Single and Multiple Frame Video Traffic Prediction Using Neural Network Models

Radu Drossu[1]

T.V. Lakshman[2]

Zoran Obradović[1, *]

C. Raghavendra[1]

[1] School of Electrical Engineering and Computer Science

Washington State University, Pullman WA 99164-2752

[2] Bellcore, Red Bank, NJ 07701

*Research sponsored in part by the NSF research grant NSF-IRI-9308523.

Abstract

The objectives of this paper are to investigate applicability of neural network techniques for single and multiple frame video traffic prediction. In the *single* and *multiple* frame traffic prediction problems, the information of previous frame sizes is used to predict either the following or several following frame sizes respectively. respectively. Accurate traffic prediction can be used to optimally smooth delay sensitive traffic [15] and increase multiplexing gain in asynchronous transfer mode (ATM) networks. Neural network models for both single and multiple frame traffic prediction problems are proposed. Two important types of video sequences are considered - video teleconferencing and entertainment video. An off-line learning method is suggested for simple traffic and an on-line learning method for complex one. Simulation studies of cell losses in an ATM multiplexer using recorded variable-bit-rate coded video teleconference data indicate reasonably good predictions for buffer delays between 0.5 and 5 ms.

1. Introduction

In the last few years a substantial amount of research has been performed in the area of neural networks. The experimental work indicates their potential for practical applications where traditional computation structures have performed poorly (e.g. ambiguous data [1] or large contextual influence [2]).

In addition to the traditional analytical approximation techniques, recently neural networks have also been proposed for some ATM control problems. For example in [7, 17] a neural networks learning method is suggested for service admission control in the ATM networks. The objective of this application is to keep the requested service quality parameters by rejecting some of the call set-up requests while connecting as many calls as possible. In [12], a neural network is used to refine a traditional analytical approximation technique for admission control for improved bandwidth efficiency. Another promising application for neural networks in ATM is call control [11]. Simulation results show that neural networks can lead to a compromise between the maximum number of calls accepted and the satisfaction of the quality of services negotiated for established calls.

In this paper we explore whether neural networks prediction techniques can be used for video traffic prediction. Accurate source traffic models and traffic predictions are needed for effective utilization and control of ATM networks. Video traffic consists of a periodic arrival of frames (25 frames per sec for our traffic sequence) with a variable number of ATM cells per frame. The number of cells per frame varies significantly. Predictions of the number of cells per frame can be used for improving

ATM network efficiency by incorporating the predictions in schemes for multiplexing, routing, smoothing and bandwidth allocation. As an example, statistical models of video sources developed in [6] have been used to optimally smooth video traffic [15] and to increase multiplexing gain. We use neural networks to predict the number of cells per frame for two important classes of video applications - video teleconferencing and entertainment video.

The computational model used in the paper is a neural network with *sigmoidal* computing units. Each unit computes the function $1/(1+e^{-\beta x})$, where x is the units weighted input sum. The computation units are organized in layers with connections only among units in adjacent layers. In the simplest form of such a *layered network,* there is just one *input layer* of source units that projects onto an *output layer* of computation units. In *multilayer networks* there is one or more *hidden layers*, whose computation units are accordingly called *hidden units*. The network is said to be *fully connected* if each node in each layer of the network is connected to every other node in the adjacent forward layer. By adding one or more hidden layers, the network is enabled to extract higher-order statistics from the data (to approximate more complicated functions). A network is called *feedforward*, if all the directed links between different layers of units start in a layer that is closer to the input layer and end in a layer that is closer to the output layer.

The proposed neural network model and learning algorithms for single frame prediction are explained in Section 2, followed by the multiple frame prediction in Section 3. In Section 4 the experimental results are presented.

