Estimation of the blocking probabilities in an ATM network node using Artificial Neural Networks for Connection Admission Control

Vincent Lemaire [a] and Fabrice Clérot

[a] France-Telecom CNET DTL/DLI
Technopole Anticipa, 2 avenue Pierre Marzin, 22307 Lannion cedex, FRANCE
email : vins.lemaire@cnet.francetelecom.fr
tel : +33 2 96 05 31 07
fax : +33 2 96 05 23 58

The aim of this paper is an application of neural networks to Connection Admission Control (CAC) in an ATM network. We propose a method to estimate if, and for how long, an ATM node is in a state of congestion. We show that estimations with neural networks are accurate, need a small observation of the traffic and can therefore react quickly to traffic changes.

1. Introduction

ATM (Asynchronous Transfer Mode) is a high speed network, able to transfer various communication services such as video, voice and data. The basic concept of ATM communications is that all users can send traffic (stream of cells) to any ATM node on demand, at a chosen rate (bandwidth) and with an arbitrary timing. In such networks the traffic characteristics can have high variability. Therefore, it is difficult to build an efficient traffic control (preventive or reactive) system able to guarantee the quality of service (QoS), measured by parameters such as cell time delay and cell loss probability.

Connection Admission Control (CAC) is a preventive traffic control. A user is allowed to establish a connection only if the network considers that the new connection will receive its required QoS and that this admission will not degrade the QoS of all the users already connected below their respective requirements [1]. Before communication, the user has to send a connection setup request to the traffic controller of the ingress node. The connection setup request specifies the QoS values it requires and its traffic parameters. These parameters define the traffic characteristics of the connection. The capacity of each node is a function of the traffic parameters and the QoS of all connections going through the node. Therefore the ATM network must estimate the post connection QoS (with the new connection) from these parameters for all the nodes used by the new connection. It accepts the new user only when the estimated value of the QoS does not violate the QoS requirements for all current users; otherwise the request is rejected.

A number of different ATM transfer capabilities have been defined by the ITU and by the ATM forum [2]. Among these capabilities, we shall consider the ATM Block Transfer (ABT) capabilities, where the user must be able to define and control a block structure

in its data stream. A Maximum Peak Cell Rate (PCR) is declared for the connection duration and a Block Cell rate (BCR), is negotiated for each block between the user and the network. All BCR are lower than the PCR . Only the PCR is used in the CAC procedure.

With this capability, the occupied bandwidth cannot be known in advance because the reservations are made dynamically block by block by all current applications already connected and can therefore have a large variability.

2. The Connection Admission Control policy

We present a MBCAC (Measurement Based Connection Admission Control) using neural networks, adapted to the ABT mode. Instead of predicting the future occupied bandwidth, we only predict the probability that future blocks sent by the sources could become congested in the ATM node for a given time. The maximum required accuracy for such blocking probabilities is about 10^{-4}.

Let us define an excursion θ as the period of time during which the traffic already accepted, $B(t)$, is above a threshold s:

$$B(t) \geq s \tag{1}$$

and its length (or duration) T is denoted by :

$$L\{B(t) \geq s\} = T \tag{2}$$

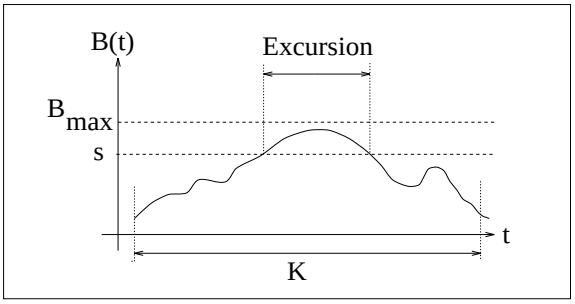

Figure 1. Définition of an excursion

Let us define a ratio η ("time with excursion"/"time without excursion") such as :

$$\eta = Pr(B(t) \geq s|n) \tag{3}$$

The probability that an excursion has a length T is therefore defined such as:

$$Pr(L(\theta) \geq T|s,n) = Pr(L(B(t) \geq s) \geq T|s,n) \tag{4}$$

with :

- θ : the excursion of $B(t)$ above the threshold s ;
- $L(\theta)$: the duration of this excursion ;
- n : the number of sources ;
- s : the threshold.

