

N° d'ordre: 03/ 2009- D/ PH

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA
RECHERCHE SCIENTIFIQUE
UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE
HOUARI BOUMEDIENNE
FACULTÉ DE PHYSIQUE



THESE

Présentée pour l'obtention du diplôme de DOCTORAT

En : PHYSIQUE

Spécialité : Physique Théorique

Par : Mme Leila Ait Gougam-Tribeche

Thème

**Application des réseaux de neurones- ondelettes
à l'approximation de tâches et à l'analyse
de certaines oscillations localisées**

Soutenue publiquement le 12/ 07/ 2009, devant le jury composé de :

Mr.	T. H. ZERGUINI	Professeur	U.S.T.H.B	Président
Mme.	F. MEKIDECHE	Professeur	U.S.T.H.B	Directrice de thèse
Mr.	B. G. GIRAUD	Directeur de Recherche	C.E.A- Saclay	Examineur
Mr.	Y. BOUDOUMA	Professeur	U.S.T.H.B	Examineur
Mr	M. BENTAIBA	Professeur	U. Blida	Examineur
Mr	B. MEZIANI	Maître de Conférences	U. Bejaia	Examineur

On pense communément que “l’ordinateur de l’avenir” sera massivement parallèle et tolérera les erreurs. Toutefois la conception d’une telle machine s’étant avérée étonnamment difficile, nous aurions abandonné depuis longtemps si le cerveau n’était pas une preuve vivante que le traitement parallèle et tolérant les erreurs est possible et très efficace.

John S. Denker, 1985

(Dans *Les Rêves de la Raison*)

REMERCIEMENTS

Le travail présenté dans cette thèse a été réalisé au sein du Laboratoire de Physique Théorique de la Faculté des Sciences-Physique de l'Université des Sciences et de la Technologie Houari Boumediene d'Alger.

Je tiens en premier lieu à adresser mes plus vifs remerciements à ma directrice de thèse Madame F. MEKIDECHE-CHAFI, Professeur à l'USTHB, pour son encadrement sans faille, ses qualités humaines et scientifiques dont elle a toujours fait preuve. Ses précieux conseils, sa constante disponibilité, son souci de clarté, de rigueur et de précision et ses encouragements sans cesse renouvelés m'ont été d'un grand secours et m'ont permis, je l'espère, de rendre fructueuses ces années passées sous sa direction.

Je remercie vivement le Professeur T.H. ZERGUINI qui nous a honorés par sa présence en acceptant de présider le jury de la présente thèse. Nous lui devons toujours l'introduction, dans notre Laboratoire, de la Physique des Plasmas.

Je suis particulièrement sensible à l'honneur que me fait Monsieur B.G. GIRAUD, Directeur de Recherche au CEA, en acceptant de juger ce travail. Durant l'évolution de cette thèse, il a toujours été disponible et très impliqué en dépit de l'éloignement. Ses messages réguliers témoignent du regard attentif qu'il a toujours porté sur ce travail. Merci infiniment pour les nombreuses idées proposées, les conseils avisés, les réponses apportées à nos diverses questions et surtout les précieux conseils issus d'un savoir, semble-t-il, inépuisable. Qu'il trouve à travers ces quelques lignes l'expression de ma plus sincère gratitude.

J'adresse mes plus vifs remerciements à Monsieur Y. BOUDOUMA, Professeur à l'USTHB, d'avoir accepté de juger ce travail de thèse. Sa présence constitue pour moi un grand hon-

neur. Merci pour vos remarques, vos critiques, vos conseils et simplement, pour l'intérêt que vous portez à mon travail.

Je remercie également Monsieur M. BENTAIBA, Professeur à l'Université de BLIDA, et Monsieur B. MEZIANI, Maître de Conférences à l'Université de BEJAIA, d'avoir accepté de faire partie de mon jury de thèse.

Je souhaite aussi témoigner toute ma reconnaissance au Professeur J. P. ANTOINE pour m'avoir fait découvrir toute la richesse et l'élégance de la théorie des ondelettes. Merci, encore une fois, Monsieur le Professeur.

Je ne saurais oublier mon collègue A. CHIKHI pour les nombreuses et fructueuses discussions que nous avons eues et qui ont permis d'éclaircir de nombreux problèmes. Je tiens à lui témoigner toute ma gratitude.

Je remercie S. Bahamida pour son aide précieuse lors de la confection des schémas des réseaux de neurones. Que toutes mes amies, en particulier Dalila et Nassima, trouvent, à travers ces quelques lignes, l'expression de ma sincère amitié.

J'aimerais associer à ce moment particulier de ma vie mes parents qui m'ont donné l'éducation nécessaire, l'amour et m'ont montré la voie. Ils ont toujours été et resteront des piliers indispensables, simplement par leur présence. Un grand merci à mes frères, mes sœurs et ma belle famille pour leur soutien et leur affection.

Enfin, *last but not least*, un grand merci à mon mari qui a guidé mes premiers pas dans le domaine très ardu des phénomènes non linéaires dans les plasmas. Je le remercie pour toutes les explications face aux problèmes ponctuels que je lui soumettais. Je resterai toujours impressionnée par son intuition remarquable, sa réelle vision scientifique et son sens de la critique.

Enfin, un gros merci chaleureux à nos enfants YANIS, RAYAN et CAMELIA, qui représentent ma plus grande réussite. Certes, je ne saurais leur faire comprendre ce qu'est un réseau de neurones mais je me contenterai de leur dire que je puise toute mon énergie de leur simple sourire.

Table des matières

Introduction	8
1 Analyse en ondelettes	15
1.1 Introduction	16
1.2 Transformée continue en ondelettes	18
1.2.1 Transformée en ondelettes	21
1.3 Transformée en ondelettes discrète	23
1.3.1 Discrétisation de la transformée continue	23
1.3.2 Analyse multirésolution et bases orthonormées d'ondelettes	24
2 Réseaux de neurones	26
2.1 Introduction	27
2.2 Neurone biologique	29
2.3 Neurone formel	31
2.4 Fonction d'activation	33
2.5 Réseaux de neurones formels	34
2.5.1 Réseaux de neurones non bouclés	36
2.5.2 Réseaux de neurones bouclés	37
2.6 Processus d'apprentissage	41
2.6.1 Algorithmes d'optimisation	42

2.7	Applications des réseaux de neurones	49
2.7.1	Propriétés spécifiques aux réseaux de neurones	49
2.7.2	Mise en oeuvre des réseaux de neurones formels	53
2.7.3	Domaines d'applications	54
		55
3	Investigation systématique d'un réseau de neurones pour l'approximation de tâches	56
3.1	Introduction	57
3.2	Architecture du réseau et apprentissage	59
3.3	Résultats numériques et discussion	61
4	Analyse d'une structure solitaire à l'aide d'un schéma de réseaux de neurones	80
4.1	Introduction	81
4.2	Onde acoustique poussiéreuse solitaire	82
4.2.1	Analyse des conditions d'existence par la méthode du pseudo-potentiel	82
4.2.2	Analyse des conditions d'existence à l'aide d'un réseau de neurones .	84
4.3	Onde solitaire dans un plasma complexe à charge variable	87
4.3.1	Modèle théorique et équations de base	89
4.3.2	Analyse des conditions d'existence par la méthode du pseudo-potentiel	92
	Conclusion générale	100

Introduction

Modéliser un phénomène physique n'est pas toujours aisé et peut même, parfois, devenir problématique. Les nombreuses expériences réalisées permettent l'acquisition d'un nombre de plus en plus important de données. Celles-ci revêtent diverses formes et sont généralement décrites par un signal, c'est à dire , par l'association d'un temps ou d'une position à une grandeur donnée. De nombreux domaines d'applications des sciences font appel à des techniques d'approximation. Il s'agit, dans la majeure partie des cas, de fournir des bases de fonctions permettant de proposer une approximation d'un signal issu de mesures physiques. L'approximation proposée doit, cependant, être la plus proche possible du signal d'origine. La multitude des techniques mises en oeuvre témoigne de la diversité des objectifs recherchés dans cette tâche. Ce peut être un souci de modélisation d'un phénomène physique, de l'extraction de certaines caractéristiques d'un signal donné ou encore de la prédiction de séries temporelles. Un des critères de choix d'un approximateur parmi d'autres est son aptitude à approcher la fonction désirée avec la structure la plus simple et un nombre de paramètres mis en jeu le plus petit possible. En effet, dans la pratique, le nombre de fonctions nécessaires pour réaliser une approximation est un critère important dans la préférence d'un approximateur par rapport à un autre. En d'autres termes, le concepteur de modèles doit toujours faire en sorte que le nombre de paramètres ajustables soit le plus faible possible: on dit que l'on cherche l'approximation la plus parcimonieuse.

Le problème d'approximation de tâche peut-être avantageusement traité par le modèle des réseaux de neurones artificiels. Ces nouvelles techniques neuronales se sont avérées très adaptées pour la résolution de tels problèmes. En effet, de nos jours, les réseaux de neurones ne cessent d'élargir leur champ d'application pour de nombreuses raisons. Nous citerons, d'une part, la propriété d'approximation parcimonieuse qui est un des résultats mathématiques solides établis[1] et qui explique pourquoi les réseaux de neurones constituent d'excellents modèles non linéaires. D'autre part, les techniques algorithmiques d'apprentissage ont fait de très grands progrès grâce notamment à l'utilisation de méthodes d'optimisation non linéaires efficaces qui, associées à l'algorithme de rétropropagation pour l'évaluation du gradient de l'erreur (mesure de l'écart entre la sortie du réseau et

la tâche à approcher), permettent des apprentissages rapides et précis. Grâce à ces propriétés, les réseaux de neurones, lorsqu'ils sont convenablement mis en oeuvre, permettent d'améliorer les résultats obtenus à partir d'autres méthodes classiques de modélisation non linéaires.

Les réseaux de neurones artificiels, dont la conception est très schématiquement inspirée du fonctionnement de vrais neurones biologiques, sont fondés sur des modèles qui tentent d'expliquer comment les cellules du cerveau et leurs interconnexions parviennent, d'un point de vue global, à exécuter des tâches complexes. Initialement, ces systèmes artificiels qui stockent et retrouvent l'information de manière "similaire" à celle du cerveau se sont avérés particulièrement adaptés aux traitements en parallèle des problèmes complexes comme la reconnaissance des formes[2] par exemple. Dans les réseaux de neurones artificiels, de nombreux processeurs appelés cellules ou unités, capables de réaliser des calculs élémentaires, sont structurés en couches successives capables d'échanger des informations au moyen de connexions qui les relient. Cette architecture connexionniste s'inspire de l'organisation neuronale du cerveau humain.

Les réseaux de neurones trouvent leur origine en 1943 avec la notion de neurone formel de McCulloch et Pitts[3] qui établirent un premier modèle mathématique du neurone biologique. Les réseaux de neurones vont prendre leur essor dans les années cinquante avec l'Intelligence Artificielle. En particulier, Rosenblatt[4] établit un premier modèle de réseaux de neurones capable d'apprendre, le perceptron, pour simuler le fonctionnement de la rétine. Cependant, ce modèle montre rapidement ses limites[5] et l'engouement pour les réseaux de neurones se tarit à la fin des années soixante. L'intérêt pour les réseaux de neurones va être ravivé par le développement de nouveaux modèles (comme les réseaux de Hopfield[6]) et par la redécouverte de l'algorithme de rétropropagation de l'erreur par Rumelhart[7], originellement inventé par Werbos[8]. Les réseaux à propagation ou réseaux feed-forward, réseaux non bouclés, sont aujourd'hui les modèles de réseaux de neurones les plus répandus. Ils sont largement utilisés dans le cadre de la modélisation statistique: classification, reconnaissance de formes, régression non-linéaire, prédiction de séries temporelles,...etc.

La propriété fondamentale des réseaux de neurones, l'approximation universelle parcimonieuse, fait de ceux-ci une représentation mathématique très avantageuse pour les modélisations statique et dynamique non linéaires des phénomènes. En effet, il a été démontré qu'un réseau de neurones de type perceptron, ayant une couche cachée est un «approximateur universel parcimonieux»[1, 9], c'est à dire qu'il peut approcher toute fonction multivariable non linéaire et ses dérivées avec une précision à priori fixée, pourvu qu'il soit soumis à un apprentissage efficace. Ce potentiel d'approximation est cependant difficile à exploiter car il n'existe à ce jour aucune méthodologie précise permettant de réaliser avec certitude l'approximation désirée pour un système donné. C'est pourquoi, le choix et l'application d'un modèle neuronal demeure un domaine de recherche très actif contrairement aux modèles classiques.

Les premiers réseaux de neurones utilisant des fonctions sigmoïdes comme fonctions d'activation, sont issus de travaux à connotation biologique[1, 9]. Par la suite, une des évolutions les plus marquantes des réseaux de neurones formels a été l'abandon de la métaphore biologique au profit de fondements théoriques solides. Cet effort de recherche s'est tout d'abord dirigé vers les réseaux de fonctions radiales (RBF, Radial Basis Functions), en particulier les gaussiennes[10]. Broom[11] semble être parmi les premiers à proposer l'idée d'utiliser des réseaux RBF pour l'approximation de fonctions non linéaires. Récemment[12], des familles de fonctions, issues du traitement du signal et de l'image, appelées ondelettes ont été utilisées avec succès pour la résolution de problèmes d'approximation de fonctions.

De manière analogue à la théorie des séries de Fourier, les ondelettes sont principalement utilisées pour la décomposition de fonctions[13]. Cette décomposition consiste à écrire une fonction donnée comme une somme pondérée de fonctions obtenues à partir d'opérations simples effectuées sur une fonction principale appelée ondelette-mère. Ces opérations consistent en des translations et dilatations de la variable de la fonction.

La décomposition en ondelettes étant une méthode d'approximation de fonctions, un lien entre cette décomposition et les réseaux de neurones a été établi par de nombreux

auteurs[14, 15] où la fonction d'activation considérée est une ondelette. Ce type de réseau de neurones est communément désigné sous l'appellation de réseau d'ondelettes.

Récemment, Giraud et Touzeau[16] ont utilisé une architecture classique à une seule couche cachée pour montrer que cette dernière produit une analyse multi-échelles raisonnablement efficace, pratique et robuste. En suivant l'évolution des paramètres d'échelles durant les premiers pas de l'apprentissage, Giraud et Touzeau ont noté et mis en évidence la fusion de deux paires d'échelles. Ils ont alors suggéré que de nouvelles cellules peuvent spontanément émerger durant le processus d'apprentissage et éventuellement représenter les dérivées de la tâche élémentaire par rapport à leurs paramètres d'échelle. Dans le but de confirmer et d'étayer cette suggestion, nous avons considéré différentes tâches contenant des dérivées d'ordre p de la fonction d'activation par rapport aux paramètres d'échelle. Cette étude nous a permis de généraliser le résultat obtenu au cas où $(p + 1)$ paramètres d'échelles fusionnent durant le processus d'apprentissage. En effet, nous avons montré que des dérivées d'ordre p , à savoir, $\frac{\partial^p f}{\partial \lambda^p}$, vont spontanément émerger dans la base fonctionnelle. Notons que pour les tâches qui adoptent un tel comportement, il nous a été possible d'améliorer l'approximation de la tâche considérée et de réduire considérablement le temps d'apprentissage en implémentant directement les tâches élémentaires ainsi que leurs dérivées dans la base fonctionnelle. Ceci nous a amené à montrer, à travers une étude systématique, les performances des réseaux d'ondelettes dans les problèmes d'approximation de fonctions[17]. En effet, en tant que modèle connexionniste, les réseaux d'ondelettes ont l'avantage de pouvoir bénéficier de nombreux résultats de l'analyse en ondelettes. Nous montrerons, en particulier, que ces réseaux présentent des capacités propres, liées notamment à la localité des fonctions utilisées. Nous comparerons, par la suite, les performances de ces réseaux d'ondelettes relativement à deux autres modèles connexionnistes bien connus: les perceptrons multicouches (MLP à fonction d'activation sigmoïdale) et les réseaux à fonctions radiales (RBF).

Dans le but de mettre en évidence l'efficacité de l'approximation à l'aide des réseaux de neurones dans les problèmes physiques, il nous a semblé nécessaire de ne pas nous contenter de cette étude, mais de montrer à travers un problème physique bien déterminé la validité

d'une telle approche[18] en empruntant à la physique des plasmas quelques modèles non linéaires[19, 20, 21]. Pour cela, nous avons choisi d'analyser les conditions d'existence des structures localisées (ondes solitaires) associées à deux modèles de plasma différents: l'un intégrable et l'autre non intégrable. Cette partie nous a permis de déterminer l'expression analytique explicite du pseudo-potentiel lorsque le système dynamique correspondant est non intégrable. L'intérêt d'une telle approximation est à rechercher du côté expérimental. Rappelons que lors des expériences sur les plasmas de laboratoire, l'opérateur, à travers une série de mesures de sonde, n'a accès, généralement, qu'aux différentes températures des espèces de particules présentes dans le plasma ainsi qu'aux profils de leurs densités respectives. Il devient alors possible de prédire l'existence ou non d'une structure solitaire lorsque les paramètres du plasma sont, à priori, connus. Notre méthode d'approximation peut alors apparaître, à bien des égards, comme une alternative, certes perfectible mais fiable, pour la description de la dynamique d'un système non intégrable.

La structure de cette thèse se présente sous la forme d'un premier chapitre où sont introduites les techniques d'analyse en ondelettes. Il s'agit beaucoup plus d'une présentation succincte qui se limitera à la présentation des résultats utiles à la mise en oeuvre des fonctions ondelettes dans les réseaux connexionnistes.

Le chapitre deux est consacré à la présentation des définitions et rappels concernant les réseaux de neurones classiques. Nous présentons notamment les architectures des réseaux bouclés et non bouclés. Un aperçu sur l'apprentissage de ces modèles est donné ainsi que les méthodes d'apprentissage qui s'appuient sur la rétropropagation de l'erreur. Il s'agit en fait de trouver les paramètres du modèle qui minimisent l'erreur. L'apprentissage étant un problème d'optimisation non linéaire, il est généralement résolu par des méthodes basées sur le gradient. Dans un souci de clarté et afin de faciliter la compréhension, il nous a paru nécessaire de distinguer, dans ce chapitre, l'étape de calcul du gradient de la fonction de coût par rapport aux paramètres (rétropropagation du gradient de l'erreur) de l'étape de modification des paramètres (méthodes du gradient simple[22], de quasi-Newton[15], de Levenberg-Marquardt[15],...etc.). Nous terminons le chapitre par un énoncé des pro-

priétés spécifiques aux réseaux de neurones, suivi d'une présentation de la méthodologie pour l'application et la mise en oeuvre de ces techniques neuronales. Cette partie s'achève par une revue générale des domaines d'applications.

Dans le chapitre trois, les capacités des réseaux d'ondelettes sont étudiées sur des problèmes d'approximation de fonctions. Dans cette étude, nous avons effectué une investigation systématique faisant intervenir tous les paramètres susceptibles d'améliorer l'approximation des tâches considérées. Sachant que les fonctions ondelettes sont caractérisées par leur décroissance rapide, elles sont donc négligeables en dehors d'une zone désignée comme la zone d'influence associée à l'ondelette. De ce fait, dans le cas des réseaux d'ondelettes, nous parlerons de la zone d'influence associée à chaque neurone, tenant compte des paramètres de translation et de dilatation. Les paramètres de translation ont pour effet de déplacer la zone d'influence par rapport à l'origine et les paramètres de dilatation de modifier l'étendue de cette zone. Nous montrons donc que cette possibilité de modifier la réponse des cellules cachées (et donc celle du réseau) en modifiant les paramètres des ondelettes, constitue une particularité du modèle tel qu'il a été défini. Cette étude nous permettra également une bonne compréhension des paramètres du modèle et de leur évolution au cours de la phase d'apprentissage.

Le chapitre quatre est consacré à la validation de nos résultats théoriques en les appliquant à un problème physique; l'analyse de certaines oscillations localisées (ondes solitaires) associées à différents modèles de plasma, ainsi que la détermination de l'expression analytique du pseudo-potentiel pour des systèmes dynamiques non intégrables.

Enfin, une conclusion générale est consacrée au rappel de la démarche globale de recherche et des objectifs. Elle s'attellera en plus à la présentation des différents prolongements et perspectives envisageables tant du point de vue théorique qu'appliqué.

CHAPITRE

1

Analyse en ondelettes

1.1 Introduction

Les perspectives de recherche actuelles sont dominées par les applications industrielles de l'analyse et du traitement du signal. Ce dernier est devenu une composante essentielle de l'activité scientifique et technologique contemporaine. Il est utilisé dans les télécommunications, dans la transmission et l'analyse des images fournies par les satellites, dans l'imagerie médicale, tout comme dans l'analyse et l'interprétation de séries temporelles complexes. Dans ce contexte, la compréhension d'un problème physique donné consiste à traiter ces signaux et d'y distinguer l'information considérée comme importante. Une première approche consiste à analyser le contenu fréquentiel de ces signaux au moyen de la transformée de Fourier. Malheureusement, cette méthode atteint vite ses limites. La transformée de Fourier, de par sa globalité, est incapable de localiser les variations abruptes (transitoires) d'une fonction, généralement séparées par des zones aux comportements plus doux. L'idée originale sur laquelle se basent les ondelettes est apparue vers les années 1940 grâce au physicien Denis Gabor[23], qui a introduit la notion de la transformée de Fourier à fenêtre glissante dans le but de remédier au problème de localisation temps- fréquence en proposant de multiplier le signal par une fonction localisée dans le temps (fenêtre) et ensuite appliquer la transformée de Fourier. L'inconvénient de cette transformée est que la taille et la forme de la fenêtre sont inchangées au cours de l'analyse. C'est à partir de là que sont nées les ondelettes qui s'adaptent d'elles mêmes à la taille et aux caractéristiques qu'elles recherchent. Elles sont très étendues pour étudier les basses fréquences (les grandes échelles) et très fines pour étudier des phénomènes plus transitoires (hautes fréquences, ou petites échelles). Afin d'employer un poncif du genre, on dit souvent que les ondelettes fonctionnent comme un "microscope mathématique" dont la résolution serait fixée par le paramètre de dilatation et l'optique serait déterminée par le choix de l'ondelette.

L'analyse par ondelettes a été donc introduite au début des années 1980, dans un contexte d'analyse du signal. Il s'agissait à l'époque de donner une représentation des signaux permettant de faire apparaître simultanément des informations temporelles (localisation dans le temps, durée) et fréquentielles, facilitant par là l'identification des caractéristiques physiques

de la source du signal. Les balbutiements de cette analyse sont généralement attribués à un géophysicien, Jean Morlet, motivé par la détection de singularités dans des signaux sismiques, et aidé par un mathématicien, Alex Grossman[24]. Leur collaboration a conduit à la transformée en ondelettes continue (CWT). A la base de la CWT se trouve l'idée de décomposer un signal, une fonction, en briques élémentaires simples, à la manière de l'intégrale de Fourier ou, plus généralement, de l'analyse harmonique. Dans le cas des ondelettes, on obtient ces "atomes" à partir d'une fonction de départ, l'ondelette mère, en agissant au moyen de translations et dilatations. La formalisation de l'analyse a été faite ensuite par quelques mathématiciens jusqu'aux travaux en commun de Stéphane Mallat (motivé par des considérations pratiques en imagerie) et Yves Meyer[25] qui ont généralisé l'analyse en ondelettes à l'analyse multirésolution basée sur une hiérarchie d'espaces d'approximation imbriqués, ce qui a conduit à des algorithmes rapides réalisant la transformée discrète en ondelettes. Puis Ingrid Daubechies[26] a généralisé l'analyse multirésolution en construisant des ondelettes à support compact performantes numériquement. Certains utilisent donc les ondelettes pour leur régularité, d'autres pour leur simplicité et leur rapidité de calcul. Certains les utilisent comme une nouvelle méthode spectrale pour résoudre les équations aux dérivées partielles (edp)[27], dans un but d'homogénéisation[28, 29, 30], d'analyse et de modélisation de phénomènes[31]. Les développements des années 90 sont allés en direction de performances numériques, de l'étude et de la résolution à partir d'ondelettes des équations aux dérivées partielles. Quelques détails concernant l'histoire de cette épopée mathématique des années 1980-1990 se trouvent dans l'ouvrage de vulgarisation de Hubbard[32]. Beaucoup de disciplines se sont donc intéressées au développement des ondelettes et de l'analyse multirésolution.

Les ondelettes n'ont depuis lors pas cessé de se développer dans de nombreuses disciplines et de trouver de nouveaux champs d'application. Elles sont devenues un outil puissant et efficace dans de nombreux domaines: en analyse harmonique, pour l'analyse fonctionnelle de données médicales (en vue d'établir un diagnostic), pour la vision par ordinateur, en

traitement du signal et en analyse de turbulences. Leur succès est dû à leur adaptativité aux données et à leur facilité d'implémentation.

Avec quelques années de recul, nous réalisons maintenant que ce sont ces origines "scientifiquement cosmopolites" qui ont donné à la théorie toute sa richesse et sa beauté, en même temps que ses vastes domaines d'application.

Nous présenterons l'analyse par ondelette de façon succincte sans trop rentrer dans les détails. Le lecteur intéressé pourra trouver de plus amples explications sur la théorie des ondelettes dans l'ouvrage de Mallat[13].

1.2 Transformée continue en ondelettes

Soit Ψ une fonction suffisamment régulière et bien localisée. Cette fonction est appelée "ondelette" si elle vérifie la condition suivante dite condition d'admissibilité dans le domaine fréquentiel[33].

$$\int_0^{+\infty} \left| \hat{\Psi}(\omega) \right|^2 \frac{d\omega}{\omega} = \int_{-\infty}^0 \left| \hat{\Psi}(\omega) \right|^2 \frac{d\omega}{\omega} < +\infty \quad (1.2.1)$$

où $\hat{\Psi}$ désigne la transformée de Fourier de Ψ . Cette condition a pour effet de restreindre l'ensemble des fonctions admissibles comme fonctions analysantes. Elle entraîne, en particulier, que l'ondelette est d'intégrale nulle, soit

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \quad (1.2.2)$$

Cette condition minimale est souvent renforcée en exigeant que l'ondelette ait $(m + 1)$ moments nuls, c'est à dire vérifie la condition suivante[13]

$$\int_{-\infty}^{+\infty} t^k \Psi(t) dt = 0 \text{ pour } k = 0, \dots, m \quad (1.2.3)$$

On dit qu'une ondelette oscille, pendant un certain temps, comme une onde et se localise grâce à un amortissement. L'oscillation d'une ondelette se mesure par le nombre de moments

nuls et sa localisation s'évalue par le segment sur lequel elle prend des valeurs assez différentes de zéro.

Les ondelettes forment donc une représentation des fonctions de $L^2(\mathbb{R})$ dans une base dont les fonctions sont bien localisées en espace et en fréquence. Elles conservent les avantages de la transformée de Fourier sans les inconvénients liés au manque de localisation. L'ondelette $\Psi(t)$ appelée "ondelette mère" engendre les autres ondelettes $\Psi_{a,b}(t)$, $a > 0$ et $b \in \mathbb{R}$, de la famille, par translation dans le temps d'un paramètre b et par dilatation d'un paramètre échelle a . Ce dernier correspond à l'inverse de la fréquence: plus a est petit, moins l'ondelette (la fonction analysante) est étendue temporellement et par conséquent la fréquence centrale de son spectre est plus élevée. La dilatation change l'étendue temporelle de l'ondelette ($a > 1$ augmente son étendue temporelle, $a < 1$ la réduit), tandis que la translation localise l'ondelette dans le temps. Nous avons alors

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi \left(\frac{t-b}{a} \right), \quad a > 0, b \in \mathbb{R} \quad (1.2.4)$$

La famille $\{\Psi_{a,b}\}$ est la famille d'ondelettes associée à Ψ . Les ondelettes gardent la même forme et le même nombre d'oscillations et sont les translatées-dilatées d'une même fonction. Nous donnons une illustration de l'influence des paramètres a et b sur l'allure de l'ondelette analysante sur les figures 1, 2, 3 et 4 suivantes.

