GLOBECOM 92

Performance of a Neural Net used as Admission Controller in ATM Systems

P. Tran-Gia and O. Gropp

Institute of Computer Science, University of Würzburg, Am Hubland, D-8700 Würzburg, F.R. Germany
Telephone: +49-931-8885509, Telefax: +49-931-8884601, e-mail: trangia@informatik.uni-wuerzburg.de

Abstract. *In this paper the capability of neural nets to control connection admission in Asynchronous Transfer Mode (ATM) networks is investigated. The general problem of connection admission control (CAC) and its formulation as a functional mapping are discussed, leading to applications of learning algorithms to CAC problems. In particular, the use of the class of feed-forward neural nets with back-propagation learning rule is considered, where various architecture alternatives are presented. As example a simple neural net structure and its use to control connection acceptance is discussed in detail. The neural net performance is compared with other connection admission control mechanisms like the peak bit rate, the equivalent bandwidth and the weighted variance methods. Numerical results for both cases, stationary load and non-stationary pulse-form overload patterns, illustrate the capability of neural nets to act as connection admission controller in ATM environments.*

1 Introduction

In broadband networks for integrated services developed to operate in high-speed environments, such as ATM (Asynchronous Transfer Mode) systems [2, 3], the connection admission control function (CAC) plays an increasingly important role, from both resource management and network management viewpoints. The aim of these efforts is to design the CAC function to maximize the system throughput while maintaining the desired quality of service. In the context of ATM networks the rather classical problem of admission control gains a higher level of difficulty, due to the diversity and the probabilistic complexity of traffic streams offered by a source and the superimposed traffic offered to the network. In addition the large variety of requirements for the quality of service to be fulfilled increases the complexity of the admission control problem.

The use of neural nets in the admission control context of ATM networks is first suggested by Hiramatsu [6, 7], where the class of back-propagation neural nets is considered. In [12] performance aspects are taken into account and the neural net application is discussed in a more general context. The neural network is used to control admission with different performance objectives being used for different call classes, where additional cell level controls are recommended. In [11] a hybrid method using both stochastic approximation and back-propagation is tested for congestion control in telecommunication networks.

The major aim of this paper is to present and to compare possible neural net structures applicable to connection admission control and to show the performance of a basic neural net under various stationary and nonstationary load conditions.

In Section 2 basic principles of the use of feed-forward neural networks with back-propagation learning in connection admission control will be discussed and different alternative neural net structures will be compared. A simple neural net will be selected as example in Section 3 and Section 4 to show the acceptance control performance and to discuss numerical aspects of neural nets under considerations.

2 Neural nets for connection admission control

Depending on the informations available to the connection admission control function and where this function is located in communication networks, different neural net structures can be developed. In this section we will briefly present these alternatives and discuss in particular the basic function and learning procedure of a back-propagation neural net used as admission controller.

2.1 Admission control in broadband networks

The connection admission control forms an important part belonging to resource allocation problems in ATM networks (cf. [3]). According to CCITT [2] the connection admission control function (CAC) is defined as:

"CAC is the set of actions taken by the network at the call set-up phase (or during the call re-negotiation phase) in order to establish whether a (virtual channel or virtual path) connection can be accepted or rejected".

We will first illustrate some problems arising in connection

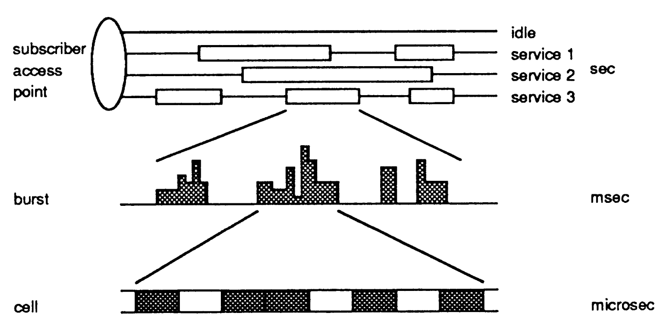

Figure 1: *Layered ATM traffic*

admission control. As shown in Fig. 1, traffic offered from a source, corresponding to a service being used, can be observed on connection (or call) layer, on burst layer or on cell layer (see also [5]). In the high-speed environment observed, the time resolution in these traffic layers differs in order of magnitude, i.e. in seconds, milliseconds or microseconds for call, burst and cell layers, respectively.