2. Single Frame Prediction

Let $u(k)$ be the frame size at time k and $y(k+1)$ be the predicted frame size at time $k+1$. We assume that $y(k+1)$ is a function of $u(k), u(k-1), \ldots, u(k-n+1)$. The model we use for predicting this function is a fully connected i feedforward multilayer neural network (FNN) with sigmoidal computation units. In our model, the FNN has n input units, one layer of hidden units and one output unit. The architecture is shown in Fig. 1, where the blocks denoted by z^{-1} represent one step delay elements. Learning is performed by the back-propagation algorithm (BP), which is a gradient-descent iterative method for minimization of the total squared prediction error on a set of training examples[16]. Here, each example has the form $\langle u(t-n), \ldots, u(t-1), u(t) \rangle$, where $\langle u(t-n), \ldots, u(t-1) \rangle$ is used as an input to the neural network and $u(t)$ is used for comparison to the predicted frame size $y(t)$.

A viable alternative to FNN with BP is the radial basis function network (RBFN) [9] which is a local approximation technique. Time series prediction research [10] indicates that this approach might give very good results and fast response if the information needed for prediction is contained in a small number of previous frames. Otherwise, the prediction would require too large radial basis networks.

Both networks have advantages and disadvantages: FNN with BP learns slowly because of the gradient descent technique used for training, but is usually the choice when dealing with complex structures having a large number of inputs, whereas RBFN is very fast, but the number of radial basis units tends to increase exponentially with the dimension of the input space, so that this approach becomes practically infeasible

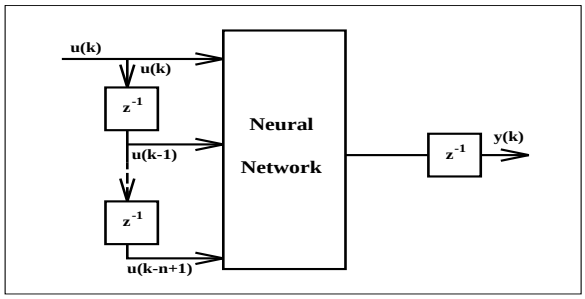

Figure 1: Neural Network Structure Used in the Simulation

when the dimensionality of the input space is high.

2.1. Off-line learning for simple video signal

For video teleconference traffic, the recorded traffic sequence does not have any abrupt fluctuations because there are no significant scene changes. So we first test if simple *off-line learning* is appropriate for the video teleconference sequence prediction problem. A sequence of T examples $\langle u(t-n), \ldots, u(t-1), u(t) \rangle$, where $n+1 \leq t \leq T$ is used for neural network training. After a training phase of N epochs the weights are frozen and the system is used for prediction on the rest of the available sequence (for $t > T$ on input $\langle u(t-n), \ldots, u(t-1) \rangle$, the predicted output is $y(t)$).

The advantage of this approach is that it is extremely fast (after the initial training phase) which makes it applicable to real-time forecasting. The drawback is that the method assumes a fixed distribution signal, which is not true for complex video sequences such as entertainment video.

2.2. On-line learning for complex video signal

For video traffic with a number of sudden scene changes, frame cuts, zooms, rapid movement of objects, the *continuous on-line learning* seems more appropriate. The learning is performed for M epochs on S consecutive examples $\langle u(t-n), \ldots, u(t-1), u(t)\rangle$, where $n+1 \leq t \leq S$ and the network is used to predict $y(S+1)$ which is the size of the next frame. Whenever a new frame becomes available, the training set is shifted to include this new frame and the oldest frame is discarded ($t \leftarrow t+1$). A new learning session of M epochs is performed on this new training set and the next frame size prediction is obtained. The process is then repeated. The weights are adjusted continuously during the whole process. After the training set is shifted, the new training session of M epochs starts from the existing weight values rather then from random values. Therefore it is reasonable to expect convergence in a significantly smaller number of epochs as compared to the off-line learning ($M \ll N$). Since the learning has to be performed in real time, we also reduce the number of training examples as compared to the off-line learning ($S < T$). The on-line learning algorithm can also have an additional initial learning phase, similar to the learning phase of the off-line algorithm, so that the on-line learning starts with properly initialized weights.