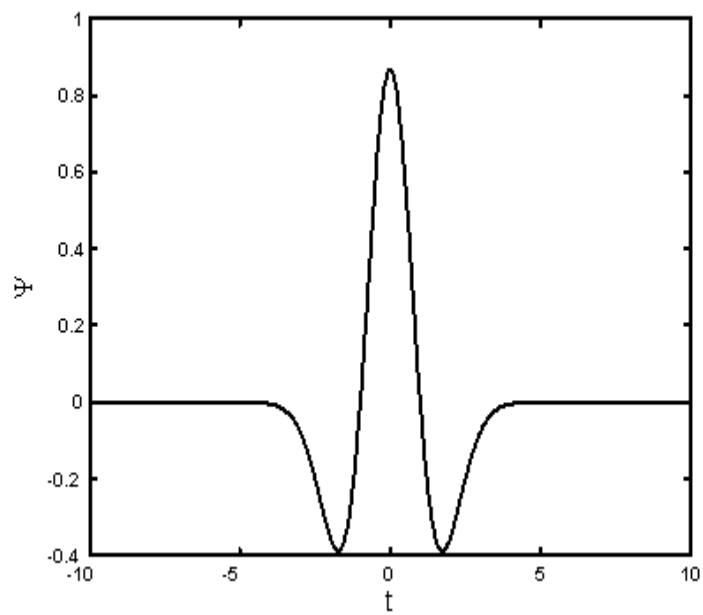


Figure 1: Ondelette "Chapeau-Mexicain".

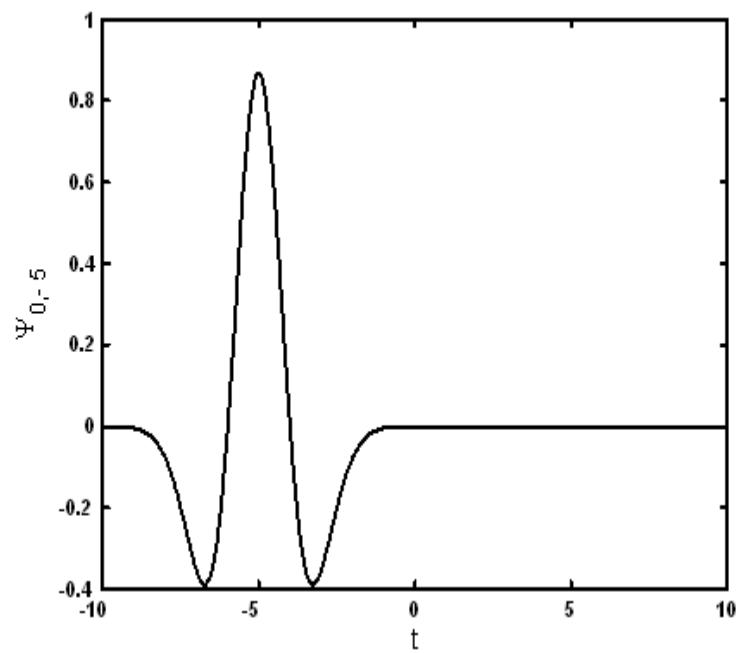


Figure 2: Ondelette "Chapeau-Mexicain" tradlatée d'un paramètre $b = -5$.

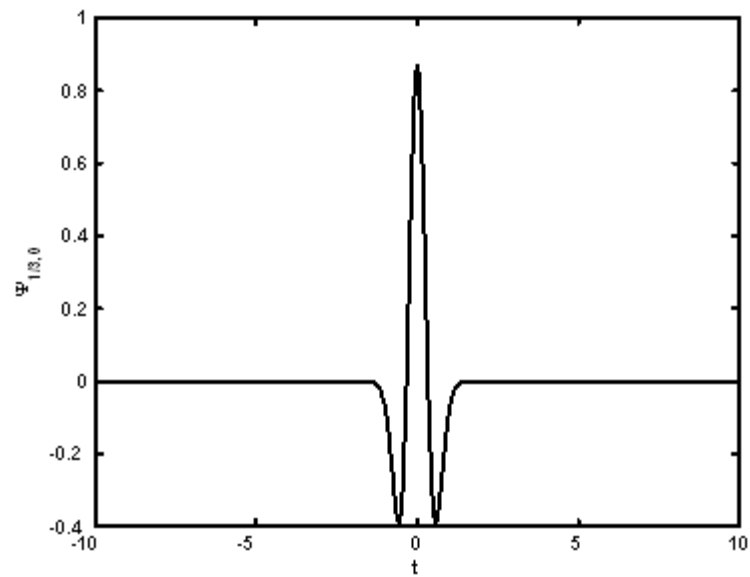


Figure 3: Ondelette "Chapeau-Mexicain" dilatée d'un paramètre échelle $a = 1/3$.

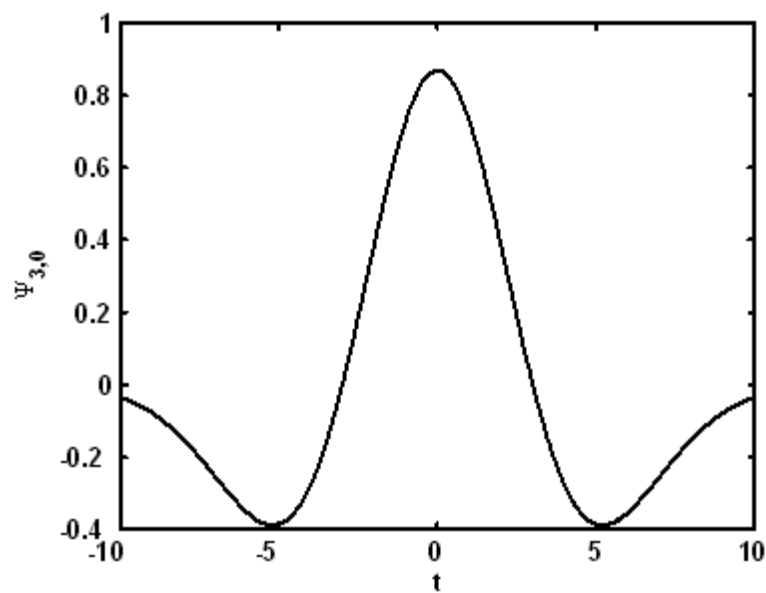


Figure 4: Ondelette "Chapeau-Mexicain" dilatée d'un paramètre échelle $a = 3$.

1.2.1 Transformée en ondelettes

Soit un signal ou une fonction $f : \mathbb{R} \rightarrow \mathbb{R}$ appartenant à $L^2(\mathbb{R})$. La transformée continue en ondelettes (CWT) définit une nouvelle description pour la fonction f . A l'inverse d'une

représentation fréquentielle pure, la CWT de cette fonction ausculte simultanément son "contenu" spatial et en échelle, la dépliant ainsi dans un espace plus grand que l'espace initial. L'exploration du contenu de f peut être réalisée par la comparaison de cette dernière avec une ondelette Ψ connue. Ce processus se réalise par un produit scalaire, engendrant ainsi les coefficients

$$\begin{aligned} W_f(a, b) &= \langle \Psi_{a,b} | f \rangle = \int_{-\infty}^{+\infty} dt f(t) \Psi_{a,b}^*(t) \\ &= \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} dt f(t) \Psi^*\left(\frac{t-b}{a}\right) \end{aligned} \quad (1.2.5)$$

où le symbole $*$ dénote la conjugaison complexe. L'ensemble des coefficients $W_f(a, b)$ caractérise univoquement f et constitue la transformée continue en ondelettes de cette dernière. Il existe également une lecture fréquentielle de cette transformée. En effet, en utilisant le théorème de convolution, la relation (1.2.5) peut être réécrite sous la forme

$$W_f(a, b) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} d\xi \hat{f}(\xi) \sqrt{a} \hat{\Psi}^*(a\xi) e^{ib\xi} \quad (1.2.6)$$

Daubechies[34] a montré que si l'ondelette est convenablement choisie, la transformée en ondelettes est inversible et il devient alors possible de reconstruire la fonction après l'analyse, soit

$$f(t) = C_{\Psi}^{-1} \int_0^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{a^2} \langle \Psi_{a,b} | f \rangle \Psi_{a,b}(t) da db \quad (1.2.7)$$

Le coefficient C_{Ψ} est donné par l'équation suivante

$$C_{\Psi} = 2\pi \int_{-\infty}^{+\infty} \left| \hat{\Psi}(\xi) \right|^2 \frac{d\xi}{\xi} \quad (1.2.8)$$

Sa condition d'existence correspond à la condition d'admissibilité de l'ondelette analysante donnée par l'équation (1.2.1). L'ensemble des coefficients définis par la relation (1.2.6) permet de passer d'une description temporelle à une description dite temps-échelle. Toutefois lors de ce passage, il faut garantir qu'aucune information n'a été perdue et pouvoir ainsi

reconstruire f au moyen de ces coefficients. Il est utile de préciser que ces affirmations peuvent être justifiées en vertu du théorème suivant[33]:

Théorème 1. *Soit $\Psi \in L^1(\mathbb{R})$ une ondelette admissible. Toute fonction $f \in L^2(\mathbb{R})$ peut être décomposée comme suit*

$$f(t) = C_{\Psi}^{-1} \int_{-\infty}^{+\infty} \int_0^{+\infty} \frac{dbda}{a^2} W_f(a, b) \Psi_{a,b}(t) \quad (1.2.9)$$

où l'égalité est vraie presque partout sur \mathbb{R} .

La projection de l'équation précédente sur f définit le corollaire suivant, qui exprime la conservation de l'énergie, définie comme $\|f\|^2$, lors du changement de représentation.

Corollaire 1. *Si $\Psi \in L^1(\mathbb{R})$ et si $C_{\Psi} < \infty$, alors*

$$\|f\|^2 = \int_{\mathbb{R}} dt |f(t)|^2 = C_{\Psi}^{-1} \int_{\mathbb{R}_+^* \times \mathbb{R}} \frac{dbda}{a^2} |W_f(a, b)|^2 \quad (1.2.10)$$

1.3 Transformée en ondelettes discrète

1.3.1 Discrétisation de la transformée continue

Il est légitime de se demander s'il est nécessaire de connaître la transformée $W_f(a, b)$ sur $\mathbb{R}^{+*} \times \mathbb{R}$ tout entier pour reconstruire f . Lorsque la réponse est négative, l'utilisation d'un sous-ensemble discret semble un objectif raisonnable. La décomposition discrète en ondelettes est obtenue en contraignant les paramètres a et b à prendre leurs valeurs dans un sous-ensemble discret de \mathbb{R} . La forme la plus courante de cette discrétisation utilise des ensembles de paramètres a et b définis par

$$a \in \{a_0^p\}_{p \in \mathbb{Z}} \text{ et } b \in \{na_0^p b_0\}_{p, n \in \mathbb{Z}} \quad (1.3.1)$$

Par conséquent, au lieu d'utiliser la famille d'ondelettes donnée par l'équation (1.2.4), on se sert pour la transformée discrète, de la famille dénombrable d'ondelettes suivante

$$\Psi_{p,n}(t) = a_0^{-p/2} \Psi(a_0^{-p}t - nb_0), \quad a_0 > 1 \text{ et } b_0 > 0 \text{ fixés avec } p, n \in \mathbb{Z} \quad (1.3.2)$$

Un choix de paramètres courant est $a_0 = 2$ et $b_0 = 1$ [34].

1.3.2 Analyse multirésolution et bases orthonormées d'ondelettes

Nous présentons dans cette partie l'analyse multirésolution de type ondelettes. Cette présentation ne se veut pas exhaustive. Le lecteur peut consulter la référence[13] pour un tour plus complet du sujet. Nous introduirons les bases orthonormées d'ondelettes en partant de la notion d'analyse multirésolution de l'espace des signaux d'énergie finie. Celle-ci fournit un cadre pour la décomposition d'un signal sous la forme d'une suite d'approximations de résolution décroissante, complétée par une suite de détails. Cette notion implique donc l'imbrication d'espaces d'approximations et elle permet de définir des algorithmes performants de transformée en ondelettes rapides. A titre d'exemple, lorsqu'un cameraman effectue un zoom, il part d'une vision globale des choses pour se focaliser sur une zone d'intérêt en faisant apparaître des détails. L'analyse multirésolution formalise ce type d'approximation. Les approximations multirésolution calculent l'approximation d'un signal $f(t)$ à diverses résolutions par projection orthogonale sur une famille d'espaces $\{V_j\}_{j \in \mathbb{Z}}$. Dans ce contexte, une analyse multirésolution est définie comme une suite de sous-espaces vectoriels fermés $\{V_j\}_{j \in \mathbb{Z}}$ de $L^2(\mathbb{R})$ vérifiant les propriétés suivantes:

$$\forall (j, k) \in \mathbb{Z}^2, f(t) \in V_j \Leftrightarrow f(t - 2^{-j}k) \in V_j. \quad (1.3.3)$$

$$V_j \subset V_{j-1}. \quad (1.3.4)$$

$$\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2 \text{ et } \bigcap_{j \in \mathbb{Z}} V_j = \{0\}. \quad (1.3.5)$$

$$\forall j \in \mathbb{Z}, f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}. \quad (1.3.6)$$

$$\exists \varphi \in V_0 \text{ tel que } \{\varphi(x - k)\}_{k \in \mathbb{Z}} \text{ forment une base orthonormée de } L^2(\mathbb{R}). \quad (1.3.7)$$

Attardons nous sur la signification de ces propriétés; la première propriété signifie que V_j est invariant pour des translations proportionnelles au paramètre d'échelle 2^{-j} . La seconde propriété traduit l'emboîtement des espaces et l'amélioration de l'approximation lorsque j décroît. La propriété (1.3.5) assure que la suite $\{V_j\}$ converge vers L^2 tout entier et donc que la suite des projections converge vers f . Enfin, la propriété (1.3.6) nous assure que la dilatation par 2 d'un signal $f(t)$ est une approximation de $f(t)$ à une résolution plus grossière 2^{-j+1} .

L'analyse multirésolution d'un signal f consiste à réaliser des projections orthogonales successives du signal sur les espaces V_j , ce qui conduit à des approximations de plus en plus grossières de f au fur et à mesure que j croit. La différence entre deux approximations consécutives représente l'information du "détail" qui est perdue au passage d'une échelle à l'autre; cette information est contenue dans le sous-espace W_j orthogonal à V_j tel que

$$V_{j-1} = V_j \oplus W_j \quad (1.3.8)$$

On dit qu'un élément de l'espace d'approximation de niveau $j - 1$ se décompose en l'approximation de niveau j , qui est plus grossière, et du détail correspondant à ce niveau j . Les $\{\varphi_{j,k}\}_{k \in \mathbb{Z}}$ engendrent V_j tandis que les $\{\Psi_{j,k}\}_{k \in \mathbb{Z}}$ engendrent W_j . Comme $L^2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_j$, tout signal est la somme de tous ses détails et les $\{\Psi_{j,k}\}_{(j,k) \in \mathbb{Z}^2}$ forment une base orthonormée d'ondelettes de $L^2(\mathbb{R})$.

CHAPITRE

2

Réseaux de neurones

2.1 Introduction

Il est rare que les grandes conquêtes technologiques s'inspirent du vivant. La nature n'est pourtant pas avare de réalisations qui nous étonnent ou tiennent en échec notre curiosité. Parmi elles, le cerveau occupe évidemment une place de choix. En effet, la pensée est produite par le cerveau ou plutôt en est une propriété. Ce dernier est tout simplement le siège de l'intelligence et de l'apprentissage chez l'humain. La physiologie du cerveau montre que celui-ci est constitué de cellules (les neurones) interconnectées. Il y'a environ 100 milliards de neurones dans un cerveau humain. Bien que ceux-ci travaillent en régime impulsionnel, on peut les assimiler à des sommateurs. Chaque neurone peut recevoir les signaux par des extensions très ramifiées de leur corps cellulaires (les dendrites) et envoient l'information par de longs prolongements (les axones). Les contacts entre deux neurones, de l'axone à une dendrite, se font par l'intermédiaire de synapses.

Un réseau de neurones artificiels est un modèle du traitement de l'information dont la conception est très schématiquement inspirée du fonctionnement de vrais neurones. En effet, l'hypothèse principale, à la base de l'essor des réseaux de neurones artificiels, est que le comportement intelligent est sous-tendu par un ensemble de mécanismes mentaux et émerge par conséquent de la structure et du comportement des éléments de base du cerveau. De façon grossièrement similaire, un réseau de neurones artificiels, basé sur un modèle simplifié de neurone, est donc constitué d'un très grand nombre de petites unités de traitement identique appelées "neurones artificiels". Ces derniers sont connectés entre eux par des liaisons pondérées, les connexions, appelées aussi "synapses" d'après le terme biologique correspondant. De tels réseaux permettent certaines fonctions du cerveau, comme l'apprentissage par l'exemple et le travail en parallèle. Cependant, le neurone artificiel reste tout de même loin de posséder toutes les capacités de son analogue biologique. Certains réseaux de neurones sont des modèles de réseaux biologiques mais d'autres ne le sont pas forcément.

Un réseau comporte plusieurs couches de neurones: une couche d'entrée, une de sortie et une ou plusieurs couches intermédiaires ou cachées. L'estimation des paramètres (poids

synaptiques) du réseau est effectuée lors d'une phase de développement du réseau appelée "phase d'apprentissage". En effet, lorsqu'on présente au réseau un exemple d'apprentissage, les neurones entrent simultanément dans un état d'activité qui provoque une modification des forces synaptiques. Il s'ensuit une reconfiguration de l'ensemble des synapses. On dit que le réseau s'auto-organise à partir d'exemples qui lui sont présentés.

L'histoire des réseaux de neurones artificiels est tissée à travers des découvertes conceptuelles et des développements technologiques survenus à diverses époques. En effet, la motivation initiale du développement des réseaux de neurones formels était "neuromimétique". Elle partait du constat que les organismes les plus simples effectuent, sans effort apparent, des tâches que les ordinateurs ne réalisent qu'imparfaitement, au prix de lourds calculs; la reconnaissance d'objets en est un exemple frappant. Les neurologues Warren Sturgis McCulloch et Walter Pitts constituèrent un modèle simplifié du neurone biologique communément appelé "neurone formel"[3]. Ils montrèrent que des réseaux de neurones formels relativement simples sont capables de réaliser des fonctions logiques et arithmétiques. Cependant, leurs travaux n'ont pas donné d'indication sur une quelconque méthode permettant d'estimer les coefficients synaptiques. Vers la fin des années quarante, Donald Hebb[35] proposa une théorie fondamentale pour l'apprentissage et en 1957, une première application concrète des réseaux de neurones est survenue avec l'invention du modèle du "perceptron" par un dénommé Franck Rosenblatt[4]. Cependant, vers la fin des années soixante, un livre publié par Marvin Minsky et Seymour Papert[5] a mis en exergue quelques limitations théoriques du perceptron et l'engouement pour les réseaux de neurones se tarit alors vers la fin des années soixante. L'intérêt pour les réseaux de neurones va être ravivé par le développement de nouveaux modèles (tels les réseaux de Hopfield[6]) et la redécouverte par Rumelhart[7] de l'algorithme de rétropropagation de l'erreur, originellement inventé par Werbos[8]. Dès lors, le domaine des réseaux de neurones artificiels a connu un essor considérable et a suscité un intérêt toujours croissant. Il est devenu un domaine où bouillonnent constamment de nouvelles théories, de nouvelles structures et de nouveaux algorithmes. Ils servent aujourd'hui à toutes sortes d'applications dans divers domaines. Ils sont utilisés dans

le traitement des signaux, pour bâtir des systèmes de vision par ordinateur, pour faire des prévisions sur le marché monétaire, pour le diagnostic médical, pour l'exploration pétrolière, en robotique,...etc. Le domaine des réseaux de neurones demeure toutefois un sujet d'un grand intérêt pour les chercheurs qui désirent améliorer les performances de ces réseaux et étendre leur champ d'applications.

2.2 Neurone biologique

La majorité des neurones sont constitués de trois parties (voir figure 5):

1. Le corps proprement dit de la cellule où se situe le mécanisme du déclenchement du neurone.
2. Son prolongement, l'axone (qui peut atteindre plusieurs dizaines de centimètres) le long duquel voyagent les impulsions nerveuses dirigées vers la périphérie et vers de minces ramifications.
3. Les dendrites, qui reçoivent des informations (impulsions nerveuses) des autres neurones et les apportent vers le corps cellulaire.

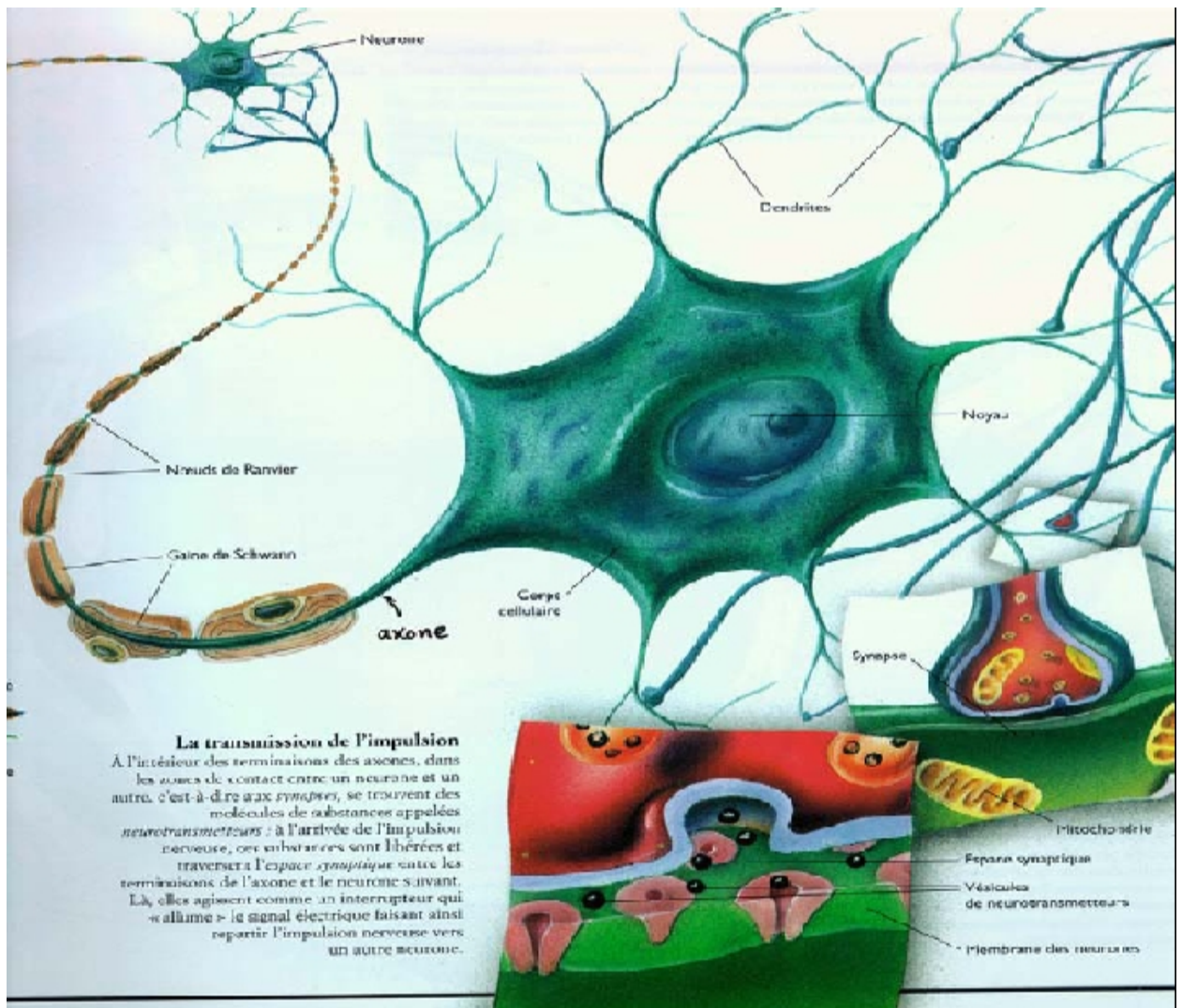


Figure 5: Neurone biologique, d'après T. Al Ani, Laboratoire A2SI- ESIEE- Paris.

Les axones sont recouverts de cellules dites cellules de Schwann, du nom de l'anatomiste allemand qui les découvrit au XIX^{ème} siècle. Lorsque ces cellules s'enroulent autour de l'axone, il se forme un revêtement appelé gaine myélinique qui présente des points découverts ou noeuds de Ranvier, nom du savant français qui les a observé en premier. L'impulsion nerveuse voyage par saut d'un noeud de Ranvier à un autre à une vitesse d'environ 200 m/s. Quand l'axone d'un neurone rencontre un autre neurone, le contact s'établit par l'intermédiaire d'une structure appelée synapse. Tous les neurones sont connectés par des

synapses dont le nombre est d'environ 10 000 milliards. Ceci signifie qu'un neurone reçoit en moyenne des excitations en provenance de 1000 neurones et qu'il émet également des informations vers 1000 neurones différents.

Il existe au moins 4 modes de communication entre neurones: liaison terminaison axonale-dendrite, liaison terminaison axonale-corps cellulaire, liaison axone-axone et liaison dendrite-dendrite. La synapse est une sorte de bulbe: entre la surface du bulbe (bouton terminal) et celle du nouveau neurone se trouve un espace dit espace synaptique. Pour que l'impulsion nerveuse, qui est un signal de nature électrique, puisse franchir cet espace, des substances électrochimiques entrent en action: les neurotransmetteurs, normalement contenus dans le bouton terminal, qui, une fois libérés, modifient les propriétés électriques des neurones suivants faisant ainsi repartir le signal électrique.

A l'apparition d'excitations qui se traduisent au niveau du noyau cellulaire par une tension supérieur à un seuil, le neurone est alors activé et cela déclenche un processus de dépolarisation: le neurone émet sur l'axone une tension positive d'environ +60 mV en se comportant comme un système non linéaire. Il est clair que le neurone biologique, qui est une entité indivisible, est l'élément fondamental sur lequel repose le système nerveux.

De façon grossièrement similaire, un réseau de neurones artificiels est un modèle du traitement de l'information dont la conception est très schématiquement inspirée du fonctionnement de vrais neurones.

2.3 Neurone formel

Un neurone formel est l'unité élémentaire d'un réseau de neurones[3]. Il reçoit un nombre variable d'entrées en provenance de neurones amonts. A chacune de ces entrées est associé un paramètre appelé coefficient ou poids représentatif de la force de la connexion. Chaque neurone est doté d'une sortie unique, qui se ramifie ensuite pour alimenter les neurones avals. D'un point de vue mathématique, un neurone formel est une fonction algébrique non

linéaire, paramétrée et à valeurs bornées. Le fonctionnement d'un neurone artificiel peut être modélisé par la représentation graphique suivante (figure 6).

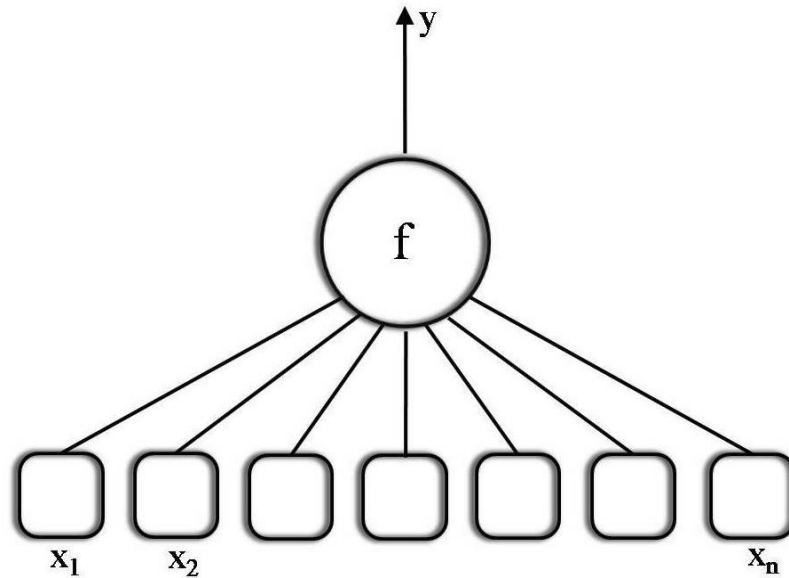


Figure 6: Représentation graphique d'un neurone formel, dont la sortie est donnée par

$$y = f\left(\sum_{i=1}^n \omega_i x_i\right).$$

Les variables $\{x_i\}$ sont désignées sous le terme d'entrées du neurone, les $\{\omega_i\}$ sont des paramètres, appelés "poids de connexions" et la valeur de la fonction désigne la sortie du neurone. Cette représentation est le reflet de l'inspiration biologique. En effet, toutes les connexions n'ont pas la même importance pour le neurone. Le poids affecté à chaque connexion mesure cette importance relative. Il faut bien comprendre que le neurone artificiel est un modèle pragmatique qui, comme nous le verrons plus loin, nous permettra d'accomplir des tâches intéressantes. Les poids des connexions représentent une grossière approximation de véritables synapses qui résultent en fait d'un processus chimique très complexe et dépendant de nombreux facteurs extérieurs encore mal connus. La fonction f est appelée fonction d'activation (fonction de seuillage ou encore fonction de transfert).

