THÈSE de DOCTORAT

présentée

DEVANT L'UNIVERSITÉ DE PARIS VI

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITÉ DE PARIS VI

Spécialité : Informatique

PAR
Vincent Lemaire

Équipe d'accueil : CNET Lannion (DTL/DLI)
Composante universitaire : LIS - P&C
Secteur : Sciences

Une nouvelle fonction de coût régularisante dans les réseaux de neurones artificiels : Application à l'estimation des temps de blocage dans un nœud ATM.

Soutenue le 21 septembre 1999 devant la commission d'examen.

Composition du jury :

Stéphane CANU
Daniel COLLOBERT
Patrick GALLINARI
Maurice MILGRAM
Hélène PAUGAM-MOISY
André THEPAUT

Résumé

La théorie de l'apprentissage statistique est à la base des réseaux de neurones artificiels. On montre que les approches connexionnistes de la caractérisation du trafic et du contrôle d'admission des connexions (CAC) utilisant des mesures temps réel apportent de nombreux avantages par rapport aux méthodes paramétriques classiques. Elles s'adaptent facilement aux changements du trafic sur lequel aucune hypothèse préalable n'est nécessaire. Elles possèdent un haut niveau de performances et permettent de généraliser à des données inconnues les résultats appris.

On présente une nouvelle méthode destinée à améliorer les performances en généralisation des perceptrons multi-couches utilisés en tant que réseaux discriminants et approximateurs de fonctions. On montre comment modifier le critère d'apprentissage afin de contrôler la distribution des erreurs au cours de l'apprentissage. Ce contrôle permet d'obtenir une meilleure marge dans les problèmes de classification. Des résultats améliorant notablement l'état de l'art sur trois différents problèmes sont présentés et valident la méthode.

Une application de cette méthode à l'estimation des périodes de congestion dans un lien ATM est présentée afin de réaliser une procédure de contrôle d'admission des connexions pour le service de type ABT-DT. On montre que les réseaux de neurones artificiels entraînés sur des trafics qualifiés de "pire cas" peuvent correctement généraliser sur d'autres types de trafics, en réalisant une estimation conservative et précise des périodes de congestion. Cette méthode non paramétrique dynamique permet de décider l'acceptation d'une nouvelle connexion au regard de ses paramètres de trafic.

mots clés : ATM, apprentissage, classification, contrôle d'admission des connexions basé sur des mesures (MBAC), régularisation, marge, réseaux de neurones, service ABT, période de congestion.

Abstract

The theory of statistical learning provides foundations for artificial neural networks. We show that connexionnist approaches to traffic characterisation and to admission control using real time measurements exhibit numerous advantages over classical methods. They easily deal with traffic changes for which no prior hypotheses are needed. They are highly reliable and enable to generalise previously learned results to new data not known before.

A novel method is presented to enhance the generalisation performances of multi-layer perceptron used as discriminant networks and function approximators. We clearly show how to modify the learning criterium in order to control the error distribution. This control allows to obtain a better margin in classification problems. Results are given for three different problems. These results substantially improve state of the art results and validate the method.

This method is applied to estimate the congestion period in an ATM link to realise the connexion admission control procedure for ABT-DT services. It is shown that artificial neural network trained on worse-case traffic correctly generalise to other traffic to realize a conservative and accurate estimation of the congestion period. This non parametric dynamic method allows to correctly decide the acceptance of a new connection in regard of its traffic parameters.

keywords : ATM, learning, classification, measurement based connexion admission control (MBAC), regularisation, margin, neural networks, ABT service, congestion period.

Remerciements

Je tiens tout d'abord à remercier Daniel Collobert pour m'avoir fait confiance d'être à même de traiter un sujet a priori éloigné de ma formation universitaire. Je voudrais lui dire toute ma gratitude de m'avoir accepté dans son équipe et de m'avoir soutenu dans les moments difficiles. Je le remercie aussi de m'avoir permis de m'écarter un peu du sujet de départ afin de développer la nouvelle fonction de coût présentée dans ce mémoire, de m'avoir même encouragé et conseillé a ce sujet, merci Daniel...

Je tiens à remercier Fabrice Clérot pour tout le savoir qu'il m'a transmis, pour toute l'attention qu'il a apportée à mes travaux et pour sa patience. Il m'a prodigué des conseils précieux, avisés et rigoureux sur mes travaux. Je le remercie aussi, en lui associant Olivier Bernier (alias Mr Red), pour avoir traduit mon Anglais dans une langue intelligible, pour tous les conseils rédactionnels qu'ils m'ont prodigués, autant sur ma thèse que sur les articles que j'ai publiés.

Je dois remercier aussi Pascal Gouzien pour son assistance informatique et pour m'avoir laissé un "peu" de temps de calcul sur les CPUs. Je le remercie aussi pour les expérimentations qu'il a bien voulu mener pour moi.

Je remercie Daniel Bardouil, véritable bibliothèque ambulante, qui m'a distribué généreusement tous les articles et toutes les références dont j'ai eu besoin.

Je remercie aussi les membres du jury, pour l'intérêt qu'ils ont porté à mes travaux : Stéphane Canu, Patrick Gallinari, Maurice Milgram, Hélène Paugam-Moisy, André Thépaut.

Je remercie tous les autres membres permanents ou non de l'équipe TNT, avec qui j'ai partagé ces trois ans : Paul Anizan, Michel Collobert, Raphaël Féraud, Joel Guerin, Sebastien Marcel (alias Mr Del), Bernard Rolland et Jean-Emmanuel Viallet. Merci à tous pour votre sympathique présence autour de la cafetière et des nombreuses discussions de travail et "philosophiques" qui y ont eu lieu. TNT c'est quelque chose...

Je tiens aussi à remercier France-Télécom qui a financé ma thèse et les travaux de recherche auxquels j'ai participé.

Et un clin d'oeil à mes ami(e)s de toujours....

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 Préambule

L'Internet connaît actuellement une forte médiatisation. Issu du projet ARPA de la défense américaine, ce réseau s'est peu à peu développé en s'ouvrant au trafic en provenance du monde de la recherche et de l'enseignement. Les conditions de déploiement de ce réseau, liées à sa généalogie ainsi qu'à sa structure décentralisée (conçue pour résister à une attaque atomique), lui ont permis de s'étendre rapidement et sur une large échelle, sans qu'il y ait pour autant de prise en charge par un opérateur particulier au niveau de la planification et des investissements. L'exemple de l'Europe est illustratif de ce point, notamment à travers l'Ebone, la dorsale majeure du réseau en Europe, dont la gestion est assurée par un ensemble de partenaires et dont le déploiement se fait par raccordement de différents sous-réseaux. Sur ce cœur de réseau se sont peu à peu installées des sociétés commerciales dont la présence a motivé la vente au détail d'accès au grand public par des fournisseurs. Un effet de mode ainsi qu'un grand nombre d'utilisateurs résidentiels ont motivé la présence sur le réseau d'autres sociétés à vocation commerciale, créant un cycle propice au développement des réseaux de consultation, comme ce fût le cas pour le réseau Télétel. Les limites de ce modèle sont cependant aujourd'hui atteintes avec l'afflux massif du trafic commercial sur le réseau (ce qui sature sa capacité) et le ralliement de l'ensemble des grands acteurs de services. A ce modèle, qui tirait principalement son financement de subventions gouvernementales de manière directe ou indirecte, est en train de se substituer un autre qui relève directement de la logique commerciale. Des opérateurs principalement issus du monde de la téléphonie proposent en effet aujourd'hui des offres de raccordement reposant sur des infrastructures qui leur sont propres, que ce soit au niveau national comme au niveau mondial. Ces offres sont principalement tournées vers les entreprises et regroupent sous le terme générique d'Intranet une offre complète de services.

Le réseau des réseaux Internet repose sur la famille de protocoles IP (Internet Protocole). IP est un protocole de transport de datagrammes[1] sans connexion. Chaque trame transporte son adresse de destination. Elles sont toutes routées indépendamment les unes des autres : c'est ce qu'on appelle un réseau de datagrammes. Ce mode de fonctionnement ne permet cependant pas d'assurer de contrôle de flux et ne peut permettre de garantir le séquencement des trames à l'arrivée. Pour pallier ces problèmes, des protocoles de niveau supérieur permettent d'émuler un mode connecté (TCP Transmission Control Protocol) ou de garantir un minimum de fonctions en mode datagramme (UDP User Datagram Protocol). Pour assurer le routage des trames vers les différentes adresses de destination, les routeurs doivent disposer de tables mises à jour par des protocoles de routage. Ces derniers permettent à chaque trame de trouver le cheminement optimal dans le réseau. Ils se répartissent entre les protocoles dits de gateways intérieurs qui définissent des cheminements à l'intérieur d'un système homogène de routeurs, et les protocoles de gateway extérieurs qui permettent à ces systèmes de dialoguer entre eux.

La croissance très importante dont fait l'objet l'Internet actuellement met à mal le modèle de fonctionnement décrit ci-dessus. En effet, les tables de routage que se transmettent les routeurs du cœur du réseau atteignent aujourd'hui des tailles très importantes et nécessitent des temps de parcours de plus en plus longs. Par ailleurs, les utilisateurs sont de plus en plus nombreux et de plus en plus exigeants sur les aspects touchant la qualité de service, très difficile à garantir avec les protocoles actuels, dans un contexte où aucun opérateur ne peut maîtriser l'intégralité du réseau. Pour pallier ces difficultés, des évolutions de protocole sont en cours, avec l'introduction de IPv6 et des études d'un meilleur couplage entre les techniques de l'Internet et celles des opérateurs de télécommunications, notamment à travers l'ATM (Asynchronous Transfer Mode). Les différents opérateurs qui prennent actuellement le contrôle de l'Internet devront utiliser au mieux ces techniques afin d'assurer à leurs réseaux un avantage déterminant sur leurs concurrents.

L'Internet, assemblage de technologie IP et d'un existant important, avait remplacé le réseau ARPA pour lequel ses concepteurs n'avait pas prévu un plan d'adressage suffisant. De même aujourd'hui l'Internet arrive à saturation dans plusieurs domaines. Les adresses actuelles permettent théoriquement d'adresser jusqu'à 4 milliards d'adresses, cependant du fait de l'adressage "à plat" (nécessité de connaître presque toute la correspondance entre les numéros et la localisation des équipements pour pouvoir effectuer le routage des adresses) et de la segmentation du plan d'adressage il arrive à saturation. Une des évolutions majeure actuellement, est la volonté de diminuer la taille de ces tables à travers, par exemple, le plan d'adressage complet de IPv6 dans lequel une hiérarchie comparable à celle

[1]Dans ce mode de transport l'étiquette porte l'adresse du destinataire, aucun chemin n'est marqué dans les nœuds du réseau

du téléphone est prévue.

Un autre axe d'évolution du réseau concerne la qualité de service. En effet, par construction et comme nous l'avons décrit plus haut, les réseaux IP ne permettent pas d'assurer une qualité de service visant à garantir un débit minimum à un utilisateur, voire à privilégier certains utilisateurs. Dès lors il est assez difficile de transporter sans dégradation de la vidéo temps réel par exemple. Bien que de nombreuses évolutions à court terme, protocoles de réservation de ressources, appel aux propriétés des réseaux ..., soient proposées, l'une des plus étudiée est la solution dite IP sur ATM. Cette dernière a pour dessein de transporter de l'IP sur ATM et essaie d'exploiter la capacité d'ATM à garantir des débits pour les connexions point à point. En identifiant les flux IP à privilégier il sera alors possible de leur associer un lien ATM, éventuellement dédié, pour assurer la classe de débit souhaitée. On peut d'ailleurs noter sur ce point que certaines solutions proposées pour apporter des garanties de qualité de service dans ce réseau sans connexion reposent en fait sur des mécanismes en mode connecté. Des évolutions de ce type permettront dans le futur de rendre sur les réseaux IP des services du type de ceux qui sont envisageables sur des réseaux connectés comme la téléphonie ou la visiophonie.

Lorsque l'Internet sera supporté en partie par la technologie ATM et à des prix réels d'exploitation on peut raisonnablement penser que les clients d'Internet seront désireux d'obtenir enfin des qualités de service en terme de débit ou de temps de transmission.

1.2 Le plan du mémoire

L'apparition du mode de transfert temporel asynchrone au début des années 80 a relancé les problèmes classiques de contrôle de flux. Le contrôle de flux, qu'il soit préventif ou réactif, apparaît être un élément clef de la gestion d'un réseau de télécommunications. L'objectif de notre thèse était de déterminer si les réseaux de neurones artificiels peuvent permettre de réaliser un contrôle d'admission des connexions efficace pour le mode de transfert ABT-DT du réseau ATM. C'est pourquoi notre travail s'est articulé autour de ces deux thèmes de travail : le contrôle d'admission des connexions (CAC) et les réseaux de neurones artificiels.

Le **chapitre 2** a pour but de donner une brève introduction au réseau ATM. Après avoir introduit rapidement la raison d'être d'un réseau asynchrone haut débit, un certain nombre de points techniques seront présentés allant de la technique de transfert asynchrone jusqu'aux paramètres de contrat de trafic en passant par des indications sur ce qu'est une cellule ATM et comment une connexion est établie dans le réseau. Ce chapitre donnera au lecteur qui ne connaît pas le réseau ATM des éléments utiles à la compréhension de certaines parties du mémoire.

Le terme “contrôle de trafic” est employé dans les recommandations de l'ITU pour couvrir une variété de fonctions agissant sur une gamme d'échelle de temps, des priorités données aux cellules individuelles (par exemple au travers de l'indication de priorité à la perte) jusqu'à la gestion globale des ressources du réseau. Dans la suite, nous désignons par contrôle du trafic les actions ou les décisions qui ont un impact sur la qualité de service (QoS) des connexions. Aussi nous présenterons une description des principaux paramètres de QoS dans la première partie du **chapitre 3**.

Le contrôle de trafic est complexe essentiellement parce que les critères de QoS dépendent de caractéristiques difficiles à connaître a priori. Il existe sous l'aspect réactif ou sous l'aspect préventif, aspects qui seront explicités. Nous montrerons aussi, toujours dans la première partie de ce chapitre, que le contrôle de trafic dépend de l'échelle de temps à laquelle on se place.

L'une des principales méthodes pour contrôler le trafic est d'en extraire les caractéristiques afin de les utiliser dans le cadre d'une modélisation. Au cours de la deuxième partie de ce chapitre, après avoir brièvement discuté de la raison d'être de la modélisation, nous détaillerons les différents types de modèles de trafic actuellement utilisés. Ensuite, des critères de choix d'un modèle de trafic seront mentionnés. Ces critères nous amènerons à présenter un modèle de trafic pire cas utilisable dans le cadre d'un contrôle d'admission. Ce contrôle d'admission des connexions, l'un des contrôles de trafic préventif les plus importants dans un réseau ATM, sera présenté au cours de la troisième partie. Les méthodes de contrôle d'admission des connexions les plus utilisées seront alors détaillées.

Les développements récents de la théorie des réseaux neuronaux et de leurs applications pratiques en font une approche naturelle pour résoudre de nombreux problèmes non linéaires dans les télécommunications, tel par exemple le problème du contrôle d'admission des connexions (CAC). Ces approches connexionnistes de la caractérisation du trafic en général et du contrôle d'admission en particulier apportent en effet de nombreux avantages par rapport aux méthodes classiques. Elles s'adaptent facilement aux changements du trafic sur lesquelles aucune hypothèse préalable n'est nécessaire. Elles possèdent un haut niveau de fiabilité et de tolérances aux fautes, permettent de généraliser à des données inconnues les résultats appris (si l'espace d'apprentissage a été judicieusement choisi et est suffisamment étendu) .

Nous présentons en détails dans le **chapitre 4** et au cours de la première section ce qu'est un réseau de neurones artificiel.

Pour l'apprentissage supervisé d'un perceptron multicouche par correction d'erreur, l'algorithme le plus utilisé est l'algorithme de descente de gradient. Le calcul du gradient se fait en utilisant l'algorithme de la rétro-propagation de l'erreur. L'algorithme d'apprentissage utilisant ce procédé reste encore aujourd'hui la méthode d'apprentissage la plus largement utilisée et nous la détaillerons au

cours de la deuxième section.

Dans un processus d'apprentissage le réseau de neurones est construit en minimisant, par exemple, une fonction de coût sur un ensemble fini d'exemples, l'ensemble d'apprentissage. Cependant, le plus important est la faculté de généraliser la représentation construite par le réseau à toutes les données, y compris celles n'appartenant pas à l'ensemble d'apprentissage. Nous discuterons de ce point au cours de la troisième section.

On présente au cours de la quatrième section une nouvelle méthode destinée à améliorer les performances en généralisation des perceptrons multicouches utilisés en tant que réseaux discriminants et approximateurs de fonctions. On montre clairement la modification du critère d'apprentissage qui permet de contrôler la forme de la distribution des erreurs au cours de l'apprentissage. Cette méthode permet de minimiser à la fois les erreurs de classification et les erreurs d'estimation par une minimisation de la variance de l'erreur quadratique. Des résultats améliorant notablement l'état de l'art sur trois problèmes sont présentés pour valider la méthode.

On s'intéressera dans le **chapitre 5** aux périodes de congestion survenant dans un lien ATM, donc dans le nœud qui nourrit ce lien quand des connexions sont multiplexées en boucle ouverte. Cette étude comme nous le verrons est orientée de manière à être utilisable dans une procédure de contrôle d'admission des connexions.

Dans la première section nous préciserons un certains nombre de points concernant la capacité de transfert ABT. Dans la seconde section, pour étudier de manière quantitative le phénomène de congestion, on introduit les paramètres de qualité de service qui vont plus particulièrement nous intéresser. La politique d'utilisation de ces paramètres dans le cadre d'une procédure de contrôle d'admission des connexions sera aussi présentée pour la capacité de transfert ABT.

La troisième section présentera trois méthodes destinées à estimer ces paramètres de qualité de service à partir de mesures du trafic. La quatrième section détaillera les bases de données de trafic qui serviront dans la section suivante à réaliser une comparaison entre les méthodes d'estimations.

On cherchera plus particulièrement à vérifier si l'utilisation des réseaux de neurones entraînés sur des trafics qualifiés de pire cas peut correctement généraliser sur d'autres types de trafics.

Nous montrerons que les réseaux de neurones en général et l'architecture connexionniste proposée en particulier peuvent être utilisés pour permettre au réseau de dimensionner les temps d'attente et/ou de blocage que vont subir les blocs transmis. Cette architecture connexionniste pourra donc être intégrée dans une procédure d'estimation des périodes de congestion ayant un caractère légèrement conservatif. Elle permettra alors de décider l'acceptation d'une nouvelle connexion au regard de ses paramètres de trafic.

Chapitre 2

Le réseau ATM

Ce chapitre a pour but de donner une brève introduction au réseau ATM. Après avoir introduit rapidement la raison d'être d'un réseau asynchrone haut débit, un certain nombre de points techniques seront présentés allant de la technique de transfert asynchrone jusqu'aux paramètres de contrat de trafic en passant par des indications sur ce qu'est une cellule ATM et comment une connexion est établie dans le réseau. Ce chapitre donnera au lecteur qui ne connaît pas le réseau ATM des éléments utiles à la compréhension de certaines parties du mémoire. Pour obtenir plus de détails[1] le lecteur pourra se reporter à [Melin, 1998] [Boisseau et al., 1994] [Handel et al., 1995].

2.1 Introduction

Les nouveaux besoins en matière de réseaux sont liés à une progression rapide des échanges d'informations, par exemple :

- l'augmentation du nombre de réseaux locaux d'entreprises qu'il devient nécessaire d'interconnecter ;
- l'utilisation croissante d'applications haut débit ;
- le besoin de faire cohabiter sur un même support des services variés (la vidéo, le son, l'image...).

Une prise en compte de ces évolutions a conduit à élaborer une technique capable d'offrir des débits importants, et qui, surtout n'impose pas son rythme d'émission (cas des réseaux synchrones), mais au contraire est capable de transporter des débits variés.

[1] Pour obtenir plus de détails tout en restant bref voir "L'écho des recherches" numéro spécial ATM 1991

De ce constat émerge également l'idée de définir une nouvelle technique de transfert qui allierait simplicité et souplesse d'utilisation. A cette fin il faudrait combiner les avantages des deux techniques de transfert que sont le mode circuit et le mode paquet.

La technologie ATM (Asynchronous Transfer Mode) possède trois caractéristiques majeures qui répondent à ce souhait :

- c'est une technologie de commutation hybride entre la technique de commutation de circuit, utilisée dans les réseaux téléphoniques, et la technologie de commutation de paquets, utilisée dans les réseaux informatiques ;
- c'est une technologie orientée connexion puisque l'appel du correspondant est préalable au transfert des informations ;
- lors de cette phase d'appel, il y a une négociation de paramètres de trafic et de qualité de service, qui se traduit par différentes possibilités d'acheminement de l'information, à la manière du courrier postal (urgent, normal, lent, accusé de réception ...).

2.2 La technique de transfert asynchrone

Les techniques informatiques hauts débits concernent le mode non commuté et des distances limitées. Pour les réseaux d'interconnexion entre réseaux locaux, ordinateurs et serveurs, qui nécessitent de hauts débits sans limitation de distance, les techniques de commutation apportent un traitement individuel des voies de commutation. Plusieurs orientations sont possibles.

On peut avoir recours à des techniques se situant dans le prolongement des applications antérieures, comme la commutation de paquets, avec une extension du protocole X25 à des débits allant jusqu'à 2 Mbit/s, ou faire appel à un service assemblant les paquets en trames (que l'on peut alors comparer à des wagons).

On peut au contraire chercher à éviter la prolifération de réseaux spécialisés à quelques applications et définir un mode de commutation adapté à un environnement multi-services. Par exemple le mode de transfert asynchrone ATM, dit aussi technique temporelle asynchrone qui est une technique de transfert en mode connecté.

Il existe deux grands modes de partage temporel du support : le mode synchrone et le mode asynchrone. Le mode de transfert synchrone s'appuie sur un repérage systématique d'intervalles de temps réguliers au sein d'une trame de longueur fixe (T). La durée d'un intervalle de temps correspond à n bits de la trame, le débit affecté à un canal (voie de communication) est donc fonction de ces deux paramètres (n,T). On dispose ainsi d'un débit lié à l'horloge du réseau,

que l'on transporte implicitement. Les sources émettrices ont un débit imposé par le réseau. A l'inverse dans le mode asynchrone, les trames du support sont utilisées de façon sporadique (les intervalles de temps entre deux trames ne sont pas réguliers). Lorsqu'une application a besoin d'émettre un message, elle utilise le nombre de trames nécessaires pour que le message complet soit envoyé. Ainsi le débit est imposé par l'application et non plus par le réseau. L'indépendance temporelle introduite demande l'utilisation du procédé des files d'attente (buffer) afin d'accueillir le flux d'une application en respectant son rythme d'émission.

Les réseaux à intégration de services large bande requièrent des mécanismes complexes de gestion de trafic car il supportent aussi bien des services à large bande que des services à bande étroite : son, données, vidéo (pour une modélisation de ces derniers sur ATM voir [Cosmas et al., 1994]). L'ATM offre une totale indépendance sémantique car le transport des cellules contenant des signaux de son, d'image ou de données est réalisé de la même manière.

2.3 Le mode paquet et la cellule

La technique ATM se fonde sur le transfert d'informations de type paquets. La ressource en débit allouée à une voie de communication n'est plus définie par une structure de multiplexage synchrone rigide, mais par l'application elle-même. Cette souplesse est rendue possible à l'aide du multiplexage par étiquetage propre au mode paquets. Le mode de transfert est celui du "paquet" c.-à-d. que l'entité de transfert est un paquet (baptisé cellule pour l'ATM) possédant son champ d'informations (qui reflète les fonctions des couches d'application) et son en-tête (qui reflète les fonctions associées à l'acheminement du paquet)[2],[3] voir figure 2.1.

Le mode ATM apporte une simplification des protocoles de multiplexage et de commutation utilisés par le réseau (en théorie), en vue d'acheminer l'information dans les meilleurs délais sans se soucier de l'intégrité de l'information. Il suppose que la qualité de la couche liaison est suffisante pour reporter d'éventuels protocoles additionnels à la périphérie.

Le mode ATM est conçu pour un environnement multi-services, comportant aussi bien des applications de données à haut débit que des applications à fortes contraintes de temps, telles que l'audiovisuel ou le son de qualité, voire des applications demandant une intégrité sémantique comme le son haute fidélité ou le service synchrone dit "émulation" de circuit. Ces contraintes conduisent à un format réduit du paquet élémentaire à 53 octets, appelé cellule, et à des nœuds de commutation à files d'attente courtes, avec des mécanismes d'allocation de

[2]La petite taille de chaque paquet (53 octets pour l'ATM) permet de diminuer la "gigue" du temps de transfert.

[3]La gigue est un défaut de la transmission numérique dû à une variation à court terme de la synchronisation des éléments binaires (fluctuation temporelle).

ressources et de contrôle d'accès garantissant une intégrité sémantique suffisante.

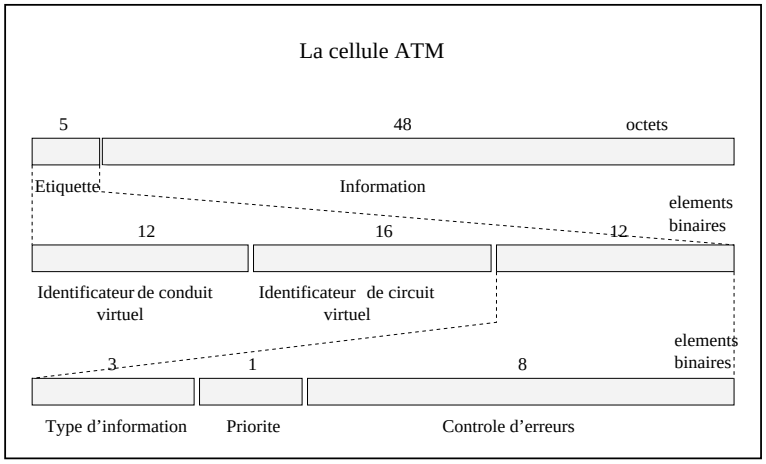

FIG. 2.1 - La cellule ATM.

En mode paquet, pour l'ATM, c'est le terminal émetteur qui définit la capacité utile du canal de communication par le biais du rythme d'émission des paquets [Boyer et al., 1996]. Il n'y a pas de réservation directe de ressources physiques mais le partage d'une ressource commune en multiplexage comme en commutation.

L'ATM peut servir de support à des réseaux commutés superposés. Il permet un accès direct aux conduits internes d'un multiplex et il associe à ces conduits les informations d'exploitation correspondantes. L'ATM apporte une grande souplesse dans la gestion des réseaux et permet d'atteindre de très haut débits (de 34 Mbits/s à 600 Mbits/s voir plus).

2.4 Acheminement des cellules

Pour qu'une communication se déroule et avant qu'une cellule de "donnée" ne transite par le réseau, il faut qu'une route soit établie entre les usagers grâce à des mécanismes de signalisation. Cette correspondance établie, le chemin physique est fixé et la séquence des paquets maintenue par le réseau car ceux-ci transitent par le même chemin. Un itinéraire doit donc être préalablement marqué dans les nœuds de commutation, via une correspondance temporaire entre une voie logique entrante sur un multiplex entrant et une voie logique sortante sur un multiplex sortant. Dans le cas du multiplexage asynchrone, on ne dispose plus d'indication temporelle, l'entité transportée doit donc contenir sa propre identification. La délimitation des cellules est un mécanisme qui permet de distinguer dans le flux transporté le début d'une cellule. Il suffit de lire l'en-tête de la cellule ainsi délimitée pour identifier la communication concernée.

Chaque cellule contient un identificateur de conduit virtuel, VP (virtuel path),

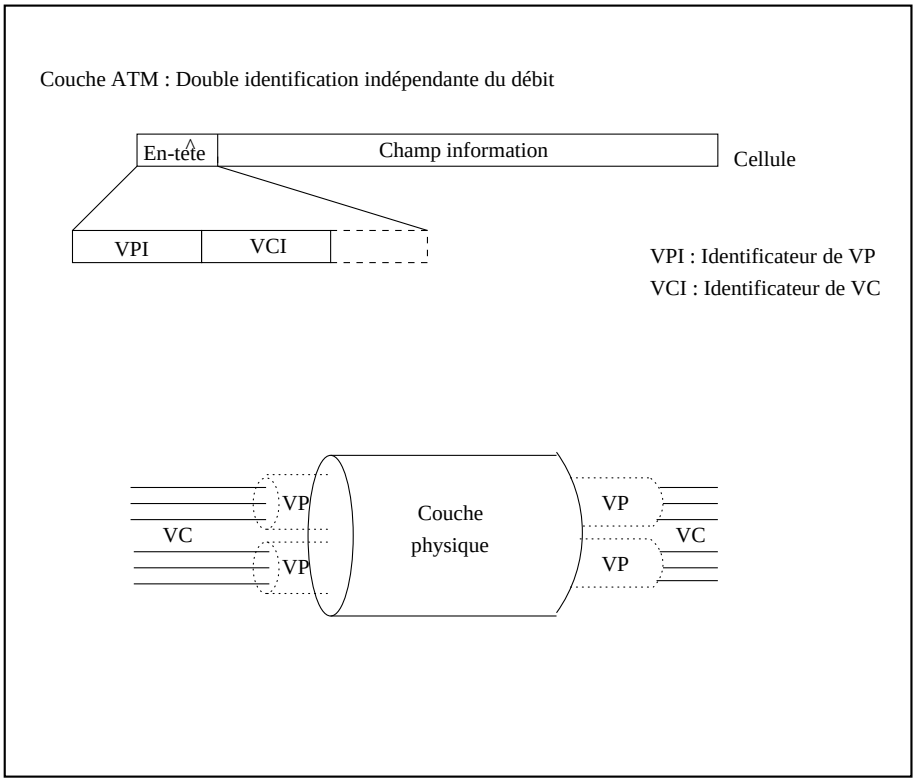

FIG. 2.2 - Le Virtual Path (VP).

et un identificateur de circuit virtuel, VC, à l'intérieur du conduit de transmission (association VP-VC). La technique comporte ainsi deux niveaux de multiplexage et distingue un réseau de conduits virtuels géré par des nœuds, les brasseurs de VP, et un réseau superposé de commutateurs de circuits virtuels opérant sur l'ensemble des conduits virtuels VP et des circuits virtuels VC.

L'identification double permet de considérer des conduits (VP) et, à l'intérieur de ces conduits, des circuits virtuels (VC) (voir figure 2.2). Ces identificateurs ne sont pas liés aux ressources, et, à condition que la somme des ressources allouées à chacun des VC dans un VP n'excède pas la capacité totale de ce VP, toutes les combinaisons sont possibles.

2.5 Le phénomène de gigue

Lorsqu'une source émet des cellules sur un réseau de télécommunication, elle le fait selon une "périodicité" définie lors de la négociation de ses paramètres de contrat de trafic. Ces cellules vont traverser des nœuds de commutation où elles vont être multiplexées avec les cellules d'autres sources dont le débit est potentiellement différent et partager avec elles les ressources disponibles. Elles

subissent alors des temps d'attente liés aux différents traitements effectués par les équipements où d'autres cellules peuvent déjà être en attente.

Le temps de traitement est supposé le même pour chaque cellule. Cependant, il n'en va pas de même pour les temps d'attente. On peut ainsi considérer ce temps d'attente comme une variable aléatoire bornée $x_{min} < x < x_{max}$. De ce fait la nature périodique du train de cellules est altérée. Cette variation est connue sous le nom de gigue et a été normalisée sous le nom de Cell Delay Variation (CDV).

Ce désordre est d'autant plus grand que la différence $x_{max} - x_{min}$ est grande et peut se retrouver au sein du réseau sous la forme d'une dispersion ou d'une agglomération. Il peut ainsi se former des "grumeaux" qui peuvent amener à une évaluation du débit instantané supérieur au débit crête et donc à un marquage ou à une destruction de cellules à tort (voir 2.7).

Les mécanismes d'allocation et de contrôle doivent prendre en compte ce phénomène dans le dimensionnement des équipements nécessaires à l'établissement d'une connexion.

2.6 L'établissement d'une connexion

L'ATM est une technique en mode connecté, c'est-à-dire qu'une phase de signalisation précède toute communication (pour les VC).

L'ATM travaille en mode paquet : on transporte des paquets de taille fixe (des cellules) qui seront multiplexés pour constituer un train de cellules. Les commutateurs permettent d'aiguiller un paquet d'un multiplex d'entrée vers un multiplex de sortie d'un même commutateur.

Pour qu'une communication se déroule, il faut qu'une route soit établie entre les usagers. L'établissement de ce chemin s'appuie sur des mécanismes de signalisation. Lorsque cette première phase à lieu, un protocole de routage s'opère de manière à trouver un chemin adapté aux exigences des usagers.

Le routage dans l'ATM qui devrait être mis en place est le routage PNNI (Private Network Network Interface). C'est un routage dynamique de la signalisation qui est destiné à être utilisé dans les réseaux privés mettant en œuvre l'adressage NSAP ([Melin, 1998] p122). Il comprend deux parties :

- un protocole de signalisation NNI (Network to Network Interface) qui reprend les informations fournis à l'UNI (User Network Interface) et les rend à leur forme d'origine à l'UNI du destinataire ;
- un protocole de routage de circuit virtuel permettant de router la signalisation dans le réseau.

Il permet la construction d'une base topologique du réseau à l'intérieur des commutateurs ATM, grâce à l'échange d'informations régulières entre les commutateurs (état des liens et des commutateurs environnants). Les informations échangées sont exprimées par des métriques et des attributs et permettent au routage PNNI de prendre en compte la qualité de service demandée par l'usager. Comme exemples de métriques et d'attributs on peut citer :

- métrique : poids administratif d'un chemin ;
- attribut : paramètres permettant d'exclure un élément de la route (si la qualité de service n'est pas satisfaite).

Grâce aux valeurs des métriques et des attributs, chaque commutateur peut obtenir à un instant donné un état estimatif du réseau. Il peut donc être en mesure d'effectuer le routage d'une requête de connexion effectuée par un usager lui étant rattaché, ce qui est appelé "routage à la source". Dans ce but une procédure de contrôle d'admission des connexions (CAC) ou "calcul de disponibilité" est implémentée. Le routage PNNI a défini un algorithme générique de contrôle d'admission des connexions (GCAC) permettant à chaque commutateur de déterminer le comportement d'un autre commutateur en fonction des données métriques qu'il possède sur ce dernier (la dernière mise à jour). Lors d'une demande de connexion le commutateur ATM de rattachement procède aux opérations suivantes :

- exécution du CAC de l'équipement, pour savoir si lui-même peut supporter la demande de l'usager ;
- détermination, en fonction du GCAC, d'une route satisfaisant les critères demandés par l'usager ;
- une fois cette route déterminée, envoi de la requête de signalisation sur cette route.

Au fur et à mesure de la progression de l'appel dans le réseau, chaque commutateur exécute sa fonction de CAC (le CAC constructeur peut être différent du GCAC). Il est possible de rencontrer un échec ce qui déclenche la fonction Crankback qui permet à un commutateur intermédiaire de recalculer un nouveau chemin selon la même procédure que le commutateur initial.

Le routage PNNI inclut pour de très gros réseaux la notion de Peer Group. Un Peer Group est un groupe de commutateurs qui sont en relation hiérarchique. Un des commutateurs du groupe est désigné pour contenir un résumé des informations de routage du groupe et peut les communiquer à l'extérieur du groupe. Ce mécanisme permet d'éviter un trafic excessif d'informations de routage dans le réseau et des calculs de routes trop complexes.

De manière à réaliser le routage en fonction des exigences des sources, un certain nombre de capacités de transfert décrites par des paramètres de contrat de trafic ont été définies et normalisées. Ces dernières vont être à présent détaillées.

2.7 Les paramètres de contrat de trafic

Avant d'entrer en communication de manière active, l'usager doit négocier un contrat de trafic en temps différé avec son prestataire de service ou en temps réel avec le réseau, contrat dans lequel sont données les caractéristiques en termes de débits et en termes de qualité de service pour la connexion demandée. L'activité de la connexion est maintenue conforme aux ressources allouées par des mécanismes de police d'accès (UPC, User Protocol Control, à l'accès usager, NPC, Network Protocol Control, à l'accès inter-réseau).

Le réseau téléphonique, basé sur un multiplexage temporel, donne une qualité de service indépendante de la charge du réseau (c'est-à-dire du nombre d'abonnés connectés en même temps). Dans le cas du réseau téléphonique commuté, le mécanisme de contrôle de flux rejette simplement les nouvelles connexions quand il n'y a plus de ressources disponibles. En effet quand tous les faisceaux du réseau sont connectés aux abonnés, il est manifeste qu'il est impossible de réaliser une nouvelle connexion. Cependant pour les abonnés déjà connectés, leur connexion leur restant propre, ils n'ont aucune conscience de ce qui se passe.

Il n'en va pas de même pour un réseau à intégration de services, certains pouvant être à débit variable dans le temps. C'est pourquoi, lors de la phase de connexion entre un usager et le réseau ATM, une phase de négociation est nécessaire : l'usager va demander une qualité de service (QoS) minimum au travers de différents paramètres [ITU, 1996b]. Une fois la connexion établie, d'une part l'usager s'engage à respecter les débits maximum et moyens négociés (selon la capacité de transfert) sous peine de pertes par sa faute, d'autre part le réseau s'engage à garantir la qualité de service accordée.

Ce protocole soulève deux difficultés : il faut être capable de prévoir quel sera l'influence de la nouvelle connexion lorsqu'elle sera admise, c'est le but du CAC, et vérifier que la source respecte son contrat de trafic établi lors de la phase de négociation.

Le but de l'administrateur de réseau est donc de garantir une QoS. Cette gestion de trafic inclut des procédures telles que :

- assurer la QoS requise pour chaque connexion ;
- surveiller et contrôler le flux du trafic à l'intérieur du réseau ;
- reconnaître et réagir aux situations anormales ;

– autoriser une nouvelle connexion au réseau si celle-ci ne produit pas une diminution de la QoS des connexions existantes.

Il y a trois différentes caractéristiques de trafic en ATM : la largeur de bande, la gigue et la latence[4]. Avec le temps quelques principaux types de services se sont dégagés à partir de ces caractéristiques : CBR, VBR, ABR, UBR et ABT (...). Des paramètres de QoS correspondants furent choisis par le forum ATM ainsi que des descripteurs de trafic à déclarer par la source [Forum, 1995a] :

– DBR (Deterministic Bit Rate) ou CBR (Constant Bit Rate) : La source déclare simplement le débit maximum qu'elle va utiliser au cours de l'appel, le "*peak cell rate*" (PCR). La réservation des ressources se fait sur la base de ce PCR.

  La définition algorithmique du PCR est reliée à une file virtuelle (donc émulée) : on considère que le débit réel est inférieur à PCR tant que la taille de la file virtuelle est inférieure à un seuil maximum L_{max}. L_{max} est reliée au temps de gigue des cellules (CDV) noté τ_{PCR} qui est un des descripteurs du trafic négocié. On a par définition la relation suivante :

$$L_{max} = PCR \,.\, \tau_{PCR} \tag{2.1}$$

  Les paramètres (PCR, τ_{PCR}) constituent les descripteurs du trafic. La définition de PCR est la même pour tous les services décrits par ailleurs.

  On réserve donc un débit constant pendant toute la durée de la connexion. Cette solution conduit les connexions à réserver un débit correspondant au débit maximum de la connexion même lorsque qu'elles n'en ont pas besoin. Cette inefficacité est d'autant plus importante que l'interactivité est forte et les variations de débit importantes.

– SBR (Statistical Bit Rate) ou VBR (Variable Bit Rate) : SBR est une capacité d'acheminement qui utilise les possibilités de multiplexage statistique et permet de négocier à la fois un débit "moyen" SCR (pour *"sustainable cell rate"*) et un PCR pour la durée de l'appel. SCR est défini de la même manière que PCR mais τ_{SCR} est plus grand.

  SCR est utilisé comme estimation locale du débit moyen valable seulement sur la durée τ_{SCR} alors que PCR est défini sur le pire cas de trafic possible conformément aux descripteurs (SCR,τ_{SCR}) et (PCR,τ_{PCR}). Ce pire cas peut être modélisé comme un trafic périodique déterministe constitué d'une rafale à PCR tous les τ_{PCR}.

[4] C'est le temps de transfert d'une donnée entre un point A et un point B d'un réseau.

La conformité au SCR se définit aussi comme un mécanisme de crédit. Ce crédit diminue lorsque l'on utilise la bande au-dessus de la valeur du débit déclaré par le SCR, et se reconstitue lorsqu'on utilise la bande en-dessous de celle-ci. Le SCR définit donc une sorte de débit moyen auquel l'utilisateur doit se conformer. Lorsque le client a épuisé son crédit, il ne peut plus dépasser la valeur SCR, jusqu'à ce que son crédit se reconstitue.

La conformité au SCR a une importante incidence sur le service rendu. Par exemple le fait de dépasser le SCR fixé à l'ouverture d'une connexion peut amener le réseau à interrompre la diffusion d'une rafale afin de garantir la QoS d'une autre source.

Il faut noter toutefois qu'avec ce type de service on n'a aucune attente et aucune perte [Remael, 1996] si on respecte le contrat passé.

Il y a deux autres paramètres existants : le CDTV (Cell Delay Variation Tolerance) et le MBS (Maximum Burst Size).

– UBR (Unspecified Bit Rate) : Ce type de contrat de trafic convient à des applications n'ayant pas des paramètres de délai contraignants, qui ne sont donc pas temps réel. Typiquement le type de contrat de trafic UBR est utilisé conjointement avec un mécanisme de contrôle de flux d'un haut niveau (tel que TCP) pour réagir correctement aux congestions. La source peut transmettre à n'importe quel débit. La congestion dans le réseau est alors évitée soit par suppression des cellules en excès, soit en marquant les cellules concernées dans le champ de leur étiquette prévu à cet effet. Il reste que le standard ne dit pas comment la source réagit à la congestion.

– ABR (Available Bit Rate) : C'est un type de contrat qui est destiné aux applications qui ne travaillent pas en temps réel, comme UBR, mais avec un effort fait pour fournir une certaine équité et garder bas le taux de pertes de cellules.

  ABR est une classe de service destinée à être utilisée pour des types de trafic ne nécessitant pas une QoS très importante notamment en ce qui concerne les paramètres relatifs au temps (les applications de courrier, transfert de fichier, ou d'autres types de trafic). L'ABR utilise les ressources laissées libres dans le réseau après une allocation de ressources des connexions établies en mode DBR et SBR. L'équité et le niveau bas du CLR (Cell Loss Ratio : Taux de perte de cellules) sont réalisés grâce à une rétroaction explicite du réseau vers la source. S'accordant avec l'information contenue dans la cellule RM[5], la source peut adapter son trafic en réponse à une situation réelle survenue dans le réseau grâce à une gestion en boucle fermée. Ce bouclage

[5]La cellule Ressource Management (RM) précède toute transmission et contient dans son champ d'information le débit réclamé par la source.

est réalisé par un contrôle de congestion adaptatif qui fait intervenir (au choix selon ce qui est implémenté) trois mécanismes :

- EFCI : Explicit Forward Congestion Indication ;
- RR : Relative Rate ;
- ERI : Explicit Rate Indication.

Evidemment la réaction de la source est faite avec un délai. Pour néanmoins préserver un certain niveau de qualité, la source peut spécifier un MCR (débit minimum), lequel sera toujours supporté par le réseau.

- ABT (ATM Block Transfert) : ABT est destiné à garantir la qualité de service au niveau du bloc plutôt qu'au niveau de la cellule : la source négocie un PCR_{max} pour toute la durée de l'appel et définit un PCR pour le bloc à venir (BCR) (voir 5.1.2 pour la définition d'un bloc). On suppose de plus que la source ne négocie pas de SCR pour la durée de la connexion.

  Les mécanismes de réservation sont basés sur PCR, τ_{PCR}, BCR, τ_{BCR}.

  Il existe un mode ABT-IT (pour Immediate Transmission) et un mode ABT-DT (pour Delayed Transmission). En ABT-IT une cellule spécifique est envoyée pour signaler le BCR requis et la trame suit immédiatement, tandis qu'en ABT-DT on attend que cette cellule spécifique réserve les ressources et revienne.

