Continuous Time Delay Neural Networks for Detection of Temporal Patterns in Signals

Reza Derakhshani
Lane Department of Computer Science and Electrical Engineering
West Virginia University
P.O. Box 6109
Morgantown, WV 26506-6109
E-mail: reza@csee.wvu.edu

Stephanie A.C. Schuckers, PhD
Department of Electrical and Computer Engineering
Clarkson University
P.O. Box 5720
Potsdam, NY 13699-5720
E-mail: sschucke@clarkson.edu

Abstract: **A method for temporal pattern recognition for continuous time signals is addressed. It is shown how a simple form of back-propagation can be used in conjunction with a temporal error signal to adapt both the weights and path delays of a continuous time delay feed forward multi-layer neural network with hard-limited output. An instance of such a network is simulated and some of the results are discussed. During the initial tests the network showed robust capabilities for detection of temporal patterns, including fast recognition of onsets of new waveforms in presence of moderately heavy noise and phase and frequency distortions.**

***Index Terms*—Time Delay Neural Networks, Signal Processing, Time Series, Adaptive Filters.**

I-Introduction

Almost every piece of information received from the real world has a continuous temporal nature. However, complexity of temporal pattern recognition makes it a challenging problem [1]. Consequently, temporal information processing by neural networks, especially in continuous time, has been less visited [2].

Biological neural networks have always motivated creation of new artificial neural network models. One of the more complicated problems in temporal neural networks has been the incorporation of short and long-term memory, and most importantly, design of efficient training algorithms. Short-term memory (STM) is easier to handle [3,4]. It usually takes the form of multiple delayed copies of a signal and thus provides a limited temporal history of events. One can find biological evidence that may suggest such phenomena in some neuronal ensembles. For instance, consider multiple synaptic connections to the same dendritic structure [5] (Figure 1).

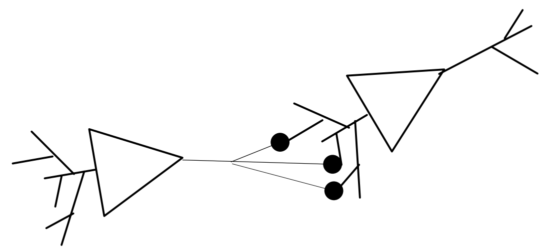

Fig. 1. A hypothetical situation where a neuron's axon makes multiple connections to a target dendritic tree.

Such configurations can be seen in neuronal tissue histology. One can hypothesize that these multiple paths, besides different cable delays, have different synaptic transmission delays due to electrochemical transport mechanism required to pass a signal through the synaptic cleft, which is usually between 0.3 to 5 ms [6,7]. This makes the model presented by Figure 2 and equation (1) biologically more plausible.

There has been research on discrete time-delay neural networks (TDNN) [8,9,10] and even their continuous time versions [11]. Here we offer a simpler, different derivation for a continuous time delay neural networks with back-propagation. In addition, enhancements such as addition of Hysteresis to the output, resolution of possible negative delays, and suggestions for different temporal error integration limits are offered. A series of simulations depict the capabilities of the suggested network.

II- The Forward Path

The proposed model is a feed-forward two-layered Perceptron with adaptive weights and adaptive continuous time delays for each connection in the input layer, which in turn are usually connected to the same source. A hidden neuron for such a network is depicted in Figure 2.

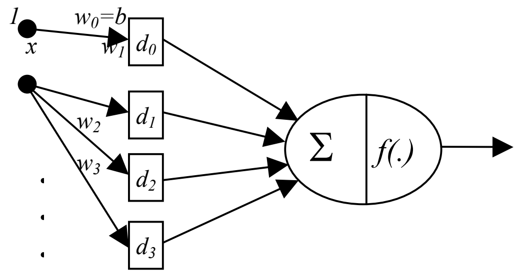

Fig. 2. A hidden layer neuron. d and w are real-valued delay and weight for each input branch, respectively. x is the input signal and *f* is the sigmoidal output nonlinearity.

These hidden neurons are connected to an output node in the output layer through weighed connections with no delays. Delay for the output layer is redundant since the synaptic integration and sigmoidal nonlinearities are time invariant and the effect of having d_j at the output of a hidden unit (input of the output node) while having d_i at the hidden neuron's inputs is equal to the effect of having the delay d_i+d_j at the inputs. The input signal x(t) passes through a series of dendrites with different synaptic weights and delays. Before explaining the rest of the network, it is instructive to analyze the behavior of such a hidden unit further.