2.4 Fonction d'activation

La fonction d'activation f sert à introduire une non linéarité dans le fonctionnement d'un neurone. Son choix se révèle être un élément constitutif important des réseaux de neurones. En effet, les propriétés de la fonction d'activation influent sur celle du neurone formel et il est donc important de bien choisir celle-ci pour obtenir un modèle utile en pratique.

Dans la formulation d'origine de Mc Culloch et Pitts, la fonction d'activation est la fonction saut de Heaviside. La plupart des neurones formels utilisés actuellement sont des variantes du neurone de Mc Culloch et Pitts dans lesquels la fonction de Heaviside est remplacée par une autre fonction d'activation. D'une manière générale, la fonction d'activation peut-être paramétrée de différentes manières. Deux types de paramétrages sont fréquemment utilisés.

1. La sortie du neurone est une fonction non linéaire d'une combinaison linéaire des entrées $\{x_i\}$ pondérées par les paramètres $\{\omega_i\}$. Dans ce cas, le choix recommandé pour f est une fonction en forme de "s". A titre illustratif, voici quelques fonctions couramment utilisées.

- La fonction sigmoïde : $y = f(x) = \frac{1}{1+\exp(-x)}$.
- La fonction tangente hyperbolique : $y = f(x) = \tanh(x)$.
- La fonction arctangente : $y = f(x) = \arctan(x)$.

Pour un choix de la fonction tangente hyperbolique, par exemple, la sortie d'un neurone admet alors pour équation[22]

$$y = \tanh(v) = \tanh(-\omega_0 + \sum_{i=1}^{n-1} \omega_i x_i) \quad (2.4.1)$$

où v est appelé "potentiel" et le coefficient ω_0 est le seuil d'activation du neurone, appelé "biais". En général, on intègre ce seuil dans les signaux d'entrée en considérant une entrée fictive x_0 de valeur constante égale à 1 à laquelle on associe le coefficient synaptique ω_0 .

2. Les paramètres interviennent directement dans la fonction d'activation, ils sont attachés à la non-linéarité du neurone. Dans ce cas les fonctions d'activation utilisées sont

- Les fonctions radiales (RBF), introduites par Powell[36] dans le cadre de l'interpolation, c'est à dire de la recherche de fonctions passant exactement par un nombre fini de points (dits points de collocation). Une fonction donnée est dite radiale si elle ne dépend que de la distance du point courant x au point de collocation x_n . On peut utiliser diverses fonctions radiales, notamment des fonctions localisées (qui tendent vers zéro dans toutes les directions de l'espace des variables) telles que des gaussiennes centrées aux points de collocation. Broom[11] semble être parmi les premiers à proposer l'idée d'utiliser des réseaux RBF pour l'approximation de fonctions non linéaires.

- Les fonctions ondelettes qui tirent leur origine de la théorie du signal[37].

A titre d'exemple, la sortie d'un neurone à non-linéarité gaussienne a pour équation

$$y = \exp \left[-\frac{\sum_{i=1}^n (x_i - \omega_i)^2}{2\omega_{n+1}^2} \right] \quad (2.4.2)$$

où les paramètres ω_i , $i = 1, \dots, n$, sont les coordonnées du centre de la gaussienne et ω_{n+1} est son écart-type.

Un neurone formel réalise donc simplement une fonction non linéaire, paramétrée de ses entrées. L'intérêt des neurones réside dans les propriétés qui résultent de leur association en réseaux.

2.5 Réseaux de neurones formels

Un réseau de neurones est constitué d'un ensemble d'unités (neurones formels) interconnectées. C'est un graphe orienté dont chaque sommet représente un neurone. Ce dernier possède des entrées, des sorties et des paramètres. Les connexions du graphe transportent les sorties de certains neurones vers les entrées d'autres neurones. Chaque neurone est capable de calculer ses sorties en fonction de ses entrées et de ses paramètres. L'exemple le plus simple de réseaux de neurones est donné par le perceptron multicouches. Ce dernier est constitué de plusieurs couches de neurones connectées entre elles de l'entrée vers la sortie. A

titre illustratif, nous présentons, sur la figure 7 suivante, le schéma d'un perceptron à trois couches.

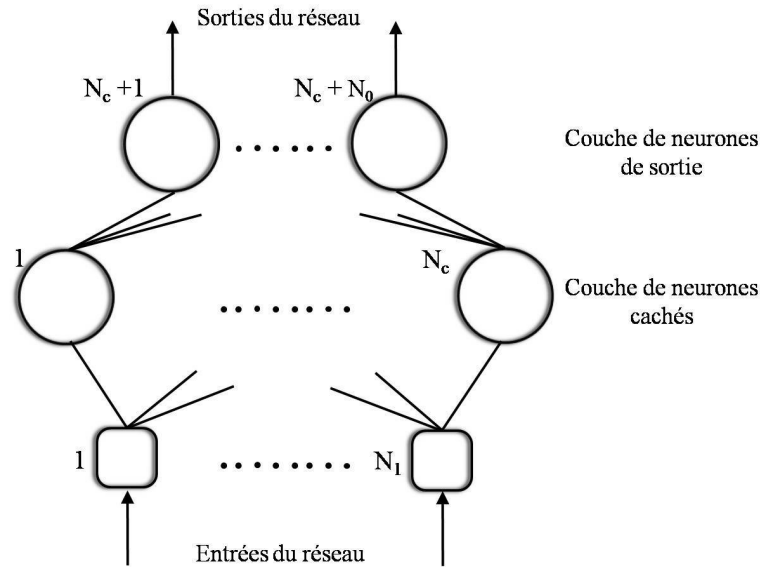


Figure 7: Perceptron à trois couches.

- La couche d'entrée est la première couche du réseau. Elle contient les variables d'entrées qui peuvent être les données sources que l'on veut utiliser pour l'analyse.
- La couche cachée est celle qui n'a qu'une utilité intrinsèque pour le réseau. Elle n'a donc pas de contact direct avec l'extérieur. Le choix de sa taille n'est pas implicite et doit être ajusté.
- La couche de sortie donne directement le résultat obtenu par le réseau après compilation. Sa taille est déterminée par le nombre de variables que l'on veut en sortie.

Théoriquement, rien n'interdit de construire un réseau possédant plusieurs sorties. Cependant, dans la pratique, il est généralement souhaitable de décomposer un problème à N sorties en N problèmes à 1 sortie. En effet, sauf cas particulier (notamment lorsqu'il s'agit de l'utilisation des réseaux de neurones comme classifieurs, dont les sorties sont des estimations des probabilités d'appartenance d'un objet à une classe ; la somme des sorties, dans ce cas, doit alors être égale à 1), il n'y a pas de lien entre les différentes grandeurs que l'on cherche à modéliser. En règle générale, les connexions entre les neurones qui composent le

réseau décrivent la topologie du modèle. Cette topologie fait le plus souvent apparaître une certaine régularité et l'on peut par conséquent distinguer différentes sortes de réseaux:

- Les réseaux multicouches classiques: il n'y a pas de connexion entre les neurones d'une même couche et les connexions ne se font qu'avec les neurones de la couche aval. Tous les neurones de la couche amont sont connectés à tous les neurones de la couche aval.

- Les réseaux à connexion locale: ce sont également des réseaux multicouches mais tous les neurones d'une couche amont ne sont pas connectés à tous les neurones de la couche aval. Le nombre de connexions est donc moins important que dans le cas d'un réseau multicouches classique.

- Les réseaux à connexions récurrentes: une ou plusieurs sorties de neurones d'une couche aval sont connectées aux entrées des neurones de la couche amont ou bien de la même couche. Ces connexions récurrentes ramènent de l'information en arrière par rapport au sens de propagation défini dans un réseau multicouches classique.

- Les réseaux à connexions complexes: chaque neurone est connecté à tous les neurones du réseau, y compris lui-même. Dans ce cas, aucun sens général de propagation n'est défini. C'est la structure d'interconnexion la plus générale possible.

D'une manière générale, on distingue deux types de réseaux de neurones: les réseaux non bouclés et les réseaux bouclés[22].

2.5.1 Réseaux de neurones non bouclés

Un réseau de neurones non bouclé, tel que représenté sur la figure 8, est un réseau représenté graphiquement par un ensemble de neurones "connectés" entre eux avec l'information circulant des entrées vers les sorties sans "retour en arrière": si l'on représente le réseau comme un graphe dont les noeuds sont les neurones et les arêtes les "connexions" entre ceux-ci, un tel graphe de connexions est dit acyclique, c'est à dire, si l'on se déplace dans le réseau, à partir d'un neurone quelconque, en suivant les connexions, on ne peut pas revenir au neurone de départ. Il convient d'insister sur le fait que le temps ne joue aucun rôle fonctionnel dans un réseau de neurones non bouclé: les entrées sont constantes et les sorties le sont

également. Le temps nécessaire pour le calcul de la fonction réalisée par chaque neurone est négligeable et fonctionnellement, on peut considérer ce calcul comme instantané. Pour cette raison, les réseaux de neurones non bouclés sont souvent appelés «réseaux statiques» par opposition aux réseaux bouclés ou «dynamiques».

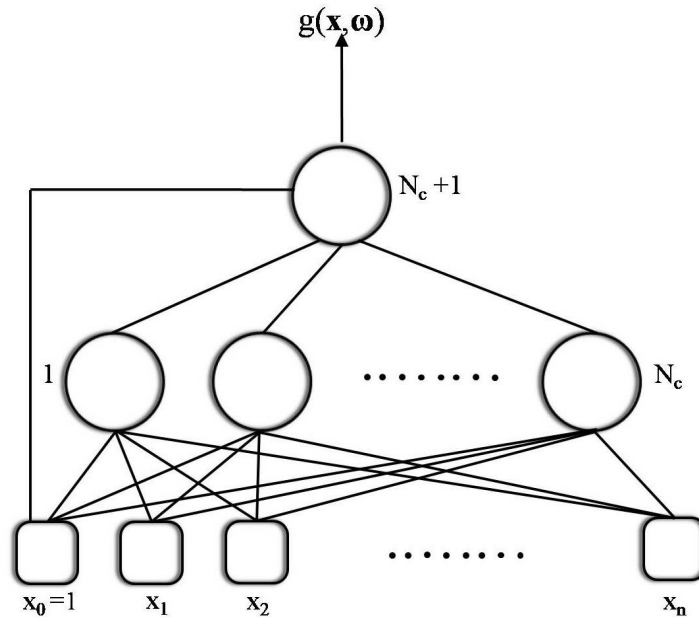


Figure 8: Réseau de neurones non bouclé à $(n + 1)$ entrées, une couche de N_c neurones cachés et un neurone de sortie linéaire.

2.5.2 Réseaux de neurones bouclés

Un réseau de neurones bouclé, appelé aussi réseau "dynamique", est un réseau dont le graphe des connexions peut contenir des cycles, c'est à dire, lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ: un tel chemin est désigné sous le terme de "cycle". La sortie d'un neurone du réseau peut donc être fonction d'elle-même; cela n'est évidemment concevable que si la notion de temps est explicitement prise en considération. Ainsi, à chaque connexion d'un réseau de neurones bouclé est attaché, outre un poids, un retard, multiple entier de l'unité de temps choisie. Un réseau de neurones bouclé à temps discret est donc un système dynamique, régi par une ou plusieurs équations aux différences non linéaires, résultant de

la composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions.

Il a été montré dans les références[38, 39] que tout réseau bouclé peut être mis sous une forme particulière, appelée forme canonique, qui est une représentation d'état minimale. Elle est constituée d'un réseau non bouclé et de connexions de retard unité ramenant les sorties de ce réseau non bouclé vers les entrées de celui-ci. La forme canonique permet de mettre clairement en évidence un ensemble minimum de variables d'état. De plus, son utilisation facilite la mise en oeuvre de l'apprentissage du réseau. La forme canonique des équations régissant un réseau de neurones bouclé est donnée par les relations suivantes

$$x(k+1) = \varphi [x(k), u(k)] \quad (2.5.1)$$

$$y(k) = \Psi [x(k), u(k)] \quad (2.5.2)$$

où φ et Ψ sont des fonctions non linéaires réalisées par un réseau de neurones non bouclé et où k désigne le temps discret. Le réseau de la figure 9 peut-être mis sous la forme canonique, strictement équivalente, mais beaucoup plus facilement manipulable, représentée sur la figure 10.

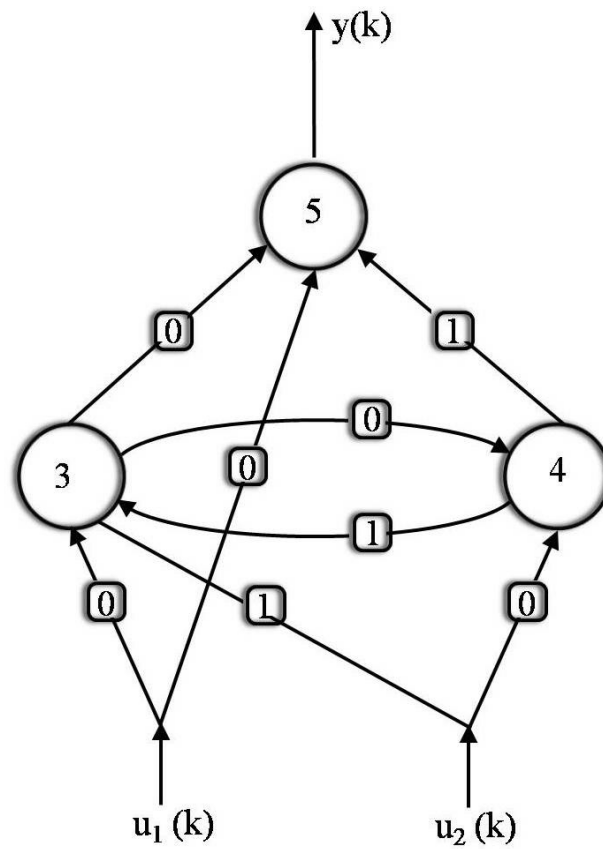


Figure 9: Réseau de neurones bouclé à deux entrées. Les chiffres dans les carrés indiquent le retard (exprimé en nombre de périodes d'échantillonnage) attaché à chaque connexion.

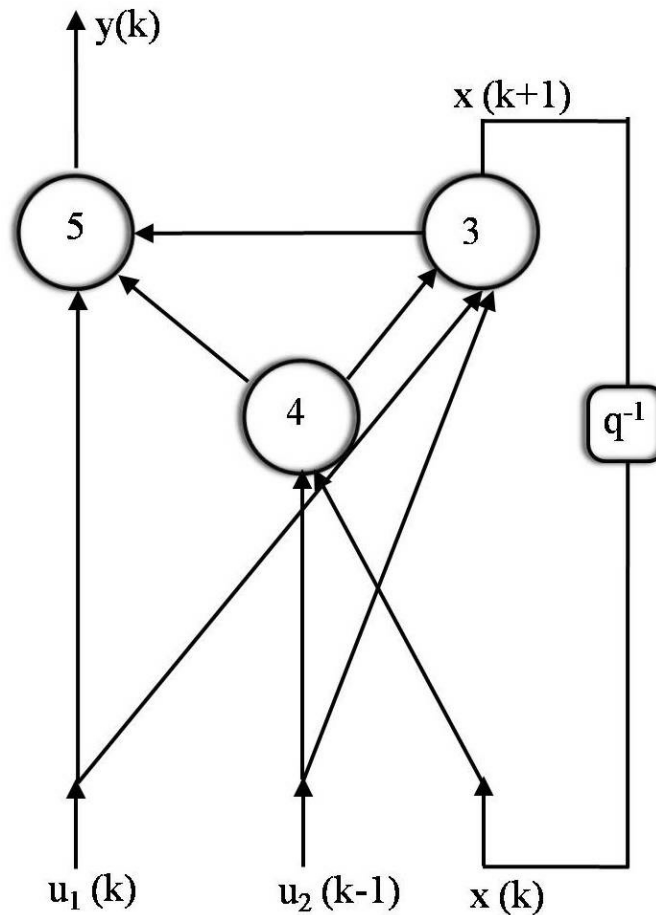


Figure 10: Forme canonique du réseau de neurones représenté sur la figure 9. q^{-1} est un symbole habituel en automatique qui représente un retard d'une unité de temps.

Ainsi, un réseau de neurones bouclé peut être mis sous une forme canonique, qui comprend un réseau de neurones non bouclé et des bouclages (récurrences) externes à celui-ci. Il est donc utile de noter que les propriétés des réseaux bouclés sont directement liées à celles des réseaux non bouclés. Ces derniers permettent de modéliser, de manière statique, des processus non linéaires qui peuvent être utilement décrits par des équations algébriques. De même, il est intéressant de mettre en oeuvre des réseaux de neurones bouclés pour modéliser, cette fois-ci, de manière dynamique, des processus qui peuvent être utilement décrits par des équations récurrentes ou équations aux différences.

2.6 Processus d'apprentissage

Après avoir introduit les notions de base concernant les réseaux de neurones et la question du choix de leur structure, nous allons présenter dans cette partie les méthodes algorithmiques permettant d'estimer les paramètres de ces réseaux, cette phase est dite phase d'apprentissage. Parmi les propriétés désirables pour un réseau de neurones, la plus fondamentale est sûrement la capacité d'apprendre de son environnement et d'améliorer ainsi sa performance à travers ce processus d'apprentissage. Ce dernier est tout simplement une phase de développement du réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. Au cours de cette phase, les paramètres des neurones sont alors estimés afin que le réseau remplisse au mieux la tâche qui lui est affectée. On distingue deux types d'apprentissage: l'apprentissage «supervisé» et l'apprentissage «non supervisé». Cette distinction repose sur la forme des exemples d'apprentissage. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie associée) alors que l'on ne dispose que des valeurs (Entrée) pour l'apprentissage non supervisé.

- Apprentissage supervisé: Lorsqu'on connaît en tout point ou en un ensemble de points les valeurs que doit avoir la sortie du réseau en fonction des entrées correspondantes, on parle d'apprentissage supervisé. Pour les réseaux à apprentissage supervisé, on présente donc au réseau des entrées et au même temps les sorties désirées. Le réseau doit alors se reconfigurer, c'est à dire calculer ses poids afin que la sortie qu'il donne corresponde bien à la sortie souhaitée. Ce type d'apprentissage est utilisé pour des tâches d'approximation, de modélisation statique et dynamique, de classification,...etc.

- Apprentissage non supervisé: Ce type d'apprentissage est utilisé dans un but de visualisation ou d'analyse de données. En effet, lorsqu'on dispose d'un ensemble de données et l'on cherche à les regrouper, selon des critères de ressemblance qui sont inconnus à priori, on parle d'apprentissage non supervisé. Dans ce cas, on présente une entrée au réseau et on le laisse évoluer librement jusqu'à ce qu'il se stabilise.

D'une manière générale, cette étape, appelée phase de développement du réseau de neurones, requiert l'utilisation de techniques d'optimisation qui consistent à déterminer les paramètres optimaux par minimisation directe d'une fonction, appelée fonction de coût, vis-à-vis des paramètres du modèle. La procédure adaptative de mise à jour des poids obéit essentiellement à un objectif; la recherche du minimum de la fonction de coût (ou fonction d'erreur) dans un espace multidimensionnel.

2.6.1 Algorithmes d'optimisation

Position du problème

On dispose d'un ensemble de N mesures $\{y^k = y(\mathbf{x}^k), k = 1, \dots, N\}$ de la sortie correspondant à N valeurs des entrées $\{\mathbf{x}^k\} = \left\{ [x_1^k, x_2^k, \dots, x_q^k]^T \right\}$, ($k = 1, \dots, N$) (Dans tout ce qui suit, les vecteurs ainsi que les matrices sont représentés en caractères gras). Conformément à l'usage dans le domaine des réseaux de neurones, nous désignons cet ensemble par l'expression "ensemble d'apprentissage". Afin de déterminer les valeurs des paramètres, composantes du vecteur $\boldsymbol{\omega}$, on minimise une fonction de coût représentative des différences entre les valeurs mesurées et les valeurs, calculées par le modèle, qui sont supposées en constituer une bonne prédiction. Pour chaque élément k de l'ensemble d'apprentissage, on définit le résidu

$$r_k = y^k - g(\mathbf{x}^k, \boldsymbol{\omega}) = y(\mathbf{x}^k) - g(\mathbf{x}^k, \boldsymbol{\omega}) \quad (2.6.1)$$

où y^k est la $k^{\text{ème}}$ mesure de la sortie et \mathbf{x}^k est la $k^{\text{ème}}$ mesure du vecteur des entrées.

La définition d'une fonction de coût est primordiale lors de la mise en oeuvre d'un réseau de neurones. Cette fonction doit permettre de mesurer l'écart entre la sortie du modèle et les mesures faites sur les exemples d'apprentissage. Si cet écart est important, la fonction de coût doit être grande et inversement. Il existe un grand nombre de fonctions possibles[40]. La fonction la plus couramment utilisée et dont nous nous sommes servi lors de notre travail,

est la fonction dite des "moindres carrés". Cette dernière est définie comme la somme, sur tous les éléments de l'ensemble d'apprentissage, des carrés des résidus, soit[40]

$$J(\boldsymbol{\omega}) = \frac{1}{2} \sum_{k=1}^N (y^k - g(\mathbf{x}^k, \boldsymbol{\omega}))^2 = \frac{1}{2} \sum_{k=1}^N J^k(\boldsymbol{\omega}) \quad (2.6.2)$$

Modèles linéaires par rapport aux paramètres

Dans le cas où le modèle est linéaire par rapport aux paramètres à ajuster, la minimisation de la fonction de coût et donc l'estimation du vecteur des paramètres $\boldsymbol{\omega}$ peut se faire à l'aide de la méthode des moindres carrés. Cette dernière ramène le problème à la résolution d'un système d'équations linéaires. La sortie du réseau étant linéaire par rapport aux paramètres, représentés par le vecteur $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_q]^T$, le modèle peut être décrit par une équation du type

$$g(\mathbf{x}^k, \boldsymbol{\omega}) = \boldsymbol{\omega}^T \boldsymbol{\varphi}(\mathbf{x}^k) \quad \text{pour } k = 1, \dots, N \quad (2.6.3)$$

où $\boldsymbol{\varphi}(\mathbf{x}^k)$ est une fonction vectorielle, de dimension q , du vecteur des variables du modèle. Ces fonctions ne contiennent pas de paramètres ajustables pendant l'apprentissage.

L'estimation des paramètres est fondée sur la minimisation de la fonction de coût des moindres carrés, définie précédemment (2.6.2). En utilisant une notation matricielle, l'expression de la fonction de coût $J(\boldsymbol{\omega})$ devient alors

$$J(\boldsymbol{\omega}) = \frac{1}{2} (\mathbf{Y} - \mathbf{X}\boldsymbol{\omega})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\omega}) \quad (2.6.4)$$

avec

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{\varphi}(\mathbf{x}^1)^T \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{\varphi}(\mathbf{x}^N)^T \end{bmatrix} \quad (2.6.5)$$

$$\mathbf{Y} = [y^1, y^2, \dots, y^N]^T \quad (2.6.6)$$

$$\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_q]^T \quad (2.6.7)$$

La matrice \mathbf{X} est appelée matrice des observations. La fonction de coût étant quadratique (par rapport au vecteur des paramètres à estimer), elle atteint son minimum pour la valeur du vecteur des paramètres annulant sa dérivée. La solution $\boldsymbol{\omega}_{mc}$ de ce problème sera donnée par:

$$[\mathbf{X}^T \mathbf{X}] \boldsymbol{\omega}_{mc} = \mathbf{X}^T \mathbf{Y} \Leftrightarrow \boldsymbol{\omega}_{mc} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{Y} \quad (2.6.8)$$

Cette solution existe à condition que la matrice $\mathbf{X}^T \mathbf{X}$ soit inversible. Cette condition est généralement vérifiée.

Modèles non-linéaires par rapport aux paramètres

Lorsque le modèle est non linéaire par rapport aux paramètres, la méthode rapide de résolution précédente n'est plus utilisables. On a alors recours à des méthodes itératives pour effectuer l'estimation des paramètres. Ces méthodes d'optimisation sont coûteuses en temps calcul mais restent simples à mettre en oeuvre et s'appliquent à toutes les fonctions de coût dérivables par rapport aux paramètres. L'apprentissage des réseaux de neurones s'intègre dans ce cadre.

Le principe de ces méthodes est de se placer en un point, de choisir une direction de descente du coût et ensuite, de se déplacer d'un pas suivant cette direction. On atteint un nouveau point et l'on recommence la procédure. On poursuit cette démarche jusqu'à satisfaction d'un critère d'arrêt. Ainsi, à l'itération k , on calcule

$$\boldsymbol{\omega}^k = \boldsymbol{\omega}^{k-1} + \mu_{k-1} \mathbf{d}_{k-1} \quad (2.6.9)$$

μ_k est le pas de la descente et \mathbf{d}_k est la direction de descente. Les méthodes d'optimisation non linéaires que nous présentons se différencient par le choix de la direction de descente et du pas. Elles font appel aux:

- Gradient de la fonction de coût $J(\boldsymbol{\omega})$, vecteur des dérivées premières de J par rapport à $\boldsymbol{\omega}$, noté $\nabla J = \mathbf{Grad}(J(\boldsymbol{\omega}))$.

- Hessien, matrice des dérivées secondes de J par rapport à $\boldsymbol{\omega}$, notée $\mathbf{H} = \mathbf{H}(J(\boldsymbol{\omega}))$.

Dans le cas des réseaux de neurones, on utilise l'algorithme de rétropropagation du gradient de l'erreur pour le calcul du gradient[7]. On utilise en général une approximation du Hessien pour les méthodes de quasi Newton. Néanmoins, le Hessien peut évidemment être calculé exactement.

Il faut noter que ces méthodes ne sont pas spécifiques aux réseaux de neurones: ce sont des méthodes très générales d'optimisation qui existaient bien avant l'introduction des réseaux de neurones.

Evaluation du gradient de la fonction de coût par rétropropagation Considérons un réseau de neurones non bouclé avec des neurones cachés et un neurone de sortie. Chaque neurone i calcule une grandeur y_i qui est une fonction non linéaire de son potentiel ν_i , soit

$$y_i = f\left(\sum_{j=1}^{n_i} \omega_{ij} x_j\right) = f(\nu_i) \quad (2.6.10)$$

Les n_i entrées du neurone i peuvent être soit les sorties d'autres neurones, soit les entrées du réseau. Dans toute la suite, x_j désignera donc indifféremment soit la sortie y_j du neurone j , soit l'entrée j du réseau. Afin d'évaluer le gradient de la fonction de coût J , il suffit d'évaluer le gradient du coût partiel $J^k(\boldsymbol{\omega})$ relatif à l'observation k et de faire ensuite la somme sur tous les exemples. L'algorithme de rétropropagation consiste essentiellement en l'application répétée de la règle des dérivées composées, soit

$$\left(\frac{\partial J^k}{\partial \omega_{ij}}\right)_k = \left(\frac{\partial J^k}{\partial \nu_i}\right)_k \left(\frac{\partial \nu_i}{\partial \omega_{ij}}\right)_k = \delta_i^k x_j^k \quad (2.6.11)$$

Calculons à présent les quantités δ_i^k . Ces dernières peuvent être avantageusement calculées d'une manière récursive en menant les calculs depuis la sortie du réseau vers ses entrées. En effet, pour le neurone de sortie i

$$\begin{aligned}
 \delta_i^k &= \left(\frac{\partial J^k}{\partial \nu_i} \right)_k = \frac{\partial}{\partial \nu_i} \left[(y^k - g(\mathbf{x}^k, \boldsymbol{\omega}))^2 \right] \\
 &= -2 (y^k - g(\mathbf{x}^k, \boldsymbol{\omega})) \left(\frac{\partial g(\mathbf{x}^k, \boldsymbol{\omega})}{\partial \nu_i} \right)_k
 \end{aligned} \tag{2.6.12}$$

Soit

$$\delta_i^k = -2 (y^k - g(\mathbf{x}^k, \boldsymbol{\omega})) f'(\nu_i^k) \tag{2.6.13}$$

Pour un neurone caché i , la fonction de coût ne dépend du potentiel du neurone i que par l'intermédiaire des potentiels des neurones m qui reçoivent la valeur de la sortie du neurone i , c'est à dire de tous les neurones qui, dans le graphe des connexions du réseau, sont adjacents au neurone i , entre ce neurone et la sortie. Dans ce cas

$$\delta_i^k = \left(\frac{\partial J^k}{\partial \nu_i} \right)_k = \sum_m \left(\frac{\partial J^k}{\partial \nu_m} \right)_k \left(\frac{\partial \nu_m}{\partial \nu_i} \right)_k = \sum_m \delta_m^k \left(\frac{\partial \nu_m}{\partial \nu_i} \right)_k \tag{2.6.14}$$

Or

$$\nu_m^k = \sum_i \omega_{mi} x_i^k = \sum_i \omega_{mi} f(\nu_i^k) \tag{2.6.15}$$

d'où

$$\left(\frac{\partial \nu_m}{\partial \nu_i} \right)_k = \omega_{mi} f'(\nu_i^k) \tag{2.6.16}$$

Finalement, nous obtenons la relation suivante

$$\delta_i^k = \sum_m \delta_m^k \omega_{mi} f'(\nu_i^k) = f'(\nu_i^k) \sum_m \delta_m^k \omega_{mi} \tag{2.6.17}$$

Ainsi les quantités δ_i^k peuvent être calculées récursivement en parcourant le graphe des connexions "à l'envers" depuis la sortie vers les entrées du réseau.