  Une telle capacité d'acheminement économise de la bande passante car les sources ont la possibilité de réserver la bande qui leur est juste nécessaire si elles connaissent leur trafic.

  Pour simplifier, si la source connaît son trafic donc son BCR, elle dit : "Je peux avoir besoin de ce débit (PCR) mais pour l'instant je n'utilise que celui-ci (BCR)".

  Pour ce type de transfert, l'ITU donne la possibilité à l'utilisateur de définir et de contrôler la structure du bloc, ce qui signifie pour l'utilisateur :

  - le choix des paramètres déclarés pour l'appel ;
  - le choix de la réaction du réseau à des situations particulières (ex : congestion) ;
  - le choix de la supervision du flot de cellules.

Pour résumer très brièvement, les aptitudes d'ATM sont :

- aptitude à traiter avec souplesse des débits allant au delà de la dizaine de mégabits par seconde et par voie de commutation ;

- aptitude à satisfaire des contraintes strictes temps réel ;
- proposer aux applications un service adapté à leurs contraintes propres sans pour autant compliquer indûment les terminaux ;
- et aptitude à garantir une qualité de service contrairement à IP.

Capacité de Transfert	Descripteurs de trafic	Qualité de Service
DBR (Deterministic Bit Rate)	PCR,τ_{PCR}	CLR $\geq 10^{-n}$ CDV petit CTD petit Pas d'usage du CLP
SBR (Statistical Bit Rate)	PCR,τ_{PCR} SCR,τ_{SCR} MBS	CLR $\geq 10^{-n}$ CDV borné CTD tolérant Usage du CLP (marquage, effacement)
ABR (Available Bit Rate)	PCR,τ_{PCR} MCR,τ_{MCR}	CLR $\geq 10^{-n}$ CDV borné CTD tolérant Usage du CLP (marquage, effacement)
ABT (ATM Block Transfer)	PCR,τ_{PCR} BCR,τ_{BCR} SCR,τ_{SCR}	Comme pour DBR mais par bloc.
UBR (Unspecified Bit Rate)	PCR,τ_{PCR}	Non garantie "Best Effort" Pas d'usage du CLP

TAB. 2.1 - Capacités de transfert du réseau ATM

Chapitre 3

Le contrôle du trafic

Le terme “contrôle de trafic” est employé dans les recommandations de l'ITU pour couvrir une variété de fonctions agissant sur une large gamme d'échelles de temps, des priorités données aux cellules individuelles (par exemple au travers de l'indication de priorité à la perte) jusqu'à la gestion globale des ressources du réseau [Cost, 96, 1, 1996]. Dans cette partie du mémoire, le contrôle du trafic désigne les actions ou les décisions qui ont un impact sur la qualité de service (QoS) des connexions. Nous présenterons une description des principaux paramètres de QoS dans la première partie du chapitre.

Le contrôle de trafic est complexe essentiellement parce que les critères de QoS dépendent de caractéristiques difficiles à connaître a priori. Il existe sous l'aspect réactif ou sous l'aspect préventif, aspects qui seront ici explicités. Nous montrerons aussi, toujours dans la première partie de ce chapitre, que le contrôle de trafic dépend de l'échelle de temps dans laquelle on se place.

L'une des principales méthodes pour contrôler le trafic est d'en extraire les caractéristiques afin de les utiliser dans le cadre d'une modélisation. Au cours de la deuxième partie de ce chapitre, après avoir brièvement discuté de la raison d'être de la modélisation, nous détaillerons les différents types de modèles de trafic actuellement utilisés. Ensuite, des critères de choix d'un modèle de trafic seront mentionnés. Ces critères nous amènerons à présenter un modèle de trafic pire cas utilisable dans le cadre d'un contrôle d'admission.

Enfin, l'un des contrôles de trafic préventif les plus importants dans un réseau ATM est le contrôle d'admission des connexions (CAC). Il sera présenté au cours de la troisième partie où les méthodes de contrôle d'admission des connexions les plus rencontrées seront détaillées.

3.1 Méthode de contrôle du trafic

3.1.1 Introduction

Dans le réseau ATM des procédures de contrôle préventif et réactif existent pour contrôler, éviter ou réagir aux problèmes de congestion et de perte. Ces procédures permettent de savoir si le réseau peut accepter de nouveaux contrats de trafic ou de vérifier que le contrat de trafic (voir 2.7) est respecté en terme, notamment, de volume. Elles doivent aussi prendre les actions nécessaires afin de protéger le réseau des violations (volontaires ou non) de contrat afin que la qualité de service de l'ensemble du réseau ne soit pas affectée.

De manière générale le contrôle du trafic peut être excessif ou laxiste. Excessif au sens ou il éliminerait trop de cellules par rapport à celles dépassant les paramètres d'un contrat de trafic donné et laxiste s'il en supprime un nombre insuffisant. La "suppression" peut d'ailleurs ne représenter qu'un simple marquage des cellules pour une priorité à la perte (si besoin était) grâce au champ CLP (Cell Loss Priority) de la cellule. Le contrôle de congestion quant à lui comprend toutes les actions qui doivent intervenir lorsqu'une congestion apparaît afin que la qualité de service (QoS) soit maintenue. La nature des performances garanties par le réseau dépend en partie de la nature et de l'efficacité de ces deux types de contrôle (contrôle réactif et préventif).

Avant d'établir la distinction entre les contrôles de type préventifs et ceux de type réactifs, tous deux ayant pour but de maintenir une certaine qualité de service, il est nécessaire d'expliciter ce qu'est la qualité de service.

3.1.2 Qualité de service

L'usager désirant se connecter au réseau ATM négocie une qualité de service, c'est-à-dire des objectifs que le réseau devra atteindre lors de la transmission des données. Cette qualité de service est traduite en terme de contrat de trafic (voir 2.7). Cependant, il n'existe pas de texte normatif concernant la procédure à suivre pour vérifier qu'une connexion reçoit bien une qualité de service compatible avec la classe de QoS négociée. On peut envisager plusieurs tests de conformité, menés sur des données réelles [Gravey et al., 1997] : mesurer la QoS offerte à une unique source sur une période de temps, mesurer la QoS offerte à plusieurs sources de même type, pour un usager donné, ou pour des usagers de même type. On peut citer les objectifs les plus courants :

– la probabilité de saturation (CLP :Cell Loss Probabilité) :

$$CLP = Pr\{\sum_{i=1}^{n} D_i(t) > C\} \tag{3.1}$$

où $D_i(t)$ représente le débit instantané d'une source i, n le nombre de sources présentes sur un lien ATM et C la capacité de ce lien. Le CLP représente la probabilité que la somme des débits instantanés (des différentes sources) soit supérieure à la capacité du lien.

– le taux de perte qui est défini comme le rapport du nombre moyen de cellules perdues au nombre moyen de cellules transmises noté CLR (Cell Loss Ratio) :

$$CLR = \frac{E\left[\left(\left(\sum_{i=1}^{n} D_i(t)\right) - C\right)^+\right]}{E\left[\sum_{i=1}^{n} D_i(t)\right]} \tag{3.2}$$

– le CTD (Cell Transfer Delay) qui représente la latence ou délai moyen de transmission. Il est défini comme la moyenne arithmétique des délais de transfert des cellules arrivées à destination.

– le CDV (Cell Delay Variation) qui représente la gigue des cellules, la variation du délai.

Il est à noter que des études de simulation, réalisées sur une gamme relativement restreinte d'exemples, montrent que dans les cas analysés les estimateurs surestiment le taux de perte de cellules et devraient donc amener à un dimensionnement conservatif [Gravey et al., 1997].

3.1.3 Contrôle réactif et préventif

Il faut distinguer le contrôle préventif du contrôle réactif. Les méthodes de contrôle réactives sont basées sur la prise en compte par les sources d'une information de congestion. En effet lors de l'apparition d'une congestion les éléments du réseau en informent les sources. Cela se traduit par un signal de réaction d'où la notion de boucle fermée. La congestion peut être alors réduite ou supprimée par la destruction des cellules qui ont été marquées précédemment pour une priorité à la perte et par une diminution du débit des sources. Il s'agit d'une technique dite "en boucle fermée" permettant au réseau d'ajuster le débit négocié vis-à-vis de son état actuel (pour une étude complète de ce système en boucle fermée voir [Forum, 1996]). Ce type de contrôle est utilisé par exemple pour les services ABR.

Dans le réseau ATM l'ordre de grandeur des débits ainsi que les distances parcourues sont grands, aussi, du fait des temps de propagation, il existe un délai entre le moment d'apparition de la congestion et celui ou les sources en sont informées. De ce fait la congestion peut avoir disparu au moment où les sources qui viennent d'être informées de son apparition réduisent leur débit. Les sources ont donc une vision décalée dans le temps de l'état du réseau en matière de congestion. On voit donc que cette méthode en boucle fermée ainsi que

d'autres méthodes réactives présentent une mauvaise adaptation et ne peuvent être utilisées exclusivement dans le réseau ATM.

Les méthodes de contrôle préventif sont basées sur des politiques d'allocation des ressources qui vont permettre d'éviter a priori les phases de congestion et/ou de pertes de cellules. L'un des principaux contrôles préventifs est le contrôle d'admission des connexions. Il sauvegarde la Qualité de Service (QoS) des connexions existantes en refusant des connexions supplémentaires si nécessaire. On peut toutefois, pour certaines d'entre elles, émettre un trafic excédant celui déclaré à condition de marquer pour une priorité à la perte les cellules concernées. Ceci permet d'aller parfois à la limite de la congestion. Il existe pour réaliser une telle procédure de contrôle préventif des mécanismes par prélèvement ou par lissage de trafic.

Les mécanismes par prélèvement se distinguent des mécanismes par lissage par le fait qu'il ne modifient pas la forme (on parle d'enveloppe d'ou le nom de REM pour Rate Envelope Multiplexing [Cost, 96, 3, 1996]) du trafic entrant, les cellules supprimées sont dites prélevées du flux original. A l'opposé, les mécanismes de lissage ont pour but de remettre en forme le trafic issu de la source (on parle de remise). Les mécanismes par lissage, eux, fonctionnement sur la base d'un fenêtrage (fenêtre sautante, glissante ...) et règlent la durée ou la position d'une fenêtre pour la traduire en terme de limites pour N et T. Les mécanismes de crédit sont des mécanismes de prélèvement qui régulent le débit d'une source en gérant un ensemble de N crédits sur une période de temps T. Toute cellule arrivant dans un intervalle de temps alors que le nombre de crédits est épuisé est détruite. L'ordre de grandeur de T et de N influencera l'ordre de grandeur d'une congestion et la quantité de cellules éliminées. Le "leaky bucket" (en anglais seau percé) fonctionne selon ce critère et permet d'obtenir un bon contrôle. Il peut être considéré comme préventif (voir [Rathgeb, 1991] pour une étude comparative des mécanismes par prélèvement).

Le contrôle préventif s'effectue généralement par "rate envelope multiplexing" (REM) ou par "rate sharing". Le REM est un multiplexage statistique du débit agrégé enveloppant et un mécanisme de lissage. Ce type de contrôle peut être réalisé en limitant le débit arrivant (de manière préventive) ou en ajustant le débit disponible dans le lien afin de s'assurer que le débit agrégé (somme des débits des connexions multiplexées) des sources $\Delta(t)$ est inférieur au débit disponible $c(t)$ ou l'excède avec une très faible probabilité ϵ. Ce débit $c(t)$, ayant été fixé du fait des paramètres de contrat de trafic des différentes sources, peut varier dans le temps (voir figure 3.1).

Ce modèle ne suppose aucune taille particulière de file d'attente mais juste son existence de manière à éviter une congestion au niveau cellule. L'objectif de ce contrôle de trafic, en limitant le nombre de connexions, est de garder le trafic agrégé en-dessous d'un seuil avec une probabilité donnée (1.10^{-9}). Il peut donc être vu comme l'implémentation d'un multiplexage sans file d'attente (bien

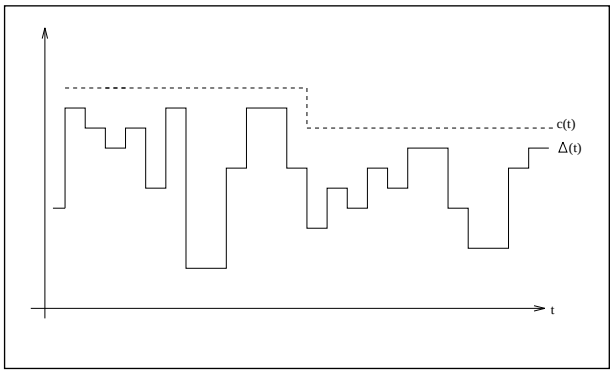

FIG. 3.1 - Rate Envelope Multiplexing

que dans la réalité il y en ait une) d'un trafic "fluide". On s'intéresse ici à la somme des trafics des différentes connexions pour effectuer le contrôle. En réalité, la file d'attente n'est destinée qu'à résorber l'asynchronisme des arrivées des cellules. Quand la taille de la file d'attente attachée à un lien de transmission est négligeable, le trafic arrivant en excès par rapport au débit du lien est écrêté et considéré comme perdu. Par conséquent quand on examine le multiplexage statistique des connexions sur un lien ATM dépourvu de file d'attente, seule l'enveloppe de débit représentant le débit instantané de la superposition des différentes connexions peut être considérée pour déterminer la perte de cellules. C'est pourquoi ce type de procédé est appelé multiplexage statistique du débit enveloppant. Ce type de contrôle est efficace lorsque les sources ont une valeur de PCR pas trop grande vis-à-vis de leur débit moyen, de manière à ce que le débit agrégé ne présente pas de grosse granularité et que l'idée d'enveloppe puisse être applicable.

Le "rate sharing" ou partage du débit disponible est un autre contrôle de trafic préventif. Ce type de contrôle utilise un multiplexage de files de grandes tailles de manière à absorber les fluctuations au niveau rafale. Dans ce cas le débit est partagé entre les différentes sources présentes dans le lien et le contrôle s'effectue sur chacune d'entre elles de manière à ce que le buffer du multiplexeur ne déborde pas. Ce contrôle est plus adapté pour des sources ayant des grandes valeurs de PCR. On raisonne alors sur la distribution de la longueur d'une file d'attente de taille infinie en fonction de la capacité du lien C.

Les différents contrôles de trafic préventifs décrits ici peuvent être comparés comme suit. Le contrôle du débit "enveloppant" peut être vu comme un calcul global où la performance est calculée indépendamment des connexions mais d'après leur superposition. Toutes les cellules de contrôle reçoivent donc la même information en fonction du trafic agrégé et du débit disponible. Au contraire, dans le cas du partage de débit la performance est calculée pour chacune des sources. Les cellules de contrôle de chaque connexion reçoivent donc une information non

seulement dépendante de la différence entre les ressources disponibles et celles utilisées mais aussi des ressources utilisées par la connexion concernée. Les avantages et inconvénients de ces deux méthodes sont alors clairs : le calcul global ne permet pas de satisfaire le critère d'équité entre les sources (impossibilité de redistribuer le débit restant et si une source viole son contrat de trafic toutes les sources sont affectées), a contrario le calcul différencié le peut mais nécessite un coût de calcul et de mesures beaucoup plus grand. Il est aussi nécessaire de connaître dans les deux cas pour quelle échelle de temps on veut réaliser le contrôle.

3.1.4 Echelle de temps

La flexibilité des débits offerts par le réseau ATM permet, au niveau de l'utilisateur, une grande variété d'applications ainsi qu'une grande diversité de terminaux connectables. Les caractéristiques du trafic produit par une connexion ATM, en terme de débit, de gigue, de latence, vont donc être assez variables et dépendantes de la nature des applications et des équipements terminaux. La prise en compte de ces contraintes nécessite de la part du réseau de transport de posséder une politique d'allocation et de partage de ses ressources adaptée au type de trafic à véhiculer et ce, aussi bien pour les connexions qu'elle décide d'accepter que pour celles qu'elle a déjà acceptées au préalable. De plus, l'utilisation des ressources doit être optimisée afin que la quantité de ressources nécessaire à chaque connexion ne soit pas surestimée, mais au contraire judicieusement calculée.

L'allocation des ressources s'effectue sur la base d'un contrat de trafic (voir 2.7), négocié entre l'utilisateur et le réseau, à partir de paramètres déclarés et d'une qualité de service désirée (QoS). Ces paramètres permettent de caractériser les débits et variations de débit de la source et permettent d'allouer les ressources. L'évaluation de la quantité de ressources nécessaire aux connexions et la décision d'acceptation ou de rejet qui en découle fait l'objet du contrôle d'admission des connexions (CAC) dont nous reparlerons.

En revanche, le partage des ressources pour des connexions dont la QoS n'est pas ou peu définie (par exemple ABR) est réalisé sur la base d'un partage équitable des ressources restantes après attribution des ressources nécessaires aux connexions gourmandes en QoS. L'actualisation des débits de sources de ce type est réalisé à l'aide d'un contrôle réactif en boucle fermée.

Ces différentes politiques d'allocation, de partage des ressources et de contrôle de trafic doivent prendre en compte l'existence de plusieurs échelles de temps aussi bien dans la description des trafics que dans l'analyse du comportement du système. On considère trois échelles de temps représentées sur la figure 3.2 :

– Niveau cellule : la fenêtre est suffisamment petite pour distinguer les cellules individuellement. Chaque source émet à un taux inférieur de plusieurs

ordres de grandeur à la capacité du lien.

- Niveau rafale : on ne distingue plus les arrivées de cellules et on considère des rafales de cellules. Les rafales peuvent être de tailles différentes. A ce niveau la granularité est ignorée. Le trafic au niveau rafale est souvent représenté par un processus de Markov à plusieurs états.

- Niveau appel : l'échelle de temps est assez grande pour ne prendre en compte que l'établissement et le retrait de connexions, selon les cas, ce niveau peut s'échelonner de quelques secondes à plusieurs heures. Le réseau se comporte en essence comme un réseau à commutation de circuits.

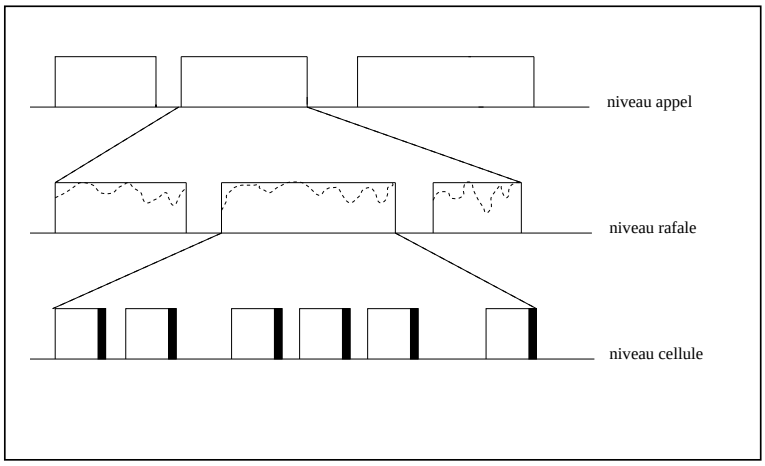

FIG. 3.2 - Les trois niveaux de temps représentant les différents types de flux.

Les flux de type CBR (voir 2.7) sont complètement caractérisés par leur débit crête PCR ce qui facilite grandement la gestion des ressources. Le réseau va déterminer une route de telle sorte que chaque lien puisse absorber un débit supplémentaire égal à PCR. Le multiplexage de ce type de connexions est donc réalisé de manière déterministe et leur étude a fait l'objet de nombreux travaux. Par contre, pour les flux à débits variable, la connaissance à elle seule de PCR ne suffit pas, aussi d'autres paramètres ont été ajoutés, tel le facteur d'utilisation (rapport débit moyen/débit crête).

De nombreux modèles de flux à débit variable ont été proposés dans la littérature [Stamoulis et al., 1994] afin d'étudier l'influence du trafic sur les différentes échelles de temps précitées. Ces modèles se distinguent par leur caractéristiques : temps continu ou temps discret ; génériques ou adaptés à un trafic particulier ; prennent en compte ou non les corrélations entre les inter-arrivées au niveau cellule. Les modèles génériques de type MMPP (Markov Modulated Poisson Process) permettent de prendre en considération les niveaux cellules et rafales. Ils décrivent des processus de Poisson dont le taux d'arrivée des cellules est modulé

par une chaîne de Markov. L'étude de ce modèle a conduit [Norros et al., 1991] à des courbes de dimensionnement de la file du multiplexeur qui présentent l'allure de la figure 3.3.

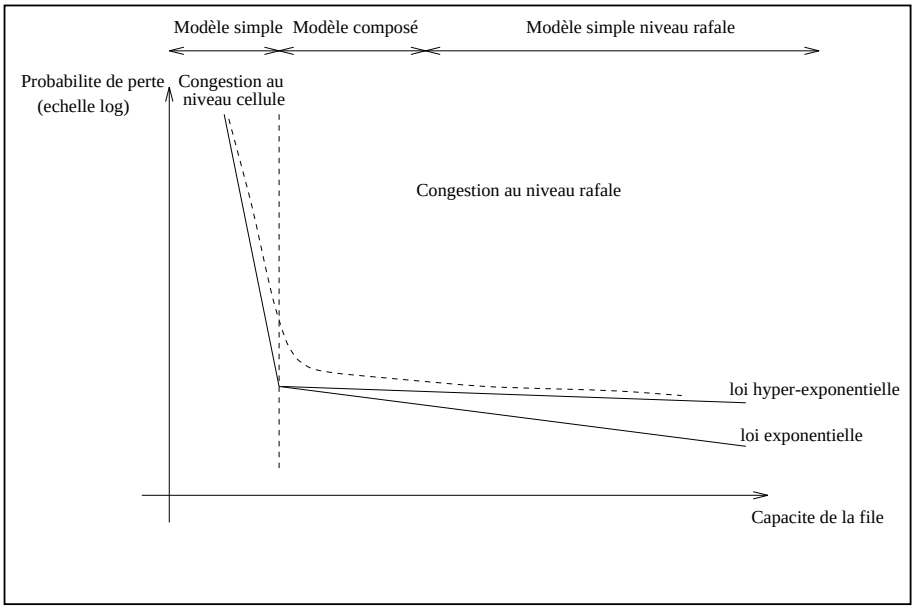

FIG. 3.3 - Impact des différents types de flux et modèles de trafic associés utilisables.

Ces courbes comportent deux parties distinctes correspondant aux régions de congestion des niveaux cellules et rafales. L'allure de la courbe de perte s'explique de la façon suivante (on trouvera dans [COST, 1991] p12-22 une description détaillée). Pour de faibles valeurs de la capacité de la file (pour une définition de la notion de capacité voir 3.2.2) le rejet est dû à l'arrivée simultanée de quelques cellules sur une courte période de temps. Pour des capacités plus grandes, cette approximation n'est plus valable et il faut prendre en compte la corrélation entre les arrivées des cellules. Les pertes sont dues aux phénomènes de rafales, la probabilité de perte diminue nettement moins vite en fonction de la capacité de la file et dépend de plus de la loi selon laquelle sont distribuées les tailles des rafales. La congestion au niveau cellule devient alors négligeable par rapport à la congestion de niveau rafale. L'étude du multiplexage peut alors se faire à l'aide d'un modèle fluide qui ne considère que le niveau rafale. L'approximation fluide consiste à gommer l'irrégularité du processus d'arrivée des cellules en considérant une rafale comme un flux continu de bits ayant un débit constant égal au débit moyen de la rafale. La présence de rafales provoque donc un phénomène de surcharge locale qui peut avoir un impact non négligeable sur le dimensionnement du système, nécessitant des files de grandes tailles.

3.2 La modélisation du trafic

3.2.1 Introduction

Une façon détournée d'analyser un système consiste à représenter son fonctionnement grâce à des outils descriptifs, mathématiques ou autres, permettant d'approcher son comportement. Cette phase de substitution se nomme modélisation et fournit une représentation, une description du système. Le principal but de la modélisation du trafic est de pouvoir par la suite réaliser une identification des paramètres du modèle de manière à pouvoir obtenir les caractéristiques statistiques importantes du trafic. Ces paramètres associés au modèle sont nécessaires aux méthodes d'évaluation des performances et de dimensionnement et donc au contrôle en général.

Dans les réseaux à haute vitesse basés sur un mode de transfert asynchrone, nous avons vu (voir 2.2) que les sources peuvent émettre des paquets ou des cellules à un rythme quelconque (dépendant de l'application connectée au réseau). Pour résoudre le problème de la différence du rythme en émission et du rythme en réception, des mémoires tampon sont utilisées. Ceci forme un système complexe à base de files d'attente qui peut être modélisé. La théorie des files d'attente dans un réseau ATM va donc jouer un rôle important dans la compréhension de son fonctionnement. Cette théorie sera le lien entre les différents types de trafics présents dans le réseau et le dimensionnement des ressources nécessaires à leur acheminement. D'une manière générale, la plupart des études analytiques menées sur le réseau ont pour but, à partir d'hypothèses sur le trafic, d'établir des distributions de probabilité relative aux taux de perte, aux taux de congestion, à la répartition du débit disponible.

De nombreux modèles de file d'attente ont déjà été résolus analytiquement notamment pour les trafics qualifiés de pire cas [Kleinrock, 1975b; Kleinrock, 1975a]. Cependant leur applicabilité dépend de la pertinence de la modélisation du trafic en entrée. Le problème se complique encore lorsque l'on s'intéresse à la superposition de plusieurs trafics, ou au mélange de trafics de caractéristiques différentes. La résolution analytique apporte une "solution exacte" à de nombreux problèmes tant que ceux-ci ne font pas intervenir un grand nombre d'hypothèses. Celles-ci se rapportant souvent aux caractéristiques du trafic en entrée du système et aux mécanismes associés à ce dernier (qui par exemple le génèrent).

Les réseaux à haute vitesse font intervenir un nombre important de paramètres différents qu'il est difficile de prendre en compte lors d'une résolution analytique. Par exemple un facteur est lié au fonctionnement même du réseau ATM qui accorde une priorité relative aux différents flux présents en son sein. On peut donc être amené à étudier des systèmes à arrivées prioritaires, ce qui complique fortement la résolution analytique proposée. Du fait de cette complexité, de nombreuses hypothèses simplificatrices sont alors utilisées, souvent restric-

tives par rapport à la réalité et dont la pertinence n'est pas toujours démontrée. L'approche analytique consiste alors souvent à étudier des systèmes où le trafic généré par une ou plusieurs sources est approximé par la superposition de sources de caractéristiques identiques.

Le modèle de trafic doit donc être approprié à l'application qu'on veut en faire. Le modèle qui sera choisi doit indiquer à quel niveau (voir 3.1.4) se placer pour contrôler le réseau, les paramètres à surveiller à ce niveau et les actions à engager lorsque des problèmes surviennent. La question est donc de classer les facteurs susceptibles d'influencer le choix d'un modèle de trafic ou d'en modifier les paramètres. De nombreux modèles de trafic à débit constant [Roberts et Virtamo, 1991; Humblet et al., 1993] ou à débit variable [Bae et Suda, 1991; Onvural, 1994; Stamoulis et al., 1994] ont été proposés dans la littérature. Il est impossible ici de détailler tout les modèles de trafic existants ou les moyens de les caractériser aussi nous n'en présenterons qu'une classification succincte (voir par exemple [Cost, 96, 3, 1996] pour plus de détails). En fait on peut dire qu'il n'existe pas de modèle universel représentant de façon adéquate tous les services proposés à ce jour.

3.2.2 Les modèles de trafic

Les caractéristiques du trafic dépendent d'un grand nombre de facteurs, qui interviennent en général à des niveaux différents. Supposons qu'on puisse les étudier séparément et donc n'étudier qu'un niveau à la fois. Ce qui veut dire supposer par exemple que les tailles et les inter-arrivées successives (des cellules ou des rafales) sont des variables aléatoires indépendantes et identiquement distribuées. Dans ce cas on peut utiliser un modèle simple qui néglige les corrélations temporelles dues aux facteurs intervenant aux échelles de temps supérieures. Ceci est possible par exemple lorsque le débit des sources est très inférieur au débit du réseau.

Ces modèles simples sont représentés par des suites aléatoires indépendantes et identiquement distribuées modélisant l'inter-arrivée entre paquets et la taille des paquets à une échelle de temps donnée. Les distributions sont ensuite paramétrées en fonction de leur premiers moments : moyenne, variance. La distribution la plus fréquemment utilisée est exponentielle. Son utilisation se justifie car elle constitue un cas limite, correspondant à la superposition d'un grand nombre de sources indépendantes.

Avec ces modèles simples les inter-arrivées successives des paquets sont supposées indépendantes or dans la réalité ce n'est pas le cas. La conséquence est une mauvaise qualité de l'estimation des véritables paramètres du trafic. Pour tenir compte des corrélations temporelles, des modèles composés à plusieurs niveaux ont été proposés. Les niveaux pouvant être décomposés en sous-niveaux

(voir 3.1.4).

Les modèles composés théoriques habituellement considérés possèdent des silences ou des inter-arrivées exponentiels. Le volume des rafales est aussi parfois exponentiel, parfois quelconque et défini par ses deux premiers moments. L'étude des modèles de trafic composés étant complexe, certains ont cherché à définir des approximations permettant d'attribuer à chaque source un débit moyen constant dit efficace. Il s'agit de la capacité équivalente.

Pour définir la capacité équivalente, on considère un système composé d'une file d'attente de taille K vidée à un débit R, et nourrie par une source on/off, modélisée par une chaîne de Markov à deux états (voir figure 3.4). Lorsque la source est active (on), elle émet ses données au débit crête D, dans l'état inactif (off) elle n'émet rien. Les périodes d'activité et d'inactivité sont donc indépendantes et identiquement distribuées suivant une loi exponentielle. Le paramètre B représente la durée moyenne des périodes d'activité et M le débit moyen sur une période activité/inactivité.

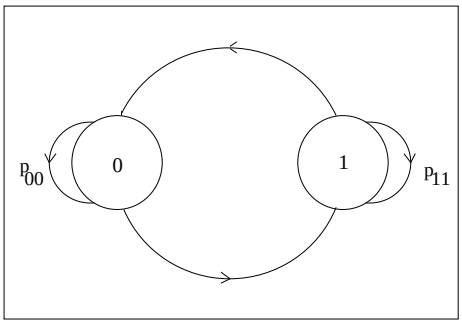

FIG. 3.4 - Modèle de Markov à deux états.

On s'intéresse alors à l'expression de la probabilité de débordement de la file soumise à ce trafic, en fonction du débit R du système. Autrement dit : "quel débit minimal faut-il fixer pour que la probabilité de débordement de la file d'attente soit inférieure ou égale à un seuil ϵ". Ce débit R est appelé capacité équivalente du trafic émis par la source. Cette valeur comprise entre M et D constitue une base pour l'allocation de ressources. La différence $D - R$ est significative de l'économie qui peut être réalisée pour la réservation de la bande passante associée à la connexion. La différence $R - M$ est quant à elle représentative du surplus de capacité qu'il faut réserver, par rapport au débit moyen, pour satisfaire une qualité de service donnée (ici la probabilité de perte). La résolution analytique de ce système existe mais notre but ici n'est pas d'en faire la démonstration. On admet généralement que la capacité équivalente est alignée sur le débit crête quand l'utilisation du système est faible et qu'il n'y a que quelques connexions, et qu'elle s'approche du débit moyen quand l'utilisation est forte et qu'il y a beaucoup de connexions.

Enfin, mais nous les avons déjà cités et en partie exploités, nous mentionnerons le modèle on/off et le modèle MMPP.

3.2.3 Critères de choix d'un modèle de trafic

Les modèles de trafic sont plus ou moins complexes et sont fonction d'un certain nombre de paramètres. La question est de savoir comment choisir le modèle approprié à un problème en particulier. Nous avons à notre disposition essentiellement deux critères de choix : l'échelle de temps concernant le problème et la complexité et/ou précision de la résolution.

L'échelle de temps dépend des facteurs concernant le problème que l'on cherche à résoudre et par conséquent de l'utilisation que l'on veut faire du modèle au sein du réseau de télécommunications. On peut dans certains cas se contenter de considérer un seul facteur, donc une seule échelle de temps et dans ce cas un modèle simple suffit. Les distributions doivent être proches de la réalité sans nécessiter une analyse trop complexe. Si nécessaire, c'est-à-dire s'il faut prendre en compte des facteurs intervenant à deux ou trois échelles de temps, on utilise alors des modèles composés.

Le choix du modèle dépend également du temps disponible pour sa résolution. Si en effet la formule pour le modéliser doit être implémentée dans un algorithme d'optimisation, on utilisera une formule simple basée par exemple sur un modèle fluide et a contrario on adoptera un modèle plus détaillé.

Enfin, les informations disponibles déterminent aussi le choix du modèle car il est inutile de choisir un modèle complexe dépendant d'un grand nombre de paramètres si l'on est incapable de lui fournir ou d'évaluer les grandeurs nécessaires à son utilisation. En effet, il faut que les paramètres du modèle qui doivent être mesurés sur le "terrain" soient accessibles. Nos discussions avec des gestionnaires réseau nous indiquent que la quantité et le nombre de mesures différentes sont très limités.

Au vu de ce qui précède, peu de modèles peuvent être choisis et les modèles les plus simples sont souvent les plus usités. Parmi ceux-ci, on peut citer le modèle on/off pour lequel le flot de cellules issues d'une source est modélisé comme une succession de périodes actives durant lesquelles les cellules sont générées et des périodes de silence durant lesquelles aucune cellule n'est émise. C'est d'ailleurs à l'aide de ce modèle que la capacité équivalente a été définie (voir 3.2.2). On peut aussi citer le modèle MMPP (Markov Modulated Poisson Process). Ce modèle, corrélé est couramment employé pour approximer le flot résultant de la superposition d'une collection de sources on/off avec un flot constant de données [Heffes et Lucantoni, 1986].

3.2.4 Un modèle conservatif pour l'étude du pire cas

Des études menées à l'INT d'Evry en collaboration avec le NIST ont cherché à trouver un modèle de trafic pire cas [Zamani, 1997; Marot, 1997a; Marot, 1997b]. Elle visent à caractériser le trafic ATM en utilisant des mesures et en en déduisant des modèles pire cas. Le but étant, pour quelques exemples de trafic ATM, de faire des mesures au niveau de la cellule afin de pouvoir obtenir des informations sur la loi du trafic. Une autre approche consiste à obtenir des informations sur les modèles des applications et des utilisateurs du réseau. Des modèles obtenus on pourrait alors déduire un générateur de trafic. L'équipe a soumis un projet de caractérisation du trafic multimédia, comportant des modèles des utilisateurs (Université de Vienne), des modèles de mobiles (Stuttgart) des modèles de satellites (Université Surrey) et des modèles de réseaux hauts débits (INT). L'un des objectifs de ce projet est de réaliser une plate-forme de simulations, comprenant des générateurs de trafics pire cas et différents réseaux.

A côté de ces études, qui semblent très spécifiques au trafic multimédia, on définit de manière générale, un trafic pire cas pour une connexion ayant un contrat de trafic donné comme le flux de cellules qui respecterait le contrat de trafic négocié tout en nécessitant le maximum de ressources au niveau du réseau. Cette section décrit le trafic pire cas pour des connexions qui ont négocié un contrat de trafic de type DBR ou SBR (voir 2.7).

Dans le cas d'une connexion DBR, le contrat de trafic étant caractérisé par le couple (T , τ), le trafic pire cas est un flux périodique de rafales dont la longueur est alignée sur le maximum admissible vis à vis du contrat de trafic (MBS) (voir figure 3.5). Dans certains cas, des flux d'arrivée plus complexes produisant de pires performances ont été proposé [Doshi, 1994]. Cependant, la présente définition du trafic pire cas est communément acceptée comme une bonne hypothèse de travail ([Roberts et al., 1996] p53).

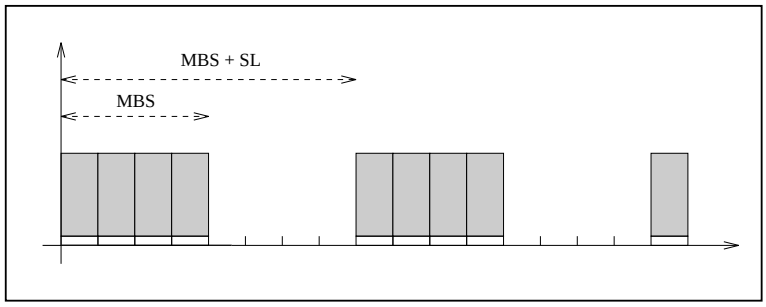

FIG. 3.5 - Trafic pire cas pour une connexion DBR.

Ce trafic pire cas est communément accepté et applicable localement pour d'autre types de connexions (SBR, ABT, ...). Supposons en effet maintenant qu'en plus du descripteur de trafic PCR (T_{PCR}, τ_{PCR}) la connexion soit aussi

caractérisée par le couple (T_{SCR}, τ_{SCR}) (voir 2.7 pour la définition du mécanisme reliée au SCR). Le trafic pire cas correspondant à ces paramètres est une source on/off périodique avec des rafales à deux niveaux (voir figure 3.6) ([Roberts et al., 1996] p55).

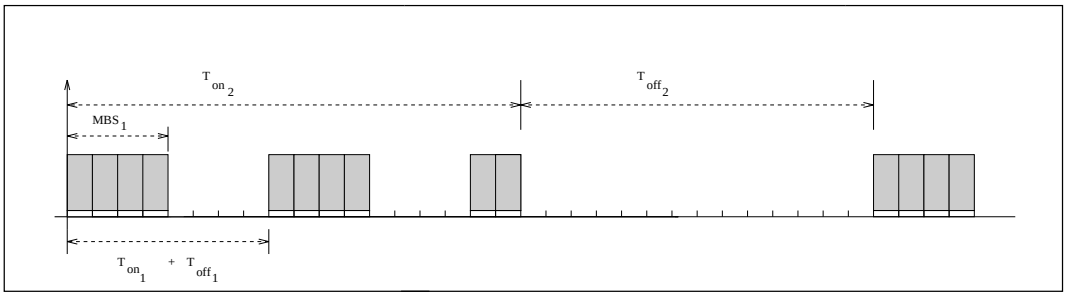

FIG. 3.6 - Trafic pire cas pour une connexion SBR = Trafic pire cas d'une connexion DBR appliqué localement pour la phase d'émission de la connexion SBR.

L'impact néfaste des trafics pires cas sur le comportement d'une file d'attente étant avéré, plusieurs possibilités s'offrent alors à l'opérateur de réseau pour ouvrir des connexions au débit crête tout en étant sûr de remplir ses engagements en terme de qualité de service. L'une d'elle consiste à développer une politique d'allocation des ressources ou d'acceptation des connexions entrantes prenant en compte ces trafics pires cas.

Le trafic pire cas conservatif pour l'allocation des ressources d'une connexion ayant un contrat de trafic ABT (voir 2.7 localement assimilable à un contrat de trafic DBR) consiste donc en une source on/off ayant un débit maximal égal au DBR négocié pour toute la durée de la connexion, débit qui est supérieur à tous les BCR.

3.3 Contrôle d'admission des connexions

3.3.1 Définition

La procédure de contrôle d'admission des connexions (CAC) est un contrôle préventif qui sauvegarde la Qualité de Service (QoS) des connexions existantes en refusant des connexions supplémentaires si nécessaire. C'est un contrôle en boucle ouverte qui ne modifie pas les caractéristiques d'émission des sources lorsque celles-ci ont été acceptées, par conséquent les conséquences potentielles d'une congestion doivent avoir été anticipées avant son apparition, ce qui diffère des contrôles réactifs (voir 3.1.3).

Afin d'évaluer les quantités de ressources nécessaires à l'acceptation de la

nouvelle connexion (acceptation au niveau cellule, rafale ou appel, voir 3.1.4), elle dispose d'un ensemble de paramètres composant un descripteur de trafic (voir 2.7) fourni par la connexion et permettant de caractériser le trafic généré. Ces paramètres ont été défini par l'ATM forum, et, bien que des trafics ayant les mêmes descripteurs peuvent avoir des "propriétés" différentes dans chaque niveau de caractérisation d'une connexion (cellule, rafale, appel), ils doivent être le principal vecteur d'entrée dans le schéma décisionnel de l'acceptation ou du rejet.

Il s'agit donc, compte tenu des descripteurs de trafic (qui ne permettent pas de cerner avec une extrême précision les caractéristiques inhérentes d'un flux) et de la superposition de ces flux due à leur multiplexage, d'accepter ou de rejeter une nouvelle connexion de manière à ce que :

- la qualité de service de la connexion soit satisfaite ;
- la qualité de service des connexions existantes soit maintenue.

Répondre à une telle question est loin d'être trivial, du fait même du multiplexage de flux hétérogènes aux propriétés statistiques variées, de plus avec des mécanismes de priorité entre ces flux. Aussi pour restreindre l'étendue du problème on se limite souvent à l'étude du CAC par classe de trafic, supposés homogènes et de même priorité.

Ce contrôle a eu une place réservée dans le réseau ATM et a été nommé, par l'ITU[1] , CAC (pour Contrôle d'Admission des Connexions) et représente une partie cruciale de la gestion temps réel du réseau ATM. Le CAC doit décider en chaque nœud d'un chemin entre deux points A et B du réseau si la nouvelle connexion peut, ou ne peut pas, être acceptée. La procédure de CAC étant basée sur l'état du réseau et des paramètres de contrat de trafic d'une nouvelle connexion, on peut différencier deux types de CAC : les CAC statiques et les CAC dynamiques.

Les CAC statiques sont utilisés lorsque seuls les paramètres de contrat de trafic déclarés de la source sont utilisés. Cette approche consiste souvent à surestimer globalement le flux considéré en allouant plus de ressources, et par conséquent, en effectuant le contrôle à un débit proche du débit crête. On essaie de garantir coûte que coûte la qualité de service individuelle de chaque source éventuellement au détriment d'une optimisation globale des ressources. Le paramètre important alors à contrôler est la gigue introduite sur le trafic (CDV). Cette approche est coûteuse car les ressources du réseau peuvent être sous-exploitées. Par exemple l'allocation basée sur le débit crête réserve une largeur de bande passante sur le lien égale au débit crête de la source. Ainsi, la nouvelle connexion est acceptée si

[1] C'est seulement récemment que certaines propositions ont été introduites dans les documents de standardisation [Forum, 1995b; ITU, 1996a].

la somme de tous les débits crêtes D_i des n sources déjà connectées et du débit maximum D_{new} de cette nouvelle source est inférieure à la capacité C du lien :

$$\sum_{i=1}^{n} D_i + D_{new} \leq C$$

Dans la suite de ce chapitre on nommera D_i le débit crête d'une source i.

Les méthodes de CAC dynamiques (ou encore appelés adaptatifs) consistent à introduire un processus d'adaptation des ressources allouées et des paramètres contrôlés, en fonction d'observations faites sur le trafic en cours et sur l'état du réseau. Beaucoup considèrent comme adaptatif un CAC qui utilise toujours le même processus de décision (statique), mais dont les paramètres d'entrée (les mesures) sont réactualisées régulièrement.

Dans les deux cas le réseau doit être à même de caractériser son trafic afin de rendre un verdict. Il existe, pour ce fait, deux méthodes : soit le CAC utilise un modèle de trafic décrivant ses propriétés statistiques, on parle de méthodes paramétriques, soit le CAC n'utilise pas de modèle de trafic, on parle alors de méthodes non paramétriques.

Le problème peut globalement être défini en estimant le niveau de violation après l'acceptation du nouvel appel. La probabilité d'accepter un nouvel appel en est déduite. Cette probabilité est conditionnée par les paramètres de contrat de trafic du nouvel appel et par l'état du réseau.

On constate ainsi que le CAC (statique ou dynamique) mis en œuvre lors d'une demande de connexion est relié aux procédures de caractérisation du trafic en cours dans le réseau (paramétriques ou non paramétriques). C'est pourquoi le problème de la représentation du trafic qui permettrait un contrôle robuste des admissions a été au centre d'intérêts de recherche ces dernières années. Ces travaux ont débouché sur de nombreuses propositions dont nous proposons un état de l'art non exhaustif. Le but ici n'étant pas de donner toutes les méthodes existantes, dont le nombre est proche du nombre d'auteurs, mais de présenter quelques-uns des CAC adaptatifs de manière à dessiner les grandes lignes des approches existantes.