Denote B_{max} the bandwidth available at the ATM node. The acceptance decision for a call with a maximum declared PCR of d_{max} is made by adding the new connection (supposed to emit at d_{max}) "on top" of the current traffic $B(t)$ and estimating the distribution of the excursion lengths above B_{max} for the aggregate traffic : $B(t) + d_{max} \geq B_{max}$. This is equivalent to estimate the distribution of the excursion lengths of the current traffic $B(t)$ above the threshold $s = B_{max} - d_{max}$.

The estimated ratio η allows us to know if an ATM node will be in a congestion state and the estimated distribution probability of the excursion lengths the duration of this state. The connection is accepted if both the ratio and the excursion length are small enough.

The scheme relies on two characterisations, a characterisation of the new connection and a characterisation of the behaviour of the occupied bandwidth.

As the new connection is accepted on the basis of its PCR (the post connection excursions are evaluated as if the new source would always send at its peak rate), this scheme is conservative. However, it is based on measurements of the pre-connection occupied bandwidth, hence allowing to benefit from the statistical gain that occurs through the multiplexing of the sources already accepted.

With four excursion lengths at fixed probability we can have an estimation of the distribution of the excursion lengths.

$T_1(s,n)$, $T_2(s,n)$, $T_3(s,n)$, $T_4(s,n)$, $\eta(s,n)$, are defined such as :

$$T_1(s,n) = Pr(L(B(t) \geq s) \geq T_1 | s,n) = 10^{-1} \quad (5)$$

$$T_2(s,n) = Pr(L(B(t) \geq s) \geq T_2 | s,n) = 10^{-2} \quad (6)$$

$$T_3(s,n) = Pr(L(B(t) \geq s) \geq T_3 | s,n) = 10^{-3} \quad (7)$$

$$T_4(s,n) = Pr(L(B(t) \geq s) \geq T_4 | s,n) = 10^{-4} \quad (8)$$

and

$$\eta(s,n) = Pr(B(t) \geq s | n) \quad (9)$$

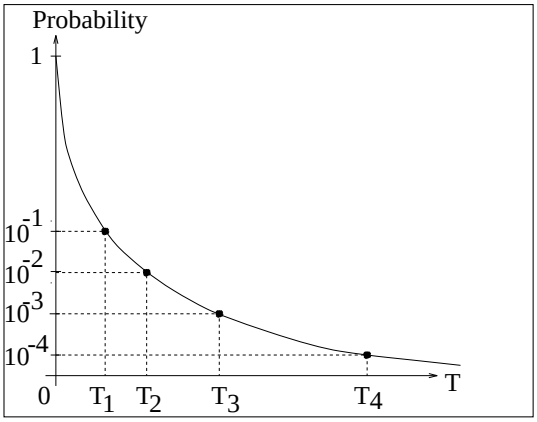

Figure 2. The estimation of the distribution of the excursion lengths.

Our goal is to estimate $T_1(s,n)$, $T_2(s,n)$, $T_3(s,n)$, $T_4(s,n)$ and $\eta(s,n)$ with $s = B_{max} - d_{max}$, using neural networks.

3. The traffic burst model

We want to learn the relation between the observed traffic and the distribution probability of the excursion lengths. A calculation of the distribution probability of the excursion lengths can be made analytically only in the limit of an infinitely large number of sources

[3]. As we do not have a sufficient number of traffic traces, we need a traffic model to constitute a data base. This parametric model is just used to build the traffic data base but the parameters will not be used during the training of the neural networks, since our goal is not an identification of the parameters of the traffic but a direct estimation of its behaviour in terms of excursion lengths. The methodology detailed below can therefore be applied to any traffic data base.

The traffic carried by an individual link is assumed to be produced by independent on/off sources. During the burst period (on), the source transmits traffic, at a constant characteristic cell rate, otherwise the source is silent (off). According to [4] the on/off sources represent the "worst case" output of the traffic enforcement function. Hence, if the connections are represented by on/off sources, the performance analysis will be conservative.

To simulate the occupied bandwidth $B(t)$, we use a superposition of a known number n of Markov Modulated sources. The source behaviour is defined by the peak cell rate, the average active and silence periods (respectively T_{ON}, T_{OFF}), or equivalently by :

- p_{00} the probability to remain in an active period ;
- p_{11} the probability to remain in a silence period ;
- $R_j(t)$ the cell rate for each source.

