Université Henri Poincaré – Nancy I

UFR S.T.M.I.A. **Département de Formation Doctorale en Informatique**
École Doctorale IAE + M

TOM, une architecture connexionniste de traitement de séquences. Application à la reconnaissance de la parole

THÈSE

présentée et soutenue publiquement le 19 décembre 1995

pour l'obtention du

Doctorat de l'Université Henri Poincaré – Nancy I
(Spécialité Informatique)

par

Stéphane Durand

Composition du jury

Président :	Jean-Marie PIERREL
Rapporteurs :	Yves BURNOD Guy MERCIER
Directeurs :	Jean-Paul HATON Frédéric ALEXANDRE
Examinateur :	René NATOWICZ

Centre de Recherche en Informatique de Nancy
INRIA–Lorraine

Résumé

La grande majorité des problèmes auxquels l'homme est confronté contient une composante temporelle. Reconnaissance de la parole, langage, motricité, génération de plans, et raisonnement sont quelques exemples d'applications que l'on tente aujourd'hui de traiter à l'aide de machines. Le connexionnisme, ou neuromimétisme, pour ses modèles les plus classiques, a montré ses capacités dans des tâches de reconnaissance de formes statiques. Il est moins approprié lorsque les tâches incluent une composante dynamique, c'est pourquoi de nouveaux systèmes, dits temporels, voient désormais le jour. Dans cette thèse, nous proposons une classification de ces systèmes selon la méthode de codage du temps, puis nous présentons le modèle TOM (Temporal Organization Map). Ce dernier est une carte neuronale dont le but est d'apprendre, de stocker et de reconnaître des séquences d'événements. Les unités connexionnistes qui la composent sont plus proches du modèle de la colonne corticale que du neurone formel de MacCulloch et Pitts. Par leur structure et leur fonctionnement, elles fournissent à la carte une tolérance aux distorsions temporelles. Ceci est particulièrement important pour des tâches sensorielles de bas niveau comme le traitement acoustique. L'architecture est générique et peut théoriquement s'appliquer à diverses modalités, à l'image du cortex biologique où une structure de base unique se retrouve sur toute la surface. Nous proposons une application de la carte TOM à la reconnaissance de chiffres parlés. Les résultats obtenus ouvrent des perspectives intéressantes pour ce genre d'architecture.

Mots-clés: Réseaux de neurones temporels, Traitement séquentiel, Propagation d'activité, Inspiration neurobiologique, Reconnaissance de la parole.

Abstract

Most real world problems to which the human being is confronted involve a temporal dimension. Speech recognition, language, motricity, scheduling and reasoning are some examples of applications which we attempt to simulate with machines. Connectionism with its classical models has shown its capabilities in static pattern recognition tasks. However, it is less adapted when tasks involve a dynamic component ; this is the reason why new systems are designed: temporal connectionist systems. We propose a classification of these systems according to their method of temporal coding and we present the TOM (Temporal Organization Map) model. This model is a neural map whose aim is to learn and recognize sequences of events. The connectionist units involved in the map are closer to the cortical column model than to the MacCulloch and Pitts neuron model. With their structure and functioning model, these units allow the map to be tolerant to temporal distortion. This is more especially important for sensory low level tasks such as acoustic processing. Like the brain cortex over which a basic structure is distributed, the TOM architecture is generic and can be theoretically applied to different modalities. We propose an application of TOM to spoken digits recognition. The results are interesting for future prospections.

Remerciements

Je remercie vivement :

- M. Jean-Marie Pierrel, professeur à l'Université Henri Poincaré, Nancy I, directeur du CRIN [1] et responsable de l'équipe Dialogue commune au CRIN et à l'INRIA [2] Lorraine, qui me fait l'honneur de rapporter la thèse et de présider ce jury.
- M. Yves Burnod, directeur de recherche à l'INSERM [3], qui est à l'origine de la modélisation de la colonne corticale et donc des idées qui sous-tendent cette thèse. Je le remercie d'avoir jugé ce travail et de siéger dans ce jury malgré sa charge de travail importante.
- M. Guy Mercier, professeur à l'ENSAT et chef de projet au CNET [4], pour l'intérêt qu'il a bien voulu accorder à mes travaux notamment dans le cadre de leur application à la reconnaissance de la parole. Je le remercie d'avoir juger mon travail et de siéger dans mon jury.
- M. Jean-Paul Haton, professeur à l'Université Henri Poincaré, Nancy I, responsable de l'équipe RFIA (Reconnaissance des formes et Intelligence Artificielle) commune au CRIN et à l'INRIA Lorraine. Je suis reconnaissant pour la confiance qu'il a placé dans ce travail ainsi que pour la rigueur scientifique qu'il a suscité en moi.
- M. René Natowicz, maître de conférence à l'ISTM [5], d'avoir bien voulu s'intéresser à ce travail. Qu'il soit à nouveau remercié pour les corrections qu'il a apporté à ce manuscrit.
- M. Frédéric Alexandre, chargé de recherche à l'INRIA Lorraine, responsable du groupe Cortex concernant les recherches en connexionnisme dans l'équipe RFIA. Je ne saurai comment exprimer mes remerciements à Frédéric, son esprit scientifique brillant et ses qualités humaines en ont fait pour moi bien plus qu'un «simple chef».

Enfin, je tiens à remercier toutes les personnes que j'ai pu cotoyer tout au long de ces trois années de thèse. Plus particulièrement, les membres du groupe Cortex, Brigitte, Emmanuel, Jean-Claude, Jean-Daniel, Jeff, Jérôme, Lionel, Nicolas, Pascal (itMan) et Yannick, les membres de l'équipe RFIA, Martine notre secrétaire.
Merci aussi à Claudine, Corinne, Lydie, Nathalies, Daniel, Jean-Mi, JP, Laurents, Serge, Thierry, ainsi qu'au personnel admistratif et technique du laboratoire et tous ceux qui ne sont pas cités ici mais qui restent présents à mon esprit.

Grâce à eux, ces dernières années ont été bien plus qu'une thèse.

Longue vie au groupe Cortex...

1. Centre de Recherche en Informatique de Nancy
2. Institut National de Recherche en Informatique et Automatique
3. Institut National de la Santé Et de la Recherche Médicale
4. Centre National d'Etudes des Télécomunications
5. Institut Supérieur de Technologie et de Management

Cette thèse est dédiée à toutes celles et à tous ceux qui ont contribué de près ou de loin,
directement ou indirectement, à son existence ;
en particulier à mes parents.

Table des matières

Table des figures

Avant-propos

Dans l'idée abstraite que l'Homme, scientifique ou non, se fait d'une « machine intelligente » il y a souvent l'image d'un humanoïde, un robot à l'apparence humaine doué de facultés proches de celles de l'homme. L'intérêt de recherches pour la conception de « machines intelligentes » ne réside pas seulement dans ce but ultime, il participe également à la compréhension du fonctionnement de l'homme en interaction avec son environnement. Encore loin de l'aboutissement imaginé par les auteurs de science-fiction, les travaux de recherche concernant « l'intelligence artificielle » ne peuvent pas se cloisonner dans un champ scientifique donné parce que la machine humaine est un tout et que ce tout n'est vraissemblablement pas la somme de ses parties.
Ainsi, les neurosciences naviguent-elles dans des domaines aussi divers que la neurobiologie, la psychologie, la pharmacologie, les mathématiques, l'informatique, ...
Dans cette thèse, trois domaines cohabitent : la neurobiologie, le traitement de la parole et l'informatique. Le souci d'intégrer des connaissances de ces domaines dans un travail homogène pourra peut-être faire penser aux spécialistes que les idées présentées sont parfois réductrices et les modèles parfois trop caricaturaux. Nous en sommes conscient mais ces simplifications sont le prix à payer aujourd'hui si l'on souhaite un jour aller plus en avant... vers un modèle du cerveau.

« Nous « réduirons » certainement un jour par la voie expérimentale la pensée à des mouvements moléculaires et chimiques dans le cerveau ; mais cela épuise-t-il l'essence de la pensée ? »

Dialectique de la nature, Friedrich Engels, 1874

Chapitre 1

Introduction

« Il est absurde de ne pas accorder à l'âme la connaissance de soi et surtout de la refuser même à la nature de l'intelligence; il est tout à fait absurde qu'elle ait la connaissance des autres choses, sans s'établir dans la connaissance et la science d'elle-même. Car c'est la sensation et, si l'on veut, la réflexion et l'opinion qui perçoivent des choses extérieures à elles, mais non pas l'intelligence. L'intelligence a-t-elle ou non connaissance de ces choses extérieures, il convient de l'examiner; en tout cas, l'intelligence connaît évidemment les intelligibles. »

Plotin, les Ennéades V, 3, 205-270.

Arrivant relativement tard chez l'enfant, la compréhension des phénomènes évolutifs, tels une histoire ou un conte, ne pose aucun problème à l'adulte. En revanche, lorsque l'on demande à un jeune enfant de retracer une succession d'événements que l'on vient de lui montrer, l'échec est quasi certain [Piaget, 1946]. Une étape de maturation cérébrale concernant l'aspect temporel intervient donc à un moment où, loin d'être capable de tenir un raisonnement, le cerveau n'en demeure pas moins équipé pour l'apprentissage et la reconnaissance de traits sensoriels [Dehaene-Lambertz et Dehaene, 1994]. Parce que l'Homme, depuis le début de son existence, a toujours voulu inventer et fabriquer des objets de toutes sortes lui permettant de s'affranchir de certaines tâches, il tente aujourd'hui, après avoir mécanisé son environnement, de concevoir des machines dites « intelligentes » capables de raisonner, de réfléchir, de penser comme lui. Les observations du comportement humain, telles celles concernant l'intégration du temps chez l'enfant, qu'elles soient neurophysiologiques ou psychologiques, permettent d'entrevoir des idées pour la définition de machines travaillant à notre image, bref pour la construction d'une « intelligence artificielle ».

L'Intelligence Artificielle (IA), en tant que domaine scientifique, est née en 1956 à Dartmouth aux Etat-Unis grâce à cinq chercheurs de disciplines scientifiques variées, J. McCarthy, M. Minsky, C. Shannon, A. Newell et H. Simon, qui se proposèrent l'étude de programmes d'ordinateurs doués d'une certaine intelligence. Cette naissance n'est pas le point de départ des idées dans ce domaine. Dès le début du siècle, des chercheurs comme Turing, Wiener, Boole, Gödel et d'autres encore préparèrent le terrain dans le domaine de la théorie des automates et de la logique mathématique [Haton et Haton, 1989]. Mais déjà, lors des siècles passés et même de l'antiquité, les philosophes s'intéressaient à l'Intelligence et plus particulièrement à l'apprentissage. Les deux grandes tendances actuelles de l'Intelligence Artificielle, à savoir l'IA symbolique et l'IA connexionniste, trouvent leurs origines dans deux courants de pensées que l'on peut faire remonter à Aristote et Platon et qui ont été plus formalisés à partir du XVII^{e} siècle.Il s'agit de l'empirisme et du rationalisme [Bechtel et Abrahamsen, 1991]. Les rationalistes conduits par les célèbres Descartes, Spinoza et Leibniz croyaient aux idées innées. A l'inverse, les empiristes parmi lesquels Bacon, Locke, Berkeley et Hume reprochaient aux inéistes leur rigidité et proposaient de tenir compte de l'expérience sensible dans le monde. Bien qu'au début de notre siècle la frontière de ces courants ne fût plus aussi tranchée, l'IA s'est décomposée dès sa création dans les années 50 en deux courants que l'on pourrait assimiler aux courants rationaliste et empirique : « making a mind » et « modeling the brain ». Un programme d'ordinateur conçu dans la première vision des choses devrait fournir des résultats corrects dès son lancement (vision rationnelle) alors qu'un programme défini dans la seconde vision devrait interagir avec son environnement pour pouvoir fournir des résultats corrects. Cette deuxième école de pensée a donné naissance au connexionnisme ou neuromimétisme : une science qui se propose l'étude et la définition de modèles inspirés du système nerveux dans le but d'aboutir à des comportements similaires à ceux des êtres vivants.

Tantôt plébiscité, tantôt attaqué, le connexionnisme n'en demeure pas moins, aujourd'hui, un domaine en plein essor. Il s'inscrit dans le cadre général de l'Intelligence Artificielle tout en constituant une voie parallèle à l'IA « classique ». Après s'être attaqué à la reconnaissance de formes tels des caractères manuscrits, le neuromimétisme, suivant l'évolution de la maturation cérébrale, doit désormais tenir compte des phénomènes temporels. En effet, tous les problèmes difficiles que l'être humain se propose de traiter à l'aide des machines incluent la dimension temporelle. Reconnaissance de la parole, évolution d'un robot, plan d'actions, raisonnement sont quelques exemples de problèmes pour lesquels les phénomènes de contexte et d'ordre d'événements sont capitaux. L'intérêt de la prise en compte de ces phénomènes dans les réseaux connexionnistes ne se cantonne pas aux seuls aspects pratiques. Il participe, parmi d'autres recherches en cours, à l'élaboration de nouvelles architectures connexionnistes dont le but est de dépasser la « simple » reconnaissance des formes.

1.1 L'évolution du connexionnisme

Se trouvant à la croisée de la neurobiologie, des mathématiques, de la physique, de la psychologie et de l'informatique, le connexionnisme doit son premier modèle à MacCulloch et Pitts [MacCulloch et Pitts, 1943] avant sa naissance officielle en 1956. En effet, ces deux chercheurs donnèrent pour la première fois un modèle du neurone biologique : le neurone formel. La découverte est de taille puisque ce modèle va constituer la base fondamentale et la pierre angulaire des systèmes connexionnistes jusqu'à nos jours. En fait, nous pouvons dire que l'intervention de la neurobiologie dans les systèmes connexionnistes va généralement s'arrêter là. A partir du premier perceptron dans les années 50, s'est développée toute une panoplie de modèles attachés à des sciences dures (mathématiques, physique) plutôt qu'à des sciences du vivant (biologie, psychologie, ...). C'est ainsi que le modèle de Hopfield [Hopfield, 1982] inspiré de la théorie physique des verres de spin ou que la rétro-propagation du gradient de l'erreur dans les perceptrons multi-couches ont vu le jour. Parallèlement à cela, on peut noter les célèbres cartes de Kohonen [Kohonen, 1989] dont les fondements mathématiques ont été extensivement étudiés et qui présentent une analogie avec les aires corticales du cerveau, en particulier les aires sensorielles. L'idée de l'automate cellulaire à seuil est présente mais abstraction est faite des différentes spécificités cellulaires, des échanges chimiques, des temps de propagation des influx nerveux, de la composition des tissus et de bien d'autres choses encore que les neurobiologistes ont parfois du mal à appréhender. Alors cette modélisation peut parfois leur paraître « brutale ». Tous ces éléments que nous venons de mettre en lumière, et qui ne sont pas, en général, pris en compte dans les modélisations, participent peut-être à l'intelligence. La recherche le dévoilera sans doute un jour ; en attendant ceci a mis en exergue un courant de pensée plus biologique dans le domaine du connexionnisme.
Même si ce courant n'est pas majoritaire, il a vu naître un certain nombre de modèles originaux. Citons en premier lieu le modèle de la colonne corticale [Burnod, 1988; Alexandre, 1990; Guyot, 1990; Alexandre *et al.*, 1991]. Ce modèle est une source d'inspiration importante dans le travail de cette thèse. Dans la même philosophie, nous trouvons les travaux de D. Béroule et son modèle à propagation guidée [Béroule, 1985; Béroule, 1990] où l'évolution d'un phénomène physique (d'un signal de parole en particulier) guide une propagation d'activité dans une couche d'automates. Il est remarquable que pour une grande majorité des modèles d'inspiration fortement neurobiologique l'aspect séquentiel du traitement de l'information soit souvent présent. C'est le cas notamment dans le modèle d'apprentissage de séquences temporelles par sélection de S. Dehaene, JP. Changeux et JP. Nadal [Dehaene *et al.*, 1987], dans la colonne de mémorisation à court terme de B. Ans [Ans, 1990; Ans *et al.*, 1994] ou encore dans les modèles à neurones intégrateurs à fuites [Wang et Arbib, 1990; Chappell et Taylor, 1993].

On le voit encore aujourd'hui, il n'existe pas de voie unique pour traiter le problème de l'intelligence. Ne serait-ce que parce que, en premier lieu, il est difficile de définir le mot « intelligence ». Le connexionnisme s'est principalement intéressé jusqu'à présent à la perception. L'ancêtre des modèles connexionnistes, le perceptron [Rosenblatt, 1958], stimula fortement les recherches dans le domaine, mais les limitations démontrées par Minsky et Papert [Minsky et Papert, 1969] portèrent un coup d'arrêt à l'activité connexionniste

dans le monde. Seuls quelques irréductibles persévérèrent et mirent au point de nouveaux algorithmes dont la célèbre rétro-propagation du gradient [Le Cun, 1985; Rumelhart *et al.*, 1986]. Aujourd'hui les capacités de classification des réseaux de neurones ne sont plus à démontrer mais ces qualités s'estompent rapidement lorsque, comme dans le cas de la parole, les formes à reconnaître dépendent du contexte, du passé.
De ce point de vue, ces dernières années ont vu la naissance de travaux de recherche mêlant les systèmes connexionnistes à d'autres techniques. Ceux-ci ont donné lieu aux modèles hybrides. On parle ainsi de modèles symbolico-connexionnistes où l'on tente d'introduire la notion de symbole plutôt propre à l'IA classique [Lallement et Durand, 1994; Lallement *et al.*, 1995], ou encore de modèles mélangeant des techniques neuronales à d'autres techniques comme les modèles de Markov [Bourlard et Morgan, 1990; Bourlard *et al.*, 1995] ou la programmation dynamique [Iso et Watanabe, 1990].

En fait, par rapport aux trois grands courants que nous venons de citer (connexionnisme « mathématique », connexionnisme d'inspiration neurobiologique et modèles hybrides), les deux derniers, plus récents, doivent leur existence au même but : donner plus d'intelligence aux réseaux connexionnistes. Les modèles connexionnistes classiques savent classer, organiser des données que l'on peut qualifier de sensorielles mais l'intelligence ne s'arrête pas à ce simple traitement statistique des données. Elle manipule ces données, en infère d'autres, bref, n'a rien à voir avec une fonction mathématique ou une boîte noire classiquement associée aux réseaux de neurones traditionnels. En IA classique, le problème est inverse, les méthodes traitent des symboles et travaillent à des niveaux d'abstraction beaucoup plus élevés que les réseaux neuronaux. Du même coup, ces méthodes se révèlent incapables de traiter de l'information sensorielle forcément numérisée à bas niveau. La granularité des systèmes d'IA classiques est donc trop grosse, d'où des tentatives pour la réduire. Les systèmes multi-agents [Chevrier, 1993] sont des exemples dans ces tentatives de réduction. En revanche, on note l'effet inverse pour les réseaux connexionnistes où l'on choisit une granularité fine pour aller vers de plus hauts niveaux d'abstraction.

Notre approche se situe dans ce contexte, avec une inspiration neurobiologique conduisant à l'utilisation d'unités de traitement d'abstraction supérieure à celle des neurones formels classiquement utilisés en connexionnisme. Bien évidemment, il ne s'agit pas de fabriquer un cerveau artificiel en copiant le nôtre. Il s'agit d'utiliser quelques idées et principes issus de la neurobiologie. On notera tout de même le penchant de l'Homme et du chercheur à s'inspirer et à utiliser le vivant : on parle aujourd'hui de réseaux de neurones mais aussi d'algorithmes génétiques [Goldberg et Holland, 1988], de processeurs neuronaux [Deville, 1994] et même d'ordinateurs moléculaires [Mashaal, 1995] !

1.2 Nature dynamique des problèmes

Nous n'allons pas revenir sur le bien fondé de tenir compte de l'aspect dynamique des problèmes. Actuellement, vous lisez ces quelques lignes et pour comprendre la signification des phrases (en espérant qu'elles soient compréhensibles), vos yeux balayent dans un ordre bien précis les pages de la thèse. Avant même de parler de traitement séquentiel dans les aires du raisonnement dans le cerveau, les muscles de vos yeux opèrent un plan d'action séquentiel parfaitement défini. C'est ce traitement séquentiel qui permet d'envoyer dans le bon ordre les syllabes et les mots eux même traités séquentiellement par des aires corticales de haut niveau. Nous observons donc que l'information qui est partie de l'encre sur cette page voyage de façon ascendante vers les niveaux d'abstraction élevés du cerveau. Mais rien n'empêche de contrôler la séquentialité par ces niveaux d'abstraction élevés. Suivante phrase la passer et droite à gauche de relire pouvez vous. La phrase précédente n'a pas de sens en français, sauf si je vous demande de modifier le programme séquentiel de vos yeux et de la relire à l'envers, ceci pour illustrer les interactions permanentes qui existent entre niveaux d'interaction haut et bas ou entre traitements de nature différente. Dans le paragraphe suivant, nous nous attardons sur le problème de la reconnaissance de la parole qui est par nature un problème à forte composante temporelle. Ce problème nous intéressera comme domaine d'application de nos travaux. On peut bien sûr citer d'autres problèmes qui pourraient être accessibles aux modèles connexionnistes temporels : compréhension et génération de mouvements par exemple. Enfin le raisonnement est également un problème séquentiel mais il paraît utopique de tenter de le traiter avec les systèmes connexionnistes actuels.

1.3 La reconnaissance de la parole

Le traitement automatique de la parole est un problème difficile, ne serait-ce que parce qu'il recouvre plusieurs niveaux d'abstraction : de la perception du signal auditif à la compréhension du langage codé par l'onde sonore [Calliope, 1989].

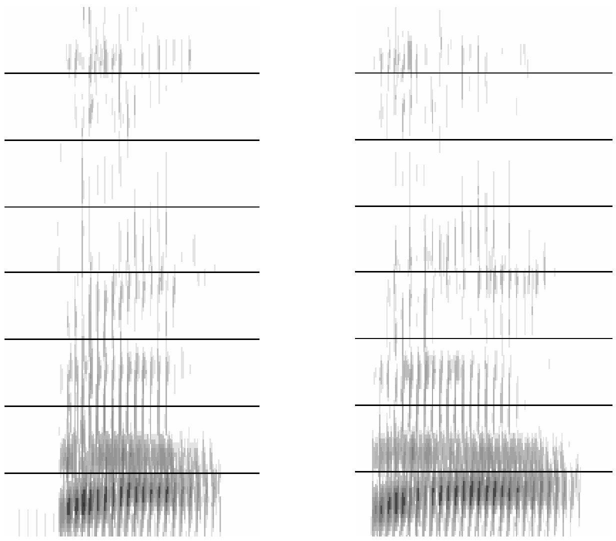


FIG. 1.1 – *Spectrogrammes de la même syllabe « ba » prononcée deux fois par le même locuteur*

Au niveau sensoriel, les difficultés rencontrées sont principalement dues à l'extrême complexité du signal de parole. En effet, rarement le signal est pur et contient uniquement le message à analyser. L'environnement n'étant jamais silencieux en situation réelle, le message est noyé dans l'environnement bruité rendant difficiles l'extraction et l'interprétation du signal utile. A cela il faut ajouter les fortes variations que peut subir un signal pour une même prononciation. Il y a bien sûr la variabilité inter-locuteur inhérente aux différences morphologiques du conduit vocal d'un individu à un autre mais aussi la variabilité intra-locuteur due aux variations de la vitesse d'élocution par exemple, ou à l'état physique lors de la prononciation. La figure 1.1 montre deux spectrogrammes (résultat d'un pré-traitement fréquentiel sur le signal, représentation temps x fréquence) d'une même syllabe ("ba") prononcée deux fois par le même locuteur. On remarquera que ces deux spectrogrammes ne sont pas superposables...

L'analyse d'une portion de signal montre une forte dépendance au contexte. Par exemple, pour un phonème particulier nous ne pouvons pas établir ses caractéristiques physiques (par exemple fréquentielles) de façon universelle, le modèle propre du phonème n'existe pas. En effet, la structure dépend du contexte de la prononciation. La figure 1.2 montre le spectrogramme du phonème /i/ prononcé dans deux contextes différents (« biz » à gauche et « dis » à droite).

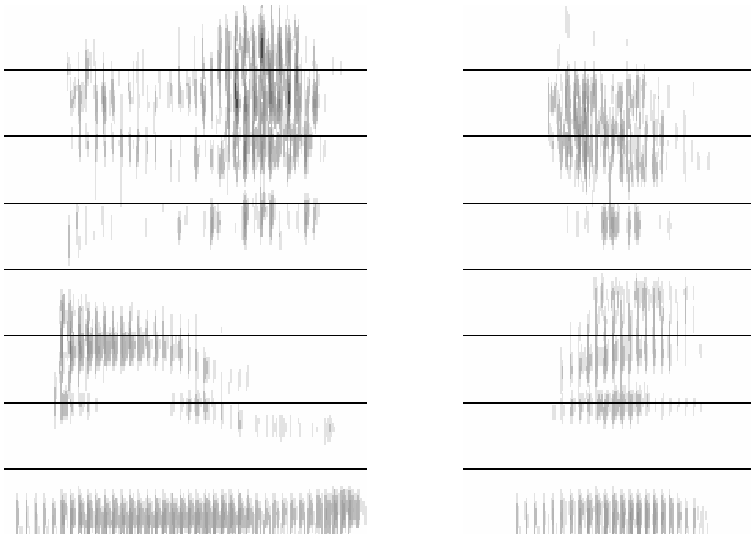


FIG. 1.2 – *Spectrogrammes du phonème /i/ prononcé dans deux contextes différents (« biz » à gauche et « dis » à droite)*

On remarque notamment que les positions des hautes énergies (barres noires sur les graphiques) ne sont pas identiques d'une prononciation à une autre. Le contexte influence cette position.

C'est l'aspect contextuel et dynamique de la parole qui nous intéresse ici par rapport à l'architecture connexionniste que nous allons présenter dans ce mémoire.

1.4 Limites du connexionnisme

Sous sa forme actuelle, le connexionnisme ne peut pas prétendre à faire plus que du traitement sensoriel de bas niveau. Certes, il y a les modèles phares du connexionnisme, les perceptrons multi-couches, dont les performances ne sont plus à démontrer et qui ont fait leur entrée dans le milieu industriel [Pican, 1995]. Seulement, ces modèles, aussi performants qu'ils soient, n'en demeurent pas moins des modélisateurs de fonctions mathématiques permettant d'effectuer de la classification de données, de donner une réponse aussi précise que possible en fonction d'une configuration fournie en entrée. Cela n'est pas une critique en soi : apprendre ces fonctions mathématiques dans des applications réelles peut se révéler très complexe.
Revenons sur l'exemple du traitement automatique de la parole. Il existe dans la littérature un certain nombre de modèles connexionnistes appliqués à la reconnaissance de la parole. Dans leur grande majorité, ils classifient les formes en entrée, bref effectuent une analyse statistique des données. Dans ce cas, nous aurions plutôt tendance à nommer ceci « traitement du signal ». En effet, la reconnaissance de la parole en tant que telle fait intervenir bien plus qu'une simple analyse du signal. En l'occurrence, un mot dans une phrase est reconnu parce que l'auditeur possède des informations acoustiques mais aussi parce qu'il est capable d'apporter des informations complémentaires sur ce mot au vu de sa position dans la phrase par exemple. Il n'est donc pas déraisonnable de penser que plusieurs modules de traitement coopèrent pour un bon fonctionnement des tâches que l'Homme traîte, aussi simples qu'elles puissent paraître.
Dans ce contexte, les architectures connexionnistes classiques ne nous semblent pas adaptées pour des problèmes réels. En fait, les modèles connexionnistes sont vraisemblablement capables de résoudre de façon ponctuelle de petites parties dans une chaîne de traitement globale d'une tâche réelle. Par exemple, pour une scène, il faut tout d'abord se focaliser sur différents endroits de la scène (un module connexionniste peut s'occuper de cela). Il faut, pour chaque portion d'image, extraire des invariants ; nous pouvons envisager l'utilisation d'un modèle connexionniste compétitif pour cela. Enfin, pour connaître la forme dans la portion, un perceptron peut très bien faire l'affaire. Il ne reste plus qu'à rassembler toutes les données issues de chaque portion afin d'établir le prototype de l'image complète, on peut alors utiliser un réseau prototypal. Bien entendu, l'énoncé de la méthode est beaucoup plus simple à dire qu'à réaliser. En particulier il faut établir l'interfaçage entre tous ces modules. Or les architectures ne sont pas unifiées et ne sont adaptées qu'aux applications pour lesquelles elles ont été conçues. C'est pourquoi, dans l'approche que nous allons décrire dans cette thèse, nous essayons de définir une architecture unique utilisable à différents niveaux d'abstraction.

Mais ceci n'est pas le point central de nos investigations. Celles-ci concernent la prise en compte des phénomènes temporels auxquels la majorité des systèmes connexionnistes actuels restent indifférents.

1.5 Le cadre du travail, notre approche et ses motivations

Parce que le connexionnisme a quelque chose de fascinant, la plupart des gens ne reste pas insensible à son attrait. Certains n'y croiront pas parce que, selon eux, le cerveau est beaucoup trop complexe et recèle beaucoup trop de mystères. Le modéliser relèverait d'une utopie. D'autres pensent au contraire que modéliser le cerveau n'est certainement pas la plus mauvaise voie pour aboutir à des machines «intelligentes». Nous faisons partie de cette dernière catégorie.

Le choix délibéré de ne pas se situer dans les modèles classiques du connexionnisme relève d'une démarche personnelle (qu'il est parfois difficile de justifier dans la mesure où il tient compte d'une certaine sensibilité humaine) mais aussi de l'histoire de notre groupe de recherche. Celui-ci fut à l'origine des premiers développements informatiques de la colonne corticale. Modèle général s'il en est, il a notamment induit des projets de recherche dont les objectifs sont de palier les manques du modèle original de la colonne bien sûr mais aussi du connexionnisme en général. En particulier, la prise en compte des phénomènes temporels n'est pas complètement élucidée dans l'architecture générique de la colonne corticale. Les problèmes de mémorisation de l'information et ceux concernant les aspects symboliques dans les réseaux se posent également au travers de la colonne [Beaugé *et al.*, 1994]. Notre travail consiste à étudier la possibilité de prendre en compte la dimension temporelle dans les réseaux connexionnistes avec pour contexte de travail le modèle de la colonne corticale.

L'objectif est donc la conception d'un modèle permettant la prise en compte des aspects dynamiques tout en voulant aboutir à une architecture utilisable pour différents problèmes et différents niveaux d'abstraction, c'est à dire un modèle générique dans la philosophie de la colonne corticale.

Tout au long de cette thèse, nous considérons un problème dynamique sous la forme de séquences d'information. Par exemple, un mot ou une syllabe est une séquence d'indices acoustiques. Nous nous proposons donc de définir une architecture permettant de traiter ces séquences tout en étant conscient qu'elles peuvent être soumises à de fortes variabilités spatiales et temporelles en fonction du niveau d'abstraction. Un système pouvant traiter les séquences possède donc au moins les propriétés suivantes :

- tolérance aux bruits spatiaux (variabilité au niveau des événements ponctuels) ;
- tolérance aux distorsions temporelles.

Nous définissons une architecture dont les entités neuronales de base sont plus proches du modèle de la colonne corticale que du neurone formel de MacCulloch et Pitts. Nous appellerons «super-unités» ces entités neuronales. L'idée fondamentale de la thèse est alors d'associer à toute forme dynamique une propagation d'activité au sein d'une assemblée de ces super-unités de base (fig. 1.3). La propagation est controlée par des paramètres de tolérance aux distorsions temporelles associés à chaque super-unité. Ces paramètres peuvent être fixés *a priori* ou appris. L'assemblée de super-unités est organisée sous la forme d'une carte que nous appelons TOM (Temporal Organization Map).

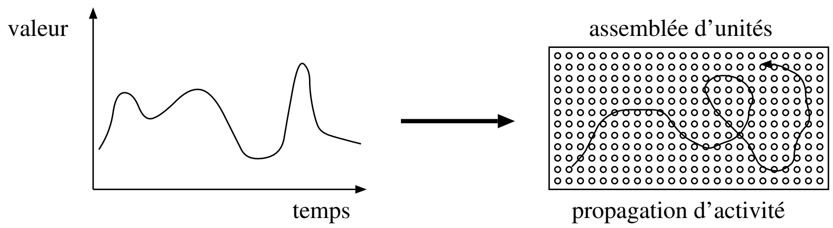


FIG. 1.3 – *Forme dynamique et propagation d'activité*

Afin d'expérimenter et valider notre approche, nous appliquons différents types d'architecture conçus à base de cartes TOM pour des tâches de reconnaissance de la parole.

1.6 Plan de la thèse

Le chapitre qui suit introduit les bases neurobiologiques qui sous-tendent les modèles connexionnistes. En particulier, nous relatons le fonctionnement du neurone biologique conduisant à sa modélisation. Nous poussons un peu plus loin les références neurobiologique pour aboutir aux fonctionnements corticaux et plus spécialement nous introduisons la colonne corticale. Bien que les données dans ce domaine ne foisonnent pas, nous traitons succinctement l'aspect dynamique du traitement cortical de l'information en donnant quelques exemples et illustrations d'aspects dynamiques connus dans le cerveau. Du cerveau au connexionnisme, nous introduisons ensuite les grandes classes de réseaux neuronaux classiques supervisés ou non ainsi que le modèle de la colonne corticale constituant une base d'idées pour nos travaux.

Le chapitre 3 s'intéresse plus particulièrement aux modèles connexionnistes temporels. Il en existe une multitude mais nous pouvons les regrouper dans trois grandes catégories suivant le mode de représentation du temps : représentation externe, représentation interne implicite ou explicite. Ceci nous permettra de situer notre approche dans cette dernière classe.

Dans le chapitre 4, nous présentons notre modèle. En fait, il se décline sous la forme d'une architecture de base, la carte TOM qui s'utilise pour la conception d'architectures plus complexes. Un nouveau type d'unité connexionniste est introduit afin de prendre en compte les aspects temporels. Nous détaillons notamment les connexions qui s'établissent au sein d'une carte d'unités ainsi qu'entre différentes cartes. Nous introduisons en outre un modèle d'architecture supervisée composé de cartes TOM. Enfin, nous discutons de la possibilité d'apprendre différents degrés de tolérance aux distorsions temporelles le long des séquences.

Dans le cadre d'applications à la reconnaissance de la parole, nous présentons brièvement au chapitre 5 quelques aspects du traitement cortical de l'information auditive et ceci d'un point de vue modélisation. Nous présentons ensuite quelques approches connexionnistes impliquées dans le traitement automatique de la parole.

Le chapitre 6 présente les expérimentations effectuées sur le modèle TOM dans le cadre du traitement de la parole. Nous présentons notamment les tests effectués sur la carte elle même puis ceux effectués sur des architectures combinant plusieurs cartes.

Enfin, le chapitre 7 conclut la thèse. Il développe notamment les perspectives associées à nos travaux ainsi que d'autres applications envisageables.

Chapitre 2

Du cerveau au connexionnisme

« Les sens nous représentent le cours des planètes comme s'effectuant tantôt en avant, tantôt en arrière, et il n'y a là ni erreur ni verité, parce que tant qu'on se borne à n'y voir qu'un phénomène, on ne porte pas du tout encore un jugement sur la condition objective de leur mouvement. Mais parce que, si l'entendement ne prend bien garde d'empêcher que ce mode subjectif de représentation ne soit tenu pour objectif, il peut facilement se produire un jugement faux, l'on dit qu'elles semblent rétrograder; l'apparence ne doit pas être mise au compte des sens, mais bien au compte de l'entendement à qui seul revient un jugement objectif d'après les phénomènes. »

Emmanuel Kant, Prolégomènes à toute métaphysique future, 1783.

2.1 Introduction

Ce chapitre a pour but de situer d'une part le domaine dans lequel nous travaillons, d'autre part de dresser l'état des lieux à partir duquel nous avons développé notre modèle. L'exposé s'articule en deux parties.
La première rappelle quelques faits neurobiologiques qui sous-tendent les bases du connexionnisme. Après avoir discuté du fonctionnement du neurone et du message nerveux, nous nous intéresserons à son implication dans les colonnes corticales. A un niveau plus macroscopique, nous parlerons de l'organisation fonctionnelle du cerveau et de son organisation en aires corticales. Enfin, même si cet aspect est moins connu en neurobiologie, nous aborderons l'aspect dynamique du traitement de l'information dans le cerveau.

La seconde partie présente les modèles classiques en connexionnisme ainsi que le modèle de la colonne corticale directement inspiré de la colonne corticale biologique. D'un point de vue plus technique, nous parlerons de la modélisation du neurone conduisant à la majorité des réseaux neuronaux d'aujourd'hui. Ces derniers, même s'ils tirent leurs origines dans la biologie, sont souvent des modèles mathématiques ou d'inspiration émanant des sciences physiques. Il nous semble cependant important d'en parler ici car, non seulement ils sont partie intégrante du connexionnisme, mais ils servent parfois de catalyseurs dans des modèles plus complexes et intégrant plus d'analogie avec le vivant. Enfin, nous décrirons le modèle de la colonne corticale dont les principes seront utilisés dans notre modèle que nous formulerons au chapitre 4.

2.2 Le cerveau

2.2.1 L'unité de base: le neurone

Architecture

Le cerveau est composé de plusieurs dizaines de milliards de cellules nerveuses ou neurones. Entité fondamentale du fonctionnement cérébral, le neurone n'en demeure pas moins relativement simple du point de vue du modélisateur lorsqu'il est étudié de façon isolée. Toute chose étant relative, un neurone est cependant connecté à un bon millier de ses homologues et on estime à 10^{14} le nombre total de ces connexions, ou synapses, dans le cerveau. C'est ce formidable réseau d'interconnexions entre automates qui génère l'énorme complexité du cerveau, cachant du même coup les secrets de son fonctionnement. Nous pouvons d'ailleurs constater que le cerveau, étant devenu si complexe au fur et à mesure de l'évolution animale, en arrive à un point où il tente désormais de s'auto-comprendre ! Quelles sont les fonctionnalités d'un neurone pris de manière isolée ? Le neurone collecte, grâce à ses dendrites, des signaux d'information émanant de son voisinage ou de structures cérébrales plus éloignées. Ces signaux sont intégrés et traités pour fournir une réponse, un influx nerveux, qui est ensuite transmis à d'autres neurones. Ces fonctionnalités sont physiquement associées aux différentes parties du neurone telles que les dendrites, le corps cellulaire et l'axone.

Les dendrites

Un neurone possède un nombre important de ramifications arborescentes qui sont des extensions de quelques dixièmes de microns de diamètre et de quelques dizaines de microns de longueur. La tâche de ces ramifications ou dendrites est de capter les signaux émanant des autres structures neuronales.

Le corps cellulaire

C'est en quelque sorte le cœur de la cellule. D'un diamètre de quelques microns, il est chargé du maintient en vie du neurone tout entier grâce aux échanges biochimiques mis

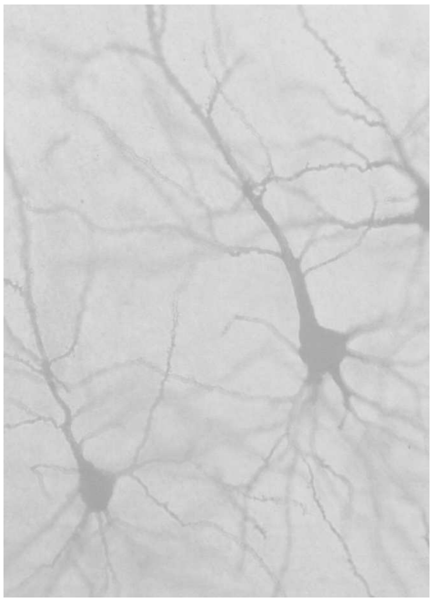


FIG. 2.1 – *Cellules nerveuses biologiques*

en jeu.

L'axone

C'est la partie terminale du neurone. Cette extension tubulaire du neurone est en général plus longue que les dendrites. Cette longueur varie d'un millimètre à quelques centimètres. Après avoir traité l'information captée par les dendrites, le corps cellulaire envoie aux autres structures neuronales une information par l'intermédiaire de cet axone.

Fonctionnement

Génération de l'influx nerveux

Le mécanisme de génération de l'influx nerveux au niveau du corps cellulaire est assez mal connu. En période de repos, la membrane cellulaire établit une différence de potentiel constante de -70mV entre l'intérieur de la cellule et l'extérieur. C'est la rupture de cet équilibre en différents points de la membrane qui caractérise l'influx. Cette rupture est engendrée par l'ouverture et la fermeture de canaux ioniques. Au niveau du corps cellulaire, cinq types de canaux différents sont présents. C'est l'action bien précise de ces canaux qui génère ou non une impulsion à propager sur l'axone.
Il est à noter que l'impulsion générée ne varie pas en amplitude (la différence de potentiel reste constante) mais en fréquence. En outre, un neurone au repos génère toujours un potentiel d'action. L'augmentation de la fréquence de génération de ces potentiels correspond au déclenchement du neurone. Nous verrons que les modélisations ne traitent pas l'activité nerveuse de cette façon.

Propagation de l'influx nerveux

La membrane cellulaire contient différents types de canaux (canaux ioniques à potassium et à sodium) qui maintiennent une différence de potentiel constante d'environ -70mV de part et d'autre de la membrane (fig. 2.2). Elle contient également des pompes à ions dont le but est de refouler les ions sodium à l'extérieur de la cellule et d'attirer les ions potassium à l'intérieur. Au repos, la concentration ionique en sodium est plus importante à l'extérieur qu'à l'intérieur. Lorsqu'un influx est généré par le corps de la cellule, le front de l'influx dépolarise la membrane et conduit à l'ouverture des canaux à sodium. Les ions sodium entrent ainsi à l'intérieur de l'axone pour propager la dépolarisation qui s'établit alors aux alentours de +40mV. Après ouverture des canaux à sodium, les canaux à potassium s'ouvrent automatiquement et les ions potassium sont refoulés à l'extérieur de la cellule rétablissant la différence de potentiel initiale. L'influx, ou potentiel d'action, se propage ainsi de proche en proche [Stevens, 1978].

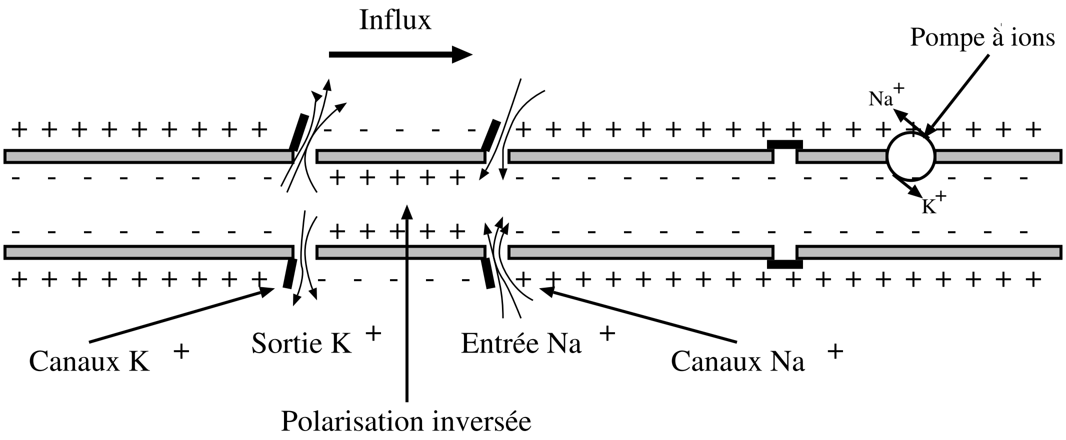


FIG. 2.2 – *Un axone et la propagation d'un influx*

2.2.2 Réseaux de neurones biologiques

Les neurones sont interconnectés entre eux pour former un réseau donnant au cerveau toute sa puissance et sa complexité. Les dendrites ne s'attachent pas directement aux axones pour transmettre les influx. Le point de connexion entre une dendrite et un axone, la synapse, permet en outre des modulations sur les signaux. La propagation de l'influx le long de l'axone est de nature électrique. Au niveau de la synapse, la propagation s'effectue par des relais chimiques (fig. 2.3).

En effet, lorsqu'un influx nerveux aboutit à une terminaison nerveuse, il stimule des vésicules qui vont alors libérer des neurotransmetteurs. Ces neurotransmetteurs vont traverser l'espace situé entre la terminaison axonique et la dendrite. Des récepteurs fixés sur cette dernière sont chargés de recevoir les neurotransmetteurs pour lesquels ils peuvent déclencher un potentiel d'action. Notons que, ces neurotransmetteurs pouvant voyager librement, leur action peut être diffusée. En particulier, certaines drogues arrivées au niveau du

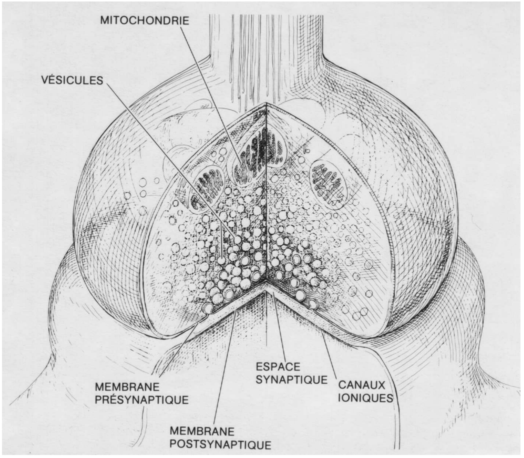


FIG. 2.3 – *Schéma d'une synapse (d'après [Stevens, 1984])*

cerveau peuvent conduire au blocage des synapses entraînant des troubles neurologiques plus ou moins graves [Iversen, 1978]. Il existe différents types de neurotransmetteurs:

- les neurotransmetteurs excitateurs qui, lorsqu'ils sont libérés dans la fente synaptique, génèrent un signal sur le neurone cible (par exemple l'acétylcholine),
- les neurotransmetteurs inhibiteurs qui bloquent la transmission de l'information (par exemple, le GABA).

2.2.3 Circuits locaux, colonnes corticales

Dans [Hubel et Wiesel, 1978], Hubel et Wiesel relatent deux découvertes fondamentales dans l'étude du cerveau. Toutes deux concernent l'organisation corticale. Plus spécifiquement, la première met en évidence la subdivision du cerveau en aires dont chacune d'entre elles possède une fonction spécifique au sein d'un ensemble de structures corticales. La seconde relate le traitement local des opérations effectuées par le cerveau.
Nous ne nous attardons pas sur la cartographie du cerveau dans ce paragraphe, celle-ci sera traitée dans le suivant.
La deuxième découverte est due à Ramon y Cajal et Lorente de No dans les années 30. Ils mettent en évidence des circuits locaux dans tout le cortex, ainsi :

> « *[...] quel que soit le rôle d'une région donnée du cortex, ce rôle est local. Dans toutes les zones du cortex où subsiste une représentation topographique systématique et détaillée il faut faire une analyse « pièce par pièce » [...]* » *[Hubel et Wiesel, 1978].*

Plus précisément

> « *L'information transportée jusqu'au cortex par une fibre unique peut, en principe, se propager à travers toute l'épaisseur des couches après trois ou quatre synapses environ, tandis que la propagation latérale le long des arborisations axonique et dendritique se limite à quelques millimètres, c'est-à-dire à une petite fraction seulement de la vaste étendue corticale.* » *[Hubel et Wiesel, 1978].*

Ceci introduit une organisation fonctionnelle des neurones du cortex en colonnes verticales, ou colonnes corticales. Plusieurs travaux, outre ceux de Ramon y Cajal et Lorente de No, ont été menés sur cette structure du cortex, citons également Szentagothai [Szentagothai, 1975], Hubel et Wiesel [Hubel et Wiesel, 1977; Hubel et Wiesel, 1978], Mountcastle [Mountcastle, 1978] et plus récemment Y. Burnod pour un aspect de modélisation [Burnod, 1988].

Une colonne corticale est un petit tube de 2mm de hauteur (l'épaisseur du cortex) et de 50 à 70 μm de diamètre. Elle constitue donc un petit circuit local d'une centaine de neurones environ et ne dépend pas structurellement de la zone du cortex où elle se trouve. En particulier, l'architecture d'une colonne corticale traitant une information auditive est la même que celle d'une colonne traitant de l'information visuelle qui est la même que celle d'une colonne traitant de l'information motrice, etc. Une colonne corticale hérite naturellement de la structure globale du cortex. En effet, comme celui-ci est organisé en couches parallèles à sa surface, une colonne corticale est également organisée en couches. Ces couches sont au nombre de 6 (figure 2.4).

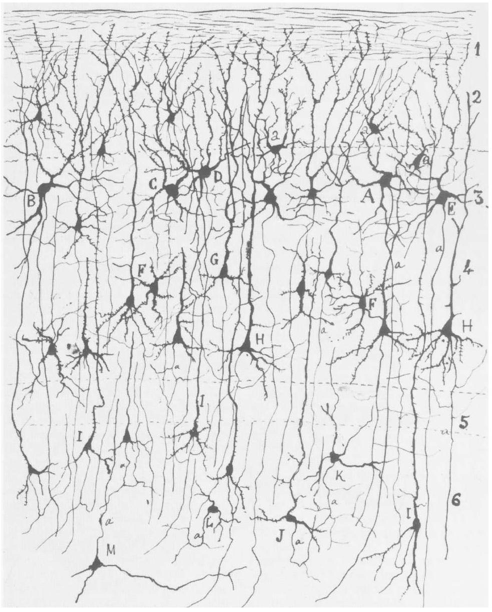


FIG. 2.4 – *Les 6 couches du cortex*

Dans la suite, nous n'emploierons plus le terme de « couche » dans le souci d'éviter toute confusion avec le même terme fréquemment employé par les modélisateurs en connexionnisme et ayant une toute autre signification. Ce terme sera remplacé par celui d'étage.

Si la composition qualitative (types de neurones) de chacun des étages ne varie pas d'une colonne à une autre, la composition quantitative et par là même l'épaisseur de chaque étage peut quant à lui varier. Afin d'éclairer cette variation, détaillons les 6 étages contenus dans une colonne (les étages sont numérotées de I à VI de la surface externe à la surface interne du cortex) :

- Les étages « supra-granulaires » I, II et III reçoivent des afférences d'autres modules corticaux. Les informations peuvent provenir d'un autre hémisphère, d'une autre aire corticale ou d'unités corticales proches. Réciproquement, ces étages envoient des connexions à d'autres modules corticaux.

- L'étage IV reçoit des connexions afférentes en provenance de l'extérieur du cortex, plus précisément d'autres structures corticales comme le thalamus ou encore d'autres aires de niveau inférieur.

- Les étages « infra-granulaires » V et VI envoient des connexions à l'extérieur du cortex. Ces connexions transitent en général par le thalamus.

On trouvera une description précise des différents étages d'une colonne corticale en termes de cellules dans [Burnod, 1988].
Les colonnes impliquées dans le traitement sensoriel possèdent un étage IV prédominant (réception sensorielle) alors que cet étage est très fin pour des colonnes situées dans des aires de haut niveau d'abstraction (cortex frontal par exemple).

2.2.4 Les aires corticales

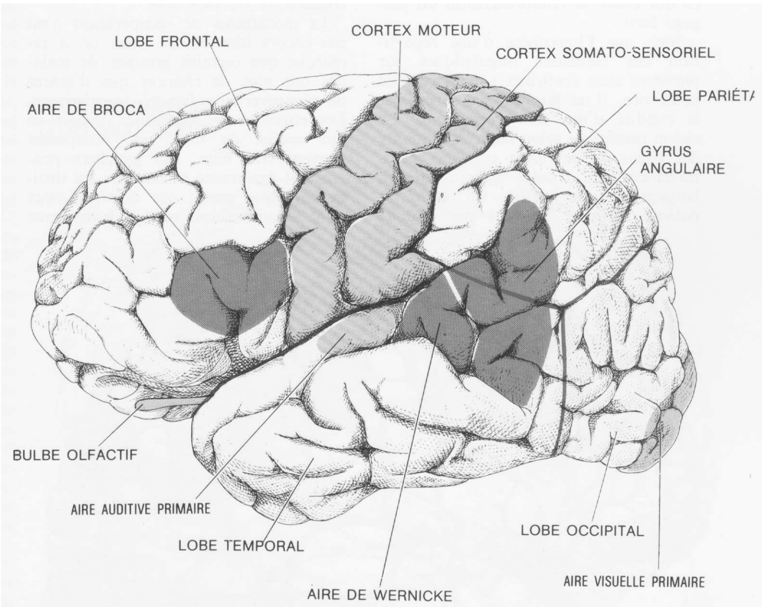


FIG. 2.5 – *Cartographie des aires corticales (d'après [Geshwind, 1984])*

Découpage fonctionnel du cerveau

On sait depuis au moins le début du siècle que le cortex peut se subdiviser en zones ou aires corticales dont chacune possède une fonctionnalité particulière. Plus globalement, la fonctionnalité de certaines structures a pu être mise en évidence, on connaît par exemple l'implication de l'hippocampe dans le stockage des informations à long terme [McNaughton et Morris, 1987]. Par ailleurs, bien que le cortex soit structurellement symétrique, ses deux hémisphères ne réalisent pas les mêmes fonctions. En particulier, la capacité linguistique se localise principalement dans l'hémisphère gauche alors que l'aptitude musicale ou encore la reconnaissance des émotions s'effectuent du coté droit [Geshwind, 1984]. En 1909, Brodmann [Brodmann, 1909] répertorie et numérote une cinquantaine d'aires. Y. Burnod, dans son travail sur la modélisation du cortex, énumère 24 aires corticales fonctionnellement homogènes [Burnod, 1988]. A partir de là, nous pouvons mettre en exergue trois grandes populations de zones corticales : les aires sensorimotrices, les aires associatives et les aires frontales. Nous les présentons brièvement dans la suite sans nommer explicitement les aires mises en jeu. Cet aspect plus technique peut être consulté dans [Burnod, 1988] ou [Alexandre, 1990; Guyot, 1990].

Aires sensorimotrices

Elles correspondent au traitement de l'information directement en relation avec l'extérieur. Ceci englobe le traitement sensoriel et le traitement moteur. Une propriété importante de ces aires est qu'elles reçoivent les informations de façon somatotopique, c'est à dire qu'il y a conservation de la topologie de l'espace de ces informations sur les aires vers lesquelles il se projette. Par exemple, au niveau du cortex somato-sensoriel ou moteur, deux zones proches sur ces aires correspondent à deux parties proches du corps (voir l'homonculus, [Geshwind, 1984]). De la même façon, la rétine se projette sur l'aire primaire du cortex visuel, tout comme la cochlée (organe de l'oreille interne, voir chap. 5) et l'aire primaire du cortex auditif. On parle alors de rétinotopie et de tonotopie. Enfin, les aires sensorimotrices tiennent compte d'informations émanant d'autres structures neuronales relevant des informations corporelles ou hormonales, il s'agit en particulier de l'hypothalamus et de l'hippocampe.
On distingue deux types d'aires dans cette catégorie : les aires primaires et secondaires. Les premières ont le rôle fondamental du codage de l'information. Par exemple, on trouve des détecteurs d'orientation dans le cortex visuel primaire [Hubel et Wiesel, 1977], des détecteurs acoustiques dans le cortex auditif [Buser et Imbert, 1987] ... Ces indices sont ensuite combinés par les aires secondaires qui traitent une information plus complexe que les indices primaires.
Bien qu'elles soient chargées de collecter l'information extérieure, les aires sensorielles peuvent aussi recevoir de l'information du cortex lui même. Ce principe est mis en évidence dans le cortex auditif primaire où certaines fréquences peuvent être privilégiées dans certains cas (milieu bruité, chuchotement,...) [Aitkin, 1990]. Ceux-ci sont déterminés par des aires de niveaux d'abstraction plus élevé pouvant ainsi modifier le codage de la perception.
Les aires sensorimotrices assurent le codage de l'information et l'action sur le monde extérieur. Elles se trouvent donc aux deux extrémités de la chaîne : stimuli-traitement-réponse.