From connection admission control point of view, the connection acceptance decision which must be made for connection time scale should be able to guarantee quality of

CONTRIBUTION TO GLOBECOM 92, ORLANDO, USA service on cell layer, e.g. keeping the resulting cell blocking probability below a predefined threshold. In current discussions in ATM system developments the range of this threshold is very low (e.g., 10^{-8}).

On the other hand, from bandwidth utilization viewpoint, traffic offered by an active source can also be characterized by a time-dependent bit rate process. Since the source characterization should be kept simple for the connection acceptance process, only a few parameters are assumed to be known in advance when a service requests a connection, e.g. the mean and the peak bit rates a connection is allowed to submit to the network during the connection duration. From network viewpoint, as depicted in Fig. 2, the mul-

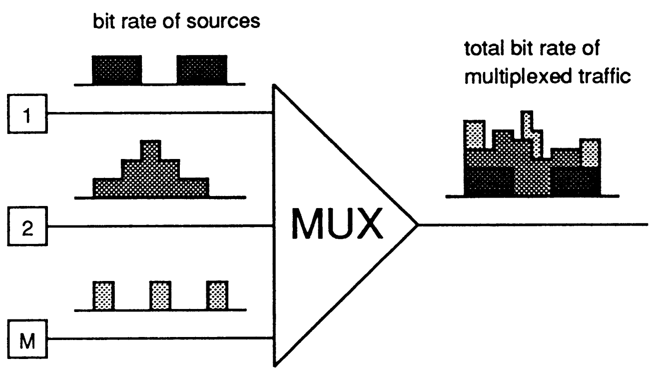

Figure 2: *Source traffic and multiplexed traffic*

tiplexed traffic has to be carried. This traffic is the result of the superposition of a number of bit rate processes generated by a number of sources of different types or services. Examples for connection or service types are constant or variable bit rate video connections, data transfer connections, interactive connections, etc.. The bit rate process, as shown in Fig. 2, is an equivalent representation of the cell arrival process. Due to the variation of this process and the finite capacity of buffer in an ATM multiplexer, delay and blocking of cells can occur and are issues of concern. The cell blocking probability and the cell delay or delay jitter are often used as indicator for the network quality of service.

We observe a connection establishment request of type i. In case of acceptance the network can be thought of to have a contract with the user: the user agrees to keep the negotiated traffic characteristics during the holding time of the connection, the network promised quality of service. The CAC has to accept new connection in such a way that all connections before and after the admission decision of the new connection are treated according to the negotiated quality of service.

Given that the CAC function is able to estimate the quality of service before and after having the requested connection, it can make an acceptance decision, i.e. the request will be rejected if the forecasted QOS could not be maintained.

In the following, we consider a number M of different connection types to be served by the network. We distinguish between two cases

i) CAC based on network state :

In this case, we assume that the entire information about the number of all connections being multiplexed is available. The system state seen by the network will be denoted by

$$X = \{n_1, n_2, .., n_M\} \tag{1}$$

with n_i is the number of active connections of type i being in the system. From mathematical point of view, the main CAC function can be represented by a mapping of the system state X to a decision vector Z defined by

$$Z = \{z_1, z_2, .., z_M\} \tag{2}$$

where $z_i = 1$ stands for the acceptance decision of a connection establishment request of type i and $z_i = 0$ for the rejection case. The CAC is thus reduced to the implementation of a mapping

$$f: \qquad X \rightarrow Z = f(X) \tag{3}$$

according to the predefined quality of service of the network. The mapping f can further be simplified by using the state

$$X^\star = \{n_1, n_2, \ldots, n_i + 1, \ldots, n_M\} \tag{4}$$

i.e. the system state just after accepting the connection request of type i. The decision vector is reduced to

$$Z^\star = \{z_i\} = \begin{cases} 0 & \text{connection i should be accepted} \\ 1 & \text{connection i should be rejected} \end{cases} \tag{5}$$

and the CAC mapping to

$$f^\star \ : \ X^\star \rightarrow Z^\star = f^\star(X^\star) \tag{6}$$

ii) CAC based on bit rate process :

