DANUBE ADRIA ASSOCIATION FOR AUTOMATION & MANUFACTURING

DAAAM INTERNATIONAL VIENNA

OFFPRINTS

Author(s):

MAJETIC, D.; BREZAK, D.; NOVAKOVIC, B. & KASAC, J.

This Publication has to be referred as:

Majetic, D.; Brezak, D.; Novakovic, B. & Kasac, J. (2003). Dynamic neural network with adaptive neuron activation function, Annals of DAAAM for 2003 & Proceedings of the 14th International DAAAM Symposium, ISBN 3-901509-34-8, ISSN 1726-9679, pp 285-286 Editor B. Katalinic, Published by DAAAM International, Vienna, Austria 2003

NOMINATION FOR DAAAM INTERNATIONAL SCIENTIFIC BOOK 2004

The Review Committee of DAAAM International Nominated this Paper for Publishing in the DAAAM International Scientific Book 2004, ISSN 1726-9687, ISBN 3-901509-38-0, Hard Cover, Editor B.[ranko] Katalinic, publisher DAAAM International Vienna, Vienna 2004. If you are interested to Publish Extended form of this Paper as a Book Chapter please contact the President of DAAAM International before 2003-11-30. The Publishing Conditions and Instructions for the Preparing of Manuscript you can Find at,

www.daaam.com

Annals of DAAAM for 2003 & Proceedings of the 14th International DAAAM Symposium, ISSN 1726-9679
ISBN 3-901509-34-8, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria 2003
45 Years of Faculty of Mechanical Engineering Sarajevo - University of Sarajevo
155 Years of Austrian Society of Engineers and Architects OIAV 1848

DYNAMIC NEURAL NETWORK WITH ADAPTIVE NEURON ACTIVATION FUNCTION

MAJETIC, D.; BREZAK, D.; NOVAKOVIC, B. & KASAC, J.

Abstract: *An attempt has been made to establish a nonlinear dynamic discrete-time neuron model, the so called Dynamic Elementary Processor (DEP). 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 function and momentum method are applied. Instead of most popular bipolar and unipolar Sigmoid neuron activation functions, the Gauss activation function with adaptive parameters is proposed. Based on the DEP neuron with adaptive activation function in hidden layer, a Dynamic Multi Layer Neural Network is proposed and tested in prediction of a Glass-Mackey time series.*

Key words: *dynamic neural network, adaptive neuron activation function, momentum, prediction, glass-mackey*

1. INTRODUCTION

Error-back propagation is one of the most famous training algorithms for multilayer perceptron. Unfortunately it can be very slow for practical applications. Over the last years many improvement strategies have been developed to speed up 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 with adaptive parameters are proposed. With applying only momentum method for speeding up the learning algorithm and proposed neuron activation function, neural network training procedure can be much efficient and faster. More over, the neural network with proposed activation function has the less number of neurons. And finally, trained neural network with smaller topology has much faster response, which is more promising in real-time domain applications.

2. DYNAMIC NEURAL NETWORK

The basic idea of the dynamic neuron concept is to introduce some dynamics to the neuron transfer function, 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):

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

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

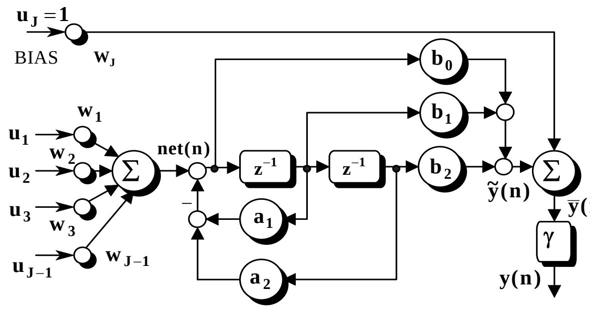

Fig. 1. Dynamic neuron model

The input of the neuron activation function (AF) is given in (3), and widely used nonlinear Sigmoid unipolar activation function and gauss activation function with adaptive parameters, are described in (4) and (5) respectively.

$$\bar{y}(n) = \tilde{y}(n) + w_J u_J \ , \qquad (3)$$

where $u_J = 1$ represents a threshold unit, also called Bias.

$$y(n) = \gamma(\bar{y}(n)) = \frac{1}{1 + e^{-\bar{y}(n)}} \ , \qquad (4)$$

$$y(n) = \gamma(\bar{y}(n)) = e^{-\frac{1}{2}\left(\frac{\bar{y}-c}{\sigma}\right)^2} \ , \qquad (5)$$

The network proposed in this study has three layers. Each i-th neuron in the first, input layer has single input which represents the external input to the neural network. The second layer is consisting 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, and one additional input with a fixed value of unity usually named as Bias. Each k-th neuron in the third, output layer has an input from every neuron in the second layer and, like the second layer one additional input with fixed value of unity (Bias).

