POLYPUBLIE
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Conception de l'algorithme d'apprentissage supervisé d'un réseau multicouche de quantrons
Auteur: Author:	Romain Lastère
Date:	2005
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Lastère, R. (2005). Conception de l'algorithme d'apprentissage supervisé d'un réseau multicouche de quantrons [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7407/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/7407/
Directeurs de recherche: Advisors:	Richard Labib
Programme: Program:	Unspecified

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

CONCEPTION DE L'ALGORITHME D'APPRENTISSAGE SUPERVISÉ D'UN RÉSEAU MULTICOUCHE DE QUANTRONS

ROMAIN LASTÈRE

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(MATHÉMATIQUES APPLIQUÉES)

AVRIL 2005

© Romain Lastère, 2005.

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-494-01350-8
Our file Notre référence
ISBN: 0-494-01350-8

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

CONCEPTION DE L'ALGORITHME D'APPRENTISSAGE SUPERVISÉ D'UN RÉSEAU MULTICOUCHE DE QUANTRONS

présenté par: LASTÈRE Romain

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. ADJENGUE Luc D., Ph.D., président

M. LABIB Richard, Ph.D., membre et directeur de recherche

M. GAGNON Michel, Ph.D., membre

À mes parents, mes soeurs

et la femme que j'aime,

REMERCIEMENTS

Je tiens à souligner en premier la contribution primordiale du docteur Richard Labib qui m'a soutenu tout au long de la réalisation de ma maîtrise. Il m'a apporté un soutien financier indispensable et une aide technique primordiale. Je tiens à le remercier tout particulièrement pour sa disponibilité, sa patience et sa minutie. J'espère avoir été à la hauteur de ses attentes.

Je tiens à remercier ma famille pour leur soutien ainsi que Julie Belley Perron pour son soutien et son aide à la correction de mon mémoire.

RÉSUMÉ

Le quantron est un nouveau modèle de neurone artificiel. Pour exploiter au maximum ces caractéristiques uniques, il est primordial de développer un algorithme d'apprentissage.

Premièrement, nous examinons les différents modèles de neurones ressemblant au quantron, incluant le perceptron. Cette exploration permet de connaître les différents algorithmes d'apprentissage supervisés existants. Malgré la multitude d'algorithmes d'apprentissage existants, aucun ne satisfait les caractéristiques uniques du quantron, nous devrons alors proposer un algorithme d'apprentissage original.

Deuxièmement, l'algorithme d'apprentissage proposé est détaillé. La principale difficulté est de gérer la conduction ou la non conduction de l'information dans le quantron. L'algorithme d'apprentissage que nous avons créé ne s'applique pas qu'au quantron et, avant de l'implanter au quantron, nous avons observé son comportement sur un modèle de quantron simplifié. Les résultats sont excellents, puisque l'algorithme d'apprentissage permet de minimiser l'erreur. Cette partie de la recherche est novatrice puisque jusqu'à ce jour aucun algorithme d'apprentissage ne permettait l'apprentissage de ce type de neurone.

Finalement, encouragé par les résultats avec le quantron simplifié, nous avons implanté l'algorithme d'apprentissage à un réseau multicouche de quantrons. Nous testons l'algorithme d'apprentissage par reconnaissance d'image. Les résultats obtenus sont très encourageants. Tel que le quantron, notre algorithme d'apprentissage est plus fidèle aux récentes découvertes de la neurobiologie. Nous croyons qu'il permettra au réseau de neurones de résoudre des problèmes plus complexes de classification et de régression.

ABSTRACT

The quantron is a new model of artificial neuron. In order to take advantage of its unique attributes, we must develop a learning algorithm.

Initially, we investigate various existing neuron models similar to the quantron. A document review reveals various supervised learning algorithms. However, from the abundance of supervised learning algorithms, none satisfies the unique attributes of the quantron. It is therefore necessary to produce an original learning algorithm.

Secondly, the propose learning algorithm is detailed. The principal difficulty is to manage the conduction or non-conduction of information in the quantron. The learning algorithm that we created does not apply only to the quantron. Before testing it with the quantron, we observe its behavior on a simplified quantron model. The results are excellent, since the learning algorithm makes it possible to minimize error. This part of the research is innovative because no other learning algorithm has been applicable to this type of neuron.

Finally, encouraged by results from the simplified quantron, we established the learning algorithm into a multilayer network of quantrons. We test the learning algorithm by image recognition. The results obtained are very encouraging. Our learning algorithm for the quantron is more accurately reflects recent discoveries in neurobiology. We

believe that it will allow the neural network to solve more complex problems of classification and regression.

TABLE DES MATIÈRES

DÉDICACE ..... **iv**

REMERCIEMENTS ..... **v**

RÉSUMÉ ..... **vi**

ABSTRACT ..... **viii**

TABLE DES MATIÈRES ..... **x**

LISTE DES TABLEAUX ..... **xii**

LISTE DES FIGURES ..... **xiii**

INTRODUCTION **1**

CHAPITRE 1 : EXPLORATION DU MONDE DU QUANTRON **6**

1.1 Neurone artificiel et algorithme d'apprentissage *7*

1.2 Le quantron *17*

CHAPITRE 2 : ALGORITHME D'APPRENTISSAGE **26**

2.1 Description *27*

2.1.1 Cas #1 : Absence de modification 29

2.1.2 Cas #2 : Excitation du réseau 29

2.1.3 Cas #3 : Inhibition du réseau 31

2.1.4 Cas #4 : Rétropropagation de l'erreur 33

2.1.5 Généralisation à plusieurs sorties 37

2.2 Expérimentation avec un modèle de quantron simplifié 38

2.2.1 Modèle de quantron simplifié 39

2.2.2 Résultat 41

2.3.3 Discussion 47

CHAPITRE 3 : IMPLANTATION DE L'ALGORITHME D'APPRENTISSAGE DANS UN RÉSEAU DE QUANTRONS 49

3.1 Propagation avant du signal d'entrée 50

3.2 Implantation des quatre cas d'apprentissage spécifiques au quantron 57

3.2.1 Excitation du quantron 58

3.2.2 Inhibition du quantron 61

3.2.3 Rétropropagation de l'erreur pour le quantron 63

3.3 Expérimentation avec le quantron 73

3.3.1 Résultats 74

3.2.3 Discussion 81

CONCLUSION 86

BIBLIOGRAPHIE 90

LISTE DES TABLEAUX

Tableau 2.1 : Énumération des quatre cas d'apprentissage du quantron 28

Tableau 2.2 : Comparaison entre l'algorithme de rétropropagation de l'erreur pour le perceptron et le neurone général 36

Tableau 2.3: Énumération des différents modèles utilisés 40

Tableau 2.4 : Énumération des différents gradients nécessaires pour l'inhibition ou l'excitation du neurone 41

Tableau 2.5 : Tableau de résultats pour une simulation avec 100 réseaux 47

Tableau 3.1 : Dérivées partielles des coefficients par rapport aux paramètres synaptiques pour le cas de l'excitation du quantron 61

Tableau 3.2 : Dérivées partielles des coefficients par rapport aux paramètres synaptiques pour le cas de la rétropropagation de l'erreur 72

Tableau 3.3 : Gradients d'erreurs nécessaires à l'algorithme d'apprentissage 73

Tableau 3.4 : Résultat pour la convergence de l'algorithme 77

Tableau 3.5 : Énoncé du problème du « ou exclusif » pour le quantron 78

Tableau 3.6 : Résultat de la résolution du problème du « ou exclusif » avec un seul quantron 78

Tableau 3.7 : Résultat pour la résolution d'un problème linéairement séparable 81

LISTE DES FIGURES

Figure 1.1 : Représentation du perceptron 9

Figure 1.2 : Représentation de la transmission de l'information dans un réseau de perceptrons et de quantrons 12

Figure 1.3 : Représentation du neurone à impulsion 15

Figure 1.4 : Schéma d'un neurone biologique 18

Figure 1.5 : Schéma de la transmission d'information aux liaisons synaptiques 19

Figure 1.6 : Noyau du quantron ou $\varphi(t)$ 22

Figure 1.7: Représentation des différents signaux internes au quantron 24

Figure 2.1 : Les étapes de l'apprentissage supervisé 27

Figure 2.2 : Exemple de réseau à exciter 29

Figure 2.3 : Différents noyaux implantable au quantron. 31

Figure 2.4 : Exemple de réseau à inhiber 32

Figure 2.5 : Modification de la structure du réseau de neurone lors de la rétropropagation de l'erreur 34

Figure 2.6 : Structure et conventions nécessaire à l'implantation de l'algorithme de rétropropagation de l'erreur 35

Figure 2.7 : Exemple de l'algorithme d'apprentissage pour un réseau comportant plusieurs sorties 38

Figure 2.8 : Représentation du modèle simplifié 40

Figure 2.9 : Superposition de l'image désirée (rond) et de l'image obtenue (croix) avant l'apprentissage pour un réseau de quantrons simplifiés quadratiques.....44

Figure 2.10 : Superposition de l'image désirée (rond) et de l'image obtenue (croix) après un apprentissage de dix époques.....45

Figure 2.11 : Résultat de la simulation avec le quantron simplifié quadratique.....46

Figure 3.1 : Fenêtre de la fonction à approximer.....51

Figure 3.2 : Extension paire de la fenêtre d'approximation.....51

Figure 3.3 : Modification du noyau du quantron.....53

Figure 3.4 : Représentation de la réponse totale à exciter au moment le plus propice59

Figure 3.5 : Représentation de la réponse totale à inhiber au moment le plus propice62

Figure 3.6 : Superposition des valeurs obtenues (croix) et des valeurs désirées (cercles) pour l'apprentissage d'un réseau multicouche de quantrons.....75

Figure 3.7 : Graphique de l'erreur en fonction de l'époque.....76

Figure 3.8 : Fonction discriminante obtenue lors de la résolution d'un problème linéairement séparable.....79

Figure 3.9 : Graphique de l'erreur en fonction de l'époque de l'apprentissage.....80

Figure 3.10 : Caractéristiques de l'image de sortie du quantron pour deux entrées.....82

INTRODUCTION

« ...pour bien conduire sa raison et chercher la vérité dans les sciences... »[1],

Il ne se passe pas une seule journée sans que quelqu'un, dans une quelconque partie du monde ne découvre une nouvelle pièce du grand casse-tête qu'est la connaissance. L'homme, à travers les siècles, n'a jamais cessé de penser, de comprendre et d'analyser. L'outil qui nous permet cette évolution est le cerveau, organe sur lequel les scientifiques se sont toujours penchés afin d'en percer les mystères.

