system		
Function	5 layers	6 layers	**[19]**
Hyperbolic tangent	0.0098	0.00950	0.00951
PWL approximation	0.01120	0.01150	0.01149
	Rössler system		
Function	5 layers	6 layers	**[19]**
Hyperbolic tangent	0.09000	0.02323	0.02842
PWL approximation	0.10000	0.04400	0.05400

which process the data through the MLP and then provides as output the predicted time series with 6-steps ahead. The connection between the PC and the MLP is done through the serial port [23], where each byte is transmitted bit by bit. The data is analyzed by Matlab, and a state machine controls the flow of information.

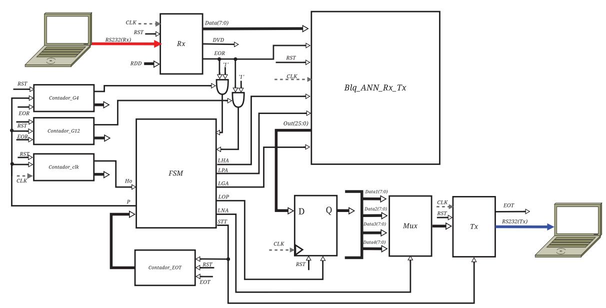

Figure 9. Prototype of the proposed chaotic time series prediction system.

V. Conclusion

The prediction of chaotic time series is a challenge to provide long-term estimation. This paper showed the usefulness of artificial neural networks, in the case of multilayer perceptron (MLP) to predict chaotic time series with good accuracy. In this case, we used three MLP topologies that were tested with two kinds of chaotic data generated from two oscillators: Rössler and a chaotic oscillator based on saturated nonlinear function (SNLF) series. All cases were compared with respect to the root-mean-square error (RMSE), and at the end, we showed the proposed electronic system that is implemented by using a field-programmable gate array (FPGA) and connected to a personal computer (PC) to feed the chaotic time series data. As a conclusion, and according to the RMSE values, we conclude that our proposed prototype exhibited a high performance in prediction of chaotic time series.

Acknowledgment

This work has been partially supported by CONACyT/México under grant 237991 and UC-MEXUS under grant CN-16-161.

References

[1] M. Moghtadaei, M. R. H. Golpayegani, and R. Malekzadeh, "Periodic and chaotic dynamics in a map-based model of tumor–immune interaction," *Journal of theoretical biology*, vol. 334, pp. 130–140, 2013.

[2] P. Jafari Moghadam Fard, M. H. Moradi, and S. Saber, "Chaos to randomness: distinguishing between healthy and non-healthy lung sound behaviour," *Australasian Physical & Engineering Sciences in Medicine*, vol. 38, no. 1, pp. 47–54, 2015.

[3] S. Raiesdana and S. M. H. Goplayegani, "Study on chaos anti-control for hippocampal models of epilepsy," *Neurocomputing*, vol. 111, pp. 54–69, 2013.

[4] V. H. Carbajal-Gómez, E. Tlelo-Cuautle, and F. V. Fernández, *Circuit Realization of the Synchronization of Two Chaotic Oscillators with Optimized Maximum Lyapunov Exponent*. Cham: Springer International Publishing, 2016, pp. 627–651.

[5] F. Zhang, "Dynamical behaviors of a generalized Lorenz family," *Discrete Continuous Dynamical Systems - B*, vol. 22, p. 3707, 2017. [Online]. Available: http://aimsciences.org//article/id/1c86bb80-d4a0-45ce-8b4d-3fd2dbffc6c7

[6] N. Stamatis, D. Parthimos, and T. M. Griffith, "Forecasting chaotic cardiovascular time series with an adaptive slope multilayer perceptron neural network," *IEEE transactions on biomedical engineering*, vol. 46, no. 12, pp. 1441–1453, 1999.

[7] M. Ishikawa and T. Moriyama, "Prediction of time series by a structural learning of neural networks," *Fuzzy Sets and Systems*, vol. 82, no. 2, 1996.

[8] J. M. P. Menezes and G. A. Barreto, "Long-term time series prediction with the NARX network: an empirical evaluation," *Neurocomputing*, vol. 71, no. 16, pp. 3335–3343, 2008.

[9] F. M. Pouzols, A. Lendasse, and A. B. Barros, "Autoregressive time series prediction by means of fuzzy inference systems using nonparametric residual variance estimation," *Fuzzy Sets and Systems*, vol. 161, no. 4, pp. 471–497, 2010.

[10] F. Di Martino, V. Loia, and S. Sessa, "Fuzzy transforms method in prediction data analysis," *Fuzzy Sets and Systems*, vol. 180, no. 1, pp. 146–163, 2011.

[11] P. Singh and B. Borah, "High-order fuzzy-neuro expert system for time series forecasting," *Knowledge-Based Systems*, vol. 46, pp. 12–21, 2013.

[12] U. Thissen, R. Van Brakel, A. De Weijer, W. Melssen, and L. Buydens, "Using support vector machines for time series prediction," *Chemometrics and intelligent laboratory systems*, vol. 69, no. 1, pp. 35–49, 2003.

[13] K.-j. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1, pp. 307–319, 2003.

[14] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks," *Neurocomputing*, vol. 73, no. 13, pp. 2540–2553, 2010.

[15] Y. Bodyanskiy and O. Vynokurova, "Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification," *Information Sciences*, vol. 220, pp. 170–179, 2013.

[16] A. Bagheri, H. M. Peyhani, and M. Akbari, "Financial forecasting using ANFIS networks with quantum-behaved particle swarm optimization," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6235–6250, 2014.

[17] J. D. Rairán Antolines, "Reconstruction of periodic signals using neural networks," *Tecnura*, vol. 18, no. 39, pp. 34–46, 2014.

[18] T. J. Ross, *Fuzzy logic with engineering applications*. John Wiley & Sons, 2009.

[19] M. Molaie, R. Falahian, S. Gharibzadeh, S. Jafari, and J. C. Sprott, "Artificial neural networks: powerful tools for modeling chaotic behavior in the nervous system," *Frontiers in computational neuroscience*, vol. 8, p. 40, 2014.

[20] T. Masters, *Practical neural network recipes in C++*. Morgan Kaufmann, 1993.

[21] U. Meyer-Baese and U. Meyer-Baese, *Digital signal processing with field programmable gate arrays*. Springer, 2007, vol. 65.

[22] H. K. Kwan, "Simple sigmoid-like activation function suitable for digital hardware implementation," *Electronics letters*, vol. 28, no. 15, pp. 1379–1380, 1992.

[23] L. Pang, H. Liu, B. Li, and B. Liang, "A data recovery method for high-speed serial communication based on fpga," pp. 664–667, Sept 2010.