ONE APPROACH FOR DISCRETE DYNAMICAL SYSTEM MODELING BASED ON FEED-FORWARD NEURAL NETWORK

Stela Angelova Stefanova

Technology School "Electronic System" associated with the Technical University of Sofia, Mladost 1, 1750 Sofia, Bulgaria, +359 2 8750040, e-mail: sstefanova@elsys-bg.org

The approach for discrete dynamical system modeling based on feed-forward neural network is considered. The proposed method is realized in two stages. The first one is training of the feed-forward neural network with given experimental data obtained for the designed system. The second stage is simulation with the neural network model. The variant of the method for feed-forward neural network training known as "backpropagation" is developed. The effective realization in matrix form of the proposed algorithm is given. This approach is successfully applied to the design problem in time domain for the recursive digital filter. Some results and graphical representations for lowpass recursive digital filters are given.

Keywords: neural networks, optimization, digital filters

1. INTRODUCTION

In this paper two stages of the approach for discrete dynamical system modeling with neural network (NN) are considered:

- Training of the neural network with given experimental data;
- Simulation with the designed neural network model.

The training procedure is realized in two steps. The first step consists in generation of training samples. At the second step, appropriated feed-forward neural network architecture has been chosen. This neural network structure has been training with training samples, obtained at the first step, so that with given appropriate signal applied to the input, the output variable of the neural network approximates a given target function in least square sense. The variant of the method for feed-forward neural network training known as "backpropagation" is developed. The effective realization in matrix form of the proposed algorithm is given.

This approach is successfully applied to the design problem in time domain for the recursive digital filter. Digital filter is a discrete dynamical system, which is described with difference equations. One approach for the 1-D FIR digital filter design based on the weighted mean square method and neural network to state the approximation problem is proposed in [1]. Some methods for the non-linear digital filters design using neural networks are considered in [2]. Time domain recursive digital filter model, based on recurrent neural network is proposed in [3]. A design approach for IIR eigenfilters with time and frequency domain constraints is published in [4]. Some basic results related to the discrete dynamical systems approximation using neural networks are discussed in [5].

2. DYNAMICAL SYSTEM MODELING BASED ON NEURAL NETWORK

The main idea of the proposed algorithm for discrete dynamical system modeling based on neural network is shown in Figure 1.

Supposing that the considered object shown in Fig. 1 is a dynamical system described by a system of differential or difference equations. Some experiments with the object are implemented and the input excitations and the output responses of the system are saved and stored.

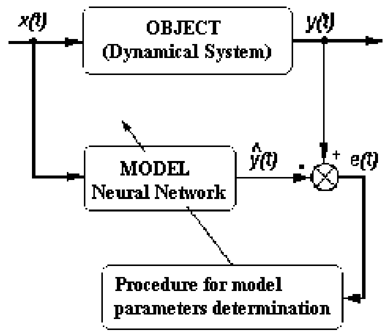

Fig. 1. Discrete dynamical system modeling based on neural network

The main goal of the modeling is to obtain the parameters of the neural network model in such way that the model output can approximate the object reaction in means square sense. To achieve this purpose it is necessary to make the difference between object reaction accepted as a target function and the neural network output i.e. to define the error function of the system model. The adjustment of the neural network parameters is realized by minimization of this error function using an appropriate optimization procedure.

3. BACKPROPAGATION ALGORITHM FOR NEURAL NETWORK TRAINING

The backpropagation algorithm for feed-forward three-layer neural network with activation function g(.) is shown in Fig. 2. This scheme expresses both computing process from input to output (forward algorithm of the method) and computation of the gradient components from output to input (backward algorithm of the method).

The error objective function is defined in the form:

(1) $$J = \frac{1}{2}\sum_{i=1}^{m} e_i^2$$

where $e_i = y_i - \hat{y}_i,\ i = 1,2,...,m$ - are the output errors of the neural network;

$\hat{y}_i$ - is the target function.

The following matrixes of weighting coefficients for the corresponding layers are written.