3.3.2 Méthodes basées sur la capacité équivalente

Les méthodes basées sur la capacité équivalente (voir 3.2.2) suivent le même principe que la méthode basée sur le débit crête mais utilisent une estimation de la capacité équivalente à réserver. Elles peuvent être statiques en n'utilisant que les paramètres de contrat de trafic déclarés, le débit crête par exemple, ou être dynamiques en mesurant certains paramètres du trafic (moyenne, variance) [Simonian et Brichet, 1999]. Par exemple, si on mesure la moyenne μ du trafic agrégé ainsi que sa variance σ, on approxime la capacité équivalente [Guerin et al.,

1991] $\hat{C}$ des sources déjà connectées, en fonction du taux de perte $\in$ recherché, par :

$$\hat{C}(\mu, \sigma^2, \in) = \mu + \alpha\sigma$$

pour

$$\alpha = \sqrt{2ln\frac{1}{\in} + ln\frac{1}{2\pi}}$$

La condition d'acceptation est alors :

$$\hat{C} + D_{new} \leq C$$

On peut aussi utiliser une approximation d'une borne supérieure de la probabilité de perte, la borne de Chernoff par exemple [Cost, 96, 1, 1996; Simonian et Brichet, 1999] ou encore la borne de Hoeffding [Floyd, 1996].

3.3.3 Méthodes basées sur l'approche bayesienne

Les méthodes basées sur l'approche bayesienne sont des méthodes paramétriques dynamiques. Le paramètre de contrat de trafic supposé connu est le débit crête (D_i) de la source désirant être connectée au réseau. Le modèle de trafic utilisé est le modèle on/off. C'est-à-dire qu'on considère que le trafic correspond à la superposition d'un nombre connu (n) de sources on-off homogènes, tel que pour chaque source on a :

$$D_i(t) = \begin{cases} 0, \text{ si source dans état } 0 \\ 1, \text{ si source dans état } 1 \end{cases} \quad \text{avec} \quad \begin{cases} Pr\{D_i(t) = 0\} = 1 - \beta \\ Pr\{D_i(t) = 1\} = \beta \end{cases}$$

On pose le débit agrégé D_Σ comme étant la somme des débits instantanés des différentes sources :

$$D_\Sigma(n, t) = \sum_{i=1}^{n} D_i(t) \tag{3.3}$$

Le problème principal est d'estimer soit le paramètre β d'activité d'une des sources de la superposition soit la fonction de densité $p(\beta)$ en utilisant l'information provenant de la charge instantanée dans le lien, c'est-à-dire le nombre de sources actives m et ceci grâce à des mesures du trafic en cours.

La probabilité conditionnelle liant m, n et β est :

$$Pr\{D_\Sigma(n, t) = m | n(t) = n, \hat{\beta} = \beta\} = \pi(m|n, \beta = \hat{\beta}) = \begin{pmatrix} n \\ m \end{pmatrix} \hat{\beta}^m (1 - \hat{\beta})^{n-m} \tag{3.4}$$

Ce qui est, pour un paramètre d'activité donné, la probabilité que le nombre de rafales actives soit égal à m quand n appels sont en cours. Si n et m sont accessibles par des mesures on peut calculer l'estimateur de maximum de vraisemblance :

$$\hat{\beta} = \frac{m}{n} \tag{3.5}$$

Le calcul d'un estimateur Bayesien donnant $p(\beta)$, étant donné la supposition du type on-off des sources et la supposition que m et n sont des valeurs mesurées sur le réseau, est :

$$Pr(\hat{\beta}|n,m) = \frac{(n)!}{m!(n-m)!}\hat{\beta}^m(1-\hat{\beta})^{n-m}, \text{ pour tout } \hat{\beta} \in [0,1]. \tag{3.6}$$

La probabilité de perte de cellules CLP quand il y a n appels en traitement peut être exprimée par :

$$CLP = \frac{E[D_\Sigma(n,t) - C]^+}{E[D_\Sigma(n,t)]} \tag{3.7}$$

Comme $D_\Sigma(n,t)$ suit une distribution binomiale cela peut être calculé directement. Un nouvel appel est accepté si la charge actuelle ne dépasse pas une valeur seuil $M(n)$ (précalculée). Cette valeur seuil, qui est à la base de ce CAC, dépend du nombre d'appels en cours et pas explicitement du temps. On voit donc qu'ici on gère un processus au niveau des rafales.

On montre [Bean, 1993b; Bean, 1994a; Bean, 1993a; Griffiths et Key, 1994] qu'en introduisant un précalcul correct des valeurs $M(n)$, il est possible de garder l'objectif des pertes (CLR) en-dessous de la valeur désirée (10^{-9}).

Avec cette méthode, pour accepter un nouvel appel il est nécessaire de calculer le taux de perte de cellule après l'acceptation de l'appel arrivant.

L'approche bayesienne peut être utilisée lorsque les informations sur les paramètres de la source sont incomplètes. Au lieu des valeurs réelles (qu'on ignore) sur les descripteurs de la source, on utilise la "croyance" que l'on a sur leur distribution (donc une présupposition sur leur modèle statistique) ainsi qu'une distribution de probabilité a posteriori sur ces paramètres grâce à des informations actualisées (mesurées) provenant du réseau. Ce CAC est applicable pour des sources homogènes de type on-off et la décision est basée uniquement sur la mesure instantanée du nombre de sources actives. Il compare une approximation d'un des paramètres du modèle du trafic avec une valeur précalculée, il est donc adaptatif au sens des mesures faites en temps réel sur le réseau en accord avec les paramètres de contrat de trafic choisis. Gibbens, Kelly et Key dans [Gibbens et al., 1995] présentent un CAC basé sur cette approche (pour une présentation "rapide" voir [Cost, 96, 1, 1996]) destinée aux sources de type CBR et VBR.

3.3.4 Méthodes basées sur l'approximation gaussienne

L'approximation gaussienne dans le cadre d'un CAC est une méthode paramétrique "dynamique". Les paramètres de trafic considérés comme connus d'une nouvelle source i sont son débit crête (D_i) et/ou son débit moyen (A_i) ainsi que la variance de son débit (σ_i^2).

Les trafics entrants et le trafic agrégé sont considérés comme des processus gaussiens. Il faut noter cependant qu'aucune déclaration de la source (en ATM) ne précise ce caractère gaussien. Cependant, grâce aux résultats du Théorème Central Limite, considérer une distribution gaussienne est intéressant car elle correspond à la superposition d'un grand nombre de sources indépendantes. Par exemple pour la superposition de N sources de type on-off ayant un paramètre d'activité donné β, on aurait pour moyenne et variance du trafic agrégé $\mu = N\beta$ et $\sigma^2 = N\beta(1-\beta)$.

Les paramètres du modèle gaussien à estimer sont donc la moyenne et la variance du trafic actuel ainsi que la moyenne et la variance du trafic du nouvel appel. Si on émet l'hypothèse que les paramètres du trafic en cours (moyenne et variance) sont soit connus, soit mesurés, alors on peut, grâce à la fonction de densité d'une distribution gaussienne, calculer la probabilité d'avoir n sources actives lors de la requête de la nouvelle source. Utilisant cette technique les résultats présentés [Joos et Verbiest, 1989; Rege, 1994; Brichet et al., 1996] permettent de calculer non seulement la probabilité de perte de cellules (CLP), mais aussi le taux de perte de cellules (CLR).

D'autres auteurs [Bean, 1993b; Bean, 1994a; Bean, 1994b] utilisent cette approximation gaussienne en lui ajoutant la technique bootstrap. Cette méthode statistique est utilisée pour réduire le nombre des mesures prises à partir du réseau réel, d'autres techniques de rééchantillonnage peuvent atteindre le même but.

L'augmentation du nombre des mesures par bootstrap ou rééchantillonnage permet de réduire la quantité de données nécessaire à une estimation. Cette technique peut donc être utilisée dans des situations où le coût des équipements de calcul décroît, mais que le coût d'échantillonnage reste stable et élevé. Elle permet aussi de mesurer plus précisément l'intervalle de confiance. Elle donne donc une voie plus sûre pour estimer les paramètres utiles à partir des observations faites sur le trafic et c'est là son intérêt par rapport à l'approche gaussienne citée précédemment.

Enfin toujours dans les méthodes utilisant l'approximation gaussienne on peut citer la proposition d'une approche hybride adaptative qui fut faite par Kröner *et al.* [Kröner et al., 1994]. Elle consiste en un processus de contrôle et en un processus de décision de niveau d'appel. Le processus de contrôle mesure la charge du réseau et la performance résultante en terme de perte de cellules. La période

de contrôle est longue et fixée a priori à 1 seconde.

La condition d'acceptation est :

$$D_\Sigma + D_i < L \tag{3.8}$$

Le processus de contrôle de la valeur de L est basé uniquement sur la perte de cellules évaluée à l'aide de l'approximation gaussienne.

Pour des conditions nominales, s'il n'y a pas de "perte" pendant la période de contrôle, L s'accroît d'une valeur adaptée au temps (TA), s'il y a des pertes elle décroît d'une valeur de réduction de surcharge (OR). Le système mémorise la charge pour laquelle la dernière surcharge est apparue L_{loss} et l'intègre dans une actualisation de OR.

Enfin, le système mesure la charge actuelle si elle est inférieure à une valeur spécifiée, la largeur de bande pour un nouvel appel arrivant est comparée à la capacité libre du système. Cette valeur limite est changée en accord avec l'information sur la surcharge survenue dans le système. On voit ici une méthode adaptative au plein sens du terme, en effet elle est non seulement adaptative au sens des mesures effectuées régulièrement, mais aussi au sens du processus d'acceptation réalisé.

Pour l'utilisation de telles méthodes on doit pouvoir connaître les μ_i et σ_i^2 du nouvel appel. On peut, grâce à un estimateur trouver $\hat{\mu}$ et $\hat{\sigma}^2$ des n sources actuellement en charge du réseau, cependant estimer μ_i et σ_i^2 du nouvel appel pour décider si on doit l'accepter donc avant même que la source n'émette est loin d'être une question triviale, bien que certains aient proposé une approche à ce sujet [Simonian, 1991; Cost, 96, 3, 1996].

3.3.5 Méthodes basées sur l'analyse spectrale

Les méthodes d'analyse spectrale visent à décrire le processus du trafic par une fonction spectrale en utilisant le cadre de la théorie du traitement du signal. L'analyse de Fourier est une approche de filtrage d'entrée et l'analyse d'ondelettes est une généralisation de l'analyse spectrale.

Un des premiers objectifs de l'analyse spectrale est de détecter les aspects déterministes du trafic. On peut trouver des périodicités à des échelles de temps importantes : quotidienne, hebdomadaire, mensuelle et annuelle. L'analyse spectrale permet d'une part d'analyser séparément ces signaux périodiques, et d'autre part d'extraire la partie non périodique du signal, sur laquelle on peut appliquer plus efficacement les techniques de modélisation décrites précédemment.

L'analyse spectrale du trafic permet par ailleurs de mettre en évidence les corrélations temporelles du trafic. Ces corrélations ne sont pas nécessairement strictement périodiques, elles peuvent aussi entrer en jeu uniquement sur des

périodes limitées. Des valeurs élevées du spectre aux hautes fréquences mettent en évidence des corrélations court terme. Elles pourront donc être contrôlées par un mécanisme intervenant aux petites échelles de temps. Au contraire, des valeurs élevées du spectre aux basses fréquences indiquent des corrélations long terme du trafic.

Dans le cadre du contrôle d'admission des connexions, on s'intéresse principalement à la partie basse du spectre. Les méthodes utilisant cette technique sont des méthodes paramétriques dynamiques. Les paramètres de contrat de trafic d'une nouvelle source à estimer sont les basses fréquences du trafic du nouvel appel : $\check{x}_{new}$.

Le processus du trafic en entrée est traité comme une fonction d'entrée ayant une densité spectrale calculable. Celle-ci peut être découpée selon les basses et hautes fréquences avec un impact dominant des basses fréquences sur les performances du réseau.

L'impact d'un trafic $x(t)$ est considéré comme étant l'impact de la somme de ses basses fréquences $\check{x}$ et de ses hautes fréquences $\hat{x}$, la limite entre ces dernières étant fixée par une fréquence notée ω_c. Li [Li et al., 1995] explique comment calculer cette fréquence de coupure.

Les paramètres à estimer seront :

$$\sum_i \check{x}_i \tag{3.9}$$

pour les connexions déjà connectées, et

$$\check{x}_{new} \tag{3.10}$$

pour l'appel arrivant.

Si on connaît la valeur de la capacité C du lien, ainsi que $\sum_i \check{x}_i$, alors la nouvelle connexion est acceptée si :

$$\sum_i \check{x}_i + \check{x}_{new} + \Delta \leq C \tag{3.11}$$

Δ représentant une marge de sécurité additionnelle pour les hautes fréquences du trafic. La valeur de la somme des basses fréquences des connexions en cours peut soit être connue à l'aide de la somme des basses fréquences déclarées, soit mesurée sur le trafic agrégé en cours.

Le but de cette méthode [Li, 1989; Li et Hwang, 1993b; Li et Hwang, 1993a; Li et al., 1995] est de faciliter le contrôle du réseau en utilisant une description simplifiée des processus d'entrée et de sortie pour le multiplexage avec ou sans buffer tout en ayant la possibilité de régler la capacité en sortie.

Bien qu'il soit difficile pour la source de déclarer $\check{x}_i$, il est cependant possible de la relier au SCR bien que la qualification d'un SCR "véridique" ne soit pas encore très précise.

L'inconvénient de l'analyse spectrale pour modéliser le trafic et ensuite réaliser une décision sur la base de celui-ci est que celui-ci est très variable, et jamais réellement stationnaire sur une période suffisante pour que l'on puisse mettre en évidence les raies du spectre. Il est donc préférable d'utiliser une fonction de convolution à support borné pour calculer le spectre à la fois en fréquences ainsi qu'au cours du temps. C'est ce que permet la transformée en ondelettes. Ce type de transformation effectue de plus une analyse multi-résolution, c'est-à-dire que plusieurs échelles de temps sont analysées simultanément (un CAC basé sur cette analyse est proposée dans [Droz, 1996]).

3.3.6 Méthodes basées sur le comportement de buffer

Ces méthodes sont basées sur le comportement de buffers dont la taille est choisie ou évaluée de manière à pouvoir analyser l'influence et/ou le comportement des rafales de cellules, le temps d'inter arrivée des cellules.

La solution de Saito [Saito, 1990a; Saito, 1990b; Saito et Shiomoto, 1991; Saito, 1992; Saito, 1994] est d'implémenter une méthode non paramétrique dynamique. Chaque appel est décrit par son débit moyen μ_i (le nombre moyen de cellules émises) et son débit maximum D_i (le nombre maximal de cellules émises). Pendant une période d'observation r on calcule $p_i(j)$ qui désigne la probabilité que la connexion i émette j cellules. Saito estime pour chaque appel cette probabilité de manière à estimer une ou plusieurs valeurs limites (le maximum du CLR ou du CTD) ou encore la probabilité de blocage des cellules, valeurs qui permettront de réaliser la décision d'acceptation.

Saito prend en compte le processus d'arrivée des cellules pendant une courte période de temps. La probabilité $p_i(j)$ est calculée soit en utilisant les débits moyens et maximums (Méthode I), soit la moyenne et la variance du débit (Méthode II), soit grâce à l'estimation de la distribution des temps d'arrivée des cellules (Méthode III), à chaque fois pendant une période d'estimation fixée r.

Connaissant la probabilité que la connexion i émette j cellules et si le modèle utilise un buffer de taille K, il existe alors une limite au "Cell Transfer Delay" (CTD) appelée T, en considérant que la longueur d'une cellule est L (en bits) :

$$K = \frac{TC}{L} \tag{3.12}$$

Les avantages de cette approche sont de présenter un CAC non paramétrique, l'incorporation de la taille du buffer et d'informations de gestion du contrôle. Elle est effective donc dans le cas de connexions multi-classes et ce indépendamment

de la classification du service (bien que prévue pour les services de type CBR et VBR). Malheureusement les valeurs du CLR préconisés par les organismes de normalisation sont très faibles (de l'ordre de 10^{-9}). La simulation ne permet pas de rendre compte de phénomènes aussi rares. Aussi la méthode consistant à observer le CLR dans plusieurs files d'attentes virtuelles, associées à une file d'attente de sortie donnée, dont les taux de service (débit entrant) sont supérieurs aux taux réels, a vu le jour. On nomme habituellement ce type de méthode "virtual output buffer" (VOB).

L'idée est de faire comme si le réseau avait un débit inférieur à sa valeur réelle. Au lieu d'essayer de prédire le CLR pour $(N+1)$ connexions et par extension pour $N(1+n)$ connexions avec une capacité C et un buffer de taille L on prédit le CLR pour une capacité $\frac{C}{1+n}$ avec N connexions et un buffer de taille K.

Pour calculer la probabilité de perte de cellules avec un tel système, il est possible d'utiliser un système virtuel qui lui traite le comportement de trois buffers virtuels ayant de plus petites capacités. Cette méthode basée sur le contrôle du comportement d'un buffer virtuel a été proposée par Courcoubetis [Courcoubetis et al., 95] et est basée sur une estimation on-line de la probabilité de perte de cellules dans les liens ATM. Si on appelle CLP_i les CLP des trois buffers et CLP celui du buffer "normal" on a : $log(CLP) = l_0 log(CLP_0) + l_1 log(CLP_1) + l_2 log(CLP_2)$ (l_0,l_1,l_2 étant déterminés à l'aide de trois observations pour différentes valeurs de CLP_0, CLP_1, CLP_2). Finalement le taux de perte de cellules pour un trafic s'accroissant peut être calculé et cette valeur est à la base du CAC. C'est-à-dire que si $CLP(n+1) > B$ (une valeur seuil) le nouvel appel n'est pas accepté.

Cette méthode n'utilise pas de modèle de trafic et est résolue de manière analytique. Dans les CAC utilisant la technique des VOB on peut aussi citer les travaux de Hiramatsu qui après avoir présenté différentes applications des réseaux de neurones artificiels à l'ATM [Hiramatsu, 1989; Hiramatsu, 1990] les a appliqués à l'aide de la technique des VOB [Hiramatsu, 1991; Hiramatsu, 1994]. Là encore, l'approche est purement non paramétrique en ce sens qu'on ne présuppose aucun modèle de trafic. C'est le réseau de neurones qui construit ce "modèle" au cours de son apprentissage sur une base de données suffisamment étendue.

Dans le cadre des VOB un réseau de neurones est associé à chaque buffer virtuel et délivre une QoS. Chaque QoS est utilisée dans l'élaboration d'une table d'exemples afin qu'un quatrième réseau de neurones juge s'il doit accepter ou rejeter le nouvel appel (voir figure 3.7). Les réseaux de neurones sont "entraînés" à l'aide d'un algorithme dit de rétro-propagation. La fonction f que l'on désire approximer à l'aide du réseau de neurones peut être : la minimisation du maximum du taux de perte (CLR), la minimisation de la perte moyenne des appels ou la maximisation de l'utilisation du lien [Hiramatsu, 1991; Hiramatsu, 1994].

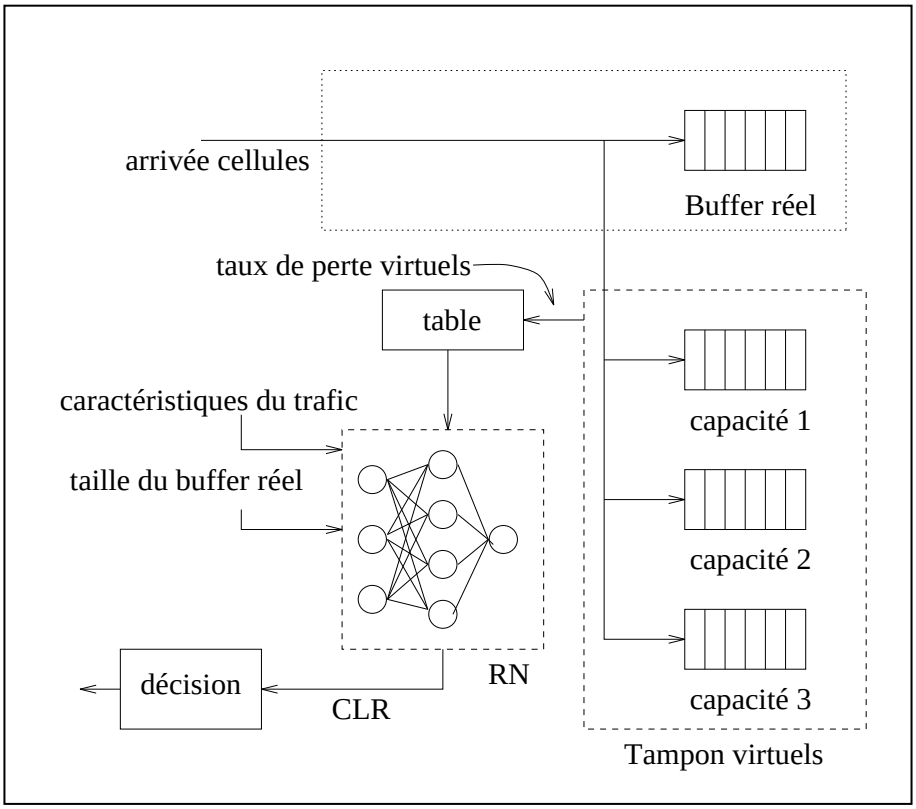

FIG. 3.7 - Utilisation de tampons virtuels pour le contrôle d'admission des connexions.

3.3.7 Discussion

Le problème de l'utilisation de mesures, de paramètres ou d'estimation pour l'incorporation d'un CAC dans un réseau ATM est loin d'être un chapitre clos du fait qu'aucune des méthodes proposées dans cet état de l'art ne s'est encore véritablement imposée. Les comparer n'est pas chose aisée car, mis à part de rares cas, les "simulations" manquent. On peut ici préciser que toutes les méthodes qui caractérisent correctement le trafic en cours peuvent être utilisées dans le cadre d'une procédure de contrôle d'admission des connexions.

La majorité des CAC, voire tous, reposent sur une hypothèse de stationnarité des trafics en ce sens qu'elles présupposent qu'il existe une "structure" du trafic qui reste valide sur de grandes échelles de temps. C'est apparemment le moins que l'on doive supposer pour tenter des études de prédiction, ou du moins faire la supposition d'un trafic localement stationnaire [Vaton, 1998]. De même, ces techniques reposent sur la disponibilité de données de trafic importantes.

Les méthodes paramétriques s'élaborent en trois temps :

– élaboration (ou choix) d'un modèle afin de représenter de manière générale

les trafics qu'on aura à traiter dans un nœud ATM. Le modèle est composé d'un ensemble de paramètres caractérisant le comportement du trafic ;

– développer une technique d'identification des paramètres du modèle à partir des observations faites sur le trafic ;

– développer une technique permettant de décrire la probabilité de congestion en fonction des paramètres de contrat de trafic de l'appel arrivant et du modèle choisi.

Il est évident que le choix du modèle est une étape cruciale pour ces techniques. Ce dernier doit être suffisamment "souple" pour pouvoir représenter une grande variété de trafics. A moins que l'acceptation d'appel que l'on désire réaliser ne s'applique à une seule variété de trafic. Il doit aussi bien se prêter à l'identification des paramètres. Il semble donc que les méthodes non paramétriques adaptatives soient les plus adaptées si on veut que le CAC utilisé recouvre un grand espace de décision. En effet elles ne souffrent pas de l'utilisation d'un modèle de trafic qui pourrait être inadapté au trafic sur lequel on l'exerce.

De plus les paramètres de trafic utilisés pour décrire les flux de trafic, et donc les paramètres de contrat de trafic, constituent une maigre description des caractéristiques statistiques des flux. Par conséquent, réaliser une politique d'allocation de ressources et/ou de contrôle d'admission des connexions uniquement basée sur leurs valeurs peut induire une sous-utilisation du réseau. Il semble plus efficace d'estimer les caractéristiques du trafic en temps réel et de réaliser un contrôle d'admission basé sur des mesures [Gibbens et Kelly, 1997; Floyd, 1996].

C'est pourquoi de nouvelles techniques, utilisant les réseaux de neurones artificiels basés sur la théorie de l'apprentissage statistique, ont vu le jour. Ces approches connexionnistes de la caractérisation du trafic en général et du contrôle d'admission en particulier apportent en effet de nombreux avantages par rapport aux méthodes classiques. Elles s'adaptent facilement aux changements du trafic sur lequel aucune hypothèse préalable n'est nécessaire. Elles possèdent un haut niveau de fiabilité et de tolérance aux fautes, permettent de généraliser à des données inconnues les résultats appris (si l'espace d'apprentissage a été judicieusement choisi et est suffisamment étendu).

Chapitre 4

Les réseaux de neurones

Les développements récents de la théorie des réseaux neuronaux et de leurs applications pratiques en font une approche naturelle pour résoudre de nombreux problèmes non linéaires dans les télécommunications, tel par exemple le problème du contrôle d'admission des connexions (CAC) évoqué au chapitre précédent. Ils ont d'ailleurs déjà été maintes fois utilisés dans le champ des télécommunications. On peut citer l'estimation de la courbe d'arrivée du trafic [Clérot et al., 1997], la prédiction de la taille maximum d'une file d'attente [Bengio et al., 1996], la prédiction du taux de perte [Nördstrom et al., 1995], ou encore la modélisation d'une file d'attente [Aussem et al., 1999]. Nous présentons en détails dans la première section ce qu'est un réseau de neurones artificiels.

Pour l'apprentissage supervisé d'un perceptron multicouche par correction d'erreur, l'algorithme le plus utilisé est l'algorithme de descente de gradient. Le calcul du gradient se fait en utilisant l'algorithme de la rétro-propagation de l'erreur. L'algorithme d'apprentissage utilisant ce procédé reste encore aujourd'hui la méthode d'apprentissage la plus largement utilisée et nous la détaillons au cours de la deuxième section.

Dans un processus d'apprentissage le réseau de neurones est construit en minimisant, par exemple, une fonction de coût sur un ensemble fini d'exemples, l'ensemble d'apprentissage. Cependant, le plus important est la faculté de généraliser la représentation construite par le réseau à toutes les données, y compris celles n'appartenant pas à l'ensemble d'apprentissage. Nous discutons de ce point au cours de la troisième section.

On présente, au cours de la quatrième section, une nouvelle méthode destinée à améliorer les performances en généralisation des perceptrons multicouches utilisés en tant que réseaux discriminants et approximateurs de fonctions. On montre clairement la modification du critère d'apprentissage qui permet de contrôler la forme de la distribution des erreurs au cours de l'apprentissage. Cette méthode permet de minimiser à la fois les erreurs de classification et les erreurs d'estima-

tion par une minimisation de la variance de l'erreur quadratique. Des résultats améliorant notablement l'état de l'art sur trois problèmes sont présentés pour valider la méthode. Celle-ci sera ensuite exploitée au cours du chapitre 5.

Les différents travaux cités dans cette section ont fait l'objet de publications [Lemaire et al., 2000; Lemaire et al., 1999a; Bernier et al., 1998a; Bernier et al., 1998b].

4.1 Présentation

4.1.1 Qu'est-ce qu'un réseau de neurones ?

En tout premier lieu, lorsque nous parlons de réseaux de neurones, nous devrions plutôt dire *"réseaux de neurones artificiels"*. En effet, les réseaux de neurones biologiques [Changeux, 1983] sont de loin plus complexes que les modèles mathématiques utilisés pour les réseaux de neurones artificiels que nous utilisons. Cependant, il est usuel d'oublier le mot *artificiels* et de dire "réseaux de neurones". Il n'y a pas de définition universelle pour caractériser un réseau de neurones. Mais la plupart de leurs utilisateurs ou concepteurs s'accordent sur le fait qu'un réseau de neurones est un réseau d'unités élémentaires interconnectées à fonctions d'activation linéaires ou non-linéaires. Ces unités se décomposent (pour les réseaux multicouches) en au moins deux sous-ensembles de neurones (figure 4.1) : un sous-ensemble de neurones d'entrée, un autre de neurones de sortie et éventuellement un ensemble de neurones cachés.

Il existe de nombreux modèles de réseaux de neurones : les réseaux de Hopfield, les réseaux de Kohonen, les réseaux à fonctions de bases radiales, les perceptrons multicouches (MLP pour Multi Layer Perceptron) ayant des architectures allant de la plus simple à la plus complexe (les différentes unités sont interconnectées aux autres, soit complètement, soit partiellement). Dans cette section nous définirons brièvement le perceptron multicouche. Pour obtenir une description plus détaillée, le lecteur pourra se rapporter par exemple à [Le Cun, 1987; Fessant, 1995].

L'ensemble des liens, convergeant vers une unité, constitue les connexions entrantes de l'unité. Ceux qui divergent vers d'autres unités constituent les connexions sortantes. A chaque *connexion*, on associe un *poids* ou efficacité *synaptique*. Le poids d'une connexion représente la force de l'influence d'une unité sur celle à laquelle est reliée sa sortie. En fait, la *"connaissance"* incluse dans le réseau de neurones est "mémorisée" dans les poids et l'architecture du réseau.

Le perceptron multicouche est structuré en couches. Chaque couche est entièrement connectée à la suivante et son graphe de connectivité ne possède pas de cycle (figure 4.1).

Nous parlerons toujours de réseau de neurones qui fonctionne en temps discret

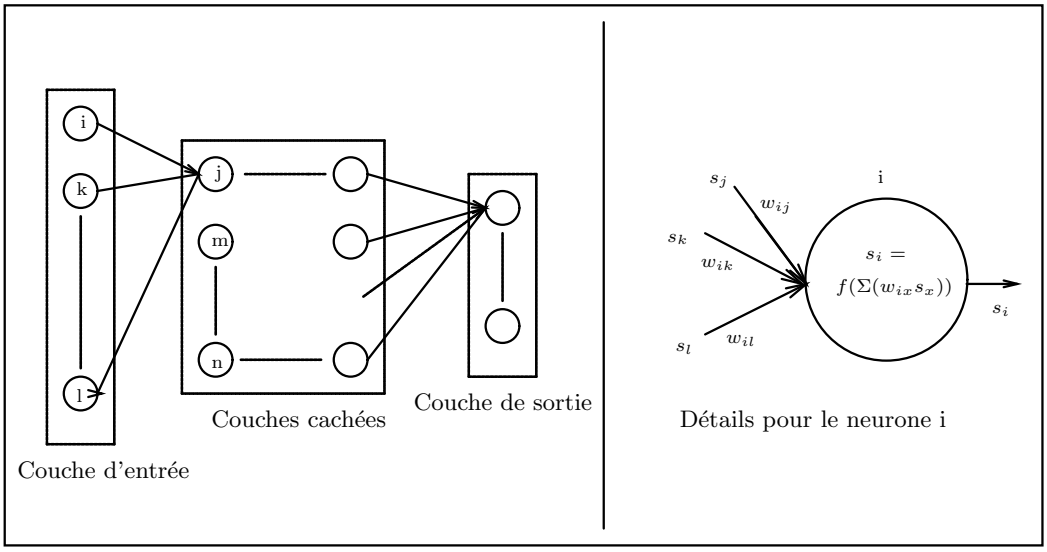

FIG. 4.1 - Un exemple de perceptron multicouche

et où chaque pas de temps est appelé cycle. Un cycle comprend tout le processus qu'effectue le réseau pour traiter une donnée (un motif d'entrée). Le fonctionnement du réseau est donc cadencé par la présentation des motifs en entrée. Un vecteur de scalaires présenté au temps t à toutes les unités d'entrées sera appelé exemple. A cet exemple sont aussi associées les valeurs (le vecteur de sortie) que l'on désire apprendre. Les poids des connexions sont éventuellement modifiés d'un cycle à l'autre par un mécanisme d'apprentissage.

Fixons à présent les notations que nous utiliserons dans la suite de ce chapitre. On appellera, au cycle t, w_{ij}^t le poids de la connexion reliant la sortie s_j d'un neurone j à l'entrée du neurone i. Le calcul de la sortie des unités à partir de leurs entrées s'appelle la règle de transition. Les règles de transition sont fondées principalement sur le modèle de Mc Culloch et Pitts [Mc Culloch, D. W. et Pitts, W., 1943]. Cette transition peut être décomposée en deux étapes : calcul de l'activation présente à l'entrée de l'unité puis calcul de la sortie de l'unité suivant la fonction d'activation qu'elle possède.

D'une façon plus générale on peut définir un neurone par les quatre éléments suivants [Rumelhart, D. E. et al., 1986] :

1. La nature de ses entrées : elles peuvent être binaires (0,1), ou prendre une valeur réelle.

2. La fonction d'entrée totale $a(.)$, qui définit le pré-traitement effectué sur les

entrées. Elle peut être affine, polynomiale, booléenne ou linéaire. Dans ce dernier cas elle s'écrit généralement comme la somme pondérée des entrées :

$$a_i = \sum_{p=1}^{n} w_{ip}^t x_{ip} \tag{4.1}$$

où a_i est l'activité du neurone i à l'instant t, x_{ip} l'entrée p du neurone i, w_{ip}^t le poids de la connexion de l'entrée x_{ip}, n le nombre d'entrées du neurone. L'entrée totale ne dépend donc que des entrées extérieures, des poids et des sorties des neurones.

3. La fonction d'activation $f(.)$ du neurone ou fonction de transfert qui définit son état de sortie en fonction de son entrée totale a, est telle que $s_i = f(a_i)$. Cette fonction peut être une fonction binaire à seuil (fonction de Heaviside ou fonction signe), une fonction sigmoïde, une fonction stochastique, ou toute autre fonction généralement choisie croissante et impaire. L'utilisation de telles fonctions permet d'introduire une non-linéarité dans le réseau de neurones. Toutes les fonctions non-linéaires, par définition, atteignent ce but et les plus utilisées sont les fonctions sigmoïdes et gaussiennes. Pour les neurones de sortie, il faut choisir une fonction d'activation adaptée à la distribution des valeurs cibles. Mais si les valeurs n'ont pas de bornes connues il est préférable d'utiliser une fonction non bornée, comme par exemple la fonction identité (si les valeurs peuvent être positives ou négatives) ou l'exponentielle (si elles sont exclusivement du même signe) ; on peut se référer à [McCullagh et Nelder, 1989] pour plus de détails).

   Par la suite la sortie s_i va être redirigée soit vers l'environnement extérieur, soit vers d'autres unités où elle contribuera au calcul de leurs états d'activation.

4. Enfin, la nature de ses sorties : comme les entrées, elles peuvent être binaires ou réelles.

Pour terminer la définition du perceptron multicouche, il nous reste à ajouter deux éléments qui lui permettent d'apprendre des données : une fonction de coût ou de perte C et un algorithme d'apprentissage.

Dans ce mémoire, nous utiliserons des entrées et des sorties à valeurs réelles, les activations seront calculées à l'aide d'une somme pondérée linéaire et les fonctions d'activation seront sigmoïdales.

4.1.2 Que peut-on faire à l'aide d'un réseau de neurones ?

En principe, les réseaux de neurones sont des approximateurs universels [White, H. et Hornik, K., 1989], c'est-à-dire qu'ils peuvent approximer n'importe quelle fonction. Cependant, bien qu'ils puissent en théorie approximer n'importe quelle transformation continue (voir [Hornik et al., 1989]) d'un espace à dimension finie vers un autre espace à dimension finie (s'ils possèdent suffisamment de neurones cachés) aussi finement qu'on le souhaite, si on ne possède pas assez d'exemples, voire de puissance de calcul, le problème peut ne pas avoir de solution acceptable (au regard d'autres méthodes). De plus, les résultats théoriques présentés dans [Hornik et al., 1989] ne fournissent aucun indice sur la méthode à utiliser pour trouver les poids correspondant à l'approximation d'une fonction recherchée, ni aucune indication sur le nombre de neurones cachés.

Suivant les architectures et les fonctions d'activation utilisées, le perceptron multicouche peut instancier beaucoup de modèles statistiques bien connus, comme la régression linéaire, la régression linéaire multi-variée, l'analyse discriminante, la régression polynomiale, l'analyse en composante principale [Sarle, 1994]. De plus, le perceptron multicouche, utilisant des fonctions d'activation non-linéaires et au moins une couche cachée, permet d'instancier des fonctions non-linéaires. Ces deux propriétés en font une *machine à apprendre* performante.

4.2 Apprentissage

4.2.1 Différentes règles d'apprentissage

L'apprentissage est le processus d'adaptation des paramètres d'un système, ici un réseau de neurones, pour donner une réponse désirée à une entrée. Le choix du processus d'adaptation des paramètres, c'est-à-dire la procédure d'apprentissage, est conditionné très fortement par la tâche que le réseau de neurones doit réaliser. On peut citer :

1. **approximation** : supposons que nous ayons une application non-linéaire qui lie à une entrée une sortie de la forme :

$$d = g(\mathbf{x}) \tag{4.2}$$

où $\mathbf{x}$ est le vecteur d'entrée et d le scalaire de sortie. La fonction $g(\cdot)$ est inconnue. On veut construire un réseau qui approxime la fonction $g(\cdot)$ non-linéaire grâce à un ensemble de couples d'exemples entrée-sortie $(\mathbf{x}_1, d_1)$, $(\mathbf{x}_2, d_2)$... $(\mathbf{x}_P, d_P)$.

2. **classification** : Le problème consiste à ranger des entrées données dans un certain nombre de classes. La discrimination est une méthode de classification pour laquelle le but est la construction d'un classifieur qui à chaque observation $\mathbf{x}$ associe une classe parmi un ensemble fixé de classes. Les réseaux de neurones sont capables de construire de façon non-paramétrique des frontières de décision séparant ces classes : ils offrent donc la possibilité de résoudre des problèmes de classification extrêmement complexes. Par exemple, on peut construire un classifieur en associant une classe à chaque sortie d'un perceptron multicouche à q sorties (q représentant le nombre de classes).

3. **prédiction**

   On possède, au temps t, des échantillons passés d'un signal :

   $$x(t-1), x(t-2), \cdots, x(t-M) \tag{4.3}$$

   et on veut prévoir $x(t)$. On peut le faire par exemple par correction d'erreur de manière supervisée en imposant $x(t)$ comme sortie désirée.

Toutes ces tâches consistent fondamentalement à construire une application à partir d'un certain nombre d'exemples de cette application. Il existe de nombreuses méthodes d'apprentissage qui peuvent être utilisées pour la réalisation de ces tâches, parmi lesquelles on citera notamment :

1. L'**apprentissage par correction d'erreur** a pour but final de minimiser une fonction de coût C construite à partir de la base d'exemples qu'on possède. Une fois que la fonction de coût a été choisie l'apprentissage devient un problème d'optimisation pour lequel il existe quantité de techniques. Une méthode connue de minimisation est la descente de gradient [Boucheron, 1992].

2. L'**apprentissage de Hebb** qui est en fait un postulat tiré de la biologie et est la plus connue et la plus vieille des règles d'apprentissage [Hebb, 1949].

3. L'**apprentissage compétitif** ou les neurones de sortie du réseau "décident" lequel d'entre eux sera actif [Kohonen, 1984]. Il n'existe donc qu'un seul neurone de sortie actif à chaque cycle.

4. L'**apprentissage de Boltzmann** qui repose sur un algorithme stochastique issu de la thermodynamique et de la théorie de l'information (car on va chercher un maximum de vraisemblance) [Mazaika, 1987].

On peut aussi distinguer deux classes d'apprentissage :

1. L'**apprentissage non-supervisé** où il n'y a pas de maître pour contrôler de l'extérieur le déroulement de l'apprentissage ; c'est pourquoi on l'appelle quelquefois auto-apprentissage. Le réseau va essayer de s'adapter aux régularités statistiques des données d'entrée. Il va donc automatiquement coder des classes dans sa représentation interne. L'apprentissage non-supervisé est réalisé, par exemple, par la règle de l'apprentissage compétitif vue plus haut. Il s'agit alors d'une stratégie appelée **winner-take-all**.

2. L'**apprentissage supervisé** : La méthode classique pour l'apprentissage supervisé consiste à se procurer un ensemble d'exemples, c'est-à-dire un ensemble fini de couple de vecteurs ($\mathbf{x_i}$,$\mathbf{y_i}$). Dans un tel couple, $\mathbf{x_i}$ désigne l'entrée du réseau et $\mathbf{y_i}$ la sortie désirée pour cette entrée. On écrit alors la fonction calculée par le réseau sous une forme paramétrique : $f(\mathbf{x},\mathbf{w})$ désigne la sortie du réseau quand on lui présente en entrée le vecteur $\mathbf{x}$ et qu'il utilise les poids synaptiques contenus dans la matrice $\mathbf{w}$. Enfin on se donne une distance sur l'espace vectoriel de sortie, c'est-à-dire un moyen de mesurer l'erreur commise en un point par le réseau. Si cette distance est notée d, on cherche alors à trouver la valeur de $\mathbf{w}$ qui minimise l'erreur totale commise par le réseau, c'est-à-dire la somme des distances entre les sorties obtenues et les sorties désirées, c'est-à-dire la somme des $d(f(\mathbf{x_i}, \mathbf{w}), \mathbf{y_i})$. Cette erreur est une fonction de $\mathbf{w}$ et on peut utiliser les techniques classiques d'optimisation de fonctions pour trouver son minimum.

4.2.2 La rétropropagation de l'erreur

Pour l'apprentissage supervisé d'un perceptron multicouche, par correction d'erreur, l'algorithme le plus utilisé est l'algorithme de descente de gradient. Le calcul du gradient se fait en utilisant l'algorithme de la rétro-propagation de l'erreur. L'algorithme d'apprentissage utilisant ce procédé a été découvert par [Rumelhart, D. E. et al., 1986; Le Cun, 1987] et reste encore aujourd'hui la méthode d'apprentissage la plus largement utilisée.

Les algorithmes d'optimisation de fonction efficaces utilisent en général la différentielle de la fonction considérée (c'est-à-dire son gradient quand elle est à valeurs réelles). Quand les fonctions de transfert utilisées dans les neurones et la fonction distance sont différentiables, alors l'erreur commise par un MLP est une fonction différentiable des coefficients synaptiques du réseau de neurones. L'algorithme de rétro-propagation permet justement de calculer le gradient de cette erreur de façon efficace : le nombre d'opérations (multiplications et additions) à faire est en effet proportionnel au nombre de connexions du réseau, comme dans le cas du calcul de la sortie de celui-ci. Cet algorithme rend ainsi possible l'ap-

prentissage d'un MLP et permet d'apporter une réponse à l'apprentissage qui devient complexe dans les réseaux multi-couches.

Un algorithme de descente de gradient repose sur une fonction de coût C, que l'on doit minimiser au cours d'une session d'apprentissage. Lors de cette dernière les couples de vecteurs d'entrée et de sorties désirées ($\mathbf{x_i}$, $\mathbf{d_i}$) sont présentés séquentiellement au réseau au cours d'un cycle. A chaque neurone de sortie du réseau, i, on associe une valeur de sortie désirée d_i. Posons ici la fonction de coût quadratique C, comme étant :

$$C = \sum_{x \in P} \sum_{i \in O} (d_i^x - s_i^x)^2, \tag{4.4}$$

où P est l'ensemble des exemples d'apprentissage, O est l'ensemble des cellules de sortie, s_i^x est la valeur du neurone de sortie i après la présentation de l'exemple x et d_i^x est la valeur désirée pour le neurone correspondant. Il est à noter que cette fonction de coût quadratique n'est pas la seule possible mais que toute fonction dérivable en s et d peut être utilisée.

La modification des poids du réseau de neurones est réalisée à l'aide d'un algorithme de gradient qui est de la forme :

$$\mathbf{W}^{t+1} = \mathbf{W}^t - \alpha \nabla C(\mathbf{W}^t) \tag{4.5}$$

où la matrice $\mathbf{W}$ représente les poids du réseau, ∇ représente le gradient de la fonction de coût par rapport aux poids $\mathbf{W}$ et α représente un coefficient de modification des poids, appelé pas d'apprentissage.