3. 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 minimises a performance index E (Zurada, 1992) as follows :

$$E = \frac{1}{2} \sum_{n=1}^{N} (O_d(n) - O(n))^2 \ , \qquad (6)$$

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, (5)) are approximated by moving in the direction of steepest descent:

$$\vartheta_{new} = \vartheta_{old} + \Delta\vartheta \qquad (7)$$

$$\Delta\vartheta = -\eta\nabla E = -\eta\frac{\partial E}{\partial\vartheta} \qquad (8)$$

where η is a user-selected positive learning constant (learning rate). To accelerate the convergence of the learning algorithm given in (7), momentum method is applied. The momentum method is given in (9) and involves supplementing the current learning parameter adjustment (8) with a fraction of the most recent parameter adjustment. This is usually done according to the formula

$$\Delta\vartheta(n) = -\eta\frac{\partial E(n)}{\partial\vartheta(n)} + \alpha\Delta\vartheta(n-1), \qquad (9)$$

where α is a user-selected positive learning constant. The arguments *n* and *n-1* are used to indicate the current and the most recent training step (instant time), respectively. All error measures will be reported using non-dimensional error index NRMS, Normalized Root Mean Square error. "Normalized" means that the root mean square is divided by the standard deviation of the target data (Lapedes & Farber, 1987).

4. EXPERIMENTAL RESULTS

Lapedes suggested (Lapedes & Farber, 1987) the Glass-Mackey time series as a good benchmark for learning algorithms, because it has a simple definition, yet its elements are hard to predict (the series is chaotic). The goal of the task is to use known values of the time series up to the point $x(t)$, to predict the value $x(t+P)$ at some point P in the future . The standard method for this type of prediction is to create a mapping $f(\bullet)$ as follows :

$$x(t+P) = f\big(x(t), x(t-\Delta), x(t-2\Delta), ..., x(t-m\Delta)\big) \qquad (10)$$

where P is a prediction time into the future, Δ is a time delay , and m is an integer. According to the equation (10) an attractor can be reconstructed from a time series by using a set of time delayed samples of a series. By choosing $P = \Delta$ it is possible to predict the value of time series at any multiple of Δ time steps in the future, by feeding the output back into the input and iterating the solution. In this study we choose to use $P = \Delta = 6$, since results can be compared with previous experiments where $P = 6$. It is obvious that for $P = \Delta = 6$ and $m = 4$ the expansion (10) has the following form :

$$x(t+6) = f\big(x(t), x(t-6), x(t-12), x(t-18), x(t-24)\big) \qquad (11)$$

According to the equation (11) the input layer consists of 5 neurons (input buffer), and output layer consists of one static neuron with linear activation function. For hidden layer we suggested 10 and 5 dynamic neurons. Lapedes and Farber (Lapedes & Farber, 1987) for the same task used 20 hidden static neurons arranged in two hidden layer architecture. Training started with random weights values between -1 and +1, while the filter coefficients a_1 and a_2 were initialized to zeros to support a stable learning procedure. The network was trained with $\eta = 0.01$ and $\alpha = 0.8$. The trained network were used to predict new sets of values $x(t)$ in the future. The learning and testing results are given in Table 1.

Neuron AF	Unipolar Sigmoid		Adaptive Gauss	
Network Topology	5-10-1	5-5-1	5-10-1	5-5-1
Learning Epoch's	70000	80000	35000	50000
Learning (NRMS)	0,069	0,053	0,027	0,057
Test 1. (NRMS)	0,069	0,071	0,048	0,043
Test 2. (NRMS)	0,071	0,067	0,052	0,058
Test 3. (NRMS)	0,073	0,078	0,050	0,052

Table 1. Learning and test results

It is obvious that proposed neuron structure modification concerning integrated ARMA filter and adaptive Gauss activation function gives very promising results. The goal was achieved with only 5 hidden nodes. Neural network with adaptive activation function learns faster and have smaller topology. To illustrate the network generalization capability, the 300 data points of test 2. for the Adaptive Gauss and Unipolar Sigmoid activation functions are given in fig. 2.

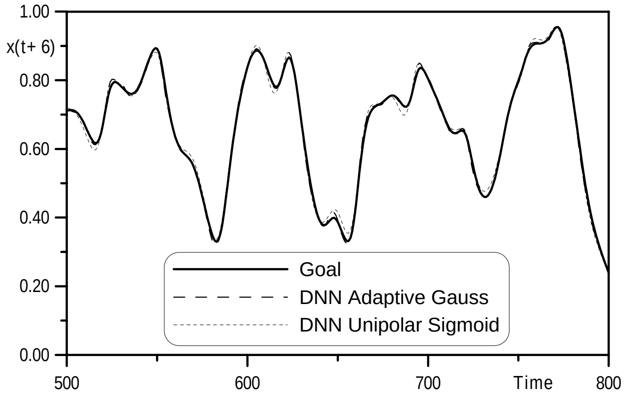

Fig. 2. Test 2. for the 5-5-1 neural network topology with Gauss and unipolar Sigmoid activation function

5. CONCLUSION

Within this approach a Multi Layer Perceptron with distributed dynamics based on the DEP neuron model and adaptive activation function was proposed to predict a time series of nonlinear chaotic system. An attempt was made within this approach to establish 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. The advantage of adaptive activation function is speeding up the learning algorithm. Such AF shows the great possibility in solving the local minima's problems. The proposed neural network offers a great potential in solving many problems that occurs in system modelling 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.

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.

Zurada, J.M. (1992). *Artificial Neural Systems*, W.P. Company, USA, ISSN: 0-314-93391-3.