Etude de la construction par réseaux neuromimétiques de représentations interprétables

Laurent Bougrain

▶ **To cite this version:**

Laurent Bougrain. Etude de la construction par réseaux neuromimétiques de représentations interprétables : Application à la prédiction dans le domaine des télécommunications. Réseau de neurones [cs.NE]. Université Henri Poincaré - Nancy 1, 2000. Français. NNT : 2000NAN10241 . tel-01746779v2

HAL Id: tel-01746779
https://inria.hal.science/tel-01746779v2

Submitted on 18 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Département de formation doctorale en informatique **École doctorale IAE + M**
UFR STMIA

Étude de la construction par réseaux neuromimétiques de représentations interprétables :

Application à la prédiction dans le domaine des télécommunications

THÈSE

présentée et soutenue publiquement le 14 novembre 2000

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Laurent Bougrain

Composition du jury

Rapporteurs :	M. Christian Pellegrini	Professeur, Université de Genève, Suisse
	M. Jerzy Korczak	Professeur, Université Louis Pasteur, Strasbourg
	M. Amedeo Napoli	Chargé de recherche, CNRS, Nancy
Examinateurs :	M. Jean-Paul Haton	Professeur, Université Henri Poincaré, Nancy
	M. Alexandre Caminada	Ingénieur de recherche et développement, France Télécom
Directeur :	M. Frédéric Alexandre	Directeur de recherche, INRIA, Nancy

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

A mes parents,
en remerciement
de leur dévouement.

Remerciements

Paradoxalement, la thèse est certainement la publication qui résulte du plus grand nombre de collaborations, de rencontres et d'influences, et pourtant, l'Histoire officielle ne retient qu'un auteur. Pour réparer cette injustice, je vais immédiatement, pareillement à mes prédécesseurs, rendre hommage à ces artisans de l'ombre.

Je commencerai par ceux qui sont quand même reconnus d'utilité publique et qui ont droit à la première page car ils ont accrédité ce travail ; mais ils ont également influencé, par leurs conseils avisés, la version définitive de ce manuscrit : je parle des membres du jury. La place d'honneur dans ces conditions revient à mon directeur de thèse, Frédéric Alexandre, dont l'efficacité est telle qu'on peut se demander s'il ne se dédouble pas (le fait que certaines personnes l'appellent Fred et d'autres Alex tend à le prouver). Je rends ici hommage à un altruiste prédicateur de l'intelligence neuromimétique convaincu et convainquant.
J'ai également beaucoup apprécié la gentillesse de Christian Pellegrini, la qualité d'écoute de Jean-Paul Haton, la qualité de relecture d'Amedeo Napoli et le professionnalisme de Jerzy Korczak. Merci à Alexandre Caminada et Thierry Balandier de France Télécom R&D de m'avoir fait bénéficier de leurs compétences et de leur enthousiasme autour d'un défi de notre temps.
Je tiens également à exprimer ma gratitude à Patrick Gallinari qui a cru en moi et l'a fait savoir pour me permettre d'entreprendre cette belle aventure.

L'équipe CORTEX et son entourage compte des membres inoubliables. Je les salue tous en les remerciant d'avoir transformé ces années de travail en semaines de bonheur. En plus de l'expérience des aînés (Nicolas P., Jean-Daniel, Dominique, Bernard, Didier, Jean-Charles, Yann G., Alister) et de l'enthousiasme des cadets (Karima, Olivier, Dudu, Bruno, Georges, Vincent), j'ai bénéficié des qualités humaines et techniques de Nicolas qui sait aussi bien prendre soin d'un robot Koala à vision monoscopique que d'une plante verte, qui milite pour la survie des accents dans les mails et n'hésite pas à payer de sa personne en participant à 33 vols paraboliques au nom de la science pour mesurer à quel point l'homme peut être désorienté (et malade) dans l'espace ; de Yann qui est tellement chaleureux qu'un T-shirt suffit à le couvrir même en hiver et qui m'a permis de bénéficier d'une secrétaire, d'une batterie de secours, d'un distributeur de monnaie et des dépêches de l'AFP sept jours sur sept ; et d'Hervé dont les discours intarissables et la bonne humeur me manquent déjà. J'adresse aussi mes amitiés en signe de remerciements à Martine, Evelyne, Alain, Arnaud et Pierrot.

Je voudrais également remercier Gaëlle, Wallis et Peggy d'avoir partagé passionnément leur intérêt pour la psychologie pendant que le président du fan club de Lacan mettait le mien en doute.

Je suis reconnaissant à Caro d'avoir compris l'importance que ce travail avait pour moi et d'avoir supporté que je sois très occupé en journée et préoccupé en soirée. Merci à la famille et aux amis qui se sont déplacés pour faire régner une atmosphère partisane lors de ma soutenance afin d'influencer favorablement le jury.

J'aimerais également remercier les personnes ou entités suivantes qui ont grandement facilité mon travail de thèse. Merci donc, dans le désordre, au vélo de Nico, à la voiture de Yann, au canapé d'Hervé, aux déménageurs nancéiens (l'équipe de nuit), aux ambulanciers du Vélodrome, à la SNCF, aux imprimantes HP, aux photocopieurs Ranx Xerox, aux ascenceurs Otis, ...

Table des matières

Introduction

1.000.000.000.000, un téraoctet, c'est la taille des bases de données disponibles pour des problèmes actuels dans le monde de l'industrie, des finances ou des communications. L'accroissement des capacités de stockage a généré une accumulation des informations. De nouvelles bases sont créées tous les jours, et tout ce qui peut accéder au statut de donnée digitale peut être facilement stocké sur une variété de supports. De plus, avec le développement d'internet et des réseaux locaux, tout le monde, professionnel de la finance ou simple citoyen, peut avoir facilement accès à une myriade de bases de données. Aujourd'hui, le problème n'est donc plus réellement le manque d'informations, mais le manque d'organisation qui ne permet pas de disposer facilement des connaissances qu'elles contiennent. En effet, obtenir des gigaoctets d'images satellitaires, de relevés météorologiques et de traités théoriques sur les sciences de la prévision ne donne pas d'information précise sur le temps qu'il fera demain en Italie. Encore faut-il savoir exploiter, dans cette masse peu structurée, les quelques octets qui nous donneront le renseignement voulu.

Le problème de l'accès à l'information s'est donc quelque peu déplacé ces dernières années. Si tout ce qui peut l'être (ou presque) est soigneusement mesuré, étiqueté et stocké dans l'attente d'une utilisation hypothétique, aucune structuration a priori de l'information n'est faite, de peur de perdre des données qui pourraient se révéler pertinentes pour une utilisation future. Dans le doute, vu le faible prix du stockage, tout est conservé. On obtient alors des bases de données gigantesques, mais sans structure. Il appartient donc désormais aux utilisateurs de bases de données d'analyser les informations mises à leur disposition, de sélectionner les plus appropriées à leur objectif et de les arranger convenablement les unes avec les autres pour arriver à la bonne formule. Le problème n'est donc plus d'accéder aux données, mais bien à l'information pertinente qu'elles contiennent, en délaissant les milliers d'autres qui ne sont que du bruit pour la requête courante. De plus, l'information pertinente n'existe pas nécessairement explicitement ; elle peut être contenue implicitement à travers d'autres informations qu'il faut interpréter afin d'en extraire le renseignement voulu. Donc, au problème de recherche de l'information pertinente vient s'ajouter celui de l'interprétation et de la structuration de données.

Ce travail de recherche directe et indirecte, tout d'abord intuitif, est devenu si fréquent, si nécessaire, il s'est révélé si ardu, et finalement si peu intuitif, qu'un nouveau domaine de recherche a émergé, afin d'explorer, de mieux comprendre et de théoriser ce domaine de la recherche d'informations dans des bases de données vastes et peu structurées. Il s'agit de la fouille de données (*data mining* [Michie *et al.*, 1994; Fayyad *et al.*, 1996; Dhar and Stein, 1997]). Un autre but, plus opérationnel, est de proposer de nouveaux outils et de nouvelles méthodologies pour aider l'utilisateur dans sa quête de la connaissance

[Simon, 2000].

Ces outils correspondent, par exemple, aux moteurs de recherche que l'on trouve sur internet pour faciliter notre navigation, et dont de nouvelles versions, annoncées plus efficaces et plus rapides, sont proposées chaque jour, preuve que de nombreuses améliorations sont à faire. Derrière leurs apparences de simples dictionnaires, en plus de réaliser des opérations élémentaires de recherche d'information, ils utilisent différents modules d'interprétation de la requête de l'utilisateur d'une part et d'analyse de données et de traitement de l'information d'autre part.

Sans prétendre à l'exhaustivité, on peut tout de même essayer de mieux qualifier la nature de ces outils pour la fouille de données. Comme le but final de la recherche d'information est d'obtenir une information plus structurée exploitable en tant que connaissance et que cette recherche commence fréquemment par une requête d'un utilisateur humain, généralement en langage naturel, des outils de manipulation de symboles sont souvent utilisés. Ils peuvent être dérivés de techniques classiques de l'intelligence artificielle symbolique. Ils tentent donc de traduire le souhait de l'utilisateur en requête exploitable. Ils peuvent incorporer au processus de sélection de l'information des connaissances d'experts du domaine.

D'autre part, des outils numériques jouent aussi un rôle central dans la fouille de données. En effet, les bases de données contiennent, dans leur grande majorité, des données numériques. Trouver la donnée pertinente parmi les autres ou extrapoler une réponse à partir d'expériences comparables revient donc fréquemment à effectuer une analyse de données pour obtenir un résultat numérique. Les outils correspondants peuvent donc être dérivés de toutes les techniques numériques relatives au traitement et à l'analyse de données, aux statistiques ou aux mathématiques.

C'est dans ce cadre du développement d'outils numériques pour la recherche d'informations pertinentes et pour la structuration de bases de données afin d'extraire de nouvelles connaissances que s'inscrit la problématique de cette thèse. Plus précisément, nous proposons d'étudier comment une famille particulière d'outils numériques de traitement de données, les réseaux de neurones artificiels, peuvent proposer une méthodologie efficace dans cette problématique d'extraction de connaissances, avec les répercussions potentielles que cela peut avoir sur le domaine de la fouille de données.

Il convient tout d'abord de mentionner que les réseaux de neurones artificiels sont des candidats naturels pour réaliser ce genre de tâche. Il s'agit d'outils numériques de traitement de données, et de nombreux travaux antérieurs ont rapporté leur proximité avec le domaine des statistiques, des mathématiques ou de l'analyse de données [Ripley, 1996]. Ces mêmes travaux ont, de plus, souvent fait remarquer leurs qualités en termes de performances, de rapidité ou de simplicité, ce qui leur donne un atout supplémentaire. Cependant, les réseaux de neurones artificiels sont également connus pour l'opacité de leur fonctionnement interne. On les qualifie fréquemment de « boîte noire ». Si leur proximité avec d'autres outils numériques classiques et leurs bonnes performances sont avérées, on ne sait aujourd'hui pas complètement décrire les processus qui en sont la cause, ni les raisons exactes qui font qu'un réseau de neurones artificiels apprend tel ou tel comportement.

On trouve donc peut-être ici le défi principal de cette thèse : expliquer le réseau de neurones artificiels en montrant que son utilisation permet d'accéder à des connaissances et que son fonctionnement revient, par apprentissage, à sélectionner l'information pertinente

ou à se construire une représentation interne qui extrait implicitement des connaissances sous-jacentes.

Les deux manières principales permettant d'extraire de la connaissance d'une base de données à l'aide d'un réseau de neurones sont, tout d'abord et de manière élémentaire, de considérer qu'un réseau de neurones réalise une opération de traitement de données (par exemple une classification, une régression, une mise en correspondance, une prédiction, etc.) et qu'il offre généralement en sortie une information différente, souvent plus élaborée que celle qu'il a reçue en entrée [Thiria *et al.*, 1997]. Dans certains cas, si les données et le réseau sont bien choisis, cette information de sortie représente une nouvelle connaissance extraite de la base de données. L'objectif sera atteint si l'on sait bien expliquer et qualifier la nature de l'opération réalisée par le réseau de neurones artificiels. Prenons un exemple pour illustrer cette procédure : un neurone artificiel binaire avec une fonction d'activation en escalier évalue la somme pondérée de ses entrées et génère une sortie si cette somme est supérieure au seuil représenté par l'escalier. Cette opération élémentaire est en fait importante, car, à travers sa non-linéarité, le neurone prend une décision porteuse d'information. A partir d'un ensemble de données numériques, le neurone sait indiquer quand l'accumulation de ces données dépasse une limite et nous propose un signal clair (son activité de sortie) qui peut nous permettre de déclencher une action adéquate. L'enjeu pour l'utilisateur est donc de bien comprendre ce mécanisme (élémentaire ici) et de savoir l'exploiter (en fixant ici un seuil adapté).

Ensuite, de manière un peu plus élaborée, on peut considérer qu'au cours de son apprentissage et afin de produire des réponses pertinentes, le réseau construit, par modification de ses poids ou de son architecture, un modèle (ou une représentation) interne du phénomène qu'il étudie et dont il reçoit des réalisations sur ses entrées. Cette représentation interne, si on sait la décrypter, contient les connaissances intermédiaires que le réseau extrait afin de réaliser sa fonction. Ici, l'objectif sera atteint si l'on sait expliquer comment cette connaissance se construit, et si l'on sait la rendre explicite en allant la chercher là où elle est, dans la structure interne du réseau. Prenons ici aussi un exemple élémentaire : en montrant les limitations du perceptron simple, Minsky et Papert ont aussi montré que chaque neurone de la couche de sortie du perceptron réalise une séparation linéaire de l'espace d'entrée, et que cette séparation linéaire est réalisée par un hyperplan dont les coordonnées sont directement accessibles par les poids de ce neurone. Il est aujourd'hui très intéressant de se servir d'un perceptron, puisqu'on sait exactement ce qu'il va faire, et qu'à l'issue de l'apprentissage on peut récupérer la valeur des poids du réseau pour obtenir exactement les séparations qui ont été réalisées, ce qui constitue une connaissance (souvent implicite) très importante de l'opération de catégorisation.

Tout au long de notre travail, notre but va donc être de démonter les mécanismes internes de différents types de réseaux afin de mieux expliquer ce qu'ils font, comment ils le font, et de mettre en œuvre ces procédés afin d'extraire effectivement de la connaissance du problème qui nous sera posé. Dans cette mesure, il s'agit bien ici d'un travail d'explication puisque l'on s'attachera à montrer que, finalement, ces réseaux retrouvent, par apprentissage, une solution analytique connue. On peut cependant noter deux choses : premièrement, expliquer complètement un fonctionnement ne revient pas forcément à rendre l'outil moins attractif. On peut au contraire être intéressé par la performance réalisée. On sera en tout cas beaucoup plus sûr de ce que l'on fait si on connaît le processus sous-

jacent ; deuxièmement, si on sait complètement décortiquer un mécanisme pour un cas simple et que l'on retrouve effectivement la solution analytique idéale, il n'en reste pas moins que ce même processus sera également applicable, avec la même facilité, pour un cas beaucoup plus complexe où la solution analytique n'est pas forcément aussi facilement accessible. Ensuite, nous verrons que, si on retrouve de nombreuses similitudes avec d'autres techniques classiques, on peut également trouver des variantes originales ou une robustesse supplémentaire qu'il faudra souligner.

A la lecture de cette problématique, nous pouvons nous rendre compte que notre démarche est aussi pragmatique. Notre but est de proposer un cheminement qui soit utile pour qui veut extraire de la connaissance à partir de vastes bases de données peu structurées. Ce but sera rempli si nous adoptons une démarche qui aboutit à une procédure robuste, c'est-à-dire qui s'applique à un grand nombre de cas, sans modification majeure, et qui répond à un besoin réel. Pour montrer l'efficacité de notre démarche et la comparer à d'autres techniques existantes, nous l'appliquerons à un problème complexe, représentatif des besoins actuels. Pour bien insister sur l'intérêt pratique de nos travaux et pour avoir une référence constante à l'élucidation de différents aspects d'un problème complexe du monde réel, nous avons choisi de présenter cette application dès le premier chapitre de ce manuscrit et, lorsqu'elle sera connue, de s'en servir tout au long des chapitres qui suivent, comme illustration des résultats obtenus.

Cette application, qui sera donc présentée au chapitre 1, se situe dans le domaine des télécommunications et plus précisément dans celui de la téléphonie mobile. Elle a été réalisée dans le cadre d'une collaboration avec France Télécom R&D[1] à Belfort. Le problème de nos interlocuteurs concerne le placement des antennes de relais pour le téléphone portable. Ce problème implique de modéliser, ou à défaut d'estimer, l'atténuation du champ radioélectrique, en fonction de l'environnement traversé et des caractéristiques de l'antenne. Pour cela, on dispose de vastes bases de données avec des informations diverses (géographiques, techniques) dont on ne connaît pas véritablement la pertinence. Nous disposons également de connaissances *a priori* et de comportements de référence, dans la mesure où ce problème a déjà été abordé par différents modèles physiques, statistiques, symboliques ou hybrides. Au-delà de l'amélioration des performances, qui sont jusqu'à présent insatisfaisantes, nos interlocuteurs à France Télécom R&D sont également demandeurs de nouvelles connaissances sur ce phénomène de propagation des ondes. C'est ce que nous tenterons de préciser au cours du chapitre 1, mais nous pouvons d'ores et déjà souligner que ce difficile problème est bien représentatif du domaine dans lequel nous souhaitons nous inscrire. Au cours de notre travail, notre démarche a toujours été la même. Pour résoudre le problème principal (l'estimation de l'atténuation) ou un des problèmes annexes qui en découlent (par exemple la sélection des variables pertinentes), nous avons utilisé divers modèles de réseaux de neurones artificiels, et ces derniers se sont généralement révélés efficaces pour la tâche qu'on leur demandait. Au-delà de la performance, nous avons essayé de mieux comprendre ce que les résultats obtenus (les sorties du réseau, mais aussi sa structure, ses poids, les performances relatives de modèles concurrents) nous apprennent sur les données et sur les connaissances qu'elles renferment implicitement. Ceci passe par l'interprétation du fonctionnement du réseau et par la mise

1. Recherche & Développement

au point de techniques pour extraire de manière plus lisible cette connaissance.

Nous avons choisi de présenter les travaux réalisés sous deux thèmes qui représentent deux manières différentes d'acquérir de la connaissance par réseaux de neurones artificiels.

Dans le chapitre 2, nous nous intéresserons tout d'abord au cas le plus classique, la structuration par adaptation. Dans ce cas, le réseau a une structure prédéterminée et son expérience (son apprentissage) lui permet d'adapter cette structure au phénomène qui lui est soumis. Comme il s'agit généralement de modèles classiques du connexionnisme, ce chapitre sera pour nous l'occasion de présenter ces modèles (réseaux auto-organisés, réseaux à couches, réseaux récurrents), et surtout les caractéristiques qui nous seront utiles pour l'extraction de connaissances. De plus, nous montrerons les liens de ces modèles neuronaux avec des techniques statistiques classiques.

Dans le chapitre 3, nous explorerons un domaine plus avancé du connexionnisme, la structuration par construction, où l'apprentissage permet non seulement d'adapter un système neuronal au problème, mais aussi de lui construire une structure adaptée au problème. Nous étudierons le cas de la structuration dynamique d'un réseau, mais aussi des cas complexes où plusieurs modules coopèrent et où des structures non triviales de réseaux sont construites.

Dans ces deux chapitres, notre démarche sera la même. Pour chaque modèle présenté, nous expliquerons ce qu'il fait de manière interne et en quoi il manipule et construit de la connaissance. Puis, nous rapporterons les résultats obtenus sur notre application et les interpréterons à la lumière de ce qui aura été dit.

Finalement, nous tenons à préciser que, si nous avons évoqué, au début de cette introduction, la fouille de données, notre travail se cantonne à un aspect bien particulier de ce domaine. Il est plutôt relatif au domaine des réseaux de neurones artificiels et pourrait d'ailleurs être exploité dans d'autres cadres que la fouille de données. Concernant sa contribution proprement dite, dans le domaine des réseaux de neurones artificiels, nous présenterons au chapitre 3 un modèle neuronal original qui a été élaboré dans un souci d'efficacité pour l'extraction de connaissances, à partir de constatations faites sur des modèles existants. Notre contribution est aussi à rechercher au niveau de la démarche que nous proposons d'adopter et de notre volonté de faire admettre les réseaux de neurones artificiels comme des outils de choix pour l'analyseur de données et le cogniticien.

« Il vaut mieux avoir une solution approchée d'un problème réel que la solution exacte d'un problème trop simplifié »

Adaptation libre d'une pensée de John Tukey.
« Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise. »

Chapitre 1

Un exemple représentatif des problèmes réels actuels : une application de prédiction dans le domaine des télécommunications

L'amélioration des techniques d'intelligence artificielle ouvre de nouvelles perspectives.

Le présent chapitre fait état d'une application complexe, mal maîtrisée, dans un des domaines les plus en vue actuellement, les télécommunications. Cette application, réalisée dans le cadre d'une collaboration avec France Télécom R&D, est également fortement représentative des besoins et des difficultés rencontrés dans les problèmes réels pour lesquels le nombre des données est très élevé et la connaissance du domaine insuffisante.

Comme nous l'avons dit dans l'introduction, la masse des informations disponibles dans les applications actuelles pose un problème d'exploitation. L'apport de connaissances sur les données peut contribuer à l'amélioration des performances du système et à une meilleure compréhension du phénomène. Pour illustrer notre propos, nous nous proposons d'appliquer les développements théoriques réalisés dans ce mémoire et de valider les modèles informatiques développés en prenant pour cadre applicatif le problème de prédiction de l'atténuation du champ radioélectrique diffusé par les antennes émettrices utilisées par

la téléphonie mobile. Pouvoir prédire avec fiabilité cette atténuation permet de connaître la couverture d'une antenne, c'est-à-dire la zone où la réception sera de bonne qualité. Il s'agit donc là d'une étape fondamentale dans la phase de planification des réseaux téléphoniques cellulaires.

Pour mieux appréhender le contexte de cette application, nous commencerons par en exposer l'importance, aux niveaux scientifique et commercial. Nous présenterons les principes de la planification des réseaux cellulaires et la plate-forme de développement utilisée actuellement par France Télécom R&D et dans laquelle les logiciels que nous développerons ont vocation à s'insérer. Puis, nous décrirons les bases de données disponibles en insistant sur la diversité des types d'information qu'elles renferment. Nous exposerons les différents types de modèles de prédiction du champ radioélectrique, les différentes étapes de leur construction et la manière d'évaluer leurs performances. Nous décrirons la constitution des corpus sur lesquels porteront différentes analyses des données et à partir desquels nous essaierons d'améliorer les performances de prédiction par rapport aux modèles existants. Finalement nous conclurons ce chapitre en rappelant l'intérêt de pouvoir illustrer le fonctionnement des modèles de prédiction et d'analyse de données existants et d'appliquer notre contribution à un domaine stratégique mal maîtrisé.

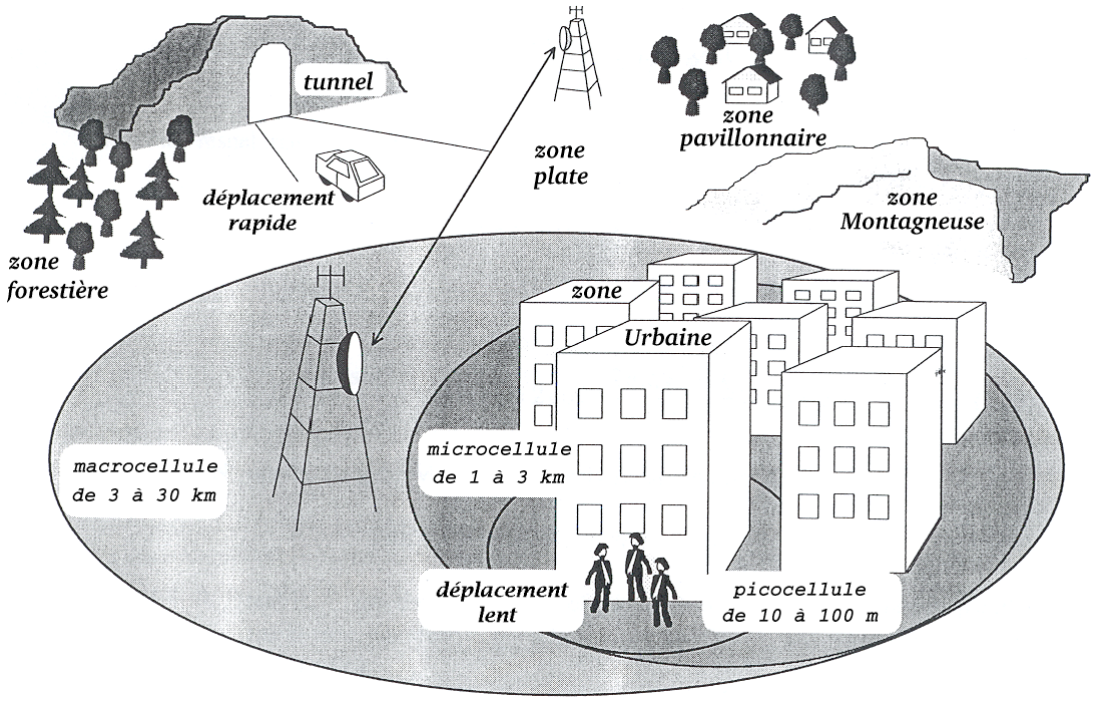


FIG. 1.1 – *La prédiction de la couverture d'une antenne émettrice d'un signal radioélectrique est un problème complexe qui dépend fortement de l'environnement de propagation (urbain, pavillonnaire, forestier, montagneux, etc.).*

1.1 De la planification cellulaire

L'optimisation de la couverture des réseaux cellulaires est un enjeu stratégique pour les nombreux opérateurs de téléphones portables car le développement d'un système efficace de transmission sans fil est un défi difficile à relever mais fondamental pour l'ingénierie des communications modernes puisqu'il ouvre les portes à un accès, pour chaque individu, non seulement à une conversation téléphonique mais aussi à toutes les sources d'information, quel que soit l'endroit où il se trouve.

La planification des cellules, c'est-à-dire des zones couvertes par les antennes, s'appuie sur la prédiction de l'atténuation du champ radioélectrique (figure **??**). La capacité à prédire l'atténuation du champ radioélectrique est donc l'élément déterminant qui conditionne le bon fonctionnement d'un réseau à moindre coût.

La propagation des ondes électromagnétiques est un phénomène complexe. Si dans l'air libre l'onde radioélectrique se propage suivant des lois physiques bien connues qui permettent d'obtenir un bon modèle théorique de propagation, en milieu urbain, où il est rare que la transmission du signal de l'émetteur au récepteur se fasse en ligne droite, les lois physiques deviennent difficilement applicables. En effet, les obstacles qui se trouvent sur la ligne directe d'une transmission ajoutent à la résistance de l'air des phénomènes de perturbation, de réflexion et de diffraction difficiles à évaluer. Si l'influence du phénomène de réflexion semble négligeable pour des fréquences de 900 MHz à 2 GHz utilisées en ville, il est important par contre de prendre en compte le phénomène de diffraction [Perrault *et al.*, 1996]. Les perturbations varient en fonction de la nature des obstacles, de leur positionnement, des caractéristiques de l'émetteur et du récepteur, et des paramètres du signal. Dans ces conditions le phénomène est si complexe que les modèles théoriques sont insuffisants pour modéliser l'atténuation de l'onde radioélectrique. Il est nécessaire d'ajouter des informations empiriques extraites de situations connues pour interpoler la valeur de l'atténuation à partir de ces situations. Dans ce cas, les modèles de propagation intègrent une phase d'apprentissage, dite supervisée, au cours de laquelle ils essaient de généraliser les exemples qui leur sont présentés pour extraire une fonction de prédiction qui puisse s'adapter correctement à de nouvelles situations. D'un autre coté, le développement des systèmes de communication par mobile en milieu urbain nécessite de réutiliser les mêmes fréquences et privilégie donc aujourd'hui l'utilisation de microcellules plutôt que de macrocellules. La multiplication des stations émettrices rend encore plus complexe la prédiction.

De nombreux modèles existent. On recense des modèles théoriques, empiriques et semi-empiriques ainsi que de multiples combinaisons qui essaient de tirer avantage des points forts de chacun. L'évaluation des performances des diverses solutions de planification cellulaire est gérée depuis un outil d'aide à l'ingénierie. Cette plate-forme de développement regroupe toutes les informations disponibles. Les différents modules de stockage et de traitement des données, de simulation et d'interprétation peuvent être sélectionnés pour évaluer sur une représentation aérienne la couverture d'une antenne (figure 1.2). Grâce à cet outil, il est possible de positionner une antenne sur la carte, de paramétrer ses carac-

téristiques et de simuler sa couverture. L'atténuation du champ peut alors être observée par un dégradé de couleurs correspondant à différentes intensités.

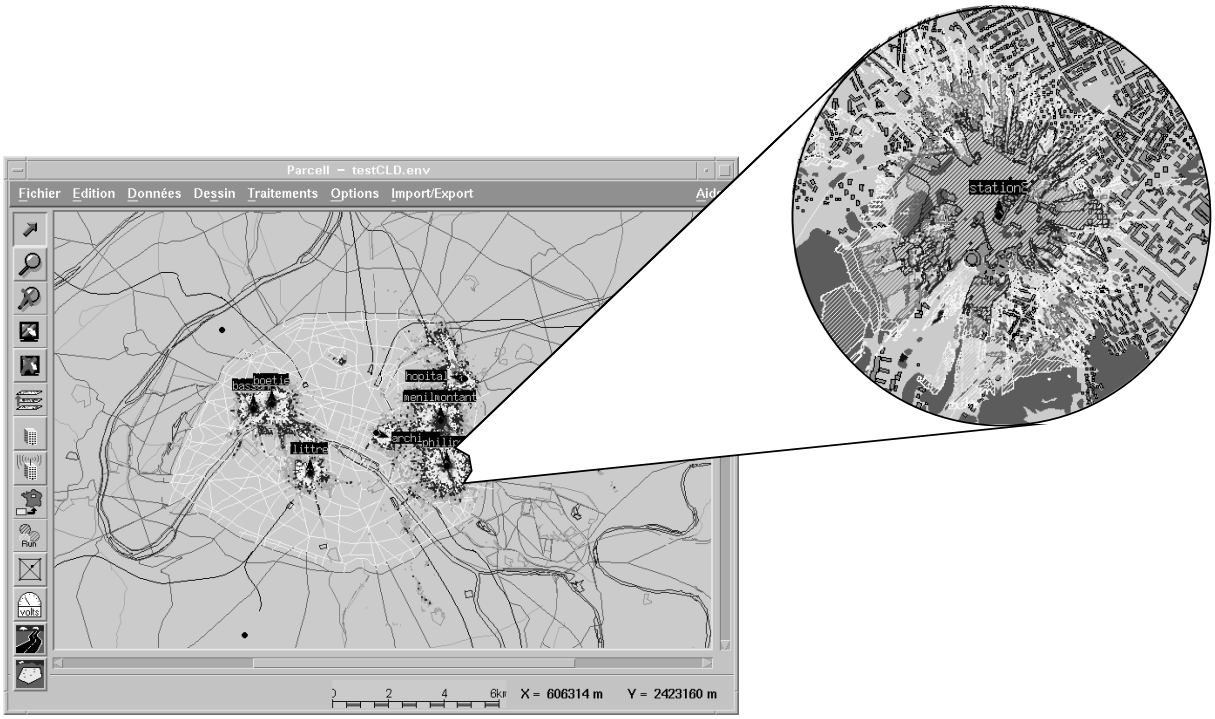


FIG. 1.2 – *Estimation de la couverture de Paris par sept stations émettrices depuis la plate-forme de développement PARCELL utilisée par France Télécom R&D. Au voisinage de chaque station, des zones de couleur différente permettent de visualiser l'atténuation du champ radioélectrique.*

La suite de ce chapitre décrit dans le détail les différents éléments constitutifs des modèles de propagation de France Télécom R&D, nécessaires à la prédiction du champ radioélectrique c'est-à-dire les données disponibles, les principes généraux des modèles existants et le contenu des corpus utilisés pour notre étude.

1.2 Bases de données

Les modèles actuels utilisés par France Télécom R&D sont donc des modèles semi-empiriques. Ils utilisent d'une part des bases de données géographiques contenant des informations sur le profil de liaison entre l'émetteur et le récepteur et d'autre part des mesures de champ servant de référence pour l'ajustement des modèles (figure 1.3). En effet, de nombreux paramètres nécessaire au calcul de phénomènes de diffusion et de diffraction ne sont pas accessibles dans une base de données géographique. Par conséquent, les paramètres empiriques déduits de mesures de champ rendent les modèles dépendant de l'environnement.

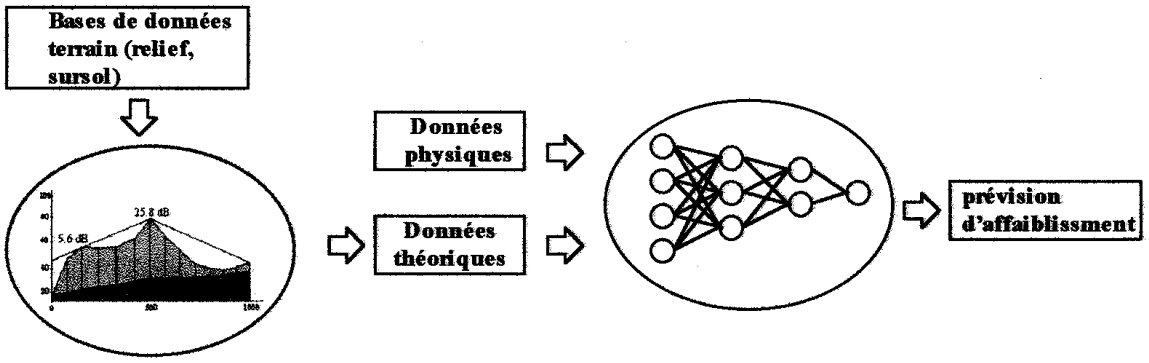


FIG. 1.3 – *Fonctionnement schématique d'un modèle neuronal de propagation. Les données d'entrée d'un modèle semi-empirique sont constituées par des paramètres physiques qui décrivent les caractéristiques de l'émetteur et du récepteur, le relief et la nature du sursol, et par des paramètres théoriques d'affaiblissement qui résultent de calculs de propagation dans l'espace libre et de phénomènes de diffraction appliqués au profil émetteur-récepteur.*

1.2.1 Données géographiques

France Télécom utilise, dans le domaine de la radio-mobile, des bases de données géographiques qui contiennent des informations sur le sol, le sursol[2], les contours administratifs, les pentes, les fonds de cartes, les fonds de plans vectoriels. Nous appuierons les méthodes de calcul de nos modèles de prédiction radioélectrique sur un sous-ensemble de ces bases de données terrain. Ce sous-ensemble décrit de manière plus ou moins fine la configuration du sol et du sursol. Les données utiles sont obtenues à partir de deux bases de données qui constituent une modélisation de l'environnement suivant différents critères.

Le modèle numérique de sursol (MNS) : le modèle numérique de sursol décrit le pourcentage d'occupation de 7 thèmes (eau, forêt, sol nu, roche, bâti diffus, bâti mixte, bâti dense) pour des mailles de 400m de coté sur tout le territoire français. Les mailles sont repérées par les coordonnées de leur centre par rapport à la maille située à l'extrême sud ouest de la zone. Les coordonnées sont exprimées en mètres.

Le modèle numérique de terrain (MNT) : le modèle numérique de terrain décrit l'altitude au sol prise tous les 100m sur la totalité du territoire français. Il s'agit d'une mesure ponctuelle et non d'une moyenne des altitudes rencontrées dans la maille. Les coordonnées du centre de la maille sont également connues suivant le même principe que le modèle MNS. Les coordonnées et l'altitude sont exprimées en mètres.

2. désigne ce qui se trouve au dessus du sol

1.2.2 Données des mesures de champ

Les modèles de propagation s'attachent à prédire la moyenne locale de l'enveloppe du champ radioélectrique en un point donné. Ces modèles, qu'ils soient théoriques ou empiriques, ne peuvent être validés qu'au moyen de mesures effectuées sur le terrain. De telles mesures permettent de relever la valeur instantanée (ou une valeur moyenne sur un court intervalle de temps) de l'enveloppe du champ électrique reçu le long d'un trajet parcouru par un véhicule. Une station émettrice fixe émet avec une puissance et une fréquence données. Un véhicule équipé d'un récepteur mesure le champ et relève les coordonnées géographiques, et ce pour des intervalles de distance fixes. L'amplitude du champ électrique mesuré est donnée sous la forme d'une tension exprimée en dBμV. Les mesures ainsi obtenues sont dites « brutes », rendant compte de l'ensemble des variations affectant le signal radio-mobile proportionnel à l'enveloppe du champ radioélectrique. Par la suite, on élimine les fluctuations rapides de ce signal afin de ne conserver que les variations lentes, seules prévisibles par les modèles de propagation du type considéré. Un sous-échantillonnage peut être appliqué pour réduire le nombre de données.
L'intervalle de confiance à 90% associé à l'estimateur de la moyenne des mesures est estimé entre +/- 1 dB et +/- 1.5 dB, pour différentes longueurs de moyennage. Il est important de connaître cette incertitude puisqu'elle influencera la conception et la validation des modèles de propagation.

La dépendance des modèles par rapport à l'environnement utilisé à la conception fait que les résultats obtenus dans un type d'environnement ne seront pas applicables à un autre.

1.3 Modèles de propagation

1.3.1 Les types de modèles

La littérature distingue trois catégories de modèles de propagation : les modèles théoriques, les modèles empiriques et les modèles hybrides.

Les modèles théoriques

Les modèles théoriques s'appuient sur une description physique de la zone (géométrie du terrain, hauteur des immeubles, densité de la végétation, etc.) et sur la théorie géométrique de la diffraction [Keller, 1962] ou bien sur un calcul de diffraction sur plusieurs arêtes [Deygout, 1966] pour évaluer, à partir de techniques de lancer de rayons, la valeur du champ comme étant une superposition d'ondes (directe, reflétées et réfractées). L'approche théorique présente l'intérêt de pouvoir être appliquée à tout type d'environnement. Mais, dans la pratique, ses modèles doivent être adaptés à la spécificité de l'environnement pour être efficaces. Ils demandent également un temps de calcul important. Pour mieux comprendre la composition de tels modèles, prenons, à titre d'exemple, le modèle théorique COST de Walfish et Ikegami. Ce modèle se base sur un profil dans lequel on prend en compte les pertes dues à la dernière diffraction au niveau du mobile (Lrts) et les pertes

dues à la diffraction multiple au-dessus d'arêtes alignées et espacées de manières équidistantes (Lmsd) [Walfish and Bertoni, 1988]. Son adaptation aux zones urbaines [Deldicque *et al.*, 1997] a fait intervenir les pertes dues à la distance à l'émetteur (L0) et les pertes dues à la diffraction sur une arête principale par la méthode de Deygout (Ldeg) [Deygout, 1966]. Les environnements rencontrés sont également pris en compte (ENV_i). L'affaiblissement prédit est une combinaison linéaire de ces différents termes sehsehh (équation 1.1). Les coefficients sont obtenus par régression linéaire.

$$aff = a.Lrts + b.Lmsd + c.L0 + d.Ldeg + \sum p_i ENV_i \tag{1.1}$$

Les modèles empiriques

Les modèles empiriques utilisent uniquement des données provenant de campagne de mesures effectuées sur des sites existants (ex. [Okumara *et al.*, 1968]). L'avantage des modèles empiriques réside dans leur capacité à s'adapter à toute sorte d'environnement pourvu qu'on dispose de mesures. Ces modèles sont généralement issus de méthodes statistiques telles que la régression linéaire. Mais les modèles statistiques mettent en jeu peu de paramètres. Ils ont donc peu de degrés de liberté ce qui limite leur capacité de modélisation. De plus, si de nouvelles données sont disponibles, il n'est pas possible de les intégrer au modèle déjà existant. Ces deux limitations peuvent être outrepassées par les réseaux connexionnistes.

Les modèles semi-empiriques

Les modèles semi-empiriques associent les avantages des deux catégories précédentes (ex. [Chan, 1991; Pérez Fontan and Hernando Rabanos, 1993]). Ils calculent de manière analytique les contributions au champ radioélectrique connues, comme la hauteur de l'antenne et la propagation d'une onde dans l'espace libre, et de manière empirique l'influence de la topologie, non maîtrisable parce que trop complexe, comme les propriétés électromagnétiques des matériaux ou la forme des toits. Les méthodes empiriques sont donc généralement utilisées pour ajouter un terme correctif à la prédiction du modèle théorique. Leur qualité dépend directement de la qualité du modèle théorique.

L'utilisation des réseaux de neurones dans les modèles

L'utilisation des réseaux de neurones artificiels pour prédire l'atténuation du champ radioélectrique est récente. Les premières études datent du début des années quatre vingt dix à travers, entre autres, un projet EURO-COST[3] [Stocker *et al.*, 1993]. Elles évoquent déjà l'intérêt d'utiliser des réseaux neuronaux dans des modèles hybrides en tant que terme correctif, l'intérêt d'introduire en entrée la hauteur de l'antenne et les capacités de généralisation des réseaux de neurones [Gschwendtner and Landstorfer, 1993]. Pour ce dernier cas, un réseau est entraîné à partir d'une représentation stylisée d'une montagne par la formule de Keller puis évalué sur une situation réelle présentant plusieurs montagnes. Ainsi, plusieurs travaux ont montré que la prise en charge, dans un

3. European cooperation in the field of scientific and technical research.

modèle semi-empirique, de l'estimation du terme correcteur lié à la spécificité de l'environnement par un réseau de neurones tel que le perceptron multicouches, en remplacement d'une méthode statistique, permettait d'améliorer la prédiction grâce à des capacités de modélisation plus flexibles (nombre plus important de degré de liberté, non linéarité) [Balandier *et al.*, 1995b; Balandier *et al.*, 1995a; Gschwendtner, 1993; Perrault *et al.*, 1996]. Ces mêmes travaux ont fait apparaître l'importance de spécialiser des réseaux sur des intervalles de distances émetteur-récepteur courts [Balandier *et al.*, 1995b; Balandier *et al.*, 1995a] voire sur des intervalles non disjoints [Caminada *et al.*, 1996].

D'autres types d'association entre un modèle théorique et un modèle connexionniste sont possibles, comme celle abordée à travers le modèle hybride COSTRN pour prédire l'affaiblissement en petites cellules à 900 MHz [Deldicque *et al.*, 1997]. Dans le cas où la visibilité entre l'émetteur et le récepteur existe, un modèle théorique prédit l'affaiblissement en se basant sur l'affaiblissement dû à la distance émetteur-récepteur et sur la fréquence de l'émetteur. En non visibilité, un modèle connexionniste prend en charge la prédiction de l'affaiblissement à partir de paramètres d'affaiblissement et de paramètres physiques.

Les réseaux de neurones proposent également des techniques de raffinement de modèles qui aboutissent à une sélection des paramètres les plus pertinents pour la prédiction (voir section 3.1.3 page 100). Des expérimentations ont été effectuées par France Télécom R&D à l'aide de ces techniques [Balandier and Governo, 1998]. Elles concernaient une base de paramètres théoriques et physiques, provenant donc de calculs d'atténuations mais aussi de mesures, comme la distance entre l'émetteur et le récepteur ou l'angle d'incidence, provenant d'un milieu urbain (Paris) pour la prédiction de l'atténuation du champ radioélectrique dans la bande des 900 MHz. La prédiction est difficile à cause de la disposition irrégulière des rues, de la variation des matériaux utilisés entre immeubles anciens et immeubles modernes et de l'étroitesse des rues. Ces travaux concluent à une amélioration des performances de prédiction et à la suppression de six des seize paramètres.

De manière générale, la comparaison entre un réseau de neurones et d'autres modèles semi-empiriques a montré qu'à partir des mêmes variables d'entrée (paramètres affaiblissement, données géographiques et mesures de champ) le réseau de neurones améliore significativement la prédiction.

1.3.2 Les contenus des modèles

Les modèles semi-empiriques sont les plus représentés parmi les modèles développés par France Télécom R&D. En général, une modélisation se base sur le principe d'une régression linéaire entre la « variable à expliquer », proche de l'affaiblissement mesuré, et les « variables explicatives » issues de la procédure suivante : dans un premier temps, les données géographiques vont permettre d'établir un profil du terrain contenu entre le point de mesure et son émetteur, rendant compte du positionnement des obstacles (figure 1.4). Dans un second temps, une partie des variables supposées liées aux phénomènes de propagation et d'affaiblissement du signal sont calculées à partir du profil établi. L'autre partie est constituée à partir des mesures et des données géographiques. Finalement, on sélectionne les attributs du modèle parmi les variables « explicatives ».

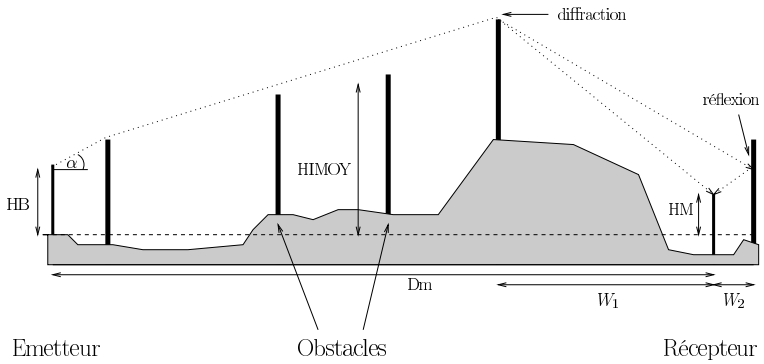


FIG. 1.4 – *Exemple de profil vertical synthétisé à partir d'un profil d'altitude (en gris) et d'un profil de hauteur (en noir). Différentes mesures peuvent être utilisées comme attribut du modèle, directement ou par combinaison, ou servir au calcul d'affaiblissements (voir tableau A.3 page 135 pour connaître leur sémantique).*

Extraction du profil

La plupart des modélisations ne prennent en considération que la partie du plan vertical contenant l'émetteur et le point de réception limitée par ces deux points. Les modèles actuels ne tiennent pas compte des phénomènes de réflexion et de diffraction provenant des objets situés au voisinage de ces limites. On distingue deux types de profil vertical : le profil d'altitude lié à la variation de l'altitude et le profil de hauteur lié au sursol. Ces deux profils sont obtenus à partir des données géographiques. De nombreuses méthodes permettent d'établir un profil d'altitude ou de sursol, de combiner plusieurs profils d'un même type et de synthétiser en un profil unique le profil d'altitude et le profil de sursol.

Attributs des modèles

Les données des mesures de champ et les données géographiques, évoquées dans la section relative aux bases de données, servent de composants élémentaires pour produire les attributs des modèles de propagation radioélectrique (la sortie étant l'affaiblissement du champ radioélectrique). La figure 1.5 schématise comment les attributs du modèle sont obtenus et présente les grandes catégories d'attributs utilisés généralement dans les modèles.

Sélection des attributs

Parmi la liste des variables explicatives, toutes ne seront pas retenues et ce, suite à deux sélections successives :

– Sélection par filtrage.
  D'un environnement à l'autre les attributs les plus pertinents changent. D'un type de modèle à l'autre également. En fonction du modèle, c'est-à-dire des besoins, on peut retenir uniquement des variables disponibles dans une zone particulière, avec des propriétés physiques spécifiques, une certitude suffisante, etc.

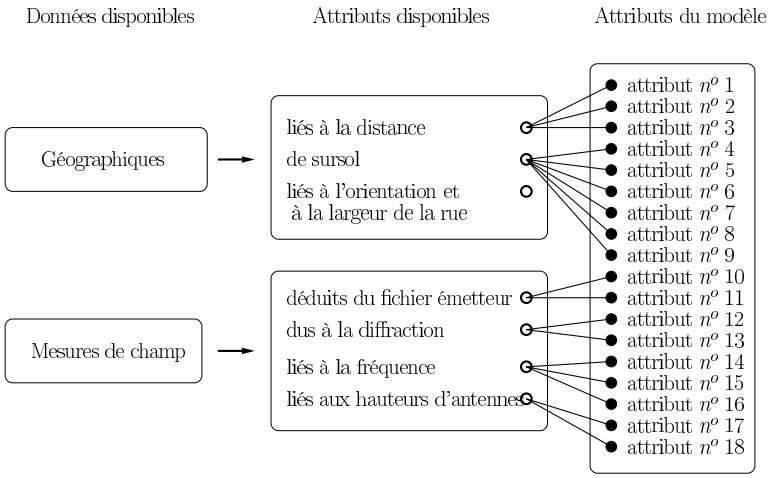


FIG. 1.5 – *Exemple de constitution d'un ensemble d'attributs pour un modèle de propagation.*

– Sélection par évaluation.
  Plusieurs études abordent la pertinence de différents paramètres et de différentes combinaisons à l'aide de réseaux de neurones [Deldicque *et al.*, 1997] [Caminada *et al.*, 1996] [Balandier and Governo, 1998]. Par exemple, des attributs exprimant le pourcentage moyen de différentes catégories de sursol, proportionnellement à leur présence le long du profil émetteur-récepteur, sont les plus pertinent pour un modèle théorique qui les associé à l'ellipsoïde de Fresnel. Il semblerait également que le nombre d'attributs utiles soit moins grand pour un modèle neuromimétique que pour un modèle théorique, et que des combinaisons d'attributs soient moins nécessaires en entrée d'un réseau, des combinaisons s'effectuant intrinsèquement [Caminada *et al.*, 1996].
  Les variables peuvent être ajoutées une par une au modèle. Une variable sera retenue si et seulement si :
  – sa corrélation partielle est la plus forte.
  – ses valeurs sont réparties de façon homogène.
  – son signe est cohérent avec la physique du phénomène.

Enfin, les situations qui génèrent de fortes erreurs ne sont généralement pas prises en considération, supprimant par là même certaines valeurs d'attributs qui correspondent *a priori* à des mauvais relevés.

1.3.3 Les critères d'évaluation des modèles

Les modèles de propagation prédisent la valeur d'atténuation du champ radioélectrique. L'erreur de prédiction, pour une situation donnée, correspond à la différence entre l'affaiblissement mesuré et l'affaiblissement prédit. L'évaluation des performances d'un modèle et la comparaison de modèles se font en utilisant plusieurs critères basés sur la distribution des erreurs de prédiction. Le modèle sera évalué comme bon si la distribution des erreurs de prédiction s'approche d'une distribution gaussienne centrée et de faible écart type. Donc la valeur moyenne et surtout l'écart type des erreurs de prédiction sont

évalués pour rendre compte de la forme de la distribution. De plus, France Télécom R&D utilise trois critères de qualité exprimant le pourcentage des situations dont la valeur absolue de l'erreur de prédiction est inférieure respectivement à 4dB, 6dB et 11dB. Ces pourcentages doivent être supérieurs respectivement à 50%, 70% et 90% de la population (figure 1.6) pour satisfaire aux exigences.

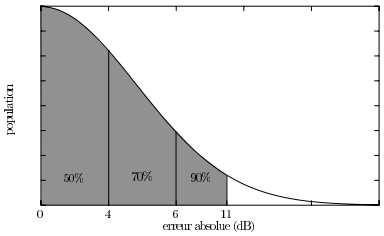


FIG. 1.6 – *Illustration des critères de qualité de France Télécom R&D. Pour que le modèle satisfasse le critère de qualité Q_4, 50% des valeurs d'atténuation prédites doivent avoir un écart inférieur à 4dB par rapport à leur valeur réelle, 70% doivent avoir un écart inférieur 6dB pour satisfaire le critère Q_8 et 90% doivent avoir un écart inférieur à 11dB pour satisfaire le critère Q_{11}.*

Afin de mieux nous rendre compte des performances pratiques des modèles, nous ne rapporterons pas la différence moyenne mais écart moyen (c'est-à-dire la différence absolue) que le modèle obtient par rapport à la valeur désirée. Par conséquent, les différents modèles présentés par la suite seront évalués par rapport à cinq mesures :

$\bar{e}$	: la valeur moyenne des valeurs absolues des erreurs de prédiction.
σ	: l'écart type des erreurs de prédiction.
Q_4	: le pourcentage des erreurs inférieures à 4 dB.
Q_6	: le pourcentage des erreurs inférieures à 6 dB.
Q_{11}	: le pourcentage des erreurs inférieures à 11 dB.

1.4 Corpus d'étude

Afin d'évaluer notre étude comparativement aux travaux réalisés jusqu'à maintenant dans le domaine de la planification de réseaux cellulaires, en plus des même critères d'évaluation cités en 1.3.3, il nous est apparu important d'utiliser des corpus relativement proches de ceux que France Télécom R&D utilise. D'abord pour réduire les différences entre notre approche et les leurs à une différence de modèle, ensuite parce que beaucoup des variables sélectionnées ont déjà fait l'objet d'études [Governo, 1997; Caminada *et al.*, 1996]. Par la suite, nous sommes susceptibles de modifier la composition des corpus en la justifiant par une analyse des variables.

Après discussion avec les représentants de France Télécom R&D, deux types de corpus ont été retenus. Le premier contient uniquement des variables géographiques et permettra d'observer les caractéristiques du sursol. Le second contient un large éventail de variables explicatives et permettra une étude approfondie de leur implication dans la prédiction de l'atténuation.

1.4.1 Le corpus géographique

Un corpus constitué uniquement de données géographiques va permettre de mieux connaître les caractéristiques du sursol indépendamment des caractéristiques de transmission. Cette information est importante car elle est invariante, dans la mesure où, dans cette application, il n'est pas possible d'agir sur l'environnement. Les conclusions auxquelles nous aboutirons par l'étude d'un tel corpus seront valables quelles que soient les caractéristiques de l'antenne.

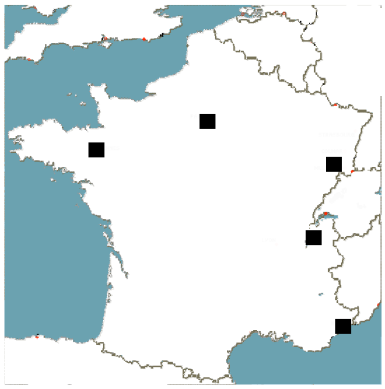


FIG. 1.7 – *Localisation des sites constituant le corpus géographique.*

France Télécom désire exporter son savoir faire en Europe et au delà. Pour que les conclusions portant sur les environnements puissent être valables dans un autre pays européen, il faut que le corpus géographique soit constitué par des zones représentatives de la géographie européenne. L'Europe peut être divisée en trois régions géographiques : l'Europe occidentale et centrale, l'Europe septentrionale et l'Europe méditerranéenne. La géographie française est incluse dans ces trois régions. Le relief et le sursol européen pourront donc être représentés par la géologie et la végétation climatique que l'on peut trouver en France. Ainsi, les trois ensembles géologiques (massifs anciens, chaînes récentes et bassins sédimentaires) et les différents types de sursol (plaine, forêt, ville, lac) pourront être décrits par 5 zones : Rennes, Paris, Belfort, Annecy et Nicev (figure 1.7).

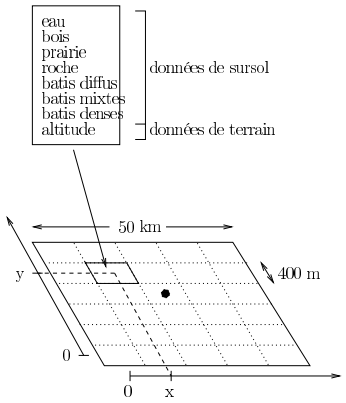


FIG. 1.8 – *Détails des informations contenues dans une zone géographique.*

Les zones mesurent 50 km de coté et sont centrées sur une ville. Une zone est divisée en parcelles de 400m de coté. Chaque parcelle regroupe des informations extraites des modèles numériques de sursol (MNS) et de terrain (MNT) à un pas de 400 m. Ainsi pour chaque maille, on possède les coordonnés x,y de son centre à l'intérieur de la zone, le pourcentage d'occupation pour les sept thèmes définis dans le format MNS et l'altitude au centre de la maille de 400 m (figure 1.8).