Aires associatives

La représentation de l'information dans le cerveau est une question importante dans le domaine des neurosciences. Même si l'idée d'une représentation distribuée s'oppose encore à l'idée d'une représentation locale [Thorpe, 1989], les aires associatives semblent donner l'avantage à la première. Un objet possède plusieurs dimensions sensorielles (visuelles, auditives, tactiles, émotionnelles, ...) et ces dimensions sont regroupées grâce aux aires associatives. On trouve ainsi dans le cortex des aires reliant les activités musculaires aux activités de vision ou d'audition.
Chaque tâche quotidienne met obligatoirement en jeu un certain nombre de modalités et rares sont les fois où un seule de ces modalités intervient. La tâche de préhension [Alexandre, 1990] par exemple, met en jeu les aires sensorielles visuelles, motrices et somatiques. Lorsqu'il s'agit de toucher un objet de la main, il faut connaître précisément, en termes de coordonnées musculaires, la position que doit avoir la main au niveau du but final. Cette position ne peut être calculée que par corrélation entre position corporelle et coordonnées visuelles. Il faut donc bien, à un moment donné ou à un autre, relier ces deux informations dans le cortex. C'est le rôle des aires associatives.

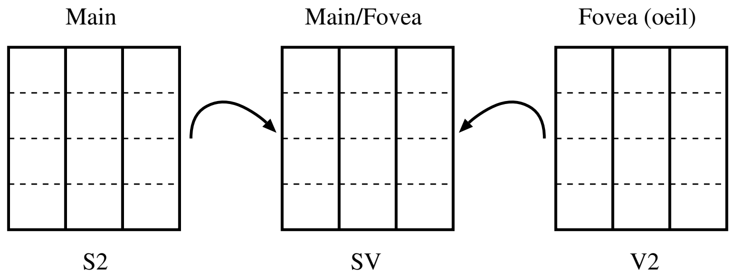


FIG. 2.6 – *Combinaison des deux aires S2 et V2 par l'aire associative SV*

La figure 2.6 montre l'aire associative SV (suivant la nomenclature donnée par Y. Burnod [Burnod, 1988]) qui établit une corrélation entre l'aire somatique secondaire (S2) et l'aire visuelle secondaire (V2). Son rôle est d'effectuer un changement de repère entre la position de la main dans V2 et les coordonnées de cette même main dans S2. De cette façon, les coordonnées d'un objet placé dans le champ visuel sont transformées en «coordonnées somatiques» dans S2 définissant le but à atteindre. Ce but est atteint grâce à la dualité M2-S2, où M2 correspond au cortex moteur.
Dans cet aspect de l'aire associative, on voit parfaitement l'intérêt pratique. A un niveau plus élevé et pour revenir à la représentation de l'information, l'idée d'une représentation des symboles par les cartes associatives semble intéressante. Cette idée suppose que les symboles n'existent pas en tant que tels et ne sont pas représentés par des unités isolées mais plutôt par des associations de flux d'information. Par exemple, le symbole de «table» n'est pas représenté par un seul et unique neurone ou groupe de neurones mais par une association de plusieurs flux d'information correspondant au symbole «table» : l'axe de la vision représente une table, l'axe auditif représente la séquence acoustique correspondant au mot «table», l'axe moteur correspond à la prononciation du mot «table», etc.

Aires frontales

Comme leur nom l'indique, elles se situent sur la partie frontale du cerveau, partie la plus neuve phylogénétiquement et plus développée chez l'homme que chez les autres espèces animales. Les aires frontales sont classiquement associées au traitement de haut niveau et en particulier au raisonnement.
Toutes les aires du cortex ont leur image dans la zone frontale. Ainsi l'aire visuelle secondaire V2 possède une image dans le cortex frontal, l'aire FV2. De plus, la connectivité établie entre les aires sensorimotrices et associatives se retrouve identique entre leurs images dans le cortex frontal. On peut donc penser que cette partie du cerveau ajoute une dimension supplémentaire à l'information traitée par le reste du cortex, il s'agit d'une dimension à caractère temporel. En effet, l'étude des lésions (accidentelles chez l'homme, ou provoquées chez les animaux) montre une implication importante du cortex frontal dans l'organisation temporelle du comportement [Pinto-Hamuy et Linck, 1965; Poppen *et al.*, 1965; Fuster, 1989]. Dans [Levine et Bapi, 1990], D. Levine montre l'implication du cortex frontal dans l'élaboration de séquences variables définissant, suivant un contexte et un problème donné, un plan d'action. Nous reviendrons dans le paragraphe suivant sur l'implication du cortex frontal dans le traitement dynamique de l'information.

2.3 Traitement dynamique de l'information

Peu de certitudes sont établies dans le traitement dynamique de l'information, en particulier pour des niveaux d'abstraction élevés. Il est en effet difficile d'explorer très finement les structures du cerveau en fonctionnement sans avoir recours à des méthodes d'investigation invasives. De nouvelles techniques d'exploration laissent tout de même entrevoir des progrès dans ce domaine. Dans ce qui suit, nous donnons quelques exemples de faits mettant en évidence les aspects dynamiques du traitement de l'information.

2.3.1 Aspects microscopiques

N. Suga et quelques autres [Dear *et al.*, 1993; Suga, 1990] ont mis en évidence les bases neuronales de l'analyse des scènes auditives chez la chauve-souris. Le mécanisme de traitement auditif par effet sonar met en jeu des neurones dont la décroissance d'activité dans le temps n'est pas identique pour tous. Afin de localiser les objets de la scène (en particulier, afin de déterminer sa distance avec ces objets), la chauve-souris émet des sons (signaux bio-sonars) et écoute les échos résultants. Ceux-ci permettent de définir l'image auditive de la scène. En effet, au niveau du cortex auditif, la chauve-souris possède des neurones spécialisés dans des délais d'échos particuliers. Par exemple, un neurone (*delay-tuned neuron*) accordé sur 10ms s'active lorsque le délai entre l'émission du signal et la réception de l'écho vaut 10ms. De plus, parce que les objets de la scène ne sont pas tous à la même distance de la chauve-souris, les échos n'aboutissent pas tous au même instant. Afin de ne pas mélanger les informations dont l'ordre d'arrivée est très important, les courbes d'accord des neurones (fig. 2.7) sont larges pour des délais longs et étroites pour des délais courts où, l'objet étant proche, la réponse est précise.

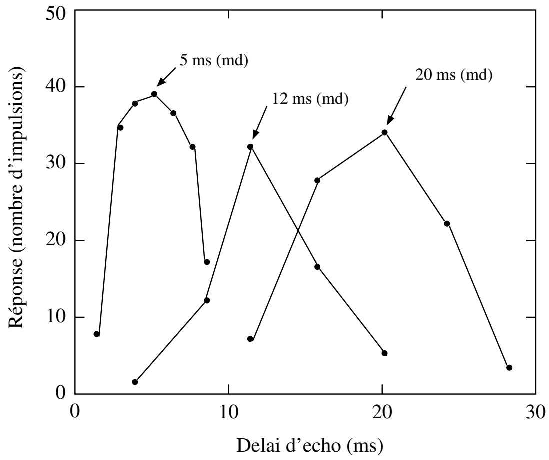


FIG. 2.7 – *Courbes d'accord des « delay-tuned neurons » chez la chauve-souris, md: meilleur délai (d'après [Dear et Al., 1993])*

Les neurones à bande étroite (délai d'écho faible) ont une latence plus longue que les neurones à bande large (délai d'écho long). Ceci permet à tout instant d'avoir une représentation unique de l'image. Un aspect dynamique du traitement cortical de l'information est donc traité au niveau même du fonctionnement du neurone, en particulier par l'intermédiaire de la latence de celui-ci.

Cet exemple montre qu'au niveau même du neurone la dimension temporelle intervient. Nous pouvons remarqué également que les modulations synaptiques peuvent également jouer un rôle important dans le traitement séquentiel de l'information. Dehaene et Changeux [Dehaene *et al.*, 1987] développent l'idée de la triade synaptique où l'activation d'un neurone par un autre via une synapse ne peut se faire qu'à la condition que cette synapse soit rendue excitatrice par un tiers neurone, il s'agit d'une modulation synaptique par ce neurone. Dans ce cas, une influence chimique permet un traitement séquentiel de très bas niveau qui, on peut le penser, n'est vraissemblablement pas sans importance.

Toujours à ce même niveau, la question de savoir si le cerveau utilise un codage temporel ou non reste pour le moment ouverte. Classiquement, on décide qu'un neurone est actif ou pas en fonction du taux de décharge du neurone. Ce taux correspond au nombre d'impulsions ou spikes émis dans un temps donné. La fréquence de décharge n'est donc pas prise en compte dans l'idée que nous avons du codage de l'information. Cependant, certains auteurs pensent que la fréquence de décharge peut intervenir pour notamment corréler des événements par synchronisation des fréquences de décharge des neurones [Milanese, 1994].

2.3.2 Aspects macroscopiques

Nous l'avons vu plus haut, le cortex frontal est peut-être la structure corticale la plus impliquée dans le traitement dynamique de l'information. Cependant, cette implication s'avère se situer à un niveau d'abstraction élevé et il est quasiment certain que les autres zones du cortex participent, à un niveau moindre, à l'élaboration du traitement séquentiel. Dans [Levine et Bapi, 1990], D Levine distingue notamment les séquences flexibles des séquences fixes. Les premières sont traitées et élaborées par le cortex frontal et définissent des plans d'actions établis en fonction d'un problème à résoudre. La constuction de ces séquences est dynamique, c'est à dire qu'elle peut être remise en cause à tout moment. En revanche, les séquences fixes sont vraisemblablement codées par d'autres aires que les aires frontales. Cela est intuitivement correct si l'on prend pour exemple le traitement de la parole. La reconnaissance de sonorités s'effectue dans les aires temporales du cortex et en particulier dans l'aire de Wernicke [Geshwind, 1984]. Or cette reconnaissance s'établit sur la base d'une séquences d'informations acoustiques. On peut donc penser que les aires de traitement de la parole réagissent sous l'impulsion de séquences.

Relatons deux expériences qui mettent en évidence le traitement dynamique par le cortex frontal.
Pinto-Hamuy et Linck [Pinto-Hamuy et Linck, 1965] ont montré l'effet de lésions au niveau du cortex frontal sur les performances de tâches séquentielles. Levine et Bapi [Levine et Bapi, 1990] développent une expérience consistant à montrer à des macaques des indices visuels (par exemple des boutons qui s'allument sur un panneau), les singes doivent alors répondre (appuyer sur les boutons allumés) en fonction des indices visuels. La première tâche consiste à allumer plusieurs boutons de façon séquentielle, le singe devant les éteindre dans le même ordre (séquences fixes). La seconde tâche consiste à allumer quelques boutons en même temps, le singe peut alors les éteindre dans l'ordre qu'il souhaite (séquences variables ou flexibles). Ces tâches sont correctement effectuées avec des singes témoins. En revanche, la tâche concernant les séquences flexibles n'est plus correctement effectuée dans le cas ou le cortex frontal du singe a été lésé alors que la tâche des séquences invariantes reste correcte. Cette expérience met donc en évidence le rôle du cortex frontal dans l'établissement des plans d'actions. De plus, plusieurs études neurophysiologiques ont mis en évidence le rôle de ce cortex dans l'intégration temporelle d'événements moteurs ou sensoriels et concluent que certains neurones codent la signification d'un stimulus sans se préoccuper de ses caractéristiques physiques (visuelles, auditives, tactiles, etc). En outre, G. Dehaene-Lambertz [Dehaene-Lambertz et Dehaene, 1994] met en évidence une activité au niveau frontal lors de la prononciation d'une syllabe nouvelle dans une série de syllabes connues chez le nourrisson [Dehaene-Lambertz et Dehaene, 1994]. La réponse tardive détecte en quelque sorte la nouveauté afin de focaliser l'attention sur celle-ci. Cette réponse est « supra-modale », c'est à dire qu'elle ne dépend pas de la nature de l'information transmise (auditive, visuelle, tactile, etc). Ce mécanisme permettrait, en outre, de pister la nouveauté et de basculer à la manière d'un central sur un autre contexte de traitement.

Enfin, pour terminer sur cet aspect dynamique et mettre en évidence l'implication d'aires coopérantes hors du cortex frontal, nous prenons l'exemple de la production de parole à partir de stimulations visuelles ou auditives. Les figures 2.8 et 2.9 montrent l'évolution du message neuronal pour la production d'un mot respectivement entendu et écrit. Le circuit d'information concernant le langage met en jeu deux aires principales : l'aire de Broca et l'aire de Wernicke. A la lumière de déficiences constatées sur des patients

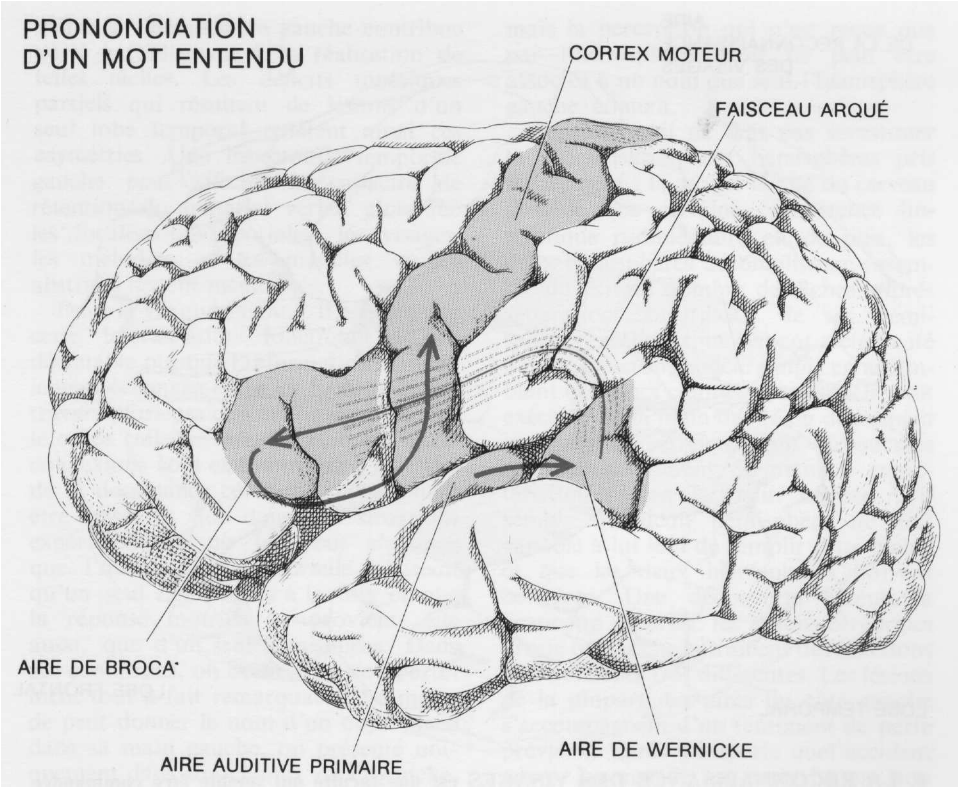


FIG. 2.8 – *Mise en jeu de différentes aires corticales lors de la prononciation d'un mot entendu (d'après [Geshwind, 1984])*

suite à des lésions, le rôle de chacune de ces aires a pu être mis en évidence. L'aire de Broca située dans la zone frontale du cortex joue un rôle d'organisation des phrases. Une lésion au niveau de cette aire entraine une difficulté pour formuler des phrases grammaticalement correctes, le style devient télégraphique. L'aire de Wernicke semble, elle, plus impliquée dans la compréhension des mots. Une lésion dans cette aire entraine la prononciation de phrases correctes grammaticalement mais dépourvue de sens. Les deux aires sont reliées entre elles par un ensemble de fibres, le faisceau arqué. Pour prononcer un mot entendu ou écrit, la phase de traitement commence bien évidemment par l'activation au niveau des aires sensorielles visuelles ou auditives. Ensuite, l'information passe par l'aire de Wernicke pour la compréhension du message verbal. L'information est ensuite envoyée dans l'aire de Broca via le faisceau arqué. A ce niveau, il y a formulation de phrases grammaticalement correctes. L'aspect temporel joue un rôle important ici, d'ailleurs l'aire de Broca se situe dans le cortex frontal, zone classiquement associée à l'organisation des événements. En fin de parcours, l'aire de Broca a généré une phrase donc un plan moteur pour produire physiquement cette phrase. Elle envoie ce plan au cortex moteur qui l'exécute.

2.3.3 Vers la modélisation

A la lumière de ce qui précède, on remarque que le mécanisme de génération et de transmission de l'influx nerveux est bien connu et qu'à un niveau plus abstrait, nous pou-

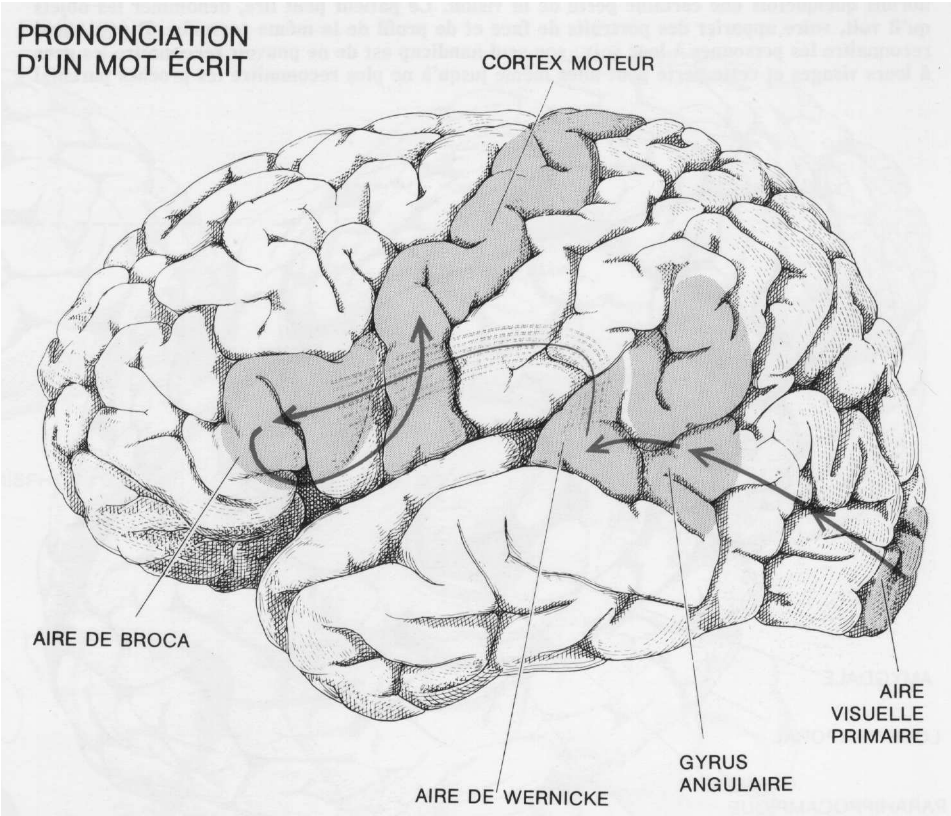


FIG. 2.9 – *Mise en jeu de différentes aires corticales lors de la prononciation d'un mot écrit (d'après [Geshwind, 1984])*

vons mettre en évidence des éléments de traitement de l'information (colonnes corticales) ainsi qu'un découpage fonctionnel du cerveau. Cependant, à un niveau encore plus haut, le fonctionnement cérébral est mal connu. Du point de vue de la modélisation, le neurone est relativement facilement réalisable et la colonne corticale bien que plus complexe peut s'envisager en termes de représentation formelle. Mais le cerveau ou les aires corticales peuvent être, en l'état actuel de nos connaissances, difficilement modélisables. La difficulté va donc croissant. Cependant, nous pouvons utiliser un certain nombre d'idées et d'hypothèses pour la modélisation qui ne tient alors forcément pas compte de tous les aspects du vivant. Pourtant, ces modélisations peuvent aider les neurobiologistes à valider et à aller plus loin dans leurs hypothèses. La démarche s'insère pleinement dans le cadre des neurosciences.

Nous présentons, dans la suite, les fondements du connexionnisme. Tous les modèles ne sont pas interprétables neurobiologiquement mais peuvent intéresser de près ou de loin nos travaux dans une modélisation d'inspiration biologique.

2.4 Modélisation du neurone

2.4.1 Modèle initial

C'est dans les années 40 que fût décrite la première modélisation du neurone. Plus précisément, en 1943 W. Mac Culloch et W. Pitts présentèrent dans [MacCulloch et Pitts, 1943] une formalisation du neurone conduisant à la description d'un automate à seuil.
Cet automate effectue une somme pondérée de ses entrées (chaque entrée est une valeur numérique qui correspond à l'activation d'un autre automate) et déclenche une réponse si cette somme dépasse un certain seuil. La figure 2.10 résume la chaîne de traitement développée par l'automate.

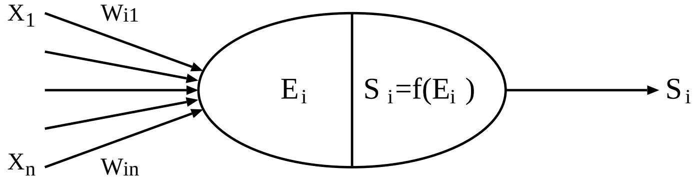


FIG. 2.10 – *Neurone formel*

La réponse finale est calculée selon la formule suivante:

$$S_i = f(\sum_{j=1}^{n} W_{ij}X_j) \tag{2.1}$$

Les X_j sont les potentiels d'action binaires émanant des autres neurones, les poids W_{ij} représentent l'efficacité synaptique entre les deux neurones i et j et S_i est la sortie calculée du neurone i. La fonction de transfert f permet un seuillage de la fonction d'entrée totale E_i $(= \sum_{j=1}^{n} W_{ij}X_j)$ et possède l'allure donnée sur la figure 2.11 où β représente le seuil (ou biais) de déclenchement du neurone i.

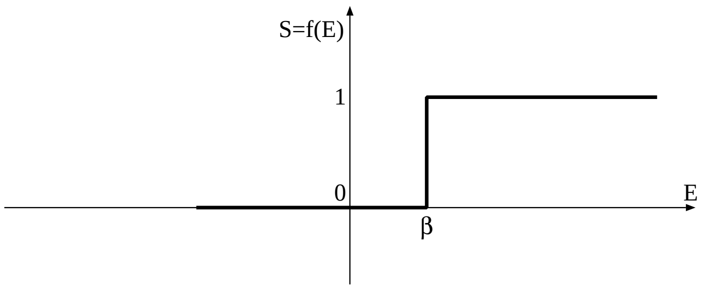


FIG. 2.11 – *Fonction de seuil du neurone formel de Mac Culloch et Pitts*

2.4.2 Extension

Le modèle originel décrit un automate à seuil booléen pour lequel les entrées et la sortie sont binaires. Nous pouvons alors étendre le neurone formel en modifiant les types des entrées et sorties, la fonction d'entrée totale (E_i) ou encore la fonction de seuil (f). Nous verrons dans la suite que les extensions de ce modèle se sont révélées nécessaires pour le dépassement de certaines limites du neurone formel.

Classiquement des entrées réelles sont utilisées et un biais β caractérisant le seuil de déclenchement du neurone est introduit. La fonction d'entrée totale devient alors une fonction affine du type :

$$E_i = \sum_{j=1}^{n} W_{ij} X_j - \beta \qquad (2.2)$$

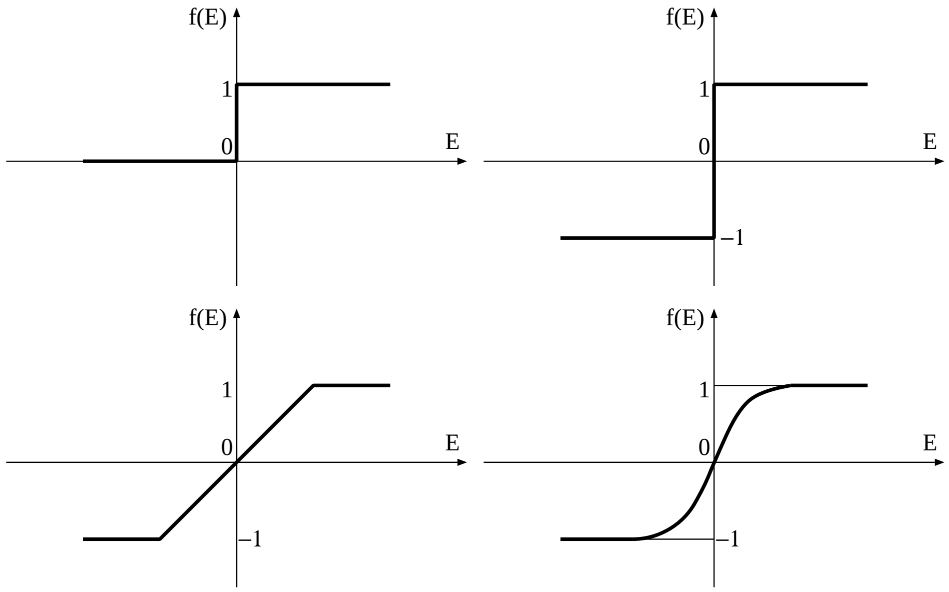


FIG. 2.12 – *Différentes fonctions de transfert*

La fonction de transfert peut prendre diverses formes comme décrites sur la figure 2.12:

- fonction de Heaviside (en haut à gauche) ;
- fonction signe dans le cas de sorties binaires (en haut à droite) ;
- fonction linéaire à seuil (en bas à gauche) ;
- fonction sigmoïde dans le cas de sorties réelles : $f(x) = a.\frac{e^{kx}-1}{e^{kx}+1}$ (en bas à droite) ;
- fonction stochastique: $f(x) = 1$ avec la probabilité $1/(1 + exp(\frac{-x}{T}))$, $f(x) = 0$ sinon, où T est un paramètre appelé température ;
- fonction gaussienne, ...

Pris de façon isolée, un neurone formel effectue une séparation linéaire de ses entrées (sauf dans le cas d'une fonction de seuil Gaussienne). Si l'espace des entrées est linéairement séparable en deux classes, il est possible de trouver les bons poids synaptiques du neurone pour définir l'hyperplan qui sépare l'espace en deux parties.
La limite de transition du neurone (sa frontière entre état actif et état inactif) est donnée par l'hyperplan d'équation $\sum_{i=1}^{n} W_{ij} X_j + \beta = 0$. La figure 2.13 montre un neurone binaire effectuant l'opération logique ET.

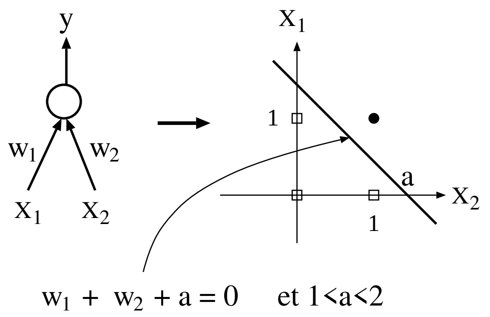


FIG. 2.13 – *Neurone effectuant un ET logique par séparation linéaire de l'espace*

2.5 Perceptrons mono et multi-couches

Comme nous venons de le voir, le neurone isolé (biologique ou formel) n'a pas un comportement que l'on pourrait qualifier d'intelligent au sens courant du terme. Le neurone exécute une fonction mathématique simple (séparation linéaire) dont l'utilisation pratique se trouve très limitée. En revanche, l'assemblage de telles entités permet l'apprentissage de fonctions plus complexes.
Les perceptrons ont été extensivement étudiés et souvent décrits. Nous ne présentons donc ici que les grandes lignes de ces modèles, ceux-ci n'étant pas directement en rapport avec nos travaux. Pour une description détaillée, le lecteur pourra se reporter à [Rumelhart et McClelland, 1986].

2.5.1 Le perceptron originel

En 1958, Rosenblatt [Rosenblatt, 1958] décrit le premier perceptron. Les propriétés du modèle (bonne spécification, comportement correct grâce à une architecture complexe, accord avec les faits biologiques ...) ainsi que ses capacités vont rapidement le rendre célèbre. L'architecture est en fait une rangée de neurones de Mac Culloch et Pitts qui apprennent chacun une fonction de transfert. Nous trouvons également une rétine composée de neurones ou cellules dont le rôle est de transcrire un stimulus, et une rangée de cellules d'association qui effectuent un pré-traitement sur la rétine (figure 2.14). Les neurones de sortie effectuent une somme pondérée de leurs entrées et calculent ensuite leur activité en passant cette somme par une fonction de Heaviside ou une fonction signe.

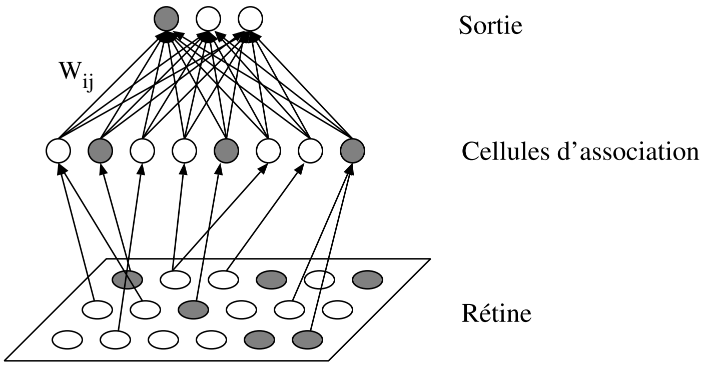


FIG. 2.14 – *Perceptron mono-couche*

La modification des poids est supervisée et s'effectue grâce à la règle de Hebb [Hebb, 1949] établie comme produit entre l'erreur calculée au niveau du neurone de sortie (différence entre sortie calculée et sortie désirée) et la valeur d'entrée pour le poids considéré. Le théorème de convergence du perceptron montre que s'il existe un jeu de poids capable de simuler la transformation alors la règle d'apprentissage convergera vers une solution en un nombre fini de pas.

2.5.2 Perceptrons multi-couches

Le perceptron mono-couche n'est pas capable de faire plus qu'une séparation linéaire de ses entrées. Afin d'obtenir des classes de formes convexes, plusieurs couches sont nécessaires (fig. 2.15).

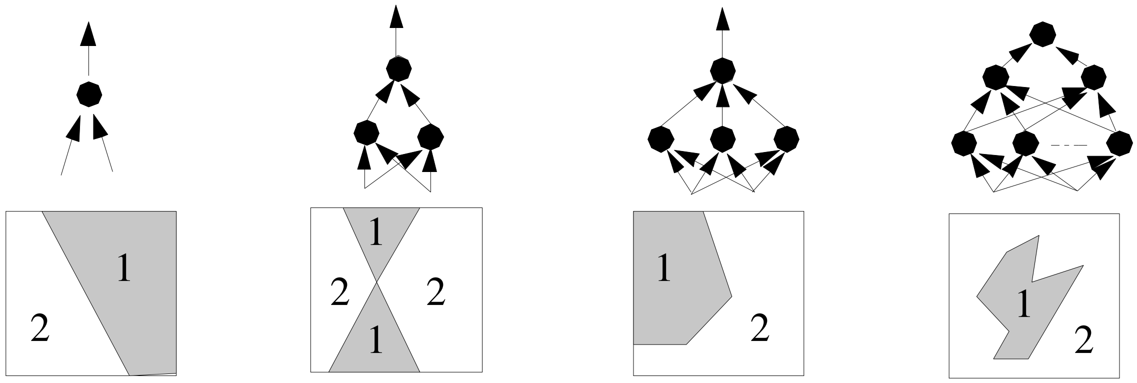


FIG. 2.15 – *Perceptrons et séparation de l'espace*

La figure 2.16 représente un perceptron à une seule couche cachée, la couche d'entrée étant composée de neurones codant le vecteur d'entrée du réseau. Suivant l'architecture (nombre de neurones, nombre de couches cachées), le réseau est capable de partitionner en classes son espace d'entrée (figure 2.15). Cependant, la procédure d'apprentissage des poids définie pour le perceptron mono-couche n'est plus valide ici puisque la sortie désirée n'est pas connue au niveau d'une couche cachée. Le problème du calcul des poids dans les réseaux multi-couches est un cas particulier du *credit assignment problem* que l'on retrouve dans les systèmes à apprentissage naturels ou artificiels. On trouvera une description de la rétro-propagation, l'algorithme de modification des poids permettant de résoudre ce problème dans un réseau multi-couches, dans [Rumelhart *et al.*, 1986].

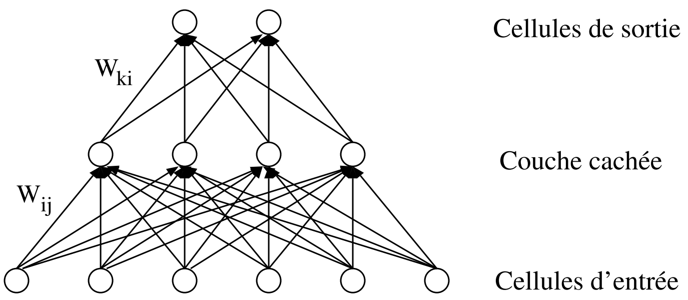


FIG. 2.16 – *Perceptron multi-couches*

2.6 Réseaux auto-organisés

Apprentissage supervisé, apprentissage par renforcement et apprentissage non supervisé sont les trois modes d'apprentissage en connexionnisme. Le premier a été décrit plus haut et utilise une mesure d'erreur pour adapter les poids du réseau. Le second est également supervisé dans le sens où un «professeur» indique si la sortie obtenue est correcte ou non sans se préoccuper de la valeur quantitative de l'erreur, les poids sont alors modifiés sélectivement. Enfin, le troisième permet d'extraire automatiquement des régularités à l'intérieur d'un corpus d'entrée.
Ces trois types d'apprentissage correspondent globalement assez bien aux modes d'apprentissage intervenant chez l'être humain. En effet, régulièrement dans notre vie nous apprenons des informations qui nous sont transmises par un support (textes, images, humains ...) mais certaines de ces informations peuvent être découvertes ou inférées. Par exemple, dans le cas de la reconnaissance de la parole, une signification est fournie à chaque mot par un maître (apprentissage supervisé) mais le codage des unités phonétiques de base semble être appris de manière non supervisée et de façon automatique [Dehaene-Lambertz et Dehaene, 1994] (apprentissage non supervisé).
Bien que les réseaux supervisés et en particulier les perceptrons soient les plus utilisés en pratique, la classe des réseaux auto-organisés ou non supervisés est souvent utilisée pour l'apprentissage des régularités ou traits contenus dans un ensemble de données. Ces réseaux déterminent les poids des connexions pour qu'une sortie particulière regroupe les entrées ayant les mêmes régularités.
Il serait trop fastidieux d'entreprendre ici une énumération complète des modèles auto-organisés, c'est pourquoi après avoir donné quelques généralités concernant ces modèles, nous présentons le modèle de Kohonen qui est un des modèles les plus utilisés et qui connaît un grand succès. Enfin, nous présentons d'autres architectures qui nous serons utiles pour la définition de notre modèle (chapitre 4).

2.6.1 Compétition et inhibition

Le principe de compétition entre les unités est la base du fonctionnement des modèles auto-organisés. Ce principe permet le traitement des problèmes où seule l'information en entrée est disponible. Ainsi un réseau auto-organisé permet :

- d'extraire des invariants du signal d'entrée, ceci permet d'isoler les traits d'une forme en entrée pour pouvoir les utiliser dans un traitement ultérieur ;
- de faire de la classification après étiquetage, les unités de sortie définissent des classes construites à partir des données du corpus d'apprentissage ;
- de faire de la réduction de dimension, le vecteur d'entrée est de dimension n et nous disposons de m neurones en sortie avec $n > m$, l'espace d'entrée se projette sur les neurones de sortie.

Dans un réseau de neurones compétitif, un seul neurone est actif à chaque pas de temps. Ceci est rendu possible grâce aux connexions inhibitrices entre les neurones. Ainsi, lorsqu'un neurone est rendu actif, les liens inhibiteurs empêchent les autres de le devenir (fig 2.17).

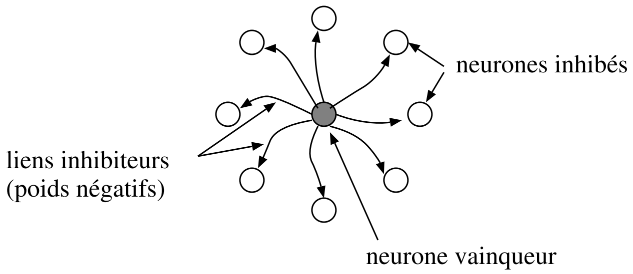


FIG. 2.17 – *Neurones et liens inhibiteurs*

Ce mécanisme connu sous le nom anglophone de *winner takes all* peut être mis en œuvre de différentes façons. La plus courante consiste à déterminer le neurone vainqueur par le calcul de son activité ou de sa distance par rapport au vecteur d'entrée :

- dans le cas de l'activité, chaque neurone du réseau calcule son activité selon la somme pondérée de ses entrées :

$$O_i = \sum_j W_{ij}.x_j = \vec{W}_i.\vec{x} \tag{2.3}$$

  alors le neurone vainqueur d'indice k est donné par :

$$k : \vec{W}_k.\vec{x} \geq \vec{W}_j.\vec{x}, \forall j \tag{2.4}$$

- dans le cas de la distance, chaque neurone calcule sa distance euclidienne avec le vecteur d'entrée. Le neurone vainqueur d'indice k est alors celui pour lequel la distance est la plus petite :

$$k : \|\vec{W}_k - \vec{x}\| \leq \|\vec{W}_j - \vec{x}\|, \forall j \tag{2.5}$$

Le neurone vainqueur est alors ajusté pour se rapprocher du vecteur d'entrée tout en restant proche des vecteurs qu'il représentait précédemment. Le placement se fait en choisissant le centre de gravité :

$$(n_k + 1)(\vec{W}_k + \Delta\vec{W}_k) = n_k\vec{W}_k + \vec{x} \tag{2.6}$$

d'où

$$\Delta\vec{W}_k = \frac{1}{n_k + 1}(\vec{x} - \vec{W}_k) \tag{2.7}$$

où n_k représente le nombre de vecteurs dans la classe représentée par $\vec{W}_k$ (figure 2.18).

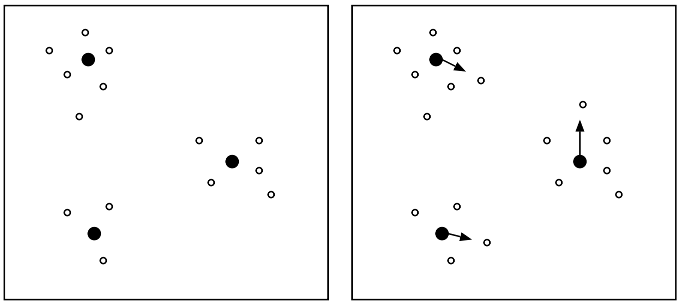


FIG. 2.18 – *Classes et centres de gravité*

Ce principe peut être utilisé pour la construction de réseaux à apprentissage concurrentiel. Ces réseaux sont organisés en couches et chacune de ces couches contient des paquets de neurones en compétition. Le nombre de paquets dans une couche détermine le nombre de caractéristiques à extraire sur la couche inférieure. La figure 2.19 représente un réseau concurrentiel avec une couche cachée qui extrait deux caractéristiques et une couche de sortie avec une caractéristique finale faisant en quelque sorte la synthèse des informations extraites à l'intérieur du réseau.

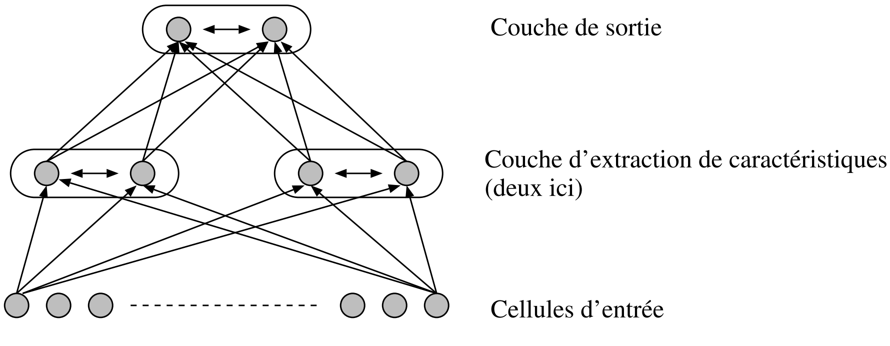


FIG. 2.19 – *Réseau à apprentissage concurrentiel*

D'autres réseaux concurrentiels ont été développés. Signalons en particulier le néocognitron de Fukushima [Fukushima, 1980], réseau organisé en couches qui extraient successivement des indices de plus en plus abstraits dans la hiérarchie du réseau. Ce modèle a été développé pour la reconnaissance de formes. Une propriété intéressante du néocognitron est sa tolérance en translation et rotation des divers indices graphiques découverts. Un réseau compétitif est souvent assimilé à des méthodes statistiques classiques comme l'analyse en composantes principales [Oja, 1982]. Le modèle de Rubner et Schulten [Rubner et Schulten, 1990] (fig. 2.20) extrait des indices à la manière d'une analyse en composantes principales grâce à des liens inhibiteurs disposés en cascade entre les neurones de sortie du réseau. Ce réseau a été appliqué à l'extraction d'indices acoustiques dans [Beaugé *et al.*, 1993].

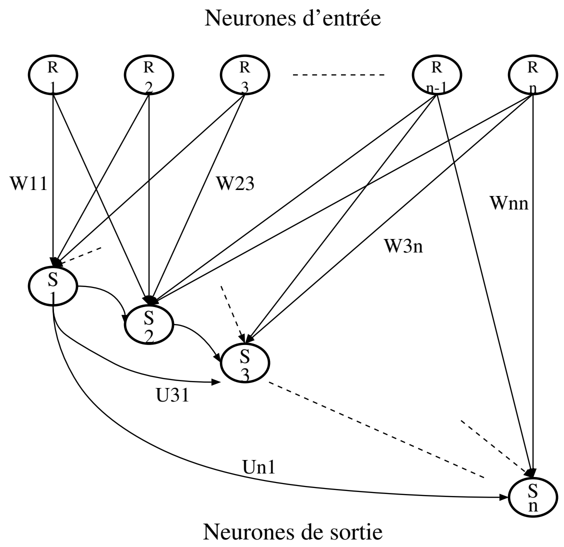


FIG. 2.20 – *Le modèle compétitif de Rubner et Schulten*

Plus formellement, on trouvera dans [Bourret *et al.*, 1991] des méthodes de simulation du comportement *winner takes all* par des liens inhibiteurs.

Dans la règle de modification de poids que nous venons de voir pour les réseaux compétitifs (eq. 2.7), seul le neurone vainqueur est modifié et il n'y a pas de relation de voisinage. Kohonen a proposé une technique pour l'apprentissage auto-organisé conservant les relations de voisinage entre l'entrée et la sortie. D'autres modèles ont également été développés pour la prise en compte de la topologie des espaces d'entrée, c'est le cas par exemple du modèle de T. Martinetz, neural-gas [Martinetz et Schulten, 1991]. Nous présentons ces modèles dans la suite.

2.6.2 Cartes auto-organisatrices de Kohonen

Les cartes auto-organisatrices de Kohonen [Kohonen, 1989] incluent la notion de topologie qui n'est pas présente dans les modèles précédemment cités. Ces cartes s'inspirent des faits neurophysiologiques selon lesquels les aires sensorielles du cortex contiennent des détecteurs dont la topologie est en correspondance avec la topologie de l'espace des stimuli. Ceci veut dire que deux stimuli proches au sens de la topologie de l'espace d'entrée activeront deux neurones proches au sens de la topologie des détecteurs c'est à dire de la carte. Plusieurs expériences ont été menées dans l'étude des aires sensorielles, principalement chez les animaux. Par exemple, des détecteurs de traits visuels ont été mis en évidence dans l'aire primaire du cortex visuel [Hubel et Wiesel, 1977] où l'on constate notamment que des zones très proches dans cette aire correspondent à des orientations dont la variation d'angle est faible.

Les cartes de Kohonen utilisent un neurone proche du neurone formel de Mac Culloch et Pitts. Soit $\mathcal{V}$ le potentiel de membrane du neurone. Ce potentiel est défini par

$$\mathcal{V} = \sum_i W_i.e_i \tag{2.8}$$

et correspond donc à la somme pondérée des entrées e_i. Soit s l'activation de ce même neurone. Cette activation dépend du potentiel d'action ainsi que d'une fonction sigmoïde (σ) permettant de modéliser la saturation du neurone qui possède un temps de réaction limité après avoir été activé une première fois. Ainsi, l'équation d'état dynamique du neurone est

$$\frac{ds}{dt} = \mathcal{V} - \sigma^{-1}(s) \tag{2.9}$$

En équilibre, $\frac{ds}{dt} = 0$ d'où

$$\mathcal{V} = \sigma^{-1}(s) \tag{2.10}$$

et la loi de fonctionnement du neurone est alors donnée par

$$s = \sigma(\sum_i W_i.e_i) \tag{2.11}$$

Nous retrouvons ainsi l'équation d'activation du neurone de Mac Culloch et Pitts.
La procédure d'apprentissage associée aux cartes de Kohonen garantit que chaque neurone du réseau s'active pour un stimulus particulier et que deux neurones proches sur la carte correspondent à deux stimuli proches en entrée. Pour cela, lors de l'adaptation des poids plusieurs neurones voient leurs poids modifiés grâce aux relations de voisinage qui existent sur la carte. La règle d'apprentissage est alors définie de la façon suivante :

$$W_{ij}(t+1) = W_{ij}(t) + \gamma h(i,k)(x_j - W_{ij}(t)) \tag{2.12}$$

où k est l'indice du neurone vainqueur et i l'indice du neurone que l'on adapte. La fonction $h(i,k)$ détermine la relation de voisinage entre le neurone gagnant et les autres. Généralement, elle représente des interactions latérales modélisées par une fonction en forme de « chapeau mexicain » (fig. 2.21).

La convergence de l'algorithme peut être améliorée en rendant cette fonction h dépendante du temps. Au cours du temps, le voisinage est diminué pour se réduire, à la fin de l'apprentissage, au neurone vainqueur.
Le choix du neurone gagnant lors de la présentation d'un stimulus $\vec{x}$ est conditionné par la position du vecteur des poids $\vec{W_i}$ du neurone i par rapport à $\vec{x}$. Soit k l'indice du neurone vainqueur, nous avons

$$\|\vec{x} - \vec{W_k}\| \leq \|\vec{x} - \vec{W_j}\|, \forall j = 1, ..., n \tag{2.13}$$

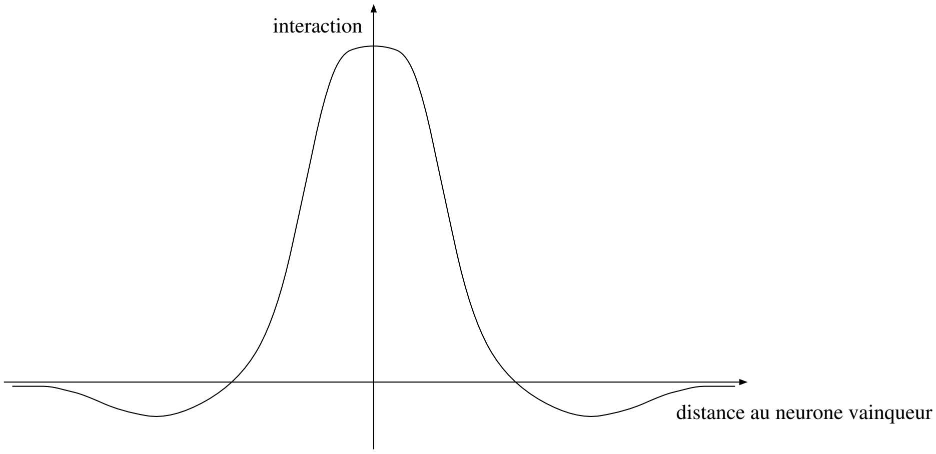


FIG. 2.21 – *Fonction de voisinage en forme de chapeau mexicain*

Le modèle de Kohonen est souvent utilisé pour diverses tâches : compression de données, mémoires auto et hétéro-associatives et a donc l'avantage d'extraire des caractéristiques avec relations topologiques. Il est en outre appliqué au domaine de la vision (segmentation d'images) ou de la parole (cartes phonétiques [Kohonen *et al.*, 1984]). Dans ces cartes, les relations de voisinage sont définies *a priori* dans le réseau. Ainsi, nous pouvons construire des modèles de Kohonen à une seule dimension (un neurone a deux voisins) ou des modèles à deux dimensions (un neurone a quatre voisins). Certains modèles n'utilisent pas de connectivité latérale *a priori* mais apprennent cette connectivité. C'est le cas pour les modèles que nous présentons dans le paragraphe 2.6.4.

2.6.3 Associations par les cartes auto-organisatrices : le modèle LASSO

Kohonen propose une version supervisée des cartes auto-organisatrices, l'algorithme LVQ (Learning Vector Quantization) [Kohonen, 1989]. Dans cette version, l'algorithme de modification des poids est guidé par un « professeur » qui choisit le bon neurone à activer, c'est-à-dire la bonne classe de vecteurs. Deux cas peuvent alors se produire :

- Un neurone s'active et ce choix est correct, les poids se rapprochent alors du vecteur ;
- Un neurone s'active et ce choix est incorrect, les poids s'éloignent alors du vecteur.

Ceci permet d'obtenir des classes de vecteurs bien séparées entre elles.

Dans [Midenet et Grumbach, 1990], S. Midenet propose un modèle d'association à partir d'une carte de Kohonen. Le principe consiste à utiliser une carte de Kohonen afin d'apprendre un espace vectoriel réunion de deux sous-espaces. Le premier sous-espace correspond aux données à présenter en entrée, le second correspond à l'information que l'on souhaite associer aux données. Par exemple, dans une utilisation pour la reconnaissance de caractères manuscrits, le premier espace correspond à une image binaire des chiffres et le second indique quel chiffre est présenté au réseau. L'apprentissage est alors en quelque sorte supervisé.

Du point de vue du fonctionnement en phase d'apprentissage, la carte utilise un algorithme classique de Kohonen. Celui-ci travaille sur des vecteurs de dimension N^C correspondant à la fusion des vecteurs de données (de dimension N^E) avec les vecteurs de sortie (de dimension N^S) ($N^C = N^E + N^S$). L'algorithme d'apprentissage pour l'adaptation des poids W^{EK} et W^{SK} est classiquement celui utilisé dans les cartes auto-adaptatives de Kohonen avec un paramètre d'apprentissage et une largeur de fenêtre se réduisant au cours du temps. L'architecture du modèle LASSO (Learning ASsociations by Self-Organization) est donnée sur la figure 2.22.

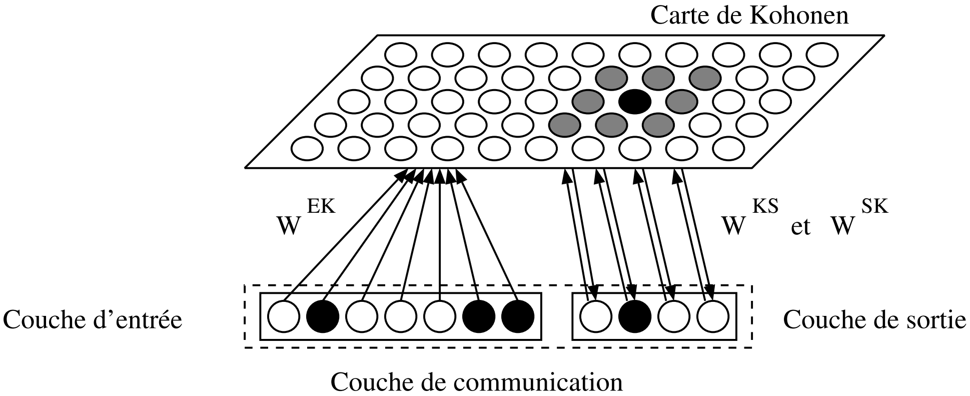


FIG. 2.22 – *Architecture du modèle LASSO*

Après apprentissage les poids W^{SK} sont inversés, d'où $W^{KS} = W^{SK}$. En phase d'exploitation, le calcul du neurone vainqueur sur la carte de Kohonen s'effectue sur le sous-espace de l'entrée de dimension N^E. Les N^S dimensions supplémentaires ne sont alors pas utilisées pour le calcul. Les neurones voisins du neurone vainqueur sur la carte de Kohonen sont activés si leur distance euclidienne avec le neurone vainqueur est inférieure à un certain seuil. La valeur d'activation des neurones de sortie est donnée par :

$$X_i^S = \frac{1}{\sum_k Y_k} \sum_{j-actifs} W_{ji}^{KS} Y_j \tag{2.14}$$

Ce principe de fonctionnement consiste à fournir une partie du vecteur d'information, le modèle LASSO se chargeant de fournir la partie manquante. On se rapproche ici des mémoires auto-associatives.

2.6.4 Le modèle neural-gas et ses extensions

neural-gas

Le modèle que nous présentons maintenant est fondamentalement assez proche du modèle de Kohonen. Le choix du neurone actif (ou vainqueur) ainsi que la loi d'adaptation des poids sont similaires à ceux proposés dans les algorithmes utilisés pour les cartes de Kohonen. La différence entre les deux méthodes réside dans la définition des liens de voisinage. Pour les cartes de Kohonen, on définit *a priori* la topologie du réseau, c'est à dire les liens entre les différents neurones. Dans le réseau neural-gas [Martinetz et Schulten, 1991], les liens entre les neurones ne sont pas fixés *a priori* mais appris. Ainsi, pendant la phase d'apprentissage, les neurones non seulement se positionnent dans l'espace des entrées mais établissent également des connexions définissant leur voisinage. De cette manière, après la phase d'apprentissage, le réseau établit une projection de l'espace d'entrée sur un espace de dimension plus réduite (dimension des vecteurs de poids) tout en conservant la topologie lors du passage d'un espace à l'autre (à deux points proches dans l'espace d'entrée correspondent deux points proches dans l'espace du réseau).

Ci-dessous, nous donnons l'algorithme d'apprentissage d'un réseau neural-gas. Les notations sont classiques pour ce qui concerne les poids ($\vec{W}$) et les vecteurs d'entrée($\vec{x}$), cependant trois variables sont introduites pour définir l'apprentissage des connexions : *(i)* un tableau de dimension 2 des connexions où la case à l'intersection de la ligne i et de la colonne j (C_{ij}) indique s'il existe une connexion du neurone i vers le neurone j (valuation à 0 ou 1), *(ii)* un tableau de dimension 2 des âges des connexions où la case à l'intersection de la ligne i et de la colonne j indique l'âge (t_{ij}) de la connexion C_{ij}, *(iii)* une variable k_i indiquant le classement du neurone i pour son activité (par exemple, si le neurone 4 est vainqueur et le neurone 2 à une activité très proche de 4 nous avons, $k_4 = 1$ et $k_2 = 2$).