In this case, only the superimposed bit rate process is available or measurable to the connection admission control function. This is e.g. the case where an intermediate ATM switching node does not have the whole system information but sees only bit rate processes to be transferred. Denoting the observed total bit rate function during the time interval $(t, t + \Delta t)$ by Y the CAC function can again be represented by the mapping

$$g \ : \ Y \rightarrow Z = g(Y) \tag{7}$$

2.2 Neural network as admission controller

As discussed in the previous subsection, the connection admission control function can be interpreted as a mapping of e.g. the state vector X into the acceptance decision vector Z. This functional mapping divides the M-dimensional state space into two regions: the accept region and the reject region. In other words, the CAC problem can be formulated like a pattern recognition problem: upon recognition of the load pattern X, a yes/no decision has to be made to accept/reject the connection request. The use of a neural net for connection control purposes in ATM systems is thus quite obvious. In this paper we will restrict the description to the class of feed-forward neural nets with back-propagation learning algorithm [14]. The basic struc-

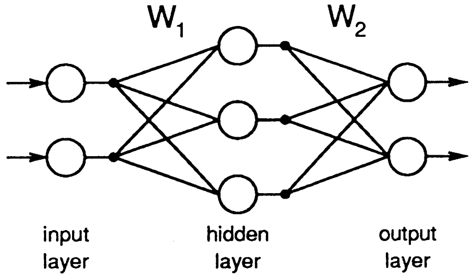

Figure 3: *Feedforward neural nets with backpropagation learning*

ture of a feedforward neural net is depicted in Fig. 3, where a three-layer net is chosen. The neural net consists of a number of neurons connected by weight vectors W_i. As shown in Fig. 3, it contains an input layer, one (or more) hidden layer and an output layer, connected in a fully meshed, feed-forward manner. Each neuron is a simplified model of a natural neuron. It is the basic processing element of the neural net, where the output v_j is computed as a weighted sum of inputs according to the sigmoidal function

$$v_j = \frac{1}{1+e^{-x_j}} \quad \text{with} \quad x_j = \sum_i w_{ij}\, v_i \quad , \tag{8}$$

w_{ij} is the weight of the connection from neuron i to neuron j. For numerical reasons a bias is often added to each x_j.
There are two operation modes: learning mode and recall mode. In the learning mode, pairs of input/output vectors are presented to the net. It computes its own output vector according to the equations mentioned above and compares the computed output vector with the presented input vector. The comparison results in an error vector, which will be used to change the weight matrices according to a learning rule. The learning phase will end if all input/output-pairs to be learned have been presented and the total error is lower then a predefined threshold. After the completion of the learning phase, the information about the input/output pairs, which represents a mapping, can be seen as stored in the weight matrices.
The use of a neural network to control connection acceptance is illustrated in Fig. 4. This basic structure has been first proposed in [6] and further developed in [7]. Traffic streams offered by different type of sources are multiplexed at the entry node of the high-speed communication network. In the proposed neural net structure in [6] the bit rate function is used as input to the neural net. The current quality of service (QOS) is measured e.g. by the cell-blocking probability B_{CELL}. This implies that the cell blocking probability at the multiplexer is taken as quality of service indicator. During the learning phase of the neural net, the input/output patterns are as follows. Inputs are formed by the bit rate pattern including the bit rate process generating by the actual connection request. The resulting QOS will be observed and compared to the target QOS. If the target QOS is still hold, the output of the current input/output pair is Z=1, i.e. the connection can be accepted, the bit rate pattern is a "good"-pattern. If the resulting quality of service is lower than the target one the output is then Z=0, i.e. the connection should not be accepted in the current load situation, the bit rate pattern is a "bad"-pattern. After the learning phase, i.e. all input/output pairs have been presented , the neural net can be used in the recall mode to perform CAC function. One of the disadvantages of this mechanism is the difficulty to generate a significant number of good- and bad-patterns for the neural net to learn. Therefore we decided to design a modified learning process for the neural net. In the current study we devote attentions