Méthode du gradient simple La méthode la plus simple à mettre en oeuvre est la méthode du gradient simple[22]. L'algorithme correspondant consiste à choisir comme direction

de descente l'opposé du gradient de la fonction de coût ($\mathbf{d}_k = -\nabla J(\boldsymbol{\omega}^k) = -\mathbf{Grad}(J(\boldsymbol{\omega}^k))$) par rapport aux paramètres $\boldsymbol{\omega} = [\omega_1, \omega_2, \dots, \omega_m]^T$. Celui-ci s'écrit

$$\nabla J = \begin{bmatrix} \frac{\partial J}{\partial \omega_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial J}{\partial \omega_m} \end{bmatrix} \quad (2.6.18)$$

Le pas μ_k est une constante. Ainsi, à l'itération k , la modification des paramètres s'écrit

$$\boldsymbol{\omega}^k = \boldsymbol{\omega}^{k-1} - \mu_{k-1} \nabla J(\boldsymbol{\omega}^{k-1}) \quad (2.6.19)$$

Un "réglage" du pas de gradient μ_k est nécessaire: en effet, une petite valeur de ce paramètre ralentit la progression de l'algorithme; en revanche une grande valeur aboutit généralement à un phénomène d'oscillation autour de la solution. Cette méthode est très utilisée. Elle a pour avantages une grande facilité de mise en oeuvre et une grande robustesse. La méthode est efficace loin d'un minimum mais la vitesse de convergence diminue lorsque l'on s'approche du minimum (la modification des paramètres est proportionnelle au gradient de la fonction de coût qui tend vers 0).

Méthodes du second ordre Les méthodes du second ordre utilisent les dérivées secondes de la fonction de coût au lieu de modifier les paramètres uniquement en fonction du gradient de la fonction de coût. Ce sont des méthodes dérivées de la méthode de Newton, dont nous présentons le principe ci-après .

La méthode de Newton utilise la courbure (dérivée seconde) de la fonction de coût pour atteindre plus rapidement le minimum. La modification des paramètres est donnée par

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k - \mathbf{H}_k^{-1} \nabla J(\boldsymbol{\omega}^k) \quad (2.6.20)$$

où $\mathbf{H}(\boldsymbol{\omega}^k)$ est la matrice Hessienne de la fonction de coût, calculée avec le vecteur des paramètres disponible à l'étape courante, elle est donnée par

$$\mathbf{H}(\boldsymbol{\omega}^k) = \begin{bmatrix} \frac{\partial^2 J}{(\partial \omega_1^k)^2} & \frac{\partial^2 J}{\partial \omega_1^k \partial \omega_2^k} & \cdots & \frac{\partial^2 J}{\partial \omega_1^k \partial \omega_m^k} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 J}{\partial \omega_m^k \partial \omega_1^k} & \frac{\partial^2 J}{\partial \omega_m^k \partial \omega_2^k} & \cdots & \frac{\partial^2 J}{(\partial \omega_m^k)^2} \end{bmatrix} \quad (2.6.21)$$

La direction de descente est dans ce cas

$$\mathbf{d}_k = -\mathbf{H}_k^{-1} \nabla J(\boldsymbol{\omega}^k) \quad (2.6.22)$$

Le pas μ_k est constant et vaut 1. Cette méthode de recherche itérative du minimum présente de très bonnes propriétés de convergence au voisinage du minimum mais nécessite l'inversion d'une matrice à chaque itération, ce qui peut s'avérer lourd en temps calcul. L'algorithme de BFGS (du nom de ses inventeurs Broyden, Fletcher, Glodfarb et Shanno), ainsi que l'algorithme de Levenberg-Marquardt, que nous présenterons dans le paragraphe suivant, sont des méthodes "quasi-newtoniennes" qui permettent de pallier à ces inconvénients.

L'algorithme de BFGS Cet algorithme est fondé sur une approximation de la méthode de Newton exposée précédemment. L'inverse de la matrice Hessienne est approché par une matrice \mathbf{M}_k définie positive et modifiée à chaque itération. La suite des matrices $\{\mathbf{M}_k\}$ est construite de manière à converger vers l'inverse du Hessien. La modification des paramètres est définie comme suit

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k - \mu_{k+1} \mathbf{M}_{k+1} \nabla J(\boldsymbol{\omega}^k) \quad (2.6.23)$$

où μ_{k+1} est le pas et \mathbf{M}_{k+1} est calculée itérativement selon la formule suivante[15]

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{I} \left(1 + \frac{\boldsymbol{\gamma}_k^T \mathbf{M}_k \boldsymbol{\gamma}_k}{\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k} \right) \frac{\boldsymbol{\delta}_k^T \boldsymbol{\delta}_k}{\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k} - \frac{\boldsymbol{\delta}_k \boldsymbol{\gamma}_k^T \mathbf{M}_k + \mathbf{M}_k \boldsymbol{\gamma}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k} \quad (2.6.24)$$

où

$$\boldsymbol{\gamma}_k = \nabla J(\boldsymbol{\omega}^k) - \nabla J(\boldsymbol{\omega}^{k-1}) \quad (2.6.25)$$

et

$$\boldsymbol{\delta}_k = \boldsymbol{\omega}^k - \boldsymbol{\omega}^{k-1} \quad (2.6.26)$$

A la première itération \mathbf{M}_0 est prise égale à la matrice identité. La vitesse de convergence de cet algorithme est beaucoup plus grande que celle de la méthode du gradient simple. De plus, elle est relativement insensible au choix du pas. Cette méthode dite quasi-newtonienne n'est efficace qu'à proximité du minimum de la fonction de coût: il convient donc de combiner l'utilisation de cet algorithme avec, dans un premier temps, une approche du minimum par une méthode du premier ordre.

L'algorithme de Levenberg-Marquardt La méthode de Levenberg-Marquardt[41, 42] consiste à modifier les paramètres selon la relation suivante

$$\boldsymbol{\omega}^{k+1} = \boldsymbol{\omega}^k - [\mathbf{H}(\boldsymbol{\omega}^k) + \mu_{k+1} \mathbf{I}]^{-1} \nabla J(\boldsymbol{\omega}^k) \quad (2.6.27)$$

où $\mathbf{H}(\boldsymbol{\omega}^k)$ est la matrice Hessienne de la fonction de coût, définie précédemment et \mathbf{I} est la matrice identité. Cette méthode effectue un compromis entre la direction du gradient et la direction donnée par la méthode de Newton. En effet, lorsque μ_{k+1} est grand, on reconnaît la méthode du gradient simple avec un pas égale à $1/\mu_{k+1}$ et lorsque μ_{k+1} est petit, l'algorithme s'approche de celui de la méthode de Newton.

2.7 Applications des réseaux de neurones

Nous commençons tout d'abord par dégager les propriétés spécifiques aux réseaux de neurones qui sont à l'origine de leur intérêt pratique toujours croissant. Nous présentons, par la suite une méthodologie pour l'application ou la mise en oeuvre de ces techniques neuronales. Cette partie s'achèvera par une revue générale des domaines d'applications.

2.7.1 Propriétés spécifiques aux réseaux de neurones

L'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans les quelques propriétés fascinantes qu'ils possèdent. Nous citons, tout d'abord, le parallélisme, la capacité

d'adaptation et la mémoire distribuée, qui proviennent directement de la modélisation des propriétés du système nerveux de l'homme.

La notion de parallélisme se situe à la base de l'architecture des réseaux de neurones, considérés comme ensembles d'entités élémentaires qui travaillent simultanément. Le parallélisme permet une rapidité de calcul mais exige tout de même de penser et de poser différemment les problèmes à résoudre.

La capacité d'adaptation se manifeste par la capacité d'apprentissage dans les réseaux de neurones, ce qui permet au réseau de tenir compte de nouvelles contraintes ou de nouvelles données du monde extérieur. Cette propriété se caractérise également, dans certains réseaux, par leur capacité d'auto-organisation.

La mémoire est distribuée sur plusieurs entités. Elle correspond, dans un réseau donné, à l'activation des neurones et surtout à la constellation ou évolution des synapses. La perte d'un élément ne correspond pas à la perte d'un fait mémorisé, ce qui n'est pas le cas en informatique, où une donnée correspond à une case mémoire et la perte de cette case entraîne la perte de la donnée stockée.

Parmi les principales propriétés qui émergent directement de l'étude du fonctionnement des réseaux de neurones formels est la capacité de généralisation. À partir d'ensembles d'exemples, les réseaux de neurones apprennent à retrouver des règles sous-jacentes ou à mimer des comportements qui permettent de résoudre le problème posé. Cette propriété s'avère être une propriété très importante et surtout essentielle dans diverses applications. Elle donne lieu à une propriété mathématique fondamentale des réseaux de neurones: l'approximation universelle parcimonieuse.

Propriété fondamentale des réseaux de neurones

Une propriété fondamentale des réseaux de neurones est *l'approximation universelle parcimonieuse*. Cette expression traduit deux propriétés distinctes: d'une part, les réseaux de neurones sont des approximateurs universels et d'autre part, une approximation à l'aide des

réseaux de neurones nécessite, en général, moins de paramètres ajustables que les approximations usuelles.

L'approximation universelle La propriété d'approximation universelle a été démontrée par Cybenko[9, 43] et peut s'énoncer de la façon suivante:

Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire.

C'est cette propriété qui justifie notre choix de l'architecture de réseaux de neurones à une couche cachée. D'un point de vue pratique, cette propriété réduit grandement la recherche de la structure adaptée à la tâche que l'on souhaite réaliser: une fois les variables du réseau déterminées, la complexité du modèle est simplement contrôlée par le nombre de neurones dans la couche cachée. Ce dernier est le seul degré de liberté qui subsiste pour la détermination de l'architecture du réseau. Il convient de souligner que cette propriété d'approximation universelle n'est pas spécifique aux réseaux de neurones: il existe bien d'autres familles de fonctions paramétrées possédant cette propriété; c'est le cas notamment des ondelettes et des fonctions radiales, par exemple. La spécificité des réseaux de neurones réside dans le caractère *parcimonieux* de l'approximation.

La parcimonie

En pratique, lorsqu'on cherche à modéliser un processus ou phénomène à partir de données, on cherche toujours à obtenir les résultats les plus satisfaisants possibles avec un nombre minimal de paramètres ajustables. Dans cette optique, Hornik[44] a montré que

Si le résultat de l'approximation (c'est à dire la sortie du réseau de neurones) est une fonction non linéaire des paramètres ajustables, elle est plus parcimonieuse que si elle est une fonction linéaire de ces paramètres. De plus, pour des réseaux de neurones à fonction d'activation sigmoïdale, l'erreur commise dans l'approximation varie comme l'inverse du nombre de neurones cachés, et elle est indépendante du nombre de variables de la fonction

à approcher. Par conséquent, pour une précision donnée, donc pour un nombre de neurones cachés donné, le nombre de paramètres du réseau est proportionnel au nombre de variables de la fonction à approcher.

Ces résultats s'appliquent aux réseaux de neurones à fonction d'activation sigmoïdale, puisque la sortie de ces neurones n'est pas linéaire par rapport à leurs coefficients. Ainsi, l'avantage des réseaux de neurones par rapport aux approximateurs universels (tels que les polynômes) est d'autant plus sensible que le nombre de variables de la fonction à approcher est grand: pour des problèmes faisant intervenir une ou deux variables, on peut généralement utiliser indifféremment des réseaux de neurones ou des polynômes. En revanche, pour des problèmes présentant trois variables ou plus, il est généralement avantageux d'utiliser les réseaux de neurones. Bien entendu, cette propriété est démontrée de manière générale et pourrait se révéler inexacte pour un problème particulier.

Qualitativement, la propriété de parcimonie peut se comprendre de la manière suivante: lorsque l'approximation est une combinaison linéaire de fonctions élémentaires fixées, il s'agit d'ajuster uniquement les coefficients de la combinaison. En revanche, lorsque l'approximation est une combinaison linéaire de fonctions non linéaires à paramètres ajustables, il s'agit d'ajuster les coefficients de la combinaison ainsi que les fonctions élémentaires à travers leurs paramètres spécifiques. On conçoit facilement que cette souplesse supplémentaire, conférée par le fait que l'on ajuste la forme des fonctions que l'on combine, permet d'utiliser un plus petit nombre de fonctions élémentaires, donc un plus petit nombre de paramètres ajustables. Il convient de noter que cette remarquable parcimonie justifie l'intérêt industriel des réseaux de neurones. En effet, en pratique, on cherche une approximation d'une fonction non linéaire modélisant un processus donné à partir de données disponibles. Le nombre de mesures nécessaires pour estimer les paramètres de manière significative est d'autant plus grand que le nombre de paramètres est grand. Ainsi, pour modéliser une grandeur avec une précision donnée à l'aide des réseaux de neurones, il faut moins de données que pour la modéliser, avec une précision comparable, à l'aide des tech-

niques classiques de modélisation non linéaires. De manière générale, on dit que les réseaux de neurones permettent de faire un meilleur usage des mesures disponibles.

2.7.2 Mise en oeuvre des réseaux de neurones formels

Il est avantageux d'utiliser des réseaux de neurones pour toute application nécessitant de trouver une relation non linéaire entre des données numériques. La mise en oeuvre d'un réseau de neurones nécessite donc:

- De déterminer les entrées pertinentes, c'est à dire les grandeurs ayant une influence significative sur le phénomène à modéliser. A titre d'exemple, lorsqu'on connaît la nature des entrées désirables pour un modèle donné, et que l'on cherche à faire un choix parmi ces grandeurs, on essaye de ne prendre en considération en entrée du modèle que les grandeurs qui agissent de manière significative sur le processus; c'est à dire, celles dont l'effet est plus important que l'incertitude de mesure.

- De collecter les données nécessaires à l'apprentissage et à l'évaluation de la performance.

- De trouver les nombres de neurones cachés pour obtenir une approximation satisfaisante. Théoriquement, l'écart entre l'approximation réalisée par un réseau de neurones et la tâche à approcher est inversement proportionnel au nombre de neurones cachés[45]. L'idée la plus naturelle consisterait à choisir le nombre de neurones cachés le plus grand possible. Cependant, en raison d'un nombre excessif de neurones cachés, et donc un nombre excessif de paramètres, la sortie du réseau peut passer avec une très grande précision par tous les points d'apprentissage, mais fournir des résultats dépourvus de signification entre ces points. Toutefois, la fonction réalisée par le réseau devrait, non seulement, passer le plus près possible, au sens des moindres carrées, des points de l'ensemble d'apprentissage, mais être également capable de généraliser de manière satisfaisante. A priori, il n'existe pas, à l'heure actuelle, de résultats précis permettant de prévoir le nombre de neurones cachés nécessaires pour obtenir une performance spécifiée du modèle, compte tenu des données disponibles. Il faudrait donc avoir recours à une procédure numérique de conception de modèle.

- D'estimer les paramètres du réseau correspondant à un minimum de la fonction de coût, c'est à dire, d'effectuer un apprentissage. La sortie d'un réseau de neurones étant non linéaire par rapport aux paramètres, la fonction de coût peut présenter des minimas locaux, et les algorithmes d'apprentissage ne donnent aucune garantie de trouver le minimum global. Le nombre d'exemples d'apprentissage joue un rôle fondamental dans l'existence des minimas locaux. En effet, lorsqu'on dispose d'un nombre suffisant d'exemples, le problème des minimas locaux ne se pose pratiquement pas. En pratique, face à ces problèmes, il suffit d'effectuer plusieurs apprentissages avec des initialisations différentes des paramètres. En procédant de la sorte, on a une plus grande chance de trouver le minimum global.

- D'évaluer les performances du réseau de neurones à l'issue de l'apprentissage. Pour cela, il convient de constituer, outre l'ensemble d'apprentissage utilisé pour le calcul des paramètres du réseau, un ensemble de test, composé de données différentes de celles de l'ensemble d'apprentissage, à partir duquel on estime les performances du réseau après un apprentissage.

2.7.3 Domaines d'applications

Les réseaux de neurones servent à toutes sortes d'applications dans divers domaines. Ils ont aujourd'hui un impact considérable et il est attendu que leur importance ira en grandissant dans le futur. Parmi ces domaines d'applications privilégiés, nous citerons:

- La reconnaissance de formes ou de caractères, domaine où les réseaux de neurones ont acquis leur notoriété et prouvé qu'ils constituent des alternatives fiables à d'autres méthodes de classification.

- Le traitement du signal et de l'image.

- La prévision et la modélisation de phénomènes statique et dynamique. La modélisation d'un processus consiste à représenter son comportement statique ou dynamique à l'aide d'un modèle mathématique paramétré. La première phase d'une modélisation consiste à rassembler les connaissances dont on dispose sur le comportement du processus, à partir d'expériences et/ou d'une analyse théorique des phénomènes physiques mis en jeu. La

seconde phase consiste à estimer les paramètres du modèle en mettant en œuvre un système d'apprentissage. L'estimation des paramètres du modèle est effectuée en minimisant la fonction de coût définie à partir de l'écart entre les sorties mesurées du processus et les valeurs prédites par le modèle. La qualité de cette estimation dépend de la richesse des séquences d'apprentissage et de l'efficacité de l'algorithme utilisé.

- L'approximation de fonctions non linéaires.
- Le diagnostic médical: on dispose d'une importante base de données de symptômes auxquels sont associés différents diagnostics qui ont été validés. Réaliser un réseau de neurones reproduisant le diagnostic peut-être effectué en utilisant les couples symptômes/diagnostics en tant qu'échantillon d'apprentissage.

CHAPITRE

3

Investigation systématique d'un réseau de neurones pour l'approximation de tâches

3.1 Introduction

Depuis maintenant plusieurs années, les modèles des réseaux de neurones ont connu une grande popularité et suscité un intérêt toujours croissant. Ils sont largement utilisés dans le cadre de la modélisation statistique: classification, reconnaissance de formes, régression non-linéaire, prédiction de séries temporelles...[46]-[48]. Ils ont été aussi utilisés avec succès pour l'identification et le contrôle des systèmes dynamiques[49]. Le problème de la description analytique d'un ensemble de données apparaît dans de nombreuses sciences et applications et peut être identifié comme étant un problème de modélisation de données ou d'identification d'un système. Les réseaux de neurones constituent un moyen adéquat de représentation, car connus pour être des approximateurs universels[1]. La tâche souhaitée est généralement obtenue grâce à une procédure d'apprentissage qui consiste en un ajustement des "poids synaptiques". Pour cela, une multitude d'algorithmes d'apprentissage ont été proposés pour ajuster ces poids. La convergence de ces algorithmes d'apprentissage est un critère déterminant pour que les réseaux de neurones soient utiles dans différentes applications[50, 51]. En réalité, un effort considérable a été consenti pour le développement de techniques d'accélération de convergence des algorithmes d'apprentissage. De plus, il est de plus en plus admis que le choix des fonctions d'activation est aussi important que le choix de l'architecture du réseau et de l'algorithme d'apprentissage. Les sigmoïdes[1, 9] ont été les premières fonctions d'activation proposées. Leur utilisation était, en fait, initialement justifiée par une analogie biologique. Il a été démontré par plusieurs auteurs qu'elles peuvent approcher une fonction continue sur un domaine compact avec une précision arbitraire pour un nombre donné de cellules ou neurones cachés. En dépit du fait que les sigmoïdes sont les fonctions les plus communément utilisées, il n'existe à priori aucune raison pour qu'elles soient optimales pour tous les cas. Il est alors légitime d'explorer les possibilités d'utilisation d'autres types de fonctions. A ce propos, plusieurs fonctions d'activation ont été proposées depuis, parmi lesquelles nous citerons la Gaussienne[52], la Lorentzienne[53] et l'onde plane[54].

Tout au long des deux dernières décennies, l'approximation sur la base de fonctions ondelettes[14, 55, 56] a fait l'objet d'un profond regain d'intérêt en tant qu'alternative aux

autres techniques plus classiques. Rappelons que le principe général de l'analyse en ondelettes repose sur l'utilisation des dilatations et des translations. En effet, en modifiant les valeurs des paramètres de dilatation, les ondelettes peuvent distinguer les caractéristiques locales d'un signal à différentes échelles et grâce aux translations, elles peuvent couvrir toute la région dans laquelle il est étudié. Il a été prouvé que toute fonction peut être écrite sous la forme d'une superposition d'ondelettes admissibles. Les coefficients d'un tel développement caractérisent complètement la fonction qu'il est possible de reconstruire de manière numériquement stable en connaissant ces coefficients. Les ondelettes constituent une classe de fonctions satisfaisant un ensemble de propriétés mathématiques importantes. Considérons, sans perte de généralité, que ces fonctions sont limitées, ont un support compact et que leurs translations et dilatations génèrent une base de fonctions pour la représentation de n'importe quel élément appartenant à une famille de fonctions de carrés sommables. Il a été démontré que ces familles de fonctions sont des approximateurs universels[34]. Par conséquent, les réseaux d'ondelettes peuvent être considérés comme une alternative aux réseaux de neurones classiques (les perceptrons multi-couches) et aux réseaux à fonctions radiales.

Dans le langage des réseaux de neurones, on désigne par modèle connexionniste, un ensemble de réseaux dont les cellules de base, reliées entre elles selon une certaine architecture, utilisent une classe donnée de fonctions d'activation. Par conséquent, à un modèle connexionniste correspond une classe de fonctions. Nous considérons, dans ce qui suit, trois modèles de réseaux de neurones à une seule couche cachée, comportant N cellules, permettant de répondre au problème d'approximation de tâches. Ces trois modèles sont:

- Le perceptron multicouches, où la fonction d'activation utilisée est la fonction sigmoïde. Cette dernière étant une fonction continue, indéfiniment dérivable et bornée, qui se prête donc facilement au calcul dans l'apprentissage des réseaux.
- Un réseau à fonctions de base radiales, où la fonction d'activation choisie est une gaussienne.

- Un réseau d'ondelettes. Pour cela, nous avons fait le choix de deux ondelettes couramment utilisées: l'ondelette de Morlet et l'ondelette «Chapeau Mexicain».

Nous disposons alors de trois modèles équivalents sur le plan théorique permettant, pour des réseaux à une couche de cellules cachées, une approximation avec une précision arbitraire. Ceci ne signifie pas que les ensembles de fonctions générées soient tous équivalents. L'utilisation pratique des résultats théoriques concernant les capacités de ces modèles connexionnistes s'avère difficile. Ce problème reste, en effet, une question d'actualité, non encore résolu pour le développeur d'applications. Pour cette raison, situer ces différents modèles connexionnistes reste toujours une question ouverte.

Dans ce chapitre, nous nous proposons d'étudier les capacités des réseaux d'ondelettes relativement aux deux autres modèles connexionnistes sus-cités. Nous effectuerons, en outre, une investigation systématique faisant intervenir tous les paramètres susceptibles d'améliorer l'approximation des tâches considérées[17].

3.2 Architecture du réseau et apprentissage

Nous allons d'abord donner un aperçu succinct du modèle et de l'architecture du réseau de neurones, ce dernier émanant directement des résultats séminaux de Kolmogorov[58, 59]. Il s'agit du second théorème de Kolmogorov qui a été subséquemment reformulé par Kurkova[60] de manière plus pratique quant aux applications des réseaux de neurones artificiels et avec moins de contraintes à satisfaire sur les fonctions à approcher. Le théorème de Kolmogorov stipule que *toute fonction continue sur le cube unité E^n de dimension n ($E = [0, 1]$), peut être représentée sous la forme*

$$f(x_1, x_2, \dots, x_n) = \sum_{j=1}^{2n+1} \Psi_j \left(\sum_{i=1}^n \phi_{ij}(x_i) \right) \quad (3.2.1)$$

où Ψ_j et ϕ_{ij} sont des fonctions réelles et continues. Les fonctions ϕ_{ij} sont indépendantes de la fonction donnée f alors que les fonctions Ψ_j sont spécifiques à f . Plus tard, Hecht-Nielsen[61] nota une ressemblance entre la structure formelle de l'expansion de Kolmogorov

sur la base de fonctions continues à travers d'autres fonctions auxiliaires avec un réseau de neurones à trois couches. Il a alors reformulé le théorème de Kolmogorov comme suit

Toute fonction continue définie sur le cube E^n à n dimensions ($E = [0, 1]$), peut être implémentée exactement grâce à un réseau à trois couches ayant $(2n+1)$ cellules dans la couche cachée avec des fonctions d'activation de type sigmoïdale de l'entrée à la couche cachée.

Autrement dit, un réseau à une seule couche cachée est capable de décrire toute fonction continue, en utilisant $(2n + 1)$ cellules, où n est le nombre d'entrées. En pratique, il est bien entendu que ces minimas théoriques ne sont atteints qu'au prix d'un nombre plus grand de neurones cachés.

En vertu de ce qui a été énoncé ci-dessus, considérons un réseau à trois couches composé d'une entrée X , d'une couche cachée de réponses élémentaires et pour les besoins de simplification, on se contentera d'un neurone de sortie linéaire. Les cellules élémentaires de la couche cachée reçoivent la même entrée X . Chaque unité ou cellule donne lieu à une sortie f qui dépend de deux paramètres: le paramètre de translation b et le paramètre échelle λ . Les poids synaptiques de sortie $\omega(b, \lambda)$ regroupent linéairement ces sorties élémentaires en une sortie globale $F(X)$. Nous obtenons alors l'expansion suivante[37]

$$F(X) = \int \omega(b, \lambda) f\left(\frac{X-b}{\lambda}\right) dbd\lambda \quad (3.2.2)$$

En discrétisant l'équation (3.2.2) à l'aide de N unités ou cellules, nous obtenons la fonction approchée suivante

$$F_{app}(X) = \sum_{i=1}^N \omega(b_i, \lambda_i) f\left(\frac{X-b_i}{\lambda_i}\right) \quad (3.2.3)$$

Pour définir la meilleure approximation possible F_{app} , il convient de minimiser l'erreur quadratique moyenne $\varepsilon = \langle F - F_{app} | F - F_{app} \rangle$ commise, d'abord par rapport aux poids synaptiques ω_i , ensuite par rapport aux paramètres translation et échelle b_i et λ_i respectivement. Par rapport aux poids ω_i , la minimisation consiste à résoudre l'équation

$$\frac{\partial}{\partial \omega_i} (\langle F - F_{app} | F - F_{app} \rangle) = 0 \quad (3.2.4)$$

En utilisant l'équation (3.2.3) et en adoptant la notation simplifiée suivante

$$f((X - b_i)/\lambda_i) = f_i \quad (3.2.5)$$

nous obtenons

$$\frac{\partial}{\partial \omega_i} \left(\langle F | F \rangle - 2 \sum_{j=1}^N \omega_j \langle f_j | F \rangle + \sum_{j,k=1}^N \omega_j \langle f_j | f_k \rangle \omega_k \right) = 0, \quad i = 1, \dots, N. \quad (3.2.6)$$

dont la solution est donnée par

$$\omega_i = \sum_{j=1}^N (g^{-1})_{ij} \langle f_j | F \rangle, \quad i = 1, \dots, N. \quad (3.2.7)$$

où g représente la matrice dont les éléments sont donnés par

$$g_{jk} = \langle f_j | f_k \rangle \quad (3.2.8)$$

Minimisons alors l'erreur quadratique moyenne par rapport aux paramètres échelle λ_i , en notant que nous n'avons besoin que des dérivées de f_i par rapport aux λ_i . Nous obtenons alors

$$\frac{\partial \varepsilon}{\partial \lambda_i} = \frac{-2\omega_i}{\lambda_i^2} \left\langle X f' \left(\frac{X - b_i}{\lambda_i} \right) | F - F_{app} \right\rangle = 0, \quad i = 1, \dots, N \quad (3.2.9)$$

La minimisation de ε par rapport aux paramètres translation b_i , donne

$$\frac{\partial \varepsilon}{\partial b_i} = \frac{-2\omega_i}{\lambda_i} \left\langle f' \left(\frac{X - b_i}{\lambda_i} \right) | F - F_{app} \right\rangle = 0, \quad i = 1, \dots, N \quad (3.2.10)$$

où f' représente la dérivée de la tâche élémentaire avant toute translation et dilatation.