Il existe deux méthodes principales de modification des poids du réseau liées à la manière de calculer le gradient, soit en utilisant un gradient total :

$$C = \sum_{x \in P} \sum_{i \in O} (d_i^x - s_i^x)^2 \tag{4.6}$$

qui est une méthode globale encore appelée "batch", soit en utilisant un gradient partiel :

$$C^x = \sum_{i \in O} (d_i^x - s_i^x)^2 \tag{4.7}$$

qui est appelée méthode stochastique. Bottou présente dans [Bottou, 1991] une comparaison des deux méthodes et il montre que la méthode stochastique est plus "rapide" que la méthode globale. Les propriétés de convergence de la rétro-propagation "standard" (telle que proposée dans [Rumelhart, D. E. et al., 1986; Le Cun, 1987]), en version stochastique et en version "batch" sont discutées

dans [Bertsekas et Tsitsiklis, 1996].

Dans le cas stochastique, l'algorithme de rétro-propagation du gradient de l'erreur se décompose en trois phases :

– Présentation d'un vecteur d'entrée, $\mathbf{x_i}$, aux neurones de la couche d'entrée puis calcul des sorties de tous les neurones du réseau de couche en couche jusqu'à obtenir les sorties des neurones de la couche de sortie. Sachant que la sortie d'un neurone i, quel qu'il soit, est :

$$s_i = f(a_i) = f(\sum_{j=0}^{n}(w_{ij}^t s_j)) \tag{4.8}$$

où a_i est l'activité présente à l'entrée du neurone, w_{ij}^t la connexion reliant le neurone i à un neurone j de la couche précédente[1], s_j la sortie du neurone j de la couche précédente, f la fonction d'activation du neurone, t le numéro du cycle d'apprentissage et n le nombre d'entrées du neurone.

– Présentation du vecteur de sortie $\mathbf{d_i}$ associé à $\mathbf{x_i}$ sur les neurones de la couche de sortie de manière à calculer l'erreur commise par le réseau.

– Application de la procédure du calcul de gradient qui permet de modifier les poids du réseau en fonction de l'erreur commise (l'algorithme de rétro-propagation lui-même). Cependant, plutôt que de calculer les gradients par rapport aux poids, on préfère calculer les gradients par rapport à la valeur de l'activation de chaque neurone a_i. En effet, elles sont en nombre plus faible et permettent de retrouver les gradients par rapport aux poids de la façon suivante :

$$\frac{\partial C^x}{\partial w_{ij}} = \frac{\partial C^x}{\partial a_i^x}\frac{\partial a_i^x}{\partial w_{ij}} = \frac{\partial C^x}{\partial a_i} s_j^x \tag{4.9}$$

On utilise dans la suite pour le gradient la notation :

$$G_i^x = \frac{\partial C^x}{\partial a_i^x} \tag{4.10}$$

Le calcul qui suit diffère selon que le neurone concerné appartient à une couche cachée ou à la couche de sortie :

– Pour un neurone i de la couche de sortie, O, et un exemple x :

[1]Rappelons que l'on se place dans le cas du perceptron multicouche sans connexion récurrente.

$$G_i^x = \frac{\partial}{\partial a_i^x} \sum_{k \in O} (d_k^x - s_k^x)^2 \tag{4.11}$$

or

$$\frac{\partial}{\partial a_i^x} \sum_{p \in O} (d_k^x - s_k^x)^2 = 0 \text{ pour } p \neq i \tag{4.12}$$

alors

$$\frac{\partial C^x}{\partial w_{ij}} = -2(d_i^x - s_i^x) f^{'}(a_i^x) s_j^x \tag{4.13}$$

– Pour un neurone i d'une couche cachée, H, suivi de la couche O ($k \in O$) et un exemple x.

$$G_i^x = \frac{\partial C^x}{\partial a_i^x} = \sum_{k=1}^{n} \frac{\partial C^x}{\partial a_k^x} \frac{\partial a_k^x}{\partial a_i^x} = \sum_{k=1}^{n} G_k^x \frac{\partial a_k^x}{\partial a_i^x} \tag{4.14}$$

donc

$$G_i^x = \sum_{k=1}^{n} G_k^x \frac{\partial a_k^x}{\partial s_i^x} \frac{\partial s_i^x}{\partial a_i^x} \tag{4.15}$$

alors

$$G_i^x = f^{'}(a_i^x) \sum_{k=1}^{n} G_k^x w_{ki} \tag{4.16}$$

La règle de modification des poids, quel que soit le poids concerné est alors :

$$w_{ij}^{t+1} = w_{ij}^t - \alpha G_i^x s_j^x \tag{4.17}$$

avec i appartenant à la couche N, j appartenant à la couche suivante dans le sens de la propagation, x l'exemple présenté et α un nombre réel positif, de faible valeur, qui représente le pas de déplacement en direction de la pente maximum. On peut résumer l'algorithme de rétropropagation à l'aide du schéma de la figure 4.2 où le sens du mot rétroprogation est bien perceptible.

La fonction de coût utilisée dans l'algorithme présenté ci-dessus possède un certain nombre de propriétés. Elle possède des minima locaux car le minimum global n'est pas forcément unique et elle peut être parsemée de plateaux qui rendent la convergence dans ces régions lente. Le choix du pas d'apprentissage est alors difficile. En effet, dans la rétro-propagation "standard" à pas fixe, avec un

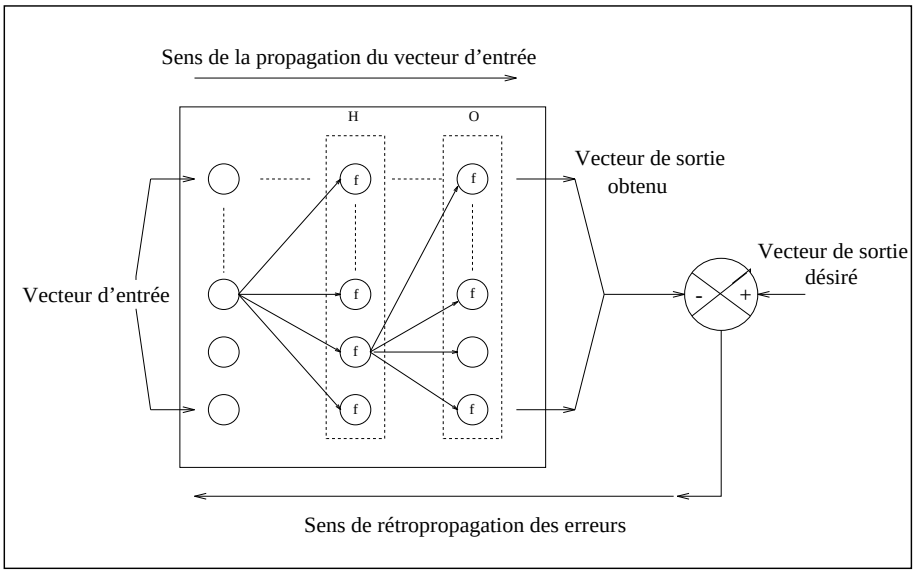

FIG. 4.2 - Représentation schématique de la rétropropagation de l'erreur.

pas d'apprentissage trop petit, le réseau de neurones apprend très lentement, a contrario avec un pas d'apprentissage trop grand les poids du réseau de neurones et la fonction d'erreur divergent. Par conséquent, trouver un "bon" pas d'apprentissage n'est pas chose aisée. Depuis 1986-1987, de nombreux chercheurs ont travaillé à l'amélioration de l'algorithme présenté ci-dessus. Une des méthodes permettant d'accélérer la convergence dans les plateaux consiste à rajouter dans l'équation de mise à jour des poids un terme de moment (ou inertie) [Plaut et al., 1986]. Dans ce cas la règle de modification des poids devient :

$$w_{ij}^{t+1} = w_{ij}^{t} - \alpha G_i^x s_j^x + \beta \Delta w_{ij}^{t} \tag{4.18}$$

On peut citer aussi d'autres travaux destinés à améliorer la vitesse de convergence de l'algorithme de rétro-propagation [Falhman, 1988; Riedmiller et Braun, 1993]. Le lecteur pourra aussi considérer d'autres méthodes de minimisation de l'erreur [Jacobs, 1988; Becker et Le Cun, 1988].

4.3 Généralisation

4.3.1 Capacité et généralisation

- *Généraliser c'est étendre à toute une classe ce qui a été observé sur un nombre limité d'éléments ou d'individus appartenant à cette classe. Dictionnaire Hachette.*

Dans un processus d'apprentissage le réseau de neurones est construit en minimisant, par exemple, une fonction de coût sur un ensemble fini d'exemples, l'ensemble d'apprentissage. Cependant, le plus important est la faculté de généraliser la représentation construite par le réseau à toutes les données, y compris celles n'appartenant pas à l'ensemble d'apprentissage. Une manière d'évaluer cette faculté consiste à mesurer les performances du réseau de neurones sur des données représentatives du problème non apprises. Il s'agit d'une évaluation de l'erreur de généralisation. La différence entre l'erreur d'apprentissage et l'erreur de généralisation représente une mesure de la qualité de l'apprentissage effectué [Vapnik, 1982; Vapnik, 1995; Boucheron, 1992].

L'erreur de généralisation dépend avant tout de trois paramètres : le nombre d'exemples utilisés pour l'apprentissage, la complexité du problème sous-jacent et l'architecture du réseau.

Les approches statistiques de la généralisation [Vapnik et Chervonenkis, 1971; Vapnik, 1982] sont un des domaines d'investigation majeur pour optimiser les performances de l'apprentissage des réseaux de neurones. Ce domaine est devenu, avec le temps, très riche et complexe aussi le lecteur est invité à considérer par exemple l'article de synthèse de Wolpert [Wolpert, 1992]. On constate qu'une amélioration de la généralisation peut être vue sous deux aspects imbriqués :

- Si la taille du réseau est fixée, quel est l'ensemble d'exemples d'apprentissage qui donnera la meilleure généralisation ?
- Si le nombre des exemples est fixé, comment choisir le réseau pour avoir la meilleure généralisation ?

Il est difficile de répondre à chacune de ces questions de manière individuelle tant la réponse à l'une nécessite de se pencher sur l'autre. Il est nécessaire ici d'introduire la notion de *capacité* : si on considère le réseau de neurones comme un système permettant de choisir une fonction parmi un ensemble déterminé par la structure du réseau, alors la capacité du système représente le nombre d'exemples que le réseau peut apprendre correctement à tout coup. Plus un système peut approximer de fonctions différentes, plus sa capacité est élevée en général plus le nombre de poids est élevé, plus la capacité augmente (figure 4.3).

Une mesure de la capacité du système est la dimension de Vapnik Chervonenkis ou VC dim [Vapnik et Chervonenkis, 1971]. Cette capacité est liée à la généralisation comme le montrent les résultats de Vapnik et Chervonenkis. Pour un nombre d'exemples fixé N, si on commence l'apprentissage avec une VC dim minimale que l'on augmente progressivement (par exemple en augmentant le nombre de connexions), l'erreur de généralisation décroît jusqu'à une valeur critique de la VC dim. Une fois ce point passé, augmenter la VC dim aura pour effet d'augmenter l'erreur de généralisation (l'erreur de généralisation est l'erreur

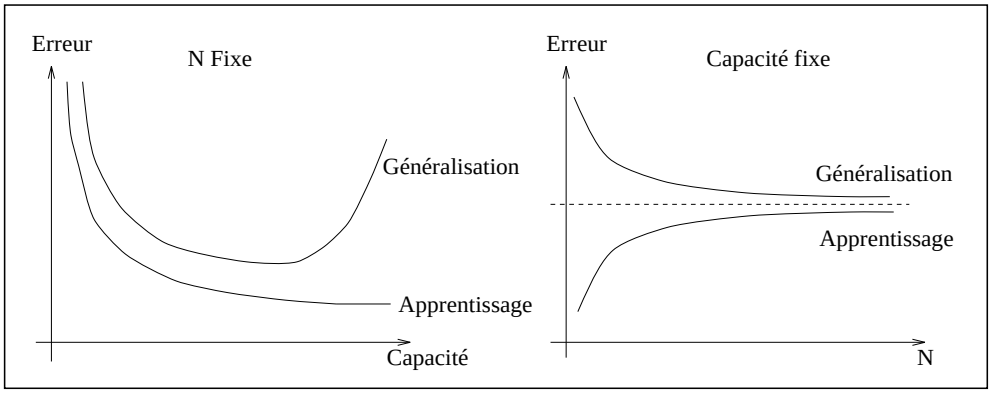

FIG. 4.3 - Relation entre capacité et généralisation des réseaux de neurones artificiels.

sur de nouveaux exemples pris en dehors du premier ensemble). De plus, pour une VC dim fixée, augmenter le nombre d'exemples n'améliorera la généralisation que jusqu'à une valeur asymptotique qui dépend de la VC dim.

4.3.2 Généralisation et critère d'arrêt pour l'apprentissage

L'une des méthodes qui permet de contrôler la capacité d'un réseau de neurones consiste à arrêter l'apprentissage "à temps". De ce fait différents critères permettant de décider quand stopper l'algorithme d'apprentissage ont été développé :

- quand l'erreur d'apprentissage a atteint un seuil fixé ;
- après un nombre fixé de cycles d'apprentissage ;
- quand une estimation de l'erreur de généralisation est minimum.

Les méthodes qui évaluent l'erreur de généralisation sont presque toutes basées sur la partition de l'ensemble des données qu'on possède en plusieurs sous-ensembles. Par exemple un ensemble utilisé pour l'apprentissage et un ensemble de validation. L'ensemble de validation est utilisé pour contrôler et mesurer la généralisation du réseau. Pendant l'apprentissage l'erreur d'apprentissage décroît continuellement, tandis que sur l'ensemble de validation elle ne diminue que jusqu'à un certain point au-delà duquel elle augmente. A partir de ce point le réseau apprend par cœur les données de l'ensemble d'apprentissage et l'apprentissage doit être stoppé (ceci est illustré par la figure 4.4). Ces deux ensembles servent à

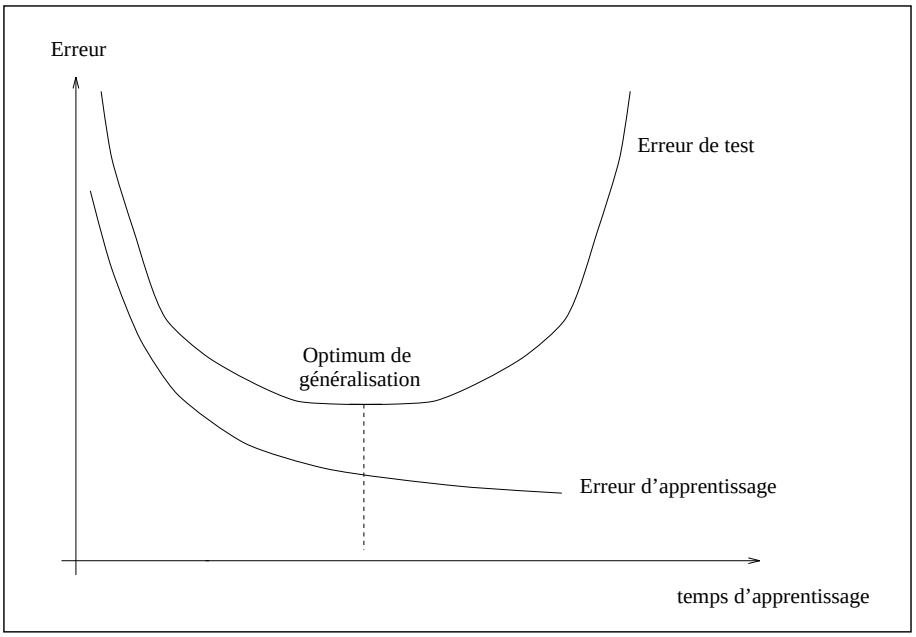

FIG. 4.4 - Evolution des erreurs d'apprentissage et de test au cours du temps.

déterminer l'architecture la plus appropriée pour le problème à traiter : pour différentes architectures (nombre de neurones cachés variable), on contrôle l'erreur de validation et on choisit l'architecture pour laquelle elle est minimale.

Parmi les méthodes utilisant cette méthodologie on peut citer les méthodes appelées "split-samples" , "Cross-validation", "bootstrapping"[2] :

La technique nommée "split-samples" réserve un troisième ensemble de données appelé ensemble de test, pour tester le réseau sur des données qui n'ont jamais été utilisées ni pour l'apprentissage ni pour la validation.

Pour la cross-validation l'ensemble des données de départ est découpé en k parties de taille égale, k pouvant être de la taille de l'ensemble de départ, si tel est le cas on parle de "leave-one-out" sinon on parle de "leave-v-out". Le réseau est entraîné k fois, chaque fois en utilisant $k-1$ parties pour l'apprentissage et la dernière pour la validation et le calcul des erreurs commises en généralisation.

Des variantes de cette méthode existent comme par exemple celle utilisée par Breiman et Spector [Breiman et Spector, 1992] appelé "10-fold cross-validation" ou encore celle nommée "bootstrapping". Le lecteur pourra trouver une comparaison de ces méthodes dans [Tibshirani, 1996]. Notre choix s'est porté sur la méthode du partage de l'ensemble des exemples en trois ensembles (apprentissage, validation, test) citée ci-dessus car nous préférons ne jamais "apprendre" les exemples de l'ensemble de validation.

[2]Les termes en anglais sont laissés car communément utilisés

4.3.3 Amélioration de la généralisation

Il existe différentes façons d'améliorer la généralisation obtenue par un réseau. Nous avons vu plus haut que la capacité d'un réseau détermine sa faculté à généraliser correctement. Cette capacité est reliée à la taille du réseau et est fonction du nombre de neurones utilisés mais aussi du nombre de poids ainsi que de leurs domaines de variation. C'est pourquoi de nombreuses méthodes visant à réduire le nombre de poids ou à contraindre leurs domaines de variations sont apparues de manière à contrôler la capacité d'un réseau de neurones :

- Recherche d'une architecture minimale avant apprentissage telle la méthode des poids partagés qui consiste à imposer des connexions locales et à forcer certaines connexions à partager les mêmes poids (et donc à réduire le nombre de paramètres libres). Ce type d'architecture est couramment utilisée dans les problèmes de reconnaissance de la parole [Waibel, 1989] et de traitement d'image [Loncelle, 1990].

- Minimisation du nombre de poids, voire de l'architecture au cours de l'apprentissage. Cette méthode proposée par Le Cun [Le Cun, Y. et al., 1990] consiste à supprimer les connexions qui n'ont qu'une petite influence sur l'erreur d'apprentissage. Depuis, d'autres méthodes de ce type appelées pruning, ont vu le jour et reposent sur le même principe.

- Introduction d'un terme incluant la complexité du réseau dans la fonction de coût à minimiser pendant l'apprentissage telles que les méthodes appelées "weight decay" [Hinton, 1986] et "weight elimination" [Rumelhart, 1988].

- Utilisation de la connaissance a priori pour structurer le réseau de neurones.

Parmi les méthodes utilisant la connaissance a priori on trouve les réseaux modulaires. En fait on peut voir un système à base de réseaux modulaires comme un très gros réseau à connexions locales et/ou ayant subi un processus d'élagage de poids. Ici l'élagage a été fait à l'aide de la connaissance a priori que l'on a sur le problème à résoudre. Les systèmes modulaires permettent de combiner des modules qui peuvent être hybrides. La justification de l'utilisation de tels modules est de combiner des sous-tâches pour pouvoir résoudre une tâche globale. Par exemple les mélanges conditionnels d'experts ont été introduits par Jacobs et Jordan [Jacobs, R. A. et al., 1991; Jacobs, R. A. et Jordan, M. I., 1991]. Ils sont composés d'une série d'estimateurs, par exemple des réseaux de neurones, et d'un réseau "porte" permettant de choisir l'expert approprié à l'entrée. Le réseau porte calcule la probabilité que l'un des experts soit le plus approprié en se basant sur la connaissance de l'entrée x.

Il y a de nombreuses raisons pratiques pour décomposer une tâche complexe en plusieurs sous-tâches. Le temps de calcul peut être réduit par l'utilisation

d'un groupe de petits réseaux plutôt qu'un seul gros réseau, des tâches de nature différentes peuvent être combinées, la maintenance et la modification de plusieurs petits réseaux est plus simple que celle d'un unique réseau et enfin une série de petits modules peut avoir de meilleures performances en généralisation qu'un gros module. L'approche modulaire permet de plus de combiner des tâches de natures différentes.

D'autres méthodes associent plusieurs réseaux de neurones pour prendre en compte le dilemme biais - variance comme par exemple les ensembles qui travaillent sur une même tâche (voir 4.4.3 pour une description). L'intérêt théorique des ensembles de réseaux de neurones est justifié par le dilemme biais - variance. En effet, si un réseau obtenu A répond d'une meilleure manière au sens du dilemme biais - variance qu'un réseau B alors on sait que le réseau A généralisera mieux que le réseau B [Geman, S. et al., 1992; Breiman, 1994; Hansen, L. K. et Salamon, P., 1990; Perrone, 1993; Wolpert, 1992; Raviv, Y. et Intrator, N., 1996].

L'utilisation d'un ensemble de réseaux de neurones permet de réduire la variance, quand les estimateurs sont identiquement et indépendamment distribués. On admet que l'hypothèse d'indépendance et de distribution identique des estimateurs est vérifiée quand ils sont entraînés sur des ensembles de données différents mais provenant de la même distribution.

De plus, l'utilisation de l'algorithme de rétro-propagation du gradient de l'erreur impose d'initialiser les poids de manière aléatoire. Les réseaux ainsi obtenus sont identiquement distribués et on peut supposer que les différents réseaux, entraînés sur les mêmes données avec une initialisation de poids aléatoire, sont indépendants.

4.4 Une nouvelle fonction de coût régularisante

Notre apport sur les réseaux de neurones est présenté ci-dessous. On présente une nouvelle méthode destinée à améliorer les performances en généralisation des perceptrons multicouches utilisés en tant que réseaux discriminants et approximateurs de fonctions.

4.4.1 Présentation

Motivation

Le choix de la capacité d'un réseau de neurones est primordial pour ses capacités d'apprentissage et de généralisation. Si le modèle est trop simple, il sera incapable d'apprendre la fonction souhaitée, s'il est trop complexe, il sera incapable de généraliser, (bien que capable sans peine de passer par tous les points de l'ensemble d'apprentissage). En fait, dans le second cas, l'espace des solutions

admissibles est beaucoup trop grand. La fonction de coût est aussi l'un des plus importants facteurs contrôlant les performances d'un perceptron multicouche.

La fonction de coût la plus utilisée est la fonction quadratique [Rumelhart, D. E. et al., 1986] avec pour critère d'arrêt le passage à un seuil de l'erreur quadratique moyenne obtenue sur l'ensemble de validation. Nous appellerons cette méthode "MSE" dans tout ce qui suit.

De nombreux algorithmes ont été proposés pour accélérer l'apprentissage, pour trouver le "bon" pas d'apprentissage, ou trouver la bonne méthode d'arrêt de l'apprentissage, mais ils sont très souvent destinés à la minimisation de cette fonction de coût MSE. On peut noter cependant que bien qu'elle vise à réduire la norme des erreurs commises par le réseau, elle n'a aucun pouvoir de contrôle sur la distribution de ces dernières au cours de l'apprentissage. Or, contrôler la "forme" de la distribution des erreurs lors de la phase d'apprentissage nous a semblé être un problème de grande importance notamment dans le cas des problèmes de classification où la notion de marge de classification s'impose d'elle-même. En d'autres termes, est-il possible de contrôler la convergence de l'algorithme d'apprentissage de manière à contrôler la forme de la distribution des erreurs et ainsi d'améliorer la robustesse de l'apprentissage ? Un contrôle de la forme de la distribution des erreurs permet-il un contrôle de la capacité du réseau de neurones et donc une meilleure généralisation ? Dans ce qui suit une nouvelle fonction de coût est proposée dans ce but.

Cette nouvelle fonction de coût et ses conditions d'utilisation sont décrites dans la présente section. La section 4.4.2 est une comparaison entre la nouvelle fonction de coût et la fonction de coût MSE, dans le cadre d'un problème de détection de visages à l'aide d'un unique perceptron multicouche.

Dans la section 4.4.3 on s'attachera à réaliser une autre comparaison entre les deux fonctions de coût sur un autre problème de classification qualifié de benchmark. Le but sera de comparer les résultats obtenus par la technique de Bagging mise au point par Breiman [Breiman, 1994] [Bishop, 1997] et plus particulièrement les résultats affichés dans [Quilan, 1998] obtenus avec la fonction de coût MSE avec ceux obtenus par la nouvelle fonction de coût. Enfin une dernière comparaison sera réalisée sur un problème de prédiction de série temporelle au cours de la section 4.4.4.

Description

Une manière de contrôler la forme de la distribution des erreurs au cours de l'apprentissage est la prise en compte d'un moment d'ordre 4 des erreurs : la variance des erreurs quadratiques, en plus de l'erreur quadratique classique.

La fonction de coût à minimiser devient :

$$C^x = \sum_{i \in O} (d_i^x - s_i^x)^2 + \sum_{i \in O} \left(\frac{1}{P} \sum_{a=1}^{P} \left((d_i^a - s_i^a)^2 - \frac{1}{P} \sum_{b=1}^{P} (d_i^b - s_i^b)^2 \right)^2 \right) \quad (4.19)$$

qui peut s'écrire sous la forme de l'addition de deux coûts :

$$C^x = \sum_{i \in O} C^x_{quad} + \sum_{i \in O} C^x_{var} \quad (4.20)$$

où P est l'ensemble des exemples d'apprentissage, O est l'ensemble des neurones de sortie, s_i^x est la valeur du neurone de sortie i après la présentation de l'exemple x et d_i^x est la valeur désirée pour le neurone correspondant.

Nous avons utilisé la méthode stochastique et dans ce cas le gradient attaché à un neurone de sortie i pour une présentation d'un exemple x est :

$$\begin{aligned} G_i^x &= \frac{\partial C_i^x}{\partial a_i^x} && (4.21) \\ &= \frac{\partial}{\partial a_i^x} \left((d_i^x - s_i^x)^2 \right) \\ &+ \frac{\partial}{\partial a_i^x} \left(\frac{1}{P} \sum_{a=1}^{P} \left((d_i^a - s_i^a)^2 - \frac{1}{P} \sum_{b=1}^{P} (d_i^b - s_i^b)^2 \right)^2 \right) && (4.22) \end{aligned}$$

donc, si f est la fonction de transfert d'un neurone :

$$\begin{aligned} G_i^x &= \frac{\partial C_i^x}{\partial a_i^x} && (4.23) \\ &= -2 f'(a_i^x)(d_i^x - s_i^x) \\ &- \frac{4}{P} f'(a_i^x)(d_i^x - s_i^x) \left((d_i^x - s_i^x)^2 - \frac{1}{P} \sum_{e=1}^{P} (d_i^e - s_i^e)^2 \right) && (4.24) \end{aligned}$$

Le gradient peut être écrit sous la forme :

$$\begin{aligned} G_i^x &= -2 f'(a_i^x)(d_i^x - s_i^x) \\ &- \frac{4}{P} f'(a_i^x)(d_i^x - s_i^x) \left((d_i^x - s_i^x)^2 - MSE \right) && (4.25) \end{aligned}$$

ou encore :

$$G_i^x = G_{i_{quad}}^x(1+\gamma) \tag{4.26}$$

avec

$$\gamma = \frac{2}{P}\left((d_i^x - s_i^x)^2 - MSE\right) \tag{4.27}$$

ou MSE représente l'erreur quadratique moyenne obtenue sur tous les exemples avant la présentation de l'exemple x.

On s'aperçoit que ce gradient peut être vu comme le gradient utilisé habituellement, dû à l'erreur quadratique, mais corrigé, pondéré, par γ qui représente une mesure entre l'erreur quadratique commise sur l'exemple x et l'erreur quadratique moyenne commise sur l'ensemble des exemples. Dans tout ce qui suivra nous appellerons cette méthode d'apprentissage "VMSE" pour *"***V***ariance and* **M***ean* **S***quared* **E***rror"*. Le principe du calcul du gradient d'un neurone caché est à l'identique de la rétro-propagation de l'erreur quadratique (méthode MSE).

La loi de modification des poids

En reprenant l'équation 4.25 on pose :

$$G_i^x = G_{i_{quad}}^x + G_{i_{var}}^x \tag{4.28}$$

avec

$$G_{i_{quad}}^x = -2f'(a_i^x)(d_i^x - s_i^x) \tag{4.29}$$

$$G_{i_{var}}^x = -\frac{4}{P}f'(a_i^x)(d_i^x - s_i^x)\left((d_i^x - s_i^x)^2 - MSE\right) \tag{4.30}$$

La loi de modification des poids devient alors :

$$\Delta w_{ij}^{t+1} = \alpha_{quad} G_{i_{quad}}^x s_j + \alpha_{var} G_{i_{var}}^x s_j + \beta \Delta w_{ij}^t \tag{4.31}$$

avec α_{quad} le pas d'apprentissage sur l'erreur quadratique, α_{var} le pas d'apprentissage sur la variance de l'erreur quadratique et β l'inertie (terme qui permet d'accélérer la convergence [Plaut et al., 1986]).

Implémentation de la rétropropagation

Du fait de la nécessité de calculer l'erreur quadratique moyenne et la variance de l'erreur quadratique moyenne après chaque modification des poids, le temps de calcul peut devenir très important. En effet pour chaque présentation d'un exemple il faut réaliser P propagations et 1 rétro-propagation. Aussi, afin de réduire les temps de calcul nous utiliserons l'algorithme suivant qui, comme nous le verrons plus loin, permet d'introduire une contrainte plus forte sur le contrôle de la forme de la distribution des erreurs et d'ajouter de la stabilité au processus d'apprentissage :

- ⋆ pour toutes les itérations
  - pour tous les exemples x
    - calcul de $f_w(x)$
    - calcul de G^x_{quad} et G^x_{var} pour les neurones de sorties
    - calcul de G^x_{quad} et G^x_{var} pour les neurones cachés
    - modifications des poids
  - calcul de l'erreur quadratique moyenne et de la variance de l'erreur quadratique moyenne

Normalisation des pas d'apprentissage

Dans les différentes études comparatives que nous allons présenter ci-après et pour étudier l'influence du terme ajouté dans la fonction de coût on définit la variable ν par :

$$\nu = \frac{\alpha_{var} P}{\alpha_{quad}} = \frac{\alpha^{'}_{var}}{\alpha_{quad}} \tag{4.32}$$

ou P représente la taille de l'ensemble utilisé dans le calcul de la MSE (le nombre d'exemples d'apprentissage).

4.4.2 Etude comparative : Classification

Introduction

Dans le cadre d'une classification par discrimination, le but recherché est la détermination d'un classifieur qui, à chaque observation x (vecteur de $\mathbb{R}^n$) associe une classe parmi un ensemble fixé de classes. Nous nous intéresserons particulièrement ici au cas où le classifieur est un perceptron multicouche et

où on utilise q neurones de sortie (où q est le nombre de classes). Le codage habituellement utilisé consiste alors à attribuer le code (d_1, ..., d_q) à la classe h avec $d_i = a$ si $i = h$, $d_i = b$ si $i \neq h$ et $a \geq b$. Les valeurs habituellement utilisées sont $(a, b) = (+1, -1)$ ou $(1, 0)$ ou encore $(0.9, 0.1)$. La sortie du réseau est un vecteur $\mathbf{y}$ de dimension q. Si $q = 2$ on peut se contenter d'utiliser un réseau à une seule sortie.

Pour les perceptrons multicouches qui utilisent des fonctions d'activation et des poids à valeurs continues, quelques valeurs particulières des sorties du réseau de neurones sont utilisées comme valeurs désirées pour les différentes classes du problème de classification à traiter. Par conséquent on observe deux types d'erreurs : l'erreur d'apprentissage liée à la minimisation de la fonction de coût que nous appellerons dans ce qui suit l'erreur d'estimation et l'erreur de classification. L'erreur d'estimation est la différence entre ce qu'on souhaitait obtenir en sortie du réseau (pour un exemple donné) et ce qu'on obtient réellement à la fin de l'apprentissage. Tandis que l'erreur de classification existe quand l'erreur d'estimation est supérieure à un seuil préétabli.

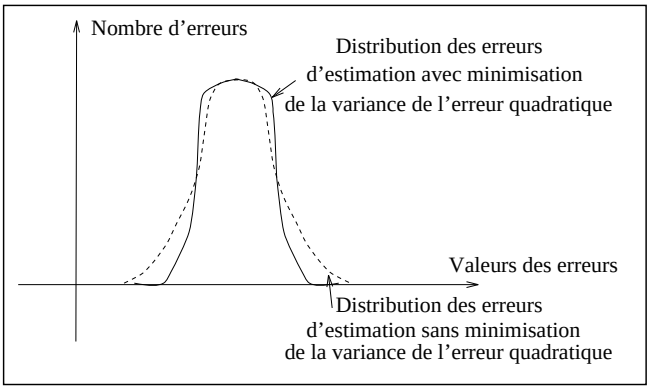

FIG. 4.5 - Influence de la minimisation de la variance de l'erreur quadratique sur la distribution des erreurs d'estimation.

Les bornes de l'erreur de généralisation d'un système composite de classifieurs ont été précédemment établies [Schapire et al., 1997][Holger et Bengio, 1998] et sont basées sur la notion de marge de classification. La taille de la marge dépend à la fois de l'erreur quadratique moyenne obtenue et de la distribution des erreurs d'estimation soit donc de la variance de l'erreur quadratique obtenue sur chaque classe (voir figure 4.5). Ainsi la performance en terme de bonne classification dépend de la forme particulière de la distribution des erreurs d'estimation. Il s'ensuit que le choix d'une fonction de coût appropriée, de manière à contrôler la forme de la distribution des erreurs d'estimation peut être crucial pour obtenir une solution optimale au problème de classification posé.

Nous proposons ici d'utiliser la nouvelle fonction de coût décrite précédemment. En effet, cette méthode approche le problème en prenant en compte un

moment d'ordre 4 (la variance de l'erreur quadratique) introduit dans la fonction de coût à minimiser lors de la phase d'apprentissage pour améliorer les résultats en généralisation. Cette approche peut être utilisée lorsque des méthodes [Schapire et al., 1997] [Breiman, 1994] utilisant plusieurs réseaux de neurones afin d'améliorer la marge de classification nécessiteraient un temps de calcul trop important pour des applications industrielles. Nous étudierons son intérêt dans le cadre de la méthode dite du Bagging un peu plus loin (voir 4.4.3).

Considérons un problème de classification à deux classes C_1 et C_2 classifiées à l'aide d'un réseau discriminant à une sortie. Le but de la phase d'apprentissage est d'obtenir les sorties suivantes pour le réseau :

- si $x \in C_1$ alors $f_w(x) = d_1$
- si $x \in C_2$ alors $f_w(x) = d_2$

avec x le vecteur d'entrée, d_1 et d_2 les sorties désirées respectivement pour un exemple de la classe C_1 et C_2 et $f_w(x)$ la réponse donnée par le réseau de neurones. A la fin de la phase d'apprentissage, ce réseau de neurones a une erreur quadratique moyenne m_1 sur la classe C_1 avec une variance σ_1^2 et une erreur quadratique moyenne m_2 sur la classe C_2 avec une variance σ_2^2 :

$$\sigma_1^2 = \frac{1}{n_1} \sum_{a=1}^{n_1} \left[(d^a - s^a)^2 - \frac{1}{n_1} \sum_{b=1}^{n_1} (d^b - s^b)^2 \right]^2 \quad a \in C_1 \quad b \in C_1 \tag{4.33}$$

$$\sigma_2^2 = \frac{1}{n_2} \sum_{c=1}^{n_2} \left[(d^c - s^c)^2 - \frac{1}{n_2} \sum_{d=1}^{n_2} (d^d - s^d)^2 \right]^2 \quad \mathrm{c} \in \mathrm{C}_2 \quad \mathrm{d} \in \mathrm{C}_2 \tag{4.34}$$

où :

- n_p est le nombre d'exemples de la classe C_p ;
- s^x est la valeur de la sortie du réseau pour l'entrée x ;
- d^x est la sortie désirée pour l'entrée x.

Comme décrit en 4.4.1 on peut prendre en compte la minimisation de la variance de l'erreur quadratique et, par extension, des variances σ_1^2 et σ_2^2, en ajoutant à l'habituelle fonction de coût, basée uniquement sur l'erreur quadratique, un terme associé à la variance de l'erreur quadratique observée sur chaque classe. Le but étant ici d'augmenter la marge de classification.

L'expression de la nouvelle fonction de coût sur un problème de classification à deux classes devient :

$$\begin{aligned} C^x = & \quad (d^x - s^x)^2 \\ & + \frac{1}{n_1} \sum_{\substack{a=1 \\ a \in C_1}}^{n_1} \left[(d^a - s^a)^2 - \frac{1}{n_1} \sum_{\substack{b=1 \\ b \in C_1}}^{n_1} (d^b - s^b)^2 \right]^2 \\ & + \frac{1}{n_2} \sum_{\substack{c=1 \\ c \in C_2}}^{n_2} \left[(d^c - s^c)^2 - \frac{1}{n_2} \sum_{\substack{d=1 \\ d \in C_2}}^{n_2} (d^d - s^d)^2 \right]^2 \end{aligned} \tag{4.35}$$

L'expression du gradient de la fonction de coût C pour le neurone de sortie i et un exemple $x \in C_1$ est :

$$\begin{aligned} \left. \frac{\partial C^x}{\partial w_{ij}} \right|_{x \in C1} &= \frac{\partial}{\partial w_{ij}} \left[(d^x - s^x)^2 \right] \\ &+ \frac{\partial}{\partial w_{ij}} \left[\frac{1}{n_1} \sum_{\substack{a=1 \\ a \in C_1}}^{n_1} \left[(d^a - s^a)^2 - \frac{1}{n_1} \sum_{\substack{b=1 \\ b \in C_1}}^{n_1} (d^b - s^b)^2 \right]^2 \right] \\ &+ \frac{\partial}{\partial w_{ij}} \left[\frac{1}{n_2} \sum_{\substack{c=1 \\ c \in C_2}}^{n_2} \left[(d^c - s^c)^2 - \frac{1}{n_2} \sum_{\substack{d=1 \\ d \in C_2}}^{n_2} (d^d - s^d)^2 \right]^2 \right] \end{aligned} \tag{4.36}$$

Le gradient, qui ne dépend pas de la variance observée sur la classe C_2, est par conséquent :

$$\begin{aligned} \left. \frac{\partial C^x}{\partial w_{ij}} \right|_{k \in C1} &= s_j^x (-2 f'(a^x)(d^x - s^x) \\ &- s_j^x \frac{4}{n_1} f'(a^x)(d^x - s^x)) \\ &\quad \left[(d^x - s^x)^2 - \frac{1}{n_1} \sum_{\substack{e=1 \\ e \in C_1}}^{n_1} (d^e - s^e)^2 \right] \end{aligned} \tag{4.37}$$

où a^x est la valeur de la somme pondérée à l'entrée du neurones i pour l'exemple x.

Le calcul est identique pour un exemple appartenant à la classe C_2. On retrouve donc la formulation proposée en 4.4.1 et le calcul du gradient attaché aux neurones cachés ne change pas dans sa forme sauf qu'il comporte deux termes. La règle de modification des poids est :

$$\Delta w_{ij}^{t+1} = \alpha_{quad} G_{quad}^x + \alpha_{var} G_{var}^x + \beta \Delta w_{ij}^t \tag{4.38}$$

avec α_{quad} le pas d'apprentissage sur l'erreur quadratique, α_{var} le pas d'apprentissage sur la variance de l'erreur quadratique et β l'inertie. On peut noter ici qu'il est possible de différencier le pas d'apprentissage sur la variance de l'erreur quadratique de la classe C_1 par rapport à la classe C_2 (et vice versa). Il est aussi possible de ne calculer qu'une seule variance, indépendamment de la classe, et chercher à ne minimiser que cette variance "globale".

Conditions expérimentales

Le problème de classification à deux classes que nous allons utiliser pour notre étude comparative est un problème de détection de visages. La nouvelle fonction de coût est testée sur le "pré-réseau" de l'application MULTRAK [Bernier et al., 1998a], qui est un système temps réel pour la détection et le suivi automatique de personnes lors d'une visioconférence. Ce système est capable de détecter et de suivre continuellement la position des visages présents dans le champ de vision de la caméra. Le cœur du système est un réseau de neurones modulaire [Féraud, R. et Bernier, O., 1997] qui détecte les visages avec précision (voir figure 4.6). Le pré-réseau est utilisé comme filtre, c'est-à-dire que ne seront présentées au réseau de neurones modulaire que les images détectées par le pré-réseau comme étant des visages. Il doit être beaucoup plus rapide que le réseau modulaire sans dégrader le taux de détection des visages pour l'ensemble du système. Pour des questions de performances temps réel, la vitesse du pré-réseau est critique et impose de n'utiliser qu'un seul réseau de neurones discriminant.

Nous avons donc entraîné deux pré-réseaux en tant que détecteurs de visages : un en utilisant la nouvelle fonction de coût décrite précédemment et l'autre uniquement à l'aide de l'erreur quadratique. Chaque réseau de neurones est un perceptron multicouche, utilisant des fonctions d'activation sigmoïdales, possédant 300 entrées (correspondant à une image de 15x20 pixels), une couche cachée de 8 neurones et une sortie. La base de données utilisée est constituée de 3 parties :

- Ensemble d'apprentissage : 7000 visages de face ou tournés et 7000 non visages ;
- Ensemble de validation : 7000 visages de face ou tournés et 7000 non visages ;
- Ensemble de test : 7000 visages de face ou tournés et 7000 non visages.

sachant qu'un non visage est une image quelconque autre qu'un visage.

Pour comparer les deux fonctions de coût utilisées, différentes expérimentations ont été réalisées. Pour chaque expérimentation, 50 apprentissages ont été réalisés avec différentes initialisations des poids. Ceci permet d'obtenir, pour chaque condition expérimentale, la moyenne et l'intervalle de confiance de chaque résultat obtenu. Chaque apprentissage a été stoppé quand le coût sur l'ensemble

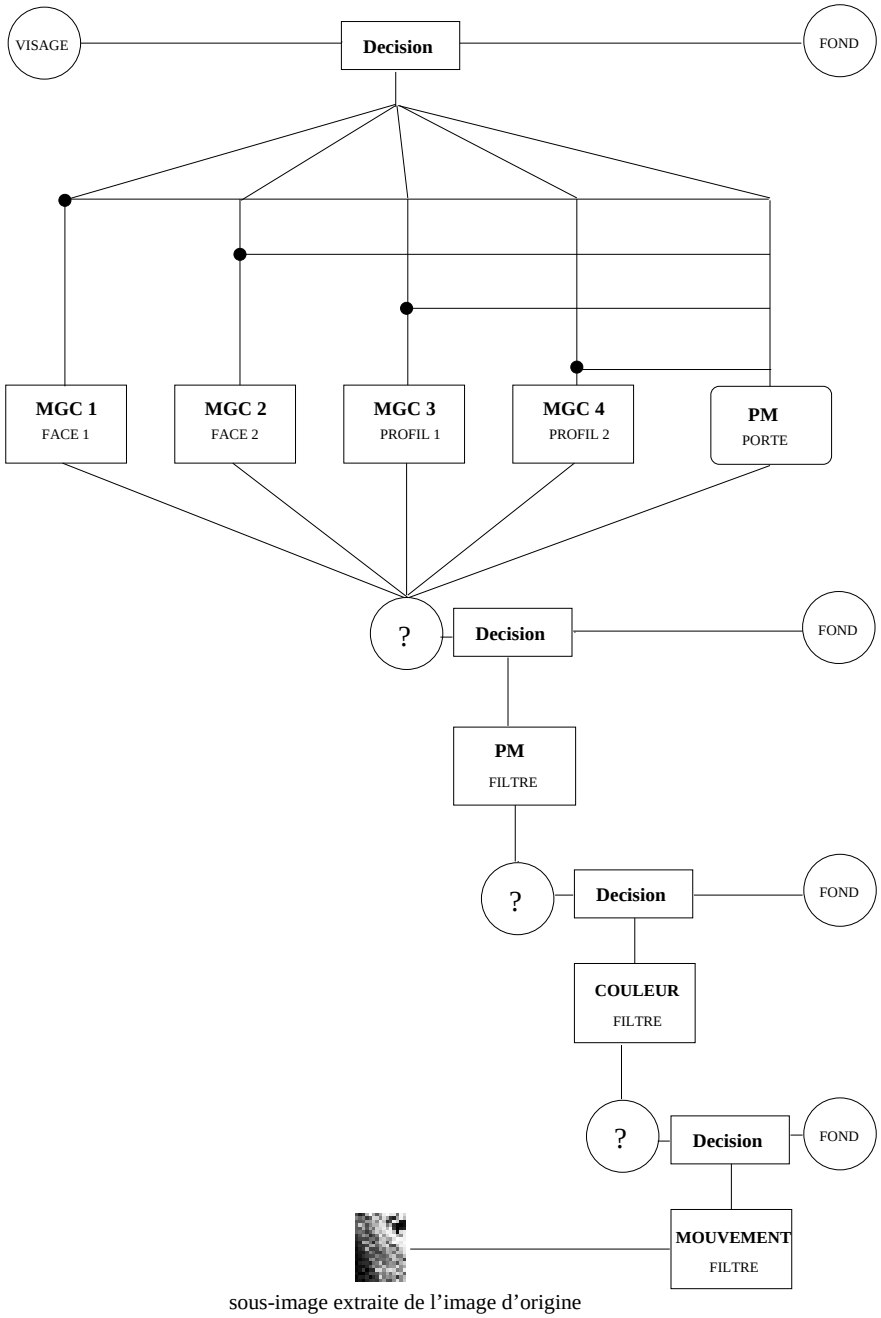

FIG. 4.6 - Le système MULTRACK : Le pré-reseau (PM) se situe après les filtres de couleurs et de mouvements (MGC : Modèle Génératif Contraint).

de validation ne diminuait plus depuis 200 itérations. Dans ce cas l'erreur quadratique moyenne, la variance de l'erreur quadratique sur chaque classe, la marge et le taux de détection sont calculés sur la meilleure configuration des poids pour cet apprentissage sur chaque ensemble (apprentissage, validation, test) et sont

détaillés ci-après.