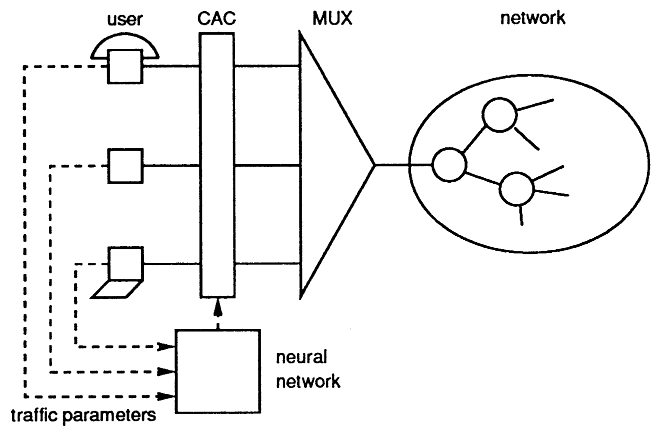

Figure 4: *Neural net for admission control*

to the neural net structure depicted in Fig. 5. The neural net is designed to perform the mapping given in eq. (6), i.e. the CAC based on network state. We consider a number M of different classes of connections, each with different known bit rate characteristic. The pairs of input/output patterns for the neural net to be learned is computed as indicated in Fig. 5. Starting with a state vector $X = \{n_1, n_2, .., n_M\}$ as the input part of a pattern the multiplexed bit rate function is determined. Having this bit rate function as traffic stream, the cell blocking probability e.g. can be estimated giving the actual quality of service. Upon a comparison of this measure with the target QOS, the acceptance decision Z can be made. This can be interpreted as the decision to be made to accept/reject a connection request of type i if actual system state is $\{n_1, n_2, .., n_i - 1, .., n_M\}$. The working mode of the neural net during the recall phase is as shown in Fig. 5, where the net will answer with an accept/reject decision Z^* for a connection request of type i when the input vector $X = \{n_1, n_2, .., n_i + 1, .., n_M\}$ is presented.
Thus, after the learning phase, the neural net performs the CAC by separating the M-dimensional input state space in two regions corresponding to a (M-1)-dimensional decision surface. The decision surface, which separates the "accept" region from the "reject" region in the state space, can be thought of as stored in the weight vectors of the neural net.

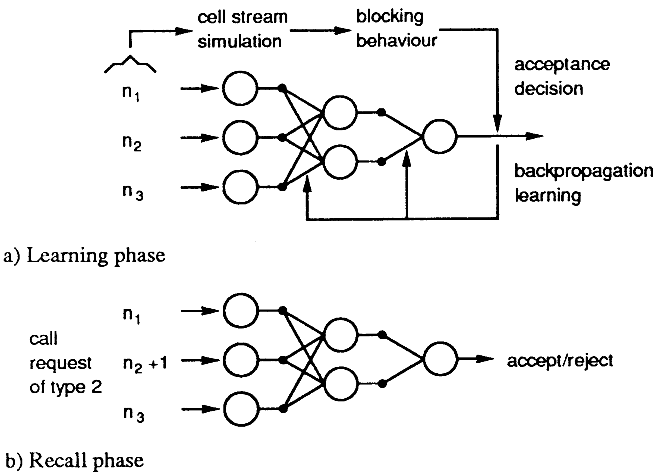

Figure 5: *Working modes of a neural net based admission controller*

2.3 Alternative neural net structures for admission control

In the previous subsection we introduced the two cases: i) CAC based on network state, where information about the number of all connection being multiplexed is available, i.e. the detailed state vector X of the system is known by the connection admission control function and ii) CAC based on bit rate process, i.e. only the superimposed bit rate process is available or measurable to the connection admission control function. According to this two cases different neural net structures can be developed, as shown in Fig. 6.

i) CAC based on network state
A simple backpropagation neural net with only one output neuron is depicted in Fig. 6 a), which implements the mapping of eqn. (6). The same functionality can be obtained using the mapping of eqn. (3) according to the neural net structure shown in Fig. 6 b).

ii) CAC based on bit rate process
Fig. 6 c) depicts a feedforward neural net for the mapping of a bit rate pattern to an accept/reject decision. This reflects a communication network architecture with less signalling efforts involved, where only the superimposed bit rate process is available or measurable to the connection admission control function.

3 Example of a neural net for admission control

In this section, to illustrate the performance of neural nets in connection admission control, we take the example of a neural net according to the structure shown in Fig. 5.

3.1 Traffic assumptions and configuration parameters

The parameters of the ATM multiplexer and the connection types are as follows. The output of the multiplexer has a capacity of 600 Mbps, the buffer space is 0.5 Mb large. To model approximately the VBR (variable bit rate) sources we