The output of a hidden neuron for a continuous time signal x(t) can be written as

$$y = f\left(\sum_{i=0}^{N} w_i x_i(t-d_i)\right) \quad (1)$$

Where N is the number of the dendritic branches stemming from the input x(t) (considering x to be the sole source of inputs to this node). w_i and d_i are the i^{th} branch weight and delay, and w_0 is the bias value (b) for the corresponding constant input of $x_0=1$.

Recalling from the linear time invariant (LTI) systems theory [12], the output of a system with an impulse response h(t) for the input x(t) can be computed from the following convolution integral

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(\tau)x(t-\tau)\,d\tau = \int_{-\infty}^{\infty} h(t-\tau)x(\tau)\,d\tau \quad (2)$$

One can see the resemblance of (2) to the argument of *f* in (1) with w_i being the samples of h(t) at t=τ. More explicitly, the argument of the nonlinear activation function *f* in (1) describes a system with this impulse response

$$h(t) = \sum_{i=1}^{N} w_i \delta(t-d_i) \quad (3)$$

Substituting (3) into (2) will yield

$$y(t) = \left(\sum_{i=1}^{N} w_i \delta_i(t-d_i)\right) * x(t) = \sum_{i=i}^{N} w_i x_i(t-d_i) \quad (4)$$

Where δ(t) is Dirac's delta function. This somewhat represents a continuous time FIR filter [13]. This configuration not only operates in continuous time, but also with its adaptive delays solves the problem of input shift register length, since in terms of a FIR tapped delay line the changing d_i can continuously slide on an input delay line to any point. Obviously, the higher the N (number of time-delayed input lines), the more flexible the system for approximation of the desired h(t).

The bias term $w_0x(t-d_0)$ does not appear in the above argument. However, the delayed assertion of the bias enables the system to increase or decrease node activations at a certain point in time, such as an anticipated onset of a recurring change in the incoming signal. The sigmoidal output nonlinearity

$$f(u) = \frac{1}{1+e^{-u}} \quad (5)$$

bends the output hyper-plane towards 0 and 1. This facilitates the classification of temporal patterns by helping the output to fit better into the binary target signal.

III-THE TRAINING ALGORITHM

We will use back-propagation for the training of our network. As we will see, back-propagation can still be easily applied to the temporal structure of this network.

First, one needs to define an error function. Considering the utilized sigmoidal nonlinearity, the target output signal can be defined as

$$D(t) = \begin{cases} 1 & \textit{desired pattern} \\ 0 & \textit{otherwise} \end{cases} \quad (6)$$

For the error function, one can extend the static form of the mean squared error (MSE) by summing through time

$$E(t)=\frac{1}{t-t_0}\int_{\tau=t_0}^{t}\left(D(\tau)-y_k(\tau)\right)^2 d\tau \qquad (7)$$

E(t) is a measure of the deviation of the actual output $y_k(t)$ from the desired output D(t), in the span t_0 to t. Ideally, the integral limits should be from the time all the copies of the input signal have reached the output to the time the first one has run its course and left the output node. That is, if the input signal x(t) is exerted during $[t_B\ t_E]$, then ideally the integral limits should be from $t_0=\mathrm{Max}\{d_{ji}\}+t_B$ to $t=\mathrm{Min}\{d_{ji}\}+t_E$, where d_{ji} denotes the value of the delay at the input of the j^{th} hidden layer neuron coming from the i^{th} branch of the input signal x(t). However, a simplified interval, $[t_0\ t]=[t_B\ t_E]$ can still be used. From the networks view, this means that the signal ends earlier by $\mathrm{Min}\{d_{ji}\}$. Since the error will be computed from t_B instead of $\mathrm{Max}\{d_{ji}\}+t_B$, the penalty for the networks with longer overall path delay will be increased.

At this point, we are ready to compute the sensitivity of the network error with respect to the adaptive parameters, namely weights and delays. They will yield the direction of the fastest descent in weight and delay spaces towards the error surface's minima. It is assumed that all the input lines (besides biases) are connected to a single input signal x(t).

A. Error Gradients with Respect to The Weights

Error rates due to and along each weight parameter in the weight space are calculated as below.