Dans les parties suivantes, nous étudions et comparons les deux fonctions de coût. Nous montrons que si le pas d'apprentissage sur le terme de variance ajouté est bien choisi :

1. la variance sur l'ensemble d'apprentissage diminue ;
2. la marge sur l'ensemble d'apprentissage s'accroît ;
3. les performances en classification sur l'ensemble de test augmentent.

Dans les figures qui vont suivre les résultats résultants de la nouvelle fonction de coût sont étiquetés "VMSE" et ceux résultants uniquement de l'erreur quadratique "MSE".

L'influence du terme sur la variance

Dans cette partie le paramètre α_{quad} relié à l'erreur quadratique possède une valeur constante de 10^{-2}. L'influence de ν est examinée sur l'intervalle $[10^{-4} : 10^{2}]$ pour évaluer comment le gradient ajouté interagit avec le gradient de l'erreur quadratique. La valeur de l'inertie, β, a elle été fixée à 0.9. Les comparaisons sont réalisées pour l'erreur quadratique moyenne globale (pour tous les exemples quelles que soient leurs classes d'appartenance) et pour la variance de l'erreur quadratique sur chaque classe. Les résultats pour le pré-réseau entraîné avec la fonction de coût MSE sont constants puisque α_{quad} est constant.

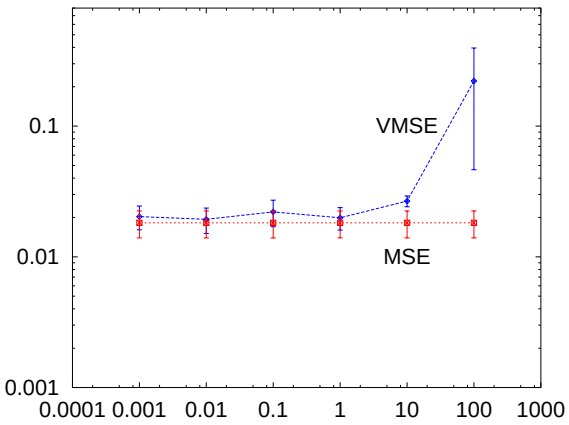

FIG. 4.7 - L'erreur quadratique moyenne globale sur l'ensemble d'apprentissage avec les deux fonctions de coût en fonction de ν.

La figure 4.7 montre les résultats obtenus sur l'erreur quadratique moyenne globale avec les deux fonctions de coût sur l'ensemble d'apprentissage. Pour

$\nu \in [10^{-4} : 10]$, les deux fonctions de coût fournissent à peu près les mêmes résultats avec le même intervalle de confiance. En revanche, pour $\nu = 100$, l'erreur quadratique moyenne globale augmente beaucoup avec la nouvelle fonction de coût. On voit ici que dans ce cas, α'_{var} est trop grand, comparé à α_{quad}. La minimisation de la variance de l'erreur quadratique au cours de l'apprentissage empêche la minimisation de l'erreur quadratique et le réseau de neurones renvoie toujours la même valeur de sortie, pour laquelle la variance de l'erreur est alors minimale.

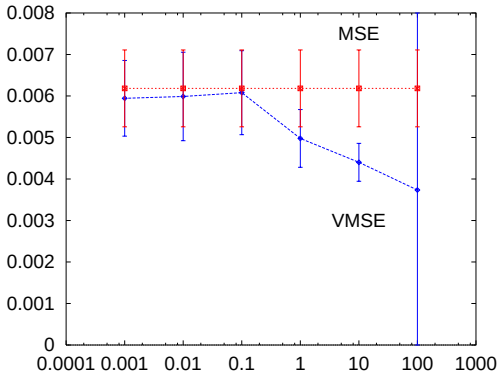

FIG. 4.8 - La variance de l'erreur quadratique pour la première classe (visages) sur l'ensemble d'apprentissage avec les deux fonctions de coût en fonction de ν.

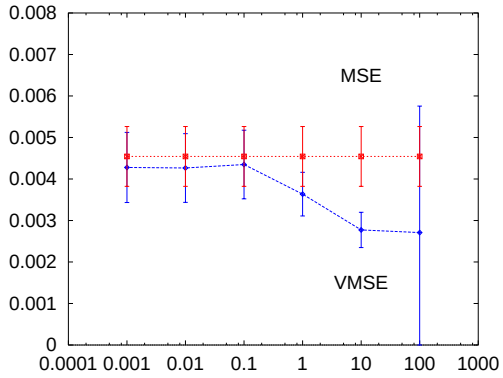

FIG. 4.9 - La variance de l'erreur quadratique pour la deuxième classe (non visages) sur l'ensemble d'apprentissage avec les deux fonctions de coût en fonction de ν.

Les figures 4.8 et 4.9 montrent les résultats obtenus sur la variance de l'erreur quadratique associée à chaque classe de l'ensemble d'apprentissage. Pour $\nu \in [10^{-4} : 10^{-1}]$ les deux fonctions de coût présentent des performances similaires. Pour $\nu \in [10^{-1} : 10]$ la nouvelle fonction de coût réduit les variances jusqu'à obtenir un gain de 37 % par rapport à la fonction de coût standard et ce avec un intervalle de confiance du même ordre. Par contre, pour $\nu = 100$, les variances sont aussi améliorées mais avec un intervalle de confiance insuffisant.

Ces résultats montrent que le terme de variance ajouté interagit avec le terme sur l'erreur quadratique. S'il est du même ordre, il permet d'améliorer les résultats obtenus sur la variance de l'erreur quadratique de chaque classe. Une valeur bien choisie du pas d'apprentissage α'_{var} améliore les résultats sur la variance de l'erreur quadratique associée à chaque classe sans dégrader l'erreur quadratique moyenne globale.

Augmentation de la marge

Une seconde expérimentation montre la relation entre la minimisation de la variance de l'erreur quadratique et la maximisation de la marge pour $\nu = 1$ (α_{quad}=0.01 ; α'_{var}=0.01).

Pour quantifier le pourcentage de la population à l'intérieur de la marge, proche de la frontière de décision, et par conséquent le taux de biens classés à l'extérieur de la marge, nous avons déterminé un seuil θ (voir figure 4.10) pour différentes valeurs de la marge. Ce seuil est réglé de manière à obtenir le meilleur taux de détection en fonction de la marge sur l'ensemble d'apprentissage, sachant qu'un exemple est considéré bien classé si :

- $f_w(k) \leq \theta$ et k $\in C_1$
- $f_w(k) \geq \theta$ + Marge et k $\in C_2$

La figure 4.11 montre que pour une marge donnée le taux de détection (sur l'ensemble d'apprentissage) avec la nouvelle fonction de coût est meilleur qu'avec la fonction de coût standard. Par conséquent, pour un même taux de détection, la marge est augmentée.

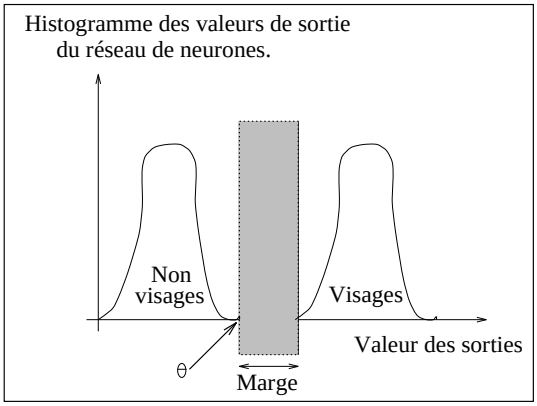

FIG. 4.10 - Explication du taux de biens classés en fonction de la marge.

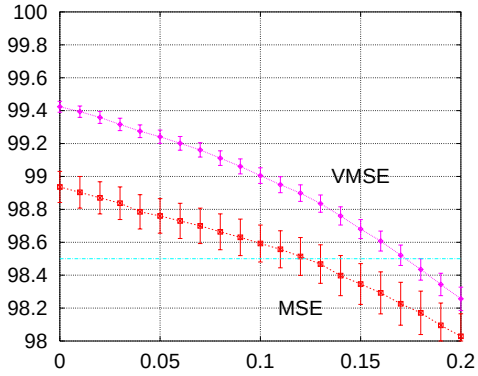

FIG. 4.11 - Le taux de biens classés en fonction de la marge pour l'ensemble d'apprentissage.

Par exemple, pour un taux de détection de 98.5 % (voir figure 4.11) la nouvelle fonction de coût fournit une marge de 0.17 $\pm$ 0.01 alors que la fonction de coût standard ne fournit que 0.12 $\pm$ 0.04. L'effet de la minimisation des variances est clairement visible ici. Il est de pousser la distribution des erreurs d'estimation sur les visages vers la droite et celle des non visages vers la gauche (voir figure 4.10).

Le deuxième but est donc atteint : la minimisation de la variance de l'erreur quadratique sur chaque classe permet d'améliorer la marge de classification.

Une meilleure marge accroît les performances en généralisation

Cette partie montre que la maximisation de la marge sur l'ensemble d'apprentissage améliore les performances en généralisation (sur l'ensemble de test). La figure 4.12 montre l'effet d'une marge améliorée : la différence entre les deux courbes représente l'amélioration obtenue pour la marge. Avec la fonction de coût MSE et pour un taux de détection de 99.5 % le taux de fausse alarme est de 8 % alors qu'avec la nouvelle fonction de coût ce taux est de seulement 5 % ce qui représente une amélioration de 37 %.

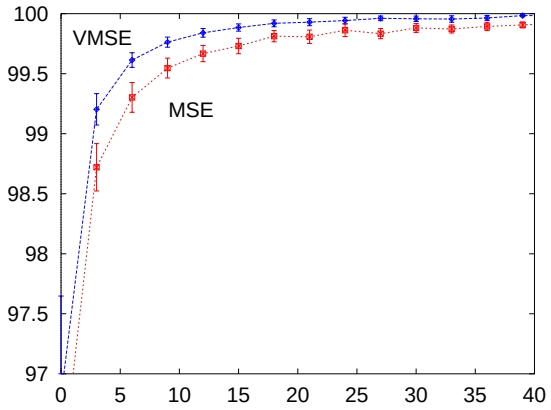

FIG. 4.12 - Le taux de détection pour les visages sur l'ensemble de test en fonction du taux de fausse alarme (non visages classés comme visages) pour les deux fonctions de coût.

Discussion

Une nouvelle méthode a été proposée et testée pour améliorer les performances en généralisation sur un problème de classification à l'aide de perceptrons multicouches. Cette méthode est utilisée dans le réseau détecteur de visages appelé pré-réseau de l'application MULTRAK [Bernier et al., 1998a]. L'utilisation de notre nouvelle fonction de coût a permis d'améliorer les performances du pré-réseau comparé à la fonction de coût basée uniquement sur l'erreur quadratique : le taux de fausse alarme est réduit de 20 % (sur l'ensemble de test A de la base de données du CMU [Hunke, 1994; Rowley et al., 1995]) pour le même taux de détection.

Cette méthode a été appliquée à un problème de classification à deux classes mais peut être étendue à des problèmes de classification possédant plus de classes. Elle peut aussi être utilisée dans les méthodes qui utilisent plusieurs réseaux de neurones pour améliorer la généralisation comme nous allons le voir dans le problème suivant.

4.4.3 Etude comparative : Bagging

Introduction

Bagging, acronyme de Bootstrap Aggregating, est une procédure de stabilisation ([Bishop, 1997] p55) qui émule la possession de répliques indépendantes d'un ensemble d'apprentissage. C'est une méthode assez récente [Breiman, 1994] destinée à améliorer les performances des systèmes classifieurs parmi d'autres méthodes existantes [Schapire et al., 1997], [Holger et Bengio, 1998], [Kohavi et John, 1995], [Ragavan et Rendell, 1993], [Utgogg et Brodley, 1990]. Ces différentes méthodes ont montré l'intérêt qu'il y a à générer et à combiner plusieurs classifieurs afin d'augmenter la précision de l'ensemble, et sont justifiées théoriquement par le dilemme biais-variance [Geman, S. et al., 1992; Breiman, 1994; Hansen, L. K. et Salamon, P., 1990; Perrone, 1993; Wolpert, 1992; Raviv, Y. et Intrator, N., 1996].

Supposons que l'on possède un ensemble de données d'apprentissage, de taille N, ou chacune de ces données appartient à une classe dont le nombre est K, et une *machine à apprendre* à l'aide de laquelle on veut construire un classifieur étant donné les données d'apprentissage. Le classifieur obtenu aura alors une précision inhérente à la représentativité des données apprises. En effet, pour un problème de régression (aux moindres carrés), l'évaluation d'un estimateur $f_{\mathcal{D}}$ entraîné sur une ensemble de données $\mathcal{D}$ de taille fixée, est faite en prenant l'erreur moyenne sur tous les exemples appartenant à $\mathcal{D}$ du critère des moindres carrés :

$$E_{\mathcal{D}}\left[(y - f_{\mathcal{D}})^2\right] = E_{\mathcal{D}}\left[(E\left[y|x\right] - f_{\mathcal{D}})^2\right] + E_{\mathcal{D}}\left[(y - E\left[y|x\right])^2\right]$$

Le second terme ne dépend que du bruit contenu dans les données. La performance d'un estimateur est donnée par :

$$E_{\mathcal{D}}\left[(E\left[y|x\right] - f_{\mathcal{D}})^2\right] = (E_{\mathcal{D}}\left[f_{\mathcal{D}}\right] - E\left[y|x\right])^2 + E_{\mathcal{D}}\left[(f_{\mathcal{D}} - E_{\mathcal{D}}\left[f_{\mathcal{D}}\right])^2\right]$$

Le premier terme est le biais, il ne dépend que de l'ensemble de fonctions Γ, où est choisie $f_{\mathcal{D}}$: si l'ensemble de fonctions Γ est suffisamment grand ou bien choisi, il peut contenir une fonction suffisamment proche de la fonction de régression (la fonction de régression est ici la fonction qui minimise l'erreur quadratique). Néanmoins, en choisissant un ensemble de fonctions de grande taille, on risque d'augmenter la variance : la distance entre la fonction $f_{\mathcal{D}}$ obtenue et la meilleure fonction que l'on peut obtenir compte tenu de l'utilisation d'un ensemble de données de taille fixé $E_{\mathcal{D}}[f_{\mathcal{D}}]$. C'est ce que l'on appelle le dilemme biais-variance. L'utilisation d'un ensemble de réseaux de neurones permet de réduire la variance, quand les estimateurs $f_{\mathcal{D}}$ sont identiquement et indépendamment distribués. Pour simplifier les notations, on prendra f_i pour $f_{\mathcal{D}}$ et E pour $E_{\mathcal{D}}$:

$$\bar{f} = \frac{1}{N}\sum_{i=1}^{N} f_i \tag{4.39}$$

La variance de la moyenne des estimateurs s'exprime par :

$$E\left[\left(\bar{f} - E\left[\bar{f}\right]\right)^2\right] = E\left[\left(\frac{1}{N}\sum_{i=1}^{N} f_i - E\left[\frac{1}{N}\sum_{i=1}^{N} f_i\right]\right)^2\right] \tag{4.40}$$

$$\Leftrightarrow E\left[\left(\bar{f} - E\left[\bar{f}\right]\right)^2\right] =$$

$$E\left[\left(\frac{1}{N}\sum_{i=1}^{N} f_i\right)^2\right] + \left(E\left[\frac{1}{N}\sum_{i=1}^{N} f_i\right]\right)^2 - 2E\left[\frac{1}{N}\sum_{i=1}^{N} f_i E\left[\frac{1}{N}\sum_{i=1}^{N} f_i\right]\right] \tag{4.41}$$

$$\Leftrightarrow E\left[\left(\bar{f} - E\left[\bar{f}\right]\right)^2\right] = E\left[\left(\frac{1}{N}\sum_{i=1}^{N} f_i\right)^2\right] - \left(E\left[\frac{1}{N}\sum_{i=1}^{N} f_i\right]\right)^2 \tag{4.42}$$

Exprimons les deux termes de l'équation :

$$E\left[\left(\frac{1}{N}\sum_{i=1}^{N} f_i\right)^2\right] = \frac{1}{N^2}\sum_{i=1}^{N} E\left[f_i^2\right] + \frac{2}{N^2}\sum_{i<j} E\left[f_i f_j\right] \tag{4.43}$$

et

$$\left(E\left[\frac{1}{N}\sum_{i=1}^{N} f_i\right]\right)^2 = \frac{1}{N^2}\sum_{i=1}^{N} \left(E\left[f_i\right]\right)^2 + \frac{2}{N^2}\sum_{i<j} E\left[f_i\right] E\left[f_j\right] \tag{4.44}$$

On obtient alors :

$$\Leftrightarrow E\left[\left(\bar{f} - E\left[\bar{f}\right]\right)^2\right] = \frac{1}{N^2}\sum_{i=1}^{N}\left(E\left[f_i^2\right] - \left(E\left[f_i\right]\right)^2\right) + \frac{2}{N^2}\sum_{i<j}\left(E\left[f_i f_j\right] - E\left[f_i\right] E\left[f_j\right]\right) \tag{4.45}$$

$$\Leftrightarrow E\left[\left(\bar{f} - E\left[\bar{f}\right]\right)^2\right] = \frac{1}{N^2}\sum_{i=1}^{N} \text{ Var } (f_i) + \frac{2}{N^2}\sum_{i<j}\left(E\left[f_i f_j\right] - E\left[f_i\right] E\left[f_j\right]\right) \tag{4.46}$$

Si les estimateurs sont tous égaux, $\forall i,\ f_i = f$, on a :

$$E\left[\left(\bar{f} - E\left[\bar{f}\right]\right)^2\right] = E\left[\left(f - E\left[f\right]\right)^2\right] = \text{ Var } (f) \tag{4.47}$$

Si les estimateurs sont tous identiquement et indépendamment distribués, on a :

$$\forall i, j \text{ on a Var } (f_i) = \text{ Var } (f_j) \tag{4.48}$$

$$\Rightarrow E\left[\left(\bar{f} - E\left[\bar{f}\right]\right)^2\right] = \frac{1}{N} \text{ Var } (f_i) \tag{4.49}$$

On admet que l'hypothèse d'indépendance et de distribution identique des estimateurs est vérifiée quand ils sont entraînés sur des ensembles de données différents mais provenant de la même distribution.

Malheureusement, dans de nombreux problèmes la taille de l'ensemble d'apprentissage est limitée, et le découper pour construire d'autres classifieurs peut diminuer la précision de chacun. On peut essayer alors de "créer" de nouveaux ensembles d'apprentissage à l'aide d'une approximation bootstrap et de l'ensemble d'apprentissage que nous qualifierons d'original. Pour construire un deuxième ensemble d'apprentissage de la même taille que l'ensemble original, on réalise N tirages indépendants avec remise dans l'original. Ce deuxième ensemble sera donc différent de l'ensemble original puisque chaque exemple qu'il contient peut apparaître plusieurs fois. Notons cependant que le deuxième ensemble provient de la même distribution de probabilité que le premier, la fréquence d'apparition des exemples étant juste différente. A présent, à l'aide du nouvel ensemble on construit un deuxième classifieur. L'opération complète pouvant être réitérée autant de fois que l'on veut, notons ce nombre T. Finalement on définit le classifieur global comme étant la combinaison des T classifieurs ainsi construits. Sa réponse étant, pour un exemple présenté simultanément à tous les classifieurs construits, la réponse moyenne de ces derniers.

Notons que la nouvelle fonction de coût VMSE ne minimise pas la variance telle que considérée dans l'équation 4.49. Ici, dans cette étude comparative, le but est de rechercher si la méthode VMSE peut permettre, après avoir amélioré le pouvoir de classification d'un unique classifieur, d'améliorer celui d'une combinaison de classifieurs ou si au contraire elle perd de son intérêt. Autrement dit, l'intérêt apporté par l'utilisation des ensembles de réseaux de neurones [Breiman, 1994] peut-il être augmenté en lui conjuguant la nouvelle fonction de coût proposée ?

Conditions expérimentales

Le problème qui nous intéresse ici est un problème de classification à deux classes sur un problème lié au crédit[3]. La base de données disponible possède 690 exemples comprenant chacun 15 attributs auxquels s'ajoute l'étiquette d'appartenance à la classe 1 ou 2. Les véritables noms des attributs ont été effacés pour devenir anonymes, on ne peut donc savoir à quoi ils correspondent. Cette base de données est intéressante parce qu'elle comprend un bon mélange d'attributs continus et discrets dont nous présentons la répartition et la normalisation que nous en avons effectué tableau 4.1. Il y a aussi des données manquantes : 37 exemples (5 %) ont un ou plusieurs attributs manquants (voir tableau 4.2).

TAB. 4.1 - La composition du fichier de données et la normalisation associée.

Attribut	Valeur de départ	Normalisation effectuée
A1	a, b	0.5, 1.0
A2	continue	[0.0,1.0]
A3	continue	[0.0,1.0]
A4	u, y, l, t	0.25, 0.50, 0.75, 1.0
A5	g, p, gg	0.3, 0.6, 0.9
A6	c, d, cc, i, j, k, m	0.07, 0.14, 0.28, 0.35, 0.42, 0.49, 0.56
	v, q, w, x, e, aa, ff	0.63, 0.70, 0.84, 0.91, 0.98
A7	v, h, bb, j, n, z, dd	0.11, 0.22, 0.33, 0.44, 0.55, 0.66, 0.77
	ff, o	0.88, 0.99
A8	continue	[0.0,1.0]
A9	t, f	0.3, 0.7
A10	t, f	0.3, 0.7
A11	continue	[0.0,1.0]
A12	t, f	0.3, 0.7
A13	g, p, s	0.2, 0.5, 0.8
A14	continue	[0.0,1.0]
A15	continue	[0.0,1.0]
A16	attribut de classe	0.1, 0.9
Ax	donnée manquante	-1.0

Rappelons que le but ici est de comparer les résultats obtenus par la technique de Bagging mise au point par Breiman [Breiman, 1994] [Bishop, 1997] et plus particulièrement les résultats affichés dans [Quilan, 1998] avec la nouvelle fonction de coût et la fonction de coût MSE. La répartition des exemples suivant les deux classes est : 307 exemples appartiennent à la classe 1, soit 44.5 %, et 383 à la

[3]On peut trouver ce fichier et d'autre informations sur les conditions de comparaison sur le site : ftp ://ftp.ics.uci.edu/pub/machine-learning-databases/credit-screening/

TAB. 4.2 - Répartition des attributs manquants.

Attribut	Nombre d'attributs manquants
A1	12
A2	12
A4	6
A5	6
A6	9
A7	9
A14	13

classe 2, soit 55.5 %.

Nous avons utilisé la même procédure d'apprentissage que celle énoncée dans [Quilan, 1998], qui est (voir figure 4.13) :

- création de duplications de la base de données de départ par tirage aléatoire uniforme avec remise (la duplication a le même nombre d'éléments que la base de départ) ;
- pour chaque nouvelle base ainsi créée (qu'on appelle D_i) :
  - on subdivise la base en 10 blocs de taille égale, qu'on appelle B_j ;
  - on réalise 10 apprentissages en utilisant 9 des blocs comme ensemble d'apprentissage et le dixième comme ensemble de "test", on appelle RB_j le résultat obtenu en utilisant le bloc j comme ensemble de test ;
  - le résultat final pour cette base est le résultat moyen obtenu sur les 10 apprentissages, qu'on appelle RD_i :

$$RD_i = \frac{1}{10} \sum_{j=1}^{10} RB_j \tag{4.50}$$

- Le résultat final obtenu R_f est fonction du nombre (N) de répliques de la base d'origine utilisées :

$$R_f = \frac{1}{N} \sum_{i=1}^{N} RD_i \tag{4.51}$$

Chaque réseau de neurones possède 15 neurones en entrée, 6 neurones cachés et un neurone en sortie. Nous précisons ici que le critère d'arrêt des différents

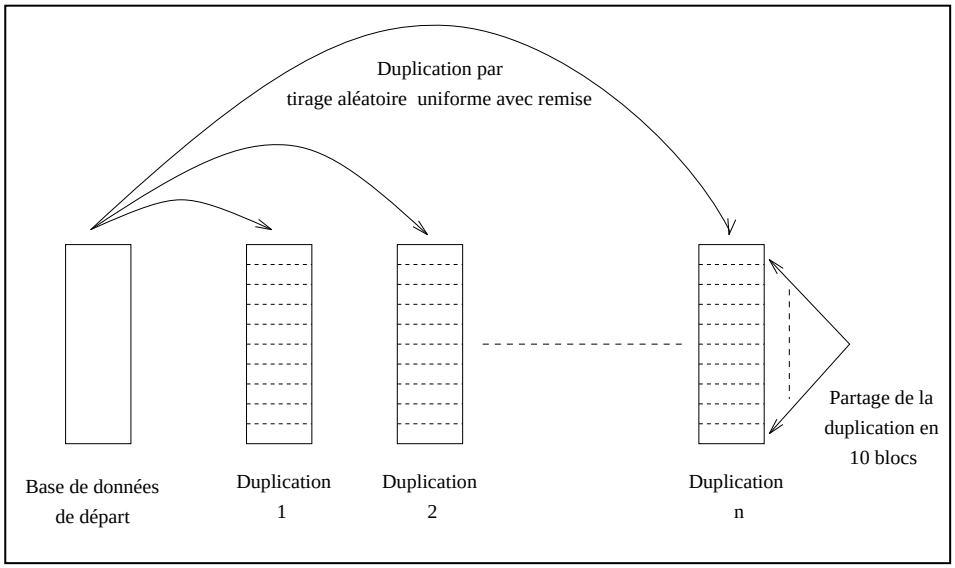

FIG. 4.13 - La "création" des différentes bases d'apprentissage.

apprentissages est basé uniquement sur l'erreur quadratique (tout comme les auteurs de [Quilan, 1998]). Le terme ajouté dans la nouvelle fonction de coût est donc utilisé comme un facteur de régularisation ce qui ne change en rien la loi de modification des poids énoncé en 4.4.1. Son influence à été mesurée pour $\nu = 1$ et $\nu = 10$ (α_{quad}=0.01, β=0.9).

Chaque apprentissage a été stoppé quand le coût sur l'ensemble de test ne diminuait plus depuis 200 itérations. Dans ce cas l'erreur quadratique moyenne, la variance de l'erreur quadratique globale et le pourcentage d'erreur sont calculés sur la meilleure configuration des poids pour cet apprentissage sur l'ensemble de test et sont détaillés ci-après.

Résultats

Les résultats présentés dans les figures 4.14, 4.15, 4.16, 4.17 sont étiquetés 'MSE' pour la fonction de coût n'utilisant que l'erreur quadratique, 'VMSE 1' et 'VMSE 10' respectivement pour $\nu = 1$ et $\nu = 10$.

On précise que pour $\nu = 0.1$ (ou inférieur) les résultats avec la méthode VMSE sont identiques à ceux obtenus avec la méthode MSE. Les valeurs de ν ont été choisies en fonction de l'expérience acquise sur le problème précédent de détection de visages.

Plus précisément on trouve figure 4.14 le résultat obtenu sur l'erreur quadratique moyenne globale, figures 4.15, 4.16 le résultat respectivement sur la variance de l'erreur quadratique pour la classe 1, 2 et figure 4.17 le pourcentage de mal classés en fonction du nombre de répliques utilisées. Sur cette dernière figure,

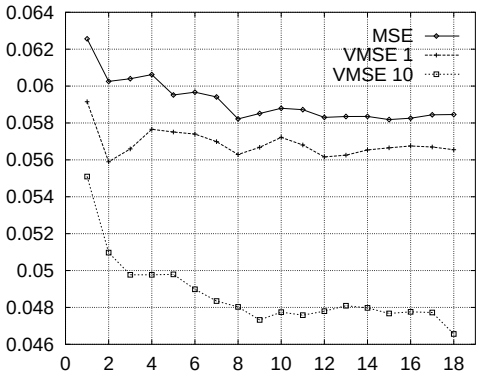

FIG. 4.14 - Résultats obtenus sur l'erreur quadratique moyenne globale en fonction du nombre de répliques.

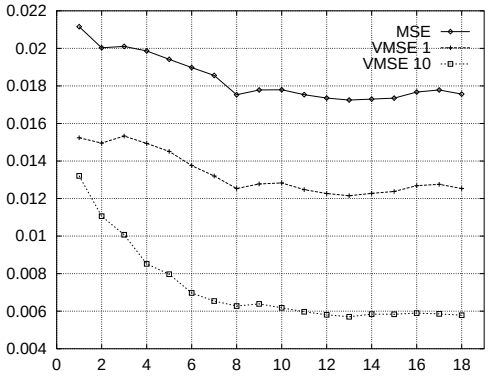

FIG. 4.15 - Résultats obtenus sur la variance de l'erreur quadratique en fonction du nombre de répliques pour la classe 1.

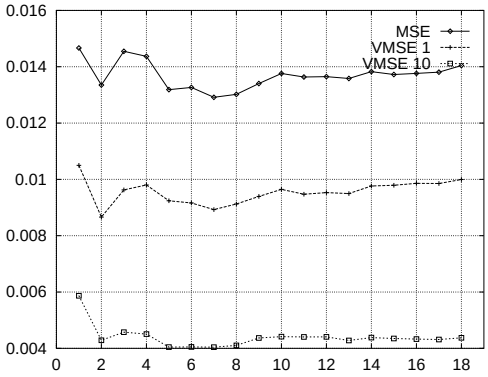

FIG. 4.16 - Résultats obtenus sur la variance de l'erreur quadratique en fonction du nombre de répliques pour la classe 2.

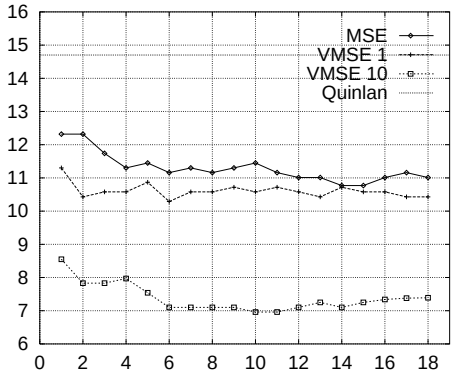

FIG. 4.17 - Résultats obtenus sur le pourcentage de mal classés en fonction du nombre de répliques.

on indique le meilleur résultat obtenu par Quilan [Quilan, 1998] (ce dont on ne dispose pas pour les trois premiers résultats affichés). Dans le cas de l'utilisation de la fonction de coût MSE ($\nu = 0$), nous n'avons pas exactement retrouvé les résultats de Quilan ce qui peut provenir d'une erreur de notre part quant à la compréhension des conditions expérimentales. Les résultats trouvés ici avec la méthode MSE sont néanmoins meilleurs et constituent donc de bonnes valeurs de comparaison.

On peut aussi noter qu'un des résultats de Breiman [Breiman, 1994] montrant de façon expérimentale que le nombre de répliques nécessaire mais suffisant est de l'ordre de 10, après quoi les performances ne sont que très peu améliorées, est retrouvé. Ceci est plus particulièrement visible sur les figures 4.14, 4.15.

Discussion

On s'aperçoit que les résultats obtenus avec la méthode 'VMSE' sont meilleurs que ceux obtenus avec la fonction 'MSE' tant sur l'erreur quadratique moyenne globale que sur sa variance et sur le pourcentage de mal classés. On peut conclure que la nouvelle fonction de coût se conjugue très bien avec la technique du Bagging. Le contrôle de la forme de la distribution des erreurs, réalisé grâce à une minimisation de la variance de l'erreur quadratique, permet une meilleure généralisation. Cette amélioration déjà constatée sur la première étude comparative n'utilisant qu'un seul réseau de neurone est conservée au fur et à mesure que le nombre de réseaux est augmenté. On voit bien sur les différentes figures de résultats que l'amélioration apportée en fonction du nombre de répliques utilisées est à peu près identique pour les trois courbes. L'amélioration des performances obtenues avec une seule base est conservé lorsque le nombre de répliques utilisées augmente.

4.4.4 Etude comparative : Prédiction

Introduction

L'idée d'utiliser les réseaux de neurones artificiels pour la prévision de séries temporelles date des années 60 avec le modèle linéaire adaptatif de Widrow appliqué à la météorologie [Hu, 1964]. Mais, c'est la mise au point de l'algorithme de rétro-propagation du gradient d'erreur qui a permis le développement des réseaux de neurones pour la prévision. Les premiers travaux sont dûs à Lapedes et Farber en 1987 [Lapedes et Farber, 1987]. Ils ont montré de façon empirique que les réseaux de neurones permettent de modéliser et prévoir à court terme des séries temporelles non-linéaires, générées par des phénomènes déterministes. Depuis, les prévisions par réseaux de neurones sur des séries réelles sont présentes dans un nombre grandissant et varié de domaines : prévision de la consommation d'eau pour la Lyonnaise des eaux [Canu et al., 1990], prévision de la consommation d'électricité [Park et al., 1991], ect... On constate également l'explosion des prévisions consacrées aux séries économiques et financières : prévision de la nature cyclique de l'économie du Royaume Uni et détection de ses points de rupture [Hoptroff et al., 1991], modélisation des performances d'une entreprise pour l'évaluation des risques et des bénéfices [Hoptroff, 1993], contrôle et gestion des stocks.

La série temporelle

Ici, nous mettrons en œuvre des perceptrons multicouches simples pour la prévision d'une série temporelle ionosphérique. L'ancien service des prévisions

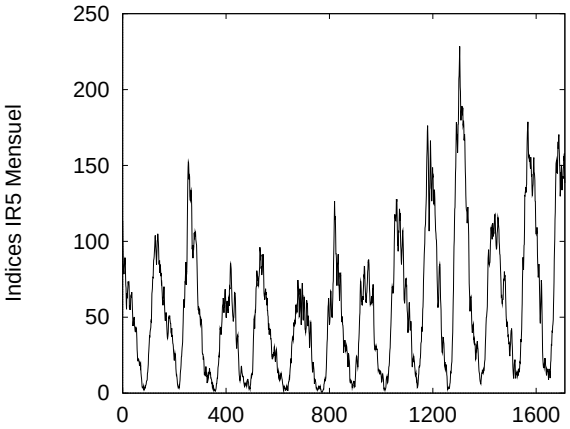

FIG. 4.18 - Indices IR5 de 1849 à 1991.

ionosphériques du CNET avait adopté comme indice d'activité solaire l'indice IR5, moyenne glissante sur cinq mois, non-centrée, du nombre de tâches solaires (voir figure 4.18) :

$$IR5(m) = \frac{1}{5}(R(m-3) + R(m-2) + R(m-1) + R(m) + R(m+1)) \quad (4.52)$$

où $IR5_m$ est l'indice rattaché au mois m et R_m la valeur moyenne du nombre journalier de tâches solaires pour le même mois. Cet indice a été préféré à l'indice R_{12} moyenné centré sur 12 mois du nombre de taches solaires, utilisé par d'autres centres de prévision [Bourdila et Hanbaba, 1984]. Nous disposons des valeurs de l'indice IR5 de 1849 a 1991 calculées, pour tous les mois, à partir des valeurs journalières du nombre de tâches solaires (figure 4.18).

Le critère de comparaison

Pour mesurer les performances des réseaux que nous allons construire, à partir d'une série temporelle x_t, il faut définir un critère de mesure des performances. La littérature statistique dispose d'un certain nombre de critères basés sur l'erreur entre la prévision calculée et la valeur réelle de la série x_t, mesurée au temps t. Pour comparer les performances des réseaux sur des séries différentes, le critère de variance moyenne relative (ARV) a été choisi :

$$ARV = \frac{\sum_{t=1}^{N}(x(t) - \hat{x(t)})^2}{\sum_{t=1}^{N}(x(t) - \hat{\mu})^2} \quad (4.53)$$

où $\hat{x(t)}$ la valeur estimée de $x(t)$ et $\hat{\mu}$ est la moyenne estimée de la série.

La division par N, nombre de données de l'ensemble de test, permet de rendre la mesure indépendante du nombre de valeurs de l'ensemble de prévision. La normalisation par la variance estimée des données permet d'enlever la dépendance sur leur aspect dynamique. Cette normalisation implique que si, à chaque instant t, la prévision de $x(t)$ est estimée par la moyenne estimée de la série, on obtiendra ARV = 1, par contre si la prévision est exacte, c'est-à-dire si $\hat{x(t)}$=$x(t)$ alors ARV = 0. Un modèle sera donc d'autant meilleur que le critère ARV sera proche de 0. C'est de cette manière que sont présentés les résultats dans [Fessant, 1995] qui sont parmi les meilleurs que nous connaissons à ce jour.

Le choix de la taille et de la constitution du vecteur d'entrée

Soit une série temporelle (x_1,..., x_M), de taille finie (avec des données mesurées à des intervalles de temps réguliers), pour laquelle on ne connaît pas les lois du système qui génère les données. Takens [Takens, 1980] a montré qu'il existe, sous certaines conditions, une fonction f et deux entiers τ et d permettant de prédire exactement une valeur future $x(t)$. Cette valeur $x(t)$ est reliée aux $d\tau$ valeurs précédentes de la série, par l'intermédiaire d'une fonction f :

$$x(t) = f(x(t-\tau), x(t-2\tau), ...x(t-d\tau)) \tag{4.54}$$

où d est appelé dimension de reconstruction (ou embedding dimension) et τ est un retard. Takens a également montré qu'il existe une limite supérieure à la dimension de corrélation : $d < 2h + 1$, où h est la dimension de l'attracteur sur lequel évoluent les données.

Les conditions permettant d'obtenir ceci supposent que l'on ait déterminé la bonne fonction f, et que le système qui génère les données soit déterministe. Dans ce théorème cependant, aucune information n'est donnée pour permettre le calcul de la fonction f, ni des paramètres d et τ. Cependant, il existe différentes méthodes heuristiques pour calculer le paramètre d, (pour une revue détaillée se reporter à l'article d'Abarbanel et al [Abarbanel et al., 1993]). En théorie, le problème du choix de τ ne se pose pas : pour une série purement déterministe et pour un nombre suffisant de données, il existera toujours une valeur de τ qui vérifiera la relation de dépendance : $x(t) = f(x(t-\tau), x(t-2\tau), ...x(t-d\tau))$. Cependant, dans la plupart des cas l'ensemble des données étant de taille limitée et celles-ci étant bruitées il faut fixer une valeur de τ. Une règle pour se fixer une limite à τ est de choisir le premier zéro de la fonction d'auto-corrélation de la série ou le premier minimum de la fonction d'information mutuelle [Abarbanel et al., 1993]. Une étude de ces différentes méthodes a conduit Fessant [Fessant, 1995] à choisir $d = 40$ pour τ=1, ce que nous utiliserons comme vecteur d'entrée.

La normalisation des données

Pour toutes les expériences réalisées dans cette partie, les données d'apprentissage et de test sont centrées par rapport à la moyenne de toute la série, puis normalisées entre [-1,1]. Cette normalisation s'obtient en divisant toutes les valeurs de la série centrée par la plus grande valeur absolue des valeurs de cette série. Ainsi, toutes les valeurs sont ramenées dans les bornes de la fonction sigmoïdale utilisée dans nos réseaux de neurones.

On pose μ la moyenne des T valeurs de la série :

$$\mu = \frac{1}{T}\sum_{t=1}^{T} x(t) \tag{4.55}$$

Les valeurs normalisées s'écrivent :

$$x(t)' = \frac{x(t) - \mu}{max|x(t) - \mu|} \tag{4.56}$$

Les données normalisées seront appliquées directement en entrée des différents réseaux sans subir aucun autre pré-traitement.

Les conditions expérimentales

Les 1477 premières valeurs de la série seront utilisées pour l'apprentissage, les 238 dernières pour le test. Ces dernières valeurs couvrent la période de février 1972 à novembre 1991. Les réseaux que nous avons utilisés sont des perceptrons multicouches dont l'architecture est : 40 unités d'entrée, 23 neurones pour la couche cachée et 6 neurones pour la couche de sortie. Ces six sorties correspondent aux prévisions : $x_{t+1}, x_{t+2}, x_{t+3}, x_{t+4}, x_{t+5}, x_{t+6}$ En effet, dans le cas de la prévision de l'indice IR5, l'objectif de prévision est la valeur à 6 pas de temps. Cette contrainte est imposée pour des raisons pratiques qui sont l'édition et la diffusion des bulletins de prévisions. Nous prenons ici les même conditions de travail que Fessant ([Fessant, 1995]) avec qui nous allons comparer nos résultats qui ne concernent que la prévision à 6 mois (donc x_{t+6}).

Résultats

Les résultats obtenus sont présenté figures 4.19, 4.20, 4.21. Ils présentent l'erreur quadratique moyenne globale, la variance de l'erreur quadratique, et l'ARV (variance moyenne relative) sur l'ensemble de test en fonction de ν sur l'ensemble

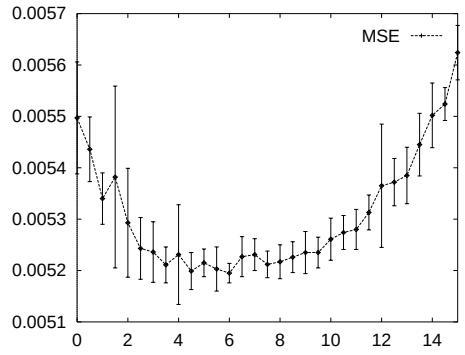

FIG. 4.19 - Résultats obtenus sur l'erreur quadratique moyenne globale en fonction de ν sur l'ensemble de "test" (Moyennes et intervalles de confiance sur 20 apprentissages).

de test. On constate que l'erreur quadratique moyenne, la variance de l'erreur quadratique moyenne et l'ARV sont respectivement améliorés de 5.5 %, de 25.3 %, et de 6.1 % pour ν=6. La nouvelle fonction de coût VMSE exhibe encore ici une meilleure généralisation que la fonction de coût MSE dans un problème d'approximation de fonction.

Pour $\nu = 1$ les deux fonctions de coût présentent des performances similaires. Pour $\nu \in [2 : 8]$ la nouvelle fonction de coût réduit les variances jusqu'à obtenir un gain de 6 % sur l'ARV par rapport à la fonction de coût standard et ce avec un intervalle de confiance du même ordre. Par contre, pour $\nu > 8$, les résultats obtenus se dégradent. Là encore si α'_{var} est trop grand, comparé à α_{quad}. La minimisation de la variance de l'erreur quadratique au cours de l'apprentissage empêche la minimisation de l'erreur quadratique et le réseau de neurones renvoie toujours la même valeur de sortie, pour laquelle la variance de l'erreur est alors minimale.