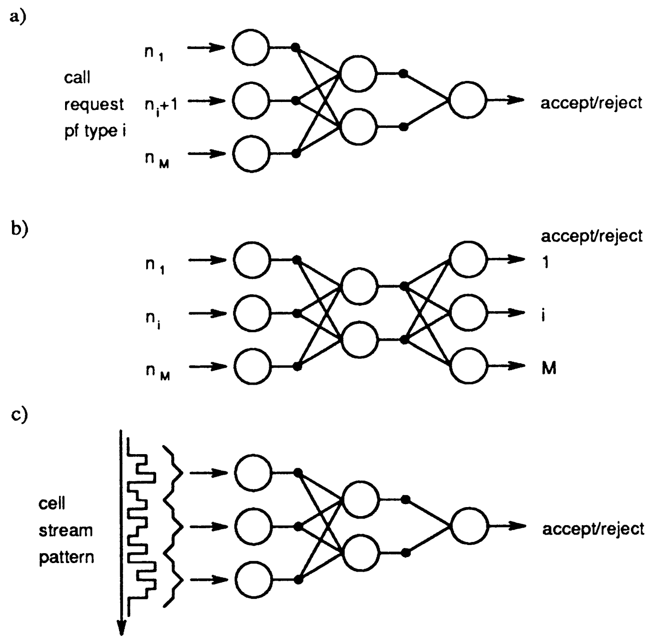

Figure 6: *Alternative neural net structures for admission control*

consider sources with first-order Markovian bit rate processes, where the two basic types are used: a) on/off-sources and b) binomial sources with two parameters: mean bit rate m and peak bit rate h. The time axis is discretized by $\Delta t = 100\,msec$. The bit rate with random variable X will be expressed in number of basic units $\Delta B = 1\,Mbps$. In each Δt we assume each source to have an independent bit rate following the distribution:

a) on/off-sources:
$p_{ON} = P\{X = \frac{h}{\Delta B}\} = p; \quad p_{OFF} = P\{X = 0\} = 1 - p;$
b) binomial sources:
$p_i = P\{X = i\} = \binom{\frac{h}{\Delta B}}{i} p^i (1-p)^{\frac{h}{\Delta B}-i}, \quad i = 0, 1, ..., h.$

We consider three connection types:
Type 1: on-off, m = 10 Mbps, h = 40 Mbps, $c_X = 1.73$.
Type 2: binomial, m = 5 Mbps, h = 40 Mbps, $c_X = 0.42$.
Type 3: binomial, m = 5 Mbps, h = 80 Mbps, $c_X = 0.19$.
c_X denotes the coefficient of variation of the bit rate X.

On connection traffic level, the arrival process of connection requests is assumed to be Poisson with a mean interarrival time chosen according to the simulated load scenario. To obtain patterns for the neural net learning process, the cell stream traffic is simulated. During the simulation time the amount of lost cells is estimated by a fluid flow model. The connection duration is assumed to be negative exponentially distributed with mean 20 sec. This mean value is intentionally chosen to be short to enable simulation runs without loosing the qualitative significancy of the results obtained. We simulate the traffic on burst level. The cell loss depends accordingly only on the actual sum of the bit rates of the sources of active connections, the capacity of the output line and the buffer space of the multiplexer. Cells are only stored in the buffer if the total bit rate exceeds 600 Mbps. Under this condition the buffer is filled and cells are lost if the buffer level exceeds the maximum.

3.2 Alternative CAC methods for performance comparison

Since an agreement on CAC mechanisms for ATM system is not yet available, we will select a few methods proposed in the literature (see [1, 4, 8, 10, 13]) to compare with the neural net CAC approach. The parameters taken into account for CAC purpose here are the numbers of active sources with given connection types, and for each connection the mean bit rate m and the peak bit rate h. The CAC aim is to keep the QOS, i.e. the cell loss rate below a given value, say 10^{-5}.

3.2.1 Peak reservation method (PR)

The most simple and robust method to limit the cell loss probability is to reserve the peak bit rate for each accepted connection. New connections are only admitted if the sum of the peak bit rates of the active connections and the new connection is smaller than the capacity of the output line. Thus no loss will ever appear. This method reduces ATM rather to STM. Obviously, for more bursty bit rate traffic the ouput channel is used in an inefficient way and the multiplexer utilization may be intolerably low. This peak bit rate reservation method is considered here only as a lower bound for admission control methods aiming to high multiplexer utilization.

3.2.2 Equivalent bandwidth method (EB)

The expression "equivalent bandwidth" is introduced in [3], in conjunction with mathematical backgrounds presented in [9]. Each source of type i has its equivalent bandwidth k_i, which depends on its mean bit rate m_i, its peak bit rate h_i and the capacity of the multiplexer output line:

$$k_i = C_1 m_i + C_2 \frac{m_i(h_i - m_i)}{c} \tag{9}$$

The constants C_1 and C_2 depend on the buffer space of the multiplexer and the maximum cell loss rate and have to be determined empirically. If a connection request of type i arrives the following condition is inquired: $K + k_i \leq c$, where K denotes the sum of the equivalent bandwidths of the actual active connections. If it holds the new connection is accepted, otherwise rejected.

