ATM Connection Admission Control using Modular Neural Networks

Chen-Khong Tham & Wee-Seng Soh

Dept of Electrical Engineering,
National University of Singapore,
Singapore 119260.
Tel. (65) 772 7959, Fax: (65) 779 1103
E-mail: `eletck@nus.sg` *&* `engp5630@nus.sg`

Abstract

Neural networks, such as multi-layer perceptron (MLP) networks which converge slowly, have been applied for traffic and congestion control in ATM networks. In this paper, we present a Connection Admission Control (CAC) scheme using modular and hierarchical neural networks for predicting the resulting cell loss rate (CLR) when calls are accepted. The fast learning and accurate predictions obtained using this architecture is shown to produce near zero CLR while maintaining a high throughput.
Keywords: ATM Congestion and Traffic Control, Call Admission Control

1 Introduction

Asynchronous Transfer Mode (ATM) [6] is a high-speed packet switching technology for the broadband integrated services digital network (B-ISDN), in which various kinds of communication services such as voice, video and data are transferred over high-speed links. This technology is gaining acceptance as the backbone high-speed network of the future, and as a high bandwidth link to homes and the desktop, e.g. for interactive multimedia services. It supports different service classes, such as Constant Bit

Rate (CBR), Variable Bit Rate (VBR), Unspecified Bit Rate (UBR) and Available Bit Rate (ABR). ATM traffic management or congestion control methods [1] are needed to ensure that quality of service (QoS) parameters such as cell delay and cell loss probability are within the agreed limits.

2 Connection Admission Control (CAC)

When a user wishes to establish a connection with another user, his terminal sends a connection set-up request to the Connection Admission Control (CAC) controller, during which it declares information such as the required QoS and its own traffic parameters, which describe the cell generation characteristics of the source, e.g. Peak Cell Rate (PCR), Average Cell Rate (ACR), burstiness and peak duration. The controller forms an estimate of the resulting network situation if the connection is accepted, and proceeds to accept the connection only if this estimate indicates that the QoS requirements of the new and existing users, especially those using the CBR and VBR service classes, will not be violated.

There is usually a difference between the declared and actual traffic parameters, making QoS estimation through analytical methods difficult. The number of connections can be large, thus compounding the estimation error in each connection. Neural network approaches rely on the actual observed values of various quantities of interest to form a mapping between different data sets.

3 Neural Networks

In the area of high-speed networking, neural networks have been applied in Connection Admission Control (CAC) [3], flow control and routing [5], ATM switch control [2] and bandwidth prediction for variable bit rate video.

The basic approach for prediction using neural networks is to construct a mapping between current and future values. To implement CAC, the neural network is trained by presenting input patterns which reflect the current buffer status - this can consist of current and previous cell arrival rates (CAR). The required output is an estimate of QoS in the future, commonly expressed in terms of average cell delay or cell loss rate (CLR). In our experiments, the average of CLR values monitored at several future time-steps, is used as the target value.

4 Modular and Hierarchical Neural Networks

Although single or monolithic function approximators, such as MLP networks, are able to model most functions, an alternative, especially when complex functions need to be modelled, is to use multiple neural networks. This method involves the construction of several different neural networks and combining them to get improved prediction accuracy. In addition, faster learning of the target function can also be achieved.

A severe drawback of MLP networks is that they require many training cycles before the required mapping function is learnt. In a CAC task performed using MLP networks, Hiramatsu [3] presented results which involve timescales of between 2,000 to 10,000 seconds. His system achieves the target CLR of 10^{-4} in "a few thousand seconds".

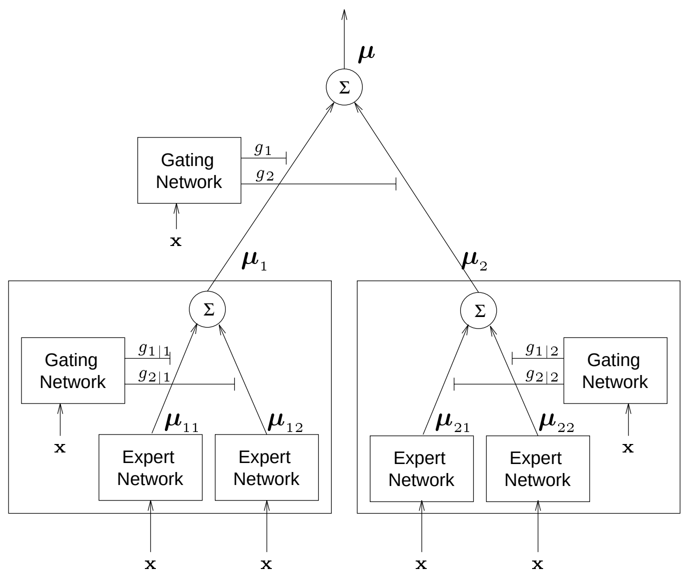

Figure 1: A two-level Hierarchical Mixtures of Experts architecture.