1. For the weights connecting the j^{th} hidden node to the output node k, using (7) we can write

$$\frac{\partial E(t)}{\partial w_{kj}}=\frac{1}{t-t_0}\int_{\tau=t_0}^{t}2\left(D(\tau)-y_k(\tau)\right)\left(\frac{\partial D(\tau)}{\partial w_{kj}}-\frac{\partial y_k(\tau)}{\partial w_{kj}}\right)d\tau$$

$$=\frac{2}{t-t_0}\int_{\tau=t_0}^{t}\left(D(\tau)-y_k(\tau)\right)\left(0-\frac{\partial y_k(\tau)}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}}\right)d\tau$$

$$\frac{\partial E(t)}{\partial w_{kj}}=\frac{-2}{t-t_0}\int_{\tau=t_0}^{t}\left(D(\tau)-y_k(\tau)\right)f'(net_k(\tau))y_j\,d\tau \qquad (8)$$

where $net_k(t)=\sum_{j=0}^{M}w_{kj}y_j(t)$ and $y_k=f\left(net_k(t)\right)$. Indices i, j, and k represent the input lines, the hidden nodes, and the output node, respectively. M is the number of hidden neurons.

2. For the weights connecting the i^{th} branch of the input x(t) to the j^{th} hidden node, using (7) we can write

$$\frac{\partial E(t)}{\partial w_{ji}}=\frac{1}{t-t_0}\int_{\tau=t_0}^{t}2\left(D(\tau)-y_k(\tau)\right)\left(\frac{\partial D(\tau)}{\partial w_{ji}}-\frac{\partial y_k(\tau)}{\partial w_{ji}}\right)d\tau$$

$$=\frac{2}{t-t_0}\int_{\tau=t_0}^{t}\left(D(\tau)-y_k(\tau)\right)\left(0-\frac{\partial y_k(\tau)}{\partial net_k}\frac{\partial net_k}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}}\right)d\tau$$

$$\frac{\partial E(t)}{\partial w_{ji}}=\frac{-2}{t-t_0}\int_{\tau=t_0}^{t}\left(D(\tau)-y_k(\tau)\right)\left(f'(net_k(\tau))(w_{kj})f'(net_j(\tau))(x_i(\tau-d_i))\right)d\tau \qquad (9)$$

where $net_j(t)=\sum_{i=0}^{N}w_{ji}x_i(t-d_i)$ and $y_j=f\left(net_j(t)\right)$.

B. Error Gradients with respect to the Delays

For error rates due to and along each delay parameter of the i^{th} branch of the input x(t) to the j^{th} hidden node, using (7) we can write

$$\frac{\partial E(t)}{\partial d_{ji}}=\frac{1}{t-t_0}\int_{\tau=t_0}^{t}2\left(D(\tau)-y_k(\tau)\right)\left(\frac{\partial D(\tau)}{\partial d_{ji}}-\frac{\partial y_k(\tau)}{\partial d_{ji}}\right)d\tau$$

$$=\frac{2}{t-t_0}\int_{\tau=t_0}^{t}\left(D(\tau)-y_k(\tau)\right)\left(0-\frac{\partial y_k(\tau)}{\partial net_k}\frac{\partial net_k}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial x_i}\frac{\partial x_i}{\partial d_{ji}}\right)d\tau \qquad (10\text{-a})$$

Note that the instance of x_i that is responsible for $y_k(t)$ (output at the time t) must have occurred at $(t-d_i)$ due to the delay element d_i. Hence for the last term in the above integrand with the corresponding time argument, i.e. $x_i=x(t-d_i)$, we can write

$$\frac{\partial x_i}{\partial d_i}=\frac{\partial x(t-d_i)}{\partial d_i}=\frac{\partial x(u)}{\partial u}\frac{\partial u}{\partial d_i}=x'(t-d_i)(-1) \qquad (10\text{-b})$$

where $u=t-d_i$ and x' is the time derivative of the input signal x(t). Since this derivative is evaluated at a past time it can be numerically determined. Substitution of (10-b) into (10-a) yields

$$\frac{\partial E(t)}{\partial d_{ji}}=\frac{2}{t-t_0}\int_{\tau=t_0}^{t}\big(D(\tau)-y_k(\tau)\big)\big(f'(net_k(\tau))(w_{kj})f'(net_j(\tau))(w_{ji})(x'_i(\tau-d_i))\big)d\tau \tag{11}$$