3.2.3 Weighted Variance method (WV)

The original method proposed in [15] has to be modified in the context of this study due to simulation reasons. The original method works only sufficiently well if the peak bit rates of the subscribers is less than one percent of the output line capacity. The modified algorithm works as follows, where m_j represents the mean bit rate of connection j, h_j its peak bit rate and c the capacity of the output line. Connection k is the new connection to be admitted, connections 1 to $k-1$ are already admitted. If $\sum_{j=1}^{k} h_j \leq p_0 c$ connection k is accepted. If this inequation does not hold, the following one is employed:

$$\alpha \sqrt{\sum_{j=1}^{k} m_j(h_j - m_j)} + \sum_{j=1}^{k} m_j + \max_{1 \leq j \leq k} h_j \leq c \tag{10}$$

If this inequation holds, connection k is accepted, otherwise it is rejected. The term $EX_j(d_j - EX_j)$ is an estimate of the variance of the bit rate of connection j. Thus the constant α determines the influence of the variances of the source s on the CAC process. The term α has to be found in advance by simulation.

3.2.4 Neural network CAC (NN)

We use a three layered feed-forward neural net to evaluate the CAC function. The neural net structure is the one depicted in Fig. 6 a).

The input consists of the vector $X^\star$ of the numbers of active sources of each class where the component of the class of the arriving request is incremented by one. The result of the feed-forward computations at the output unit is a real number between 0 and 1. If the output value is less than a threshold (say 0.5) the new connection is accepted, otherwise rejected. The decision of the neural net depends on its internal set of weight matrices which have to be determined in advance during the learning phase as discussed in the previous section.

3.3 Neural net convergence and numerical issues

As mentioned before the neural net needs a learning process to fix its weight vectors. This process uses a set of patterns to be learned. Each pattern consists of an input vector $X^\star$ and the corresponding output value $Z^\star$, which have to be chosen so that the network is able to learn the mapping function $f^\star$ for CAC (see section 2.1). We obtain this pattern using a simulation of the multiplexer state process, i. e. we fix the numbers of active connections of each traffic class at certain values and determine the corresponding loss rate at the multiplexer buffer. If this loss rate is less than a predefined value (in this study 10^{-5}) this set of connection can be accepted, otherwise it should be rejected. We perform this simulation for the vectors

$$X_i^\star = \{i_1 k, i_2 k, .., i_M k\} \quad \text{with} \quad 0 \leq i_j \leq \frac{600Mbps}{m_j} \tag{11}$$

and step size k. Thus we get an equally spaced M dimensional grid whose nodes are named with A(ccept) or R(eject). This grid can be separated by an $M-1$ dimensional decision surface in an "accept" and a "reject" region. As an example Fig. 7 shows this grid for $M = 2$, where n_1 and n_2 denote the numbers of active connections of class 1 and 2. In this case the decision surface is just a line. To complete the pattern each $X_i^\star$ gets a $Z_i^\star = 0.2$ if the node name is "A" and a $Z_i^\star = 0.8$ if the node name is "R". The values 0.2 and 0.8 instead of 0 and 1 are used to obtain a more appropriate learning algorithm.