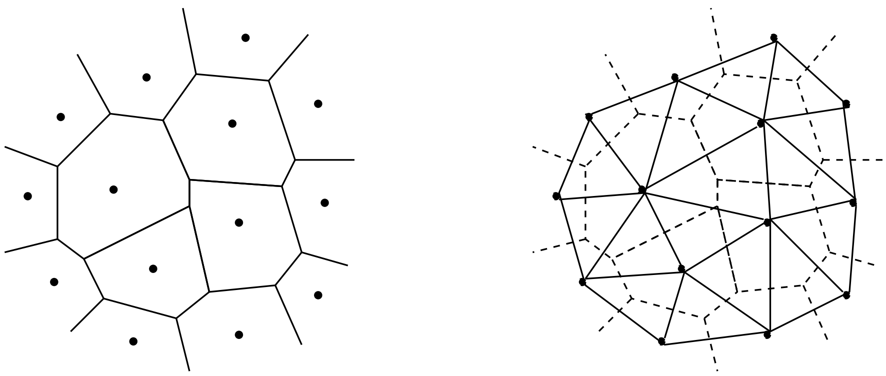
1. Initialiser les poids de façon aléatoire et le tableau des connexions à zéro ($C_{ij} = 0$).
2. Choisir un vecteur $\vec{x}$ dans la base d'apprentissage.
3. Pour chaque neurone i, déterminer son ordre k_i, c'est à dire la séquence :
$$\|\vec{x} - \vec{W_{i_0}}\| < \|\vec{x} - \vec{W_{i_1}}\| < ... < \|\vec{x} - \vec{W_{i_{N-1}}}\|. \tag{2.15}$$
4. Calculer l'adaptation de tous les neurones suivant la formule :
$$\vec{W_i}(t+1) = \vec{W_i}(t) + \varepsilon . e^{-k_i/\lambda} . (\vec{x} - \vec{W_i}). \tag{2.16}$$
5. Si $C_{i_0 i_1} = 0$, alors $C_{i_0 i_1} = 1$ et $t_{i_0 i_1} = 0$
   Si $C_{i_0 i_1} = 1$, alors $t_{i_0 i_1} = 0$.
6. Augmenter l'âge de toutes les connexions de i_0 avec $t_{i_0 j} = t_{i_0 j} + 1$ pour tous les j tels que $C_{i_0 j} = 1$.
7. Supprimer toutes les connexions de i_0 qui excèdent leur durée de vie en annulant $C_{i_0 j}$ pour tout j avec $C_{i_0 j} = 1$ et $t_{i_0 j} > T$.
8. Aller en 2.

Il y a donc dispersion d'un certain nombre de neurones dans l'espace d'entrée puis arrangement de ces neurones suivant une règle proche de la règle des cartes de Kohonen : on adapte fortement le neurone vainqueur, puis un peu le second neurone, un peu moins le troisième, etc. Cet ordre est donné par l'ordre de proximité des neurones vis-à-vis du vecteur d'entrée. De plus, à chaque itération une connexion s'établit ou se réinitialise entre les deux neurones les plus actifs ; cette connexion disparaît lorsque son utilisation (différence d'activité trop importante entre les deux neurones qu'elle connecte) se raréfie.

Plus formellement, nous pouvons interpréter géométriquement l'effet de l'algorithme mis en œuvre dans neural-gas [Martinetz et Schulten, 1994]. Celui-ci construit un diagramme de Voronoi à partir de l'ensemble des points formant le corpus d'apprentissage. Un diagramme de Voronoi est un ensemble de polygones de Voronoi. Le polygone de Voronoi $\mathcal{V}_i$ du point W_i est défini par l'ensemble des vecteurs $\vec{v}$ de la façon suivante ;

$$\mathcal{V}_i = \{\vec{v} \in \mathcal{R}^D / \|\vec{v} - \vec{W}_i\| \leq \|\vec{v} - \vec{W}_j\|, j = 1, ..., N\} \tag{2.17}$$

Le diagramme de Voronoi fournit un pavage complet de l'espace de dimension D et du point de vue connexionniste chaque polygone $\mathcal{V}_i$ correspond au champ récepteur du neurone i. Ce problème est le problème dual du problème de la triangulation de Delaunay [Martinetz et Schulten, 1994], où les point centraux des polygones de Voronoi sont utilisés pour le tracé d'un graphe : deux sommets correspondant à deux polygones adjacents sont connectés par une arrête (fig. 2.23(b)).



(a) (b)

FIG. 2.23 – *Diagramme de Voronoi (a) et triangulation de Delaunay (b) (d'après [Martinetz et Schulten, 1991])*

Dans [Martinetz et Schulten, 1994], Martinetz montre que l'utilisation de la règle de Hebb dans l'algorithme de neural-gas revient à effectuer une triangulation de Delaunay. Sur la figure 2.24, nous voyons l'état d'un réseau neural-gas au cours de l'apprentissage. La topologie à apprendre consiste en une forme en trois dimensions (parallépipède), une forme en deux dimensions (rectangle) et une forme d'une seule dimension (cercle). On constate à la fin de l'apprentissage que le réseau « épouse » les formes de la topologie en conservant les relations de voisinage définissant la topologie.

growing neural-gas

Growing neural-gas de B. Fritzke décrit dans [Fritzke, 1995] est une extension du modèle de Martinetz. La philosophie générale de l'algorithme reste la même, c'est à dire

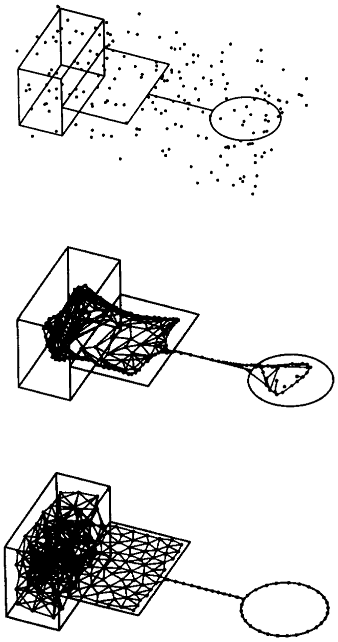


FIG. 2.24 – *Apprentissage d'une topologie par un réseau neural-gas (d'après [Martinetz et Schulten, 1991])*

qu'il y a apprentissage des poids des neurones ainsi que de la topologie de l'espace d'entrée grâce aux connexions entre ces neurones. Cependant, alors que le modèle neural-gas est initialisé avec un certain nombre de neurones repartis aléatoirement dans l'espace, le modèle growing neural-gas est un modèle évolutif. En effet, le système utilisé ici est incrémental, c'est à dire qu'au cours de l'apprentissage de nouveaux neurones sont créés et positionnés pour améliorer la segmentation de l'espace. Le principal avantage de cet algorithme par rapport au modèle neural-gas correspond au fait qu'il n'est donc pas nécessaire de spécifier la taille du réseau *a priori*.
Nous donnons ci-dessous l'algorithme d'apprentissage :

1. Démarrer avec deux unités W_a et W_b dont les poids sont initialisés de façon aléatoire.

2. Choisir un vecteur $\vec{x}$ dans la base d'apprentissage.

3. Trouver le neurone le plus proche (s_1) et le second neurone le plus proche (s_2).

4. Incrémenter l'âge de toutes les connexions émanant de s_1.

5. Ajouter le carré de la différence entre le vecteur $\vec{x}$ et le plus proche neurone à la variable d'erreur cumulée locale à s_1 :

$$\Delta err(s_1) = \|W_{s_1} - \vec{x}\|^2 \tag{2.18}$$

6. Bouger s_1 et ses voisins directs vers $\vec{x}$ avec les coefficients ε_b et ε_n selon :

$$\Delta W_{s_1} = \varepsilon_b(\vec{x} - W_{s_1}) \tag{2.19}$$

$$\Delta W_{s_n} = \varepsilon_n(\vec{x} - W_{s_n}) \tag{2.20}$$

pour tous les voisins directs de s_1.

7. Si s_1 et s_2 sont connectés, mettre l'âge de la connexion à 0. Si la connexion n'existe pas, la créer.

8. Supprimer toutes les connexions dont l'âge est supérieur à un âge maximum (a_{max}). Supprimer toutes les unités n'ayant plus de connexions.

9. Si le nombre d'itérations est un multiple de λ, insérer une nouvelle unité de la façon suivante:

   – Choisir l'unité q avec la plus grande erreur cumulée.

   – Insérer une nouvelle unité r entre q et f l'unité voisine ayant la plus grande erreur cumulée :

$$W_r = \frac{1}{2}(W_q + W_f) \tag{2.21}$$

   – Connecter r avec q et r avec f. Supprimer la connexion entre q et f.

   – Diminuer l'erreur cumulée de q et f en la multipliant par une constante α. Initialiser l'erreur de r avec la nouvelle valeur de l'erreur de q.

10. Diminuer les erreurs cumulées de toutes les unités en les multipliant pas une constante d.

11. Si un critère d'arrêt n'est pas atteint, retourner en 2.

L'algorithme est donc assez similaire à celui présenté pour le modèle neural-gas, si ce n'est maintenant la gestion des créations/destructions d'unités au sein du réseau. A cette fin, une nouvelle variable est attachée à chaque neurone : l'erreur cumulée (Δerr). Cette variable permet à chaque itération de connaître l'unité statistiquement la plus éloignée des vecteurs d'entrée précédemment présentés au réseau. Cette unité indique alors la position (représentée par le vecteur des poids $\vec{W_i}$) dans l'espace où nous devons rajouter une unité pour améliorer la segmentation.
Les résultats obtenus avec cet algorithme sont similaires à ceux obtenus avec neural-gas. En outre, à la fin de l'apprentissage, les configurations de réseaux obtenues pour les deux algorithmes sont sensiblement les mêmes. Cependant, les configurations intermédiaires caractérisent plus fidèlement la topologie dans le cas de growing neural-gas.

2.7 Le modèle de la colonne corticale

Parallèlement aux travaux menés dans le développement de modèles d'inspiration plus mathématique, nous trouvons des recherches inspirées plus directement du fonctionnement cortical et constituant un courant du connexionnisme. Parmi ces travaux, le modèle de la colonne corticale est prépondérant [Szentagothai, 1975; Hubel et Wiesel, 1977; Mountcastle, 1978; Burnod, 1988]. Nous le décrivons ici d'une part pour la remarque précédente et d'autre part parce que sa philosophie a inspiré les travaux de cette thèse.

2.7.1 Architecture

La réalité neurobiologique de la colonne corticale a été évoquée au paragraphe 2.2.3 de ce chapitre, nous nous intéressons maintenant à sa modélisation.
L'automate cellulaire modélisant la colonne corticale est organisé en niveaux à l'image de son homologue biologique. Ainsi, nous trouvons :

- un niveau regroupant les entrées et sorties vers des colonnes situées sur d'autres aires ainsi que les entrées/sorties vers des colonnes de la même aire ;
- un niveau dont les afférences proviennent du monde extérieur ;
- un niveau envoyant de l'information vers le monde extérieur.

L'automate met alors un certain nombre de connexions en jeu :

- des *liens lointains* avec des automates situés dans d'autres aires ;
- des *liens locaux* avec des automates de la même aire ;
- des *liens externes* avec le monde extérieur ;
- des *liens voisins* avec les automates voisins.

Pour tous ces liens, on considère les connexions réciproques. La figure 2.25 représente une colonne modélisée.

Les connexions voisines s'établissent entre colonnes adjacentes. Les connexions lointaines entre aires différentes respectent la topologie des aires, c'est à dire que deux colonnes voisines seront reliées à des colonnes voisines sur d'autres aires.

Enfin, les connexions externes permettent une représentation du monde extérieur. En particulier, les entrées/sorties s'effectuent sur de petites parties connexes d'une *carte de codage* du monde extérieur. On suppose alors l'existence d'une telle carte de codage. Ces connexions observent deux propriétés : une colonne est connectée à une seule zone de codage et deux colonnes voisines sont connectées à deux zones de codage voisines.

Nous n'avons pas parlé ici de la symétrie des connexions lointaines. Cette symétrie privilégie en particulier certaines recombinaisons d'information. Le lecteur trouvera de plus amples détails dans [Alexandre, 1990; Guyot, 1990]. Une différence majeure que l'on peut noter par rapport au connexionnisme « classique » est la possibilité qu'a la colonne corticale de différencier ses entrées.

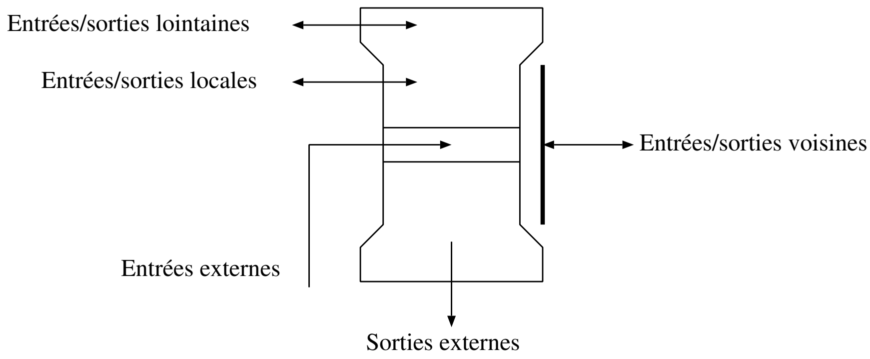


FIG. 2.25 – *Entrées/sortie de la colonne corticale. (D'après [Alexandre, 1990])*

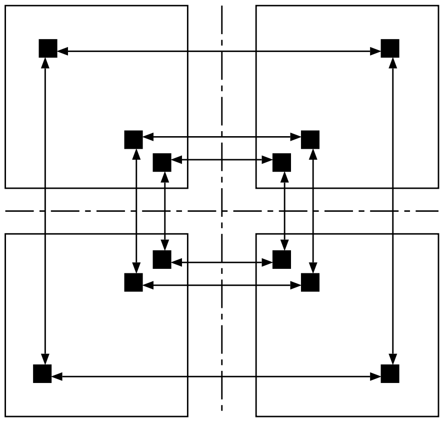


FIG. 2.26 – *Connexions entre colonnes corticales dans un réseau à quatre aires. (D'après [Alexandre, 1990])*

2.7.2 Fonctionnement

Trois états d'activation sont mis en évidence à la lumière des activations des colonnes biologiques :

- un état inhibé, E0 ;
- un état peu activé, E1 ;
- un état fortement activé, E2.

Ces états discontinus sont modulés plus finement par une variation locale C et s'interprètent en termes comportementalistes et psychologiques : E1 est un état de sensibilisation ou de pré-activation, E2 est un état d'action et E0 est un état inhibé. L'activation d'une colonne corticale dépend bien évidemment des entrées externes, lointaines, locales et voisines mais aussi du type de l'aire considérée (ou variabilité cytoarchitecturale : aires sensorielles, motrices, associatives ...), de l'apprentissage et de l'état précédent de l'automate. La table 2.1 donne les sorties externes et lointaines en fonction des entrées et de l'état précédent.

Entrées		Etat	Sorties		Commentaires	Dépend de :
Lointaines	Externes	Précédent	Lointaines	Externes		
E0	E0		E0	E0		
E0	E1		E2 E1 E0	E2 E0 E0	Sensorielle Associative Motrice	Type de l'aire
E1	E0		E0 E1 E2	E0 E0 E2	Inhibition Decl. cond. Decl. incond.	Apprentissage
E1	E1		E2	E2	Amplification	
E2		E1	E1	E0	Retour	Apprentissage
E2		E2	E2	E2	Réactivation	

TAB. 2.1 – *Table d'entrées/sorties (entrées lointaines et externes). (D'après [Alexandre, 1990])*

Les entrées externes modélisent des invariants spécifiques qui sont dépendant de la position corticale. Par exemple, les entrées/sorties externes correspondent à des indices (orientations visuelles, indices acoustiques ...) pour les aires sensorielles, des déclenchements d'actions pour les aires motrices, etc. La force des connexions est supposée définie *a priori*.
Les entrées lointaines s'interprètent en termes logiques. En supposant qu'un réseau d'automates soit capable d'identifier les sous-problèmes d'un problème global, l'état E1 (état de déséquilibre) correspond à une question posée sur un problème. Il y a alors propagation de cet état vers d'autres automates susceptibles de modifier l'état d'équilibre E1 apportant ainsi une réponse au problème.
Les entrées locales définissent une méthode de transmission de l'information de proche en proche entre automates de même aire et ceci grâce à la variable C. La transmission fait décroître cette variable à chaque pas franchissant un automate. L'unité réceptrice en fonction de la valeur de C peut ainsi localiser la provenance du signal.
Enfin les entrée voisines réactivent ou inhibent un automate suivant les conditions : il y a inhibition latérale entre automates voisins réagissant à des invariants exclusifs et ré-activation lorsque ces automates ont des fonctions de transfert voisines.

2.7.3 Apprentissage

L'apprentissage des quatres types de connexions définis plus haut ne se fait qu'en cas de besoin. Il est spécifique pour chacun des liens.

Entrées externes

Ces entrées sont câblées *a priori* mais peuvent subir quelques petites variations. Une modification est engagée s'il existe une différence entre le masque de l'automate et le stimulus, un message d'erreur précisant l'importance du stimulus, l'automate ne devant

Coefficients		Sorties		Commentaires
P2	P0	Lointaines	Externes	Commentaires
?	?	E0,E1	E0	Aléatoire
0	1	E0	E0	Inhibition
>0	<1	E1	E0	Decl. cond.
1	0	E2	E2	Decl. incond

TAB. 2.2 – *Conséquences des valeurs de P0 et P2 sur l'activation d'une colonne (d'après [Alexandre,1990])*

pas être inhibé. La règle de Hebb est alors employée pour l'adaptation du poids des connexions.

Entrées lointaines

Comme nous l'avons vu, les entrées lointaines jouent un rôle fondamental dans la décomposition de problèmes en sous-problèmes. Une façon de décider qu'un problème B est sous-problème d'un problème A est d'utiliser la corrélation temporelle. Si B se produit avant A on dira de B qu'il est sous-problème de A. On définit alors deux coefficients statistiques P0 et P2 mesurant cette corrélation. P2 est la probabilité que l'automate B soit à E2 avant A et P0 est la probabilité que B soit à E2 avant A à E0. D'où l'algorithme d'apprentissage suivant (donné dans [Alexandre, 1990]) :

1. on donne un problème par l'intermédiaire de l'aire de codage qui provoque des déséquilibres (E1) dans le réseau ;

2. on force les réponses (E2) sur une aire de codage en sortie ;

3. on met à jour les coefficients P0 et P2.

L'activité d'une colonne en fonction de P0 et P2 est donnée dans le tableau 2.2.

Entrées locales

La valeur de la variable C, qui indique l'éloignement de l'émetteur lorsqu'un message significatif est reçu, est mémorisée.

Entrées voisines

Ces entrées opèrent une division/fusion de l'espace d'entrée projeté sur les automates. Ce mécanisme simule les regroupements fonctionnels de colonnes et les inhibitions latérales différenciant ces regroupements.

2.7.4 Conclusion

Nous venons de décrire rapidement le modèle de la colonne corticale ; le lecteur pourra se reporter à [Burnod, 1988; Alexandre, 1990; Guyot, 1990] pour un exposé beaucoup plus détaillé. L'architecture ainsi que les modes de fonctionnement et d'apprentissage mettent en lumière quatre principes importants dans le cadre de simulations de comportements humains : principe de connectivité locale, principe d'intégration temporelle, principe de décomposition fonctionnelle, principe de haut niveau d'intégration. Dans le cadre de cette thèse, nous nous intéressons plus particulièrement au deuxième principe mais les trois autres sont également présents à l'esprit. On l'a vu, les liens lointains expriment des relations d'implication induisant un séquencement de l'information et par là même une prise en compte du phénomène temporel. Ceci reste tout à fait valable à un niveau d'abstraction élevé, on parle d'ailleurs de buts et de sous-buts, notions que l'on place volontiers à un niveau cognitif plutôt que sensoriel. Cependant, le traitement temporel intervient également à un niveau cognitif bien moins élevé. La reconnaissance de la parole est un exemple parmi d'autres où un phonème, une syllabe ou un mot sont des séquences acoustiques. Or dans ces cas, la granularité de l'information est beaucoup plus importante qu'à des niveaux d'abstraction plus élevés engendrant de fortes variations d'une occurrence à une autre. Nous l'avons vu, une colonne corticale réagit parce qu'elle correspond à une information spatiale particulière et aussi parce qu'elle peut être déclenchée par des colonnes correspondant à des sous-buts. Il y a donc introduction de la notion de séquence. Mais comment différencier l'activation d'une colonne par un stimulus particulier sous un certain contexte de l'activation sous un autre contexte ? Ceci est fondamental dans une grande partie des problèmes sensoriels mais n'est pas explicitement traité dans le modèle de la colonne corticale.

2.8 Neurobiologie, modèles et théorie unificatrice

Ce chapitre consacré aux bases du connexionnisme n'avait pas pour but d'être une énumération exhaustive des modèles que l'on rencontre classiquement. A partir des observations faites en neurobiologie, nous avons décrit le courant majoritaire en connexionnisme, c'est-à-dire l'ensemble des modèles développés et des recherches effectuées en connexionnisme d'inspiration mathématique ou physique. Nous nous sommes de plus intéressé au courant d'inspiration plus neurobiologique, d'une part parce qu'il nous semble prometteur d'autre part parce que notre travail s'y insère. Le modèle de la colonne corticale reste volontairement général pour pouvoir s'appliquer à différentes situations, il se veut une *modélisation fonctionnelle du cortex*. Dans ce sens, il pourrait être considéré comme théorie unificatrice incluant les principes de base du connexionnisme. Mais parce qu'elle n'en est qu'à ses débuts, la colonne souffre de manques qu'ils convient de combler pour des applications pratiques. En particulier, la prise en compte du temps n'est pas complètement intégrée au modèle.

Notre objectif n'est pas d'ajouter un module au modèle de la colonne corticale pour pouvoir traiter du temps mais plutôt de repartir sur un nouveau modèle fondé sur les principes et la philosophie de la colonne corticale et plus particulièrement dédié au traitement temporel. Ceci nous permet, entre autres, de simplifier certains mécanismes de la colonne pour mieux se focaliser sur le problème qui nous préoccupe.

Chapitre 3

Le temps dans les systèmes connexionnistes

« Qu'est-ce en effet que le temps ? Qui serait capable de l'expliquer facilement et brièvement ? Qui peut le concevoir, même en pensée, assez nettement pour exprimer par des mots l'idée qu'il s'en fait ? Est-il cependant notion plus familière et plus connue dont nous usions en parlant ? Quand nous en parlons, nous comprenons sans doute ce que nous disons ; nous comprenons aussi, si nous entendons un autre en parler.
Qu'est-ce donc que le temps ? Si personne ne me le demande, je le sais ; mais si on me le demande et que je veuille l'expliquer, je ne le sais plus. Pourtant, je le déclare hardiment, je sais que si rien ne passait, il n'y aurait pas de temps passé ; que si rien n'arrivait, il n'y aurait pas de temps présent. [...] »

Saint Augustin, Les confessions, livre 11, ch XV, 400.

L'intérêt grandissant pour la prise en compte du paramètre temporel dans les réseaux neuromimétiques a fait apparaître un certain nombre de nouveaux modèles. Tous ces modèles n'abordent pas le problème de la dimension temporelle de la même façon et nous pouvons dénombrer un nombre assez conséquent de systèmes aux architectures assez variées. Dans leur majorité, les systèmes temporels sont des extensions de modèles connexionnistes statiques, ils peuvent également faire appel à des techniques non neuronales (programmation dynamique, chaînes de Markov ...) constituant ainsi des « modèles hybrides ». Cependant, de nouvelles approches sont mises en œuvre dans des modèles le plus souvent issus de l'inspiration du vivant. Si ces derniers modèles ne fournissent pas

encore de résultats au moins égaux à ceux générés par les modèles classiques, ils offrent une nouvelle approche qui semble relativement élégante pour traiter les problèmes du monde réel. Car le temps n'est pas la seule motivation dans ces recherches sur les modèles temporels. Le connexionnisme se cantonne souvent à des problèmes de perception et ses modèles souvent à des « boîtes noires » pour lesquelles il est difficile d'interpréter quoi que ce soit. Or il semble difficile de traiter les problèmes du monde réel avec une boîte noire. Un raisonnement, par exemple, est une succession d'états inférés grâce aux précédents, cette suite logique doit pouvoir être prise en compte par un système « intelligent » pour pouvoir fournir des résultats corrects. Certes, nous n'en sommes pas à concevoir des systèmes connexionnistes capables de raisonner, mais il paraît opportun de concevoir des architectures permettant des traitements séquentiels.
Dans ce chapitre, nous décrivons de façon non exhaustive des architectures connexionnistes intégrant la notion de temps. Plusieurs critères de classification de ces modèles peuvent être utilisés. Une façon serait de poser l'antagonisme entre les systèmes d'inspiration plus mathématique et les systèmes d'inspiration plus biologique. Une autre consisterait à classer les systèmes selon leur application et selon leurs aptitudes au flou spatial et temporel. Nous avons choisi de classer les différents systèmes temporels en fonction du codage du temps mis en œuvre.

3.1 Classification des modèles temporels

Le temps est vu ici sous l'aspect séquentiel, on considère donc l'ordonnancement de l'information sous la forme de séquences. En fonction d'une telle séquence d'informations, un réseau connexionniste peut fournir ou prédire une réponse. Les informations composant les séquences sont des événements que l'on qualifiera de ponctuels. Nous supposons les avoir obtenus par un prétraitement (échantillonnage), connexionniste ou non.
Selon la méthode de prise en compte des séquences d'événements, nous pouvons dégager deux grandes classes de réseaux connexionnistes temporels [Durand et Alexandre, 1994b; Durand et Alexandre, 1995b]. La première inclut les modèles pour lesquels la séquence est en fait représentée dans sa totalité à l'extérieur du réseau. Le problème revient alors à un problème de classification de forme statique. La seconde classe correspond aux modèles qui intègrent le temps à l'intérieur de leur architecture. Dans ce cas, l'information parvient au réseau au fur et à mesure de son émission dans le temps et le système fournit une réponse au bout d'un temps donné. Dans cette seconde classe de modèles, nous distinguons deux sous-types de représentation qui distinguent le traitement temporel par la succession des états stables du réseau (représentation *implicite*) du traitement temporel par propagation d'activité (représentation *explicite*). Les modèles d'inspiration biologique semblent se classer parmi les modèles de cette dernière sous-classe. Dans ce qui suit, nous donnons quelques exemples de modèles temporels. L'énumération respecte le critère de classification énoncé ci-dessus.

3.2 Modèles à représentation externe

C'est le mode de représentation le plus courant actuellement. La technique consiste à « spatialiser » le temps, c'est à dire à considérer le paramètre temporel comme une dimension équivalente aux autres. Ainsi, nous pouvons facilement représenter une forme dynamique (évoluant dans le temps) par une «image» que le réseau utilise comme entrée. La concaténation des vecteurs composant une séquence revient à considérer un seul vecteur d'entrée.

3.2.1 Classification de séquences par des méthodes classiques

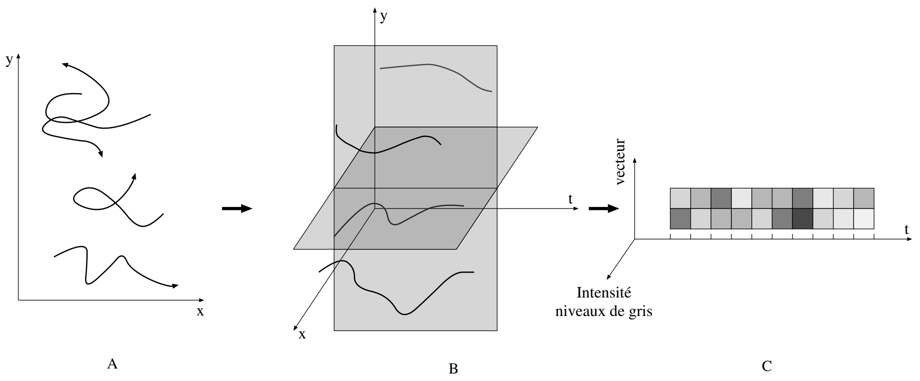


FIG. 3.1 – *Classification de séquences*

Sur la figure 3.1(A), sont représentées quatre séquences. L'espace des stimuli étant en deux dimensions (x,y), nous ajoutons un axe supplémentaire (t) pour coder le temps. Une séquence acceptable par un réseau est représentée sur la partie B. Le problème revient alors à classer une image, une forme particulière, c'est à dire à trouver une fonction de transfert entre l'espace d'entrée tel que celui représenté sur la partie C et la sortie désirée. Les perceptrons multi-couches [Rumelhart et McClelland, 1986] sont particulièrement bien adaptés à ce genre de tâche et servent souvent de base à des systèmes plus complexes comme nous allons le voir dans le paragraphe suivant.

L'architecture et l'apprentissage du perceptron ont été présentés au chapitre 2. Dans l'aspect temporel qui nous préoccupe, les séquences d'information sont transformées en une matrice telle que celle représentée sur la partie C de la figure 3.1. Les MLPs ont la propriété importante de pouvoir généraliser par rapport aux formes précédemment apprises, cela permet donc de reconnaître des séquences un peu différentes de celles qui ont été apprises. Bien que ces réseaux soient considérés comme des classifieurs universels, les performances des perceptrons multi-couches (MLPs) sont largement tributaires de la base de données d'apprentissage utilisée. Ainsi, lorsque le problème dynamique à étudier introduit des distorsions temporelles assez importantes (voir le cas de la parole par exemple), les «images» présentées au réseau peuvent différer fortement des «images» apprises. Dans ce cas, la relaxation du réseau (calcul des activités des neurones) conduit à un échec. Ceci est encore plus valable lorsqu'un décalage temporel est introduit au niveau de l'entrée. Ces configurations sont représentées sur la figure 3.2.

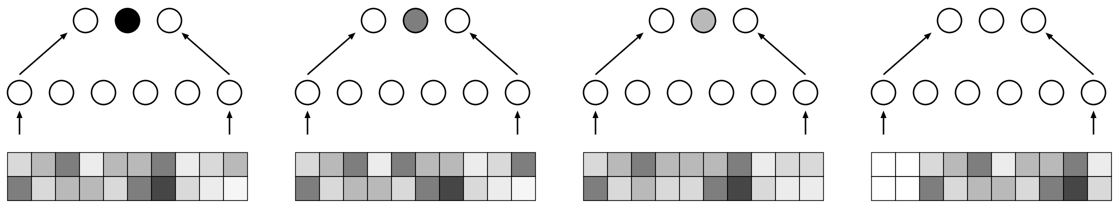


FIG. 3.2 – *Séquences bruitées et réactions du perceptrons*

On le voit bien, un perceptron ne peut donc pas intrinsèquement prendre en compte des variations temporelles. Ceci constitue un handicap certain lorsque le problème à traiter est soumis à de fortes variations temporelles comme en reconnaissance de la parole. Il s'avère donc nécessaire de définir d'autres types d'architectures.

Bien que moins utilisés pour le traitement temporel, des systèmes connexionnistes auto-organisés sont utilisés pour des applications où l'information existe sous la forme de séquences. Par exemple, dans [Ferran *et al.*, 1994], il est proposé une méthode de classification de séquences protéiques à base d'une carte de Kohonen [Kohonen, 1989]. Après apprentissage, la carte obtenue définit des classes de séquences : les séquences protéiques voisines se retrouvent codées par le même neurone ou par des neurones voisins. Tout le problème du temps est traité dans le codage de l'entrée de la carte. Dans le cas des séquences protéiques, des matrices dérivées de la composition dipeptidique des protéines sont utilisées.
Souvent, le choix de l'utilisation d'un réseau classique (MLP ou carte de Kohonen) dépend fortement du type de problème traité. Dans le problème de la reconnaissance et la classification de séquences protéiques, il y a relativement peu de variabilité entre deux occurrences d'une même séquence. La carte de Kohonen ne traitant pas vraiment la variabilité temporelle, de fortes différences temporelles dans les séquences n'auraient pas conduit à des résultats satisfaisants.
Dans la suite, nous citons d'autres types de réseaux connexionnistes temporels issus des modèles classiques mais commençant à tenir compte des variabilités temporelles, notamment la distorsion.

3.2.2 Le TDNN (Time Delay Neural Network)

Le TDNN est un réseau multi-couches dont la composante de base est un MLP. Il a été développé dans le but d'apprendre des structures spectrales spécifiques à l'intérieur de vecteurs de parole consécutifs. D'une couche à l'autre, l'information spectrale est compressée pour obtenir sur la dernière couche un vecteur de neurones de dimension réduite correspondant aux formes dynamiques à apprendre, en l'occurrence les trois occlusives voisées /B,D,G/ dans l'article original (voir l'application du TDNN à la reconnaissance de la parole au chapitre 5).

Le TDNN a montré sa capacité de reconnaissance avec des décalages temporels au niveau du signal d'entrée. Cependant, il conserve les limitations des MLPs pour le problème de la distorsion temporelle. Pour des applications à la reconnaissance de séquences acoustiques plus longues que celles correspondant à des phonèmes, des systèmes composés de plusieurs TDNN ont été mis en œuvre. Dans la suite, nous décrivons l'architecture du TDNN puis le mode d'apprentissage. Enfin, nous étudions le comportement du réseau en phase de reconnaissance.

Architecture temporelle

Chaque neurone effectue une somme pondérée avec ses $N * (D+1)$ poids et utilise une sigmoïde comme fonction de transfert. N représente la dimension d'entrée du vecteur $\vec{X}(t)$ et $(D+1)$ représente le nombre de vecteurs $\vec{X}$ successifs que le neurone utilise pour calculer son activation, c'est à dire la taille de la fenêtre temporelle sur laquelle le neurone réagit. La figure 3.3 représente un tel neurone.

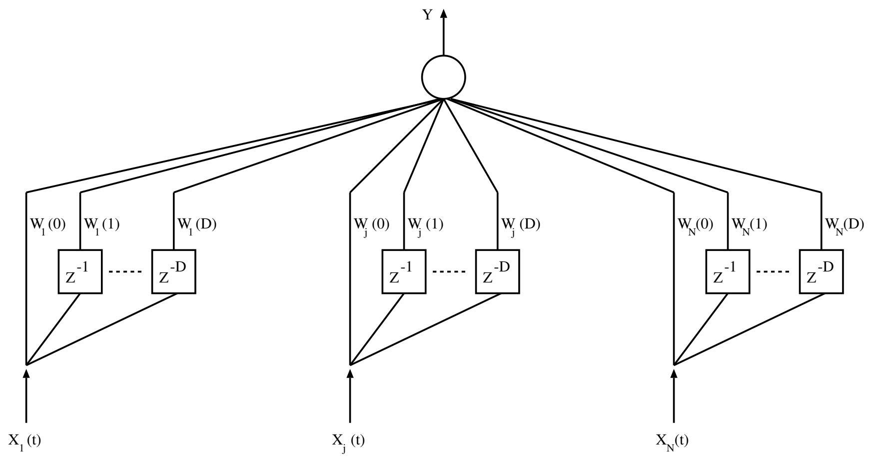


FIG. 3.3 – *Architecture du neurone du TDNN*

Une couche cachée est une colonne de neurones complètement interconnectés avec tous les neurones correspondant à une fenêtre temporelle de longueur $(D+1)$ sur la couche inférieure. Cela signifie que lorsqu'une couche cachée est connectée à une autre couche cachée, il faut mémoriser les activités de la colonne de la couche inférieure sur un pas de temps correspondant au délai des neurones de la couche supérieure. La couche de sortie possède une connectivité plus particulière puisqu'un neurone à délai de la couche cachée n'intègre l'information que sur une seule dimension de la colonne de la dernière couche cachée. Ceci implique que la séquence d'information pour une seule dimension de la dernière couche cachée est spécifique d'une classe particulière à coder. De plus, les poids pour tous les neurones de sortie avec la dernière couche cachée sont identiques tout comme les poids attachés à deux colonnes différentes d'une même couche cachée.
L'architecture du TDNN est résumée sur la figure 3.4.

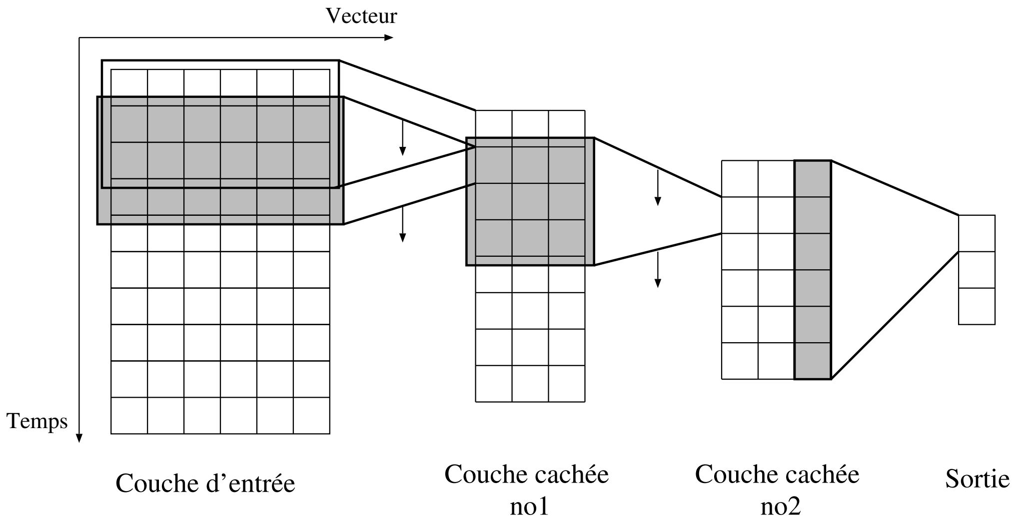


FIG. 3.4 – *Architecture du TDNN*

Apprentissage

L'apprentissage des poids de connexion du TDNN utilise l'algorithme de rétro-propagation classique. Cependant, du fait qu'un neurone, pour pouvoir calculer complètement son activation, a besoin d'attendre l'information sur un certain pas de temps, il n'est pas possible de modifier les poids à tout instant entre la couche de sortie et la couche d'entrée. C'est la raison pour laquelle, avant de « lancer » l'algorithme de rétro-propagation, il y a mémorisation des colonnes nécessaires à l'activation des autres colonnes et duplication des poids qui sont donc partagés. La figure 3.5 montre la mémorisation de colonnes entre deux couches ainsi que la duplication associée des poids.

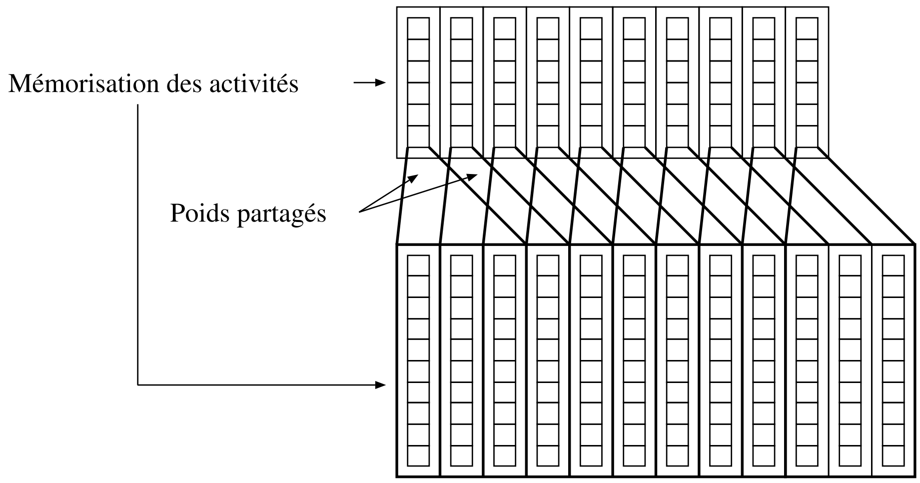


FIG. 3.5 – *Stockage des informations au-cours du temps pour l'apprentissage des poids du TDNN*

Le TDNN revient alors à un perceptron classique pour lequel nous pouvons classiquement appliquer la rétro-propagation [Rumelhart et McClelland, 1986] du gradient de l'erreur de sortie. Cependant, les modifications à apporter au niveau des poids sont conservées pour toutes les colonnes et une moyenne est calculée pour chacun des poids partagés. Cette moyenne est ensuite ajoutée à chacun des poids correspondants et nous retrouvons une matrice de poids équivalente pour chaque colonne.

Comportement

Si l'on observe la dernière couche cachée d'un TDNN, nous pouvons constater une relative invariance de la configuration des activités au sein de cette couche pour des contextes différents. Par exemple, dans le modèle appliqué à la reconnaissance des trois occlusives /B,D,G/, les deux configurations de la dernière couche cachée pour la prononciation des deux syllabes « da » et « do » sont pratiquement identiques. Ceci montre l'abstraction du contexte qui peut être faite par un TDNN pour la représentation d'un phonème quel que soit son contexte (voir fig. 5.9 du chapitre 5).

De plus, il faut noter la bonne performance du TDNN dans le cas d'un décalage temporel du signal d'entrée (par exemple un spectrogramme de parole). Ceci est dû aux poids partagés permettant la détection d'une information à tout instant même si elle a été apprise à un instant particulier. La figure 3.6 montre la configuration des activités d'un TDNN pour la présentation de deux signaux décalés à gauche puis à droite par rapport au signal appris. La réponse du réseau est toujours correcte dans les deux cas.

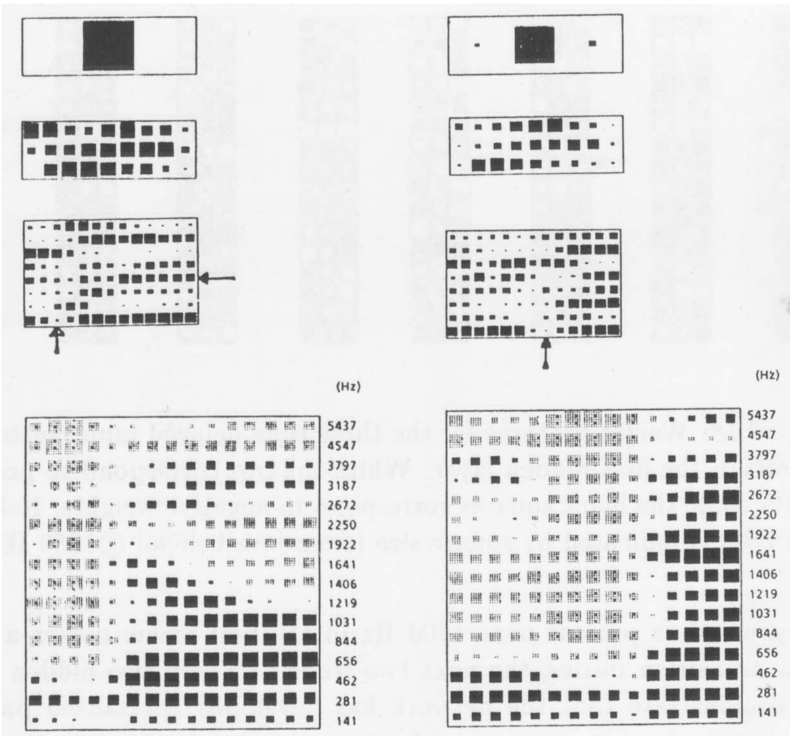


FIG. 3.6 – *Activités du TDNN pour la prononciation de "do" décalé par rapport à la syllabe apprise de +30ms à gauche et -30ms à droite (d'après [Morgan, 1991])*

3.2.3 Extensions du TDNN

Pour des applications de reconnaissance sur des chaînes d'événements acoustiques plus longues (reconnaissance de mots multi-locuteurs, par exemple) des systèmes à bases de TDNNs, plus complexes, semblent requis [Haffner et Waibel, 1992; Hampshire II et Waibel, 1992]. Ceci est dû principalement au fait que les réseaux neuronaux dans leur globalité n'intègrent pas de procédure d'alignement temporel. Or, ceci paraît fondamental pour obtenir de bonnes performances dans les applications, en particulier dans le cas de la reconnaissance de la parole. C'est la raison pour laquelle bon nombre de systèmes connexionnistes sont couplés avec des procédures de traitement temporel – on parle alors de modèles hybrides. Le TDNN n'y échappe pas. Nous n'allons pas décrire ici les extensions du TDNN, ceci sera fait au chapitre 5 car les applications à la reconnaissance de la parole sont, dans la plupart des cas, à l'origine de ces modèles.

3.2.4 Le modèle NPM

[Iso et Watanabe, 1990] propose un modèle de reconnaissance de la parole à base de MLP et de recalage temporel par programmation dynamique. L'architecture utilisée est une séquence de N MLPs (neural prediction model, NPM). Chaque MLP dans la séquence prend τ vecteurs en entrée afin de prédire le $(\tau + 1)$-ème vecteur. En phase de reconnaissance, le mot à tester est découpé en N morceaux. La segmentation optimale est obtenue par programmation dynamique. Dans ce système, on voit nettement apparaître la notion de séquences. Cependant, le découpage effectué sur le signal n'est pas calculé par le réseau lui même mais par un algorithme «extérieur». Le temps est ainsi représenté à l'extérieur du réseau.

3.2.5 Discussion

Ces quelques exemples montrent une technique possible pour tenir compte de la dimension temporelle de certains problèmes comme la reconnaissance de mots. Les résultats obtenus pour des applications bien ciblées sont appréciables. La majeure partie de ces systèmes est cependant très influencée par le domaine d'application. Le découpage en segments, la taille de la fenêtre d'intégration des neurones sont quelques paramètres non pas traités par le réseau mais par l'utilisateur ou des algorithmes extérieurs. Une approche alternative à ces systèmes concerne la prise en compte d'un flot d'information événement par événement avec un système dont le but est de fournir une réponse au bout d'un temps fini. Ce mode de prise en compte de formes dynamiques est discuté ci-dessous.

3.3 Modèles à représentation interne et implicite

Dans ce type de représentation, les vecteurs composant la forme à étudier sont intégrés par le réseau au cours du temps. Ceci est également valable pour le type de représentation décrit au paragraphe suivant. Cependant, dans cette alternative, le temps n'est pas explicitement codé sur le réseau. C'est à dire que le passé ou des morceaux du passé sont codés par des états internes du système.

3.3.1 Les systèmes dynamiques

La définition donnée au terme « dynamique » est souvent générale et naturellement opposé à « statique ». Dans le domaine du connexionnisme, les relations entre entrées et sorties sont considérées comme instantanées pour des modèles statiques. A partir du moment où cette relation est durable dans le temps on considère le système connexionniste comme dynamique. En particulier, certains modèles à représentation externe du temps, comme par exemple le TDNN, sont considérés dans cette nomenclature comme dynamiques. Dans notre classification nous affinons ce terme en précisant le type de codage du temps. Nous introduisons cet aspect des choses au niveau de la représentation interne et implicite du temps parce que les modèles dynamiques originaux se servent de ce type de représentation. En effet, les réseaux d'automates [Gas, 1994], en particulier le jeu de la vie de Conway, ont des états qui évoluent au cours du temps pour se stabiliser sur des positions finales à partir d'un état initial.

Dans la suite, nous présentons le modèle de Hopfield dont la philosophie rejoint celle de l'évolution dynamique des états. Une petite perturbation en entrée du réseau induit une évolution de celui-ci vers un état d'équilibre. Nous présentons également les modèles récurrents à couche pour lesquels l'évolution de l'état du réseau se fait sous l'impulsion d'une séquence et non pas d'une perturbation des entrées.

3.3.2 Le modèle de Hopfield

L'approche de Hopfield publiée dans [Hopfield, 1982] marque le renouveau du connexionnisme au début des années 80. Elle suppose que le système nerveux recherche en permanence des états stables attracteurs. Hopfield propose un modèle de réseaux utilisant des neurones de MacCulloch et Pitts qui sont complètement interconnectés entre eux. Le système est vu comme une mémoire adressable par son contenu, c'est à dire qu'il suffit d'exciter une partie du réseau avec une partie correcte de l'entrée pour aboutir sur un état stable correspondant à l'entrée totale correcte. Du point de vue temporel, le modèle de Hopfield rentre dans le cadre des systèmes dynamiques, c'est-à-dire que son état à un instant t dépend de son état aux instants passés. Des résultats rigoureux sur ce modèle peuvent être trouvés dans [Bourret *et al.*, 1991].

Architecture

Un modèle de Hopfield est un réseau complètement connecté (fig. 3.7) de N neurones de MacCulloch et Pitts. Chacun de ces neurones calcule donc une somme pondérée de ses entrées $I_i = \sum_j T_{ij} Xj$ où T_{ij} représente le poids de connexion entre les deux neurones i et j. La sortie Y_i du neurone i calculée à l'instant $t+1$ est donnée par l'équation :

$$Y_i(t+1) = \begin{cases} +1 & \text{si } \sum_{j=1}^{N} W_{ij} Y_j(t) > 0 \\ -1 & \text{si } \sum_{j=1}^{N} W_{ij} Y_j(t) \leq 0 \end{cases} \tag{3.1}$$

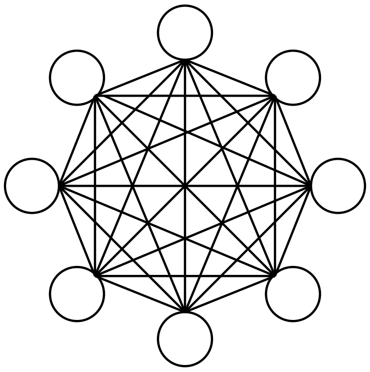


FIG. 3.7 – *Réseau de Hopfield*

L'état du système est donné par les valeurs de sortie des N neurones et peut donc être représenté par un mot de N bits. L'ordre de modification de l'état d'un neurone dans le temps est donné de façon aléatoire. A chaque top d'horloge, on choisit donc aléatoirement un neurone pour calculer son activité. Cependant, d'autres dynamiques peuvent être choisies comme par exemple une modification synchrone de l'activité de tous les neurones ou une modification suivant un ordre pré-déterminé.

Apprentissage

Le mode d'apprentissage des modèles de Hopfield est relativement différent du mode d'apprentissage par essai-erreur du perceptron. La connaissance de l'ensemble des états à mémoriser conduit directement au calcul des poids du réseau. Alors que les travaux théoriques concernant le perceptron se focalisent principalement sur les méthodes et la convergence de l'algorithme d'apprentissage, les études pour le modèle de Hopfield se concentrent plus sur la relaxation, en particulier l'étude des bassins d'attraction.
On suppose vouloir apprendre un ensemble de p états $\{\vec{X}\}_s$ correspondant aux états attracteurs du réseau. Ces états sont retrouvables à partir d'états légèrement différents. La loi d'affectation des poids pour le stockage de ces états est donnée par la règle :

$$T_{ij} = \sum_p Y_i^p . Y_j^p \tag{3.2}$$

$$T_{ii} = 0 \tag{3.3}$$

La formulation 3.2 traduit l'application de la règle de Hebb après codage d'un état sur le réseau.

Formellement, une condition suffisante de stabilité du réseau est que les états soient deux à deux orthogonaux. En effet, nous avons :

$$\begin{aligned} \sum_j W_{ij}.Y_j^p &= \sum_j (\sum_k Y_i^k.Y_j^k).Y_j^p && (3.4) \\ &= \sum_k Y_i^k.(\sum_j Y_j^k.Y_j^p) && (3.5) \\ &= \sum_k Y_i^k. < Y^p, Y^k > && (3.6) \end{aligned}$$

d'où $Y_i^p(t+1) = Y_i^p(t)$ si les états sont deux à deux orthogonaux.

Dynamique du modèle de Hopfield

Par rapport au domaine qui nous intéresse dans nos travaux, le modèle de Hopfield n'est pas à proprement parler un modèle permettant d'apprendre et de stocker des séquences d'informations dans le traitement des problèmes incluant une dimension temporelle. Cependant, intrinsèquement le réseau utilise ses états passés pour évoluer vers un état d'équilibre, c'est en ce sens que l'on peut le qualifier de temporel : l'état du réseau à un instant t dépend de l'état à l'instant précédent $E(t) = f(E(t-1))$. Le réseau de Hopfield est donc parfois utilisé comme principe de base pour des modèles d'apprentissage de séquences comme nous allons le voir dans la suite.

Limites du modèle de Hopfield

L'affectation des poids, décrite par la règle de Hebb dans la formule 3.2, introduit généralement des états attracteurs indésirables. En déséquilibre suite à l'initialisation de l'état du réseau, celui-ci peut tomber dans un « état poubelle ». Afin de remédier à ce problème, différentes techniques ont été proposées, en particulier des techniques issues de théories de la physique. Une telle technique est présentée dans le paragraphe suivant.
Nous avons vu précédemment la condition d'orthogonalité deux à deux des vecteurs d'états comme condition suffisante de stabilité. Cependant, il est parfois difficile d'assurer l'orthogonalité des états. C'est pourquoi dans [Personnaz *et al.*, 1986], il est proposé une méthode de modification des poids permettant une stabilité des états orthogonaux ou non.
Enfin, le modèle de Hopfield peut stocker un nombre d'états stables limité par la taille du réseau. Ainsi le rapport du nombre d'exemples à stocker sur le nombre optimal de neurones avoisine 0.14. Ce rapport atteint, l'ajout d'un exemple supplémentaire conduit à faire disparaître de la mémoire des exemples précédemment appris. Ce phénomène est connu sous le nom « d'oubli catastrophique ».

3.3.3 Extension du modèle de Hopfield pour la dimension temporelle

B. Gas dans [Gas, 1994] propose une extension du modèle de Hopfield dans le domaine temporel afin d'apprendre de manière auto-organisée des séquences temporelles. Les séquences sont ici des séquences d'informations binaires représentant des trains de pics (fig. 3.8).

Pour chacune des connexions w_{ij} entre les neurones, une variable, dite *temps d'intégration*, est associée à w_{ij}. L'apprentissage intervient exclusivement sur ces variables.
Le principe de fonctionnement consiste à introduire deux types de calcul de l'activité des unités. Ainsi, l'activité peut se calculer de façon déterministe et dépend uniquement des sorties des cellules pré-synaptiques (fonction classique de seuillage), on parle alors d'activité évoquée, ou bien elle peut se calculer de façon aléatoire et indépendamment de ses entrées (générateur aléatoire), on parle alors d'activité spontanée. Le choix de la fonction

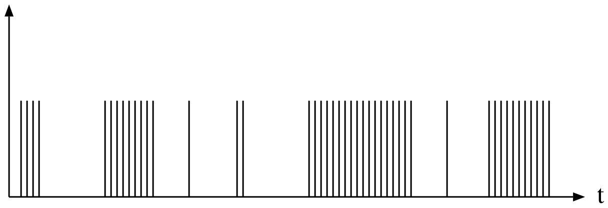


FIG. 3.8 – *Trains de pics (ou spikes)*

d'activation (évoquée ou spontanée) se fait en évaluant l'activité pré-synaptique évoquée et spontanée (voir [Gas, 1994]).
La reconnaissance d'une séquence en entrée est définie qualitativement comme suit : *une séquence est reconnue par le réseau s'il existe un sous ensemble de cellules calculant dans le mode évoqué dont l'activité est synchronisée avec celle des cellules d'entrée.* En phase d'apprentissage, les temps d'intégration sont adaptés afin d'augmenter le nombre d'unités évoquées à l'instant suivant.
Le principe de reconnaissance consiste à mettre en «résonance» le réseau avec les séquences qui lui sont présentées.

