JOURNAL OF ENGINEERING
ANNALS OF FACULTY OF ENGINEERING HUNEDOARA

PREDICTION OF THE SIGNALS USING THE NEURONAL NETWORKS

[1] Gelu Ovidiu TIRIAN, [1]Stela RUSU-ANGHEL, [2]Camelia PINCA-BRETOTEAN

[1] "Politechnica" University of Timisoara, Faculty of Engineering of Hunedoara, Department of Electrical Engineering and Industrial Informatics, ROMANIA
[2] "Politechnica" University of Timisoara, Faculty of Engineering of Hunedoara, Department of Engineering and Management, ROMANIA

Abstract:
This paper work refers to the prediction problems which are used with the help of the neuronal networks. The network is made of a neuron whose function is linear and who has the past few 5 input values of the useful signal x(t) – this signal must be predicted.
The training algorithm is Widrow-Hoff. This algorithm decreases the number of square errors between the output of the network and the required value, and it eventually establishes the "weight" factor.
Keywords:
Prediction, algorithm, neuronal networks, signal

1. INTRODUCTION

This identification [2], [3] is based on some neuronal methods and it has been lately used a research method due to mathematical research about the approximation properties of the neuronal networks - MLP (Multilayer Perceptron) and RBF (Radial-Basis Function) type [1]. The development of this new domain has been spectacular due to a remarkable contribution of some Automatics scientists. They have used the MLP networks for the non-linear identification and they have decided to use the RBF-type patterns. Thus, we have to point out the fact that the most important automatic application of those techniques, who are specific for the neuronal networks, is the use of the MLP features with sinoid nodes inside the hidden layer(s). On the other hand, those companies who produce technical-scientific software have begun developing specific facilities for the neuronal networks, so that the first version of the Neuronal Network Toolbox (NNT) should be enclosed in the MATLAB 4.2 environment – it had a major impact on the interest of the automatics specialists. NNT, from the earliest to the most recent version, has been designed to cover a range of applications of the neuronal networks, the designers had no intention to develop the Simulink blocks simultaneously [4]. These blocks are used for designing dynamic patterns.

2. NEURONAL NETWORKS. TRAINING NEURONAL NETWORKS

The neuronal networks can be made of simple processing elements, such as perceptrons or neuronals (Figure no. 1), and they make up one-layer networks, or of multiple elements and they make up multi-layer networks [1]. All these network-types enclose some elements that are distributed within the connection „weights" amongst different layers that make up each network [6]. The properties of the neuronal networks are: contain memory, shape acknowledgment, control and identification of the non-linear processes, etc. These properties are obtained through learning, such as in the case of the physiological systems. Specific training algorithms could be used in order to determine the values of those weights/percentages who represent the solution to the problems we solve by using the neuronal networks. The training algorithms could be divided in two separate categories: supervised instruction methods and non-supervised instruction methods. In the case of those methods belonging to the first category, the instruction is called supervised because we know both the input and the output sizes. The system is shaped up with the help of a neuronal network and the weight amongst the layers have random values. By comparing the input sizes that we already know and those of the output of the network (after we have used the input data set), we get an error signal. This signal helps us establish and adjust the weights amongst the layers of the network, in order to diminish a performance criterion [8].

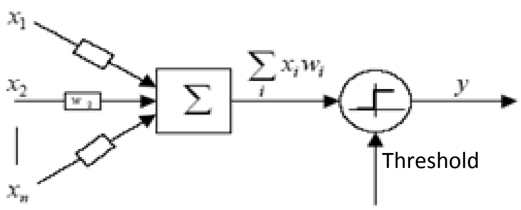

Figure no. 1 Classic perceptron model

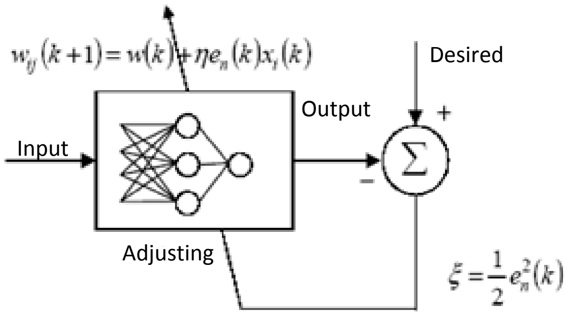

Figure no. 2 Supervised instruction mechanism for neuronal networks

The non-supervised learning methods do not uses already-known input sizes during the training stage of the neuronal network, by using only the input sizes for adjusting the weights. Thus, we can establish some input categories who correspond to some inputs from the data set, or „winner take all" outputs – in this case, the output neuron who has the highest activity wins and turns active, meanwhile the others do not work. This method is called self-organisation and we could use it successfully in matters of shape acknowledgement.