Cette obsession de la compréhension n'a pas toujours été profitable et les expérimentations effectuées sur le cerveau n'ont pas toujours été fructueuses. Plusieurs pratiques chirurgicales, telle que la lobotomie, ont fourni une preuve des limites de la compréhension de l'être humain. Par contre, ce désir de comprendre a mené les professionnels du monde médical à décrire le fonctionnement du cerveau et identifier ses caractéristiques pour, notamment, aider les mathématiciens à en reproduire certains éléments. À partir de la représentation du neurone biologique, les chercheurs ont reproduit la structure du cerveau pour résoudre des problèmes divers. Il en résulta un nouveau champ de recherche : les réseaux de neurones artificiels.

---

[1] Descartes, René, *Discours de la méthode, 2 ième partie*

Pendant près de vingt ans, le psychiatre McCulloch a réfléchi à une représentation artificielle du neurone biologique dans le but de comprendre mieux le fonctionnement du cerveau. Au point tournant de ses recherches, Pitts, un mathématicien reconnu, s'est joint à lui pour lui permettre de modéliser ses idées. Ensemble en 1943, ils ont écrit un article présentant une unité de calcul logique provenant de leur étude neurobiologique. Cet article, lu par de nombreux scientifiques, a été, notamment, l'inspiration de von Neumann pour la création d'une architecture électronique qui est maintenant la base des microprocesseurs (Haykin (1996)).

En 1958, Rosenblatt introduit le perceptron, premier neurone artificiel ayant été créé sur la base des travaux de McCulloch et Pitts. La découverte du perceptron est accompagnée d'une nouvelle méthode d'apprentissage supervisé. L'engouement pour le perceptron a pris fin abruptement par la publication du livre « Perceptrons » de Minsky et Papert (1988) qui démontrait les limites du perceptron. Par contre, une trentaine d'années plus tard, le développement de l'algorithme de rétropropagation de l'erreur par Rumelhart, Hinton et Williams (1986) a fait revivre l'intérêt pour le perceptron en permettant l'utilisation du perceptron en réseau multicouche plutôt qu'en une seule couche. Depuis ce temps, le réseau multicouche de perceptrons est le réseau de neurones le plus populaire (Haykin (1996)). Aujourd'hui, ce type de réseau est utilisé pour la reconnaissance des cancers (Ng et al. (1993)), l'aide au diagnostic de la maladie de l'Alzeimer (Chan et al. (1993)) et beaucoup d'autres applications.

Avec les récentes découvertes neurobiologiques, le perceptron est devenu un modèle rudimentaire qui ne représente pas le fonctionnement réel du neurone biologique et qui gagnerait à être amélioré. Labib (1999) a développé un modèle de neurone artificiel, le quantron, davantage fidèle à nos connaissances sur la neurobiologie. Par contre, ce neurone artificiel ne possède pas d'algorithme d'apprentissage efficace, outil indispensable à l'exploitation du neurone. En effet, le rôle de l'algorithme d'apprentissage est de modifier les paramètres des neurones du réseau pour que celui-ci puisse s'adapter à son environnement ou, dans notre cas, au problème à résoudre. La création de l'algorithme d'apprentissage original approprié au quantron est le sujet de cette recherche.

Le premier chapitre de notre recherche s'intéresse aux différents modèles de neurones artificiels ressemblant au quantron afin de connaître leur algorithme d'apprentissage. Le premier modèle étudié est le perceptron qui utilise l'algorithme d'apprentissage classique de rétropropagation de l'erreur sur lequel on se basera, malgré son improbabilité biologique. Le second modèle étudié est le neurone à impulsion qui possède de grandes similarités avec le quantron puisqu'il se base sur les nouvelles découvertes en neurobiologie. Cependant, des différences importantes existent, ce qui nous permet de comprendre que le nouvel algorithme d'apprentissage doit être original puisque aucun algorithme d'apprentissage existant ne satisfait toutes les caractéristiques du quantron. Pour nous permettre de découvrir l'algorithme d'apprentissage répondant au besoin du quantron, les caractéristiques de celui-ci sont détaillées.

Le deuxième chapitre détaille l'algorithme d'apprentissage proposé. La principale difficulté consiste à gérer la conduction ou la non conduction de l'information au sein du quantron. Nous devons identifier les différents cas d'apprentissage pour s'adapter aux propriétés uniques du quantron. Un apprentissage est introduit pour chacun de ces cas. Avant l'implantation de notre algorithme d'apprentissage au quantron, nous observons son comportement sur un modèle de quantron simplifié. Ce modèle a été développé en simplifiant le quantron original tout en gardant ses propriétés. Nos tests avec le quantron simplifié consistent à effectuer de la reconnaissance d'image. Les résultats révèlent une nouvelle découverte, celle d'un algorithme permettant l'apprentissage de ce type de neurone.

Le dernier et troisième chapitre traite de l'implantation de l'algorithme d'apprentissage à un réseau multicouche de quantrons. Cette implantation est plus complexe qu'avec le modèle simplifié puisque nous devons effectuer une approximation de la réponse totale pour déterminer la sortie du quantron. Nous devons également pallier aux particularités de chacun des cas d'apprentissage identifiés au chapitre deux. L'implantation de ces cas pose certaines difficultés que nous surmonterons par différentes opérations. Par la suite, nous testons l'algorithme d'apprentissage implanté par la reconnaissance d'image pour trois problèmes distincts.

Nous croyons que les résultats que nous obtiendrons permettent de conclure que l'algorithme d'apprentissage permet au réseau multicouche de quantrons de résoudre des problèmes plus complexes de classification et de régression.

Notre recherche tend à présenter la nécessité de l'algorithme d'apprentissage dans le développement d'un neurone artificiel. L'intérêt principal est d'obtenir un réseau de neurones plus puissant que le réseau multicouche de perceptrons qui résoudra des problèmes plus complexes, plus rapidement.

CHAPITRE 1 :

EXPLORATION DU MONDE DU QUANTRON

Depuis sa création, l'homme tente de percer le mystère du fonctionnement du corps humain. L'agencement des organes et de la matière qui forment l'être humain est un exemple complexe à reproduire au sein duquel chaque partie de notre corps joue un rôle spécifique. La complexité des opérations intellectuelles que l'homme peut résoudre s'explique par l'usage de son cerveau, dont l'unité de base est le neurone.

La littérature scientifique sur les réseaux de neurones artificiels a débuté par la reproduction d'un simple neurone pour finalement découvrir les avantages d'utilisation du neurone en réseau. Afin de reproduire un réseau de neurones, la création d'un algorithme d'apprentissage s'avère indispensable pour la résolution de tout problème, car il permet d'adapter les paramètres du neurone dans le réseau pour résoudre le problème donné. De nombreux scientifiques ont proposé des modèles de neurones ou de réseau de neurones artificiels assortis de leur algorithme d'apprentissage. Après une description sommaire d'un type de neurones artificiels, nous concentrerons notre revue de la littérature sur les algorithmes d'apprentissage supervisés qui ont été créés pour divers types de modèles de réseau de neurones artificiel. Finalement, nous présenterons un nouveau modèle de neurone extrêmement prometteur : le quantron.

1.1 Neurone artificiel et algorithme d'apprentissage

Le quantron est un nouveau modèle de neurone artificiel, développé par Labib (1999), se basant sur des observations biologiques récentes et sur l'étude du mouvement brownien des liaisons synaptiques. Comme les neurones biologiques, le quantron est stimulé par des potentiels d'action à ses synapses. Cette stimulation produit une onde variant dans le temps à chacune des synapses. Les ondes des entrées sont sommées. L'onde produite correspond à l'activité du neurone. À l'instant où l'activité du neurone dépasse le seuil critique, le quantron émet un nombre représentant la période des potentiels d'action. Dans le cas où l'activité n'atteint pas le seuil critique, le neurone ne transmet aucune information. Ce nouveau modèle de neurone peut résoudre des problèmes classiques en utilisant un nombre de paramètres inférieur à tout autre modèle de neurone. Cependant, il n'existe pas d'algorithme d'apprentissage. La prémise de Labib (2000) est qu'« une meilleure compréhension du fonctionnement du cerveau humain devrait se traduire par la création d'une machine neuronale artificielle plus robuste »[2]. Dans cette optique, l'algorithme que nous développerons tentera d'être plausible biologiquement pour respecter cette idée. Cette description du quantron sera approfondit dans la prochaine section mais elle est suffisante pour déterminer les modèles de neurone ressemblant au quantron et connaître leurs algorithmes d'apprentissage.