The advantage of the on-line approach is the continuous learning on new examples that should cope with changes in the distribution for complex sequences. The problem of this approach is that it might be computationally too expensive for real-time response if experiments show that for accuracy reasons M or S have to be large. In such a case, an alternative to standard backpropagation learning is a more effi-

cient parallel learning that we have proposed earlier and applied successfully to other problems of large scale [8, 13, 18].

Another possible learning approach for complex signals is the recently proposed growing cell structures network [5]. This local approximation algorithm is very fast and in contrast to regular RBFN, it is still appropriate for larger dimension problems. For very complex signals, an appropriate approach is to combine domain specific prior knowledge and constructive learning from examples into a hybrid system, as proposed in [3, 4].

3. Multiple Frame Prediction

To optimally smooth delay sensitive traffic we examine to what extent neural networks can do a multiple frame size prediction. More precisely, in an s-frame prediction problem given the actual frame sizes $u(k), u(k-1), \ldots, u(k-n+1)$, the network has to predict not only $y(k+1)$ as earlier, but also $y(k+2), \ldots, y(k+s)$, where $s > 1$. Obviously, this is a more challenging problem compared to the previous 1-frame prediction, and so the accuracy might be compromised if s is too large. It is an experimental question of how many previous frames are needed for s-frame prediction and how many frames ahead (maximum value for s) neural networks can predict without a significant decay in accuracy.

The first approach for multiple frame prediction (*incremental approach*) has a training phase as earlier (using an off-line or on-line technique depending on the signal complexity). In the prediction phase $y(t+1)$ is computed using $\langle u(t-n), \ldots, u(t) \rangle$

as the network input. Prediction for the frame size at time $t+2$ is computed using $\langle u(t-n+1), \ldots, u(t), y(t+1)\rangle$ (oldest actual frame size is discarded and the previous prediction - $y(t+1)$ is used instead of the corresponding actual value). Similarly, $y(t+3)$ is predicted using $\langle u(t-n+2), \ldots, u(t), y(t+1), y(t+2)\rangle$, and the process continues until $y(t+s)$ is computed using $\langle u(t-n+s-1), \ldots, u(t), y(t+1), \ldots, y(t+s-1)\rangle$.

Another approach for multiple frame prediction (*direct approach*) is to learn a mapping $f : N^n \rightarrow N^s$ from the previous n to the next s frames. In this model the network has s output units rather than one. The training phase uses examples of the form $\langle u(t-n), \ldots, u(t+s)\rangle$ where the first n components are used as the network input and the rest to compare the computed response versus the actual frame sizes. The advantage of this direct approach is that the prediction error is not accumulating as in the incremental approach. On the other side this approach has more free parameters in the system. So the learning process is computationally more demanding and also a larger training set is needed for a good generalization.

4. Experimental Results

A number of experiments were performed by varying the training set size, the network input layer size, the number of hidden units, the training time (number of epochs) and the learning rate. This section contains a short summary of the obtained results. Sections 4.1 and 4.2 present the results for the single frame prediction problem, while Section 4.3 presents the multiple frame prediction.

4.1. Video Teleconference Data

The experiments on video teleconference sequence forecasting, were done by using recorded and compressed real video conference data consisting of 40,000 frames previously used in [6]. The first 1000 frames are shown in Fig. 2.

Reasonably good prediction results are obtained using the off-line learning on a neural network with 5 inputs, 5 units in one hidden layer and a single output unit. Learning is performed on the first 666 frames for 10000 epochs and testing is done on the next 329 frames. Fig. 3 shows the quantile-quantile plot of predicted versus actual data. The generalization stays approximately the same for smaller number of training epochs and longer test sequences (up to all available 40,000 frames). Fig. 4 shows the results obtained using an identical network and the same training examples as previously, but with 2000 training epochs and a prediction of 9000 frames. The autocorrelation function for the actual and the predicted data corresponding to Fig. 4 is shown in Fig 5. The autocorrelation functions are exponentially decreasing and the fit between actual and predicted data is good.