3.3.4 Les réseaux récurrents à couches

Les approches précédemment décrites font partie de la classe générale des réseaux récurrents. Nous parlons de récurrences lorsque les activités de deux neurones i et j dépendent directement ou indirectement l'un de l'autre par opposition aux liens feed-forward où l'activité va de couche en couche. Dans ces modèles, l'aspect temporel est vu sous l'angle de la dynamique d'évolution du réseau. En effet, sous l'influence d'une perturbation des neurones d'un réseau, celui-ci génère un certain nombre d'états (valeurs d'activation des neurones) pour finalement se stabiliser sur un état stable ou attracteur. Il n'y a pas à proprement parler de fonction de transfert à apprendre. Parmi les réseaux récurrents, nous pouvons distinguer la classe particulière des réseaux à couches avec des connexions récurrentes. Une différence importante avec les modèles complètement interconnectés (Hopfield, Boltzmann, ...) concerne l'apprentissage d'une fonction de transfert. Etant organisés en couches, ces réseaux possèdent une entrée et une sortie et apprennent donc des relations à la manière des perceptrons : couples (entrée, sortie). Cependant, alors que l'apprentissage est complètement statique dans les perceptrons, l'entrée globale des réseaux à couches récurrents contient une information sur le passé. La sortie du réseau dépend donc de l'information qui est fournie au système à l'instant t et du passé de ce même système sur une plage de temps $[t - \Delta t, t[$.
Soit une suite d'événements ponctuels $\{e_t\}$. Dans la littérature la prédiction de séries temporelles peut se faire classiquement par l'utilisation de filtres linéaire [Gershenfeld et Weigend, 1994]. Le filtre FIR (Finite Impulse Response) dont la description mathématique est donnée dans l'équation 3.7 permet l'expression d'une suite $\{x_t\}$ en fonction de la donnée d'une suite en entrée.

$$x_t = \sum_{n=0}^{N} b_n e_{t-n} \tag{3.7}$$

Les b_n représentent les coefficients du filtre. Dans les cas où l'on ne souhaite pas « gouverner » la série de façon extérieure, nous avons recours aux filtres auto-régressifs ou IIR (Infinite Impulse Response) définis mathématiquement par la formule 3.8.

$$x_t = \sum_{m=1}^{M} a_m x_{t-m} + \sum_{n=0}^{N} b_n e_{t-n} \tag{3.8}$$

Dans cette expression, e_t représente les variables de contrôle et x_t les états de sortie précédents du filtre. Cette formulation est connue sous le nom de modèle ARMA paramétré par M et N [Gershenfeld et Weigend, 1994].
Les réseaux à couches récurrents simulent des filtres impulsionnels à réponses infinies (IIR) avec la possibilité d'une non-linéarité. La relation entre la sortie calculée d'un réseau en fonction des états d'entrée (e_t) et des états internes passés (x_t) est alors régie suivant la règle générale :

$$x_t = f(x_{t-1}, ..., x_{t-M}, e_t, ..., e_{t-N}) \tag{3.9}$$

Pratiquement, les réseaux neuronaux récurrents calculent leur sortie en fonction du passé et de l'entrée courante, c'est à dire de l'information présente à l'instant t. En l'occurrence, il n'y a généralement pas mémorisation d'une suite d'informations passées. Dans ce cas, l'équation 3.9 s'écrit plus particulièrement avec $N = 0$. Le schéma général d'un réseau récurrent à couche est décrit sur la figure 3.9.

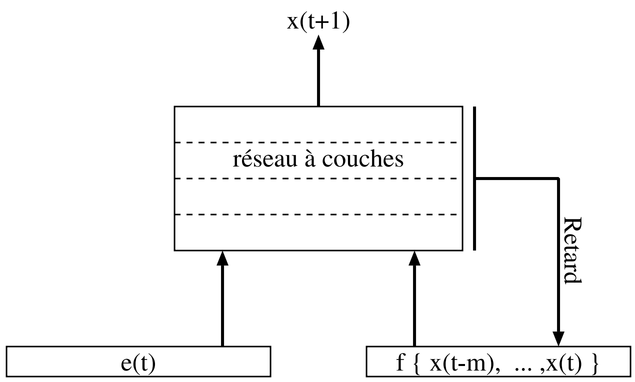


FIG. 3.9 – *Architecture générale des réseaux récurrents*

Plusieurs architectures récurrentes ont été proposées pour l'apprentissage séquentiel dans les réseaux neuronaux. Nous présentons dans la suite les modèles les plus courants.

Le modèle de Jordan

L'architecture définie par Jordan [Jordan, 1986] a pour base un perceptron multicouche à une seule couche cachée. Il y a donc interconnexion totale entre les couches successives. De plus, les neurones de la couche de sortie sont connectés à une partie des neurones de la couche d'entrée. Ces neurones correspondent aux cellules codant les états du réseau (x_t). Les connexions récurrentes entre ces deux groupes de neurones sont pratiquement des lignes à retard permettant au réseau de disposer à un instant t de sa sortie à l'instant $t-1$, elles sont valuées à 1. De plus, chacun des neurones d'état possède une boucle récurrente. Les poids de connexion sont fixes tout au long de l'apprentissage qui utilise une généralisation de l'algorithme de rétro-propagation du gradient de l'erreur. Les neurones restant sur la couche d'entrée définissent le plan de la séquence. L'ensemble des neurones définissant ce plan reste figé durant la génération de la séquence, il constitue en quelque sorte la clef de la séquence. A partir de cette clef et de l'état courant, le réseau génère un nouvel état qui est utilisé à l'instant suivant grâce aux connexions récurrentes.

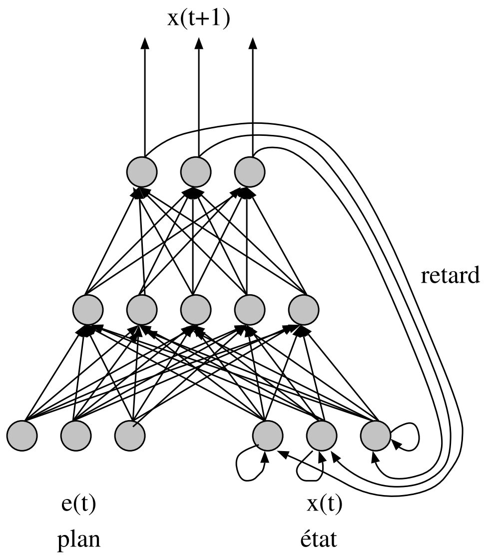


FIG. 3.10 – *Architecture du modèle de Jordan*

Nous pouvons adapter l'équation 3.9 au cas du réseau de Jordan de la façon suivante :

$$x_t = f(x_{t-1}, e) \tag{3.10}$$

où e représente le plan d'entrée et x_t l'état du réseau à l'instant t. Ce modèle a notamment été appliqué dans le domaine de la production de parole [Lippmann, 1989].

Le modèle de Elman

Le modèle de Elman [Elman, 1990], dont l'architecture est donnée sur la figure 3.11, fonctionne différemment du modèle de Jordan. En effet, alors que ce dernier modèle utilise une «clef» comme générateur de séquence, le modèle de Elman intègre de l'information au cours du temps pour fournir une séquence d'informations en sortie. Comme pour le modèle de Jordan, l'architecture du modèle de Elman est basée sur un perceptron à une couche cachée. La couche d'entrée se décompose en deux ensembles de neurones : un ensemble codant le vecteur d'entrée et un ensemble codant l'état passé du réseau définissant ainsi le contexte. Grâce aux connexions récurrentes entre la couche cachée et les unités de contexte, il y a recopie ou mémorisation de l'activité de la couche cachée sur une partie de la couche d'entrée. Ces connexions récurrentes sont des lignes à retard dont les poids sont immuables au cours de l'apprentissage et sont valués à 1. Une généralisation de l'algorithme de rétro-propagation est utilisée pour l'adaptation des poids.

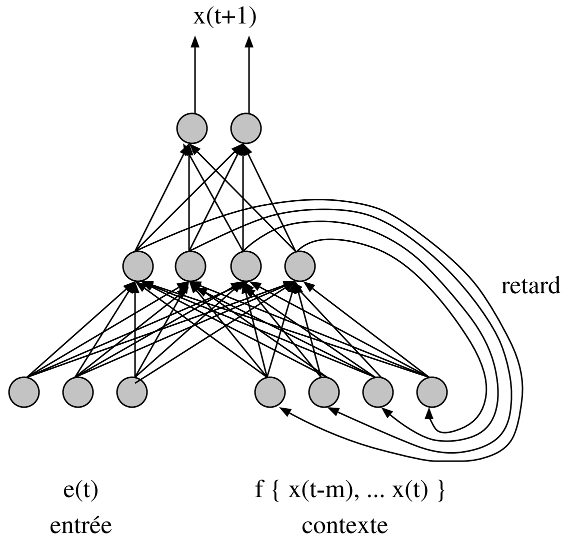


FIG. 3.11 – *Architecture du modèle de Elman*

Dans le cas particulier de ce modèle, l'équation 3.9 devient :

$$x_t = f(x_{t-1}, ..., x_{t-M}, e_t) \tag{3.11}$$

Ceci traduit le fait que la couche cachée (ou la partie contextuelle de la couche d'entrée) code un état du réseau qui correspond à une prise en compte globale des évolutions passées du réseau. Nous verrons un peu plus bas que ce phénomène permet entre autres d'affiner le comportement d'automates d'états finis associé à ce genre de modèle.

Apprentissage dans les réseaux récurrents à couches

Dans ce qui précède, les connexions récurrentes sont supposées figées lors de l'apprentissage. Ceci permet d'utiliser l'algorithme de rétro-propagation de l'erreur sans adaptation au cas particulier de ces réseaux récurrents. Toutefois, dans le cas général des réseaux récurrents (pas forcément organisés en couche) où les poids ne sont pas figés, nous pouvons étendre l'algorithme de rétro-propagation. Nous ne donnons pas ici cette extension qui peut être consultée dans [Gas, 1994].

Propriétés des réseaux récurrents à couches

Schématiquement, un réseau récurrent se comporte comme un automate d'états fini. L'état du système est codé par la configuration d'activités de la partie contextuelle de l'entrée. Une information est lue sur l'autre partie de l'entrée et en fonction de l'état courant le système tombera dans un nouvel état. L'algorithme d'apprentissage est supervisé et utilise la règle de rétro-propagation du gradient. Une application originale de ce genre d'architecture est donnée dans [Servan-Schreiber *et al.*, 1991]. Il s'agit de reconnaître des séquences de lettres de l'alphabet ordonnées selon une grammaire que le réseau devra apprendre. Si les résultats obtenus avec des séquences de lettres montrent bien la capacité des réseaux récurrents à construire des séquences suivant des règles précises, il n'en demeure pas moins que ce genre de problèmes reste assez éloigné de problèmes plus stochastiques comme la parole.

3.3.5 Les synapses FIR

Une alternative pour traiter ces problèmes est la technique des réseaux à synapses FIR (Finite Impulse Response)[Wan, 1994]. Fondamentalement, une synapse FIR est assez proche du modèle du neurone du TDNN qui intègre l'information sur une fenêtre temporelle. Dans le cas du neurone FIR, chaque synapse bénéficie d'un traitement temporel spécifique sur l'entrée qui lui parvient. Le dessin de droite de la figure 3.12 montre, pour une fenêtre de 4 pas de temps, le type de traitement effectué. Il s'agit d'un filtre linéaire [Gershenfeld et Weigend, 1994]. Dans [Wan, 1994], E. Wan montre qu'un réseau multi-couches de neurones FIR peut se transformer en un simple MLP d'une combinatoire largement plus élevée.

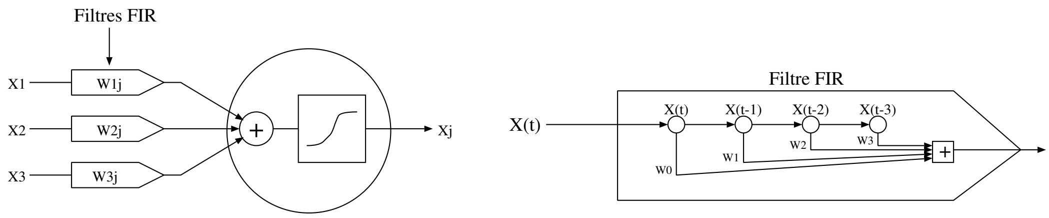


FIG. 3.12 – *Le modèle du neurone FIR. A droite, détail d'une synapse FIR. (D'après [Wan, 1994])*

Ce type de réseau est efficace pour la prédiction de séries temporelles telles que celle décrites dans [Weigend et Gershenfeld, 1994].

3.3.6 Sommateurs à fuites et cartes de Kohonen temporelles

Sommateur à fuites

Sur la base d'observations neurobiologiques faisant état de réponses neuronales dépendant de la nature temporelle des signaux d'entrée (notamment dans l'hippocampe), Reiss et Taylor dans [Reiss et Taylor, 1991] proposent un modèle de neurone dont la constante de temps importante permet d'intégrer de l'information séquentielle, il s'agit du neurone sommateur à fuites. L'activité du neurone $A(t)$ est définie de la façon suivante :

$$A(t+1) = f(I).I(t) + (1 - f(I)).A(t) \tag{3.12}$$

où $I(t)$ est l'entrée du neurone à l'instant t comprise entre 0 et 1 et où la fonction $f(I)$ est une fonction linéaire définie par :

$$f(I) = d.(1 - I) + a.I \tag{3.13}$$

où a et d sont deux constantes, a détermine la durée de montée de l'activité A à partir de l'information perçue en entrée et b la durée de la décroissance de cette même activité. La sortie finale du neurone est donnée par :

$$out(t+1) = H(A(t) - \frac{1}{2}) \tag{3.14}$$

où H est la fonction de Heaviside.
La figure 3.13 donne les courbes d'activité pour ce modèle de neurone.

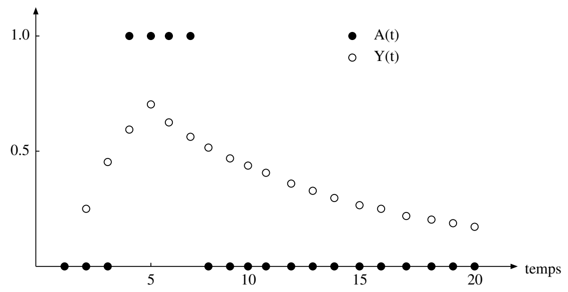


FIG. 3.13 – *Le modèle du neurone sommateur à fuite (leaky integrator neuron). D'après [Reiss et Taylor, 1994])*

Application aux cartes de Kohonen

Reiss et Taylor proposent dans [Reiss et Taylor, 1991] une extension de la carte de Kohonen pour la prise en compte du contexte dans la reconnaissance de phonèmes.
Dans le cas « statique » des cartes de Kohonen, les neurones représentent un prototype présenté en entrée. En utilisant des neurones sommateurs à fuites, un neurone gagne le processus de compétition sur la carte à l'instant t grâce au prototype présenté à l'instant t mais aussi grâce aux prototypes antérieurs. Les applications présentées dans [Reiss et Taylor, 1991] montrent une bonne capacité des cartes à base de neurones intégrateurs à fuites pour le codage et la représentation de séquences, ceci s'obtenant de manière non supervisée. Cependant, les expérimentations présentées utilisent des séquences fixes peu soumises aux distorsions temporelles.

3.4 Modèles à représentation interne et explicite

Les études en neurophysiologie sont une source très importante pour la conception de nouvelles architectures connexionnistes. Dans ce paragraphe, nous tentons de mettre en lumière quelques idées et modèles issus de la neurobiologie et s'intéressant plus particulièrement au problème du traitement temporel de l'information. Dans le chapitre 5, on trouvera quelques éléments de neurophysiologie concernant le traitement auditif. Face à une description neurophysiologique, il faut pouvoir extraire des éléments pertinents pour une modélisation opérationnelle. Ceci constitue une difficulté importante dans la mesure où les descriptions neurologiques sont rarement simplistes. Ceci ne veut en aucun cas signifier que les modélisateurs conçoivent des modèles simplistes, cela veut simplement dire qu'il doit exister une bonne synergie entre modélisateurs et neurophysiologistes. Dans la suite, nous présentons donc quelques modèles neurophysiologistes avec l'esprit du modélisateur. Nous ne prétendons pas être très précis quant au fonctionnement réel de ces systèmes biologiques, encore moins exhaustif. Ceci car en premier lieu, nous ne sommes pas neurophysiologiste et en second lieu, nous voulons extraire la philosophie de ces modèles.

3.4.1 La colonne de mémorisation à court terme

Dans [Ans, 1990], B. Ans propose un modèle original de stockage et de rappel de séquences temporelles. L'élément de base de ce modèle est la colonne de mémorisation à court terme. Celle-ci est une structure ressemblant à une pile de groupes de neurones où les différents événements viennent s'empiler pour former la séquence, ou en quelque sorte une mémoire de l'histoire récente.

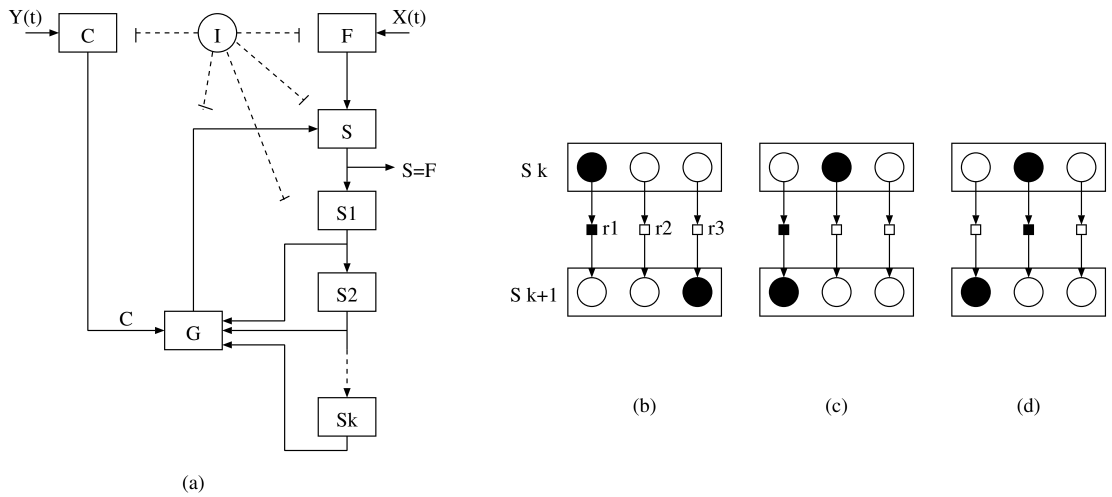


FIG. 3.14 – *Architecture du modèle de la colonne de mémorisation à court terme (d'après [Ans, 1990])*

La figure 3.14 représente l'architecture utilisée pour l'apprentissage des séquences. Contrairement à bon nombre de modèles, le rappel ne s'effectue pas par un extrait de la séquence mais par un *item* dit contextuel. Le flot d'information $X(t)$ qui arrive en entrée du module F et composant la séquence est échantillonné et digitalisé grâce à ce module. Le mécanisme neuronal mis en jeu dans ce module consiste à utiliser des petits groupes de trois neurones. Il existe un principe de compétition locale à ces groupes. A partir d'un item d'information $X(t)$, on obtient un vecteur d'information binaire calculé par F : $\{(010)(001)...(100)(010)\}$. Sur la figure, les flèches verticales représentent des connexions spécifiques, une unité source connecte une seule unité cible. Les flèches horizontales représentent des connexions réparties (chaque unité d'une couche source connecte toutes les unités de la couche cible). La couche S est une recopie de la couche F mais peut aussi être activée par la couche G (dont le principe de fonctionnement est explicité plus bas). Les activités entre la couche S_k et S_{k+1} sont propagées telles quelles avec un retard entre chaque couche (fig. 3.14 (b)(c)(d)). L'information s'empile en quelque sorte et le passage des activités d'une couche à une autre est déclenché par le module I qui est en fait l'horloge du système, il détermine en outre la fréquence d'échantillonnage du flot d'entrée X. Les poids des connexions de C vers G, des S_k vers G et de G vers C sont adaptables et permettent, via le module G, de rappeler une séquence grâce à l'item Y donné en entrée de C. Ce dernier module fonctionne de la même manière que le module F, il digitalise donc l'information. Celle-ci n'est pas échantillonnée puisque l'item est constant pendant toute la durée de passage de la séquence. Sans rentrer dans les détails, le module G apprend à associer l'information qu'il collecte en entrée, c'est à dire les coordonnées de l'item (Y) ainsi que les activités issues des modules S_k. En phase de rappel, la présence de l'item active le module S qui va lui même amorcer une descente d'information dans la colonne. Celle-ci va à son tour ré-activer G qui va lui même donner une nouvelle information sur S, injectant l'information suivante dans la colonne. Par ce mécanisme, la séquence « s'auto-rappelle ».

Dans sa version originale, le système décrit ne traite que des séquence de même longueur (taille fixe de la colonne de mémorisation à court terme). Cet inconvénient a été effacé dans une nouvelle version récemment développée [Ans *et al.*, 1994]. La colonne y est remplacée par un réseau récurrent permettant un traitement de séquences de longueurs variables. De ce fait, le mode de représentation du temps se réoriente vers une représentation implicite comme décrite plus haut.

3.4.2 Modèle de Wang et Arbib

Wang et Arbib [Wang et Arbib, 1990] proposent un modèle de neurone original afin de pouvoir apprendre des séquences de symboles. Le principe d'activation de ce neurone se rapproche de celui du neurone sommateur à fuite (cf. paragraphe précédent) afin d'obtenir une activité à valeurs réelles décroissantes et non binaire. Pour cela, le neurone se décompose en deux neurones interconnectés entre eux (neurones duaux) pour jouer le rôle d'oscillateurs. La figure 3.15 montre deux neurones duaux entretenant une activité qui décroît au cours du temps.

Ce mécanisme d'activation est mis à profit pour apprendre des séquences de symboles que les auteurs classent suivant leur complexité. Une séquence simple ne contient pas de sous-séquence contrairement aux séquences complexes. Par exemple, $A-B-C-D-E$ est une séquence simple alors que $C-A-B-D-A-B-E$ est une séquence complexe parce qu'elle contient une sous-séquence $A-B$. Ceci implique des contextes (sous-séquences) de longueur variable pour un symbole donné. Par exemple, les contextes de longueur 2 des symboles D et E ne permettent pas de les distinguer, il faut nécessairement des contextes de longueur trois (on dit que D et E sont de degré trois). Le modèle proposé par Wang et Arbib permet de traiter ce problème de sous-séquences de longueurs variables. Le réseau employé contient des neurones duaux et il y a inter-connexion totale entre les neurones N_1 de chaque oscillateur. A partir d'un apprentissage de type Hebbien, le potentiel des neurones suit une courbe croissante jusqu'à saturation. A la fin de l'apprentissage, on mémorise cette saturation par un seuil qui, lorsqu'il sera atteint en phase de reconnaissance,

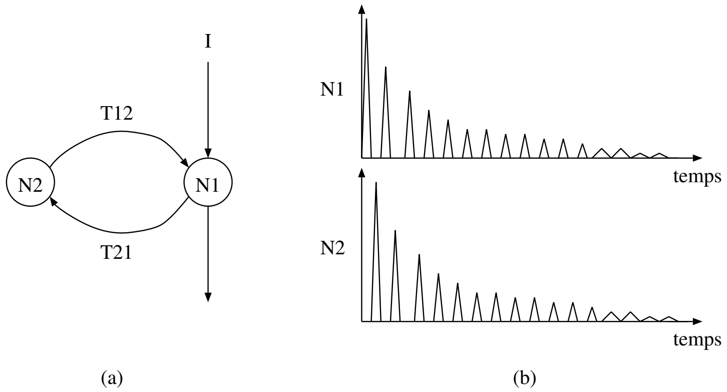


FIG. 3.15 – *Neurones duaux (a) et mémorisation à court terme de l'activité (b) (d'après [Wang et Arbib, 1990])*

caractérisera une séquence particulière.

Il semble, d'après les expériences énoncées dans [Wang et Arbib, 1990] que le réseau apprenne rigoureusement les séquences de symboles. La suppression ou l'ajout d'un seul symbole dans une séquence n'aboutit pas à la reconnaissance. Le système ne semble donc pas tolérant aux distorsions (il n'avait *a priori* pas été formulé pour cela).

3.4.3 Apprentissage de séquences par sélection et modèle de la triade synaptique

Sur la base d'observations biologiques S. Dehaene, JP. Changeux et JP. Nadal [Dehaene *et al.*, 1987] proposent un modèle d'apprentissage de séquences temporelles par sélection. L'étude de l'acquisition du chant chez les oiseaux (en particulier *Melospiza georgiana* et *Melospiza melodia*) montre que celle-ci s'effectue en deux phases : une phase sensorielle où l'oiseau écoute et mémorise le chant des adultes suivie d'une phase sensori-motrice ou d'imitation. Lors de cette dernière phase, il y a production et mémorisation d'un nombre élevé de sous-chaînes sonores. Ce nombre est quatre à cinq fois plus élevé que chez l'adulte. Ainsi, une étape de sélection des séquences de chant utiles est opérée pour aboutir finalement au répertoire de l'adulte.
Le réseau proposé s'appuie sur ces faits. Afin de traiter l'aspect séquentiel de l'information, il met en jeu la notion de triade synaptique. Celle-ci est un modèle à trois neurones où l'efficacité synaptique entre deux des neurones est modulée par l'intervention du troisième (voir figure 3.16).

Pour pouvoir activer le neurone B, la synapse A-B doit être excitatrice et rendue efficace par une activité prolongée (d'une durée d'environ 0.1 seconde) du neurone modulateur C. Une longue période de silence du neurone C diminue grandement l'efficacité synaptique. Ainsi, pour que le neurone B soit actif, il faut que le neurone A le soit précédemment et que la synapse ait été rendue excitatrice, c'est à dire que l'activité du neurone C précède immédiatement l'activité du neurone A avec une durée d'activation au moins égale à 0.1 seconde. La triade définit de cette manière la séquence *A-C-B*.

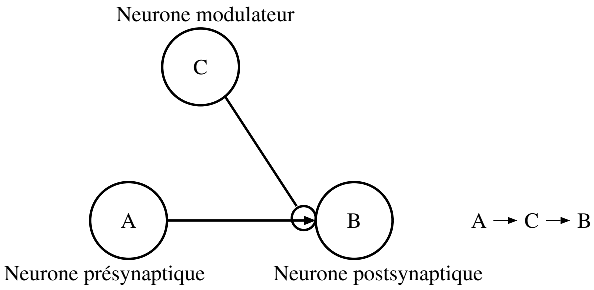


FIG. 3.16 – *Modèle de la triade synaptique*

Le réseau n'est pas un ensemble de neurones interconnectés entre eux mais un ensemble de groupes de neurones (environ une centaine) interconnectés (fig. 3.17).

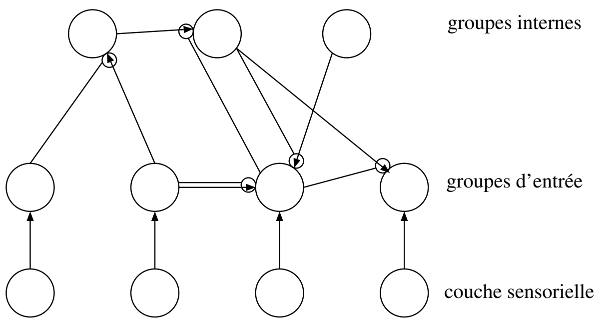


FIG. 3.17 – *Réseau de groupes neuronaux (d'après [Dehaene et Al., 1987])*

L'activité d'un groupe est représentée par la fraction de neurones actifs dans ce groupe. Deux types de connectivité sont mises en œuvre : une connectivité statique et une connectivité modulatrice. La première est excitatrice pour les neurones d'un même groupe et inhibitrice entre les neurones de groupes distincts. La seconde s'établit par paquets entre groupes. Ainsi, si un groupe A est relié par connectivité modulable à un groupe B et si le groupe C module cette connexion, cela signifie qu'il s'établit des triades synaptiques entre les neurones de A, de B et de C.
Le principe d'apprentissage utilisé est une règle de Hebb : lorsque une synapse A-B a récemment contribué à l'activation de B, c'est à dire si A et C ont été activés dans cet ordre alors l'efficacité synaptique entre A et B est augmentée. Inversement, si après l'activation de A, B reste silencieux alors l'efficacité synaptique tend vers zéro. La formalisation de l'apprentissage pourra être consultée dans [Dehaene *et al.*, 1987] ou [Gas, 1994].

La couche d'entrée représente les symboles des séquences et un seul des groupes de cette couche ne peut s'activer. Au départ, les groupes de la couche interne sont connectés aléatoirement entre eux et avec les groupes de la couche d'entrée. Cette structure et le mécanisme d'apprentissage hebbien permettent un apprentissage sélectif de séquences à la manière de l'apprentissage du champ chez les oiseaux (attrition sélective).

3.4.4 La propagation guidée

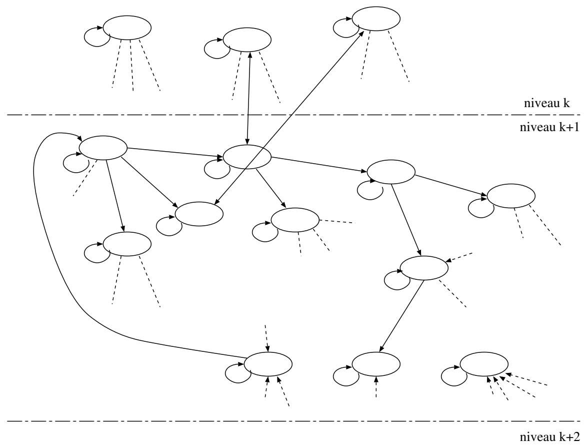


FIG. 3.18 – *Modèle ADAM*

Le modèle ADAM (mémoire adaptative, dynamique et associative) proposé par D. Béroule [Béroule, 1985; Béroule, 1990] consiste en une propagation d'énergie au travers d'un réseau d'unités. Cette propagation d'énergie dépend de stimulations successives produites par des éléments de mémoire associés à des événements se réalisant dans l'environnement du réseau. Une unité u_k pré-active l'unité qui lui succède u_{k+1}. L'arrivée d'une énergie simultanément sur u_{k+1} l'activera définitivement. La pré-activation constitue un phénomène d'hypothèse, en effet, après chaque unité de traitement, plusieurs autres unités peuvent succéder. Un mécanisme d'inhibition latérale permet d'éliminer un certain nombre d'hypothèses après qu'une unité en ait confirmé une (activation). La figure 3.18 montre une représentation schématique et arborescente du modèle.
L'aspect séquentiel est parfaitement représenté dans le modèle et semble particulièrement bien adapté à des niveaux tels que le niveau syntaxique en traitement automatique de la parole. Cependant, pour de plus bas niveaux (par exemple, décodage acoustico-phonétique), il faut une adéquation forte avec l'espace d'entrée et une tolérance assez forte aux distorsions temporelles ce qui n'est pas explicitement pris en compte dans le modèle.
Le modèle a constitué un cadre de travail orignal et constitué l'origine de modèles novateurs [Blanchet, 1992].

3.4.5 La colonne corticale

Nous avons présenté au chapitre 2 la colonne corticale et sa modélisation [Burnod, 1988; Alexandre *et al.*, 1991]. Une des fonctionnalités offertes par ce modèle est sa propension à tenir compte des aspects dynamiques grâce aux liens établis entre colonnes de même aire. Nous avons vu que la force des connexions s'établit statistiquement suivant la décorrélation temporelle des colonnes. Ainsi, l'activation d'une colonne est vue comme l'aboutissement d'un but atteint grâce à la réalisation de plusieurs sous-buts représentés par d'autres colonnes envoyant des connexions afférentes vers la colonne but. De cette façon, une colonne but peut « appeler » des sous-buts afin de s'activer, c'est le principe des arbres d'appel (fig. 3.19).

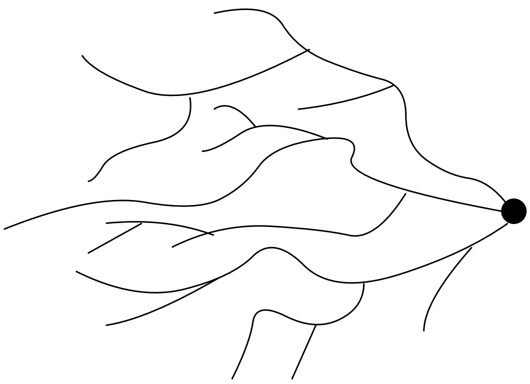


FIG. 3.19 – *Arbre d'appel*

Ce mécanisme permet donc un traitement séquentiel ou temporel de l'information. Dans l'approche que nous proposons, nous souhaitons mettre en oeuvre ce genre de mécanisme et ceci à un niveau bas du traitement où la distorsion temporelle est importante (codage de séquences acoustiques par exemple). Dans le cas des arbres d'appel, la notion de séquencement de l'information est évidente. Cependant, celle de la prise en compte des fluctuations temporelles l'est beaucoup moins. Notre travail consistera en partie à s'intéresser à ce dernier problème.

3.5 Conclusion

La classification que nous venons de présenter a l'intérêt de situer les différents modèles connexionnistes intégrant la notion de temps les uns par rapport aux autres. Elle nous permet également de positionner le modèle sur lequel nous travaillons par rapport aux systèmes existants.
D'un point de vue applicatif, la première classe de modèles (représentation externe) est la plus souvent utilisée, notamment en reconnaissance de la parole (TDNN, perceptron et modèles dérivés). La classe des modèles à représentation interne et implicite voit quelques applications mais à un niveau d'abstraction plus élevé que celui traité généralement par les modèles à représentation externe. Par exemple, un perceptron est capable d'effectuer un codage acoustico-phonétique mais traite difficilement du langage, un réseau récurrent en revanche est beaucoup moins adapté au traitement acoustique mais peut se substituer à un automate d'états finis pour traiter du langage. Enfin, la dernière classe (représentation interne explicite) se caractérise par des modèles originaux, souvent d'inspiration neurobiologique, mais avec des performances qui ne sont pas encore (ou tout juste) à la hauteur de modèles plus classiques.
Cependant, nous pensons que ces derniers modèles peuvent apporter une dimension supplémentaire dans le champ d'investigation des réseaux de neurones, en particulier pour

dépasser le traitement sensoriel. En se voulant plus généraux, ces modèles perdent en performance mais peuvent espérer s'améliorer grâce à leur possibilité d'intégration permettant la prise en compte d'autres types d'information émanant de différents niveaux d'abstraction.

Chapitre 4

TOM, un modèle de traitement de séquences

« *[...] Ce qui fait encore problème, c'est de savoir si oui ou non, en l'absence de l'âme, le temps pourrait être. Car s'il est impossible qu'existe un nombrant quelconque, il est du même coup impossible qu'existe un nombrable, pas plus, évidemment, qu'un nombre : le nombre est, en effet, soit le nombré, soit le nombrable. Mais si la nature n'accorde qu'a l'âme la faculté de nombrer, et plus précisément à cette partie de l'âme qu'est l'intellect, il est impossible que le temps soit sans l'âme, sauf quant à son substrat, au sens où l'on dit que le mouvement peut être sans l'âme. Et l'antériorité et la postériorité se trouvent dans le mouvement, et constituent le temps dans la mesure où elles sont nombrables. [...]* »

Aristote, 335-332 av. J.-C.

4.1 Introduction

Au chapitre 3, nous avons introduit une classification des représentations temporelles dans les systèmes connexionnistes. Si nous pouvons dégager un nombre limité de classes (trois dans notre approche), le nombre de modèles temporels composant ces classes est quant à lui important et il ne nous serait pas possible de tous les décrire ici. Cette multitude de systèmes conçus le plus souvent à partir de réseaux classiques (perceptrons, cartes auto-organisatrices ...) ne va pas dans le sens du développement d'un modèle temporel « universel » adapté aux différents problèmes de nature dynamique pouvant être posés.

Evidemment, un tel modèle paraît bien illusoire aujourd'hui, mais tenter de concevoir une architecture unique pour différents types de problèmes ayant tous une caractéristique dynamique relève d'une démarche volontariste qui n'est pas dénuée de sens : l'architecture fonctionnelle d'une aire particulière du cortex est identique aux autres quelle que soit la tâche à laquelle elle est dédiée [Burnod, 1988]. Bien qu'il soit difficile de concevoir un système connexionniste générique sous l'aspect d'un module (ré)-utilisable quelle que soit la tâche à traiter, nous voulons nous orienter vers cet objectif. Notre approche ne se base pas sur des systèmes classiques mais plutôt sur des modèles d'inspiration neurobiologique. Ce choix comporte un certain nombre d'avantages mais aussi un danger important : le système nerveux est complexe et recèle encore beaucoup de mystères. C'est la raison pour laquelle nous ne souhaitons pas « copier » le cerveau mais plutôt utiliser les idées et certaines hypothèses qui sous-tendent son fonctionnement.

Le modèle que nous proposons dans cette thèse s'inscrit dans l'idée de la conception d'un modèle neuromimétique prenant en compte la composante temporelle et pouvant être utilisé pour des problèmes incluant une composante dynamique. Dans la mesure du possible, nous avons essayé de ne pas orienter l'architecture du modèle en fonction de l'application (la parole dans le cadre de cette thèse, voir chapitre 5). Ainsi, nous nous sommes inspiré du modèle de la colonne corticale [Burnod, 1988] et plus précisément du regroupement de ces colonnes en aires dédiées à des fonctionnalités particulières. L'idée qui sous-tend l'architecture que nous proposons repose sur la notion de propagation d'activité au sein d'un groupe d'unités représentant chacune une information ponctuelle. TOM (Temporal Organization Map) est le nom donné au modèle. Il précise son organisation sous forme de carte. L'architecture proposée se veut donc une contribution au champ de recherche qui ne cesse de se développer concernant la prise en compte du temps dans les systèmes connexionnistes. TOM est une alternative face aux modèles d'inspiration classique. Il constitue une « brique » de base et une plate-forme de recherche pour des problèmes temporels comme la reconnaissance de la parole, la motricité, la vision active, la planification ...

D'un point de vue plus technique, l'architecture de base TOM est une carte neuronale dont les deux principales fonctionnalités sont d'apprendre des indices spatiaux et des séquences temporelles de ces indices et issues du flot d'entrée de la carte. Ses composantes de base, les *super-unités*, se rapprochent plus du modèle de la colonne corticale [Alexandre *et al.*, 1991] que du neurone formel de Mac Culloch et Pitts [MacCulloch et Pitts, 1943]. Ainsi, une super-unité possède plusieurs types d'entrées, elle réagit à des stimuli particuliers et contient, en outre, différentes petites unités dont les interconnexions sur la carte forment des chemins de propagation d'activité qui correspondent à des traits séquentiels particuliers du signal d'entrée. TOM introduit donc différents types de connexions : des liens feed-forward pour coder les stimuli, des liens locaux entre super-unités pour définir une topologie au sein de la carte, et des liens intra-carte pour définir des champs récepteurs temporels. Des extensions du modèle nous conduisent à ajouter d'autres types de liens permettant de gérer par exemple l'interaction entre différentes cartes.

Dans ce chapitre, nous présentons tout d'abord une version de base de l'architecture, c'est-à-dire une version permettant l'apprentissage et la reconnaissance de séquences d'informations. Nous étendons ensuite ce modèle pour la prise en compte des interactions entre différentes cartes. Le rôle de carte associative que nous pouvons attribuer au modèle TOM nous permet d'étudier une version supervisée de l'architecture. Enfin, nous étendons le modèle pour la prise en compte de la tolérance temporelle sélective et adaptative.

4.2 Généralités, propriétés

Comme nous l'avons vu en introduction, la prise en compte du temps a plusieurs objectifs. En particulier, elle a pour but d'améliorer l'apprentissage et la reconnaissance de formes dynamiques, c'est à dire de formes représentées sur un intervalle de temps. Il s'agit donc, à partir du monde réel dans lequel les formes évoluent, d'aboutir à une représentation formelle de ce monde. Cette représentation doit pouvoir conduire à une reconnaissance robuste des formes.

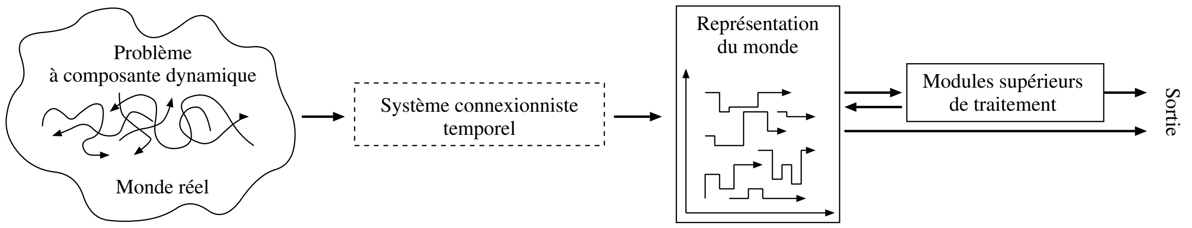


FIG. 4.1 – *Du problème réel à sa représentation*

Une carte TOM, que nous décrivons dans la suite, permet le passage du monde réel représenté sous la forme d'un corpus de formes dynamiques (séquences) à une représentation synthétique robuste permettant un traitement ultérieur. Ce dernier traitement peut lui même être effectué par une carte TOM, il s'agit alors d'un réseau de cartes adapté à une tâche particulière.
En résumé, nous partons d'un espace (qui peut être grand) incluant une dimension temporelle. Cet espace est «projeté» sur un espace plus petit grâce à un système connexionniste temporel, ce dernier espace pouvant lui même être « projeté » dans le cadre d'une tâche donnée et ainsi de suite.

4.2.1 Modèle de base et discrétisation du temps

Architecture

Dans cette partie, nous considérons une carte TOM de façon isolée. L'analogie avec les aires corticales du cerveau s'arrêtera donc à la prise en compte des liens intra-aires. En particulier, nous faisons abstraction des coopérations qui peuvent exister entre les différentes cartes. Ces coopérations, matérialisées sous forme de liens inter-cartes seront étudiées par la suite. Ainsi, une carte TOM est vue ici comme une « brique » de base dont la fonction principale est d'extraire et de représenter des séquences d'information (ou formes dynamiques) représentées par des séquences d'activités.

L'architecture se présente sous la forme d'une assemblée d'unités connexionnistes (dont nous allons décrire la nature ci-dessous) réparties sur un plan. Chaque unité sur ce plan, ou carte, réagit à un stimulus particulier, ceci à l'image des spécificités sensorielles des colonnes corticales que l'on peut trouver, par exemple, dans l'aire primaire du cortex visuel [Hubel et Wiesel, 1977].

Discrétisation du temps

Avant toute chose, il s'agit de définir ce qu'est une forme dynamique. Dans la suite, on emploiera plutôt le terme de séquence rendant plus explicite la notion de dynamique. Par opposition aux formes statiques qui correspondent à des événements ponctuels, c'est à dire une « photographie » à un instant donné, une forme dynamique se produit sur un intervalle de temps. Un événement temporel dans le monde réel est donc mesuré sur un intervalle de temps particulier $[t, t+\Delta t]$. Une multitude d'exemples peut être trouvée pour illustrer la notion d'événement temporel, en voici quelques uns :

- action d'aller saisir un objet de la main initialement au repos (événement sensori-moteur) ;
- action de reconnaissance d'un mouvement dans un champ visuel (événement visuel) ;
- action de reconnaissance d'une phrase, un mot, une syllabe parlée (événement auditif) ;
- action de prononcer une phrase, un mot, une syllabe (événement moteur) ;
- etc.

Tous ces événements temporels sont continus, or nous travaillons avec des réseaux connexionnistes qui sont, par essence, numériques et discrets. Il faut donc discrétiser un événement temporel en une suite d'événements discrets acceptables successivement par le réseau. Pour cela, nous pouvons dégager deux méthodes :

- le flot des données est prétraité pour donner une séquence d'événements ponctuels (fig. 4.2(a)) ;
- le système isole régulièrement une information extraite dans le flot des données (échantillonnage) (fig. 4.2(b)).

Le réseau intègre alors successivement dans le temps les différentes informations composant la séquence. De par sa topologie en forme de carte, le réseau que nous proposons réagit à un stimulus donné à l'instant t par l'activation d'une ou plusieurs de ses unités neuronales. Cette activation concentrée sur une zone géographique particulière de la carte se déplace ou se propage lorsqu'une séquence de stimuli est présentée au réseau (figure 4.3).

Dans ce qui suit, une information à un instant donné t sera définie par un vecteur $\vec{V}(t)$. De cette façon, une forme dynamique sera représentée par une suite de vecteurs : $\{\vec{V}(t), \vec{V}(t+1), ..., \vec{V}(t+\Delta t)\}$

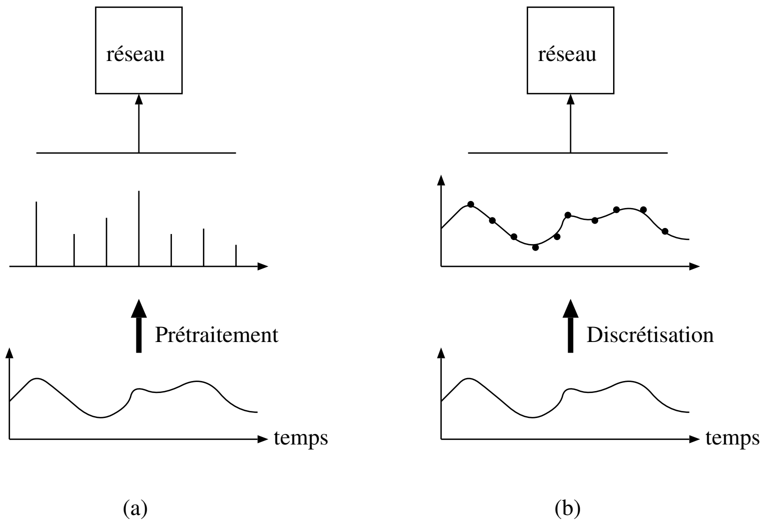


FIG. 4.2 – *Discrétisation du temps*

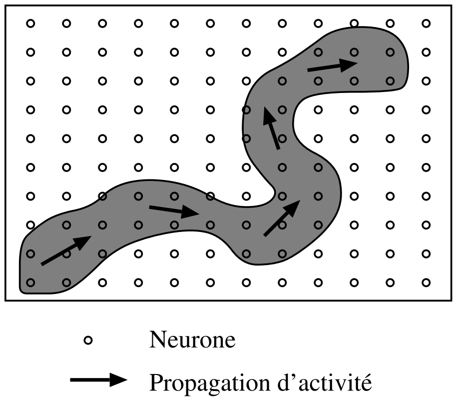


FIG. 4.3 – *Propagation d'activité au sein d'une carte d'unités neuronales*

4.2.2 Propriétés

Nous venons de définir la notion de séquence utilisée dans notre modèle. Voyons maintenant les propriétés que l'on souhaite attacher à ce modèle. Une carte TOM a la charge d'apprendre et de représenter des séquences d'informations contenues dans un corpus de formes dynamiques. Dans un premier temps, nous devons pouvoir retrouver les propriétés généralement attachées aux réseaux de neurones, en particulier pour la partie spatiale de la carte :

- apprentissage ;
- pouvoir de généralisation ;
- tolérance au bruit ;
- représentation distribuée.

Dans un second temps, des propriétés spécifiques au traitement temporel des séquences sont souhaitables, en particulier pour des applications à forte composante stochastique comme dans le cas du traitement de la parole (cf. chapitre 5 et 6). En effet, dans la majeure partie des problèmes réels, les séquences correspondant à une occurrence particulière d'une forme n'ont pas toutes forcément exactement le même aspect. En outre, les longueurs peuvent varier d'une instance à l'autre et le contenu des séquences peut également légèrement différer. C'est pourquoi, si l'on souhaite pouvoir reconnaître une séquence qui n'a pas été apprise par le système, nous devons être «tolérants» spatialement (au niveau des événements qui composent la séquence) et temporellement pour autoriser les distorsions temporelles. Sur ce dernier point, les propriétés que l'on souhaite pour les séquences sont les suivantes :

- tolérance aux insertions d'événements dans les séquences de référence du système ;
- tolérance aux suppressions d'événements dans les séquences de référence du système.

La phase d'apprentissage consistera à apprendre des liens de précédence et de succession afin d'aboutir à une réponse inconditionnelle à un stimulus séquentiel appris.
Toutes ces propriétés ont, entre autres, motivé les choix d'architecture et de fonctionnement décrits dans la suite.

4.3 Architecture

Dans ce qui suit, nous décrivons l'architecture de TOM avant d'en donner les principes de fonctionnement. Comme nous l'avons précisé ci-dessus, les choix concernant le modèle ainsi que les justifications associées à ces choix se feront dans l'optique de l'apprentissage et de la reconnaissance de séquences d'événements avec une robustesse au bruit et à la distorsion temporelle.

Une carte TOM est composée d'un ensemble de *super-unités* que l'on définit ainsi parce qu'elles correspondent à des unités fonctionnelles plus complexes que le neurone utilisé dans les perceptrons ou les cartes auto-organisatrices. Différents types de liens aboutissent sur ces super-unités, chacun d'eux possède une fonctionnalité et une utilité particulière. La figure 4.4 montre le détail d'une super-unité qui possède une fonctionnalité spatiale et une fonctionnalité temporelle.

Du point de vue spatial, les éléments qui assurent cette fonctionnalité sont décrits comme suit :

- des liens feed-forward décrivent le stimulus pour lequel la super-unité réagit. Ces liens sont représentés par un vecteur des poids $\vec{W}$ correspondant au vecteur prototype de la super-unité. Ils sont à rapprocher des liens feed-forward que l'on peut trouver pour les neurones des cartes auto-organisatrices. Il est à noter que la fonctionnalité de ces liens peut être différente dans le cas d'un réseau de cartes (voir plus loin).

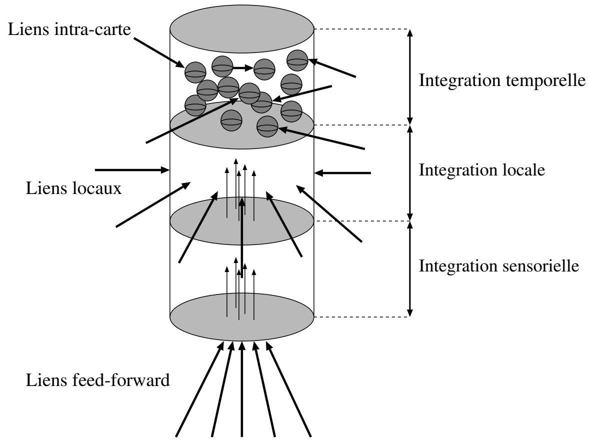


FIG. 4.4 – *L'élément fondamental d'une structure TOM : la super-unité*

– des liens de voisinage s'établissent entre les super-unités. Leur rôle est de définir une topologie spatiale au sein de la carte. En particulier, nous souhaitons conserver la topologie de l'espace des vecteurs d'entrée, ceci afin d'introduire une continuité sur la carte et la notion de flou spatial où, à un vecteur de l'espace d'entrée ne correspond pas une seule et unique super-unité mais une zone de super-unités sur la carte. Dans ce sens, deux super-unités reliées par un lien local réagissent à deux stimuli (vecteurs de l'espace d'entrée) relativement proches.

Enfin, la déclinaison temporelle est assurée par :

– un ensemble de petites *unités* incluses dans la super-unité. Ce sont ces unités qui sont à la base de la notion de séquence dans notre modèle. Chaque unité, pour être déclenchée (activité maximale), a besoin d'une part, que la super-unité dans laquelle elle réside soit activée par un stimulus et d'autre part, que son activité contextuelle soit correcte. L'activité d'une unité ne dépend donc pas seulement de la présence d'un stimulus, il est nécessaire que le passé des activités (ou la séquence des activités) soit en correspondance avec le passé qu'elle aura appris.

– ce passé est matérialisé par des liens entre les unités à l'intérieur des super-unités. Ces liens sont nommés « liens intra-carte » et définissent en quelque sorte la « topologie temporelle » de l'espace d'entrée. Chaque unité possède alors un ensemble de liens intra-carte qui constituent son *champ récepteur temporel*, c'est-à-dire ses contextes possibles d'activation.

Sur la figure 4.4, nous avons schématiquement séparé les composants de la super-unité. Ainsi, nous définissons une zone d'intégration des informations sensorielles, une zone d'intégration des informations locales et une zone d'intégration des informations temporelles ou contextuelles.

Les liens intra-carte définissent les possibilités de propagation d'activité au sein d'une carte. Ces possibilités caractérisent ainsi des formes dynamiques particulières. Il est également à noter que dans l'état actuel de notre modèle, toutes les super-unités possèdent le même champ récepteur. Nous représentons sur la figure 4.5 un exemple de carte TOM.

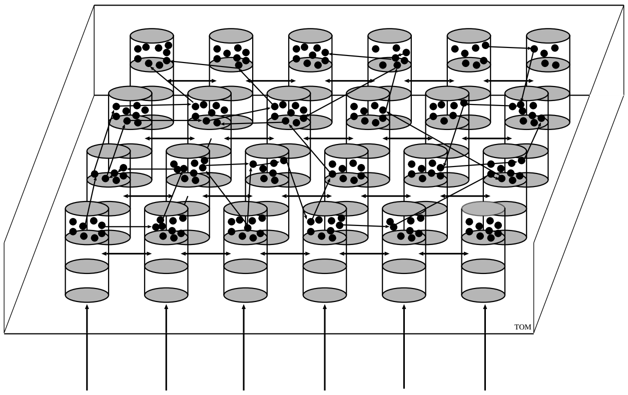


FIG. 4.5 – *Une carte à organisation temporelle (TOM)*

Dans la suite, nous détaillons les principes utilisés pour le codage des stimuli, l'influence locale et la prise en compte des phénomènes temporels.

4.4 Principes de fonctionnement

4.4.1 Le codage des stimuli, activités spatiales

Comme il est décrit sur la figure 4.4, une super-unité intègre de l'information sensorielle. Cette information est codée par l'intermédiaire d'un champ récepteur spatial défini par des liens feed-forward. Ces liens sont équivalents à ceux que l'on peut trouver dans les réseaux auto-organisés et en particulier dans les cartes auto-organisatrices de Kohonen. A ce niveau, nous pouvons donc considérer une super-unité comme un neurone formel (avec un vecteur de poids spatiaux $\vec{W}$) en faisant abstraction de la zone d'intégration temporelle. Les lois de calcul des activités ainsi que de l'apprentissage se ramènent alors aux algorithmes classiques associés aux cartes auto-organisatrices comme nous l'avons décrit au chapitre 2.

Soit $SA_i(t)$, l'activité calculée à l'instant t de la super-unité i. Cette activité est binaire (0 ou 1). Elle est calculée suivant le principe du *winner takes all* ou *winner takes most* décrit au chapitre 2. Ainsi, si la super-unité k est telle que

$$k : \|\vec{W}_k(t) - \vec{V}(t)\| \leq \|\vec{W}_j(t) - \vec{V}(t)\|, \forall j, \tag{4.1}$$

nous avons

$$SA_k(t) = 1, \tag{4.2}$$

et

$$SA_j(t) = 0, \text{pour} j \neq k. \tag{4.3}$$

Les liens de voisinage permettent une inhibition latérale. Nous pouvons ainsi mettre en œuvre le principe de compétition associé aux algorithmes compétitifs.

Après apprentissage, chaque super-unité code un stimulus particulier à un instant donné. Ce stimulus est représenté par le champ récepteur spatial $\vec{W}$ de la super-unité. Pour fixer un peu plus les idées et par analogie avec la neurobiologie, nous pouvons nous placer dans le cas d'un problème de vision. Les champs récepteurs des super-unités correspondent alors à des masques visuels en forme de grille. Dans l'aire primaire du cortex visuel, on trouve par exemple des colonnes corticales avec des masques du type de ceux décrits sur la figure 4.6.

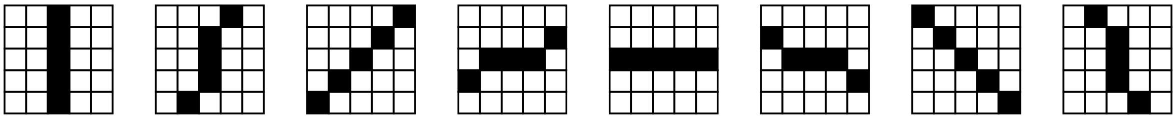


FIG. 4.6 – *Masques visuels déterminés par apprentissage*

Ce sont ces masques que l'on tente de faire apprendre à nos super-unités. Dans l'exemple de la figure 4.6, les masques sont binaires (cases allumées ou éteintes). Cependant, dans nos simulations et en particulier pour l'application à la reconnaissance de la parole, les vecteurs des poids représentant les masques sont à valeurs réelles.