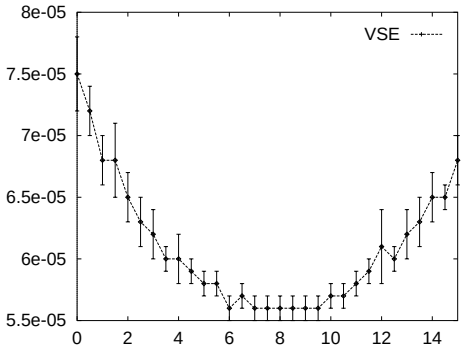

FIG. 4.20 - Résultats obtenus sur la variance de l'erreur quadratique en fonction de ν sur l'ensemble de "test" (Moyennes et intervalles de confiance sur 20 apprentissages).

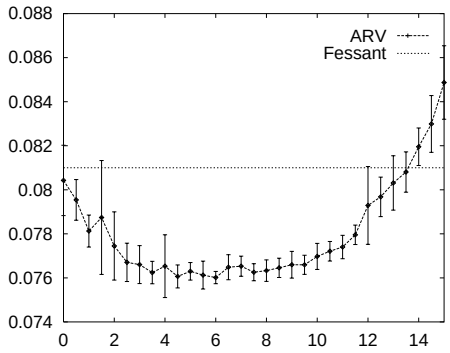

FIG. 4.21 - Résultats obtenus sur l'ARV en fonction de ν sur l'ensemble de "test" (Moyennes et intervalles de confiance sur 20 apprentissages).

4.4.5 Conclusion

On a présenté dans cette section une nouvelle méthode destinée à améliorer les performances en généralisation des perceptrons multicouches utilisés en tant que réseaux discriminants et approximateurs de fonctions. On a montré clairement comment modifier le critère d'apprentissage afin de contrôler la distribution des erreurs au cours de l'apprentissage. Cette méthode permet de minimiser à la fois les erreurs de classification et les erreurs d'estimation par une minimisation de la variance de l'erreur quadratique.

Cette méthode utilisée en tant que fonction de régularisation est très facilement implémentable même dans les cas où il y a plus d'un neurone de sortie. En effet il suffit de calculer la MSE sur chaque neurone de sortie du réseau. Par exemple pour un problème à cinq classes où on utilise cinq neurones de sortie il faut calculer cinq MSE. Ces MSE ayant été calculées, le calcul des cinq gradients G_{var} est alors immédiat conformément à l'équation 4.30. Les modifications à apporter dans un programme d'apprentissage où la méthode MSE était utilisée, sont alors réellement minimes (une application à des réseaux possédant un grand nombre de poids est présentée Annexe A).

Reste à savoir comment choisir de façon simple et efficace les paramètres α_{quad}, α_{var} et β. Pour β nous renvoyons le lecteur à [Plaut et al., 1986] du fait que nous avons remarqué que nombre d'utilisateurs de réseaux de neurones introduisent ce terme dans la règle de modification des poids alors que d'autres n'y trouvent pas un grand intérêt, expérimentalement parlant. Pour ce qui est de α_{quad} et de α_{var}, le critère de choix, comme nous l'avons montré, est imposé par leur ratio illustré dans cette section par la variable ν.

Trois études comparatives ont été présentées sur trois problèmes différents. Les résultats expérimentaux obtenus améliorent sensiblement l'état de l'art :

– pour le problème de détection de visages : baisse de 37 % du taux de fausse

alarme (pour un taux de détection de 99.5 %) pour $\nu = 1$;

- pour le problème de crédit : amélioration du taux de bien classé de 30.4 % pour ν=10 ;
- pour la série temporelle : amélioration de l'ARV de 6.1 % pour $\nu = 6$.

L'idée de contrôler la forme de la distribution des erreurs au cours de l'apprentissage afin d'obtenir une meilleure généralisation se trouve validée de manière expérimentale. Nous présentons, dans le chapitre suivant, une application des réseaux de neurones et de la nouvelle méthode explicitée ci-dessus, à l'estimation des délais dans un réseau ATM dans le but de réaliser un CAC adaptatif et ceci pour la capacité de transfert ABT-DT.

Chapitre 5

Estimation des temps de blocage dans un lien ATM

On s'intéresse dans ce chapitre aux périodes de congestion survenant dans un lien ATM, donc dans le nœud qui nourrit ce lien quand des connexions sont multiplexées en boucle ouverte et ce dans le cadre d'un procédé de type REM (voir 3.1.3). Cette étude est orientée de manière à être utilisable dans une procédure de contrôle d'admission des connexions.

Dans la première section nous préciserons un certain nombre de points concernant la capacité de transfert ABT dont nous n'avons fait qu'une brève introduction en 2.7.

Dans la deuxième section, pour étudier de manière quantitative le phénomène de congestion, nous introduisons les paramètres de qualité de service qui vont plus particulièrement nous intéresser. La politique d'utilisation de ces paramètres dans le cadre d'une procédure de contrôle d'admission des connexions sera aussi présentée pour la capacité de transfert ABT.

La troisième section présentera trois méthodes destinées à estimer ces paramètres de qualité de service à partir de mesures du trafic. La quatrième section détaillera les bases de données de trafic qui serviront dans la section suivante à réaliser une comparaison entre les méthodes d'estimation.

On cherchera plus particulièrement à vérifier si l'utilisation des réseaux de neurones entraînés sur des trafics qualifiés de pire cas peuvent correctement généraliser sur d'autres types de trafics.

Les différents travaux cités dans ce chapitre ont fait l'objet de publications [Lemaire, 1997; Lemaire et Clérot, 1999; Lemaire et al., 1999b].

5.1 La capacité de transfert ABT

5.1.1 Introduction

La recommandation I.371 de l'UIT-T [ITU, 1996a; ITU, 1997] et la spécification 4.0 (section Trafic) de l'ATM Forum [Forum, 1996] ont été achevées courant 1996, après quatre années de travail. A travers la spécification d'un certain nombre de capacités de transfert ATM, ces deux documents marquent une avancée importante en matière de gestion de trafic dans les réseaux ATM.

De nombreuses études ont montré qu'il pouvait être dangereux pour un réseau de laisser entrer un volume important de trafic imprévisible en excès par rapport aux capacités de transmission, à cause notamment des phénomènes de congestion de niveau rafale [Roberts, 1991]. C'est pourquoi la stratégie du CNET en matière de gestion de trafic a été de promouvoir des méthodes de réservation rapide de bande passante, en particulier par l'intermédiaire des protocoles de réservation rapide [Boyer et Tranchier, 1992; Guillemin, 1999]. Il est clair que, eu égard à la transmission d'information, deux principes peuvent être envisagés :

- soit la bande passante nécessaire est préalablement réservée dans le réseau avant la transmission d'informations, ceci donne naissance au protocole de réservation rapide avec transmission retardée ; le délai de réservation est alors clairement une latence incompressible du mécanisme ;
- soit le bloc d'informations est transmis sans attendre l'accord du réseau, il essaie alors de réserver de proche en proche dans les nœuds successifs du réseau la bande passante nécessaire à sa transmission ; dans ce cas il n'y a pas de délai de réservation mais le risque de perdre tout un bloc de données existe.

Après un affinage progressif des protocoles de réservation rapide, notamment en matière de probabilité de réussite de transfert de bloc, la capacité de transfert de blocs ATM (ABT pour *ATM Block Transfer*) est née avec les variantes transmission retardée (ABT-DT pour *ATM Block Transfer with Delayed Transmission*) et transmission immédiate (ABT-IT pour *ATM Block Transfer with Immediate Transmission*).

5.1.2 Le bloc ATM et son transfert

Le concept de bloc repose sur l'observation suivante : quand on examine le trafic engendré par un certain nombre d'applications, on s'aperçoit que le débit crête varie par paliers, c'est-à-dire que le débit instantané de l'application est constant par morceaux (exemples : les périodes d'activité et d'inactivité d'une

source). Cette remarque vaut aussi pour aussi pour les conduits virtuels (VP) qui contiennent plusieurs canaux virtuels (VC).

Cette notion de palier étant définie il n'y a qu'un pas à franchir pour concevoir un procédé de multiplexage qui consiste à négocier les ressources d'une connexion palier par palier. C'est précisément le but des protocoles de réservation rapide, aussi bien sous leur forme transmission immédiate que retardée. La caractéristique principale de ces protocoles est qu'ils font usage d'une nouvelle entité de contrôle, à savoir la cellule RM (pour Ressource Management). Pour négocier les ressources d'un palier, un certain nombre de cellules RM doivent être échangées entre l'utilisateur et le réseau, suivant le mode de transmission. Un bloc ATM est alors défini de la manière suivante (voir figure 5.1) :

- Un bloc ATM est formé d'un groupe de cellules consécutives d'une connexion délimité par deux cellules RM, une en tête juste avant la première cellule du bloc et une autre en queue, juste derrière la dernière cellule du bloc. Les cellules RM ne font pas partie du bloc

La délimitation des blocs ATM peut être effectuée soit par la source elle-même (on parle alors de source ABT native), soit par le réseau qui est alors responsable de l'assemblage et de la délimitation des blocs ATM.

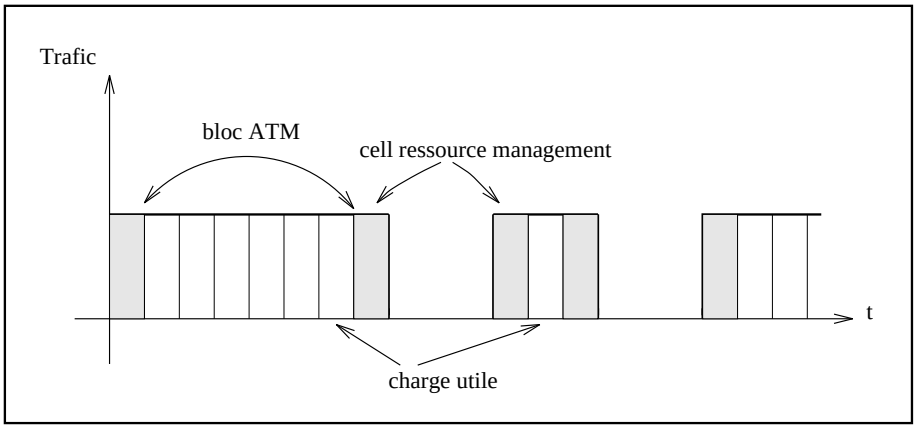

FIG. 5.1 - Le trafic ABT.

La notion de bloc ATM étant introduite, il est naturel d'envisager une méthode de multiplexage qui manipule des blocs ATM et qui consiste à réserver des ressources sur la base des blocs ATM pour assurer le transfert à travers le réseau. Ceci donne naissance à la capacité de transfert de blocs ATM. Comme nous l'avons mentionné plus haut, il existe deux modes de transmission, à savoir retardée (ABT-DT) ou immédiate (ABT-IT).

Dans le cas d'ABT-DT, toutes les conditions pour assurer le transfert d'un bloc avec une qualité de transfert similaire à celle de la capacité DBR sont réunies avant qu'une source ne soit autorisée à émettre. C'est pour cela que la capacité ABT-DT peut être qualifiée de DBR par morceaux. Dans le cas de la capacité

ABT-IT, la source émet le bloc juste après l'émission de la cellule RM de tête, chargée de la réservation des ressources dans le réseau de proche en proche. Les cellules d'un bloc accepté sont multiplexées sur la base du débit de bloc BCR (pour Block Cell Rate). En effet, rappelons que le mode ABT est destiné à garantir la qualité de service au niveau du bloc plutôt qu'au niveau de la cellule : la source négocie un PCR_{max} pour toute la durée de l'appel et définit un PCR pour le bloc à venir (BCR).

Chacune de ces capacités peut être rigide ou élastique. En mode rigide, le débit demandé est soit accepté soit refusé par le réseau. Dans le cas d'un refus la demande peut être réitérée plus tard ou changée (par exemple diminuée). En mode élastique, le débit demandé est soit accepté, soit refusé, mais le réseau peut alors proposer un autre débit à l'utilisateur. Le bloc peut aussi être stocké et s'il le faut lissé à un BCR inférieur. L'élasticité d'un bloc est spécifié par un bit de la cellule RM.

5.1.3 Contrat de trafic et qualité de service

Le contrat de trafic négocié entre l'utilisateur et le réseau à l'établissement d'une connexion ABT spécifie un certain nombre de paramètres qui permettent l'allocation de ressources à la connexion et de configurer les mécanismes de contrôle. La liste des paramètres est la suivante :

- le débit crête du flux (hormis le flux de signalisation et de maintenance ainsi que celui des cellules RM), qui fixe une borne supérieure sur les BCR possibles ;
- le débit crête du flux de signalisation et de maintenance OAM (pour Organization And Maintenance). Ce paramètre est optionnel ;
- le couple de paramètres statistiques (SCR, MBS), qui caractérise le débit moyen de la source ainsi que son caractère de sporadicité ;
- le débit crête des cellules RM de requête de bande passante ;

Les paramètres ci-dessus sont dits statiques et il n'est pas envisagé à l'heure actuelle de les renégocier en cours d'appel pour éviter les interférences entre la procédure ABT et la signalisation ou la gestion de réseau. En matière de gigue, une tolérance de gigue est associée à chacun des quatre flux cités ci-dessus.

Le couple de paramètres (SCR, MBS) introduit la notion de bande passante garantie. Il permet d'améliorer les performances de la capacité ABT vis à vis de la probabilité de destruction d'un bloc ATM. Cette probabilité est définie lorsqu'un couple (SCR, MBS) est négocié à l'établissement de la connexion. La bande passante est reliée au montant des ressources réservées dans le réseau et

sa valeur est égale au SCR négocié. La moyenne temporelle du montant des ressources (en terme de débit) qui peuvent être potentiellement réservées dans le réseau est au moins égale à SCR. De plus, tant que le trafic émis par une source est conforme au couple (SCR, MBS) négocié, la probabilité qu'un bloc soit détruit dans le réseau est plus petite qu'une certaine valeur spécifiée à l'établissement de la connexion.

La qualité de service pour ABT-DT comme pour ABT-IT s'exprime à la fois au niveau cellule et au niveau bloc. Au niveau cellule les exigences en matière de qualité de service sont équivalentes à celle de la capacité de transfert DBR à savoir :

- tant que le trafic est conforme au BCR négocié les engagements en termes de perte de cellule (CLR et/ou CLP) et délai de transfert sont respectés ;
- la qualité de service est assurée à toutes les cellules conformes ;
- si certaines cellules sont non conformes, le réseau a la liberté de déclarer la connexion non conforme et de ne pas respecter la qualité de service négociée ; s'il décide de ne pas rejeter la connexion, la qualité de service n'est assurée qu'à un volume de cellules satisfaisant aux tests de conformité.

Des objectifs de qualité de service au niveau bloc sont spécifiés quand un couple (SCR, MBS), différent de zéro, est négocié à l'établissement de la connexion. Autrement seul le BCR est pris en compte. Si le trafic devient non conforme ou si le couple (SCR, MBS) négocié est nul alors le réseau ne garantit aucun objectif de qualité de service au niveau bloc. Ces objectifs de qualité de service au niveau bloc sont :

- pour la capacité ABT-DT, une estimation du temps d'attente du bloc avant qu'ils soit accepté (transmis) par le réseau ;
- pour la capacité ABT-IT, un objectif en terme de taux de perte de blocs (ces derniers pouvant être détruits si un seul élément du réseau ne les accepte pas au cours de la négociation de proche en proche).

On voit que pour la capacité ABT le réseau va devoir chercher à dimensionner les temps d'attente et/ou de blocage que vont subir les blocs transmis. Ce dimensionnement fait l'objet de la section suivante.

5.2 Les temps de blocage dans un lien ATM

5.2.1 Introduction cadre

Dans cette section on s'intéresse au débit instantané $B(t)$ créé par la superposition de différentes connexions sur un lien ATM. La perte due au multiplexage est généralement estimée par le coefficient d'écrêtage, ξ donné par :

$$\xi = \frac{E[(B(t) - B_{max})^+]}{E[B(t)]} \tag{5.1}$$

où B_{max} est la capacité, la bande passante maximale disponible, du lien de transmission. L'avantage de cette analyse est sa simplicité pour déterminer la probabilité de perte de cellules. En effet si on considère le multiplexage de N sources on/off identiques, chacune ayant la probabilité p d'être active, alors, en prenant leur débit crête comme unité de débit, $B(t)$ suit une loi binomiale et on a :

$$\xi = \frac{1}{Np} \sum_{k \geq B_{max}}^{N} \left(\begin{array}{c} N \\ k \end{array} \right) p^k (1-p)^{N-k} \tag{5.2}$$

L'analyse faite ci-dessus est valable à un instant arbitraire dans le temps mais ne reflète en rien ce qui se passe lors d'une période de congestion du lien. En effet, il n'est pas incompatible d'avoir une probabilité de perte très faible et certaines périodes de congestion très longues. De même, on peut avoir des périodes de congestion courtes avec une quantité importante d'informations perdue durant ces périodes.

Pour combler en partie cette lacune, on s'intéresse dans ce chapitre à la durée des périodes de congestion du lien de transmission quand des connexions sont multiplexées en boucle ouverte. Cette étude étant orientée de manière à être utilisable plus tard dans une procédure de contrôle d'admission des connexions, on s'intéressera à la durée des périodes de congestion au-dessus d'une fraction du lien concerné.

On note B_{max} la bande passante maximale disponible sur un lien ATM. La décision d'acceptation, pour un appel ayant un débit crête déclaré à $d_{max} = PCR$, sera faite en ajoutant la nouvelle connexion (supposée émettre à d_{max}) au dessus du trafic en cours et en estimant la distribution des temps de blocage au-dessus de B_{max} pour le trafic agrégé : $B(t) + d_{max} \geq B_{max}$. Ceci est équivalent à estimer la distribution des temps de blocage du trafic en cours $B(t)$ au-dessus du seuil $s = B_{max} - d_{max}$.

On définit η, la probabilité que B(t) soit supérieur à s :

$$\eta = Pr(B(t) > s) \tag{5.3}$$

La connaissance des temps de blocage permet d'évaluer la probabilité qu'un nouveau bloc puisse être "inséré" dans la bande passante disponible et le paramètre η permet de connaître la fréquence des périodes de congestion. La connexion est acceptée si η et les temps de blocage sont suffisamment petits.

Ce processus est donc basé sur deux caractérisations, une caractérisation de la nouvelle connexion et une caractérisation du comportement de la bande passante utilisée par le trafic en cours.

Comme la nouvelle connexion est acceptée sur la base de son PCR (les futurs temps de blocage sont évalués comme si la nouvelle source voulait toujours émettre à son débit crête), ce processus est conservatif (prudent, légèrement pessimiste). Cependant, il est basé sur des mesures du trafic réalisées avant la demande de connexion et permet donc de bénéficier du gain opéré par le multiplexage statistique des sources déjà acceptées.

5.2.2 Un temps de blocage

Pour étudier de manière quantitative le phénomène des temps de blocage, on introduit les variables suivantes :

- On appelle un blocage θ un événement tel que $B(t)$, la bande passante utilisée par les sources présentes, est supérieur à un seuil s pour une durée non nulle.

- La durée T de l'événement θ, le temps de blocage, au-dessus de s est définie par (voir figure 5.2) :

$$T = L(\theta) \tag{5.4}$$

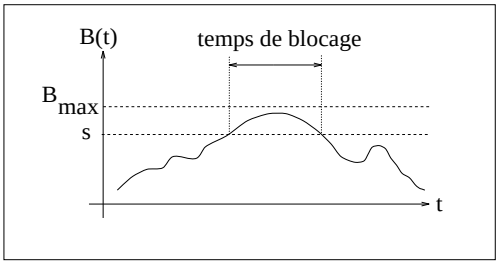

FIG. 5.2 - Illustration graphique d'un temps de blocage au-dessus d'une fraction de la bande passante d'un lien ATM

Dans l'illustration graphique proposée figure 5.2, on ne distingue pas le niveau cellule car on suppose que les rafales sont suffisamment grandes devant la taille des cellules. Ainsi, on peut supposer que l'information arrive continûment pendant une rafale.

5.2.3 La distribution de probabilité des temps de blocage

Maintenant que la durée T, d'un blocage est définie on introduit le paramètre suivant :

$$Pr(L(\theta) \geq T_* | s, \sum_{PCR}) = k(T_*, s, \sum_{PCR}) \tag{5.5}$$

qui représente la probabilité qu'un temps de blocage ait une durée supérieure ou égale à T_* pour un seuil s et connaissant la somme des débits crêtes des connexions produisant $B(t)$, avec :

- θ : le blocage de $B(t)$ au-dessus du seuil s ;
- $L(\theta)$: la durée de ce blocage ;
- $\sum_{PCR}$: la somme des débits crêtes des sources connectées au moment de l'estimation,
- s : le seuil.

Afin d'estimer la distribution de probabilité des temps de blocage il est suffisant d'estimer k pour différents objectifs de probabilité i, définis par :

$$10^{-i} = k(T_i, s, \sum_{PCR}) \tag{5.6}$$

Grâce à quatre objectifs, par exemple pour i=1, 2, 3, 4 pour respectivement T_1,T_2,T_3 et T_4, on peut obtenir une estimation raisonnable de la distribution des durées des temps de blocage (voir figure 5.3).

$T_1(s, \sum_{PCR})$, $T_2(s, \sum_{PCR})$, $T_3(s, \sum_{PCR})$, $T_4(s, \sum_{PCR})$ et $\eta(s, \sum_{PCR})$ sont donc définis par :

$$Pr(L(\theta) \geq T_1 | s, \sum_{PCR}) = 10^{-1} \tag{5.7}$$

$$Pr(L(\theta) \geq T_2 | s, \sum_{PCR}) = 10^{-2} \tag{5.8}$$

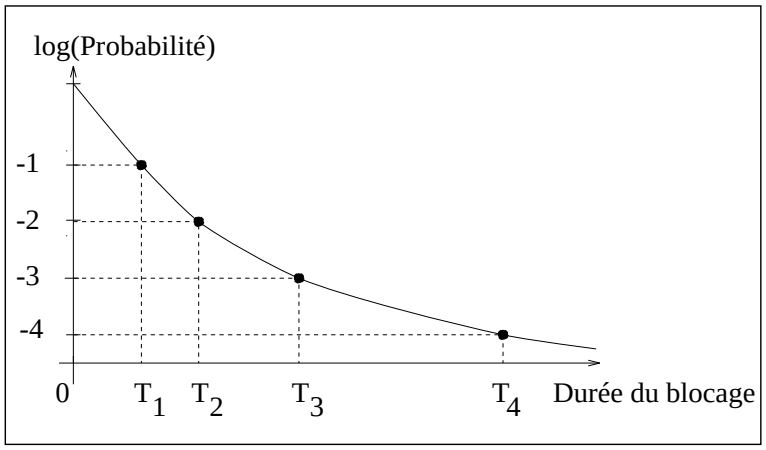

FIG. 5.3 - Distribution des temps de blocage

$$Pr(L(\theta) \geq T_3 | s, \sum_{PCR}) = 10^{-3} \tag{5.9}$$

$$Pr(L(\theta) \geq T_4 | s, \sum_{PCR}) = 10^{-4} \tag{5.10}$$

$$\eta(s, \sum_{PCR}) = Pr(\theta | s, \sum_{PCR}) \tag{5.11}$$

On veut réaliser cette estimation "on-line", aussi on suppose ne posséder qu'un nombre limité (d) d'observations du trafic en cours, que l'on note Φ_{obs} :

$$\Phi_{obs} = B(t-\tau), B(t-2\tau), ..., B(t-d\tau) \tag{5.12}$$

Avec une telle configuration, on suppose donc qu'on ne dispose que des éléments suivants :

- un seuil au-dessus duquel on veut connaître la distribution de probabilité des temps de blocage ;
- une connaissance de la bande passante maximale utilisable au nœud (B_{max}) ;
- une description des appels en cours (la somme des débits crêtes déclarés par les connexions présentes au lien ATM au moment de l'estimation, $\sum_{PCR}$) ;
- une caractérisation du trafic $B(t)$ passant par le lien sur une certaine durée d'observation (Φ_{obs}).

Le but à présent va être d'estimer $\hat{T_1}(s, \sum_{PCR}, \Phi obs)$, $\hat{T_2}(s, \sum_{PCR}, \Phi obs)$, $\hat{T_3}(s, \sum_{PCR}, \Phi obs)$, $\hat{T_4}(s, \sum_{PCR}, \Phi obs)$ et $\hat{\eta}(s, \sum_{PCR}, \Phi obs)$.

5.3 Estimation de la distribution de probabilité des temps de blocage

Dans cette section nous décrivons trois méthodes permettant une estimation des 5 paramètres de qualité de services $T_1(s, \sum_{PCR})$, $T_2(s, \sum_{PCR})$, $T_3(s, \sum_{PCR})$, $T_4(s, \sum_{PCR})$ et $\eta(s, \sum_{PCR})$. La première partie décrit une méthode empirique qui utilise de très longues fenêtres d'observation. Elle servira à fournir des valeurs étalon dans le reste de ce chapitre. La deuxième partie explicite l'utilisation de l'approximation gaussienne pour le calcul des T_i et de η. Enfin la troisième partie décrit une méthode utilisant les réseaux de neurones artificiels. Nous expliciterons dans cette partie comment l'architecture neuronale a été choisie.

5.3.1 L'estimation empirique

Si on possède un nombre d'échantillons du trafic observé suffisamment important alors la probabilité peut être approximée par la fréquence d'apparition. La formulation d'un temps de blocage recherché devient :

$$Pr(L(\theta) \geq T_i | s, \sum_{PCR}) \sim \frac{\sum_{t=0}^{K} (L(\theta) \geq T_i)}{\sum_{t=0}^{K} (L(\theta))} \tag{5.13}$$

et

$$\eta = \frac{\sum_{t=0}^{K} (\theta)}{K} \tag{5.14}$$

avec K la période d'observation et t le pas d'échantillonnage.

Plus K est grand, meilleure sera l'estimation effectuée, aussi nous avons choisi K selon les critères suivants :

- la valeur maximale du temps de blocage (T_{max}) que l'on cherche à estimer sachant qu'au-dessus d'une valeur on est sûr que les excursions sont trop grandes pour accepter tout nouvel appel. Il faut que la valeur de K permette de mesurer T_{max} un nombre de fois suffisant pour estimer qu'il apparaît avec une fréquence de 10^{-k} ;
- la précision que l'on désire avoir sur les T_i. La valeur de K conditionne la valeur approximée du T_i si on doit approximer la probabilité qui lui est attachée à un arrondi près ;
- l'objectif maximal en terme de probabilité sachant que la précision maximale requise pour de tels temps de blocage est de l'ordre de 10^{-4}. Il est inutile en effet d'avoir une valeur de K qui ferait apparaître des valeurs de T_i à 10^{-9}.

Lors de ce calcul empirique de probabilité il est possible de ne pas pouvoir mesurer l'un 5 paramètres de qualité de service que nous nous sommes fixé pour l'estimation de la distribution des temps de blocage. Par exemple pour T_1 on peut, pour un seuil s, avoir mesuré deux durées telles que :

$$Pr(L(\theta) \geq T_i^-|s,n) = 0.102 \tag{5.15}$$

et

$$Pr(L(\theta) \geq T_i^+|s,n) = 0.098 \tag{5.16}$$

mais pas de durée T_i telle que

$$Pr(L(\theta) \geq T_i|s,n) = 0.1 \tag{5.17}$$

Dans ces conditions on utilise une approximation localement linéaire (voir figure 5.4). La procédure est alors la suivante :

– mesurer la valeur la plus proche de T_i par défaut, T_i^- :

$$Pr(L(\theta) \geq T_i^-|s,n) = 10^{-i} + \varepsilon_1 = Pr_i^- \tag{5.18}$$

– mesurer la valeur la plus proche de T_i par excès, T_i^+ :

$$Pr(L(\theta) \geq T_i^+|s,n) = 10^{-i} - \varepsilon_2 = Pr_i^+ \tag{5.19}$$

– estimer T_i tel que :

$$\frac{T_i - T_i^-}{T_i^+ - T_i^-} = \frac{10^{-i} - Pr_i^-}{Pr_i^+ - Pr_i^-} \tag{5.20}$$

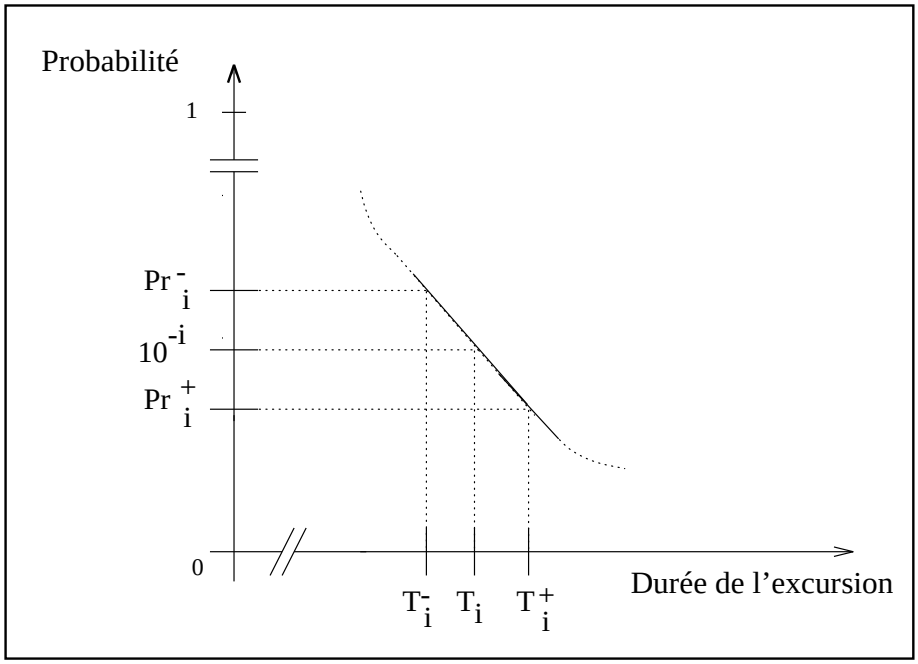

FIG. 5.4 - Approximation localement linéaire des T_i

5.3.2 L'approximation gaussienne

Dans le cas de l'approximation gaussienne on suppose que le trafic observé provient d'un processus échantillonné de manière indépendante et identique à partir d'une distribution gaussienne. Ce trafic a une moyenne et un écart type qui sont estimés par :

$$\hat{\mu_B} = \frac{1}{K}\sum_{t=1}^{K} B(t) \tag{5.21}$$

$$\hat{\sigma_B} = \sqrt{\frac{1}{K-1}\sum_{t=1}^{K}\left(B(t)-\mu_B\right)^2} \tag{5.22}$$

avec K la taille de la fenêtre d' observation Φ_{obs}.

Les paramètres η et T_i sont estimés par :

$$Q(s) = 1 - \int_{s}^{+\infty} \frac{1}{\sqrt{2\pi\hat{\sigma_B}^2}} e^{\left(-\frac{1}{2}\left(\frac{B(t)-\hat{\mu_B}}{\hat{\sigma_B}}\right)^2\right)} \tag{5.23}$$

$$Pr\left(B(t) \geq s\right) = Q(s) \tag{5.24}$$

$$Pr\left(B(t) < s\right) = 1 - Q(s) \tag{5.25}$$

$$Pr\left(B(t) < s, B(t+1) \geq s\right) = Q(s)(1 - Q(s)) \tag{5.26}$$

alors

$$Pr\left(L(\theta) \geq T^*\right) = \sum_{T \geq T^*}^{\infty} \left(Q(s)^T \left[1 - Q(s)\right]^2\right) \tag{5.27}$$

$$= \left[1 - Q(s)\right]^2 \sum_{T \geq T^*}^{\infty} Q(s)^T \tag{5.28}$$

$$= \left[1 - Q(s)\right]^2 Q(s)^{T^*} \sum_{T \geq 0}^{\infty} Q(s)^T \tag{5.29}$$

$$= \left[1 - Q(s)\right]^2 Q(s)^T \frac{1}{1 - Q(s)} \tag{5.30}$$

donc

$$\eta = Q(s) \tag{5.31}$$

$$T_i = \frac{-i}{log_{10}\left(Q(s)\right)} \tag{5.32}$$

5.3.3 L'approche neuronale

Les réseaux de neurones choisis pour toutes les expérimentations décrites dans ce chapitre sont des perceptrons multicouches possédant une couche cachée (voir 4.1.1). Ils sont construits à partir de trois couches de neurones : la couche d'entrée qui reçoit les données, une couche cachée et la couche de sortie qui délivre les sorties désirées.

Choix du vecteur d'entrée

Tous les réseaux de neurones ont des entrées communes qui sont :

- le seuil, s, pour lequel l'estimation des probabilités de blocage est désirée ;
- la somme des débits crête déclarés ($\sum_{PCR}$) ; cette entrée contient l'information qu'au-dessus de la bande passante relié à $\sum_{PCR}$ il ne peut y avoir de blocage (si les contrats de trafics sont respectés) ;

– une caractérisation du trafic passé.

La caractérisation du trafic passé est constituée :

– des valeurs du trafic sur une fenêtre d'observation

$$\Phi_{obs} = (B(t), B(t-\tau), ..., B(t-d\tau)), \tag{5.33}$$

– de la moyenne et de la variance du trafic estimées sur la même fenêtre

$$\hat{\mu_B} = \frac{1}{d} \sum_{t-d\tau}^{t} B(t) \tag{5.34}$$

$$\hat{\sigma_B} = \sqrt{\frac{1}{d} \sum_{t-d\tau}^{t} (B(t) - \hat{\mu_B})^2} \tag{5.35}$$

où τ est un délai et d est la taille du vecteur d'entrée Φ_{obs}.

Les meilleures méthodes utilisées pour sélectionner τ sont basées sur l'hypothèse que deux valeurs successives du vecteur d'entrée doivent être choisies de manière à maximiser l'information contenue dans Φ_{obs}. Par exemple on peut choisir le premier zéro de la fonction d'auto-corrélation. Le choix de τ a été fait pour le cas du trafic homogène présenté dans la section 5.4.2 et pour ce dernier l'analyse de sa fonction d'auto-corrélation nous a amené à choisir τ=1.

La valeur de d est liée à la dimension de reconstruction (voir 4.4.4). En utilisant une technique par dichotomie[1] nous avons choisi d=80 et le meilleur nombre de neurones pour la couche cachée comme étant 18 ; cependant la précision de ces valeurs n'est pas cruciale.

Les entrées $\hat{\mu_B}$ et $\hat{\sigma_B}$ permettent d'introduire l'information que la somme des débits crêtes peut ne pas être toujours la borne supérieure pour laquelle il ne peut y avoir de blocage. Par exemple dans le cas où les sources n'émettent jamais à leur débit crête. Ces deux entrées ont permis d'améliorer les résultats de [Lemaire et al., 1999b] par rapport à [Lemaire et Clérot, 1999].

Choix de l'architecture et de la fonction de coût

Les différentes autres entrées des réseaux de neurones ainsi que l'architecture choisie après de nombreux essais sont présentées figures 5.5 et 5.6. Cette architecture est une combinaison de 5 réseaux de neurones.

[1] On essaye différentes valeurs pour le nombre de neurones d'entrée et cachés, l'arrêt de l'apprentissage étant effectué au moyen de la technique nommée "split-sample" voir 4.3.2.

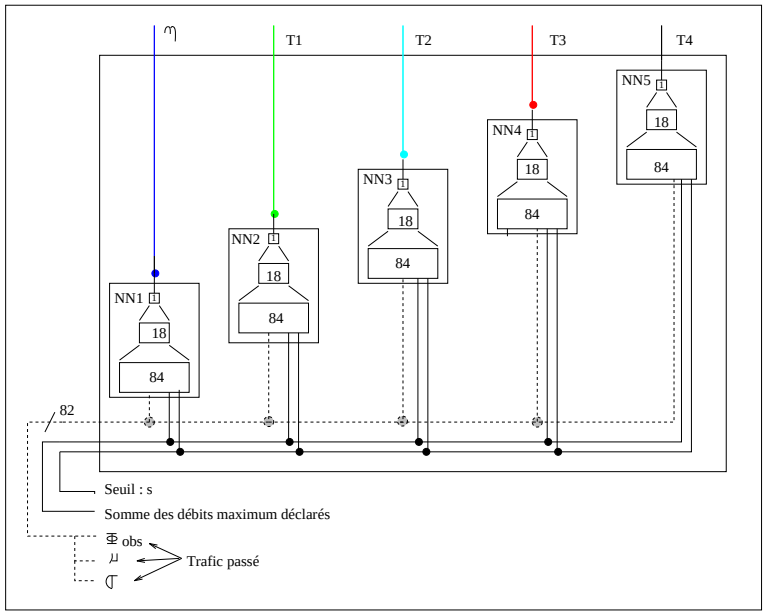

FIG. 5.5 - Architecture neuronale sans cascade pour l'estimation de la distribution de probabilité des temps de blocage dans un lien ATM.

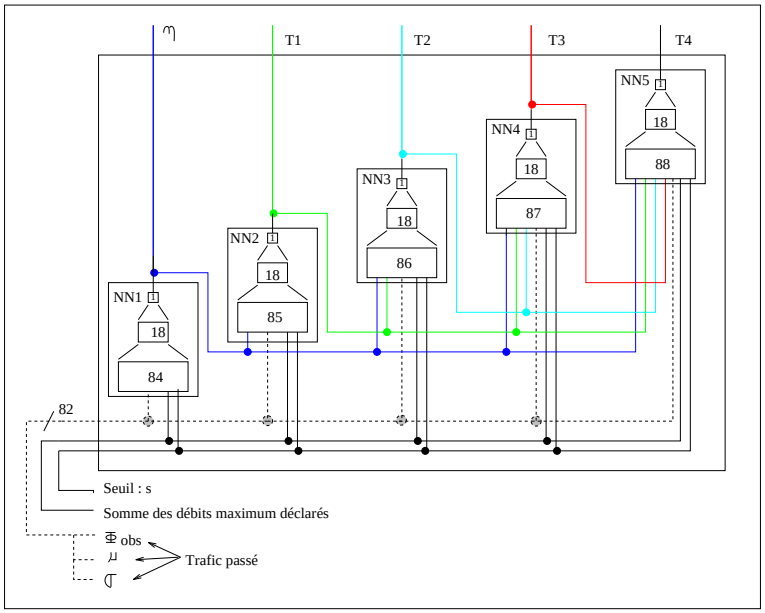

FIG. 5.6 - Architecture neuronale avec cascade retenue pour l'estimation de la distribution de probabilité des temps de blocage dans un lien ATM.

On présente figure 5.5 une architecture neuronale dite sans cascade où les 5 paramètres de qualité de service sont estimés indépendamment. Pour la figure 5.6 on voit qu'il s'agit d'une architecture neuronale où les réseaux de neurones sont

organisés en cascade. En effet il nous a semblé plus facile d'estimer par exemple T_2 si on possède déjà une estimation de T_1.

On présente figure 5.7 la distribution des erreurs commises (pour la signification du graphique voir 5.5.1) avec et sans la nouvelle fonction de coût pour le cas sans cascade et avec la nouvelle fonction de coût pour le cas avec cascade (la nouvelle fonction de coût est utilisée en régularisation voir 4.4.3). Ceci dans le cas du trafic homogène présenté en 5.4.2 mais dans le cas où $B_{max} = \infty$ [Lemaire, 1997]. On constate sur cette figure l'amélioration apportée par la nouvelle fonction de coût sur la répartition de la distribution des erreurs et l'amélioration apportée par la cascade sur la symétrie de la distribution.

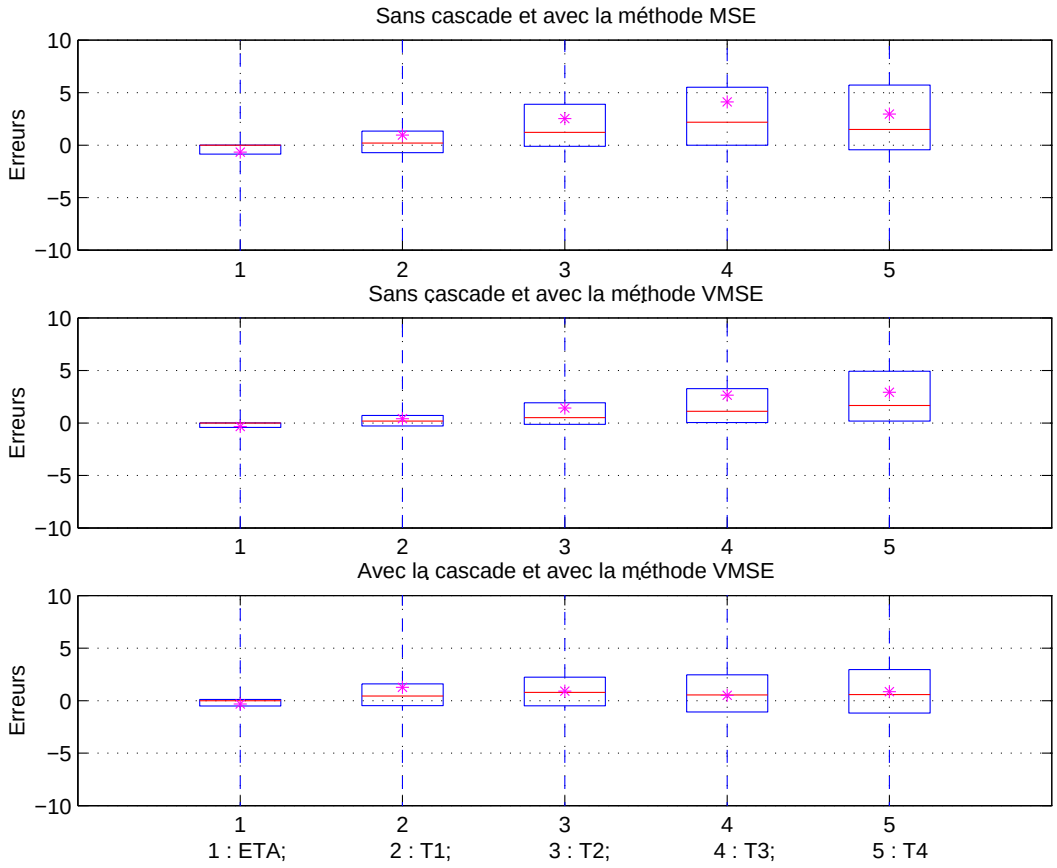

FIG. 5.7 - Représentation graphique de la distribution des erreurs commises sur la base de données de trafics homogènes pour $B_{max} = \infty$: En haut dans le cas sans cascade avec la méthode MSE (ν=0), Au milieu dans le cas sans cascade avec la méthode VMSE (ν=1), En bas dans le cas avec cascade avec la méthode VMSE (ν=1).

Précisons que, dans le cas de la cascade, chacun des réseaux de neurones a été entraîné individuellement. Un apprentissage de NN_1 a été tout d'abord effectué.

Puis NN_2 a été entraîné à l'aide des estimations réalisées par NN_1 et ainsi de suite jusqu'à NN_5.

Le tableau 5.1 présente une comparaison réalisée avec et sans la structure en cascade ainsi qu'une comparaison entre les résultats obtenus avec la fonction de coût MSE (voir 4.4) et la fonction de coût VMSE. On peut constater le gain en performance sur le module moyen de l'erreur que produisent respectivement l'utilisation de l'architecture en cascade et la méthode VMSE.

TAB. 5.1 - Module moyen de l'erreur sur les $T_i \in [0, 200]$ and $\eta \in [0, 1]$.

Module moyen de l'erreur sur :	Sans cascade		Avec cascade	
	MSE	VMSE	MSE	VMSE
	$\nu = 0$	$\nu = 1$	$\nu = 0$	$\nu = 1$
Ensemble d'apprentissage				
η	0.0124	0.0124	0.0124	0.0124
T_1	4.49	4.33	4.67	4.36
T_2	6.03	5.89	5.49	5.28
T_3	7.84	7.03	5.71	5.67
T_4	9.54	6.89	6.34	6.08
Ensemble de validation				
η	0.0127	0.0127	0.0127	0.0127
T_1	4.41	4.37	4.66	4.39
T_2	5.95	5.79	5.43	5.30
T_3	7.71	6.98	5.68	5.75
T_4	9.36	7.13	6.35	6.25
Ensemble de test				
η	0.0124	0.0124	0.0124	0.0124
T_1	4.41	4.34	4.59	4.23
T_2	5.91	5.74	5.18	5.10
T_3	7.78	5.74	5.43	5.55
T_4	9.23	7.01	6.09	5.99

Au vu de ces résultats nous avons décidé, pour d'autres bases de données de trafic, d'utiliser l'architecture neuronale décrite ci-dessus et de l'entraîner à l'aide de la méthode VMSE avec ν=1 (voir 4.4 pour la signification de ν).