4.1 Hierarchical Mixtures of Experts (HME)

The Hierarchical Mixtures of Experts (HME) [4] is an architecture (an example is shown in Figure 1) which is based on the principle of *divide-and-conquer* in which a large and difficult function approximation problem is broken into several smaller and easier-to-solve sub-problems. This involves dividing the input space into a nested set of regions and fitting surfaces to the data that fall in these regions. A separate function-approximator or 'expert' network is assigned through competition to each of these regions. Each expert network may itself be composed of competing sub-experts. Details such as the training procedure will be described in the full paper.

In this paper, the HME is used for predicting the average cell loss rate (CLR) that would arise when new call set-up requests are accepted. These predictions form the basis of the proposed ATM Connection Admission Controller.

5 HME-based Connection Admission Control

The HME architecture is integrated into the CAC algorithm shown in Figure 2 - its role is to predict the average cell loss rate (CLR), when calls are accepted, from an input pattern made up of the current and previous values of the cell arrival rate at the Broadband Terminal Equipment (BTE). Learning takes place in real-time, i.e. control decisions are based on the output of the HME, these control actions affect the subsequent input patterns and target values, which in turn affect future HME predictions and control decisions etc. However, note that the HME is trained only after an ACCEPT decision to ensure that a consistent mapping is formed. Once the prediction of the average CLR $J(t)$ is available, the CAC policy is straightforward: accept new calls in the next time interval Δt if $J(t)$ is less than the target CLR, and reject them otherwise.

6 Experimental Details

6.1 Experiments

The following experiments using four different CAC schemes were carried out:

Admit all : This the "no CAC" case. All arriving calls are accepted by the BTE.

Simple : A simple control policy, in which a call is accepted if the BTE output buffer is less than 90% full when the call arrives, is implemented.

MLP : The MLP-based algorithm proposed by Hiramatsu [3] is implemented.

HME : The HME-based CAC scheme described in Section 5 which uses predictions of average CLR from a 3-level HME network with eight expert networks is used.

All four CAC schemes observe the same call arrival pattern, i.e. calls have exponentially distributed inter-arrival times with a mean of 10ms.

CAC Scheme	No. of cells received at Destination	No. of cells discarded/lost
Admit all	3,186,150	77,942
Simple	3,107,786	10,302
MLP ($\eta = 0.01$)	2,838,130	69,109
MLP ($\eta = 0.05$)	1,859,974	16,105
HME	3,023,512	669

Table 1: Throughput and cell loss results at the end of 10s of simulation time.

7 Results

The HME-based CAC approach is able to achieve near zero CLR throughout the 10s simulation period, while maintaining high throughput, i.e. just a little lower than the 'admit all' and 'simple' CAC schemes - see Table 1 and Figure 3(d). The number of cells discarded is significantly lower than the other schemes.

The ACCEPT/REJECT graph shows a high degree of switching which indicates that the HME-based CAC scheme can react quickly to different traffic situations. Calls are not rejected unnecessarily, but enough calls are rejected to keep CLR low.

The good performance of the HME-based scheme can be largely attributed to its ability to predict the average CLR accurately and quickly, which is a direct consequence of the modular and hierarchical nature of the HME.

8 Conclusion

An ATM Connection Admission Control scheme using the modular and hierarchical HME architecture has been proposed in this paper which achieves near zero cell loss rate while maintaining high throughput in an ATM network. It is shown to be superior to the method described in [3], which uses an MLP network to perform classification, in terms of both cell loss rate and throughput. We believe that the main reason for the good performance of the HME-based CAC scheme is that the HME reacts quickly and accurately to new traffic patterns.

References

[1] ITU-T Study Group 13. Traffic Control and Congestion Control in B-ISDN. Recommendation I.371, ITU-T, Perth, Nov 1995.

[2] T.X. Brown. Neural networks for switching. In B. Yuhas and N. Ansari, editors, *Neural Networks in Telecommunications*, pages 11–36. Kluwer, 1994.

[3] A. Hiramatsu. ATM communications network control by neural networks. *IEEE Transactions on Neural Networks*, 1(1):122–130, 1990.

[4] M.I. Jordan and R.A. Jacobs. Hierarchical Mixtures of Experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[5] J.E. Neves, M.J. Leitão, and L.B. Almeida. Neural networks in B-ISDN flow control: ATM traffic prediction or network modeling ? *IEEE Communications Magazine*, pages 50–56, Oct 1995.

[6] Händel R., M. Huber, and S. Schröder. *ATM Networks - Concepts, Protocols, Applications*. Addison-Wesley, 2nd edition, 1994.