Nous disposons de 15625 exemples par zone, ce qui donne 62500 exemples pour le corpus d'apprentissage constitué de Rennes, Paris, Belfort et Annecy. Pour vérifier que les traitements mis au point suite à l'analyse des variables géographiques restent pertinents pour de nouvelles régions, nous ne faisons pas intervenir les données de la zone niçoise

dans le corpus principal mais nous les réservons à un corpus de test. La région niçoise est intéressante car elle comprend plusieurs types d'environnements, ce qui la rend difficile à traiter. Elle comprend également 15625 exemples.

Pour rendre compte du type de codage utilisé pour les données de sursol, on expose, à titre d'exemple, le début du fichier contenant les données relatives à la zone d'Annecy (tableau ci-dessous). x et y représentent la position de la parcelle dans sa zone. La somme des pourcentages d'occupation des 7 thèmes vaut 100.

x	y	eau	$bois$	$prairie$	$roche$	$b.diffus$	$b.mixte$	$b.dense$
0	0	0	8	92	0	0	0	0
0	400	0	0	100	0	0	0	0
0	800	0	37	62	0	0	0	0
0	1200	0	31	69	0	0	0	0
0	1600	0	14	86	0	0	0	0
0	2000	0	2	98	0	0	0	0
0	2400	0	20	77	3	0	0	0
0	2800	0	0	100	0	0	0	0
...	...	...	...	...	...	...	...	...

On remarquera que les pourcentages d'occupation d'une maille sont répartis dans seulement deux ou trois thèmes. Cette répartition entraîne donc de forts pourcentages pour les thèmes présents.

1.4.2 Le corpus complet

Le corpus complet comprend un ensemble de variables regroupant un large éventail des attributs disponibles à partir des données géographiques et des données de mesures de champ pour, d'une part, assurer une bonne modélisation et, d'autre part, permettre l'étude des variables les plus pertinentes.

Il constitue une base de données de situations réelles décrivant les conditions et la valeur de la propagation du signal radioélectrique dans 5 villes de France (Lyon, Mulhouse, Nice, Paris, Strasbourg). Les informations regroupées dans la base s'appuient sur les relevés effectués à partir des déplacements des véhicules d'acquisition de France Télécom R&D (figure 1.10). Les données disponibles ne correspondent donc pas au quadrillage d'une zone ni à un relevé systématique de toutes les artères d'une agglomération. La taille des sous-corpus correspondant à une ville est inégale et dépend du nombre et de la longueur des parcours, et du nombre d'antennes présentes dans la ville. La base d'évaluation comprend au total 50081 exemples de 33 paramètres dont la description est donnée dans l'annexe A page 135.

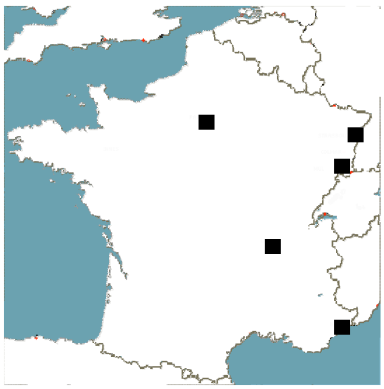


FIG. 1.9 – *Localisation des sites constituant le corpus complet.*

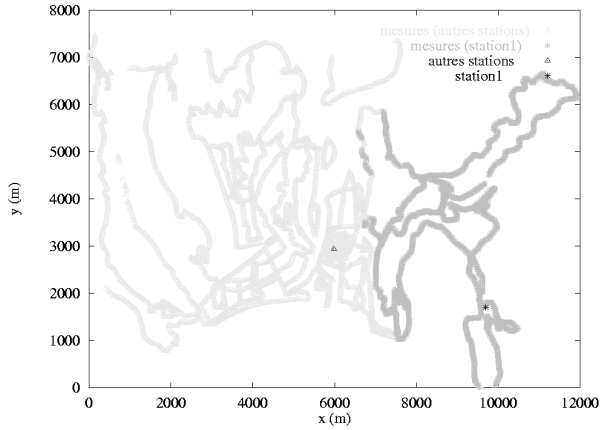


FIG. 1.10 – *Parcours des relevés liés à la station n°1 de Nice.*

1.4.3 Analyse descriptive

Plusieurs études statistiques ont été menées pour mieux connaître d'une part les caractéristiques générales de chaque variable, avec une attention particulière pour la variable à prédire, et d'autre part, leurs relations les unes avec les autres.

Remarque : l'histogramme et la régression linéaire ne s'appliquent qu'au corpus complet dans la mesure où seul ce corpus contient une variable particulière à prédire.

Histogramme de l'affaiblissement

L'affaiblissement du champ radioélectrique, ou plus exactement la combinaison d'affaiblissements représentée par la variable n°33 (AMES), est la plus importante des variables disponibles dans la base de données. C'est en effet sa valeur que le modèle doit prédire en fonction de la valeur des attributs (variables n°1 à n°32). Nous devons donc accorder une attention particulière aux réalisations de cette variable car les réalisations rendent compte de la fonction de densité, laquelle est fort utile pour appréhender la valeur d'une variable. La distribution des réalisations peut être approximée par un histogramme [Saporta, 1990]. La figure 1.11 page 21 présente l'histogramme de 180 classes d'amplitude unitaire pour un intervalle allant de 0 dB à 180 dB et contenant toutes les valeurs des réalisations. Les réalisations de l'atténuation sont distribuées sensiblement suivant deux gaussiennes centrées en 14.1 dB et 132.1 dB avec respectivement un écart type de 2.8 et de 16.3. Seules 1781 valeurs sont inférieures à 30 dB. La quasi totalité des atténuations mesurées (57250 exactement, soit 97 % des données) sont supérieures à 30 dB. Les réalisations inférieures à 30 dB appartiennent toutes à la même campagne de mesure (celle relative à la station n°3 de Strasbourg). Elles correspondent à deux zones de relevés relativement plates et dégagées par rapport à la station n°3. La considération de l'ensemble des données, sans distinguer le groupe des faibles atténuations de celui des fortes atténuations, n'est pas une bonne solution. Les paramètres statistiques issus de la totalité des réalisations de la variable n°33 donnent une moyenne de 128.5 dB et un écart type de 25.8. La distribution des réalisations est loin de correspondre à une distribution gaussienne ayant

les paramètres pré-cités. Les cas où l'affaiblissement est faible sont en nombre insuffisant pour être correctement assimilés par un modèle connexionniste unique. En revanche, des modèles plus complexes les traiteront efficacement. Nous garderons donc de coté les 3% des données correspondant au cas particulier de l'affaiblissement du signal dans un environnement dépourvu d'obstacles facilement modélisables par un modèle mathématique. Cette option n'est bien évidemment valable que si les situations qui génèrent un faible affaiblissement sont identifiables à partir de la valeur de leurs attributs. L'application de l'algorithme C4.5 de Quinlan [Quinlan, 1986] à un corpus 40000 d'exemples contenant 1215 cas d'affaiblissement sans obstacle permet d'obtenir un arbre de décision (voir figure A.1 dans l'annexe A) qui classifie correctement plus de 99,97% des 19031 exemples du corpus de test. Nous nous intéresserons donc, au niveau du corpus complet et sauf avis contraire, à l'ensemble des réalisations correspondant au cas général représenté par des valeurs d'atténuations supérieures à 30 dB et représentant 97% des données disponibles.

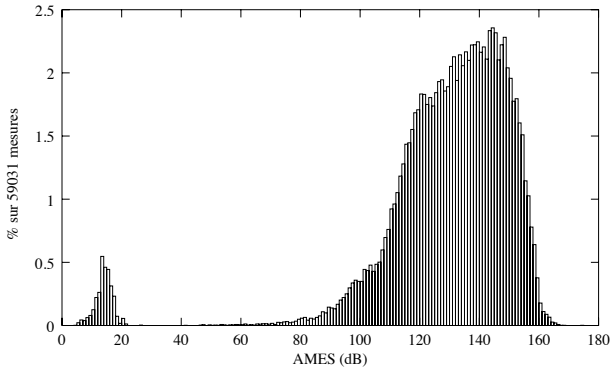


FIG. 1.11 – *Distribution des réalisations de l'atténuation du champ radioélectrique.*

Paramètres statistiques usuels

La distribution de chaque variable a été résumée par l'étude statistique des paramètres usuels (minimum, maximum, moyenne et écart type) sur l'ensemble des réalisations. Les résultats de ces études peuvent être consultés dans l'annexe A. Leur lecture nous indique que certaines variables pourraient être remplacées par une constante égale à la moyenne des réalisations au regard de leur faible écart type. Mais bien que les attributs n°5, n°10 et n°32 du corpus complet soient quasi constants et quasi nuls, leur influence sera étudiée de façon plus approfondie et leur éventuelle suppression se fera au cours de l'application de méthodes de sélection de variables.

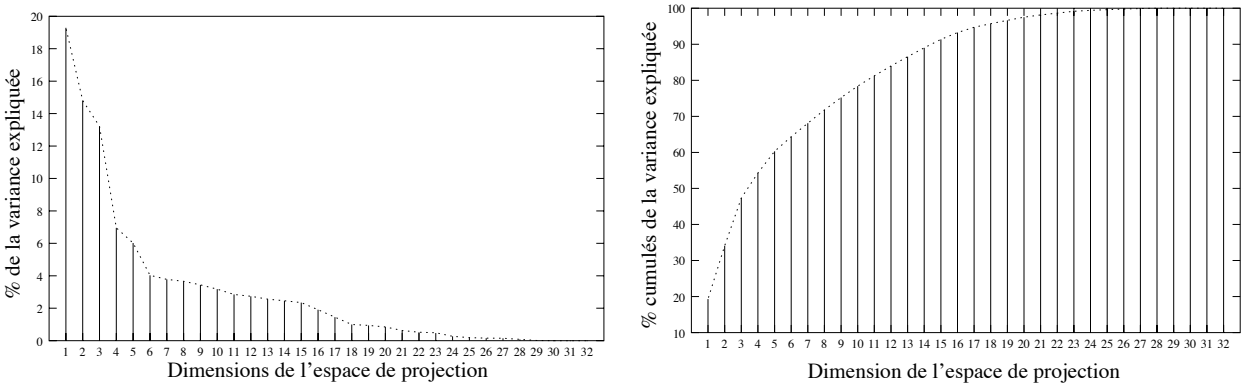
Corrélation

Une bonne mesure des liaisons entre les variables est donnée par les valeurs de la matrice des corrélations. Les tableaux correspondant à la matrice de corrélation du corpus géographique et du corpus complet sont inclus dans l'annexe A. On remarquera que certaines variables du corpus complet sont totalement corrélées (n°9 et n°30, n°21 et n°22). Les indices de corrélation supérieurs à $|0.7|$ ont été mis en gras. Une forte corrélation entre

la variable à prédire et les autres présente un réel intérêt. Avec des indices respectifs de 0.8, 0.7 et 0.7, on peut dire que l'affaiblissement est fortement corrélé à l'atténuation de la distance émetteur-récepteur, à la distance émetteur-récepteur et à la distance de l'émetteur à l'immeuble le plus haut. Une attention particulière sera portée sur ces variables lors de l'application des méthodes de sélection de variables.

Analyse en composantes principales

L'analyse en composantes principales est une méthode statistique qui permet de déterminer itérativement, dans l'espace des variables, les axes de projection sur lesquels les données ont la plus grande variance. Les vecteurs propres solutions de la diagonalisation de la matrice de variance-covariance des données sont les supports des axes de projection. Les valeurs propres indiquent, quant à elles, la variance des données sur leur axe respectif. Le rapport d'une valeur propre sur la somme des valeurs propres correspond au pourcentage de variance globale expliquée par la projection des données sur son axe (voir figure 1.12 (a)). Le cumul des pourcentages des plus fortes valeurs propres permet de se rendre compte de la quantité d'information préservée par la réduction de la dimension des données dans un sous-espace de dimension inférieure (voir figure 1.12 (b)).



(a) variance expliquée par chaque dimension de projection.

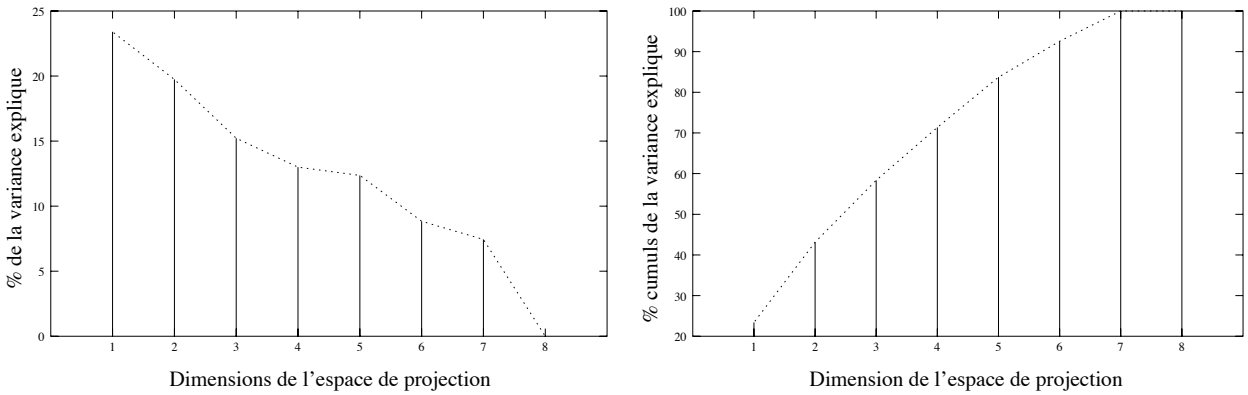
(b) variance expliquée par rapport au nombre de dimensions du sous-espace de projection.

FIG. 1.12 – *Analyse en composantes principales du corpus complet.*

Nous pouvons voir que 21 attributs, combinaisons des variables mesurées, suffiraient pour expliquer plus de 98% de la variance des données. Mais l'influence des variables mesurées sur la prédiction serait plus complexe à mettre évidence. Le traitement du problème réalisé par un modèle ne permettra pas une interprétation claire. De plus, le nombre de variables n'est pas excessivement important. Une fois encore, les méthodes de sélection de variables déduiront, si besoin est, le nombre de variables pertinentes et donc la dimension suffisante de l'espace d'entrée mais en conservant une lisibilité quant à l'interprétation de la modélisation. Signalons toutefois que 29 dimensions suffisent à expliquer la totalité de la variance. Ce qui signifie que trois variables peuvent être obtenues par combinaison linéaire des 29 autres. Néanmoins, nous les gardons pour évaluer l'efficacité des algorithmes

de sélection de variables.

L'analyse en composantes principales des attributs du corpus géographique est résumée sous la forme des deux graphiques de la figure 1.13 . Un attribut peut être obtenu par combinaison linéaire des autres, ce que nous savions déjà puisque la somme des 7 premiers attributs vaut 100. Ici, tous les attributs sont conservés pour éviter de nouvelles manipulations.



(a) variance expliquée par chaque dimension de projection.

(b) variance expliquée par rapport au nombre de dimensions du sous-espace de projection.

FIG. 1.13 – *Analyse en composantes principales du corpus géographique.*

Régression linéaire

Une régression linéaire classique basée sur une méthode de Kerlinger et Pedhazur [Kerlinger and Pedhazur, 1973], permet d'apprécier les performances obtenues par un modèle de prédiction statistique linéaire (tableau 1.1). Les performances du modèle ne satisfont pas les critères de qualités (cf. section 1.3.3). Il est donc nécessaire d'envisager des modèles plus performants.

Modèle	Phase	$\bar{e}$ (dB)	σ	Q4 (%)	Q6 (%)	Q11 (%)
régression linéaire	*Apprentissage*	4.84	3.98	50,8	68,9	92,3
	Test	4.91	4.02	50,4	68,5	91,8

TAB. 1.1 – *Performances d'une régression linéaire statistique sur le problème de prédiction du champ radioélectrique.*

1.4.4 Prétraitement

Les attributs ont subi des transformations nécessaires à leur bon traitement par des modèles numériques en général et par des réseaux neuromimétiques en particulier.

Normalisation

La standardisation est le traitement le plus adéquat pour rendre homogènes les variations des attributs. Chaque variable, à l'exception des variables binaires n°10 et n°12, est centrée-réduite en fonction de la moyenne et de l'écart-type calculés sur la totalité des données. Le fait d'inclure les données du corpus de test dans le calcul de la moyenne et de l'écart-type s'explique par le fait que dans l'application finale, la totalité des mesures pourront servir à estimer les paramètres de la moyenne et de l'écart-type afin d'obtenir une meilleure standardisation.

Répartition des corpus

1. Les corpus géographiques des villes de Rennes, Paris, Strasbourg et Belfort sont regroupés et normalisés pour constituer le corpus d'apprentissage. Le corpus géographique de la ville de Nice constituera le corpus de test.
2. Une fois normalisé, le corpus complet est réparti dans 5 corpus qui serviront à faire une validation croisée, de telle manière que chaque région soit équitablement représentée dans chaque corpus. Ensuite, quatre des cinq corpus seront groupés en corpus d'apprentissage, le cinquième servira de corpus de test.

1.5 Conclusion

Ce chapitre avait pour but de présenter l'état actuel des connaissances d'un problème particulier de télécommunication. Les données et les modèles mis en jeu dans la prédiction de l'atténuation du champ radioélectrique sont caractéristiques des problèmes réels actuels : les données sont présentes en grand nombre, nécessitant une phase d'investigation pour identifier l'implication réelle d'une variable observable, connaître les relations entre les variables observables, réduire le nombre de variables explicatives ; les modèles apportent des indications sur la manière dont les experts traitent le problème et sur les performances qu'ils obtiennent. Nous avons expliqué comment les variables d'entrée des modèles, c'est à dire leurs attributs, sont obtenues à partir des données observables, mettant par là même en évidence la grande variété de variables productibles et les différentes natures qu'elles peuvent avoir.
C'est la méconnaissance du phénomène qui est la source de la multiplication des variables et de l'obtention de performances sous optimales. Nous allons donc montrer tout au long de ce document comment extraire de la connaissance d'un problème à l'aide en particulier de réseaux neuromimétiques. Les corpus issus de la modélisation de l'atténuation du champ radioélectrique nous permettront d'illustrer notre propos de façon concrète.

Chapitre 2

Structuration par adaptation

La modélisation a pour but d'établir un ensemble d'équations et de relations servant à représenter et à étudier un système complexe. Le résultat d'une modélisation est la définition d'un modèle, c'est-à-dire une *fonction paramétrée*. Les modèles peuvent être définis à partir d'informations obtenues auprès des experts du domaine considéré ou publiées dans la littérature scientifique. Dans ce cas, les paramètres du modèle sont définis *a priori* en nombre et en valeur. Mais lorsque la qualité des sources d'information, la précision de l'estimation des paramètres, ou plus généralement, lorsque toutes ou une partie de ces informations manquent, ce qui constitue le cas le plus courant, les paramètres du modèle peuvent être estimés à partir d'une phase d'apprentissage basée sur un ensemble de situations connues. Ce processus consiste à structurer l'information pour rendre compte des relations liant la description d'une situation au résultat escompté. Par *structuration*, nous entendons une modélisation automatique aboutissant à des modèles ajustés au problème et interprétables. L'*adaptation* correspond au cas où la forme de la fonction est déterminée *a priori* par la forme même du modèle, et il ne reste plus qu'à ajuster ses paramètres. La structuration correspond au choix de la fonction et l'adaptation au choix des paramètres.

Les réseaux neuromimétiques sont utilisés pour des tâches aussi différentes que la discrimination, la catégorisation, la prévision, le contrôle, le diagnostic, etc. A chacune de ces opérations correspond généralement une famille particulière de réseaux. Quelles particularités internes expliquent le bon fonctionnement d'une famille de réseaux dans un traitement spécifique? Pour répondre à cette question, nous commencerons par introduire les fondements des méthodes d'analyse de données dont les techniques ont été développées pour réaliser des tâches comparables de réduction de la dimension du problème, de catégorisation, de classification, de régression et d'extraction de caractéristiques. Après avoir présenté les fondements des réseaux neuromimétiques, les liens que nous établirons entre les méthodes statistiques d'analyse de données et les réseaux neuromimétiques nous permettront d'apporter des explications supplémentaires sur le fonctionnement interne de certains réseaux de neurones. Ce chapitre est consacré aux grandes familles de réseaux neuromimétiques, dont nous expliquerons le fonctionnement, pour montrer que l'adaptation de ces modèles aux données du problème permet d'obtenir une représentation interne porteuse de sens qui renseignera sur le phénomène étudié. A partir de ces investigations sur des modèles fondamentaux, il sera plus facile d'appréhender des réseaux plus complexes qui construisent la fonction en adéquation avec les besoins et que nous dé-

velopperons dans le troisième chapitre. Le but de notre travail est, rappelons le, de mieux comprendre l'organisation des réseaux neuromimétiques pour mieux les exploiter et les améliorer. L'application réelle, présentée dans le chapitre un, nous fournira les éléments nécessaires à la validation de notre travail.

2.1 Les fondements de l'analyse multivariée

Malgré une différence de terminologie [Hastie and Tibshirani, 1994], les réseaux de neurones et les méthodes d'analyse multivariée ont des liens théoriques étroits [Friedman, 1994; Vapnik, 1995]. En dehors de la modélisation neurobiologique (voir par exemple [Reiss and Taylor, 1991; Sutton and Barto, 1981; Dehaene and Changeux, 1989; Grossberg, 1982]), les réseaux de neurones peuvent être vus comme une classe particulière de modèles statistiques, même si, par rapport aux méthodes statistiques classiques, cette classe présente des avantages que nous détaillerons. Aussi, la connaissance de techniques d'analyse multivariée, comme l'analyse factorielle et la catégorisation, permettra d'expliquer l'adaptation des réseaux de neurones à la lumière de la statistique. Nous présentons donc, entre autres, dans cette partie, le but et les stratégies sous-jacentes à l'analyse en composantes principales, à l'analyse en composantes indépendantes et à la quantification vectorielle pour expliquer dans la partie suivante une part des représentations internes obtenues par les réseaux neuromimétiques.

Les statisticiens ne raisonnent pas en termes de systèmes dynamiques, où l'on parle d'entrées et de sorties à un instant donné, mais sur un tableau de données avec des lignes et des colonnes, des individus et des variables. Les données proviennent d'un échantillon de variables observables pris sur un ensemble d'individus. Il n'existe aucune garantie que les variables observables soient effectivement les variables responsables du phénomène étudié et idéales pour la modélisation désirée. Les variables observables sont une représentation bruitée des variables internes au phénomène. Pour augmenter la probabilité que les variables internes puissent être expliquées par les variables observées, les données sont relevées en masse. On se retrouve donc confronté à un problème désormais classique d'analyse de données multidimensionnelles sur de grands échantillons pour déterminer des variables qui rendent compte au mieux du phénomène.

2.1.1 Analyse factorielle

L'analyse factorielle fait référence à un ensemble de techniques statistiques dont l'objectif commun est de représenter des variables observées par des variables latentes hypothétiques non observables : les facteurs. L'analyse factorielle sert à expliquer les corrélations entre les variables observées en utilisant un minimum de variables latentes (donc si les corrélations entre les variables sont nulles, l'analyse factorielle n'est pas intéressante). La logique de l'analyse factorielle est d'établir une explication causale entre le facteur latent et les facteurs observés. Comme il existe une indétermination inhérente aux structures causales, les méthodes d'analyse factorielle sont régies par trois principes : un principe

d'indépendance conditionnelle locale (qui fait prévaloir les solutions qui minimisent les coefficients de corrélation partielle), un principe de parcimonie (conduisant à préférer une solution constituée d'un minimum de variables), et un principe de la structure simple (qui fait appel à l'opérateur de rotation pour obtenir une représentation plus simple). Donc l'analyse factorielle recherche une solution simple et interprétable qui se traduit par la possibilité d'obtenir une représentation géométrique des variables observées.

En analyse de données, les données sont représentées par un tableau à deux dimensions dont les lignes correspondent aux individus (ou observations, équivalents aux exemples dans la terminologie des réseaux de neurones) et les colonnes correspondent aux variables observées. Les données sont donc contenues dans une matrice dans laquelle les n lignes représentent un nuage de n individus dans l'espace $\mathbb{R}^p$ des variables et les p colonnes représentent un nuage de p variables dans l'espace $\mathbb{R}^n$ des individus.

L'analyse en composantes principales

L'analyse en composantes principales (ACP) applique une matrice de rotation aux variables observables pour les disposer dans un nouveau système de coordonnées dans lequel la variance est maximisée, quelle que soit la dimension du sous-espace conservé. Les nouvelles variables sont décorrélées. La projection dans un sous-espace respecte la topologie. L'intention originale de Pearson en 1901 était de trouver, de manière automatique, l'équation de la ligne ou du plan qui permettait de visualiser des données multidimensionnées dans un espace à 1 ou 2 dimensions, en préservant au mieux la distribution initiale des points [Pearson, 1901]. Par la suite, Hotelling intégra l'analyse en composantes principales à la statistique mathématique [Hotelling, 1933].

Le principe de base de l'approche algébrique[4] de l'ACP est de considérer que si deux variables sont corrélées, elles ont au moins une source de variation commune. Pour la trouver, on analyse la variance des données en considérant que les variables X_1,X_2,X_3,..,X_p sont des combinaisons linéaires de facteurs non corrélés Y_1,Y_2,Y_3,..,Y_p. Les coefficients a_{ij} de la combinaison, c'est-à-dire les poids des facteurs, sont appelés *saturations* (équation 2.1).

$$X_1^n = a_{11}Y_1^n + a_{21}Y_2^n + ... + a_{p1}Y_p^n \tag{2.1}$$

Une saturation traduit l'importance d'un facteur dans l'explication des valeurs observées sur une variable (équation 2.1). La saturation est la corrélation entre une variable et un facteur. La somme, sur tous les facteurs, du produit des saturations observées entre deux variables donne la corrélation entre ces deux variables (figure 2.1).

Le carré de la saturation entre une variable et un facteur représente la variance de la variable expliquée par le facteur. La *communauté* h_i^2 est la variance d'une variable X_i expliquée par les d premiers facteurs. C'est donc la somme des carrés des saturations observées sur ces d facteurs pour cette variable :

$$h_i^2 = \sum_{j=1}^{d} a_{ji}^2$$

4. L'approche géométrique est basée sur la maximisation des distances entre les points après projection et aboutit à la même solution

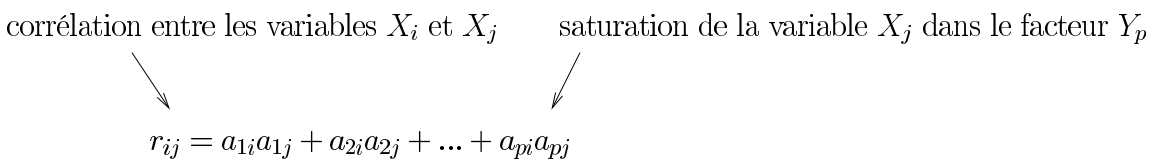


FIG. 2.1 – *La somme du produit des saturations observées entre deux variables X_i et X_j donne la corrélation entre ces deux variables.*

La somme de toutes les communautés divisée par le nombre d'individus soumis à l'analyse (et multipliée par 100) donne le pourcentage de variance totale expliqué par les facteurs. Cette valeur traduit l'importance du système de facteurs dégagés.

L'ensemble des saturations des p variables sur un facteur j définit ce qu'on appelle un *vecteur propre*.

$$\mathbf{v}_j = (a_{j1},a_{j2},...,a_{jp})^T$$

La *valeur propre* est la somme des carrés des saturations sur un facteur. La *valeur propre* représente la part de la variance totale expliquée par un facteur, c'est-à-dire la quantité d'inertie du nuage de points sur ce facteur.

$$\lambda_j = \sum_{i=1}^{p} a_{ji}^2$$

Le rapport $\frac{\lambda_j}{p}$ (multiplié par 100) de la valeur propre sur le nombre de variables soumises à l'analyse donne le pourcentage de variance expliquée par le facteur j. La valeur propre la plus élevée désigne le premier facteur car elle rend compte du maximum de variance. La seconde valeur propre rend compte du maximum de variance restant à expliquer. Elle désigne le second facteur. Et ainsi de suite... Cette propriété est due aux contraintes fixées lors de la méthode d'extraction des facteurs, qui se base sur le théorème suivant :

Théorème 1
Soit E_d un sous-espace portant l'inertie maximale, alors le sous-espace de dimension $d+1$ portant l'inertie maximale est la somme directe de E_d et du sous-espace de dimension 1 orthogonal à E_d portant l'inertie maximale.

Une démonstration du théorème est proposée dans [Saporta, 1990, page 167].

Finalement, la matrice constituée des vecteurs propres, rangés par ordre d'importance de leur valeur propre, projette les données dans un espace de représentation dont la dimension est égale au nombre de vecteurs engagés dans la matrice de transformation (voir « réduction de dimension » section 2.1.3). En pratique, pour obtenir les vecteurs propres et les valeurs propres, on procède à une diagonalisation de la matrice de corrélations des variables obtenue à partir des données.

L'analyse en composantes indépendantes

L'analyse en composantes indépendantes (ACI) [Hérault and Ans, 1984; Jutten and Hérault, 1991] est une extension de l'analyse en composantes principales. Alors que l'ana-

lyse en composantes principales cherche à décorréler les variables pour obtenir la relation sur les facteurs Y_i, $E[Y_1Y_2] = E[Y_1]E[Y_2]$ où E est l'espérance mathématique, l'analyse en composantes indépendantes cherche à rendre les variables statistiquement indépendantes, ce qui se traduit par $p(Y_1 = y_1, Y_2 = y_2) = p(Y_1 = y_1)p(Y_2 = y_2)$. Sous la condition d'indépendance, on a $E[Y_1Y_2] = E[Y_1]E[Y_2]$ mais la réciproque est fausse. L'ACI est donc bien une extension de l'ACP.
L'indépendance entre les facteurs est accrue si l'on utilise des moments d'ordre supérieur [Karhunen and Joutsensalo, 1994], généralement d'ordre 3 ou 4. L'ACI introduit des fonctions non linéaires impaires dans la règle d'apprentissage des coefficients de la matrice de transformation pour générer des moments d'ordre supérieur à deux (il suffit de développer en série de Taylor des fonctions impaires dérivables sur les facteurs pour voir apparaître dans le produit des moments d'ordre supérieur de la forme $y_1^{2q+1}y_2^{2r+1}$). La transformation n'en demeure pas moins linéaire. Les facteurs extraits des variables observables par ACP n'apportent pas toujours d'amélioration. L'ACI apporte une consistance aux facteurs. Cette méthode est utilisée pour la séparation de sources.

L'analyse en composantes curvilignes

Les méthodes précédentes sont basées sur des transformations linéaires. Une méthode de projection non linéaire, appelée analyse en composantes curvilignes, met également en correspondance l'espace des données avec un espace de dimension réduite. On commence par effectuer une quantification vectorielle de l'espace des données, puis on cherche les images des prototypes dans l'espace de projection à partir de la recopie locale de leurs distances deux à deux. Habituellement, lors d'une projection, on cherche à minimiser le critère :

$$E = \sum_i \sum_j (d_{ij} - d'_{ij})^2$$

où d_{ij} est la distance entre les points x_i et x_j dans l'espace des données et d'_{ij} est la distance dans l'espace de représentation des projetés y_i et y_j de x_i et x_j. Mais ce critère conduit uniquement à une représentation qui a subi une translation, une rotation ou une inversion d'axe. En considérant le critère :

$$E = \sum_i \sum_j (d_{ij} - d'_{ij})^2 f(d'_{ij})$$

où $f(.)$ est une fonction décroissante et monotone dont l'étendue est contrôlée par un paramètre λ ajustable (exemple figure 2.2). La fonction $f(.)$ défavorise les longues distances d'_{ij} entre des prototypes éloignés dans l'espace de représentation. Seules les distances les plus faibles sont prises en compte. La procédure tend à conserver la topologie au moins à un niveau local. L'analyse en composantes curvilignes réalise un dépliage des données. Pour plus d'informations sur l'analyse en composantes curvilignes, nous conseillons la lecture de [Demartines and Hérault, 1997].

L'analyse en facteurs communs et spécifiques

Le terme d'analyse factorielle (AF) est souvent réservé à l'analyse en facteurs communs et spécifiques [Spearman, 1904; Thurstone, 1947]. Elle suppose que les variables observées

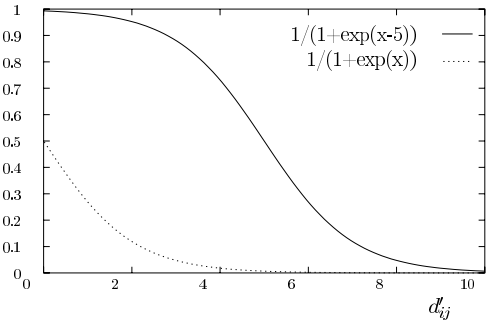


FIG. 2.2 – *La fonction* $f_\lambda(d'_{ij}) = \frac{1}{1+\exp(d'_{ij}-\lambda)}$ *utilisée en analyse en composantes curvilignes néglige les différences de distances* (d_{ij}- d'_{ij}) *avant et après projection pour les prototypes éloignés. Elle permet de conserver la topologie au niveau local.*

sont les résultantes de deux types de facteurs, ceux communs à plusieurs variables et ceux spécifiques à chacune des variables. Les facteurs spécifiques, aussi appelés facteurs uniques, ne contribuent en rien aux relations entre les variables. Le but de l'analyse factorielle va consister à expliquer ou rendre compte des corrélations observées et non plus de la variance totale des données comme dans l'ACP.

L'analyse factorielle des correspondances

L'analyse factorielle des correspondances (AFC ou AC) développée par Benzécri [Benzécri, 1973] permet de répondre aux situations dans lesquelles les observations donnent lieu à des tableaux de fréquences ou à des tableaux de contingence, c'est-à-dire lorsque les variables sont nominales. L'AC se généralise au cas où plus de deux ensembles sont mis en correspondance. La méthode la plus connue est l'analyse en correspondances multiples (ACM). Ces techniques s'appliquent à des variables nominales et par extension à des tableaux binaires. Elles ne seront pas plus détaillées dans ce manuscrit car notre approche est axée essentiellement sur des variables numériques à valeurs continues. Pour une étude approfondie de ces méthodes, on peut consulter par exemple [Lebart *et al.*, 1995].

2.1.2 Catégorisation

La grande quantité d'informations disponibles dans les bases de données rendent indispensables la sélection et le rangement. La catégorisation [Jain *et al.*, 1999], aussi appelée partitionnement, est une structure classificatoire particulière de la classification automatique qui répartit les données dans des groupes distincts et homogènes. D'un point de vue général, la classification automatique permet d'étiqueter un grand ensemble d'individus en rapprochant les individus sans étiquette d'individus étiquetés sur la base de leur similarité vis à vis d'une notion de distance. La classification automatique ne nécessite aucune connaissance *a priori*, en particulier sur le nombre et le genre des classes (dans le domaine des réseaux de neurone artificiels, on parlera d'apprentissage non supervisé). Elles permettent, d'une part, d'extraire de la connaissance des données au bénéfice de la compréhension du problème, et, d'autre part, de confronter les classes obtenues avec la conception qu'on s'en faisait. On retrouve les approches descriptive et exploratoire de l'analyse

factorielle. Pour que le système soit totalement efficace, le regroupement des objets par similitude doit s'accompagner d'une évolution permanente des classes pour permettre son adaptation à l'évolution des informations et la création de nouveaux concepts, apportant par là même de nouvelles connaissances. La classification automatique adapte les classes à l'évolution des données. Le partitionnement est adapté aux données récentes. Il est même possible d'intégrer un phénomène d'oubli à l'encontre des informations obsolètes. Les classements nous donnent une représentation simplifiée et ordonnée d'un domaine de connaissance, indispensable pour pouvoir prendre une décision. L'intérêt du classement ne se résume pas à la taxinomie. Par l'aspect structurel qu'il apporte, il peut être partie prenante de la réussite de la reconnaissance des formes. Les méthodes présentées dans cette section sont des méthodes statistiques de classification automatique couramment utilisées, telles que les méthodes des *k-means*, des centres mobiles et des nuées dynamiques. Les méthodes statistiques ne fournissent pas d'assistance à la construction de mesures de ressemblance entre les objets, mais elles permettent l'interprétation des résultats par l'application de tests statistiques performants.

L'une des difficultés provient de la nature multidimensionnée des données. Les unités des variables peuvent être différentes. De plus, les variables peuvent être corrélées. L'évaluation de la distance euclidienne entre deux individus ignore malencontreusement ces particularités. Or la notion de similitude entre deux individus est liée à la définition d'une distance, laquelle s'appuie sur l'utilisation d'une métrique particulière. Nous présenterons donc d'autres métriques, dont celle de Mahalanobis qui présente des propriétés intéressantes. Nous verrons également que les distances usuelles présentent une insuffisance géométrique, due à la dispersion des classes, particulièrement pénalisante lors de la recherche de l'affectation d'un individu. Ici encore, certaines métriques sont en mesure de remédier au problème.

Les méthodes de partitionnement

Notre présentation des méthodes statistiques de partitionnement s'appuie sur la décomposition faite par Lechevallier et présentée dans [Lechevallier, 1997].

Le but des méthodes de partitionnement est de construire une partition d'un ensemble d'éléments en regroupant ces derniers par similitude, afin d'obtenir les classes les plus homogènes possible.

Définition 1
Une partition d'un ensemble $E = \{e_1,e_2,...,e_N\}$ d'individus est un ensemble $P = (P_1,P_2,...,P_K)$ de K parties non vides de E, d'intersections vides deux à deux et dont la réunion forme E.

1. $\forall i = 1,...,K \quad P_i \neq \emptyset$
2. $\bigcup_{i=1}^{K} P_i = E$
3. $\forall i,j = 1,...,K$ et $i \neq j \quad P_i \cap P_j = \emptyset$

La structure classificatoire recherchée ici est une partition[5]. La recherche de la meilleure

5. D'autres structures classificatoires, telles qu'un recouvrement, une hiérarchie ou une pyramide,

structure nécessite un critère d'évaluation *a priori*. La solution recherchée est la partition qui satisfait le mieux ce critère. Généralement, le critère à minimiser est l'inertie intra-classe, qui exprime l'homogénéité des éléments qui composent une classe.

Si l'ensemble E contient un nombre fini de N individus, l'ensemble $\mathcal{P}_K(E)$ contenant les partitions possibles de E en K classes est fini et avoisine $\frac{K^N}{K!}$. Puisque le nombre de combinaisons est fini, la recherche exhaustive de la solution est possible mais rarement réalisable. En effet, signalons à titre indicatif que le nombre de répartitions de 100 individus dans 5 classes est supérieur à 10^{67}. Aussi, la recherche de la solution se fait généralement par optimisation itérative. On part d'une partition quelconque qu'on évalue par rapport au critère, puis on la modifie pour obtenir une nouvelle partition. La nouvelle partition est évaluée. Parmi l'ancienne et la nouvelle partition, la partition modifiée à l'étape suivante sera celle qui optimise le critère. L'algorithme s'arrête lorsque la nouvelle partition a déjà été rencontrée. La ou les partitions explorées appartiennent généralement au voisinage de la partition retenue à un instant donné. Le voisinage d'une partition est constitué des partitions qui diffèrent de la partition actuelle seulement de quelques individus et bien souvent d'un seul. La partition retenue vis à vis du critère correspond à une solution optimale locale.

Nous allons illustrer le fonctionnement des algorithmes de voisinage par les deux algorithmes les plus représentatifs : l'algorithme de transfert et l'algorithme des *k-means*. Tous deux utilisent l'inertie intra-classe comme critère d'optimisation. L'inertie intra-classe I_w est la somme de l'inertie I_j de chaque classe pondéré par le nombre d'éléments n_j de la classe (équation 2.2).

$$I_w = \sum_{j=1,...,K} n_j I_j = \sum_{j=1,...,K} n_j \sum_{i/e_i \in \{P_j\}} \frac{1}{n_j} d^2(\mathbf{x}_i, \mathbf{g}_j) \tag{2.2}$$

où j est l'indice de la classe, e_i le i$^{\text{ème}}$ élément et $\mathbf{x}_i$ son vecteur de description, $\mathbf{g}_j$ le centre de gravité de la classe j, et d est la distance euclidienne. La modification de l'inertie d'une classe suite à l'ajout d'un élément e_i à la classe j est :

$$I_j = I(P_j \cup \{e_i\}) = I(P_j) + 2\frac{n_j}{n_j + 1} d^2(\mathbf{x}_i, \mathbf{g}_j)$$

Le théorème de Huygens prouve la décroissance du critère.

Théorème 2 (théorème de Huygens)
Le moment d'inertie, par rapport à un point M, d'un système S pondéré $\{(x_1, n_1),...,(x_i, n_i),...\}$ de masse totale N est égal au moment d'inertie par rapport au centre de gravité G de S, augmenté du moment d'inertie par rapport à M d'une masse appliquée en G et égale à celle de S.

$$\sum_i p_i ||\mathbf{x}_i - \mathbf{m}||^2 = \sum_i p_i ||\mathbf{x}_i - \mathbf{g}||^2 + N||\mathbf{g} - \mathbf{m}||^2$$

Dans le cas probabiliste, N = 1 (somme des probabilités).

$$\sum_i \frac{1}{n} ||\mathbf{x}_i - \mathbf{m}||^2 = \sum_i \frac{1}{n} ||\mathbf{x}_i - \mathbf{g}||^2 + ||\mathbf{g} - \mathbf{m}||^2$$

existent et pourront être évoquées le cas échéant, mais ne seront pas abordées dans le détail.

Algorithme de transfert Le premier algorithme de voisinage présenté est un algorithme de transfert général. On part d'une partition $P = (P_1,...,P_K)$ de l'ensemble $E = \{e_1,e_2,...,e_N\}$ de N individus décrits par les vecteurs $\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N\}$. Chaque élément e_i appartient à une classe P_j de cardinal n_j et de centre de gravité $\mathbf{g}_j$. On considère les éléments les uns après les autres. Si le centre de gravité de la classe d'un élément n'est pas le plus proche, alors l'élément est transféré dans la classe dont le centre de gravité est le plus proche et les deux centres de gravité sont recalculés. L'algorithme s'arrête quand plus aucun transfert n'est à effectuer. L'algorithme n°1 page 33 détaille la procédure.

Algorithme 1 Algorithme de transfert

Données: Le nombre d'individus n_j pour chaque classe j
Une partition $P = (P_1,...,P_K)$
Résultats: Une autre partition

1. Initialisation
   pour tout j de 1 à K **faire**
   Calculer le centre de gravité $\mathbf{g}_j$ de la classe
   fin pour
2. Etape itérative
   $test \leftarrow 0$
   pour tout i de 1 à N **faire**
   Déterminer la classe de e_i, notée s
   Déterminer c tel que $c = \arg\ \min \left\{ \min_{\substack{j=1,...,K \\ j \neq c}} \left\{ \frac{n_j}{n_j+1} d^2(\mathbf{x}_i,\mathbf{g}_j) \right\}, \frac{n_s}{n_s-1} d^2(\mathbf{x}_i,\mathbf{g}_s) \right\}$
   si $s \neq c$ **alors**
   $test \leftarrow 1$
   $\mathbf{g}_c \leftarrow \frac{n_c \mathbf{g}_c + \mathbf{x}_i}{n_c+1}$
   $\mathbf{g}_s \leftarrow \frac{n_s \mathbf{g}_s - \mathbf{x}_i}{n_s-1}$
   $P_c \leftarrow P_c \cup \{e_i\}$
   $P_s \leftarrow P_s - \{e_i\}$
   $n_c \leftarrow n_c + 1$
   $n_s \leftarrow n_s - 1$
   fin si
   fin pour
3. Test de continuation
   si $test = 0$ **alors**
   Aller en 2
   fin si

Algorithme des *k-means* Le second algorithme de voisinage présenté est l'algorithme des *k-means* de Mac Queen [MacQueen, 1967]. Pour minimiser l'inertie à chaque étape, il suffit de trouver au moins un centre de gravité plus proche de l'élément considéré que celui auquel il est actuellement associé. Aussi, l'algorithme des *k-means* utilise la même procédure que l'algorithme n°1 mais n'évalue pas toutes les distances. L'élément est transféré dans la classe du premier centre de gravité établi plus proche que celui de la classe actuelle. L'algorithme complet est donné page 34. Le nom de l'algorithme fait référence à l'utilisation des k centres de gravité pour partitionner les individus.

Algorithme 2 Algorithme des *k-means*

Données: Le nombre d'individus n_j pour chaque classe j
Une partition $P = (P_1,...,P_K)$
Résultats: Une autre partition
1. Initialisation
pour tout j de 1 à K **faire**
Calculer le centre de gravité $\mathbf{g}_j$ de la classe
fin pour
2. Etape itérative
$test \leftarrow 0$
pour tout i de 1 à N **faire**
Déterminer la classe de e_i, notée s
Déterminer c tel que $c = \arg \min_{j=1,...,K} d^2(\mathbf{x}_i,\mathbf{g}_j)$
si $s \neq c$ **alors**
$test \leftarrow 1$
$\mathbf{g}_c \leftarrow \frac{n_c\mathbf{g}_c+\mathbf{x}_i}{n_c+1}$
$\mathbf{g}_s \leftarrow \frac{n_s\mathbf{g}_s-\mathbf{x}_i}{n_s-1}$
$P_c \leftarrow P_c \cup \{e_i\}$
$P_s \leftarrow P_s - \{e_i\}$
$n_c \leftarrow n_c + 1$
$n_s \leftarrow n_s - 1$
fin si
fin pour
3. Test de continuation
si $test = 0$ **alors**
Aller en 2
fin si

D'autres algorithmes étendent le voisinage contenant les partitions qui peuvent être évaluées à l'ensemble $\mathcal{P}(E)$ des partitions. Dans ce cas, la totalité des individus est ré-

affectée avant la mise à jour des centres de gravité, c'est-à-dire avant que la totalité des individus soit réaffectée d'une partition à l'autre. Deux étapes s'enchaînent : à chaque itération, chaque individu est réaffecté à la classe dont il est le plus proche (étape d'affectation) puis chaque centre de gravité est recalculé (étape de représentation). L'ordre de présentation des individus n'a plus d'influence sur le résultat.

Algorithme des centres mobiles La différence entre l'algorithme des centres mobiles [Forgy, 1965] et l'algorithme des *k-means* réside dans la mise à jour des centres de gravité. Ici, le centre de gravité est mis à jour après le passage de tous les individus. On peut démontrer qu'une solution localement optimale de l'algorithme des centres mobiles est une solution de l'algorithme des *k-means*, et inversement. Le critère à optimiser est toujours l'inertie intra-classe. L'algorithme est donné page 36.

Algorithme des nuées dynamiques La particularité de l'algorithme des nuées dynamiques [Diday, 1971] est qu'il ne fait pas nécessairement appel au centre de gravité pour attribuer un élément à une classe mais à un noyau. Le noyau d'une classe peut être constitué par un ensemble d'éléments centraux, une droite principale ou un plan principal, ou une loi de probabilité. Il faut en fait avoir une fonction de représentation L qui associe à l'ensemble des éléments d'une classe son noyau. Il faut également une fonction d'affectation ϕ qui associe à l'ensemble des éléments leur classe. La distance entre un élément x et son noyau L est donnée par une fonction positive $d(\mathbf{x},L)$. Une classe P_j est définie par $\{e \in E/\phi(\mathbf{x}) = j\}$. Le critère à optimiser est donné par une fonction D qui mesure l'adéquation entre les éléments d'une classe P_j et un noyau L. On a $D(P_j,L_j) = \sum_{\mathbf{x}\in P_j} d(\mathbf{x},L_j)$. L'algorithme complet est donné page 37.

A l'issue d'une classification automatique, il est nécessaire de se poser la question difficile, mais souvent nécessaire, de la validité des classes obtenues. Trois critères, ou plutôt trois types de notions, sont habituellement employés pour justifier l'existence d'une classe : sa compacité, son isolation et sa stabilité. Mais la formalisation mathématique de chacune de ces notions n'est pas réellement satisfaisante. Une approche, non paramétrique, basée sur la technique du « Bootstrap », peut tester à la fois l'isolation, la compacité, et divers degrés de stabilité d'une classe. Dans le même ordre d'idée, le choix du nombre de classes est souvent évoqué mais rarement abordé. L'article de Bock fournit un début de réponse à ce problème [Bock, 1985] (cf. également section 3.1.2).

Le rôle de la métrique

La notion de similarité entre les éléments, que l'on rencontre dans toutes ces méthodes, nécessite de définir une mesure de distance entre les individus. Différentes distances peuvent être utilisées. Toutes se réfèrent à une métrique spécifique. Le choix de la métrique va modifier la mesure de similarité et donc le processus de partitionnement. Dans le cas général, celui de la physique, les dimensions sont des longueurs, elles ont toutes la même unité. La distance entre deux points est la distance euclidienne donnée

Algorithme 3 Algorithme des centres mobiles

Données: Le nombre d'individus n_j pour chaque classe j
Une partition $P = (P_1,...,P_K)$
Résultats: Une autre partition
1. Initialisation
pour tout j de 1 à K **faire**
Calculer le centre de gravité $\mathbf{g}_j$ de la classe
fin pour
2. Etape d'affectation
$test \leftarrow 0$
pour tout i de 1 à N **faire**
Déterminer la classe de e_i, notée s
Déterminer c tel que $c = \arg \min_{j=1,...,K} d^2(\mathbf{x}_i,\mathbf{g}_j)$
si $s \neq c$ **alors**
$test \leftarrow 1$
$P_c \leftarrow P_c \cup \{e_i\}$
$P_s \leftarrow P_s - \{e_i\}$
fin si
fin pour
3. Etape de représentation
pour tout j de 1 à K **faire**
$\mathbf{g}_c \leftarrow \frac{n_c\mathbf{g}_c+\mathbf{x}_i}{n_c+1}$
$\mathbf{g}_s \leftarrow \frac{n_s\mathbf{g}_s-\mathbf{x}_i}{n_s-1}$
$n_c \leftarrow n_c + 1$
$n_s \leftarrow n_s - 1$
fin pour
4. Test de continuation
si $test = 0$ **alors**
Aller en 2
fin si

par la formule de Pythagore :

$$\begin{aligned} d^2(\mathbf{x}_i,\mathbf{x}_j) &= \sum_{k=1}^{p}(x_i^k - x_j^k)^2 \\ &= (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j) \end{aligned}$$

Souvent, les individus sont exprimés suivant des caractères dont les dimensions sont différentes. Il peut être nécessaire de contrôler une dimension par rapport aux autres

Algorithme 4 Algorithme des nuées dynamiques

Données: Une partition $P = (P_1,...,P_K)$
Résultats: Une autre partition
1. Initialisation
  pour tout j de 1 à K **faire**
    Définir le noyau L_j de la classe
  fin pour
2. Etape d'affectation
  $test \leftarrow 0$
  pour tout i de 1 à N **faire**
    Déterminer la classe de e_i, notée $s = \phi(e_i)$
    Déterminer c tel que $c = \arg \min_{j=1,...,K} d(\mathbf{x}_i,L_j)$
    si $s \neq c$ **alors**
      $test \leftarrow 1$
      $P_c \leftarrow P_c \cup \{e_i\}$
      $P_s \leftarrow P_s - \{e_i\}$
    fin si
  fin pour
3. Etape de représentation
  pour tout j de 1 à K **faire**
    Calculer le nouveau noyau L_j de la classe P_j qui vérifie $D(P_j,L_j) = \min_{L \in \mathcal{L}} D(P_j,L)$
  fin pour
4. Test de continuation
  si $test = 0$ **alors**
    Aller en 2
  fin si

pour que l'importance de la variation d'un caractère soit comparable à celle d'un autre caractère. Par exemple, les données géographiques de notre application sont constituées de pourcentages et de l'altitude. La variation entre un pourcentage d'eau de 0 et un pourcentage de 90 est très importante du point de vue de la signification, mais on peut supposer que la variation entre les altitudes de 1400m et de 1500m n'a pas la même importance. Pourtant, une parcelle décrite par le relevé (0,.,.,.,.,.,.,.,1400) sera évaluée plus similaire, du point de vue de la distance euclidienne, au relevé (90,.,.,.,.,.,.,.,1400) qu'au relevé (0,.,.,.,.,.,.,.,1500). La solution est de pondérer les variations. La formule précédente

devient :

$$\begin{aligned} d^2(\mathbf{x}_i,\mathbf{x}_j) &= \sum_{k=1}^{p} \alpha_k (x_i^k - x_j^k)^2 \\ &= (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) \quad \text{avec } \mathbf{M} = \begin{pmatrix} \alpha_1 & & & 0 \\ & \alpha_2 & & \\ & & \ddots & \\ 0 & & & \alpha_p \end{pmatrix} \end{aligned}$$

Pondérer $(x_i^k - x_j^k)^2$ par α_k revient à multiplier x_i^k et x_j^k par $\sqrt{\alpha_k}$ ($\alpha_k \geq 0$).

La formule de Pythagore n'est valable que si les axes sont orthogonaux. En statistique rien ne l'oblige. Il est donc possible de prendre n'importe quelle matrice $\mathbf{M}$ symétrique définie positive. A une matrice $\mathbf{M}$ correspond une métrique. Les choix les plus courants pour la matrice $\mathbf{M}$ sont :

1. Matrice identité ($\mathbf{M} = \mathbf{I}$)
   Les coefficients de la diagonale α_k valent 1. On retrouve la distance euclidienne correspondant au produit scalaire usuel. Les variables pour lesquelles les différences entre les individus sont les plus fortes, i.e. les variables les plus dispersées, influencent plus nettement le résultat du calcul.

2. Matrice diagonale des inverses de variances ($\mathbf{M} = \mathbf{D}_{1/s_k^2}$)
   Les coefficients de la diagonale sont égaux à l'inverse des variances $\alpha_k = 1/s_k^2$. La distance ne dépend plus des unités des variables puisque tous les caractères ont une variance de 1. L'influence d'une variable sur la distance est la même pour toutes. x_i^k/s_k^2 est sans dimension.

3. Matrice inverse de la matrice de variance-covariance ($\mathbf{M} = \mathbf{\Sigma}^{-1}$)
   La matrice $\mathbf{M}$ n'est plus diagonale mais égale à $\mathbf{\Sigma}^{-1}$, où $\mathbf{\Sigma}$ est la matrice carré d'ordre p de variance-covariance des variables X_1,X_2,...,X_p définie par :

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1^2 & \text{cov}(X_1,X_2) & \cdots & \text{cov}(X_1,X_p) \\ & \sigma_2^2 & & \vdots \\ \vdots & & \ddots & \\ & \cdots & & \sigma_p^2 \end{pmatrix}$$

   où σ_p^2 est la variance de X_p.
   Pour comprendre l'intérêt de la métrique définie par la matrice inverse de la matrice de variance-covariance, reprenons l'exemple des données géographiques de l'application. 7 variables décrivent le pourcentage d'occupation des thèmes eau, forêt, sol nu, roche, bâti diffus, bâti mixte et bâti dense pour une parcelle de 400m de coté. Il n'est pas étonnant de constater que le pourcentage de bâti diffus et le pourcentage de bâti mixte sont corrélés positivement. La variation du pourcentage de bâti diffus va aller de pair avec la variation du pourcentage de bâti mixte. Donc, si on se

rapproche de la ville, le pourcentage de bâti mixte augmentera avec le pourcentage de bâti diffus. La variation de la distance entre les valeurs du point de départ et les valeurs du point d'arrivée semblera importante. On aura l'impression de s'être plus éloigné qu'on ne l'est en réalité. Il en résultera une erreur d'appréciation. Deux individus peuvent paraître fortement éloignés si des variables sont fortement corrélées. L'utilisation de la matrice inverse de la matrice de variance-covariance en tant que métrique va permettre de décorréler les variables avant d'appliquer le produit scalaire. La mesure de similarité est plus juste puisqu'elle s'affranchit des covariances.