3.3 Résultats numériques et discussion

Nous allons maintenant procéder à la présentation de nos résultats numériques. Pour cela, considérons la tâche suivante présentant plusieurs oscillations localisées[16]

$$F(X) = 0.10167 \exp(-X/10) \{0.60717 \tanh[4(X - 1.66133)] - 4.33575 \tanh[4(X - 9.56591)]\} \quad (3.3.1)$$

Dans la recherche de la meilleure approximation de F dans l'intervalle $[0, 20]$, nous utilisons un réseau à $N = 5$ cellules cachées. L'apprentissage du réseau est effectué par la descente du gradient avec un pas fixe, en minimisant l'erreur quadratique moyenne commise sur l'approximation par rapport à l'ensemble des paramètres du réseau. Ces paramètres sont ensuite corrigés en fonction de la direction fournie par le calcul. L'initialisation des paramètres est faite soit de manière aléatoire, soit donnée par la formule heuristique, traditionnellement utilisée pour les paramètres échelles des réseaux d'ondelettes, $\lambda_i = 2^{i-(N+1)/2}$. Après environ 80 pas, $\|F_{app}\|^2$ se sature et une assez bonne comparaison entre F et F_{app} peut être reproduite. Signalons au passage que toute exécution numérique de notre code de calcul converge de manière régulière vers une saturation de la norme $\|F_{app}\|^2$. La figure 11 illustre nos résultats obtenus dans le cas d'un réseau de type perceptron, avec comme fonction d'activation une sigmoïde $\left[f(x) = \frac{1}{1+\exp(-x)} \right]$. Les deux courbes (la tâche et sa meilleure approximation), après un léger rapprochement, dévient l'une de l'autre. Cette déviation devient moins prononcée dans le cas des réseaux à fonctions gaussiennes $\left[f(x) = A \exp\left(-\frac{x^2}{2}\right) \right]$ comme l'indique la figure 12. Ce résultat est principalement dû au profil gaussien qui semble mieux convenir que la sigmoïde dans l'approximation de la tâche considérée (3.3.1). F_{app} est alors développée, successivement, sur la base de la famille des ondelettes de "Morlet" $\left[f(x) = \cos(6x) \exp\left(-\frac{x^2}{2}\right) \right]$ et "Chapeau Mexicain" $\left[f(x) = (1 - x^2) \exp\left(-\frac{x^2}{2}\right) \right]$ et nos résultats sont tracés, respectivement, sur les figures 13 et 14. Ces dernières montrent une nette amélioration par rapport aux deux cas précédents (en particulier, dans le cas de l'ondelette "Chapeau Mexicain"). Il apparaît clairement que le choix des fonctions de base peut affecter la qualité de l'approximation.

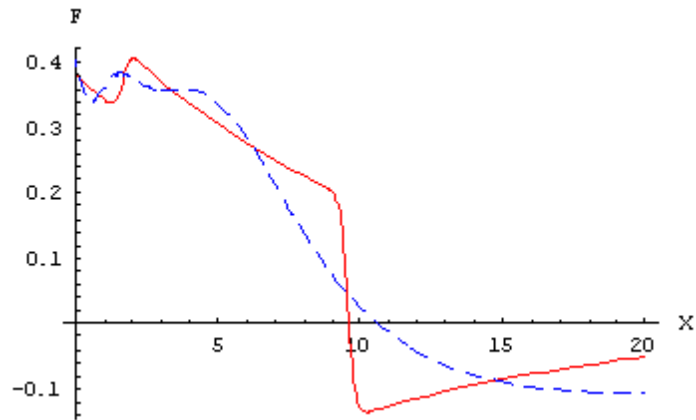


Figure 11: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de la sigmoïde.

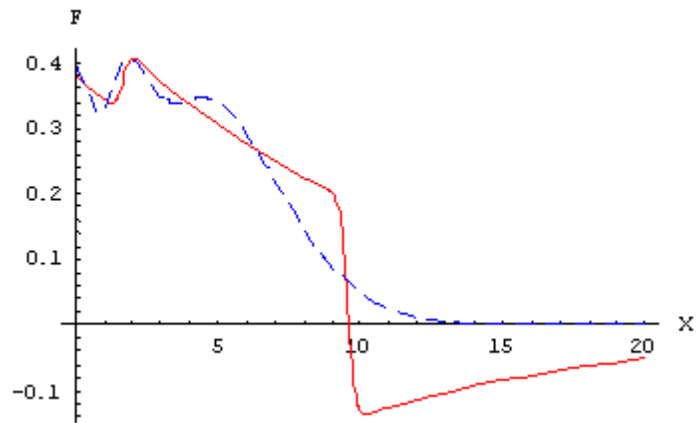


Figure 12: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de la gaussienne.

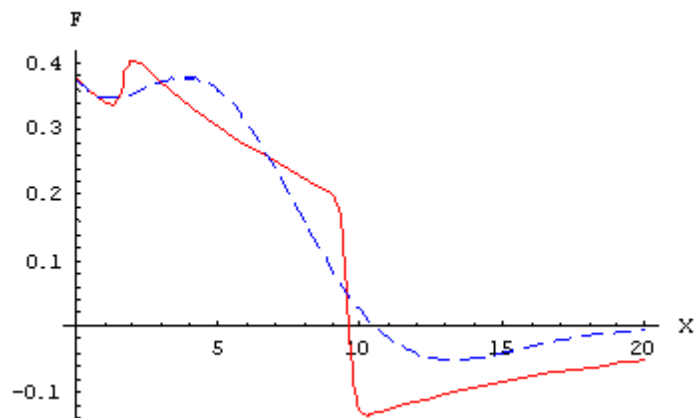


Figure 13: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette de Morlet.

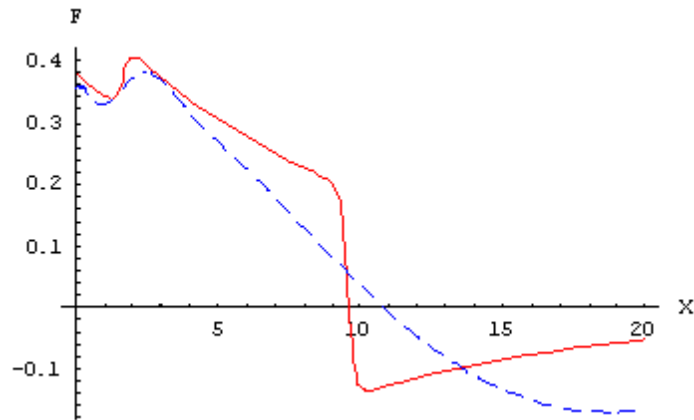


Figure 14: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain".

Il est important de signaler que contrairement aux cas des réseaux à fonctions sigmoïdales où l'initialisation des paramètres est effectuée de manière aléatoire, pour les réseaux d'ondelettes, une attention particulière doit être portée à cette phase d'initialisation des paramètres. Les ondelettes étant des fonctions à décroissance rapide, une initialisation aléatoire des paramètres de translation et de dilatation serait inefficace: en effet, si les translations sont initialisées à l'extérieur du domaine contenant les exemples (ou encore le domaine de définition de la tâche à approcher), ou si les dilatations choisies sont trop petites, la sortie de l'ondelette est pratiquement nulle, de même que sa dérivée. L'algorithme d'adaptation des paramètres étant fondé sur une technique de gradient, il devient inopérant.

Comme l'algorithme utilisé ajuste de manière itérative les poids du réseau pour minimiser l'erreur quadratique moyenne (noté ci-après MSE), qui indique la déviation moyenne par rapport à la tâche, nous avons alors décidé de l'utiliser comme critère d'évaluation de la précision des différents modèles connexionnistes utilisant différentes fonctions d'activation. Prenant $N = 5$ et un nombre d'itérations $itr = 71$, les résultats obtenus, résumés dans le tableau suivant,

Fonction d'activation	MSE (%)
Sigmoïde	13.6
Gaussienne	13
Ondelette de Morlet	12
Ondelette "Chapeau Mexicain"	11.6
Ondelette "Chapeau Mexicain" avec translation	5.2

révèlent que l'ondelette "Chapeau Mexicain" produit la meilleure précision. L'amélioration de la précision est plus significative dans le cas où le paramètre de translation est pris en compte. Tenant compte de cela, nous avons alors décidé de choisir l'ondelette "Chapeau Mexicain" comme fonction d'activation pour l'étude de l'influence du nombre de neurones et du nombre d'itérations sur la meilleure approximation.

Pour montrer l'influence du nombre d'itérations sur la meilleure approximation, nous avons considéré, respectivement, 501 et 1001 pas et les résultats sont tracés sur les figures 15 et 16. Nos résultats indiquent qu'une augmentation du nombre d'itérations se traduit par une légère amélioration de la précision.

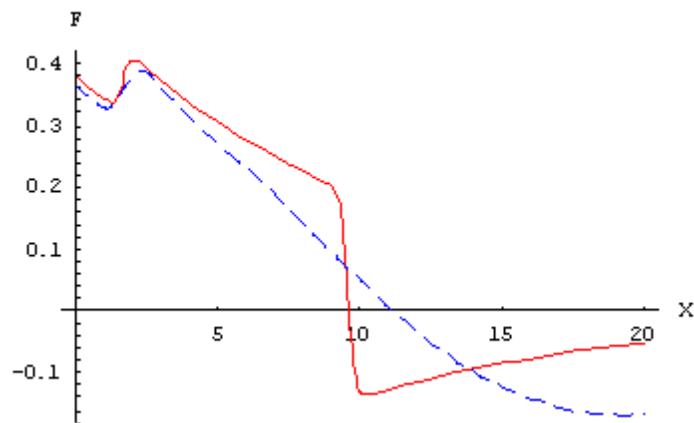


Figure 15: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" (figure 4) avec un nombre d'itérations égal à $itr = 501$.

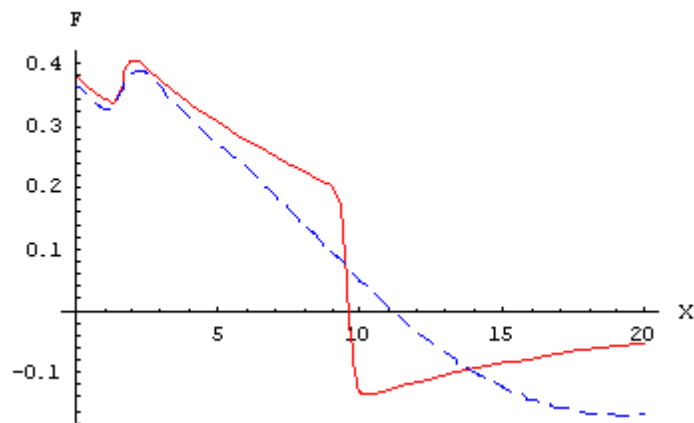


Figure 16: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) avec un nombre d'itérations égal à $itr = 1001$.

La figure 17 montre les variations de l'erreur quadratique moyenne en fonction du nombre d'itérations dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 5$. Les résultats obtenus montrent qu'une augmentation du nombre d'itérations entraîne une diminution de la valeur du MSE. Ce dernier décroît de manière abrupte avant d'atteindre une valeur fixe égale à 0.114. La prise en compte du paramètre de translation reproduit qualitativement des résultats similaires (figure 18) avec, cependant, une meilleure précision et une nette amélioration de l'approximation comme l'atteste la figure 19 (à comparer avec la figure 14). En effet, l'erreur sur l'approximation converge rapidement vers une valeur faible. Ceci est vraisemblablement lié à la convergence des paramètres de translation des cellules cachées. On comprend alors bien le comportement de l'algorithme d'apprentissage, qui consiste en fait à "aligner" les centres des fonctions de base sur les extremas de la fonction à approcher. Les paramètres de dilatation sont ensuite modifiés de manière à améliorer l'approximation fournie. Ce phénomène est principalement dû à l'utilisation de modèles à fonctions locales, pour l'approximation de fonctions présentant des variations ou des oscillations localisées. D'ailleurs, ce sera souvent le cas des fonctions réelles, qui sont généralement définies par un ensemble d'apprentissage. En effet, en dehors du domaine couvert par le vecteur d'entrée, ne disposant pas d'informations sur la fonction, on peut donc supposer que ses variations sont locales.

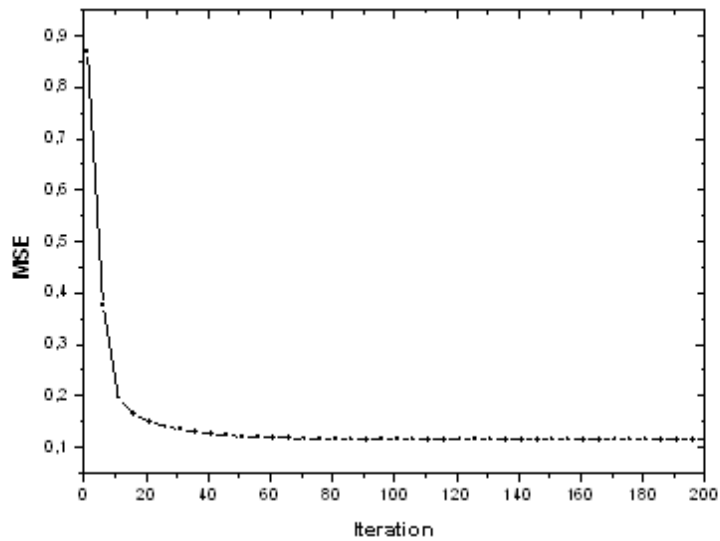


Figure 17: Tracé de la norme de l'erreur quadratique moyenne (MSE) en fonction du nombre d'itérations dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 5$. Le paramètre de translation est négligé.

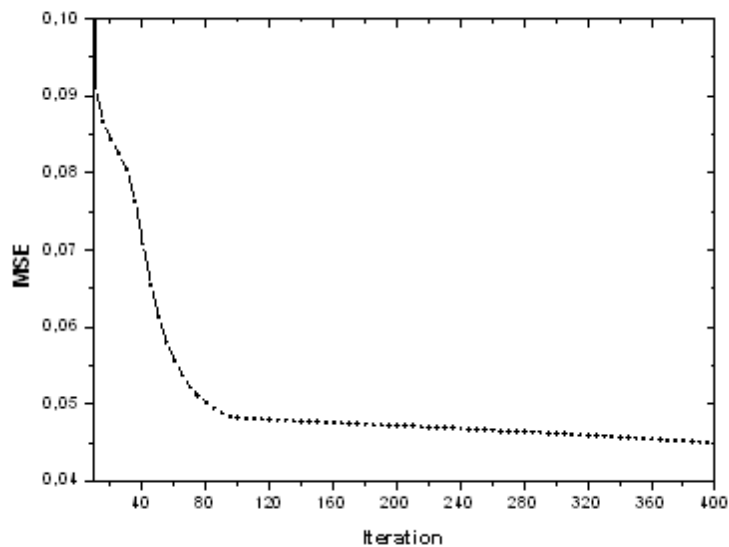


Figure 18: MSE en fonction du nombre d'itérations dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 5$. Le paramètre de translation est pris en compte.

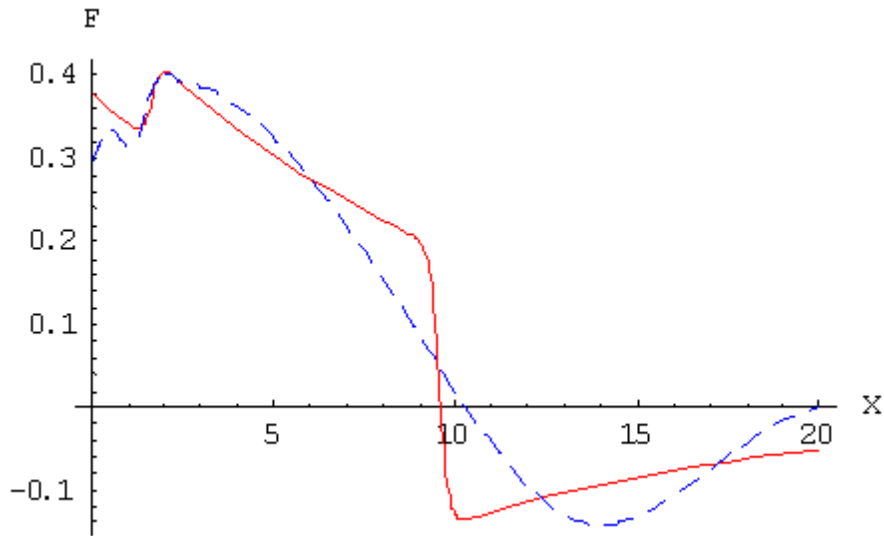


Figure 19: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 5$ et $itr = 71$. La paramètre de translation est pris en compte.

Par ailleurs, en variant la dimension N de la base des tâches élémentaires de 7 à 15, nous avons noté que ce dernier peut affecter la performance du réseau de neurones et, par conséquent, la meilleure approximation F_{app} . Les résultats sont résumés dans les figures 20-24, où la fonction et sa meilleure approximation sont représentées pour 5 valeurs choisies de N , soient $N = 7, 9, 11, 13$, et 15 . Il est clair qu'une augmentation de N entraîne une nette amélioration de F_{app} . La figure 25 représente l'erreur quadratique moyenne (MSE) en fonction du nombre de neurones dans le cas de l'ondelette "Chapeau Mexicain" avec un nombre d'itération égal à 71. Nous constatons qu'à mesure que le nombre de neurones N augmente, l'erreur quadratique moyenne décroît, entraînant une nette amélioration de la meilleure approximation. Cette amélioration est plus nette et plus visible dans le cas où le paramètre de translation est pris en compte (figure 26) donnant ainsi lieu à de très faibles valeurs du MSE comme l'indique d'ailleurs la figure 27 (à comparer avec la figure 24) en dépit du caractère oscillatoire de la tâche choisie.

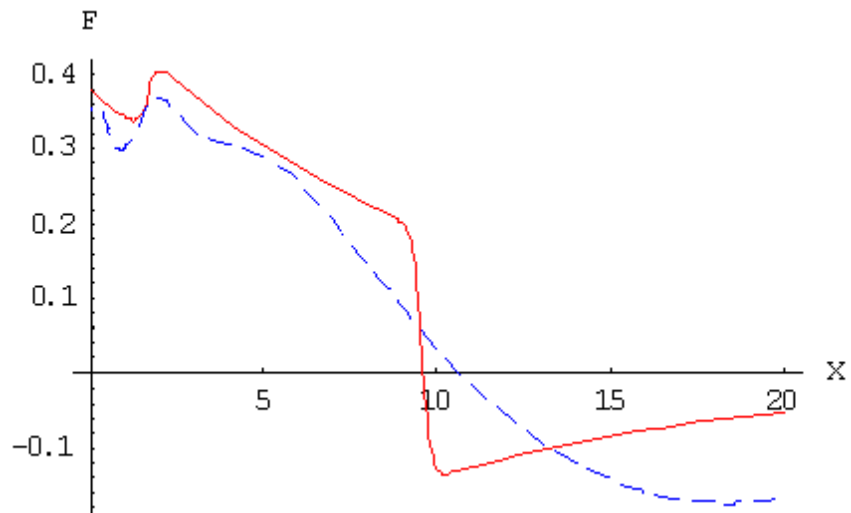


Figure 20: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 7$.

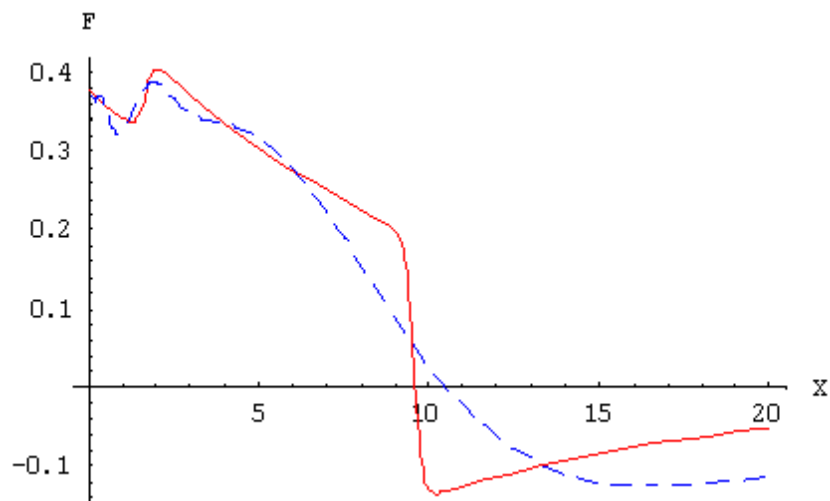


Figure 21: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 9$.

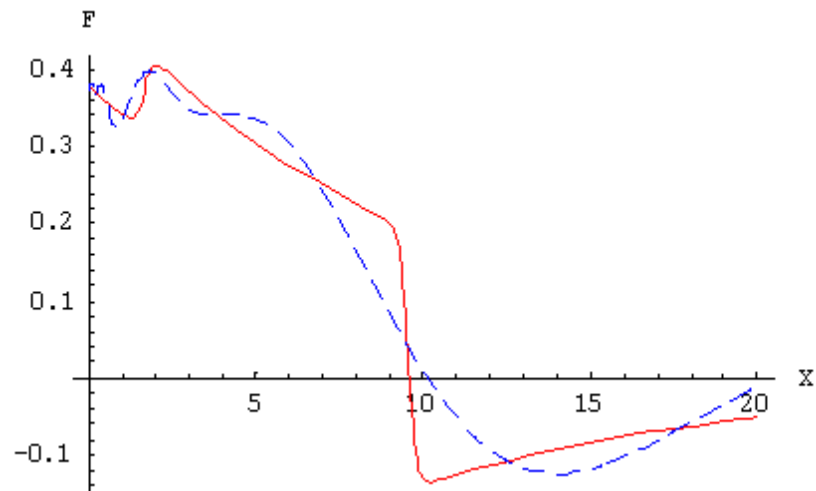


Figure 22: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 11$.

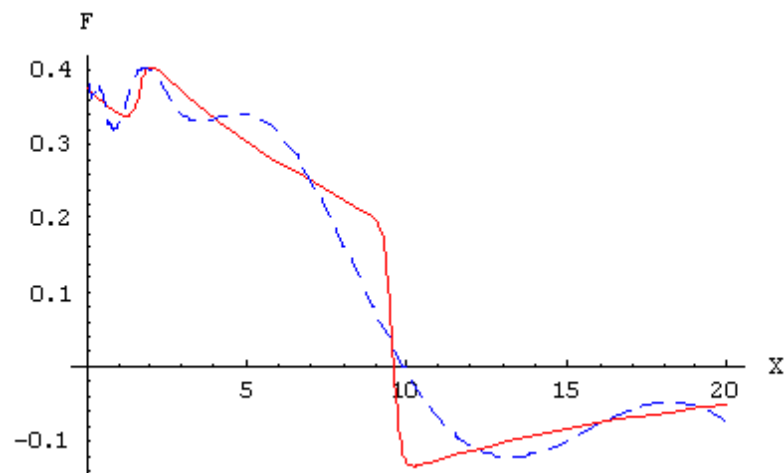


Figure 23: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 13$.

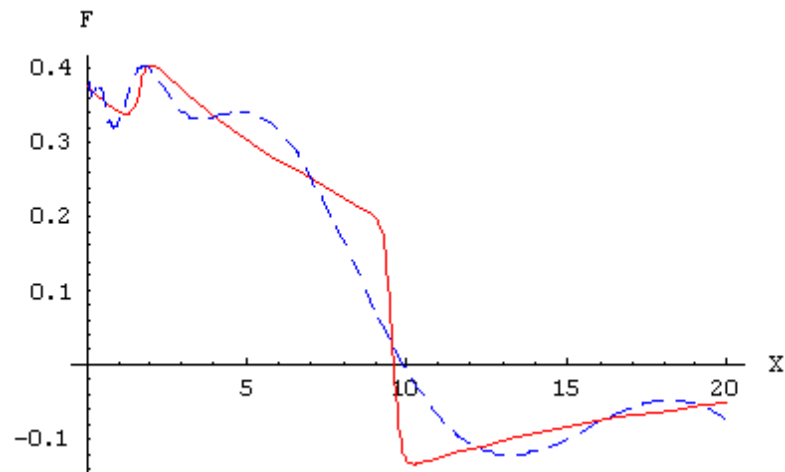


Figure 24: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 15$.

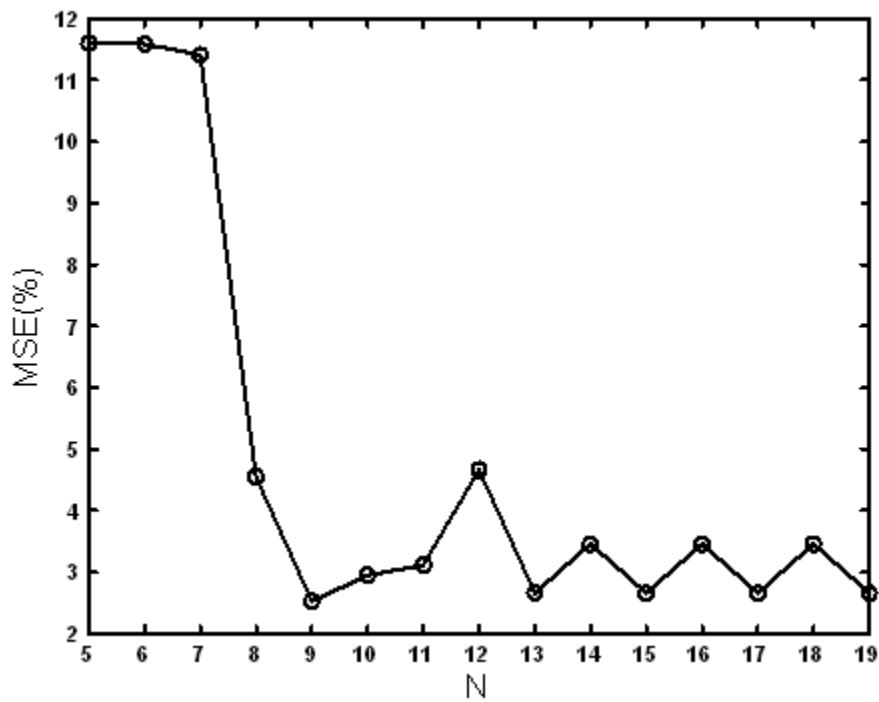


Figure 25: MSE en fonction du nombre de neurones N dans le cas de l'ondelette "Chapeau Mexicain". Le nombre d'itérations est $itr = 71$ et le paramètre de translation est négligé.

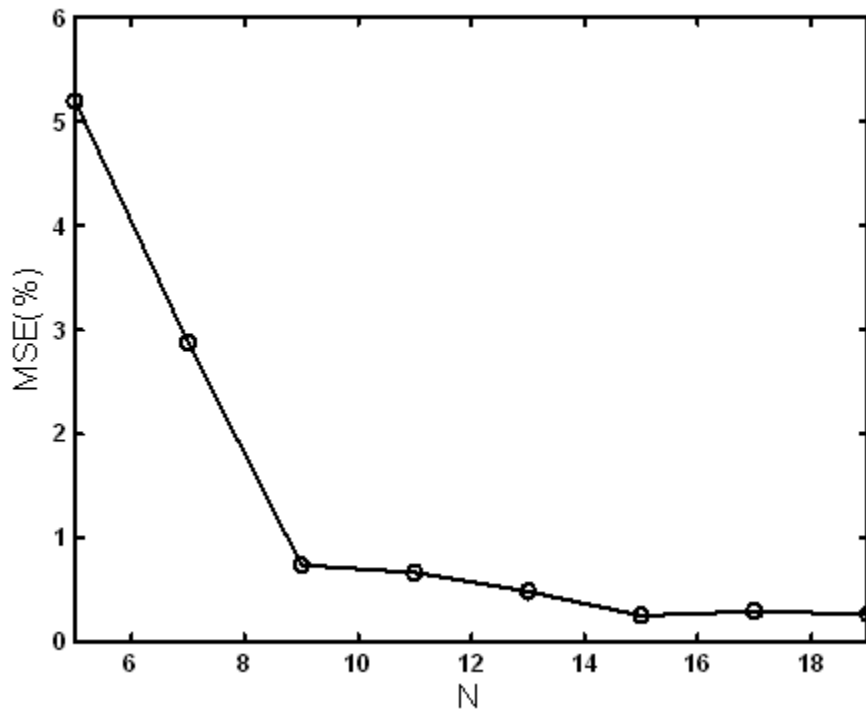


Figure 26: MSE en fonction du nombre de neurones N dans le cas de l'ondelette "Chapeau Mexicain". Le nombre d'itérations est $itr = 71$ et le paramètre de translation est pris en compte.

