The 16th INTERNATIONAL DAAAM SYMPOSIUM
"Intelligent Manufacturing & Automation: Focus on Young Researchers and Scientists"
19-22nd October 2005

NEURAL NETWORK WITHOUT BIAS NEURON FOR HIDDEN LAYER

Majetic, D.; Brezak, D.; Novakovic, B. & Kasac, J.

Abstract: *In this paper the nonlinear dynamic discrete-time neuron model, the so-called Dynamic Elementary Processor (DEP) is proposed. This dynamic neuron disposes of local memory, in that it has dynamic states. To accelerate the convergence of proposed extended dynamic error-back propagation learning algorithm, the adaptive neuron activation is applied. Instead of most popular unipolar and bipolar Sigmoidal neuron activation functions, the Gauss activation function with adaptive parameters is proposed. Based on the DEP neuron with adaptive activation function in hidden layer, and without Bias neuron for hidden layer, a Dynamic Multi Layer Neural Network is proposed and used for the identification of discrete-time nonlinear dynamic system.*

Key words: *dynamic neural network, adaptive activation function, bias neuron, identification*

1. INTRODUCTION

Error-back propagation is one of the most famous training algorithms for multilayer neural networks. Over the last decade, many improvement strategies have been developed to speed up the error-back propagation, and improve neural network learning and generalization features. All of these strategies can be separated in three basic categories. The first category deals with the improvement of the error back-propagation learning algorithm (Smagt, 1994). The second category deals with the neurons weights initial values (Nguyen & Widrow, 1990; Darken & Moody, 1991) and the third category deals with neural network topology optimization (Lawrence at al., 1996).

In this paper the neuron structure modification and activation function (AF) with adaptive parameters are proposed. With adaptive Gauss AF we wish to eliminate the Bias neuron for hidden layer. Before doing that, we must answer to some fundamental questions. The first one is, why is the Bias neuron so important, and the second one is, can we eliminate Bias neuron for hidden layer?

It is well known that any nonlinear, smooth, differentiable, and preferably non-decreasing function can be used as AF in hidden layer. The two most popular activation functions are the unipolar Logistic and the bipolar Sigmoidal functions. For those types of activation functions, Bias neuron is very important, and the error-back propagation neural network without Bias neuron for hidden layer does not learn (Kecman, 2001). Shortly, the Bias weights control shapes, orientation and steepness of all types of Sigmoidal functions through data mapping space. However, if one uses Gauss AF in hidden layer with adaptive parameters, than the Bias neuron and his weights can be neglected. Gauss function parameters controls the AF shapes and orientation, and the position of activation functions in data mapping space.

2. DYNAMIC NEURAL NETWORK

The basic idea of the dynamic neuron concept is to introduce some dynamics to the neuron activation function (AF), such that the neuron activity depends on the internal neuron states. In this study, an ARMA (Auto Regressive Moving Average) filter is integrated within the well-known static neuron model. Such a filter allows the neuron to act like an infinite impulse response filter, and the neuron processes past values of its own activity and input signals. The structure of a proposed dynamic neuron model is plotted in Fig. 1. The filter input and output at time instant (n) are given in (1) and (2) respectively (Novakovic at al., 1998):

$$\mathrm{net}(n) = \sum_{j=1}^{J-1} w_j u_j \,, \tag{1}$$

$$\tilde{y}(n) = b_0 \mathrm{net}(n) + b_1 \mathrm{net}(n-1) + b_2 \mathrm{net}(n-2) - \\ - a_1 \tilde{y}(n-1) - a_2 \tilde{y}(n-2) \,. \tag{2}$$

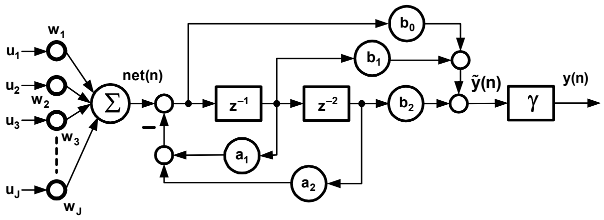

Fig. 1. Discrete-time dynamic neuron model

Widely used nonlinear Sigmoidal bipolar AF and Gauss AF with adaptive parameters, are described in (3) and (4) respectively.