(2) $$\mathbf{W_1} = \begin{bmatrix} w_{11}^{(1)}\ w_{12}^{(1)} \ ...w_{1n}^{(1)} \\ w_{21}^{(1)}\ w_{22}^{(1)} \ ...w_{2n}^{(1)} \\ .... \\ w_{p1}^{(1)}\ w_{p2}^{(1)} \ ...w_{pn}^{(1)} \end{bmatrix} \quad \mathbf{W}_2 = \begin{bmatrix} w_{11}^{(2)}\ w_{12}^{(2)} \ ...w_{1p}^{(2)} \\ w_{21}^{(2)}\ w_{22}^{(2)} \ ...w_{2p}^{(2)} \\ .... \\ w_{q1}^{(2)}\ w_{q2}^{(2)} \ ...w_{qp}^{(2)} \end{bmatrix} \mathbf{W}_3 = \begin{bmatrix} w_{11}^{(3)}\ w_{12}^{31)} \ ...w_{1q}^{(3)} \\ w_{21}^{(3)}\ w_{22}^{(3)} \ ...w_{2q}^{(3)} \\ .... \\ w_{m1}^{(1)}\ w_{m2}^{(1)} \ ...w_{mq}^{(3)} \end{bmatrix}$$

$\mathbf{W_1} \in R_{p\times n}$ $\mathbf{W}_2 \in R_{q\times p}$ $\mathbf{W}_3 \in R_{m\times q}$

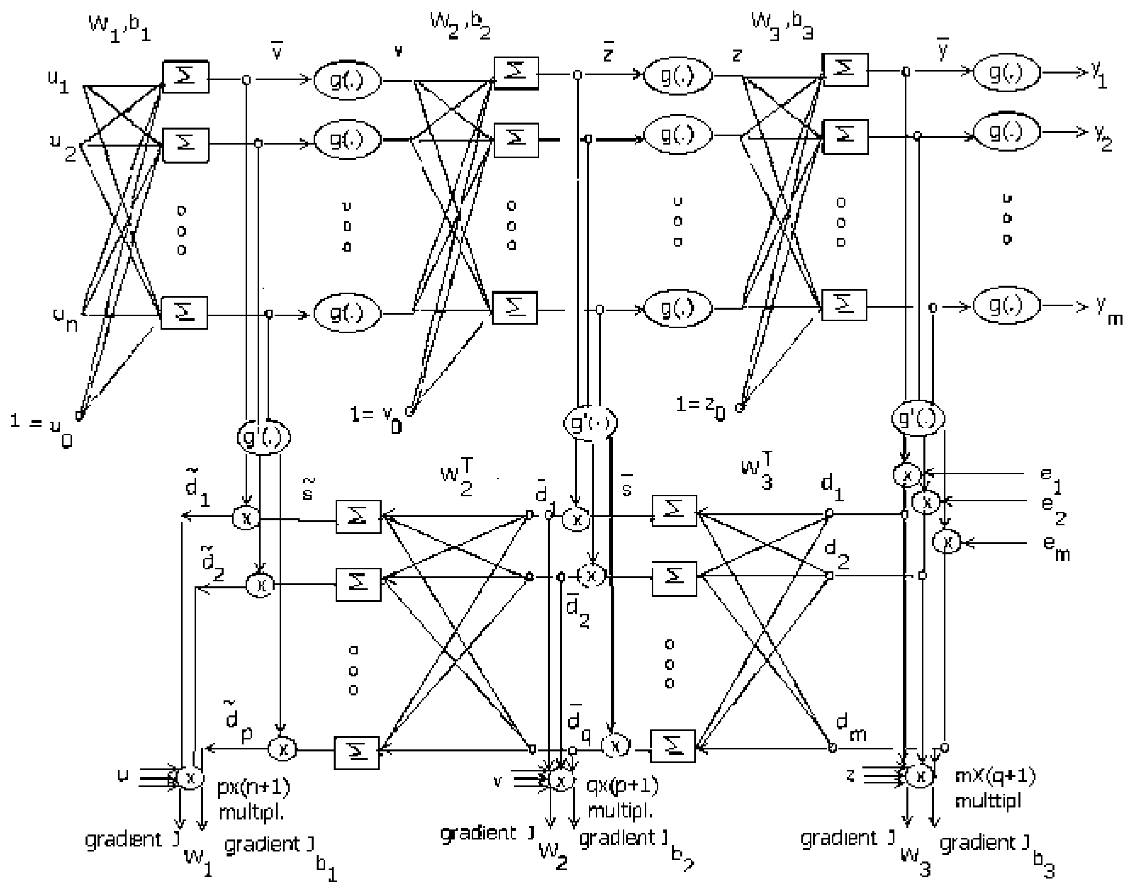

Fig. 2. Backpropagation algorithm for feed-forward three-layer neural network

3.1. Forward backpropagation algorithm

Main connections for the three-layer neural network are given as follows:

$$\mathbf{W_1 u} + \mathbf{b_1} = \bar{\mathbf{v}} \qquad \mathbf{W}_2\mathbf{v} + \mathbf{b}_2 = \bar{\mathbf{z}} \qquad \mathbf{W}_3\mathbf{z} + \mathbf{b}_3 = \bar{\mathbf{y}} \tag{3}$$

$$\mathbf{v} = [g(\bar{v}_1), g(\bar{v}_2),..., g(\bar{v}_p)]^T \quad \mathbf{z} = [g(\bar{z}_1), g(\bar{z}_2),..., g(\bar{z}_q)]^T \quad \mathbf{y} = [g(\bar{y}_1), g(\bar{y}_2),...g(\bar{y}_m)]^T$$

and the calculations are made from the input to the output in accordance with the scheme in fig. 2.