The occupied bandwidth is defined by the aggregate traffic :

$$B(t) = \sum_{j=1}^{n} R_j(t) \tag{10}$$

If we have a sufficient number of samples the probability can be estimated from the frequency on a given time window. Therefore :

$$Pr(L(\theta) \geq T_i | s, n) \sim \frac{\sum_{t=0}^{K} \left(L \left(\sum_{j=1}^{n} R_j(t) \geq s \right) \geq T_i \right)}{\sum_{t=0}^{K} \left(L \left(\sum_{j=1}^{n} R_j(t) \geq s \right) \right)} \tag{11}$$

and

$$\eta \sim \frac{\sum_{t=0}^{K} \left(\sum_{j=1}^{n} R_j(t) \geq s \right)}{K} \tag{12}$$

with K the estimation period.

A simulation (see figure 3) of the aggregate traffic was built with this traffic burst model. This simulation is based on a limited available bandwidth (B_{max}) at the output link.

For $B_{max} = \infty$, the results of the estimation of the distribution of the excursion length was reported in [5]. Because of the limited bandwidth, the incoming traffic blocks may be stored temporarily in a memory buffer. Each traffic source, in this adaptative simulation, has a dedicated buffer whose size is fixed at two blocks. To decide which source can start the emission of its block, a random choice is made then the chosen sources become prioritary for the duration of its block. This allow each block to be sent within a finit time. All the prioritary sources are then multiplexed and provide the aggregate traffic $B(t)$.

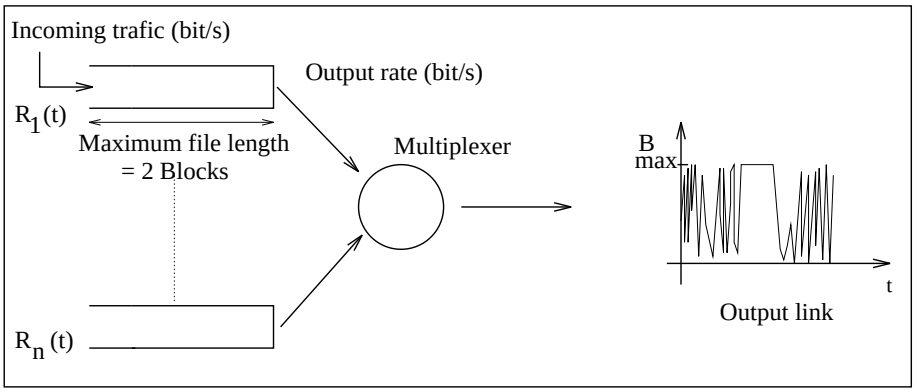

Figure 3. The traffic simulation.

4. The database

Traffic was generated according to this model for the homogeneous case (all source have the same activity parameters and peak rate taken as the bandwidth unit), for various activity parameters and numbers of sources. This traffic database generated by this model was divided in three subsets such as :

- Training set :
  - $p_{00} = 0.15 + o \cdot 0.1$; $p_{00} \in [0.15 - 0.85]$
  - $p_{11} = 0.10 + p \cdot 0.1$; $p_{11} \in [0.10 - 0.90]$
  - $n = 10 + q \cdot 3$; $n \in [10 - 26] \cup [28 - 80]$
- Validation set :
  - $p_{00} = 0.10 + o \cdot 0.1$; $p_{00} \in [0.10 - 0.90]$
  - $p_{11} = 0.10 + p \cdot 0.1$; $p_{11} \in [0.10 - 0.90]$
  - $n = 10 + q \cdot 3$; $n \in [10 - 26] \cup [28 - 80]$
- Test set :
  - $p_{00} = 0.15 + o \cdot 0.1$; $p_{00} \in [0.15 - 0.85]$
  - $p_{11} = 0.15 + p \cdot 0.1$; $p_{11} \in [0.15 - 0.85]$
  - $n = 10 + q \cdot 1$; $n \in [10 - 80]$
- $o, p, q \in \mathbb{N}^+$, $t \in \mathbb{N}$.

The training set is used to train the neural networks and the validation set allows to monitor the generalization capacity of the networks during the training process. The test set is never used during the training and only serves for performance evaluation purposes.