A partir de cette architecture, nous pouvons mettre en œuvre tout type d'algorithme auto-organisé pour calculer les poids feed-forward. En particulier, nous pouvons utiliser l'algorithme de Kohonen qui donne de bons résultats. Les liens locaux, dans ce cas, sont définis *a priori*. Comme nous l'avons vu au chapitre 2, d'autres algorithmes peuvent être mis en œuvre notamment pour apprendre les liens locaux de la même façon que les liens feed-forward sont appris. Nous avons utilisé trois algorithmes différents dans nos simulations : algorithme de Kohonen, algorithme neural-gas, et algorithme growing neural-gas. Les commentaires concernant ces algorithmes seront donnés dans le chapitre des applications.

4.4.2 Influence locale

Une super-unité est influencée par ses voisins géographiques. Ce phénomène est modélisé par les liens locaux attachés à chaque super-unité et définissant le voisinage de celle-ci. Nous avons déjà discuté de l'utilité de ces liens ainsi que de leur mode de création dans le paragraphe précédent. Nous nous focalisons ici sur leur utilité notamment dans l'optique de la reconnaissance de formes dynamiques incluant des variabilités spatiales.

Un exemple de variabilité spatiale est donnée sur la figure 4.7 où à l'instant $t = 3$ le vecteur d'information présenté en entrée ne correspond pas exactement au stimulus attendu par rapport à la séquence de référence. Il est bien évident que pour être appliqué à des situations concrètes, le système doit être capable de tolérer de petites différences entre le vecteur d'entrée et celui qui est le résultat d'une période d'apprentissage. Dans l'exemple de la figure 4.7 la séquence devrait être reconnue. Nous parlons alors de *flou* spatial pour bien spécifier qu'une unité ne réagit pas seulement à un stimulus parfaitement précis mais à une classe de stimuli proches. Les liens locaux permettent une tolérance

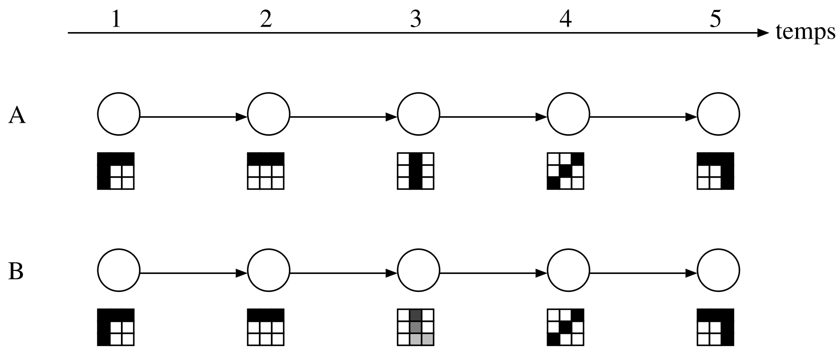


FIG. 4.7 – *Illustration de la variabilité spatiale*

spatiale au niveau des séquences.
Illustrons ceci par un exemple. La figure 4.8 représente deux cartes avec une séquence représentée en grisé sur le dessin. A gauche de la figure, aucune des super-unités ne possède de lien de voisinage. La séquence n'est pas reconnue car des super-unités à l'intérieur de la séquence n'ont pas été activées. Cependant, des super-unités proches de celles qui auraient due être activées l'ont été. En élargissant le champ de la séquence, c'est-à-dire en introduisant des liens de voisinage entre les super-unités topologiquement proches, les super-unités liées localement vont «s'entre-activer» et permettre ainsi la reconnaissance.

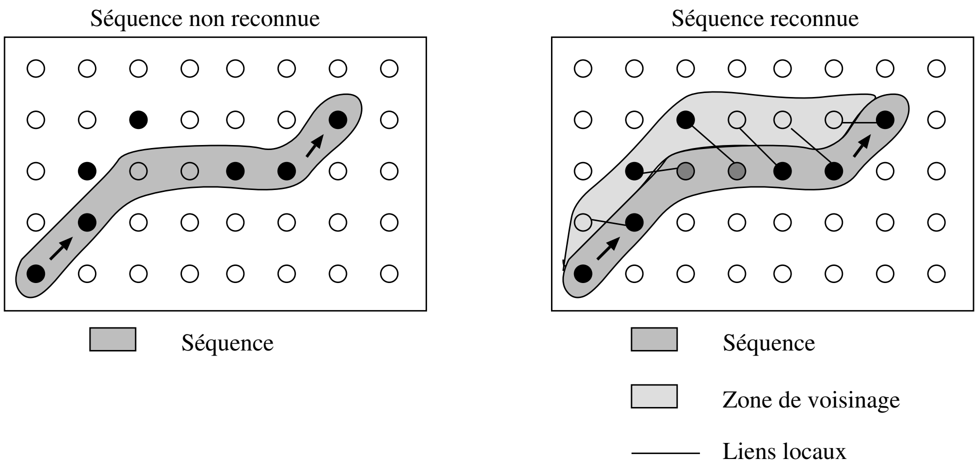


FIG. 4.8 – *Séquence et voisinage. A gauche, pas de voisinage, à droite, avec voisinage*

Dans nos simulations, l'utilisation des liens locaux ne se fera que lors de la phase d'apprentissage des liens feed-forward et lors de la phase de reconnaissance d'une séquence. Dans ce dernier cas, la super-unité gagnante est calculée et celle-ci activera également les super-unités voisines auxquelles elle est localement connectée. Dans la phase d'apprentissage des liens intra-carte, ce principe ne sera pas utilisé afin de ne pas augmenter la combinatoire de création des liens formant les séquences.

4.4.3 La préactivation temporelle, activité des unités

Nous nous intéressons maintenant aux unités des super-unités. Nous avons vu que ces unités sont chargées de la fonctionnalité temporelle d'une super-unité. Plus particulièrement, nous nous intéressons aux mécanismes d'activation de ces unités, ces dernières étant les organes principaux de la reconnaissance de séquences.

Une unité réagit tout d'abord à une information spatiale, un stimulus que sa super-unité est capable de coder. Alors que la super-unité possède un comportement complètement « aveugle » dans le temps, c'est-à-dire qu'elle réagit à un instant donné à ce qui lui est présenté, l'unité quant à elle, tient compte pour s'activer des informations disponibles aux instants précédents, elle tient compte du passé, c'est-à-dire du contexte temporel d'activation. Matériellement, une unité se spécialise dans un contexte temporel d'activation particulier grâce à un champ récepteur temporel. Ce champ récepteur est en quelque sorte sa mémoire du passé, il est composé de liens intra-carte, c'est à dire des liens provenant d'autres unités de la même carte. La figure 4.9 montre une unité (U) et son champ récepteur temporel composé de 5 liens afférents (numérotées de 1 à 5). A l'instant t, si la super-unité dans laquelle se trouve l'unité U est activée (présence du bon stimulus), celle-ci se déclenchera effectivement si l'une ou plusieurs des unités 1 à 5 étaient activées au temps $t-1$.

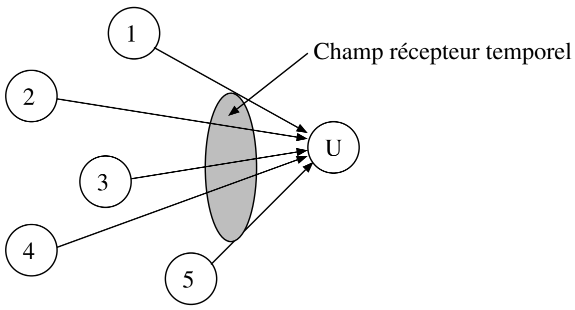


FIG. 4.9 – *Une unité et son champ récepteur temporel*

En effet, une unité ne s'active que lorsque le stimulus de la super-unité dans laquelle elle se trouve est présent, et ceci à la condition qu'une autre unité reliée par le champ récepteur temporel ait été précédemment activée.

Afin de pouvoir tenir compte des activités « précédentes », chaque unité possède une mémoire à court terme (STM). En d'autres termes, lorsqu'une unité est activée, cette activité ne retombe pas à zéro au temps suivant mais décroît en fonction d'un paramètre de décroissance aussi appelé *decay*. Ce principe se retrouve assez souvent dans la littérature, notamment dans les modèles de neurones intégrateurs à fuite. Ainsi la courbe d'activation d'une unité au cours du temps a, par exemple, l'allure donnée sur la figure 4.10.

Comme nous l'avons dit ci-dessus, une forme évoluant dans le temps est caractérisée par une succession d'activités des petites unités sur la carte. Le calcul de ces activités dépend à la fois du stimulus codé par la super-unité abritant les unités et du champ récepteur temporel de l'unité.

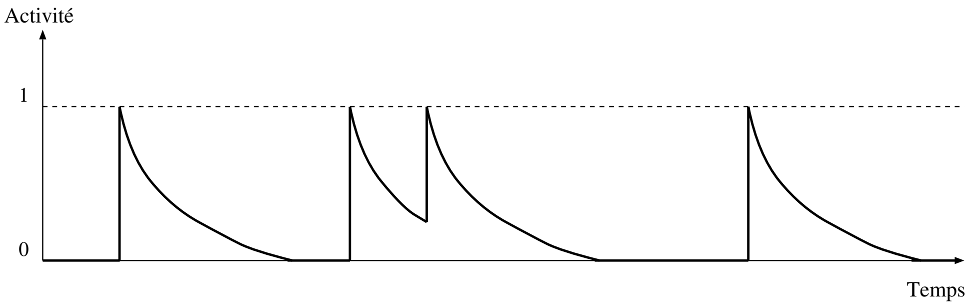


FIG. 4.10 – *Loi d'activation d'une unité*

Bien que les activités des unités soient à valeurs réelles du fait du paramètre de décroissance, le déclenchement de ces activités s'effectue grâce à deux variables d'activations binaires. L'une indique la présence d'un stimulus, l'autre indique l'activation précédente d'une unité dans le champ récepteur temporel d'une unité.
$SA(t)$ est l'activité spatiale d'une super-unité à l'instant t (voir paragraphe 4.4.1). Cette fonction traduit la présence du stimulus et vaut 1 lorsque celui-ci est présent, 0 sinon. Chaque unité à l'intérieur de la super-unité hérite de cette fonction.

Soit $TA(t)$ l'activité temporelle d'une unité au temps t. Si une unité est active dans le champ récepteur temporel de l'unité, $TA(t)$ vaut 1, sinon 0. Le déclenchement de l'activité d'une unité est donné par la formule suivante :

$$\overline{Act(t)} = \frac{1}{\alpha + \beta}(\alpha SA + \beta TA) \tag{4.4}$$

Les coefficients α et β règlent l'influence respective de chaque composante. L'égalité des deux coefficients traduit que le déclenchement d'une unité ne peut s'opérer qu'à la condition de la présence effective de SA et TA, *ie.* que le stimulus soit présent et que le contexte soit correct. Si un des coefficient est plus fort que l'autre, cela peut entraîner un déclenchement sans la présence de l'une des deux composantes.
Examinons maintenant l'activité réelle de l'unité ($Act(t)$) :

$$Act(t) = \begin{cases} 1 & \text{si } \overline{Act(t)} \geq 1 \\ decay.Act(t-1) & \text{sinon} \end{cases} \tag{4.5}$$

Cette équation n'est autre que l'expression mathématique de la courbe donnée figure 4.10. Une unité est déclenchée par la présence d'un stimulus particulier dans une condition temporelle particulière. En dehors de ceci, l'activité de l'unité décroît au cours du temps. La décroissance est contrôlée par le paramètre *decay* qui influence donc la mémoire à court terme de l'unité.

4.4.4 Les séquences d'activités

L'architecture décrite dans ce qui précède permet d'apprendre et d'organiser des séquences d'informations contenues à l'intérieur d'un corpus traduisant un problème dynamique (par exemple, un corpus de séquences acoustiques représentant des mots en traitement de la parole). Chaque unité peut donc être vue comme le maillon d'une chaîne qui ne peut s'activer que si les maillons précédents dans la chaîne ont été activés dans un passé plus ou moins lointain. Chacun des maillons réagit à un événement spatial particulier.

Parce que l'apparition d'un stimulus peut se faire dans diverses conditions, en particulier dans divers contextes temporels, il nous faut pouvoir distinguer les stimuli à l'information spatiale identique mais au passé différent. C'est la raison pour laquelle, à chaque fois qu'un nouveau contexte d'activation est connu pour un stimulus donné, nous créons une nouvelle unité à l'intérieur de la zone d'intégration temporelle de la super-unité.

Le fait de «créer» un certain nombre d'unités à l'intérieur des super-unités nous permet d'éviter les phénomènes de boucles et de confusion de séquences. En effet, pour illustrer ce fait supposons 7 stimuli dénommés A, B, C, D, E, F, et G. Nous supposons disposer de 7 super-unités codant chacune un stimulus particulier. De plus, plus d'une seule unité ne peut être présente dans la zone d'intégration temporelle. Le corpus ne contient que les deux séquences suivantes : ABCD et EBCFG. Le graphe réseau obtenu est dessiné sur la figure 4.11(a).

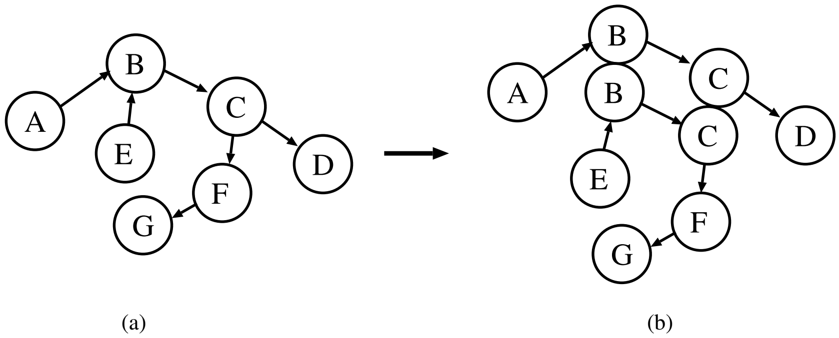


FIG. 4.11 – *Exemple d'unités inter-connectées et duplication d'unités*

Nous voyons que si les séquences ABCD et EBCFG sont bien présentes sur la carte, il y a eu génération de deux autres séquences (ABCFG et EBCD) qui n'auraient pas dû être représentées sur le réseau. Ceci est dû au fait qu'il n'y a pas eu distinction dans la fonctionnalité spatiale de chacune des super-unités. D'où l'introduction de plusieurs unités dont la fonction est de distinguer le contexte d'activation (figure 4.11(b)).

4.4.5 Propagation d'activité à partir d'une forme dynamique

L'algorithme de reconnaissance d'une forme dynamique donnée en entrée d'une carte TOM est donné ci-dessous. Une forme dynamique est une suite de vecteurs $\{\vec{V}(t)\}$.

Pour une forme dynamique à reconnaître V de durée D faire

Pour t de 1 à D faire

- Calcul de l'activité spatiale SA(t) de chaque super-unité
- Activation des super-unités voisines de la super-unité gagnante
- Calcul de l'activité de chaque unité dans les super-unités
- Si les unités possédant le label de V sont actives (act>SEUIL)

  alors la forme est reconnue

fpour

fpour

Nous parlons de reconnaissance dans la mesure où les séquences apprises sur la carte ont été étiquetées. En réalité, l'algorithme décrit la méthode d'activation des unités conduisant à une propagation d'activité. Ainsi, l'algorithme introduit la notion de label. En effet, d'un point de vue pratique il faut pouvoir décider si une séquence est reconnue pour représenter telle ou telle forme apprise. C'est pourquoi, chaque unité en fin de séquence possède une étiquette qui indique que si cette unité est activée elle reconnaîtra la forme associée à l'étiquette de l'unité. Informatiquement, le label est une variable associée à l'unité ou un lien vers une autre structure (ou même une unité) qui se trouve déclenchée quand l'unité qu'elle connecte est activée. Nous verrons plus loin dans ce chapitre une exploitation possible de cette propagation d'activité.

4.5 Apprentissage

Nous nous intéressons maintenant aux mécanismes d'apprentissage des différents types de liens inclus dans la carte. L'apprentissage s'effectue séquentiellement en deux étapes. Il y a d'abord construction des champs récepteurs spatiaux associés au problème à étudier. Puis, il y a construction des liens intra-carte pour l'apprentissage des combinaisons temporelles (séquences).

4.5.1 Apprentissage des liens feed-forward

Nous avons déjà discuté plus haut des liens feed-forward. En particulier, nous avions précisé que nous pouvions considérer la carte TOM comme une simple carte auto-organisée en faisant abstraction des unités et en considérant les super-unités comme de simples neurones formels avec des vecteurs de poids $\overrightarrow{W}$. Nous reprenons ces considérations dans la partie de l'apprentissage des liens feed-forward.
Les algorithmes mis en œuvre sont ceux décrits pour les réseaux auto-organisés au chapitre 2. Il s'agit donc d'adapter la super-unité gagnante ainsi que son voisinage décrit par

les liens locaux.

L'algorithme le plus simple qui vient à l'esprit est celui des cartes auto-organisatrices de Kohonen [Kohonen *et al.*, 1984; Kohonen, 1989]. Dans cette version, les liens de voisinage sont définis *a priori*. Nous pouvons cependant envisager d'apprendre ces liens pour mieux apprendre la topologie spatiale de l'espace d'entrée. C'est ce à quoi s'intéresse le réseau neural-gas [Martinetz et Schulten, 1991; Martinetz *et al.*, 1993]. Pour aller encore plus loin dans la finesse de la classification, nous pouvons adjoindre des super-unités lorsque la classification se fait mal à certains endroits de l'espace. C'est le propos de l'algorithme growing neural-gas [Fritzke, 1995].

Nous ne détaillons pas ici ces algorithmes puisqu'ils ont été présentés au chapitre 2.

4.5.2 Apprentissage des liens intra-carte

Il reste à déterminer les liens intra-carte, fondamentaux pour la modélisation temporelle des formes à étudier. Dans la forme de base de l'algorithme, les liens sont binaires (existent ou n'existent pas). Il s'agit donc d'apprendre pour chaque unité son champ récepteur temporel. L'algorithme est donné ci-dessous.

<u>Pour chaque forme dynamique</u> V <u>de durée</u> D <u>faire</u>
    <u>Pour</u> t <u>de</u> 1 <u>à</u> D <u>faire</u>

(Traitement du vecteur de V au temps t : V(t))

- Calcul de l'activité spatiale SA(t) de chaque super-unité

- Calcul de l'activité de chaque unité dans les super-unités

- <u>Si</u> il existe une unité U sans champ récepteur temporel dans SU
et telle que act(U)=1.0
    <u>alors</u>
    - <u>Pour toutes</u> les unités $U' \neq U$ de la carte <u>faire</u>
    <u>Si</u> act(U')>$thres\_act$
        <u>alors</u>
        ajouter U' dans le champ récepteur temporel de U
    <u>fsi</u>
<u>fpour</u>

    <u>sinon</u>
    - Créer une unité U dans SU
    - act(U) $\leftarrow$ 1.0
    - <u>Pour toutes</u> les unités $U' \neq U$ de la carte <u>faire</u>
    <u>Si</u> act(U')>$thres\_act$
        <u>alors</u>
        ajouter U' dans le champ récepteur temporel de U

fsi
fpour
fsi
• Si t=D (fin de la séquence)
alors mettre un label à toutes les unités U telles que act(U)>SEUIL
fpour
fpour

Comme pour l'algorithme de reconnaissance, nous utilisons des labels positionnés en fin de séquence. Ceci n'a essentiellement d'intérêt que pour un aspect pratique et n'influence en rien l'apprentissage des séquences.

Quelle est la philosophie générale de cet algorithme? Pour répondre à cette question, nous allons regarder l'évolution d'une carte TOM contenant 4 super-unités. Pour simplifier le problème et se concentrer sur l'aspect temporel (les unités et leurs liens intra-carte), nous attachons à chaque super-unité une étiquette, une lettre de l'alphabet. Nous souhaitons apprendre une séquence de lettres, par exemple, BCDAB. Au départ, la carte contient 4 super-unités, chacune d'elles étant dédiée à un stimulus particulier (une lettre). La zone d'intégration temporelle ne contient pas d'unités. La figure 4.12 au temps $t = 0$ (coin supérieur gauche) donne l'allure de la carte.

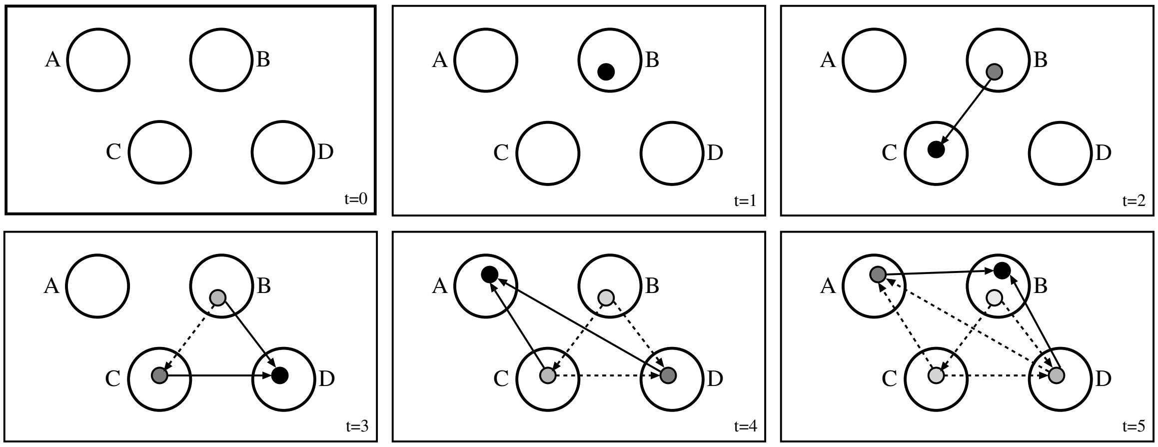


FIG. 4.12 – *Apprentissage d'une séquence, évolution sur la carte*

Le premier stimulus de la séquence ($t = 1$) est représenté par le label B. Comme la super-unité B ne contient pas d'unité dans sa zone d'intégration temporelle, nous en créons une. Son activité est alors très forte (égale à 1). Puis, au temps $t = 2$, le stimulus C est présenté. La super-unité C ne contenant pas d'unité active, nous en créons une. Afin de mettre à jour son champ récepteur temporel, nous examinons toutes les autres unités ayant une activité supérieure à un certain seuil. Graphiquement, nous voyons que l'unité précédemment créée dans la super-unité B est encore assez active. Nous ajoutons donc celle-ci dans le champ récepteur temporel de l'unité que nous venons de placer. A $t = 3$, D arrive. Par le même principe que précédemment, nous créons une unité dans D. Puis nous mettons à jour le champ récepteur. Comme illustré sur la figure, deux unités sont ajoutées dans le champ récepteur temporel. En fait, ceci dépend de deux paramètres (*decay* et *thres_act*) qui influencent la composition du champ récepteur. Si le seuil d'activation est bas, le champ récepteur ira chercher des unités assez faiblement active. A $t = 4$, le stimulus A est présenté, une unité est créée avec son champ récepteur. A la fin de la séquence ($t = 5$), le stimulus B est à nouveau présent. Si ce stimulus est « spatialement » le même que celui présenté à $t = 1$, il n'est « spatio-temporellement » pas le même. C'est la raison pour laquelle une nouvelle unité est créée dans la zone d'intégration temporelle de la super-unité B. Dans la suite, nous discutons de l'intérêt d'une telle construction en particulier pour des problèmes où les séquences sont extrêmement variables dans le temps et dans l'espace, comme c'est le cas dans la parole.

4.6 Comportement

Comme nous l'avons vu plus haut, la phase d'apprentissage s'effectue en deux temps. Une première phase, spatiale, consiste à «fixer» les poids feed-forward afin de prédisposer chaque super-unité à un stimulus particulier. La seconde phase, temporelle, créé les liens intra-carte permettant de définir les champs récepteurs temporels de chaque unité dans la zone d'intégration temporelle.
La durée de la phase d'adaptation des liens feed-forward dépend du nombre de super-unités utilisées sur la carte, elle dépend également de la dimension de l'espace d'entrée (et donc de la dimension du vecteur des poids feed-forward). En fonction de la taille du corpus d'apprentissage et du type du problème à traiter, cette durée peut être relativement longue. En revanche, la mise en place des liens intra-carte est plus rapide car elle ne nécessite pas de répétition du passage du corpus. En effet, la carte extrait des prototypes de séquences qui vont servir de références pour la tâche de reconnaissance. Ces prototypes ont la propriété d'être résistants au bruit et à la distorsion temporelle. Par bruit, on entend le fait:

- qu'un stimulus codé dans une séquence soit transformé en un stimulus légèrement ou complètement différent dans la séquence de test (variabilité) ;
- qu'un stimulus soit absent dans la séquence (élision) ;
- qu'un stimulus soit inséré dans la séquence (insertion).

La distorsion temporelle fait également référence aux deux derniers points. Sur un exemple, nous allons examiner le comportement du système face à la reconnaissance d'une séquence bruitée.
Sur la figure 4.13, un exemple artificiel montre une séquence apprise par une carte TOM. La séquence est isolée pour une meilleure compréhension. La forme dynamique ayant généré ce graphe est $ABCDE$. Nous remarquons que chaque unité dispose de deux unités dans son champ récepteur temporel. Le nombre de connexions générées à un instant donné dépend du paramètre *thres_act* (voir l'algorithme d'apprentissage temporel). Par induction, ce paramètre contrôle le phénomène d'élision d'un événement. Ainsi, la séquence $ACDE$ est facilement reconnue car C réagit en contexte B mais aussi en contexte A. Le choix de ce paramètre est donc très important. En particulier, il faut veiller à ce qu'il ne soit ni trop fort (risque de séquences trop lâches), ni trop faible (séquences n'admettant

pas d'élision). Le phénomène d'insertion est contrôlé par la décroissance de l'activité d'une unité. Ainsi, le paramètre *decay* définit en quelque sorte une mémoire à plus ou moins court terme. Lorsqu'une unité s'active (à 1), elle peut être intégrée dans un champ récepteur. Cette utilisation est conditionnée par un autre paramètre, ACT. Si l'activité est supérieure à ce seuil, l'unité peut être choisie pour un champ récepteur temporel. Si ACT est fort (proche de 1) peu ou pas d'insertion sera tolérée. Sur notre exemple, $ABACDE$ est reconnue. En fonction des paramètres *decay* et *thres_ act* $ABAACDE$ pourrait l'être encore.
Comme nous l'avons vu plus haut, la super-unité code un stimulus particulier grâce à ces liens feed-forward. Par convolution, c'est l'unité la plus proche qui réagit. En conséquence, un stimulus un peu différent de celui appris peut déclencher la super-unité. Ainsi, $AB'CDE$ est reconnue (B' n'étant spatialement pas trop éloigné de B).

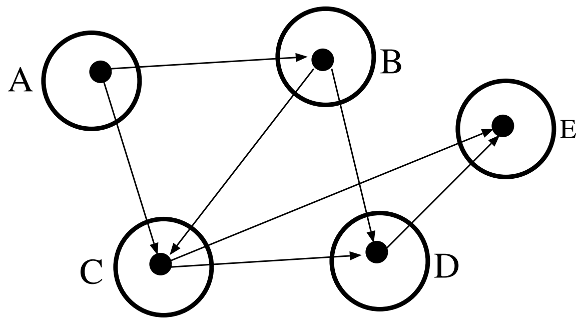


FIG. 4.13 – *Une exemple de séquence extraite par TOM*

La combinatoire de la carte (*i.e.* le nombre d'unités créées ainsi que le nombre de liens intra-carte générés) dépend de la taille du problème traité mais ne croît pas linéairement en fonction du temps. En effet, au début du corpus, il y a création d'un nombre important de séquences (unités et liens intra-carte) mais des occurrences différentes d'une même forme temporelle ne génèrent pas nécessairement deux séquences différentes. Si les deux formes dynamiques sont proches (au sens spatial et temporel), il y aura plutôt affinage d'une séquence déjà existante. Ainsi, la répétition d'une forme dont des occurrences sont déjà codées par la carte n'aura pas d'influence sur la combinatoire de cette carte.

Un des intérêts d'une telle représentation du temps par rapport à des représentations telles que les représentations externes ou internes-implicites est que l'on peut envisager une explication de la réponse donnée par le système. Dans leur large majorité, les réseaux neuronaux sont vu comme des boîtes noires pour lesquelles nous fournissons un vecteur d'entrée et nous attendons une sortie sans vraiment pouvoir expliquer comment elle a pu être obtenue. Avec une architecture du type de celle présentée ci-dessus, nous pouvons entrevoir un début d'explication quant à la façon dont le résultat a été obtenu.

4.7 Remarques et critiques sur le modèle de base

Nous venons de présenter la base du modèle sur lequel nous travaillons dans le cadre du développement d'un système connexionniste intégrant une composante temporelle. La carte TOM permet donc l'apprentissage (et le codage) de séquences d'événements. Une des propriétés requises pour ce modèle est la tolérance au bruit. Ce bruit se caractérise sous deux formes : spatiale et temporelle. La distorsion temporelle des séquences est un phénomène que l'on retrouve régulièrement dans des problèmes du monde réel notamment en traitement automatique de la parole. Cependant, à d'autres niveaux, cette tolérance aux distorsions temporelles peut ne pas avoir lieu. Lorsqu'il s'agit, par exemple, de se remémorer une séquence de chiffres (un numéro de téléphone par exemple) nous ne pouvons pas tolérer d'élision ou d'insertion d'un événement (un chiffre) dans la séquence. Les paramètres de tolérance temporelle (décroissance d'activité, *decay*, et seuil de branchement des liens intra-carte *thres_act*) dépendent donc fortement de l'application que l'on souhaite traiter. Ainsi, pour une application à la reconnaissance de la parole (chap. 6) il faudra être relativement tolérant en distorsion temporelle alors que pour une application de mémorisation de séquences de chiffres il ne faudra pas être tolérant.

Prise de façon isolée, la carte TOM permet une propagation d'activité uniquement (avec certaines propriétés dont nous avons déjà parlé). Il faut pouvoir l'exploiter pour une application particulière. Dans sa version de base telle que celle décrite ci-dessus, l'architecture de la carte permet la représentation de séquences apprises à partir d'un corpus de formes dynamiques. Il ne s'agit pas d'une mémorisation systématique de séquences présentées en entrée du réseau mais plutôt de la construction de prototypes de séquences qui réagissent pour des séquences proches mais pas forcément identiques.

Les cartes auto-organisatrices (telles que les cartes de Kohonen) partagent l'espace des données d'entrée en classes, chaque neurone sur la carte étant le centre d'une classe particulière. De plus, et c'est peut-être là le point le plus intéressant, ces cartes conservent les relations topologiques qui existent dans l'espace d'entrée (deux neurones proches sur la carte au sens de la métrique usuelle correspondent à deux vecteurs proches au sens d'une distance euclidienne). Dans le cas de notre problème, il serait sans doute intéressant d'introduire une notion de distance entre les séquences pour pouvoir les regrouper en classes. Ceci permettrait de définir une topologie dans l'espace des formes dynamiques. L'intérêt pour une application pratique est clair : à partir d'une séquence donnée à reconnaître, on peut alors retrouver à quelle classe la séquence appartient et activer non pas une seule séquence mais un groupement (une bulle d'activation) à la manière des cartes de Kohonen.

Dans la version de base de TOM, une séquence est détectée lorsqu'une propagation d'activité atteint la fin d'une chaîne d'unités (séquence). Il s'agit d'un déclenchement binaire et il n'y a donc pas de notion de proximité entre les séquences. Au paragraphe suivant nous proposons une amélioration de la carte permettant la classification de séquences et la définition d'une topologie.

Enfin, jusqu'à présent, tout ce que nous avons proposé relève du domaine non supervisé. Nous n'avons pas considéré l'existence possible d'une influence « externe » sur une carte. Cette influence peut être exercée soit par un « professeur » indiquant quelles doivent être les réponses correctes du système, soit par une autre carte de niveau supérieur. La coopération entre cartes est ainsi une évolution naturelle pour des applications réalistes de notre système. Nous traitons de ce dernier point au paragraphe 4.9.

4.8 Classification de séquences, conservation de la topologie

4.8.1 Détection de séquences

Lorsque l'on parle de classification ou de quantification vectorielle, on suppose implicitement un ensemble de vecteurs constituant un espace vectoriel de dimension fixe et finie. Dans le cas de nos séquences, nous ne pouvons plus faire cette supposition puisque nous ne pouvons pas représenter toutes les séquences sur un vecteur de dimension fixe au vu de la variabilité des longueurs des séquences.
Cependant, nous pouvons rendre une super-unité sensible à une séquence en «branchant» son champ récepteur spatial sur une suite d'unités connectées (une séquence) comme représenté sur la figure 4.14. Le principe de fonctionnement d'une super-unité détectrice de séquence au niveau spatial (déclenchement de l'activité spatiale) revient alors à un filtre impulsionnel à réponse finie [Wan, 1994]. L'activité spatiale de la super-unité est alors donnée par

$$SA(t) = \sum_{n=0}^{T} \vec{C}_n.\vec{V}(t-n) \tag{4.6}$$

où $\vec{C}_n$ définit les coefficients du filtre.

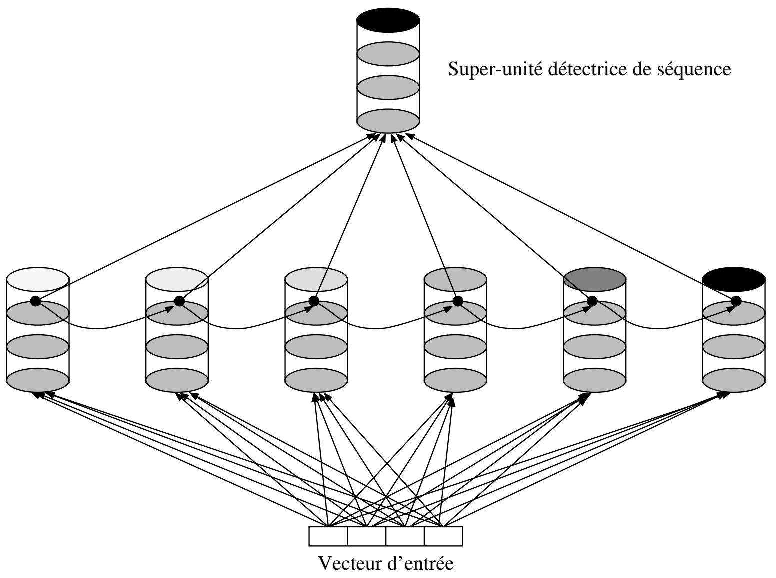


FIG. 4.14 – *Une super-unité avec un champ récepteur spatial sur une séquence*

Afin d'améliorer le codage des séquences et en particulier introduire une métrique permettant une classification des séquences, nous adjoignons à une carte TOM chargée du codage des séquences une seconde carte contenant des super-unités détectrices de séquences. La figure 4.15 montre une première carte à propagation d'activité connectée à une carte détectrice de séquences.

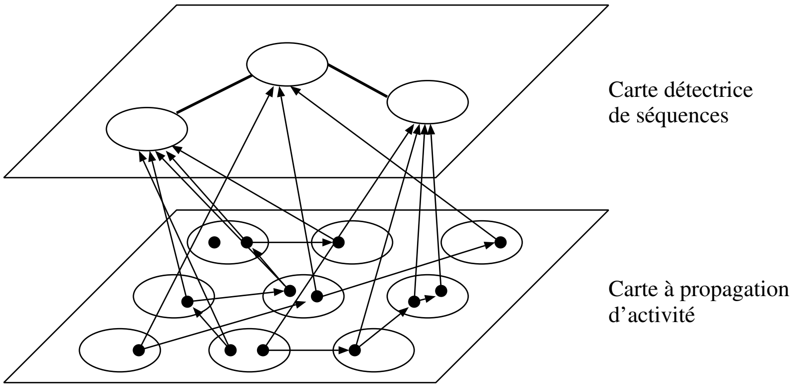


FIG. 4.15 – *Codage et détection de séquence par deux cartes TOM*

Les deux cartes sont exactement du même type (carte TOM). Cependant, la différence de fonctionnalité réside dans le type d'afférence pour la zone d'intégration sensorielle des super-unités. Dans la première carte, celle dédiée à la propagation d'activité due aux séquences d'événements, le champ récepteur sensoriel (les liens feed-forward) se connecte sur un vecteur représentant l'information à un instant t dans la séquence. Dans la seconde carte (détectrice de séquences), le champ récepteur sensoriel se connecte sur les unités qui représentent une séquence complète.

Ce dernier mode de connexion impose un mode de fonctionnement légèrement différent du simple déclenchement par stimulus. En effet, pour un champ récepteur sur un vecteur, l'information est disponible entièrement à l'instant voulu et le calcul de l'activité peut donc être effectué. Dans le cas d'un champ récepteur sur des unités, il est bien évident que nous ne pouvons pas disposer de toute la configuration d'activité de ces unités à un instant donné t. C'est la raison pour laquelle lorsqu'une unité a été activée (bon stimulus et bon contexte temporel), nous conservons la trace de cette activité grâce à une variable de mémorisation.

Pour plus de clarté, donnons un exemple. La figure 4.16 représente une super-unité (sous la forme d'un carré) connectée à des unités (sous la forme de cercles) formant une séquence (par souci de clarté, les super-unités de la séquence n'ont pas été représentées sur ce schéma). Examinons au-cours du temps, l'activation spatiale de la super-unité.
La méthode d'activation spatiale vue plus haut dans le modèle de base utilise une convolution, c'est-à-dire que le vecteur des poids est comparé au vecteur d'entrée par un calcul de distance pour décider de l'activation. Dans le cas d'une connexion sur une chaîne d'unités, on utilise un filtre comme pour les neurones des perceptrons : il s'agit d'une somme pondérée des entrées. Cette somme pondérée n'utilise pas directement les activités des unités si l'on veut pouvoir déterminer l'activité spatiale de la super-unité à tout instant. En effet, si l'on se place à un instant correspondant à la fin de la séquence, l'activité de la première unité, même si elle a été conservée un petit laps de temps grâce à sa mémoire à court terme, est devenue quasi nulle. C'est pourquoi, la somme pondérée utilise, pour son

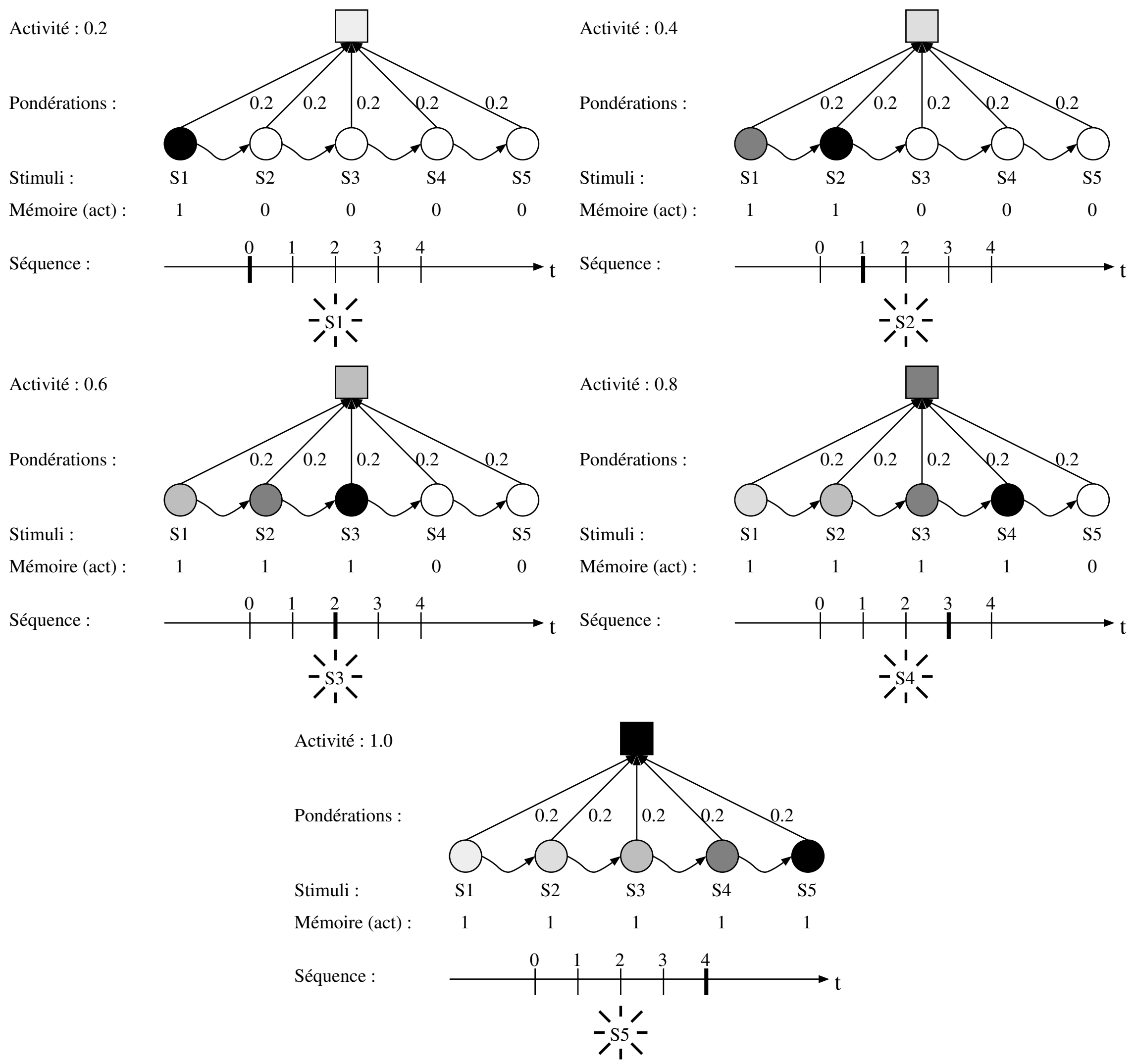


FIG. 4.16 – *Activation d'une super-unité par une séquence*

calcul, une variable qui indique si une unité a été déclenchée lors de la présentation d'une séquence en entrée du réseau. La figure 4.16 montre ainsi l'évolution de l'activité d'une super-unité détectrice de séquence. Cette activité est (re-)calculée à chaque pas de temps. Elle aurait cependant pu être calculée uniquement à la fin de la séquence, le résultat étant alors identique.

D'un point de vue technique et pratique, la mise en place des super-unités détectrices de séquences s'effectue en même temps que l'apprentissage des liens intra-carte dont l'algorithme a été donné plus haut. Nous pouvons plus formellement donner l'algorithme de mise en place de la détection de séquences par deux cartes TOM :

Soient C1 la carte à propagation d'activité et C2 la carte détectrice de séquences :
Pour chaque séquence S faire
    Si S n'active pas de séquence existante sur C1

alors
- Créer une nouvelle suite d'unités dans C1 selon l'algorithme d'apprentissage des liens intra-carte
- Créer une nouvelle super-unité dans C2
- Connecter la super-unité créée aux unités de la nouvelle séquence avec des pondérations à 1

fsi

fpour

- Normaliser les poids de connexions de chaque super-unité

4.8.2 Relations topologiques

Il est bien évident qu'à partir du moment où nous faisons abstraction du type des entrées des super-unités (unités ou vecteurs), nous retrouvons exactement les même fonctionnalités pour la carte détectrice de séquences que celles décrites pour le modèle de base (propagation d'activités). Nous pouvons donc construire des séquences de séquences pour les besoins d'une application particulière. Une super-unité détectrice de séquence est alors considérée comme une simple super-unité codant un stimulus particulier au même titre qu'une information spatiale et ponctuelle.

Dans ce paragraphe, nous nous focalisons sur les liens de voisinage entre super-unités, ceci dans le but d'établir une topologie au niveau des séquences. Dans le paragraphe précédent, nous avons décrit comment il était possible d'utiliser une carte TOM pour détecter des séquences d'événements. La carte contenant les super-unités détectrices n'utilise cependant pas toutes les possibilités offertes par l'architecture TOM. En particulier, elle ne fait appel qu'aux liens feed-forward (qui servent pour son déclenchement) et ignore les liens voisins et intra-carte qui peuvent être mis en œuvre dans ce type d'architecture. La figure 4.17 montre des groupements de séquences de même nature grâce aux liens locaux pouvant s'établir entre les super-unités.

On définit ainsi une carte topographique en accord avec les données neurobiologiques où les aires corticales conservent les relations topologiques des données qui se projettent sur ces aires.

L'algorithme de mise en place des liens locaux entre super-unités détectrices de séquences est similaire au développement des liens de voisinage dans l'algorithme Neural-Gas [Martinetz et Schulten, 1991]. Il s'agit de repérer statistiquement les corrélations d'activité entre les super-unités. L'algorithme est le suivant :

Soit C_{ij} la matrice des connexions locales entre les super-unités (si $C_{ij} = 1$ alors la connexion existe sinon elle n'existe pas). Soit t_{ij} la matrice des âges des connexions.

1. Initialiser les connexions : $C_{ij} = 0, \forall i, j$

2. Choisir une séquence dans la base d'apprentissage

3. Relaxer la carte de propagation d'activité (passage de toute la séquence)

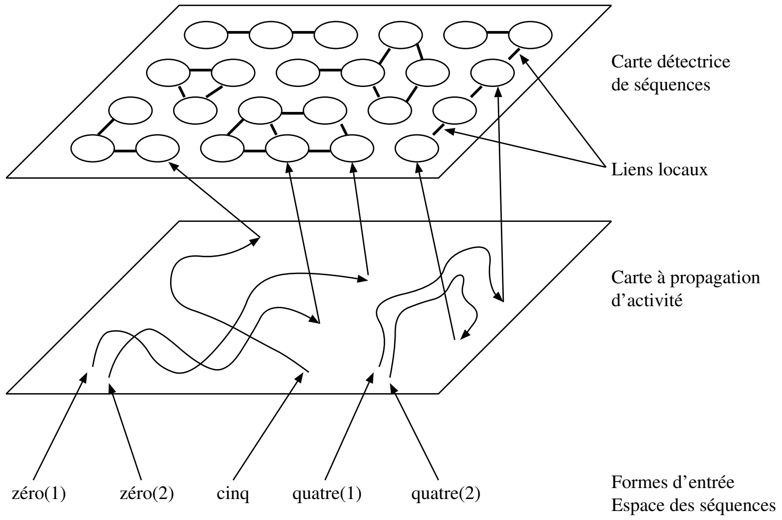


FIG. 4.17 – *Conservation de la topologie de l'espace spatio-temporel*

4. Chercher les deux super-unités les plus actives sur la carte des séquences : i_0 et i_1.
5. Si $C_{i_0 i_1} = 0$ alors $C_{i_0 i_1} = 1$ et $t_{i_0 i_1} = 0$
6. Si $C_{i_0 i_1} = 1$ alors $t_{i_0 i_1} = 0$
7. Augmenter l'âge des connexions de i_0 avec les j tels que $C_{i_0 j} = 1$ avec $t_{i_0 j} = t_{i_0 j} + 1$
8. Supprimer toutes les connexions de i_0 vers j tel que $C_{i_0 j} = 1$ et dépassant la limite d'âge : $t_{i_0 j} > T$
9. aller en 2

Pratiquement, l'intérêt de ces liens est la possibilité d'activer à travers eux non pas une seule super-unité mais tout un groupe correspondant à une entité à apprendre.

4.9 Coopération de cartes, modèles supervisés

Jusqu'à présent, nous ne sommes pas intervenu sur le modèle lors de son fonctionnement, nous en sommes resté à des méthodes non supervisées. Comme nous le verrons au chapitre 6, les résultats enregistrés par le modèle de base sur une application à la reconnaissance de chiffres parlés sont intéressants dans la mesure où aucune connaissance acoustique n'est fournie *a priori* au système. Cependant, il paraît opportun d'apporter des connaissances extérieures au système c'est-à-dire de définir une version supervisée d'un système à base de cartes TOM. Pour cela, deux raisons principales viennent à l'esprit :

- l'objectif d'améliorer les performances des réseaux construits avec des cartes TOM ;
- l'analogie avec les systèmes neurobiologiques où une zone corticale, pour prendre une décision, peut être aidée par une information de niveau supérieur extérieure à cette zone.

Nous présentons donc dans ce paragraphe une méthode pour intégrer des connaissances et pour aider à la mise en correspondance de formes. La technique que nous proposons consiste à pratiquer des associations entre formes provenant de flux de données différents. Cependant, plusieurs autres solutions peuvent être envisagées et testées, nous discuterons de ces possibilités dans la partie consacrée aux perspectives du modèle.

Dans les paragraphes précédents, nous avons utilisé au plus deux cartes TOM. Une carte TOM, directement connectée au signal d'entrée via un vecteur d'entrée, code les séquences du corpus et permet la propagation d'une activité caractérisant une forme dynamique. Cette carte peut, en outre, être connectée à une seconde carte chargée de détecter des séquences comme nous l'avons vu dans le paragraphe précédent. Nous allons maintenant introduire la notion de *réseau de cartes* pour permettre une intervention «extérieure» sur les données codées dans les cartes. A l'intérieur d'un tel réseau de cartes, nous allons considérer une carte dont l'architecture est toujours identique aux cartes TOM mais dont la fonctionnalité sera particulière : il s'agit d'une carte d'association ou carte associative. Cette carte a pour but de mettre en correspondance deux informations émanant de deux flux de données différents.

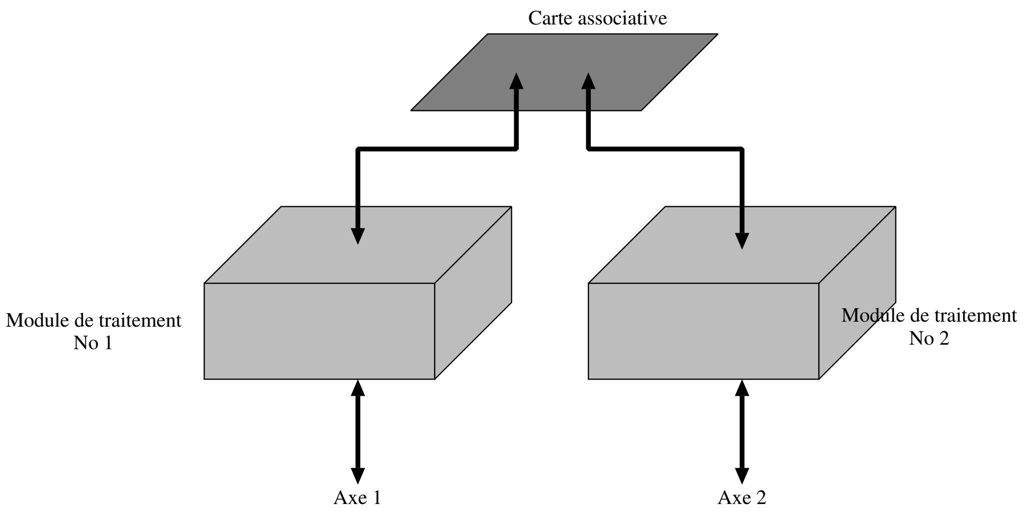


FIG. 4.18 – *Association de deux axes de traitement*

Sur la figure 4.18, nous avons représenté une carte associative connectée à deux modules de traitement dédiés chacun à une tâche particulière. Par module, on entend un réseau connexionniste pouvant se résumer à une seule carte TOM ou à un ensemble de cartes connectées entre elles. La carte associative se connecte sur les unités de sortie de ce module. Un module correspond à un axe de traitement particulier. Par exemple, il peut s'agir d'un module de reconnaissance de visages (axe visuel), d'un module de reconnaissance de mots isolés (axe auditif), d'un module de commande motrice (axe moteur), etc. Dans la nature, ces différent axes de traitement ne sont pas isolés et co-existent toujours. Par exemple, pour reconnaître des visages, il faut bien pouvoir associer à chacun de ces visages un nom de personne, pour reconnaître un mot prononcé il faut pourvoir l'associer à quelque chose permettant de l'identifier (une image par exemple)... L'association ne sert pas seulement à «fabriquer» des couples pour un aspect pratique, elle permet en général d'aider la classification des informations d'un axe donné. Par exemple, dans le cas de l'apprentissage des mots isolés, nous pouvons indiquer au système dans quelle classe de mots nous pouvons ranger la forme correspondant au mot présenté au réseau.

Examinons maintenant ce principe d'association d'un point de vue plus technique. La carte associative est une carte TOM pour laquelle nous n'utilisons pas *a priori* les liens intra-carte. En effet, nous utilisons la carte comme une simple carte de Kohonen permettant d'effectuer une classification des vecteurs qui lui sont présentés en entrée. La fusion des données en sortie des deux axes à associer constitue le vecteur d'entrée de la carte associative. Ceci est illustré sur la figure 4.19. Le principe de fonctionnement de la carte associative est alors basé sur celui du modèle LASSO [Midenet et Grumbach, 1990] présenté au chapitre 2. La figure 4.19 montre le détail des connexions entre une carte associative et deux autres cartes TOM chacune dédiée à un codage particulier.

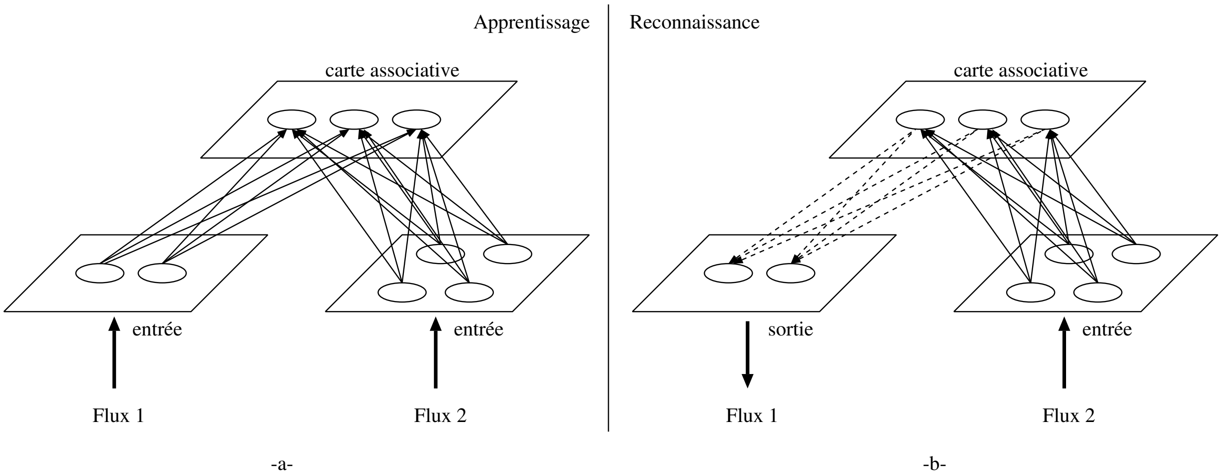


FIG. 4.19 – *Exemple de configuration pour une carte associative*

Les super-unités de la carte associative fonctionnent par convolution pour le calcul de l'activité spatiale. Ces super-unités sont donc connectées aux super-unités des cartes afférentes et non aux unités. En phase d'apprentissage, les connexions sont feed-forward, des cartes correspondant aux deux flux vers la carte associative. Celle-ci apprend donc en quelque sorte des couples d'informations provenant de ces deux flux. En phase de reconnaissance, les connexions sont inversées entre la carte associative et l'axe pour lequel on souhaite obtenir de l'information. Par exemple, et ceci sera le cas que nous traiterons au chapitre 6, en reconnaissance de la parole nous pouvons envisager une architecture à deux flux d'information. Le premier flux consiste à coder et à traiter le signal de parole tel qu'il parvient à l'oreille, le second flux correspond à un niveau symbolique. Par exemple, pour la reconnaissance de mot isolés le niveau symbolique est constitué d'une seule carte avec des super-unités qui correspondent chacune à un mot isolé. En phase d'apprentissage, on détecte quelles sont les séquences acoustiques reconnues et on active la super-unité du mot à apprendre. La carte associative s'ajuste alors en conséquence (figure 4.19.a). En phase de reconnaissance, la présentation du signal acoustique générant une propagation d'activité au niveau du flux acoustique activera une super-unité « symbole » grâce à la carte associative (figure 4.19.b).