1. Start of control cycle. Set $t = 0$.
2. Get cell arrival rate $p(t)$ from multiplexer.
   (obtained by counting number of cells arriving the last time interval Δt)
3. Get cell loss rate $l(t)$ from multiplexer.
   (obtained by counting number of cells lost when output buffer of BTE multiplexer is full, divided by total number of cells arriving, in the last time interval Δt)
4. If $t \geq k-1$, form new cell arrival pattern $P(t)$ (k is the size of the cell arrival pattern):
$$P(t) = [p(t), p(t-1), ..., p(t-k+1)]$$
5. Pass $P(t)$ to HME and ask for $J(t)$, which is the prediction of average CLR in the next averaging period $T \times \Delta t$ (*note*: $T \geq 1$).
6. Make control decision: (L_t is the target CLR)
   if $J(t) < L_t$, set $r(t)$ to 1 and ACCEPT all set-up requests in the next Δt
   if $J(t) \geq L_t$, set $r(t)$ to 0 and REJECT all set-up requests in the next Δt
7. Save $P(t)$ - the $T+1$ most recent $P(t)$ values need to be saved.
   This can be implemented efficiently by saving $p(t)$ at each time step in a buffer of size $k+T$, e.g. $P(t-T)$ can be formed from the oldest k values in this buffer.
   $l(t)$ is saved in another buffer of size T - every value in this buffer is used to form the target value in Step 8.
8. If $t \geq k+T-1$ AND $r(t-T) = 1$, i.e. ACCEPT decision was made T intervals ago, train HME with the following input-output pair:
   Input pattern: $P(t-T)$
   Target value: $y(t) = \text{compand}([\text{avrg}(l)](t-T))$, where
$$[\text{avrg}(l)](t-T) = \frac{l(t+1-T)+...+l(t)}{T}$$
   and
$$\text{compand}(x) = \frac{\log(2+\mu x)}{\log(2+\mu)} \qquad [\mu = 10^6]$$
   else do not train HME
9. Repeat the control cycle: set $t \leftarrow t+1$ and go to Step 2.

Figure 2: The algorithm for CAC using the HME. *Note*: t and T are treated as the number of intervals Δt.

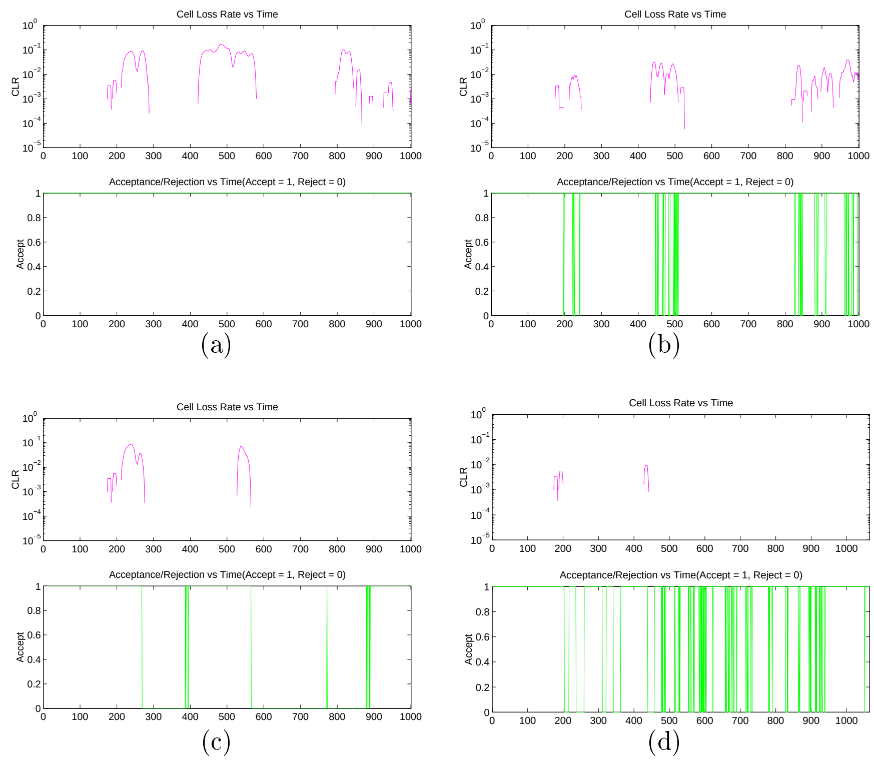

Figure 3: The average CLR achieved and the ACCEPT/REJECT decisions throughout the simulation period of 10s, using the four different CAC schemes: (a) Admit all. (b) Simple. (c) MLP ($\eta = 0.05$). (d) HME. *Note*: The number on the horizontal axis denotes the number of 10ms intervals which has elapsed.