First we transform the learning process into a function mi-

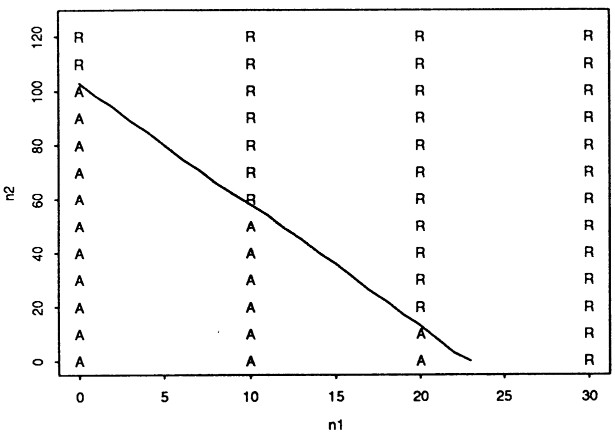

Figure 7: *Learning patterns of neural net*

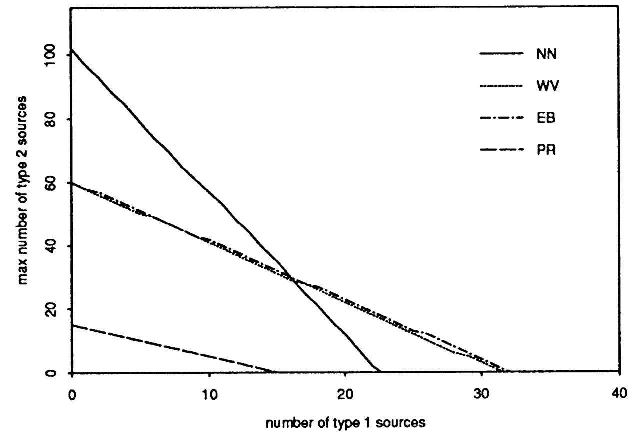

Figure 8: *Decision surface for CAC*

nimization problem. Given the set of learning patterns we define an error function

$$E(W) = \frac{1}{2N} \sum_{i=1}^{N} (Z_i^\star - F(X_i^\star, W))^2 \tag{12}$$

The term W denotes the vector of the weights and $F(X_i^\star, W)$ the output of the neural net upon $X_i^\star$ presented to the input layer. Using the BFGS algorithm - well known from the theory of optimization - $E(W)$ is minimized in the weight space.

For the recall phase the internal weights of the neural network are now fixed to their final values. The neural network is thought of to has learned the functional mapping $f^\star(X^\star) = Z^\star$ correctly only for the training patterns. It is then able to perform this mapping also for all other input patterns with the help of the learned decision surface. This property is often referred to as "learning by examples".

3.4 Performance results and discussion

The load control performance of the neural net will be discussed in this subsection, taking into account stationary and non-stationary load conditions. The neural net operates in the recall mode. Results are obtained by means of simulations with different mixtures of the three connection types described in Section 3.1. For the "Equivalent Bandwidth" and the "Weighted Variance" methods the parameters C_1 and C_2 or respectively, α , had to be determined to guarantee a cell loss rate smaller than the threshold 10^{-5}. Tab. 1 shows the multiplexer utilization for the CAC methods considered. The column 'Mix' indicates the mixture of the connection types used. Without any admission control the utilization of the multiplexer would be about 91 %, without maintaining the desired QOS. As expected PR is the most restrictive method and has a bad performance, whereas the other methods perform almost on the same level. Only in the case with all the three connection types involved a slight advantage of the NN control can be observed. The two methods EB and WV

Mix	PR	EB	WV	NN
1/2	20.2 %	47.9 %	47.4 %	47.5 %
1/3	17.9 %	45.4 %	43.6 %	48.0 %
2/3	10.3 %	69.1 %	67.0 %	66.7 %
1/2/3	15.8 %	49.5 %	50.6 %	55.4 %

Table 1: *Multiplexer utilization*

have almost the same connection blocking probabilities. The conclusion of the comparison of their performance with the NN solution is that the NN method rejection decision depends strongly on the mean bit rate of the connection type while the decision of EB and WV depend on mean bit rate and variance.

Fig. 8 shows the decision surfaces of the considered connection admission control methods, which separates the accept and reject regions. The accept region lies on the left hand side of the decision surface. The two methods EB and WV have almost the same decision line, which again indicates the

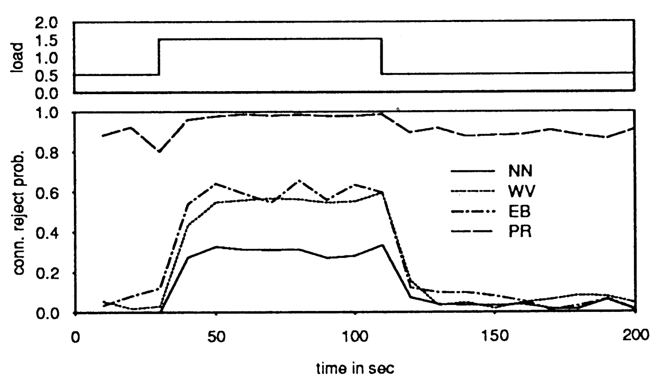

Figure 9: *Comparison of overload performances*

similarity of their performances. The NN decision surface is extremely different. It can be observed that the NN algorithm accepts much more sources with small mean bit rate (type 2) and less sources with high bit rate (type 1) than the EB and WV methods. From communication network point of view, this results in the same multiplexer utilization (Tab. 1, row '1/2'), whereas from user viewpoint the differences for the user groups using connection type 1 or 2 are significant.