3.2. Backward backpropagation algorithm

The gradient of the objective function (1) in matrix form for the three layers of the neural network have to be obtained by the realization the procedure of the backward backpropagation algorithm, shown as parallel scheme in Fig. 2.

$$[\frac{\partial J}{\partial \mathbf{w}_3} \vdots \frac{\partial J}{\partial \mathbf{b}_3}] = \begin{bmatrix} d_1z_1\ d_1z_2\ ...d_1z_q\ \vdots d_1z_0 \\ d_2z_1\ d_2z_2\ ...d_2z_q \vdots d_2z_0 \\ .... \\ d_mz_1\ d_mz_2\ ...d_mz_q \vdots d_mz_0 \end{bmatrix} ; \mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ .... \\ d_m \end{bmatrix} = \begin{bmatrix} e_1g'(\bar{y}_1) \\ e_2g'(\bar{y}_2) \\ .... \\ e_mg'(\bar{y}_m) \end{bmatrix} \tag{4}$$

$$(5)\quad [\frac{\partial J}{\partial \mathbf{w}_2} \vdots \frac{\partial J}{\partial \mathbf{b}_2}] = \begin{bmatrix} \bar{d}_1 v_1 & \bar{d}_1 v_2 & \ldots \bar{d}_1 v_p & \vdots \bar{d}_1 v_0 \\ \bar{d}_2 v_1 & \bar{d}_2 v_2 & \ldots \bar{d}_2 v_p & \vdots \bar{d}_2 v_0 \\ \ldots & \ldots & \ldots & \ldots \\ \bar{d}_q v_1 & \bar{d}_q v_2 & \ldots \bar{d}_q v_p & \vdots \bar{d}_q v_0 \end{bmatrix}; \bar{\mathbf{d}} = \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \\ \ldots \\ \bar{d}_q \end{bmatrix} = \begin{bmatrix} \bar{s}_1 g'(\bar{z}_1) \\ \bar{s}_2 g'(\bar{z}_2) \\ \ldots \\ \bar{s}_q g'(\bar{z}_q) \end{bmatrix}; \begin{bmatrix} \bar{s}_1 \\ \bar{s}_2 \\ \ldots \\ \bar{s}_q \end{bmatrix} = \mathbf{W}_3^T \begin{bmatrix} d_1 \\ d_2 \\ \ldots \\ d_m \end{bmatrix}$$

$$(6)\quad [\frac{\partial J}{\partial \mathbf{w}_1} \vdots \frac{\partial J}{\partial \mathbf{b}_1}] = \begin{bmatrix} \tilde{d}_1 u_1 & \tilde{d}_1 u_2 & \ldots \tilde{d}_1 u_n & \vdots \tilde{d}_1 u_0 \\ \tilde{d}_2 u_1 & \tilde{d}_2 u_2 & \ldots \tilde{d}_2 u_n & \vdots \tilde{d}_2 u_0 \\ \ldots & \ldots & \ldots & \ldots \\ \tilde{d}_p u_1 & \tilde{d}_p u_2 & \ldots \tilde{d}_p u_n & \vdots \tilde{d}_p u_0 \end{bmatrix}; \tilde{\mathbf{d}} = \begin{bmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \ldots \\ \tilde{d}_p \end{bmatrix} = \begin{bmatrix} \tilde{s}_1 g'(\bar{v}_1) \\ \tilde{s}_2 g'(\bar{v}_2) \\ \ldots \\ \tilde{s}_p g'(\bar{v}_p) \end{bmatrix}; \begin{bmatrix} \tilde{s}_1 \\ \tilde{s}_2 \\ \ldots \\ \tilde{s}_p \end{bmatrix} = \mathbf{W}_2^T \begin{bmatrix} \bar{d}_1 \\ \bar{d}_2 \\ \ldots \\ \bar{d}_q \end{bmatrix}$$