For each set of parameters (p_{00}, p_{11}, n) 95.10^6 time steps of traffic (K) were generated to have a good precision on the $T_i(s, n)$ and $\eta(s, n)$ (see figure 4). The thresholds s are in percentage of the maximum available bandwidth and $\in$ [0:100].

The number of examples used for the three subsets is, for T_1 (B_{max}=50) :

- Learning set : 72812
- Validation set : 37031
- Test set : 59766

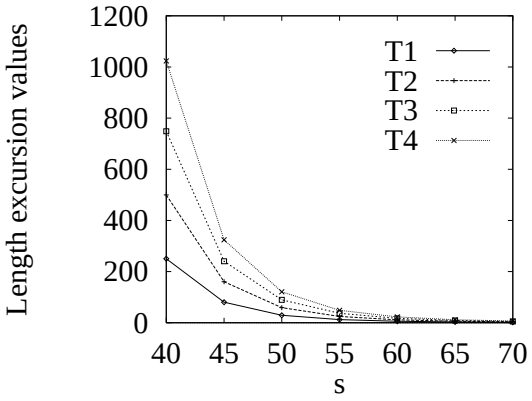

Figure 4. An example for the T_i : p_{00}=0.15, p_{11}=0.60, n=50, B_{max}=50.

5. Excursion Length Estimations

5.1. Artificial Neural Network

We show that using neural networks we can estimate :

$$T_i \text{ such that } Pr(L(\theta) \geq T_i | s, n, \mathbf{\Phi_{obs}}) = 10^{-i} \text{ i=1, 2, 3, 4} \quad (13)$$

for any number of sources n and threshold s, and with only a short observation of the traffic $\Phi_{obs} = (B(t), B(t-\tau), ..., B(t-d\tau))$ where τ is a delay and d is the size of input vector Φ_{obs} (d is link to the reconstruction dimension [6]).

We need just the actual number of already connected sources n, the rate required by the new user d_{max} ($s = B_{max} - d_{max}$) and a traffic vector $\mathbf{\Phi_{obs}}$ to estimate $T_1(s,n)$, $T_2(s,n)$, $T_3(s,n)$, $T_4(s,n)$ and $\eta(s,n)$.

The architecture chosen after several tests is a combination of five neural networks, NN_1 to NN_5, (Multilayer Layer Perceptron with a standard sigmoidal function) with one hidden layer. We trained the neural networks to estimate the following five functions :

$$\begin{aligned}
\hat{\eta} &= f_1(\Phi_{obs}, n, s,) \\
\hat{T_1} &= f_2(\Phi_{obs}, n, s, \hat{\eta}) \\
\hat{T_2} &= f_3(\Phi_{obs}, n, s, \hat{\eta}, \hat{T_1}) \\
\hat{T_3} &= f_4(\Phi_{obs}, n, s, \hat{\eta}, \hat{T_1}, \hat{T_2}) \\
\hat{T_4} &= f_5(\Phi_{obs}, n, s, \hat{\eta}, \hat{T_1}, \hat{T_2}, \hat{T_3})
\end{aligned}$$

Using cross validation we have determined the best number of hidden units to be 18 and $d = 80$ but the precise values are not crucial (if large enough). The analysis of the traffic correlation led us to choose $\tau = 1$. The number of neurons for each neural network is then 82, 83, 84, 85, 86 respectively for η, T_1, T_2, T_3, T_4 for the input layer, 18 for the hidden layer and 1 for the output layer. Each neural network was trained with the usual algorithm of backpropagation [7] to minimize the mean squared error. Training was stopped at the minimum of the validation error, so as to get a good generalization capacity.

5.2. Gaussian estimation

We review below the estimation of the excursion length in the case of a process independently and randomly sampled from a gaussian distribution. This estimation will be used below for comparison purposes. This traffic has a mean and a standard deviation estimated by :

$$\bar{B} = \frac{1}{N}\sum_{t=1}^{N} B(t) \tag{14}$$

$$\sigma_B = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(B(t)-\bar{B}\right)^2} \tag{15}$$

with N the size of the observation (Φ_{obs})