$$y(n) = \gamma(\tilde{y}(n)) = \frac{2}{1 + e^{-\tilde{y}(n)}} - 1 \,, \tag{3}$$

$$y(n) = \gamma(\tilde{y}(n)) = e^{-\frac{1}{2}\left(\frac{\tilde{y}-c}{\sigma}\right)^2} \,, \tag{4}$$

The network proposed in this study has three layers. Each i-th neuron in the first, input layer has single input that represents the external input to the neural network. The second layer consists of dynamic neurons, which are presented by Fig. 1. Each j-th dynamic neuron in hidden layer has an input from every neuron in the first layer. Each k-th neuron in the third, output layer has an input from every neuron in the second layer.

3. THE LEARNING ALGORITHM

The goal of the learning algorithm is to adjust the neural network learning parameters ϑ in order to determine the optimal parameter set that minimizes a performance index E (Kecman, 2001) as follows :

$$E = \frac{1}{2} \sum_{n=1}^{N} (O_d(n) - O(n))^2 \,, \tag{5}$$

where N is the training set size, and the error is the signal defined as difference between the desired response $O_d(n)$ and the actual neuron response O(n). This error is propagated back to the input layer through the dynamic filters of dynamic neurons in hidden layer. Iteratively, the optimal parameters weights, filter coefficients and DEP activation function parameters (c and σ) are approximated by moving in the direction of steepest descent:

$$\vartheta_{new} = \vartheta_{old} - \eta \nabla E = \vartheta_{old} - \eta \frac{\partial E}{\partial \vartheta}, \tag{6}$$

where η is a user-selected positive learning constant (learning rate). According to (6) and for the purpose of easier comparison of different activation functions, we did not use any off learning accelerator algorithms. All error measures are reported using non-dimensional Normalized Root Mean Square error index NRMS (Lapedes & Farber, 1987, Novakovic et al., 1998).

4. EXPERIMENTAL RESULTS

As an interesting application of the proposed neural network algorithm, the identification of the dynamic discrete-time nonlinear system is performed. The system behavior is governed by the 1st order difference equation (Kecman, 1994),

$$x(n+1) = \left(0.9 - 0.003\, x(n)\right) x(n) + 0.2\, u(n), \tag{7}$$

with sampling time T_0=1s and a state-dependent time constant of about $T \cong 10$s. Such system is difficult to identify by classical methods when the mathematical structure of nonlinearity is unknown, because nonlinearity cannot be separated from the linear dynamics like e.g. a Hammerstein model. In order to obtain a good model of a nonlinear process, it is important that the learning data completely cover the relevant state space and contain a rich spectrum of frequencies. Thus, the process is excited with a pseudo-random binary noise (PRBS) signal with amplitude modulation. Output data are spoiled with pink noise of variance 0.05 x_{max}. The set of 621 data samples is plotted in Fig. 2.

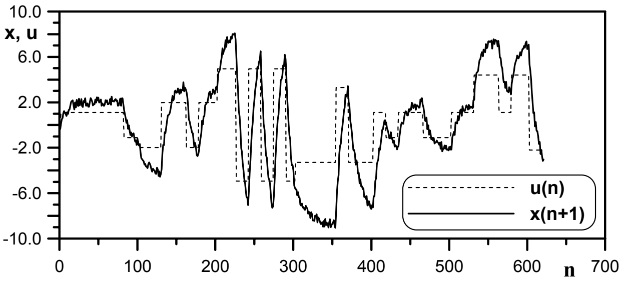

Fig.2. The set of 621 data samples

The identification of above described system is performed by neural network topology with 1 input neuron (u(n)), 5 dynamic neurons in hidden layer and 1 static neuron (x(n+1)) in output layer. For the training procedure the first 350 values of single input-output data set plotted in Fig. 2 is used. The goal for the neural network with adaptive Gauss AF was achieved with only 5 hidden nodes and in only 5000 learning steps. The neural network with bipolar Sigmoid AF in hidden layer (with Bias neuron weights) achieved the similar learning error (NRMS<0.08) after 35.000 learning steps. To illustrate the networks generalization capabilities, the new 800 test data points for the adaptive Gauss and bipolar Sigmoid AF are given in fig. 3.