---

[2] Labib (200) p.81

Le premier neurone artificiel ayant été créé se nomme le perceptron. Il fut développé en 1958 par Rosenblatt à partir des travaux de McCulloch et Pitts (Haykin, 1999). Ces derniers avaient affirmé que la sortie du neurone biologique suivait un comportement « tout ou rien ». Rosenblatt a modélisé ce comportement par une fonction échelon pour être utilisée comme fonction d'activation du perceptron. Ce modèle est aujourd'hui très rudimentaire étant donné les récentes découvertes à propos du comportement du neurone biologique.

Le modèle du perceptron est une pondération par des poids synaptiques ω_1, ω_2, ..., ω_n des entrées x_1, x_2, ..., x_n. Un biais θ est ajouté pour ajuster le niveau des contributions des entrées. Ensuite, la somme de tous les signaux passe par une fonction d'activation φ. Les fonctions d'activation les plus utilisées sont la sigmoïde, la tangente hyperbolique et la fonction échelon. La figure 1.1 montre une perceptron ainsi que les fonctions entre les entrées et la sortie.

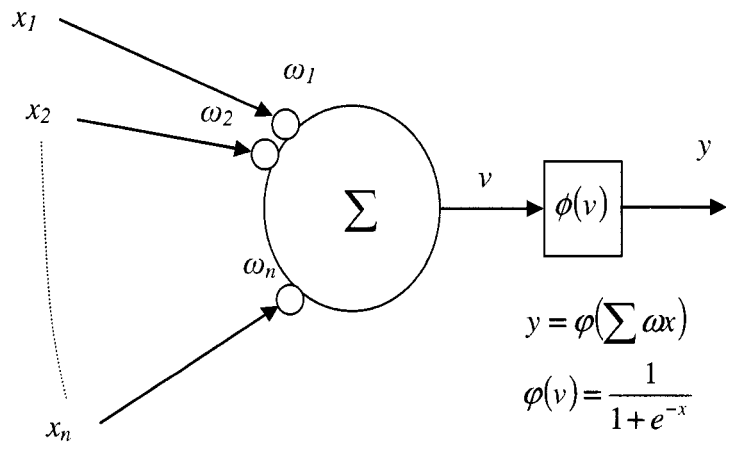


Figure 1.1 : Représentation du perceptron

L'ouvrage « Perceptrons » de Minsky et Papert (1988) a présenté les limites du perceptron. Cet ouvrage a énoncé le problème classique du « ou exclusif » ou « XOR » que le perceptron est incapable de résoudre seul. L'effet de cet ouvrage fut catastrophique pour la recherche sur les neurones artificiels puisqu'il a démontré que le perceptron ne pouvait résoudre les problèmes non linéaires.

L'intérêt pour le perceptron fut renouvelé par l'introduction de l'algorithme d'apprentissage de rétropropagation de l'erreur popularisé par Rumelhart, Hinton, et Williams (1986). Il n'était plus question d'un seul perceptron, mais d'un réseau multicouche de perceptrons. Cet algorithme propose de rétropropager l'erreur grâce aux dérivées partielles à l'intérieur d'un réseau multicouche de perceptrons, aussi nommé « MLP ». La nécessité d'obtenir les dérivées partielles du perceptron a obligé la modification de la fonction d'activation

échelon par la fonction d'activation sigmoïde ou tangente hyperbolique. Cette structure combinée avec l'algorithme de rétropropagation de l'erreur devint l'un des réseaux de neurones les plus populaires.

Plusieurs améliorations ont été proposées à l'algorithme d'apprentissage de rétropropagation de l'erreur assorti d'une fonction d'activation sigmoïde ou tangente hyperbolique. L'utilisation d'un gradient d'ordre deux, plutôt que le gradient initial d'ordre un, a été utilisé, notamment, par la méthode de Newton, développée par White (1990) et par la méthode de Levenberg-Marquardt, créée par Hagan et Manhaj (1994). D'autres améliorations ont modifié le gradient en lui ajoutant son conjugué (Hagan et Menhaj (1994)). Également, le poids de chaque synapse a été modifié selon le signe de sa dérivée partielle de l'erreur en fonction du poids synaptique dans le RPROP (*Resilient propagation*) de Riedmiller et Braun (1993). De nombreuses autres modifications ont été apportées à cet algorithme par une combinaison de différentes modifications, tel que l'algorithme de Jeong et Lee (1996).

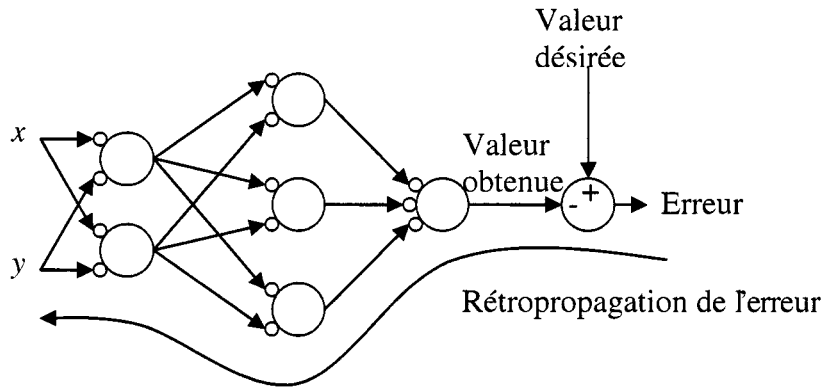
Des algorithmes d'apprentissage assortis d'une fonction d'activation échelon ont aussi été développés pour les réseaux multicouches de perceptrons. Toms (1990) a proposé un algorithme où la fonction d'activation est une somme de la fonction sigmoïde et de la fonction échelon qui sont liées par un paramètre α. Ce dernier paramètre est compris entre zéro et un, il diminue en fonction de l'erreur quadratique des sorties. Cependant, en effectuant les dérivées, Toms (1990) a obtenu un algorithme de rétropropagation

classique du perceptron. Goodman et Zeng (1994) ont obtenu un algorithme très similaire qui revient également à l'algorithme de rétropropagation classique du perceptron. L'algorithme obtenu par Barlett et Downs (1992) utilise un entraînement aléatoire des poids. Cet algorithme nécessite une descente du gradient tel que dans la rétropropagation de l'erreur.

Indépendamment de la modification de l'algorithme, la rétropropagation de l'erreur doit être effectuée. Ceci présuppose que l'information peut circuler peu importe les entrées et la configuration du réseau. La circulation inconditionnelle de l'information est un des problèmes que nous rencontrerons avec le réseau multicouche de quantrons, aussi nommé « MLQ », que nous décrirons en détail dans la deuxième partie de ce chapitre. Pour le moment, retenons que dans certaines situations le quantron peut ne pas conduire d'information, ce qui entraîne l'impossibilité de propager le signal vers la sortie donc l'impossibilité de rétropropager l'erreur aux entrées. La figure 10.2 montre justement cette différence cruciale entre les réseaux MLP et MLQ. Le premier réseau présente la transmission de l'information à travers le réseau multicouche de perceptrons. Cette transmission, représentée par une flèche, débute aux entrées et circule jusqu'à la sortie pour l'obtention d'une valeur qu'on nomme valeur obtenue. La différence entre la valeur obtenue et la valeur désirée représente l'erreur. Cette erreur est rétropropagée à travers le réseau par l'algorithme d'apprentissage de rétropropagation de l'erreur. Le second réseau est un « MLQ » ou un réseau multicouche de quantrons. Sous certaines conditions, le quantron peut ne pas transmettre l'information. Graphiquement, la non conduction est

représentée par une flèche pointillée recouverte d'une croix. Le réseau multicouche de quantrons présenté ne conduit pas le signal jusqu'à la sortie. Il est donc impossible d'appliquer l'algorithme d'apprentissage de rétropropagation de l'erreur puisque la transmission de l'information est coupée.

Transmission de l'information dans un MLP



Transmission de l'information dans un MLQ ou réseau de multicouche quantrons

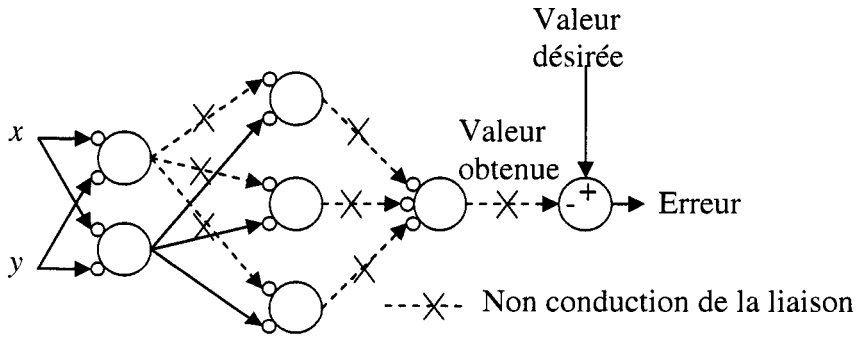


Figure 10.2 : Représentation de la transmission de l'information dans un réseau de perceptrons et de quantrons

Un autre modèle de neurone, appelé le TAF (*Trainable Activation Function*), a été introduit par Wu, Zhao et Ding (1997) et est constitué d'un modèle pour les synapses ainsi que pour la fonction d'activation. Cependant, les fonctions modélisant les synapses et la fonction d'activation sont continues et dérivables en tous points, ce qui diffère du quantron. Encore une fois, la technique de la rétropropagation de l'erreur est utilisée pour l'algorithme d'apprentissage du réseau de TAF.