4. Matrice inverse de la matrice de corrélation ($\mathbf{M} = \mathbf{R}^{-1}$)
   La métrique définie par la matrice $\mathbf{M} = \mathbf{R}^{-1}$, où $\mathbf{R}$ est la matrice de corrélation, regroupe les propriétés des métriques $\mathbf{M} = \mathbf{D}_{1/s^2}$ et $\mathbf{M} = \mathbf{\Sigma}^{-1}$ définies ci-dessus. Elle permet non seulement d'avoir des dimensions du même ordre de grandeur, mais également d'avoir des variables non corrélées. En effet, si les variables X_i sont réduites, la matrice de variance-covariance $\mathbf{\Sigma}$ s'identifie avec la matrice de corrélation $\mathbf{R}$.

$$\mathbf{R} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1p} \\ & 1 & & \vdots \\ \vdots & & \ddots & \\ & \cdots & & 1 \end{pmatrix}$$

   La métrique ainsi définie est connue sous le nom de métrique de Mahalanobis et renvoie à la distance du même nom.

Nos données étant normalisées, la métrique la plus intéressante est la métrique ($\mathbf{M} = \mathbf{\Sigma}^{-1}$). Mais la dispersion des classes fait intervenir, lors de l'affectation d'un nouvel individu à une classe, des considérations supplémentaires que nous allons expliquer.

L'affectation

Ayant trouvé le meilleur partitionnement des n individus en k classes, on peut alors chercher à affecter un nouvel individu à l'un des groupes. La règle naturelle consiste à calculer les distances de l'individu à classer au centre de gravité de chaque classe et à affecter cet individu à la classe dont le centre de gravité est le plus proche. Pour affecter les données aux classes, en plus des métriques présentées ci-dessus et qui peuvent tenir compte de la dispersion des paramètres ou de leur corrélation les uns par rapport aux autres pour l'ensemble des données (métriques globales), les classes étant déterminées, de nouvelles métriques peuvent être utilisées. En effet, les éléments qui caractérisent une classe fournissent une description de la classe à travers la dispersion ou la corrélation de ses éléments suivant les différents paramètres. L'utilisation des métriques que nous venons de voir prend en compte tous les éléments, toutes classes confondues. La matrice $\mathbf{M}$ est calculée sur la totalité des données. Nous allons voir l'intérêt d'utiliser une matrice spécifique à chaque classe.

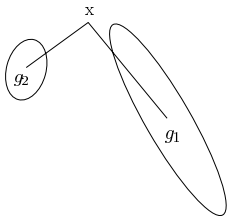


FIG. 2.3 – *Illustration de l'importance d'utiliser, pour l'affectation d'un élément, une métrique qui tiennent compte des dispersions des classes. Classiquement,* $\mathbf{x}$ *sera affecté à la classe 2. Or, au regard de la dispersion des classes,* $\mathbf{x}$ *s'apparente plus à certains éléments de la classe 1.*

Matrices catégorielles Dans un cas comme celui illustré sur la figure 2.3, le calcul de la distance en utilisant une matrice globale donnera $d^2(\mathbf{x},\mathbf{g}_1) > d^2(\mathbf{x},\mathbf{g}_2)$ et $\mathbf{x}$ sera affecté à la classe 2. Or, au regard de la dispersion des classes, $\mathbf{x}$ s'apparente plus à certains éléments de la classe 1. Puisque la dispersion de chaque classe influence l'affectation d'un élément, la solution serait de prendre des matrices locales $\mathbf{M}_i$ qui tiennent compte de la particularité des éléments de chaque classe. On aurait donc :

$$d^2(\mathbf{x},\mathbf{g}_i) = (\mathbf{x} - \mathbf{g}_i)^T \mathbf{M}_i (\mathbf{x} - \mathbf{g}_i)$$

où $\mathbf{M}_i$ est calculée d'après la valeur des variables des éléments de la classe i.

Les matrices catégorielles sont :

- Métrique tenant compte de la dispersion des éléments de la classe.
  Le calcul de la distance de la forme à classer au prototype tient compte de la dispersion des éléments de la classe. La matrice M est spécifique à chaque classe : $\mathbf{M}_i = (D_{1/s^2})_i$
- Métrique tenant compte des covariances des éléments de la classe.
  Le calcul de la distance de la forme à classer au prototype tient compte des covariances des éléments de la classe. La matrice M est spécifique à chaque classe : $\mathbf{M}_i = \mathbf{\Sigma}_i^{-1}$. De plus, les matrices $\mathbf{\Sigma}_i^{-1}$ doivent être normalisées pour que les distances soient réellement comparables. Normaliser une matrice $\mathbf{M}$ de dimension p revient à lui appliquer l'opération suivante :

  $$\mathbf{M}' = \frac{\mathbf{M}}{det(\mathbf{M})^{1/p}}$$

  où $\mathbf{M}'$ est la matrice normalisée de $\mathbf{M}$.
- Métrique tenant compte des dispersions et des covariances des éléments de la classe.
  Conséquence directe des observations précédentes, l'utilisation d'une matrice M spécifique à chaque classe $\mathbf{M}_i = \mathbf{R}_i^{-1}$ cumule les propriétes relatives à la dispersion et à la corrélation des éléments d'une classe. Ici aussi, les matrices $\mathbf{\Sigma}_i^{-1}$ doivent être normalisées.

Finalement, les méthodes de partitionnement permettent une représentation simplifiée et ordonnée des données, qui apporte une connaissance sur le domaine considéré. Elles

facilitent la prise de décision et donc la reconnaissance des formes. Les méthodes statistiques de partitionnement présentées sont parmi les plus représentatives. Nous avons vu que la notion de similarité est déterminante pour le résultat d'une classification et que certaines métriques peuvent améliorer l'homogénéité des éléments d'une même classe.

La section suivante présente les capacités des méthodes d'analyse multivariée à structurer des données pour augmenter la connaissance du problème et l'efficacité du traitement.

2.1.3 Capacités des méthodes d'analyse multivariée

Les méthodes d'analyse factorielle qui ont été présentées sont descriptives (on dit aussi exploratoires), par opposition aux méthodes d'analyse factorielle confirmatoires qui permettent de confronter l'adéquation de modèles théoriques aux données. Par conséquent, les méthodes abordées visent à décrire les données par des processus de transformation qui fournissent une explication du phénomène à travers l'étude des relations existant entre les données. Les diverses capacités des méthodes d'analyse multivariée peuvent être regroupées en trois catégories en fonction du but à atteindre :

Extraction de caractéristiques

L'application d'une analyse factorielle aboutit à une transformation de l'espace des données en l'espace des facteurs (ou caractéristiques). L'étude des caractéristiques va permettre de trouver un ensemble de variables internes susceptibles d'expliquer autrement le phénomène. Cet aspect est particulièrement intéressant pour comprendre un phénomène observé mais non maîtrisé. Généralement, le phénomène étudié est surdimensionné suivant un principe de précaution. Un traitement s'avère alors nécessaire pour obtenir l'ensemble des variables responsables du phénomène. Les transformations linéaires qui lient les facteurs aux variables observables donnent la possibilité de connaître les variables observables utiles et par conséquent de limiter les relevés à ces variables. Enfin, la sélection des caractéristiques les plus pertinentes déterminera les variables internes à utiliser pour une réduction efficace de la dimension.

Réduction de la dimension

Conformément aux souhaits de son auteur, l'analyse en composantes principales constitue une technique de représentation des données dans un espace à 2 ou 3 dimensions. De manière plus générale, elle permet une réduction de dimensions. Il suffit de faire abstraction des dimensions de faible variance. On ne retiendra que les dimensions dont les variances sont les plus importantes pour conserver un maximum d'information. De ce point de vue, l'analyse en composantes principales permet la compression de données. Il est important de réduire la dimension de l'espace des données pour plusieurs raisons. D'abord, pour des raisons de performances. Par exemple, il est difficile de modéliser une fonction gaussienne à une dimension par des fonctions noyaux à deux dimensions. Ensuite, pour des raisons de compréhension. Il est plus facile d'expliquer quelque chose de simple plutôt que quelque chose de complexe. La réduction de la dimension des données se

répercute directement sur le nombre de paramètres du modèle. Plus le nombre d'attributs est petit, moins il y a de combinaisons possibles, donc de paramètres libres. Le modèle sera plus facile à mettre au point et la solution plus rapidement trouvée. Par extension, on peut vouloir estimer la taille de l'échantillon suffisant à la modélisation.

Réduction de l'échantillon

La réduction du nombre d'éléments de l'échantillon permet de simplifier le problème en temps et en explication. La sélection des données peut même être une nécessité si leur nombre est trop important. Les données résultant de la réduction de la taille de l'échantillon doivent être choisies avec attention car dans certains cas le processus de sélection peut débruiter les données. Généralement, les données n'occupent qu'une petite partie d'espace. Elles peuvent être résumées par un petit nombre de représentants. Ces représentants ne sont pas nécessairement issus d'une sélection. Il est souvent préférable de créer des données prototypiques par quantification vectorielle. De nombreuses méthodes de quantification vectorielle existent. Elles réalisent toutes, par une compression de l'information, une représentation en quantité ou en qualité de la répartition des données dans son espace. En nombre restreint, les représentants peuvent être utilisés comme prototypes de classes. Ils permettent un partitionnement de l'espace des données. Les données seront regroupées par similarité en fonction de leur distance au prototype le plus proche.

Cette section avait pour but de se familiariser avec les fondements des méthodes d'analyse de données afin d'examiner par la suite les méthodes neuronales sous cet aspect. En effet, le travail de structuration réalisé par les méthodes d'analyse factorielle et de partitionnement sur les données permet d'extraire de la connaissance sur le problème. De plus, cela nous permettra, après la présentation des fondements des réseaux de neurones, de relever, dans les sections suivantes, les similitudes existant entre certaines méthodes des deux domaines, et donc de fournir des interprétations supplémentaires de la structuration des réseaux neuromimétiques, voire d'envisager des développements neuronaux qui permettent de réaliser des techniques d'analyse de données plus élaborées. En effet, nous verrons que les réseaux neuromimétiques sont capables, entre autres, d'effectuer les différents traitements évoqués dans cette section. Ils sont de plus capables d'étendre ces traitements à des structures non-linéaires.

2.2 Les fondements des réseaux neuromimétiques

Les réseaux neuromimétiques, aussi appelés réseaux de neurones artificiels, ou encore réseaux connexionnistes, sont un sujet de recherche couvrant de nombreuses disciplines parmi lesquelles l'intelligence artificielle, les statistiques, la neurobiologie, la psychologie et les sciences cognitives. Les réseaux de neurones artificiels sont souvent désignés par simplification sous le nom de réseaux de neurones voire de réseaux. Ces différents termes pourront être utilisés dans la suite du manuscrit de façon équivalente. Les réseaux de neurones réalisent un traitement d'information distribué. Ils sont constitués d'unités de calcul primitives (les neurones) qui fonctionnent en parallèle et sont reliées entre elles par des

connexions (les synapses). Le principe général de fonctionnement d'un réseau connexionniste est la transmission de l'activité d'un groupe de neurones à un autre groupe de neurones via les synapses, de manière analogue au fonctionnement du cerveau. Si les capacités du cerveau humain constituent une source d'inspiration très motivante, tous les réseaux connexionnistes n'essaient pas d'imiter le modèle biologique, mais peuvent être perçus simplement comme une classe d'algorithmes parallèles performants.
Dans ce type de modèle, la connaissance est généralement répartie à travers le réseau. Elle est stockée dans la topologie des neurones et le poids des connexions. Les réseaux de neurones s'organisent par des méthodes d'apprentissage automatique, ce qui permet une utilisation simple et rapide pour développer une application. Il n'est pas besoin de connaître les règles logiques qui modélisent le processus, ce qui constitue un intérêt pratique important. De plus, ces règles peuvent être extraites du modèle [Andrews *et al.*, 1995; Towell and Shavlik, 1993]. Un autre intérêt des modèles connexionnistes est leur grande tolérance au bruit et aux détériorations. Pour plus d'informations sur l'historique des réseaux neuromimétiques, les ouvrages suivants peuvent être consultés [Cowan, 1989] [Minsky and Papert, 1969].

2.2.1 Les principes d'une représentation distribuée

Les neurones

Le fonctionnement d'un neurone est simple. Il reçoit un signal de la part de chaque neurone i qui lui est connecté en entrée sous la forme d'une valeur $a(x_i,w_i)$, fonction de la valeur de sortie x_i du neurone i et du poids w_i de la connexion qui le relie au neurone i. A partir de cet ensemble de valeurs, le neurone calcule sa propre valeur de sortie y, ou activation, via une fonction d'activation $g(.)$. Finalement la valeur de sortie sera transmise instantanément à tous les neurones auxquels il est relié en sortie par l'intermédiaire de connexions (figure 2.4).

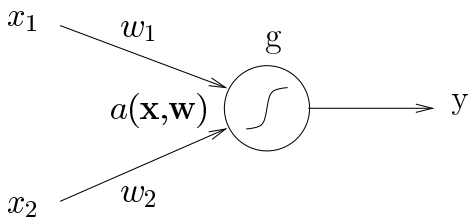


FIG. 2.4 – *Formalisation mathématique d'un neurone. Un neurone combine ses valeurs d'entrée (x_1 et x_2) avec le poids de leur connexion respective (w_1 et w_2), généralement sous la forme d'une somme pondérée ($a(\mathbf{x},\mathbf{w}) = x_1w_1 + x_2w_2$). La fonction d'activation $g(.)$ du neurone utilise cette combinaison pour calculer son activation, c'est-à-dire sa valeur de sortie y, qui pourra éventuellement servir de valeur d'entrée à d'autres neurones.*

Au sein d'un réseau de neurones, on distingue trois types d'unités : les neurones d'entrée qui reçoivent des valeurs extérieures au réseau, les neurones de sortie qui transmettent leur valeur en dehors du réseau et les neurones cachés qui n'ont pas de lien avec l'extérieur.

Les connexions

Une connexion, aussi appelée synapse en référence au vocabulaire neurobiologique, est unidirectionnelle. Elle relie un neurone pré-synaptique à un neurone post-synaptique. Elle est caractérisée par une valeur : son poids, noté w. Une valeur de poids positive contribue à une excitation du neurone tandis qu'une valeur négative contribue à une inhibition.

Les fonctions d'activation

La fonction d'activation $g(.)$ prend comme entrée l'état courant du neurone $y(t)$, c'est-à-dire la valeur de son activation à un instant t, et une combinaison des valeurs d'entrée a(t) pour calculer l'état du neurone à l'instant suivant $y(t+1)$.

$$y(t+1) = g(y(t),a(t))$$

Généralement, la valeur d'entrée d'un neurone est la somme de ces entrées pondérées par le poids de leur connexion plus un biais noté θ.

$$a_j(t) = \sum_i w_{ji}(t)x_i(t) + \theta_j(t)$$

Les neurones qui utilisent une telle combinaison sont appelés unités *sigma*. Un autre type de combinaison, multiplicative, a été introduit par Feldman et Ballard [Feldman and Ballard, 1982] et est connu sous le nom de règle de propagation des unités sigma-pi.

$$a_j(t) = \sum_i w_{ji}(t) \prod_m x_{i_m}(t) + \theta_j(t)$$

Le choix de la fonction d'activation dépend de la souplesse qu'on veut lui donner et des bornes de l'espace de sortie. Généralement on utilise des fonctions à seuil, semi-linéaires, linéaires ou sigmoïdales (voir figure 2.5).

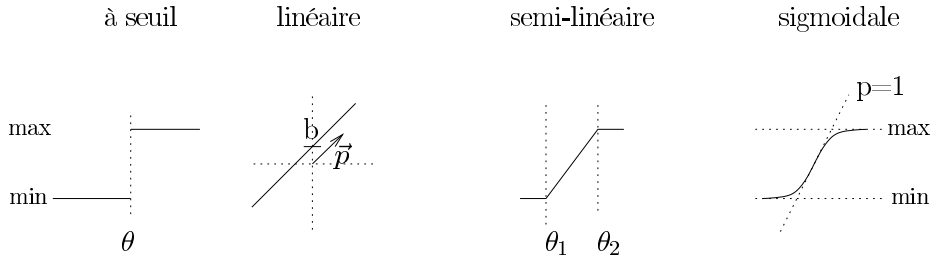
La valeur de l'activation peut ne pas être déterministe. Dans ce cas, la valeur d'entrée détermine la probabilité p qu'un neurone ait une forte activation. La fonction d'activation stochastique est du type :

$$g(x) = \frac{1}{1+\exp(-a(x)/T)} = p(y \leftarrow 1)$$

où la température T règle la pente de la fonction probabiliste.

2.2.2 La topologie

Nous venons de voir le principe de fonctionnement d'un neurone, nous allons maintenant nous intéresser au fonctionnement d'ensembles de neurones. Les neurones sont fréquemment rassemblés en groupes, lesquels constituent des couches. Le nombre de couches, le nombre de neurones par couche et la connectivité qui les relie définissent l'architecture d'un réseau connexionniste. Deux grandes catégories de réseaux se distinguent en fonction de leur connectivité :



$$g(x) = \begin{cases} min & \text{si } x < \theta \\ max & \text{si } x > \theta \end{cases} \quad g(x) = px + b \quad g(x) = \begin{cases} min & \text{si } x \leq \theta_1 \\ px + b & \text{si } \theta_1 < x < \theta_2 \\ max & \text{si } \theta_2 \leq x \end{cases} \quad g(x) = \frac{max - min}{1 + \exp(-px)} + min$$

où min et max représentent les valeurs minimale et maximale de sortie,
p la pente,
et θ, θ_1 et θ_2 différents seuils.

FIG. 2.5 – *Différentes fonctions d'activation d'un neurone. Le résultat de la fonction d'activation correspond à la valeur de sortie du neurone. Du type de cette fonction dépend la nature du traitement effectué par le modèle. Les fonctions à seuil introduisent dans le modèle une discrimination, les fonctions non linéaires une non linéarité, etc. Généralement les neurones d'une même couche ont la même fonction d'activation.*

Les réseaux récurrents

Les réseaux récurrents autorisent et contiennent des cycles dans le graphe d'état orienté que forme un réseau de neurones. Certaines connexions relient donc un neurone à un neurone de la même couche ou d'une couche antérieure. On parle de connexion récurrente ou feed-back. Cette catégorie comprend par exemple les réseaux de Hopfield [Hopfield and Tank, 1986], les cartes auto-organisatrices de Kohonen [Kohonen, 1989] etc. Nous détaillerons ces modèles dans la section 2.4.

Les réseaux feed-forward

Les réseaux dits feed-forward propagent le flux d'information de manière unidirectionnelle depuis la couche d'entrée jusqu'à la couche de sortie. Il n'y a pas de cycle, on les dit acycliques. Ce type de fonctionnement est le plus couramment observé dans la littérature. Cette catégorie comprend par exemple le perceptron, le perceptron multicouches, et le réseau adaline.

2.2.3 L'apprentissage

Un réseau de neurones doit traiter l'information présentée en entrée de telle manière que les sorties produites répondent aux attentes. Lorsqu'on ne connaît pas *a priori* les caractéristiques à donner au réseau (ce qui est souvent le cas), la gestion du processus se fait à l'aide d'un apprentissage itératif, par modification des poids et plus rarement par modification de l'architecture. Les transformations apportées dépendent de règles d'apprentissage. Seuls les deux cas principaux d'apprentissage sont présentés :

Les paradigmes d'apprentissage

1. Les méthodes supervisées
   Si l'on connaît explicitement les valeurs à fournir en sortie, la méthode d'apprentissage est dite supervisée. Une fonction de coût mesure la différence entre les sorties désirées et les sorties produites, et une règle d'apprentissage modifie le réseau pour réduire cette différence.

2. Les méthodes non supervisées
   Ici, le réseau doit découvrir des régularités dans les formes qui lui sont présentées. Le nombre de neurones de sortie spécifie le nombre de catégories que l'on veut voir émerger. Le système doit développer sa propre représentation des formes d'entrée en retenant les traits statistiquement redondants. Aucune connaissance ne lui est donnée *a priori*.

La modification des poids

Quel que soit le type d'apprentissage, il a pour effet de modifier la valeur des poids. La règle d'apprentissage détermine l'importance de cette modification à l'égard de la contribution du poids sur la qualité du résultat final.

1. La première règle d'apprentissage a été définie par Hebb [Hebb, 1949]. Elle porte le nom d'apprentissage hebbien et s'énonce de la manière suivante : si deux neurones y_i et y_j interconnectés s'activent au même moment, alors la connexion qui les unit voit son poids w_{ij} renforcé.

$$\Delta w_{ij} = w_{ij}(t+1) - w_{ij}(t) = \eta y_i y_j$$

   avec η une constante positive représentant le coefficient d'apprentissage.
   L'apprentissage hebbien est non supervisé.
2. Une autre règle de modification des poids, connue sous le nom de règle de Windrow-Hoff ou règle du delta, est à l'origine de nombreuses variantes. Elle s'applique lorsqu'on connaît la valeur désirée d_j du neurone post-synaptique j. L'apprentissage est donc supervisé. La valeur de la modification du poids w_{ij} est pondérée par le coefficient d'apprentissage η et par la valeur y_i du neurone pré-synaptique i.

$$\Delta w_{ij} = \eta y_i (d_j - y_j)$$

La fréquence des modifications peut être nuancée :

1. Apprentissage on-line : la correction des poids a lieu après chaque présentation.
2. Apprentissage off-line : la correction des poids a lieu après le passage de tous les exemples i.e. après un cycle. Les modifications à apporter aux poids sont cumulées et prennent effet en fin de cycle.

Le corpus d'apprentissage et le corpus de test

L'apprentissage est stoppé en fonction d'un critère d'arrêt. Pour les méthodes supervisées, il est généralement donné par la performance obtenue sur un corpus de test après apprentissage des données d'un corpus d'apprentissage :

1. Le corpus d'apprentissage est constitué de la plus grande partie des données. Le modèle modifie ses paramètres, c'est-à-dire ses poids, en fonction des erreurs qu'il observe sur les exemples du corpus d'apprentissage. La fonction est apprise sur les données qu'il contient.
2. Le corpus de test estime la performance du modèle sur des données nouvelles, donc dans des conditions réelles d'utilisation. Il indique quand arrêter l'apprentissage, i.e. le moment où le modèle a fini d'apprendre les caractéristiques générales de la tâche et commence à apprendre par cœur les exemples du corpus d'apprentissage.

Ainsi se termine la présentation des fondements des réseaux de neurones. Elle avait pour but d'introduire le vocabulaire propre aux réseaux de neurones et de présenter, de manière concise, les différents secteurs du domaine, tous concernés par les explications, réflexions et améliorations présentées dans la suite de ce manuscrit. Nous allons maintenant interpréter, dans le détail, le fonctionnement des réseaux de neurones sur des tâches d'extraction de connaissances qui nécessitent toutes une adaptation du réseau au problème localisé, problème qui peut être résolu en structurant l'information.

2.3 Les réseaux feed-forward

Dans cette section, nous expliquons les capacités de représentation des réseaux feed-forward à une couche, c'est-à-dire sans couche cachée, puis celles des réseaux feed-forward à plusieurs couches.

2.3.1 Les réseaux à une couche

Les réseaux connexionnistes à une couche calculent l'activation des neurones de sortie directement à partir des entrées du réseau via le poids des connexions qui les relient (figure 2.6). Les paramètres du réseau se résument aux poids et à la fonction d'activation. Pour ce type de réseaux, le champ de manœuvre est donc faible. Nous allons profiter de l'extrême simplicité du réseau pour montrer que, malgré la faible liberté qui nous est donnée, le choix judicieux de la fonction d'activation va permettre d'effectuer une tâche de classification, voire de régression. Nous allons donc expliquer les capacités particulières des fonctions à seuil, des fonctions linéaires et des fonctions sigmoïdales. Ces capacités resteront valables pour des réseaux plus complexes. Donc l'étude des réseaux à une couche est intéressante à deux titres : pour eux-mêmes, car ces réseaux réalisent déjà des tâches intéressantes de classification et de régression dont nous allons expliquer les mécanismes ; et pour mieux comprendre les opérations effectuées par des réseaux plus complexes mais qui, localement, adoptent un comportement similaire, explicable à partir du type de la fonction d'activation.

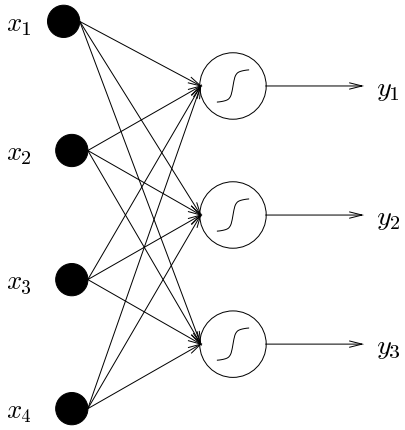


FIG. 2.6 – *Illustration d'un réseau feed-forward à une couche. Les différentes sorties y_j du réseau sont le résultat d'une opération directe sur les entrées x_i. Ce réseau comporte 4 entrées, 3 sorties et les neurones de la couche de sortie possèdent une fonction d'activation sigmoïdale. Il est entièrement connecté.*

Les réseaux à fonctions à seuil :

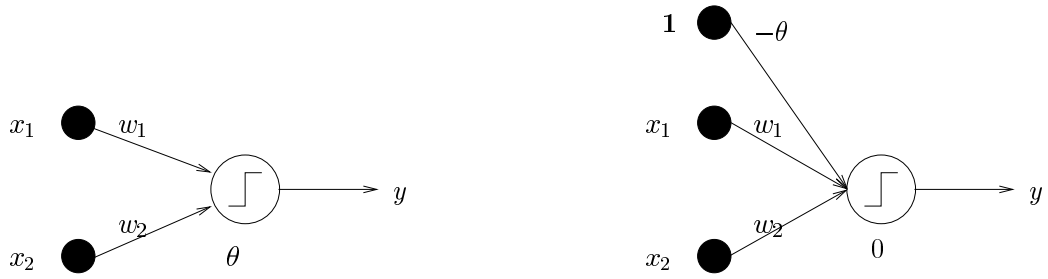
Les fonctions d'activation à seuil sont des fonctions bi-stables qui changent d'état au-delà d'un seuil (équation 2.3). Elles passent généralement de 0 à 1, ou de -1 à 1 (nous détaillerons leur utilisation un peu plus loin).

$$g(\mathbf{x}) = \begin{cases} min & \text{si } \sum_i w_i(t)x_i(t) < \theta \\ max & \text{si } \sum_i w_i(t)x_i(t) \geq \theta \end{cases} \tag{2.3}$$

Toutes les fonctions à seuil peuvent se ramener par translation à une autre fonction dont le seuil est zéro (équivalences 2.4). Dans ce cas, cela revient à ajouter un neurone d'entrée toujours actif à 1 avec un poids de connexion égal à $-\theta$. Cette modification a pour intérêt de permettre l'apprentissage de la valeur du seuil (figure 2.7).

$$\begin{array}{lcll} \sum_i^N w_i(t)x_i(t) & < & \theta & \\ \sum_i^N w_i(t)x_i(t) - \theta & < & 0 & \\ \sum_i^{N+1} w_i(t)x_i(t) & < & 0 & \text{avec } w_{N+1}(t) = -\theta \text{ et } x_{N+1}(t) = 1 \end{array} \tag{2.4}$$

Pour bien comprendre l'opération réalisée par une fonction d'activation à seuil, plaçons nous dans le cas d'un réseau à deux entrées et une seule sortie (figure 2.7). En tenant compte de la remarque ci-dessus, l'équation 2.3 dans un espace à deux dimensions définit un hyperplan séparateur H d'équation : $w_1x_1 + w_2x_2 - \theta = 0$ (figure 2.8). La fonction d'activation à seuil réalise une classification des entrées dans deux classes. Donc un réseau à une couche dont le neurone de sortie a une fonction d'activation à seuil représente une fonction linéaire discriminante. Dans le cas de plusieurs neurones de sortie, le réseau représente une tâche de partitionnement. Chaque neurone de sortie identifie une région de l'espace d'entrée.



(a) Le seuil de discrimination d'un réseau feed-forward sans biais est fixe et égal à θ. Généralement θ vaut zéro.

(b) Le seuil de discrimination d'un réseau feed-forward avec biais est variable et opposé au poids θ de la connexion issue du neurone biais.

FIG. 2.7 – *La valeur du seuil θ de discrimination par un neurone comportant une fonction d'activation du même nom peut être apprise par l'utilisation d'un neurone supplémentaire en entrée, appelé le biais, dont la valeur est toujours égale à 1. C'est l'ajustement du poids de cette connexion supplémentaire qui déterminera le seuil de discrimination.*

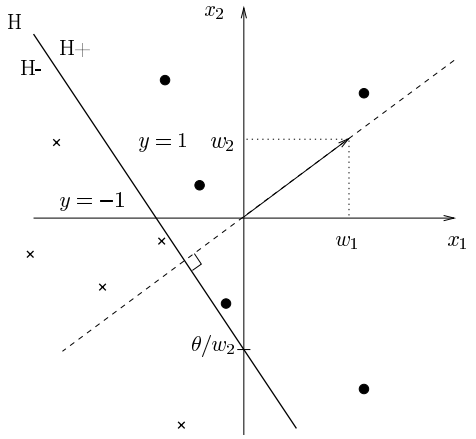


FIG. 2.8 – *Illustration de l'utilisation d'un neurone à fonction à seuil pour effectuer une séparation de l'espace d'entrée par un hyperplan H. Son orientation dépend de la valeur des poids car elle est perpendiculaire au vecteur $(w_1,w_2)^T$ et son décalage par rapport à l'origine dépend du seuil θ. La valeur de sortie du neurone, positive ou négative, détermine le sous-espace auquel appartient la forme d'entrée $(x_1,x_2)^T$. Ce type de neurone peut donc apprendre, par l'apprentisage de ses poids, à discriminer au mieux des éléments de deux classes,* x *et* •.

La compréhension des mécanismes sous-jacents à l'utilisation des fonctions à seuil dans un réseau à une couche (voir section 2.3.1) nous permet, lorsque l'on connaît le diagramme de Voronoï d'un partitionnement [Bose and Garga, 1993], de créer le réseau de neurone qui réalisera exactement ce partitionnement. On peut donc passer d'une solution théorique à son implantation de manière simple et efficace. On rappelle qu'un diagramme de Voronoï est constitué d'un ensemble de sites qui déterminent une partition de l'espace (voir figure 2.9). Cette partition se compose de régions (ensemble des points plus proches d'un site que de tous les autres), de côtés (ensemble des points équidistants de 2 sites) et de sommets

(ensemble des points équidistants d'au moins 3 sites). La méthode de Bose et Garga nous permet de calculer, à partir des sites, l'équation des hyperplans délimitant les régions. Il suffit ensuite de déduire la valeur des poids à partir de l'équation des hyperplans (ce qui est immédiat).

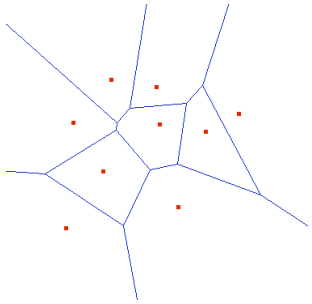


FIG. 2.9 – *Le diagramme de Voronoi représente les frontières délimitant les plus proches voisins d'un ensemble de sites. Chaque région, ou cellule, contient l'ensemble des points qui se trouvent plus proches du site contenu dans la cellule que de tout autre site. Le diagramme de Voronoï a de nombreuses propriétés et applications (voir [Aurenhammer, 1991]). Il permet, entre autres, de visualiser la limite de classes dont on connaît les représentants.*

Nous avons vu que le seuil pouvait être ramené au poids d'une connexion et par conséquent à un paramètre variable. Pour obtenir une classification adéquate il faut bien catégoriser les entrées en jouant sur la valeur du seuil en appliquant la règle de modification des poids tel qu'on le ferait pour un poids quelconque. L'apprentissage d'une classification est supervisé. L'ajustement des droites séparatrices s'obtient en modifiant la valeur des poids. Pour une tâche de classification, si le réseau donne la réponse $\mathbf{y}$ lorsque l'entrée $\mathbf{x}$ appartient à la classe identifiée par le vecteur $\mathbf{d}$, la règle d'apprentissage est la suivante :

si $y_j \neq d_j$ **alors** $\Delta w_{ij} = \eta x_i d_j$ où $d_j \in \{-1, +1\}$ et η est le coefficient d'apprentissage.

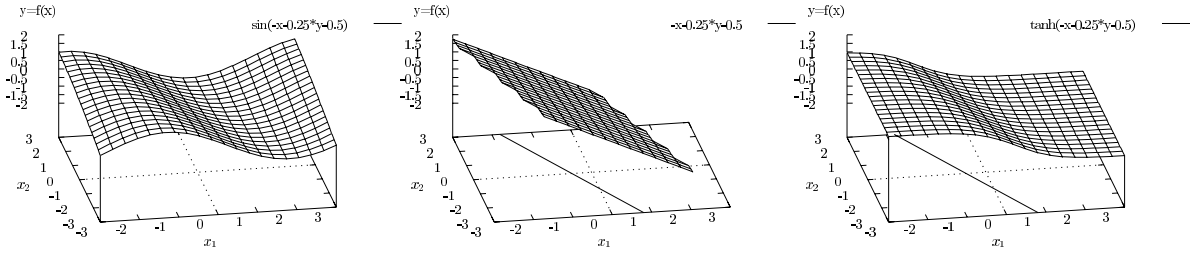
La règle est très proche de la règle de Hebb, mais ici, si la classification est bonne les poids ne varient pas. L'écriture de cette règle sous une forme qui ne nécessite pas de test aboutit à la règle de Widrow-Hoff présentée en section 2.

$$\Delta w_{ij} = \eta x_i (d_j - y_j)$$

Théorème 3
S'il existe un ensemble de poids qui permettent $\mathbf{y} = d(\mathbf{x})$*, la règle d'apprentissage convergera vers une solution identique ou équivalente en un nombre fini de passages et ce quelle que soit l'initialisation des poids.*

La preuve de ce théorème peut être trouvée dans de nombreux ouvrages dont [Rosenblatt, 1961; Hertz *et al.*, 1991].

Rosenblatt est à l'origine de la preuve du théorème et il appliqua la règle d'apprentissage ci-dessus à un réseau de neurones à une couche appelé perceptron ou perceptron



(a) La fonction à modéliser est $-\sin(x + 1/4y + 1/2)$.

(b) Une approximation linéaire de cette fonction.

(c) Une approximation sigmoïdale de cette fonction.

FIG. 2.10 – *Illustration de l'effet de la fonction d'apprentissage sur l'approximation de fonction.*

simple [Rosenblatt, 1961]. Ce modèle apprenait une transformation de $f : \{-1,1\}^N \rightarrow \{-1,1\}$ à partir d'exemples de couples ($\mathbf{x}$,y). tels que $f(\mathbf{x}) = y$. Par extension, les réseaux de neurones à plusieurs couches ont pris le nom de perceptrons multicouches.

Donc, un neurone muni d'une fonction d'activation à seuil réalise une séparation linéaire des entrées. Un réseau de neurone à une couche dont les neurones de sortie ont des fonctions d'activation à seuil réalisera correctement une tâche de classification à condition que les classes soient linéairement séparables.

Les réseaux à fonctions linéaires

Un réseau de neurones à une couche dont les fonctions d'activation sont linéaires établit des relations linéaires entre les sorties et les entrées, du type :

$$y_j = \sum_i w_{ij} x_i + \theta_j$$

où θ_j est un biais, c'est-à-dire le poids d'une connexion reliant un neurone supplémentaire d'activation égale à 1 et le neurone de sortie d'indice j.

Un tel modèle réalise une tâche d'approximation de fonctions. En effet, pour chaque neurone de sortie, les poids définissent un hyperplan. Nous pouvons entraîner le réseau pour que l'hyperplan résume au mieux les données, c'est-à-dire obtenir une régression linéaire (figure 2.10).

Pour un vecteur d'entrées $\mathbf{x}^n$, le modèle produit un vecteur de sortie $\mathbf{y}^n$ qui diffère de la sortie désirée $\mathbf{d}^n$ de $(\mathbf{d}^n - \mathbf{y}^n)$. La fonction de coût E qui mesure les performances du modèle est définie comme la somme des différences au carré :

$$E = \sum_n^N E^n = 1/2 \sum_n^N (\mathbf{d}^n - \mathbf{y}^n)^2$$

où E^n est l'erreur produite sur le $n^{\text{ème}}$ exemple.
Widrow et Hoff ont défini une règle d'apprentissage, connue sous le nom de la règle des

moindres carrés, qui minimise l'erreur de la fonction de coût par une méthode de descente du gradient le long de la surface d'erreur. Cette méthode consiste à modifier les poids proportionnellement et inversement à la valeur de la dérivée de l'erreur :

$$\Delta w_{ij} = -\eta \frac{\partial E^n}{\partial w_{ij}}$$

La dérivée peut s'écrire :

$$\frac{\partial E^n}{\partial w_{ij}} = \frac{\partial E^n}{\partial y_j^n} \frac{\partial y_j^n}{\partial w_{ij}}$$

avec

$$\frac{\partial E^n}{\partial y_j^n} = -(d_j^n - y_j^n)$$

et

$$\frac{\partial y_j^n}{\partial w_{ij}} = x_i^n$$

d'où

$$\Delta w_{ij} = \eta (d_j^n - y_j^n) x_i^n$$

Widrow et Hoff ont appliqué leur algorithme à un réseau linéaire à une couche prenant en entrée et en sortie des éléments de $\{-1,1\}$. Ce réseau, du nom de ADALINE, signifiait à l'origine ADApatative LInear NEuron pour devenir ADAptative LINear Element lorsque les réseaux de neurones sont devenus moins populaires, à la fin des années 60 [Widrow and Hoff, 1960] [Widrow and Lehr, 1990] . En effet, les réseaux à une couche sont limités dans leur capacité à représenter la frontière entre deux classes ou la fontion qui associe les entrées aux valeurs désirées. Dans une tâche de classification, sa représentation des classes est restreinte à des frontières de type hyperplan. Si les données ne sont pas linéairement séparables, la solution ne sera qu'approchée. Dans une tâche d'approximation de fonctions, la forme de l'approximation est induite par la fonction d'activation. Elle est donc généralement linéaire ou sigmoïdale.

Ces limitations sont à l'origine de la conception de réseaux à plusieurs couches dont les capacités de représentation sont augmentées. Néanmoins, les réseaux feed-forward à une couche ont l'avantage d'avoir des algorithmes d'apprentissage qui convergent vers la solution optimale, ce qui n'est pas le cas des réseaux multicouches non-linéaires (bien sûr, les réseaux multicouches peuvent ne comporter que des fonctions linéaires mais ils n'obtiendraient pas de meilleurs résultats qu'un réseau à une couche puisqu'on pourrait toujours se ramener dans ce cas à une combinaison linéaire des entrées).

Expérimentation : Pour évaluer les performances en régression d'un perceptron linéaire sur notre application, nous avons utilisé le corpus complet normalisé sur un perceptron comportant 32 entrées et 1 sortie. L'apprentissage a duré 100 cycles avec un coefficient d'apprentissage fixe de 10^{-4}. La qualité de la prédiction du champ radioélectrique est fournie par le tableau 2.1.

L'unique sortie est égale à $\sum_i w_i x_i + \theta$. La lecture directe des poids donne la valeur des coefficients de régression.

Modèle	Phase	$\bar{e}$ (dB)	σ	Q_4 (%)	Q_6 (%)	Q_{11} (%)
Perceptron simple	*Apprentissage*	4.83	3.98	50.9	69.1	92.3
	Test	4.91	4.02	50.5	68.5	91.7

TAB. 2.1 – *Performances d'un perceptron simple sur le problème de prédiction du champ radioélectrique.*

La fonction reconstituée est :
$AMES = 0.16\ X_1 + 0.86\ X_2 + 0.05\ X_3 - 0.02\ X_4 - 0.01\ X_5 - 0.16\ X_6 - 0.02\ X_7 - 0.12\ X_8 + 0.18\ X_9 - 1.54\ X_{10} + -0.02\ X_{11} + 0.02\ X_{12} + 0.13\ X_{13} + 0.12\ X_{14} - 0.04\ X_{15} + 0.25\ X_{16} - 0.02\ X_{17} - 0.03\ X_{18} - 0.04\ X_{19} - 0.04\ X_{20} - 0.29\ X_{21} + 0.23\ X_{22} - 0.04\ X_{23} + 0.02\ X_{24} - 0.01\ X_{25} + 0.06\ X_{26} + 5.10^{-3}\ X_{27} - 0.13\ X_{28} - 0.02\ X_{29} + 0.05\ X_{30} + 0.07\ X_{31} + 0.05\ X_{32} - 0.13$

Comparativement à la régression statistique (voir page 23), la méthode neuronale ne nécessite pas de pré-traitement particulier pour ne conserver que les variables indépendantes.

Les réseaux à fonctions sigmoïdales

Les fonctions sigmoïdales sont des fonctions non linéaires bornées de la forme :

$$g(x) = \frac{max - min}{1 + \exp(-px)} + min$$

où min et max représentent les bornes des valeurs de sortie, et p représente la pente, autrement dit le passage plus ou moins rapide d'une extrême à l'autre. Dans le cas où $min = -1$, $max = 1$ et $p = 2$ on retrouve la fonction tangente hyperbolique.
Les fonctions d'activation sigmoïdales sont plus souples que les fonctions à seuil. Elles donneront une indication sur la confiance de l'appartenance d'un exemple à une classe, en ce sens que la réponse du neurone ne sera pas blanche ou noire mais grise à proximité de la frontière. Elles vont également permettre d'obtenir une régression non linéaire. Il sera donc plus facile d'ajuster la fonction de régression aux exemples. De plus, le fait qu'elles soient bornées va permettre d'éviter une explosion des valeurs de sortie, en particulier dans les réseaux multicouches où les calculs s'enchaînent. C'est pourquoi elles sont largement utilisées dans ces réseaux.

Nous avons vu que les unités de la couche de sortie d'un perceptron reçoivent une combinaison linéaire des entrées. Munir ces unités d'une fonction d'activation linéaire revient à définir un modèle de régression linéaire multiple multivarié. Dans le cas d'une fonction sigmoïdale, nous avons un modèle de régression logistique. Si la fonction d'activation est à seuil, nous avons un modèle linéaire de discrimination simple dans le cas du réseau connexionniste ADALINE, multivarié s'il y a plusieurs sorties. Mais il est plus avantageux d'utiliser dans ce dernier cas une fonction *s*oftmax, c'est-à-dire une fonction logistique multiple, qui permet d'estimer la probabilité conditionnelle de chaque classe.

Les réseaux à liens fonctionnels

Il est également possible d'obtenir une régression polynômiale de degré N si on utilise un réseau à lien fonctionnel avec N unités sur la couche cachée ayant chacune une fonction d'activation $g(.)$ faisant correspondre à $x \mapsto g(x) = x^n$ (figure 2.11). Un lien fonctionnel ne comporte pas de paramètre ajustable mais une fonction. La valeur du neurone présynaptique est l'unique paramètre de la fonction. Un lien fonctionnel est équivalent à une connexion de poids fixe égal à 1 suivi par un neurone dont la fonction d'activation est la fonction du lien fonctionnel. Les réseaux à liens fonctionnels ont une couche supplémentaire par rapport au perceptron mais un seul niveau de poids à estimer. Les poids reliant l'entrée aux neurones de la couche cachée sont constants. On peut donc leur appliquer les algorithmes d'apprentissage des réseaux à une couche.

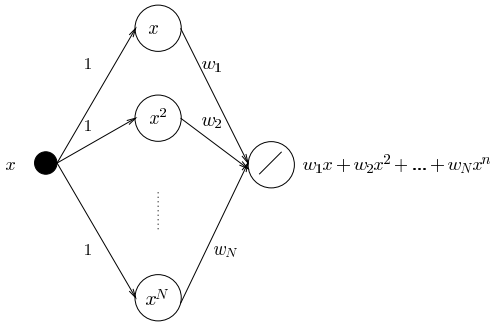


FIG. 2.11 – *Illustration d'un réseau à liens fonctionnels. Une couche cachée constituée de fonctions différentes, correspondant chacune à une élévation de la valeur d'entrée x à une puissance comprise entre 1 et N, permet d'obtenir un polynôme de degré N en sortie du réseau dont les poids w_i sont les coefficients du polynôme (la fonction de sortie est linéaire). Un tel réseau réalise une modélisation polynômiale.*

Ce type de réseau a été développé pour permettre de faire une régression polynômiale. On pourrait, par exemple, obtenir le développement de Taylor au voisinage de 0 d'une fonction. On pourrait également utiliser d'autres fonctions telles que les fonctions gaussiennes qui définissent une famille particulière des réseaux de neurones, les réseaux à fonctions à base radiale.

Finalement, la présentation des réseaux de neurones à une couche nous a permis de mieux connaître leur fonctionnement interne et de mettre à jour le type de représentation du monde dont ils sont capables. Leur structuration est intimement liée à la fonction d'activation qu'ils utilisent.

2.3.2 Les réseaux multicouches

Minsky et Paper [Minsky and Papert, 1969] ont montré que les réseaux à une couche n'étaient capables de résoudre que des problèmes linéairement séparables.

L'ajout successif de couches (figure 2.12) permet intuitivement d'associer des demi-espaces pour obtenir, avec un réseau à deux couches, des régions convexes et, avec un

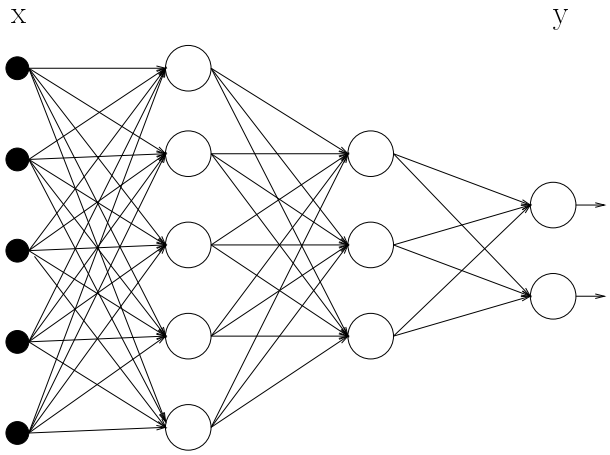
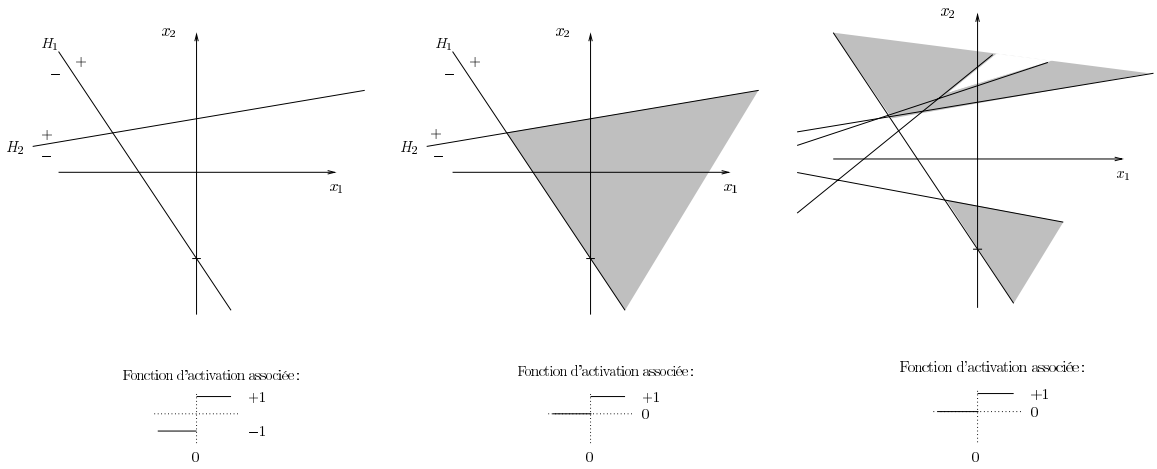


FIG. 2.12 – *Illustration d'un réseau multicouches. Ce réseau comporte 3 couches, une de sortie et deux cachées (i.e. sans lien directe avec les entrées ou les sorties). La couche d'entrée n'est pas comptée dans la taille d'un réseau. Elle peut éventuellement effectuer une transformation des valeurs d'entrée.*



(a) La division de l'espace d'entrée en demi-espaces est obtenue par un réseau à 1 couche.

(b) Les régions convexes, définies par des demi-espaces, sont obtenues par un réseau à 2 couches.

(c) Les régions non-convexes et disjointes, définies par des demi-espaces, sont obtenues par un réseau à 3 couches.

FIG. 2.13 – *Illustration de l'effet du nombre de couches sur la capacité de discrimination d'un réseau à fonction à seuil.*

réseau à trois couches, des régions non convexes et disjointes (figure 2.13). Les classes peuvent donc ne pas être linéairement séparables. Les fonctions modélisables peuvent être plus complexes, voire très complexes.

Dans les réseaux à une couche, tous les neurones peuvent corriger leur poids en comparant leur valeur de sortie à la valeur désirée. Le problème dans le cas de couches inter-

médiaires est que les neurones de ces couches n'ont pas de valeur désirée à comparer à leur sortie pour modifier leur poids. La solution a été, entre autres, fournie par Rumelhart, Hinton et Williams [Rumelhart *et al.*, 1986a] sous la forme d'une règle d'apprentissage qui généralise la règle de Widrow-Hoff aux fonctions d'activation dérivables d'une part, et permet son application à des couches intermédiaires d'autre part. La règle consiste à transmettre une valeur désirée aux couches intermédiaires en calculant la contribution d'un neurone puis de ses poids à l'erreur. Cette règle a pour nom la règle de rétro-propagation du gradient de l'erreur. Elle est une généralisation de la règle du Delta aux fonctions d'activation dérivables.

Algorithme de la rétro-propagation

Nous avons vu, dans la section 2.3.1, la définition de la règle de modification des poids de Widrow-Hoff pour des unités linéaires dans le cas d'un réseau à une couche. Si on réécrit cette formule en généralisant la fonction d'activation au cas d'une fonction dérivable, on a :

$$y_j^n = g(a_j^n)$$

avec

$$a_j^n = \sum_{i=1}^{d} w_{ij} y_i^n$$

et

$$\Delta w_{ij} = -\eta \frac{\partial E^n}{\partial a_j^n} \frac{\partial a_j^n}{\partial w_{ij}}$$

si on pose $\delta_{j^L}^n = -\dfrac{\partial E^n}{\partial a_{j^L}^n}$, alors

$$\Delta w_{ij^L} = \eta \delta_{j^L}^n y_{i^{L-1}}^n$$

On peut écrire :

$$\delta_{j^L}^n = -\frac{\partial E^n}{\partial y_{j^L}^n} \frac{\partial y_{j^L}^n}{\partial a_{j^L}^n}$$

on a :

$$\frac{\partial y_{j^L}^n}{\partial a_{j^L}^n} = g'(a_{j^L}^n)$$

Pour le calcul de $\dfrac{\partial E^n}{\partial y_{j^L}^n}$, deux cas se présentent. Si nous sommes sur la couche de sortie :

$$\frac{\partial E^n}{\partial y_j^n} = -(d_j^n - y_j^n)$$

et finalement :

$$\delta_{j^L}^n = (d_j^n - y_j^n) g'(a_{j^L}^n)$$

Sinon,

$$\begin{aligned}\frac{\partial E^n}{\partial y_j^n} &= \sum_k^{d^L} \frac{\partial E^n}{\partial a_{k^L}^n} \frac{\partial a_{k^L}^n}{\partial y_j^n} \\ &= \sum_k^{d^L} \frac{\partial E^n}{\partial a_{k^L}^n} \frac{\partial \sum_{j=1}^{d} w_{jk} y_j^n}{\partial y_j^n} \\ &= \sum_k^{d^L} \frac{\partial E^n}{\partial a_{k^L}^n} w_{jk} \\ &= \sum_k^{d^L} \delta_{k^L}^n w_{jk}\end{aligned}$$

et finalement :

$$\delta_{j^l}^n = \sum_k^{d^L} \delta_{k^L}^n w_{jk} g'(a_{j^l}^n)$$

Nous disposons donc d'un calcul récursif des δ, c'est-à-dire des erreurs de chaque unité, à partir desquelles nous pouvons obtenir la modification des poids à appliquer aux connexions qui leur sont destinées. Ceci constitue l'algorithme de rétro-propagation de l'erreur applicable aux réseaux multicouches constitués de fonctions d'activation dérivables.

Approximateurs universels de fonctions

Un réseau feed-forward multicouches (MultiLayer Perceptron en anglais (MLP)) à une couche cachée dont les fonctions d'activation de la couche cachée sont sigmoïdales et dont les fonctions d'activation de la couche de sortie sont linéaires est un modèle de régression non linéaire. S'ils disposent de suffisamment de données et d'unités sur la couche cachée, les perceptrons multicouches sont des approximateurs universels de fonctions, c'est-à-dire qu'ils peuvent modéliser avec une précision aussi grande que nécessaire n'importe quelle fonction [Cybenko, 1989; White, 1992; Hecht-Nielsen, 1989; Hornik *et al.*, 1989]. White l'explique en disant que l'apprentissage de la combinaison de sigmoïdes hiérarchiques a pour effet de créer une association de fonctions dont les « fréquences », les « phases » et les « amplitudes » s'organisent pour ajuster au mieux la fonction à modéliser.

Expérimentation : Le problème de prédiction de l'atténuation du champ radioélectrique est un problème complexe. Les résultats obtenus jusqu'à présent avec une régression linéaire statistique et un perceptron simple ne permettent pas d'atteindre les critères de qualité souhaités. Ce problème nécessite des modèles de prédiction plus performants tels qu'un perceptron multicouches paramétré pour la régression. Nous avons donc évalué un perceptron multicouches constitué d'une couche cachée avec 10 neurones dont la

fonction d'activation est la fonction tangente hyperbolique. Donc l'architecture du réseau comporte 32 unités sur la couche d'entrée, 10 unités non linéaires sur la couche cachée et une unité linaire sur la couche de sortie. L'apprentissage a duré 1000 cycles avec un coefficient d'apprentissage de 10^{-4}. Les résultats sont reportés dans le tableau 2.2 et sont à comparer avec les résultats obtenus par les régressions linéaires pages 23 et 53.

Modèle	Architecture	Phase	$\bar{e}$ (dB)	σ	Q_4 (%)	Q_6 (%)	Q_{11} (%)
MLP	*32*x*10*x*1*	*Apprentissage*	4.18	3.38	56.9	75.7	95.6
		Test	4.21	3.43	56.8	75.3	95.4
MLP	*32*x*15*x*1*	*Apprentissage*	4.00	3.34	59.8	77.6	96.4
		Test	4.05	3.26	58.2	76.2	96.2

TAB. 2.2 – *Performances de perceptrons multicouches sur le problème de prédiction du champ radioélectrique.*

Les critères de qualité sont atteints. Les travaux présentés par la suite auront donc davantage pour tâche d'apporter de la connaissance sur le problème que d'améliorer proprement dit les résultats (il suffirait pour cela d'augmenter le nombre d'unités sur la couche cachée (voir, à titre d'exemple, les performances d'un MLP avec 15 unités cachées dans le tableau 2.2)).

Les réseaux feed-forward s'adaptent facilement à toutes sortes de régression [Canu, 1996]. Si le réseau comporte plusieurs entrées, le modèle de régression est dit multiple, et s'il comporte plusieurs sorties, le modèle de régression est multivarié. Si des connexions complémentaires relient directement les entrées aux sorties, on ajoute ce que les statisticiens appellent un effet principal des données sur la régression. Si le nombre d'unités sur la couche cachée est faible, comparativement au nombre d'entrées, le MLP devient un modèle paramétrique qui constitue une alternative intéressante à la régression polynômiale. L'augmentation des unités sur la couche cachée le transforme en modèle semi-paramétrique telle que la poursuite en projection, à ceci prêt que le MLP utilise des fonctions d'activation prédéterminées, alors que la méthode statistique utilise des lissages non linéaires adaptatifs. Si maintenant le nombre d'unités de la couche cachée est élevé, le modèle devient non paramétrique et comparable à la régression par fonctions noyaux et par fonctions splines. L'intérêt des MLPs est donc de pouvoir évoluer d'un modèle paramétrique de base jusqu'à un modèle non paramétrique très adaptatif. De plus, les MLPs sont plus stables que les polynômes de degré élevé et ne nécessitent pas d'information supplémentaire comme les nœuds pour les splines, mais il n'existe pas de preuve qu'ils atteignent le minimum global. De plus, à la différence de polynômes et des splines, les MLPs s'étendent au cas multiple multivarié sans accroissement exponentiel du nombre de paramètres.