The predicted sequence is used in simulations in which 25 multiplexed video sources feed a finite buffer queue which models the queue at the output port of an ATM switch. The output rate from the queue is 56 Mbps. The buffer size at the queue is specified in terms of maximum delay that can be tolerated at the queue. Maximal delay is varied from 0.5 ms to 35 ms. Incoming traffic (in the form of ATM cells) is lost if it arrives to a full buffer. The results of a number of simulations is shown in Table 1 and 2. In these tables the buffer size is specified in terms of maximal delay (i.e. 0.5

buffer	0.5	1	2	3	4	5
predicted	4.104e-05	3.830e-05	3.447e-05	3.109e-05	2.802e-05	2.495e-05
actual	7.444e-05	6.988e-05	6.184e-05	5.592e-05	5.183e-05	4.808e-05

Table 1: Simulation using small buffer and predicted teleconference sequence

buffer	11	12	13	14	15
predicted	1.156e-05	1.002e-05	8.486e-06	6.937e-06	5.768e-06
actual	3.647e-05	3.497e-05	3.347e-05	3.196e-05	3.046e-05

Table 2: Simulation using larger buffer and predicted teleconference sequence

ms, 1 ms, 2 ms, etc) and the cell loss probabilities corresponding to these buffer sizes are specified for the predicted and the actual sequences. The fit is reasonably good for traffic purposes for small buffers (accurate to within a factor of 2 or 3) but for large buffers the actual and predicted losses diverge.

Slightly better results were obtained by using on-line learning. Fig. 6 shows the quantile-quantile plot of predicted versus actual data for predicting 1000 values. The neural network used has 5 input units, one hidden layer with 5 units and one output unit and learning is performed using a window of $S = 100$ frames and $M = 200$ epochs.

4.2. Entertainment Video Data

Entertainment video has frequent scene changes which makes the prediction problem very difficult. Supervised neural networks learning seems quite appropriate for such noisy prediction problems. The experiments on entertainment video sequence forecasting, have been done by using recorded and compressed real data consisting of

12,000 frames. The global distribution for this data set is unknown, although certain small subsequences of data can be identified to have a gamma distribution. The first 4000 frames are shown in Fig. 7.

Fig. 8 shows the quantile-quantile plot of predicted versus actual data obtained using the off-line learning on a neural network with 5 inputs, 5 units in one hidden layer and a single output unit. Learning is performed on the first 666 frames for 2000 epochs and testing is done on the next 3000 frames. From Fig. 8 it is clear that the losses are caused mainly by the scene changes (peaks). The autocorrelation function for the actual and the predicted data used in the entertainment video test corresponding to Fig. 8 is shown in Fig. 9. It can be observed that the fit between actual and predicted data is still good, although the exponential character of the curves is lost.

The on-line experiments were similar to those performed for the video teleconference data, but the results were fairly poor. Although the predictions had a tendency of sensing the scene changes, the errors were excessive. We believe that the reason for this behavior is the slow backpropagation learning.

4.3. Multiple Frame Prediction for Video Teleconference Data

The previously explained incremental approach is used for 2-frame and 3-frame prediction experiments on video teleconference data. The neural network and the data set used were exactly the same as those used in Test 2 (corresponding to Fig. 4). Figures 10 and 11 present the quantile-quantile plots of predicted versus actual data for the 2-frame and the 3-frame problems respectively. In the 2-frame problem, the

accuracy of the prediction is measured by comparing the second predicted frame size versus the corresponding actual value. Similarly, in the 3-frame problem the accuracy is measured by comparing the third predicted frame size versus the corresponding actual value. By comparing Figures 4, 9 and 10 it can be observed that in the s-frame prediction problem the accuracy of the prediction is decreasing as s increases.