À l'époque de sa création, le perceptron découlait d'observations biologiques. Il est évident que ces observations étaient trop simples pour représenter fidèlement la réalité mais en est-il de même pour l'algorithme d'apprentissage de rétropropagation de l'erreur? Crick et Koch (1989) et Zisper et Anderson (1988) ont conclu que cet algorithme n'était pas biologiquement plausible à cause du retour de l'information à travers le réseau de neurones. Pour sa part, Stork (1989) a renversé la question : que faudrait-il pour que la rétropropagation de l'erreur soit biologiquement plausible? Malheureusement, même si Stock (1989) amène certains éléments de réponses, il apparaît évident que la structure classique de l'algorithme d'apprentissage de rétropropagation de l'erreur n'est pas plausible. L'intérêt de

Randall et O'Reilly (1998) ont proposé six principes pour obtenir un réseau de neurones biologiquement plausible. L'un des principes est qu'il doit y avoir apprentissage supervisé par propagation de l'erreur et non par rétropropagation, c'est pourquoi l'algorithme d'apprentissage de rétropropagation de l'erreur ne peut être biologiquement plausible. La

littérature à ce sujet écarte l'algorithme de rétropropagation de l'erreur, sans proposer d'alternative puisque les auteurs ne s'accordent pas sur tous les points de l'apprentissage d'un réseau de neurones biologiques.

Récemment, d'autres modèles ont été développés à la lumière des récentes découvertes en neurobiologie. Il en résulte une famille de neurones appelée *spiking neurons* ou plutôt neurone à impulsion. Plusieurs modèles existent, les plus populaires sont décrits par Gerstner et Kistler (2002). Les neurones à impulsion proviennent du modèle de Hodgkin-Huxley (Hodgkin et Huxley (1952)), mais ce modèle étant trop complexe, des simplifications ont été effectuées et un grand éventail de modèles de neurone à impulsion en ont résulté.

Dans le neurone à impulsion, les entrées ainsi que les sorties sont des impulsions à un moment précis, le temps joue un rôle primordial. L'impulsion entre par les synapses du neurone qui génère un potentiel d'action. Ces potentiels d'action sont sommés dans le corps du neurone. Si le potentiel cumulatif dépasse le seuil de transmission, une impulsion est générée et transmise par l'axone et les dendrites aux autres neurones. Lors d'une transmission, une période de relaxation s'en suit. Pendant cet instant, aucune autre impulsion ne peut être émise. Les différences entre les modèles se situent au niveau de la forme du potentiel d'action et la période de relaxation. La figure 1.3 montre les différents éléments du neurone à impulsion.

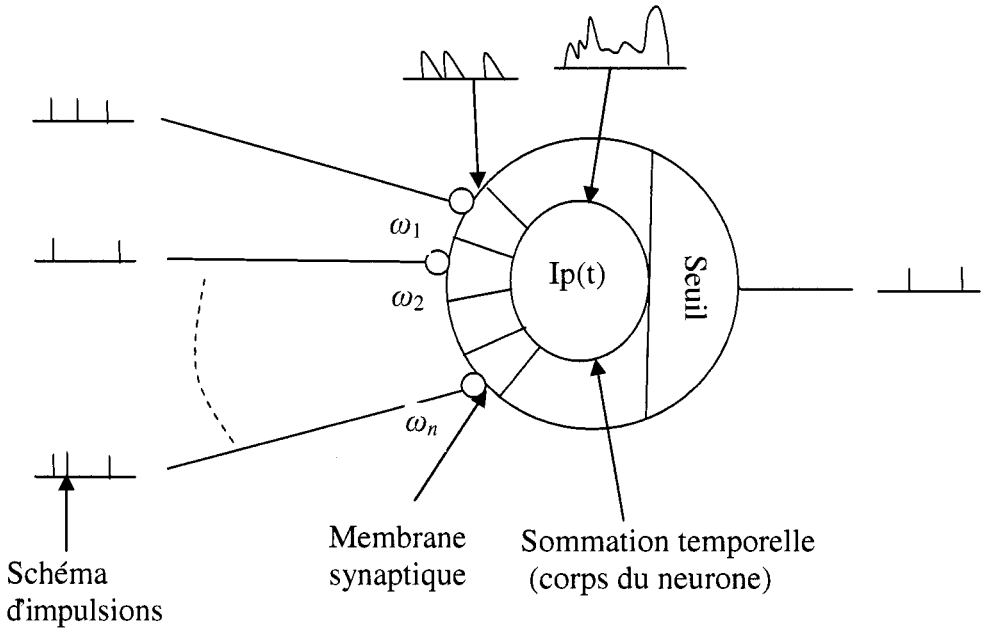


Figure 1.3 : Représentation du neurone à impulsion

La sortie d'un tel réseau de neurones est une série d'impulsion variant dans le temps composant un schéma d'impulsion. L'information de sortie est contenue dans ces impulsions. La principale difficulté est de connaître de quelle façon l'information est contenue dans ces séries d'impulsions. Ce problème consiste à déterminer le code utilisé par les neurones pour emmagasiner de l'information. Maass et Bishop (1999) décrivent les deux sortes de code utilisé par les neurones à impulsion. Le premier est le *pulse code* ou le code relatif aux impulsions, qui est relatif au moment de l'impulsion et qui peut être traduit par trois types de codes : le temps du premier déclenchement, la phase entre les impulsions des sorties et le synchronisme des sorties. Le second est le *rate code* ou le taux de sortie, également traduit par trois types de codes : le compte du nombre

d'impulsions pour une période donnée, la densité d'impulsions au régime permanent et l'activité moyenne d'une population. Ces différentes mesures s'appliquent à différents problèmes. L'un de ceux-ci est de simuler une population de neurones et d'en obtenir les résultats pour ensuite tenter de comprendre l'activité cérébrale. D'ailleurs, il a été démontré qu'une population de neurone à impulsions peut se comporter comme le perceptron (de Kamps et van der Velde (2001)).

L'apprentissage du réseau de neurones à impulsion est généralement un apprentissage non supervisé (Maass et Bishop (1999)). Par contre, les problèmes résolus par un apprentissage non supervisé s'éloignent considérablement de la résolution de problèmes de régression et de classification que le quantron résout. La sortie du quantron révèle une fréquence d'impulsions traduite par une valeur numérique tandis que la sortie du neurone à impulsion présente un schéma d'impulsions. Cette différence fondamentale mène vers des axes de recherche complètement différents puisque leurs champs d'applications divergent. C'est pourquoi, les algorithmes d'apprentissage des neurones à impulsion ne sont pas intéressants pour le quantron, malgré leurs ressemblances biologiques.

1.2 Le quantron

Le quantron est un nouveau modèle de neurone artificiel, développé par Labib (1999), se basant sur des observations biologiques récentes et sur l'étude du mouvement brownien des liaisons synaptiques. Tout comme les neurones biologiques, le quantron est stimulé par des potentiels d'action à ses synapses. Le quantron possède une plus grande quantité de caractéristiques du neurone biologique que tout autre modèle de neurone. C'est pourquoi nous croyons qu'il peut résoudre des problèmes plus complexes. Avant de présenter le fonctionnement et les caractéristiques du quantron, voici une explication sommaire du fonctionnement d'un neurone biologique dont la compréhension est indispensable pour comprendre le quantron.

La figure suivante présente les principaux composants du neurone biologique.

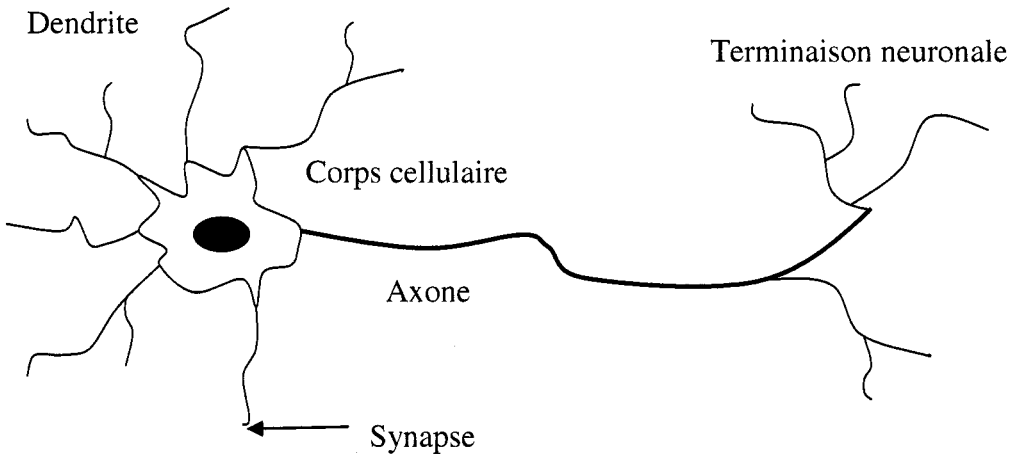


Figure 1.4 : Schéma d'un neurone biologique

La chaîne de transmission de l'information dans un neurone biologique débute par l'arrivée d'un potentiel d'action (ou influx nerveux) au niveau de la jonction synaptique. Cette arrivée active la libération des neurotransmetteurs, des particules jouant le rôle de messagers chimiques. Ces particules se propagent à travers la fente synaptique de manière aléatoire et un pourcentage de neurotransmetteurs se fixent à la membrane post-synaptique, qui est composée de récepteurs. La combinaison neurotransmetteurs-récepteurs modifie le potentiel électrique de la membrane post-synaptique. La modification est positive s'il s'agit d'une synapse excitatrice et est négative s'il s'agit d'une synapse inhibitrice. Ce processus est illustré à la figure 1.5.