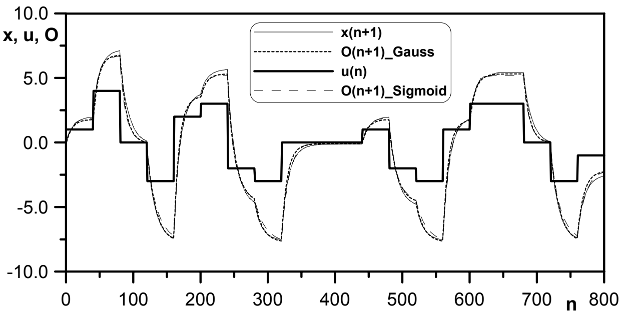

Fig. 3. Test for the 5-5-1 neural network topology with Gauss and bipolar Sigmoid AF

According to the Fig. 3 it is obvious that both neural networks solved the problem. These, and some other experiments (Majetic, et al., 2003) shows that neural network with adaptive activation function performs better mapping. Adaptive Gauss AF eliminates the Bias neurons weights, which means that the number of neurons and the number of those neurons learning parameters are reduced.

Therefore, the proposed neuron structure modification concerning integrated ARMA filter and adaptive Gauss AF gives very promising results.

5. CONCLUSION

We established a basic dynamic neuron model, which processes multi inputs and does not require past values of the process measurements or prior information about its activity functions.

The main advantage of proposed dynamic neuron model is that it reduces the network input space. Additionally, because of elimination of the Bias neuron for hidden layer, the neural network with adaptive Gauss activation function has the less number of neurons and learning parameters. It reduces CPU time and memory needed; it learns much faster and has better generalization property. Such AF shows the great possibility in solving the local minima's problems. Finally, trained neural network with smaller topology has much faster response, which is more promising in real-time domain applications.

The proposed neural network offers a great potential in solving many problems that occurs in system modeling with a special emphasis on the systems with characteristics such as nonlinearity, time delays, saturation or time-varying parameters.

6. REFERENCE

Darken, C. & Moody, J. (1991). Note of Learning Rate Schedules for Stochastic Optimization, *Neural Information Processing Systems*, pp. 832-838.

Lapedes, A.S. & Farber R. (1987). Nonlinear Signal Processing Using Neural Networks: Prediction And System Modeling, *Technical Report*, Los Alamos National Laboratory, Los Alamos, New Mexico.

Lawrence, S.; Giles, C.L. & Tsoi, A.C. (1996). What Size of Neural Network Gives Optimal Generalization, Convergence Properties of Backpropagation, *Technical Report UMIACS-TR-96-22 and CS-TR-3617*, Institute of Advanced Computer Studies, Maryland.

Kecman, V. & Pfeifer, B-M. (1994). Exploiting The Structural Equivalence of Learning Fuzzy Systems and Radial Basis Function Neural Networks*, Proceedings of EUFIT'94, Aachen*, Vol. 1, pp. 58-66.

Kecman, V. (2001). *Learning and Soft Computing*, MIT Press, ISBN: 0-262-11255-8, Cambridge, Massachusetts, England.

Majetic, D.; Brezak, D; Novakovic, B. & Kasac, J. (2003) Dynamic Neural Network With Adaptive Neuron Activation Function, *14.th DAAAM International Symposium*, Katalinic, B. (Ed.), pp. 251-252, ISNB 3-901509-34-8, Sarajevo, Bosna i Hercegovina, 2003, DAAAM International, Vienna.

Nguyen, D. & Widrow, B. (1990). Improving the Learning Speed of Two-Layer Networks by Choosing Initial Values of the Adaptive Weights, *Proceedings of International Joint Conference on Neural Networks*, Vol. 3, pp. 21-26, Sand Diego, CA, USA.

Novakovic, B.; Majetic, D. & Siroki, M. (1998). *Artificial Neural Networks*, Faculty of Mechanical Engineering and Naval Architecture, ISBN 953-6313-17-0, Zagreb, Croatia.

Smagt, P. (1994). Minimization methods for training feed-forward networks, *Neural Networks 7*, pp. 1-11.