Récemment dans l'histoire des statistiques, de nombreuses méthodes ont apporté une solution non linéaire au problème de régression. L'idée commune est d'insérer entre les données et l'approximation linéaire une modélisation non linéaire des résidus issus de l'optimisation de l'approximation linéaire par l'application d'un critère des moindres carrés ou d'un critère de vraisemblance. Nous allons voir que ces méthodes sont facilement

réalisables par des perceptrons multicouches [Gallinari *et al.*, 1988; Gallinari *et al.*, 1991; Sarle, 1994]

La poursuite en projection (en régression)

Deux statisticiens, Friedman et Stuetzle, ont proposé une méthode dite de poursuite en projection (Projection Pursuit Regression [Friedman and Stuetzle, 1981]), pour laquelle les données sont projetées sur j axes révélateurs qui, après transformation non linéaire ϕ_j, se combinent linéairement. Les fonctions non linéaires ne sont pas statiques. Elles sont obtenues à partir des résidus, par des méthodes de lissage unidimensionnel comme l'interpolation par des splines cubiques, courbes de Bézier, fonctions polynômes etc.

- Dans la version d'origine, le nombre de fonctions ϕ_j augmente progressivement pour contrôler la complexité du modèle. La régression est donnée par la formule :

$$y_k = \sum_{j=1}^{M} \phi_j(\mathbf{u}_j^T \mathbf{x}) \tag{2.5}$$

  Le vecteur $\mathbf{u}_j$ de dimension p projette les données sur une ligne directrice correspondant à une direction révélatrice.
- Dans une version ultérieure, appelée SMART ([Friedman, 1985]), les fonctions non linéaires sont partagées par les sorties. L'optimisation des fonctions est globale.

$$y_k = \sum_{j=1}^{M} w_{jk} \phi_j(\mathbf{u}_j^T \mathbf{x} + u_{j0}) + w_{0k} \tag{2.6}$$

Ces modèles sont facilement réalisables avec un perceptron multicouches. Les fonctions non linéaires sont apprises par des sous-réseaux munis de fonctions non linéaires sur la couche cachée (figure 2.14).

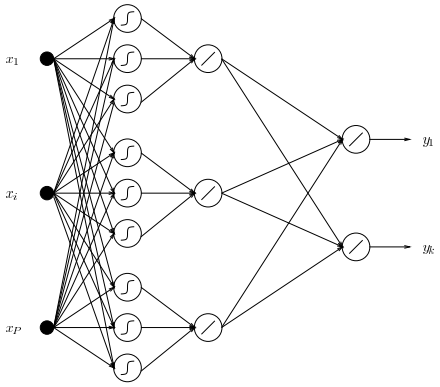


FIG. 2.14 – *Illustration de la réalisation d'une poursuite en projection par un réseau connexionniste.*

La comparaison entre les modèles de poursuite en projection et les MLPs [Hwang *et al.*, 1992] a montré une grande similarité de comportement.

Les modèles additifs généralisés

Les modèles additifs généralisés de Hastie et Tibshirani (Generalized Additive Model [Hastie and Tibshirani, 1990]) sont des modèles de prédiction à mi-chemin, du point de vue de la capacité et de la complexité, entre les modèles de régression linéaire et les modèles de régression de types MLPs non linéaires, polynômes de haut degré et autres fonctions splines. Ce sont des modèles non linéaires de régression dont la particularité est que chacune des entrées calcule séparément une partie de la prédiction. Le résultat est la somme des prédictions isolées. De fortes restrictions résultent de cette méthode. Il est évident que toute fonction avec des termes mixtes tels que $x_1 x_2$ ne peut pas être modélisée. L'intérêt est de pouvoir visualiser la part d'une donnée sur la prédiction comparativement aux autres. En effet, la prédiction y_k est modélisée à partir de la somme de fonctions non linéaires $\phi(.)$ appliquées individuellement à chaque entrée x_i (équation 2.7).

$$y_k = g\left(\sum_{i=1}^{P} \phi_i(x_i) + w_0\right) \quad \text{avec } g(.) \text{ une fonction non linéaire} \tag{2.7}$$

Ici encore, il est possible de réaliser cette technique sous forme neuronale (figure 2.15). Chaque fonction $\phi_i()$ est approximée par un MLP constitué de fonctions d'activation sigmoïdales.

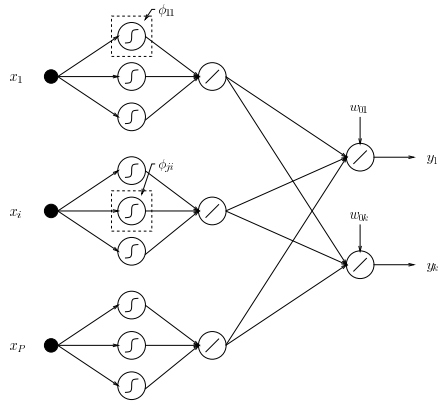


FIG. 2.15 – *Illustration de la réalisation d'un modèle additif de régression par un réseau connexionniste.*

Les réseaux connexionnistes peuvent représenter d'autres modèles additifs, comme par exemple celui de Geiger (Topologically Distributed Encoding [Geiger *et al.*, 1990]) qui utilise quant à lui des fonctions gaussiennes.

MARS (Multivariate Adaptative Regression Splines) proposé par Friedman étend les modèles additifs en permettant les interactions entre les variables [Friedman, 1991].

$$y_k = \sum_{i=1}^{P} \phi_i(x_i) + w_0 \tag{2.8}$$

Une analyse des similitudes peut être obtenue par un réseau linéaire à une couche cachée [Rao, 1960]. Si l'on insère une couche de fonctions non linéaires comme prétraitement, on crée une généralisation du modèle. La couche cachée de faible dimension correspond toujours à une réduction de la dimension.
De nombreux travaux successifs ont montré le lien entre l'analyse discriminante et les réseaux de neurones. Dans un premier temps, Gallinari et al. [Gallinari *et al.*, 1991] ont démontré qu'un perceptron à une couche cachée réalise une analyse linéaire discriminante si on fait correspondre les entrées-sorties du réseau de neurones avec les doubles entrées d'un tableau de contingence (ou tableau croisé) de manière supervisée. Le cas non linéaire a été abordé par Asoh et Otsu [Asoh and Otsu, 1989]. Finalement, le cas général incluant un apprentissage non supervisé a été proposé par Baldi et Hornik [Baldi and Hornik, 1989].

Enfin, signalons que bien qu'un MLP soit un approximateur universel de fonctions, d'autres réseaux multicouches présentent des caractéristiques intéressantes du point de vue statistique en général et prédictif en particulier. Ainsi si un réseau à une couche cachée peut modéliser n'importe quelle fonction, un réseau à plusieurs couches cachées peut obtenir la même modélisation pour une architecture réduite.

2.3.3 Les arbres de décision et régression

Les arbres de classification ou de régression, dont les meilleurs représentants sont l'algorithme CART (Classification And Regression Trees [Breiman *et al.*, 1984]) et ID3 (Induction of Decision Trees [Quinlan, 1986]), sont des méthodes récursives de partitionnement binaire de l'espace des données. Ces algorithmes positionnent des hyperplans perpendiculairement aux axes des variables d'entrée pour délimiter l'espace (figure **??**). Chaque région ainsi définie correspond à une classe. Les nœuds de l'arbre correspondent aux surfaces séparatrices. Certaines méthodes permettent qu'elles ne soient pas perpendiculaires aux axes, voire non planes. Les feuilles de l'arbre correspondent aux différentes régions.

Une représentation neuronale peut être facilement déduite d'un arbre de décision (figure **??**). Le passage d'une représentation à l'autre est détaillé dans [Sirat and Nadal, 1990; Sethi, 1990; Brent, 1991]. L'apprentissage des critères de séparation est locale (i.e. pour un nœud) ou optimisé (pour un nœud et ses fils [D'arché-Buc *et al.*, 1994]).

Expérimentation : L'extension de l'algorithme ID3, nommé C4.5 par Quinlan [Quinlan, 1986], a été utilisée avec succès pour séparer deux groupes dont les valeurs d'atténuation différaient trop pour être assimilées à une seule classe (voir page 21). L'avantage de pouvoir définir un réseau connexionniste qui effectuera cette discrimination de manière automatique réside dans l'utilisation du même formalisme pour discriminer et prédire. Le réseau qui effectue l'opération de discrimination peut être assemblé en amont du réseau prédictif. Dans ce cas, le prétraitement est réalisé par le modèle.

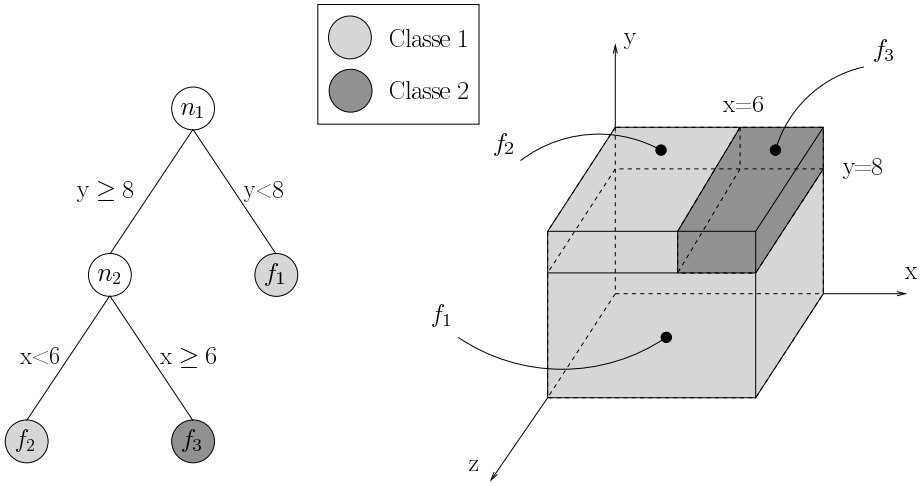


FIG. 2.16 – *Exemple de partitionnement de l'espace d'entrée par un arbre de décision. A gauche, un arbre de décision comportant deux nœuds (n_1 et n_2) et trois feuilles (f_1, f_2 et f_3), discrimine deux classes en fonction des valeurs des deux variables d'entrée x et y. A droite, l'arbre de décision a pour effet de partitionner l'espace d'entrée par des hyperplans perpendiculaires aux axes dont les équations correspondent aux valeurs de transition distinguant une branche de l'autre.*

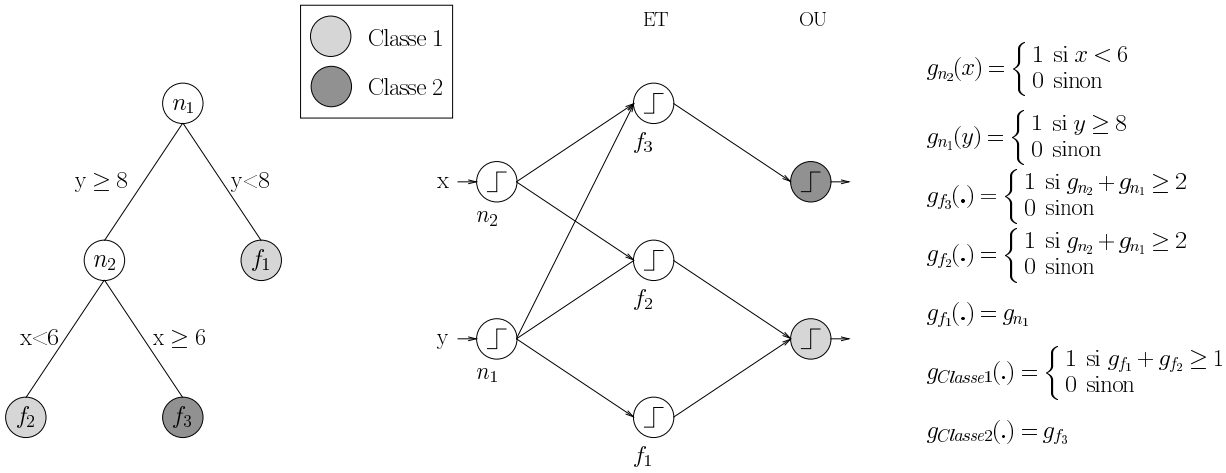


FIG. 2.17 – *L'arbre décisionnel, représenté à gauche, peut être modélisé par un réseau connexionniste en adoptant l'architecture présentée au centre et en prenant par exemple les fonctions à seuil de droite (les poids sont pris égaux à 1). Les neurones de la première couche correspondent aux nœuds et le résultat de leur fonction d'activation indique quelle branche est empruntée, ceux de la deuxième couche correspondent aux feuilles et sont connectés aux nœuds qui permettent d'y accéder, et ceux de la dernière couche correspondent aux classes.*

2.4 Les réseaux récurrents

Les réseaux récurrents [Hertz *et al.*, 1991, chap. 6] peuvent être vus comme une extension des réseaux feed-forward. Ils contiennent au moins une connexion, dite récurrente, qui met en relation un neurone avec un neurone de la même couche ou d'une couche précédente, introduisant par là même un cycle à l'intérieur du réseau. Un cycle va per-

mettre à une information de rester dans le système. Le modèle gardera ainsi une trace de l'activité de certains neurones à l'instant précédent. Puisque la valeur de sortie d'un neurone est le résultat d'un calcul initié à partir de la forme d'entrée, les réseaux récurrents permettent une mémorisation à court terme, sous une forme codée, des données précédemment vues. Cette mémoire est une information supplémentaire qui vient s'ajouter aux données courantes. L'intérêt de tels réseaux résulte de l'influence bénéfique que peuvent avoir les formes précédentes, ou plus exactement des calculs obtenus à partir des formes précédentes, sur le traitement de la forme courante. L'influence ne sera pertinente que si les formes présentées successivement au modèle sont rattachées les unes aux autres. La relation qui les unit génère, par l'intermédiaire des connexions récurrentes, un contexte qui influencera positivement le traitement en cours.
Les réseaux récurrents s'appliquent donc principalement aux problèmes qui présentent une continuité temporelle. L'évolution des données peut bien sûr être prise en compte dans les modèles feed-forward si l'on considère en entrée non plus $\mathbf{x}(t)$ mais la concaténation de $\mathbf{x}(t-n)$,...,$\mathbf{x}(t-1)$,$\mathbf{x}(t)$ ou encore la concaténation de $\mathbf{x}(t)$,$\mathbf{x'}(t)$,...,$\mathbf{x^{(n)}}(t)$. Dans le dernier cas, la complexité du calcul des dérivées d'ordre élevé s'ajoute à l'augmentation de la taille du vecteur d'entrée et donc du réseau. L'apprentissage devient plus difficile et plus lent.
Les modèles récurrents de Jordan et de Elman offrent deux solutions différentes à ce problème en proposant deux mémorisations différentes des formes précédentes. Dans ces types de modèles récurrents, le contexte c est ajouté à la forme courante $\mathbf{x}$ sur la couche d'entrée. Pour le premier élément, le contexte n'existe pas, il faut donc amorcer le système en initialisant les entrées contextuelles à zéro, ou à des valeurs aléatoires.

2.4.1 Le modèle de Jordan

Le réseau de Jordan [Jordan, 1986b; Jordan, 1986a] est un des premiers réseaux récurrents à être apparu, et aussi un des plus utilisés. Dans un réseau récurrent de Jordan, les valeurs de sortie, obtenues pour la forme précédente, sont recopiées sur la couche d'entrée, comme autant de neurones supplémentaires (couche de contexte) et sont totalement connectés à la couche cachée (figure 2.18) (a)). Deux types de connexions récurrentes interviennent. L'une, au niveau du neurone de sortie et de poids égal à un, active le souvenir des résultats obtenus à l'instant d'avant. L'autre, auto-connexion de chaque neurone de la couche de contexte et de poids égal au coefficient α, entretient, plus ou moins fortement, le souvenir des résultats obtenues jusqu'à présent en privilégiant les plus récents. Le contexte à l'instant t, $c(t)$, dépend de la couche de sortie $y(t-1)$ et du contexte $c(t-1)$, à l'instant $t-1$, suivant la formule :

$$c(t) = y(t-1) + \alpha.c(t-1)$$

où $\alpha \in [0,1[$ est un coefficient dont dépend l'étendue de la mémoire. Si $\alpha = 0$, le contexte se résume aux valeurs de sortie de la forme précédente. L'augmentation du coefficient α augmente l'étendue de la mémoire. L'influence ne se limite pas à la situation

immédiatement précédente mais s'étend à toutes les formes présentées.

$$c(t) = y(t-1) + \alpha.(y(t-2) + \alpha.c(t-2))$$
$$c(t) = y(t-1) + \alpha.y(t-2) + \alpha^2.c(t-2)$$
$$c(t) = y(t-1) + \alpha.y(t-2) + \alpha^2.y(t-2) + \alpha^3.y(t-3) + ... + \alpha^n.y(t-n)$$

L'influence d'une sortie s'amoindrit exponentiellement au fur et à mesure que le temps s'écoule. Si le coefficient α est proche de un, la valeur du neurone peut être élevée si les sorties désirées ne sont pas standardisées ou au moins centrées. Le modèle de Jordan introduit une mémorisation à court terme et un phénomène d'oubli par le bias des deux types de connexions récurrentes.

2.4.2 Le modèle de Elman

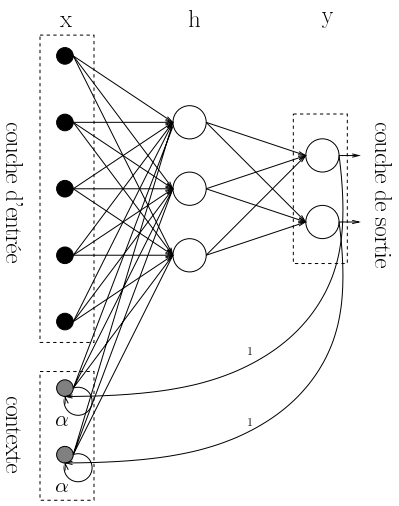
Dans un réseau récurrent de Elman [Elman, 1990], le contexte $c(t)$ à l'instant t est la copie de la couche cachée $h(t-1)$ à l'instant $t-1$.

$$c(t) = h(t-1)$$

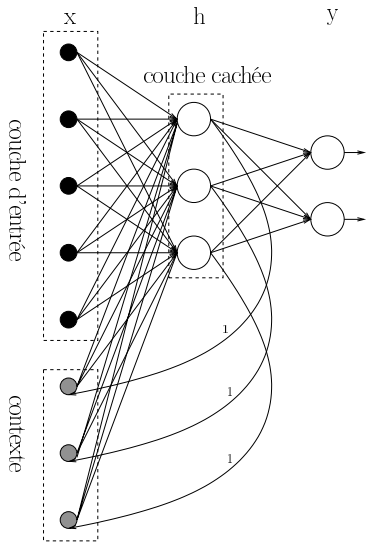
L'influence des formes précédentes est à la fois plus simple à percevoir puisqu'elle ne concerne que la forme vue à l'instant d'avant, et plus complexe car les valeurs de la couche cachée ne sont pas aussi faciles à interpréter que les valeurs des variables de sortie. Néanmoins, nous pouvons dire que le réseau récurrent de Elman mémorise, sous une forme codée, les valeurs d'entrée du réseau à l'étape précédente. Ici donc, l'influence proviendra plus d'une transformation, dans la majeure partie des cas non linéaire, des entrées que des sorties désirées. Une autre comparaison avec le modèle Jordan nous signale que le contexte sera représenté par un nombre plus important de valeurs car, dans la grande majorité des cas, le nombre de neurones de la couche cachée est supérieur au nombre de neurones de la couche de sortie. Donc, bien souvent, le contexte du modèle de Elman est plus riche en information et sera plus influent face au nombre de neurones d'entrée du réseau. Le modèle de Elman introduit donc une mémorisation à très court terme par l'ajout d'une connexion récurrente par neurone de la couche cachée et à destination d'un neurone supplémentaire situé sur la couche d'entrée (figure 2.18) (b)).

D'autres réseaux récurrents traitent des séquences en introduisant des connexions réflexives sur les neurones de la couche d'entrée [Stornetta *et al.*, 1988] ou de la première couche cachée [Mozer, 1989]. Citons également le modèle de Hopfield qui permet de mémoriser des représentations binaires, et les machines de Boltzmann [Hinton and Sejnowski, 1986; Ackley *et al.*, 1985] qui introduisent des phénomènes stochastiques dans le calcul neuronal.

Expérimentation L'atténuation du champ radioélectrique est un phénomène continu (figure 2.19 (b)). Des situations géographiquement proches auront en général des atténuations proches. Puisque les relevés sont obtenus par le déplacement continu d'un véhicule équipé d'instruments de mesure (figure 2.19 (a)), nous pouvons tirer avantage de ces circonstances pour améliorer la prédiction de l'atténuation, en utilisant la similitude qui



(a) Le réseau récurrent de Jordan ajoute aux valeurs de la forme d'entrée courante les valeurs de sortie obtenues pour la forme précédente introduisant par là même une relation entre les formes successives. Elle dépend de l'ordre de présentation des formes. Une connexion auto-récurrente de poids α régule la mémoire temporelle de cette relation.

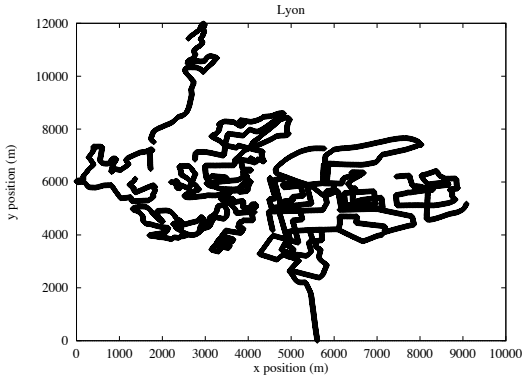


(b) Le réseau récurrent de Elman ajoute aux valeurs de la forme d'entrée courante les valeurs de la couche cachée obtenues pour la forme précédente (i.e. sa représentation interne). Il introduit ainsi une relation entre les formes successives qui dépend de l'ordre de présentation des formes.

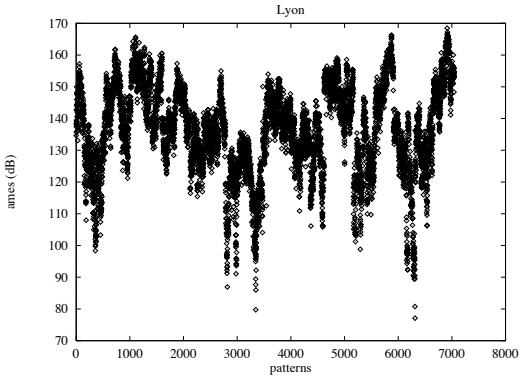
FIG. 2.18 – *Illustrations des réseaux récurrents de Jordan et de Elman.*

existe entre deux relevés successifs. L'ajout des informations relatives aux situations voisines constitue une information contextuelle qui pourra être bénéfique au modèle. Il reste à déterminer quelle forme aura cette information. Nous avons précisé les complications engendrées par l'augmentation des entrées suite à la concaténation de plusieurs vecteurs. Les modèles récurrents s'avèrent donc intéressant à expérimenter dans ce cadre [Bougrain and Alexandre, 1999a; Bougrain, 1998a].

Pour tester notre hypothèse que l'ajout d'une information contextuelle produite par les situations voisines améliore les résultats, la valeur de l'atténuation, mesurée pour la situation précédente, est ajoutée aux valeurs d'entrée du modèle (figure 2.20 (a)). Il ne s'agit donc pas encore ici d'un réseau récurrent mais de la simple augmentation de la taille de la couche d'entrée par un contexte mesuré physiquement et non pas estimé par le réseau. L'erreur de la prédiction sur le corpus de test baisse de moitié, ce qui correspond à une augmentation de la précision de la prédiction de 96,7% à 98% (tableau 2.3. Bien évidemment, ce résultat ne constitue pas une amélioration en soi puisque la valeur exacte de l'atténuation du voisin ne sera pas connue pendant la phase d'utilisation, mais il valide notre hypothèse et donne une indication sur la capacité limite d'un modèle de Jordan.



(a) Les mesures relevées dans une zone (ici Lyon) proviennent d'un ensemble de parcours continus et sinueux.



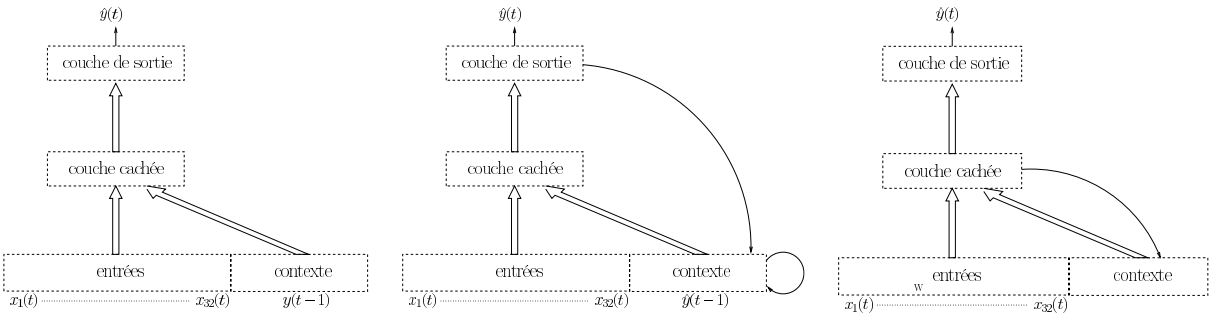
(b) Les mesures de l'atténuation du champ radioélectrique (ici de Lyon) témoignent de la continuité du phénomène et de sa variabilité.

FIG. 2.19 – *Continuités des relevés et de l'atténuation du champ radioélectrique.*

Nous avons appliqué le modèle de Jordan sur une architecture similaire à celle du perceptron multicouches, seul le neurone supplémentaire de la couche d'entrée et la connexion récurrente de la sortie faisant la différence (figure 2.20 (b)). Le contexte est donné par une seule valeur, résultat d'un calcul sur l'ancienne valeur du contexte et la valeur prédite à l'instant précédent. Le paramètre α règle la taille de la fenêtre temporelle qui détermine l'influence des prédictions précédentes. L'erreur a été diminuée (tableau 2.3, preuve que le contexte apporte une information utile. La modification du paramètre α ne semble pas affecter les performances. On peut donc en déduire que les situations lointaines sont trop espacées pour renforcer véritablement la prédiction.

Dans le modèle de Jordan, le contexte ne représente que $1/33^{\text{e}}$ des valeurs d'entrée. C'est peut être insuffisant pour influencer pleinement la prédiction. Dans le modèle de Elman, le contexte est représenté par un nombre de neurones égal à la taille de la couche cachée, soit 10 valeurs dans notre cas (figure 2.20 (c)). Le contexte représentera alors le quart de l'information d'entrée. En choisissant le modèle de Elman, les performances sont à nouveau améliorées (tableau 2.3, diminuant l'erreur d'un quart de décibel et montrant l'intérêt d'introduire un contexte plus riche.

Au delà de l'erreur moyenne, il faut s'intéresser à l'évolution des performances dans le temps. En effet, les réseaux récurrents utilisent des valeurs calculées à l'instant précédent. Si les valeurs précédentes ne sont pas pertinentes, les calculs en cours ne donneront pas de valeurs pertinentes. La prédiction sera de moins en moins bonne et l'on assistera à une dérive. L'observation de l'évolution des performances dans le temps nous informe sur l'éventuelle dégradation de la prédiction, et donc sur la dégradation du contexte. L'analyse révèle que, dans notre cas, les performances sont constantes dans le temps. Les prédictions



(a) Perceptron multicouches incluant en entrée la valeur mesurée de l'atténuation de la situation précédente $y(t-1)$ pour estimer l'atténuation $\hat{y}(t)$ de la situation courante.

(b) Réseau de Jordan utilisant la valeur estimée de l'atténuation à la situation précédente $\hat{y}(t-1)$ pour estimer l'atténuation $\hat{y}(t)$ de la situation courante.

(c) Réseau de Elman utilisant les valeurs de la couche cachée calculées à la situation précédente pour estimer l'atténuation $\hat{y}(t)$ de la situation courante.

FIG. 2.20 – *Réseaux utilisés pour évaluer l'intérêt d'une information contextuelle dans la prédiction de l'atténuation du champ radioélectrique.*

Modèle	Phase	$\bar{e}$ (dB)	σ (%)	Q_4 (%)	Q_6 (%)	Q_{11} (%)
MLP	*Apprentissage*	4.27	3.45	56,2	74,8	95,1
	Test	4.36	3.50	54,8	73,7	94,8
MLP+AMES(t-1)	*Apprentissage*	2.62	2.12	56.9	75.8	96.4
	Test	2.68	2.15	55.6	74.7	96.1
JORDAN	*Apprentissage*	4.18	3.39	56.5	75.2	95.6
	Test	4.28	3.43	55.1	74.1	95.3
ELMAN	*Apprentissage*	4.03	3.26	56.7	75.6	96.1
	Test	4.12	3.30	55.4	74.5	95.8

TAB. 2.3 – *Performances de modèles contextuels sur le problème de prédiction du champ radioélectrique.*

sont donc suffisamment bonnes pour ne pas entraîner à la longue une détérioration du contexte.

Jusqu'à présent nous avons mis en avant l'intérêt d'utiliser des modèles contextuels pour augmenter les performances en prédiction. Mais en choisissant une information contextuelle extraite du voisinage temporel, nous devrions également observer une diminution de la fluctuation des prédictions contiguës. L'écart entre les atténuations mesurées peut nous donner une indication sur la similitude des situations voisines avec la situation courante. Plus l'écart est petit, plus les voisins auront une influence bénéfique sur la prédiction, à condition que la fonction à modéliser soit continue, ce qui n'est pas le cas par exemple en cryptologie. La moyenne des écarts entre les prédictions de deux situations successives a été calculée et comparée à la moyenne des écarts entre les atténuations mesurées (tableau 2.4). Le premier réseau que nous avons testé présente un écart plus important que le

MLP. Il est donc plus à même de suivre les fluctuations des atténuations grâce à un bon contexte. La moyenne des écarts pour les réseaux de Jordan et de Elman est inférieure à celle d'un perceptron multicouches, ce qui signifie que les réseaux récurrents effectuent un lissage des valeurs de sortie. Dans un voisinage, les prédictions sont plus cohérentes les unes avec les autres.

Modèle	Ecart moyen (dB)	
	Apprentissage	Test
Mesuré	2.75	2.90
MLP	2.12	2.17
MLP+AMES(t-1)	2.27	2.32
JORDAN	1.51	1.55
ELMAN	1.60	1.63

TAB. 2.4 – *Moyenne des écarts entre les affaiblissements, mesurés et prédits, de deux situations voisines géographiquement.*

Finalement, nous avons montré qu'un réseau muni d'une entrée supplémentaire correspondant à la valeur désirée de l'exemple précédent permet d'évaluer l'intérêt d'ajouter une information de contexte temporel et d'estimer la performance limite réalisable par un réseau de Jordan. Le modèle de Jordan améliore les performances, mais son contexte est trop faible par rapport au reste des données d'entrée pour influencer suffisamment la prédiction. Le modèle de Elman permet d'augmenter l'influence du contexte et d'améliorer les performances. La convergence est plus rapide pour le MLP augmenté de l'atténuation précédente et pour le réseau de Jordan. L'information supplémentaire rend la prédiction plus facile. Par contre, le réseau de Elman converge plus lentement en raison de la forte augmentation du nombre de poids à apprendre. Nous avons attiré l'attention sur l'importance de ne pas cantonner, pour ce type de réseau, l'étude à la moyenne des performances, mais d'étudier également l'évolution des performances dans le temps afin d'observer qu'il n'y avait pas de dérive. Enfin, au delà de la diminution de l'erreur, les réseaux récurrents permettent d'améliorer la cohérence des réponses de situations proches ce qui est particulièrement important dans de nombreuses applications, dont la nôtre. L'élargissement de la fenêtre temporelle augmentera le lissage. L'ajout de nouveaux voisins doit s'accompagner de l'assurance qu'ils soient suffisamment utiles, c'est-à-dire, dans notre cas, qu'ils ne soient pas trop éloignés. De plus, le renforcement de la contrainte de cohérence pourrait finir par amoindrir la variabilité des sorties. Donc, l'utilisation des modèles récurrents permet d'augmenter les performances et de diminuer les fortes erreurs par un lissage des prédictions.

2.5 Les modèles auto-organisés

La particularité des modèles connexionnistes qui appartiennent à cette catégorie réside dans le fait que l'information pertinente ne proviendra pas d'une mise en correspondance de deux ensembles d'exemples (source-cible), mais elle sera extraite uniquement de

l'organisation des données d'entrée. Dans ces circonstances, l'information recherchée par les méthodes auto-organisées est différente de celle recherchée par les méthodes hétéro-associatives.
Après avoir présenté les éléments nécessaires à la compréhension des réseaux connexionnistes auto-organisés les plus intéressants, nous détaillerons les différents types de tâches qu'ils permettent de réaliser, parmi lesquelles citons le partitionnement des données, la quantification de l'espace des données, la réduction de la dimension des données ou encore l'extraction de caractéristiques à partir des données.

2.5.1 Apprentissage compétitif

Les méthodes compétitives [Rumelhart and Zipser, 1985] contraignent les neurones de sortie à une compétition dont l'enjeu est l'attribution de la forme d'entrée au neurone vainqueur. En retour, le neurone gagnant modifie ses poids pour encore mieux caractériser la forme d'entrée. De cette manière, les neurones de la couche de sortie se spécialisent sur une partie des données. Le but commun des méthodes compétitives est de distribuer un nombre de vecteurs égal au nombre de neurones de sortie dans l'espace des données. La distribution de ces vecteurs doit rendre compte de la probabilité de distribution de la population, qui n'est généralement pas connue explicitement mais peut être appréhendée au travers d'un certain nombre d'exemples. Nous verrons que cette simple opération peut avoir des conséquences intéressantes. Pour la réaliser, deux couches de neurones suffisent : la première représente les entrées, la seconde les sorties i.e. les groupes. Les deux couches sont entièrement connectées (figure 2.21 (a)), donc le nombre de connexions qui arrivent à chaque neurone de sortie est égal au nombre de neurones d'entrée. Les poids d'un neurone de sortie constituent un vecteur particulier dans l'espace des données, puisque ce vecteur est de même dimension que la dimension des formes d'entrée. Chaque neurone de sortie est donc caractérisé par un vecteur de référence, son vecteur poids.

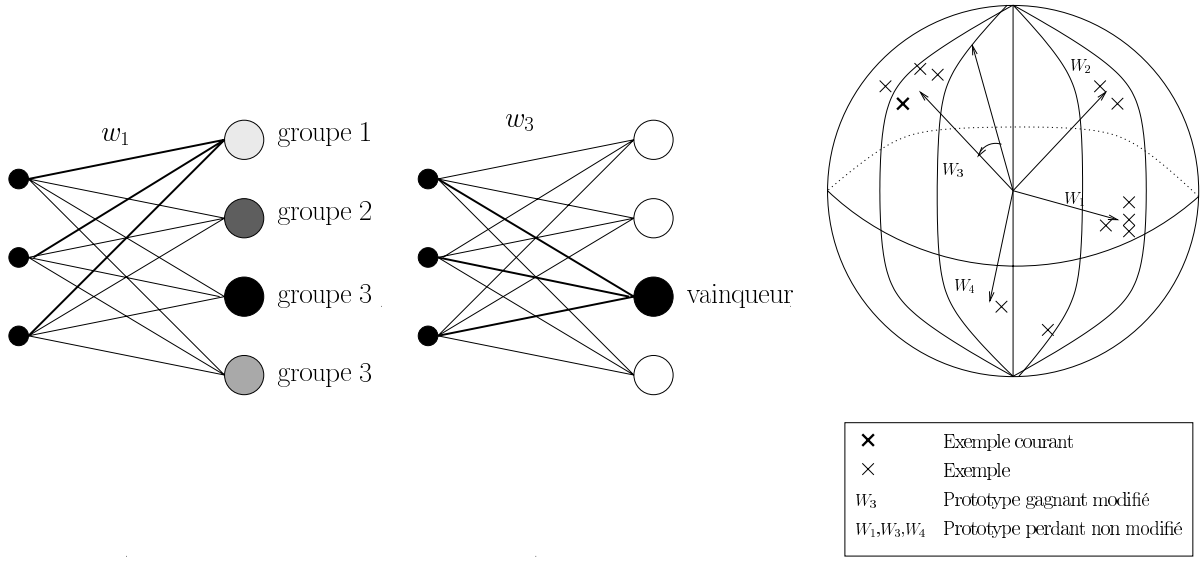
Dans les méthodes compétitives, lorsqu'on présente une forme d'entrée, un seul neurone de la couche de sortie, le vainqueur, s'active. C'est celui dont le vecteur répond le mieux au critère de sélection basé sur la comparaison entre le vecteur d'entrée du réseau et le vecteur poids de chaque neurone de sortie. On peut vouloir maximiser la ressemblance ou minimiser la différence.

Maximiser la ressemblance :

Le vecteur d'entrée $\mathbf{x}$ est comparé à un vecteur de référence $\mathbf{w}_j$ par calcul du produit cartésien.

$$y_j = \sum_i w_{ij} x_i = \mathbf{w}_j^T \mathbf{x}$$

Cette valeur de ressemblance constitue la valeur d'activation du neurone de sortie. L'estimation de la ressemblance par le produit cartésien constitue une solution biologiquement plausible [Kohonen, 1982], mais elle nécessite la normalisation des vecteurs. Prenons comme exemples les situations de la figure 2.22. Dans le premier cas, c'est-à-dire lorsque les vecteurs sont normalisés, on aimerait affecter $\mathbf{x}$ à la classe 1. Le produit cartésien permet bien de réaliser cette opération. En effet, $\mathbf{w_1}^T\mathbf{x} > \mathbf{w_2}^T\mathbf{x}$. Dans le second cas, sans



(a) Phase 1 (b) Phase 2 (c) Espace d'entrée.

FIG. 2.21 – *Dans un réseau compétitif, plus l'activité d'un neurone de sortie est forte (sombre sur la figure), plus il est représentatif de la forme d'entrée (a). Le neurone représentant le groupe 3 est déclaré vainqueur. Il s'active, les autres se désactivent (b). Le représentant de la classe 3, w_3, est donc le plus proche du point de vue du produit cartésien de la forme courante. Il est le seul à se déplacer dans la direction de l'individu de la classe qu'il caractérise (c).*

normalisation, on aimerait affecter $\mathbf{x}$ à la classe 2 mais le produit cartésien désigne $\mathbf{w_1}$ comme vainqueur, car ici aussi on a $\mathbf{w_1}^T\mathbf{x} > \mathbf{w_2}^T\mathbf{x}$. L'opération de mise à jour du vecteur de référence est donnée par la formule :

$$\mathbf{w_j}(t+1) = \frac{\mathbf{w_j}(t) + \eta(\mathbf{w_j}(t) - \mathbf{x}(t))}{||\mathbf{w_j}(t) + \eta(\mathbf{w_j}(t) - \mathbf{x}(t))||}$$

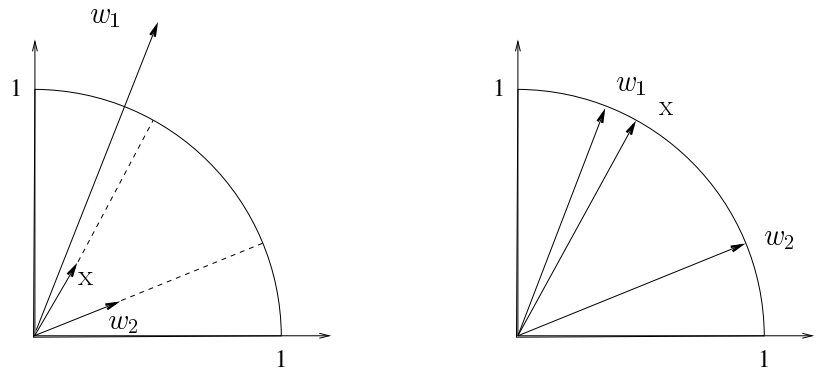
Minimiser la différence :

L'évaluation de la différence par calcul de la distance euclidienne entre deux vecteurs ne nécessite pas la normalisation des vecteurs. Le gagnant est déterminé par la formule suivante : $\forall j \in [1,N] \quad ||\mathbf{w_c} - \mathbf{x}|| \leq ||\mathbf{w_j} - \mathbf{x}||$. Cette formule vérifie la formule du produit cartésien dans le cas où les vecteurs sont normalisés (on a $||\vec{w}_c - \vec{x}|| \leq ||\vec{w}_j - \vec{x}|| \Rightarrow \cos(\vec{w}_c,\vec{x}) > \cos(\vec{w}_j,\vec{x})$). L'opération de mise à jour du vecteur de référence n'est plus une rotation mais une translation. On a :

$$\mathbf{w_j}(t+1) = \mathbf{w_j}(t) + \eta(\mathbf{w_j}(t) - \mathbf{x}(t)) \tag{2.9}$$

L'initialisation :

L'initialisation des vecteurs de référence est une étape importante, spécialement quand l'espace d'entrée est grand ou possède une dimension élevée. On imagine aisément qu'un



(a) Sans normalisation (b) Avec normalisation

FIG. 2.22 – *Influence de la normalisation des vecteurs dans la désignation du vecteur de référence le plus proche du point de vue du produit cartésien. Si les vecteurs ne sont pas normalisés, l'élément x sera affecté à la classe 2 dont w_2 est le représentant (a). Si les vecteurs sont normalisés, il sera affecté à la classe 1 (b).*

vecteur choisi aléatoirement dans l'espace d'entrée peut se situer si loin des exemples par rapport aux autres vecteurs de référence qu'il ne gagnera jamais, et par conséquent restera dans une zone désolée allant à l'encontre de sa vocation. Plusieurs solutions existent :

1. Les vecteurs de références peuvent être pris aléatoirement parmi les exemples.
2. Les vecteurs de références peuvent être organisés sur le plan principal d'inertie puis repositionnés dans l'espace d'entrée [Elemento, 1999]. Les prototypes sont ainsi répartis régulièrement dans les régions où les données sont présentes (figure 2.23).

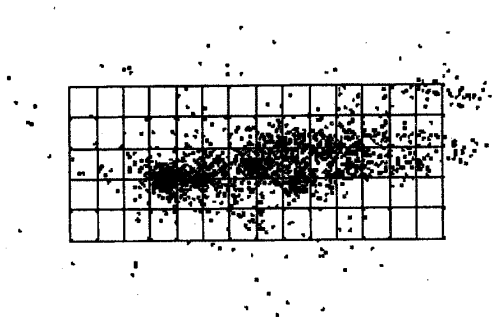


FIG. 2.23 – *Répartition initiale régulière des prototypes sur le plan principal des données.*

Une alternative à l'isolement de vecteurs de référence est de contraindre ceux qui ne gagnent pas à modifier leur position en direction des données.

3. Les perdants peuvent être légèrement déplacés. La modification de la règle de modification des poids a donné naissance à de nouveaux apprentissages compétitifs connus sous le nom de *leaky learning*.

4. La fréquence des victoires peut intervenir dans la détermination du gagnant. Les neurones qui perdent souvent augmentent leurs chances de sélection ([Ahalt *et al.*, 1990; DeSieno, 1988]).

La fonction de coût :

Les données d'entrée sont regroupées par similarité. La notion de similarité fait généralement appel à un critère de distance. Pour mesurer la qualité du regroupement, on utilise un critère quadratique tel que : $E = \sum_n ||\mathbf{W_j} - \mathbf{x}^n||^2$. Ce critère est minimisé par la règle de modification des poids de l'équation 2.9.

Nous présentons maintenant un algorithme général de compétition.

Algorithme 5 Algorithme compétitif

1. Normaliser les vecteurs d'entrée.
2. Initialiser les vecteurs de référence $\mathbf{w_j} \in \mathbb{R}^n$ (avec $i \in [1,N]$ et N est le nombre de classes) suivant $p(\mathbf{x})$, où $\mathbf{x} \in \mathbb{R}^n$ est une forme d'entrée choisie aléatoirement dans l'espace d'entrée $\mathbb{R}^n$ avec la fonction de densité de probabilité de sélection $p(\mathbf{x})$.
3. Initialiser le paramètre temporel t :
t = 0.
4. Choisir aléatoirement un vecteur d'entrée $\mathbf{x}$ selon $p(\mathbf{x})$.
5. Calculer l'activation des neurones de sortie par produit cartésien :
$y_j = \sum_i w_{ij} x_i = \mathbf{w_j}^T \mathbf{x}$
6. Déterminer le vecteur de référence $\mathbf{w_c}$ le plus proche de $\mathbf{x}$:
$\forall j \in [1,N] \quad \mathbf{w_c}^T \mathbf{x} \geq \mathbf{w_j}^T \mathbf{x}$.
7. Réaffecter les valeurs de sortie :
$\begin{cases} y_c = 1 \\ y_j = 0 \quad \forall j \neq c \end{cases}$
8. Modifier les poids du vainqueur suivant la formule suivante :
$$\mathbf{w_j}(t+1) = \frac{\mathbf{w_j}(t) + \eta(\mathbf{w_j}(t) - \mathbf{x}(t))}{||\mathbf{w_j}(t) + \eta(\mathbf{w_j}(t) - \mathbf{x}(t))||}$$
9. Incrémenter t :
t=t+1.
10. Retourner à l'étape 4, si le critère d'arrêt n'est pas validé.

L'étape 6 peut être réalisée automatiquement en ajoutant des connexions entre les neurones représentant les classes avec des poids inhibiteurs [Lippmann, 1989]. L'étape 7 constitue l'aspect compétitif, dit winner-takes-all. Seul le vainqueur est activé, les autres sont désactivés. La règle d'apprentissage fait effectuer au vecteur gagnant une rotation en direction de la forme courante. Finalement, il va se positionner à proximité du centre de gravité de l'ensemble des points qui correspondent à sa classe.

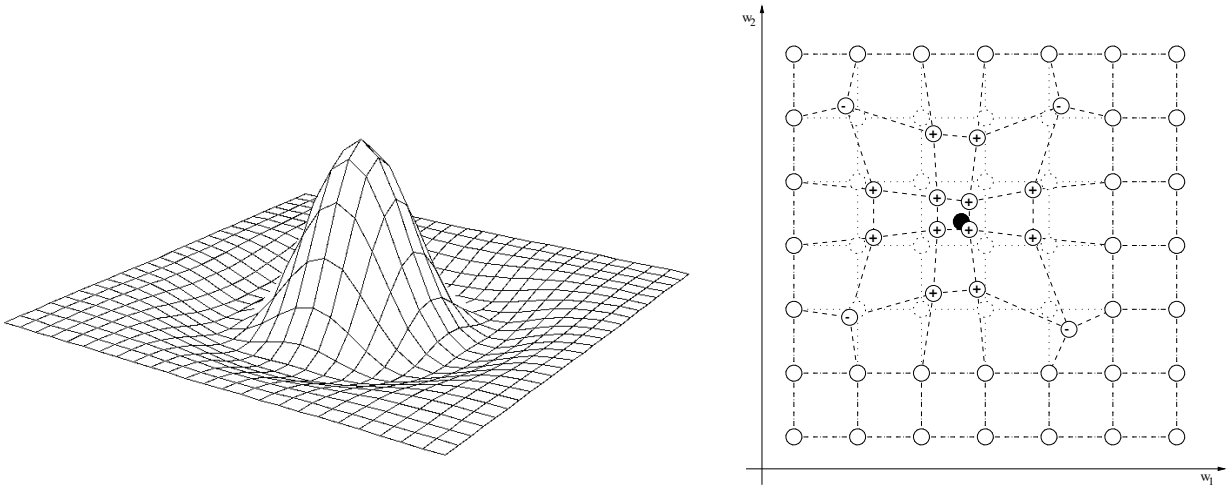
2.5.2 Cartes auto-organisatrices de Kohonen

Les réseaux de Kohonen [Kohonen, 1982] [Kohonen, 1989] peuvent être vus comme une extension des réseaux compétitifs, même si cela est chronologiquement faux et que leurs domaines d'application sont différents. Le but est d'obtenir une catégorisation des exemples avec en plus une contrainte de voisinage entre les classes qui impose que deux éléments de classes voisines sont voisins dans l'espace des données (la réciproque n'est pas toujours vraie). La topologie est définie en fonction du problème. Généralement elle se positionne dans un espace à 2 ou 3 dimensions et aura pour forme élémentaire une grille avec des cellules carrées, hexagonales ou cubiques. Les différentes topologies définissent un nombre de voisins et des relations de distance spécifiques. La distance entre les classes sur la grille est utilisée pour définir la force avec laquelle les représentants des classes sont modifiés par rapport au gagnant. De cette manière, la topologie des classes reflète la topologie des entrées. La distance peut être par exemple la distance euclidienne ou de Manhattan. Si la dimensionalité de la grille est égale à celle des entrées, la topologie est conservée de manière réciproque, même si les données sont uniformément distribuées dans tout l'espace. La dimensionalité de la grille peut être réduite si les données d'entrée sont localisées dans une sous-région de l'espace, sinon la réciproque risque de ne pas être vérifiée. Les cartes auto-organisatrices s'inspirent de l'organisation du cerveau. Kohonen s'est inspiré des travaux sur l'organisation des neurones du cortex visuel [Hubel and Wiesel, 1977] et leur modélisation [von der Malsburg, 1973][Willshaw and von der Malsburg, 1976]. On constate également une projection localisée des stimuli corporels et des contrôles moteurs sur le cortex. L'utilisation des représentations topographiques est tellement répandue qu'elle constitue manifestement un traitement de l'information pertinent. Pour défendre l'existence d'une structure comparable dans le cortex, Kohonen remarque que les inhibitions latérales entre les neurones peuvent être modélisées par des connexions inférentes entre ces neurones. Ce type de relations a la forme d'un chapeau mexicain (figure 2.29 (b)).

L'analyse de données a popularisé l'algorithme des *k-means*, l'algorithme des centres mobiles ou encore l'algorithme des nuées dynamiques. Tous ont pour critère de convergence la minimisation de l'inertie intra-classe. L'algorithme des nuées dynamiques peut être confondu avec les cartes auto-organisatrices de Kohonen si l'on remplace la phase d'optimisation du premier par la descente du gradient du second [Anouar *et al.*, 1997]. Il est néanmoins nécessaire d'ajouter un critère d'arrêt, puisque la stationnarité de la classification n'est pas rapidement applicable aux cartes de Kohonen.

2.5.3 Algorithme compétitif fair-play de DeSieno

Les algorithmes compétitifs sont aussi appelés *winner-takes-all*, cela signifie que seul le prototype vainqueur (et éventuellement son voisinage, dans le cas des cartes auto-organisatrices) est récompensé, c'est-à-dire modifié. En conséquence, il peut arriver que certains représentants ne gagnent jamais et donc ne soient jamais rapprochés des données. A l'inverse, si tous les prototypes sont rapprochés de la forme courante, ils vont rester autour du centre de gravité des données et ne pas se spécialiser véritablement sur une partie. DeSieno propose un algorithme [DeSieno, 1988] qui tient compte de ces considé-



(a) Interactions latérales autour du prototype gagnant en fonction de la distance : les prototypes proches seront excités, les lointains seront inchangés et les intermédiaires seront inhibés.

(b) Modification de la topologie initiale d'une carte de Kohonen suite à la présentation d'un exemple (•).

FIG. 2.24 – *Effet de la fonction de voisinage sur l'apprentissage.*

Algorithme 6 Algorithme des cartes auto-organisatrices de Kohonen

1. Choisir la topologie des relations entre les N classes.
2. Initialiser les vecteurs de référence $\mathbf{w_j} \in \mathbb{R}^n$ ($i \in [1,N]$) suivant $p(\mathbf{x})$, où $\mathbf{x} \in \mathbb{R}^n$ est une forme d'entrée choisie aléatoirement dans l'espace $\mathbb{R}^n$ avec la fonction de densité de probabilité de sélection $p(\mathbf{x})$..
3. Initialiser le paramètre temporel t :
t = 0.
4. Choisir aléatoirement un vecteur d'entrée $\mathbf{x}$ selon $p(\mathbf{x})$.
5. Déterminer le vecteur de référence $\mathbf{w_c}$ le plus proche de $\mathbf{x}$:
$\forall j \in [1,N] \quad \mathbf{||x - w_c|| \leq ||x - w_j||}$.
6. Modifier les vecteurs de référence $\mathbf{w_j}$ en fonction de leur distance à $\mathbf{w_c}$ sur la grille $d(i,c)$:
$\mathbf{w_i}(t+1) = \mathbf{w_i}(t) + \alpha(t) * \exp\left(-\frac{d(i,c)}{2\sigma^2(t)}\right) * (\mathbf{x} - \mathbf{w_i}(t))$.
où le coefficient d'apprentissage $\alpha(t)$ et la fonction de voisinage $\sigma(t)$ sont des fonctions f décroissantes et monotones du temps avec $0 < f(t) < 1$.
7. Incrémenter t :
t=t+1.
8. Retourner à l'étape 4, si le critère d'arrêt n'est pas validé.

rations [Rumelhart and Zipser, 1985; Rumelhart and Zipser, 1986]. L'évaluation de la distance entre une forme et un prototype considère un facteur supplémentaire basé sur

la fréquence des victoires. Il joue le rôle d'une conscience, en ce sens que les vainqueurs les plus fréquents ont un handicap. On pourrait dire, en référence aux courses de trot, qu'ils rendent une distance supplémentaire pour laisser la possibilité aux autres de gagner. De cette manière, la distribution finale des prototypes ne présentera pas une forte inégalité des probabilités de sélection. L'algorithme de DeSieno ne contient pas de notion de voisinage. Donc, la topologie des données n'est pas conservée.

Algorithme 7 Algorithme de DeSieno

1. Initialiser les vecteurs de référence $\mathbf{w_i} \in \mathbb{R}^n$ (avec $i \in [1,N]$ et N le nombre de classes) suivant $p(\mathbf{x})$, où $\mathbf{x} \in \mathbb{R}^n$ est une forme d'entrée choisie aléatoirement dans l'espace d'entrée $\mathbb{R}^n$ avec la fonction de probabilité de densité de sélection $p(\mathbf{x})$.
2. Initialiser la fréquence de victoire f_i de chaque vecteur de référence :
$f_i = 0$.
3. Initialiser le paramètre temporel t :
$t = 0$.
4. Choisir aléatoirement un vecteur d'entrée $\mathbf{x}$ selon $p(\mathbf{x})$.
5. Déterminer le vecteur de référence $\mathbf{w_c}$ le plus proche de $\mathbf{x}$:
$\forall i \in [1,N] \quad \mathbf{||x - w_c||} \leq \mathbf{||x - w_i||}$.
6. Modifier les fréquences de victoire :
$f_i(t+1) = f_i(t) + B * (y_i - f_i(t))$,
avec $y_c = 1$ et $y_{i \neq c} = 0$, et où B est une constante telle que $0 < B \ll 1$.
7. Déterminer le nouveau vecteur de référence $\mathbf{w_{nc}}$ le plus proche de $\mathbf{x}$:
$\forall i \in [1,N] \quad \mathbf{||x - w_{nc}||}^2 - b_{nc} \leq \mathbf{||x - w_i||}^2 - b_i$,
où $b_i = C * (1/N - f_i)$ et C est une constante de pondération du handicap.
8. Modifier les vecteurs de référence $\mathbf{w_i}$:
$\mathbf{w_i(t+1)} = \mathbf{w_i(t)} + A * (\mathbf{x - w_i(t)}) * z_i$,
où $z_{nc} = 1$ et $z_{i \neq nc} = 0$, et où A est le coefficient d'apprentissage.
9. Incrémenter le paramètre t :
$t = t + 1$.
10. Retourner à l'étape 4, si les conditions d'arrêt ne sont pas vérifiées.