Les fonctions que l'on désire donc estimer pour chaque réseau de neurones sont donc les suivantes :

$$\hat{\eta} = f_1(\Phi_{obs}, \sum_{\text{PCR}}, s, \hat{\mu_B}, \hat{\sigma_B}) \quad (5.36)$$

$$\hat{T_1} = f_2(\Phi_{obs}, \sum_{\text{PCR}}, s, \hat{\mu_B}, \hat{\sigma_B}, \hat{\eta}) \quad (5.37)$$

$$\hat{T_2} = f_3(\Phi_{obs}, \sum_{\text{PCR}}, s, \hat{\mu_B}, \hat{\sigma_B}, \hat{\eta}, \hat{T_1}) \quad (5.38)$$

$$\hat{T_3} \quad f_4(\Phi_{obs}, \sum_{\text{PCR}}, s, \hat{\mu_B}, \hat{\sigma_B}, \hat{\eta}, \hat{T_1}, \hat{T_2}) \quad (5.39)$$

$$\hat{T_4} = f_5(\Phi_{obs}, \sum_{\text{PCR}}, s, \hat{\mu_B}, \hat{\sigma_B}, \hat{\eta}, \hat{T_1}, \hat{T_2}, \hat{T_3}) \quad (5.40)$$

5.4 Description des bases de données de trafics

5.4.1 Introduction

Nous voulons étudier la relation entre le trafic observé et la distribution de probabilité des temps de blocage. Nous utilisons un modèle de trafic afin de constituer des bases de données pour l'apprentissage car nous ne possédons pas un nombre suffisant de traces de trafic et cela nous permet d'étudier l'influence des différents paramètres constituant une trace de trafic. Ce modèle paramétrique est utilisé pour construire les bases de données de trafic mais les paramètres ne sont pas utilisés dans le cadre d'une estimation à l'aide de réseaux de neurones. En effet le but n'est pas une identification des paramètres du trafic mais une estimation directe de son comportement en termes de temps de blocage. La méthodologie utilisée ici peut donc être utilisée pour n'importe quelle base d'apprentissage.

Etant donné que nous considérons que la superposition de sources homogènes on/off est un trafic pire cas (voir 3.2.4), une base de données de trafic constituée de la superposition de sources on/off homogènes a été construite. Cette base de données sera utilisée pour réaliser l'apprentissage de la structure neuronale présentée en 5.3.3. Pour vérifier que les réseaux de neurones entraînés sur ce type de trafic qualifié de pire cas peuvent correctement généraliser sur d'autres types de trafic, une base de données de trafics hétérogènes (on/off hétérogènes) a été constituée pour des mesures de performances. Enfin, étant donné que lors de la section 5.5 nous comparerons l'estimation réalisée par les réseaux de neurones et l'approximation gaussienne, nous avons aussi construit une base de données de trafics gaussiens. On pourra alors comparer les performances de l'architecture neuronale et de l'approximation gaussienne sur des trafics très favorables à cette dernière.

Les modèles de trafic

Pour simuler la bande passante occupée, $B(t)$, nous avons utilisé deux modèles de trafic :

1. une superposition d'un nombre connu n de source on/off. Le trafic porté par un unique lien ATM est supposé avoir été produit par des sources on/off indépendantes.

   Pendant la période d'activité (on), la source émet un trafic ayant des caractéristiques, en terme de débit, constantes. Hors de cette période la source est silencieuse (off).

   Dans ce cas le comportement d'une source est défini par son débit crête, les périodes moyenne d'activité et de silence (respectivement T_{on} et T_{off}) ou de manière équivalente par :

   - p_{00} la probabilité de rester dans une période d'activité ;
   - p_{11} la probabilité de rester dans une période d'inactivité ;
   - $R_j(t)$ le débit correspondant de la source j.

   La bande passante occupée est définie par le trafic agrégé :

   $$B(t) = \sum_{j=1}^{n} R_j(t) \tag{5.41}$$

2. un tirage aléatoire dans une distribution de probabilité gaussienne. La bande passante occupée provient de tirages aléatoires uniformes et indépendants dans une distribution gaussienne de paramètres μ, σ :

   $$B(t) = \mathcal{N}(\mu, \sigma) \tag{5.42}$$

La structure de la simulation

Deux architectures de simulation ont été construites avec les modèles de trafic précités et sont présentées figure 5.8. Ces simulations sont basées sur le fait que sur le lien de sortie d'un nœud ATM la bande passante est limitée à une valeur maximale définie par les équipements ou l'allocation de ressources mise en place. Nous notons cette bande passante maximale utilisable sur le lien de sortie B_{max}. Du fait de cette limite le trafic entrant peut être temporairement stocké dans une ou plusieurs files d'attente de taille limitée.

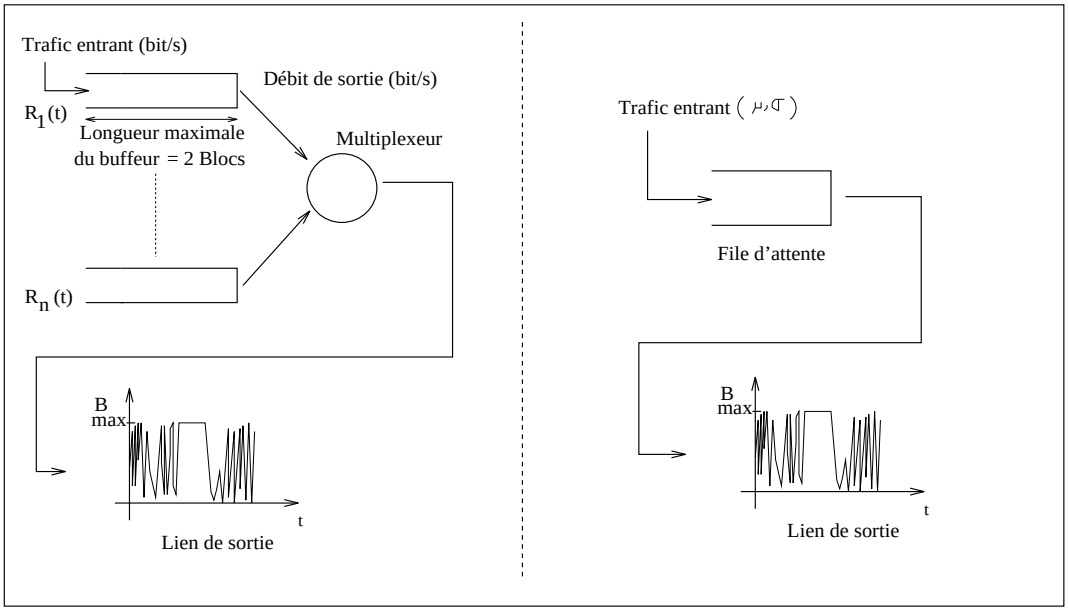

FIG. 5.8 - Les architectures de simulation pour la création de base de données de trafic (à gauche pour les sources on/off ; à droite pour le trafic gaussien).

Nous avons généré (voir ci-dessous) trois bases de données de traces de trafic : une base de données de trafics homogènes (sources on/off), une base de données de trafics hétérogènes (sources on/off) et une base de données de trafics gaussiens.

Pour chaque trace de trafic qui contient 95 10^6 échantillons de trafic, les 5 paramètres de qualité de service ont été calculés à l'aide de l'estimation empirique décrite en 5.3.1 (en utilisant toute la trace générée) et serviront de valeurs étalon.

Définition du couple de vecteurs d'entrées/sorties

Pour chaque trace de trafic la définition d'un couple de vecteurs d'entrées/sorties est (voir figure 5.9) :

- un vecteur de sortie constitué des 5 paramètres de qualité de service estimés à l'aide de l'estimation empirique présentée en 5.3.1 en utilisant toute la trace (donc K=95 10^6).

- un vecteur d'entrée dont la caractérisation du trafic est telle que définie en 5.3.3.

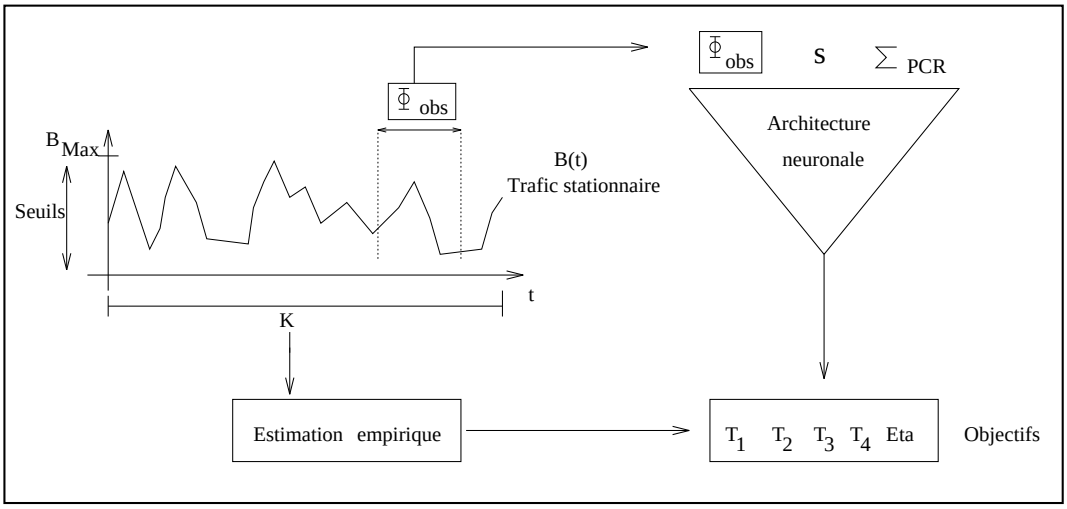

FIG. 5.9 - Illustration graphique des couples d'entrées/sorties.

5.4.2 La base de données de trafics homogènes

Les trafics ont été générés en accord avec le modèle de source on/off et la simulation décrite ci-dessus (5.4.1). Une discrétisation de l'espace des traces de trafic couvert par ce modèle, qui est fonction des trois paramètres p_{00}, p_{11} et $\sum_{PCR}$, a été effectuée pour des paramètres d'activité et des nombres de sources variés.

TAB. 5.2 - Pseudocode de l'algorithme pour générer la base de donnés de trafics homogènes (B_{max}=50)

pour p_{00}=0.05 **à** 0.95 (p_{00}=p_{00}+0.05) **faire**
  pour p_{11}=0.05 **à** 0.95 (p_{11}=p_{11}+0.05) **faire**
    pour $\sum_{PCR}$=nombre de sources=10 **à** 80 ($\sum_{PCR}$=$\sum_{PCR}$+1) **faire**
      pour t=1 **à** t=95 10^6 (t=t+1) **faire**
        établir un échantillon de $B(t)$
      fin pour
      pour s=1 % de B_{max} **à** s=100 % de B_{max} (s=s+1) **faire**
        calcul des $T_i(s, \sum_{\mathrm{PCR}})$ et de $\eta(s, \sum_{\mathrm{PCR}})$
        suivant l'estimation empirique (5.3.1)
      fin pour
    fin pour
  fin pour
fin pour

Dans cette base de données de trafics homogènes toutes les sources possèdent les mêmes paramètres d'activité ; le débit crête de chaque source est fixé à une

unité de la bande passante. Chaque trace de trafic est stationnaire et dépend des paramètres p_{00}, p_{11} et n (dans ce cas $\sum_{PCR} = n$). Le pseudocode de l'algorithme qui a permis de générer cette base de données, avec B_{max}=50, est donné dans le tableau 5.2.

Cette base de données a été divisée en trois sous ensembles : l'ensemble d'apprentissage, de validation et de test (avec $p_{00} \in [0.05, 0.95]$, $p_{11} \in [0.05, 0.95]$ et $\sum_{\text{PCR}} \in [10, 80]$) et ce pour réaliser un apprentissage de l'architecture neuronale détaillée en 5.3.3. Chaque sous ensemble contient différents cas de trafic (différentes valeurs des paramètres (p_{00}, p_{11}, $\sum_{\text{PCR}}$)) qui ne sont pas inclus dans les autres sous ensembles ; une représentation en est donnée figure 5.10.

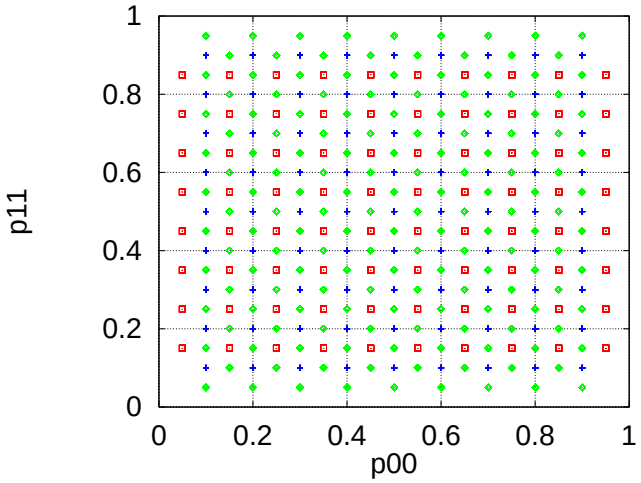

FIG. 5.10 - Base de données de traces de trafics homogènes : Répartition des trois ensembles (apprentissage (□), validation (+), test (◇)) sur un des plans constitué d'axes p_{00}, p_{11} à $\sum_{PCR}$ constant.

En fonction de $\sum_{PCR}$, tous les cas sont segmentés comme ci-dessus sauf pour le cas où $\sum_{PCR}$=27 qui est entièrement réservé pour l'ensemble de test.

L'ensemble d'apprentissage sera utilisé pour entraîner les réseaux de neurones et l'ensemble de validation permet de contrôler leur faculté de généralisation au cours de la phase d'entraînement (voir 4.3.2). L'ensemble de test n'est jamais utilisé pendant l'apprentissage et sert seulement à des mesures de performance. On possède ainsi 2880, 2916 et 11502 traces de trafic pour respectivement l'ensemble de test, de validation et l'ensemble d'apprentissage (soit donc 1.643 10^{12} échantillons de trafic).

5.4.3 La base de données de trafics hétérogènes

Les trafics ont été générés en accord avec le modèle de source on/off et la simulation décrite ci-dessus (5.4.1). Cette fois-ci il n'y a qu'un ensemble de test

étant donné que cette base de données de trafic n'est destinée qu'à une mesure de performance. Pour chaque trace de trafic générée les probabilités p_{00} et p_{11} de chaque source sont fixées à l'aide d'un tirage aléatoire uniforme dans l'intervalle $[0.05, 0.95]$. Elles ont donc des paramètres d'émission différents les unes des autres. Tous les débit crêtes sont pris comme étant l'unité de la largeur de bande sachant que $\sum_{\text{PCR}} \in [10, 80]$.

25270 traces de trafics ont été générées pour cette base de données soit 2.4 10^{12} échantillons de trafic. Le pseudocode de l'algorithme qui a permis de générer cette base de données, avec B_{max}=50, est donné dans le tableau 5.3.

TAB. 5.3 - Pseudocode de l'algorithme pour générer la base de donnés de trafics hétérogènes (B_{max}=50)

pour k=1 **à** 300 (k=k+1) **faire**
  pour $\sum_{PCR}$=nombre de source=10 **à** 80 ($\sum_{PCR}$=$\sum_{PCR}$+1) **faire**
    pour chaque source i
      p_{00}^i=random(0.05,0.95)
      p_{11}^i=random(0.05,0.95)
    fin pour
    pour t=1 **à** t=95 10^6 (t=t+1) **faire**
      établir un échantillon de $B(t)$
    fin pour
    pour s=1 % de B_{max} **à** s=100 % de B_{max} (s=s+1) **faire**
      calcul des $T_i(s, \sum_{\text{PCR}})$ et de $\eta(s, \sum_{\text{PCR}})$
      suivant l'estimation empirique (5.3.1)
    fin pour
  fin pour
fin pour

5.4.4 La base de données de trafics gaussiens

La base de données des traces de trafics gaussiens est basée sur un modèle fluide (voir 3.1.3) car une seule file d'attente est implémentée pour absorber la congestion au niveau cellule. Chaque trace de trafic dépend de deux paramètres : la moyenne et l'écart type du trafic agrégé. Par définition pour un modèle gaussien il difficile de déterminer ce à quoi peut correspondre la somme des débits crêtes déclarés par les sources qui engendrent ce trafic. Aussi pour limiter le trafic à une *"somme des débits crêtes déclarés"* chaque échantillon de trafic est tiré dans une densité de probabilité gaussienne tronquée dans l'intervalle $[\mu - 5\sigma, \mu + 5\sigma]$, par conséquent $\sum_{\text{PCR}} = \mu + 5\sigma$. Si l'échantillon de trafic a une valeur négative un autre échantillon est alors tiré pour le remplacer et si la valeur de l'échantillon de

trafic est supérieure à bande passante maximale, B_{max} la file d'attente est alors utilisée. Dans ce cas on remplit le lien de sortie et ce qui reste est posé dans la file d'attente.

Nous avons généré 1106 traces de trafic suivant cet algorithme avec $\mu \in [10.0, 49.0]$ et $\sigma \in [0.5, 7.0]$ soit 1.05 10^{11} échantillons de trafic. Le pseudocode de l'algorithme qui a permis de générer cette base de données, avec B_{max}=50, est donné dans le tableau 5.4. Là encore, il n'y a qu'un ensemble de test étant donné que cette base de données de trafic n'est destinée qu'à une mesure de performance.

TAB. 5.4 - Pseudocode de l'algorithme pour générer la base de donnés de trafics gaussien(B_{max}=50).

pour μ=10.0 **à** 49.0 (μ=μ+0.5) **faire**
  pour σ=0.5 **à** 7.0 (σ=σ+0.5) **faire**
    pour t=1 **à** t=95 10^6 (t=t+1) **faire**
      construction de $B(t) \in [\mu - 5\sigma, \mu + 5\sigma]$, B(t) ≥ 0
    fin pour
    pour s=1 % de B_{max} **à** s=100 % de B_{max} (s=s+1) **faire**
      calcul des $T_i(s, \sum_{\text{PCR}})$ et de $\eta(s, \sum_{\text{PCR}})$
      suivant l'estimation empirique (5.3.1)
    fin pour
  fin pour
fin pour

5.5 Comparaison des différentes méthodes d'estimation

5.5.1 Introduction

On cherche à vérifier si l'utilisation des réseaux de neurones entraînés sur des trafic qualifiés de pire cas peuvent correctement généraliser sur d'autres types de trafics. C'est pourquoi l'apprentissage de l'architecture neuronale n'a été fait qu'avec la base de données de trafics homogènes. L'ensemble de test de cette base de données permettra de vérifier si l'apprentissage a été efficace, au sens de la capacité de généralisation des réseaux de neurones. Les deux autres bases de données constitueront donc aussi des bases de test pour l'architecture neuronale.

Pour chacune des trois bases de données de trafic (homogène, hétérogène, gaussien) les comparaisons sont effectuées entre les valeurs étalons établies par

l'estimation empirique sur une trace entière de trafic (stationnaire) et l'estimation "on line" fournie soit par l'approximation gaussienne soit par l'architecture neuronale.

Les critères de comparaison retenus sont les suivants :

– l'erreur moyenne sur un ensemble :

$$\text{Erreur Moyenne} = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{d}_r(s,\sum_{\text{PCR}}) - d_r(s,\sum_{\text{PCR}})\right) \tag{5.43}$$

– le module moyen des erreurs obtenues sur un ensemble :

$$\text{Module moyen de l'erreur} = \frac{1}{R}\sum_{r=1}^{R}\left(|\hat{d}_r(s,\sum_{\text{PCR}}) - d_r(s,\sum_{\text{PCR}})|\right) \tag{5.44}$$

avec

– R la taille de l'ensemble considéré ;

– d_r la valeur étalon ;

– $\hat{d}_r$ la valeur obtenue par l'estimateur ;

– r l'indice du couple de vecteur d'entrées/sorties.

Le signe de l'erreur moyenne nous permettra de qualifier le caractère de l'estimation. S'il est négatif l'estimation est optimiste, dans le cas contraire elle est conservative. Le module moyen de l'erreur permet de quantifier ce caractère.

Nous considérerons aussi la distribution des erreurs en utilisant la représentation proposée figure 5.11. Cette représentation permet de qualifier les erreurs commises ainsi que l'intervalle de confiance comme suit :

– erreur moyenne et une médiane proches de zéro → erreurs faibles ;

– erreur moyenne proche de la médiane → distribution des erreurs symétrique ;

– faible IQR → distribution étroite.

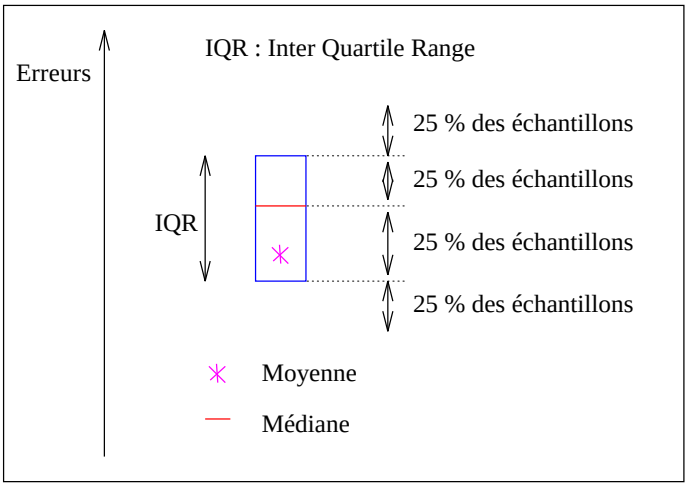

FIG. 5.11 - Représentation graphique des erreurs commises : L'étoile indique la position de l'erreur moyenne. La "boite" représente l'inter quartile range (IQR). 50 % des échantillons sont contenus entre la ligne supérieure et la ligne inférieure de cette boite et la médiane représente le milieu de la distribution.

5.5.2 Résultats sur le trafic homogène

La figure 5.12 présente les résultats obtenus avec l'architecture neuronale pour une fenêtre d'observation (Φ_{obs}) de 80 et l'approximation gaussienne pour une fenêtre d'observation de 80 et de 180. Les résultats sont donnés pour l'ensemble de test. Les valeurs précises des résultats obtenus sur l'erreur moyenne sont présentés tableau 5.5.

Sur cette figure on voit que :

- l'architecture neuronale surestime légèrement les temps de blocage alors que l'approximation gaussienne les sous-estime. L'approximation gaussienne sur ce type de trafic est donc optimiste tandis que l'architecture neuronale est conservative.
- la distribution des erreurs avec l'architecture neuronale est symétrique alors que la distribution des erreurs avec l'approximation gaussienne est fortement asymétrique.
- l'IQR de la distribution des erreurs avec l'architecture neuronale est nettement inférieure à celui de l'approximation gaussienne.
- les erreurs commises par l'approximation gaussienne peuvent être diminuées si on augmente la taille de la fenêtre d'observation, afin d'avoir une meilleure évaluation de la moyenne et de l'écart type du trafic. Cependant

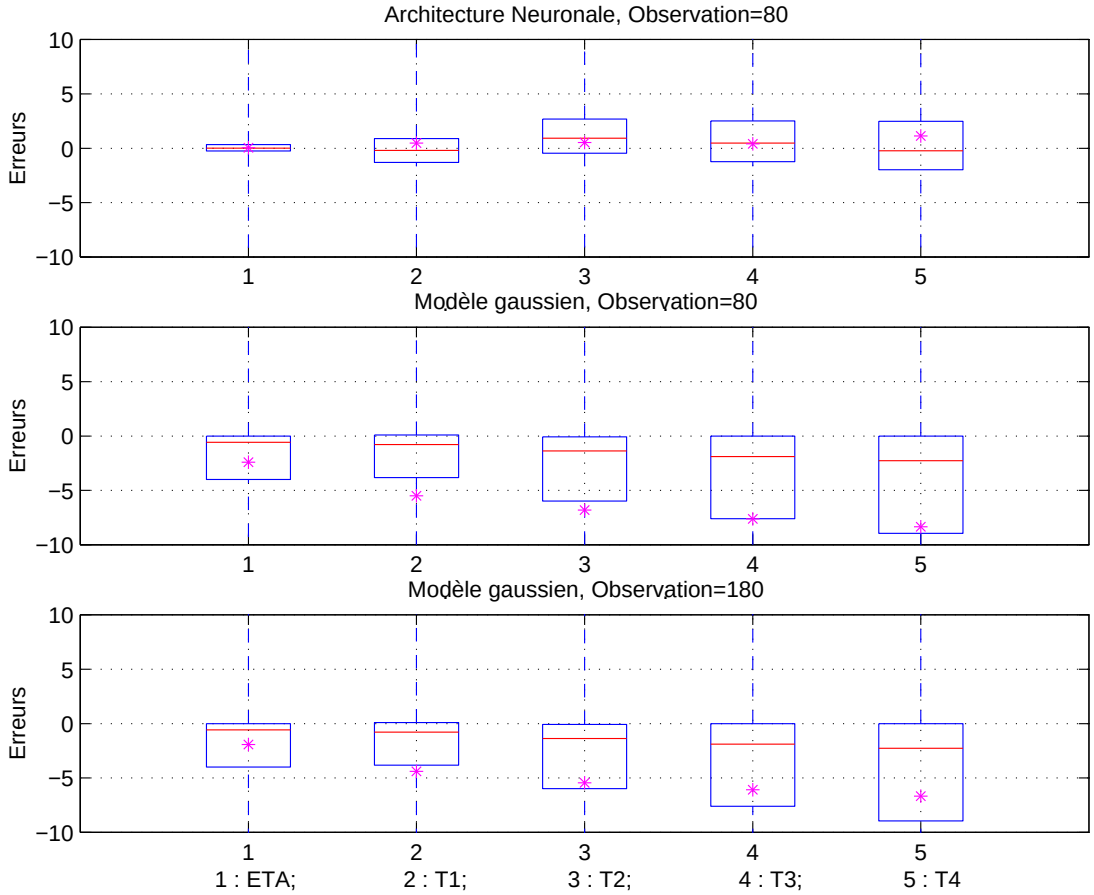

FIG. 5.12 - Représentation graphique de la distribution des erreurs commises sur la base de données de trafics homogènes : en haut avec l'architecture neuronale, au milieu avec l'approximation gaussienne pour une fenêtre d'observation de 80, au bas avec l'approximation gaussienne pour une fenêtre d'observation de 180.

plus la fenêtre d'observation est grande plus le temps nécessaire pour réagir aux variations du trafic est grand. L'estimation réalisée par l'architecture neuronale permet une adaptation rapide aux variations du trafic étant précise même avec une petite fenêtre d'observation.

On conclut que les estimations réalisées par l'architecture neuronale sont meilleures que celles produites par l'approximation gaussienne.

On précise (voir tableaux 5.5, 5.6) que les scores obtenus par les réseaux de neurones sont approximativement les mêmes sur l'ensemble d'apprentissage, de validation et de test. On peut en conclure que le processus d'apprentissage a été efficace et que le réseau a une bonne capacité de généralisation.

TAB. 5.5 - Les erreurs pour chaque $T_i \in [0, 200]$ et $\eta \in [0, 1]$ pour la base de données de trafics homogènes.

Erreur moyenne sur :					
Ensemble	T_1	T_2	T_3	T_4	η
Estimation par l'architecture neuronale Φ_{obs}=80					
Apprentissage	0.39	0.61	0.32	0.80	0.0038
Validation	0.36	0.57	0.21	0.88	0.0059
Test	0.12	0.95	0.58	0.57	0.0033
Approximation gaussienne Φ_{obs}=80					
Test	-5.43	-7.04	-8.09	-8.96	-0.0206
Approximation gaussienne Φ_{obs}=180					
Test	-4.06	-5.64	-6.97	-7.86	-0.0207

TAB. 5.6 - Les erreurs pour chaque $T_i \in [0, 200]$ et $\eta \in [0, 1]$ pour la base de données de trafics homogènes.

Module moyen de l'erreur sur :					
Ensemble	T_1	T_2	T_3	T_4	η
Estimation par l'architecture neuronale Φ_{obs}=80					
Apprentissage	4.45	5.38	5.64	5.90	0.0123
Validation	4.52	5.49	5.86	6.13	0.0126
Test	4.50	5.37	5.63	5.91	0.0125
Approximation gaussienne Φ_{obs}=80					
Test	8.29	9.32	9.93	10.51	0.0267
Approximation gaussienne Φ_{obs}=180					
Test	7.47	8.54	9.39	10.12	0.0245

5.5.3 Résultats sur le trafic hétérogène

La figure 5.13 présente les résultats obtenus avec l'architecture neuronale et l'approximation gaussienne pour une fenêtre d'observation (Φ_{obs}) de 80. Les valeurs précises des résultats obtenus sur l'erreur moyenne et le module moyen de l'erreur sont présentés tableau 5.7 .

Sur cette figure on voit que :

– l'architecture neuronale surestime légèrement les temps de blocage alors que l'approximation gaussienne les sous-estime. La surestimation produite par l'architecture neuronale est un peu plus importante que dans le cas homo-

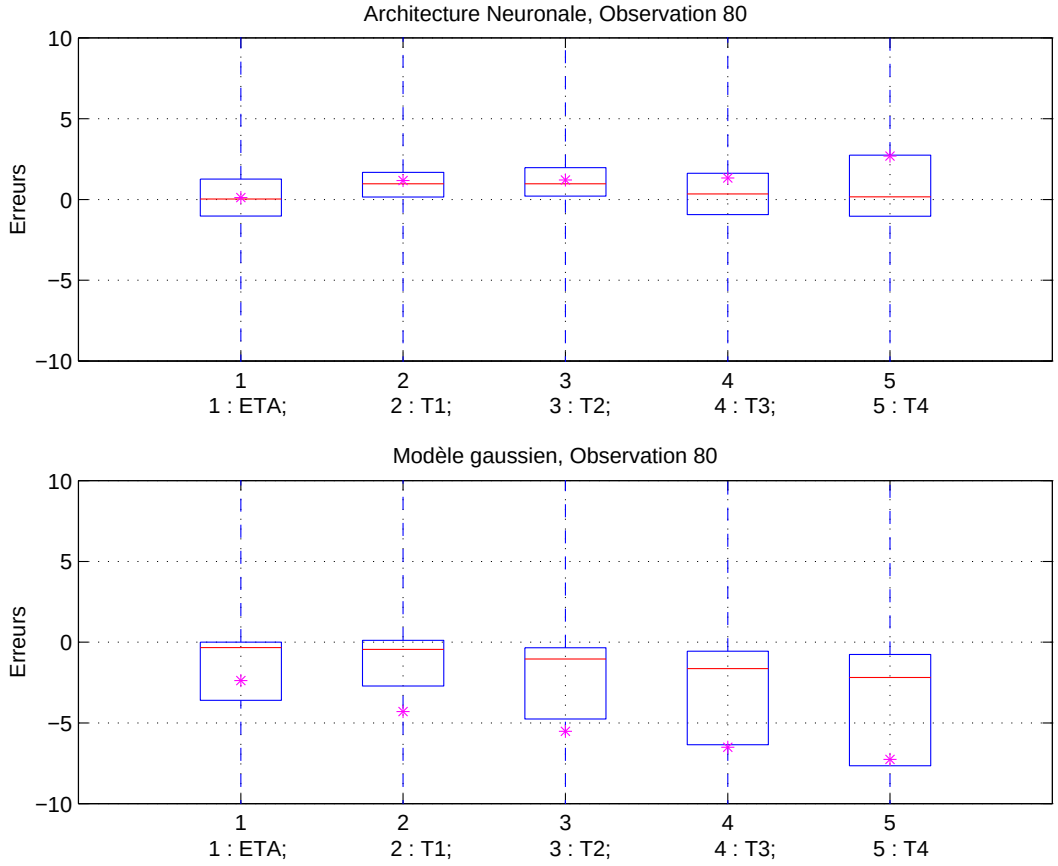

FIG. 5.13 - Représentation graphique de la distribution des erreurs commises sur la base de données de trafics hétérogènes (en haut avec l'architecture neuronale, en bas avec l'approximation gaussienne).

gène. Comme pour le cas homogène l'approximation gaussienne sur ce type de trafic est optimiste tandis que l'architecture neuronale est conservative.

– la distribution des erreurs avec l'architecture neuronale est symétrique (mis à part pour T_4) alors que la distribution des erreurs avec l'approximation gaussienne est fortement asymétrique.

– l'IQR de la distribution des erreurs avec l'architecture neuronale est nettement inférieur à celui de l'approximation gaussienne.

Les résultats obtenus sur cette base de données, montrent que le processus d'apprentissage sur la base de données homogène (considérée comme des trafics pire cas) a été efficace. Il permet d'obtenir une bonne généralisation sur la base de données de trafics hétérogènes.

TAB. 5.7 - Les erreurs pour chaque $T_i \in [0, 200]$ et $\eta \in [0, 1]$ pour la base de données de trafics hétérogènes.

Erreur moyenne sur :					
	T_1	T_2	T_3	T_4	η
Estimation par l'architecture neuronale Φ_{obs}=80	1.47	1.77	1.66	2.73	0.0011
Approximation gaussienne Φ_{obs}=80	-3.96	-5.33	-6.39	-7.23	-0.0204

Module moyen de l'erreur sur :					
	T_1	T_2	T_3	T_4	η
Estimation par l'architecture neuronale Φ_{obs}=80	5.48	5.66	5.91	6.57	0.0166
Approximation gaussienne Φ_{obs}=80	7.36	8.19	8.70	9.29	0.0235

5.5.4 Résultats sur le trafic gaussien

La figure 5.14 présente les résultats obtenus avec l'architecture neuronale et l'approximation gaussienne pour une fenêtre d'observation (Φ_{obs}) de 80. Les valeurs précises des résultats obtenus sur l'erreur moyenne et le module moyen de l'erreur sont présentés tableaux 5.8, 5.9.

Sur cette figure on voit que :

- l'architecture neuronale ainsi que l'approximation gaussienne surestiment ensemble les temps de blocage. Cette fois-ci l'approximation gaussienne sur ce type de trafic et l'architecture neuronale sont conservatives.
- les distributions des erreurs avec l'architecture neuronale et avec l'approximation gaussienne sont asymétriques.
- l'IQR de la distribution des erreurs avec l'approximation gaussienne est nettement inférieur à celui de l'architecture neuronale.

On peut noter cependant que cette base de données est très favorable à l'approximation gaussienne du fait qu'il s'agit de trafics gaussiens. Les erreurs commises par l'approximation gaussienne proviennent de la taille finie de la fenêtre d'observation.

On s'aperçoit là encore que les estimations produites par l'architecture neuronale sont assez bonnes. Néanmoins elles restent inférieures aux erreurs de l'ap-

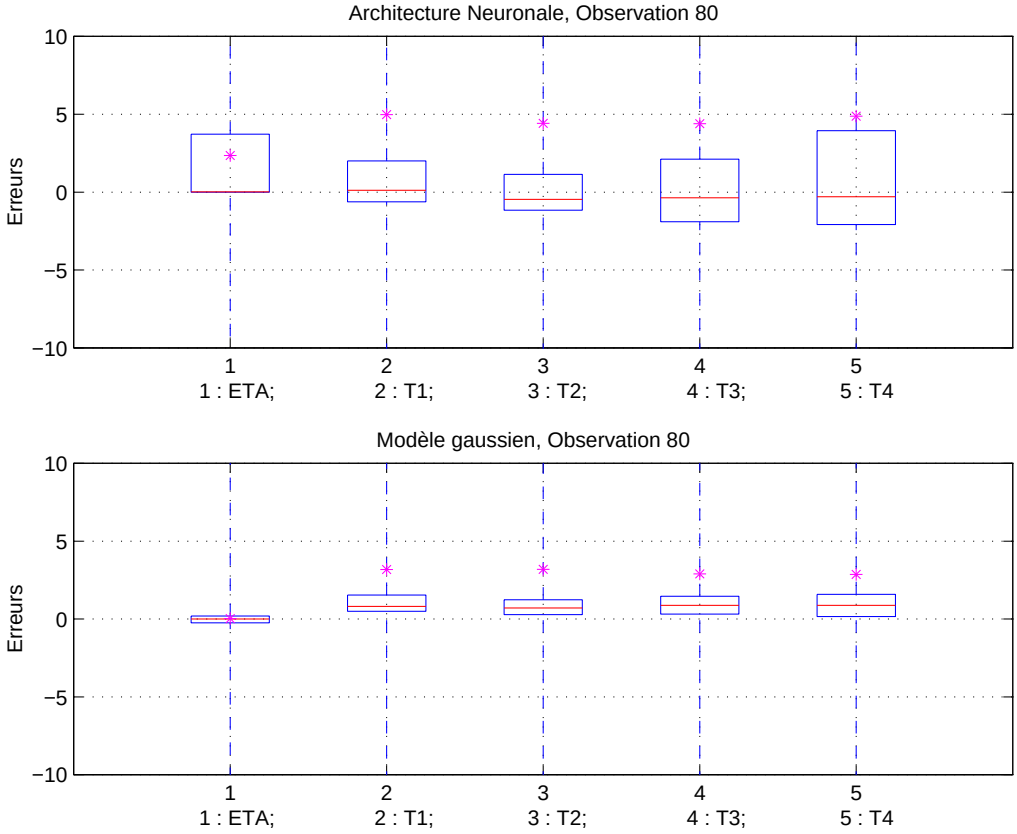

FIG. 5.14 - Représentation graphique de la distribution des erreurs commises sur la base de données de trafics gaussien (en haut avec l'architecture neuronale, en bas avec l'approximation gaussienne).

proximation gaussienne sur les deux bases de données précédentes. De plus les résultats obtenus sur cette base de données, montrent que le processus d'apprentissage sur la base de données homogène (considérés comme des trafics pire cas) a été efficace : les estimations restent conservatives.

TAB. 5.8 - Les erreurs pour chaque $T_i \in [0, 200]$ et $\eta \in [0, 1]$ pour la base de données de trafics gaussien.

Erreur moyenne sur :					
	T_1	T_2	T_3	T_4	η
Estimation par l'architecture neuronale Φ_{obs}=80	4.66	4.42	4.21	4.61	0.0207
Approximation gaussienne Φ_{obs}=80	3.43	3.33	3.48	3.39	0.0020

TAB. 5.9 - Les erreurs pour chaque $T_i \in [0, 200]$ et $\eta \in [0, 1]$ pour la base de données de trafics gaussien.

Module moyen de l'erreur sur :					
	T_1	T_2	T_3	T_4	η
Estimation par l'architecture neuronale Φ_{obs}=80	8.02	8.80	8.30	8.42	0.0238
Approximation gaussienne Φ_{obs}=80	7.24	7.48	7.39	7.38	0.0132

Pour comprendre d'ou viennent les erreurs commises par l'architecture neuronale et par le modèle gaussien on présente figure 5.15 :

– l'espace d'apprentissage de l'architecture neuronale dans le plan (μ,σ) ;

– l'espace des traces de trafics gaussiens utilisés en test dans le plan (μ,σ).

Ces espaces sont mesurés au niveau du lien de sortie en ayant utilisé l'architecture de simulation présentée figure 5.8.

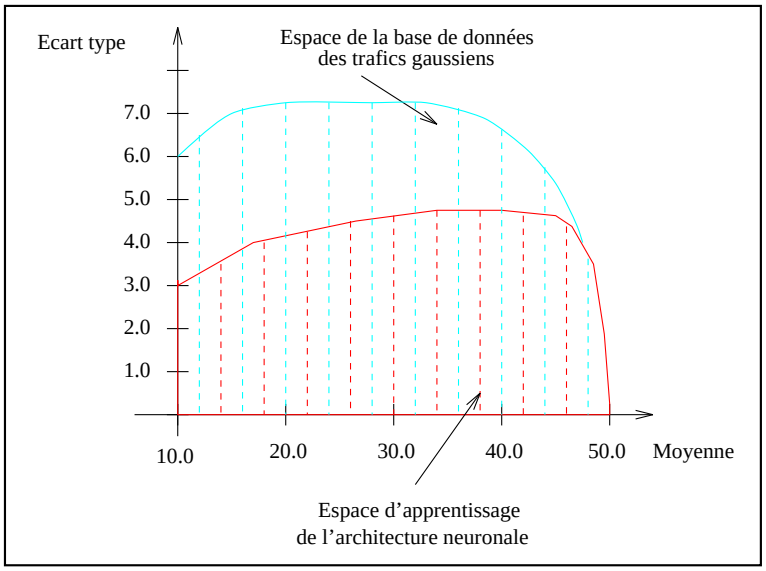

FIG. 5.15 - Représentation mesurée de l'espace d'apprentissage mesuré (au niveau du lien de sortie) de l'architecture neuronale et de la base de données de trafics gaussien dans le plan (μ,σ) en ayant utilisé l'architecture de simulation présentée figure 5.8.

Précisons que cette représentation est basée uniquement sur les deux premiers moments des traces de trafics. Bien que les deux espaces se recouvrent en partie

les trafics correspondants ne sont pas identiques. Ils ne possèdent pas les mêmes moments d'ordre supérieur à 2 et les mêmes degrés de corrélation. La limite supérieure de l'espace d'apprentissage de l'architecture neuronale correspond au cas ou n tend vers l'infini. Cette limite provient du modèle de source (on/off à deux états) utilisé. Cet espace d'apprentissage pourrait être étendu par exemple en utilisant un modèle de source possèdant un nombre d'états plus important.

La figure 5.16 présente les grandes erreurs commises, dans le plan (μ,σ) par, respectivement, l'architecture neuronale et le modèle gaussien pour T_4.

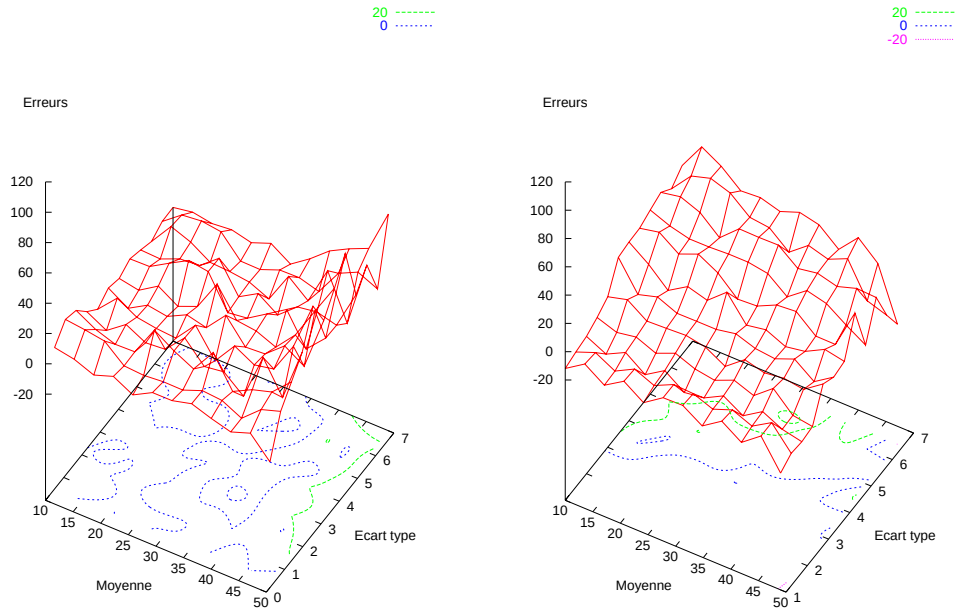

FIG. 5.16 - Erreurs du modèle gaussien (à gauche) et de l'architecture neuronale (à droite) sur la base de données de trafics gaussiens dans le plan (μ,σ) : les courbes de niveau représente la frontière entre les estimations pessimiste et optimiste (0.0 en bleu) ainsi que les erreurs supérieure à 20.0 (en vert).