The parameters η and T_i are estimated by :

$$\eta = Pr\left(B(t) \geq s\right) = Q(s) = 1 - \int_{s}^{+\infty} \frac{1}{\sqrt{2\pi\sigma_B^2}} e^{\left(-\frac{1}{2}\left(\frac{B(t)-\bar{B}}{\sigma_B}\right)^2\right)} \tag{16}$$

$$Pr\left(L(\theta) \geq T^*\right) = Q(s)^{T^*} \tag{17}$$

$$\Rightarrow T_i = \frac{-i}{log_{10}\left(Q(s)\right)} \tag{18}$$

6. Discussion

In this section we present our results, compare them to the gaussian estimation and show how to use the estimation. To qualify the results on the different subsets we can consider, as usual, the mean error and the mean error modulus of the errors obtained (R is the size of the considered data set) :

$$\text{Mean error} = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{T_i^r}(s,n) - T_i^r(s,n)\right) \tag{19}$$

$$\text{Mean error modulus} = \frac{1}{R}\sum_{r=1}^{R}\left(\mid |\hat{T_i^r}(s,n) - T_i^r(s,n)| \mid\right) \tag{20}$$

Tables 1,2 presents the results obtained, for the mean error and the mean error modulus for different sizes of the observation window, Φ_{obs}, of the traffic. As we see , for Φ_{obs}=80, the estimations realised by the neural networks are better than with the gaussian estimator.

The scores obtained by the neural networks being approximately the same on the training, validation and test sets, we can conclude that the training process was efficient with a good generalization capacity.

From these tables it is clear that neural networks slightly overestimate the excursion lengths while the gaussian estimator is much more inaccurate and underestimates the

Table 1
The Mean errors for each $T_i \in [0-200]$ and $\eta \in [0-1]$

	Mean error on :				
Subset	T_1	T_2	T_3	T_4	η
Artificial Neural Network Estimation Φ_{obs}=80					
Trai. set	0.66	1.15	1.11	1.11	0.0024
Vali. set	0.51	1.10	0.83	0.98	0.0036
Test. set	0.48	1.19	0.82	0.86	0.0035
Gaussian Estimation Φ_{obs}=80					
Trai. set	-5.11	-6.55	-7.50	-8.28	-0.0205
Vali. set	-5.41	-6.96	-8.12	-8.92	-0.0207
Test. set	-5.43	-7.04	-8.09	-8.96	-0.0206
Gaussian Estimation Φ_{obs}=180					
Trai. set	-3.85	-5.10	-6.31	-7.19	-0.0204
Vali. set	-4.13	-5.64	-6.91	-7.91	-0.0206
Test. set	-4.06	-5.64	-6.97	-7.86	-0.0207

excursion lengths. Therefore, the neural networks can allow a tight and conservative CAC.

The errors of the gaussian estimation can be decreased by increasing the size the observation window to get a better evaluation of mean and variance of the traffic. However the larger the observation window, the longer the time to react to traffic variations. Being accurate even with a small observation window, the neural network estimations allow to adapt rapidly to traffic variations.

Such results show that neural network estimators can be used for a tight, conservative and rapidly adaptative MBCAC.

To qualify the error in term of bandwidth required and to make sure that the variance of the error does not degrade the result (see figure 5), let us define the probability of an error of magnitude k (in percent of the maximum bandwidth available) as:

$$Pr(E(k)) = \frac{1}{R}\sum_{r=1}^{R} e(k) \tag{21}$$

with

$$e(k) = \ 0 \text{ if } (T_i^r(s-k,n) \geq \hat{T_i^r}(s,n) \geq T_i^r(s+k,n) \tag{22}$$

$$\text{else } e(k) = 1 \tag{23}$$

Because both of the error and its variance are small, the probability $Pr(E(2))$ is always lower than 0.05; therefore we have a probability of 0.95 that the accuracy is of the order of 2 percent of the bandwidth.