4.10 Tolérance temporelle sélective

4.10.1 Définition et intérêt

La notion de séquences d'événements sous-tend l'architecture et le fonctionnement de notre modèle. Comme nous l'avons vu, un événement est une information (un vecteur) à un instant donné parmi un ensemble d'événements déterminés par échantillonnage du flot temporel de l'information. Jusqu'à présent, chaque événement a la même importance dans la séquence (les paramètres de mémoire à court terme, *decay*, et de suppression d'information, *thres_ act*, sont valables pour tout le réseau et sont identiques pour chaque unité). Or, on s'aperçoit que dans la réalité, un événement n'a pas toujours la même importance suivant sa position dans la séquence ou suivant sa fréquence d'apparition dans le corpus.
Par exemple en parole, si l'on considère une syllabe composée d'une occlusive suivie d'une voyelle, l'information caractérisant l'occlusive se trouve confinée dans un intervalle de temps réduit par rapport à l'intervalle de temps utile pour la voyelle. Pour cette dernière, une tolérance temporelle assez large peut intervenir alors que pour l'occlusive il n'est pas possible de tolérer des suppressions et des insertions d'événements (fig. 4.20).

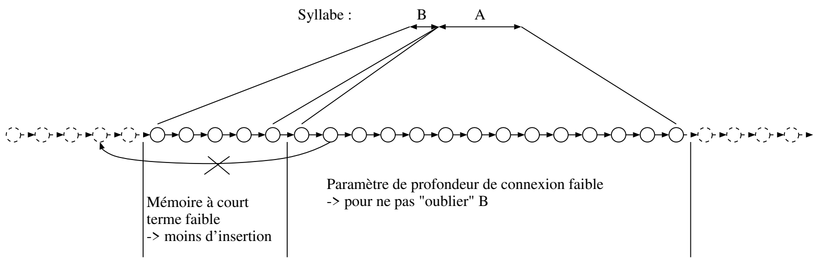


FIG. 4.20 – *Ordre et durée de l'information*

Il nous a donc paru judicieux de ne plus considérer que toutes les unités dans le réseau possèdent la même tolérance temporelle (même *decay* et *thres_ act*) et d'affecter à chacune de ces unités une tolérance qui lui est propre. De plus, parce qu'il paraît difficile d'affecter empiriquement des valeurs de tolérance à chaque unité, nous décidons de les faire apprendre. Le paragraphe suivant est consacré à la méthode d'apprentissage que nous avons choisi de mettre en œuvre.

4.10.2 Mise en œuvre de la tolérance sélective dans TOM

L'algorithme de modification des paramètres de tolérance se base sur trois choix possibles lorsqu'une nouvelle séquence est présentée à la carte :

– ne rien faire sur les paramètres de tolérance,

– fusionner des séquences, modifier les paramètres,

– séparer des séquences, modifier les paramètres.

Ces trois choix sont issus des observations que l'on peut faire lors du fonctionnement d'une carte et en particulier dans les cas de confusion de séquences. Cette remarque implique que les séquences sont étiquetées dans la carte et que le réseau a connaissance des étiquettes des séquences qui lui sont présentées. Ainsi, les trois choix précédents s'argumentent de la façon suivante :

– une séquence est présentée au réseau. Il s'en suit une propagation d'activité relativement faible et n'aboutissant pas au déclenchement d'une séquence codée. Nous considérons donc la séquence présentée comme nouvelle et la codons directement dans la carte en lui affectant des valeurs de tolérance (*decay* et *thres_ act*) empiriques et définies *a priori*.

– Une séquence est présentée au réseau et induit une propagation d'activité conduisant à une reconnaissance correcte. La séquence ayant conduit à cette reconnaissance est repérée et fusionnée avec la séquence d'entrée. Les unités communes à ces deux séquences sont considérées comme étant les unités principales de la séquence (responsables de son activité), c'est la raison pour laquelle les paramètres de tolérance sont adaptés pour forcer leur présence dans les activations futures. Une façon de renforcer l'importance de ces unités est d'augmenter le *decay* pour chacune d'elle et de diminuer la profondeur de connexion (*thres_ act*) pour les autres (fig. 4.21).

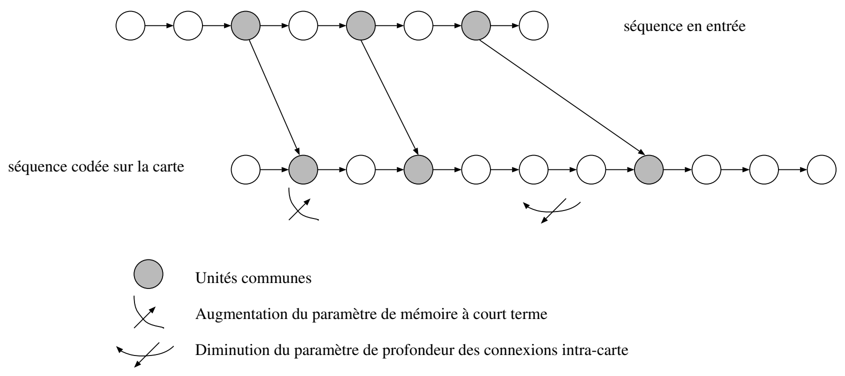


FIG. 4.21 – *Fusion de séquences et adaptation des paramètres de tolérance temporelle*

– Une séquence est présentée au réseau et induit une propagation d'activité aboutissant à une confusion dans les étiquettes (par exemple, la séquence S_1 est présentée et c'est la séquence S_2 qui s'active dans le réseau). Dans ce cas, il est nécessaire de « séparer » les deux séquences. Si les deux séquences ont été activées c'est que des unités communes l'ont été. Il convient d'empêcher par la suite l'activation de ces unités communes. La technique employée ici consiste à diminuer les deux paramètres *decay* et *thres_ act* (fig. 4.22).

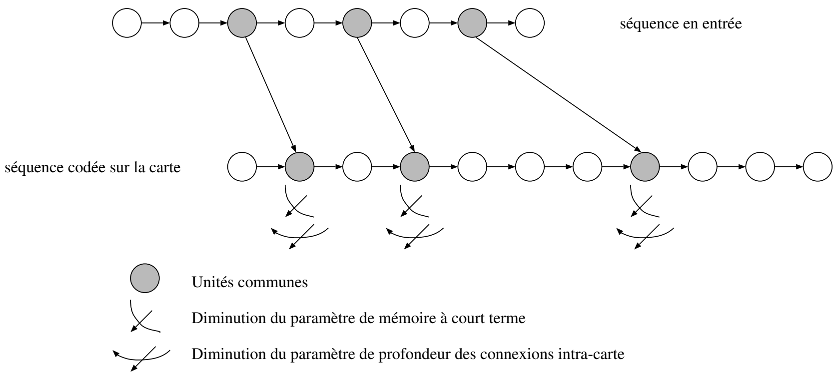


FIG. 4.22 – *Séparation de séquences et adaptation des paramètres de tolérance temporelle*

Il ne reste plus qu'à appliquer ce processus sur la totalité du corpus d'apprentissage répété plusieurs fois, en espérant que la carte des séquences se stabilisera dans une configuration stable des paramètres de tolérance.

La méthode de modification des paramètres de tolérance aux distorsions temporelles est intuitive. Elle demande a être validée par l'expérience et éventuellement à être modifiée. Cette partie de nos travaux constitue le développement actuel.

4.11 Résumé sur l'architecture

Le lecteur l'aura bien compris tout au long de la lecture de ce chapitre, TOM n'est pas un modèle « tout fait » pour telle ou telle application. Il s'agit d'une architecture capable de procéder à la classification d'un espace incluant une dimension temporelle. En particulier, une carte TOM possède des propriétés intéressantes notamment la tolérance aux distorsions temporelles. Le tableau de la figure 4.23 résume les différents constituants et les connexions possibles entre eux dans une architecture à base de cartes TOM.

Cette conception modulaire d'un réseau connexionniste se retrouvera dans les programmes mis en œuvre pour les diverses simulations que nous avons développées. Des éléments concernant l'implantation informatique sont donnés en annexe.

4.12 Conclusion

Dans ce paragraphe, nous ne posons pas un point final à ce chapitre en concluant sur un modèle complètement fini. Tout d'abord parce que le modèle n'est pas terminé dans le sens où il n'est pas un système fourni « clef en main » pour une application donnée. Ensuite, parce que nous n'avons pas encore évalué les capacités et les possibilités de l'architecture, ceci sera fait au chapitre 6. Nous pouvons cependant résumer d'un point de vue théorique les motivations qui nous ont conduits à ce type d'architecture connexionniste, discuter des points forts et des points faibles qui sont à nos yeux attachés au modèle.

Nous l'avons dit en introduction à cette thèse, la plupart des problèmes auxquels l'homme et la machine sont confrontés incluent une composante dynamique, un aspect temporel. Il était donc souhaitable que les réseaux de neurones artificiels prennent en compte cet aspect temporel dans le cadre d'applications diverses comme la reconnaissance de la parole. La communauté scientifique internationale en connexionnisme s'est donc tout naturellement attaquée à ce problème. Ainsi, divers modèles ont été proposés, nous en avons vu les grandes lignes au chapitre 3. Notre approche ne se situe pas dans le courant majeur des modèles connexionnistes temporels. Il s'agit en effet de s'inspirer de faits neurobiologiques pour construire une architecture temporelle. Nous ne sommes évidemment pas les premiers à avoir tenté ce genre d'approche : avant nous D. Béroule [Béroule, 1990] et son système à propagation guidée, F. Alexandre et F. Guyot [Alexandre *et al.*, 1991] et le modèle de la colonne corticale et d'autres encore. Dans cette droite ligne, nos travaux gardent à l'esprit deux objectifs : inspiration neurobiologique et facilité d'application à des problèmes concrets.

Il est cependant bien difficile d'atteindre simultanément ces deux objectifs. En voulant « copier » des structures neuronales nous faisons abstraction de l'application. Les performances enregistrées sont alors médiocres car pour une application fixée il faut toujours « régler » le système propre à cette application, positionner divers paramètres, bref, effectuer un travail de mise au point qui est d'autant plus important que le modèle d'inspiration neurobiologique est général. En partant de l'application, la mise au point du modèle se trouve nécessairement influencée par cette application. Les résultats enregistrés sont alors assez bons mais le modèle a perdu en généricité. L'application à un autre problème risque alors de causer quelques déboires. Dans notre travail, nous avons essayé de ne pas tomber dans l'un de ces écueils.

L'idée qui est à l'origine de la carte TOM et la philosophie mise en jeu proviennent d'une observation macroscopique qui consiste à considérer une succession d'informations pour aboutir à une réponse ou à un autre type d'information pouvant à nouveau être exploité. Si l'on se place à un niveau d'abstraction assez élevé, le niveau du raisonnement par exemple, une démonstration mathématique est une suite d'hypothèses générant d'autres hypothèses permettant d'aboutir à une conclusion. A un niveau moindre, un mouvement de la main est une succession d'actions musculaires pour atteindre un but. Le cerveau est bien entendu capable de raisonner et de faire bouger des membres, qui plus est, les architectures cellulaires au niveau des zones corticales chargées des tâches précédemment citées sont identiques [Burnod, 1988]. A partir de ces observations, nous avons voulu développer une architecture qui soit identique quelle que soit la tâche à traiter et qui puisse être capable d'apprendre et d'organiser des séquences d'informations. Ce type d'architecture n'est pas souhaitable uniquement pour l'analogie avec le cortex et ses aires corticales, il permet d'interconnecter plus facilement des réseaux qui sont homogènes. La carte TOM permet donc de collecter des informations sous la forme de propagation d'activité directement exploitable pour une action ou une application donnée.

La généricité de l'architecture semble donc être un avantage pour la conception de systèmes assez complexes traitant, en particulier, de plusieurs axes sensoriels comme par exemple un système de contrôle de mouvements par la voix. Nous le verrons plus en avant dans le chapitre consacré aux applications, une carte TOM possède des propriétés tout à fait intéressantes dans le domaine spatial et temporel. En particulier, une carte possède une résistance aux distorsions temporelles des séquences qu'elle perçoit. Il est de plus possible de représenter, par l'intermédiaire d'une telle carte, l'espace des séquences projeté sur la carte. On peut cependant opposer quelques critiques au modèle parmi lesquelles :

- Une séquence, lorsqu'elle est créée sur la carte, peut être légèrement adaptée pour tenir compte d'autres occurrences. Cependant, elle se généralise difficilement de façon intrinsèque par opposition aux modèles de Markov [Rabiner, 1990] où les paramètres du modèle sont appris statistiquement.

- Le nombre de paramètres associés à chaque carte TOM est important. Ces paramètres peuvent de plus être différents d'une carte à l'autre dans le cas d'un réseau de cartes. La mise au point des paramètres dans une application donnée risque donc de poser des problèmes pour la convergence du système.

Ces inconvénients font l'objet d'études pour l'avenir, nous en reparlerons dans les perspectives associées au modèle.

Nous allons regarder les cartes TOM sous un angle plus technique en les testant sur des applications de traitement automatique de la parole (chapitre 6). Mais avant cela, nous dressons un panorama non exhaustif des modèles connexionnistes appliqués à la reconnaissance de la parole (chapitre 5).

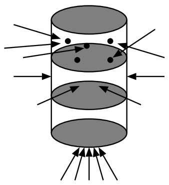
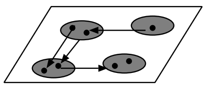
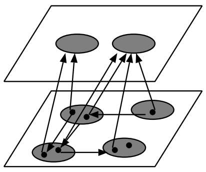
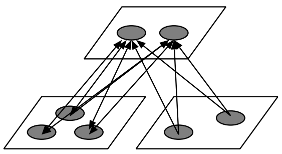
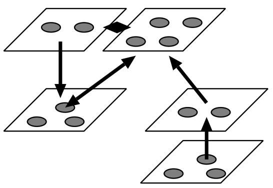
		Description / Fonctionnalité	Connectivité / Apprentissage
Super-unités et unités		- Codage des stimuli (codés sous la forme de vecteurs) - Influences voisines, groupement fonctionnel - Réaction des unités conditionnées aux séquences d'activations passées	- Liens feed-forward, déclenchement de l'activité spatiale par convolution ou filtrage suivant le type d'afférence (unité ou super-unité) - Liens locaux ou de voisinage - Liens intra-cartes pour la définition du contexte temporel d'activation
Carte de super-unités TOM		- Carte TOM : assemblage de super-unités - Apprentissage et codage de séquences à partir d'un corpus de formes dynamiques - Tolérance aux distorsions temporelles	- Connectivité intra-carte entre les unités des super-unités - Activités des unités du type des intégrateurs à fuite et connexion non linéaire → tolérance insertion/suppression évt.
Assemblage de cartes Détection de séquences		- Deux niveaux d'abstraction : niveau signal d'entrée, niveau détection de séquences - Qualification et quantification des séquences	- Deux types de connectivité ascendante - Connectique en fin de séquence → déclenchement de la super-unité (0/1) - Connectique sur toutes les unités de la séquence → filtre impulsionnel, activité continue
Carte TOM d'association de flux		- Utilisation d'une carte TOM en tant que carte associative - Permet de superviser l'apprentissage d'une carte grâce aux informations contenues sur une autre	- Les connexions feed-forward se font de super-unité à super-unité - Un des deux axes sert de professeur - Les connexions sont "retournées" pour cet axe en phase d'apprentissage
Réseaux de cartes TOM		- Assemblage de cartes TOM en vue d'une application particulière - Plusieurs niveaux de fonctionnalités - Interfaçage de plusieurs axes fonctionnels différents	- Tous les types de connectivité sont mis (ou peuvent être mis) en oeuvre dans le cas d'un système complet

FIG. 4.23 – *Tableau récapitulatif du modèle*

Chapitre 5

Le neuromimétisme et la parole

« C'est dans les mots que nous pensons. Nous n'avons conscience de nos pensées déterminées et réelles que lorsque nous leur donnons la forme objective, que nous les différencions de notre intériorité, et par la suite, nous les marquons d'une forme externe, mais d'une forme qui contient aussi le caractère de l'activité interne la plus haute. C'est le son articulé, le mot, qui seul nous offre une existence où l'externe et l'interne sont si intimement unis. Par conséquent, vouloir penser sans les mots, c'est une tentative insensée. [...] Et il est également absurde de considérer comme un désavantage et comme un défaut de la pensée cette nécessité qui lie celle-ci au mot. On croit ordinairement, il est vrai, que ce qu'il y a de plus haut, c'est l'ineffable. Mais c'est là une opinion superficielle et sans fondement; car, en réalité, l'ineffable, c'est la pensée obscure, la pensée à l'état de fermentation, et qui ne devient claire que lorsqu'elle trouve le mot. Ainsi le mot donne à la pensée son existence la plus haute et la plus vraie. »

Georg Wilhelm Friedrich Hegel, Philosophie de l'Esprit, 1817.

5.1 Introduction

Le traitement automatique de la parole est un champ de recherche important depuis maintenant une cinquantaine d'années. Bien que d'énormes progrès aient été accomplis tout au long de ces années, les machines et les programmes n'atteignent pas encore les performances des êtres humains dans des conditions réelles. La difficulté de la tâche est due principalement à l'incroyable variabilité et à la distorsion du signal de parole [Klatt, 1986]. Cette variabilité peut être spatiale, les caractéristiques acoustiques d'une sonorité étant variables d'une prononciation à une autre et d'une personne à une autre. Elle peut également être temporelle. En effet, la dilatation ou la compression du signal est un phénomène courant et pose des problèmes importants aux systèmes automatiques. De plus, les phénomènes contextuels sont intrinsèques à la parole. En effet, lors de la prononciation d'un phonème, la structure acoustique de ce phonème dépend du contexte dans lequel le phonème a été prononcé, c'est à dire des phonèmes précèdent et suivant. De plus, comme la majeure partie des applications se déroulent en situations réelles, l'onde sonore n'est jamais pure et isolée, elle circule dans un environnement bruité. On le voit donc, le problème de la reconnaissance de la parole est relativement complexe et constitue un véritable défi pour les années à venir.

La recherche dans le domaine de la parole a donné lieu à bon nombre de techniques et algorithmes. Deux d'entre eux ont connu et connaissent encore un grand succès accompagné de performances intéressantes. Il s'agit du recalage temporel dynamique (« DTW ») et des modèles de Markov cachés (« HMM »). Dans le premier cas, l'algorithme compare par programmation dynamique une forme à reconnaître parmi un ensemble de formes de référence. Pour cela, on établit une fonction de coût qui permet de comparer avec ces formes de référence. Le DTW exige une construction manuelle de la base des formes. A l'inverse, les HMM permettent un apprentissage automatique de chacune des formes. Cependant, la création d'un modèle par forme est nécessaire. Un inconvénient majeur est qu'il n'est pas possible d'apprendre la structure topologique des formes, en particulier il n'est pas possible d'extraire des sous-structures [Lippmann, 1989]. C'est sans doute à ce niveau que les réseaux neuronaux peuvent surclasser les autres techniques.
En effet, relevant de la reconnaissance des formes, le connexionnisme a tout naturellement été utilisé pour le traitement automatique de la parole et a obtenu des résultats honorables dans certains cas bien précis. La démarche utilisée est en général une démarche applicative : les réseaux classiques sont appliqués modulo un pré-traitement adéquat du signal de parole. Nous avons vu au chapitre 2 un certain nombre de modèles classiques et au chapitre 3 une classification des modèles connexionnistes temporels. Une grande partie de ces modèles a été appliquée au domaine de la parole. Dans ce chapitre, nous n'allons pas revenir sur les descriptions architecturales et fonctionnelles de ces réseaux, nous citons simplement les modèles qui nous semblent assez bien représentatifs de l'utilisation des réseaux connexionnistes en parole.
Parce que le neuromimétisme puise ses origines dans le fonctionnement biologique, il peut être intéressant d'étudier la structure et la fonctionnalité de l'axe cortical auditif afin de s'en inspirer pour concevoir des systèmes connexionnistes. La tâche devient nettement plus difficile parce qu'en premier lieu, peu de données certaines concernant le cortex auditif de haut niveau sont en notre possession. En revanche, les mécanismes de la perception auditives sont relativement bien connus, en particulier la transduction onde-message auditif [Buser et Imbert, 1987][Pujol, 1990]. Au-delà, concernant les mécanismes corticaux de la reconnaissance proprement dite de la parole et du langage, nous passons dans le domaine des hypothèses. Finalement, cela ne constitue pas véritablement un handicap pour nous, modélisateurs. En effet, cela entraîne une collaboration avec les neurophysiologistes afin de proposer, de faire émerger des idées, et de tester des modèles sur ordinateur.

Le chapitre s'articule en trois parties. La première présente brièvement l'axe cortical auditif de l'onde sonore au cortex. L'étude du traitement biologique peut nous être d'une aide précieuse pour la conception de modèles. Par exemple, l'étude du fonctionnement cochléaire a donné lieu à des modèles de la cochlée, l'organisation phonotopique des cartes de Kohonen possède une analogie avec la conservation de la topologie lors des projections sur les aires corticales (tonotopie par exemple). La seconde partie présente quelques modèles connexionnistes, certains sont inspirés de la neurobiologie d'autres non, il nous paraît cependant intéressant de les citer afin de situer notre approche. Enfin la dernière partie traite des modèles hybrides combinant la programmation dynamique ou les modèles de Markov cachés avec les réseaux de neurones. Ces modèles semblent être actuellement une voix prometteuse et donnent de bons résultats.

5.2 L'onde sonore de l'oreille au cortex : point de vue du modélisateur

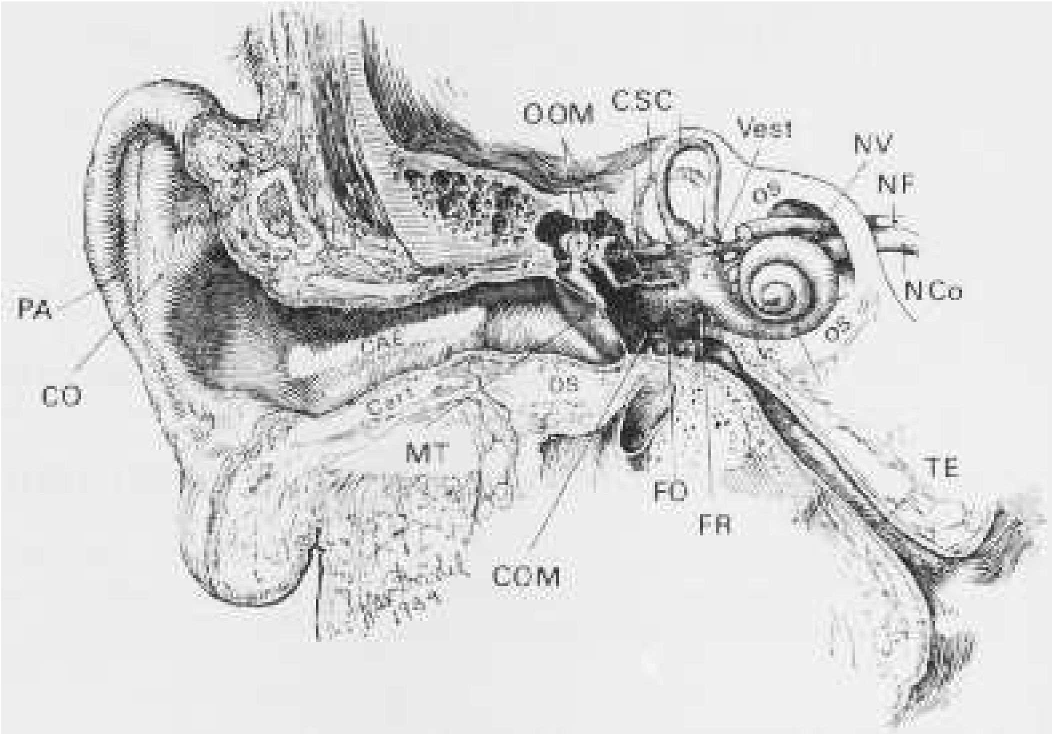
5.2.1 Biologie auditive et modélisation

Dans le but de concevoir des systèmes neuromimétiques capables de reconnaître la parole, il peut être intéressant d'étudier les organes et les mécanismes de la compréhension sonore chez les êtres vivants. En particulier, l'étude de la reconnaissance de la parole chez l'homme peut être source d'inspiration pour la conception d'outils de reconnaissance. Malheureusement, peu d'information concernant le traitement de la parole dans le cortex est disponible. Les nouvelles méthodes d'investigations non invasives (enregistrement des activités physiologiques en surface, techniques telles que l'imagerie par résonance magnétique ou la tomographie par émission de positons) permettront sans doute à l'avenir une étude plus fine du cortex auditif en particulier. Cependant, si les informations aux niveaux corticaux élevés ne foisonnent pas, le traitement acoustique par le cerveau semble mieux connu. Les études par implantation d'électrodes au sein même du cerveau des animaux (du chat en particulier) permettent une bonne connaissance du traitement acoustique. Dans la suite, l'exposé simplifié de l'axe auditif dans le cerveau sera fait selon le point de vue du modélisateur et du neuromiméticien en particulier. Il ne s'agit donc pas d'une présentation fine et précise des mécanismes corticaux de l'audition mais plutôt d'un énoncé des structures de base ainsi que de leur fonctionnement (dans la mesure où il est connu). La présentation des ces structures se fera selon l'axe ascendant du traitement sonore, de l'oreille au cortex.

5.2.2 Réception du message auditif

L'oreille externe et moyenne

La cochlée (tube osseux enroulé en spirale) dont nous détaillerons le fonctionnement plus loin, est l'organe fondamental de la réception sonore. En effet, c'est elle qui transforme l'onde acoustique en message auditif accepté par le système nerveux. Cependant, si



PA : pavillon ; CO : conque ; CAE : conduit auditif externe ; Cart : cartilage ; MT : membrane tympanique ; O.OM : osselets de l'oreille moyenne ; CSC : canaux semi-circulaires ; FO : fenêtre ovale ; FR : fenêtre ronde ; COM : cavité de l'oreille moyenne : Vest : vestibule : NV : nerf vestibulaire ; NF : nerf facial ; NCo : nerf cochléaire ; TE : trompe d'Eustache

FIG. 5.1 – *Coupe longitudinale de l'appareil auditif humain*

la cochlée devait recevoir directement l'onde sonore (c'est-à-dire si la cochlée était directement en contact avec l'air ambiant) seulement 2% de la puissance du message ondulatoire serait absorbé [Buser et Imbert, 1987]. En réalité, grâce à la chaîne de traitement contenue dans l'oreille externe et moyenne, 67% de la puissance est absorbée (estimations d'après [Buser et Imbert, 1987] pour une zone fréquentielle moyenne de 1kHz à quelques kHz chez l'homme). L'oreille peut donc être modélisée par une suite de modules de transfert dont l'entrée est l'onde sonore du milieu ambiant et la sortie, les influx nerveux se propageant sur le nerf auditif et aboutissant aux premiers complexes nerveux de traitement de l'information auditive (cf. fig. 5.2).

Sur la figure, les modules I et II peuvent être assimilés à la phase de « prise de son » alors que le module III commence à traiter réellement l'onde sonore. Par rapport à un système de reconnaissance de la parole, ce module participe au prétraitement du signal. Le conduit auditif (module I) intervient pour un accroissement de la pression. Cet accroissement est fonction de la fréquence et est maximal pour 2.5kHz. Ce conduit serait également impliqué dans la localisation des sources sonores. L'oreille moyenne (module II : tympan, osselets, fenêtre ovale de la cochlée ...) agit comme un transformateur de l'impédance du milieu. L'impédance acoustique est une mesure physique caractérisant le pouvoir de transmission du milieu. Plus ce paramètre est grand, plus l'information ondu-

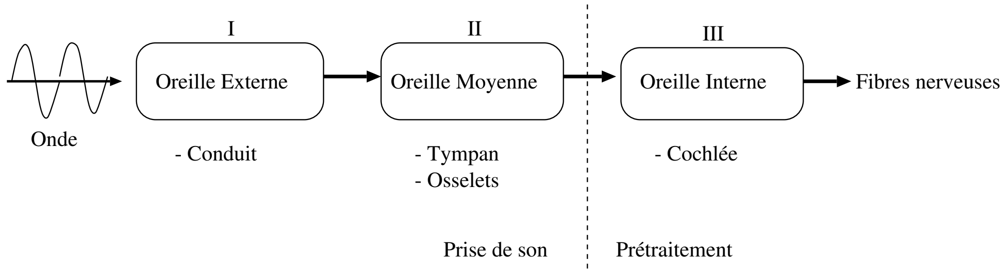


FIG. 5.2 – *Modules de traitement d'information dans l'oreille*

latoire transmise sera importante. Si les vibrations de l'air devaient aboutir directement sur l'ouverture de la cochlée, 98% de l'énergie vibratoire serait réfléchie. Grâce à l'oreille moyenne, le coefficient de transmission se trouve considérablement accru. Structure du tympan, osselets (marteau, enclume et étrier) et articulation des osselets en liaison directe avec la cochlée permettent l'augmentation de la transmission. On trouvera de plus amples détails dans [Buser et Imbert, 1987].

L'oreille interne, transduction cochléaire

La transformation du message ondulatoire en message auditif est réalisée par la cochlée (module III). La mécanique de cet organe est complexe dans sa mise en œuvre mais finalement assez simple sur le principe. A la base de la cochlée, on trouve deux ouvertures [Buser et Imbert, 1987; Pujol, 1990; Aitkin, 1990] permettant, pour l'une (la fenêtre ovale) la transmission des vibrations tympaniques par l'intermédiaire des osselets, et pour l'autre (la fenêtre ronde) la bonne propagation des ondes. La cochlée est un tube osseux en forme de spirale (cf. fig.5.1) et remplie d'un liquide (la périlymphe). Elle possède un canal central lui même empli d'un liquide (l'endolymphe) et limité par la membrane basiliaire supportant les cellules réceptrices. Ces cellules sont appelées *cellules ciliées* parce qu'elles forment à leur extrémité des cils plongés dans le liquide cochléaire. Les mouvements de ces cils conduisent à une décharge au niveau cellulaire. Deux grands types de ces cellules interviennent au niveau du canal cochléaire (fig. 5.3).

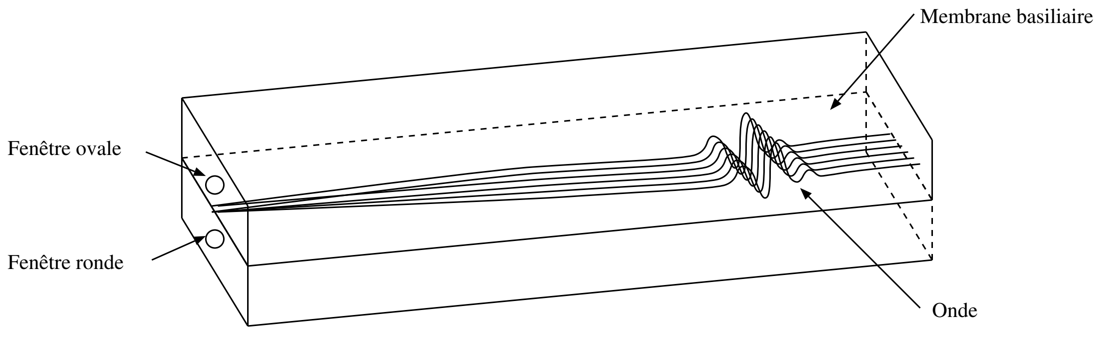


FIG. 5.3 – *Schéma simplifié d'une cochlée déroulée*

Réparties linéairement le long de la cochlée, de la base à l'apex (centre de la spirale), les cellules du premier type, les *cellules ciliées internes* forment une seule rangée et sont au

nombre d'environ 3500 ; les cellules du second type, les *cellules ciliées externes*, forment 3 ou 4 rangées et sont au nombre d'environ 12000. Les cellules internes codent à elles seules 95% de l'onde sonore et forment 95% du nerf auditif. Nous pouvons nous étonner du fait que les cellules externes, en plus grand nombre, n'interviennent alors que pour 5% du codage du signal. Des études ont montré que la destruction des cellules externes [Aitkin, 1990] entraînait une perte de l'audition aux intensités normales. Ces cellules jouent donc un rôle fondamental dans l'audition. L'hypothèse la plus commune concernant leur fonctionnalité est qu'elles jouent un rôle d'amplification des mouvements liquidiens grâce à des battements commandés par un système efférent de niveau supérieur comme le cortex. Chaque cellule interne réagit à une fréquence donnée. Il y a une répartition ordonnée des fréquences le long de la cochlée, la base réagissant aux hautes fréquences et l'apex aux basses fréquences. Ceci serait rendu possible grâce aux propriétés mécaniques de la membrane basiliaire sur laquelle reposent les cellules ciliées, celle-ci étant plus dure à la base et plus souple au niveau de l'apex.
En résumé, la cochlée est le premier organe intervenant dans la reconnaissance des sons. Elle assure une représentation tonotopique des fréquences et envoie l'information fréquentielle au niveau du cerveau. De plus, un mécanisme de rétroaction par le système nerveux sur la cochlée est mis en évidence. Ce système permet une « focalisation » ou un « réhaussement » de certaines bandes de fréquences (voir le schéma du mécanisme général de la transduction cochléaire fig. 5.4).

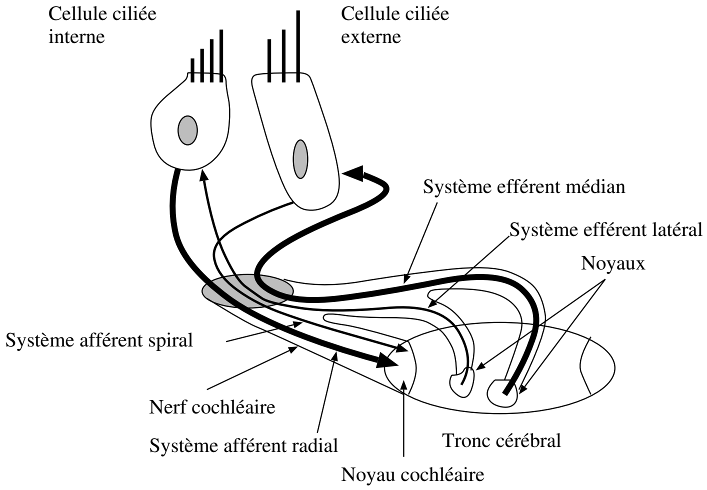


FIG. 5.4 – *Schéma du mécanisme général de la transduction cochléaire*

5.3 Le cortex auditif

Si le transcodage de l'onde sonore en influx nerveux est relativement bien connu, le traitement auditif de haut niveau, c'est-à-dire en particulier la reconnaissance de la parole, l'est beaucoup moins [de Ribaupierre, 1986; Aitkin, 1990; Buser et Imbert, 1987]. Des aires corticales dédiées à l'audition ont pu être mises en évidence mais leur rôle fonctionnel n'est pour l'instant pas élucidé. Il est cependant intéressant d'examiner l'organisation de l'audition dans le cortex ainsi que les hypothèses émises concernant la fonctionnalité des zones corticales auditives.

5.3.1 Topographie

Le cortex possède un certain nombre d'aires corticales auditives, par exemple, chez le chat : l'aire auditive primaire (AI), les aires secondaires (AII) et tertiaire (AIII), la région ectosylvienne postérieure (Ep), la frange suprasylvienne (SF) et l'insula (INS) [de Ribaupierre, 1986]. A cela, il faut ajouter les aires associatives combinant l'audition à d'autres modalités. Des découpages encore plus fins donnent lieu à d'autres aires, nous ne les détaillons pas ici car notre propos se situe plutôt sur un aspect fonctionnel. Chez l'Homme, on notera également des aires plus spécifiques comme l'aire de Wernicke pour la reconnaissance du langage ou encore l'aire de Broca pour la production du langage [Geshwind, 1984] (voir fig. 2.9 et 2.8). Une aire peut recevoir des projections de sous-structures corticales, comme le thalamus, mais aussi d'autres aires. En particulier, la cochlée se projette sur l'aire primaire, celle-ci conservant une tonotopie précise. Enfin, comme nous l'avons précisé au chapitre 2, la colonne corticale se retrouve partout dans le cerveau quelle que soit la fonctionnalité :

> « *[...] une aire donnée peut être considérée comme formée d'une série de modules fonctionnels unitaires juxtaposés. Ces modules, ou colonnes corticales, sont constitués par l'ensemble des neurones compris dans un cylindre de 200 à 300 μm de diamètre, perpendiculaire à la surface et traversant le cortex dans toute son épaisseur. Tous les neurones d'un module partagent des propriétés fonctionnelles identiques, car ils reçoivent des entrées thalamiques communes via la couche IV, qui sont redistribuées à l'intérieur du module par l'intermédiaire de ses connexions intrinsèques. L'information est ainsi traitée à l'intérieur de la colonne, puis redistribuée vers d'autres modules corticaux et vers les structures profondes. La sortie d'un module résulte donc, à chaque instant, du traitement local effectué en fonction du câblage propre à ce module sur les entrées qu'il reçoit du thalamus et les réafférences corticales qui proviennent d'autres modules proches ou lointains. Le tout est encore modulé par des entrées diffuses provenant du tronc cérébral.*»
> *F. de Ribaupierre [de Ribaupierre, 1986]*

5.3.2 Fonctionnalité

Le cortex extrait des paramètres auditifs comme par exemple les fréquences sonores ou encore la durée sonore dans le cas de l'écholocation [Suga, 1990]. Il est également capable d'interagir sur les structures afférentes. Ainsi, le cortex agirait sur le thalamus, fonctionnant en filtre adaptatif, pour obtenir certaines fréquences au détriment d'autres.
Sur le plan temporel, plusieurs expériences ont été proposées et ont montré l'importance du paramètre temps dans le codage de l'information. Par exemple, la configuration d'activation pendant la présentation d'un son pur peut être différente de la configuration pour la présentation d'un même son pur suivant la nature du son qui précède. En outre,

l'aspect séquentiel de l'information (pas seulement dans le cortex auditif) est un grand débat : *l'information n'est peut-être pas simplement codée par la fréquence instantanée des neurones, mais aussi par la séquence spatio-temporelle de leur activation. [de Ribaupierre, 1986].*

5.4 Modélisation et sous-problèmes de la reconnaissance de la parole

Après cette brève introduction au traitement auditif du signal de parole par le cortex auditif, nous allons maintenant donner quelques exemples de modèles permettant un traitement automatique de la parole. Nous allons bien évidemment mettre l'accent sur les modèles connexionnistes, cependant nous citons d'autres méthodes dont les résultats sont remarquables et qui sont de plus en plus fréquemment combinées avec des systèmes connexionnistes (modèles hybrides). Nous l'avons vu au chapitre 3, nous pouvons distinguer plusieurs classes de modèles connexionnistes temporels. Ces systèmes sont naturellement utilisés en parole bien que les modèles statiques soient toujours d'actualité. Dans notre exposé, nous présentons successivement les modèles statiques appliqués à la parole puis les modèles intégrant la notion de temps. Le traitement automatique de la parole recouvre plusieurs champs d'investigation. :

- reconnaissance de phonèmes, syllabes, mots ;
- synthèse de la parole ;
- reconnaissance et vérification du locuteur.

Ces trois grands centres d'intérêt impliquent un certain nombre de problèmes qu'il convient de pouvoir résoudre dans le cas d'un système pouvant fonctionner en situation réelle :

- Problème de l'alignement et du recalage temporel. En phase de reconnaissance, les systèmes doivent tenir compte de la variabilité temporelle du signal de parole. Les variabilités intra et inter-locuteurs peuvent engendrer en général une compression/dilatation du signal.
- Extraction de traits acoustiques. Ceux-ci sont nécessaires pour la représentation de l'information acoustique.
- Elimination du bruit. En situation réelle les enregistrements se font rarement dans le silence le plus complet. Plusieurs types de bruits peuvent être présents dans le signal.
- Codage du signal. En particulier, ce problème intervient dans les problèmes de communication, de stockage des données ou encore de cryptage de l'information.
- Identification du locuteur. Il s'agit ici d'extraire des indices pertinents pour pouvoir caractériser les locuteurs.

– Localisation de mots et segmentation. En parole continue, il peut s'avérer nécessaire d'isoler certaines parties du signal afin d'améliorer les performances.

Dans ce qui suit, nous ne parlons pas des systèmes de reconnaissance de parole fondés sur l'utilisation explicite de connaissances. Nombre de systèmes ont été développés (par exemple, APHODEX [Fohr, 1986], KEAL [Mercier, 1990]) et beaucoup de travaux sont encore aujourd'hui entrepris dans ce domaine notamment dans l'étude des caractéristiques fréquentielles.

5.5 Modèles connexionnistes sans paramètre temporel

Ces modèles regroupent principalement les perceptrons et les modèles compétitifs comme les cartes auto-organisatrices de Kohonen. Les architectures de ces modèles ont déjà été présentées, nous nous intéressons ici à leur application au traitement de la parole.

5.5.1 Les perceptrons

Le perceptron est utilisé en tant que classifieur ou associateur de formes. L'utilisation la plus connue du perceptron en parole est sans nul doute le réseau NetTalk de Sejnowski [Sejnowski et Rosenberg, 1987]. Les résultats obtenus avec ce réseau ont mis l'algorithme de rétro-propagation de l'erreur sur le devant de la scène. Le réseau est un transcripteur graphèmes-phonèmes (synthétiseur de parole) et comporte déjà un aspect temporel ou plutôt contextuel.

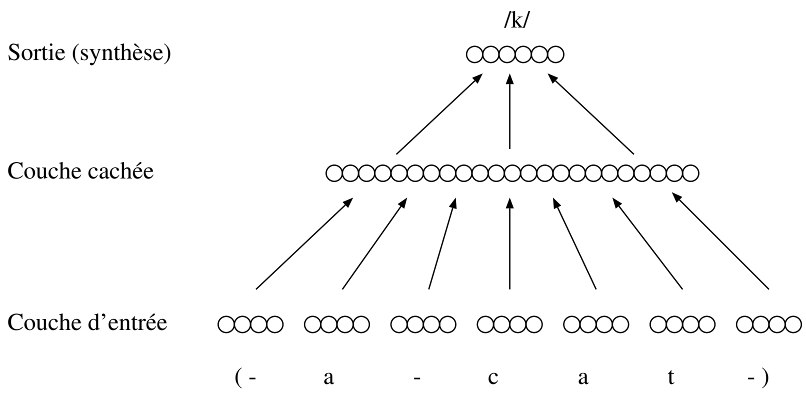


FIG. 5.5 – *Architecture du réseau NETtalk*

En effet, comme le montre la figure 5.5, la couche d'entrée est en fait composée de groupes de neurones d'entrée. Le groupe central correspond au phonème à prononcer. Comme la prononciation d'un phonème est fortement dépendante des phonèmes environnants, la couche d'entrée utilise des groupes de neurones associés aux graphèmes environnants définissant ainsi le contexte de la prononciation. La couche de sortie définit des paramètres qu'un synthétiseur de parole peut utiliser pour prononcer le phonème central. Comme dans tout perceptron, l'algorithme d'adaptation des poids utilise un «professeur» qui indique pour un graphème particulier sa prononciation dans le contexte qui est fourni par les autres groupes de neurones.

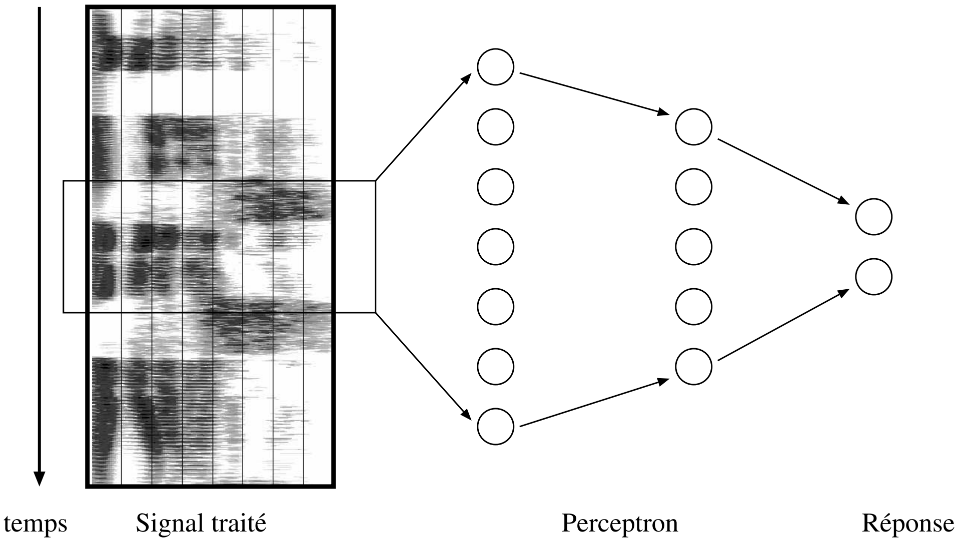


FIG. 5.6 – *Perceptron et parole*

Le perceptron est cependant plus largement utilisé pour des tâches de classification (fig. 5.6). La difficulté n'est pas tant dans la conception du perceptron mais plutôt dans le choix du codage de l'information en entrée. Il paraît illusoire de vouloir traiter directement le signal de parole brut. La majeure partie des systèmes utilise au minimum une analyse fréquentielle (transformée de Fourier rapide par exemple, FFT en anglais). La figure 5.7 représente l'architecture décrite par Huang et Lippmann [Huang *et al.*, 1988] ainsi que la segmentation de l'espace obtenue. Les vecteurs présentés en entrée du système sont de dimension 2 et contiennent les positions fréquentielles respectives des deux premiers formants de chaque mot présenté. Les neurones de sortie correspondent chacun à un mot à apprendre.

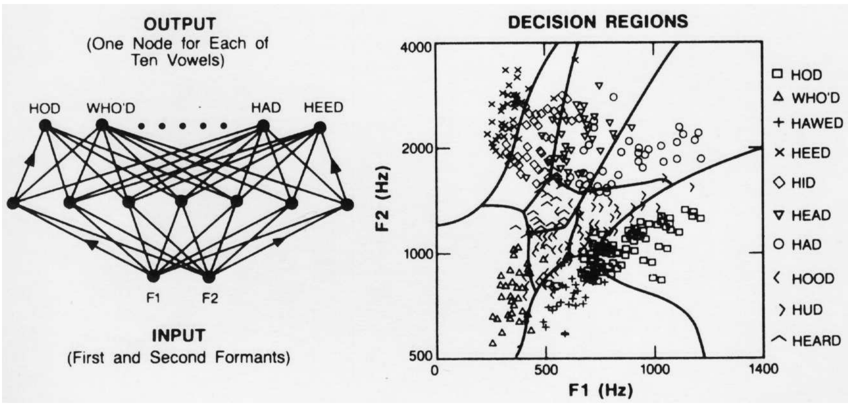


FIG. 5.7 – *Classification à partir de formants (d'après Huang et Lippmann)*

D'autres indices peuvent être utilisés comme entrées de tels réseaux, nous allons le voir au paragraphe suivant. La difficulté consiste à choisir des traits pertinents pour la reconnaissance. Bien que les réseaux connexionnistes permettent une généralisation et une tolérance aux déformations physiques sur les formes apprises, il est bien évident qu'une représentation adéquate [Davis et Mermelstein, 1980] et relativement invariante face aux « agressions » sur le signal ne peut qu'améliorer la classification opérée par le perceptron.

5.5.2 Modèles compétitifs

Les modèles compétitifs, nous l'avons vu au chapitre 2, entrent dans le cadre des systèmes auto-organisés ou non supervisés. Ils effectuent une analyse statistique des données. Kohonen propose notamment une carte topologique permettant de classer les différents phonèmes [Kohonen *et al.*, 1984]. Les neurones de la carte correspondent chacun à une classe de sonorité différente et des relations topographiques sont établies entre ces différentes classes. Grâce aux relations de voisinage proche (excitatrices) et lointain (inhibitrices) on obtient une carte « phonotopique ». La carte de Kohonen représente ainsi les aires à indices généralement observées dans les systèmes biologiques, tout en servant de système opérationnel pour la classification de formes.

Nous pouvons envisager d'autres modèles auto-organisés pour l'extraction de traits caractéristiques. Rubner et Schulten dans [Rubner et Schulten, 1990] proposent une architecture où les neurones de sortie sont connectés entre eux en cascade par des liens inhibiteurs (fig. 2.20). Le modèle effectue une analyse en composantes principales, chacun des neurones de sortie correspondant à une composante particulière [Oja, 1982].
Ce genre de systèmes est intéressant pour le codage de l'information. En particulier, un système compétitif peut constituer le premier niveau d'un système plus complexe utilisant des indices calculés au premier niveau. Par exemple, dans le cas de la reconnaissance des voyelles, on connaît l'importance de la position et de l'orientation des formants. Dans le cadre de notre stage de DEA, nous avons proposé un codage de l'information adapté au traitement des voyelles pour l'extraction et la représentation par des neurones des caractéristiques des voyelles. L'entrée du réseau est une carte (fréquence x variation d'énergie) que l'on a découpé en trois sous-cartes correspondant chacune respectivement au premier, deuxième et troisième formant. Un modèle de Rubner et Schulten composé de 4 neurones de sortie est connecté sur chaque carte. Les masques obtenus sont présentés sur la figure 5.8.

Les zones claires ou foncées au centre de chaque masque indiquent l'orientation des formants sur le spectrogramme. Par exemple, le premier neurone de la carte N^o1 réagit lorsque le formant est droit, les autres neurones réagissent pour des transitions formantiques différentes.

Même si parfois ces réseaux traitent de l'aspect dynamique de l'information par l'intermédiaire du codage de l'entrée, ils n'en demeurent pas moins statiques. Cependant, quelques modèles intégrant la notion de temps ont été proposés dans le domaine de la parole. Nous en citons quelques-uns ci-dessous.

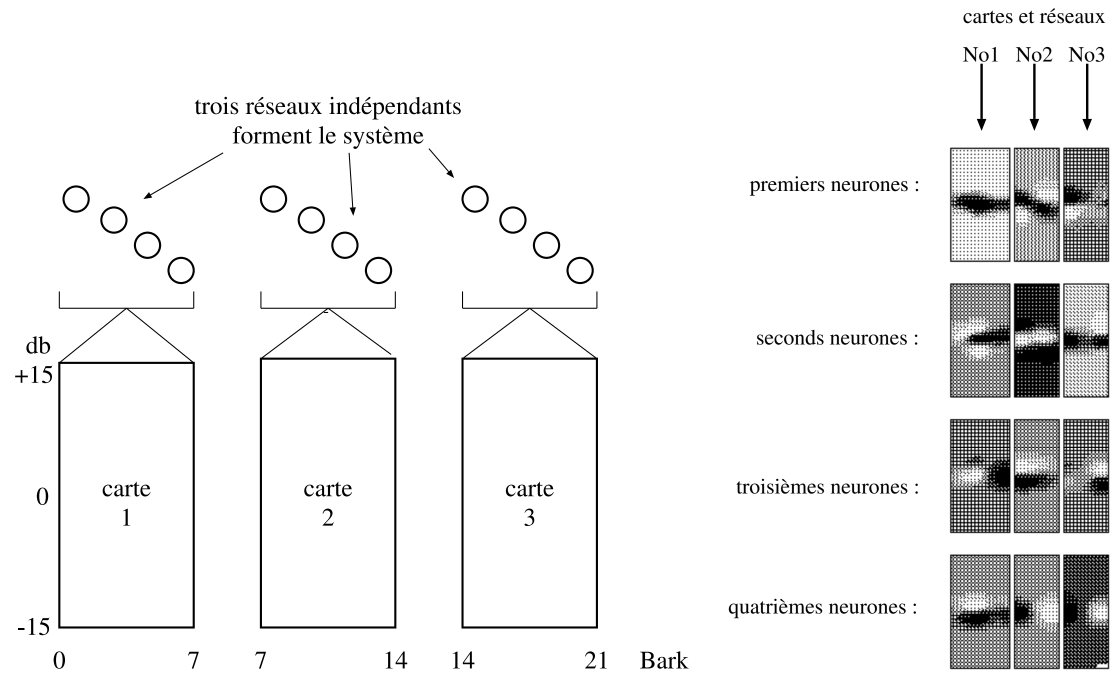


FIG. 5.8 – *Modèle compétitif de Rubner et Schulten pour l'extraction de la position et de l'orientation des formants : masques obtenus*

5.6 Modèles connexionnistes avec paramètre temporel

Au chapitre 3 nous avons établi une classification des différents modèles connexionnistes temporels. Dans la première classe, il s'agissait de représenter le temps de manière externe au réseau (représentation externe). Dans la seconde, les séquences d'information sont codées par les états internes successifs du réseau (représentation interne implicite). Enfin, dans la dernière, les réseaux permettent une propagation explicite de l'information dans des structures adéquates (représentation interne explicite, cas du modèle TOM). En parole, la fréquence d'utilisation de ces classes de représentation du temps va avec l'ordre de la présentation que nous venons d'en faire. En effet, la représentation externe semble la plus usitée, la représentation interne et implicite l'est moins et se situe à un niveau plus haut (langage par exemple), enfin la représentation interne et explicite est plus marginale et manque de modèles performants. Cette dernière est encore du domaine des laboratoires. Nous présentons dans la suite deux modèles représentant respectivement les modèles à représentation externe et les modèles à représentation interne et implicite du temps.

5.6.1 Le TDNN et ses extensions

L'architecture du TDNN a été présentée au chapitre 3. Dans le premier article concernant le TDNN [Waibel *et al.*, 1989], Waibel *et al.* décrivent une application à la reconnaissance des trois plosives voisées dans le vocabulaire Japonais, /b,d,g/. 800 phonèmes produits par trois locuteurs masculins constituent la base d'apprentissage et de test. Pour chaque phonème en contexte, le TDNN utilise 15 vecteurs de parole contenant chacun 16 coefficients issus d'une transformée de Fourier rapide (FFT) sur une fenêtre de Hamming de 256 points sur le signal de parole. Cette transformée est calculée toutes les 5 ms. Le protocole d'apprentissage et de test se situe dans le domaine mono-locuteur. Les tests effectués donnent l'avantage au TDNN avec 98,5% de reconnaissance correcte sur le corpus de test face au HMM conventionnel avec 93,7% dans les mêmes conditions d'expérience. Le modèle possède plusieurs couches qui intègrent l'information de façon de plus en plus abstraite. La structure d'un phonème est différente d'un contexte à l'autre, ainsi la configuration des 15 vecteurs d'entrée est donc différente pour deux contextes différents. Cependant, on constate [Morgan et Scofield, 1991] que l'effet de contexte s'estompe dans la configuration d'activités de la dernière couche cachée du réseau. La figure 5.9 montre la configuration d'activité de la couche cachée pour la prononciation du phonème /d/ en contexte /a/ et /o/ ("da", "do").

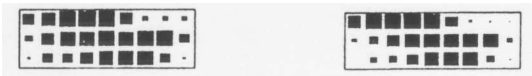


FIG. 5.9 – *Activité de la dernière couche cachée du TDNN pour la syllabe "da" (à gauche) et "do" (à droite).*

Le TDNN possède de bonnes propriétés pour le recalage temporel et il est bien adapté à la reconnaissance de phonèmes. Cependant, il se révèle moins performant pour des tâches de discrimination de mots notamment à cause de la durée de l'apprentissage mais aussi à cause de problèmes liés aux distorsions temporelles. C'est la raison pour laquelle des extensions du modèle ont été développées afin de dépasser le problème de la reconnaissance de phonèmes. Parmi ces extensions, nous pouvons noter le MS-TDNN [Haffner et Waibel, 1992]. Nous considérons celui-ci comme un modèle hybride (voir plus loin) nous le présenterons donc dans le paragraphe consacré à ce genre de systèmes.