2.5.4 *Neural gas* ou l'apprentissage de la topologie

La préservation de l'information topologique des données est souvent très utile. Or, la topologie des prototypes obtenue par les cartes auto-organisatrices est fortement contrainte. Les prototypes ont un nombre fixe de voisins et une relation fixe (carrée, hexagonale, etc.). Avec l'algorithme de *neural gas*, Martinetz [Martinetz, 1993] propose de construire la topologie à l'aide d'un apprentissage hebbien. Pour chaque exemple présenté, une connexion est créée entre le prototype champion et le vice-champion, connexion qui perdure pendant une période fonction d'un paramètre temporel. Le nombre de connexions dépend de ce paramètre puisqu'il contrôle la durée de vie des connexions.

Algorithme 8 Algorithme de *Neural gas*

1. Initialiser les vecteurs de références $\mathbf{w_i} \in \mathbb{R}^n$ (avec $i \in [1,N]$ et N le nombre de classes) suivant $p(\mathbf{x})$, où $\mathbf{x} \in \mathbb{R}^n$ est une forme d'entrée choisie aléatoirement dans l'espace d'entrée $\mathbb{R}^n$ avec la fonction de probabilité de densité de sélection $p(\mathbf{x})$.
2. Initialiser l'ensemble C des connexions à l'ensemble vide :
$C = \emptyset$.
3. Initialiser le paramètre temporel t :
$t = 0$.
4. Choisir aléatoirement un vecteur d'entrée $\mathbf{x}$ selon $p(\mathbf{x})$.
5. Déterminer le numéro d'ordre k_i $(0 \leq k_i < N)$ de chaque vecteur de référence $\mathbf{w_{i_k}}$ d'après la distance euclidienne entre $\mathbf{w_j}$ et $\mathbf{x}$:
$||\mathbf{x} - \mathbf{w_{i_0}}|| \leq ||\mathbf{x} - \mathbf{w_{i_1}}|| \leq .. \leq ||\mathbf{x} - \mathbf{w_{i_k}}|| \leq .. \leq ||\mathbf{x} - \mathbf{w_{i_{N-1}}}||, \quad j = i_k$.
6. Modifier les vecteurs de référence $\mathbf{w_i}$:
$\mathbf{w_i(t+1)} = \mathbf{w_i(t)} + \epsilon(t) * \exp\left(-\frac{k_i}{\lambda(t)}\right) * (\mathbf{x} - \mathbf{w_i(t)})$,
où le coefficient d'apprentissage est donné par la fonction $\epsilon(t)$ et où la fonction $\lambda(t)$ est une fonction décroissante monotone f qui diminue l'importance de la modification en fonction du rang (par exemple telle que $f(t) = f_{max}(f_{min}/f_{max})^{t/t_{max}}$).
7. Modifier l'ensemble C des connexions :
si $(i_0,i_1) \notin C$ alors $C = C \cup \{(i_0,i_1)\}$.
8. Incrémenter l'âge de la connexion issue de i_0 :
$age_{(i_0,i_1)} = 0$,
si $(i_0,j) \in C$ alors $age_{(i_0,j)} = age_{(i_0,j)} + 1$.
9. Supprimer les connexions trop vieilles issues de i_0
si $age_{(i_0,j)} > age_{max}(t)$ alors $C = C \setminus \{(i_0,j)\}$ la fonction $age_{max}(t)$ est aussi une fonction décroissante monotone f.
10. Incrémenter le paramètre t :
$t = t + 1$.
11. Retourner à l'étape 4, si $t < t_{max}$.

2.5.5 Réseaux auto-associatifs

Un réseau auto-associatif est caractérisé par la mise en correspondance de la même forme en entrée et en sortie d'un système. Dans ces conditions, le modèle cherche, par apprentissage supervisé, à reproduire en sortie le signal d'entrée. Si les couches intermédiaires sont de dimensions inférieures, le réseau doit réaliser une compression puis une décompression des informations contenues dans les données. Les deux opérations inverses sont accomplies par un réseau symétrique. La transformation est linéaire (figure 2.25).

Nous allons montrer qu'un réseau connexionniste, muni d'une couche cachée dont les fonctions d'activation sont linéaires, réalise une compression de l'information qui s'apparente à celle de l'analyse en composantes principales (section 2.1.1)[Oja, 1992].

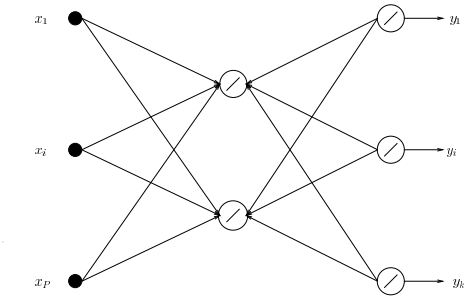


FIG. 2.25 – *L'analyse en composantes principales (ACP) par un réseau de neurones revient à utiliser un réseau comportant un nombre d'entrées et de sorties égaux au nombre de variables. Une couche intermédiaire de compression contient un nombre de neurone définissant la taille de l'espace de projection. La règle d'apprentissage de Oja fera converger les valeurs des poids reliant les entrées à un neurone intermédiaire vers un des vecteurs de projection obtenu par ACP [Oja, 1982]. La règle d'apprentissage de Sanger les donnera dans l'ordre [Sanger, 1989].*

Considérons un modèle à un seul neurone de sortie dont la fonction d'activation est linéaire. Sa valeur de sortie y est la combinaison linéaire des entrées $\mathbf{x}(t)$ par ses poids $\mathbf{w}(t)$, $y(t) = \mathbf{w}(t)^T\mathbf{x}(t) = \mathbf{x}(t)^T\mathbf{w}(t)$.

L'application de la règle d'apprentissage de Hebb a l'inconvénient de faire croître indéfiniment les poids. Cet inconvénient peut être outrepassé si les poids sont normalisés par la formule suivante :

$$\mathbf{w}(t+1) = \frac{\mathbf{w}(t) + \eta y(t)\mathbf{x}(t)}{L(\mathbf{w}(t) + \eta y(t)\mathbf{x}(t))} \tag{2.10}$$

avec $L(.)$ l'opérateur qui rend la longueur d'un vecteur et η un coefficient d'apprentissage faible. La règle de Hebb normalisée est aussi référencée comme la règle du delta pour l'apprentissage compétitif.

L'opérateur de normalisation peut être approximé par un développement de Taylor au voisinage de $\eta = 0$.

$$L(\mathbf{w}(t) + \eta y(t)\mathbf{x}(t)) = 1 + \eta \left.\frac{\partial L}{\partial \eta}\right|_{\eta=0} + O(\eta^2)$$

De plus pour η proche de 0 on a :

$$\frac{1}{1+a\eta} = 1 - a\eta + O(\eta^2)$$

d'où

$$\frac{1}{L(\mathbf{w}(t) + \eta y(t)\mathbf{x}(t))} = \frac{1}{1 + \eta \left.\frac{\partial L}{\partial \eta}\right|_{\eta=0} + O(\eta^2)} = \left(1 - \eta \left.\frac{\partial L}{\partial \eta}\right|_{\eta=0} + O(\eta^2)\right) \tag{2.11}$$

Le calcul de la dérivée donne :

$$\begin{aligned}
L(\mathbf{w}(t)+\eta y(t)\mathbf{x}(t)) &= \big((\mathbf{w}(t)+\eta y(t)\mathbf{x}(t))^T(\mathbf{w}(t)+\eta y(t)\mathbf{x}(t))\big)^{1/2} \\
&= \big(\mathbf{w}^T(t)\mathbf{w}(t)+\eta y(t)(\mathbf{w}^T(t)\mathbf{x}(t)+\mathbf{x}^T(t)\mathbf{w}(t))+\eta^2 y^2(t)\mathbf{x}^T(t)\mathbf{x}(t)\big)^{1/2} \\
&= \big(\mathbf{w}^T(t)\mathbf{w}(t)+\eta y(t)(y(t)+y(t))+\eta^2 y^2(t)\mathbf{x}^T(t)\mathbf{x}(t)\big)^{1/2} \\
&= \big(1+2\eta y^2(t)+0(\eta^2)\big)^{1/2} \\
\frac{\partial L}{\partial \eta} &= \frac{y^2(t)}{1+2\eta y^2(t)} \\
\left.\frac{\partial L}{\partial \eta}\right|_{\eta=0} &= y^2(t) \qquad (2.12)
\end{aligned}$$

Finalement, en regroupant les résultats (2.10), (2.11) et (2.12), on a :

$$\begin{aligned}
\mathbf{w}(t+1) &= \mathbf{w}(t)+\eta y(t)\mathbf{x}(t)(1-\eta y^2(t)+O(\eta^2)) \\
&= \mathbf{w}(t)+\eta y(t)\mathbf{x}(t)-\eta y^2(t)\mathbf{w}(t)-\eta^2 y^3(t)\mathbf{x}(t)+O(\eta^2) \\
\mathbf{w}(t+1) &= \mathbf{w}(t)+\eta y(t)(\mathbf{x}(t)-y(t)\mathbf{w}(t))+O(\eta^2) \qquad (2.13)
\end{aligned}$$

La dernière formule est la règle de Oja [Oja, 1982]. Sa première partie correspond à la règle de Hebb, c'est donc la seconde qui assure la normalisation.

Nous allons voir maintenant que cette règle peut extraire les composantes principales.

Soient N variables X_1,X_2,...,X_N dont les réalisations x_1,x_2,...,x_N sont regroupées dans un vecteur $\mathbf{x}(t)$. Supposons que les variables X_i soient centrées et notons $\mathbf{R}$ la matrice de corrélation de l'échantillon.
L'espérance des poids pour la règle de Oja vaut :

$$\begin{aligned}
E(\mathbf{w}(t+1)|\mathbf{w}(t)) &= E(\mathbf{w}(t))+E(\eta y(t)(\mathbf{x}(t)-y(t)\mathbf{w}(t))) \\
&= \mathbf{w}(t)+\eta E(y(t)\mathbf{x}(t)-y^2(t)\mathbf{w}(t)) \\
&= \mathbf{w}(t)+\eta E(y(t)\mathbf{x}(t))-\eta E(y^2(t)\mathbf{w}(t)) \\
&= \mathbf{w}(t)+\eta E(\mathbf{x}^T(t)\mathbf{w}(t)\mathbf{x}(t))-\eta E((\mathbf{w}^T(t)\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{w}(t))^2\mathbf{w}(t)) \\
&= \mathbf{w}(t)+\eta\mathbf{R}\mathbf{w}(t)-\eta\mathbf{w}^T(t)\mathbf{R}\mathbf{w}(t))\mathbf{w}(t)
\end{aligned}$$

On peut déduire de sa version continue le théorème suivant :

Théorème 4
Soient $\mathbf{e}_i$ *les vecteurs propres de la matrice de corrélation* $\mathbf{R}$ *associés aux valeurs propres* λ_i *classées de manière décroissante telle que* $\lambda_1 > \lambda_2 > ... > \lambda_N$. *L'équation*

$$\frac{d}{dt}\mathbf{w}(t) = \mathbf{R}\mathbf{w}(t) - (\mathbf{w}^T(t)\mathbf{R}\mathbf{w}(t))\mathbf{w}(t)$$

fait converger $\mathbf{w}$ *vers* $\pm\mathbf{e}_1$

La démonstration ci-dessus prouve que les poids d'un seul neurone de sortie convergent vers le vecteur propre de la matrice de corrélation qui a la plus grande valeur propre. Le vecteur de poids est orienté dans le sens de la plus grande variance des données. Pour trouver les autres vecteurs propres, décomposons le vecteur d'entrée dans la base des vecteurs propres :

$$\mathbf{x} = \sum_{i}^{N} \alpha_i \mathbf{e}_i$$

Si maintenant nous considérons le nouveau vecteur

$$\tilde{\mathbf{x}} = \mathbf{x} - \alpha_1 \mathbf{e}_1$$

le coefficient α_1 de $\tilde{\mathbf{x}}$ vaut zéro. Donc si nous appliquons la même procédure à ce vecteur avec un nouveau neurone de sortie, ses poids vont prendre la direction du vecteur propre de la seconde plus grande valeur propre i.e. celle dont la variance est la plus grande après $\mathbf{e}_1$, et ainsi de suite. Nous pouvons écrire :

$$y = \mathbf{w}^T \mathbf{x} = \mathbf{e}_1^T \sum_{i}^{N} \alpha_i \mathbf{e}_i = \alpha_1$$

et puisque $\mathbf{w} = \mathbf{e}_1$, on a :

$$\tilde{\mathbf{x}} = \mathbf{x} - y\mathbf{w}$$

L'application de cette transformation aux données permet, en appliquant à nouveau la règle de Oja sur les nouvelles valeurs, d'obtenir une estimation du second vecteur propre. L'ensemble des vecteurs propres peut être ainsi déterminé de manière itérative.

Des méthodes plus élaborées se servent de connexions latérales entre les neurones de la couche cachée pour orthogonaliser les poids des neurones et obtenir la convergence simultanée des poids vers les vecteurs propres. De plus, le formalisme des réseaux connexionnistes permet de réaliser facilement une généralisation de l'analyse en composantes principales, en ajoutant deux couches cachées non linéaires avant et après la couche cachée linéaire (figure 2.26). La compression ainsi obtenue sera non linéaire.

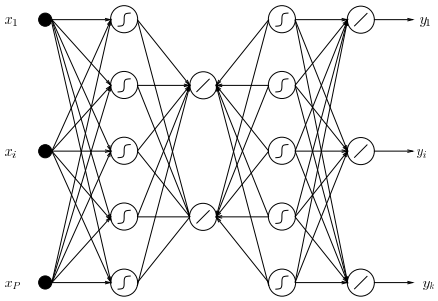
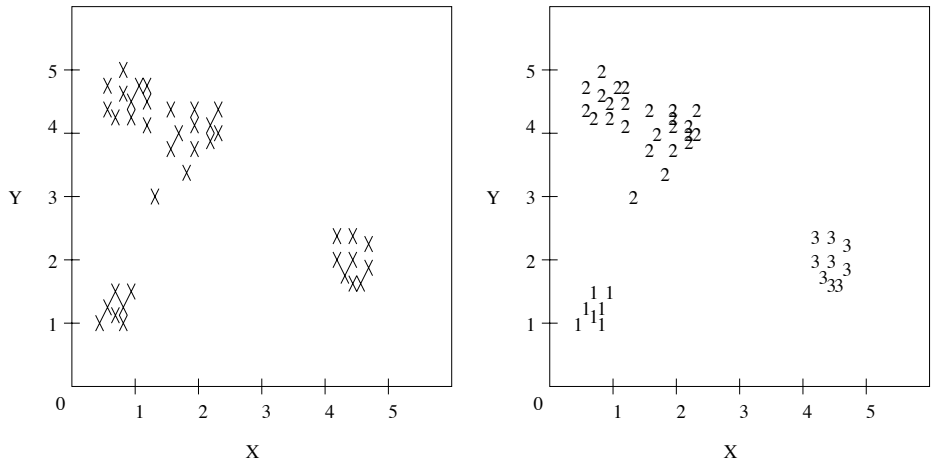


FIG. 2.26 – *Une analyse en composantes principales non linéaire peut être effectuée par un réseau de neurones sur le même principe que celui d'une analyse linéaire (voir figure 2.25) en insérant une couche de neurones à fonction non linéaire avant et après la couche de projection.*

Les parties suivantes présentent de quelle manière des problèmes comme le partitionnement des données, la quantification vectorielle de l'espace des données, la diminution de l'espace des données et l'extraction de caractéristiques peuvent être traités par une approche connexionniste distribuée. Elles expliquent comment, sans apprentissage supervisé, un système de compétition peut aboutir à la réalisation de traitements élaborés, et surtout quelles connaissances sont utilisées et générées par le modèle pour y parvenir.

2.5.6 Catégorisation

La catégorisation a déjà fait l'objet d'une définition dans la section 2.1.2. A cette occasion, nous avons abordé les enjeux et les stratégies utilisées par des méthodes statistiques. Rappelons simplement que les données peuvent ainsi être regroupées par similitude en classes homogènes. Tous les algorithmes compétitifs produisent un regroupement inhérent aux données. L'information utile pour le partitionnement est le neurone gagnant, puisqu'il est associé à la classe à laquelle appartient la donnée courante (figure 2.27).



(a) Répartition des données. (b) Catégorisation en trois classes.

FIG. 2.27 – *Schématisation d'une catégorisation.*

L'apprentissage non supervisé est basé sur un critère de compétition entre les unités de sortie et sur une notion de distance. Un réseau compétitif qui utilise une distance euclidienne entre les formes d'entrée et les poids des unités de sortie réalise un partitionnement des données d'entrée identique à l'algorithme des *k-means* [Hartigan, 1975]. Les réseaux compétitifs s'adaptent facilement aux nombreuses variantes des *k-means*, comme sa version axiale, et aux nuées dynamiques. De plus, l'information topologique peut être utile pour établir une continuité entre les classes, et, par extension, pour mélanger les prédictions de plusieurs modèles voisins.

Expérimentation : Actuellement, il n'existe pas de modèle théorique général pour expliquer la complexité du phénomène de propagation d'un signal radioélectrique. A l'inverse, un modèle statistique peut facilement être mis en œuvre pour estimer les contextes

géographiques, à partir d'exemples, et obtenir de bonnes performances si les données sont homogènes. Dans ce but, nous proposons de partitionner l'environnement pour parvenir à des classes de données suffisamment homogènes afin de produire par la suite (cf. section 3.2.1) un modèle prédictif de l'atténuation du champ radioélectrique sur chaque classe. Nous venons de décrire un ensemble d'algorithmes non supervisés qui peuvent réaliser une telle tâche. Le grand nombre de techniques existantes indique qu'il n'y a pas de méthode donnant des résultats optimaux qui soit valable pour tous les problèmes. Parmi les techniques présentées, certaines conservent l'information topologique, d'autres contrôlent la probabilité de sélection d'une classe. Nous allons donc utiliser les cartes auto-organisatrices de Kohonen, l'algorithme de DeSieno et le réseau *neural gas* pour partitionner les données pour, par la suite, spécialiser sur chacune de ces partitions un modèle de régression tel que le perceptron multicouches dans le but d'améliorer la prédiction de l'atténuation du champ radioélectrique.
Puisque les lois changent avec l'environnement, nous allons utiliser le corpus géographique, qui contient une description de l'environnement traversé par le signal, pour expérimenter les algorithmes non supervisés. La répartition des données a été faite en 16 classes pour obtenir une grille carrée sur la carte auto-organisatrice de Kohonen et permettre l'existance de vecteurs de référence pour des petits groupes de données parfois isolés.

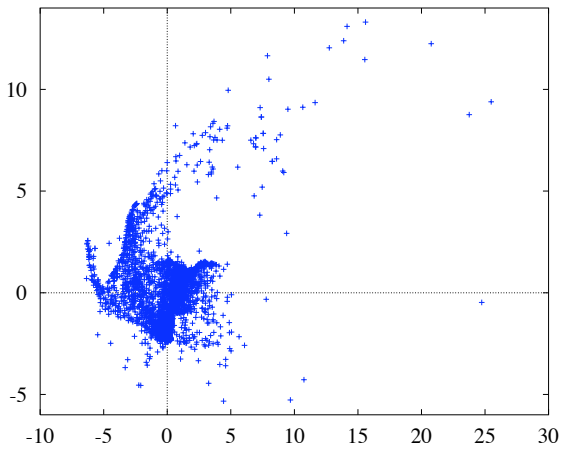


FIG. 2.28 – *Données géographiques projetées.*

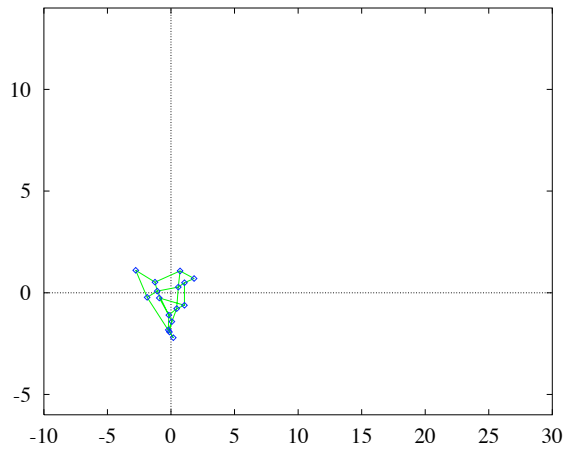


FIG. 2.29 – *Carte auto-organisatrice projetée.*

En projetant, par l'algorithme de Sammon[6] [Sammon, 1969], l'ensemble des données et les vecteurs de référence de chaque méthode dans un espace à deux dimensions, nous pouvons visualiser le positionnement des uns par rapport aux autres pour comprendre leur stratégie et leurs spécificités. La figure 2.28 présente la répartition des données après projection par l'algorithme de Sammon. Certaines zones de forte densité peuvent ne pas apparaître si beaucoup de données ont une description identique, puisqu'elles seront toutes

6. L'algorithme de Sammon tente de préserver la distance inter-points après projection dans un espace à 2 ou 3 dimensions. C'est une alternative à l'analyse en composantes principales. Les valeurs sur les axes de projection n'ont pas de signification particulière. La méthode s'affranchit des contraintes d'orthogonalité de l'ACP pour permettre une meilleure visualisation de la répartition des données.

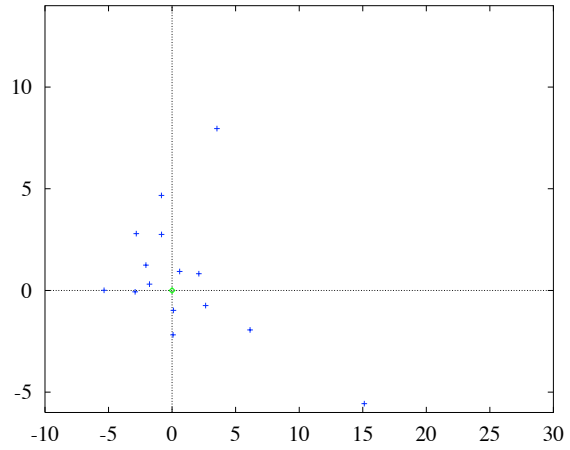


FIG. 2.30 – *Prototypes de DeSieno projetés.*

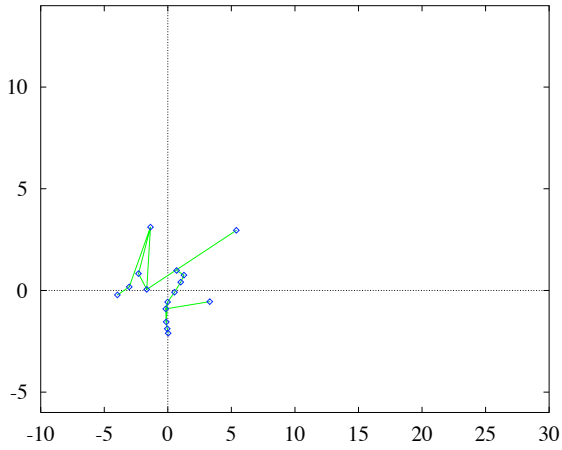


FIG. 2.31 – *Réseau de* neural gas *projeté.*

représentées par un unique point. Pour notre application, 22% des données sont redondantes et n'apparaîtront donc pas sur la figure. Le reste des figures représente les vecteurs de référence par des points et leurs relations topologiques par des liens lorsque l'information existe. La figure 2.29 montre que les vecteurs de référence obtenus par une carte auto-organisatrice sont concentrés autour du centre de gravité des données. La définition de la topologie *a priori* contraint la liberté de mouvement des vecteurs. Ils caractérisent donc tous plus ou moins la même zone, en l'occurrence celle située autour du centre de gravité des données. Nous pourrions diminuer la contrainte de voisinage mais sur une grille de petite taille cela aura pour conséquence de ne plus avoir de voisinage. La relation topologique entre les neurones n'aurait alors plus de véritable signification. La figure 2.30 montre que la contrainte de l'algorithme de DeSieno, qui tend vers une équiprobabilité de sélection des neurones, a permis aux vecteurs de référence de mieux se disperser. Cette dispersion peut être contrôlée par la valeur du coefficient C qui, s'il est réduit à zéro, ne contraint pas les neurones les plus victorieux et confond l'algorithme de DeSieno avec celui de Kohonen, sans information topologique. Le figure 2.31 rapportant les résultats de l'algorithme de *neural gas* montre ce qu'un algorithme compétitif classique obtiendrait avec en plus une information topologique construite par apprentissage hebbien. En effet, bien que tous les neurones soient modifiés à chaque itération, la fonction qui pondère l'ampleur de la modification des poids est fortement décroissante et s'apparente à un *leaky learning*, c'est-à-dire que les vecteurs éloignés sont déplacés pour ne pas tomber dans l'oubli. On peut donc considérer que seul le neurone gagnant est véritablement modifié. L'intérêt particulier de *neural gas* est l'apprentissage de la topologie des entrées par les vecteurs de référence. Tous les neurones sont connectés à au moins un neurone, ce qui prouve qu'ils ont tous été récemment (i.e. depuis moins de age_{max} d'itérations) premier ou second d'une compétition. Un neurone sans connexion signifierait qu'il caractérise une zone peu peuplée, voire déserte en début d'apprentissage. Pour conclure l'analyse des catégorisations, nous ferons observer que bien que le critère de la variance intra-classe soit minimisé dans le cas d'un nombre de catégories égal au nombre d'individus, en pratique le nombre de catégories ne doit pas être élevé pour que la catégorisation soit significative. En conséquence du nombre réduit de catégories, tous les neurones de la carte sont de proches

voisins. Ils sont déplacés un peu vers chaque forme et finissent proches les uns des autres. Cette relation, on pourrait presque dire cet enchaînement, peut aboutir à positionner une catégorie dans une zone exempte de données parce qu'elle se trouve au milieu de deux zones très représentées.

Du point de vue de l'application, nous nous sommes intéressés aux algorithmes compétitifs parce qu'ils permettent de définir des groupes de données plus homogènes, dans le but de faciliter l'adaptation d'un modèle de prédiction de l'atténuation à un sous-groupe de données. Les catégories obtenues par la carte auto-organisée ne caractérisent pas bien les zones de faible densité. Le choix de la dimension de la carte est crucial pour obtenir une bonne représentation. Mais dans les problèmes réels de dimension élevée, la répartition des données est généralement trop complexe pour être appréhendée par une relation de petite dimension entre les neurones. Fortement contraint, l'algorithme de DeSieno définit des catégories qui ont un nombre relativement proche d'éléments et non, comme nous le voudrions, des éléments relativement similaires. Faiblement contraints, les neurones perdent leur fair-play et retrouvent leurs instincts combatifs intrinsèques aux algorithmes compétitifs. Nous disposons dans ce cas des mêmes catégories que celles de la carte de Kohonen. Le réseau de *neural gas* est plus libre. Il n'est pas contraint par la topologie. Finalement, les observations faites sur la visualisation de la répartition des données et des vecteurs de référence ne peuvent suffire à déterminer la catégorisation optimale. Certains critères, comme l'inertie intra-classe, peuvent rendre compte de l'homogénéité des classes et donc donner une idée de la meilleure catégorisation. Seule l'application d'un modèle de prédiction de l'atténuation spécialisé par catégorie nous donnera une estimation de l'intérêt de chaque méthode compétitive.

Les détails de ces études peuvent être consultés dans divers articles [Bougrain and Alexandre, 1998; Bougrain and Alexandre, 1999b; Bougrain, 1997a].

2.5.7 Quantification vectorielle

Un système de quantification vectorielle divise l'espace d'entrée en sous-espaces disjoints. Il fournit une représentation discrétisée de l'espace d'entrée à partir des données. La fonction ainsi modélisée prend en entrée un élément de l'espace et rend l'identification du sous-espace auquel il appartient. La quantification vectorielle se différencie de la catégorisation en ce sens qu'elle ne recherche pas les similitudes entre les formes d'entrée mais la répartition des données dans l'espace entier. Le but est d'obtenir une estimation de la fonction de densité de chaque sous-espace. La probabilité est plus forte dans les régions où de nombreuses données sont présentes. L'apprentissage compétitif intervient au niveau de la recherche des zones les plus peuplées. Les vecteurs de quantification vont se positionner dans les zones de plus forte densité, c'est-à-dire dans les régions où le plus d'individus sont apparus. De cette manière, la quantification vectorielle peut permettre d'effectuer une compression de l'information pour le stockage.

Expérimentation : La quantification vectorielle ne diffère du partitionnement que dans l'interprétation des résultats des algorithmes compétitifs (voir figures 2.27 et 2.33). Les

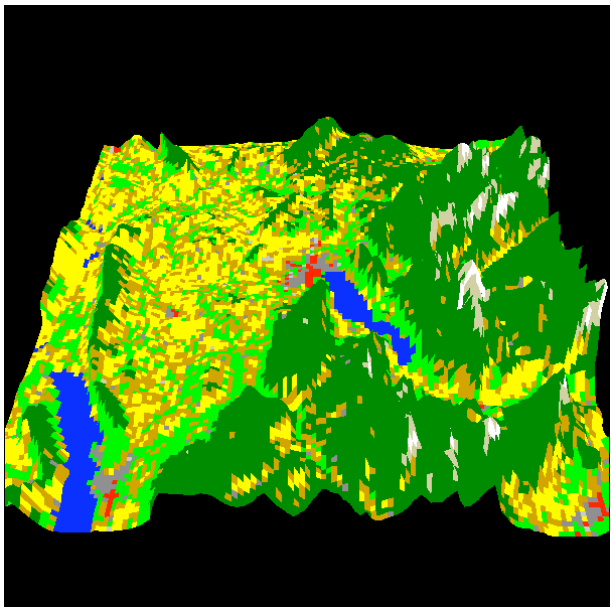
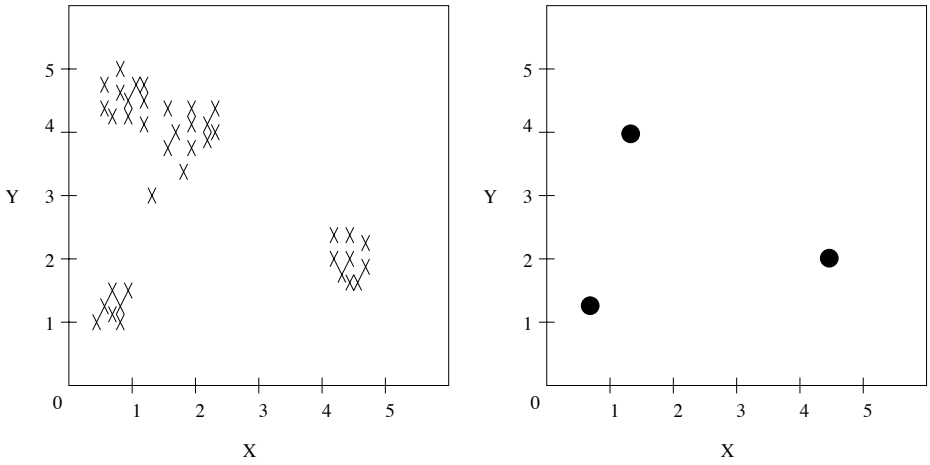


FIG. 2.32 – *Exemple de classification automatique des données géographiques de Annecy par une carte auto-organisatrice de Kohonen comportant 16 prototypes (chaque classe est représentée par une couleur différente). Le réseau réalise une classification cohérente de la géographie.*



(a) Répartition des données.

(b) Quantification vectorielle sur trois vecteurs.

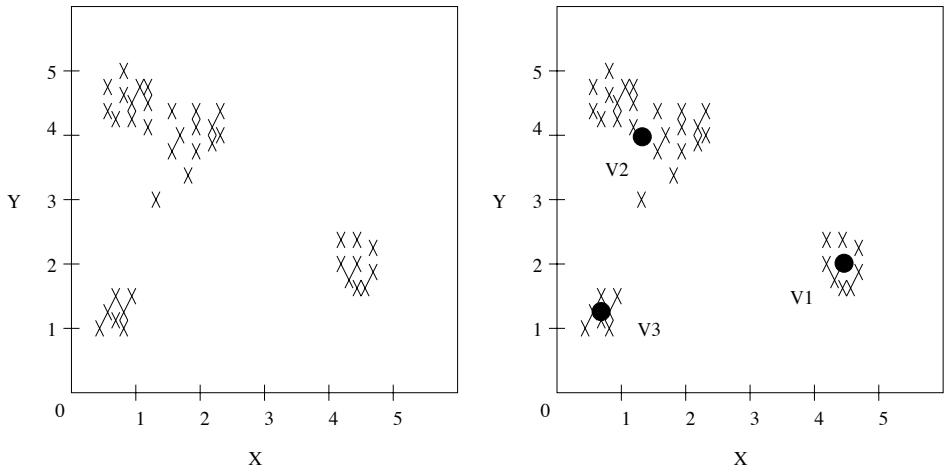
FIG. 2.33 – *Schématisation d'une quantification vectorielle.*

résultats obtenus par les différents algorithmes compétitifs présentés sur le problème d'atténuation du champ radioélectrique ont été visualisés page 81. Les vecteurs nous montrent la répartition des données dans un espace projeté. L'algorithme de DeSieno nous donne une meilleure information sur la répartition de ces données dans l'espace d'origine car chaque vecteur représente environ le même nombre d'exemples, ce qui compense une par-

tie de la perte d'information due à la projection.

2.5.8 Extraction de caractéristiques

L'extraction de caractéristiques est liée à la réduction de la dimension du problème. Dans la section suivante, nous rappellerons les conséquences d'une réduction du nombre de variables et les différentes manières d'y parvenir à l'aide de réseaux de neurones. Dans cette section, nous allons évoquer l'extraction de caractéristiques obtenue par les méthodes compétitives suite à la création d'un petit nombre de prototypes. L'utilisation des vecteurs de référence revient à diminuer le nombre d'individus en ne gardant que les individus les plus représentatifs, ou plus exactement en créant, pour chaque groupe d'individus, un individu caractéristique (figure 2.34). Donc, ici, on s'intéresse à la valeur des vecteurs de référence. Les méthodes compétitives extraient les caractéristiques communes à des groupes de données. Les cartes de Kohonen n'ont pas vraiment de méthodes statistiques équivalentes. Elles produisent une réduction de la dimension d'entrée (celle des individus) de manière non linéaire. Une information topologique vient s'ajouter à l'extraction de caractéristiques réalisée par l'apprentissage compétitif. Les cartes de Kohonen constituent une extension de l'algorithme des *k-means* et peuvent être adaptées aux nuées dynamiques.



(a) Répartition des données.

(b) Extraction des caractéristiques en trois groupes : v1=(4.5,2), v2=(1.5,4), v3=(0.7,1.3).

FIG. 2.34 – *Schématisation d'une extraction de caractéristiques.*

Expérimentation : L'extraction de caractéristiques ne diffère des deux précédentes méthodes que dans l'interprétation des résultats des algorithmes compétitifs (voir figures 2.27, 2.33 et 2.34). L'analyse des valeurs prises par les composantes des différents vecteurs de référence nous décrit les caractéristiques centrales des différents groupes et donc de l'ensemble des données.

2.5.9 Réduction de la dimensionalité

Les réseaux présentés dans les sections précédentes peuvent être vus comme des transformations vectorielles non linéaires. Ils font correspondre à un vecteur d'entrée un vecteur de sortie binaire indiquant le neurone vainqueur. Les modifications imposées aux poids sont telles que les vecteurs de référence peuvent être considérés comme des vecteurs prototypes (i.e. des vecteurs moyens) pour les vecteurs d'entrée pour lesquels ils sont gagnants. Les transformations auto-organisées présentées dans cette section font subir à l'espace d'entrée une rotation pour que les vecteurs de sortie soient aussi décorrélés que possible et que l'énergie (i.e. les variances des exemples) soit principalement concentrée dans un minimum de neurones de sortie. Après la rotation, la variance des variances a augmenté. Certaines dimensions représentent mieux la variance des données au détriment des autres. La variance des données est maximisée sur un nombre plus petit de dimension. Cette transformation est proche d'une transformation par vecteur propre qui, dans le cas d'un traitement d'images, code l'image dans une dimension inférieure pour la recomposer par la suite.
L'analyse en composantes principales et les réseaux auto-associatifs peuvent être utilisés pour associer à des données d'un espace continu de dimension élevée leur représentant dans un espace continu de dimension inférieure, tandis que les cartes de Kohonen associent à des données d'un espace continu leur représentant dans un espace discret.

Conclusion

Les relations entre les réseaux connexionnistes et les statistiques sont plus étroites que la différence de terminologie ne le laisse penser. Ainsi, ce qu'en réseaux neuromimétiques nous appelons « apprendre à généraliser à partir d'un ensemble d'exemples bruités » n'est autre que de l'inférence statistique. Bien sûr, certains modèles neuromimétiques ont peu, ou pas du tout, de points communs avec les statistiques, mais la plupart des modèles connexionnistes sont une classe particulière des modèles de discrimination et de régression. Les statisticiens ont étudié les propriétés de cette classe [Bishop, 1995; Ripley, 1996]. L'application de la théorie statistique aux réseaux connexionnistes est abordée dans le détail dans [Bishop, 1995; Ripley, 1996]. De nombreux résultats théoriques obtenus sur les modèles statistiques peuvent être appliqués directement aux réseaux. Les réseaux de neurones sans couche cachée sont des modèles linéaires généralisés. Les réseaux de neurones feedforward à une couche cachée sont étroitement liés à la régression. Par exemple, le perceptron multicouches s'applique aux mêmes situations que la poursuite en projection en régression, c'est-à-dire quand la dimension des données est élevée, et qu'une grande partie des informations réellement utiles à la prédiction se trouve contenue dans un sous-espace de dimension faible. Le calcul des valeurs de prédiction est plus simple et plus rapide dans le deuxième cas. De plus, la technique neuronale s'adapte mieux à des fonctions problématiques comme les fonctions discontinues. On trouve plusieurs avantages aux réseaux connexionnistes qu'il nous semble intéressant d'exploiter et de développer. Les arguments souvent avancés se basent sur la non-linéarité et la facilité d'utilisation des méthodes neuronales. Des méthodes d'optimisation utilisées pour les

modèles d'ajustement non linéaires, telles que les algorithmes de Levenberg-Marquardt et du gradient conjugué, peuvent être directement appliquées aux réseaux pour accélérer la convergence. Nous avons également vu que les méthodes neuronales de quantification vectorielle sont similaires aux méthodes de classification automatique de type *k-means*. Les cartes auto-organisatrices de Kohonen à une dimension sont des approximateurs discrets de courbes (les cartes à deux dimensions approximent des surfaces principales). De nombreux autres algorithmes neuronaux tels que l'algorithme de DeSieno et l'algorithme de *neural gas* permettent également d'obtenir de manière non supervisée une représentation des données. L'apprentissage hebbien s'apparente à l'analyse en composantes principales. Il permet, grâce aux corrélations qui existent entre les variables, de sélectionner des variables pertinentes et de concentrer l'information dans un nombre restreint de variables. Nous avons aussi montré que, de toutes les méthodes abordées, il est possible d'extraire de la connaissance.

Les réseaux connexionnistes sont plus pensés en termes de résultats qu'en termes de moyens. Pour combler cette lacune et combattre la bête noire des réseaux de neurones (i.e. la boîte noire), il faut essayer de comprendre leur fonctionnement à un niveau moins localisé. Les liens entre certaines méthodes statistiques et les réseaux connexionnistes étant établis, il devient évident que la théorie dans le domaine des statistiques apporte de précieuses informations en termes de méthodologie (hypothèses, optimisation, intervalles de confiance, visualisation) et d'interprétation. Les réseaux connexionnistes présentés dans ce chapitre sont des méthodes classiques. La compréhension précise de leur fonctionnemnt va constituée une base solide pour étudier des modèles plus complexes et développer un modèle original contextuel et interprétable. Le chapitre suivant aborde la façon dont certains réseaux connexionnistes déterminent leur propre architecture et le procédé de modélisation qu'ils utilisent en fonction du but à atteindre.

Chapitre 3

Structuration par construction

Le chapitre précédent expose les connaissances que les grandes familles de réseaux de neurones sont capables d'acquérir implicitement. Pour ces réseaux qui obtiennent une représentation interne du monde par adaptation, l'espace de liberté est contraint par une architecture fixe et une combinaison des poids relativement simple.

Pour passer outre ces restrictions, d'autres réseaux de neurones peuvent être mis en place. Ces modèles sont caractérisés par une organisation singulière de leur architecture. Celle-ci se structure en fonction de la tâche à modéliser, on dira qu'elle s'organise ou se construit. Pour prendre une image, la fonction à modéliser n'est plus un corps pour lequel on cherche le bon vêtement (comprenez le modèle) auquel on retouche les ourlets (ce sont les paramètres du modèle) mais un corps pour lequel on confectionne un vêtement sur mesure.

En début de chapitre, nous discuterons de deux approches neuronales différentes qui présentent cette particularité. Puis nous montrerons que les avantages de ces deux approches peuvent être unifiés à travers l'introduction d'un nouveau modèle.

La première des deux approches qui structure la connaissance par construction est constituée de modèles dynamiques. Pour trouver une fonction qui modélise la tâche, l'espace de liberté des modèles classiques peut être élargi. Deux grandes stratégies permettent d'ajuster librement le modèle à la fonction : la méthode expansive qui consiste à « tricoter » une structure jusqu'à l'obtention du bon modèle, et la méthode réductrice qui consiste à « patronner » une structure à partir d'un grand modèle. Nous présenterons les réseaux dynamiques, non pas en procédant à un état de l'art énumératif qui peut être trouvé dans de nombreux ouvrages tels que [Kwok and Yeung, 1997], mais en présentant les principaux mécanismes de construction sur lesquels se développe la grande majorité des modèles dynamiques.

La seconde approche est constituée par des réseaux modulaires. Nous nous intéresserons à la façon dont une tâche peut être répartie entre différents modules et à la manière dont ils se coordonnent pour collaborer efficacement. Pour revenir à notre parallèle avec le monde de la confection, nous pourrions dire qu'ici le modèle n'est pas une combinaison intégrale mais l'association de plusieurs éléments (chemise, pantalon, ceinture, etc.). Nous

examinerons successivement quelles peuvent être les raisons de la décomposition d'un système en modules, les méthodes de décomposition et les différents types de relation qui peuvent exister entre les modules. Dans ce type de réseaux, le travail est plus organisé et la connaissance plus structurée.

Finalement, pour parfaire notre étude, nous reprendrons les avantages des différentes approches dans un modèle original qui présentera une composante dynamique pour obtenir l'architecture la plus efficace, une composante contextuelle pour mieux s'adapter à la situation courante et une composante explicative pour fournir des informations sur le problème.

L'étude des modèles nous apprendra leur fonctionnement, et nous continuerons à utiliser notre application pour enrichir notre propos en montrant concrètement quelles sont leurs capacités, c'est-à-dire quelles sont leurs performances en classification ou prédiction, sous quelle forme la connaissance est structurée et donc comment l'interpréter. Nous présenterons de plus les résultats obtenus par l'implantation de notre modèle sur machine parallèle, en termes de fonctionnement, de performances et d'explication.

3.1 Construction dynamique

La question la plus fréquente dans le domaine des réseaux connexionnistes est : « combien d'unités cachées dois-je utiliser ? ». Si la réponse est si recherchée, c'est parce que l'architecture d'un réseau, c'est-à-dire le nombre d'unités et la façon dont elles sont connectées, a une influence importante sur les performances. En effet, le nombre de fonctions modélisables par un réseau de neurones dépend du nombre de paramètres libres, c'est-à-dire du nombre de poids synaptiques, dont il dispose. Et le nombre de poids dépend directement de la taille de l'architecture. Mais si la question est si fréquente, c'est surtout parce qu'il n'existe pas de réponse précise. Il y a bien des pseudo-lois du style « Jacques a dit : mettez un nombre d'unités cachées compris entre le nombre d'unités d'entrée et le nombre d'unités de sortie », mais ces lois empiriques ne sont pas très fiables (quand elles ne sont pas fausses).

Le problème de la modélisation d'une tâche revient à chercher dans l'espace des fonctions celle qui constitue un bon approximateur de la fonction cible. La recherche n'étant pas réalisable dans tout l'espace, elle s'effectue dans un sous-espace restreint par un ensemble de contraintes appliquées au modèle. Généralement, l'architecture fixe du modèle constitue la contrainte la plus importante. Mais si cette contrainte réduit fortement l'espace de recherche, elle réduit également la possibilité de trouver une solution optimale. La difficulté de la recherche n'est pas exactement liée à la grande taille de l'espace des fonctions mais à la taille de l'espace des fonctions modélisables par le réseau, c'est-à-dire l'espace à explorer. Donc, si la recherche est plus efficace, elle peut être appliquée à un espace plus grand. La recherche doit s'organiser à partir d'une position de départ déterminée par les conditions initiales, du but à atteindre spécifié sous la forme de critères d'arrêt, et d'une stratégie pour passer de l'une à l'autre. Dans le cadre des modèles dynamiques, l'espace de recherche est élargi par suppression de la contrainte de taille appliquée

généralement à l'architecture du modèle. De nouveaux algorithmes tentent d'augmenter l'efficacité de la recherche dans cet espace. La recherche s'effectue en deux parties : la recherche de l'architecture optimale et la recherche des paramètres optimaux.

Donc, pour être certain de pouvoir modéliser la fonction recherchée, on est tenté d'utiliser une grande architecture. Mais ce choix engendre un accroissement du temps d'apprentissage et d'utilisation, un nombre plus important d'exemples pour ajuster correctement les nombreux paramètres, des perturbations dues aux variables d'entrée inutiles et une difficulté d'interprétation. Pour pallier ces problèmes, deux approches dynamiques peuvent être utilisées. La première consiste à partir d'un réseau faiblement dimensionné et à augmenter sa capacité par l'ajout d'unités. La seconde approche consiste à partir d'un réseau surdimensionné capable de modéliser des fonctions complexes et à le simplifier par la suppression de connexions peu utiles voire d'unités. Finalement, un réseau dynamique est un réseau ayant une plasticité architecturale en plus d'une plasticité synaptique,

Pour bien comprendre l'intérêt des réseaux dynamiques, il est nécessaire de reconsidérer brièvement les conditions de l'apprentissage et ses effets en termes de généralisation et plus particulièrement de surapprentissage.

3.1.1 Capacité de généralisation et surapprentissage

Le but d'un modèle est de rendre compte d'un phénomène sous la forme d'une fonction f. La fonction et ses paramètres $\mathbf{w}$ ayant été déterminés, la qualité d'un modèle se mesure en comparant, pour chaque situation $\mathbf{x}$, sa réponse $f(\mathbf{x},\mathbf{w})$ à la réponse que l'on voudrait qu'il ait $\mathbf{d}$. En théorie, l'erreur E est évaluée sur l'ensemble des situations, en accord avec leur fonction de densité de distribution p et d'après une fonction de coût locale e : $E = \int_{\mathbf{x}} e(f(\mathbf{x},\mathbf{w}),\mathbf{d})p(\mathbf{x},\mathbf{d})d\mathbf{x}$. L'erreur réelle correspond alors à l'espérance de la performance du modèle sur une situation quelconque. Mais la fonction de densité de probabilité n'étant généralement pas connue, il faut évaluer la qualité du modèle à partir de l'erreur produite par un ensemble fini d'exemples $\{\mathbf{x}_n,\mathbf{d}_n\}^N$ dont les caractéristiques sont connues à un bruit près. L'erreur ainsi calculée est une erreur empirique basée sur un ensemble d'apprentissage : $E_{emp} = \sum_{i=1}^{N} e(f(\mathbf{x}_i,\mathbf{w}),\mathbf{d}_i)$. Elle est à distinguer de l'erreur réelle du modèle, aussi appelée erreur en généralisation. Le but est de faire que l'erreur empirique soit un bon approximateur de l'erreur réelle. Mais une erreur empirique faible ne garantit pas toujours une erreur en généralisation faible. Le nombre fini d'exemples pose le problème de la spécialisation du modèle sur les exemples lorsqu'elle conduit à un apprentissage par cœur qui ne présente aucun intérêt en généralisation. Dans ce cas, on parle de surapprentissage.

Le surapprentissage se produit lorsque le nombre de paramètres du modèle est trop élevé par rapport au nombre d'exemples disponibles. La théorie de l'apprentissage et les mathématiques nous fournissent des outils pour mieux appréhender ce phénomène. D'un point de vue mathématique, l'approximation de fonctions correspond à une tentative de rendre compte (d'approximation) d'une fonction complexe connue par une fonction plus simple dans un but computationnel. L'interpolation désigne quant à elle la recherche d'une

fonction qui passe par un nombre de points précis dans le but de fournir des estimations pour de nouvelles situations situées à l'intérieur du domaine couvert par les exemples. L'extrapolation, comme son nom l'indique, cherche à estimer la valeur de points situés en dehors de cet intervalle. Ainsi, ce qu'on entend dans le domaine des réseaux de neurones par approximation de fonctions correspond, en mathématiques, à l'interpolation voire à l'extrapolation. Les familles de fonctions les plus utilisées pour l'interpolation, en mathématiques, sont les fonctions polynômiales, les fonctions rationnelles (quotients de polynômes) et les fonctions trigonométriques. Les méthodes neuronales utilisent généralement quant à elles, des fonctions linéaires, sigmoïdales ou encore gaussiennes. Dans le cas des fonctions polynômiales, les mathématiques nous donnent des théorèmes pour calculer la fonction exacte qui passe par un nombre précis de points. Il est connu que par deux points il passe une seule droite. Par trois points, une seule quadratique etc... Par un nombre N de points, il passe une unique fonction polynômiale de degré $N-1$. Les observations faites sur l'utilisation de fonctions polynômiales pour modéliser un phénomène, au même titre que les réseaux de neurones, va nous permettre de comprendre l'influence du nombre d'exemples sur le degré de liberté, c'est-à-dire du nombre de paramètres d'un modèle. Plus le nombre de paramètres est élevé, plus le modèle est complexe. Par conséquent, plus il est à même de reproduire les situations qui lui sont présentées en apprentissage.

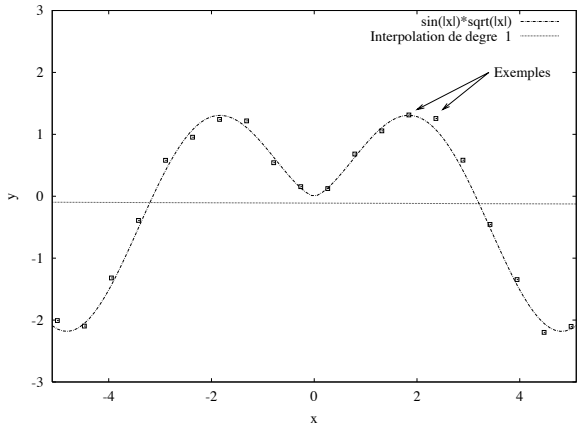


FIG. 3.1 – *Approximation polynômiale de degré 1 de la fonction* $\sqrt{|x|}\sin(|x|)$ *à partir de réalisations bruitées.*

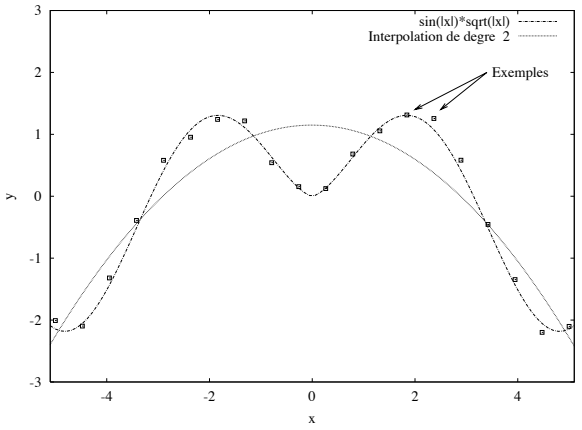


FIG. 3.2 – *Approximation polynômiale de degré 2 de la fonction* $\sqrt{|x|}\sin(|x|)$ *à partir de réalisations bruitées.*

Nous allons illustrer nos propos en observant deux modèles d'apprentissage : un polynôme et un réseau de neurones. Un ensemble d'apprentissage a été créé en évaluant la fonction $y = \sin(|x|) * \sqrt{(}|x|) + \epsilon$ pour 20 valeurs de x uniformément réparties entre -5 et 5, ϵ étant un bruit aléatoire uniformément distribué entre -0.2 et 0.2. Dans un premier temps, des interpolations polynômiales de degré 1,2,10 et 19 sur la base d'apprentissage tentent d'approximer la fonction. Le résultat de ces interpolations est présenté dans les figures 3.1, 3.2, 3.3 et 3.4. Nous pouvons observer que l'interpolation linéaire n'est pas adaptée à ce type de fonction. Un polynôme de degré 2 est insuffisant. Un polynôme de

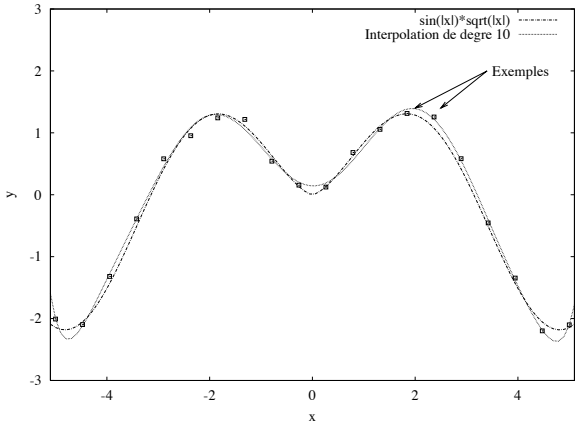


FIG. 3.3 – *Approximation polynômiale de degré 10 de la fonction* $\sqrt{|x|}\sin(|x|)$ *à partir de réalisations bruitées.*

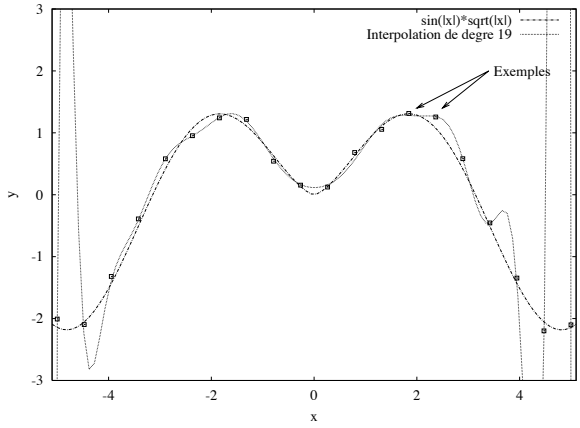


FIG. 3.4 – *Approximation polynômiale de degré 19 de la fonction* $\sqrt{|x|}\sin(|x|)$ *à partir de réalisations bruitées.*

degré 10 généralise bien les exemples. Le polynôme de degré 19 passe par tous les exemples mais il interpole mal les situations intermédiaires. Ainsi, les modèles qui ont peu de paramètres ne peuvent modéliser que des fonctions simples tandis que les modèles complexes s'adaptent très bien aux données d'apprentissage mais leur souplesse nécessite un nombre plus important d'exemples faute de quoi la généralisation n'est pas bonne.