Equations (8), (9) and (11) provide the required gradients for the temporal back-propagation learning. Since the gradient of E(t) points towards the direction of maximal change (fastest ascent) on the error surface in the corresponding space, $-\eta\nabla E$ will denote the incremental change towards the minimum on a (local) fastest descent path, where η is the learning rate. Thus, for each iteration n for a node j, considering

$$\vec{\nabla}_{\vec{w}_j}E=\begin{bmatrix}\vdots\\ \frac{\partial E}{w_{ji}}\\ \vdots\end{bmatrix},\ \vec{\nabla}_{\vec{d}_j}E(t)=\begin{bmatrix}\vdots\\ \frac{\partial E}{d_{ji}}\\ \vdots\end{bmatrix};$$

$$\vec{w}_j=\begin{bmatrix}\vdots\\ w_{ji}\\ \vdots\end{bmatrix},\qquad \vec{d}_j=\begin{bmatrix}\vdots\\ d_{ji}\\ \vdots\end{bmatrix}$$

We can write

$$\vec{w}_j(n+1)=\vec{w}_j(n)-\eta_w\vec{\nabla}_{w_j}E(n) \tag{12-a}$$

$$\vec{d}_j(n+1)=\vec{d}_j(n)-\eta_d\vec{\nabla}_{d_j}E(n) \tag{12-b}$$

η_w and η_d are the learning rates for the weight and delay vectors of the j^{th} neuron.

Equations (12-a) and (12-b) show the online training, which has a noisy iterative adaptation. Nevertheless, one can compute the gradient steps for all the patterns in each epoch without updating the operating point and later update the **d** and **w** vectors with an average of the calculated gradients (batch training). We used the smoother batch mode for our simulations.

C. The Case of Negative Delays

Equation (11) along with (12-b) can sometimes generate negative time delays. Since this is not possible in a causal system, and since the network is time invariant, the absolute value of the most negative delay can be added to all the delays in the same iteration to include the effect while reducing the negative delay to zero, that is

$$\exists d<0 \Rightarrow d_{ji}=d_{ji}-Min\{d\},\ \forall i,j \tag{13}$$

Where d denotes any input delay in the hidden layer and d_{ji} is the delay from i^{th} input branch to the j^{th} hidden node.

IV- SIMULATION ANS RESULTS

In this section the results of the simulations of a continuous time delay neural network described by the previous equations will be shown and discussed.

A. Methodology

For the following simulations, Matlab 6.5 and Simulink for continuous systems were used. Simulation settings include

- Number of hidden nodes: 3; number of output nodes: 1.
- Number inputs for each node (besides bias): 3.
- Start time $t_0=t_B=0$, stop time $t=t_E=10$ sec.
- Solver: ode45 (Dormand-Prince) with variable-step.
- Auto min, max, initial step size, and absolute tolerance.
- Relative tolerance=10^{-2}.
- $\eta_d=\eta_w=0.1$ for al the nodes.
- Batch training with two input-target waveform pairs per epoch.

This network passes its output through a Hysteresis block with thresholds set at 0.4 and 0.6. This reduces the effects of possible spurious transients in the 0.4-0.6 band. Needless to say, the back-propagation uses the raw output signal before the Hysteresis block.

B. Waveform Detection

During this simulation, the described network was exposed to 2 training signals, a square wave with a 50% duty cycle, zero phase, unit amplitude, frequency of 1 Hz, and an assigned target of D(t)=0. The second training signal was a zero phase, unit amplitude, 1 Hz sine wave with an assigned target of D(t)=1. Weights and delays were initialized within [-2 2] and [0 1] ranges respectively, using a uniform random distribution. In this simulation, the maximum network delay after training was 1.2941 sec, which is the settle time and the start of the valid output time window $t_0=Max\{d_{ji}\}+t_B$, as discussed in section III. Examining the latency of the network during the training sessions, detection of the temporal pattern (incoming signal x(t)) usually takes place in less than 1 second. Considering the frequency of 1 Hz for the training waveforms, this latency of one period is reasonable. Frequency distortion tests of the trained network show that changing the test input's frequency by a factor of 1.25 falls within the network's generalization span. The same is true for time shifts.