We illustrate below the use of such estimations for CAC. Two connections request to be accepted on a node with 25 connections already connected. Both connections have a PCR of 29.4 and their QoS is defined in term of maximum excursion lengths with probabilities

Table 2
The Mean errors modulus for each $T_i \in [0-200]$ and $\eta \in [0-1]$

	Mean error modulus on :				
Subset	T_1	T_2	T_3	T_4	η
Artificial Neural Network Estimation Φ_{obs}=80					
Trai. set	4.73	6.04	6.67	6.61	0.0134
Vali. set	4.73	6.15	6.88	6.80	0.0138
Test. set	4.50	5.89	6.67	6.45	0.0133
Gaussian Estimation Φ_{obs}=80					
Trai. set	7.95	8.95	9.45	9.93	0.0255
Vali. set	8.03	9.21	9.95	10.53	0.0265
Test. set	8.29	9.32	9.93	10.51	0.0267
Gaussian Estimation Φ_{obs}=180					
Trai. set	7.11	8.09	8.84	9.46	0.0233
Vali. set	7.29	8.49	9.38	10.07	0.0243
Test. set	7.47	8.54	9.39	10.12	0.0245

10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}. The post connection excursion lengths are therefore estimated from the current traffic at the threshold s=70.6 for n=25. The results are shown on the figure 6.

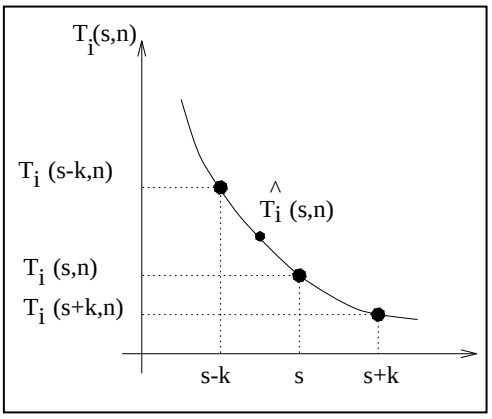

Figure 5. An explanation for $e(k)$.

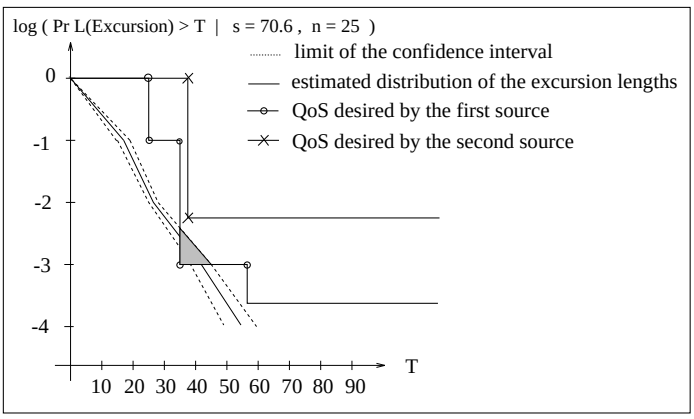

Figure 6. The acceptance zone

Guaranteeing that the Quality of Service for the already connected sources is respected, the second source will be accepted but the first one will be rejected, because the blocking probabilities are too high with regards to its quality of service.

7. Conclusion

In this contribution we have shown a new method for Connection Admission Control in the ABT/ATM network which is based on an estimation of the blocking probabilities in an ATM node using the generalization ability of neural networks. This non parametric method allows us to correctly decide the acceptance of a new connection in regard of its traffic parameters. The estimations with neural networks are accurate, need a small observation of the traffic and can therefore react quickly to traffic changes. In a future work we will study the influence of the buffer size, of traffic heterogeneity.

REFERENCES

1. Methods for the performance evaluation and design of broadband multiservice networks. The COST 242 Final Report, June 1996. Part I, Traffic Control June 4-5, 1996.
2. The ATM Forum. *Trafic management specification.* The ATM Forum, version 4.0 edition, Februar 1996. ATM Forum/95-0013R10.
3. F. Guillemin, A. Dupuis, and B. Sericola. —. In *Queuing Systems and their applications*, 1997.
4. K. Kvols and S. Blaabjerg. Bounds and approximations for the periodic on/off queue with applications to ATM traffic control. In *Proceedings of INFOCOM*, 1992. Session A.3.1, Florence, Italy, May.
5. V. Lemaire. Connection Admission Control in ATM network using an artificial neural network. In *Proceedings of Helnet 97 International Workshop On Neural Network, Montreux, Switzerland*, october 1997. http://www.robo.jussieu.fr/lemaire/publi/HELNET97.ps.
6. F. Takens. Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence, Springer-Verlag, Berlin*, 898:366–381, 1980.
7. Y. Le Cun. *Modèles connexionnistes de l'Apprentissage.* PhD thesis, Université Paris 6, 1987.