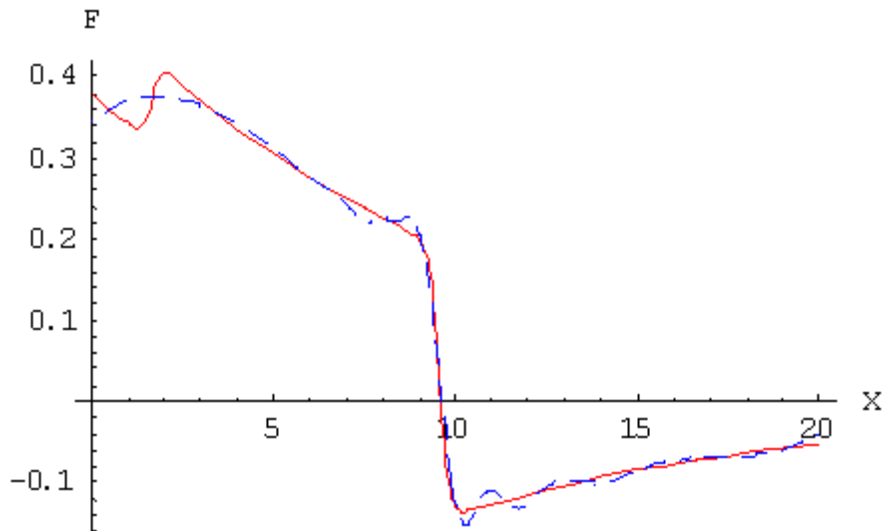


Figure 27: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 15$ et $itr = 71$. Le paramètre de translation est pris en compte.

Il est utile de noter que généralement un petit nombre de neurones est suffisant pour atteindre une très bonne approximation dans le cas de tâches relativement lissées. En effet, en considérant les deux tâches suivantes

$$F(X) = (1 + 2X - X^2) \exp(-X^2/2) \quad (3.3.2)$$

$$F(X) = \exp[-(X + 1)/2] X \sin(3X + 7) \quad (3.3.3)$$

communément utilisées dans la littérature, nous avons constaté qu'une très bonne approximation a été atteinte dans les deux cas (voir figures 28-29), bien que le paramètre de translation ait été négligé (MSE=0.88 · 10⁻⁴ et 0.15 · 10⁻² dans le cas, respectivement, des tâches (3.3.2) et (3.3.3)).

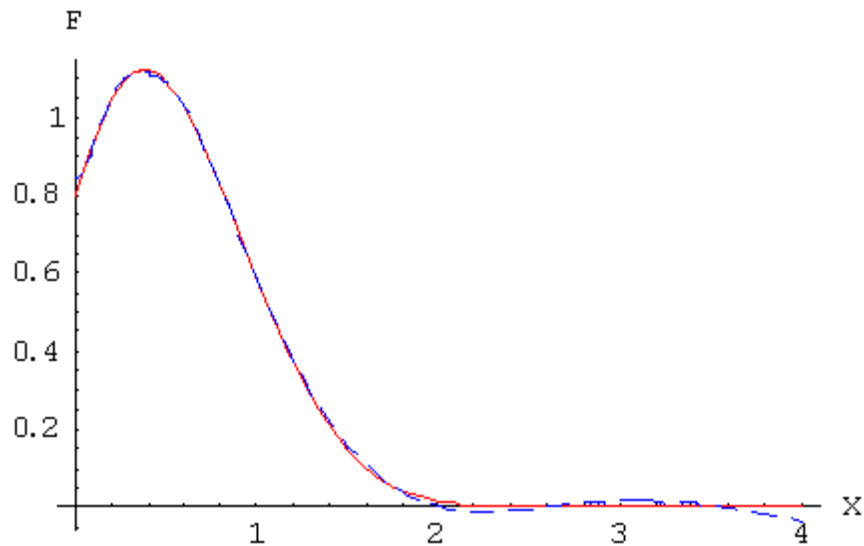


Figure 28: Tracé de la fonction $F(X) = (1 + 2X - X^2) \exp(-X^2/2)$ (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 5$ et $itr = 71$. Le paramètre de translation est négligé.

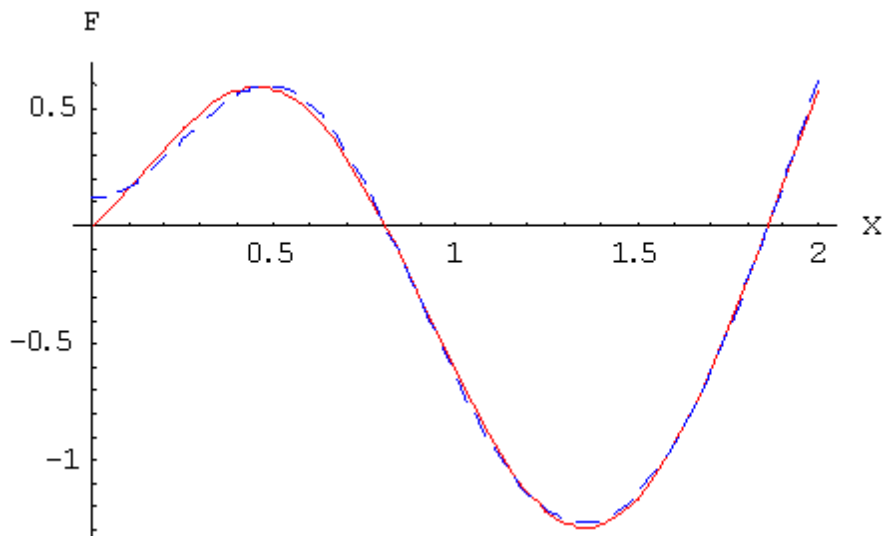


Figure 29: Tracé de la fonction $F(X) = \exp[-(X+1)/2] X \sin(3X+7)$ (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "Chapeau Mexicain" avec $N = 5$ et $itr = 71$. Le paramètre de translation est négligé.

Afin de se rapprocher plus du minimum de la fonction de coût, nous avons également considéré une méthode de gradient du second ordre: l'algorithme BFGS, décrit dans le chapitre précédent. En effet, cette méthode, dite "quasi-newtonienne", n'est efficace que si elle est appliquée au voisinage d'un minimum. D'autre part, la règle du gradient simple est efficace lorsqu'on est loin du minimum et sa convergence ralentit considérablement lorsque la norme du gradient diminue (c'est à dire lorsqu'on s'approche du minimum). Ces deux techniques sont donc complémentaires. De ce fait, nous avons opté pour une optimisation en deux étapes: utilisation de la règle du gradient simple pour approcher un minimum, et de l'algorithme BFGS pour l'atteindre. Au delà d'environ 80 itérations, nous avons alors complété l'apprentissage avec l'algorithme BFGS. Nos résultats (figure 30) indiquent que la valeur de MSE passe de 11.7% à 71 itérations à une valeur de 7.18% à 1001 itérations dans le cas de la tâche (3.3.1). Il est important de signaler que la difficulté essentielle lors de l'application de cet algorithme réside dans le choix de la condition de passage du gradient simple à la méthode de BFGS.

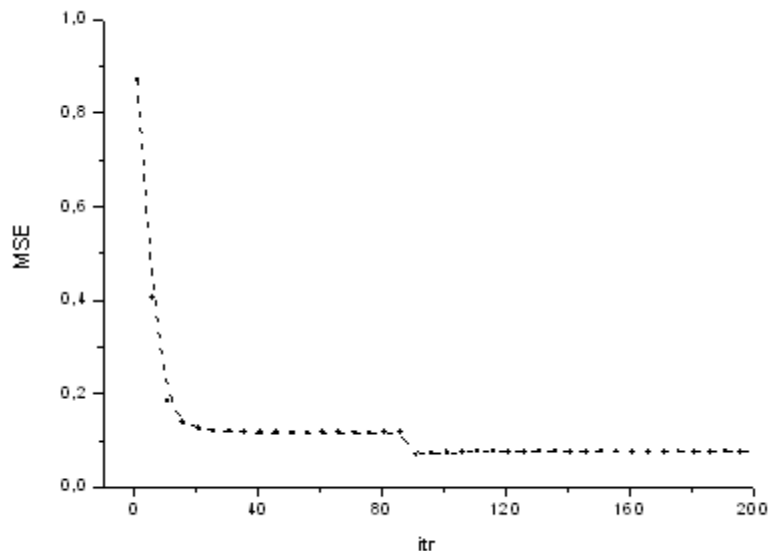


Figure 30: MSE en fonction du nombre d'itérations. L'apprentissage est d'abord effectué à l'aide de l'algorithme du gradient simple et, au delà de 80 itérations, l'algorithme BFGS est implémenté.

Notons finalement qu'il est quasiment impossible de conclure de manière définitive quant à la supériorité d'un modèle en général, tant la diversité des fonctions à tester est grande. Nous avons constaté, néanmoins, sur une multitude de fonctions considérées que les différents modèles fournissent des approximations plus ou moins bonnes, selon la proximité du modèle de l'approximation avec la structure de la fonction à approcher.

Dans une étude précédente[16], Giraud et Touzeau ont, tout en montrant la manière avec laquelle évoluent les paramètres d'échelle durant les 100 premiers pas de la descente du gradient, noté et mis en évidence la fusion de deux paires d'échelles. Ils ont alors suggéré que de nouvelles cellules peuvent spontanément émerger durant le processus d'apprentissage et éventuellement représenter les dérivées de la tâche élémentaire par rapport à leurs paramètres d'échelle. Dans le but de confirmer et d'étayer cette suggestion, nous avons considéré différentes tâches contenant la première, la seconde et la troisième dérivée de la fonction d'activation par rapport aux paramètres d'échelle. Tel que indiqué sur la figure 31, où la première dérivée de la fonction d'activation autour de $\lambda = 4$ est prise comme tâche à

approcher, l'évolution de $\text{Log}_2(\lambda_i)$, durant les premiers pas du processus d'apprentissage, exhibe une tendance à la fusion entre λ_1 et λ_2 confirmant ainsi la suggestion de Giraud et Touzeau. La même tendance persiste même si un neurone formel quelconque est ajouté à la tâche précédente: λ_1 et λ_2 adoptent une tendance similaire à la fusion alors que l'évolution de λ_3 dévie des deux autres valeurs (figure 32). En réalité, lorsque la fusion a lieu, i.e., $\lambda_i \simeq \lambda_j$, la base fonctionnelle dégénère compte tenu du fait que f_i et f_j ne sont plus distinctes ($f_i \simeq f_j$). La représentation strictement équivalente suivante

$$\omega_i f_i + \omega_j f_j \equiv (\omega_i + \omega_j)/2 \times (f_i + f_j) + (\omega_i - \omega_j)/2 \times (f_i - f_j) \quad (3.3.4)$$

montre qu'une renormalisation du terme $(f_i - f_j)$ fait que la base fonctionnelle contiendra, en plus du premier terme $(\omega_i + \omega_j)/2 \times (f_i + f_j)$, une nouvelle réponse élémentaire

$$\frac{\partial f}{\partial \lambda} = \lim_{\lambda_i \rightarrow \lambda_j} \frac{f_i - f_j}{\lambda_i - \lambda_j} \quad (3.3.5)$$

De manière analogue, une deuxième (troisième) dérivée de la fonction d'activation entraîne une tendance à la fusion entre $\lambda_{1,2,3}$ ($\lambda_{1,2,3,4}$) comme l'indique la figure 33 (figure 34). Dès que $\lambda_1 \simeq \lambda_2 \simeq \lambda_3$, les tâches élémentaires correspondantes f_1 , f_2 et f_3 deviennent quasi égales et donc la base fonctionnelle contient une nouvelle réponse élémentaire $\frac{\partial^2 f}{\partial \lambda^2}$. Ces nouvelles tâches élémentaires qui apparaissent spontanément, représentent les dérivées secondes des tâches élémentaires. Ce résultat peut donc être généralisé au cas où $(p + 1)$ paramètres d'échelles fusionnent durant le processus d'apprentissage: des dérivées d'ordre p , à savoir, $\frac{\partial^p f}{\partial \lambda^p}$, vont spontanément émerger dans la base fonctionnelle. Notons que pour les tâches qui exhibent un tel fusionnement, l'approximation de la fonction peut être légèrement améliorée et le temps d'apprentissage considérablement réduit en implémentant directement les tâches élémentaires ainsi que leurs dérivées directement dans la base fonctionnelle. Un tel résultat peut être déduit à partir de la figure 35 (à comparer avec la figure 14). Celle-ci illustre les résultats obtenus lorsque les dérivées premières de la tâche élémentaire sont directement implémentées dans la base fonctionnelle.

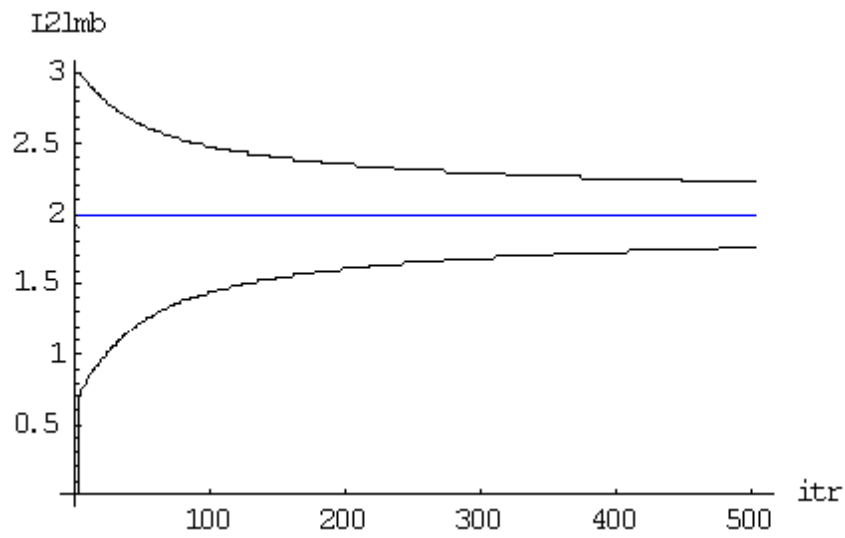


Figure 31: Variation de $\text{Log}_2(\lambda_i)$ en fonction du nombre d'itérations. La dérivée première de la fonction d'activation autour de $\lambda = 4$ est prise comme tâche à approximer.

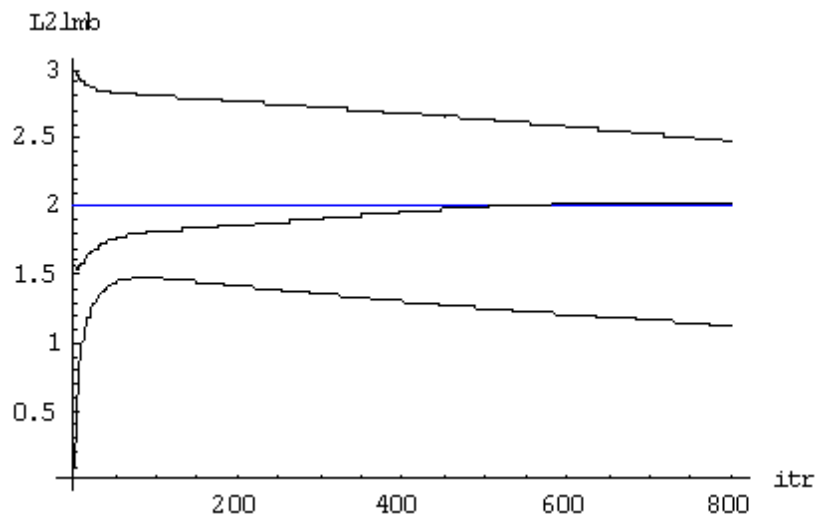


Figure 32: Variation de $\text{Log}_2(\lambda_i)$ en fonction du nombre d'itérations. Un neurone formel arbitraire est rajouté à la tâche de figure 31.

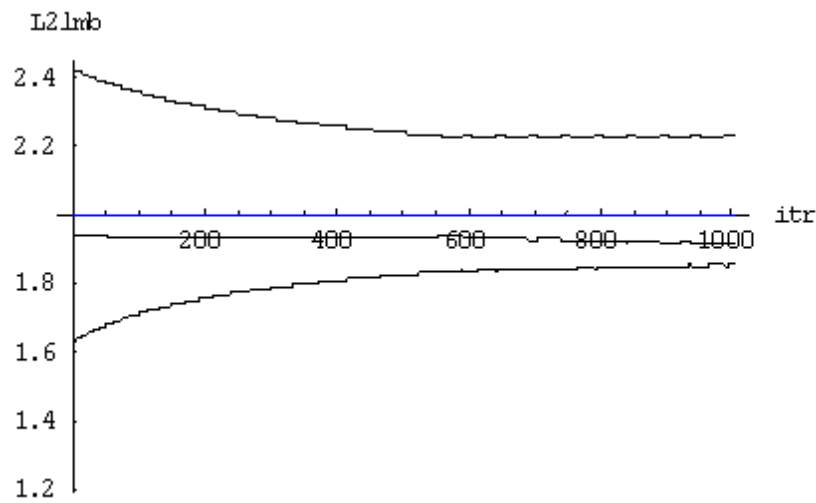


Figure 33: Variation de $\text{Log}_2(\lambda_i)$ en fonction du nombre d'itérations. La dérivée seconde de la fonction d'activation autour de $\lambda = 4$ est prise comme tâche à approximer.

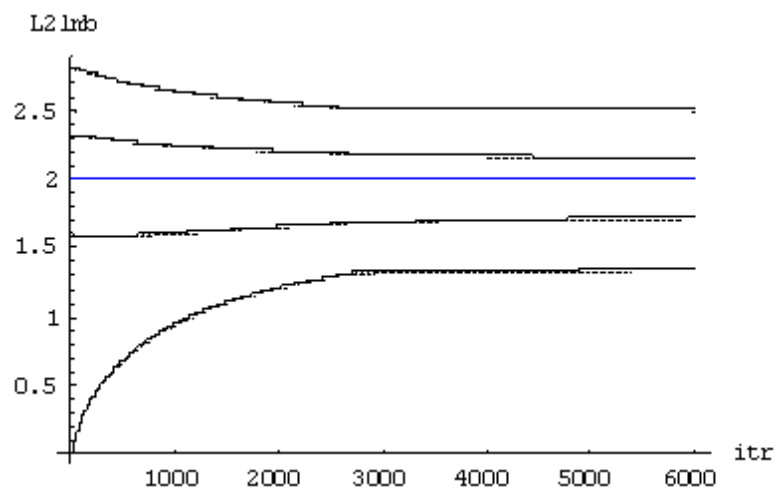


Figure 34: Variation de $\text{Log}_2(\lambda_i)$ en fonction du nombre d'itérations. La dérivée troisième de la fonction d'activation autour de $\lambda = 4$ est prise comme tâche à approximer.

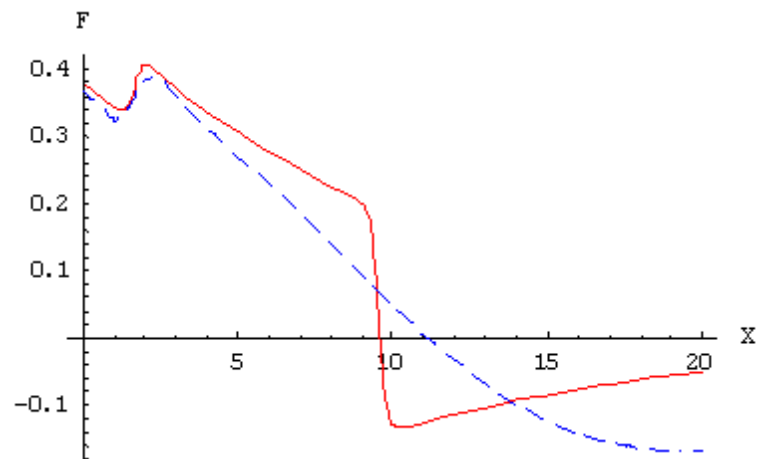


Figure 35: Tracé de la tâche (3.3.1) (trait plein) et sa meilleure approximation (tirets) dans le cas de l'ondelette "chapeau mexicain" lorsque les dérivées premières de la tâche élémentaire sont rajoutées dans la base fonctionnelle.

CHAPITRE

4

Analyse d'une structure solitaire à l'aide d'un schéma de réseaux de neurones

4.1 Introduction

Dans le chapitre précédent, nous avons procédé à l'investigation systématique d'un réseau de neurones pour l'approximation d'une fonction. Nous avons alors noté que les réseaux de neurones, eu égard à leur qualité d'approximateurs universels, peuvent aussi bien modéliser un ensemble de données que décrire le comportement non linéaire d'un système dynamique. C'est pourquoi, l'objectif principal du présent chapitre sera d'appliquer nos résultats précédents pour l'analyse de certaines oscillations localisées (ondes solitaires) associées à différents modèles de plasma. Mais, tout d'abord définissons de manière succincte l'état plasma.

Il est communément admis que 99% de la matière de l'univers est à l'état plasma, milieu composé de particules neutres, d'ions et d'électrons[62]. Un plasma se définit donc comme étant un gaz électrifié dont les atomes sont dissociés en ions à charge positive et en électrons à charge négative[63]. Les atmosphères, les nébuleuses et la majeure partie de l'hydrogène interstellaire sont à l'état plasma. Notre planète est elle même entourée d'un plasma à une distance comprise entre 90 et 500 km de sa surface. D'un autre côté, dans notre vie quotidienne, le nombre d'exemples d'états plasma est quelque peu limité et restreint. Nous citerons, à titre d'exemple, les gaz conducteurs à l'intérieur des tubes fluorescents tels les tubes à néon, les propulseurs spatiaux et les plasmas couramment utilisés dans l'industrie notamment en micro-électronique. Il est utile de noter que n'importe quel gaz ionisé ne se trouve pas forcément à l'état plasma. Un gaz présente toujours un certain degré d'ionisation. Ce n'est qu'au moment où la concentration des particules chargées provoque une charge d'espace capable de limiter le libre parcours moyen des ions et des électrons que les caractéristiques du gaz changent de manière significative[64]. Un plasma se définit comme étant un gaz quasi-neutre composé de particules chargées et de particules neutres qui exhibent un comportement collectif. Lorsque les particules chargées se meuvent dans un plasma, des concentrations locales de charge positive et de charge négative sont générées au sein de ce plasma. Ces concentrations, et donc ces charges d'espace, donnent naissance à des champs électriques. En outre, le mouvement des particules chargées génère des courants et crée, par conséquent, des champs électromagnétiques. Ces champs électromagnétiques

affectent et perturbent le mouvement d'autres particules chargées qui à leur tour peuvent influencer le reste des particules du plasma. En fait, toute perturbation affectant la neutralité d'un plasma provoque de forts champs électromagnétiques qui tendent à la restaurer. La façon la plus naturelle d'obtenir un plasma est de chauffer un gaz jusqu'à ce que l'énergie moyenne des particules soit comparable à l'énergie d'ionisation de l'espèce considérée. Dans un plasma, il existe de nombreux processus à travers lesquels les modes instables peuvent se saturer et atteindre de grandes amplitudes. Lorsque ces amplitudes sont suffisamment grandes, les non linéarités ne peuvent plus être négligées[65]. De manière générale, les non linéarités proviennent de divers phénomènes tels la force de Lorentz, le piégeage des particules dans le potentiel de l'onde,...etc. Dans un plasma, ces non linéarités contribuent à la localisation des ondes donnant naissance à différentes structures non linéaires telles les ondes solitaires, les ondes de choc, les double couches, les vortex,...etc. Un soliton résulte de l'équilibre de deux effets: la non linéarité et la dispersion. Cependant, quoique la plupart des ondes dans les plasmas présentent de la dispersion et que le plasma lui-même se comporte comme un milieu non linéaire, seulement un nombre restreint d'ondes est connu pour admettre des solutions solitaires. Les ondes acoustiques ioniques exhibent de telles solutions localisées et ont été largement étudiées tant sur le plan théorique qu'expérimentale. D'ailleurs la redécouverte de l'équation de Korteweg de Vries pour ce type d'onde dans les plasmas a été sans nul doute le premier pas pour démêler quelque peu le domaine très ardu des phénomènes non linéaires.

4.2 Onde acoustique poussiéreuse solitaire

4.2.1 Analyse des conditions d'existence par la méthode du pseudo-potentiel

Récemment, Mamun et *al.*[66] ont considéré un plasma composé de grains d'impureté à charge négative, $q_d = -Z_d e$, et d'ions positifs de température T_i , distribués selon la loi de Maxwell-Boltzmann. A l'équilibre, $n_{i0} = Z_d n_{d0}$ où n_{i0} et n_{d0} désignent respectivement

les densités numériques non perturbées des ions et des grains de poussière. Les oscillations acoustiques de faible vitesse de phase associées à un tel plasma sont alors gouvernées par les équations fluides normalisées suivantes

$$\frac{\partial N_d}{\partial T} + \frac{\partial(N_d V_d)}{\partial X} = 0 \quad (4.2.1)$$

$$\frac{\partial V_d}{\partial T} + V_d \frac{\partial V_d}{\partial X} = \frac{\partial \Psi}{\partial X} \quad (4.2.2)$$

$$\frac{\partial^2 \Psi}{\partial X^2} = N_d - \exp(-\Psi) \quad (4.2.3)$$

La densité des grains N_d , leur vitesse fluide V_d , le potentiel électrostatique Ψ , le temps T et la variable d'espace X sont normalisés par n_{d0} (valeur à l'équilibre), $C_d = (Z_d T_i / m_d)^{1/2}$ (la vitesse acoustique poussiéreuse), T_i / e , $\omega_{pd}^{-1} = (m_d / 4\pi n_{d0} Z_d^2 e^2)^{1/2}$ (la période plasma poussiéreuse) et $\lambda_{pd} = (T_i / 4\pi n_{d0} Z_d e^2)^{1/2}$ (la longueur de Debye), respectivement. Dans le but d'étudier les propriétés de l'onde acoustique poussiéreuse de large amplitude, les équations (4.2.1)-(4.2.3) sont habituellement réécrites en introduisant la variable $\xi = X - MT$, où M est appelé nombre de Mach (vitesse du soliton / C_d). En imposant des conditions aux limites appropriées aux solutions localisées ($\Psi \rightarrow 0$, $V_d \rightarrow 0$, $N_d \rightarrow 1$ lorsque $\xi \rightarrow \pm\infty$), nous obtenons à partir de (4.2.1) et (4.2.2) la relation

$$N_d = \frac{M}{\sqrt{M^2 + 2\Psi}} \quad (4.2.4)$$

En portant la relation (4.2.4) dans l'équation de Poisson (4.2.3), en multipliant chaque membre de l'équation résultante par $(d\Psi/d\xi)$, en intégrant une fois et en imposant les conditions aux limites appropriées aux solutions localisées ($\Psi \rightarrow 0$, $d\Psi/d\xi \rightarrow 0$ lorsque $\xi \rightarrow \pm\infty$), nous obtenons la quadrature

$$\frac{1}{2} \left(\frac{d\Psi}{d\xi} \right)^2 + V(\Psi) = 0 \quad (4.2.5)$$

où

$$V(\Psi) = 1 + M^2 - M^2(1 + 2\Psi/M^2)^{1/2} - \exp(-\Psi) \quad (4.2.6)$$

représente le potentiel de Sagdeev[67] ou pseudo-potentiel. L'équation (4.2.5) peut être perçue comme étant l'équation du mouvement d'une pseudo-particule de masse unité, de

position Ψ et de vitesse $(d\Psi/d\xi)$, oscillant dans un potentiel $V(\Psi)$. Il est aisé de vérifier à partir de l'équation (4.2.6) que $V(\Psi)$ et $dV(\Psi)/d\Psi$ sont nuls en $\Psi = 0$. Les conditions d'existence d'une solution en onde solitaire de l'équation (4.2.5) peuvent alors être énoncées comme suit : (i) $[d^2V(\Psi)/d\Psi^2]_{\Psi=0} < 0$ (on dit alors que le point fixe à l'origine est instable), (ii) Ψ admet un extremum non nul noté Ψ_m tel que $V(\Psi_m) = 0$, (iii) $V(\Psi) < 0$ lorsque Ψ varie de 0 à Ψ_m . L'analyse numérique de (4.2.6) pour différentes valeurs de M montre que le potentiel de Sagdeev ne pourra jamais s'annuler pour toute valeur de $\Psi > 0$ et $M > 1$, écartant ainsi la possibilité d'existence des solitons acoustiques poussiéreux compressifs ($\Psi > 0$). Cependant, une analyse similaire indique que des solitons de raréfaction à potentiel négatif ($\Psi < 0$) existent. Des ondes acoustiques poussiéreuses solitaires d'amplitude finie existent pour $1 < M < 1.58$.