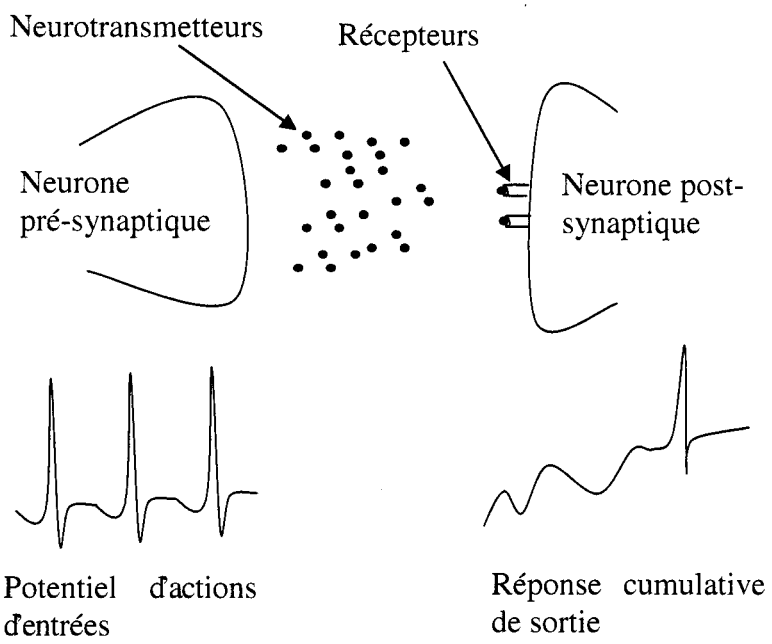


Figure 1.5 : Schéma de la transmission d'information aux liaisons synaptiques

L'influx nerveux est conduit par la dendrite vers le corps cellulaire et les influx nerveux se cumulent pour créer le niveau d'excitation. Si l'excitation totale causée par les entrées est supérieure au seuil critique de dépolarisation, un potentiel d'action sera émis et se propagera dans l'axone pour se répandre vers les terminaisons neuronales où le neurone excitera les neurones connectés à celui-ci (Maass et Bishop (1999)). Cependant, si l'excitation totale n'atteint pas le seuil critique, le neurone ne transmettra aucune information. Cette simple explication suffit pour amorcer l'étude du quantron.

Tel que nous le savons, la transmission de l'information dans un neurone débute à la synapse. Dans le quantron, l'activité synaptique a été modélisée à partir du principe de

diffusion en se basant sur le mouvement brownien. Il en résulte une fonction exprimant le voltage *V(t)* induit dans la membrane post-synaptique :

$$V(t) = \frac{2\omega}{\sqrt{2\pi t}} \int_{\ln a}^{\infty} e^{-x^2/2t} dx = 2\omega\left(1 - \Phi\left(\frac{\ln a}{\sqrt{t}}\right)\right) \tag{1.1}$$

où a est la longueur de la fente synaptique et sera considérée comme une constante. Le paramètre ω est une constante de proportionnalité. Il équivaut au poids synaptique qui pondère les valeurs d'entrées d'un perceptron. La fonction $\Phi(t)$ correspond à la fonction de répartition de la variable aléatoire gaussienne centrée réduite.

Le potentiel post-synaptique ne peut croître indéfiniment c'est pourquoi le temps d'interruption S a été introduit. La variation du voltage initial (*V.V.I.*) de la membrane post-synaptique a été définit comme suit :

$$V.V.I. \equiv V(t)(u(t) - u(t - S)) \tag{1.2}$$

où $u(t)$ représente la fonction échelon. À partir du point d'interruption, la membrane réintègre son état initial au même rythme. Ce retour à l'état initial correspond à la variation du voltage final (*V.V.F.*) :

$$V.V.F. \equiv -V(t-S)(u(t-S)-u(t-2S))+V(S)(u(t-S)-u(t-2S)) \qquad (1.3)$$

Ces deux étapes permettent d'exprimer le niveau global de la polarisation dans le neurone post-synaptique. Cette fonction $\varphi(t)$ est donnée par :

$$\varphi(t) \equiv V(t)u(t)+(V(S)-V(t)-V(t-S))u(t-S)+(V(t-S)-V(S))u(t-2S) \qquad (1.4)$$

Cette dernière équation détermine la variation du niveau de polarisation dans une synapse. La variation totale du potentiel synaptique $\varphi(t)$ se compare à un noyau ou *kernel*.

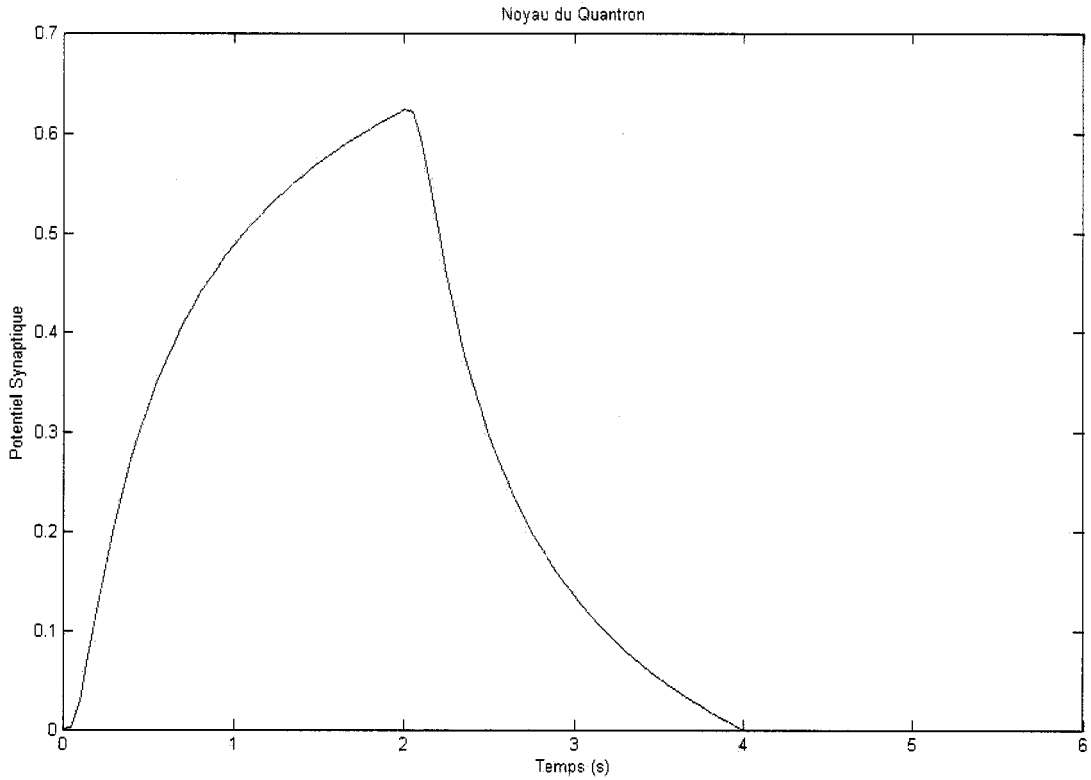


Figure 1.6 : Noyau du quantron ou $\varphi(t)$

Tel que vu pour le modèle biologique, les potentiels électriques de chaque dendrite se cumulent au corps du neurone. Mathématiquement, cette opération se traduit par une sommation temporelle, il en résulte la variation de voltage cumulative ($V.V.C.$) :

$$V.V.C. \equiv \sum_{i=0}^{\infty} \varphi(t - ix) \tag{1.5}$$

où x^{-1} est la fréquence pré-synaptique des signaux d'entrées. Contrairement au perceptron où l'entrée correspond à l'amplitude du signal d'entrée, l'entrée du quantron est la fréquence d'arrivée. En ajoutant un délai θ qui représente la période de temps avant l'arrivée du premier potentiel d'action pré-synaptique, on obtient la réponse totale *R(t)* du neurone post-synaptique :

$$R(t) = \sum_{i=0}^{\infty} \vartheta(t - \theta - ix) \tag{1.6}$$

Finalement, pour que le quantron émette un potentiel d'action à la sortie, la réponse totale doit être supérieure au seuil critique de dépolarisation Γ. Dans le cas contraire où la réponse totale est insuffisamment élevée, il y a absence de transmission de signal et il en résulte une impossibilité de propagation de l'information au-delà de la synapse. La sortie du quantron se traduit mathématiquement par :

$$\Omega = \begin{cases} 0 & si \quad \sum_{j=1}^{n} R(t) < \Gamma \qquad \forall t \in [0, \infty) \\ \alpha & \quad Sinon \end{cases} \tag{1.7}$$

où n est le nombre d'entrées synaptiques et où α représente le premier instant de la jonction du seuil critique et de la réponse totale :

$$\alpha = \inf\{t > 0 : R(t) = \Gamma\} \tag{1.8}$$

La figure 1.7 illustre les différents signaux menant à la sortie du quantron. Les signaux ayant un trait plein montre le voltage cumulatif de chacune des trois synapses. Un des signaux est inhibiteur tandis que les deux autres sont excitateurs. En sommant ces signaux, on obtient la réponse totale du quantron qui est représentée par le trait en pointillé fin. La valeur α représentant la jonction entre la réponse totale et le seuil critique est illustrée par un cercle. Le trait en pointillé grossier représente le seuil critique du quantron.