On s'aperçoit que les fortes erreurs positives (pessimistes) du modèle gaussien sont localisées à la frontière de la largeur de bande maximale utilisable B_{max} pour $\mu \simeq 50$. Cette frontière constitue en effet la limite du domaine de validité du modèle. En effet l'écart type mesuré $\hat{\sigma}$ du trafic provient uniquement des variations du trafic situées en dessous de B_{max} alors que cette valeur $\hat{\sigma}$ est utilisée par le modèle comme étant symétrique par rapport à la moyenne mesurée d'où des estimations pessimistes. Ce sont ces fortes valeurs qui rendent la distribution

des erreurs légèrement asymétrique (voir figure 5.14). En dehors de cette zone les erreurs positives et négatives, plus faibles en module, sont réparties assez uniformément dans le plan (μ,σ).

Pour l'architecture neuronale les fortes erreurs ne proviennent pas de la partie "apprise" du plan (μ,σ). Ces dernières sont localisées dans la partie du plan où se situent de forts ratio σ/μ. Cette partie du plan correspond à des trafics très fortement sporadiques. Ce sont là aussi ces fortes valeurs qui rendent la distribution des erreurs asymétrique (médiane # moyenne voir figure 5.14). On note que sur cette partie du plan les estimations sont pessimistes. Ce résultat est satisfaisant si on veut réaliser un contrôle d'admission prudent.

5.5.5 Autres résultats et utilisation

La valeur que nous nous sommes fixée pour T_{max} étant de 200 (valeur en temps discret) le choix de la valeur de K a été de 95 millions d'échantillons de trafic. La valeur de T_{max} est arbitraire du fait que nous travaillons en temps discret et que par conséquent l'unité de temps est elle même arbitraire. Si le trafic utilisé pour l'estimation des temps de blocage est échantillonné sur des pas de 1 ms alors la valeur de T_{max} est de 200 ms. Cependant pour acquérir des valeurs supérieures à 200 ms il suffit de rééchantillonner le trafic avec un autre pas. Par exemple avec un pas de 10 ms la valeur de T_{max} passe à 2 s. Pour obtenir des valeurs plus petites la démarche est à l'identique.

Dans le cadre d'un contrôle d'admission des connexions il peut être important de qualifier les erreurs d'estimation commises en terme de débit demandé. Pour cela on définit la probabilité d'une erreur d'amplitude k (k est en pourcentage du débit maximum utilisable au lien ATM) par (voir figure 5.17) :

$$Pr(E(k)) = \frac{1}{R}\sum_{r=1}^{R} e(k) \tag{5.45}$$

avec

$$e(k) = \ 0 \text{ si } (T_i^r(s-k, \sum_{\text{PCR}}) \geq \hat{T_i^r}(s, \sum_{\text{PCR}}) \geq T_i^r(s+k, \sum_{\text{PCR}}) \tag{5.46}$$

$$\text{sinon } e(k) = 1 \tag{5.47}$$

Du fait que les erreurs commises par l'architecture neuronale ainsi que la distribution de ces erreurs sont "petites" la probabilité, constatée sur les différentes bases de données, $Pr(E(2))$ est toujours inférieure à 0.05. On en déduit que nous avons une probabilité de 0.95 que la précision obtenue soit de l'ordre de 2% de la bande passante du lien.

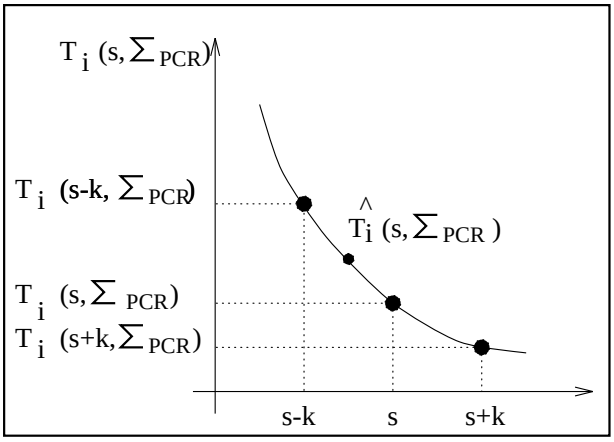

FIG. 5.17 - Illustration graphique des erreurs en fonction de la bande passante.

L'utilisation de l'estimation de la distribution de probabilité des temps de blocage dans un lien ATM est illustrée figure 5.18 pour deux connexions n'ayant pas les mêmes exigences. Ces deux connexions demandent à être acceptées sur un nœud où des connexions ont été déjà acceptées pour une somme des PCR de 25 ($B_{max} = 100$). Chacune des deux nouvelles connexions requiert un PCR de 29.4 et leur qualité de service est définie en terme d'un gabarit de temps de blocage (pour les probabilités 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4} voir figure 5.18). Les temps de blocage a posteriori sont par conséquent estimés à partir du trafic en cours pour un seuil s de 70.6 ($s = B_{max} - d_{max}$) et pour $\sum_{\text{PCR}}$=25.

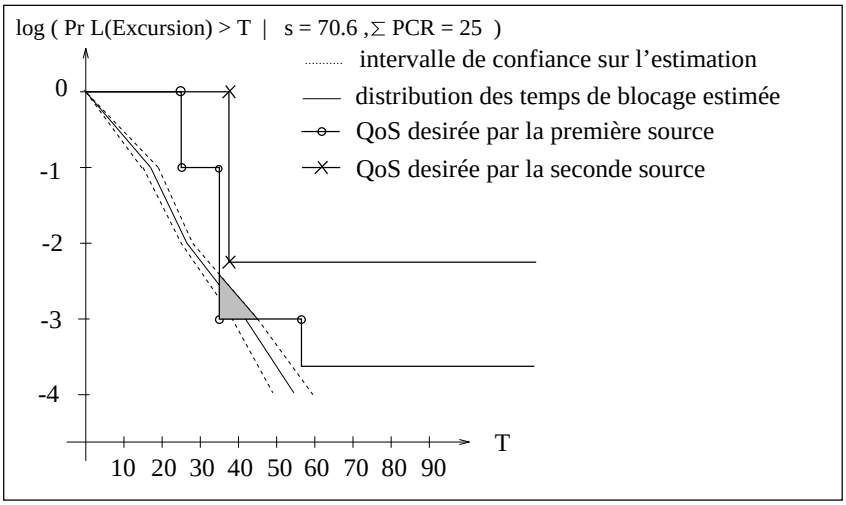

FIG. 5.18 - Illustration graphique de l'utilisation de l'estimation de la distribution de probabilité des temps de blocage dans un lien ATM

Sous l'hypothèse que la qualité de service des connexions déjà acceptées est respectée, la seconde source, moins exigeante, sera acceptée alors que la deuxième

sera refusée, du fait que des temps de blocage trop longs ont une probabilité d'apparaître supérieure à ses exigences de qualité de service.

5.6 Discussion

Ces résultats montrent le pouvoir de généralisation des réseaux de neurones sur des exemples non appris. Le but recherché, qui était de réaliser un apprentissage sur un trafic qualifié de pire cas, de manière à réaliser une estimation des temps de blocage conservative sur d'autres types de trafics, a été atteint. L'architecture neuronale possède un bon niveau de fiabilité permettant de généraliser les résultats appris à des données inconnues car l'espace d'apprentissage a été judicieusement choisi et est suffisamment étendu.

Cette méthode ne souffre pas de l'utilisation d'un modèle de trafic qui pourrait être inadapté au trafic sur lequel on l'exerce. En effet, l'approximation gaussienne s'est montré performante sur la base de données de trafics gaussien mais bien moins précise sur les deux autres bases de données. L'architecture neuronale peut s'adapter facilement aux changements du trafic sur lesquels aucune hypothèse préalable n'est nécessaire. De plus elle est précise même avec une petite fenêtre d'observation.

On voit dès lors que les réseaux de neurones en général et l'architecture neuronale proposée en particulier peuvent être utilisés pour permettre au réseau de dimensionner les temps d'attente et/ou de blocage que vont subir les blocs transmis. Cette architecture neuronale peut donc être intégré dans une procédure de contrôle d'admission des connexions ayant un caractère légèrement conservatif.

Chapitre 6

Conclusion et perspectives

L'apparition du mode de transfert temporel asynchrone au début des années 80 a relancé les problèmes classiques de contrôle de flux. Le contrôle de flux, qu'il soit préventif ou réactif, apparaît être un élément clef de la gestion d'un réseau de télécommunications. L'objectif de notre thèse était de déterminer si les réseaux de neurones artificiels peuvent permettre de réaliser un contrôle d'admission des connexions efficace pour le mode de transfert ABT-DT du réseau ATM. C'est pourquoi notre travail s'est articulé autour de ces deux thèmes de travail : le contrôle d'admission des connexions (CAC) et les réseaux de neurones artificiels.

Le premier point de notre étude a porté sur le contrôle d'admission des connexions. Deux grandes familles de méthodes de contrôle d'admission des connexions existent : soit le contrôle d'admission des connexions utilise un modèle de trafic décrivant ses propriétés statistiques (méthode paramétrique) ou pas (méthode non paramétrique). Il est évident que le choix du modèle est une étape cruciale pour ces techniques. Ce dernier doit être suffisamment "souple" pour pouvoir représenter une grande variété de trafic. Il doit aussi bien se prêter à l'identification des paramètres. Il semble que les méthodes non paramétriques soient les plus adaptées si on veut que le contrôle d'admission des connexions utilisé recouvre un grand espace de décision. En effet, ces méthodes ne souffrent pas de l'utilisation d'un modèle qui pourrait être inadapté au trafic sur lequel on l'exerce. De plus, les paramètres utilisés pour décrire les flux de trafic, et donc les paramètres de contrat de trafic, constituent une maigre description des caractéristiques statistiques des flux. Par conséquent réaliser une politique d'allocation de ressources et/ou de contrôle d'admission des connexions uniquement basée sur leur valeurs peut induire une sous utilisation du réseau. Il semble plus efficace d'estimer les caractéristiques du trafic en temps réel et de réaliser un contrôle d'admission basé sur des mesures. C'est pourquoi nous nous sommes orientés vers un contrôle d'admission des connexions non paramétrique basé sur des mesures.

De nouvelles techniques de contrôle d'admission basées sur des mesures utilisant les réseaux de neurones artificiels, basés sur la théorie de l'apprentissage

statistique, ont vu le jour. Ces approches connexionnistes de la caractérisation du trafic en général et du contrôle d'admission en particulier apportent en effet de nombreux avantages par rapport aux méthodes classiques. Elles s'adaptent facilement aux changements du trafic sur lequel aucune hypothèse préalable n'est nécessaire. Elles possèdent un haut niveau de fiabilité et de tolérance aux fautes, permettent de généraliser à des données inconnues les résultats appris.

Le deuxième point de notre étude a été de définir un modèle de trafic afin de constituer des bases de données pour l'apprentissage de réseaux de neurones car nous ne possédions pas un nombre suffisant de traces de trafic. Le modèle choisi se devait d'être pire cas de manière à réaliser un contrôle d'admission des connexions prudent. Nous avons, au vu de l'état de l'art, déterminé que le trafic pire cas pour l'allocation des ressources d'une connexion ayant un contrat de trafic ABT consiste en une source on/off ayant un débit maximal égal au débit crête négocié pour toute la durée de la connexion. Ce modèle de trafic a donc été plus tard utilisé pour construire des traces de trafic variées. Ces traces de trafic ont permis de réaliser l'apprentissage de réseaux de neurones.

Notre contribution dans le domaines des réseaux de neurones artificiels a été de présenter une nouvelle méthode destinée à améliorer les performances en généralisation des perceptrons multicouches utilisés en tant que réseaux discriminants et approximateurs de fonctions. On a montré comment modifier le critère d'apprentissage afin de contrôler la distribution des erreurs au cours de l'apprentissage. Des résultats, améliorant notablement l'état de l'art sur trois problèmes, ont été présentés pour valider la méthode. Cette méthode utilisée en tant que fonction de régularisation est très facilement implémentable.

Cette contribution a été appliquée dans le cadre du contrôle d'admission des connexions du réseau ATM. Nous avons proposé un contrôle d'admission des connexions basé sur deux caractérisations, une caractérisation de la nouvelle connexion et une caractérisation du comportement de la bande passante utilisée par le trafic en cours. Ce contrôle est basé sur le dimensionnement des temps d'attente et/ou de blocage que vont subir les blocs transmis. Au vu du choix du modèle de trafic décrit ci-dessus, le but recherché était de réaliser un apprentissage sur un trafic qualifié de pire cas, de manière à réaliser une estimation des temps de blocage conservative et précise sur d'autres types de trafics. Ce but a été atteint et la nouvelle fonction de coût régularisante proposée pour les réseaux de neurones artificiels s'est là encore montrée performante. L'architecture neuronale proposée possède un bon niveau de fiabilité permettant de généraliser les résultats appris à des données inconnues car l'espace d'apprentissage a été judicieusement choisi et est suffisamment étendu.

Cette méthode ne souffre pas de l'utilisation d'un modèle de trafic qui pourrait être inadapté au trafic sur lequel on l'exerce car les paramètres du modèle de trafic utilisé pour construire les bases de données de trafic n'ont pas été utilisés dans le cadre d'une estimation à l'aide de réseaux de neurones. L'architecture

connexionniste peut s'adapter facilement aux changements du trafic sur lesquels aucune hypothèse préalable n'est nécessaire. De plus elle est précise même avec une petite fenêtre d'observation.

On voit que les réseaux de neurones en général et l'architecture connexionniste proposée en particulier peuvent être utilisés pour permettre au réseau de dimensionner les temps d'attente et/ou de blocage que vont subir les blocs transmis. Cette architecture connexionniste peut donc être intégrée dans une procédure d'estimation des périodes de congestion ayant un caractère légèrement conservatif. Elle permettra alors de décider l'acceptation d'une nouvelle connexion au regard de ses paramètres de trafic.

Annexe A

Application de la nouvelle fonction de coût régularisante au modèle CGM.

LISTEN est un système temps réel de suivi de personnes [Collobert et al., 1996]. Dans ce système, les visages sont détectés par un réseau de neurones dans des zones de teinte chair. Marcel et al. [Marcel et al., 1999; Marcel, 1999] présentent une extension de LISTEN utilisant la reconnaissance des postures de la main pour exécuter une commande. Cette reconnaissance des postures de la main dans une image est effectuée à l'aide d'un modèle de réseau de neurones.

Il ne s'agit pas là de reconnaître toutes les postures possibles de la main dans une image mais seulement un petit nombre. Aussi, un alphabet de postures de la main a été constitué (A, B, C, Cinq, Pointe et V). Pour chacune d'elle, Marcel et al. ont construit une base de plusieurs milliers d'exemples sur des fonds uniformes et complexes divers (80 % servant lors de la phase d'apprentissage et le reste pour la validation et le test). Les tailles des fenêtres des postures dans l'image sont : 20x20 pour A, 18x30 pour B, 18x20 pour C et Cinq, et 18x30 pour Pointe et V (voir figure A.1).

FIG. A.1 - Les différentes postures (de gauche à droite) : A, B, C, Cinq, Pointe et V.

Pour rechercher une posture, les images sont explorées à différentes échelles

et les fenêtres sont testées à différentes positions par un réseau de neurones. Ce dernier donne la probabilité d'avoir une posture de la main.

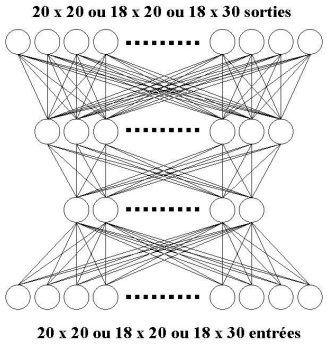

FIG. A.2 - Le modèle génératif contraint.

Les réseaux de neurones, tels les modèles discriminants [Murakami et Taguchi, 1991] ou les cartes de Kohonen [Boehm et al., 1994] ont été précédemment appliqués à la reconnaissance des postures de la main. Marcel et al. proposent d'utiliser un modèle déjà appliqué à la détection de visages : le modèle génératif contraint (CGM) [Féraud, 1997] (voir figure A.2).

Le but de l'apprentissage génératif contraint est d'évaluer la distance d'un point de l'espace d'entrée à l'ensemble des données que l'on cherche à apprendre. Pour cela, un réseau de neurones non-linéaires à compression est entraîné avec des exemples, mais aussi avec des contre-exemples. Chaque exemple de main est reconstruit à l'identique et chaque contre-exemple est contraint à être reconstruit comme une moyenne du voisinage de l'exemple le plus proche. Pour chaque posture, la classification est faite en mesurant la distance entre l'entrée présentée et la sortie obtenue (voir figures A.3, A.4).

Marcel et al. ayant pris connaissance de la méthode d'apprentissage que nous proposons (voir 4.4) ont décidé de l'appliquer sur leur modèle afin d'en améliorer les performances. L'expérimentation a été mené avec $\nu = 10$ ($\alpha_{var} = 0.001, \alpha_{quad} = 0.01$). Sur les figures A.3 A.4 on présente l'histogramme des distances de reconstruction sans la nouvelle fonction de coût ($\nu = 0$) et avec la nouvelle fonction de coût ($\nu = 10$). On remarque que la variance des distances de reconstruction sur les exemples est plus faible avec la méthode VMSE. L'amélioration ainsi apportée permet une meilleure classification entre les exemples et les contre-exemples.

Le tableau A.1 décrit les résultats obtenus sur les 6 postures de l'alphabet de Marcel et al. On constate les améliorations obtenues sur la classification des différentes postures réalisées grâce à la nouvelle fonction de coût. On précise que le critères d'arrêt des apprentissages a été l'erreur quadratique moyenne sur les ensembles de validation lorsque celle-ci ne diminue plus. La nouvelle méthode ayant donc été utilisée comme fonction de régularisation.

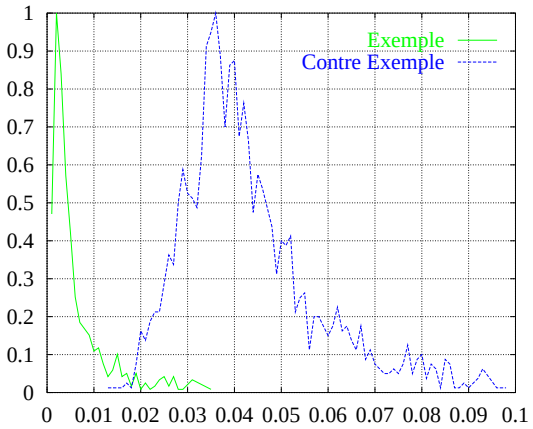

FIG. A.3 - Histogramme des distances de reconstruction pour la posture Pointe avec la méthode MSE.

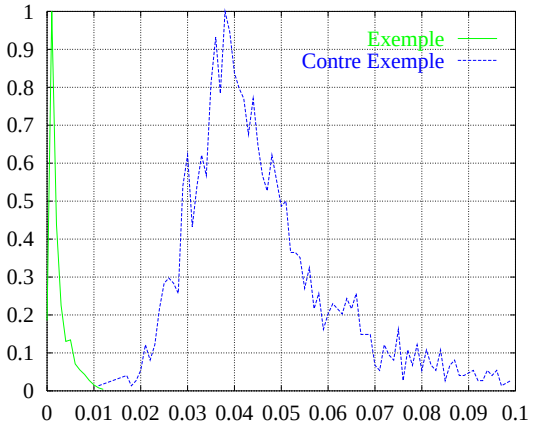

FIG. A.4 - Histogramme des distances de reconstruction pour la posture Pointe avec la méthode VMSE ($\nu = 10$).

TAB. A.1 - Les taux de classifications sur les différentes postures de la main avec MSE et VMSE.

Sur l'ensemble de test				
	Résultat MSE		Résultat VMSE	
	Taux	Fausse	Taux	Fausse
	détection.	Alarme	détection.	Alarme
Posture 'A'	95.68	1.03	98.27	0.51
Posture 'B'	96.61	0.56	100.0	0.00
Posture 'C'	97.33	1.04	98.66	0.52
Posture 'C5'	85.48	0.53	90.32	0.53
Posture 'P'	96.66	3.76	96.66	0.53
Posture 'V'	87.27	1.08	96.36	1.08

.

Index

Bibliographie

Abarbanel, H., Brown, R., Sidorowich, J., et Tsimring, L. (1993). The analysis of observed chaotic data in physical systems. *Review of Modern Physics*, 65 :1331–1392.

Aussem, A., Rouxel, S., et Marie, R. (1999). Neural-based queueuing system modelling for service quality estimation in communications networks. In *ICANN'99*.

Bae, J. et Suda, T. (1991). Survey of traffic control schemes and protocols in ATM networks. In *IEEE*, volume 79, pages 170–189.

Bean, N. G. (1993a). Robust connection acceptance control for ATM networks with incomplete source information. In *Annals of Operation Research*, volume 48, pages 357–379.

Bean, N. G. (1993b). *Statistical Multiplexing in Broadband Communication Networks*. PhD thesis, University of Cambridge.

Bean, N. G. (1994a). Dynamic effective bandwidths using network observation and the bootstrap. Technical report, Teletraffic Research Centre Report. Adelaida, http ://www.math.adelaide.edu.au/Applied/staff/nbean.html.

Bean, N. G. (1994b). Estimation and control in ATM networks. Technical report, Teletraffic Research Centre Report. Adelaida.

Becker, S. et Le Cun, Y. (1988). Improving the convergence of backpropagation learning with second order methods. In *Connectionist Models Summer School*, pages 29–37, Pittsburg.

Bengio, S., Clerot, F., Gravey, A., et Collobert, D. (1996). *Dynamical ressource reservation in an ATM network using neural network-based traffic prediction.*

Bernier, O., Collobert, M., Féraud, R., Lemaire, V., Viallet, J., et Collobert, D. (1998a). MULTRAK : a system for Automatic Multiperson Localization and tracking in real-time. In *International Conference on Image Processing.*

Bernier, O., Collobert, M., Féraud, R., Lemaire, V., Viallet, J., et Collobert, D. (1998b). MULTRAK : Un system de localisation et de suivi de personnes pour visioconférences. In *CORESA*, page 141.

Bertsekas, D. P. et Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. MA : Athena Scientific, Belmont. ISBN 1-886529-10-8.

Bishop, C. M., editor (1997). *Neural Networks and Machine Learning*, volume 168 of *F*, chapter I-11. NATO ASI Series.

Boehm, K., Broll, W., et Sokolewicz, M. (1994). Dynamic Gesture Recognition Using Neural Networks : A Fundament for Advanced Interaction Construction. In *SPIE, Conference Electronic Imaging Science and Technology*.

Boisseau, M., Demange, M., et Munier, J. M. (1994). *Réseaux ATM*. Eyrolles, Paris.

Bottou, L. (1991). *Une approche théorique de l'apprentissage connectioniste ; applications à la reconnaissance de la parole.* PhD thesis, Université de Paris 11, Orsay, France.

Boucheron, S. (1992). *Théorie de l'apprentissage*. Hermès, Paris.

Bourdila, A. et Hanbaba, R. (1984). Long term ionospheric radiopropagation predictions, choices of indices. In *Solar Terrestrial Predictions Proceeding*, pages 491–499, Workshop at Meudon France.

Boyer, P., Coudreuse, J. P., Gonet, P., Legras, J., et Pays, G. (1996). Réunion d'animation technique, le mode de transfert asynchrone (ATM). Technical report, France Telecom CNET. NT/LAA/DIR/02 et NT/LAA/RSL/68.

Boyer, P. E. et Tranchier, D. P. (1992). A reservation principle with applications to the ATM traffic control. In *Comput. Networks ISDN Systems*, pages 57,58. 24 :321-334.

Breiman, L. (1994). Bagging predictors. Technical Report TR-421, University of California, Berkley.

Breiman, L. et Spector, P. (1992). Submodel selection and evaluation in regression : The X-random case. *International Statistical Review*, 60 :291–319.

Brichet, F., Roberts, J., et Simonian, A. (1996). Estimation of blocking probabilities for variable bit rate traffic. Technical report, France Telecom CNET. NT/PAA/ATR/GTR/4777.

Canu, S., Sobral, R., et Lengelle, R. (1990). Formal neural networks as an adaptative model for water demand. In *International Neural Network Conference (INNC)*, pages 131–135, Paris France.

Changeux, P. J. (1983). *L'homme neuronal*. Changeux, P. J.

Clérot, F., Gouzien, P., Bengio, S., Gravey, A., et Collobert, D. (1997). Dynamical resource reservation scheme in an ATM network using neural-network-based traffic prediction. In *5th IFIP Workshop on Performance Modelling and Evaluation in ATM Networks*. Chapman Hall.

Collobert, M., Féraud, R., Le Tourneur, G., Bernier, O., Viallet, J. E., Mahieux, Y., et Collobert, D. (1996). LISTEN : System for Locating and Tracking Individual Speakers. In *Second International Conference on Automatic Face and Gesture Recognition*, pages 283–288, Vermont USA.

Cosmas, J. P., Petit, G. D., Lehnert, R., Blondia, C., et Kontovassilis, K. (1994). A review of voice, data, and video traffic models for ATM. In *European Transactions on Telecommunications*, volume 5. n. 2.

COST, . (1991). Performance evaluation and design of multiservice networks.

Cost, 96, 1 (1996). Methods for the performance evaluation and design of broadband multiservice networks. The COST 242 Final Report. Part I, Traffic Control June 4-5, 1996.

Cost, 96, 3 (1996). Methods for the performance evaluation and design of broadband multiservice networks. The COST 242 Final Report. Part III, Traffic models and queueing analysis, June 4-5, 1996.

Courcoubetis, C., Kesidis, G., Ridder, A., Walrand, J., et Weber, R. (95). Admission control and routing in ATM networks using inferences from measured buffer occupancy. In *IEEE Transactions on Communications*, volume 43. n. 2–4.

Doshi, B. T. (1994). Deterministic rule based traffic descriptors for broadband ISDN : Worst case behaviour and connection acceptance control. In *Proceedings of ITC*, volume 14. Elsevier Science.

Droz, P. (1996). Traffic estimation and resource allocation based on periodical wavelet analysis. In *sub. for publication*. http ://www.zurich.ibm.com/dro/.

Falhman, S. (1988). Faster-learning variations on back-propagation : An empirical study. In Touretzky, D., Hinton, G., et Sejnowski, T., editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages Morgan Kaufmann, 38–51.

Fessant, F. (1995). *Prédiction de séries temporelles par réseaux de neurones artificiels : application aux séries temporelles ionosphériques*. PhD thesis, Université de Rennes I.

Floyd, S. (1996). Comments on measurement-based admission control for controlled-load services. Technical report, Lawrence Berkeley National Laboratory. http ://www.aciri.org/floyd/admit.html.

Forum, T. A. (1995a). *ATM Forum Traffic Management Specification*. The ATM Forum, version 4.0 edition. ATM Forum/95-0013R8.

Forum, T. A. (1995b). *User-Network Interface Specification*. The ATM Forum, version 4.0 edition. ATM Forum/95-1434R8.

Forum, T. A. (1996). *Trafic management specification*. The ATM Forum, version 4.0 edition. ATM Forum/95-0013R10.

Féraud, R. (1997). PCA, Neural Networks and Estimation for Face Detection. *NATO ASI, Face Recognition : from Theory to Applications*, pages 424–432.

Féraud, R. et Bernier, O. (1997). Ensemble and modular approaches for face detection : a comparison. In *Neural Information Processing System*, volume 10, pages 472–478.

Geman, S., Bienenstock, E., et Doursat, R. (1992). Neural networks and the bias-variance dilemma. *Neural Computation*, 4 :1–58.

Gibbens, R., Kelly, F. P., et Key, P. B. (1995). A decision-theoretic aproach to call admission control in ATM networks. In *IEEE Journal on Selected Areas in Communications*, volume 13. n. 6.

Gibbens, R. J. et Kelly, F. P. (1997). Measurement-based connection admission control. In *International Teletraffic Congress*.

Gravey, A., Sévilla, K., Roberts, J., et Simonian, A. (1997). Le contrôle d'admission des connexions pour le service sbr de l'offre multiservice sur ATM. Technical report, France Telecom CNET. NT/DAC/ARP/GTR/5031.

Griffiths, R. et Key, P. (1994). Adaptive call admission control in ATM networks. In *Proceedings of ITC*, volume 14. Elsevier Science.

Guerin, R., Ahmadi, H., et Naghshineh, M. (1991). Equivalent capacity and its application to bandwidth allocation in high-speed networks. In *IEEE JSAC*, volume 9. n. 7.

Guillemin, F. (1999). *Modélisation mathématique pour le contrôle de trafic dans les réseaux temporels asynchrones.* PhD thesis, Université de Pierre et Marie Curie, Paris VI. Habilitation à diriger des recherches.

Handel, R., Huber, M. N., et S., S. (1995). *Comprendre ATM*. Addison-Wesley, France.

Hansen, L. K. et Salamon, P. (1990). Neural networks ensembles. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 12, pages 993–1001.

Hebb, D. O. (1949). *The organization bahaviour*. J. Wiley and Sons.

Heffes, H. et Lucantoni, D. M. (1986). Markov Modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE J. Sel. Areas Commun.*, SAC-4(6) :856–867.

Hinton, G. E. (1986). Learning distributed representations of concepts. In *8 Annual Conference of the Cognitive Science Society*, pages 1–12.

Hiramatsu, A. (1989). ATM communications network control by neural network. In *IEEE-IJCNN Washington (DC)*, volume 1, pages 259–266.

Hiramatsu, A. (1990). ATM communications network control by neural network. In *IEEE Trans Neural Networks*, volume 1, pages 122–130.

Hiramatsu, A. (1991). Integration of ATM call admission control and link capacity control by distributed neural networks. In *IEEE Journal on Selected Areas in Communications*, volume 9. n. 7.

Hiramatsu, A. (1994). ATM call admission control using neural network trained with virtual output buffer method. In *IEEE International conference on neural networks, ICNN94*, pages 3611–3616.

Holger, S. et Bengio, Y. (1998). Training method for adaptative boosting of neural networks. In *Neural Information Processing System*.

Hoptroff, R. (1993). The principles and practice of time series forecasting and business modelling using neural nets. *Neural Computation and application*, pages 59–66.

Hoptroff, R. G., Bramson, M. J., et Hall, T. J. (1991). Forecasting economics turning points with neural nets. In *IEEE INNS*, volume 1, pages 347–352, Seattle.

Hornik, K., Stinchcombe, M., et White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2 :359–366.

Hu, M. J. C. (1964). *Application of the Adaline System to Weather Forecasting*. PhD thesis, Standford electronic laboratory.

Humblet, P., Bhargava, A., et Hluchyj, M. (1993). Ballot theorem applied to the transcient analysis of nd/d/1 queues. In *IEEE/ACM Trans. on Networking*, volume 1, pages 81–95.

Hunke, H. M. (1994). Locating and tracking of human faces with neural network. Technical Report CS-94-155, CMU.

ITU (1996a). *Methods for cell level traffic control in B-ISDN*. ITU. Draft Recommendation E.73x.

ITU (1996b). Traffic control and congestion control in B-ISDN. Technical report, ITU, Geneva. Recommendation I.371 (pp 3, 4, 47, 48, 51, 54, 55, 59, 78, 116, 118).

ITU (1997). *Traffic control and congestion control in B-ISDN - ABT and ABR conformance definition*. ITU-T Recommendation I.371, Séoul.

Jacobs, R. A. (1988). Increased rates of convergence through learning rates adaptation. *Neural Networks*, 1 :295–307.

Jacobs, R. A. et Jordan, M. I. (1991). A competitive modular connexionist architecture. In *Neural Information Processing System 3*.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., et Hinton, G. E. (1991). Adaptative mixtures of local experts. *Neural Computation*, 3 :79–87.

Joos, P. et Verbiest, W. (1989). A statistical bandwidth allocation and usage monitoring algorithm for ATM networks. In *ICC*. paper 13.5.

Kleinrock, L. (1975a). *Queueing systems, Computer applications*, volume 2. Wiley, New-York.

Kleinrock, L. (1975b). *Queueing systems, Theory*, volume 1. Wiley, New-York.

Kohavi, R. et John, G. H. (1995). Automatic parameter selection by minimizing estimated error. In Kauffman, M., editor, *International Conference on Machine Learning*, volume 12, pages 304–311, San Francisco.

Kohonen, T. (1984). *Self-Organization and Associative Memory*. Springer-Verlag.

Kröner, H., Renger, T., et Knobling, R. (1994). Performance moedelling of an adaptive CAC strategy for ATM networks. In *Proceedings of ITC*, volume 14. Elsevier Science.

Lapedes, A. et Farber, R. (1987). Nonlinear signal processing using neural networks : prediction and system modelling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory.

Le Cun, Y. (1987). *Modèles connexionnistes de l'Apprentissage*. PhD thesis, Université Paris 6.

Le Cun, Y., Denker, J., et Solla, S. (1990). Optimal brain damage. In *Neural Information Processing Systems*, volume 2, pages 598–605.

Lemaire, V. (1997). Connection Admission Control in ATM network using an artificial neural network. In *Proceedings of Helnet 97 International Workshop On Neural Network, Montreux, Switzerland*. http ://www.robo.jussieu.fr/lemaire/publi/HELNET97.ps.

Lemaire, V., Bernier, O., Collobert, D., et Clérot, F. (1999a). Une nouvelle fonction de coût régularisante dans les réseaux de neurones artificiels : Application à la classification. In *Conférence d'Apprentissage, CAP99*. http ://www.robo.jussieu.fr/lemaire/publi/CAP99.ps.

Lemaire, V., Bernier, O., Collobert, D., et Clérot, F. (2000). A new method to increase the margin of multilayer perceptrons. *Neural Processing Letter*. http ://www.robo.jussieu.fr/lemaire/publi/NPL99.ps.

Lemaire, V. et Clérot, F. (1999). Estimation of the blocking probabilities in an ATM network node using Artificial Neural Networks for Connection Admission Control. In *International Teletraffic Congress*, volume 16, Edinburgh.

Lemaire, V., Clérot, F., et Collobert, D. (1999b). Using Neural Network for Measurement Based Connection Admission Control. In *submited to Journal of System Management*.

Li, S. Q. (1989). Study on information loss in packet voice system. In *IEEE Transactions on Communications*, volume 37, pages 1192–1202. n. 11.

Li, S. Q., Chong, S., et Hwang, C. L. (1995). Link capacity allocation and network control by filtered input rate in high-speed networks. In *IEEE/ACM Transactions on Networking*, volume 3.

Li, S. Q. et Hwang, C. L. (1993a). Queue response to input correlation functions : continuous spectral analysis. In *IEEE/ACM Transactions on Networking*, volume 1.

Li, S. Q. et Hwang, C. L. (1993b). Queue response to input correlation functions : discrete spectral analysis. In *IEEE/ACM Transactions on Networking*, volume 1.

Loncelle, J. (1990). *Contribution des réseaux connexionistes au traitement d'image bas niveau*. PhD thesis, Université Paris 11.

Marcel, S. (1999). Hand posture recognition in a body-face centered space. In *ACM SIGCHI Conference on Human Factors in Computer Systems*, Pittsburg, Pennsylvania USA.

Marcel, S., Bernier, O., et Collobert, D. (1999). Reconnaissance de la main pour les interfaces gestuelles. In *CORESA'99*, Sophia-Antipolis France.

Marot, M. (1997a). Etude du trafic multimédia. Master's thesis, Université de Paris 6. DEA Recherche opérationnelle.

Marot, M. (1997b). Etude du trafic multimédia. Technical report, IIE.

Mazaika, P. K. (1987). A mathematical model of the boltzmann machine. In *IEEE First International Conference on Neural Networks*, San Diego, Californie, 21-24 juin 1987.

Mc Culloch, D. W. et Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. In *Bull. Math. Biophysics*, volume 5.

McCullagh, P. et Nelder, J. (1989). *Generalized Linear Models*. Chapman and Hall, 2nd ed., London.

Melin, J. L. (1998). *Pratique des réseaux ATM*. Eyrolles, Paris.

Murakami, K. et Taguchi, H. (1991). Gesture recognition using Recurrent Neural Networks. In *CHI'91*, pages 237–242.

Nördstrom, E., Gallmo, O., Gustafsson, M., et Asplund, L. (1995). Neural Networks for adaptative traffic control in ATM networks. In *IEEE Communications Magazine*.

Norros, I., Roberts, J., Simonian, A., et Virtamo, J. (1991). The superposition of variable bit rate in an ATM multiplexer. In *Proceeding of IEEE JSAC*, number 3 in 9, pages 378–387.

Onvural, R. (1994). Asynchronous Transfer Mode networks : performance issue. In *Artech House*, Boston.

Park, D. C., El-Sharkawi, A., et Marks, J. (1991). Electric load forecasting using an artificial neural network. In *IEEE Transaction on Power Systems*, volume 6, pages 442–449.

Perrone, M. P. (1993). *Improving Regression Estimation : Averaging Methods for Variance Reduction with Extensions to General Convex Measure Optimization*. PhD thesis, Brown University, Institute - Brain and Neural Systems.

Plaut, D., Nowlan, S., et Hinton, G. (1986). Experiments on learning by back-propagation. Technical Report CMU-CS-86-126, Department of computer science, Carnegie Mellon University.

Quilan, J. R. (1998). Bagging, Boosting and C4.5. Technical report, University of Sydney.

Ragavan, H. et Rendell, L. (1993). Lokkahead feature construction for learning hard concepts. In Kauffman, M., editor, *International Conference on Machine Learning*, volume 10, pages 252–259, San Francisco.

Rathgeb, E. (1991). Modeling and performance comparison of policing mechanisms for ATM networks. In *Proceeding of IEEE JSAC*, number 3 in 9, pages 325–334.

Raviv, Y. et Intrator, N. (1996). Bootstrapping with noise : An effective regularization technique. *Connection Science*, 8 :355–372.

Rege (1994). Equivalent bandwidth and related admission criteria for ATM systems - a performance study. In *International Journal of Communication Systems*, volume 7, pages 181–197.

Remael, F. A. (1996). Le multiplexage statistique des connexions audiovisuelles. Technical report, France Telecom CNET. NT/LAA/RSL/82.

Riedmiller, M. et Braun, H. (1993). A direct adaptive method for faster backpropagation learning : The rprop algorithm. In *IEEE International Conference on Neural Networks*, San Francisco.

Roberts, J., Mocci, U., et Virtamo, J. (1996). *Broadhand Network Teletraffic*, chapter 2, page 58. Springer.

Roberts, J. et Virtamo, J. (1991). The supperposition of periodic cell arrival streams in an ATM multiplexer. In *IEEE Trans. Comm.*, volume 32, pages 298–303.

Roberts, J. W. (1991). Variable-bit-rate traffic control in B-ISDN. In *IEEE Communications Magazine*, volume 29, pages 50–57. n. 9.

Rowley, H., Baluja, S., et Kanade, T. (1995). Human face detection in visual scenes. Technical report, School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213, CMU-CS-95-158R.

Rumelhart, D. (1988). Learning and generalization. In *IEEE International conference on Neural Networks*, San Diego.

Rumelhart, D. E., Hinton, G. E., et Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distribued Processing : Explorations in the Microstructures of Cognition*, volume 1, pages 318–362.

Saito, H. (1990a). Call admission control in an ATM network without using traffic measurement. Technical report, IEIEC Technical Report. SSE89-155, http ://www.ieice.or.jp/.

Saito, H. (1990b). New dimensioning concept for ATM networks. In *7th Int. Teletraffic Congress Seminar*, Morristown.

Saito, H. (1992). Call admission control in an ATM network using upper bound of cell loss probability. In *IEEE Transactions on Communications*, volume 10. n. 7.

Saito, H. (1994). *Teletraffic Technologies in ATM Networks*. Artech House, Boston, London.

Saito, H. et Shiomoto, K. (1991). Dynamic call admission control in ATM networks. In *IEEE Journal on Selected Areas in Communications*, volume 9. n. 7.

Sarle, W. S. (1994). Neural network and statistical models. In *Proceeding of the Nineteenth Annual SAS Users Group International Conference*.

Schapire, R. E., Freund, Y., Bartlett, P., et Lee, W. (1997). Boosting the margin : A new explanation for the effectiveness of voting methods. In *Machines That Learn*. http ://www.research.att.com/ schapire/.

Simonian, A. (1991). Stationary analysis of a fluid queue with input rate varying as an orstein-uhlenbeck process. In *SIAM J. APPL. MATH.*, volume 51. n. 3.

Simonian, A. et Brichet, F. (1999). Conservative models for measurement-based admission control. In *International Teletraffic Congress, ITC16*.

Stamoulis, G., Anagnostou, M., et Georgantas, A. (1994). Traffic source models for ATM network : a survey. In *Proceeding of Computer communication*, number 6 in 17, pages 428–438.

Takens, F. (1980). Detecting strange attractors in turbulence. *Dynamical Systems and Turbulence, Springer-Verlag, Berlin*, 898 :366–381.

Tibshirani, R. (1996). A comparison of some error estimates for neural network models. *Neural Computation*, 8 :152–163.

Utgogg, P. E. et Brodley, C. (1990). An incremental method for finding multivariate splits for decision trees. In Kauffman, M., editor, *International Conference on Machine Learning*, volume 7, pages 58–65, San Francisco.

Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*. Springer-Verlag New York Heidelberg Berlin.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York Heidelberg Berlin.

Vapnik, V. N. et Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its applications*, 16 :264–271.

Vaton, S. (1998). *Modélisation statistique de trafic sur réseau local : application au contrôle dynamique de bande passante*. PhD thesis, ENST Paris.

Waibel, A. (1989). Modular construction of time delay neural networks for speech recognition. *Neural Computation*, 1.

White, H. et Hornik, K. (1989). Mutilayer feedforward networks are universal approximators. *Neural Network*, 2 :359–366.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5 :241–259.

Zamani, K. (1997). A real time MPEG-2 transport stream analyser. Technical report, NIST. http ://isdn.ncsl.nist.gov/.

Résumé

La théorie de l'apprentissage statistique est à la base des réseaux de neurones artificiels. On montre que les approches connexionnistes de la caractérisation du trafic et du contrôle d'admission des connexions (CAC) utilisant des mesures temps réel apportent de nombreux avantages par rapport aux méthodes paramétriques classiques. Elles s'adaptent facilement aux changements du trafic sur lequel aucune hypothèse préalable n'est nécessaire. Elles possèdent un haut niveau de performances et permettent de généraliser à des données inconnues les résultats appris.

On présente une nouvelle méthode destinée à améliorer les performances en généralisation des perceptrons multi-couches utilisés en tant que réseaux discriminants et approximateurs de fonctions. On montre comment modifier le critère d'apprentissage afin de contrôler la distribution des erreurs au cours de l'apprentissage. Ce contrôle permet d'obtenir une meilleure marge dans les problèmes de classification. Des résultats améliorant notablement l'état de l'art sur trois différents problèmes sont présentés et valident la méthode.

Une application de cette méthode à l'estimation des périodes de congestion dans un lien ATM est présentée afin de réaliser une procédure de contrôle d'admission des connexions pour le service de type ABT-DT. On montre que les réseaux de neurones artificiels entraînés sur des trafics qualifiés de "pire cas" peuvent correctement généraliser sur d'autres types de trafics, en réalisant une estimation conservative et précise des périodes de congestion. Cette méthode non paramétrique dynamique permet de décider l'acceptation d'une nouvelle connexion au regard de ses paramètres de trafic.

mots clés : ATM, apprentissage, classification, contrôle d'admission des connexions basé sur des mesures (MBAC), régularisation, marge, réseaux de neurones, service ABT, période de congestion.

Abstract

The theory of statistical learning provides foundations for artificial neural networks. We show that connexionnist approaches to traffic characterisation and to admission control using real time measurements exhibit numerous advantages over classical methods. They easily deal with traffic changes for which no prior hypotheses are needed. They are highly reliable and enable to generalise previously learned results to new data not known before.

A novel method is presented to enhance the generalisation performances of multi-layer perceptron used as discriminant networks and function approximators. We clearly show how to modify the learning criterium in order to control the error distribution. This control allows to obtain a better margin in classification problems. Results are given for three different problems. These results substantially improve state of the art results and validate the method.

This method is applied to estimate the congestion period in an ATM link to realise the connexion admission control procedure for ABT-DT services. It is shown that artificial neural network trained on worse-case traffic correctly generalise to other traffic to realize a conservative and accurate estimation of the congestion period. This non parametric dynamic method allows to correctly decide the acceptance of a new connection in regard of its traffic parameters.

keywords : ATM, learning, classification, measurement based connexion admission control (MBAC), regularisation, margin, neural networks, ABT service, congestion period.