4.2.2 Analyse des conditions d'existence à l'aide d'un réseau de neurones

Supposons maintenant que le système dynamique précédent ne soit pas intégrable, c'est à dire qu'il n'est pas possible de trouver l'expression analytique explicite du pseudo-potentiel $V(\Psi)$ (4.2.6). Approchons alors le membre de droite de l'équation de Poisson (4.2.3) en l'écrivant sous la forme

$$\frac{d^2\Psi}{d\xi^2} = \frac{M}{\sqrt{M^2 + 2\Psi}} - \exp(-\Psi) \simeq \sum_{i=1}^{N=5} \omega_i f_i \quad (4.2.7)$$

où

$$f_i = f(\Psi/\lambda_i) = \left(1 - \frac{\Psi^2}{\lambda_i^2}\right) \exp\left(-\frac{\Psi^2}{2\lambda_i^2}\right) \quad (4.2.8)$$

représente l'ondelette "Chapeau-Mexicain" et la sommation porte sur le nombre de neurones. ω_i et λ_i désignent, respectivement, le poids et le paramètre échelle dont les valeurs sont obtenues grâce au schéma d'apprentissage construit précédemment. Sachant que

$$\frac{d^2\Psi}{d\xi^2} = -\frac{dV}{d\Psi} \quad (4.2.9)$$

nous obtenons aisément l'expression approchée de $V(\Psi)$ suivante

$$V(\Psi) \simeq - \sum_{i=1}^{N=5} \omega_i \Psi \exp(-\Psi^2/2\lambda_i^2) \quad (4.2.10)$$

Dans le but d'obtenir la meilleure approximation possible et de voir dans quelle mesure celle-ci dépend du domaine d'approximation, nous avons choisi d'entreprendre un apprentissage sur trois domaines proches mais quelque peu différents, à savoir, $\Psi = [-1.1, 0]$, $[-1.12, 0]$ et $[-1.124, 0]$. Pour amorcer le processus d'apprentissage, les valeurs des paramètres d'échelle sont initialisées à 1, 2, 5, 8 et 20. Nos résultats sont alors résumés dans les trois tableaux suivants:

Intervalle 1, $\Psi = [-1.1, 0]$

i	1	2	3	4	5
ω_i	1117.39	1097.95	-5361.85	-970.38	4117.03
λ_i	3.36277	3.36377	5.34902	8.00071	20.0074

Intervalle 2, $\Psi = [-1.12, 0]$

i	1	2	3	4	5
ω_i	888.593	873.311	-4239.89	-779.509	3257.68
λ_i	3.37317	3.37417	5.35283	8.00076	20.0075

Intervalle 3, $\Psi = [-1.124, 0]$

i	1	2	3	4	5
ω_i	886.965	873.357	-3974.27	-862.61	3076.75
λ_i	3.4936	3.4946	5.40501	8.0007	20.0089

et représentés sur la figure 36.

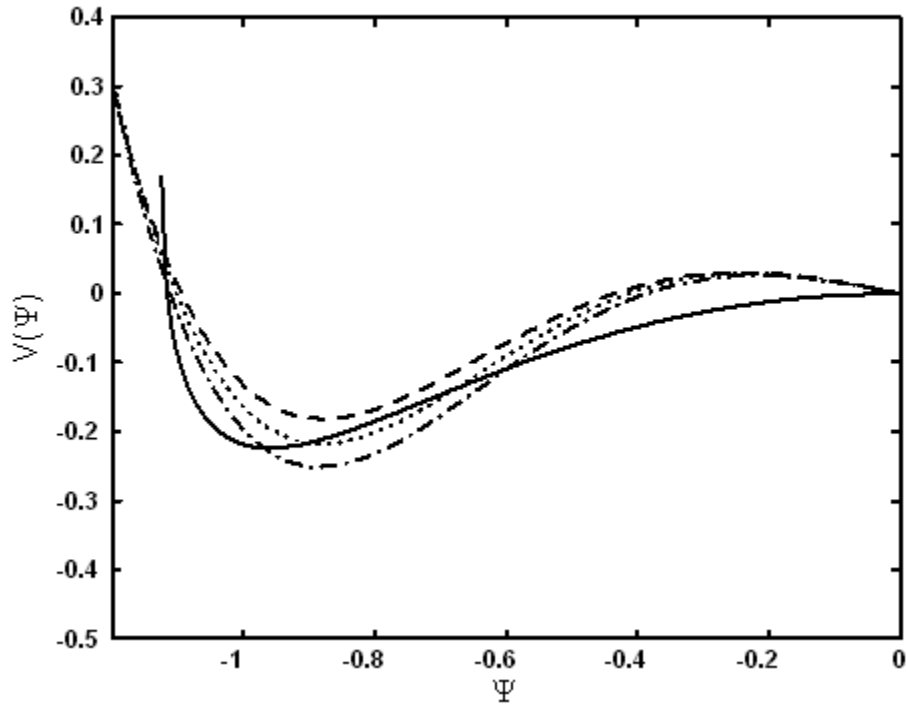


Figure 36: Pseudo-potential associé à l'onde acoustique poussiéreuse avec $M = 1.5$. Trait plein: expression exacte. Tirets: intervalle 1. Pointillés: intervalle 2. Tirets-pointillés: intervalle 3.

Celle-ci montre les profils du pseudo-potential approché et celui exact tracé à partir de l'expression (4.2.6). L'accord est quelque peu remarquable dans la mesure où les résultats de l'intervalle 2 (pointillés) prévoient l'existence d'une structure solitaire qui possède les mêmes propriétés intrinsèques que celle associée au pseudo-potential exact dont l'expression est donnée par la relation (4.2.6): même profondeur du puits et même point d'intersection avec l'axe des potentiels Ψ . Rappelons que la profondeur du puits et le point d'intersection du pseudo-potential avec l'axe- Ψ traduisent, respectivement, la largeur de la structure solitaire ainsi que son amplitude. Par commodité, nous reproduisons, sur la figure 37, les profils du pseudo-potential exact ainsi que de celui que l'on peut construire à partir des résultats de l'approximation associée à l'intervalle 2.

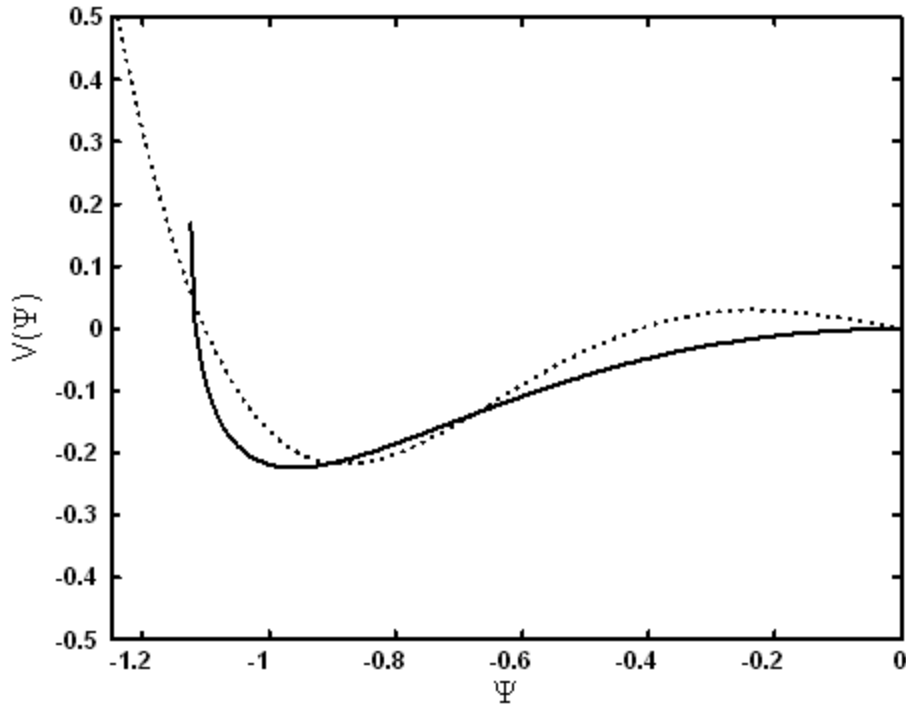


Figure 37: Pseudo-potential associé à l'onde acoustique poussiéreuse avec $M = 1.5$. Trait plein: expression exacte. Pointillés: intervalle 2.

L'intérêt d'une telle approximation est à rechercher du côté expérimental. Rappelons que lors des expériences sur les plasmas de laboratoire, l'opérateur, à travers une série de mesures de sonde, n'a accès, généralement, qu'aux différentes températures des espèces de particules présentes dans le plasma ainsi qu'aux profils de leurs densités respectives. Il devient alors possible, en procédant de la même manière que précédemment, de prédire l'existence ou non d'une structure solitaire lorsque les paramètres du plasma sont, à priori, connus.

4.3 Onde solitaire dans un plasma complexe à charge variable

Le modèle de plasma précédent appartient à la classe des systèmes intégrables dans la mesure où les expressions analytiques de leurs pseudo-potentiels peuvent être obtenues de manière exacte. Cette classe englobe, entre autres exemples, l'équation classique de Korteweg-de

Vries (K-dV), l'équation de Korteweg-de Vries modifiée (mK-dV) ainsi que l'équation non linéaire de Schrödinger (NLSE). Il n'en est plus de même lorsqu'il s'agit, par exemple, de l'équation K-dV à coefficients variables et, dans ce cas là, un traitement perturbatif, un développement en séries de puissances ou encore une intégration numérique s'avèrent, parfois, plus que nécessaires. Les chercheurs ont de tout temps redoublé d'efforts dans leur quête de rendre intégrables des systèmes dynamiques qui ne le sont pas et le sujet reste toujours d'actualité. Ce problème est, en quelque sorte, l'analogue de celui inhérent aux systèmes hamiltoniens (et non hamiltoniens) ou encore, à celui relatif aux systèmes à comportement chaotique (et régulier). Notre méthode d'approximation peut alors apparaître, à bien des égards, comme une alternative pour la description de la dynamique d'un système non intégrable. Pour la mettre en application, nous emprunterons à la physique des plasmas un modèle de plasma non intégrable en nous inspirant des travaux que nous avons menés parallèlement à ceux sur l'approximation, les ondelettes et les réseaux de neurones. On se propose, pour cela, de construire le pseudo-potentiel approché, associé à une onde solitaire dans un plasma complexe contenant des grains d'impureté à charge variable dont une fraction est piégée dans les gradients du potentiel de l'onde solitaire. Mais que veut dire plasma complexe?

A ce propos, rappelons que la définition d'un plasma à deux composantes (électrons et ions) telle qu'énoncée au début de ce chapitre peut s'avérer, pour plusieurs raisons, incomplète et restrictive. En effet, un plasma réel contient toujours une composante additionnelle d'impuretés chargées communément appelées poussières[68]. La présence de ces extra particules chargées positivement ou négativement (selon les conditions de leur environnement) et de taille micrométrique ou sub-micrométrique rend la nature du plasma beaucoup plus complexe. Un plasma complexe est un gaz de basse température, complètement ou partiellement ionisé comprenant des électrons, des ions et une composante additionnelle de grains de poussière chargés et extrêmement massifs. Les plasmas complexes sont aussi bien présents dans le milieu interplanétaire que dans les queues et chevelures des comètes, les anneaux planétaires, les boucles solaires et les nuages moléculaires interstellaires. La

présence de grains de poussière d'origine météoritique a d'ailleurs été détectée dans la mésosphère terrestre à des altitudes allant de 80 à 100 km grâce à des mesures directes de sonde. Au vu des nouveaux phénomènes qu'ils introduisent et font intervenir, tels que la fluctuation de la charge électrique, l'appauvrissement électronique et la dissipation anormale de l'énergie, ces grains de poussière massifs et hautement chargés peuvent modifier les propriétés intrinsèques du plasma traditionnel à deux composantes. Un grain d'impureté immergé dans un plasma acquiert une charge électrique qui peut valoir plusieurs milliers de fois celle de l'électron. La charge du grain de poussière provient généralement d'un concours de plusieurs processus physiques telles que la collection des électrons et des ions du plasma ambiant, l'émission photo-électronique dans les milieux radioactifs, l'émission électronique secondaire, l'émission par ions énergétique,...etc. Dans les plasmas de laboratoire de basses températures, la collection des électrons et des ions semble être le mécanisme de charge le plus dominant.

4.3.1 Modèle théorique et équations de base

Considérons un plasma non magnétisé composé d'électrons, d'ions à charge positive et de grains d'impuretés (ou poussières) de densités n_e , n_i , et n_d , respectivement. Les grains de poussière sont supposés être des sphères de rayon r , de même masse m_d et de même charge électrique négative $q_d = -eZ$, où Z représente le nombre de charge. Sur une échelle temporelle caractéristique du mouvement des grains de poussière (massifs et relativement lents), les électrons et les ions peuvent être supposés en équilibre thermodynamique (distribués selon la loi de Maxwell-Boltzmann). Leurs densités numériques (nombre de particules par unité de volume) sont alors données par[63]

$$n_j = n_{j0} \exp\left(-\frac{q_j \phi}{T_j}\right) \quad (4.3.1)$$

où $j = e, i$ dénote, respectivement, les électrons et les ions, ϕ représente le potentiel électrostatique, $q_{j=e,i} = \mp e$ les charges électriques et T_j les températures. L'indice 0 indique des quantités à l'équilibre ($\phi = 0$). Les grains de poussière, dont une partie est piégée dans

le potentiel de l'onde, sont modélisés grâce à la fonction de distribution, dite de Schamel, suivante[69]

$$f_d = f_{df} + f_{dt} \quad (4.3.2)$$

où

$$f_{df}(x, v_d) = n_{d0} \left(\frac{m_d}{2\pi T_{df}} \right)^{1/2} \exp \left(-\frac{m_d v_d^2 / 2 - e \int_0^\phi Z d\phi}{T_{df}} \right) \text{ pour } |v_d| \succ \sqrt{\frac{2e \int_0^\phi Z d\phi}{m_d}} \quad (4.3.3)$$

$$f_{dt}(x, v_d) = n_{d0} \left(\frac{m_d}{2\pi T_{df}} \right)^{1/2} \exp \left(-\beta \frac{m_d v_d^2 / 2 - e \int_0^\phi Z d\phi}{T_{df}} \right) \text{ pour } |v_d| \preceq \sqrt{\frac{2e \int_0^\phi Z d\phi}{m_d}} \quad (4.3.4)$$

L'indice $f(t)$ représente la contribution des grains de poussière libres (piégés). β est un paramètre déterminant le nombre de grains de poussière piégés et sa valeur est définie comme étant le rapport de la température des grains libres T_{df} sur celle des grains piégés T_{dt} , i.e., $|\beta| = T_{df}/T_{dt}$. Le paramètre de piégeage β permet l'introduction de différentes températures dans la fonction de distribution des vitesses du grain de poussière, $\beta = 0$ (ou 1) correspondant à une fonction de distribution en forme de plateau (ou Maxwellienne). Notons qu'un trou dans la région de piégeage, correspondant à un dépeuplement ou appauvrissement en particules piégées, est représenté par les valeurs négatives de β (fonction de distribution de type vortex). En intégrant la fonction de distribution (4.3.2) précédente sur tout l'espace des vitesses, nous obtenons pour $\beta < 0$

$$n_d(\phi) = \int_{-\infty}^{+\infty} f_d(x, v_d) dv_d = n_{d0} \left\{ \exp \left(-\frac{\chi}{T_{df}} \right) \operatorname{erf} c \left(\sqrt{-\frac{\chi}{T_{df}}} \right) + \frac{2}{\sqrt{-\pi\beta}} \operatorname{W} \left(\sqrt{\beta \frac{\chi}{T_{df}}} \right) \right\} \quad (4.3.5)$$

où $\chi = -e \int_0^\phi Z d\phi$ et $\operatorname{W}(x)$ est l'intégrale de Dawson définie par $\operatorname{W}(x) = \exp(-x^2) \int_0^x \exp(t^2) dt$.

L'équation de Poisson peut alors s'écrire sous la forme (pour $\beta < 0$)

$$\frac{d^2\Psi}{dX^2} = \exp(\Psi) - f \exp(-\sigma\Psi) + \frac{(f-1)}{Q_{d0}} Q_d N_d \quad (4.3.6)$$

où

$$N_d = n_d/n_{d0} = \exp(-\gamma\xi) \operatorname{erf} c \left(\sqrt{-\gamma\xi} \right) + \frac{2}{\sqrt{-\pi\beta}} \operatorname{W} \left(\sqrt{\beta\gamma\xi} \right) \quad (4.3.7)$$

$$\frac{d\xi}{dX} = Q_d \frac{d\Psi}{dX} \quad (4.3.8)$$

$f = n_{i0}/n_{e0}$, $\sigma = T_e/T_i$ et $\gamma = rT_e^2/e^2T_{df}$. Le système de variables adimensionnelles suivant

$$\Psi = \frac{e\phi}{T_e}, \xi = \frac{e^2}{rT_e^2}\chi, X = \frac{x}{\lambda_{De}}, Q_d = \frac{eq_d}{rT_e} \quad (4.3.9)$$

a été introduit et $\lambda_{De} = (T_e/4\pi n_{e0}e^2)^{1/2}$ représente la longueur de Debye électronique. La condition de quasi-neutralité de la charge électrique requiert $n_{e0} - n_{i0} + Z_0 n_{d0} = 0$.

Dans le modèle standard de la sonde électrostatique appliqué au grain de poussière[70], la charge électrique du grain provient essentiellement de la collecte des électrons et des ions du plasma de base. Par conséquent, la charge du grain poussière $q_d = -eZ$ est déterminée de manière self-consistante par

$$v_d \frac{dq_d}{dx} = I_e + I_i \quad (4.3.10)$$

En réalité, l'équation précédente traduit le principe de conservation de la charge électrique. I_e et I_i représentent, respectivement, les courants de charge électronique et ionique et sont définis par[70]

$$I_j = \pi r^2 q_j (8T_j/\pi m_j)^{1/2} n_j W_j \quad (4.3.11)$$

où $r \ll \lambda_e$ représente le rayon du grain de poussière et $m_{j=i,e}$ les masses. Pour $q_d < 0$, nous avons[71, 72]

$$W_e = \exp(eq_d/rT_e) \quad (4.3.12)$$

et

$$W_i = 1 - eq_d/rT_i \quad (4.3.13)$$

Les grains de poussière sont initialement accélérés à une vitesse v_{d0} grâce à un faible champ électrique. Par conséquent, nous obtenons à partir de l'équation de conservation du nombre de particules, après intégration, la relation

$$n_d v_d = n_{d0} v_{d0} \quad (4.3.14)$$

En réarrangeant les termes de l'équation (4.3.10), nous obtenons l'équation de la charge adimensionnelle suivante

$$\frac{dQ_d}{dX} = kN_d \left\{ -\exp(\Psi + Q_d) + \sqrt{\frac{\mu}{\sigma}} f \exp(-\sigma\Psi) (1 - \sigma Q_d) \right\} \quad (4.3.15)$$

où

$$k = \sqrt{\frac{2n_{e0}e^2r^2}{m_e v_{d0}^2}} \quad (4.3.16)$$

et $\mu = m_e/m_i$. L'équation (4.3.15) est une équation dynamique additionnelle couplée aux autres équations du plasma à travers les courants de charge. La charge électrique du grain de poussière devient alors une nouvelle variable dynamique couplée de manière self-consistante aux autres variables dynamiques du plasma telles la densité numérique des particules et le potentiel électrostatique. A l'équilibre et en l'absence de toute perturbation ($\Psi = 0$), l'équation (4.3.15) requiert

$$f = \sqrt{\frac{\sigma \exp(Q_{d0})}{\mu (1 - \sigma Q_{d0})}} \quad (4.3.17)$$

Dans la simulation numérique qui va suivre, la valeur de f est déduite à partir de la relation (4.3.17) ci-dessus, tandis que les valeurs des autres paramètres sont préalablement définies.

4.3.2 Analyse des conditions d'existence par la méthode du pseudo-potentiel

En multipliant chaque membre de l'équation de Poisson (4.3.6) par $d\Psi/dX$ et en intégrant une fois, nous obtenons la quadrature suivante

$$\frac{1}{2} \left(\frac{d\Psi}{dX} \right)^2 + V(\Psi) = 0 \quad (4.3.18)$$

où $V(\Psi)$ représente le pseudo-potentiel ou potentiel de Sagdeev[67] et est donné par

$$V(\Psi) = -\exp(\Psi) - \frac{f}{\sigma} \exp(-\sigma\Psi) - \frac{(f-1)}{Q_{d0}} \int_0^\Psi Q_d N_d d\Psi + 1 + \frac{f}{\sigma} \quad (4.3.19)$$

Le temps caractéristique du mouvement du grain de poussière est de l'ordre de dizaines de millisecondes alors que son temps de charge est typiquement de l'ordre de 10^{-8} s. Le déplacement du grain de poussière est donc négligeable devant l'échelle spatiale du problème. Par conséquent, le processus de charge peut être traité comme un phénomène local et le terme convectif de l'équation (4.3.15) peut être négligé. Nous obtenons alors la relation

$$I_e + I_i \simeq 0 \quad (4.3.20)$$

à partir de laquelle nous établissons l'expression de la charge du grain suivante

$$Q_d = -\text{Lambert} \left\{ \frac{\exp [\Psi (1 + \sigma) + 1/\sigma]}{\sqrt{\mu\sigma}f} \right\} + \frac{1}{\sigma} \quad (4.3.21)$$

La fonction de Lambert[73], appelée aussi fonction omega, est la fonction inverse de $y = xe^x$ et certains problèmes bien connus en électrostatique et en mécanique quantique peuvent être traités avec une grande aisance en l'utilisant. Lambert(x) est réelle pour $x \geq -1/e$ et il est aisé de vérifier que l'argument $\frac{\exp[\Psi(1+\sigma)+1/\sigma]}{\sqrt{\mu\sigma}f}$ [voir équation (4.3.21)] est toujours positif. L'énergie potentielle ξ est alors donnée par

$$\begin{aligned} \xi &= \int_0^\Psi Q_d d\Psi = \\ &= \frac{1}{2(1+\sigma)} \left\{ \text{Lambert}^2 [\exp (1/\sigma) / \sqrt{\mu\sigma}f] - \text{Lambert}^2 [\exp (\sigma\Psi + \Psi + 1/\sigma) / \sqrt{\mu\sigma}f] \right\} \\ &+ \frac{1}{(1+\sigma)} \left\{ \text{Lambert} [\exp (1/\sigma) / \sqrt{\mu\sigma}f] - \text{Lambert} [\exp (\sigma\Psi + \Psi + 1/\sigma) / \sqrt{\mu\sigma}f] \right\} + \frac{\Psi}{\sigma} \end{aligned} \quad (4.3.22)$$

A l'instar de la section précédente, on se propose maintenant d'approcher le terme

$$\begin{aligned} Q_d N_d &= \left[-\text{Lambert} \left\{ \frac{\exp [\Psi (1 + \sigma) + 1/\sigma]}{\sqrt{\mu\sigma}f} \right\} + \frac{1}{\sigma} \right] \\ &\left[\exp (-\gamma\xi) \operatorname{erf} c \left(\sqrt{-\gamma\xi} \right) + \frac{2}{\sqrt{-\pi\beta}} \text{W} \left(\sqrt{\beta\gamma\xi} \right) \right] \end{aligned} \quad (4.3.23)$$

en l'écrivant sous la forme

$$\begin{aligned} Q_d N_d &= \left[-\text{Lambert} \left\{ \frac{\exp [\Psi (1 + \sigma) + 1/\sigma]}{\sqrt{\mu\sigma}f} \right\} + \frac{1}{\sigma} \right] \\ &\left[\exp (-\gamma\xi) \operatorname{erf} c \left(\sqrt{-\gamma\xi} \right) + \frac{2}{\sqrt{-\pi\beta}} \text{W} \left(\sqrt{\beta\gamma\xi} \right) \right] \\ &= \sum_{i=1}^{N=5} \omega_i f_i \end{aligned} \quad (4.3.24)$$

où

$$f_i = f(\Psi/\lambda_i) = \frac{1}{1 + \exp(-\Psi/\lambda_i)} \quad (4.3.25)$$

représente la fonction Sigmoidale et la sommation porte sur le nombre de neurones. ω_i et λ_i désignent, respectivement, le poids et le paramètre d'échelle dont les valeurs sont obtenues

grâce au schéma d'apprentissage construit précédemment. Nous obtenons alors

$$\begin{aligned}
 V(\Psi) &= -\exp(\Psi) - \frac{f}{\sigma} \exp(-\sigma\Psi) - \frac{(f-1)}{Q_{d0}} \sum_{i=1}^{N=5} \omega_i \int_0^\Psi \frac{d\Psi}{1 + \exp(-\Psi/\lambda_i)} + 1 + \frac{f}{\sigma} \\
 &= -\exp(\Psi) - \frac{f}{\sigma} \exp(-\sigma\Psi) - \frac{(f-1)}{Q_{d0}} \sum_{i=1}^{N=5} \omega_i \lambda_i \log \left[\frac{1 + \exp(-\Psi/\lambda_i)}{2 \exp(-\Psi/\lambda_i)} \right] + \\
 &\quad 1 + \frac{f}{\sigma}
 \end{aligned} \tag{4.3.26}$$

Pour amorcer le processus d'apprentissage, les valeurs des paramètres d'échelle sont initialisées à 2, 200, 200, 40000 et 4000. Nos résultats sont alors résumés dans le tableau suivant

i	1	2	3	4	5
ω_i	-0.6753108	10.8819398	10.88178957	-12.07508619	-12.07508619
λ_i	$1.99999 \cdot 10^{-5}$	$200 \cdot 10^{-5}$	$200.001 \cdot 10^{-5}$	$22 \cdot 10^{-2}$	$22 \cdot 10^{-2}$

et tracés sur la figure 38. L'accord est plus que remarquable dans la mesure où V_{app} prévoit l'existence d'une structure solitaire (figure 39) dont l'amplitude et la largeur sont quasi-similaires à celles de la structure solitaire que nous obtenons par intégration numérique de l'équation (4.3.6) et en faisant appel aux relations (4.3.7), (4.3.21) et (4.3.22) (voir figure 40). Il serait peut être utile de noter que le graphe en tirets (voir figure 40) a été obtenu en implémentant directement l'expression approchée (4.3.24) dans le programme numérique qui nous a permis de résoudre l'équation de Poisson (4.3.6). Par ailleurs, la même approche peut être utilisée pour différentes valeurs des paramètres de notre modèle de plasma. Notons enfin qu'il est toujours possible de procéder de la même manière que précédemment pour étudier d'autres systèmes non intégrables.