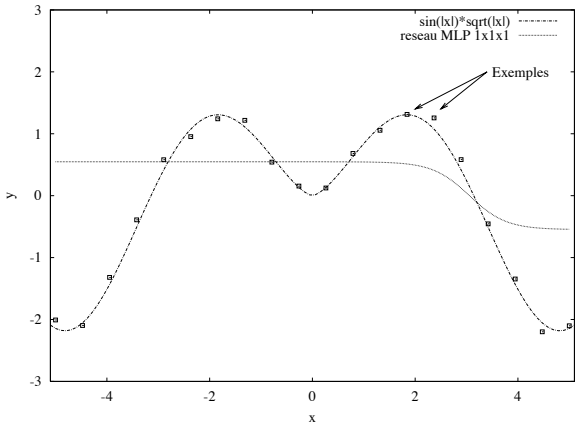


FIG. 3.5 – *Approximation neuronale avec 1 unité cachée (2 poids) de la fonction* $\sqrt{|x|}\sin(|x|)$ *à partir de réalisations bruitées.*

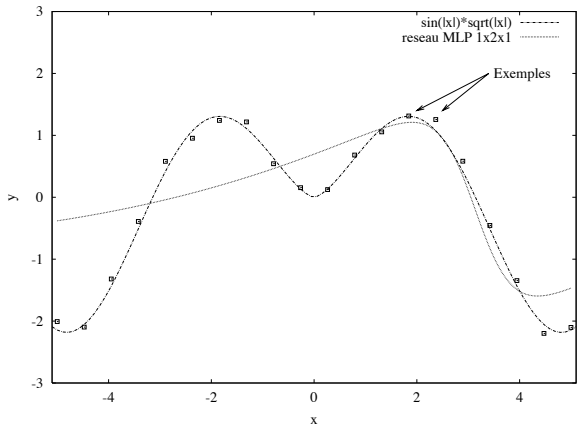


FIG. 3.6 – *Approximation neuronale avec 2 unités cachées (8 poids) de la fonction* $\sqrt{|x|}\sin(|x|)$ *à partir de réalisations bruitées.*

Dans le cas des réseaux de neurones, les performances en généralisation ont été évaluées après 1000 cycles d'apprentissage sur un ensemble de test constitué de 200 valeurs uniformément réparties entre -5 et 5. Quatre réseaux ont été testés avec un nombre différent de neurones sur la couche cachée. Toutes les unités cachées ont une fonction sigmoïde. La sortie est linéaire. Il n'y a pas de biais. Le premier réseau n'a qu'un neurone caché. Les performances sont faibles. On observe l'effet de la fonction sigmoïde (figure 3.5). Le second neurone à deux unités cachées. Les performances ne sont guère meilleures pour la fonction

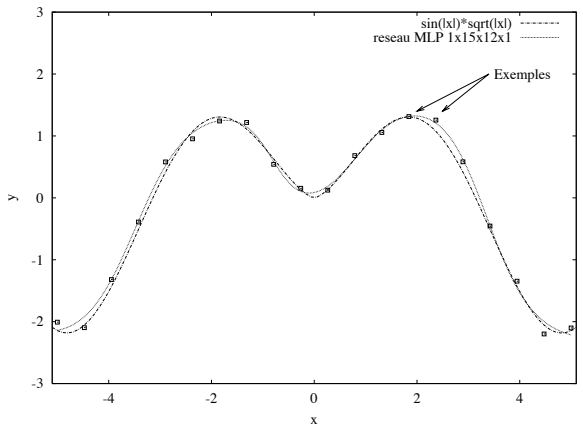


FIG. 3.7 – *Approximation neuronale avec un réseau 1x15x12x1 (207 poids) de la fonction* $\sqrt{|x|}\sin(|x|)$ *à partir de réalisations bruitées.*

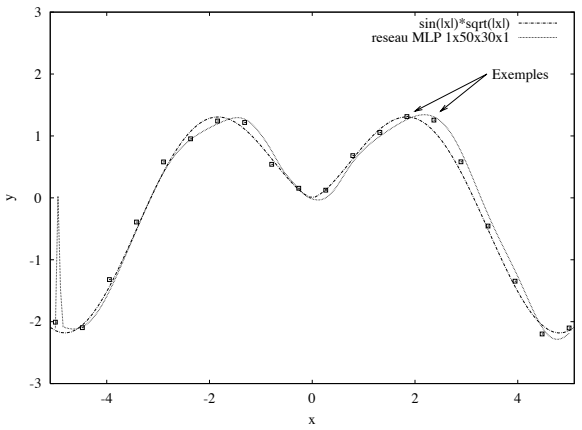


FIG. 3.8 – *Approximation neuronale avec un réseau 1x50x30x1 (1580 poids) de la fonction* $\sqrt{|x|}\sin(|x|)$ *à partir de réalisations bruitées.*

choisie (figure 3.6). Le troisième réseau comporte 27 neurones cachés. Ces performances sont très correctes (figure 3.7). La généralisation se passe bien malgré le peu d'exemples. Finalement, un réseau avec un nombre très important de poids (plus de 1500) par rapport aux 20 exemples montre des signes de surapprentissage identiques au polynôme de degré 19 (figure 3.8). Mais ces performances sont remarquables pour une telle disproportion entre le nombre de paramètres et le nombre d'exemples car les exemples sont équirépartis, donc l'espace est représenté de façon homogène. Afin d'éviter, ou en tout cas de limiter, le phénomène de surapprentissage dans les réseaux de neurones, diverses techniques ont été mises au point parmi lesquelles citons la sélection de modèles, l'arrêt anticipé, le weight decay et l'élagage (voir section 3.1.3).

En conclusion, si le nombre de paramètres est insuffisant par rapport à la complexité de la tâche à modéliser, les performances en généralisation ne seront pas bonnes par manque de flexibilité. A l'inverse, si le nombre de paramètres est excessif par rapport au nombre d'exemples, les performances en généralisation ne seront pas toujours bonnes par manque d'information sur le problème. On comprend mieux maintenant l'influence du nombre de paramètres sur les performances et l'intérêt de pouvoir modifier ce nombre au cours de l'apprentissage afin d'éviter la multiplication des modèles à tester.

3.1.2 Les algorithmes incrémentaux

Rappelons avant toute chose que l'utilisation des couches cachées dans les réseaux de neurones artificiels ne se justifie que si les éléments ne sont pas linéairement séparables dans le cas d'une classification ou si une régression linéaire ne suffit pas dans le cas d'un problème de régression.

Le principe général des algorithmes incrémentaux est le suivant :

– Partir d'une architecture initiale minimale :

Trois cas peuvent servir de point de départ à un apprentissage incrémental : les entrées ne sont pas connectées aux sorties (on part de la fonction nulle), les entrées sont entièrement connectées aux sorties, une architecture multicouches particulière est mise en place.

– Évaluer les performances du modèle.

– Si les performance ne sont pas satisfaisantes, modifier l'architecture.

Entrons dans le détail des algorithmes incrémentaux avec un modèle supervisé applicable à des problèmes de régression et un modèle non supervisé applicable à des problèmes de catégorisation.

Cascade-correlation

Cascade-correlation [Fahlman and Lebiere, 1990] est certainement le modèle incrémental le plus utilisé. L'idée est d'ajouter des unités cachées pour qu'elles se spécialisent sur les erreurs effectuées jusqu'alors par le réseau. L'algorithme applique à une architecture en cascade un apprentissage des poids basé sur une maximisation de la corrélation (en fait il s'agit d'une covariance) entre les poids de la nouvelle unité et les erreurs obtenues en son absence.

– Une architecture en cascade :
Le réseau est initialisé en connectant complètement et uniquement la couche d'entrée à la couche de sortie. Il s'agit donc d'un perceptron simple. Si les critères de performance ne sont pas atteints après un nombre limité d'apprentissages, une unité cachée est ajoutée au réseau. Ses entrées sont les entrées du réseau plus les sorties des unités cachées déjà ajoutées. Sa sortie sera connectée aux sorties du réseau après apprentissage des poids du neurone. L'architecture présente donc une succession de couches cachées ne comportant qu'une unité laquelle est entièrement connectée à toutes les unités antérieures ce qui a pour effet de rappeler la forme d'une cascade ou d'une pyramide suivant le sens de la représentation (voir figure 3.9).

– Un apprentissage centré sur une corrélation :

1. L'apprentissage des poids entre les entrées et les sorties se fait de manière classique en appliquant l'algorithme de rétro-propagation ou une variante telle que l'algorithme Quickprop [Fahlman, 1989].
2. A l'ajout d'une nouvelle unité, les poids déjà existants sont gelés pour permettre l'apprentissage des poids $\{w_i\}$ de cette nouvelle unité, lesquels seront modifiés afin de maximiser la valeur absolue de la covariance (équation 3.1) sur l'ensemble d'apprentissage, entre la sortie du neurone et l'erreur obtenue par le réseau sans prendre en compte cette nouvelle unité. A cet instant, les connexions entre la sortie de la nouvelle unité et les neurones de sortie n'existent pas. La valeur à maximiser est :

$$v = \sum_{j=1}^{J} \left| \sum_{n=1}^{N} (z^n - \bar{z})(e_j^n - \bar{e}_j) \right| \tag{3.1}$$

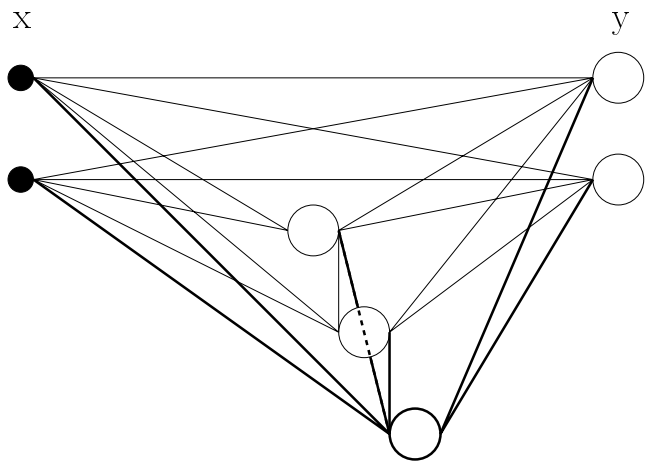


FIG. 3.9 – *Architecture évolutive de l'algorithme* cascade-correlation. *Les éléments représentés en gras correspondent à la dernière extension du réseau dont le but est d'augmenter sa capacité de modélisation.*

où $z = g(\sum_i w_i x_i)$ est la sortie de la nouvelle unité et $e_j^n (= y_j^n - d_j^n)$ est l'erreur produite par le neurone de sortie d'indice j pour l'exemple d'indice n entre la valeur calculée y_j^n et la valeur désirée d_j^n. $\bar{e}_j$ représente la moyenne des valeurs du neurone j.

La modification à apporter aux poids $\{w_i\}$ pour maximiser v peut être obtenue de manière classique par un algorithme de descente de gradient ou un algorithme du second ordre, plus rapide, tel que la méthode du gradient conjugué échelonné [Moller, 1993]. A titre d'information, la dérivée de v (équation 3.1) par rapport au poids w_i vaut :

$$\frac{\partial v}{\partial w_i} = \sum_{j=1}^{J} sign(v_j) \sum_{n=1}^{N} g'(x_i^n)(e_j^n - \bar{e_j})$$

3. Les nouveaux poids $\{w_i\}$ sont gelés. Les connexions entre la sortie du nouveau neurone et les neurones de sortie deviennent effectives. Tous les poids aboutissant aux neurones de sortie sont réajustés par apprentissage.

Il est intéressant de constater la rapidité d'apprentissage de cet algorithme. Tout d'abord, peu de poids sont entraînés en même temps (que ce soient les poids du nouveau neurone ou les poids des neurones de sortie). Ensuite, les poids des neurones cachés étant figés une fois pour toutes, leur valeur de sortie pour chaque exemple ne change pas (il est donc possible de la conserver en mémoire). Les algorithmes du second ordre peuvent lui être appliqués.

Le réseau d'initialisation de l'algorithme incrémental de cascade-correlation est un perceptron simple. Le réseau réalise donc au départ une régression linéaire (voir section 2.3.1). La fonction modélisée par le réseau est une projection sur l'axe principal. Après l'ajout d'une nouvelle unité, le modèle s'enrichit d'un perceptron multicouches. Ce dernier

modélise une fonction plus flexible qui complétera une partie des erreurs de l'approximation linéaire. Ensuite, l'ajout d'un neurone correspond à l'ajout d'une fonction de plus en plus flexible qui se spécialise sur les erreurs du modèle. Donc le réseau est semi-distribué, c'est-à-dire que certaines connexions apportent une légère modification à la réponse du modèle alors que d'autres contribuent à déterminer la valeur moyenne de sortie. Puisque les connexions n'ont pas la même importance le modèle est moins robuste. De plus, puisque les nouveaux poids sont dédiés nécessairement à l'affinage de la fonction, la puissance de calcul du réseau est diminuée. Mais l'organisation particulière du réseau le rend facilement interprétable localement.

Finalement, l'algorithme cascade-correlation présente une solution efficace pour, à partir d'une architecture minimale, construire un modèle qui apprendra à combler ses erreurs. Nous pouvons le rapprocher en cela du modèle SMART (section 2.3.2) qui, pour sa part, de possède pas d'architecture en cascade puisque les nouvelles unités ont uniquement pour entrées les entrées du réseau.

Suite à cet exemple d'apprentissage incrémental supervisé, nous allons décrire maintenant un algorithme incrémental non supervisé permettant d'effectuer de façon dynamique une quantification vectorielle.

Growing neural gas

Growing neural gas est un algorithme de quantification vectorielle (voir section 2.5.7 page 83) qui conserve la topologie des formes d'entrée dans l'espace de sortie. Il se différencie des cartes auto-organisatrices de Kohonen par deux modifications majeures. La première opérée par Martinetz donna naissance à l'algorithme neural gas [Martinetz, 1993] (voir section 2.5.4). Pour ce modèle, la topologie n'est pas déterminée au départ, elle se construit et évolue en fonction d'un apprentissage hebbien compétitif. Par la suite, Fritzke [Fritzke, 1995] en fit une version dynamique en ajoutant périodiquement une classe et dans certain cas extrêmes en en supprimant une. L'apprentissage de la topologie est identique à celui de l'algorithme de *neural gas*. En voici le fonctionnement complet :

Partant de deux prototypes aléatoires, un prototype est ajouté périodiquement entre le prototype présentant le cumul d'erreur le plus important et son voisin ayant le cumul d'erreur le plus important. Seuls le prototype le plus proche de la forme d'entrée et ses voisins ont leurs poids modifiés par l'apprentissage, suivant le même type de loi de celle de l'algorithme de Kohonen. Si une unité se retrouve sans connexion, elle est supprimée. Le nombre de classes augmente périodiquement jusqu'à un nombre maximum prédéfini ou plus généralement jusqu'à ce que le critère d'arrêt soit atteint.

Le domaine d'utilisation de *growing neural gas* est similaire à celui des méthodes auto-organisées de la section 2.5.1. Il permet le partitionnement des données, la quantification vectorielle et l'extraction de caractéristique. Mais il résout également le problème du choix du nombre de vecteur de référence c'est-à-dire du nombre de classes. Un autre aspect particulièrement de cet algorithme est la visualisation de la zone qui est le moins bien représenté puisque sa localisation coïncide avec le lieu d'apparition d'un nouveau prototype.

Algorithme 9 Algorithme de *growing neural gas*

1: L'ensemble A des prototypes est initialisé avec deux prototypes aléatoires a et b.
L'ensemble C des connexions et l'ensemble des voisins de a et de b sont vides.

2: Choisir une forme $\mathbf{x}$ de l'ensemble d'apprentissage.

3: Déterminer les deux prototypes de A, $s1$ et $s2$, les plus proches de $\mathbf{x}$:
$||\mathbf{x} - \mathbf{w_{s1}}|| \leq ||\mathbf{x} - \mathbf{w_{s2}}|| \leq ||\mathbf{x} - \mathbf{w_i}||, \ \forall i \in A$

4: Créer, si elle n'existait pas, une connexion entre s1 et s2
Si ($s1$,$s2$) $\in C$, alors $C = C \cup \{(s1,s2)\}$ et $N_{s1} = N_{s1} \cup \{s2\}$
où N_{s1} est l'ensemble des voisins topologiques de s1.

5: Mettre l'âge de la connexion à zéro.
$age_{s1,s2} = 0$

6: Mettre à jour l'erreur locale du prototype gagnant :
$\mathbf{E_{s1}}(t+1) = \mathbf{E_{s1}}(t) + ||\mathbf{x} - \mathbf{w_{s1}}||^2$

7: Mettre à jour le prototype gagnant et ses voisins :
$$\begin{aligned} \mathbf{w_{s1}}(t+1) &= \mathbf{w_{s1}}(t) + \epsilon_b(\mathbf{x} - \mathbf{w_{s1}}) \\ \mathbf{w_i}(t+1) &= \mathbf{w_i}(t) + \epsilon_n(\mathbf{x} - \mathbf{w_i}), \quad \forall i \in N_{s1} \end{aligned}$$

8: Incrémenter la durée de vie de tous les voisins du neurone gagnant i.e.
si $s1,j \in C$ alors $age_{s1,j}(t+1) = age_{s1,j}(t) + 1 \ \ \forall i \in N_{s1}$

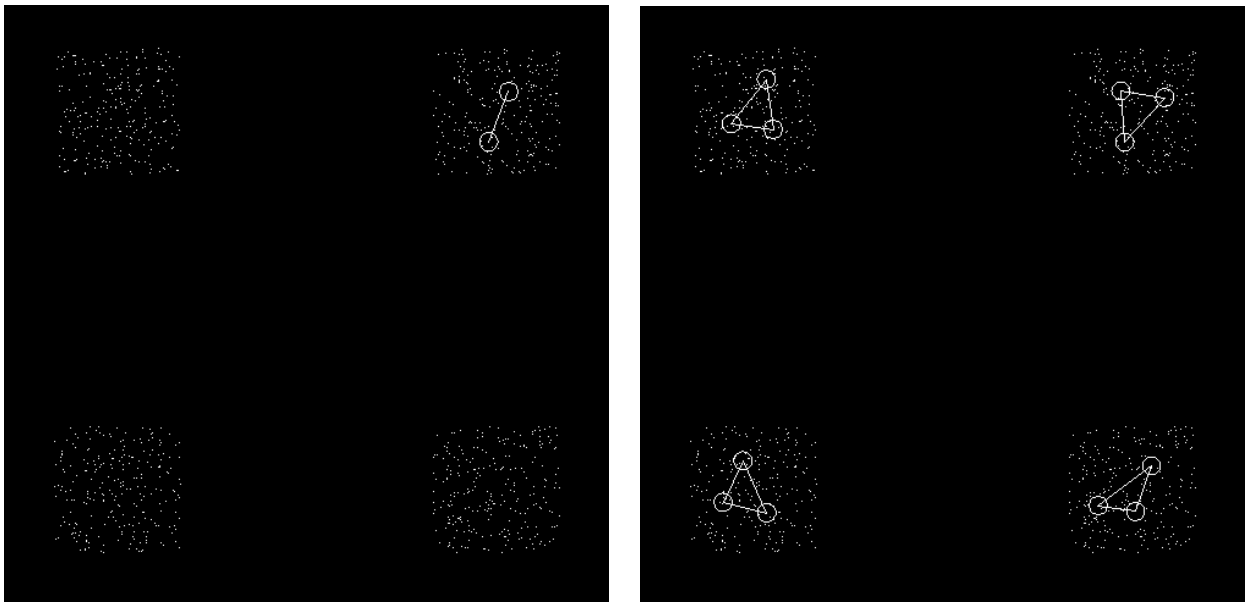
9: Supprimer toutes les connexions du neurone gagnant qui excèdent la durée de vie ie
si $age_{s1,j} > age_{max}$ et $s1,j \in C$ alors $C = C \setminus s1,j$
Si un prototype n'est plus connecté, i.e. n'a plus de voisins, alors il est supprimé.

10: Si le nombre d'exemples présentés est un multiple de la période λ, alors créer un prototype :

1. insérer un nouveau prototype r entre le prototype q dont l'erreur locale est la plus importante et son voisin f ayant l'erreur locale la plus forte.
   $A = A \cup \{r\}$
   $\mathbf{w_r} = 0.5(\mathbf{w_q} - \mathbf{w_f})$
2. connecter r à q et f.
   $C = C \cup \{(r,q),(r,f)\}$
   et supprimer la connexion entre q et f.
   $C = C \setminus (q,f)$
3. décroître l'erreur locale de q et f
   $\mathbf{E_q}(t+1) = \mathbf{E_q}(t) - \alpha\mathbf{E_q}(t)$
   $\mathbf{E_f}(t+1) = \mathbf{E_f}(t) - \alpha\mathbf{E_f}(t)$
   interpoler l'erreur locale de r :
   $\mathbf{E_r} = 0.5(\mathbf{E_q} + \mathbf{E_f})$

11: Décroître l'erreur locale de tous les prototypes :
$\mathbf{E_i}(t+1) = \mathbf{E_i}(t) - \beta\mathbf{E_i}(t) \ \ \forall i \in A$

12: Si le critère d'arrêt (taille du réseau ou mesure de performance) n'est pas atteint, alors aller en 2.



(a) Exemple d'initialisation de *growing neural gas*. Ici, les deux prototypes initiaux sont initialisés aléatoirement à deux éléments du corpus d'apprentissage. Puis, le voisinage de ces deux prototypes est établi.

(b) Quantification vectorielle par *growing neural gas*. Au fur et à mesure de l'apprentissage, le nombre de prototypes augmente et leurs voisinages sont appris en fonction de la distribution des données.

FIG. 3.10 – *Illustration de l'évolution* growing neural gas.

Expérimentation : Nous avons expérimenté l'algorithme de *Growing Neural Gas* dans le même cadre que l'étude menée dans la section 2.5.6 sur d'autres méthodes auto-organisées. Le but était d'obtenir un partitionnement de l'environnement à partir de la base de données géographiques. Les vecteurs de référence de *Growing Neural Gas* (figure 3.12) s'apparentent à ceux de l'algorithme de *Neural Gas* (figure 2.31). Ils sont légèrement plus dispersés puisque tous les vecteurs ne bougent pas en direction de la forme courante comme cela est le cas avec *Neural Gas*. Il est extrêmement rare de constater la suppression d'un vecteur de référence. L'algorithme est quasiment toujours strictement croissant.

Les algorithmes incrémentaux ont ceci d'intéressant que les différentes architectures qui aboutissent à la solution finale maximisent la performance malgré l'insuffisance de leurs ressources ce qui les oblige, à chaque étape, à extraire et à manipuler les informations les plus utiles. Chaque solution intermédiaire a donc une valeur informative qui vient s'ajouter à la performance finale et qui nous intéresse particulièrement dans le cadre de cette thèse.

Remarque : de nombreux algorithmes incrémentaux ont été inspirés par des méthodes statistiques de partitionnement, d'analyse discriminante ou de régression [Ripley, 1996]. A ce titre, la lecture d'articles faisant le lien entre les deux domaines est riche en compléments

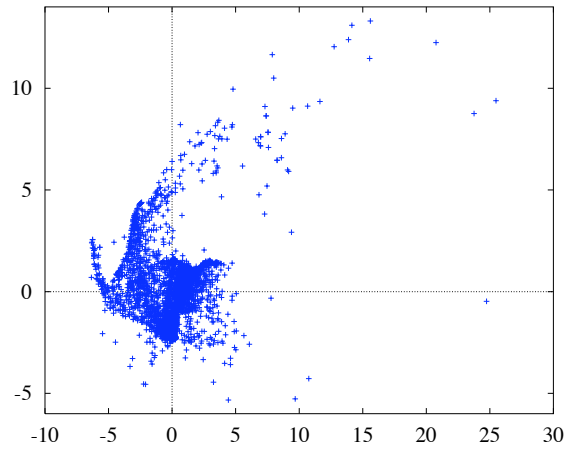


FIG. 3.11 – *Données géographiques projetées.*

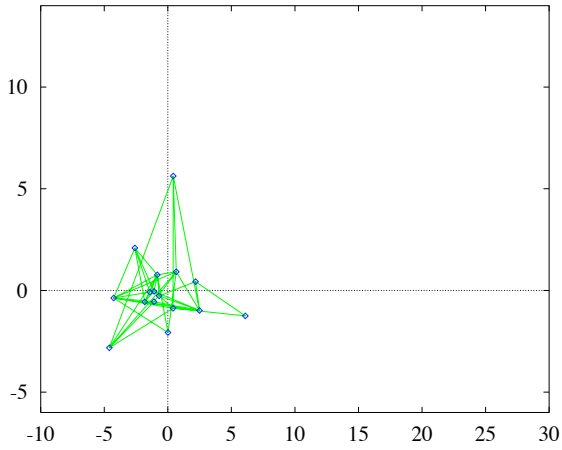


FIG. 3.12 – *Réseau de* growing neural gas *projeté.*

d'information pour qui veut mieux comprendre les raisons profondes d'un algorithme particulier et donc ce qu'il peut apporter en qualité de compréhension du problème, au delà d'une amélioration des performances.

3.1.3 Les algorithmes d'élagage

Le propos des algorithmes d'élagage est de supprimer les connexions et/ou les neurones qui contribuent le moins à la réponse du système. Le but recherché est multiple. Du point de vue des performances, par la suppression de paramètres superflus, il est possible de supprimer le bruit qu'ils génèrent dans le réseau et ainsi d'améliorer la généralisation. On aboutit à une architecture en adéquation avec le problème. Du point de vue de l'implémentation, la diminution des paramètres et des unités diminue les besoins de ressource (mémoire,CPU,composant) et accélère le temps de calcul. Du point de vue informatif, la suppression de connexions permet d'extraire un nombre raisonnable de règles et la suppression de neurones d'entrée permet de sélectionner les variables d'entrée [Remm, 1996].

En fonction du but et de la méthode, la suppression peut s'appliquer à une connexion, à un neurone d'une couche cachée ou à un neurone d'entrée. Reed a consacré un article à ce sujet dans lequel il décrit principalement des méthodes neuronales [Reed, 1993]. Un article de Cibas et al. se focalise sur la sélection de variables [Cibas *et al.*, 1996]. Pour l'extraction de règles on pourra consulter [Andrews *et al.*, 1995]. Pour une utilisation dans les apprentissages bayésiens citons [Williams, 1995]. Pour un recoupement avec les statistiques, l'analyse de Ripley est intéressante [Ripley, 1996]

Suppression de connexions

La première idée qui vient à l'esprit est de supprimer les poids de valeur faible puisque les sommes pondérées dont ils font partie varieront peu dans ces circonstances. La valeur

absolue du poids est appelée magnitude dans les méthodes basées sur ce principe. Mais c'est ne pas tenir compte de la valeur de x_i qui intervient au même titre que w_i dans la somme pondérée $\sum_i x_i w_i$. La standardisation des entrées et des sorties réduira la variation des x_i mais l'absence de théorie donne peu de crédit à ces méthodes. En effet, la suppression d'un connexion dont le poids est proche de zéro peu avoir une conséquence importante si la valeur de ce poids permettait de faire basculer la réponse de la fonction d'activation d'un neurone à seuil.

L'idée commune à de nombreuses méthodes est de construire une approximation de la surface d'erreur au voisinage d'un minimum local par un développement de Taylor (eq 3.2) pour permettre une analyse des perturbations causées par la suppression d'un poids.

$$\delta E = \sum_i \frac{\partial E}{\partial w_i}\,\delta w_i + \frac{1}{2}\sum_i\sum_j \frac{\partial^2 E}{\partial w_i\,\partial w_j}\,\delta w_i\,\delta w_j + O(||\delta \mathbf{W}||^3) \tag{3.2}$$

certaines hypothèses sont émises pour simplifier les calculs :
- Si la variation considérée pour un poids est sa suppression, alors $\delta w_i = -w_i$.
- On suppose que l'apprentissage des poids converge vers un minimum local. Les termes $\partial E/\partial w_i$ sont donc nuls.
- On suppose qu'en ce minimum la surface de l'erreur est approximativement quadratique. Les termes contenus dans $O(||\delta \mathbf{W}||^3)$ sont donc négligeables.

L'équation 3.2 se résume donc aux termes faisant intervenir les dérivées partielles d'ordre deux qui sont, par définition, les coefficients de la matrice hessienne $\mathbf{H}$.

$$\delta E = \frac{1}{2}\sum_i\sum_j \frac{\partial^2 E}{\partial w_i\,\partial w_j}\,w_i\,w_j \tag{3.3}$$

Optimal Brain Damage (OBD) proposé par Le Cun, Denker et Solla en 1989 [Le Cun *et al.*, 1990] est à l'origine de cette démarche et constitue l'algorithme neuromimétique de référence en matière d'élagage. Il offre un bon compromis entre simplicité et efficacité. En effet, ses auteurs étaient confrontés au problème désormais célèbre de la reconnaissance manuscrite des codes postaux américains. Les 2600 poids de leur réseau rendaient difficile le calcul d'une matrice de 6,5 millions termes et ce à chaque approximation. Ils ont donc simplifié l'approximation en supposant que la matrice hessienne était diagonale ce qui revient à faire l'hypothèse que la variation de l'erreur causée par la suppression de plusieurs poids n'est autre que la somme des variations engendrées par la suppression individuelle de chaque poids.

$$\delta E = \frac{1}{2}\sum_i \frac{\partial^2 E}{\partial w_i\,\partial w_i}\,w_i^2 = \sum_i \delta E(w_i)$$

Dans ces conditions, la sensibilité s_i de la fonction de coût à la suppression d'un poids w_i vaut :

$$s_i = \delta E(w_i) = \frac{1}{2}\frac{\partial^2 E}{\partial w_i^2}\,w_i^2 \tag{3.4}$$

Le principe de OBD est donc de calculer la sensibilité de la fonction de coût pour chacune des connexions et de supprimer la connexion dont l'absence entraîne la moindre croissance de l'erreur. La suppression ne doit pas toucher un grand nombre de poids ou un poids ayant une grande valeur absolue car les calculs sont valables pour une petite variation au voisinage de la position actuelle.

L'algorithme général est le suivant :
1. Choisir une architecture de départ
2. Faire converger le réseau
3. Calculer les dérivées secondes de l'erreur par rapport à chaque poids
4. En déduire les sensibilités du réseau pour chaque poids
5. Ordonner les poids dans l'ordre croissant des sensibilités
6. Supprimer les poids dont la sensibilité est faible
7. Retourner à l'étape 2

Remarque : Il est possible de ne calculer qu'une dérivée seconde de remplacement pour toutes les connexions d'un même neurone, et d'en déduire simplement les sensibilités [Remm, 1996].

Pour éviter de faire converger à nouveau le réseau après la suppression de quelques connexions, il faudrait pouvoir calculer les corrections à apporter aux poids maintenus connaissant les poids supprimés. Dans ce cas, il devient envisageable de calculer la matrice hessienne en relâchant l'hypothèse contraignante d'une matrice diagonale. C'est ce que proposent Hassibi et Stork [Hassibi and Stork, 1993] avec Optimal Brain Surgeon (OBS). Reprenons l'équation 3.3 sous forme matricielle.

$$\delta E = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} \tag{3.5}$$

On cherche la plus petite variation de l'erreur $\delta \mathbf{w}$ dans le cas d'une absence de w_i. La recherche d'un minimum de δE sous contrainte $g(\delta \mathbf{w}) = \mathbf{e}_i^T \delta \mathbf{w} + w_i = 0$ nous renvoie à la recherche d'un minimum sans contrainte de la fonction lagrangienne :

$$L_i(\delta \mathbf{w}, \lambda) = \delta E(\delta \mathbf{w}) + \lambda g(\delta \mathbf{w}) = \frac{1}{2}\delta \mathbf{w}^T \mathbf{H}\, \delta \mathbf{w} + \lambda(\mathbf{e}_i^T \delta \mathbf{w} + w_i)$$

où λ est un multiplicateur de Lagrange et $\mathbf{e}_i$ est le vecteur unitaire parallèle à l'axe w_i.

$$\begin{aligned}
\frac{\partial L}{\partial(\delta w_i)} &= \mathbf{H}\,\delta\mathbf{w}\lambda\mathbf{e}_i^T = 0 \\
\delta\mathbf{w} &= -\lambda\mathbf{H}^{-1}\mathbf{e}_i^T && (3.6)\\
\frac{\partial L}{\partial(\delta\lambda)} &= \mathbf{e}_i^T\delta\mathbf{w} + w_i = 0 && (3.7)\\
&= -\lambda\mathbf{e}_i^T\mathbf{H}^{-1}\mathbf{e}_i^T + w_i = 0 \\
\lambda &= \frac{w_i}{(\mathbf{H}^{-1})_{ii}} && (3.8)\\
\delta\mathbf{w} &= -\frac{w_i}{(\mathbf{H}^{-1})_{ii}}\,\mathbf{H}^{-1}\mathbf{e}_i^T && (3.9)
\end{aligned}$$

En remplaçant $\delta\mathbf{w}$ dans l'équation 3.5, on obtient l'augmentation de l'erreur dans le cas d'une suppression de w_i :

$$\begin{aligned}
\delta E(w_i) &= \frac{1}{2}\delta\mathbf{w}^T\mathbf{H}\left(-\frac{w_i}{(\mathbf{H}^{-1})_{ii}}\mathbf{H}^{-1}\mathbf{e}_i\right) \\
&= -\frac{1}{2}\frac{w_i}{(\mathbf{H}^{-1})_{ii}}\delta\mathbf{w}^T\mathbf{e}_i \\
&= \frac{1}{2}\frac{w_i^2}{(\mathbf{H}^{-1})_{ii}} && (3.10)
\end{aligned}$$

Si la matrice hessienne est diagonale on retrouve le résultat de OBD (équation 3.4 page 101).

Afin de comparer et d'interpréter facilement les trois techniques d'élagage les plus classiques, la figure 3.13 montre les différents choix tactiques opérés respectivement par les algorithmes Magnitude-Based Pruning, Optimal Brain Damage et Optimal Brain Surgeon. Depuis le point de convergence, supposé être un minimum, un algorithme d'élagage basé sur la magnitude supprimera le plus petit poids, soit w_2 ce qui le conduira à se déplacer dans l'espace des poids en (w_1,0). Optimal Brain Damage tient compte du rayon de courbure préférant supprimer w_1 pour aboutir en (0,w_2). Optimal Brain Surgeon supprimera w_1 pour les mêmes raisons que OBD et il corrigera également le poids w_2 en (0, $w_2 + \delta w_2$). On constate d'après l'ellipse isométrique extérieure que le choix de MBP provoquera la plus grande augmentation de l'erreur et celui de OBS la plus petite.

En théorie, le nombre de poids qu'il est possible de supprimer en une seule passe doit être faible puisqu'il doit respecter l'hypothèse d'un faible déplacement autour du minimum. De plus, l'organisation séquentielle des couches d'un réseau de neurone fait que la suppression d'un poids peut avoir des répercutions non négligeables sur les autres poids. Donc l'idéal serait de supprimer les poids un par un. Mais dans la pratique, il n'est pas rare d'utiliser un seuil, un pourcentage voire un nombre fixe de suppressions de poids.

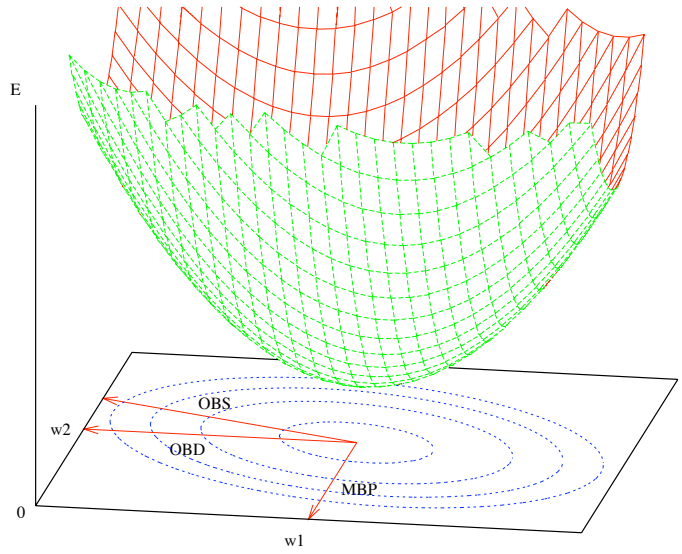
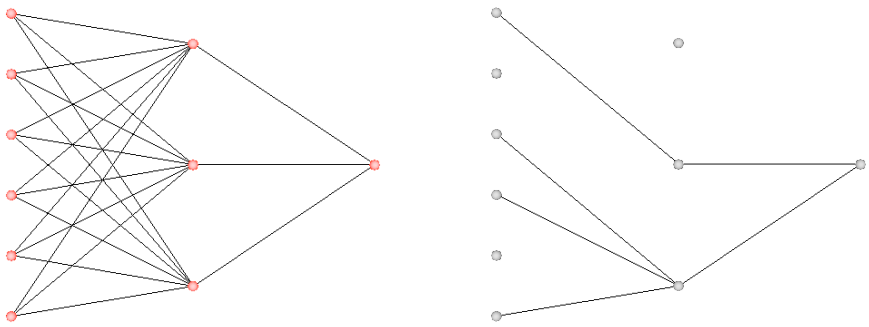


FIG. 3.13 – *Modification opérée dans l'espace des poids par différentes techniques d'élagage (MBP : magnitude-based pruning, OBD : optimal brain damage, OBS : optimal brain surgeon) et leurs conséquences sur l'accroissement de l'erreur.*



(a) initialisation. (b) état final.

FIG. 3.14 – *Perceptron multicouches avant et après élagage. A l'état initial, le réseau est entièrement connecté (a). A l'état final, les variables d'entrée les plus pertinentes sont conservées, l'architecture du réseau est simplifiée pour donner un modèle qui exprime les relations essentielles entre les entrées et la sortie, et à partir desquelles il est possible d'extraire des règles.*

Suppression de neurones

La « squelettisation » (skeletonization en anglais) définie par Smolensky et Mozer [Mozer and Smolensky, 1989] utilise les mêmes considérations d'élagage que OBD [Le

Cun *et al.*, 1990] et OBS [Hassibi and Stork, 1993] à savoir l'estimation de l'accroissement de l'erreur dans le cas de la suppression d'un élément, mais ici, il s'agit d'un neurone. Pour chaque neurone i, une variable binaire α_i (strength attentional) indique la présence ou l'absence du neurone à l'intention des neurones post-synaptiques auxquels il est connecté pour qu'ils puissent calculer leur entrée a_j en conséquence. Ce qui se traduit par :

$$a_j = \sum_i \alpha_i x_i w_{ij}$$

Donc quand $\alpha_i = 0$ le neurone n'existe pas aux yeux des autres (si tant est que les neurones aient des yeux). La sensibilité de l'erreur à la suppression du neurone i vaut :

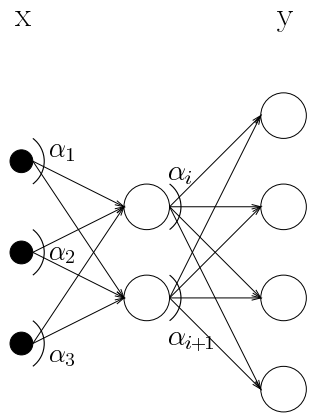


FIG. 3.15 – *Les variables binaires α_i de l'algorithme de « skeletonization » permettent de supprimer l'influence de certains neurones lorsqu'elles sont nulles.*

$$s_i = E_{\alpha_i=0} - E_{\alpha_i=1} \approx - \left. \frac{\partial E}{\partial \alpha_i} \right|_{\alpha_i=1}$$

Lorsque s_i est inférieur à un seuil, le neurone i est supprimé.

De nombreuses méthodes de suppression de neurones existent, en particulier la littérature est abondante en ce qui concerne la suppression des neurones d'entrée. Un état de l'art est disponible dans [Cibas *et al.*, 1996].

Expérimentation : Pour extraire de la connaissance du problème de prédiction du champ radioélectrique et améliorer les performances d'un perceptron multicouches (cf. section 2.3.2), nous lui avons appliqué l'algorithme d'élagage *Optimal Brain Damage* (les méthodes basées sur la magnitude des poids n'étant pas suffisamment fiables et l'algorithme *Optimal Brain Surgeon* nécessitant trop de temps de calcul sur une grosse application).

Nous sommes partis d'une architecture de dimension 32x10x1 (modèle A du tableau 3.1). Cette situation correspond à la situation de départ représentée par un A sur la figure 3.27. Les lettres A, B, C et D de la figure 3.27 représentent les situations importantes

caractéristiques d'une procédure d'élagage. Le point A correspond aux performances du réseau non élagué. Le point B correspond à une amélioration des performances par suppression du bruit interne au réseau dû à une surparamétrisation. Le point C correspond au cas où le réseau présente les mêmes performances que la situation de départ mais avec une architecture simplifiée. Finalement le point D correspond à un réseau très simplifié. Les performances de ce réseau sont moins bonnes mais son intérêt est à rechercher dans la simplicité de son architecture qui permet l'extraction de règles [Andrews *et al.*, 1995].

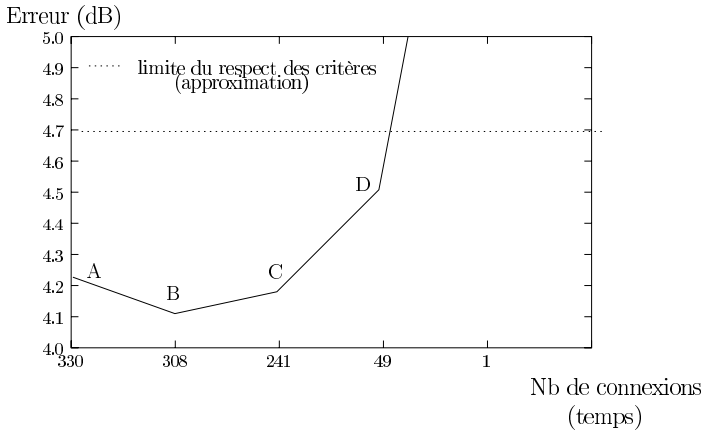


FIG. 3.16 – *Points stratégiques d'une procédure d'élagage. A : performance sans élagage, B : meilleure performance obtenue par suppression des connexions superflues, C : même performance qu'au départ mais pour un réseau réduit, D : performance moins bonne mais possibilité d'extraire plus de connaissances.*

Les connexions ont été supprimées une par une, et chaque suppression a été suivie d'un réapprentissage de 20 cycles pour compenser la perte de la connexion. Les meilleures performances sont obtenues pour un réseau dont 7% des connexions d'origine ont été supprimées (modèle B du tableau 3.1). La poursuite de l'élagage aboutit à la situation C où un réseau avec seulement 8 unités cachées et 73 % des connexions d'origine parvient aux mêmes performances que le réseau de départ (modèle C du tableau 3.1). Finalement, nous avons obtenu un réseau fortement élagué qui ne contient plus que 15% des connexions d'origine, 2 unités sur la couche cachée et 27 des 32 variables d'entrée, et qui continue de respecter les critères de qualité (modèle D du tableau 3.1). Ainsi nous apprenons que les variables 4, 5, 22, 27 et 29 sont les moins pertinentes parce qu'elles n'interviennent plus dans le calcul de la réponse du réseau par suite de l'élagage de toutes les connexions qui reliaient leur unité au réseau.

Signalons également que d'autres méthodes dites de régularisation ont les mêmes propriétés que celles présentées ci-dessus[Reed, 1993; Cibas *et al.*, 1996]. Elles ajoutent un terme de régularisation au terme correctif du poids ce qui revient à demander au réseau de modéliser une autre fonction. Le terme de régularisation ajoute une contrainte de mobilité au poids qui aura pour effet de réduire la capacité du réseau à interpoler strictement les valeurs du corpus d'apprentissage. Le réseau modélise une fonction plus simple qui fait abstraction de la part des fluctuations qui est due au bruit. Les petits poids limites

Modèle	Architecture	Phase	$\bar{e}$ (dB)	σ	Q_4 (%)	Q_6 (%)	Q_{11} (%)
A	*32*x*10*x*1 (330)*	*Apprentissage*	4.18	3.38	56.9	75.7	95.6
		Test	4.21	3.43	56.8	75.3	95.4
B	*32*x*10*x*1 (308)*	*Apprentissage*	4.10	3.31	57.7	76.6	96.0
		Test	4.11	3.31	57.0	76.5	96.1
C	*32*x*8*x*1 (241)*	*Apprentissage*	4.19	3.38	56.8	75.5	95.7
		Test	4.20	3.39	56.5	75.2	95.7
D	*27*x*2*x*1 (49)*	*Apprentissage*	4.50	3.66	53.9	72.3	94.1
		Test	4.51	3.64	53.3	71.8	94.0

TAB. 3.1 – *Performances de* Obtimal Brain Damage *sur le problème de prédiction du champ radioélectrique.*

les fortes variations en sortie. Les poids proches de zéro peuvent être réduits à zéro et leur connexions supprimées. Dans le cas d'une élimination massive, des neurones cachés ou des neurones d'entrée pourront se retrouver sans connexion de sortie et seront supprimés. Dans le dernier cas, cela aboutira à une sélection des variables d'entrée les plus pertinentes. Des expérimentations avec ce type de méthode ont été effectuées par France Télécom R&D sur le problème de prédiction du champ radioélectrique (voir [Balandier and Governo, 1998] et section 1.3.1 page 14).

Nous avons présenté les méthodes dynamiques les plus utilisées, dont s'inspirent de nombreuses variantes. Ces méthodes recherchent l'architecture minimale qui va satisfaire les critères de qualité. Ces critères sont difficiles à déterminer dans le cas des algorithmes incrémentaux car généralement l'augmentation de la taille du réseau va entraîner une amélioration des performances. Par exemple, dans le cas des méthodes de partitionnement, généralement le critère d'évaluation est basé sur l'inertie intra-classe. Or la solution optimale pour ce critère est d'avoir une classe par élément. Cette solution ne présente aucun intérêt pour l'extraction de connaissances. D'un autre coté, les algorithmes d'élagage simplifient le réseau en supprimant des connexions ce qui permet d'obtenir des règles de décision ou des neurones ce qui s'avère particulièrement instructif lorsque ces neurones appartiennent à la couche d'entrée. Dans le dernier cas, l'élagage conduit à une sélection des variables du problème. La prochaine section décrit un autre type d'organisation structurée.

3.2 Construction modulaire

Basée d'une part sur la réalité biologique qui nous montre que notre cerveau sous-traite les tâches à réaliser à des architectures spécifiques [Alexandre and Guyot, 1995] et d'autre part sur la devise belge « l'union fait la force », la combinaison de plusieurs réseaux de neurones peut conduire à l'amélioration des capacités du système tout en introduisant une meilleure compréhension sur son fonctionnement.

Avant toute chose, pour définir le module et la construction modulaire, commençons par dire ce qu'ils ne sont pas. Il paraît en effet important de préciser la distinction entre l'approche ensembliste et l'approche modulaire [Sharkey, 1999].
L'approche ensembliste [Hansen and Salamon, 1990; Sharkey, 1996] fait référence à l'utilisation d'une combinaison de réseaux connexionnistes, désignée par les termes de *comité* ou d'*assemblée*, dont les membres réalisent de manière redondante une même tâche mais par des moyens différents ou dans des conditions différentes afin de confronter leurs résultats pour obtenir une solution finale, par vote par exemple.
L'approche modulaire [Sharkey, 1997] divise le problème en sous-tâches [Hilario *et al.*, 1994; Lallement *et al.*, 1995]. Chaque sous-tâche est appréhendée par un module. Les différents modules sont combinés pour fournir la réponse du système. Les raisons de la décomposition peuvent tenir à une capacité spéciale intrinsèque à un module qui ne pourrait pas être obtenue autrement, à une amélioration des performances, à une facilitation des modifications ou à une meilleure compréhension. Les méthodes de décomposition peuvent être soit automatiques, dans ce cas l'algorithme de partitionnement est indépendant de l'application, soit explicites, c'est-à-dire induites par la spécificité des modules ou les connaissances implicites du problème. Finalement, on distingue quatre types de relation entre les modules. La relation peut être séquentielle : à un instant donné seul un module fonctionne et transmet son résultat au module suivant. La relation peut être coopérative : l'interaction est bijective. La relation peut être supervisée : un module coordonne le travail des autres modules. Ou bien, la relation peut être compétitive : on rejoint ici l'approche ensembliste, mais bien souvent la compétition à pour but une meilleure connaissance des compétences de chaque module ce qui conduira finalement à apprendre à sélectionner le bon module ou à combiner leur réponse ce qui reviendrait à de la supervision. Pour en savoir plus sur l'approche modulaire le lecteur peut lire les ouvrages ou articles suivants : [Sharkey, 1999; Rouzier, 1998].

3.2.1 Combinaisons de prédicteurs spécialisés sur un sous-espace

Le partitionnement des données en sous-groupes plus homogènes donc généralement plus faciles à caractériser permet généralement d'améliorer les performances en définissant un modèle spécialisé sur chaque sous-groupe. La procédure se fait en deux phases : premièrement, l'affectation de la donnée à son groupe ; deuxièmement, l'application d'un traitement spécifique au groupe.

Partitionnement de l'espace d'entrée

Le partitionnement de l'espace d'entrée est le plus couramment utilisé. Il permet d'essayer de mieux comprendre la distribution des données (cf. sections 2.1.2 et 2.5.6). La quasi totalité des fonctions à modéliser sont continues. Par conséquent, les éléments d'un même groupe auront des valeurs de sortie proches les unes des autres. Un modèle en charge de ce groupe sera plus à même de distinguer leurs particularités.

Expérimentation : Le partitionnement opéré par les méthodes auto-organisées, dont les résultats ont été présentés dans les sections 2.5.6 et 3.1.2, peut être exploité en spécialisant un modèle prédictif par groupe. Nous avons utilisé des perceptrons multicouches comme modèles prédictifs et non des réseaux récurrents de Elman, pourtant plus performants, car notre but était uniquement d'évaluer les différences entre les algorithmes compétitifs présentés. Nous avons étudié l'influence du nombre de classes sur les performances en prédiction suite à une catégorisation par cartes de Kohonen (tableau 3.2). Le choix d'une répartition en 9 catégories a été retenu pour comparer les méthodes compétitives (carte de Kohonen, algorithme de DeSieno, algorithme de *neural gas* et algorithme de *growing neural gas* sur des performances en prédiction (tableau 3.3)).

	Apprentissage		Test	
	μ	σ	μ	σ
SOM avec 6 classes	3.58	4.61	3.59	4.69
SOM avec 9 classes	3.36	4.45	3.48	4.57
SOM avec 16 classes	3.27	4.23	3.51	4.65

TAB. 3.2 – *Comparatif de l'influence du nombre de catégories produites par les cartes auto-organisatrices de Kohonen sur des performances en prédiction.*

	Apprentissage		Test	
	μ	σ	μ	σ
Carte de Kohonen	3.36	4.45	3.48	4.57
Algorithme de DeSieno	3.41	4.41	3.52	4.58
Algorithme de *neural gas*	3.58	4.63	3.59	4.69
Algorithme de *growing neural gas*	3.49	4.53	3.58	4.73
sans partitionnement	4.10	5.24	4.03	5.18

TAB. 3.3 – *Comparatif de méthodes compétitives sur des performances en prédiction.*

L'utilisation de modèles adaptés à des environnements spécialisés permet d'améliorer les performances, quelle que soit la méthode de partitionnement employée. On constate également que c'est plus l'approche que la méthode proprement dite qui est à l'origine de l'amélioration. L'approche modulaire présente d'autres intérêts, comme la possibilité de remplacer un prédicteur par un autre, mieux adapté au sous-ensemble considéré.

On peut se demander si une séparation en un nombre plus important de classes n'aboutirait pas à un nouveau gain. En fait, nous avons observé que certaines méthodes généraient des sous-environnements avec très peu d'éléments, sans pour autant réduire significativement l'erreur de prédiction. Remarquons qu'on n'est pas toujours certain d'obtenir de meilleurs résultats en fractionnant l'espace d'entrée, même si cela est fait sur un critère de proximité. En effet, si on réduit le nombre d'exemples, on peut se retrouver avec un nombre insuffisant d'exemples en apprentissage et perdre en performances de généralisation. Néanmoins, ici les performances montrent qu'on n'est pas touché par ces problèmes, ce qui prouve que les éléments à l'intérieur de chaque partition sont homogènes. Les méthodes de partitionnement sont donc efficaces.

L'amélioration des performances en prédiction est de l'ordre de 20% [Bougrain and Alexandre, 1999c; Bougrain, 1998b]. Nous avons non seulement réduit la moyenne des erreurs mais nous avons également réduit leur écart type. Les résultats sont plus fiables que précédemment. A l'avenir, le système de calcul de l'affaiblissement devrait comporter des méthodes capables de réaliser une prise en compte de la spécificité de environnement auquel appartient la situation pour améliorer la prédiction.

Partitionnement de l'espace de sortie

Lorsque les valeurs à obtenir en sortie du système sont connues, un partitionnement de l'espace de sortie peut également apporter une meilleure compréhension du problème et une amélioration des performances.

Expérimentation : Nous avons vu au chapitre 1 que les valeurs de l'atténuation pouvaient être séparées en deux sous-ensembles. Jusqu'à présent nous avons travaillé sur le sous-ensemble constitué de 97% des données dont l'atténuation est supérieure à 30 dB. Si nous essayons de modéliser les atténuations de l'ensemble des données du corpus complet avec un seul modèle, les performances sont nettement plus mauvaises que si nous utilisons un modèle de discrimination tel qu'une machine à *support vectors*, un perceptron multi-couches ou un arbre de décision pour séparer les données et que nous appliquons un réseau prédictif sur chacune des deux parties (tableau 3.4) [Bougrain, 1999b]. On peut donc avoir intérêt à utiliser des modèles spécialisées sur des sous-groupes de données discriminées à partir de leurs valeurs désirées.

Modèle	Phase	$\bar{e}$ (dB)	σ	Q_4 (%)	Q_6 (%)	Q_{11} (%)
sous-corpus (ames $\geq$ 30 dB)	*Apprentissage*	4.18	3.38	56.9	75.7	95.6
	Test	4.21	3.43	56.8	75.3	95.4
sous-corpus (ames <30 dB)	*Apprentissage*	0.43	0.39	99.9	100	100
	Test	0.46	0.44	100	100	100
corpus séparé	*Apprentissage*	4.07	nc	58.2	76.4	95.7
	Test	4.10	nc	58.1	76.0	95.5
corpus entier	*Apprentissage*	8.25	9.38	36.4	51.4	76.6
	Test	8.35	9.51	36.3	51.6	77.1

TAB. 3.4 – *Performances de modèles spécialisés sur un sous-espace de l'espace de sortie.*

Partitionnement des erreurs

Les performances des systèmes se résument bien souvent à une mesure statistique moyenne. Mais l'erreur commise peut varier fortement en fonction de la forme présentée. Ainsi, quelques situations difficiles à traiter peuvent dégrader fortement les performances (tableau 3.5). Partant de la distribution normale des erreurs généralement produites par

les réseaux neuromimétiques [Lawrence *et al.*, 1997], une détection et un traitement particulier peuvent être appliqués aux données qui génèrent les plus fortes erreurs dans un problème de régression, ceci afin d'améliorer les performances.

Expérimentation : La définition de différents critères de qualité basés sur une erreur limite particulière pour l'évaluation des performances d'un modèle de prédiction du champ radioélectrique montre que le pourcentage des erreurs supérieures à 11 dB est à distinguer des autres. Le tableau 3.5 sépare l'erreur moyenne des erreurs les plus fortes (10%) des autres.

Performances	Apprentissage		Test	
	$\bar{e}$ (dB)	σ	$\bar{e}$ (dB)	σ
prédicteur global	4.13	5.29	4.15	5.44
Faibles erreurs (90% du corpus)	3.33	4.03	3.36	4.08
Fortes erreurs (10% du corpus)	11.31	11.45	11.82	12.23

TAB. 3.5 – *Influence des fortes erreurs sur les performances.*

Nous voyons que 10% des données peuvent dégrader fortement les performances moyennes. Fixons un seuil distinguant les fortes erreurs des faibles erreurs, par exemple à 8.8 dB (i.e. que 10% des erreurs produites sur le corpus d'apprentissage seront considérées comme de fortes erreurs). Il est possible d'apprendre à discriminer à partir du corpus d'apprentissage les données qui risquent de produire de fortes erreurs. Un prédicteur spécialisé sur ces données permet d'améliorer les résultats (tableaux 3.5 et 3.6) [Bougrain, 1999b; Bougrain, 1999a]. Ici, encore nous avons pu voir que l'étude des données ou des résultats obtenus par réseaux de neurones pouvaient être exploités pour organiser les données de manière à améliorer les performances.

