Neural Network Based ATM Connection Admission Controllers – A Comparative Study

Tong-Seng Quah, Poh-Keng Lim
School of Electrical and Electronic Engineering
Nanyang Technological University
Nanyang Avenue
Singapore 639798
Republic of Singapore

Keywords
ATM Network, Back Propagation Neural Network, Learning Vector Quantization, Supervised Training, Cell Loss Probability

Abstract
The congestion control problem is one of the challenging problems facing ATM designers. Proper control of congestion is absolutely necessary for providing QoS for all the services supported by the network. Multimedia sources such as voice and video are bursty in nature. There may be many bursts during a call and the bursts themselves may have a variable number of cells. Since the idea of high-speed ATM networks is to provide bandwidth on demand, the effective bandwidth provided to a source could be much less than its peak rate. Reactive traffic control is based on the use of feedback information from the congested node to the source nodes for flow control. Such schemes are not very effective at high speeds. The alternative strategy is preventive control. By employing effective schemes at the network entry points, preventive control prevents the network from reaching undesirable levels of congestion. As a result, admission control is absolutely necessary and critical for the proper functioning of the ATM network. This research studies the effectiveness of two neural architectures for use in connection admission controllers and performs a comparison.

1. Introduction
The Asynchronous Transfer Mode (ATM) is meant to be the transfer technique for Broadband Integrated Services Digital Networks (B-ISDN). ATM is a packet-oriented switching and multiplexing technique designed to meet the different bandwidth and quality of service (QoS) demands of B-ISDN services. One of the key issues in ATM networks that is under heavy research is congestion control. The main preventive congestion control schemes are connection admission control (CAC), traffic policing, priority scheduling, and adaptive routing. The focus of this research is on CAC.

CAC is defined as the set of actions taken by the ATM network during the call setup phase in order to determine whether a virtual channel/virtual path connection request can be accepted or rejected. CAC is a key component in preventive congestion control. Different analytical approaches have been proposed to develop an effective admission control mechanism. However, although accurate traffic performance models exist, they are often too complex to be employed in a real-time ATM network.

Thus artificial neural networks have been proposed to perform the CAC function, since neural networks have the ability to capture complex nonlinear relationships and classify traffic conditions into acceptable and unacceptable levels. This project seeks to conduct a comparative study of two traffic controllers of different neural architectures.

Specifically, the following neural architectures are studied and compared:
- Back-Propagation Neural Network
- Learning Vector Quantization Neural Network

2. Background
Over the past decade, a substantial number of CAC schemes have been proposed. Generally, CAC schemes may be classified
as conventional, fuzzy-logic-based, or neural-net-based [2].

Neural-network-based CAC schemes provide learning and adaptation capabilities, which reduce the estimation errors of conventional CAC, and achieve performance similar to that of a fuzzy logic controller. The self-learning capability of neural networks has been applied in many research work to characterize the relationship between input traffic and system performance [3,6,8]. The ability of neural networks to adapt to changing traffic situations combined with prediction capabilities has made them increasingly popular alternatives to analytical methods.

There are many different versions of neural CACs which vary mainly in the traffic parameters that are to be related to one or more quality of service (QoS) parameters (e.g. cell delay, and cell loss rate). These parameters may be an observation window for the arrival cell process, or more frequently, the state of the system represented by a vector of active connections for each traffic type [1,4,5,7]. In any case, all the above methods need to acquire patterns to train the neural network that decides whether to accept or reject a set-up request from a new connection. In other words, the CAC problem is being transformed to a pattern recognition problem.

3. Methodology

Two neural network architectures are chosen for this research: back-propagation (BP) network and Learn Vector Quantization (LVQ) network. As BP network is fairly well known, we will assume knowledge of such network in this paper.

The Learning Vector Quantization (LVQ) network is a hybrid network, which uses both unsupervised and supervised learning to form classifications. It basically trains the competitive Kohonen layer in a supervised manner. The Kohonen layer will automatically learn to classify input vectors. However, the classes that the competitive layer identifies are dependent only on the distance between input vectors. Thus, if two input vectors are very similar, the competitive layer will put them in the same class. There is no mechanism in a strictly competitive layer design to say whether any two input vectors are in the same or different class. LVQ networks, however, learn to classify input vectors into target classes specified by the user. This network is described in detail in :Kohonen, T. Self-Organization and Associative Memory, 2nd Edition, Berlin: Springer-Verlag, 1987.

4. Research Implementation

Implementation of the Neural Admission Controller can be summarized as follows:

- To use the Neural Network Toolbox for MATLAB to construct the various neural networks.
- To generate training and testing data sets using burst scale loss analysis. This includes data normalization during the pre-processing phase.
- After sufficient training, to test and compare performance of the networks.

In ATM, not only must the route be found, but also a check must be made at each link on a proposed route to ensure that the new connection, with whatever traffic characteristics, can be supported without violating the negotiated performance requirements of connections established over each link.