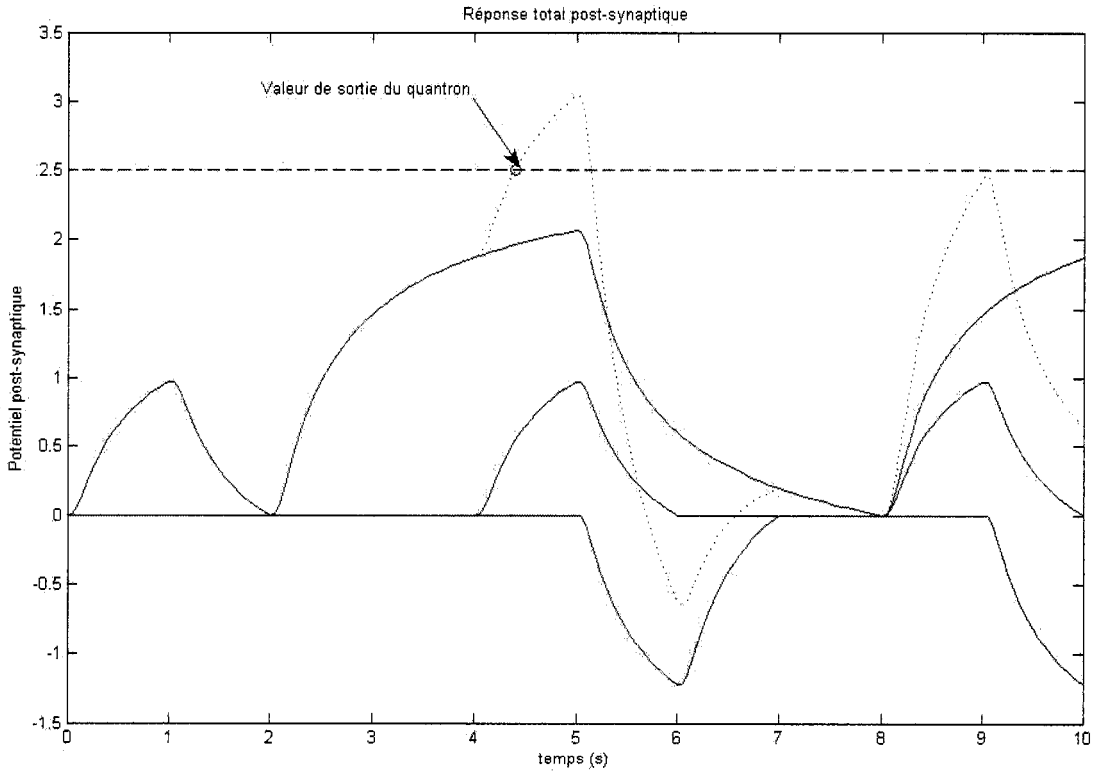


Figure 1.7: Représentation des différents signaux internes au quantron

À partir des dernières équations, nous obtenons le comportement entre les entrées et la sortie du quantron tel qu'illustré à la figure 1.7. Il a été prouvé par Labib (1999) qu'un seul quantron permet de résoudre le problème classique du XOR alors que trois perceptrons sont nécessaires pour la résolution du même problème. Ce qui signifie que le quantron possède un grand pouvoir discriminant, caractéristique primordiale pour effectuer des classifications non linéaires. Ce nouveau modèle de neurone possède de grandes possibilités, mais pour les exploiter un algorithme d'apprentissage doit être implanté. Le perceptron est la preuve qu'un algorithme d'apprentissage est nécessaire pour le développement d'un modèle de neurone. En effet, tandis que Minsky et Papert en 1988 condamnaient à mort le perceptron, la rétropropagation de l'erreur redonnait vie à celui-ci en lui permettant de devenir aujourd'hui le réseau de neurones le plus populaire.

CHAPITRE 2 : ALGORITHME D'APPRENTISSAGE

Chaque jour, des milliers d'enfants dans le monde entier se rendent à l'école pour apprendre et comprendre les différentes matières que doivent leur transmettre les professeurs. Le processus d'apprentissage est un processus complexe au sein duquel le professeur joue un rôle important. Son rôle est de superviser l'apprentissage et de tester les connaissances de ses élèves. Ceux-ci reçoivent la matière et doivent l'assimiler pour la comprendre et la mémoriser. L'algorithme d'apprentissage, supervisé d'un réseau de quantrons, suit le même processus d'apprentissage que l'élève.

L'apprentissage le plus approprié pour la résolution de problèmes de classification ou de régression par le réseau de quantrons est un apprentissage supervisé, tel que l'élève et son professeur. Les trois étapes de l'apprentissage sont illustrées à la figure 2.1. La première étape est de propager le signal à travers le réseau multicouche pour obtenir une valeur de sortie. Ensuite, cette valeur de sortie est comparée à la valeur désirée et la différence entre les deux valeurs constitue l'erreur. Finalement, en fonction de l'erreur, l'algorithme d'apprentissage modifie les paramètres synaptiques du neurone. Tel que le professeur guidant ses élèves pour l'obtention des bonnes réponses, l'algorithme d'apprentissage guide le réseau multicouche pour l'obtention de la valeur désirée par la modification des paramètres synaptiques. Ces modifications, qui surviennent à chaque erreur, permettent au réseau multicouche de mémoriser et de comprendre le problème pour lequel il doit

s'adapter. Une des difficultés du processus d'apprentissage est de l'arrêter au moment opportun pour que le réseau multicouche comprenne le problème à résoudre. Si l'arrêt n'est pas effectué, le réseau multicouche mémorise les entrées et ce phénomène se nomme le « surapprentissage », la difficulté classique des réseaux de neurones artificiels.

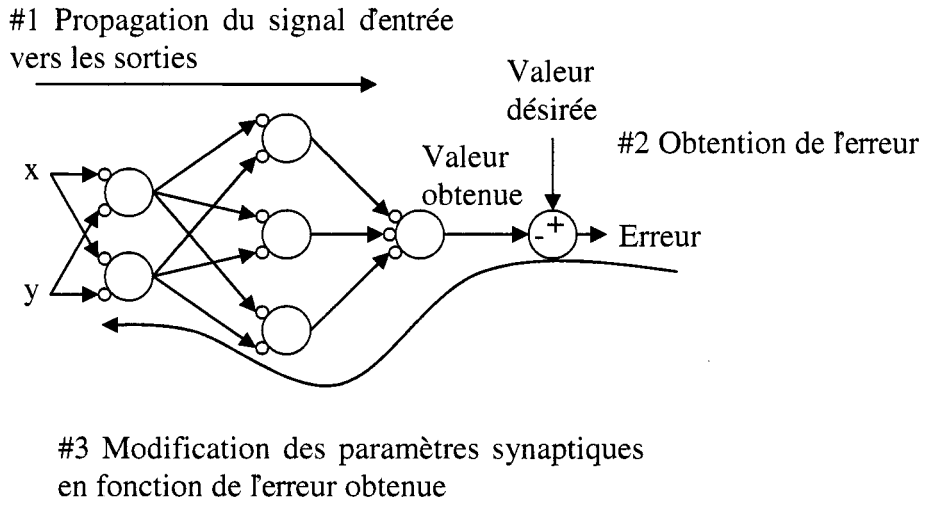


Figure 2.1 : Les étapes de l'apprentissage supervisé

2.1 Description

Tel que nous l'avons expliqué, pour obtenir des performances maximales, un algorithme d'apprentissage doit être implanté au réseau multicouche de quantrons. L'algorithme que nous présentons permet un apprentissage supervisé pour toute configuration de réseau de

neurones. La grande difficulté à surmonter est la possibilité d'absence de conduction de l'information au sein du quantron. Cette difficulté se révèle un avantage pour des problèmes de classification car la conduction d'information peut correspondre à une classe et la non conduction à l'autre. Le tableau 2.1 énumère les quatre cas différents d'apprentissage.

Tableau 2.1 : Énumération des quatre cas d'apprentissage du quantron

	Valeur désirée	Valeur obtenue	Opération à effectuer
Cas #1	Non conduction	Non conduction	Aucune modification
Cas #2	Conduction	Non conduction	Exciter le réseau
Cas #3	Non conduction	Conduction	Inhiber le réseau
Cas #4	Conduction	Conduction	Rétropropagation conventionnelle

Il est important de noter que notre algorithme d'apprentissage ne s'applique pas seulement au quantron, mais à tous les modèles possédant les mêmes caractéristiques de conduction d'information. À la section 2.2, l'algorithme d'apprentissage sera testé sur un modèle de quantron simplifié afin de le valider.

Pour les cas du tableau 2.1, des opérations sont à effectuer et chacune d'elles est présentée ci-après.

2.1.1 Cas #1 : Absence de modification

Ce cas est trivial puisque la valeur obtenue est égale à celle désirée, l'action portée sera de ne rien modifier aux paramètres synaptiques.

2.1.2 Cas #2 : Excitation du réseau

Ce cas survient lorsque l'on désire qu'il y ait conduction, mais que le circuit ne propage pas l'influx. La figure 2.2 présente ce cas.