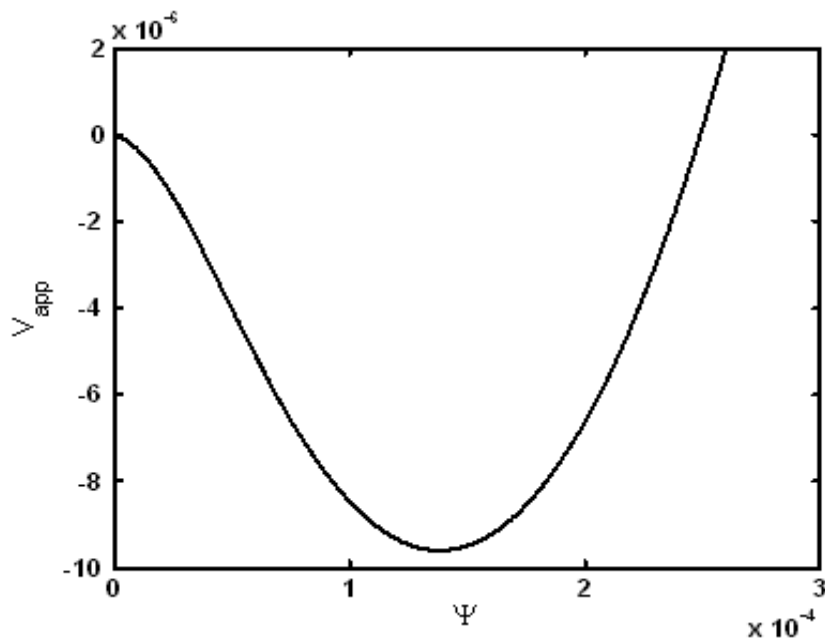


Figure 38: Pseudo-potential approximated associated with the solitary wave with $\beta_d = -0.9$, $Q_{d0} = -1.5$, $T_e = 1\text{eV}$, $\sigma = 10$, $T_{df} = T_i/10$ et $r_d = 0.1\mu\text{m}$.

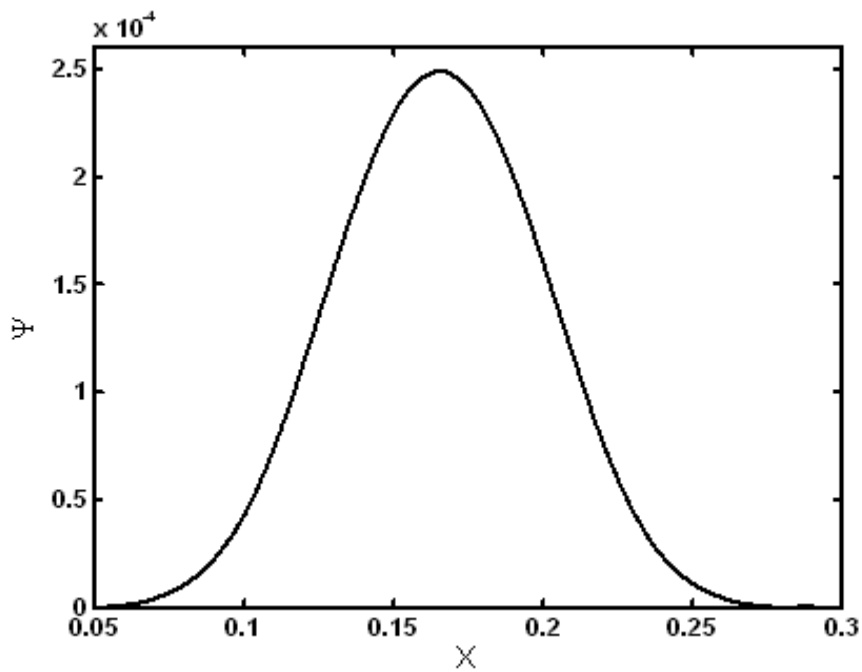


Figure 39: Structure solitaire obtenue à partir du pseudo-potential approximated.

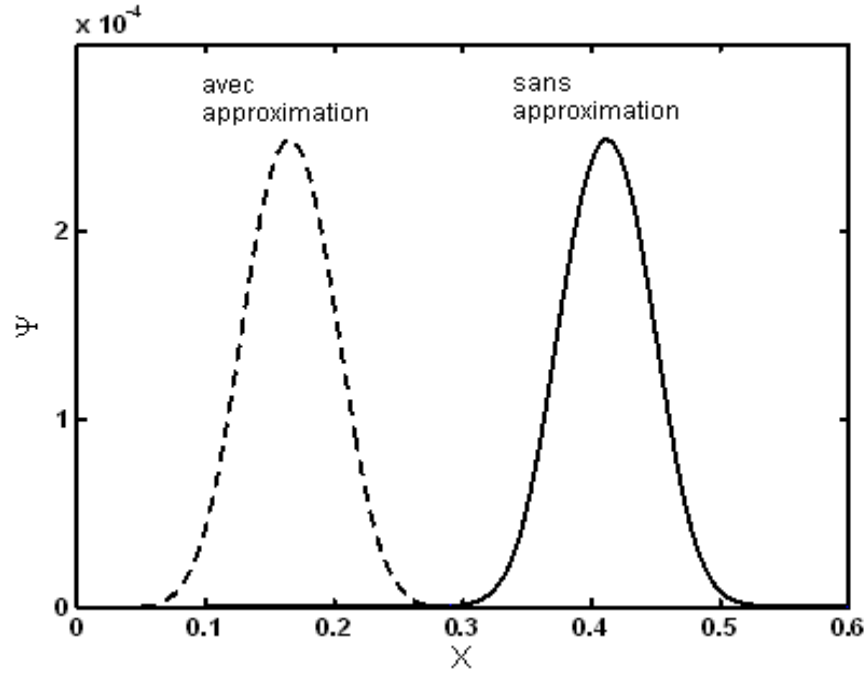


Figure 40: Structure solitaire obtenue à l'aide de l'intégration numérique des équations de base sans approximation (trait plein) et celle obtenue en utilisant l'approximation (tirets).

Intéressons nous maintenant au cas des faibles amplitudes. A partir de l'équation (4.3.20), il est possible d'établir la relation transcendantale suivante

$$(1 + \sigma) \Psi = \ln \left\{ \frac{1 - \sigma Q_d}{1 - \sigma Q_{d0}} \exp(Q_{d0} - Q_d) \right\} \quad (4.3.27)$$

En négligeant les termes d'ordre $\theta(\sigma^{-1})$, la relation précédente devient

$$\sigma \Psi = Q_{d0} - Q_d + \ln \frac{Q_d}{Q_{d0}} \quad (4.3.28)$$

En supposant que la variation relative de la charge du grain de poussière est suffisamment petite

$$\ln \frac{Q_d}{Q_{d0}} \simeq (Q_d - Q_{d0}) / Q_{d0} \quad (4.3.29)$$

nous obtenons

$$\frac{Q_d}{Q_{d0}} = 1 - \frac{\sigma \Psi}{Q_{d0} - 1} \quad (4.3.30)$$

En utilisant les équations suivantes

$$Q_d N_d = Q_{d0} \left(1 - \frac{\sigma \Psi}{Q_{d0} - 1} \right) \left\{ \exp(-\gamma \xi) \operatorname{erfc}(\sqrt{-\gamma \xi}) + \frac{2}{\sqrt{-\pi \beta}} W(\sqrt{\beta \gamma \xi}) \right\} \quad (4.3.31)$$

$$\xi = \int_0^\Psi Q_d d\Psi = Q_{d0} \left[\Psi - \frac{\sigma \Psi^2}{2(Q_{d0} - 1)} \right] \quad (4.3.32)$$

$$N_d = 1 - \gamma Q_{d0} \Psi + \frac{4(1-\beta)}{3\sqrt{\pi}} \gamma Q_{d0} \sqrt{-\gamma Q_{d0}} \Psi^{3/2} + \frac{\gamma Q_{d0}}{2} \left(\frac{\sigma}{Q_{d0} - 1} + \gamma Q_{d0} \right) \Psi^2 + 0(\Psi^{5/2}) \quad (4.3.33)$$

il devient alors possible d'établir l'expression, en faible amplitude, du pseudo-potentiel $V(\Psi)$ suivante

$$V(\Psi) = A_1 \Psi^2 + A_2 \Psi^{5/2} + 0(\Psi^3) \quad (4.3.34)$$

où

$$\begin{aligned} A_1 &= \frac{1}{2} \left\{ (f-1) \gamma Q_{d0} + \frac{(f-1)\sigma}{Q_{d0}-1} - 1 - f\sigma \right\} \\ A_2 &= \frac{8(\beta-1)}{15\sqrt{\pi}} (f-1) \gamma Q_{d0} \sqrt{-\gamma Q_{d0}} \end{aligned} \quad (4.3.35)$$

La solution de l'équation

$$\frac{1}{2} \left(\frac{d\Psi}{dX} \right)^2 + A_1 \Psi^2 + A_2 \Psi^{5/2} = 0 \quad (4.3.36)$$

est alors donnée par

$$\Psi(X) = \left(2 \frac{A_1}{A_2} \right)^2 \operatorname{sech}^4 \left[\sqrt{-\frac{A_1}{8}} X \right] \quad (4.3.37)$$

Celle-ci représente une solution solitaire de faible amplitude à condition que

$$\frac{f-1}{1+f\sigma} \left(\gamma Q_{d0} + \frac{\sigma}{Q_{d0}-1} \right) < 1 \quad (4.3.38)$$

La solution solitaire (4.3.37) diffère du soliton, dit de Korteweg-de Vries, suivant

$$\Psi(X) = \Psi_M \operatorname{sech}^2 \left[\sqrt{\frac{1}{6}} \Psi_M X \right] \quad (4.3.39)$$

de par la puissance et l'argument de la fonction Sech. Dans le but de mieux asseoir et étayer les différences entre les solutions solitaires (4.3.37) et (4.3.39), nous avons décidé de procéder à un apprentissage en faisant appel au réseau de neurones déjà établi[18]. Les figures 41 et 42 illustrent l'évolution de $\operatorname{Log}_2(\lambda_i)$ pour (4.3.39) (figure 41) et (4.3.37) (figure 42). En l'absence des particules piégées (figure 41), l'évolution des paramètres d'échelle λ_i

exhibe un comportement très intéressant. Une tendance à la fusion, similaire à celle déjà rapportée dans les chapitres précédents, entre λ_1 , λ_2 , et λ_3 est alors observée. Cette tendance (au merging) disparaît en présence des particules piégées comme l'indique la figure 42 où les λ_i , au terme d'un accroissement itératif, se maintiennent à des valeurs constantes. Ce phénomène peut alors être attribué à la saturation produite, de concert, par les processus de charge et de piégeage du grain de poussière. On peut alors établir une certaine analogie avec le travail de Buti[74] qui a montré qu'une petite fraction de grains de poussière chargés peut éliminer le chaos généralement observé dans les systèmes Alfvéniques non dissipatifs. Expérimentalement (où un ensemble de données est générée), l'existence ou non du phénomène de fusion peut, dans une certaine mesure, nous renseigner sur l'existence ou non des particules piégées résonantes. Rappelons que des fonctions de distribution électronique en forme de plateau, caractéristiques de la présence des électrons piégés, sont généralement observées dans les expériences portant sur l'onde acoustique ionique.

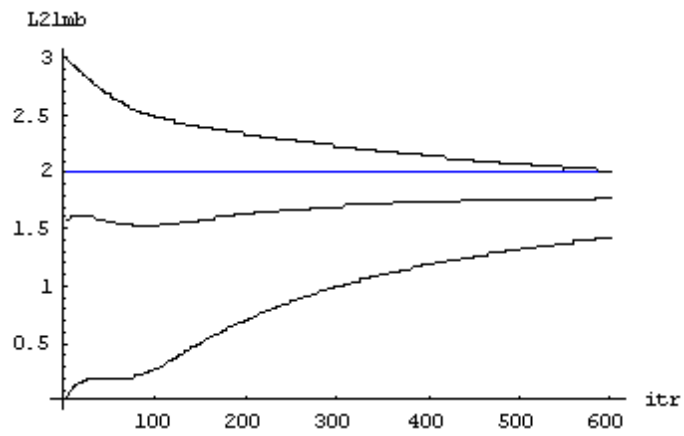


Figure 41: Tracé de $\text{Log}_2(\lambda_i)$ en fonction du nombre d'itérations dans le cas $F(X) = \text{sech}^2(X)$.

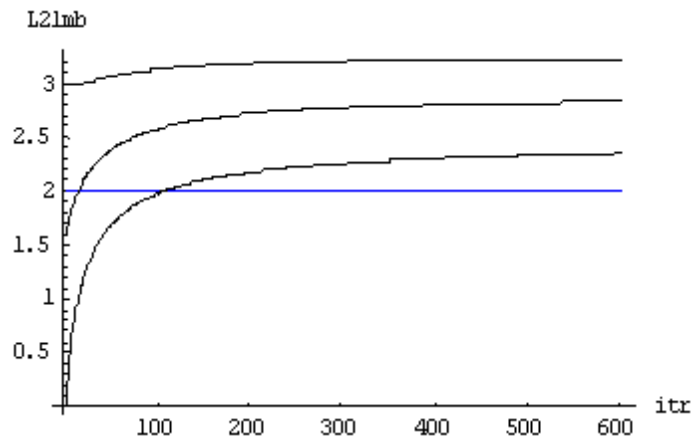


Figure 42: Tracé de $\text{Log}_2(\lambda_i)$ en fonction du nombre d'itérations dans le cas $F(X) = \text{sech}^4(X)$.

Conclusion générale

Les travaux présentés dans cette thèse ont porté sur l'application des réseaux de neurones-ondelettes à l'approximation de tâches et à l'analyse de certaines oscillations localisées associées à deux modèles de plasmas dont l'un appartient à la classe des problèmes non intégrables. De nos jours, ces réseaux sont perçus comme un puissant outil pour développer des solutions alternatives de plus en plus efficaces.

Dans un premier temps, la technique de l'analyse en ondelettes a été présentée de manière succincte. Une description des réseaux connexionnistes a ensuite été donnée. Un aperçu sur l'apprentissage de ces modèles ainsi que les méthodes d'apprentissage qui s'appuient sur la rétropropagation de l'erreur ont été alors rapportés. Nous avons, par la suite, dégagé les propriétés spécifiques aux réseaux de neurones qui sont à l'origine de leur intérêt pratique toujours croissant. Nous avons alors présenté une méthodologie pour l'application et la mise en oeuvre de ces techniques neuronales. Cette partie s'est achevée par une revue générale des domaines d'applications.

Les capacités des réseaux d'ondelettes ont été étudiées sur des problèmes d'approximation de fonctions. En particulier, nous avons montré, à travers une étude systématique, les performances des réseaux d'ondelettes. Nous avons alors pu situer les capacités de tels réseaux relativement à deux autres modèles connexionnistes bien connus: les perceptrons multicouches et les réseaux à fonctions radiales. Bien qu'il semble difficile de conclure de manière définitive quant aux mérites des différents réseaux, nos résultats ont clairement plaidé en faveur des réseaux d'ondelettes tenant compte des deux paramètres structurels des fonctions utilisées (les translations et les dilatations). En effet, nous avons noté que l'erreur sur l'approximation converge rapidement vers une valeur faible lors de la prise en compte des deux paramètres des ondelettes dans l'apprentissage du réseau utilisé. Cette convergence est vraisemblablement liée à la convergence des paramètres de translation des cellules cachées. On comprend alors le comportement de l'algorithme d'apprentissage consistant en fait à "aligner" les centres des fonctions de base sur les extremas de la fonction à approcher. Les paramètres de dilatation sont ensuite modifiés de manière à améliorer l'approximation fournie. Ce phénomène est principalement dû à l'utilisation de modèles à fonctions lo-

cales pour l'approximation de fonctions présentant des variations ou oscillations localisées. D'ailleurs, ce sera souvent le cas des fonctions réelles qui sont généralement définies par un ensemble d'apprentissage. En effet, en dehors du domaine couvert par le vecteur d'entrée, ne disposant pas d'informations sur la fonction, on peut supposer que ses variations sont locales. Nous avons également pu tester les performances d'un algorithme du second ordre couramment utilisé: l'algorithme BFGS. Ce dernier a pu réduire considérablement l'erreur commise sur l'approximation. Ceci laisse encore augurer des améliorations possibles qui pourraient être apportées par d'autres algorithmes plus performants. Par ailleurs, nous avons clairement démontré que la tendance à la fusion des paramètres d'échelles au cours du processus d'apprentissage donne lieu à de nouvelles tâches élémentaires correspondant aux dérivées de la fonction d'activation relativement au paramètre d'échelle. Nous avons alors noté que pour les tâches qui exhibent un tel comportement, l'approximation de la fonction peut être légèrement améliorée et le temps d'apprentissage considérablement réduit en implémentant directement les tâches élémentaires ainsi que leurs dérivées dans la base fonctionnelle. Dans le but de mettre en oeuvre les techniques d'approximation développées ci-dessus, il nous a semblé nécessaire de montrer à travers un problème physique bien déterminé la validité d'une telle approche en empruntant à la physique des plasmas deux modèles non linéaires. Pour cela, nous avons choisi d'analyser les conditions d'existence des structures localisées (ondes solitaires) associées à deux modèles de plasma différents dont l'un appartient à la classe des systèmes non intégrables. Cette partie nous a permis de construire le pseudo-potentiel correspondant à un système dynamique non intégrable. Notre méthode d'approximation peut alors apparaître, à bien des égards, comme une alternative, certes perfectible mais fiable, pour la description de la dynamique d'un système non intégrable.

Différentes applications peuvent tirer profit du phénomène de fusion (merging) observé lors de notre investigation. En réalité, pour les systèmes biologiques, la diversification des neurones entre tâches élémentaires et leurs dérivées pourrait constituer des concepts pouvant revêtir un intérêt certain. En effet, le système visuel par exemple est constitué

de capteurs qui sont les yeux, du nerf optique qui assure la liaison entre les capteurs et le cerveau et du cortex visuel qui assure une partie importante du traitement de l'information visuelle. Les fonctions de l'oeil sont au nombre de deux: la vision et le regard. Ce dernier étant un mouvement de poursuite oculaire, donc faisant intervenir la notion de vitesse, une analogie pourrait, dans une certaine mesure, révéler l'existence probable de cellules particulières dérivées dans le réseau responsable de la fonction regard. En effet, de nos jours, ce n'est plus le paradigme neurobiologique qui contribue au développement des réseaux de neurones. Au contraire, ce sont les réseaux de neurones formels qui contribuent, de plus en plus fréquemment, à la compréhension des systèmes nerveux vivants, car ils constituent des outils précieux pour en construire des modèles, simples mais utiles. Il y'a là un champ de recherche fascinant, complètement ouvert.

Par ailleurs, ce phénomène de fusion nous a permis de conclure quant à l'existence (ou non) des particules piégées résonantes. Une identification dynamique nous a alors permis de distinguer entre différents types de structures solitaires.

Les perspectives du présent travail de recherche sont nombreuses. On se propose à l'avenir d'étendre notre analyse au cas multi-dimensionnel. Par ailleurs, il serait intéressant de considérer un problème de modélisation d'un phénomène physique complexe en faisant appel à des réseaux à plusieurs entrées et plusieurs couches cachées. L'utilisation de réseaux bouclés nous permettront de décrire la dynamique d'un système physique donné en faisant appel, à titre d'exemple, à l'équation de Duffing. Il serait également souhaitable de considérer d'autres systèmes non intégrables.

Finalement, nous pouvons affirmer (et c'est là une intime conviction), que dans un avenir très proche, on peut s'attendre à ce que les architectures parallèles soient généralisées. Ces dernières constitueront probablement la première génération de neuro-ordinateurs. Ainsi, les réseaux de neurones, dotés peu à peu de supports matériels de plus en plus adaptés et puissants, mieux étudiés et mieux connus, deviendront une véritable technologie nouvelle.

Bibliographie

- [1] K. Hornik, M. Stinchcombe et H. White, *Neural Networks* **2**, 359 (1989).
- [2] S. Knerr, *Une méthode nouvelle de création automatique de réseaux neurones pour la classification de données: Application à la reconnaissance de chiffres manuscrits*, thèse de Doctorat de l'Université Pierre et Marie-Curie, Paris (1991).
- [3] W. S. McCulloch et W. Pitts, *Bulletin of Mathematical Biophysics* **5**, 115 (1943).
- [4] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Spartan, Washington DC, 1962).
- [5] M. Minsky et S. Papert, *Perceptrons* (MIT Press, Cambridge, 1969).
- [6] J. J. Hopfield, *Proceedings of the National Academy of Sciences, Etats-unis* **79**, 2554 (1982).
- [7] D. E. Rumelhart et J. L. McClelland, *Parallel Distributed Processing* (MIT Press, Cambridge, 1986).
- [8] P. J. Werbos, *Beyond regression : new tools for prediction and analysis in the behavioural sciences*, Ph. D. thesis, Harvard University (1974).
- [9] G. Cybenko, *Mathematics of Control, Signals and Systems* **2**, 303 (1989).
- [10] F. Girosi, M. Jones et T. Poggio, *Neural Computation* **7**, 219 (1995).
- [11] D. S. Broomhead et D. Lowe, *Complex Systems* **2**, 321 (1988).

-
- [12] Y. C. Pati et P. S. Krishnaprasad, *IEEE Trans. on Neural Networks* **4**, 73 (1993).
- [13] S. Mallat, *A Wavelet Tour of Signal Processing* (Academic Press, New York, 1998).
- [14] A. Benveniste et Q. Zhang, *IEEE Trans. Neural Networks* **6**, 889(1992).
- [15] Y. Oussar, *Réseaux d'ondelettes et réseaux de neurones pour la modélisation statique et dynamique de processus*, thèse de Doctorat de l'Université Pierre et Marie-Curie, Paris (1998).
- [16] B. G. Giraud et A. Touzeau, *Phys. Rev. E* **65**, 01609 (2001).
- [17] L. Ait Gougam, M. Tribeche et F. Mekideche, *Neural Networks* **21**, 1311 (2008).
- [18] L. Ait Gougam, M. Tribeche et F. Mekideche, *Journal of Plasma Physics* **73**, 901 (2007).
- [19] M. Tribeche, L. Ait Gougam et T. H. Zerguini, *Physica Scripta* **75**, 354 (2007).
- [20] L. Ait Gougam, M. Tribeche et F. Mekideche, *Journal of Plasma Physics* **73**, 331 (2007).
- [21] M. Tribeche, L. Ait Gougam et K. Aoutou, *Physics of Plasmas* **13**, 092105 (2006).
- [22] G. Dreyfus, J.-M. Martinez, M. Samuelides *et al.*, *Réseaux de neurones, méthodologie et applications* (Editions Eyrolles, Paris, 2002).
- [23] D. Gabor, *J. Inst. Elect. Eng.* **93**, 429 (1946).
- [24] A. Gossmann et J. Morlet, *J. Math. Anal.* **15**, 723 (1984).
- [25] Y. Meyer, *Wavelets and Applications* (Springer Verlage, 1991).
- [26] I. Daubechies, *Comm. on Pur and Applied Math.* **XLI**, 909 (1988).
- [27] S. Goedecker, *Wavelets And their Applications, for the Solution of Partial Differential Equations in Physics* (Presses polytechniques et universitaires romandes, Lausanne, 1998).

-
- [28] S. Dumont, *Ondelettes, Homogénéisation Périodique et Elasticité*, PhD thesis, Université de Montpellier II (1996).
- [29] M. Brewster et G. Beylkin, *Applied and Computational Harmonic Analysis* **2**, 327 (1995).
- [30] G. Beylkin, M. Brewster et A. Gilbert, *Applied and Computational Harmonic Analysis* **5**, 450 (1998).
- [31] M. Farge, N. Kevlahan, V. Perrier et K. Schneider, *Turbulence analysis, modeling and computing using wavelets* (J. Van Den Berg, Cambridge University Press, 1999).
- [32] B. Bulke Hubbard, *Ondes et Ondelettes, la Saga D'un Outil Mathématique* (Diffusion Belin, Paris 1995).
- [33] B. Torrèsani, *Analyse continue par ondelettes, Savoirs actuels* (InterEdition/ CNRS edition, 1995).
- [34] I. Daubechies, *Ten Lectures on Wavelets* (CBMS-NSF Regional Series in Applied Mathematics, SIAM, Philadelphia, 1992).
- [35] D.O. Hebb, *The organization of behavior: a neuropsychological theory* (Wiley, NewYork, 1949).
- [36] M. J. D. Powell, *Radial Basis Functions for Multi-variable Interpolation : A Review*, IMA Conference on Algorithms for the Approximation of Functions and Data (RMCS Shrivenham, UK, 1985).
- [37] S. Mallat, *IEEE Trans. Pattern Anal. Machine* **11**, 674 (1989).
- [38] O. Nerrand, *Réseaux de Neurones pour le Filtrage Adaptatif, l'Identification et la Commande de Processus*, Thèse de Doctorat de l'Université Paris VI (1992).
- [39] G. Dreyfus et Y. Idan, *Neural Computation* **10**, 133 (1998).

- [40] C. Bishop, *Neural Networks for Pattern Recognition* (Clarendon Press – Oxford, New York, 1995).
- [41] K. Levenberg, *Quarterly Journal of Applied Mathematics* **II** (2), 164 (1944).
- [42] D.W. Marquardt, *Journal of the Society of Industrial and Applied Mathematics* **11**, 431 (1963).
- [43] K. Funahashi, *Neural Networks* **2**, 183 (1989).
- [44] K. Hornik, M. Stinchcombe, H. White et P. Auer, *Neural Computation* **6**, 1262 (1994).
- [45] A. Barron, *IEEE Transactions on Information Theory* **39**, 930 (1993)
- [46] J. J. Hopfield et D. W. Tank, *Science* **233**, 625 (1986).
- [47] R. P. Lippmann, *IEEE ASSP Magazine* **4**, 4 (1987).
- [48] S. Haykin, *Neural Networks: A Comprehensive Foundation* (Macmillan, New York, 1994).
- [49] K. S. Narendra et K. Parthasarathy, *IEEE Trans. Neural Networks* **1**, 4 (1990).
- [50] S. Arik et Chaos, *Solitons & Fractals* **26**, 1407 (2005).
- [51] J. Liang et J. Cao, *Chaos, Solitons & Fractals* **22**, 7 (2004).
- [52] E. J. Hartman, J. D. Keeler et J. M. Kowalski, *Neural Computation* **2**, 210 (1990).
- [53] B. G. Giraud, A. Lapedes, L. C. Liu et J. C. Lemm, *Neural Networks* **8**, 757 (1995).
- [54] B. G. Giraud, L.C. Liu, C. Bernard et H. Axelrad, *Neural Networks* **4**, 803 (1991).
- [55] Y. Oussar et G. Dreyfus, *Neurocomputing* **34**, 131 (2000).
- [56] J. Zhang, G. G. Walter, Y. Miao et W. N. W. Lee, *IEEE Trans. Signal Processing* **43**, 1485 (1995).

-
- [57] G. Erlebacher, M. Y. Hussaini et L. M. Jameson, *Wavelet Theory and Applications* (Oxford University Press, Oxford, 1996).
- [58] A. N. Kolmogorov, Dokladi Akademii Nauk SSSR **108**, 358 (1956).
- [59] A. N. Kolmogorov, Dokladi Akademii Nauk SSSR **114**, 953 (1957).
- [60] V. Kurkova, Neural Networks **5**, 501 (1992).
- [61] R. Heht-Nielsen, *Kolmogorov's mapping neural network existence theorem*, in Proceedings of the IEEE Intl. Joint Conference on Neural Networks **3**, 11 (1987).
- [62] L. Spitzer Jr., *Physics of Fully Ionized Gases* (Wiley, New York, 1962).
- [63] F. F. Chen, *Introduction to Plasma Physics* (Plenum Press, New York, 1974).
- [64] B. Held, *Cold plasma Physics* (Masson, Paris, 1994).
- [65] N. A. Krall and A. W. Trivelpiece, *Principles of Plasma Physics* (McGraw-Hill, New York, 1973).
- [66] A.A. Mamun, R.A. Cairns et P. K. Shukla, Phys. Pasma **3**, 702 (1996).
- [67] R. Z. Sagdeev, in *Reviews of Plasma physics*, edited by M. A. Leontovich (Consultant Bureau, New York ,1966) Vol. 4, p. 23.
- [68] P. K. Shukla et A. A. Mamun, *Introduction to Dusty Plasma Physics* (Institute of Physics, Bristol, 2002).
- [69] H. Schamel, Phys. Repts. **140**, 161 (1986).
- [70] J. E. Allen, Phys. Scr. **45**, 497 (1992).
- [71] J. X. Ma, Jin-yuan Liu et M. Y. Yu, Phys. Rev. E **55**, 4627 (1997).
- [72] Jin-yuan Liu et J. X. Ma, Phys. Plasmas **4**, 2798 (1997).
- [73] R. M. Corless, G. H. Gonnet, D. E. G. Hare *et al.*, Adv. Comput. Math. **5**, 329 (1996).

- [74] B. Buti, Phys. Lett. A **235**, 241 (1997).