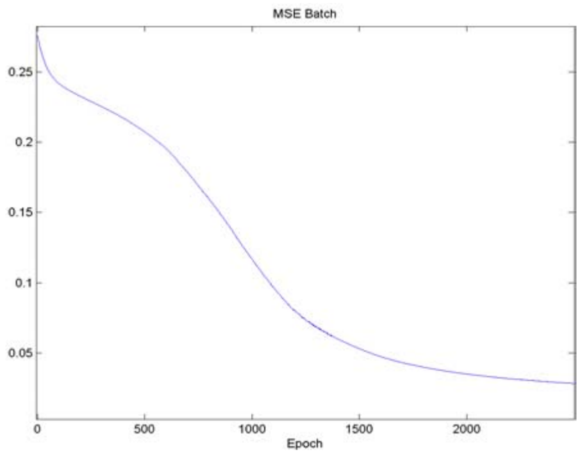

Fig. 3. MSE of the network trained to recognize a sine wave from square wave.

The small jitters in the frequency change tests are an indicator of the departure of the test input from the training pattern. In such cases, the duty cycle of the Hysteresis-conditioned output can be used as a measure of the resemblance of the input to the previously learned temporal patterns. Figure 4 shows a real time switching between a square and a sine waveform at the input, which generates the corresponding expected outputs in less than 1 period.

Probably the most interesting result is the one shown in Figure 5. The network shows a very good noise immunity and robust behavior by correctly detecting both the noisy square wave and then the following noisy sine wave. The additive band-limited noise signal (power=0.0015 and sample time=0.04 sec) has noticeably distorted the test input signal. Besides the transient bounces after the onset at t=4 sec, no other output and classification jitters can be seen.

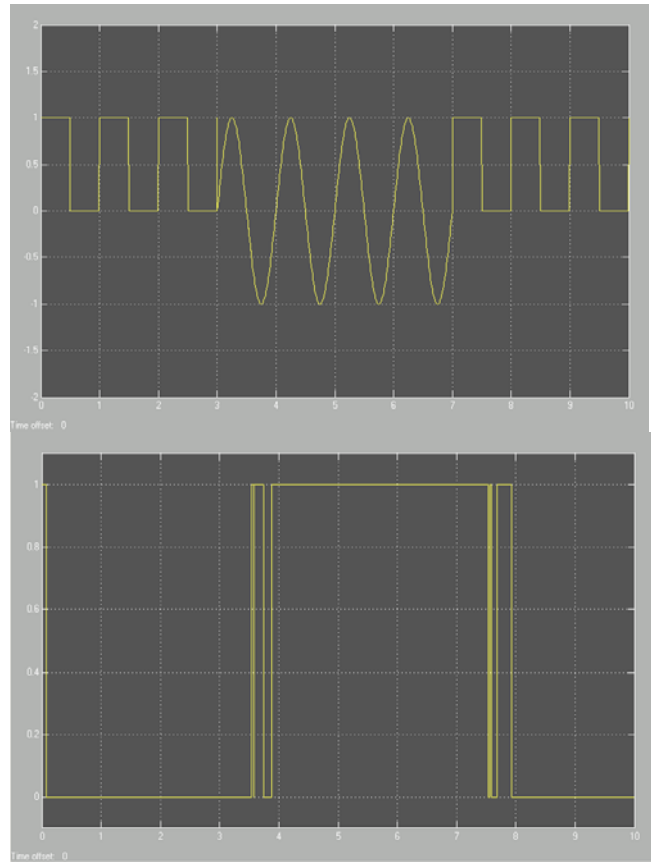
Fig. 4. Top: network's onset detection test by switching from square wave to sine wave at t=3 and back to square at t=7. Bottom: output after Hysteresis.

V- CONCLUSIONS AND FUTURE WORK

Spatio-temporal pattern recognition is an important field that has been less discovered. Time delay neural networks with their short-term memory provided by delayed signal paths have shown promising results in some applications. Here we derived a set of equations that utilize back-propagation of temporal errors to adapt both the weights and the delays of a continuous time delay neural network. We showed that the resulting network could acquire robust temporal pattern recognition capabilities for continuous time signals. This category of classifiers is especially important since there are increasingly massive volumes of temporal data that are being produced by researchers (especially in medical fields) in form of continuous time series and signals. With the promising initial results depicted here, the next step will be training such a network with massive and complex real world data. However, as mentioned earlier during the analysis of the hidden nodes, the number of input lines per each node and the network topology in general are the factors that determine the learning capability of the network. Even though no analytical solution is available for finding such an optimal network configuration, one can use general search methods such as evolutionary computing techniques to optimize network's topology [14] for the task at hand, which we shall pursue in future. It will also be interesting to see how the temporal delay patterns will evolve in such networks. This may shed a light on the controversial issues regarding the role of synaptic delays in neuronal ensembles.