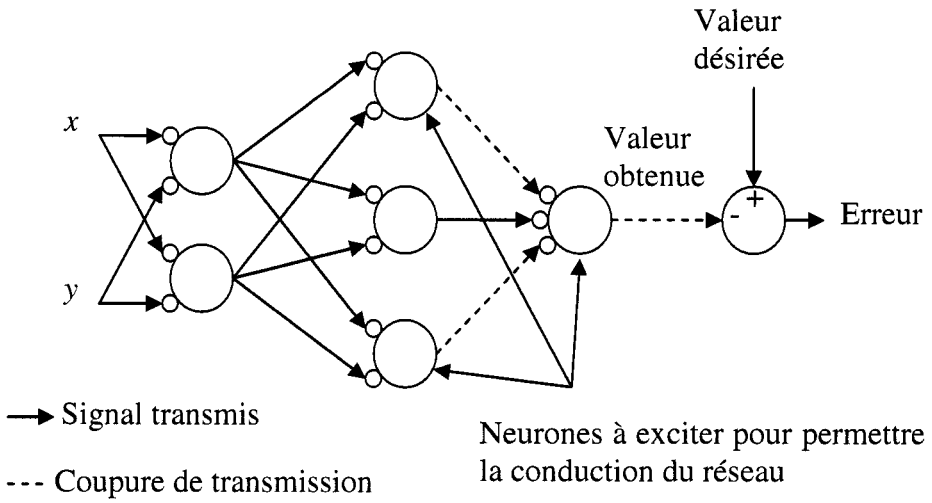


Figure 2.2 : Exemple de réseau à exciter

Le signal ne peut se propager jusqu'à la sortie puisque certains neurones ne conduisent pas. Il s'agit alors de permettre la conduction du réseau en stimulant les neurones qui ne conduisent pas.

Stimuler ces neurones équivaut à la modification des paramètres synaptiques. Cette modification s'effectue en tenant compte du gradient de sortie du neurone en fonction du paramètre synaptique à modifier. Le gradient apporte une information locale sur la direction des mises à jour à effectuer pour stimuler le neurone. La mise à jour des paramètres s'effectuera dans le sens du gradient. On obtient alors la règle d'apprentissage suivante :

$$P_{i,k} = P_{i,k-1} + \eta \frac{\delta y}{\delta P_{i,k-1}} \tag{2.1}$$

où η est un taux d'apprentissage et $P_{i,k-1}$ est un des paramètres synaptiques du neurone. Ces paramètres sont déterminés à partir du noyau utilisé dans le quantron. Dans notre cas, le noyau utilisé est le noyau rectangulaire et les paramètres sont ω, θ et S. Cependant une multitude de noyaux pourraient être implantés dans le quantron. Trois types de noyaux sont illustrés à la figure 2.3: le noyau standard développé par Labib (2000), le noyau rectangulaire et le noyau triangulaire. L'avantage de pouvoir implanter une multitude de noyaux différents est l'obtention d'une variété de quantrons pouvant résoudre certains types de problèmes selon le noyau implanté.

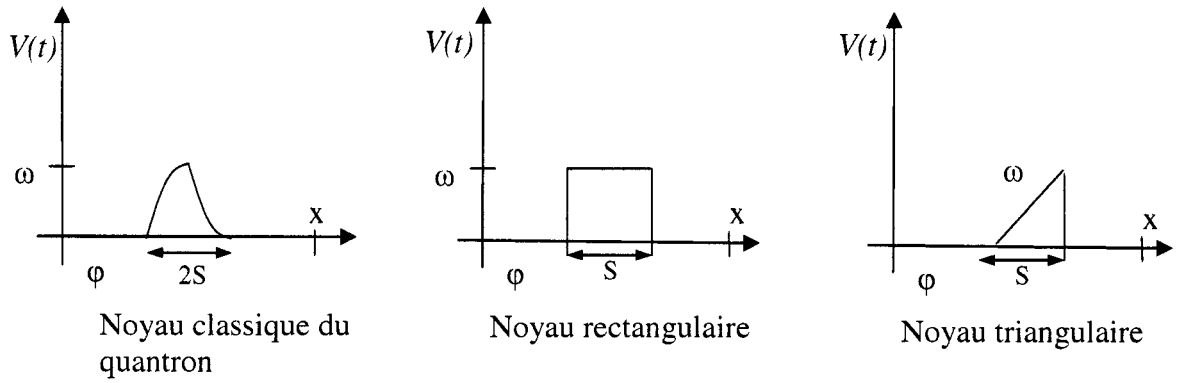


Figure 2.3 : Différents noyaux implantable au quantron.

2.1.3 Cas #3 : Inhibition du réseau

Ce cas est l'opposé du cas précédent. Il survient lorsque l'on désire qu'il n'y ait pas de conduction, mais que le circuit propage l'influx. La figure 2.4 illustre ce cas. Il s'agit alors d'arrêter la conduction du réseau en inhibant les neurones qui conduisent.

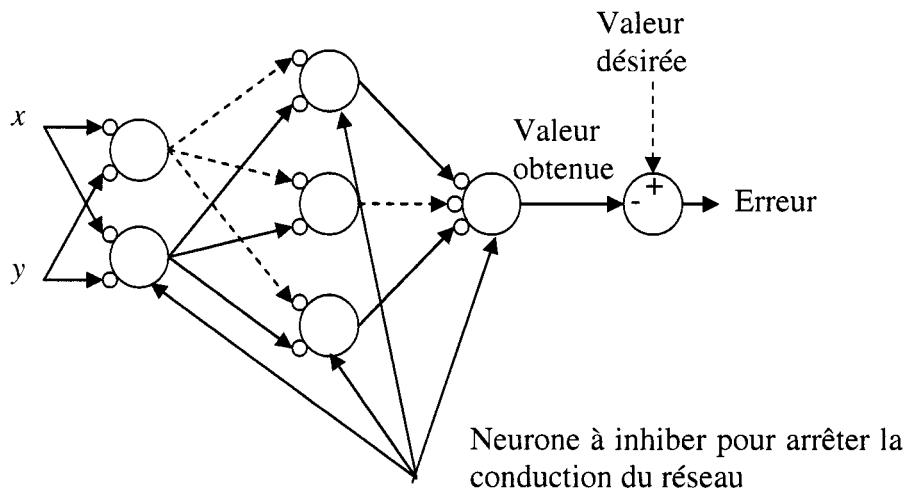


Figure 2.4 : Exemple de réseau à inhiber

Inhiber ces neurones équivaut à diminuer leur excitation en modifiant les paramètres synaptiques. Cette modification s'effectue en tenant compte du gradient de sortie du neurone en fonction du paramètre synaptique à modifier. Contrairement au cas précédent, pour inhiber le neurone la mise à jour des paramètres s'effectuera dans le sens contraire du gradient, ce qui s'exprime par un négatif. On obtient alors la règle d'apprentissage suivante :

$$P_{i,k} = P_{i,k-1} - \eta \frac{\delta y}{\delta P_{i,k-1}} \tag{2.2}$$

où η est un taux d'apprentissage et $P_{i,k-1}$ est un des paramètres synaptiques du neurone.

2.1.4 Cas #4 : Rétropropagation de l'erreur

L'algorithme d'apprentissage de rétropropagation de l'erreur a été développé pour l'apprentissage supervisé du réseau multicouche de perceptrons. Le but de cet algorithme est de rétropropager l'erreur à travers le réseau. Pour rétropropager l'erreur, la dérivation en chaîne permet de déterminer les gradients d'erreur nécessaires à la mise à jour des paramètres synaptiques. L'algorithme de rétropropagation de l'erreur pour un réseau multicouche de perceptrons est détaillé dans le tableau 2.2.

La rétropropagation de l'erreur nécessite une valeur de sortie ainsi qu'une valeur désirée. Ces deux valeurs forment l'erreur de sortie. Le quatrième cas de l'algorithme d'apprentissage assure la définition de l'erreur. Lors de cette étape, les neurones qui ne conduisent pas sont éliminés temporairement puisque la rétropropagation de l'erreur ne s'effectue qu'à travers les neurones qui transmettent le signal. La figure 2.5 présente la modification de la structure du réseau de neurones où les neurones rayés représentent ceux qui ne conduisent pas. La rétropropagation s'effectuera donc dans le réseau résultant.

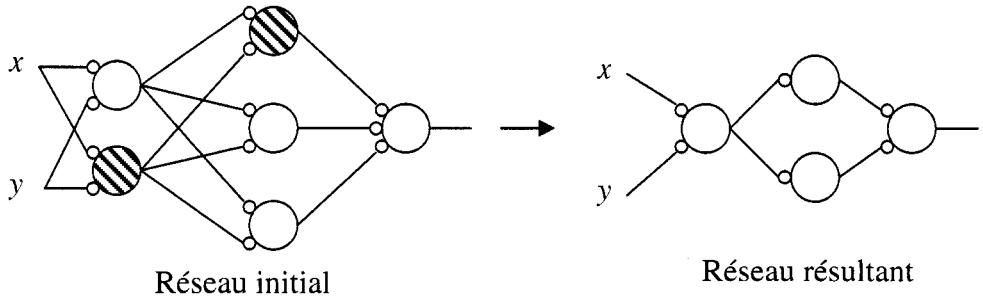


Figure 2.5 : Modification de la structure du réseau de neurone lors de la rétropropagation de l'erreur

Avant de présenter la généralisation de l'algorithme d'apprentissage de rétropropagation de l'erreur, dans la figure 2.6, nous présentons les conventions que nous suivrons pour l'explication de l'algorithme. Dans cette figure, le réseau de neurones présenté possède trois couches, mais il est à noter que l'algorithme se généralise à un réseau variant d'une couche à n couches. La variable x_n représente le vecteur d'entrée de la couche n tandis que la variable y_n représente la sortie de la couche n.