These preliminary results on the multiple frame prediction problem suggest that the direct approach might give better results since the error accumulated in the incremental approach is too large even for predicting the third following frame.

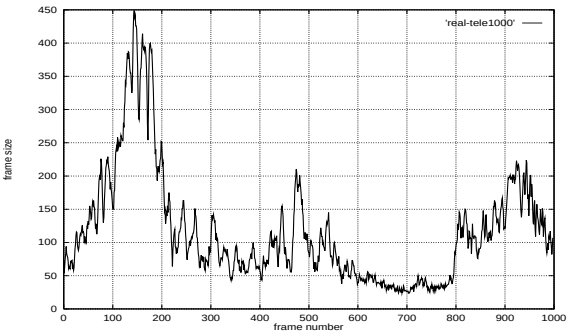

Figure 2: Video Teleconference Data

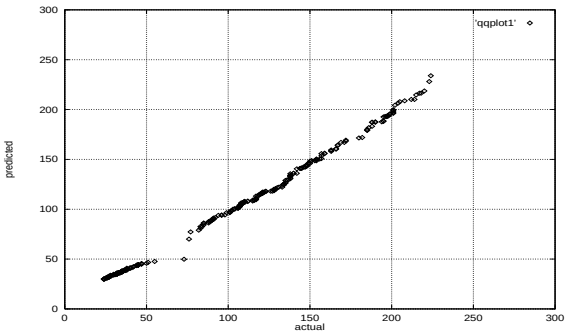

Figure 3: Video Teleconference Test 1

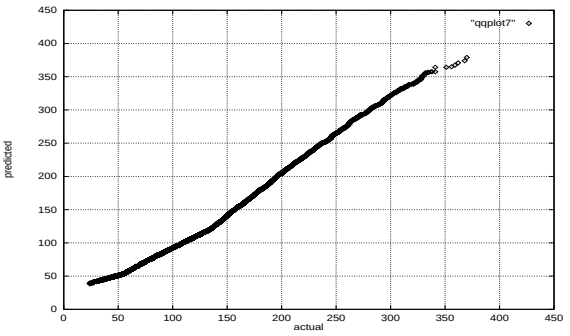

Figure 4: Video Teleconference Test 2

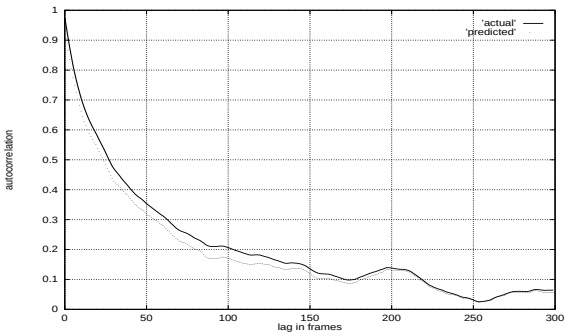

Figure 5: Autocorrelation for Test 2

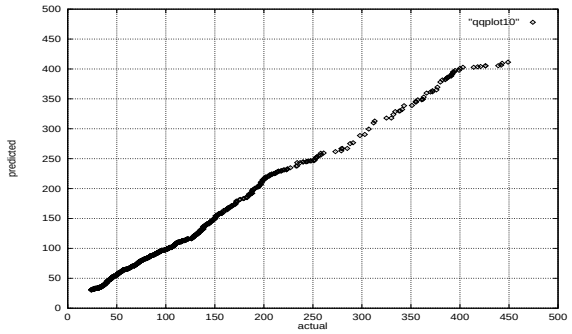

Figure 6: Video Teleconference Test 3

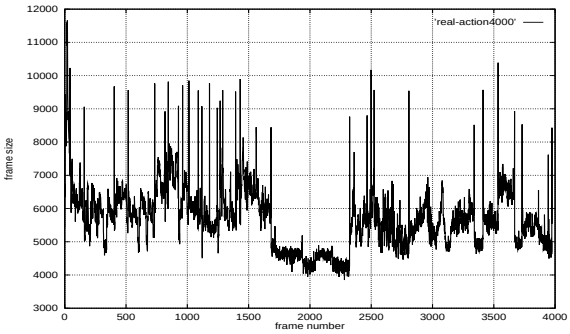

Figure 7: Entertainment Video Data

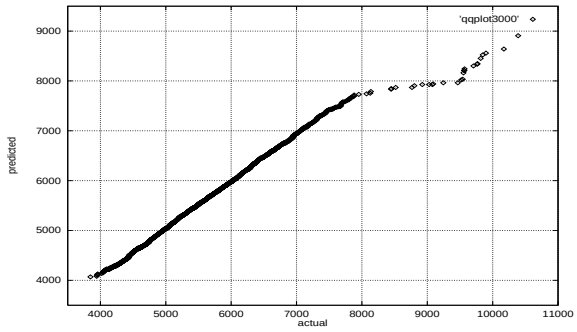

Figure 8: Entertainment Video Test

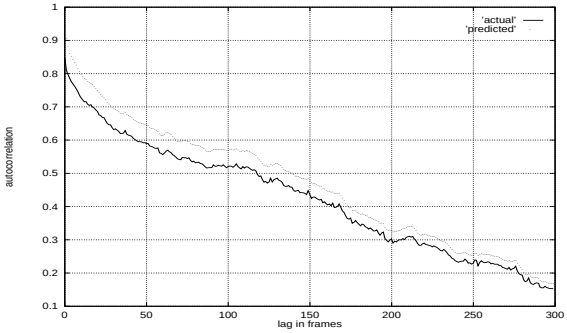

Figure 9: Autocorrelation for Video Test

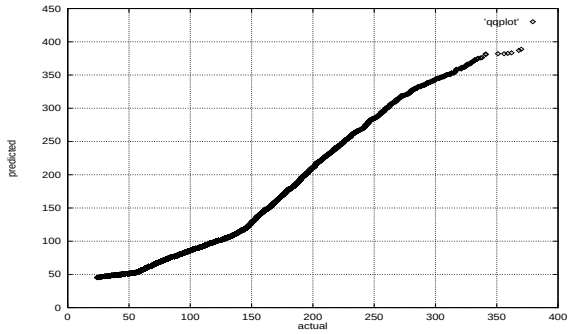

Figure 10: 2-nd Frame for Test 2

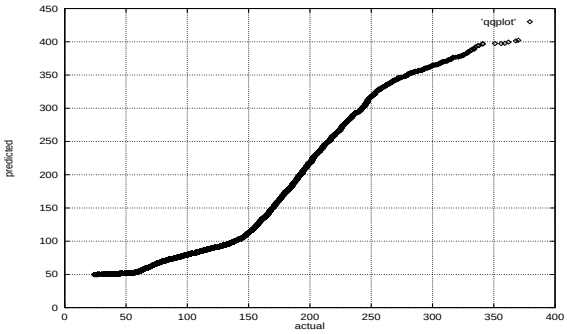

Figure 11: 3-rd Frame for Test 2

5. Conclusion

The experimental results indicate that the single frame problem for the video teleconference data can be predicted with reasonably good accuracy using both on-line and off-line learning methods. The entertainment video data is more difficult to predict

but the off-line learning method still gives promising results. The on-line learning method in its actual form can not be used for predicting bursty data in real-time.

Preliminary results on the multiple frame problem using the incremental learning approach indicate that the method is inappropriate for prediction using only information contained in a small number of previous frames. The direct learning approach for this problem is still under investigation. Also, the experimentation with radial basis function networks for on-line learning of the video entertainment data is in progress.

Traffic prediction has application in smoothing video traffic in an ATM network, which is useful in network traffic management. Probabilistic models used for video traffic prediction have been successfully applied to traffic smoothing with some constraints [14]. The application of the neural networks based traffic prediction model to traffic smoothing and to other traffic management problems are currently under consideration.