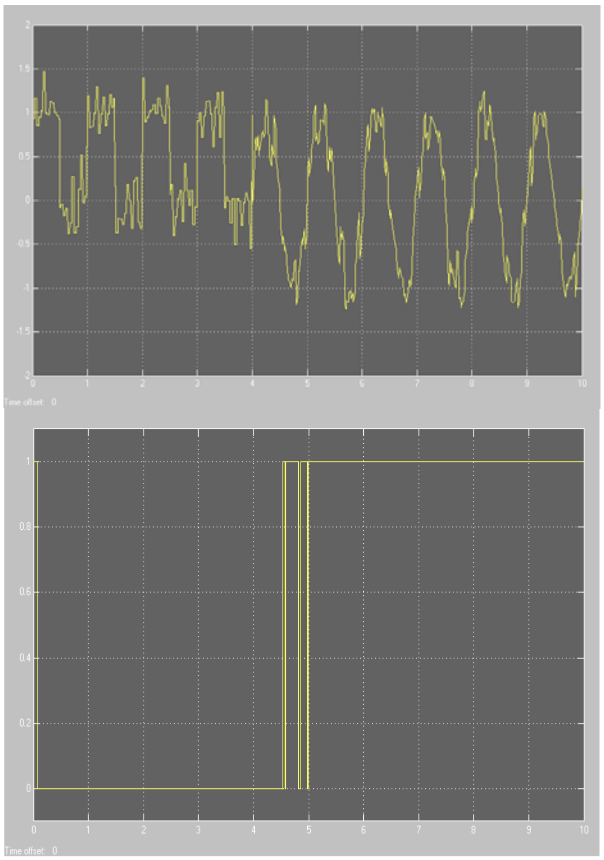

Fig. 5. Top: network's noise tolerance and onset detection test by switching from a distorted square wave to a distorted sine wave at t=4, while a band limited noise signal (power=0.0015, sample time=0.04 sec) is constantly being added. Bottom: output after Hysteresis.

References

[1] Wang, D. (1995) "Temporal Pattern Processing," The Handbook of Brain Theory and Neural Networks, Arbib M. Ed. MIT Press, Cambridge, MA.

[2] Derakhshani R, and Schuckers SAC (2002) "Biologically Inspired Evolutionary Temporal Neural Circuits." Proceedings of IEEE World Congress on Computational Intelligence, Honolulu, HI,

[3] Guigon, A., and Burnod Y. (1995) "Short Term Memory," The Handbook of Brain Theory and Neural Networks, Arbib M. Ed. MIT Press, Cambridge, MA.

[4] Bourlard, H., and Morgan, N. (1995) "Speech Recognition, Pattern Matching," The Handbook of Brain Theory and Neural Networks, Arbib M. Ed. MIT Press, Cambridge, MA.

[5] Mead C., (1989), Analog VLSI and Neural Systems, Reading, MA, Addison-Wesley.

[6] Milton, J, et al (1995) "Traveling Activity Waves," The Handbook of Brain Theory and Neural Networks, Arbib M. Ed. MIT Press, Cambridge, MA.

[7] Johnston, D. and Wu, S.(1997), Foundations of Cellular Neurophysiology, MIT Press, Cambridge, MA.

[8] Principe JC, Euliano NR, Lefebvre WC (2000), Neural and Adaptive Systems, John Wily, NY

[9] Waibel A., Hanazawa T., Hinton G., and Lang, K. (1989), "Phoneme recognition using time-delay neural networks," IEEE Trans. Acoust.,Speech, Signal Processing, vol. 37, pp. 328–339,.

[10] Wan E. A., (1990) "Temporal backpropagation for FIR neural networks," in IEEE Int. Joint Conf. Neural Networks, San Diego, CA, pp.575–580.

[11] W¨ohler C. and Anlauf J. (1999) "An Adaptable Time-Delay Neural-Network Algorithm
for Image Sequence Analysis," IEEE Trans. on Neural Networks, November vol. 10, No. 6, pp 1531-1536.

[12] S. P. Day and M. R. Davenport, (1993). "Continuous-time temporal backpropagation with adaptable time delays," IEEE Trans. Neural Networks,vol. 4, pp. 348–354.

[13] Oppenheim, A. and Willsky, A. (1983), Signals and Systems,Prentice Hall, Englewood Cliffs, NJ.

[14] Yao, X. (1999), "Evolving Artificial Neural Networks," Proceedings of IEEE, September, vol 87, No. 9,pp 1423-1447.