Let introduce the following diagonal matrixes:

$$\mathbf{V} = \mathrm{diag}(g'(\bar{v}_1), g'(\bar{v}_2), \ldots, g'(\bar{v}_p));\ \mathbf{Z} = \mathrm{diag}(g'(\bar{z}_1), g'(\bar{z}_2), \ldots, g'(\bar{z}_q));$$

$$\mathbf{Y} = \mathrm{diag}(g'(\bar{y}_1), g'(\bar{y}_2), \ldots g'(\bar{y}_m))$$

Then for the backward backpropagation algorithm is true the following statement:

Statement: The gradient of the objective function (1) can be written in the matrix as:

- for weighting coefficients at the third layer:

$$(7)\qquad [\frac{\partial J}{\partial \mathbf{w}_3} \vdots \frac{\partial J}{\partial \mathbf{b}_3}] = \mathbf{d}[\mathbf{z}^T \vdots z_0]; \qquad \mathbf{d} = \mathbf{Ye}$$

- for weighting coefficients at the second layer:

$$(8)\qquad [\frac{\partial J}{\partial \mathbf{w}_2} \vdots \frac{\partial J}{\partial \mathbf{b}_2}] = \bar{\mathbf{d}}[\mathbf{v}^T \vdots v_0]; \qquad \bar{\mathbf{d}} = \mathbf{Z}\mathbf{W}_3^T \mathbf{d}$$

- for weighting coefficients at the first layer:

$$(9)\qquad [\frac{\partial J}{\mathbf{w}_1} \vdots \frac{\partial J}{\partial \mathbf{b}_1}] = \tilde{\mathbf{d}}[\mathbf{u}^T \vdots u_0]; \qquad \tilde{\mathbf{d}} = \mathbf{V}\mathbf{W}_2^T \bar{\mathbf{d}}$$

The obtained gradient can be used for the objective function (1) optimization by standard quasi-Newton procedure.

4. NEURAL NETWORK MODEL OF RECURSIVE DIGITAL FILTER

The recursive or IIR digital filter can be considered as a discrete dynamic system described in time domain with n – order difference equation that has been stated using the delayed samples of the excitation input signal and the response signal at the output:

$$(10)\qquad v(kT) = \sum_{i=0}^{n} a_i u(kT - iT) - \sum_{i=1}^{n} b_i v(kT - iT)$$

where $a_i, b_i, i = 1..n$ are the IIR digital filter coefficients, $u(kT), v(kT)$ - are the excitation input signal, respectively the response signal at the filter output.

Transforming the difference equation (10), the state space description of the recursive digital filter can be obtained as a system of n – number of first order equations in the form:

$$\begin{aligned}\mathbf{x}(nT+T) &= \mathbf{A}\,\mathbf{x}(nT)+\mathbf{B}\,\mathbf{u}(nT)\\ y(nT) &= \mathbf{C}\,\mathbf{x}(nT)+\mathbf{D}\,\mathbf{u}(nT)\end{aligned} \tag{11}$$

where **A**, **B**, **C**, **D** are matrixes and **x**(nT) is state space vector.

Taking into consideration the results obtained in [5], the IIR digital filter have to be modeling by one-layer recurrent neural network with number of neurons that correspond to the order of IIR filter. When the recursion is introduced in the feed-forward neural network this means that the training sequences will be defined from the input signal and from the delayed previous filter responses.

5. MODELING RESULTS

The model of the lowpass IIR digital filter is developed. The requirements to the digital filter are: passband 0 - 800 rad/s; passband attenuation 0,5dB; stopband > 1600 rad/s; stopband attenuation – 45dB; sampling frequency 5000 rad/s. The IIR digital filter is designed as a 5-th order Chebyshev filter. The filter impulse response is calculated and this time characteristic is used as a target function to form the training sequences of the recurrent neural network with five neurons.

The impulse responses of the target 5-th order IIR Chebyshev filter and the neural network model with 5 neurons are shown in Fig. 3a and Fig.3b respectively.