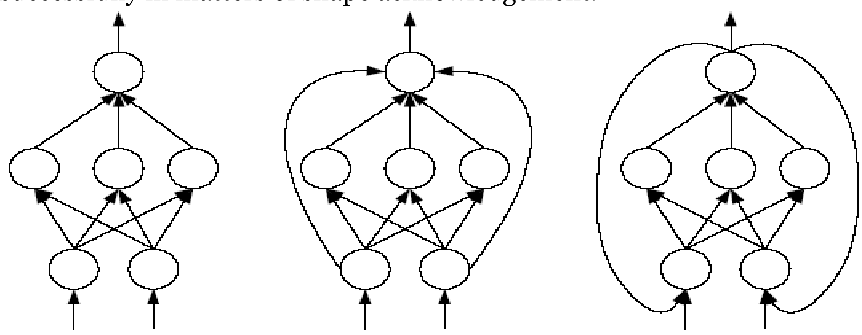

Figure no. 3„Feed forward" neuronal networks topologies a), b) and „feedback" neuronal networks topology c)

According to this topology, the artificial neuronal networks could be classified in two categories - „feedback" and „feed forward". In the case of a „feed forward" network the neuronal output is sent to other neurons who do not receive any information from the input neurons from the surface layers - Figure 3 a) and b).

3. NEURONAL NETWORKS USED FOR SIGNAL PREDICTION

The design of the linear prediction neuron is described in Figure no. 4. The network is made of a linear active neuron and the input receives the last five values of the useful signal x(t) - this signal must be predicted [4], [7].

We write the matrix P, $p=[x(t) \quad x(t-1) \quad x(t-2) \quad x(t-3) \quad x(t-4)]$, and the five delay values of the x(t) signal, who are going to be represented at the input of the neuronal network. The matrix and the values are going to represent the data we need for the supervised training, by using the Widrow-Hoff algorithm. This algorithm decreases the sum of the square errors between the output and the required value, and establishes the weight vector who is able to solve the problem.

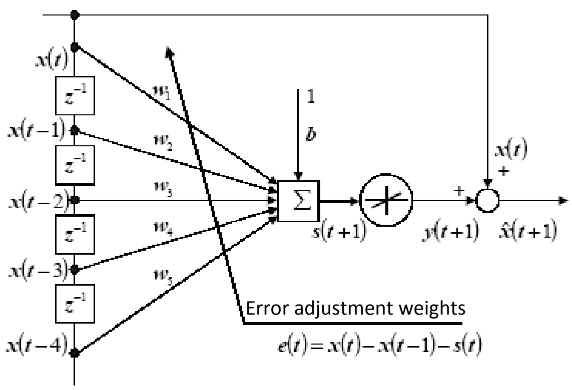

Figure no. 4 Linear prediction neuron

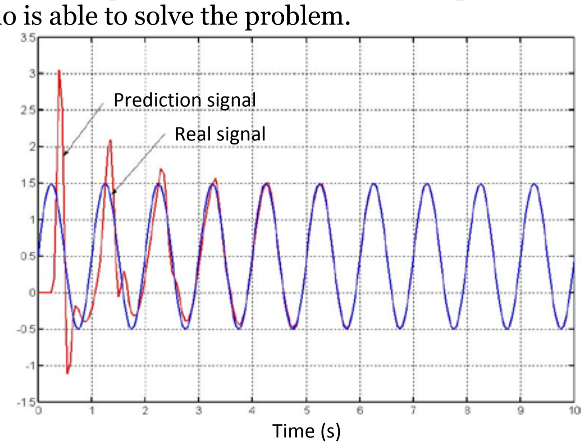

Figure no. 5 The proposed sinusoidal signal (blue) and the network predicted signal (red)

We consider a sinusoid signal $x(t) = 0.5 + \sin(2 \cdot \pi \cdot t)$ - in Figure no. 5 it is coloured in blue. The learning ratio of the training algorithm is $\eta = 0.1$, considering that it is constant during the training process. The network turns active and the training algorithm is used for a period of 200 steps. We could see that after about 100 repetitions the network is able to predict the signal we propose – Figure no. 5, where the red signal represents the output of the neuronal network during the training process. Figure no. 6 describes the prediction error - the difference between the real and the predicted signal. This signal tends to reach 0 after a certain time.