Le modèle originel du TDNN est performant dans le cas mono-locuteur mais l'est beaucoup moins dans le cas multi-locuteurs. J. Hampshire et A. Waibel dans [Hampshire II et Waibel, 1992] proposent un modèle de reconnaissance de phonèmes dans le cas multi-locuteurs. Ils utilisent pour cela un classifieur connexionniste modulaire pour la reconnaissance de formes multi-sources. Les sources sont représentées à la base par des TDNNs qui reconnaissent chacun des formes particulières (par exemple les trois plosives /b,d,g/) et pour un locuteur donné. Les sorties des TDNNs de base sont connectées à des unités qui sont les unités de sortie du réseau Meta-Pi. Ce dernier permet, en phase de reconnaissance, de « choisir » le TDNN susceptible de fournir la bonne réponse. L'architecture du réseau Meta-Pi est donnée figure 5.10.

5.6.2 Les réseaux récurrents à couches

Les réseaux récurrents à couches sont peu utilisés dans le traitement bas niveau de la parole. Ils possèdent cependant d'intéressantes propriétés à de plus hauts niveaux notamment dans le cas des grammaires de langages. Dans [Servan-Schreiber *et al.*, 1991], Servan-Schreiber *et al.* montrent qu'un réseau récurrent du type de celui de Elman est

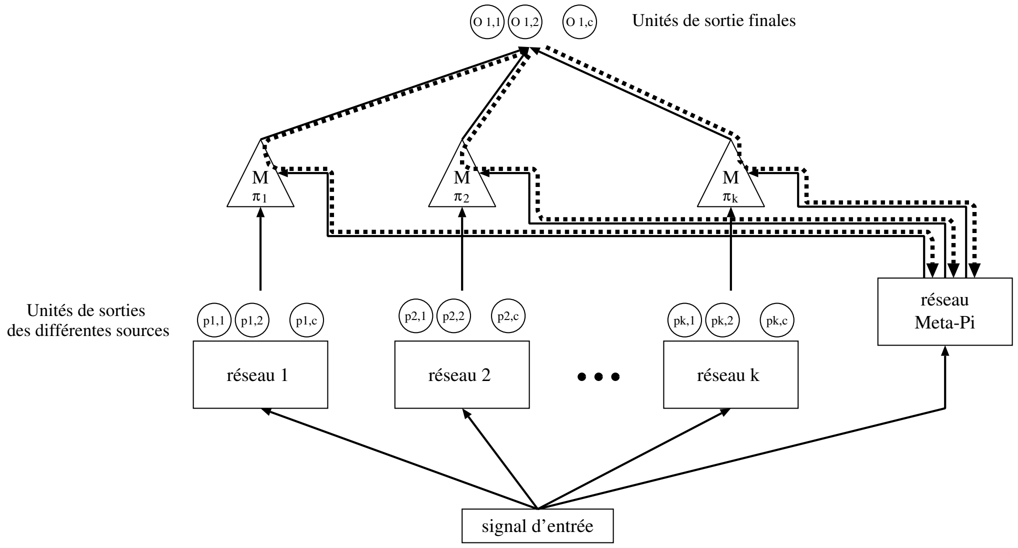


FIG. 5.10 – *Architecture du réseau Meta-Pi (d'après [Hampshire and Waibel, 1992])*

capable d'apprendre une grammaire d'états finis. Le réseau se comporte alors comme un automate d'états finis. Cependant, le réseau connexionniste semble être avantageux dans le cas de distinctions contextuelles précises. Par exemple, soit la grammaire L composée des trois terminaux A, B et C. L'automate décrivant la grammaire est donné à la figure 5.11.

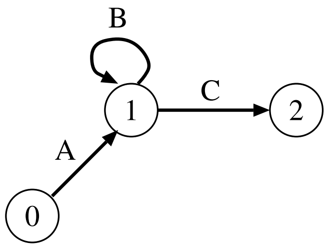


FIG. 5.11 – *Automate d'états finis*

Cet automate n'est pas capable de faire la différence entre le mot "ABC" et le mot "ABBC". Il acceptera les deux mots dans le langage de la même manière sans établir de différence entre les deux. Un réseau récurrent, lorsqu'il est correctement entraîné pour apprendre la grammaire d'un langage, modélise un « pseudo-automate » plus précis permettant de différencier les deux mots "ABC" et "ABBC" comme le montre la figure 5.12.

Ce genre de réseau est donc particulièrement bien adapté au traitement du langage. Les applications proposées, notamment dans [Servan-Schreiber *et al.*, 1991], le sont cependant pour des langages de taille relativement courte. Il n'est pas certain que ces réseaux puisse correctement modéliser des grammaires de langages relativement complexes comme celle des langues naturelles.

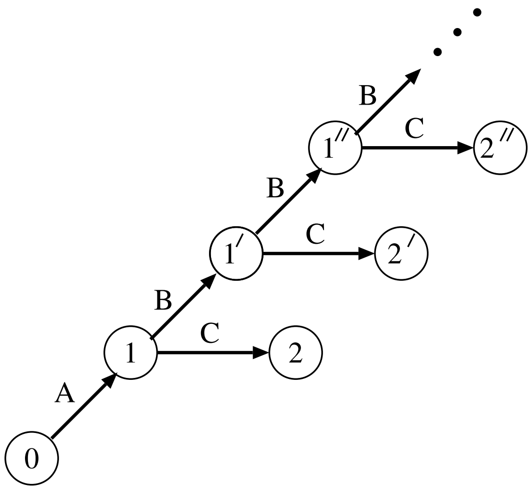


FIG. 5.12 – *Automate d'états finis représenté par un réseau récurrent*

5.7 Autres techniques et modèles hybrides

La classification de phonèmes par réseaux de neurones artificiels donne souvent de bons résultats. Cependant les capacités s'estompent lorsque l'on passe au niveau du mot par exemple. Le problème vient principalement du fait de l'alignement temporel qui n'est pas modélisé correctement par les réseaux classiques. Cette remarque a donné naissance à des modèles hybrides, en particulier à des systèmes combinant les réseaux neuronaux avec des techniques plus classiques d'alignement temporel comme la programmation dynamique. En outre, beaucoup de travaux ont vu le jour et ont encore cours concernant l'utilisation des modèles de Markov dans les réseaux. Ce paragraphe est consacré à ce genre d'approche. Nous décrivons d'abord et de façon succincte deux méthodes classiquement utilisées en parole et issues de la recherche opérationnelle d'une part (le DTW) et de la statistique d'autre part (les HMM). Ces méthodes sont désormais bien connues mais il nous a paru intéressant de les exposer ici car elles développent une fonctionnalité séquentielle. Enfin, nous citons quelques modèles hybrides impliquant ces méthodes hybrides.

5.7.1 Le recalage temporel dynamique (DTW)

L'algorithme de DTW consiste à mettre en correspondance du mieux possible deux séquences d'information acoustique codant un phonème, un mot ... Il s'agit du recalage temporel des deux formes. L'algorithme consiste à faire évoluer une fonction de coût à chaque progression dans les deux chaînes. La figure 5.13 montre une progression possible pour les deux chaînes.

Pour atteindre le point (i,j) plusieurs possibilités sont offertes :

– effectuer une substitution dans les deux chaînes en venant de (i-1,j-1) ;

– insérer un élément dans la chaîne à reconnaître venant de (i-1,j) ;

– supprimer un élément dans la chaîne à reconnaître en venant de (i,j-1).

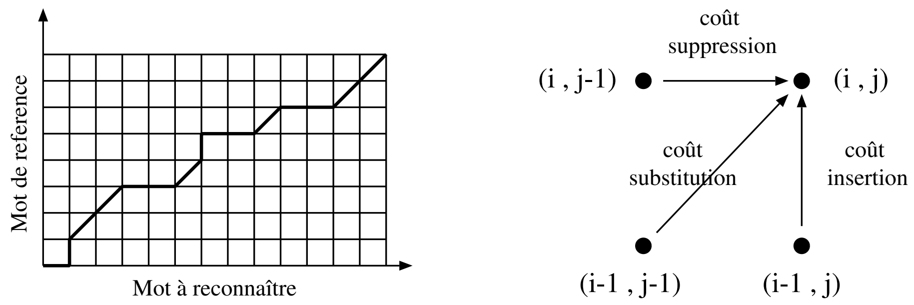


FIG. 5.13 – *Chemin de recalage entre deux chaînes (DTW)*

Chacune de ces opérations comporte un coût qui est ajouté au coût calculé au noeud (i-1,j-1). Le chemin de recalage est trouvé en sélectionnant à partir d'un point, celui des trois points pouvant être atteint dont le coût minimum calculé est le plus faible (voir figure 5.13).
Il s'agit alors de calculer le chemin de recalage pour chaque mot de référence dans la base de données et de choisir comme résultat le mot de référence dont la longueur du chemin associé est la plus petite.

5.7.2 Les modèles de Markov

L'algorithme de recalage temporel par le DTW effectue une comparaison forme à forme et nécessite donc une base de références importante afin de couvrir les différents cas possibles. Les méthodes stochastiques pallient ce problème en représentant un ensemble de formes de même type par un modèle stochastique et en représentant les formes par une suite d'états d'observation et de transitions entre ces états (fig. 5.14).

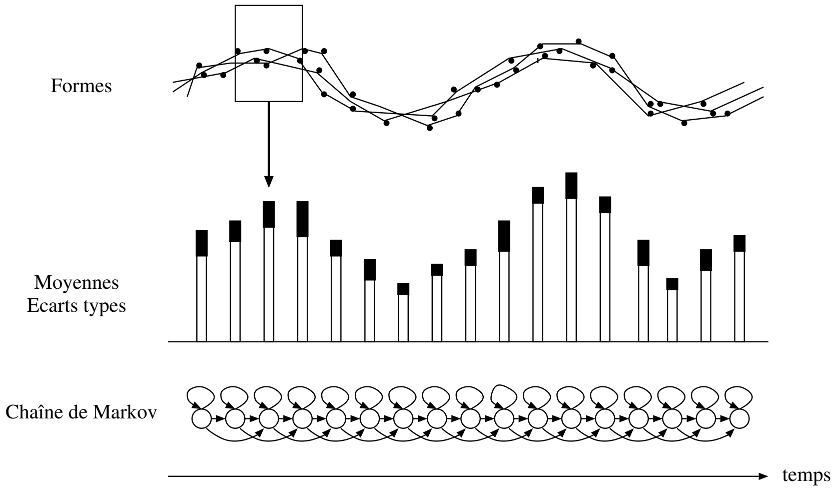


FIG. 5.14 – *Principe des modèles de Markov*

L'automate d'état est *markovien* dans le sens où la probabilité d'atteindre un état dépend de l'état courant et de l'état précédent. Classiquement en reconnaissance de la parole, on utilise des modèles de Markov *cachés* ou HMM (Hidden Markov Model). Le terme « caché » est employé ici pour préciser que la chaîne modélise un processus stochastique qui n'est pas observable mais qui affecte la séquence observée d'événements. La chaîne modélise donc deux processus stochastiques :

- pour chaque état, la probabilité d'apparition d'un événement (observation) ;
- la séquence d'états.

Un modèle de Markov caché est caractérisé par :

- Un nombre N d'états inter-connectés entre eux suivant une logique qui dépend de l'application mais qui consiste en général dans les applications de parole à orienter les transitions dans le sens du temps. L'états courant à l'instant t est repéré par q_t qui peut prendre ses valeurs dans l'ensemble des états $S = \{S_1, S_2, ..., S_N\}$.
- Un nombre M d'observations possibles pour chaque état. Une observation est un symbole pris dans un ensemble de symboles $V = \{v_1, v_2, ..., v_M\}$.
- Une distribution de probabilité de transition entre les états $A = \{a_{ij}\}$ où a_{ij} est définie par
$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq N. \tag{5.1}$$
- La distribution de probabilité d'observation d'un symbole dans un état j, $B = \{b_j(k)\}$ où
$$b_j(k) = P(v_k | q_t = S_j), 1 \leq j \leq N, 1 \leq k \leq M. \tag{5.2}$$
- La distribution de probabilité initiale $\pi = \{\pi_i\}$ où
$$\pi_i = P(q_1 = S_i), 1 \leq i \leq N. \tag{5.3}$$

Etant donné un modèle λ définit par A, B et π, la probabilité de la séquence d'observations $O = O_1 O_2 ... O_T$ est donnée par

$$P(O|\lambda) = \sum_{tousQ} P(O|Q, \lambda) P(Q|\lambda) \tag{5.4}$$
$$= \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) ... a_{q_{T-1} q_T} b_{q_T}(O_T) \tag{5.5}$$

Ce calcul nécessite un nombre d'itérations important pour trouver la probabilité d'observation car tous les chemins possibles sont explorés. Une procédure plus efficace en temps de calcul a été mise au point, il s'agit de la procédure *forward-backward* [Rabiner, 1990].

La suite d'états la plus probable pour une suite d'observations O et un modèle λ est donné par l'algorithme de Viterbi [Rabiner, 1990]. Enfin, l'ajustement des paramètres (probabilités de transition et d'observations) pour maximiser la probabilité d'observation pour un modèle ($P(O|\lambda)$) s'effectue grâce à l'algorithme de Baum-Welch décrit dans [Rabiner, 1990].

Dans ce qui précède, la transition d'un état à un autre dépend de l'état courant et de l'état précédent. Les modèles de Markov ont été étendus pour prendre en compte non seulement l'état précédent mais aussi l'état de deux pas de temps en arrière. Ces systèmes sont appelés modèle de Markov d'ordre 2 et semblent donner de meilleurs résultats.

5.7.3 Approches hybrides

Réseaux de neurones et modèles de Markov cachés

Les réseaux de neurones classiques sont efficaces en classification de formes mais ne tiennent pas compte de la nature dynamique de la parole. Or ceci est bien traité par les modèles de Markov cachés. D'où l'idée d'utiliser conjointement les réseaux de neurones et les chaînes de Markov.
Un perceptron correctement entraîné est équivalent à un estimateur de probabilités *a posteriori* [Bourlard et Wellekens, 1990].

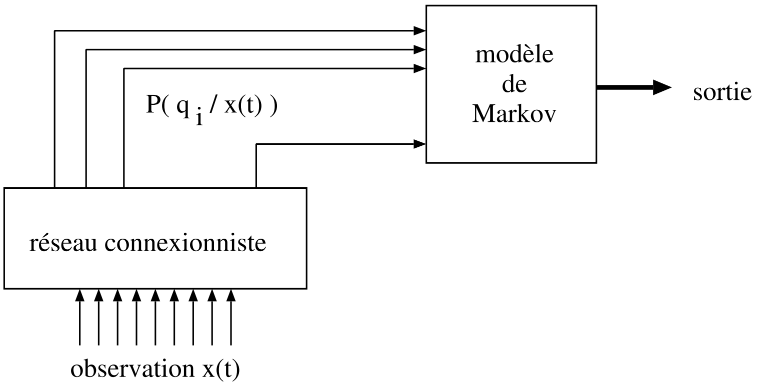


FIG. 5.15 – *Structure d'un modèle hybride particulier: NN → HMM*

En particulier, un perceptron (un réseau récurrent, ou un modèle RBF) peut être utilisé pour estimer les paramètres d'un modèle de Markov (fig. 5.15) [Bourlard et Morgan, 1990; Renals *et al.*, 1994; Bourlard *et al.*, 1995]. Une autre alternative (se rapprochant du modèle MS-TDNN) consiste à utiliser le perceptron (ou tout autre modèle connexionniste adapté) pour étiqueter les échantillons du signal de parole. Les séquences d'étiquettes sont alors modélisées par un HMM [Le Cerf, 1994].

Le MS-TDNN

Le TDNN a été combiné avec les modèles de Markov [Dugast *et al.*, 1994] mais aussi avec des techniques de programmation dynamique. L'architecture MS-TDNN [Haffner et Waibel, 1992] (Multi State TDNN) peut être classée selon les auteurs comme une méthode hybride dans la mesure où elle incorpore un algorithme de programmation dynamique. En effet, la méthode consiste à utiliser une architecture de type TDNN comme décrite dans [Waibel *et al.*, 1989]. Cependant la couche de sortie est supprimée pour n'utiliser que la dernière couche cachée permettant de donner à chaque pas de temps les scores pour chaque état possible. Par exemple, dans le cas de la reconnaissance des trois plosives /b,d,g/ la dernière couche cachée comporte des colonnes de trois neurones correspondant à trois états distincts. A un instant donné, ces neurones, par leur valeur, indiquent dans quel état se trouve le système. Ainsi, pour la prononciation d'une plosive on obtient une séquence d'états que l'on compare par programmation dynamique à des séquences de référence. Le principe se généralise pour d'autres types de sonorités comme des mots. Dans ce dernier cas, les résultats sont honorables et aboutissent à des scores compris entre 90% et 95%.

5.8 Conclusion

Le connexionnisme a naturellement été appliqué à la reconnaissance de la parole au vu des bonnes capacités en classification de formes. Cependant, les modèles connexionnistes n'ont pour l'instant pas surpassé les HMM. Ceci est en grande partie due à leur relative inadéquation au problème du traitement séquentiel de l'information. C'est pourquoi une grande tendance dans le domaine du neuromimétisme consiste à étendre les systèmes classiques ou à développer de nouveaux modèles pour prendre en compte les variabilités inhérentes à la parole. A partir des modèles statiques tel le perceptron, les structures ont été adaptées pour tenir compte de l'aspect temporel sans pour autant changer complètement l'architecture des réseaux. Des modèles hybrides ont été conçus afin d'additionner les qualités de chacun des modèles fusionnés mais sans réellement homogénéiser l'architecture.
Enfin, une autre voie consiste à développer de nouvelles architectures avec un bagage important de modèles adaptés à différentes situations. La neurobiologie dans ce cas, peut être une source d'inspiration intéressante pour cette voie. C'est la raison pour laquelle nous avons brièvement présenté l'axe cortical auditif en tête de ce chapitre. Il est bien évident que cette voie de recherche n'a pas autant d'acquis et d'expérience que la voie utilisant des modèles classiques. Il nous a paru cependant important de l'explorer en particulier dans le cas de la reconnaissance de la parole.

Chapitre 6

TOM et la reconnaissance de la parole

« Supposons donc qu'au commencement l'âme est ce qu'on appelle une table rase, vide de tous caractères, sans aucune idée, quelle qu'elle soit. Comment vient-elle à recevoir des idées ? D'où puise-t-elle tous ces matériaux qui sont comme le fond de tous ses raisonnements et de toutes ses connaissances ? A cela je reponds en un mot, de l'expérience : c'est là le fondement de toutes nos connaissances, et c'est de là qu'elles tirent leur première origine. Les observations que nous faisons sur les objets extérieurs et sensibles, ou sur les opérations intérieures de notre âme, que nous apercevons et sur lesquelles nous réfléchissons nous-même, fournissent à notre esprit les matériaux de toutes ses pensées. Ce sont là les deux sources d'où découlent toutes les idées que nous avons, ou que nous pouvons avoir naturellement. »

John Locke, Essai philosophique concernant l'entendement humain, II, 1 à 4, 1690.

6.1 Introduction

Le traitement automatique de la parole est un des grands défis de la recherche en Intelligence Artificielle. En particulier, il recouvre plusieurs champs d'investigation qui ont chacun leur importance dans le processus de communication utilisé par l'homme : traitement du signal acoustique, reconnaissance des formes, traitement lexical, analyse syntaxique, traitement sémantique et pragmatique à un niveau beaucoup plus élevé. Tous ces problèmes revêtent un caractère dynamique mais. pour des raisons évidentes, nous

nous limitons dans le cadre de cette thèse, au traitement acoustique tout en considérant le reste des problèmes comme des applications tout à fait intéressantes dans la perspective d'évolution de notre modèle.
Les différentes applications que nous développons dans ce chapitre mettent en lumière les qualités et les défauts du modèle tel qu'il existe actuellement. En particulier, nous nous focalisons sur la capacité de généralisation des différents systèmes construits à partir de cartes TOM.
Il est bien évident qu'il est difficile de prétendre faire mieux que des modèles largement étudiés et extensivement testés comme les modèles de Markov. Ceci parce que le modèle que nous proposons est relativement nouveau et, bien que ses principes nous paraissent tout à fait pertinents pour la reconnaissance de la parole, son développement et sa mise au point constituent un travail de longue haleine et nécessitent encore des adaptations.

Nous avons effectué les expérimentations sur des mots parlés français ou américains et plus précisément les chiffres. Avant de décrire précisément les différents tests, nous discutons du prétraitement du signal et du codage des entrées que nous avons utilisés. Ceux-ci influencent de façon non négligeable le fonctionnement des réseaux et constituent un des problèmes sur lequel il convient de réfléchir, nous en discuterons au chapitre de conclusion.
Ce chapitre s'articule selon quatre axes. Le premier paragraphe présente les résultats obtenus avec la carte TOM de base et isolée. Le second discute de la version supervisée par une carte associative. Le troisième introduit les expérimentations sur le modèle fonctionnant à l'heure actuelle avec des paramètres de distorsion locaux. Enfin, dans le dernier paragraphe, nous donnons quelques résultats obtenus avec les chaînes de Markov et le TDNN afin de comparer ceux-ci avec le modèle TOM.

6.2 Le pré-traitement du signal de parole

Le signal de parole pur est difficilement interprétable et utilisable par un système automatique. La grande quantité d'information qu'il véhicule se trouve confinée dans un espace de représentation (temps x amplitude) relativement réduit. C'est la raison pour laquelle ce signal pur n'est quasiment jamais directement utilisé, une représentation de type (temps,fréquence,énergie) lui est préférée (spectrogramme). On trouvera dans [Calliope, 1989] une description détaillée des différentes méthodes de prétraitement et d'analyse du signal. Pour notre part, nous décrivons ici l'analyse MFCC (Mel-Frequency Cepstrum Coefficients) que nous utiliserons dans les expériences menées sur TOM.

6.2.1 Analyse MFCC

Le signal qui est obtenu en sortie de l'appareil phonatoire résulte de l'inter-modulation source/conduit et ne représente donc pas uniquement la modulation par les conduits vocal et nasal. C'est la raison pour laquelle on sépare en général le signal source excitateur de la contribution des conduits considérés comme des filtres impulsionnels. L'élimination du signal excitateur (ou Pitch) conduit au *cepstre* et c'est sur ce signal qu'une transformée

de Fourier est effectuée pour le calcul du spectrogramme.
Des techniques de filtrage plus ou moins complexes peuvent encore être ajoutées à cette dernière analyse. En particulier, l'analyse MFCC permet d'obtenir des vecteurs de coefficients robustes au changement de locuteur [Davis et Mermelstein, 1980]. Cette méthode utilise un banc de N filtres triangulaires et produit M coefficients. Leur formulation est donnée par l'équation ci-dessous :

$$MFCC_i = \sum_{k=1}^{N} X_k.cos[i.(k - \frac{1}{2}).\frac{\pi}{N}] \tag{6.1}$$

où k est le numéro du filtre, X_k le logarithme de l'énergie calculée sur le $k^{ème}$ filtre et i varie de 1 à M. Pour sa relative indépendance aux variations inter-locuteur, nous utiliserons cette méthode d'analyse.

6.2.2 Codage de la parole pour le modèle TOM

Dans la suite, toutes les expériences décrites utilisent le même processus de codage de la parole modulo quelques paramètres que nous définirons. Nous utiliserons une analyse MFCC à 12 coefficients. Le signal est échantillonné à 16kHz et la fenêtre sur le signal pour le calcul des coefficients contient 512 points. Le décalage de la fenêtre entre deux calculs est de 4 ms (64 points). Une première remarque peut être faite concernant le choix de ces paramètres. Les valeurs données ci-dessus sont des valeurs classiquement utilisées, en particulier dans notre laboratoire[6]. Nous savons qu'elles influencent l'analyse spectrale de la parole [Calliope, 1989], cependant, parce que le modèle que nous présentons possède un nombre de paramètres non négligeables, nous fixons *a priori* les paramètres de traitement du signal aux valeurs données ci-dessus.

Le problème majeur auquel nous sommes confrontés concerne la distorsion temporelle du signal de parole, cette distorsion étant répercutée sur le résultat du prétraitement. Il est bien évident que plus le signal comporte de distorsions par rapport aux signaux appris par un système quelconque plus ce système aura de mal à identifier le signal. En particulier, plus une séquence acoustique est longue, plus elle contiendra statistiquement de distorsions. C'est pourquoi nous n'utiliserons pas directement les séquences de vecteurs produites par l'analyse MFCC mais des séquences de vecteurs moyens calculés sur toute la longueur des séquences générées par le prétraitement (fig. 6.1).

Les meilleurs résultats ont été obtenus avec des groupes de 3,4 ou 5 vecteurs sans recouvrement. De cette façon, la taille des séquences est réduite et l'information est moyennée, assouplissant ainsi les distorsions.

6. Crin-CNRS / Inria Lorraine, équipe RFIA

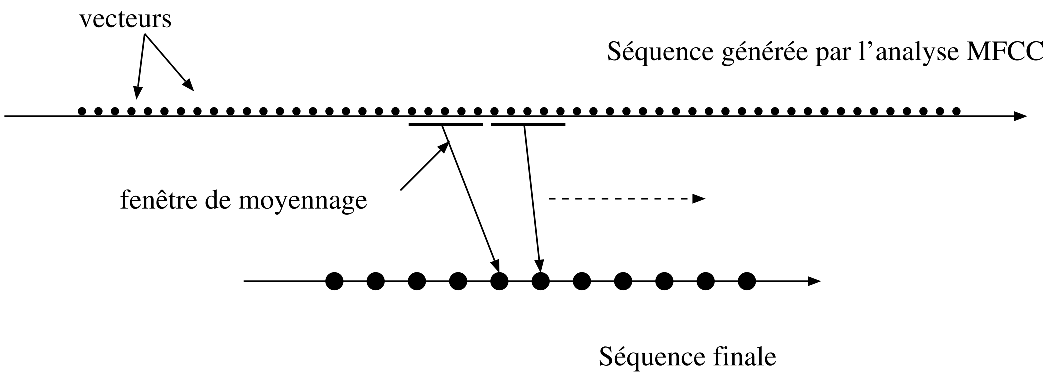


FIG. 6.1 – *Génération des séquences utiles*

6.2.3 Les corpus utilisés

Nos tests ont été effectués à partir de chiffres parlés isolés. Deux bases de données sont utilisées :

- Une base de chiffres en français prononcés par un seul locuteur masculin. Une centaine de chiffres (0 à 9) composent la base. 60 sont utilisés pour l'apprentissage et 40 pour le test.

- Une base de chiffres en anglais, TI DIGIT (base de données de chiffres parlés multi-locuteurs conçue par la société américaine Texas Instrument). Au total, 111 locuteurs masculins prononcent deux occurrences de chaque chiffres plus le mot « o » (variante de « zero »). 55 locuteurs servent pour l'apprentissage et 56 pour le test.

6.3 L'apprentissage des masques des super-unités

Suivant l'algorithme d'apprentissage présenté au chapitre 4, les masques des super-unités doivent être définis avant l'apprentissage des séquences. Les trois méthodes auto-organisatrices introduites comme moyens de base pour évaluer ces masques (carte topographique de Kohonen [Kohonen *et al.*, 1984], réseau Neural-Gas [Martinetz et Schulten, 1991] et réseau Growing Neural-Gas [Fritzke, 1995]) n'ont pas fourni de résultats significativement différents entre elles. Le choix de tel ou tel réseau compétitif s'opère plutôt sur la base de l'architecture. En outre, le modèle Growing Neural-Gas introduit des neurones sur la carte, là où des confusions de vecteurs sont constatées. Dans l'algorithme original [Fritzke, 1995], l'instant d'ajout d'un neurone est donné *a priori* mais nous pouvons imaginer et envisager que cet instant soit contrôlé par un mécanisme d'action indiquant quand et où le réseau doit être plus précis. Cet algorithme nous semble donc plus adapté à des mécanismes de rétro-actions, c'est-à-dire des mécanismes où des cartes ou des réseaux de neurones de haut niveau agissent sur la carte de codage de l'information sensorielle. La comparaison entre ces différents modèles se situe donc plus sur un plan qualitatif que quantitatif.

Dans l'approche que nous avons tout d'abord mené, nous avons utilisé l'algorithme de Kohonen pour le calcul des poids feed-forward. Les tests ont été conduit principalement avec des connexions voisines et latérales à une dimension (Kohonen linéaire). Une géographie topographique à deux dimensions a apporté une amélioration substantielle des résultats. Nous le verrons au paragraphe 6.6.

6.4 Tests préliminaires sur le modèle de base

Nous présentons ici les premiers tests effectués sur une carte TOM pure, c'est-à-dire isolée et non inter-connectée avec une autre carte ou un autre système connexionniste. Chaque fin de séquence est cependant étiquetée avec le symbole du chiffre correspondant à la séquence. Les deux paramètres de distorsion temporelle *decay* et *thres_ act* sont fixes pendant toute la durée de l'apprentissage est valent respectivement $0,90$ et $0,75$.

6.4.1 Généralisation sur les séquences

Dans un premier temps, nous examinons la combinatoire de la carte pour un nombre croissant d'occurrences d'un même chiffre. Il s'agit du chiffre «zéro» en français. La base de donnée a été construite par nos soins et l'enregistrement effectué en situation bruitée (bureau) avec un micro de faible qualité. 170 occurrences du chiffre zéro sont apprises par la carte. Le nombre de séquences générées en fonction du nombre de séquences prononcées est donné sur la figure 6.2.

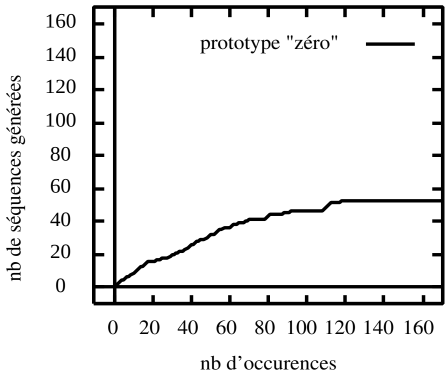


FIG. 6.2 – *Génération de séquences pour un chiffre parlé (zéro français)*

On voit que l'évolution n'est pas linéaire mais plutôt logarithmique et donc que la carte permet une généralisation par rapport aux séquences apprises. Les séquences acoustiques sont de longueurs variables et sont prononcées à des distances variables du micro. Il en résulte une bonne capacité de généralisation de la carte pour un même locuteur en situation réelle.

6.4.2 Reconnaissance de chiffres

L'architecture utilisée ici comporte une carte TOM de base à paramètres de distorsions globaux fixés *a priori* pendant toute la durée de l'apprentissage. Les séquences acoustiques apprises sont étiquetées par des «neurones grand-mère». Pour des raisons d'homogénéïté et dans le souci d'évolutions futures, nous modélisons ces neurones grand-mère par des super-unités. Le système général comporte alors deux cartes : une carte de traitement et de codage de séquences ainsi qu'une carte «symbolisant» les chiffres. Cette dernière contient 10 super-unités, une pour chaque chiffre. Les connexions de la carte des séquence vers la carte des chiffres sont feed-forward (liens inter-cartes) et repèrent toutes les séquences. L'algorithme de création de ces liens est le suivant :

1. Choisir un chiffre dans le corpus.

2. Passer la séquence acoustique pour générer une propagation d'activité.

3. Tisser les liens binaires entre les unités encore actives sur la carte des séquences avec la super-unité correspondant au chiffre prononcé sur la carte des chiffres. Ces liens constituent les champs récepteurs feed-forward des super-unités.

4. Retourner au 1.

Cet algorithme relie les fins de séquence aux super-unités indiquant l'identité des chiffres. L'activité spatiale de la super-unité est calculée grâce à son champ récepteur sur les séquences selon la formule suivante :

$$SA(SU_i) = \sum_{j=1}^{N} act(U_{ij}) \tag{6.2}$$

où $SA(SU_i)$ est l'activité spatiale de la super-unité SU_i, N est le nombre de liens feed-forward sur la super-unité i et $act(U_{ij})$ est l'activité de l'unité j sur la carte des séquences et se trouvant dans le champ récepteur de SU_i. Ce calcul traduit le fait qu'une super-unité codant un chiffre s'active si la propagation d'activité aboutit à au moins une des unités fin de séquences qu'elle connecte. L'architecture est résumée sur la figure 6.3.

Tests en français

Le corpus d'apprentissage contient 60 chiffres, il est passé une seule fois pour apprendre les liens intra-carte (cf. l'algorithme de base donné au chapitre 4). Le temps d'apprentissage est de l'ordre de 35 secondes sur une station de travail SUN Sparc 20. Les pourcentages ne sont pas réellement significatifs dans la mesure ou le corpus est petit, cependant, nous les donnons à titre indicatif dans le tableau suivant (6.1). Un seul chiffre n'est pas reconnu dans le corpus de test. Les paramètres du réseau ont les valeurs suivantes : $\alpha = 1.0$, $\beta = 1.0$, *thres_ act*$= 0.75$, et *decay*$= 0.90$.

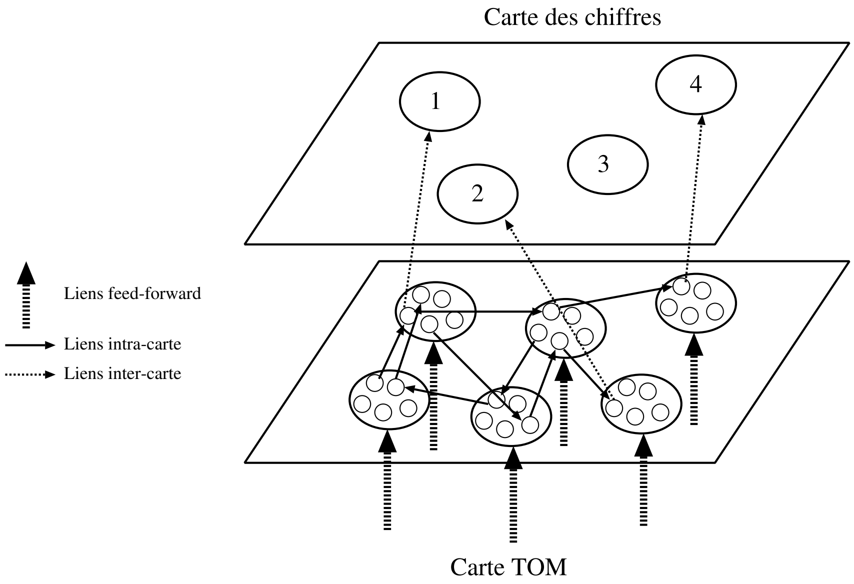


FIG. 6.3 – *Architecture de base pour la reconnaissance de chiffres*

Modèle	Train	Test
français	100,00%	97,50%

TAB. 6.1 – *Résultats obtenus avec une simple carte TOM sur le corpus français*

Tests en anglais

Nous testons maintenant la carte avec le corpus de chiffres américains issus de la base TI DIGIT.
L'expérience se décompose en deux tests.

- Le premier test (test1) utilise 10 locuteurs prononçant chacun les dix chiffres pour l'apprentissage, et les même locuteurs prononçant une nouvelle fois chacun les dix chiffres pour le test.

- Le second test (test2) utilise 10 locuteurs prononçant chacun deux occurrences de chaque chiffre (20 chiffres au total) pour l'apprentissage, et 10 autres locuteurs prononçant chacun deux nouvelles occurrences de chaque chiffre.

L'architecture est identique à celle présentée au paragraphe suivant. Contrairement au test français qui utilisait l'algorithme de Kohonen pour l'apprentissage des liens feed-forward, nous utilisons ici l'algorithme growing neural-gas. Nous rappelons que le principe de cet algorithme consiste à introduire de nouveaux neurones là ou les confusions sont les plus fortes. Dans notre cas, il s'agit d'ajouter non pas des neurones mais des super-unités. Suivant l'algorithme donné dans [Fritzke, 1995], une super-unité est ajoutée tous les L pas de temps d'apprentissage. Les liens topologiques sont coupés lorsque leur âge (leur fréquence d'utilisation) est supérieur à un seuil MA.
Nous présentons, dans le tableau 6.2, les pourcentages de reconnaissance obtenus avec

différentes valeurs de L et MA et les valeurs suivantes des paramètres de la carte : $\alpha = 1.0$, $\beta = 1.0$, *thres_ act*= 0.75, and *decay*= 0.90.

	L	MA	Nb cycles	Nb super-unités	Reconnaissance
Test 1	1000	100	50	103	85%
	200	50	50	154	89%
	500	100	50	204	93%
Test 2	200	50	50	154	76,5%
	500	100	50	211	80,5%
	700	100	50	230	82,5%

TAB. 6.2 – *Résultats obtenus avec une simple carte TOM et le corpus américain*

6.5 Un système supervisé pour la reconnaissance de chiffres parlés

Dans ce paragraphe, nous présentons les résultats obtenu sur l'architecture supervisée incluant ce que nous avons appelé au chapitre 4 une carte associative. En réalité, il s'agit plutôt ici de tester l'adéquation à notre modèle de l'algorithme LASSO [Midenet et Grumbach, 1990] présenté au chapitre 2. En phase d'apprentissage, cet algorithme ce comporte comme une carte auto-organisatrice. En phase de reconnaissance, le neurone vainqueur est choisi sur la base d'un sous-espace de l'espace de ses entrées. Il envoie ensuite une impulsion sur le sous-espace restant (cf. chap. 2) afin de compléter l'information manquante.

6.5.1 Architecture

Dans notre cas, quatre cartes TOM composent le système supervisé de reconnaissance. A la base, une carte (CSIG) code et apprend les séquences acoustiques (comme au paragraphe précédent), une carte de détection de séquences (CSEQ) permet d'évaluer quantitativement les séquences d'activité mises en jeu, une carte de chiffres (CC) joue le rôle de professeur et une carte associative de type LASSO (CA) apprend les couples <chiffre,séquence>. Cette architecture est représentée sur la figure 6.4.

Les super-unités de la carte (CSEQ) fonctionnent comme des filtres sur les séquences. Leur activité n'est donc pas binaire mais à valeurs réelles.

La carte CA est utilisée ici comme une carte de Kohonen. Seuls les liens feed-forward et de voisinage sont donc mis à contribution, nous n'exploitons pas la fonctionnalité temporelle de la carte. Les vecteurs que classe la carte sont de dimension $Nc + Ns$ où Nc est le nombre de super-unités sur la carte des chiffres (nombre de chiffres, $Nc = 10$) et Ns est le nombre de super-unités sur la carte des séquences (nombre de séquences extraites). En phase d'apprentissage, le système active la super-unité correspondant au chiffre dont la séquence acoustique est présentée au réseau, le vecteur de dimension Nc est alors du type (0,0,1,0,0,0,0,0,0,0) pour le chiffre deux par exemple.

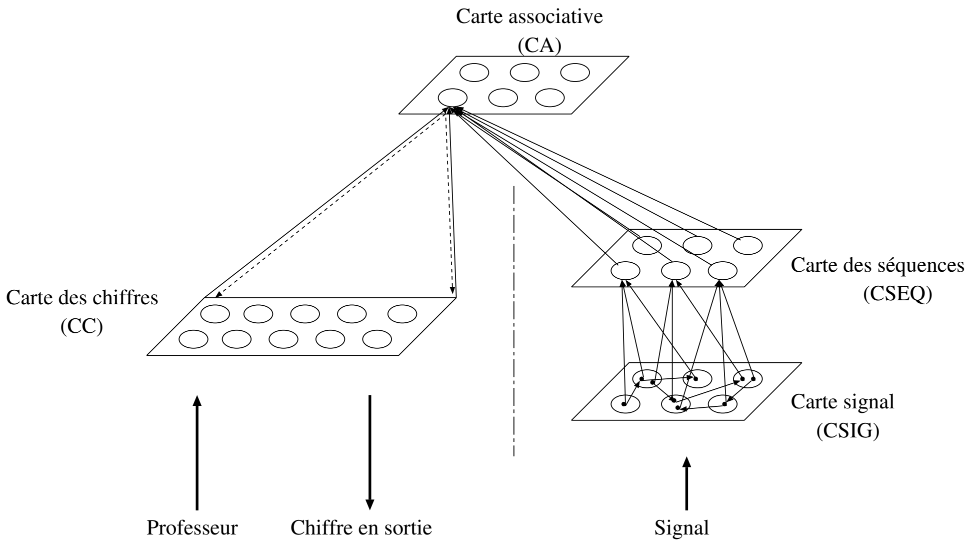


FIG. 6.4 – *Architecture supervisée pour l'apprentissage et la reconnaissance de chiffres parlés*

6.5.2 Les expériences avec la carte associative

Les indices acoustiques (vecteurs MFCC) sur la carte CSIG sont appris classiquement avec l'algorithme de Kohonen. La phase d'apprentissage des poids feed-forward de la carte associative CA est désormais plus longue puisque le corpus d'apprentissage doit être présenté plusieurs fois pour faire converger ces poids. L'ordre de grandeur de la durée d'apprentissage varie en fonction du corpus. Il peut aller jusqu'à plusieurs heures pour un très gros corpus sur une station de travail SUN Sparc 20. Afin de pouvoir comparer à ce qui précède et ce qui va suivre, nous testons d'abord sur le corpus français. Les résultats sont donnés au tableau 6.3.

Train	Test
96,66%	90,00%

TAB. 6.3 – *Résultats obtenus avec une carte associative (LASSO). Corpus français mono-locuteur*

Nous donnons au tableau 6.4, les résultats obtenus à partir d'un corpus d'apprentissage multi-locuteurs contenant 11 ou 55 locuteurs prononçant chacun deux occurrences de chaque chiffre (plus deux occurrences du mot «o» représentant le chiffres zéro en anglais). Le corpus de test contient 56 autres locuteurs prononçant également deux occurrences de chaque chiffres soit un total de 1232 mots.

Nb Loc	Train	Test
11	99,58%	75,81%
55	93,14%	88,47%

TAB. 6.4 – *Résultats obtenus avec une carte associative (LASSO). Corpus américain multi-locuteurs*

6.6 Modèle à paramètres de distorsion locaux

Nos travaux actuels consistent en la mise au point de cartes TOM avec des paramètres de distorsion locaux aux unités et appris. Nous pensons que ces paramètres peuvent apporter une amélioration significative dans les performances du modèle. L'algorithme de modification de ces paramètres (vu au chapitre 4) est loin d'être un algorithme universel et mérite sans doute d'être amélioré en fonction des résultats obtenus. En outre, le choix de modifier tel ou tel paramètre (*decay* et *thres_act*) sur telle ou telle unité a été fait empiriquement et intuitivement. Il demandera vraisemblablement des aménagements.

Dans les expériences concernant cet apprentissage des paramètres de distorsion temporelle, nous utilisons un système similaire à celui présenté au paragraphe 6.4 : une carte pour le codage des séquences et une carte contenant les super-unités détectrices de l'activation des séquences. Cependant, contrairement au paragraphe 6.4, le nombre de super-unités sur la carte détectrice de séquences est laissé libre : à chaque nouvelle séquence créée sur la carte des séquences, une nouvelle super-unité est créée et reliée sur la carte détectrice.

6.6.1 Première phase

La première vague de tests concernant les paramètres de distorsion appris a été menée avec des liens de voisinage linéaires (Kohonen 1D). 150 super-unités sont utilisées. Les valeurs initiales de *decay* et *thres_act* sont respectivement 0.90 et 0.53.
Le premier test utilise le corpus français (60 chiffres d'apprentissage, 40 de test). Le taux de reconnaissance sur le corpus d'apprentissage et de test sont donnés dans le tableau (6.5).

Train	Test
100,00%	97,50%

TAB. 6.5 – *Résultats obtenus avec une carte TOM à paramètres de distorsion locaux appris . Première phase, corpus français mono-locuteur*

Les résultats obtenus sur ce corpus sont équivalents à ceux obtenus avec la carte simple. Nous évaluons maintenant les performances avec le corpus d'apprentissage contenant 10 locuteurs (10 chiffres chacun soit 100 chiffres au total) et le corpus de test contenant les 10 mêmes locuteurs (10 autres chiffres chacun soit 100 chiffres). Les résultats sont contenus dans le tableau 6.6.

Nb cycles	Train	Test
1	88,00%	73,00%
2	92,00%	87,00%
3	100,00%	85,00%

TAB. 6.6 – *Résultats obtenus avec une carte TOM à paramètres de distorsion locaux appris. Première phase, corpus américain multi-locuteurs*

6.6.2 Seconde phase, amélioration

Nous avons poursuivi les tests en ajustant de différentes manières les paramètres de la carte, en particulier *decay* et *thres_act*. Nous avons de plus noté une amélioration des performances en utilisant une carte de Kohonen à deux dimensions pour les liens de voisinage et un nombre plus important de super-unités (400) par rapport aux tests précédents. Enfin, il se trouve que le critère de sélection des séquences actives n'était peut-être pas idéal dans les tests précédents. En effet, une séquence était déclarée active lorsque l'activité se propageait jusqu'aux unités fin de séquence. Une procédure de sélection de séquences actives améliorant nettement les résultats consiste à choisir la séquence ayant le plus d'unités actives proportionnellement à sa longueur. Le choix de cette procédure, l'apprentissage des masques spatiaux avec une carte de Kohonen bi-dimensionnelle ainsi que les valeurs des paramètres *decay* et *thres_act* respectivement à 0.96 et 0.63 ont sensiblement amélioré les résultats qui sont donnés dans le tableau 6.7.
Dans le test, dit «pseudo multi-locuteurs» dans le tableau, avec 10 locuteurs d'apprentissage (100 chiffres) et les 10 mêmes locuteurs de test (100 autres chiffres) aucune erreur n'est constatée. Dans le test complètement multi-locuteur nous utilisons 55 locuteurs prononçant chacun deux occurrences de chaque chiffres et 17 locuteurs de tests prononçant 2 occurrences de chaque chiffres également [Durand et Alexandre, 1996].

Type test	Train	Test
Pseudo multi-locuteur	100,00%	100,00%
Multi-locuteur	98,75%	98,52%

TAB. 6.7 – *Résultats obtenus avec une carte TOM à paramètres de distorsion locaux appris. Seconde phase, corpus américain multi-locuteurs*

6.7 Comparaison avec d'autres approches

Nous présentons dans ce paragraphe les résultats obtenus sur des applications du même type que celles présentées pour TOM ceci avec deux méthodes classiques utilisées en reconnaissance de la parole : le TDNN et les chaînes de Markov. Nous l'avons dit tout au long de cette thèse, le but de nos travaux à court terme n'était pas directement de concurrencer ces modèles. Il s'agissait de concevoir une architecture connexionniste temporelle d'inspiration neurobiologique et adaptée à tout type de problème incluant une dimension temporelle. Cependant, nous pouvons examiner les résultats fournis par ces deux techniques afin de nous situer par rapport à elles et prévoir les orientations futures. La première expérience consiste à utiliser un TDNN sur le corpus français mono-locuteur. La seconde, consiste à utiliser des chaînes de Markov du premier et du second ordre sur le corpus américain.

6.7.1 Utilisation du TDNN en reconnaissance de chiffres

Plusieurs types de TDNN ont été testés. La différence d'architecture réside dans le nombre de couches cachées (une ou deux) et dans le nombre de neurones dans ces couches cachées. Le TDNN utilise une fenêtre de taille T fixée en entrée, il nous a donc fallu empiriquement choisir une taille de séquence et adapter chacune des séquences acoustiques à cette contrainte. Lorsque la longueur d'une séquence dépasse la taille T, elle est tronquée, seul les T premiers vecteurs sont utilisés. Lorsque la taille de la séquence est plus petite que T, des vecteurs nuls sont ajoutés. Le choix de T s'est effectué sur la base de la taille moyenne des séquences, c'est-à-dire $T = 15$. Deux modèles sont définis, ils ont tous les deux le même format d'entrée, à savoir 15 vecteurs de dimension 12 et une couche de sortie comprenant 10 neurones (un pour chaque chiffre). Chaque modèle se différencie de la façon suivante :

- Modèle 1 : une seule couche cachée (13 vecteurs de 10 neurones chacun).
- Modèle 2 : deux couches cachées, première couche (13 vecteurs de 11 neurones), deuxième couche (8 vecteurs de 10 neurones).

Les deux réseaux sont des TDNN non optimisés et ont été développés sur le simulateur SNNS (Stuttgart Neural Network Simulator). Les résultats obtenus après un apprentissage de 60.000 cycles et un apprentissage de 160.000 cycles sont donnés dans les deux tableaux 6.8 et 6.9.

Modèle	Train	Test
modèle 1	60,00%	53,33
modèle 2	50,00%	27,50

TAB. 6.8 – *Résultats obtenus avec 60.000 cycles sur les TDNN*

On voit que le TDNN n'est pas du tout adapté au traitement de mots, il n'égale donc pas le modèle TOM dans sa version la plus simple ($97,50\%$ sur le même test).
Ceci était prévisible au vu des travaux effectués pour l'extension du TDNN à ces problèmes [Haffner et Waibel, 1992]. Nous aurions pu envisager l'expérience sur un système plus complexe à base de TDNN comme un réseau Méta-Pi ou un MS-TDNN. Cependant,

Modèle	Train	Test
modèle 1	60,00%	60,00%
modèle 2	50,00%	25,00%

TAB. 6.9 – *Résultats obtenus avec 160.000 cycles sur les TDNN*

le temps d'apprentissage de l'architecture TOM (modèle simple) sur une station de travail SUN Sparc20 est de l'ordre de la demi-heure alors qu'elle est de plus de 5 heures pour les TDNN présentés ici. Une architecture plus complexe à base de TDNN prendrait vraisemblablement plus de temps pour apprendre. Il ne nous a donc pas paru nécessaire de tester ces modèles.

6.7.2 Les modèles de Markov

Nous présentons ici les résultats obtenus avec des modèles de Markov cachés du premier et du second d'ordre. Les modèles de Markov sont actuellement les modèles aux performances les plus remarquables, prouvant du même coup leur adéquation au traitement du phénomène temporel. Le corpus utilisé provient de la base TI DIGIT avec cinq corpus d'apprentissage contenant respectivement 11, 19, 27, 43 et 55 locuteurs, le corpus de tests contient 56 locuteurs. Le tableau 6.10 donne les résultats obtenus avec un modèle de Markov du premier ordre, le tableau 6.11 donne les résultats pour un modèle d'ordre 2.

Nb de locuteurs d'apprentissage	9 boucles	11 boucles
11	94,48%	95,21%
19	96,75%	97,24%
27	97,89%	97,80%
43	98,29%	98,37%
55	98,62%	98,70%

TAB. 6.10 – *Reconnaissance avec des modèles de Markov d'ordre 1.*

Nb de locuteurs d'apprentissage	9 boucles	11 boucles
11	95,37%	95,53%
19	97,80%	97,64%
27	98,29%	97,97%
43	98,53%	99,02%
55	98,70%	99,26%

TAB. 6.11 – *Reconnaissance avec des modèles de Markov d'ordre 2.*

Les performances que l'on trouve ici, notamment avec 43 et 55 locuteurs, tout en étant supérieures à nos performances, ne sont pas si éloignées. On peut donc penser que dans un futur proche, notre système pourra égaler les performances des HMMs. En effet, ceux-ci ont été extensivement développés et testés sur une plage de temps relativement longue par rapport à celle dont nous disposions pour la thèse [Bourlard *et al.*, 1995].

6.8 Conclusion

A l'heure du bilan chiffré, nous pouvons faire les remarques suivantes pour chacun des types de réseaux présenté dans ce chapitre.

- La carte TOM isolée montre une bonne capacité de généralisation lorsque le corpus n'est pas caractérisé par une multitude de variations (cas du corpus français). Elle nous paraît parfaitement adaptée pour des tâches de reconnaissance limitées à un seul locuteur pour peu que l'apprentissage ait été exécuté sur ce même locuteur. En phase multi-locuteur, la carte conserve des performances honorables lorsque les locuteurs de test ont été appris au préalable. Ces performances sont en baisse lorsque les locuteurs de test n'ont pas été appris.

- Les performances enregistrées avec le modèle supervisé (carte associative) s'approchent des 90% pour le gros corpus américain (55 locuteurs d'apprentissage et 56 de test). Ce résultat est un peu décevant dans la mesure où le système ayant été rendu supervisé, nous nous attendions à de meilleurs résultats. Ceci ne remet cependant pas en question l'idée des axes de traitement reliés par une carte associative. Le modèle LASSO n'est peut-être pas adapté pour relier ces axes. D'autres algorithmes traitant des liens inter-carte seront peut-être plus profitables.

- Le modèle à paramètres de distorsion locaux, après avoir été testé dans diverses configurations, a donné des résultats tout à fait intéressants par rapport aux résultats obtenus avec les HMMs, les modèles les plus performants actuellement. Ce mode, avec tolérances distinctives le long des séquences, semble donc améliorer sensiblement les résultats et constitue une voie dans laquelle nous devons poursuivre notamment en étudiant les lois de modification des paramètres.

Les résultats que nous obtenons sont donc tout à fait honorables dans certains cas et demandent à être encore travaillés dans d'autres. Nous nous y employons notamment dans la dernière version de la carte. Cependant, l'originalité de l'architecture ainsi que sa courte existence jusqu'à maintenant (ne pouvant autoriser des expérimentations extensives) ne pouvaient pas permettre de concurrencer les modèles phares du moment. On notera tout de même que la représentation interne et explicite du temps a permis des performances meilleures que dans le cas d'une représentation externe.

Chapitre 7

Conclusion

« Le temps est la condition formelle a priori de tous les phénomènes en général. L'espace, comme forme pure de toute intuition externe, ne sert de condition a priori qu'aux phénomènes extérieurs. Au contraire, comme toutes les représentations, qu'elles aient ou non pour objets des choses extérieures, apartiennent toujours par elles-mêmes, en tant que détermination de l'esprit, à un état intérieur, et que cet état intérieur, toujours soumis à la condition formelle de l'intuition interne, rentre ainsi dans le temps, le temps est une condition a priori de tous les phénomènes en général, la condition immédiate des phénomènes intérieurs (de notre âme), et, par là même, la condition médiate des phénomènes extérieurs. [...] »

Emmanuel Kant, Critique de la raison pure, Esthétique transcendantale, 1781.

7.1 Bilan

Le connexionnisme a connu trois grandes époques depuis sa création : l'ère de l'engouement à partir de 1958, date de la création du perceptron mono-couche [Rosenblatt, 1958], où l'on pensait à tort que les réseaux neuronaux allaient pouvoir résoudre d'innombrables problèmes, l'ère de la désillusion suite au coup d'arrêt porté en 1969 par Minsky et Papert [Minsky et Papert, 1969], le perceptron mono-couche n'était en fait pas capable de réaliser la simple fonction logique du OU exclusif, puis l'ère du renouveau suite à la mise au point de la procédure d'adaptation des poids dans les perceptrons multi-couches [Le Cun, 1985;

Rumelhart *et al.*, 1986].

Une quatrième ère est-elle sur le point de s'amorcer ? Si l'on examine la question d'un point de vue historique, nous remarquons que chaque nouvelle ère a commencé par un événement majeur (perceptron, coup d'arrêt en 69, rétro-propagation). Or il n'y a pas réellement eu, ces dernières années, d'événement fondamental lançant le connexionnisme sur une nouvelle voie. Cependant, on constate désormais une volonté de dépasser les limitations actuelles du connexionnisme. Aussi, de nouveaux types de systèmes voient le jour, inspirés de la neurobiologie, de la psychologie ou mixant des techniques connexionnistes avec d'autres symboliques ou stochastiques (modèles hybrides). Dans ce sens, la question de savoir si le connexionnisme est à un tournant de son existence mérite d'être posée. Dans tous les cas, l'ère du perceptron multi-couches devra évoluer pour dépasser la « simple » classification de formes. Quels sont alors les points sur lesquels nous devons nous focaliser pour gommer les défauts actuels ?