References

[1] Y. Le Cun et al., "Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning," *IEEE Communications*, 1989, pp. 41-46.

[2] R.C. Eberhart, R.W. Dobbins, "Case Study I: Detection of Electroencephalogram Spikes," in *Neural Networks PC Tools*, ed. R.C. Eberhart, R.W. Dobbins, San Diego, CA, Academic Press, 1990.

[3] J. Fletcher, Z. Obradovic, "Combining Prior Symbolic Knowledge and Constructive Neural Networks," *Connection Science Journal,* vol. 5, Nos 3 and 4, 1993, pp. 365-375.

[4] J. Fletcher, Z. Obradovic, "Parallel Constructive Neural Network Learning," *Proc. IEEE 2nd Int. Symp. on High-Performance Distributed Computing,* Spokane, WA, 1993, pp. 174-178.

[5] B. Fritzke, "Growing Cell Structures - A Self-Organizing Network in K Dimensions," in *Artificial Neural Networks 2,* eds. I. Aleksander and J. Taylor, North-Holland, 1992, pp. 1051-1056.

[6] D. P. Heyman, A. Tabatabai, T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks," *IEEE Transactions on Circuits and Systems for Video Technology,* Vol. 2, No. 1, March 1992, pp. 49-59.

[7] A. Hiramatsu, "ATM Communications Network Control by Neural Networks," *IEEE Transactions on Neural Networks,* Vol. 1, No. 1, March 1990, pp. 122-130.

[8] I. Mehr, Z. Obradovic, "Parallel Neural Network Learning Through Repetitive Bounded Depth Trajectory Branching," *Proc. 8th IEEE Int. Parallel Processing Symposium*, Cancun, Mexico, 1994, pp. 784-791.

[9] J. Moody, C. Darken, "Learning with Localized Receptive Fields," Connectionist Models Summer School, Morgan Kaufmann, 1988, pp. 133-143.

[10] K. S. Narendra, "Adaptive Control of Dynamical Systems Using Neural Networks," in *Handbook of Intelligent Control,* ed. D.A. White and D.A. Sofge, Van Nostrand Reinhold, 1992, pp. 141-183.

[11] J.E. Neves, L.B. de Almeida, M.J. Leitao, "ATM Call Control by Neural Networks," *Proc. Int. Workshop on Applications of Neural Networks to Telecommunications,* Princeton, N.J., 1993.

[12] E. Nordstrom, "A Hybrid Admission Control Scheme for Broadband ATM Traffic," *Proc. Int. Workshop on Applications of Neural Networks to Telecommunications,* Princeton, N.J., 1993.

[13] Z. Obradovic, R. Srikumar, "Evolutionary Design of Application Tailored Neural Networks," *Proc. IEEE Int. Symp. on Evolutionary Computation,* Orlando, FL, 1994.

[14] T.J. Ott, T. V. Lakshman, A. Tabatabai, "Smoothing Traffic with Real-Time Constraints Using Arrival Forecasts," Manuscript submitted to *IEEE Transactions on Communications.*

[15] T. J. Ott, T. V. Lakshman, A. Tabatabai, "A Scheme for Smoothing Delay-Sensitive Traffic Offered to ATM Networks", *Proc. of INFOCOM 1992,* pp. 776-785.

[16] D.E.Rumelhart, G.E.Hilton and R.J.Williams, "Learning Internal Representations by Error Propagation," *Parallel and Distributed Processing*, Eds. D.E.Rumelhart and J.L.McClelland, Cambridge, MA, MIT Press, 1986.

[17] T. Takahasni, A. Hiramatsu, "Integrated ATM Traffic Control by Distributed Neural Networks," *Proc. 13th Int. Switching Symp.*, Vol. 3, pp. 42-56.

[18] R. Venkateswaran, Z. Obradovic, "Efficient Learning through Cooperation." *Proc. 1994 World Congress on Neural Networks,* San Diego, CA, 1994.