There are three traffic parameters which are important in determining the type of queueing behavior: peak cell rate, mean cell rate, and the average active state duration. If we allow the peak cell rate to exceed the service capacity, this is one form of "statistical bit-rate (SBR) transfer capability" (see ITU Recommendation I.371 Traffic Control and Congestion Control in B-ISDN, Geneva, July 1995).

In particular, the behavior of variable bit-rate (VBR) traffic is modeled and studied. The VBR source can be viewed as an ON/OFF source, which will generate a burst of cells during an ON period. Queueing occurs when the total
number of cells arriving from simultaneous
bursts exceeds the number of cell slots in
that "simultaneous" period. For VBR, there
is a long inactive OFF period followed by
an active state producing a burst of cells.

Based on the burst scale loss
analysis, and that each VBR source has an
average rate of \( m \) cells/s, so with \( N \) sources
and bandwidth of \( C \) cells/s, the utilization is
given by

\[
\rho = \frac{Nm}{C}
\]

Unfortunately there is no simple
approximate formula that can be
manipulated to give the admissible load \( \rho \) as
an explicit function of the traffic contract
parameters. The best that can be done is to
use the approximate formula for the burst
scale loss factor:

\[
CLP \approx \frac{1}{(1-\rho)^2 N_0} \left( \rho N_0 \right)^{N_0} e^{-\rho N_0}
\]

where \( N_0 \) is the number of peak cell rates
which fit into the service capacity. Thus, the
formula can be used to generate a table of
maximum admissible load, which allows us
to specify the required cell loss probability
and the source peak cell rate and find out the
maximum allowed utilization. We can then
calculate the maximum number of sources
of this type (with mean cell rate \( m \)) that can
be accepted using

\[
N = \frac{\rho C}{m}
\]

The aim then is to construct the
neural network to learn the function above.
The CAC problem is then formulated as a
pattern recognition problem of the traffic
load pattern reflected in the load table.

In this case,

\[
X = \{ CLP, N_0, \rho \} \quad \text{and} \quad Z =
\begin{cases}
0, & \text{for a reject decision} \\
1, & \text{for an accept decision}
\end{cases}
\]

After the learning phase, the neural
network can be used in the recall mode to
perform the CAC function. The neural
network performs the CAC by separating the
3-dimensional input state space into 2
regions of "Accept" and "Reject"
corresponding to a 2-dimensional decision
boundary surface.

For the LVQ network, the inputs are
a set of values of a combination of CLP, \( N_0 \)
and \( \rho \). Each combination is labeled as a
"Accept" or "Reject" situation, i.e. strictly

\[
CLP = |0\log CLP|
\]

"1" or "0", which is hard-limiting in nature.
Thus, the trained LVQ network has to
decide upon presentation of an input vector
of CLP, \( N_0 \) and \( \rho \), whether to classify it as
an "Accept" or "Reject" case. For the
network to train properly, some pre-
processing of the data needs to be
performed. The CLP is transformed using:

For the BP network, the inputs are
also the vector combination of CLP, \( N_0 \)
and \( \rho \). Similar pre-processing was also
performed. In this case, there is no hard
limiting of output as in the case of LVQ.
Each combination is mapped to a 2-element
output of (0,1) or (1,0). However, the values
can assume decimal values, as the function
used is log-sigmoid.

5. Results

The performance is rated according to how
successful the 2 networks are in
generalization of the test data.

Table: Comparison of LVQ and BP
networks.

<table>
<thead>
<tr>
<th></th>
<th>LVQ</th>
<th>BP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Size</td>
<td>3x32x2</td>
<td>3x16x16x2</td>
</tr>
<tr>
<td>Size of testing samples</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Success rate in</td>
<td></td>
<td></td>
</tr>
<tr>
<td>generalization (%)</td>
<td>79.5</td>
<td>90.7</td>
</tr>
<tr>
<td>Number of training</td>
<td>550</td>
<td>550</td>
</tr>
<tr>
<td>samples</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training time / session</td>
<td>6 hrs</td>
<td>10 hrs</td>
</tr>
</tbody>
</table>

As shown above, the BP network
offers superior performance compared to the
LVQ network in determining the decision boundary, and hence better classification of the input traffic conditions into a connection admission decision. For the BP network, if the set of inputs is very close to the boundary surface, say \{15, 15, 69\}, then the BP gives an output of \((0.545, 0.455)\). This gives an indication of the "closeness" of the current set of inputs to the decision boundary. On the other hand, the LVQ network performs relatively poorly at the boundary surface. This is because for LVQ network, a neuron's weight vector may have to travel through a region of a class that it does not represent, to get to a region that it does represent. This would imply tearing down boundaries that have already been formed. Because the weights of such a neuron will be repulsed by vectors in the region it must cross, it may not be able to cross, and so it may never properly classify the region it is being attracted to. This may have contributed to the poorer generalization performance by the LVQ network.

References