Tout d'abord, les réseaux statiques sont des associateurs, ils apprennent à apparier deux vecteurs dans deux espaces différents. Il réalisent ainsi une fonction qui peut être relativement complexe. Cette fonctionnalité est tout à fait intéressante, notamment pour du traitement de bas niveau et plus précisément du traitement sensoriel. Les réseaux sont ainsi vus comme des boîtes noires et hermétiques pour lesquelles l'interprétation des constituants adaptables est difficile voire impossible. Il est clair que l'architecture et la philosophie même des structures telles que le perceptron ne vont pas dans le sens d'une interprétation possible des résultats. Pour pouvoir traiter des problèmes de plus haut niveau que ceux de la perception, nous devons envisager des réseaux dont le fonctionnement et l'architecture soient capables « d'expliquer » leur démarche.

Ensuite, parce que le temps est une dimension très importante dans la majeure partie des problèmes que nous nous posons, il est nécessaire de prendre en compte ce paramètre. Le temps, comme nous l'avons vu, se décline sous deux aspects : un aspect séquentiel ou d'ordre des événements et un aspect de durée de ces événements. Des modèles connexionnistes temporels ont été développés et nous en avons vu une classification. Presque tous s'occupent de l'aspect séquentiel mais souffrent encore de leur manque de tolérance en particulier pour la distorsion. Qui plus est, très peu prennent en compte la notion de durée.

Le traitement symbolique n'est d'autre part aucunement traité par la majeure partie des modèles connexionnistes. Ceux-ci sont en effet des systèmes numériques et ne sont donc pas, par nature, adaptés au traitement symbolique. Pourtant, cette notion semble importante pour des tâches de haut niveau, elle constitue en particulier la base des modèles d'IA classique et des systèmes logiques.

Le cerveau est le produit de son passé et, à cette image, un système intelligent doit en être de même. En l'occurrence l'information doit en premier lieu pouvoir être stockée. Nous devons, en outre, pouvoir récupérer les relations temporelles qui existent entre les différents éléments composant l'information.

Enfin, les réseaux connexionnistes ont plutôt une vue « localiste » des choses. Peu de systèmes intègrent des informations de nature différente, de bas et de haut niveau ou encore d'axes de traitement différents. D'un point de vue technique et pratique, il est bien évident que la conception d'un modèle complet du cerveau n'est pas réalisable aujourd'hui. On s'intéresse alors à différents sous-problèmes qui sont bien souvent eux-mêmes assez complexes. Cependant, ce qui sous-tend « l'intelligence » est peut-être un tout, et, comme le suggère la théorie Gestaltiste [Rock et Palmer, 1991], ce tout n'est peut-être pas la somme de ses parties.

Les points évoqués ci-dessus nous ont conduit à définir et à utiliser un automate connexionniste d'une fonctionnalité plus importante que le neurone classiquement utilisé dans le domaine du neuromimétisme.
Cet automate réagit sous l'influence de quantités d'information qu'il peut distinguer en fonction de leur provenance. Ainsi, dans sa version actuelle, l'automate possède deux types de connexions entrantes : les liens feed-forward et les liens intra-carte.
Les premiers définissent une information spatiale (que l'on nomme événement) de bas niveau ou émanant d'un niveau inférieur, il s'agit par exemple de coder un indice visuel [Hubel et Wiesel, 1977; Hubel et Wiesel, 1978], acoustique [Buser et Imbert, 1987], tactile, une position, etc, ou à un niveau plus haut, une séquence d'information correspondant par exemple à un mot, un mouvement, etc.
Les seconds expriment les relations temporelles qui existent entre les événements et plus particulièrement la notion d'ordre. Organisé sous la forme d'une carte, un ensemble d'automates (appelé ici super-unités) permet de définir des séquences d'information codées par une propagation d'activité au travers de cette carte.
Du point de vue du codage des événements, la carte est une carte auto-organisatrice et permet d'organiser topographiquement l'information. Ceci est en accord avec les principes d'organisation isotopique que l'on retrouve classiquement dans les aires sensorielles du cerveau : rétinotopie dans l'aire primaire du cortex visuel, tonotopie dans le cortex auditif, somatotopie pour la surface du corps projetée sur les aires somato-sensorielles, etc.
Le traitement séquentiel de l'information est assuré par des unités à l'intérieur des super-unités. Chaque unité définit un contexte d'activation pour l'information qui correspond à l'entrée *feed-forward* de l'unité. La carte construit ainsi des séquences d'unités par l'intermédiaire des liens intra-carte permettant une propagation d'activité. Afin de permettre une telle propagation, nous avons introduit une mémoire à court terme pour les unités de telle sorte que l'activité n'est pas « tout ou rien » mais décroît au cours du temps à partir d'une impulsion et selon une pente donnée en paramètre ou apprise. En particulier, nous avons vu que ce paramètre permet l'insertion d'information dans une chaîne. Le nombre d'événements insérables (et tolérés) est fonction de ce paramètre. De la même façon, chaque unité ne récolte pas l'information uniquement de l'unité précédente dans la chaîne mais également d'unités lointaines et cela afin d'autoriser la détérioration ou la suppression d'événements codés dans la carte. Ce phénomène est également contrôlé par un paramètre de profondeur de connexion dans la chaîne. Ces deux paramètres contrôlent la tolérance aux distorsions temporelles dans la chaîne. Ils peuvent être fixés *a priori* et être identiques pour chaque unité, ils peuvent également être appris. En effet, la tolérance

en distorsion temporelle ne doit pas être égale sur toute la longueur d'une chaîne d'information. L'exemple d'une syllabe contenant une plosive suivie d'une voyelle le montre bien. La partie occlusive (l'explosion) est beaucoup plus courte que la voyelle et la totalité de l'information de cette occlusive est vraisemblablement plus pertinente que la totalité de l'information pour la voyelle.

La carte TOM que nous avons présentée dans cette thèse permet donc l'apprentissage et le rappel de séquences d'information avec une tolérance à la distorsion temporelle sélective dans la séquence. Elle constitue un élément de base et a pour vocation de s'insérer dans un réseau afin d'en exploiter toutes ses possibilités. En outre, nous avons présenté une méthode permettant une supervision grâce à une carte TOM jouant le rôle de carte associative. Celle-ci permet d'apprendre des associations de formes, en particulier de fabriquer des couples <symbole,séquence>. Les objectifs et les perspectives de nos travaux seront exposés plus loin, mais, à la lumière des étapes de développement du modèle et des expérimentations effectuées nous pouvons faire les remarques suivantes :

- La carte, prise de façon isolée, n'est pas *a priori* construite pour une application particulière. Elle permet une propagation d'activité aboutissant ou non à une propagation caractéristique déjà apprise. Cette propagation doit être exploitée pour aboutir à des systèmes opérationnels, l'exploitation pouvant se faire soit par un algorithme classique soit par un autre système connexionniste ou encore par une autre carte TOM.

- Pour des applications complexes telle que la reconnaissance de la parole en mode multi-locuteurs, le nombre d'événements acoustiques et de séquences de ces événements est relativement élevé et rend difficile l'examen de la carte lorsque les résultats ne sont pas ceux espérés.

- Le modèle a été développé dans l'optique d'être applicable à différents problèmes incluant une composante temporelle. Pour cette raison, il se veut le plus général possible. A l'avantage d'être générique nous pouvons directement opposer le risque d'être moins pragmatique et par là même de perdre en performance. Cependant, nous pensons que ce que nous perdons à court terme en performance, nous le gagnerons à long terme grâce à la capacité d'intégration de différents niveaux de la carte TOM. Car celle-ci à l'ambition de s'intégrer dans un système multi-sensoriel qui reste à définir.

7.2 Perspectives d'utilisation du modèle

Bien que l'architecture que nous avons proposée ne soit qu'une étape dans un vaste champ de recherche, elle n'en demeure pas moins susceptible d'être appliquée à différents problèmes impliquant une dimension temporelle. En l'occurrence, après la parole, nous pouvons nous intéresser à d'autres axes sensoriels tels que la vision, la production de parole, ou encore la motricité. Dans ce paragraphe, nous proposons trois applications possibles à partir du modèle tel qu'il est présenté dans ce manuscrit. Bien évidemment, la carte TOM peut ne pas être complètement adaptée à ces applications, mais ces dernières serviront pour relever les défauts du modèle afin de lui apporter les améliorations adéquates.

7.2.1 Vision active

Le problème de la vision active consiste à coder les mouvements qui peuvent s'opérer dans un champ de vision. Par exemple, le mouvement d'une barre horizontale du haut vers le bas peut supposer être codé par une activité neuronale particulière. C'est ce codage que nous tentons de mettre en évidence dans l'application de la carte TOM qui suit.
En analyse de mouvements dans un champ visuel, la notion de distorsion temporelle est présente au même titre qu'en parole [Buser et Imbert, 1987]. Il convient donc de pouvoir la prendre en compte et à ce titre une carte TOM peut s'avérer intéressante. La figure 7.1 montre une architecture possible pour la modélisation de mouvements sur une rétine.

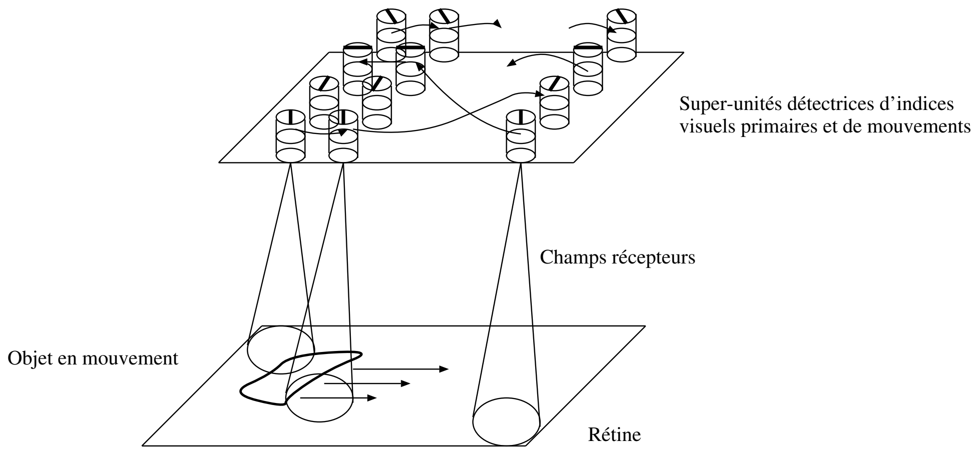


FIG. 7.1 – *Application de TOM pour la modélisation du mouvement sur une rétine*

Le principe consiste à apprendre les séquences d'indices visuels obtenues à partir de corpus des scènes visuelles. La rétine est découpée en petite zones qui servent de champs récepteurs aux super-unités de la carte de traitement. Sur une zone donnée, plusieurs super-unités projettent leur champ récepteur afin d'extraire des indices visuels sur cette zone [Hubel et Wiesel, 1977]. A bas niveau, ces indices sont des orientations (verticales, horizontales, obliques à différents degrés). En faisant abstraction de l'aspect temporel, on sait qu'une carte auto-organisatrice est capable d'apprendre de tels indices. Par rapport au modèle général de TOM où la compétition s'opère sur la totalité de la carte, nous souhaitons ici apprendre les indices sur les différentes zones. Une fois les indices codés, les propriétés de TOM vont nous permettre de coder la dynamique de ces indices.
En effet, le mouvement d'un objet de forme quelconque va générer une séquence d'indices visuels sur chacun des champs récepteurs. Cette séquence, au même titre que les séquences acoustiques apprises dans les expérimentations présentées au chapitre précédent, peuvent être codées sur une carte TOM via les liens intra-carte.
Un tel exemple est une application directe d'une carte TOM pour une application de

relativement bas niveau. Les combinaisons de cartes peuvent apporter une perspective intéressante dans l'utilisation du modèle. Nous examinons cette possibilité dans ce qui suit.

7.2.2 Suivi visuel de mouvements

Nous pouvons coupler la carte détectrice de mouvement sur une image décrite ci-dessus avec une carte motrice permettant de diriger un «oeil» sur l'objet en mouvement. Deux axes sensori-moteurs sont alors en présence ici, l'axe visuel de détection de mouvements sur une image et l'axe moteur permettant de diriger l'oeil sur l'objet en mouvement. Ces deux axes sont reliés par une carte associative dont le but est d'associer un mouvement oculaire (décrit par une séquence d'événements moteurs) à une séquence d'indices visuels caractérisant le déplacement d'un objet.
Pour réaliser un tel système, il faut au préalable avoir «appris» les deux axes en présence : apprentissage des mouvements sur une image, apprentissage des mouvements possibles de l'oeil. Ensuite, le système apprend les correspondances mouvement de l'objet / mouvement de l'oeil en fonction des différents cas de figures qui peuvent se présenter. Ces correspondances peuvent être traitées par l'aspect spatial de la carte TOM qui joue là le rôle de carte associative.

7.2.3 Commande motrice auditive

Nous citons enfin une application possible dans le domaine de la commande vocale d'un instrument automatique (robot, bras manipulateur ...). La figure 7.2 montre une architecture à base de cartes TOM pour une telle application.

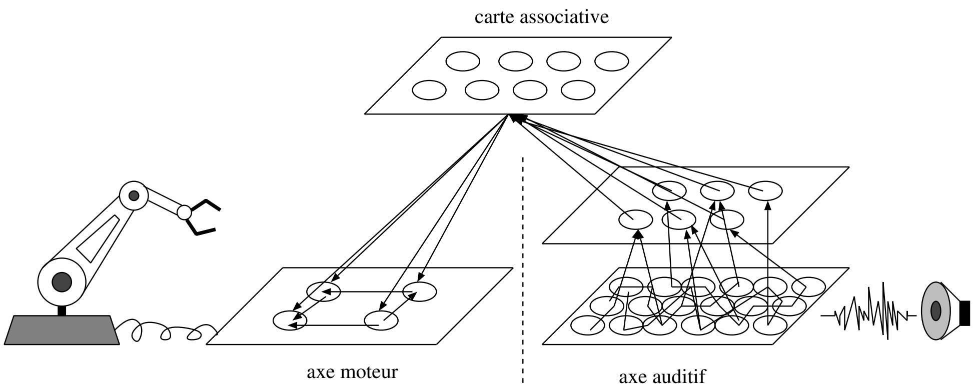


FIG. 7.2 – *Application de TOM pour la commande vocale*

Là encore, il s'agit de mettre en correspondance deux espaces de représentation différents. Par rapport à l'architecture supervisée décrite plus en détail au chapitre 4, la carte de l'axe moteur peut intégrer des séquences. Il s'agit alors d'associer non plus des symboles aux séquences d'informations acoustiques mais des séquences de commandes motrices s'il y a lieu.

Les applications que nous proposons ici n'ont pas été implantées, mais elles nous semblent traitables par des cartes TOM. Cependant, il y a souvent un décalage important entre la théorie et la pratique, c'est pourquoi ces applications constituent une phase importante dans le développement des cartes TOM. Certaines applications pourraient donc entrer dans le paragraphe suivant concernant les perspectives d'amélioration du système.

7.3 L'avenir de TOM, perspectives d'amélioration

Au vu de toutes ces remarques, comment peut-on envisager l'avenir d'un tel modèle? Tout d'abord précisons une fois encore qu'il ne s'agit en rien d'un modèle universel et encore bien moins d'un modèle du cerveau. Nous avons tenté de concevoir un système capable d'apprendre, de stocker et de reconnaître des séquences d'information avec des propriétés de tolérance spatio-temporelle et de généralisation, tout cela dans une approche résolument nouvelle par rapport aux modèles connexionnistes temporels que l'on trouve classiquement dans la littérature. L'aspect temporel était le but à court terme de notre architecture mais il sous-tend un certain nombre de buts plus éloignés qui ont été esquissés plus haut. Il reste donc un travail conséquent non seulement pour perfectionner encore le système du point de vue temporel mais aussi pour pouvoir traiter des problèmes dépassant la perception.

A court terme, les cartes TOM devront être testées de façon plus extensive en parole avec d'autres types de tests que ceux présentés ici. Sans modifier fondamentalement les principes architecturaux et de fonctionnement, d'autres types d'architectures à base de cartes TOM pourront être développés afin d'exploiter au mieux les ressources de TOM et d'en découvrir éventuellement certaines faiblesses.
Afin d'aboutir à un système ouvert permettant des intégrations multiples, il s'avère nécessaire de tester le système dans d'autres types d'applications que la parole. En particulier, diverses expériences peuvent être envisagées : problèmes de vision active, conduite d'une robot par la voix, couplage de différents axes de traitement (voir paragraphe précédent). Ces différentes applications auront pour but d'apprécier le comportement de TOM dans différentes situations mais aussi d'effectuer des améliorations techniques en rapport avec les applications.
Pour tout ce qui précède, une base logicielle adéquate avec un noyau de fonctions gérant les cartes TOM est nécessaire. Ce noyau, en langage C, existe déjà en partie mais demande à évoluer pour construire et faire fonctionner rapidement toute application à base de cartes TOM. Ce noyau constitue ainsi une plate-forme à laquelle on peut adjoindre diverses fonctions graphiques pour une gestion et une interprétation rapide des résultats.

D'un point de vue plus théorique sur le modèle, nous devrons poursuivre les investigations dans le domaine neurobiologique afin d'étudier plus finement les mécanismes de traitement temporel modélisables dans un système tel que celui présenté dans cette thèse. Des faits psychologiques seront également d'une aide précieuse.
Les points qui nous paraissent importants dans l'objectif de l'amélioration du modèle sont

décrits ci-dessous.

- Dans la version présentée ici, les liens reliant deux cartes de niveaux différents ont été extensivement étudiés dans le sens *feed-forward*, c'est à dire dans le sens d'abstraction ascendante. En revanche, nous ne nous sommes pas attardé sur le sens *feed-back* ou une carte de niveau supérieur interagit sur une carte de niveau inférieur. En particulier, nous avons vu la relative dépendance des résultats face au codage des entrées et plus spécifiquement de la quantification vectorielle de l'espace d'entrée. Celle-ci est effectuée indépendamment de l'apprentissage temporel or il paraît important de pouvoir l'affiner au cours de celui-ci. De nouveaux algorithmes de quantification vectorielle, tel Growing Neural-Gas [Fritzke, 1995], sont adaptés pour pouvoir tenir compte d'une intervention extérieure.

- Par ailleurs, l'activité de déclenchement des unités des super-unités est binaire. Une activité à valeurs réelles (certes plus difficile à interpréter dans le fonctionnement du réseau) pourrait être une voie à étudier pour mieux quantifier les séquences.

- D'autre part, dans les versions supervisées, l'association se fait entre une séquence (à apprendre) et le symbole (la signification) de cette séquence. Bien souvent dans la réalité, les données fournies par le « professeur » sont également temporelles et peuvent se décliner sous la forme de séquences. Il serait donc intéressant de pouvoir superviser une séquence par une séquence.

Dans le domaine de la recherche concernant la modélisation du fonctionnement du cerveau humain, notre travail et notre modèle ne sont que deux gouttes d'eau.
Cependant, nous envisageons à long terme une formalisation plus poussée de notre architecture. Nous nous sommes situés, dans le cadre de cette thèse, sur des applications limitées. Nous avons en particulier utilisé au maximum quatre cartes pour une application à la reconnaissance de la parole. Dans la perspective d'un modèle du cortex, il est bien évident que le nombre d'axes corticaux mis en jeu (et donc le nombre de cartes) sera sans commune mesure avec ce qui a pu être présenté ici. Il faudra donc pouvoir définir proprement les interactions de tous ces axes et en particulier les différentes échelles de temps qu'ils pourront utiliser.
Il semble que les structures sous-corticales jouent un rôle important dans le traitement de l'information (par exemple, cochlée-noyaux cochléaires [Buser et Imbert, 1987]). Celle-ci est représentée de différentes manières et il convient donc de modéliser le plus fidèlement possible les structures de représentation dans toute leur diversité et voir dans quelle mesure elle intègre, déjà à leur niveau, la notion temporelle.
Enfin, d'un point de vue strictement temporel, le futur modèle devra permettre une prise en compte totale du temps. Séquencement et durée sont des problèmes que nous avons commencé à traiter dans cette thèse. Bien d'autres problèmes restent encore à travailler. Par exemple, nous n'avons pas parlé de la notion de rythme : comment le cerveau représente et mémorise t-il le rythme musical ?

En résumé, le modèle que nous avons présenté dans cette thèse constitue un cadre de travail pour lequel diverses applications pratiques peuvent être envisagées et un point de départ vers des recherches encore plus ambitieuses aboutissant à la modélisation des fonctions cérébrales.

7.4 Conclusion personnelle

Les formalisations, les théories, les modèles sont la part visible de toute science mais la science ne serait pas sans ses interactions humaines. Ceci est encore plus vrai dans le domaine des neurosciences où la confrontation de cultures différentes impose un dialogue continuel entre les chercheurs. L'image du chercheur dans sa tour d'ivoire est véritablement un mythe. Ces trois années de recherche ne se concrétisent pas seulement par ce manuscrit. Il y a aussi tout ce qui ne peut pas s'écrire, tout ce qui a été acquis inconsciemment, et l'expérience humaine qu'a impliqué ce travail de recherche.
« Les machines ne pourront jamais être intelligentes parce qu'elle n'ont pas d'histoire... », cette remarque entendue au cours de conversations constitue à mon sens un véritable défi pour la recherche en connexionnisme. Je ne saurais dire si ce défi est utopique ou non, en tout cas, beaucoup de chercheurs travaillent actuellement à introduire la notion de temps dans les systèmes neuromimétiques. La tâche est gigantesque mais passionnante, ce travail de thèse a été pour moi une formidable expérience et un point de départ vers des développements futurs...

Aspect logiciel

Le modèle TOM a été programmé en langage C. Une application utilisant une ou plusieurs cartes TOM se décline sous la forme d'un programme général utilisant des fonctions de manipulation sur les carte TOM (construction, destruction, apprentissage, propagation d'activité ...). Ces fonctions constituent un « noyau » utilisable pour tout type d'application.
Ci-dessous, nous donnons la structure en langage C que le noyau manipule.

Structure pour le codage des événements (vecteurs) sensoriels :

```
typedef struct _sensory_map
{
  float *sensory_vect;
  struct _sensory_map *fol;
} SensoryMap;
```

Structure d'un champ recepteur *feed-forward*:

```
typedef struct _receipt_field
{
  union {  /* union definissant le type de lien */
    float *link;                 /* lien sur un element du vecteur d'entree */
    struct _unit *u_link;        /* lien sur une unite */
    struct _super_unit *su_link; /* lien sur une super-unite */
  } u;
  float pond;                    /* ponderation du lien */
  struct _receipt_field *fol;
} ReceiptField;
```

Structure d'une unité (à l'intérieur des super-unités) :

```
typedef struct _unit
{
  int Id;                       /* etiquette de l'unite */
```

```
  struct _super_unit *su;        /* super-unite dans laquelle se
       trouve l'unite */
  struct _l_unit *intra_connect; /* liens intra-carte (entree) */
  struct _l_unit *inter_connect; /* liens inter-carte (entree) */
  float act;                     /* activite de l'unite */
  float new_act;                 /* pour le calcul des nouvelles act. */
  short mem_act;                 /* memoire de l'activite passee d'une unite */
  short top;                     /* marqueur */
  short adaptable;            /* repere indiquant si l'unite est adaptable */
  float stm;                  /* memoire a court terme, terme de decroissance
 de l'activite de l'unite au cours du temps */
  int fbc;                    /* seuil permettant d'aller chercher l'info plus
 ou moins loin en profondeur dans le champ
 recepteur temporel (feed-back connection) */
} Unit;
```

Liste d'unités :

```
typedef struct _l_unit
{
  struct _unit *elt;
  float pond;           /* ponderation du lien. Uniquement lorsque la
   structure est utilisee pour avoir la liste
   des liens intra-carte */
  struct _l_unit *fol;
} L_Unit;
```

Structure d'une super-unité :

```
typedef struct _super_unit
{
  int Id;                       /* etiquette de la super-unite */
  struct _l_super_unit *local;  /* liens locaux  */
  struct _receipt_field *rf;    /* champ recepteur (entree inter-carte) */
  struct _l_unit *units;        /* les unites de la super-unit */
  int size;                     /* nombre d'unites utiles */
  float act;                    /* activite calculee de la super-unite */
  float err;                    /* err cumulee pour l'algo GNG */
  short top;                    /* marqueur */
  struct _tom *map;             /* carte dans laquelle se trouve la super-unite */
  int etiq;                     /* Etiquette associee a la super-unite */
} SuperUnit;
```

Liste des super-unités :

```
typedef struct _l_super_unit
{
  struct _super_unit *elt;
  int pond;                    /* ponderation, dans le cas d'une utilisation
  pour les connexions locales */
  struct _l_super_unit *fol;
} L_SuperUnit;
```

Structure d'une carte TOM :

```
typedef struct _tom
{
    int Id;                        /* etiquette de la carte */
    struct _l_super_unit *map;     /* liste des super-unites */
    int size;                      /* Nombre de super-unites dans la carte */
    int max_unit;                  /* Nombre maximum d'unites par super-unites */
    struct _tom *send_connection;  /* Carte de niveau superieur vers laquelle
                                      cette carte se connecte */
    struct _tom *connection_from;  /* Carte envoyant des connexions */
    short type_feed_forward;       /* type des connexions feed-forward :
      0 -> sur vecteur
      1 -> sur unite
      2 -> sur super-unite */
} TOM;
```

Structure du réseau de cartes :

```
typedef struct _tomnet
{
  struct _tom *tom;
  struct _tomnet *fol;
} TOM_NET;
```

Bibliographie

[Ackley *et al.*, 1985] D. Ackley, G. Hinton, et T. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[Ahissar et Vaadia, 1992] E. Ahissar et E. Vaadia. Dependence of cortical plasticity on correlated activity of single neurons and on context. *Science*, 257:1412–1415, september 1992.

[Aitkin, 1990] L. Aitkin. *The auditory cortex. Structural and functional bases of auditory perception*. Chapman and Hall, 1990.

[Alexandre *et al.*, 1991] F. Alexandre, F. Guyot, J.P. Haton, et Y. Burnod. The cortical column: a new processing unit for multilayered networks. *Neural networks*, 4:15–25, 1991.

[Alexandre et Guyot, 1995] F. Alexandre et F. Guyot. Neurobiological Inspiration for the Architecture and Functioning of Cooperating Neural Networks. Dans *Proceedings International Workshop on Artificial Neural Networks*, Malaga (Spain), June 1995.

[Alexandre, 1990] F. Alexandre. *Une modélisation fonctionnelle du cortex : la colonne corticale. Aspects visuels et moteurs.* PhD thesis, Université Nancy I, 1990.

[Ans *et al.*, 1994] B. Ans, Y. Coiton, J.C. Gilhodes, et J.L. Velay. A neural network model for temporal sequence learning and motor programming. *Neural Networks*, 1994.

[Ans, 1990] B. Ans. Modèle neuromimétique du stockage et du rappel de séquences temporelles. t311, série iii, C. R. Acad. Sci. Paris, 1990.

[Beaugé *et al.*, 1993] L. Beaugé, S. Durand, et F. Alexandre. Plausible self-organizing maps for speech recognition. Dans *Artificial neural nets and genetic algorithms*, rédacteurs R. F. Albrecht, C. R. Reeves, et N. C. Steele, pages 221–226. Springer–Verlag, 1993.

[Beaugé *et al.*, 1994] L. Beaugé, S. Durand, et Y. Lallement. De nouvelles perspectives pour le connexionnisme. *Revue VALGO*, (94-2), 1994.

[Bechtel et Abrahamsen, 1991] W. Bechtel et A. Abrahamsen. *Connectionism and the mind: an introduction to parallel processing in networks.* Cambridge, Mass., 1991.

[Bechtel et Abrahamsen, 1993] W. Bechtel et A. Abrahamsen. *Le connexionnisme et l'esprit : introduction au traitement parallèle par réseau.* édition la découverte, 1993.

[Blanchet, 1992] P. Blanchet. *Une architecture connexionniste pour l'apprentissage par l'expérience et la représentation des connaissances.* PhD thesis, Université Paris XI, 1992.

[Bourlard *et al.*, 1995] H. Bourlard, Y. Konig, et N. Morgan. Remap: Recursive estimation and maximisation of a posteriori probabilities. application to transition-based connectionist speech recognition. Rapport Technique TR-94-064, International Computer Science Institute, Berkeley, California, 1995.

[Bourlard et Morgan, 1990] H. Bourlard et N. Morgan. A continuous speech recognition system embedding MLP into HMM. Dans *Advances in Neural Information Processing Systems 2*, rédacteur D. Touretzky, pages 186–193. Morgan Kaufmann, 1990.

[Bourlard et Wellekens, 1990] H. Bourlard et C. J. Wellekens. Links between Markov models and multilayer perceptrons. *IEEE Transaction on pattern analysis and machine intelligence*, 12(12):1167–1178, 1990.

[Bourret *et al.*, 1991] P. Bourret, J. Reggia, et M. Samuelides. *Réseaux neuronaux, un approche connexionniste de l'Intelligence Artificielle.* Teknea, 1991.

[Bressloff, 1992] P. C. Bressloff. Complex dynamics of a discrete time model of a neuron. Dans *Neural Network Dynamics*, rédacteurs J.G. Taylor, E.R. Caianiello, R.M.J. Cotterill, et J.W. Clark, pages 103–121. Springer-Verlag, 1992.

[Bridle, 1990] J.S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. Dans *Advances in Neural Information Processing Systems 2*, rédacteur D.S. Touretzky, pages 211–217, Los Alto, CA, 1990. Morgan Kaufman.

[Brodmann, 1909] K. Brodmann. *Vergleichende Lokalisationslehre der grobhirnrinde.* J.A. Barth, Leipzig, 1909.

[Béroule, 1985] D. Béroule. *Un modèle de mémoire adaptative, dynamique et associative pour le traitement automatique de la parole.* PhD thesis, Université de Paris-sud, Orsay, 1985.

[Béroule, 1990] D. Béroule. Guided propagation : current state of theory and application. Dans *Neurocomputing. NATO ASI Serie*, rédacteurs F. Fogelman Soulié et J. Hérault. Springer Verlag Berlin Heidelberg, 1990.

[Burnod, 1988] Y. Burnod. *An adaptive neural network: The cerebral cortex.* Masson Paris, 1988.

[Buser et Imbert, 1987] P. Buser et M. Imbert. *Neurophysiologie fonctionnelle, Audition.* Hermann, 1987.

[Calliope, 1989] Calliope. *La parole et son traitement automatique.* Masson, Paris, Milan, Barcelone, 1989.

[Carpenter et Grossberg, 1988] G. A. Carpenter et S. Grossberg. The art of adaptative pattern recognition by a self-organizing neural network. *Computer*, (21):77–88, 1988.

[Changeux, 1992] J.P. Changeux. Les neurones de la raison. *La recherche*, 23(244):704–713, juin 1992.

[Chappell et Taylor, 1993] G. J. Chappell et J. G. Taylor. The temporal kohonen map. *Neural Networks*, 6:441–445, 1993.

[Chevrier, 1993] V. Chevrier. *Etude et mise en œuvre du paradigme multi-agents : de Atome à Gtmas.* PhD thesis, Université Henri Poincaré, Nancy I, 1993.

[Cho et Kim, 1995] S. B. Cho et J. H. Kim. An hmm/mlp architecture for sequence recognition. *Neural Computation*, (7):358–369, 1995.

[Christophe et Morton, 1994] A. Christophe et J. Morton. Comprehending baby-think. *Nature*, 370:250–251, July 1994.

[Cleeremans, 1993] A. Cleeremans. *Mechanisms of implicit learning.* MIT Press, 1993.

[Colnet et Durand, 1995] B. Colnet et S. Durand. Application of temporal neural networks to source localisation. Dans *International Conference on Artificial Neural Networks and Genetic Algorithms*, rédacteurs D. W. Pearson, N. C. Steele, et R. F. Albrecht, pages 456–459, Alès, France, 1995. Springer-Verlag.

[Colnet et Martino, 1995] B. Colnet et J.C. Di Martino. Bearing estimation with time-delay neural networks. Dans *Proceedings of the IEEE International Conference on Acousti cs, Speech, and Signal Processing*, Detroit, Michigan, May 1995.

[Davalo et Naim, 1989] E. Davalo et P. Naim. *Des réseaux de neurones.* Eyrolles, 1989.

[Davis et Mermelstein, 1980] S. B. Davis et P. Mermelstein. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on acoustics, speech, and signal processing*, ASSP-28(4):357–366, 1980.

[de Ribaupierre, 1986] F. de Ribaupierre. Traitement de l'information auditive dans le thalamus et le cortex cérébral. Dans *Le système auditif central - Anatomie et physiologie*, rédacteur R. Romand, pages 183–226. INSERM / SFA, série audition, 1986.

[Dear *et al.*, 1993] S. P. Dear, J. A. Simmons, et J. Fritz. A possible neuronal basis representation of acoustic scenes in auditory cortex of the big brown bat. *Nature*, 364:620–626, august 1993.

[Dehaene *et al.*, 1987] S. Dehaene, J.P. Changeux, et J.P. Nadal. Neural networks that learn temporal sequences by selection. *Proc. Natl. Acad. Sci. USA, Biophysics*, 84:2727–2731, May 1987.

[Dehaene-Lambertz et Dehaene, 1994] G. Dehaene-Lambertz et S. Dehaene. Speed and cerebral correlates of syllable discrimination in infants. *Nature*, 370:292–294, July 1994.

[Delgutte et Kiang, 1984] B. Delgutte et N. YS. Kiang. Speech coding in the auditory nerve: I. vowel-like sounds. *acoustical society of America*, 75(3), march 1984.

[Delgutte, 1984] B. Delgutte. Speech coding in the auditory nerve: Ii. processings schemes for vowel-like sounds. *acoustical society of America*, 75(3), march 1984.

[Deville, 1994] Y. Deville. Trends in hardware implementations of neural networks. *Valgo*, 94-1:25–44, 1994.

[Dugast *et al.*, 1994] C. Dugast, L. Devillers, et X. Aubert. Combining tdnn and hmm in a hybrid system for improved continuous-speech recognition. *IEEE Transaction on Speech and Audio Processing*, 2(1):217–223, 1994. part II.

[Durand et Alexandre, 1994a] S. Durand et F. Alexandre. Mécanisme d'apprentissage spatio-temporel pour l'organisation sensori-motrice. Dans *Actes NSI94*, pages 71–74, Chamonix, 1994.

[Durand et Alexandre, 1994b] S. Durand et F. Alexandre. A neural network based on sequence learning: Application to spoken digits recognition. Dans *7th international conference on Neural Networks and Their Applications*, pages 290–298, Marseilles, 1994.

[Durand et Alexandre, 1995a] S. Durand et F. Alexandre. Learning speech as acoustic sequences with the unsupervised model, TOM. Dans *NEURAP, 8th International Conference on Neural Networks and their Applications*, Marseilles, France, 1995.

[Durand et Alexandre, 1995b] S. Durand et F. Alexandre. Spatio-temporal mask learning: Application to speech recognition. Dans *International Conference on Artificial Neural Networks and Genetic Algorithms*, rédacteurs D. W. Pearson, N. C. Steele, et R. F. Albrecht, pages 132–135, Alès, France, 1995. Springer-Verlag.

[Durand et Alexandre, 1996] S. Durand et F. Alexandre. Tom, a new temporal neural net architecture for speech signal processing. Dans *ICASSP, IEEE International Conference on Acoustic Speech and Signal Processing*, Atlanta, USA, 1996.

[Durand, 1993] S. Durand. Apprendre des indices acoustiques et les organiser dans le temps, la solution connexionniste... Dans *VALGO, Cinquièmes journées annuelles de l'ACTH*, pages 22–31, 1993.

[Durand, 1994] S. Durand. Réseaux neuromimétiques spatio-temporels pour l'organisation des sens. application à la parole. Dans *Actes Rencontres Nationales des Jeunes Chercheurs en Intelligence Artificielle*, page 297, Marseille, 1994.

[Durand, 1995] S. Durand. *TOM, une architecture connexionniste de traitement de séquences. Application à la reconnaissance de la parole.* PhD thesis, Université Henri Poincaré, Nancy I, 1995.

[Elman, 1990] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[Evans et Whitfield, 1964] E. F. Evans et I. C. Whitfield. Classification of unit responses in the auditory cortex of the unanaesthetized and unrestrained cat. *Journal of Physiology*, (171):476–493, 1964.

[Ferran *et al.*, 1994] E. Ferran, D. Shire, et P. Ferrara. Neural network clustering of protein sequences. Dans *7th International Conference on Neural Networks and their Applications*, pages 36–48, Marseilles, 1994.

[Fohr, 1986] D. Fohr. *APHODEX : Un système expert en décodage acoustico-phonétique de la parole continue.* PhD thesis, Université Henri Poincaré, Nancy I, 1986.

[Fritzke, 1995] B. Fritzke. A growing neural gas network learns topologies. Dans *Advances in Neural Information Processing Systems 7*, rédacteurs G. Tesauro, D.S. Touretzky, et T.K. Leen. MIT Press, Cambridge MA, 1995.

[Fukushima, 1980] K. Fukushima. Neocognitron : a self organizing neural network for a mechanism of pattern recognition unaffected by a shift in position. *Biological Cybernetics*, (36):193–202, 1980.

[Fuster, 1989] J. M. Fuster. *The prefrontal cortex. Anatomy, physiology, and neuropsychology of the frontal lobe.* Raven Press New York, 1989.

[Gas, 1994] B. Gas. *Un modèle connexionniste non supervisé pour l'apprentissage et la reconnaissance de séquences temporelles.* PhD thesis, Université de Paris-Sud, Orsay, 1994.

[Gershenfeld et Weigend, 1994] N. A. Gershenfeld et A. S. Weigend. The future of time series: Learning and understanding. Dans *Time Series Prediction. Forecasting the Future and Understanding the Past*, rédacteurs A. Weigend et N. Gershenfeld. SFI Studies in the Sciences of Complexity, Addison-Wesley, 1994.

[Geshwind, 1984] N. Geshwind. Les spécialisations du cerveau humain. *Le Cerveau, Belin*, 1984.

[Goldberg et Holland, 1988] D. E. Goldberg et J. H. Holland. Special issue on genetic algorithms. *Machine Learning*, 3(2/3), Octobre 1988.

[Greenberg, 1988] S. Greenberg. The ear as a speech analyzer. *Journal of Phonetics*, (16):139–149, 1988.

[Guigon *et al.*, 1991] E. Guigon, B. Dorizzi, Y. Burnod, et W. Schultz. Learning of sensori-motor sequences by prefrontal cortical circuits: a model. *Soc Neurosci Abstr*, (17):307, 1991.

[Guigon, 1993] E. Guigon. *Modélisation des propriétés du cortex cérébral. Comparaison entre aires visuelles, motrices et préfrontales.* PhD thesis, Ecole Centrale de Paris, 1993.

[Guyot *et al.*, 1989] F. Guyot, F. Alexandre, et J.P. Haton. Toward a continuous model of the cortical column: application to speech recognition. Dans *Proceedings of the International Conference on Acoustic Speech and Signal Processing*, Glasgow, 1989.

[Guyot, 1990] F. Guyot. *Une modélisation fonctionnelle du cortex : la colonne corticale. Aspects auditifs et visuels.* PhD thesis, Université Nancy I, 1990.

[Haffner et Waibel, 1992] P. Haffner et A. Waibel. Multi-state time delay neural networks for continuous speech recognition. Dans *Advances in Neural Information Processing Systems 4*, rédacteurs J.E. Moody, S.J Hanson, et R.P. Lipmann, pages 135–143. Morgan Kaufmann, San Mateo, CA, 1992.

[Hajislam, 1994] R. Hajislam. *Décodage acoustico-phonétique et robustesse en reconnaissance de la parole.* PhD thesis, Université Henri Poincaré, Nancy I, 1994.

[Hampshire II et Waibel, 1992] J. B. Hampshire II et A. Waibel. The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(7):751–769, July 1992.

[Haton et Haton, 1989] JP. Haton et MC. Haton. *L'intelligence artificielle.* Que sais-je? Presse Universitaire de France, 1989.

[Hebb, 1949] D. O. Hebb. *The organization of behaviour.* Wiley, New-York, 1949.

[Hopfield, 1982] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Dans *Proceedings of the National Academy of Sciences, USA*, pages 2554–2558, 1982.

[Huang *et al.*, 1988] W. Huang, R. Lippmann, et T. Nguyen. Neural nets for speech recognition. Dans *Conf. Acoustical Society of America*, 1988.

[Hubel et Wiesel, 1977] D. H. Hubel et T. N. Wiesel. Functional architecture of macaque monkey visual cortex. *Ferrier Lecture Proc. Roy. Soc. Lond.B*, pages 1–59, 1977.

[Hubel et Wiesel, 1978] D. H. Hubel et T. N. Wiesel. *Les mécanismes cérébraux de la vision.* Le Cerveau, Belin, 1978.

[Hush et Horne, 19993] D. R. Hush et B. G. Horne. Progress in supervised neural networks. what's new since lipmann? *IEEE Signal Processing Magazine*, pages 8–39, january 19993.

[Iso et Watanabe, 1990] K. Iso et T. Watanabe. Speaker-independent word recognition using a neural prediction model. Dans *Readings in Speech Recognition*, rédacteur A. Waibel, pages 443–446. Springer-Verlag, 1990.

[Iversen, 1978] L. Iversen. *La chimie du cerveau.* Le Cerveau, Belin, 1978.

[Jodouin, 1990] J.F. Jodouin. Présentation des modèles connexionnistes. *intellectica*, pages 9–39, 1990.

[Jordan, 1986] M. I. Jordan. Attractor dynamics and parallelism in a connectionist sequential machine. Dans *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, rédacteur Hillsdale. Erlbaum, 1986.

[Kaukoranta *et al.*, 1987] E. Kaukoranta, R. Hari, et O. V. Lounasmaa. Responses of the human auditory cortex to vowel onset after fricative consonants. *Experimental Brain Research*, (69):19–23, 1987.

[Klatt, 1986] K. H. Klatt. The problem of variability in speech recognition and models of speech perception. Dans *Invariance and variability in speech processes*, rédacteurs J.S. Perkell et D.H. Klatt, pages 300–324. New Jersey: Lawrence Erlbaum, 1986.

[Kohonen *et al.*, 1984] T. Kohonen, K. Makisara, et T. Saramaki. Phonotopic maps – insightful representation of phonological features for speech recognition. Dans *IEEE Proceedings of the 7th International Conference on Pattern Recognition*, 1984.

[Kohonen, 1989] T. Kohonen. *Self-Organization and Associative Memory*. Springer Series in Information Sciences. Springer-Verlag, third édition, 1989.

[Lallement *et al.*, 1995] Y. Lallement, M. Hilario, et F. Alexandre. Neurosymbolic integration: Cognitive grounds and computational strategies. Dans *World Conference on the Fundamentals of Artificial Intelligence*, rédacteurs M. DeGlas et Z. Pawlak, Paris, July 1995.

[Lallement et Durand, 1994] Y. Lallement et S. Durand. Ancrage de symboles dans des systèmes connexionnistes dédiés à des tâches de perceptions sensorielles. Dans *Actes NSI94*, pages 221–224, Chamonix, 1994.

[Large et Kolen, 1995] E. W. Large et J. F. Kolen. Resonance and the perception of musical meter. *Connection Science*, 6(1):177–208, 1995.

[Le Cerf, 1994] P. Le Cerf. Multilayer perceptrons as labelers for hidden Markov model. *IEEE Transaction on Speech and Audio Processing*, 2(1):185–193, 1994. part II.

[Le Cun, 1985] Y. Le Cun. Une procédure d'apprentissage pour réseau à seuil asymétrique. Dans *Proceedings of Cognitiva 85*, pages 599–604, Paris, 1985.

[Levine et Bapi, 1990] D. S. Levine et R. S. Bapi. Networks modeling the involvement of the frontal lobes in learning and performance of flexible movement sequences. Dans *Proceedings of International Joint Conference on Neural Networks, vol. 2*, pages 759–764, 1990.

[Levine et Parks, 1992] D. S. Levine et R. Parks. Frontal lesion effects on verbal fluency in a network model. Dans *Proceedings of the International Joint Conference on Neural Networks, Vol. 2*, pages 39–44, 1992.

[Lippmann, 1989] R. P. Lippmann. Review of neural networks for speech recognition. *Neural Computation*, 1:1–38, 1989.

[Locke, 1983] J L. Locke. *Phonological acquisition and change.* Academic press, 1983.

[MacCulloch et Pitts, 1943] W. S. MacCulloch et W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[Martinetz *et al.*, 1993] T. M. Martinetz, S. G. Berkovich, et K. J. Schulten. "neural-gas" network for vector quantization and its application to time-series prediction. *IEEE Transaction on Neural Networks*, 4(4):558–569, 1993.

[Martinetz et Schulten, 1991] T. M. Martinetz et K. J. Schulten. A "neural-gas" network learns topologies. Dans *Artificial Neural Network*, rédacteurs T.Kohonen, K. Mäkisara, O. Simula, et J. Kangas, pages 397–402. Elsevier Science Publishers, B.V. (North-Holland), 1991.

[Martinetz et Schulten, 1994] T. M. Martinetz et K. J. Schulten. Topology representing networks. *Neural Networks*, 7(3):507–522, 1994.

[Martinetz, 1993] T. M. Martinetz. Competitive hebbian learning rule forms perfectly topology preserving maps. Dans *ICANN'93, International Conference on Artificial Neural Networks*, pages 427–434, Amsterdam, 1993. Springer.

[Mashaal, 1995] M. Mashaal. Les balbutiements prometteurs des ordinateurs à adn. *La Recherche*, (279):860–862, 1995.

[McNaughton et Morris, 1987] B. L. McNaughton et R. G. M. Morris. Hippocampal synaptic enhancement and information storage within a distributed memory system. *TINS*, 10(10), 1987.

[Mercier, 1990] G. Mercier. Recognition of speaker dependent continuous speech with keal. Dans *Readings in speech recognition*, rédacteur A. Waibel. Morgan Kaufmann Publishers, 1990.

[Metropolis *et al.*, 1953] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, et E. Teller. Equation of state calculations for fast computing machines. *Journal of Chemical Physics*, 6:1087, 1953.

[Midenet et Grumbach, 1990] S. Midenet et A. Grumbach. Supervised learning based on Kohonen's self-organizing feature maps. Dans *Proceedings of INNC Paris*, pages 773–776, july 1990.

[Milanese, 1994] R. Milanese. Feature binding through synchronized neuronal oscillations: a preliminary study. Rapport Technique TR-94-044, ICSI, international computer science institute, august 1994.

[Minsky et Papert, 1969] M. Minsky et S. Papert. *Perceptrons.* Cambridge, MIT Press, 1969.

[Minsky, 1986] M. Minsky. *La société de l'esprit.* InterEditions, 1986.

[Morgan et Scofield, 1991] D. P. Morgan et C. L. Scofield. *Neural networks and speech processing.* Kluwer Academic Publishers, 1991.

[Mougeot et Asencorr, 1990] M. Mougeot et R. Asencorr. Auto-organisation de cellules sélectives à l'orientation dans le cortex visuel et modification après experiences visuelles. Dans *NSI90*, may 1990.

[Mountcastle, 1978] V. B. Mountcastle. *An organizing principle for cerebral function. The unit module and the distributed system.* The mindful brain, MIT Press, Cambridge, 1978.

[Nenov et Dyer, 1993] V. I. Nenov et M. G. Dyer. Perceptually grounded language learning: Part1–a neural network architecture for robust sequence association. *Connection Science*, 5(2):115–138, 1993.

[Nenov et Dyer, 1994] V. I. Nenov et M. G. Dyer. Perceptually grounded language learning: Part2–dete: A neural/procedural model. *Connection Science*, 6(1):3–41, 1994.

[Oja, 1982] E. Oja. A simplified neuron model as a principal component analyzer. *J. Math. Biology*, 15:267–273, 1982.

[Pal *et al.*, 1993] N. R. Pal, J. C. Bezdek, et E. C-K Tsao. Generalized clustering networks and kohonen's self-organizing scheme. *IEEE Transaction on Neural Networks*, 4(4):549–557, 1993.

[Pearlmutter, 1990] B. A. Pearlmutter. Dynamic necurrent neural networks. Rapport Technique CMU-CS-90-196, Carnegie Mellon University, December 1990.

[Personnaz *et al.*, 1986] L. Personnaz, I. Guyon, et G. Dreyfus. Collective computationnal properties of neural networks: new learning mechanisms. *Physical Review A*, 34(3), 1986.

[Piaget, 1946] J. Piaget. *Le developpement de la notion de temps chez l'enfant.* PUF, 1946.

[Pican, 1995] N. Pican. *Approches statique et Dynamique de la modulation des efficacités synaptiques dans les réseaux de neurones.* PhD thesis, Université Henri Poincaré, Nancy I, 1995.

[Pinto-Hamuy et Linck, 1965] T. Pinto-Hamuy et P. Linck. Effect of frontal lesions on performance of sequential tasks by monkeys. *Experimental neurology*, 12:96–107, 1965.

[Poppen *et al.*, 1965] A. L. Poppen, K. H. Pribram, et R. S. Robinson. Effect of frontal lobotomy in man on the performance of a multiple choice task. *Experimental neurology*, 11:217–229, 1965.

[Pujol, 1990] R. Pujol. Le traitement du son dans l'oreille interne. *Pour la science*, (154):20–29, 1990.

[Rabiner et Juang, 1993] L. Rabiner et B-H Juang. *Fundamentals of speech recognition.* Signal processing series. PTR, Prentice Hall, 1993.

[Rabiner, 1990] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Dans *Readings in speech recognition*, rédacteur A. Waibel, pages 267–296. Morgan Kaufmann Publishers, 1990.

[Reiss et Taylor, 1991] M. Reiss et J. G. Taylor. Storing temporal sequences. *Neural Networks*, 4:773–781, 1991.

[Renals *et al.*, 1994] S. Renals, N. Morgan, H. Bourlard, M. Cohen, et H. Franco. Connectionist probability estimators in hmm speech recognition. *IEEE Transaction on Speech and Audio Processing*, 2(1):161–174, 1994. part II.

[Rock et Palmer, 1991] I. Rock et S. Palmer. L'héritage du gestaltisme. *Pour la Science*, (160):64–70, fevrier 1991.

[Romand,] rédacteur R. Romand. *Le système auditif central - Anatomie et physiologie.* Série Audition. INSERM-SFA, ?

[Rosenblatt, 1958] F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Review*, (65):386–408, 1958.

[Rubner et Schulten, 1990] J. Rubner et K. Schulten. Development of feature detectors by self-organization. *Biological cybernetics*, (62):193–199, 1990.

[Rumelhart *et al.*, 1986] D. E. Rumelhart, G. E. Hinton, et R. J. Williams. Learning internal representations by error propagation. Dans *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, volume 1. Cambridge, MA: MIT Press, 1986.

[Rumelhart et McClelland, 1986] D. E. Rumelhart et J. L. McClelland. *Parallel Distributed Processing: Exploration in the Microstructure of Cognition.* Cambridge, MA: MIT Press, 1986.

[Schwartz, 1987] JL. Schwartz. Représentations auditives de spectres vocaliques. *Thèse de docteur d'état*, juillet 1987.

[Segui, 1993] J. Segui. Surdité phonologique et perception du langage. *Revue de Neuropsychologie*, 3(4):397–406, 1993.

[Sejnowski et Rosenberg, 1987] T. J. Sejnowski et C. R. Rosenberg. Parallel networks that learn to pronounce english text. *Complex System*, 1:145–168, 1987.

[Servan-Schreiber *et al.*, 1991] D. Servan-Schreiber, A. Cleeremans, et J. McClelland. Graded state machines: the representation of temporal contingencies in simple recurrent networks. *Machine Learning*, 7:161–193, 1991.

[Steinschneider *et al.*, 1982] M. Steinschneider, J. Arezzo, et H. G. Vaughan Jr. Speech evoked activity in the auditory radiations and cortex of the awake monkey. *Brain Research*, (252):353–365, 1982.

[Steinschneider *et al.*, 1994] M. Steinschneider, C. E. Schroeder, J. C. Arezzo, et H. G. Vaughan Jr. Speech-evoked activity in primary auditory cortex: effects of voice onset time. *Electroencephalography and clinical neurophysiology*, (92):30–43, 1994.

[Stevens, 1978] C. Stevens. *Le neurone.* Le Cerveau, Belin, 1978.

[Stuss et Benson, 1986] D. T. Stuss et D. F. Benson. *The Frontal Lobes.* Raven Press, New York, 1986.

[Suga, 1990] N. Suga. Cortical computational maps for auditory imaging. *Neural Networks*, 3:3–21, 1990.

[Sutton, 1998] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, (3):9–44, 1998.

[Szentagothai, 1975] J. Szentagothai. The "module concept" in cerebral cortex architecture. *Brain Research*, 95:475–496, 1975.

[Tamura et Waibel, 1988] S. Tamura et A. Waibel. Noise reduction using connectionist models. Dans *Proc. ICASSP, New York, NY*, pages 553–556, April 1988.

[Thorpe, 1989] S. J. Thorpe. Local vs. distributed coding. *Intellectica*, 2(8):2–40, 1989.

[Touzet, 1990] C. Touzet. *Contribution à l'étude et au développement de modèles connexionnistes séquentiels.* PhD thesis, Université Montpellier II, 1990.

[Waibel *et al.*, 1989] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, et K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 37(3):328–339, 1989.

[Waibel, 1992] A. Waibel. Neural network approaches for speech recognition. Dans *Advances in Speech Signal Processing*, rédacteurs Sakaoki Furui et M. Mohan Sondhi, pages 555–595, New York, 1992. Marcel Decker, Inc.

[Walter *et al.*, 1991] J. A. Walter, T. M. Martinetz, et K. J. Schulten. Industrial robot learns visuo-motor coordination by means "neural-gas" network. Dans *Artificial Neural Networks*, rédacteurs T.Kohonen, K. Mäkisara, O. Simula, et J. Kangas, pages 357–364. Elsevier Science Publishers B.V. (North-Holland), 1991.

[Wan, 1994] E. Wan. Time series prediction by using a connectionist network with internal delay lines. Dans *Time Series Prediction. Forecasting the Future and Understanding the Past*, rédacteurs A. Weigend et N. Gershenfeld. SFI Studies in the Sciences of Complexity, Addison-Wesley, 1994.

[Wang et Arbib, 1990] D. Wang et M. A. Arbib. Complex temporal sequence learning based on short-term memory. *Proceedings of the IEEE*, 78(9):1536–1543, 1990.

[Weigend et Gershenfeld, 1994] rédacteurs A. S. Weigend et N. A. Gershenfeld. *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley, 1994.

[Werbos, 1990] P. J. Werbos. Backpropagation throught time: what it does and how to do it. *proceedings of the IEEE*, 78(10), october 1990.

[Whitfield et Evans, 1965] I. C. Whitfield et E. F. Evans. Responses of auditory cortical neurons to stimuli of changing frequency. *Journal of Neurophysiology*, (28):655–672, 1965.

[Widrow et Hoff, 1960] B. Widrow et M. E. Hoff. Adaptive switching circuits. Dans *1969 IRE WESON Convention Record, New-York*, pages 96–104, 1960.

[Zipser, 1991] D. Zipser. Recurrent network model of the neural mechanism of short-term active memory. *Neural Computation*, 3:179–193, 1991.

Index