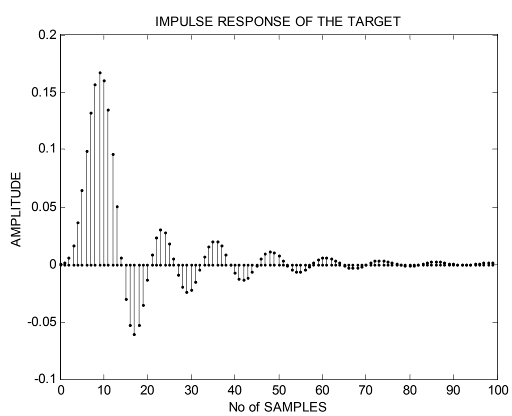

Fig 3a. Impulse response of the target IIR filter

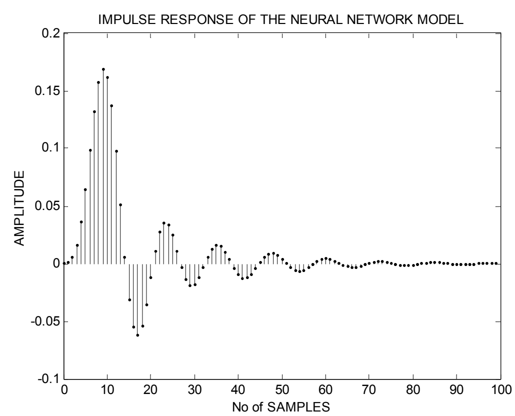

Fig. 3b. Impulse response of the neural network model

The comparison between the behavior of the impulse response of the target IIR Chebyshev filter (curve with solid line) and the impulse response of neural network model of IIR digital filter (curve with symbol * line) is shown in Fig. 4.

In order to examine that the IIR digital filter satisfies the requirements it is necessary to transform the impulse response in frequency domain using FFT.

The neural network training is performed by harmonic sinusoidal signal with frequency in the digital filter passband. The training sequence based on harmonic sinusoidal signal (curve with solid line) and the response of the IIR digital filter neural network model (curve with symbol * line) are shown in Fig. 5.

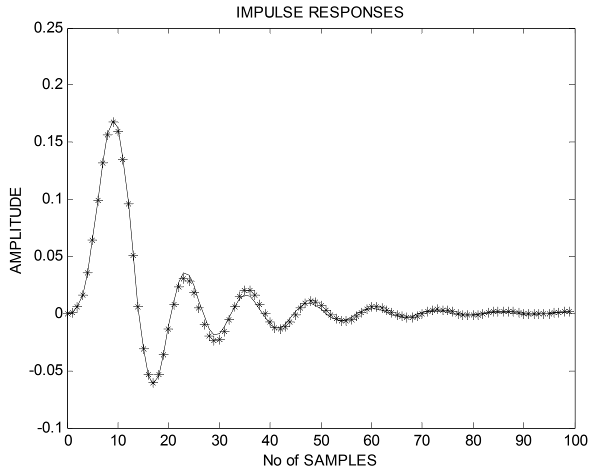

Fig. 4. Impulse responses of the target IIR filter and the neural network model

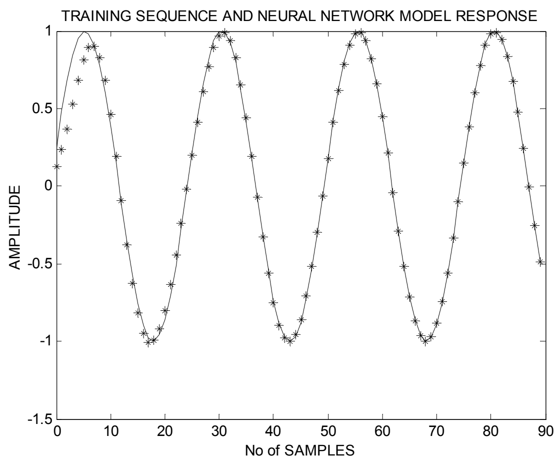

Fig.5. Training sequence and response of the IIR filter neural network model

6. REFERENCES

[1]. Bhattacharya D. and A. Antoniou, *Real-time Design of FIR Filter by Feedback Neural Network*, IEEE Signal Processing Letters, Vol.3, No 4, May 1996, pp. 158 – 161.

[2]. Pedersen M., *Optimization of Recurrent Neural Networks for Time Domain Series Modeling*, Ph.D Thesis, Department of Mathematical Modeling, Technical University of Denmark, 1997.

[3]. Stefanova S., *Time Domain Recursive Digital Filter Modeling Based on Recurrent Neural Network Training,* 13-th International Scientific and Applied Science Conference "ELECTRONICS ET' 2004", 22-24 September 2004, Sozopol, book 2, pp 181 - 186.

[4]. Pei S., Ch. Hsu, P. Wang, *Design of a class of IIR eigenfilters with time and frequency domain constraints.* IEEE Trans. on Circuits and Systems-II Analog and Digital Signal Processing, vol.49, No 2, February 2002, pp 145 - 151

[5]. Jin L., P. Nikiforuk, M. Gupta, *Approximation of Discrete-Time State-Space Trajectories Using Dynamical Recurrent Neural Networks*, IEEE Transactions on Automatic Control, Vol.40, No 7, 1995.