To show the overload control performance of the CAC methods under consideration, it is necessary to study the CAC response on a non-stationary overload pattern. In the diagrams to follow we use as overload pattern a rectangular overload pulse as illustrated in Fig. 9 and observe the time-dependent CAC reaction in term of cell and connection blocking probabilities. Clearly, a better CAC mechanism should survive the overload phase with smaller connection blocking probability while keeping the cell loss rate on the same level as under normal load conditions.

Fig. 9 shows a comparison of the non-stationary connection blocking probabilities of connection type 3 of the four CAC control methods. In this case it can be seen that the overload performance of the neural net solution is the most efficient.

4 Conclusion and outlook

In this paper different aspects concerning possible use of neural nets to perform connection admission control (CAC) in broadband integrated services networks have been discussed. The formulation of CAC problem as functional mapping and in consequence, the use of learning algorithms to represent the required mapping were shown. Various architecture alternatives for the CAC neural nets using the class of feed-forward structures in conjunction with back-propagation learning were depicted. In order to discuss performance aspects a basic net example has been investigated. The neural net performance has been compared with other connection admission control mechanisms like the peak bit rate, the equivalent bandwidth and the weighted variance methods. Numerical results for stationary and non-stationary pulse-form overload patterns have been obtained to illustrate the capability of neural nets used as connection admission controller in ATM environments.

In most of the load scenarios under consideration the CAC performance of the investigated neural net structure is comparable with and in some cases better than the CAC methods mentioned above, even by using a very small and simple neural net. To improve the performance of CAC by neural nets, other neural net structures or other input representations can be developed. One promising candidate is a combined solution of an adaptive neural net with learning patterns, which contain more information about the past of the observable load situation.

References

[1] Castelli P., Cavallero E., Tonietti A., "Policing and call admission problems in ATM networks", Proc. 13th Int. Teletraffic Congress, Copenhagen 1991, pp. 847-858.

[2] CCITT, Temporary Document 43/XVIII: "On networking and resource management", Study group XVIII, Matsuyama 1990.

[3] COST 224 Committee (ed. Roberts J.), COST 224 Final report: "Performance evaluation and design of multiservice networks", Paris, October 1991.

[4] Galassi G., Rigolio G., Verri L., "Resource management and dimensioning in ATM networks", IEEE Network Magazine, (1990)5, pp. 8-17.

[5] Gihr O., Tran-Gia P., "A layered description of ATM cell traffic streams and correlation analysis", Proc. IEEE INFOCOM '91, Florida, April 1991.

[6] Hiramatsu A., "ATM communications network control by neural networks", Proc. IEEE Int. Conf. on Neural Networks 1989, Vol. I, pp. 259-266.

[7] Hiramatsu A., "ATM communications network control by neural networks", IEEE Transactions on Neural Networks, (1990)1, pp. 122-130.

[8] Jacobson S.B., Moth K., Dittmann L., "Load control in ATM networks", Proc. XIII Int. Switching Symposium, Stockholm 1990, pp. 131-138.

[9] Lindberger K., "Analytical methods for the traffical problems with statistical multiplexing in ATM-networks", Proc. 13th ITC: "Teletraffic and Data-traffic in a period of change", Copenhagen, June 1991, pp. 807-813.

[10] Lombardo A., Palazzo S., Panno D., "Admission control over mixed traffic in ATM networks", Int. Journal of Digital and Analog Communication Systems, (1990)3, 155-159.

[11] Milito R., Guyon I., Solla S., "Neural network implementation of admission control", In: Advances in neural information processing systems, Vol 3, Morgan Kaufmann 1991, pp. 493-499.

[12] Morris R.J.T., "Prospects for neural networks in broadband network resource management", Proc. 13th ITC: "Teletraffic and Datatraffic in a period of change", Copenhagen, June 1991, pp. 335-340.

[13] Rasmussen C., Sørensen J., Kvols K.S., Jacobsen S.B., "Source-independent acceptance procedures in ATM networks", Journal on Selected Areas of Communications, JSAC-(1991)4, pp. 351-358.

[14] Rumelhart D.E., Hinton G.E., Williams R.J., "Learning internal representation by error propagation", In: Parallel distributed processing – Explorations in the microstructure of cognition, Vol. I: Foundations, MIT Press 1986, pp. 318-362.

[15] Wallmeier E., "A connection acceptance algorithm for ATM networks based on mean and peak bit rates", Int. Journal of Digital and Analog Communication Systems, (1990)3, 143-153.