Performances	Apprentissage		Test	
	$\bar{e}$ (dB)	σ	$\bar{e}$ (dB)	σ
MLP (Fortes erreurs)	7.80	9.29	9.06	11.27
MLP (Faibles erreurs)	3.36	4.12	3.44	4.26
prédicteurs spécialisés	3.8	5.06	3.94	5.51

TAB. 3.6 – *Performances de modèles spécialisés sur une catégorie d'erreurs.*

L'utilisation d'un prédicteur spécialisé sur un sous-espace permet généralement une amélioration significative des performances. Mais il n'y a pas de raisons de se restreindre à la décision du prédicteur le plus proche, d'une part, parce que le prototype le plus proche peut être malgré tout à une grande distance de l'exemple, d'autre part, parce que les estimations des autres prédicteurs peuvent être intéressantes voire primordiales si d'autres prototypes se situent presque à la même distance de l'exemple que le prototype vainqueur. La section suivante présente une méthode qui tient compte de ces considérations.

3.2.2 Mélanges d'experts

L'apprentissage à partir d'exemples conduit souvent à étudier leur distribution pour mieux prédire ou classer. Il est rare que cette distribution puisse être évaluée correctement par un seul modèle tout simplement parce que les données sont souvent localisées dans des régions particulières de l'espace d'entrée. Il est donc intéressant de diviser l'espace en sous-régions pour avoir une représentation plus précise de la répartition des données et, connaissant mieux les données, il devient plus facile de les utiliser.
Jacobs et al. ont défini un modèle spécifique pour ce type de problème gouverné par le principe de « diviser pour régner » [Jacobs *et al.*, 1991]. En plus d'utiliser un modèle spécifique par région (l'expert), la procédure les répartit dans l'espace et un superviseur, disons le portier pour traduire le terme anglais qui lui est attribué « gating », recueille les différentes réponses des experts et leur accorde plus ou moins de crédit en fonction de la réputation de l'expert c'est-à-dire par rapport à l'adéquation entre le domaine de prédilection de l'expert et le type de situation rencontrée.

L'algorithme d'apprentissage a pour but de permettre au portier de faire une bonne décomposition de l'espace d'entrée en M sous-régions et de former un spécialiste sur chaque région. Le portier tient à jour une liste constituée d'un représentant par région. Ces représentants sont des vecteurs de référence de même dimension que l'espace d'entrée. La similitude entre un exemple $\mathbf{x}$ et un représentant $\mathbf{v}$ est mesurée par un produit scalaire $\mathbf{x}^T\mathbf{v}$. Le partitionnent de l'espace d'entrée se fait donc par des hyperplans $\mathbf{x}^T(\mathbf{v}_i - \mathbf{v}_{i'}) = 0$ où cette fois-ci $\mathbf{x}^T$ désigne le vecteur générique $(x_1,x_2,...,x_d)$ (avec $i,i' \in \{1,2,...,M\}$ et $i \neq i'$). Mais les frontières de séparation entre les régions, et donc les critères d'intervention des experts, ne sont pas si strictes. Les données peuvent appartenir à plusieurs régions. Inspirée de la régression logistique multiple, la fonction *softmax* détermine la probabilité a posteriori des classes [Bridle, 1990a; Bridle, 1990b] (voir figure 3.19). L'utilisation de la fonction *softmax* pour juger de la confiance à donner à chaque expert fait que plusieurs experts peuvent collaborer à la décision finale. En fait, ils collaborent tous mais l'avis de certains n'est pas pris en compte. Les régions se recouvrent.
L'apprentissage donne simultanément les modifications à apporter aux poids de tous les experts ainsi qu'aux vecteurs de référence du portier.

La fonction d'erreur est obtenue en prenant l'opposé du logarithme néperien de la vraisemblance.
Dans sa forme la plus simple, le vecteur d'entrée est présenté au portier qui détermine la confiance qu'il a en chaque expert. L'expert avec la plus grande confiance se voit présenter à son tour la forme d'entrée pour produire une réponse qui dans ce cas est également la réponse du système.

Jacobs et al. [Jacobs *et al.*, 1991] ont montré l'efficacité des mélanges d'experts sur un problème de reconnaissance des voyelles. La décomposition de l'espace des voyelles qu'ils ont observée alors était très instructive. C'est cet aspect de découverte de caractéristiques qui nous a poussé à présenter cette technique.
Jordan et Jacobs [Jordan and Jacobs, 1994] ont étendu le modèle de mélange d'experts à

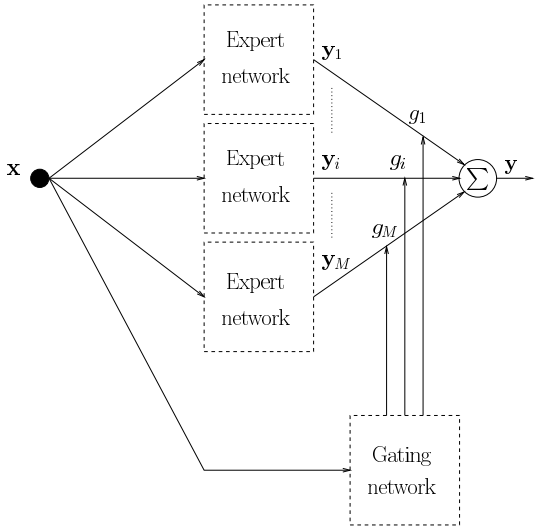


FIG. 3.17 – *Illustration de l'architecture d'un mélange d'experts. La pondération g_i de la prédiction y_i de chaque expert i est calculée par un réseau prédictif (*gating network*) en fonction de la forme courante x.*

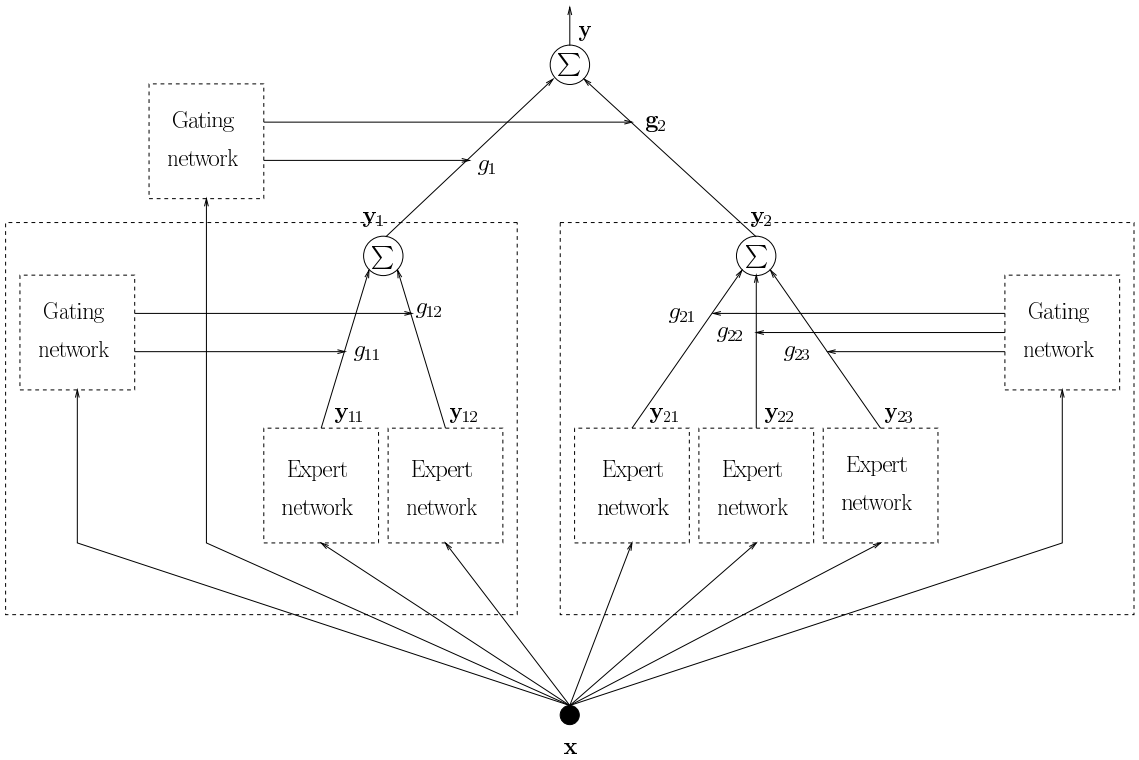


FIG. 3.18 – *Illustration de l'architecture d'un mélange hiérarchique d'experts. La pondération g_i de la prédiction y_i de chaque mélange d'expert (voir figure 3.17) est calculée par un réseau prédictif (*gating network*) en fonction de la forme courante x.*

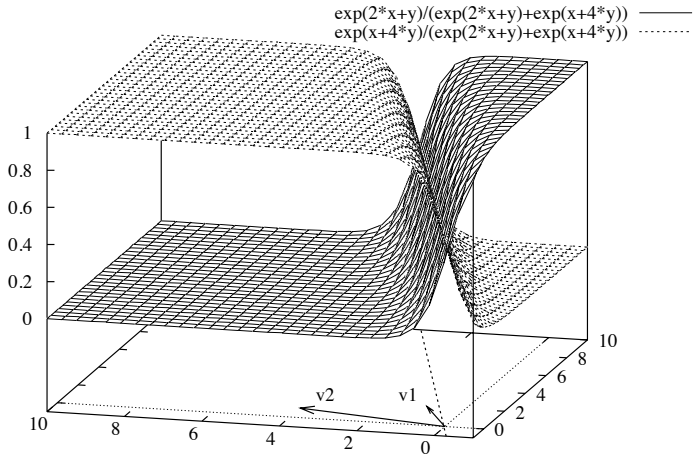


FIG. 3.19 – *La fonction softmax calcule la contribution de chaque expert à la décision finale en fonction de la similitude (i.e. du produit scalaire) du vecteur de référence v_i de l'expert avec la forme présentée.*

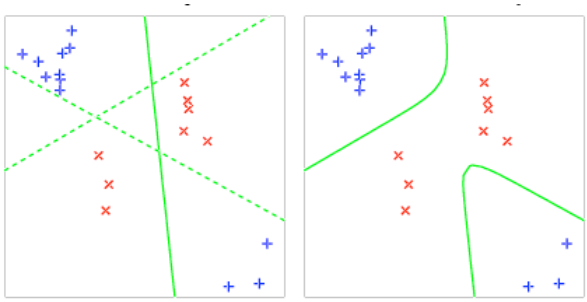


FIG. 3.20 – *Illustration de l'utilisation d'un mélange de deux experts sur un exemple de classification constitué de deux classes (*x *et* +*). Sur la figure de gauche, la ligne pleine marque la frontière de séparation en deux sous-problèmes posée par le portier. Les lignes en pointillé représentent les frontières de discrimination des experts. La figure de droite montre les frontières de décision obtenues par le modèle de mélange (d'après Moerland [Moerland, 2000]).*

une architecture hiérarchique. Chaque expert peut lui même être le résultat d'un mélange d'experts (voir figure 3.18). Cette technique conduit à l'obtention d'arbre comparable à d'autres algorithmes qui utilisent également le principe de « diviser pour régner » comme CART [Breiman *et al.*, 1984], MARS [Friedman, 1991] et ID3 [Quinlan, 1986] (voir sections 2.3.2 et 2.3.3). Mais ici, la séparation n'est pas stricte ce qui constitue un avantage du point de vue de la diminution de la variance puisse le nombre d'exemples appartenant à chaque région est plus important dans ces conditions.
L'apprentissage a également été amélioré par l'adaptation de l'algorithme EM (Expectation-

Maximization) [Dempster *et al.*, 1977] à ce type de modèle [Jordan and Jacobs, 1994].

La procédure est ascendante ce qui sollicite tout le système quelle que soit la région. Une procédure descendante serait plus appréciable.

3.2.3 Orthogonal Weight Estimator

Les perspectives offertes par l'utilisation des unités sigma-pi décrites ci-dessous et les observations neurobiologiques présentées ensuite sont à l'origine du développement d'un modèle connexionniste modulaire par Nicolas Pican [Pican *et al.*, 1994] : l'orthogonal weight estimator (OWE).

Unités sigma-pi et réseaux d'ordre élevé

Nous avons présenté (page 57) la propriété d'approximation universelle d'un réseau de neurones à une couche cachée. Cette propriété résulte de sa capacité à approcher, avec une précision aussi fine que désirée, n'importe quelle fonction. Mais une modélisation correcte d'une fonction aussi simple que la fonction $f : x \mapsto f(x) = x^2$ nécessite un réseau de grande taille comportant deux couches cachées avec respectivement 16 et 8 neurones [Pican, 1995, pages 75 à 78] (bien qu'une seule couche cachée suffise, l'utilisation de deux couches cachées permet de réduire la taille du réseau [Chester, 1990]). La difficulté rencontrée, avec les fonctions assimilables à des polynômes de degré supérieur à un, réside dans la combinaison faite entre les valeurs pré-synaptiques x_i et les valeurs synaptiques w_i. L'association en question correspond généralement au produit scalaire des deux vecteurs, et aboutit donc à une somme pondérée $... + w_i x_i + ...$ sans terme mixte $x_i x_j$. Pour pallier cette situation, des formes plus complexes d'association ont été définies et sont désignées sous le terme d'unités d'ordre élevé (higher order units) par opposition aux unités additives telles qu'elles ont été définies dans le neurone formel [Rumelhart *et al.*, 1986a]. Pour mieux saisir la supériorité des unités d'ordre élevé, intéressons nous à l'une d'entre elles : l'unité sigma-pi dont la définition est la suivante :

$$\sum_{k} \sum_{i=(i_1,...,i_k)} w_{ij} \Pi x_{i_1} x_{i_2} ... x_{i_k} \tag{3.11}$$

L'unité j reçoit une somme pondérée de produits d'entrées. k est le nombre d'entrées x impliquées dans la combinaison i.

A l'aide de ce type d'unités, la modélisation parfaite de la fonction x^2 peut être obtenue en utilisant une seule unité dont l'unique combinaison fait intervenir deux fois l'entrée x (figure 3.21) i.e. :

$$x^2 \longleftrightarrow \begin{cases} i = 1 \\ j = 1 \\ w_{11} = 1, x_{1_1} = x, x_{1_2} = x \end{cases}$$

De même, le problème de discrimination lié à la fonction *xor* peut être résolu en utilisant un simple perceptron dont l'unité de la couche de sortie est du second degré (figure 3.22). On a :

$$x - 2xy + y \longleftrightarrow \begin{cases} i = 3 \\ j = 1 \\ w_{11} = 1, x_{1_1} = x \\ w_{21} = -2, x_{2_1} = x, x_{2_2} = y \\ w_{31} = 1, x_{3_1} = y \end{cases}$$

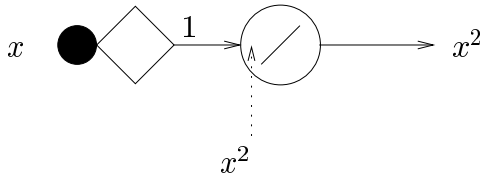


FIG. 3.21 – *Exemple de réseau sigma-pi modélisant la fonction x^2.*

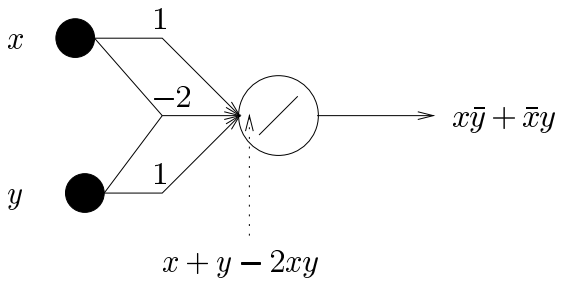


FIG. 3.22 – *Exemple de réseau sigma-pi modélisant la fonction xor.*

Les réseaux munis d'unités sigma-pi sont dit d'ordre élevé [Rumelhart *et al.*, 1986b]. Ils permettent d'obtenir une fonction polynômiale de degré supérieur ou égal à 1. Les unités additives sont un cas particulier des unités sigma-pi. Pour s'en rendre compte, il suffit de poser dans la formule désignant une unité sigma-pi (éq 3.11) $k = 1$ et $x_{i_1} = x_i$ pour aboutir à la formule d'une unité additive :

$$\sum_i w_{ij} x_i$$

j étant l'indice de l'unité.

L'utilisation d'unités plus complexes que les unités additives, généralement adoptées dans les réseaux connexionistes, permet de simplifier le modèle mais surtout elle permet une meilleure approximation de la fonction à modéliser. Un réseau capable de reproduire une fonction polynômiale de degré élevé est à même de réaliser des performances similaires à l'association de plusieurs réseaux classiques avec en plus une simplification du fonctionnement et une amélioration de la cohérence des réponses. La sous-section suivante apporte une justification biologique à la définition et à l'utilisation des unités sigma-pi.

Observations biologiques

Les unités additives, classiquement manipulées dans les réseaux connexionnistes, sont issues de la formalisation d'un neurone biologique par un physicien, McCulloch, et un biologiste, Pitts [McCulloch and Pitts, 1943]. Elles utilisent des connexions axo-dendritiques observées en nombre dans le cortex. En termes généraux, une connexion relie par l'intermédiaire de la synapse un neurone à un autre. Le neurone situé en amont de la synapse par rapport au sens de l'influx nerveux est appelé pré-synaptique, le neurone situé en aval de la synapse est appelé neurone post-synaptique. Dans le cas d'une connexion axo-dendritique, La synapse relie l'axone du neurone pré-synaptique à une dendrite du neurone post-synaptique (figure 3.23). Des recherches en neurobiologie ont montré qu'il existe d'autres types de connexions entre neurones telles que les connexions axo-somatiques (transmission directe de l'activité pré-synaptique au soma), axo-axoniques (transfert d'activité post-synaptique à un autre axone) et axo-synaptiques (connexion d'une synapse sur une autre synapse modulant ainsi son efficacité) (figure 3.23) [Delacour, 1987] [Chen, 1983] .

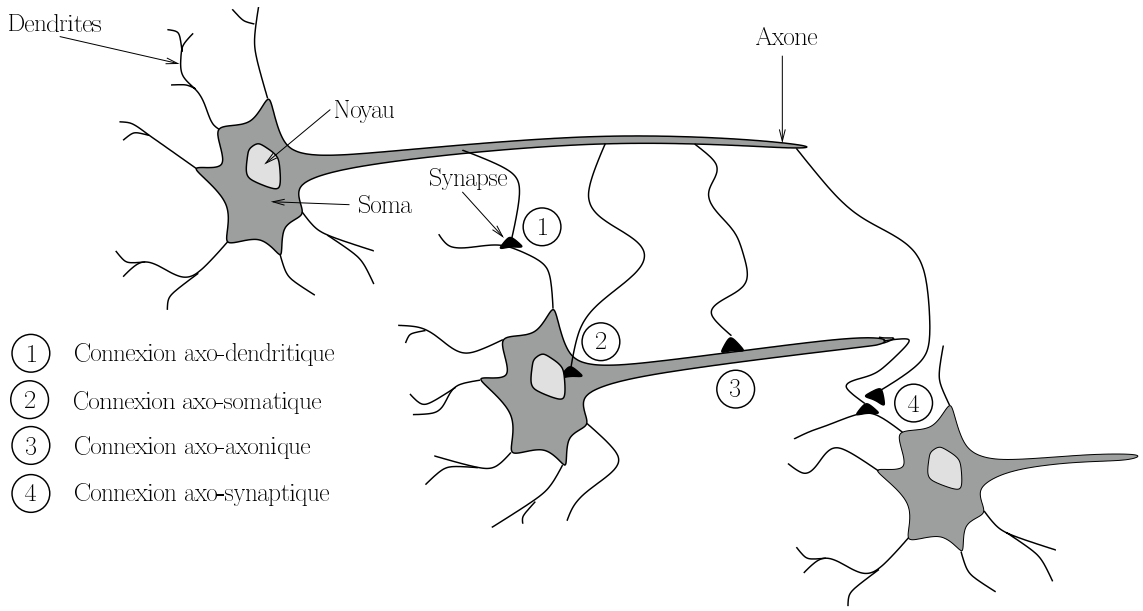


FIG. 3.23 – *Différentes connexions biologiques.*

Dans les réseaux de neurones artificiels, les connexions entre neurones sont généralement de type axo-dendritique. L'activation d'un neurone afférent j est pondérée par l'efficacité de sa synapse, c'est-à-dire le poids de sa connexion w_{ij}, et transmise au neurone suivant i (voir Figure 3.24).

Une connexion axo-synaptique met en jeu trois neurones (i,j,k) avec une connexion axo-dendritique S_{ij} entre j et i et la terminaison synaptique du neurone k connectée sur la synapse S_{ij} (voir Figure 3.24). Le principe consiste à augmenter la probabilité de libération de neuromédiateurs de la synapse S_{ij} dans son espace synapto-dendritique par l'action du neurone k appelé neurone modulateur. Le calcul de l'activation du neurone i peut alors se traduire par l'équation suivante [Pican, 1996] :

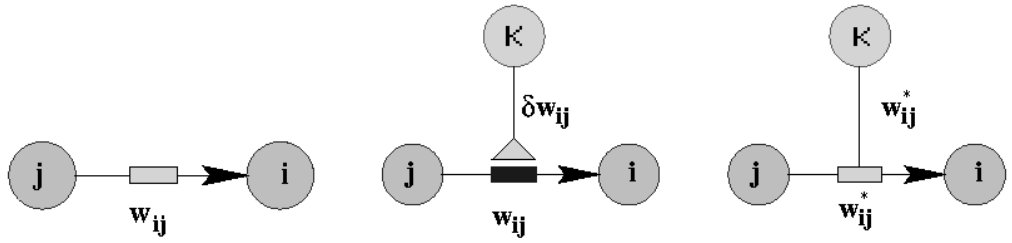


FIG. 3.24 – *Connexions axo-dendritique classique (a) et axo-synaptiques de ODWE (b) et de OWE (c).*

$$Y_i = F_i((w_{ij} + \delta w_{ij})Y_j) \tag{3.12}$$

qui se généralise par :

$$Y_i = F_i(\sum_J (w_{ij} + \delta w_{ij})Y_j) \tag{3.13}$$

dans laquelle δw_{ij} représente la modulation (sortie du neurone modulateur k), w_{ij} le poids de la connexion axo-synaptique entre j et i supposé figé (i.e. le poids statique de la synapse S_{ij}), Y_i et Y_j l'activation des neurones i et j et F_i la fonction de transfert du neurone i.

Description du modèle

L'idée, inspirée des connexions axo-synaptiques (figure 3.23), suggère que les paramètres du modèle pourraient être ajustés au cours de son utilisation en fonction du contexte. En termes connexionnistes, nous dirions que l'efficacité d'une synapse, c'est-à-dire la valeur d'un poids, pourrait être modulée par l'influence d'un autre neurone. Dans ces conditions, l'utilisation d'un modèle dont les paramètres varient équivaudrait à utiliser plusieurs modèles. Le modèle deviendrait donc hautement adaptatif puisqu'il construirait une fonction adaptée à chaque situation.

Architecture. Dans le cas courant, la phase d'apprentissage d'un réseau connexionniste abouti à la définition d'un seul modèle puisque la valeur des poids est fixée après convergence et demeurera constante pendant la phase d'utilisation. L'orthogonal weight estimator (OWE) est un réseau connexionniste prédictif dont l'unique sortie définit le poids d'une connexion (voir figure 3.25). En affectant un OWE à chaque connexion du réseau principal, il est possible de contrôler les paramètres du modèle au cours de la phase d'utilisation. Les OWEs sont des perceptrons multicouches. Ces derniers ont de bonnes performances en prédiction. Cela permet également de conserver une homogénéïté au système. Les modèles prédictifs présentent une architecture similaire, aucune règle universelle ne permettant d'identifier les poids les plus difficiles à ajuster. De plus, dans le cas d'un fonctionnement parallèle, si la complexité des OWEs est identique, tous les poids du réseau principal seront connus en même temps. Le réseau principal peut être un réseau

connexionniste quelconque. Pican a appliqué le principe des OWEs aussi bien à un réseau multicouches supervisé [Pican and Alexandre, 1994a] qu'aux cartes auto-organisatrices de Kohonen [Pican, 1997]. En effet, puisque le réseau principal est un réseau connexionniste quelconque, quel que soit son algorithme d'apprentissage, rétro-propagation du gradient ou autre, il est possible de connaître la modification à apporter à la valeur de chaque paramètre. Cette modification à apporter représente l'erreur de prédiction faite par un modèle prédictif sur le paramètre qu'il caractérise. Par répercutions, il est ensuite possible de corriger la valeur des poids de chaque réseau prédictif en fonction de sa contribution à l'erreur de prédiction telle qu'on le ferait pour un réseau classique.

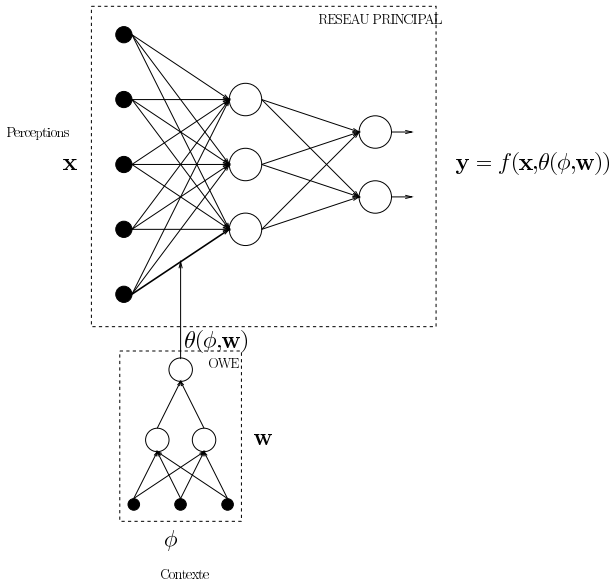


FIG. 3.25 – *Illustration d'une association OWE-MLP. Le contexte ϕ est présenté aux réseaux OWEs (dont la figure ne présente qu'un exemplaire) pour prédire la valeur d'un poids du réseau principal. Ensuite, la forme courante* $\mathbf{x}$ *est présentée au réseau principal pour obtenir de manière classique la valeur de sortie* $\mathbf{y}$.

Répartition des données. Les informations d'entrée se répartissent en deux catégories :

1. Les informations contextuelles fournissent des informations qui permettent de définir les caractéristiques globales du modèle. Elles constituent les variables d'entrées des OWEs. Tous les OWEs ont les mêmes entrées.
2. Les informations perceptives permettent d'affiner la réponse du modèle. Elles constituent les entrées du réseau principal.

Principe de fonctionnement.

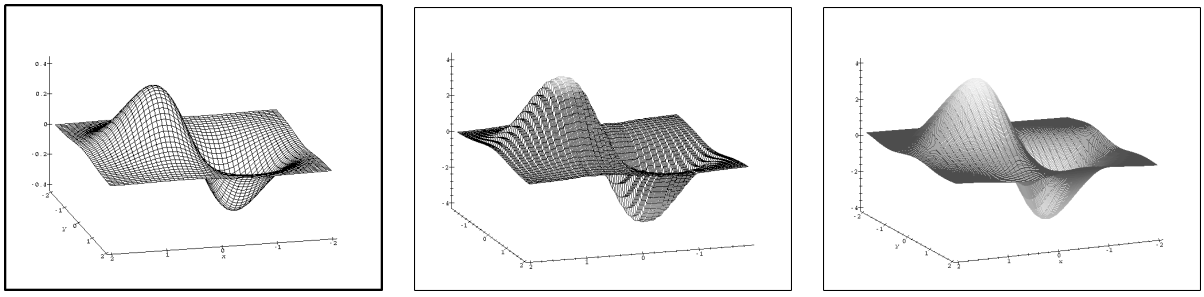
1. Les informations contextuelles sont présentées à tous les OWEs.
2. Les OWEs calculent leur sortie respective. Les poids du réseau principal sont donc déterminés.

3. Les informations perceptives sont présentées au réseau principal.
4. Le réseau principal calcule ses sorties.

Principe d'apprentissage.

1. La règle d'apprentissage fournit la correction à apporter à chaque poids du réseau principal (la correction à apporter à un poids est en fait l'erreur de prédiction de son OWE).
2. L'algorithme de rétro-propagation du gradient détermine les modifications à apporter aux poids.
3. La règle d'apprentissage modifie les poids des OWEs.

Les premiers travaux de Nicolas Pican l'on conduit à étudier les capacités d'interpolation d'un tel modèle à partir d'un ensemble discrétisé de contextes (figure 3.26). L'heuristique de l'approche est basée sur le postulat que s'il existe une continuité de comportements en relation avec une continuité de jeux de paramètres alors les poids du modèle contextuel peuvent reproduire cette continuité. Les avantages d'un modèle capable de reproduire le comportement de plusieurs modèles sont déjà intéressants en soi mais le principal intérêt provient de l'interpolation des paramètres qui correspondent à une multitude de contextes intermédiaires. La continuité permet d'avoir un meilleur comportement du système pour des situations limitrophes à deux contextes ou plus, ou encore éloignées de la situation de base caractéristique du contexte. Un contexte particulier détermine un jeu de paramètres spécifiques. Donc si le contexte ne change pas, il n'est pas nécessaire de recalculer les poids du réseau principal ce qui permet de gagner en rapidité de calcul.



(a) $f(x,\phi) = 10\phi e^{-\phi^2+x^2}$

(b) Modélisation de $f(x,\phi)$ par plusieurs MLPs spécialisés sur des intervalles distincts $[\phi_i,\phi_f]$.

(c) Modélisation de $f(x,\phi)$ par un MLP+OWEs. ϕ est présenté aux OWEs puis x l'est au MLP.

FIG. 3.26 – *Continuité inter-contextuelle obtenue par OWEs.*

Le modèle neuronal modulaire qui vient d'être présenté a été établi à partir de données neurobiologiques soulignant le rôle modulateur de certains neurones sur l'apprentissage d'autres neurones. Aujourd'hui, ce modèle est assez éprouvé pour être utilisé simplement et efficacement sur des applications industrielles de vraie grandeur.

Modèle	Architecture	Phase	$\bar{e}$ (dB)	σ
MLP	(32x10x1)	*Apprentissage*	4.18	3.38
		Test	4.21	3.43
OWE+MLP	(7x4x1)+(25x10x1)	*Apprentissage*	3.56	4.55
		Test	4.01	5.18
Multi-MLPs	9 MLPs (32x10x1)	*Apprentissage*	3.36	4.45
		Test	3.48	4.57

TAB. 3.7 – *Performances d'un modèle OWE sur le problème de prédiction du champ radioélectrique.*

Expérimentation : Comme nous avons pu le voir dans les sections 2.4 et 2.5.6, le problème de prédiction du champ radioélectrique est fortement contextuel car il dépend de la nature de l'environnement. Sa complexité nécessite la considération de paramètres spécifiques à la zone d'étude pour influencer la prédiction comme cela a été fait avec l'utilisation de modèles spécialisés sur une partie des données.

Le tableau 3.7 montre clairement que, par l'utilisation du modèle OWE, les performances de réseaux classiques (MLP) sont améliorées, et qu'elles s'approchent de celles obtenues avec une architecture Multi-MLPs. Par contre, deux avantages sont en faveur de l'utilisation des OWEs : ils ne nécessitent qu'une seule phase d'apprentissage et incluent une faculté d'interpolation entre les contextes [Bougrain *et al.*, 1998; Bougrain, 1997b].

Les avantages en termes de simplification, d'adaptation, d'interpolation et le parallèle possible avec un fonctionnement neurobiologique rendent le modèle OWE intéressant et utile.

3.3 Une construction modulaire dynamique

La première partie de ce chapitre nous a permis de soulever le problème central du surapprentissage intervenant intrinsèquement dans la qualité finale d'un modèle et d'évaluer les techniques qui permettent de circonscrire ce problème. Les méthodes incrémentales se heurtent avec difficulté à la mise au point du critère d'arrêt. Bien souvent, l'accroissement de la taille du système implique une amélioration des performances, et finalement le système s'arrête pour avoir atteint la limite des ressources dont il dispose et non pas pour avoir satisfait à un véritable critère d'arrêt basé sur un bon compromis entre performances et complexité du système. Non seulement les méthodes d'élagage ne présentent pas cette fâcheuse tendance à la gloutonnerie, mais elles permettent en plus de trouver le système minimal nécessaire à un certain niveau de performance, d'améliorer les performances en réduisant le bruit interne dû à un système surparamétré et d'obtenir des informations sur le problème traité au travers de la sélection progressive des variables et de la lecture des combinaisons les plus importantes qui résultent du processus d'élagage.
La seconde partie, consacrée à l'approche modulaire, nous a permis de mettre en évidence l'intérêt de partitionner le problème, ou mieux de laisser le système partitionner le

problème, pour obtenir un réseau spécialisé sur un sous-espace du problème. Ici aussi, la structure finale du réseau est ajustée à la complexité du problème. Certains sous-espaces plus difficiles à traiter se voient attribuer un plus grand nombre d'experts. La façon dont le problème a été décomposé est une source d'informations, mais la principale difficulté réside justement dans le partitionnement du problème et la connexion des modules. Nous avons vu que le modèle OWE permet d'éviter la multiplication des modèles spécialisés tout en conservant, et même en augmentant, l'adaptation du réseau à la particularité de la situation.
Dans cette section, nous allons exploiter les observations faites sur les réseaux appartenant à ces deux approches pour définir un modèle original qui combine l'approche dynamique et l'approche modulaire, ceci afin d'augmenter les avantages d'une construction automatique de la structure du modèle en fonction de la tâche à réaliser.

Le point de départ de notre modèle est l'architecture du modèle contextuel OWE. L'idée d'adapter les poids du réseau selon le contexte de la situation est intéressante mais la véritable plus-value de ce modèle par rapport à un ensemble de réseaux entraînés sur des contextes différents est sa capacité à interpoler les contextes, et donc le paramétrage de la fonction réalisé à partir des contextes connus par le modèle. De plus, la taille de l'architecture est indépendante du nombre de contextes. La réalisation de cette idée sous la forme d'une architecture modulaire où chaque poids est évalué par un réseau de neurones prédictif permet une bonne compréhension de la répartition des tâches. Les mécanismes internes sont clairs. De plus, le principe de fonctionnement est applicable à tous les réseaux de neurones. A partir de ce modèle, nous présentons dans les sections suivantes les améliorations possibles permettant d'aboutir à un nouveau modèle contextuel adapté à l'extraction de connaissance que nous avons baptisé Pruned Orthogonal Weight Estimator (POWE).

3.3.1 Le problème du partage des variables d'entrée du OWE

Jusqu'à présent, nous avons établi l'intérêt d'une approche contextuelle (sections 2.4, 3.2.1, 3.1.2, 3.2.3) mais sans connaissance précise des variables pertinentes à utiliser pour caractériser le contexte. Le modèle OWE définit le contexte *a priori* et ni les différentes méthodes de partitionnement ni les réseaux récurrents ne permettent d'extraire un lot de variables significatives. Pourtant, un modèle contextuel tel que OWE sera d'autant plus performant que le contexte sera adapté au problème. Nous allons tenter de répondre à la question du choix des variables contextuelles pour des problèmes multidimensionnels.

– Tout d'abord, le choix des variables contextuelles peut se faire d'après leur nature après discussion avec les experts du domaine.

  Expérimentation : Pour étudier le nombre de variables pertinentes qui décriront le contexte dans le modèle OWE, nous avons fait varier le nombre de variables utilisées pour définir le contexte, d'un nombre restreint (8) à la quasi totalité (28). Les résultats montrent qu'un contexte constitué d'un petit nombre de variables bien choisies est le plus performant (tableau 3.8). Rappelons que les variables sont

Modèle	Architecture	Phase	$\bar{e}$ (dB)	σ
MLP	(32x10x1)	*Apprentissage*	4.18	3.38
		Test	4.21	3.43
OWE+MLP (petit contexte)	(8x8x1)+(24x8x1)	*Apprentissage*	3.82	4.84
		Test	4.07	5.23
OWE+MLP (large contexte)	(28x6x1)+(4x4x1)	*Apprentissage*	28.72	36.79
		Test	32.88	42.27

TAB. 3.8 – *Performances d'un modèle OWE sur le problème de prédiction du champ radioélectrique en fonction de la taille du contexte.*

Modèle	Architecture	Phase	$\bar{e}$ (dB)	σ
MLP	(32x10x1)	*Apprentissage*	4.18	3.38
		Test	4.21	3.43
OWE+MLP (contexte fortement corrélé)	(7x4x1)+(25x10x1)	*Apprentissage*	3.56	4.55
		Test	4.01	5.18
OWE+MLP (contexte faiblement corrélé)	(9x4x1)+(23x10x1)	*Apprentissage*	29.57	37.75
		Test	34.15	43.93

TAB. 3.9 – *Performances d'un modèle OWE sur le problème de prédiction du champ radioélectrique en fonction de la composition du contexte.*

réparties entre le contexte et les perceptions, ce qui signifie que lorsque le contexte est de taille réduite, les variables perceptives sont plus nombreuses. Il n'est donc pas possible de conclure explicitement puisque nous avons deux variables indépendantes [Bougrain *et al.*, 1998].

– D'autre part, nous voulons vérifier l'hypothèse selon laquelle les variables les plus corrélées à la valeur à prédire sont de bonnes variables contextuelles.

Expérimentation : Puisque nous avons vu qu'un petit nombre de variables pouvait constituer un contexte pertinent pour OWE, seules les 7 variables les plus corrélées à la valeur à prédire ($|\text{corrélation}| > 0.56$) et les 9 variables les moins corrélées ($|\text{corrélation}| < 0.074$) seront utilisées pour définir les deux contextes de test. Les résultats confirment que les variables fortement corrélées à la valeur à prédire constituent un meilleur choix de contexte pour le modèle OWE que les valeurs subjectives choisies par les experts (tableau 3.9). De plus, les variables peu corrélées sont un très mauvais choix. Elles ne permettent pas d'obtenir une bonne spécialisation du modèle OWE [Bougrain *et al.*, 1998].

Nous avons cherché à définir objectivement quelles variables d'un problème doivent être considérées comme étant contextuelles. Les résultats reportés ici vont dans le sens d'une meilleure connaissance de ces critères et devraient être approfondis pour mieux les appréhender.

3.3.2 Intérêts d'élaguer le modèle OWE

Pour éviter le problème de la répartition des variables d'entrée en variables perceptives et variables contextuelles, toutes les variables peuvent être placées à la fois dans l'ensemble perceptif et l'ensemble contextuel. Cette solution présente un autre avantage, celui de pouvoir modéliser des fonctions à l'aide de tous les couples de produit des entrées. Par contre, la taille du réseau augmente considérablement dans ces conditions. De plus, il est peu probable que tous les poids du modèle principal nécessitent la totalité des variables contextuelles pour être prédits. L'application d'une procédure d'élagage nous fournira la liste des variables contextuelles nécessaires à la prédiction de chaque poids. Et nous conservons tous les avantages liés à la minimisation de l'architecture en fonction de la tâche à réaliser.

Expérimentation : Pour être appliquée au problème prédictif du champ radioélectrique, l'architecture de départ du POWE (Pruned OWE) est constituée pour le réseau principal d'une forme 32x8x1, soit 264 paramètres à estimer, nécessitant autant de réseaux prédictifs d'architecture 32x2x1. Les 18768 connexions sont supprimées par groupe de 10%.

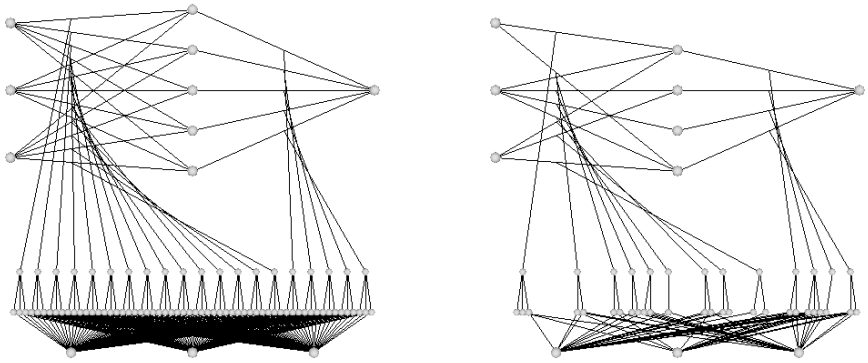
Modèle	Phase	$\bar{e}$ (dB)	σ	Q_4 (%)	Q_6 (%)	Q_{11} (%)
MLP	*Apprentissage*	4.18	3.38	56.9	75.7	95.6
	Test	4.21	3.43	56.8	75.3	95.4
POWE (avant élagage)	*Apprentissage*	4.02	3.31	56.0	76.4	94.7
	Test	4.08	3.40	56.1	75.8	94.5
POWE (après élagage)	*Apprentissage*	4.15	3.36	57.3	75.0	94.2
	Test	4.17	3.37	57.1	74.6	93.2

TAB. 3.10 – *Performances d'un modèle POWE sur le problème de prédiction du champ radioélectrique.*

Nous avons observé, au cours de l'élagage, la même évolution des performances pour notre modèle que celle que nous avons décrite dans la section 3.1.3 pour un réseau classique, comme le perceptron multicouches. Les premières connexions supprimées entraînent une diminution du bruit et une amélioration des performances. La situation du modèle après élagage présentée dans le tableau 3.10 correspond, à performances égales, à un réseau dont l'irrégularité des connexions (par rapport à un réseau entièrement connecté) nous permet d'apprécier les combinaisons importantes des variables. Par la suite, le taux de 10% de connexions élaguées est trop élevé pour permettre au réseau de converger efficacement vers l'architecture minimale avec des performances correctes. D'autres investigations doivent être menées pour connaître plus en détails les interprétations possibles du modèle et donc du phénomène dans le cas de la prédiction du champ radioélectrique.

3.3.3 Parallélisation

L'intérêt de l'implantation parallèle des réseaux de neurones a été abordé, entre autres, par Misra [Misra, 1996]. Il s'agit de profiter du formalisme distribué des réseaux connexion-



(a) initialisation (b) état final

FIG. 3.27 – *Illustration d'un réseau POWE. Partant d'une architecture entièrement connectée (a), le modèle contextuel POWE supprime les connexions et les neurones inutiles pour adapter le modèle à la complexité du phénomène à modéliser, puis il supprime les connexions et les neurones les moins pertinents pour effectuer une sélection des variables d'entrée et obtenir une architecture adaptée à l'extraction de règles (b).*

nistes pour répartir le calcul sur plusieurs processeurs, afin d'augmenter la rapidité de calcul et la robustesse du système. Dans le cas du réseau POWE, puisque l'on procède par simplification, la difficulté est engendrée par la grande taille du réseau de départ, nécessaire à une bonne exploration de l'espace des solutions, et par le grand nombre d'opérations, nécessaire à l'obtention de l'architecture optimale (l'ordre de grandeur pour notre tâche de prédiction est au départ de 15000 neurones, 100000 connexions et 50000 données en 32 dimensions). L'apprentissage de ce type de réseaux est donc relativement long, tandis que l'utilisation du réseau final élagué sera très rapide. Le réseau POWE a été implanté à l'aide d'une librairie de développement parallèle spécifique aux réseaux de neurones [Boniface, 2000]. Cette librairie nous offre deux avantages. Premièrement, le temps d'implantation se trouve considérablement réduit par l'utilisation des fonctions offertes. Deuxièmement, il est possible, avec le même programme, d'effectuer l'apprentissage sur machines parallèles MIMD à mémoire partagée, et d'utiliser le réseau émergent sur machines séquentielles classiques.

Expérimentation : L'accélération obtenue sur *origin 2000* en fonctionnement multi-utilisateurs est de deux sur trois/quatre processeurs et de six sur une vingtaine de processeurs [Bougrain and Boniface, 2000]. Au-delà de l'accélération, le gain en temps de calcul est très appréciable (en terme d'heures)(Figure 3.28).

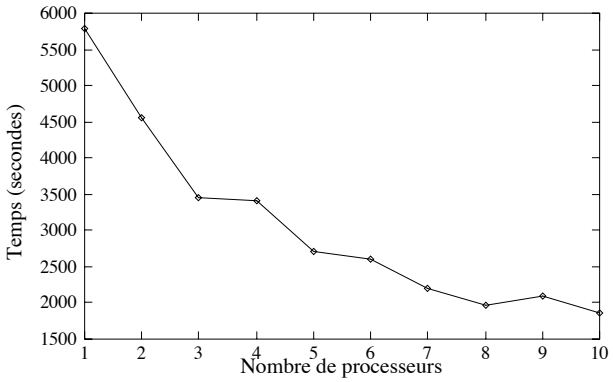


FIG. 3.28 – *Performances en parallélisation du modèle POWE.*

Conclusion

L'application d'une méthode d'élagage, telle que Optimal Brain Damage, à l'architecture modulaire d'un Orthogonal Weight Estimator permet de cumuler les performances de ces deux méthodes dans un nouveau modèle baptisé POWE pour Pruned Orthogonal Weight Estimator. Le modèle POWE permet de réunir les bénéfices d'une architecture minimale et d'un modèle hautement adaptatif, bénéfices qui se traduisent par une meilleure compréhension du problème et par une amélioration des performances en termes de sélection de variables, d'extraction de règles, de rapidité de calcul, de réduction du bruit interne au modèle et d'utilisation réduite des ressources. L'application de la technique d'élagage à un *Orthogonal Weight Estimator* permet de lever certaines contraintes comme la séparation des variables d'entrée en variables perceptives et variables contextuelles, choix qui peut parfois s'avérer arbitraire. Il est possible d'obtenir tous les produits de variables y compris des produits d'une même variable. Enfin l'utilisation du même contexte pour tous les OWEs n'est plus obligatoire. L'implémentation parallèle du modèle diminue le temps d'apprentissage pour que cette méthode puisse déterminer rapidement une architecture de réseaux neuromimétiques construite sur les besoins de la tâche. Une fois l'architecture obtenue, la phase d'utilisation peut être exécutée soit sur machines parallèles, soit sur machines séquentielles.

Conclusion générale

Actuellement, les informations disponibles dans les banques de données posent plusieurs problèmes d'exploitation. Premièrement, les banques de données sont, bien souvent, pas ou peu organisées. L'abondance des informations, due principalement au développement de nouveaux médias tels qu'internet, rend difficile leur organisation et diminue fortement la possibilité de trouver rapidement une information. Deuxièmement, nous ne savons pas quelle confiance accorder à une information. Certaines sont bruitées, d'autres ne correspondent pas tout à fait à l'information pertinente que nous recherchons. Souvent, les variables observables, d'où les données sont issues, ne sont qu'une représentation bruitée des variables réellement impliquées dans un phénomène. Troisièmement, nous avons de plus en plus besoin d'extraire de la connaissance des données pour expliquer un phénomène. Les tâches à réaliser sont de plus en plus complexes et les connaissances théoriques et empiriques des experts du domaine ne suffisent plus à expliquer un phénomène. Tous ces problèmes d'exploitation proviennent de la méconnaissance des données et des phénomènes auxquels elles sont liées.

Une analyse des données et de leur implication dans la modélisation d'un phénomène peut permettre d'acquérir de nouvelles connaissances, et, en conséquence, d'améliorer les performances d'une tâche qui fait appel à elles. Ce type d'analyse peut être réalisé par des outils de statistique et d'intelligence artificielle (apprentissage, algorithmes génétiques, réseaux de neurones, etc.). Les réseaux de neurones sont de bons outils de modélisation (efficaces, simples à mettre en œuvre et rapides) mais ils ont la réputation d'être difficiles à interpréter. L'objectif de notre travail était, dans un premier temps, de réunir divers travaux théoriques pour étudier les points communs entre certains réseaux de neurones classiques et des méthodes statistiques d'analyses de données, afin de mieux comprendre le fonctionnement des réseaux de neurones et de valider leur utilisation en tant qu'outils d'acquisition de connaissance ; puis, dans un deuxième temps, d'étudier et de comprendre les particularités des réseaux connexionnistes à la lumière de ce qui précède afin de les exploiter en concevant un nouveau modèle d'extraction de connaissances.

Notre démarche pour y parvenir fut d'examiner quels sont les réseaux de neurones qui permettent de par leur mode de fonctionnement d'extraire de la connaissance.

Dans le chapitre 1, nous avons défini, à travers l'exemple de la prédiction du champ radioélectrique pour la téléphonie mobile, une problématique représentative des besoins et des difficultés rencontrés dans les problèmes réels pour lesquels le nombre de données est très élevé et la connaissance du domaine insuffisante. Les banques de données rassemblées pour ce problème nous ont fourni les éléments nécessaire à l'illustration de nos investigations tout au long de notre démarche.

Dans le chapitre 2, nous nous sommes intéressés aux réseaux de neurones qui structurent la connaissance par simple adaptation de leurs poids. Pour montrer quels types de connaissance peuvent être extraits de ces réseaux, nous avons examiné leur fonctionnement à la lumière de méthodes statistiques d'analyse de données, telles que des méthodes d'analyse factorielle et de partitionnement. Ainsi, il a été montré que les réseaux connexionnistes sont une classe particulière des modèles de régression et de discrimination. Les réseaux supervisés présentent une architecture qui s'adapte facilement à toutes sortes de régression. Ainsi, nous rappelons par exemple qu'un réseau comportant plusieurs entrées constitue de manière naturelle un modèle de régression multiple et qu'avec plusieurs sorties, c'est un modèle de régression multivarié. Le choix de la fonction d'activation (linéaire, sigmoïdale ou polynômiale) définit une régression du même nom. De plus, nous avons détaillé comment un réseau auto-associatif peut réaliser une analyse en composantes principales linéaire ou non linéaire. Les réseaux connexionnistes non supervisés tels que les cartes auto-organisatrices se comportent comme la méthode de partitionnement des *k-means*. Nous avons également souligné divers travaux qui font état d'extensions possibles des méthodes d'analyse de données grâce au formalisme des méthodes connexionnistes, comme par exemple la possibilité d'avoir des versions non linéaires de méthodes statistiques linéaires, ou d'obtenir des informations topologiques des classes.

Dans le chapitre 3, l'intérêt des réseaux de neurones classiques comme outils d'analyse de données et d'extraction de connaissances étant établi, nous avons étudié des modèles connexionnistes qui organisent leur architecture, de manière dynamique ou modulaire, pour mieux modéliser la tâche. Ces réseaux structurent l'information en construisant une famille de fonctions plus à même de modéliser la tâche, et non plus uniquement en adaptant leurs poids. La compréhension des avantages et des inconvénients des deux approches nous a amené à sélectionner un représentant particulier de chaque approche pour les réunir dans un nouveau modèle qui bénéficie des avantages de chacun et élimine certains inconvénients. Nous sommes partis du modèle modulaire OWE de Pican [Pican and Alexandre, 1994b; Pican *et al.*, 1994]. Plutôt que d'utiliser des poids fixes et des combinaisons de poids linéaires, les paramètres du modèle, appelé ici modèle principal, sont calculés par un ensemble de réseaux prédictifs. L'utilisation d'un modèle dont les paramètres varient équivaut à utiliser plusieurs modèles. Le modèle est donc hautement adaptatif. Il construit une fonction adaptée à chaque situation. De nombreux avantages découlent de la conception modulaire, qui consiste à utiliser des prédicteurs connexionnistes pour calculer les paramètres du modèle en charge de la modélisation de la fonction proprement dite. Les modèles prédictifs viennent se greffer sur les connexions du modèle principal. Ce principe peut s'appliquer à n'importe quel réseau connexionniste (dont les plus utilisés : perceptron multicouches et cartes auto-organisatrices de Kohonen) pour des tâches aussi différentes que la discrimination, la catégorisation, la prévision, le contrôle et le diagnostic. De même qu'un réseau connexionniste classique réalise une interpolation des situations qui lui ont été présentées, le réseau prédictif réalise une interpolation des contextes qui lui ont été présentés. Il est donc possible d'obtenir une continuité de la prédiction d'un contexte à l'autre, ce qui revient à obtenir une continuité entre les classes, si le contexte est la classe d'appartenance d'une forme. Mais cette conception présentait un inconvénient que notre modèle a supprimé. Tout d'abord, il était nécessaire de faire une séparation *a priori* des données en données perceptives et données contextuelles. Peu d'études ont été faites à ce

sujet et celles que nous avons réalisées ne permettent pas d'aboutir à un précepte théorique. Or la définition d'un contexte adéquat est la clé du succès de la méthode. Dans notre modèle, toutes les variables sont utilisées au départ comme entrées perceptives et contextuelles. Notre algorithme inclut une technique de raffinement du modèle pour bénéficier de plusieurs avantages. L'utilisation d'une méthode d'élagage aussi éprouvée que *optimal brain damage* permet de déterminer rapidement les connexions les moins utiles à la réalisation de la tâche. Les effets bénéfiques de cette approche dynamique se traduisent d'abord par la suppression de perturbations internes au modèle générées par la présence de variables inutiles et de combinaisons trop complexes. On limite également de cette façon le phénomène de surapprentissage. Ensuite, la suppression de connexions entraîne la suppression de certains neurones, ce qui permet de réduire l'architecture du modèle à son minimum pour des contraintes de performance déterminées *a priori*. De plus, la suppression de neurones d'entrée engendre une sélection de variables et constitue une première forme d'extraction de connaissances. Les variables les plus utiles à la perception ou au contexte apparaissent peu à peu. Il est possible d'extraire des règles simples à partir de la version minimale du modèle, ce qui constitue une autre forme d'extraction de connaissances. Ces apports de connaissances sont bénéfiques à la compréhension du phénomène modélisé. De plus, la réduction de la taille du modèle permet un calcul plus rapide de la valeur de sortie du système, d'où une réduction du temps de calcul. L'association de modèles *OWE*s à un réseau principal définit au départ une architecture plus complexe que celle du réseau principal seul. Mais la modulation des poids par les *OWE*s permet de réduire plus fortement l'architecture du réseau principal. Donc, dans le cas où plusieurs situations appartenant au même contexte sont présentées les unes après les autres, les paramètres du modèle principal ne bougent pas. Le calcul de la réponse du système peut alors devenir plus rapide que celle du modèle principal élagué.

L'implantation de notre modèle sur machine parallèle (*origin* 2000) nous a permis de diminuer considérablement le temps d'apprentissage. Elle a été facilitée par l'utilisation d'une librairie spécifique aux réseaux connexionnistes, et qui permet de compiler et d'exécuter notre algorithme, indifféremment sur machine séquentielle ou sur machine parallèle.

Nous avons appliqué notre démarche au problème de prédiction en téléphonie mobile. Suite à cette étude, France Télécom Mobile[7] a reconnu l'intérêt d'intégrer les réseaux de neurones artificiels aux outils d'aide à l'ingénierie qu'il utilise pour prédire l'atténuation du champ radioélectrique et aider à mieux comprendre le phénomène complexe de la propagation des ondes dans des milieux hétérogènes. Nos travaux sont donc exploités commercialement aux niveaux national et international. Du point de vue des performances, nous avons apporté des améliorations successives à la prédiction du champ, réduit le pourcentage de fortes erreurs et l'intervalle de confiance, et nous avons montré qu'il est possible d'avoir des prédictions de situations géographiquement proches plus cohérentes les unes avec les autres. Du point de vue de l'acquisition de connaissances, nous avons évalué la pertinence des variables d'entrée du système et opéré une sélection des plus utiles pour la tâche à modéliser. L'intérêt de notre démarche est que ces plus-values peuvent être

7. Filiale commerciale de France Télécom chargée de l'exploitation des communications avec mobile.

obtenues de la même manière pour d'autres applications.