Figure no. 7 describes the values of the simple perception weights during the training process. The identification time of the process lasts according to the value we choose for the learning ratio. If we want a faster identification then, the value must be increased, but the values we have estimated during the first stages reach important values. We could also use another training method that should vary the learning ratio throughout the process, for improving the methods.

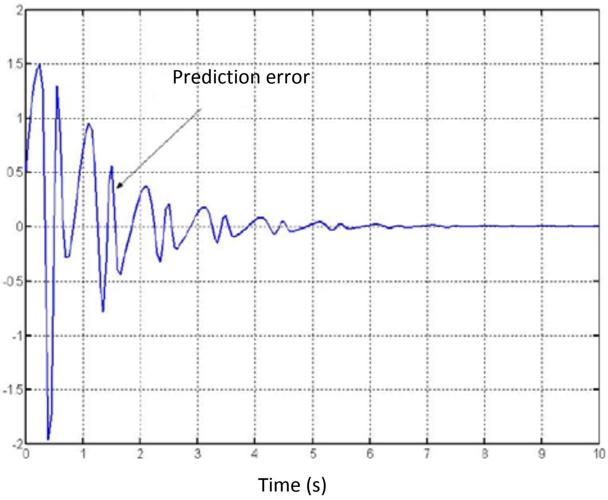

Figure no. 6 Prediction error between the real and predicted signal

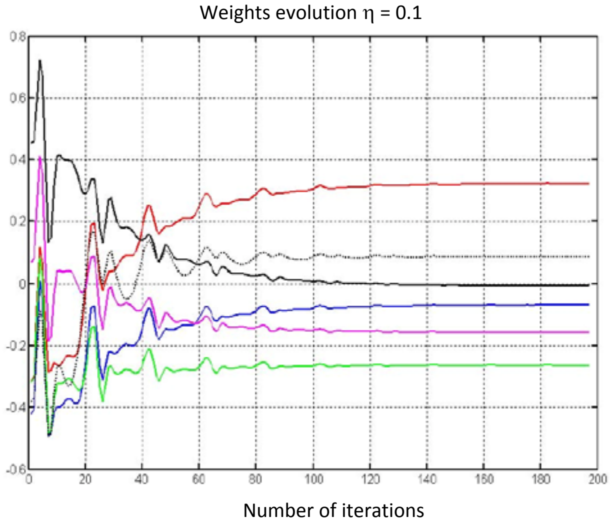

Figure no. 7 The evolution of the weights during the training process

4. CONCLUSIONS

All the neuronal networks could be made of simple processing elements, such as perceptions or neurons, so they should make up some single-layer networks, or made of several elements and they should make up some multi-layer networks. The „information" within all these networks is distributed within the connection weights amongst the different layers that make the network. Studies have proven that the prediction of the neuronal networks signals is extremely effective.

This paper work has described the linear neuron used for predictions, the training process, and the results we had obtained. The network is made of one neuron, according to the linear active process, which has the last five input values we have to predict.

REFERENCES

[1] Dumitrescu, D., Costin, H. „Reţele neuronale teorie şi aplicaţii" Editura Teora, 1996.
[2] Ljung, L. "System Identification" Theory for the User Prentice Hall, Inc., Englewood Cliffs, New Jersey 07632.
[3] Ljung, L., Soderstrom T. "Theory and Practice of Recursive Identification", MIT Press, 1983.
[4] Moktani, M, Marie, M. "Engineering Application of Matlab 5.3", Springer, 2000.
[5] Narendra, K. S., Parthasaraty, K. "Identification and Control of Dynamical Systems Using Neural Networks", IEEE Transactions on Neural Networks, vol. 1 nr. 1 1990, pag. 4-27.
[6] Tirian G.O. „High speed neuronal estimator for the command of the induction machine" International multidisciplinary scientific symposium "UNIVERSITARIA SIMPRO 2006" October, 13-14, pg.62...65, 2006.
[7] G.O.Tirian „Abordari moderne in estimarea si identificarea sistemelor", Referat doctorat, Timisoara, 2005.
[8] Widrow, B., Lehr, M. A. "30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation", Proceedings of IEEE, vol. 78 nr. 9 1990, pag. 1415-1439