Perspectives

Tout d'abord, d'un point de vue technique, l'élagage s'applique à toutes les connexions du modèle indépendamment de leur localisation. Cette solution permet d'obtenir de meilleures performances puisque l'algorithme détermine pour chaque réseau modulaire l'architecture la plus adaptée. En particulier, l'architecture de chaque OWE s'adapte aux besoins de sa prédiction indépendamment des autres. Ainsi, les variables d'entrée sélectionnées pour prédire un paramètre sont généralement différentes pour chaque réseau prédictif. Par conséquent, la définition du contexte est locale. La définition d'un contexte commun à tous les OWEs diminuera les performances mais aura plus de signification. Nous pourrons connaître les variables qui caractérisent le mieux la différence de classes. Pour parvenir à uniformiser les couches d'entrée des *OWE*s, il suffit de modifier l'algorithme pour que l'on connaisse durant l'apprentissage la connexion la moins utile, tout OWE confondu. Ce critère peut s'obtenir en cumulant sur tous les OWEs la pertinence d'une même connexion. L'architecture de départ est la même pour tous les OWEs. Par conséquent, si on supprime dans chaque réseau prédictif la connexion qui en moyenne est la moins pertinente pour tous les OWEs, en fin d'apprentissage, ils auront la même architecture. Ils auront donc en particulier le même contexte.
Les techniques d'élagage sont largement utilisées dans le but d'appliquer, par la suite, au réseau élagué un procédé d'extraction des règles [Andrews *et al.*, 1995]. L'utilisation de ces procédés nécessite une adaptation pour être applicable à notre modèle. En effet, les poids ne sont pas constants. Donc, dans certains cas les coefficients apparaissant dans les règles changeront pour chaque contexte. Si chaque situation possède un contexte différent toutes les règles seront uniques. Il y aura plus d'exceptions à la règle que de situations concernées par celle-ci. En d'autres termes, l'extraction de règles n'est intéressante que si les règles extraites s'appliquent à un grand nombre de cas. Nous pouvons donc l'envisager dans le cas où le nombre de contextes différents est limité. Nous obtenons sans modification un jeu de règles pour chaque contexte. Si les formes appartiennent à des catégories, nous auront une formule générale par classe. Nous pouvons même envisager de dégager des règles qui, à partir des variables contextuelles, déterminent la classe à laquelle appartient la forme.
L'algorithme de notre modèle est compartimenté. Les phases d'apprentissage, d'élagage et d'extraction de règles sont distinctes. Il est donc possible d'améliorer un algorithme sans avoir à modifier les autres. De plus, tous les algorithmes neuronaux de modification des poids fournissent la valeur δw_i nécessaire au calcul des modifications à apporter aux poids du réseau prédictif OWE en charge de l'estimation du w_i (où i est une connexion du réseau principal). Il est donc possible d'appliquer un algorithme plus performant que celui de la rétro-propagation du gradient [Le Cun, 1985; Rumelhart *et al.*, 1986a] tel que des variantes [Fahlman, 1989; Riedmiller and Braun, 1993], ou encore des algorithmes du second ordre [Battiti, 1992; Moller, 1993; Johansson *et al.*, 1991; Tollenaere, 1990]. De même, de nombreuses techniques d'élagage [Reed, 1993; Le Cun *et al.*, 1990; Hassibi and Stork, 1993; Mozer and Smolensky, 1989] fournissent une valeur de la pertinence d'une connexion s_i utilisée par notre algorithme (où i est une connexion quelconque du modèle). Il sera donc

facile d'appliquer un algorithme mieux adapté au problème.

Les possibilités d'application de notre travail sont donc larges et dépassent le cadre de la fouille de données. Ses bénéfices peuvent être obtenus, de la même manière que celle qui nous a servi d'exemple, pour d'autres applications où la connaissance du domaine ne suffit pas à modéliser correctement un phénomène. Le modèle que nous avons développé et les réseaux connexionnistes que nous avons réunis et interprétés pourront alors, à partir des données, enrichir la compréhension du phénomène en analysant et en organisant les informations par rapport à la tâche à accomplir.

La mienne est sur le point de l'être. La dernière remarque que je voudrais faire va au delà des résultats de l'application et même des méthodes d'extraction de connaissances ou d'interprétation de réseaux neuronaux que nous avons abordés. Je tiens ici à souligner que, selon moi, la démarche, illustrée par une application technologique et des outils numériques, rejoint d'autres approches plus cognitives, relatives aux neurosciences, à la psychologie et même à la philosophie. En effet, nous travaillons ici sur les liens entre des concepts aussi fondamentaux que la structure et la fonction et, à notre niveau mais dans le même courant que des personnages comme Piaget, Hannequin, Reuchlin ou Ehrlich, nous essayons de voir comment l'organisation d'un support physique peut faire émerger de la connaissance.

Annexe A

Descriptif des corpus

Identificateur	Sémantique des paramètres
EAU	Pourcentage d'eau sur une parcelle de 400m de coté
BOIS	Pourcentage de zones boisées sur une parcelle de 400m de coté
PRAIRIE	Pourcentage de zones de prairie sur une parcelle de 400m de coté
ROCHE	Pourcentage de zones rocheuses sur une parcelle de 400m de coté
BATI DIFFUS	Pourcentage de zones comportant des bâtis peu imposants sur une parcelle de 400m de coté
BATI MIXTE	Pourcentage de zones comportant des bâtis semi imposants sur une parcelle de 400m de coté
BATI DENSE	Pourcentage de zones comportant des bâtis imposants sur une parcelle de 400m de coté
ALTITUDE	Altitude du centre de la parcelle

TAB. A.1 – *Description des attributs du corpus géographique.*

N°	Attributs	min	max	μ	σ
1	EAU	0.00	100.00	8.14	26.59
2	BOIS	0.00	100.00	32.18	37.51
3	PRAIRIE	0.00	100.00	46.47	39.53
4	ROCHE	0.00	100.00	1.37	8.84
5	BATI DIFFUS	0.00	100.00	3.93	13.35
6	BATI MIXTE	0.00	100.00	6.39	18.00
7	BATI DENSE	0.00	100.00	1.49	8.59
8	ALTITUDE	9.00	2271.00	405.66	391.42

TAB. A.2 – *Moyenne et écart type de chaque paramètre du corpus géographique.*

Identificateur	Sémantique des paramètres
ADIFF	atténuation due à la dernière diffraction et à la reflexion respectivement sur les immeubles devant et derrière le mobile
ATRAJ	atténuation due à la distance émetteur-récepteur
ASOL	atténuation due à la multi-diffraction sur les arêtes
ANGLE_INCIDENCE	angle entre l'horizontale et la droite reliant l'émetteur au premier pic de diffraction
BOIS	indique si le mobile est situé dans la végétation
HB	hauteur de l'émetteur au dessus du sol
HM	hauteur du mobile
Dm	distance émetteur-récepteur
HIMOY	hauteur moyenne des bâtiments sur le profil émetteur-récepteur
LOS	1 si visibilité directe, 0 sinon
N	nombre d'arêtes entre l'émetteur et le récepteur
LOSMNT	1 si visibilité directe à travers le MNT, 0 sinon
ADEGT	atténuation due à la diffraction sur une arête principale
dBAT	distance cumulée de bâtiments sur le profil émetteur-récepteur
dPAV	distance cumulée de petites constructions sur le profil émetteur-récepteur
dBOIS	distance cumulée de végétation sur le profil émetteur-récepteur
dhBAT	surface cumulée de bâtiments sur le profil émetteur-récepteur
dhPAV	surface cumulée de petites constructions sur le profil émetteur-récepteur
dhBOIS	surface cumulée de végétation sur le profil émetteur-récepteur
B	espacement moyen des bâtiments
W	largeur de la rue ou se trouve le mobile
W1	distance entre le mobile et la dernière arête avant le mobile
W2	distance entre le mobile et l'arête suivante (derrière le mobile)
HiMob	hauteur de l'immeuble avant le mobile
HiMax	hauteur de l'immeuble le plus haut sur le profil émetteur-récepteur
D_HiMax	distance à l'émetteur de l'immeuble le plus haut
phi	angle rue/axe base/mobile en degrés
distVegEmttDernierBat	distance de végétation traversée entre l'émetteur et le dernier Bâti
distVegDernierBatMob	distance de végétation traversée entre le dernier bâti et le mobile
HIMOY-HM	cf + haut
HB-HIMOY	cf + haut
N/Dm	densite d'obstacles sur le profil émetteur-récepteur
AMES	affaiblissement mesure sur le terrain

TAB. A.3 – *Description des attributs du corpus complet.*

N°	Attributs	min	max	μ	σ
1	ADIFF	-17.20	51.85	25.99	8.29
2	ATRAJ	54.18	127.35	96.92	8.04
3	ASOL	-18.54	43.11	10.14	7.97
4	ANGLE_INCIDENCE	-177.80	178.94	-0.46	2.83
5	BOIS	0.00	1.00	0.02	0.14
6	HB	12.00	55.42	32.29	7.40
7	HM	0.98	10.24	1.64	0.27
8	Dm	13.16	9833.61	2545.29	1983.05
9	HIMOY	0.00	103.83	18.03	7.33
10	LOS	0.00	1.00	0.01	0.11
11	N	0.00	157.00	30.24	20.46
12	LOSMNT	0.00	1.00	0.85	0.36
13	ADEGT	0.00	42.12	3.69	8.13
14	dBAT	0.00	3584.46	611.36	439.53
15	dPAV	0.00	446.09	6.46	16.66
16	dBOIS	0.00	2158.95	126.24	270.12
17	dhBAT	0.00	81865.50	12200.28	10495.60
18	dhPAV	0.00	54171.07	141.77	654.82
19	dhBOIS	0.00	312005.47	3611.80	10624.80
20	B	0.00	4999.30	88.75	133.81
21	W	3.85	7408.85	240.37	600.44
22	W1	0.33	7204.58	210.13	597.20
23	W2	1.00	685.33	30.24	24.76
24	HiMob	0.00	307.37	18.10	12.05
25	HiMax	-34.48	342.23	14.37	16.13
26	D_HiMax	0.00	9824.73	2324.01	1942.81
27	phi	0.00	90.00	47.32	28.10
28	distVegEmttDernierBat	0.00	1898.05	99.31	230.33
29	distVegDernierBatMob	0.00	1948.49	13.49	92.86
30	HIMOY-HM	-2.70	102.23	16.39	7.34
31	HB-HIMOY	-91.83	54.67	14.25	8.42
32	N/Dm	0.00	0.13	0.01	0.01
33	**AMES**	**41.45**	**174.30**	**132.11**	**16.27**

TAB. A.4 – *Moyenne et écart type de chaque paramètre du corpus complet.*

	ADIFF	ATRAJ	ASOL	ANGLE	BOIS	HB	HM	Dm	HIMOY	LOS	N	LOSMNT	ADEGT	dBAT	dPAV	dBOIS
ADIFF	1.0	-.24	-.13	-.076	-.14	.32	-.036	-.25	.48	-.35	.15	.054	-.067	.23	-.018	-.25
ATRAJ	-.24	1.0	.71	.41	.025	.00070	.063	.87	-.012	.095	.65	-.27	.31	.56	.11	.35
ASOL	-.13	**.71**	1.0	.31	-.23	.016	.018	.60	.36	.14	.47	-.40	.49	.39	.31	.27
ANGLE_INCIDENCE	-.076	.41	.31	1.0	.014	-.16	-.023	.25	.026	-.089	.10	-.24	.25	.079	.15	.10
BOIS	-.14	.025	.016	.014	1.0	-.042	-.023	.0085	-.036	-.017	-.046	-.032	.039	-.055	.0061	.11
HB	.32	.00070	-.23	-.16	-.042	1.0	.064	-.034	.35	-.019	.25	.069	-.12	.34	-.22	-.021
HM	-.036	.063	.018	-.023	-.023	.064	1.0	.068	-.0080	.0021	.17	.067	-.064	.21	-.064	-.062
Dm	-.25	**.87**	.60	.25	.0085	-.034	.068	1.0	-.083	-.0053	.67	-.23	.29	.53	.082	.42
HIMOY	.48	-.012	.36	.026	-.036	.35	-.0080	-.083	1.0	-.11	.093	-.30	.36	.18	.16	.0022
LOS	-.35	.095	.14	-.089	-.017	-.019	.0021	-.0053	-.11	1.0	-.12	.046	-.045	-.11	-.034	-.042
N	.15	.65	.47	.10	-.046	.25	.17	.67	.093	-.12	1.0	-.020	.041	.91	-.015	.099
LOSMNT	.054	-.27	-.40	-.24	-.032	.069	.067	-.23	-.30	.046	-.020	1.0	-.89	.042	-.40	-.16
ADEGT	-.067	.31	.49	.25	.039	-.12	-.064	.29	.36	-.045	.041	**-.89**	1.0	-.031	.47	.21
dBAT	.23	.56	.39	.079	-.055	.34	.21	.53	.18	-.11	**0.91**	.042	-.03	1.0	-.13	**-.0092**
dPAV	-.018	.11	.31	.15	.0061	-.22	-.064	.082	.16	-.034	-.015	-.40	.47	-.13	1.0	.052
dBOIS	-.25	.35	.27	.10	.11	-.021	-.062	.42	.0022	-.042	.099	-.16	.21	-.0092	.052	1.0
dhBAT	.33	.43	.41	.064	-.057	.41	.21	.36	.44	-.10	**.78**	-.055	.052	.92	-.10	-.043
dhPAV	.034	.13	.38	.12	**.0043**	**-.091**	**-.036**	.12	.39	**-.024**	.011	-.38	.51	**-.071**	**.75**	.13
dhBOIS	-.11	.27	.35	.12	.050	-.054	-.045	.31	.25	-.033	.043	-.29	.45	-.034	.23	.67
B	-.13	.21	.16	.21	.0078	-.25	-.022	.20	-.056	-.018	-.12	-.23	.25	-.14	.12	.16
W	-.63	.28	.26	.17	.10	-.21	-.025	.26	-.14	.24	-.17	-.17	.21	-.18	.035	.25
W1	**-.62**	.28	.26	.16	.095	**-.21**	**-.027**	.25	**-.14**	.24	**-.17**	-.17	.21	**-.18**	.034	.25
W2	-.20	.11	.086	.058	.19	-.083	.040	.085	-.056	.012	.0034	-.037	.040	-.0080	.039	.069
HiMob	.24	-.11	.078	.036	-.020	.18	.047	-.12	.44	-.078	-.041	-.14	.22	.027	.048	-.0098
HiMax	.37	-.026	.13	.074	-.055	.12	.027	-.037	.44	-.17	.069	-.17	.28	.11	.13	-.011
D_HiMax	**-.027**	**.79**	.52	.19	**-.027**	.045	.080	**.94**	**-.040**	**-.099**	**.76**	-.17	.21	.61	.066	.34
phi	.12	-.0034	-.074	-.084	-.023	.14	.0043	.014	.0033	-.046	.11	.15	-.17	.11	-.22	.042
distVegEmtfDernierBat	-.060	.36	.26	.074	.063	-.0065	-.055	.45	.022	-.044	.19	-.15	.20	.064	.058	**.82**
distVegDernierBatMob	-.21	.079	.089	.023	0.92	.036	-.022	.078	.036	-.0062	-.038	-.038	.059	-.055	.018	.44
HIMOY-HIM	.48	**-.014**	.36	.027	**-.036**	.35	**-.047**	**-.086**	**1.0**	**-.11**	.087	-.31	.36	.17	.17	.0047
HB-HIMOY	-.12	.011	-.52	-.16	-.0066	.60	.064	.040	-.54	.073	.15	.32	-.42	.15	-.34	-.021
N/Dm	.51	-.57	-.39	-.34	-.090	.29	.0028	-.52	.15	-.13	.075	.34	-.37	.099	-.13	-.40
AMES	.13	**.80**	.67	.36	-.0071	.074	.021	**.70**	.28	-.20	.66	-.34	.39	.61	.12	.32

TAB. A.5 – *Matrice de corrélation des variables du corpus complet (partie 1).*

	dBAT	dPAV	dBOIS	B	W	W1	W2	HiMob	HiMax	D_HiMax	phi	distVegEmttDernierBat	distVegDernierBatMob	HIMOY-HM	HB-HIMOY	N/Dm	AMES
ADIFF	.33	.034	-.11	-.13	-.63	-.62	-.20	.24	.37	-.027	.12	-.060	-.21	.48	-.12	.51	.13
ATRAJ	.43	.13	.27	.21	.28	.28	.11	-.11	-.026	.79	-.0034	.36	.079	-.014	.011	-.57	.80
ASOL	.41	.38	.35	.16	.26	.26	.086	.078	.13	.52	-.074	.26	.089	.36	-.52	-.39	.67
ANGLE_INCIDENCE	.064	.12	.12	.21	.17	.16	.058	.036	.074	.19	**-.084**	.074	.023	.027	**-.16**	**-.34**	.36
BOIS	-.057	.0043	.050	.0078	.10	.095	.19	-.020	-.055	-.027	-.023	.063	.092	-.036	-.0066	-.090	-.0071
HB	.41	-.091	-.054	-.25	-.21	-.21	-.083	.18	.12	.045	.14	-.0065	.036	.35	.60	.29	.074
HM	.21	-.036	-.045	-.022	-.025	-.027	.040	.047	.027	.080	.0043	-.055	-.022	-.047	.064	.0028	.021
Dm	.36	.12	.31	.20	.26	.25	.085	**-.12**	**-.037**	.94	.014	.45	.078	**-.086**	.040	**-.52**	.70
HIMOY	.44	.39	.25	-.056	-.14	-.14	-.056	.44	.44	-.040	.0033	.022	.036	1.0	-.54	.15	.28
LOS	-.10	-.024	-.033	-.018	.24	.24	.012	-.078	-.17	-.099	-.046	-.044	-.0062	-.11	.073	-.13	-.20
N	.78	.011	.043	-.12	-.17	-.17	.0034	-.041	.069	.76	.11	.19	-.038	.087	.15	.075	.66
LOSMNT	**-.055**	**-.38**	**-.29**	**-.23**	**-.17**	**-.17**	**-.037**	**-.14**	**-.17**	**-.17**	.15	**-.15**	**-.038**	**-.31**	.32	.34	**-.34**
ADEGT	.052	.51	.45	.25	.21	.21	.040	.22	.28	.21	-.17	.20	.059	.36	-.42	-.37	.39
dBAT	**.92**	-.071	-.034	-.14	-.18	-.18	-.0080	.027	.11	.61	.11	.064	-.055	.17	.15	.099	.61
dPAV	-.10	**.75**	.23	.12	.035	.034	.039	.048	.13	.066	-.22	.058	.018	.17	-.34	-.13	.12
dBOIS	**-.043**	.13	.67	.16	.25	.25	.069	**-.0098**	**-.011**	.34	.042	.82	.44	.0047	**-.021**	**-.40**	.32
dhBAT	1.0	-.021	-.031	-.14	-.17	-.17	-.026	.14	.18	.44	.090	.019	-.047	.43	-.0059	.15	.56
dhPAV	-.021	1.0	.39	.090	.051	.050	.027	.21	.27	.094	-.16	.12	.070	.39	-.41	-.14	.18
dhBOIS	-.031	.39	1.0	.16	.16	.17	.047	.25	.40	.24	-.050	.60	.25	.25	-.26	-.31	.29
B	-.14	.090	.16	1.0	.016	.015	.040	-.028	.022	.19	-.099	.18	.017	-.055	-.17	**-.39**	.13
W	**-.17**	.051	.17	.016	1.0	1.0	.15	**-.079**	**-.26**	**-.091**	-.13	**-.0096**	.27	**-.14**	**-.071**	**-.44**	.038
W1	-.17	.050	.17	.015	**1.0**	1.0	.11	-.077	-.26	-.093	-.13	-.011	.27	-.14	-.070	-.43	.038
W2	-.026	.027	.047	.040	.15	.11	1.0	-.049	-.069	.045	-.099	.023	.029	-.058	-.027	-.16	.00072
HiMob	.14	.21	.25	**-.028**	**-.079**	**-.077**	**-.049**	1.0	.73	**-.087**	**-.0048**	**-.029**	.13	.44	**-.21**	.11	.058
HiMax	.18	.27	.40	.022	-.26	-.26	-.069	**.73**	1.0	.035	.0059	.080	**-.0088**	.44	-.27	.14	.17
D_HiMax	.44	.094	.24	.19	**-.091**	**-.093**	.045	-.087	.035	1.0	.064	.46	-.015	**-.043**	.075	**-.37**	.70
phi	.090	-.16	-.050	**-.099**	**-.13**	**-.13**	**-.099**	**-.0048**	.0059	.064	1.0	.048	.019	.0032	.12	.12	.048
distVegEmttDernierBat	.019	.12	.60	.18	**-.0096**	**-.011**	.023	**-.029**	.080	.46	.048	1.0	.031	.024	**-.025**	**-.32**	.34
distVegDernierBatMob	**-.047**	.070	.25	.017	.27	.27	.029	.13	**-.0088**	**-.015**	.019	.031	1.0	.037	.0016	**-.16**	.071
HIMOY-HM	.43	.39	.25	**-.055**	**-.14**	**-.14**	**-.058**	.44	.44	**-.043**	.0032	.024	.037	1.0	**-.54**	.15	.28
HB-HIMOY	**-.0059**	**-.41**	**-.26**	**-.17**	**-.071**	**-.070**	**-.027**	**-.21**	**-.27**	.075	.12	**-.025**	.0016	**-.54**	1.0	.13	**-.17**
N/Dm	.15	**-.14**	**-.31**	**-.39**	**-.44**	**-.43**	**-.16**	.11	.14	**-.37**	.12	**-.32**	**-.16**	.15	.13	1.0	**-.33**
AMES	.56	.18	.29	.13	.038	.038	.00072	.058	.17	**.70**	.048	.34	.071	.28	**-.17**	**-.33**	1.0

TAB. A.6 – *Matrice de corrélation des variables du corpus complet (partie 2).*

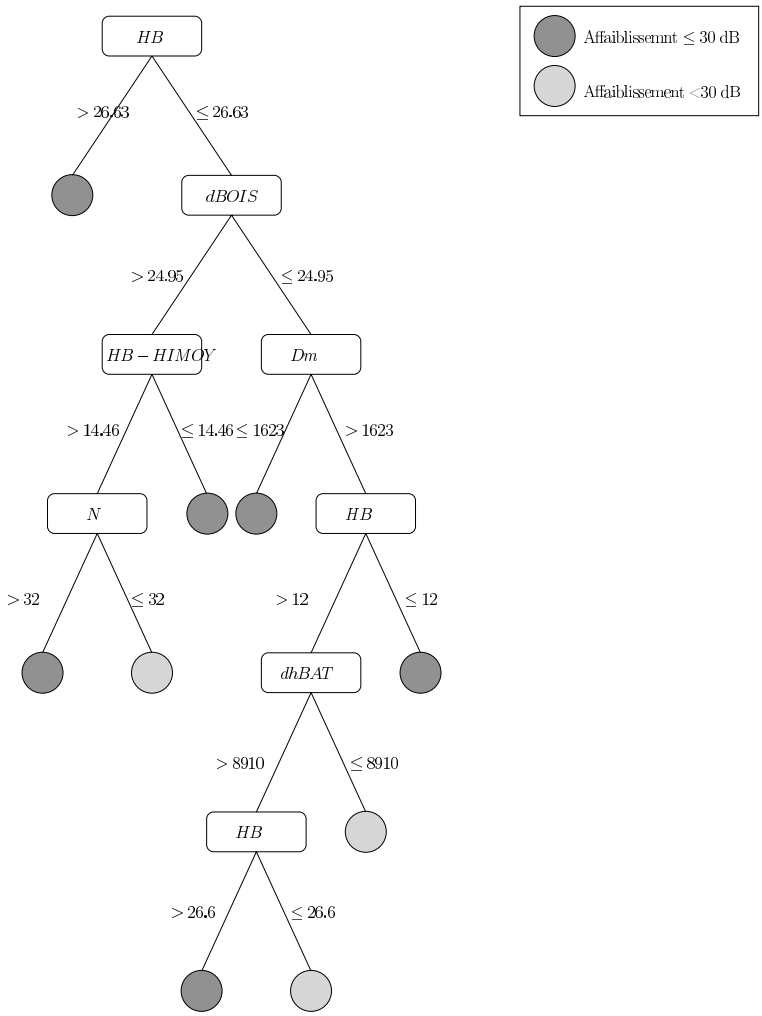


FIG. A.1 – *Arbre de discrimination des affaiblissements faibles et forts*

Liste des figures

Liste des tableaux

Liste des algorithmes

Bibliographie

[Ackley *et al.*, 1985] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, 9 :147–169, 1985. Ré-imprimé dans [Anderson and Rosenfeld, 1988].

[Ahalt *et al.*, 1990] S. C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, 3 :277–290, 1990.

[Alexandre and Guyot, 1995] F. Alexandre and F. Guyot. Neurobiological inspiration for the architecture and functioning of cooperating neural networks. In *Proceedings of Int. Workshop on ANNs*, Malaga, Spain, Juin 1995.

[Anderson and Rosenfeld, 1988] J. A. Anderson and E. Rosenfeld, editors. *Neurocomputing : Foundations of Research*. The MIT Press, Cambridge, MA, 1988.

[Andrews *et al.*, 1995] R. Andrews, J. Diederich, and A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificical neural networks. *Knowledge Based Systems*, 8(6) :378–389, Dec. 1995.

[Anouar *et al.*, 1997] F. Anouar, F. Badran, and S. Thiria. *Statistique et méthodes neuronales*, chapter Cartes topologiques et nuées dynamiques. Dunod, 1997.

[Asoh and Otsu, 1989] H. Asoh and N. Otsu. Non linear discriminant analysis and multilayer perceptrons. In *Proceedings of the International Joint Conference on Neural Networks*, pages 411–415, San Diego, 1989.

[Aurenhammer, 1991] F. Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23 :345–405, 1991.

[Balandier and Governo, 1998] T. Balandier and D. Governo. Comparison of different regularization terms and pruning methods for function approximation. In *IEEE Neural Net Conference*, Alaska, USA, 1998.

[Balandier *et al.*, 1995a] T. Balandier, A. Caminada, Lemoine V., and F. Alexandre. 170 mhz field strength prediction in urban environment using neural nets. In *IEEE 6th international symposium on Personal, Indoor and Mobile Radio Communication, Toronto, Canada*, pages 120–124, 1995.

[Balandier *et al.*, 1995b] T. Balandier, A. Caminada, Lemoine V., and F. Alexandre. Multi-layer perceptron for field strenght prediction in cellular network design. In *Engineering Applications of Neural Networks, Helsinki, Finland*, 1995.

[Baldi and Hornik, 1989] P. Baldi and K. Hornik. Neural networks and principal component analysis : Learning from examples without local minima. *Neural Networks*, 2(1) :53–58, 1989.

[Battiti, 1992] T. Battiti. First and second-order methods for learning : Between steepest descent and Newton's method. *Neural Computation*, 4(2) :141–166, 1992.

[Benzécri, 1973] J.-P. Benzécri. *L'analyse de données, 2 : L'analyse en correspondances.* Dunod, Paris, 2nd edition, 1973.

[Bishop, 1995] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[Bock, 1985] H. H. Bock. On some signification tests in cluster analysis. *Journal of classification*, 2 :3–24, 1985.

[Boniface, 2000] Y. Boniface. *Etude et développement d'une bibliothèque d'adaptation du parallélisme neuromimétique au parallélisme MIMD.* PhD thesis, Université Henri Poincaré, Nancy (France), 2000.

[Bose and Garga, 1993] N. K. Bose and Amulya K. Garga. Neural network design using Voronoi diagrams. *IEEE Transactions on Neural Networks*, 4(5) :778–787, September 1993.

[Bougrain and Alexandre, 1998] Laurent Bougrain and Frédéric Alexandre. Unsupervised Algorithms for Vector Quantization : Evaluation on an environmental data set. In *NEURAP, Fourth International Conference on Neural Networks and their Applications, Marseille, France*, pages 347–350, March 1998.

[Bougrain and Alexandre, 1999a] L. Bougrain and F. Alexandre. Recurrent neural networks for mobile phone cell planning using topological information. In *Proc. of Engineering Applications of Neural Networks*, Varsovie, 1999.

[Bougrain and Alexandre, 1999b] Laurent Bougrain and Frédéric Alexandre. Unsupervised Connectionist Algorithms for Clustering an environmental data set : a comparison. *Neurocomputing*, 1-3(28) :177–189, October 1999.

[Bougrain and Alexandre, 1999c] Laurent Bougrain and Frédéric Alexandre. Unsupervised Connectionist Clustering Algorithms for a better Supervised Prediction : Application to a radio communication problem. In *International Joint Conference on Neural Networks, Washington (D.C.), USA*. International Neural Networks Society, July 1999.

[Bougrain and Boniface, 2000] L. Bougrain and Y. Boniface. Détermination de l'architecture de modèles neuromimétiques par implantation parallèle. In *Journées scientifiques du centre Charles Hermite*, 2000.

[Bougrain *et al.*, 1998] L. Bougrain, N. Pican, and F. Alexandre. Rôle du contexte dans le modèle owe : un réseau de neurones artificiels utilisant des connexions axo-synaptiques. In *Proc. of NeuroSciences pour Ingenieur*, Munster, 1998.

[Bougrain, 1997a] L. Bougrain. Classification d'environnements : réseaux neuromimétiques et méthodes non supervisées. Technical Report 1, France Télécom R&D, Belfort, 1997.

[Bougrain, 1997b] L. Bougrain. Prédiction de l'atténuation du champ radioélectrique : évaluation des outils de prédictions. Technical Report 2, France Télécom R&D, Belfort, 1997.

[Bougrain, 1998a] L. Bougrain. Prédiction continue : réseaux récurrents et mixture d'experts. Technical Report 4, France Télécom R&D, Belfort, 1998.

[Bougrain, 1998b] L. Bougrain. Prédiction de l'atténuation du champ radioélectrique : approche modulaire. Technical Report 3, France Télécom R&D, Belfort, 1998.

[Bougrain, 1999a] L. Bougrain. Traitement des prédictions difficiles et utilisation de contextes spatiaux. Technical Report 5, France Télécom R&D, Belfort, 1999.

[Bougrain, 1999b] Laurent Bougrain. Détection et traitement de "données à problèmes". In *7ième jounnées de la Société Francophone de Classification, Nancy, France*, pages 333–339, September 1999.

[Breiman *et al.*, 1984] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. Technical report, Wadsworth International, Monterey, CA, 1984.

[Brent, 1991] R. P. Brent. Fast training algorithms for multilayer neural nets. *IEEE Transactions on Neural Networks*, 2 :346–354, 1991.

[Bridle, 1990a] J. S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In F. Fogelman Soulié and J. Hérault, editors, *Neuro-computing : Algorithms, Architectures and Applications*, pages 227–236, Berlin, 1990. Springer.

[Bridle, 1990b] J. S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 211–217, San Mateo, CA, 1990. Morgan Kaufman Publishers Inc.

[Caminada *et al.*, 1996] A. Caminada, R. Deldicque, and T. Balandier. Neural nets : an alternative method for field strength prediction. Technical report, European cooperation in the field of scientific and technical research, EURO-COST 231 TD(96) 7, jan. 1996. [Reprinted in [Caminada *et al.*, 1997]].

[Caminada *et al.*, 1997] A. Caminada, R. Deldicque, and T. Balandier. Neural nets : an alternative method for field strength prediction. In *IEEE Vehicular Technology Conference*, 1997.

[Canu, 1996] S. Canu. Des réseaux de neurones à la régression flexible, 1996. Mémoire d'habilitation à diriger les recherches.

[Chan, 1991] G. Chan. Propagation and coverage prediction for cellular radio system. *IEEE Transactions on Vehicular Technology*, 40(4) :665–670, nov. 1991.

[Chen, 1983] C. H. Bailey M. Chen. Morphological basis of long term habituation and sensitization in aplysia. *Science*, 220 :91–93, 1983.

[Chester, 1990] D. L. Chester. Why two hidden layers are better than one. In *Proceedings of the International Joint Conference on Neural Networks*, volume 1, pages 265–268. Erlbaum, 1990.

[Cibas *et al.*, 1996] T. Cibas, F. Fogelman Soulié, P. Gallinari, and S. Raudys. Variable selection with neural networks. *Neurocomputing*, 12 :223–248, 1996.

[Cowan, 1989] Jack D. Cowan. Neural networks : The early days. In *[Touretzky, 1989]*, pages 828–842, 1989.

[Cybenko, 1989] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4) :303–314, 1989.

[D'arché-Buc *et al.*, 1994] F. D'arché-Buc, J.-P. Nadal, and D. Zwiersky. Hybrid trees for supervised learning of decision rules, combining symbolic and neural processing. In *European Conference on Artificial Intelligence*, 1994. Workshop.

[Dehaene and Changeux, 1989] Stanislas Dehaene and Jean-Pierre Changeux. A simple model of prefrontal cortex function in delayed-response task. *Journal of Cognitive Neuroscience*, 1(3) :244–261, 1989.

[Delacour, 1987] J. Delacour. Apprentissage et mémoire : une approche neurobiologique. *Masson (Ed.)*, 1987.

[Deldicque *et al.*, 1997] R. Deldicque, A. Caminada, and T. Balandier. Costrn : Modèle hybride analytique/connexionniste de prévision de l'affaiblissement en petites cellules à 900 mhz. In *3ème journées d'étude : propagation électromagnétique dans l'atmosphère du décamétrique à l'angström*. Groupe ouest de la Société des electriciens et des electroniciens, 1997.

[Demartines and Hérault, 1997] P. Demartines and J. Hérault. Curvilinear component analysis : A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transactions on Neural Networks*, 8(1) :148–154, January 1997.

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society series B*, 39 :1–38, 1977.

[DeSieno, 1988] D. DeSieno. Adding a conscience to competitive learning. In *IEEE International Conference on Neural Networks*, volume 1, pages 117–124, New York, 1988. (San Diego 1988), IEEE.

[Deygout, 1966] L. Deygout. Multiple knife-edge diffracion of microwaves. *IEEE Transaction on Antennas and Propagation*, 14(4) :480–489, 1966.

[Dhar and Stein, 1997] V. Dhar and R. Stein. *Seven Methods for Transforming Corporate data Into Business Intelligence*. Prentice-Hall, 1997.

[Diday, 1971] E. Diday. Une nouvelle méthode en classification automatique et reconnaissance des formes : La méthode des nuées dynamiques. *Revue de Statistique Appliquée*, 19(2) :19–33, 1971.

[Elemento, 1999] O. Elemento. Apport de l'analyse en composantes principales pour l'initialisation et la validataion de cartes de kohonen. In *7ième journée de la Société Francophone de Classification*, pages 325–331, Nancy, sept. 1999.

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

[Fahlman and Lebiere, 1990] S.E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2, pages 524–532, San Mateo, 1990. (Denver 1989), Morgan Kaufmann.

[Fahlman, 1989] S.E. Fahlman. Fast-learning variations on back-propagation : An empirical study. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 38–51, San Mateo, 1989. (Pittsburg 1988), Morgan Kaufmann.

[Fayyad *et al.*, 1996] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery : An Overview. In U.M. Fayyad, G. Piatetsky-Shapiro,

P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI Press / MIT Press, Menlo Park, California, 1996.

[Feldman and Ballard, 1982] J.A. Feldman and D.H. Ballard. Connectionist models and their properties. In *Cognitive Science*, volume 6, pages 205–254, 1982.

[Forgy, 1965] E. W. Forgy. Cluster analysis of multivariate data : efficiency vs interpretability of classifications. *Biometrics*, 21 :768–769, 1965.

[Friedman and Stuetzle, 1981] J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76 :817–823, 1981.

[Friedman, 1985] J. H. Friedman. Classification and multiple response regression through projection pursuit. Technical Report LCS012, Stanford Univ., 1985.

[Friedman, 1991] J. H. Friedman. Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19 :1–141, 1991.

[Friedman, 1994] J. H. Friedman. An overview of predictive learning and function approximation. In V. Cherkassky, J. H. Friedman, and H. Wechsler, editors, *From Statistics to Neural Networks, Theory and Pattern Recognition Applications*, volume 136 of *NATO ASI Series F*, pages 1–61. Springer, 1994.

[Fritzke, 1995] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 625–632. The MIT Press, 1995.

[Gallinari *et al.*, 1988] Patrick Gallinari, Sylvie Thiria, and Francoise Fogelman-Soulie. Multilayer perceptrons and data analysis. In *Proceedings of the IEEE Annual International Conference on Neural Networks (ICNN88)*, volume I, pages 391–399, San Diego, CA, July 1988.

[Gallinari *et al.*, 1991] P. Gallinari, S. Thiria, F. Badran, and F. Fogelman-Soulie. On the relations between discriminant analysis and multilayer perceptrons. *Neural Networks*, 4(3) :349–360, June 1991.

[Geiger *et al.*, 1990] D. Geiger, T. Verma, and J. Pearl. Recognizing independence in Bayesian networks. *Networks*, 20 :507–534, 1990.

[Governo, 1997] D. Governo. Optimisation de réseaux de neurones pour l'ingénieurie des réseaux cellulaires. Mémoire de dess informatique, Université Henry Poincaré, Nancy, 1997.

[Grossberg, 1982] S. Grossberg. *Studies of the Mind and Brain : Neural principles of learning, perception, development, cognition, and motor control.* Reidel Press, Boston, MA, 1982.

[Gschwendtner and Landstorfer, 1993] B.E. Gschwendtner and F.M. Landstorfer. An application of neural networks to the prediction of terrestrial wave propagation. Technical report, European cooperation in the field of scientific and technical research, COST 231 TD(93) 21, jan. 1993.

[Gschwendtner, 1993] B.E Gschwendtner. Some results using neural networks for field strength prediction. Technical report, European cooperation in the field of scientific and technical research, COST 231 TD(93) 117, sept. 1993.

[Hansen and Salamon, 1990] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12 :993–1001, 1990.

[Hartigan, 1975] J. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[Hassibi and Stork, 1993] B. Hassibi and D. G. Stork. Second order derivatives for network pruning : Optimal brain surgeon. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5. Proceedings of the 1992 Conference*, pages 164–171, San Mateo, CA, 1993. Morgan Kaufmann.

[Hastie and Tibshirani, 1990] T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, 1990.

[Hastie and Tibshirani, 1994] Trevor Hastie and Robert Tibshirani. Nonparametric regression and classification. In V. Cherkassky, J. H. Friedman, and H. Wechsler, editors, *From Statistics to Neural Networks, Theory and Pattern Recognition Applications*, volume 136 of *NATO ASI Series F*, pages 62–69. Springer, 1994.

[Hebb, 1949] D.O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949.

[Hecht-Nielsen, 1989] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *International Joint Conference on Neural Networks*, volume 1, pages 593–605, New York, 1989. (Washington 1989), IEEE.

[Hertz *et al.*, 1991] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the theory of neural computation*. Addison Wesley, 1991.

[Hilario *et al.*, 1994] M. Hilario, C. Pellegrini, and F. Alexandre. Modular Integration of Connectionist and Symbolic Processing in Knowledge-Based Systems. In *Proceedings International Symposium on Integrating Knowledge and Neural Heuristics*, Pensacola Beach (Florida, USA), May 1994.

[Hinton and Sejnowski, 1986] G.E. Hinton and T.J. Sejnowski. Learning and relearning in boltzmann machines. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing*, volume 1, chapter 7, pages 282–317. MIT Press, Cambridge, 1986.

[Hopfield and Tank, 1986] J.J. Hopfield and D.W. Tank. Computing with neural circuits : A model. *Science*, 233 :625–633, 1986.

[Hornik *et al.*, 1989] Kurt Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2 :359–366, 1989.

[Hotelling, 1933] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24 :417–441 and 498–520, 1933.

[Hérault and Ans, 1984] J. Hérault and B. Ans. Réseau de neurones à synapses modifiables : décodage de messages sensoriels composites par apprentissage non supervisé et permanent. In *CRAS*, volume 13 of *III*, pages 525–528, Paris, 1984.

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

[Hwang *et al.*, 1992] J.-N. Hwang, H. Li, M. Maechler, D. Martin, and J. Schimert. A comparison of projection pursuit and neural network regression modeling. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4. Proceedings of the 1991 Conference*, pages 1159–1166, San Mateo, CA, 1992. Morgan Kaufmann.

[Jacobs *et al.*, 1991] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.

[Jain *et al.*, 1999] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, September 1999.

[Johansson *et al.*, 1991] E. M. Johansson, F. U. Dowla, and D. M. Goodman. Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems*, 2:291–302, 1991.

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

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

[Jordan, 1986b] M. I. Jordan. Serial order: A parallel distributed processing approach. Technical Report ICS Report 8604, Institute for Cognitive Science, University of California at San Diego, La Jolla, CA, May 1986.

[Jutten and Hérault, 1991] C. Jutten and J. Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[Karhunen and Joutsensalo, 1994] J. Karhunen and J. Joutsensalo. Representation and separation of signals using non-linear pca type learning. *NNks*, 7(1):188–127, 1994.

[Keller, 1962] J. B. Keller. Geometrical theorie of diffraction. *Journal of the Optical society Am.*, 52:116–130, 1962.

[Kerlinger and Pedhazur, 1973] F. N. Kerlinger and E. J. Pedhazur. *Multiple Regression in Behavioral Research.* Holt Rinehart Winston, New York, 1973.

[Kohonen, 1982] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 1982.

[Kohonen, 1989] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 3 edition, 1989.

[Kwok and Yeung, 1997] T.-Y. Kwok and D.-Y. Yeung. Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Transactions on Neural Networks*, 8(3):630–645, May 1997.

[Lallement *et al.*, 1995] Y. Lallement, M. Hilario, and F. Alexandre. Neurosymbolic Integration: Cognitive Grounds and Computational Strategies. In M. DeGlas and Z. Pawlak, editors, *Proceedings World Conference on the Fundamentals of Artificial Intelligence*, Paris, July 1995.

[Lau, 1992] C. Lau, editor. *Neural Networks: Theoretical Foundations and Analysis.* IEEE Press, New York, 1992.

[Lawrence *et al.*, 1997] Steve Lawrence, A.D. Back, A.C. Tsoi, and C. Lee Giles. On the distribution of performance from multiple neural network trials. *IEEE Transactions on Neural Networks*, 8(6):1507–1517, 1997.

[Le Cun *et al.*, 1990] Y. Le Cun, J.S. Denker, and S.A. Solla. Optimal brain damage. In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems II (Denver 1989)*, pages 598–605, San Mateo, 1990. Morgan Kaufmann.

[Le Cun, 1985] Y. Le Cun. Une procédure d'apprentissage pour Réseau à seuil assymétrique. In *Cognitiva 85 : A la Frontière de l'Intelligence Artificielle des Sciences de la Connaissance des Neurosciences*, pages 599–604, Paris, 1985. (Paris 1985), CESTA.

[Lebart *et al.*, 1995] L. Lebart, A. Morineau, and M. Pinon. *Statistique exploratoire multidimensionnelle*. Dunod, 1995.

[Lechevallier, 1997] Y. Lechevallier. *Statistique et méthodes neuronales*, chapter Classification non supervisée. Dunod, 1997.

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

[MacQueen, 1967] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, CA, 1967. University of California Press.

[Martinetz, 1993] Thomas Martinetz. Competitive Hebbian learning rule forms perfectly topology preserving maps. In Stan Gielen and Bert Kappen, editors, *Proc. ICANN'93, Int. Conf. on Artificial Neural Networks*, pages 427–434, London, UK, 1993. Springer.

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

[Michie *et al.*, 1994] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.

[Minsky and Papert, 1969] Marvin Minsky and Seymour Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

[Misra, 1996] M. Misra. Parallel environments for implementing neural network. *Neural Computing Survey*, 1 :48–60, 1996.

[Moerland, 2000] P. Moerland. *Mixture models for unsupervised and supervised learning*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Suisse, 2000.

[Moller, 1993] Martin F. Moller. A scaled conjugate grandient algorithm for fast supervised learning. *Neural Networks*, 6 :525–533, 1993.

[Mozer and Smolensky, 1989] M. C. Mozer and P. Smolensky. Skeletonization : A technique for trimming the fat from a network via relevance assessment. *Connection Science*, 11 :3–26, 1989.

[Mozer, 1989] M.C. Mozer. A focused back-propagation algorithm for temporal pattern recognition. *Complex Systems*, 3 :349–381, 1989.

[Oja, 1982] E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 16 :267–273, 1982.

[Oja, 1992] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5 :927–935, 1992.

[Okumara *et al.*, 1968] Y. Okumara, E. Ohmori, T. Kawano, and K. Fukura. Field strength and variablili in vhf and uhf land mobile radio service. *Rev. Elec. Comm. Lab.*, 16(9-10) :825–873, 1968.

[Pearson, 1901] K. Pearson. On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2 (sixth series) :559–572, 1901.

[Perrault *et al.*, 1996] O. Perrault, J-P. Rossi, T. Balandier, and A. Levy. Predicting field strength with a neural ray-tracing model. Technical report, European cooperation in the field of scientific and technical research, COST 231 TD(96), jan. 1996.

[Pican and Alexandre, 1994a] N. Pican and F. Alexandre. Highly Adaptative Neural Networks for Adaptative Neuro-Control : the OWE Architecture. In *Proceedings IEEE SMC'94, San Antonio (Texas, USA)*, October 1994.

[Pican and Alexandre, 1994b] N. Pican and F. Alexandre. Weight Interpretation with OrthogonalWeight Estimator (OWE) Architecture. In *Proceedings IMACS International Symposium on Signal Processing, Robotics and Neural Networks*, pages 550–555. IEEE, 1994.

[Pican *et al.*, 1994] N. Pican, J.-C. Fort, and F. Alexandre. A Lateral Contribution Learning Algorithm for Multi MLP Architecture. In M. Verleysen, editor, *Proceedings European Symposium on Artificial Neural Networks, Brussels (Belgium)*, pages 229–234. D. Facto Publications, 1994.

[Pican, 1995] N. Pican. *Approche statique et dynamique de la modulation de l'efficacité synaptique dans les réseaux de neurones*. Doctorat d'université, Université Henri Poincaré, Nancy 1, January 1995.

[Pican, 1996] N. Pican. An orthogonal delta weight estimator for mlp architectures. *ICNN'96. Washington, DC, USA*, 1996.

[Pican, 1997] Nicolas Pican. Contextual Kohonen SOM with Orthogonal Weight Estimator principle. In *7th International Conference on artificial Neural Networks - ICANN'97, Lausanne, Suisse*, volume 1327 of *Lecture notes in computer science*, pages 667–672. Springer-Verlag, October 1997.

[Pérez Fontan and Hernando Rabanos, 1993] F. Pérez Fontan and J.M. Hernando Rabanos. Selection of digital terrain database parameters for radiocommunications system planning applications. *IEEE Transaction on Broadcasting*, 39(3) :335–342, sept. 1993.

[Quinlan, 1986] J R Quinlan. Induction of decision trees. *Machine Learning*, 1(1) :81–106, 1986. QUINLAN86.

[Rao, 1960] C. R. Rao. Multivariate analysis : an indispensable statistical aid in applied research. *Sankhyā*, 22 :317–338, 1960.

[Reed, 1993] Russel Reed. Pruning algorithms — A survey. *IEEE Transactions on Neural Networks*, 4(5) :740–746, 1993.

[Reiss and Taylor, 1991] M. Reiss and J.G Taylor. Storing temporal sequences. *Neural Networks*, 4 :773–787, 1991.

[Remm, 1996] J.F. Remm. *Extraction de connaissances par réseaux neuronaux : application au domaine du radar*. PhD thesis, Université Henri Poincaré, Nancy I, 1996.

[Riedmiller and Braun, 1993] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning : The RPROP algorithm. In *Proc. of the IEEE Intl. Conf. on Neural Networks*, pages 586–591, San Francisco, CA, April 1993.

[Ripley, 1996] Brian D. Ripley. *Pattern recognition and Neural networks*. Cambridge University Press, Cambridge, UK, 1996.

[Rosenblatt, 1961] F. Rosenblatt. *Principles of Neurodynamics : Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington DC, 1961.

[Rouzier, 1998] Sophie Rouzier. *Réseaux neuronaux et Modularité*. PhD thesis, Institut National Polytechynique de Grenoble, 1998.

[Rumelhart and Zipser, 1985] D. E. Rumelhart and D. Zipser. Feature discovery by competitive learning. *Cognitive Science*, 9 :75–112, 1985.

[Rumelhart and Zipser, 1986] D. E. Rumelhart and D. Zipser. Feature discovery by competitive learning. In D. Rumelhart, J. McClelland, and the PDP Research Group, editor, *Parallel Distr. Processing*, volume 1, page 151. MIT Press, Cambridge, MA, 1986.

[Rumelhart *et al.*, 1986a] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323 :533–536, 1986.

[Rumelhart *et al.*, 1986b] D. E. Rumelhart, J. L. McClelland, and the PDP research group. *Parallel distributed processing : Explorations in the microstructure of cognition. Vols. I and II.* MIT Press, Cambridge, MA, 1986.

[Sammon, 1969] John W. Jr Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 5 :401–409, 1969.

[Sanger, 1989] T.D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2 :459–473, 1989.

[Saporta, 1990] G. Saporta. *Probabilités, Analyse des données et Statistiques*. Éditions Technip, 1990.

[Sarle, 1994] Warren S. Sarle. Neural networks and statistical models. In *Proceedings of the Nineteenth Annual SAS Users Group International Conference, April, 1994*, pages 1538–1550, Cary, NC, April 1994. SAS Institute.

[Sethi, 1990] I. K. Sethi. Entropy nets : from decision trees to neural networks. *Proceedings of the IEEE*, 78 :1605–1613, 1990. [Reprinted in [Lau, 1992]].

[Sharkey, 1996] A.J.C. Sharkey. Special issue : Combining artificial neural nets : Ensemble approaches. *Connection Science*, 8(3&4), december 1996.

[Sharkey, 1997] A.J.C. Sharkey. Special issue : Combining artificial neural nets : Modular approaches. *Connection Science*, 9(1), march 1997.

[Sharkey, 1999] Amanda J.C. Sharkey, editor. *Combining Artificial Neural Nets : Ensemble and Modular Multi-Net Systems*. Springer-Verlag London Ltd, 1999.

[Simon, 2000] A. Simon. *Outils classificatoires par objets pour l''extraction de connaissances dans des bases de données*. PhD thesis, Université Henri Poincaré, Nancy I, 2000.

[Sirat and Nadal, 1990] J. A. Sirat and J.-P. Nadal. Neural trees : a new tool for classification. *Network*, 1 :423–438, 1990.

[Spearman, 1904] C. Spearman. General intelligence, objectively determined and measured. *American Journal of Psychology*, 15 :201–293, 1904.

[Stocker *et al.*, 1993] K.E. Stocker, B.E. Gschwendtner, and F.M. Landstorfer. Neural network approach to prediction of terrestrial wave propagation for mobile radio. *IEE Proceeding-H*, 140(4) :315–320, 1993.

[Stornetta *et al.*, 1988] W.S. Stornetta, T. Hogg, and B.A. Huberman. A dynamical approach to temporal pattern processing. In D.Z. Anderson, editor, *Neural Information*

Processing Systems, pages 750–759, New York, 1988. (Denver 1987), American Institute of Physics.

[Sutton and Barto, 1981] R. S. Sutton and A. G. Barto. Toward a modern theory of adaptative network : Expectation and prediction. *Pattern Recognition*, 88(2) :135–170, 1981.

[Thiria *et al.*, 1997] S. Thiria, Y. Lechevallier, O. Gascuel, and S. Canu, editors. *Statistique et méthodes neuronales*, chapter Représentation structurée. dunod, 1997.

[Thurstone, 1947] L. L. Thurstone. *Multiple factor analysis*. University of Chicago Press, 1947.

[Tollenaere, 1990] Tom Tollenaere. SuperSAB : Fast adaptive back propagation with good scaling properties. *Neural Networks*, 3(5) :561–573, 1990.

[Touretzky, 1989] David S. Touretzky, editor. *Advances in Neural Information Processing Systems 2*, San Mateo, CA, 1989. Morgan Kaufman Publishers Inc.

[Towell and Shavlik, 1993] Geoffrey G. Towell and Jude W. Shavlik. The extraction of refined rules from knowledge-based neural networks. *Machine Learning*, 13(1) :71–101, 1993.

[Vapnik, 1995] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[von der Malsburg, 1973] Christof von der Malsburg. Self-organizing of orientation sensitive cells in the striate cortex. *Kybernetik (Biological Cybernetics)*, 14 :85–100, 1973.

[Walfish and Bertoni, 1988] J. Walfish and L. Bertoni. A theoritical model of uhf propagation in urban environments. *IEEE Transaction on Antennas and Propagation*, 36(12) :1788–1796, 1988.

[White, 1992] H. White. *Artificial Neural Networks : Approximation and Learning Theory*. Blackwell, Oxford, 1992.

[Widrow and Hoff, 1960] B. Widrow and M. E. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, volume 4, pages 96–104. IRE, New York, 1960.

[Widrow and Lehr, 1990] B. Widrow and M. A. Lehr. 30 years of adaptive neural networks : Perceptron, Madaline, and backpropagation. *Proceedings of the IEEE*, 78(9) :1415–1442, 1990.

[Williams, 1995] Peter M. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7(1) :117–143, 1995.

[Willshaw and von der Malsburg, 1976] David J. Willshaw and Christoph von der Malsburg. How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London, Series B*, 194 :431–445, 1976.

Etude de la construction par réseaux neuromimétiques de représentations interprétables

Résumé : Les réseaux de neurones artificiels sont de bons outils de modélisation (efficaces, facilement adaptables et rapides) mais ils ont la réputation d'être difficiles à interpréter et sont généralement comparés à des boîtes noires dont il n'est pas facile de comprendre l'organisation interne, pourtant responsable de leurs bonnes performances. Pour obtenir une meilleure compréhension du fonctionnement des réseaux connexionnistes et une validation de leur utilisation en tant qu'outils d'acquisition de connaissances, nous avons, dans un premier temps, réuni divers travaux théoriques pour montrer les points communs existant entre certains réseaux de neurones classiques et des méthodes statistiques de régression et d'analyses de données. Dans un deuxième temps et à la lumière de ce qui précède, nous avons expliqué les particularités de réseaux connexionnistes plus complexes, tels que des réseaux dynamiques ou modulaires, afin d'exploiter leurs avantages respectifs en concevant un nouveau modèle d'extraction de connaissances adapté à la complexité du phénomène à modéliser. Les réseaux connexionnistes que nous avons réunis et interprétés et le modèle que nous avons développé peuvent, à partir des données, enrichir la compréhension du phénomène en analysant et en organisant les informations par rapport à la tâche à accomplir comme nous l'illustrons à travers une application de prédiction dans le domaine des télécommunications où la connaissance du domaine ne suffit pas à modéliser correctement le phénomène. Les possibilités d'application de notre travail sont donc larges et s'inscrivent dans le cadre de la fouille de données et dans le domaine des sciences cognitives.

Mots-clé : réseaux de neurones artificiels, extraction de connaissances, fouille de données, télécommunications

Artificial neural networks arrangements leading to knowledge extraction

Abstract : Artificial neural networks constitute good tools for certain types of computational modelling (being potentially efficient, easy to adapt and fast). However, they are often considered difficult to interpret, and are sometimes treated as black boxes. However, whilst this complexity implies that it is difficult to understand the internal organization that develops through learning, it usually encapsulates one of the key factors for obtaining good results. First, to yield a better understanding of how artificial neural networks behave and to validate their use as knowledge discovery tools, we have examined various theoretical works in order to demonstrate the common principles underlying both certain classical artificial neural network, and statistical methods for regression and data analysis. Second, in light of these studies, we have explained the specificities of some more complex artificial neural networks, such as dynamical and modular networks, in order to exploit their respective advantages in constructing a revised model for knowledge extraction, adjusted to the complexity of the phenomena we want to model. The artificial neural networks we have combined (and the subsequent model we developed) can, starting from task data, enhance the understanding of the phenomena modelled through analysing and organising the information for the task. We demonstrate this in a practical prediction task for telecommunication, where the general domain knowledge alone is insufficient to model the phenomena satisfactorily. This leads us to conclude that the possibility for practical application of our work is broad, and that our methods can combine with those already existing in the data mining and the cognitive sciences.

keywords : Artificial neural networks, knowledge extraction, data mining, telecommunication