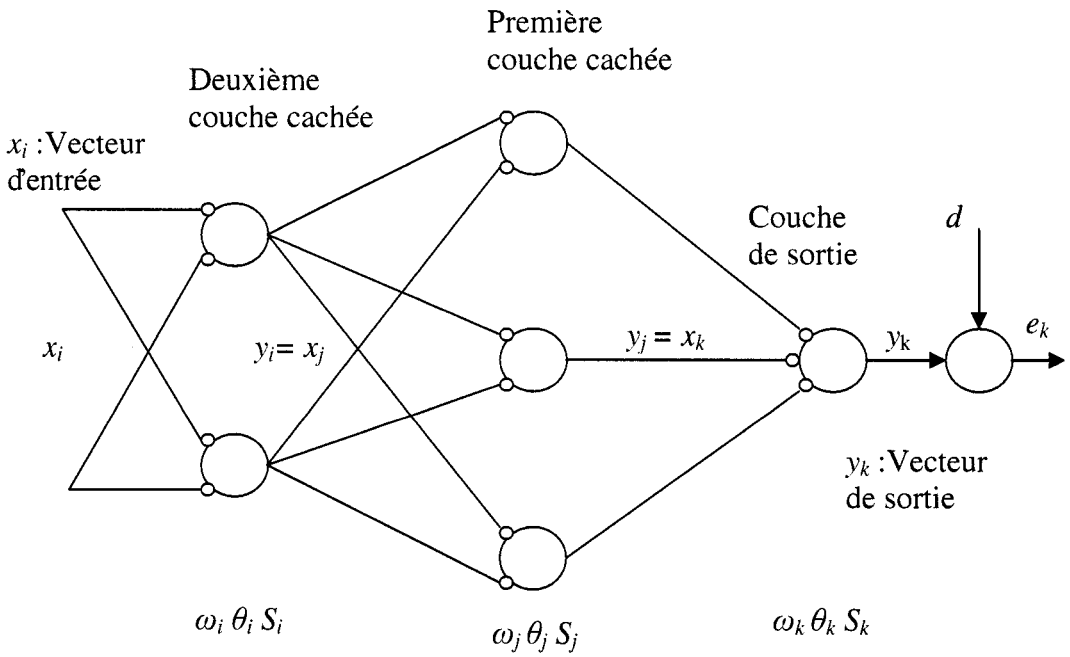


Figure 2.6 : Structure et conventions nécessaires à l'implantation de l'algorithme de rétropropagation de l'erreur

Le tableau 2.2 récapitule l'algorithme d'apprentissage pour le perceptron tout en montrant l'algorithme d'apprentissage implanté pour le neurone général. L'algorithme d'apprentissage est initialement développé pour le quantron. Cependant, nous l'avons généralisé pour le neurone possédant la caractéristique de conduction et de non conduction de l'information, c'est-à-dire le neurone général. Cette explication est sommaire de l'algorithme de rétropropagation puisqu'il existe une quantité énorme de référence tel que Haykin (1999). L'important à retenir sur l'algorithme de rétropropagation de l'erreur est que l'on modifie les paramètres synaptiques du réseau en propageant l'erreur de la sortie du réseau vers les entrées. Cette rétropropagation

s'effectue à partir des gradients obtenus de la fonction entre les entrées et la sortie du neurone. Ce gradient d'erreur est obtenu en effectuant une dérivée en chaîne, il en résulte que les poids synaptiques qui contribuent à engendrer une erreur importante se verront modifier de manière plus significative que les poids qui ont engendré une erreur marginale.

Tableau 2.2 : Comparaison entre l'algorithme de rétropropagation de l'erreur pour le perceptron et le neurone général

	Perceptron	Neurone général
Fonction d'entrées et de sortie	$v=\sum_{i=1}^{n}\omega_i x_i \qquad y=\varphi(v)=\frac{1}{1+e^{-v}}$	$y=f(x,P_0,\ldots,P_t)$
Dérivée des fonctions	$\varphi'(v)=y(1-y)$	$\frac{\delta y}{\delta x}=\frac{\delta f(x,P_0,\ldots,P_t)}{\delta x}$ $\frac{\delta y}{\delta P}=\frac{\delta f(x,P_0,\ldots,P_t)}{\delta P}$
Mise à jour des paramètres	$\omega=\omega-\eta\delta_{couche}x$	$P=P-\eta\frac{\delta E}{\delta P}$
Gradient d'erreur		
Couche de sortie k	$\delta_k=-(d_k-y_k)\varphi_k'$	$\frac{\delta E}{\delta P_k}=-(d_k-y_k)\frac{\delta y_k}{\delta P_k}$ $\delta_k=-(d_k-y_k)$
Couche cachée j	$\delta_j=-\left(\sum_k \delta_k\omega_k\right)\varphi_j'$	$\frac{\delta E}{\delta P_j}=-\sum_k\left(\delta_k\frac{\delta y_k}{\delta x_k}\right)\frac{\delta y_j}{\delta P_j}$ $\delta_j=-\sum_k\left(\delta_k\frac{\delta y_k}{\delta x_k}\right)$
Couche d'entrée i	$\delta_i=-\left(\sum_j \delta_j\omega_j\right)\varphi_i'$	$\frac{\delta E}{\delta P_i}=-\sum_j\left(\delta_j\frac{\delta y_j}{\delta x_j}\right)\frac{\delta y_i}{\delta P_i}$ $\delta_i=-\sum_j\left(\delta_j\frac{\delta y_j}{\delta x_j}\right)$

2.1.5 Généralisation à plusieurs sorties

Jusqu'à présent l'algorithme d'apprentissage présenté modifie le réseau de neurones à partir de l'erreur d'un seul neurone de sortie. Les problèmes à résoudre nécessite souvent des réseaux ayant plusieurs sorties. La généralisation de l'algorithme est simple, il s'agit d'effectuer l'algorithme à chacune des sorties successivement. La figure 2.7 illustre un exemple d'apprentissage de réseau comportant plusieurs sorties. Sur le premier neurone de sortie, nous exciterons les neurones des couches cachées qui ne conduisent pas ainsi que le neurone de sortie puisque l'on obtient un cas d'excitation du réseau à la sortie de ce neurone. Sur le second neurone de sortie, nous effectuerons la rétropropagation de l'erreur puisque l'on obtient un cas de rétropropagation du réseau à la sortie de ce neurone.

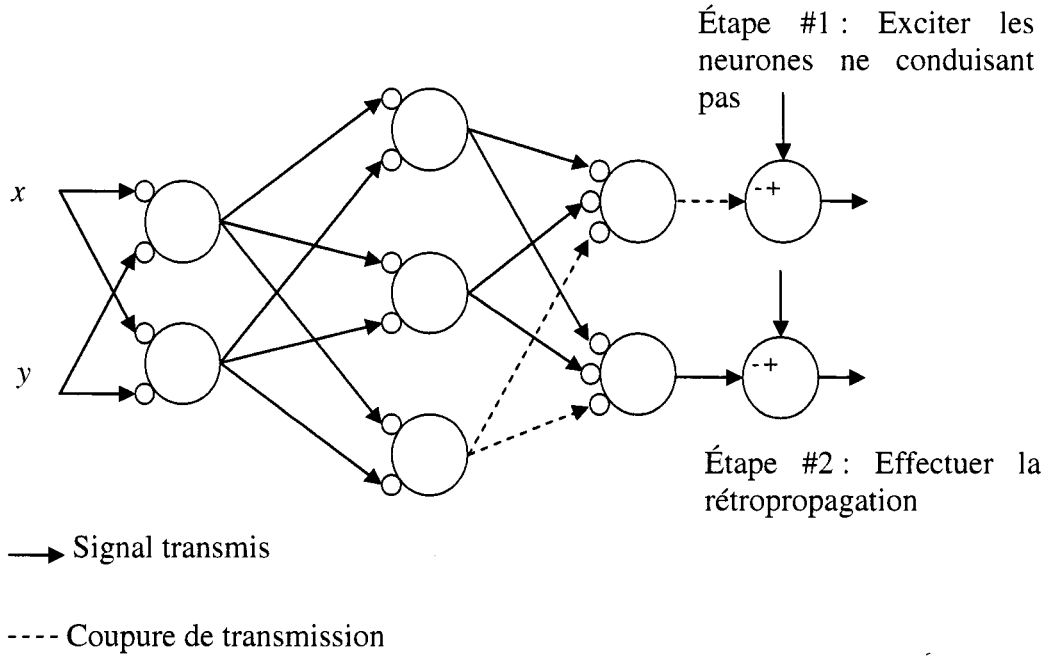


Figure 2.7 : Exemple de l'algorithme d'apprentissage pour un réseau comportant plusieurs sorties

2.2 Expérimentation avec un modèle de quantron simplifié

Ce nouvel algorithme d'apprentissage apporte une solution au problème de conduction du quantron. Avant d'entreprendre l'implantation de l'algorithme sur un réseau multicouche de quantrons, nous nous sommes assurés qu'il permettait un apprentissage adéquat. Pour tester l'algorithme d'apprentissage, nous avons créé un modèle simplifié de quantron se basant sur les caractéristiques du quantron et nous l'avons testé avec notre algorithme.

2.2.1 Modèle de quantron simplifié

Le quantron simplifié se base sur le quantron. Les caractéristiques recherchées sont une coupure de transmission ainsi qu'une réponse non linéaire lors de la transmission. Nous avons modélisé ce comportement par l'équation suivante :

$$y = \begin{cases} S(\omega, x) & si \quad T(\omega, x) > K \\ Non \quad Transmission & Sinon \end{cases} \tag{2.3}$$

où K est une constante. La fonction $T(\omega,x)$ correspond à la fonction permettant la transmission. Si sa valeur est supérieure à une certaine constante K, la transmission a lieu. La valeur de sortie est donnée par la fonction $S(\omega,x)$. Dans le cas contraire, la transmission est coupée. Ce modèle simplifié représente sommairement les caractéristiques de sortie du quantron, mais il est suffisant pour tester l'algorithme d'apprentissage. La figure 2.8 illustre la représentation du neurone.

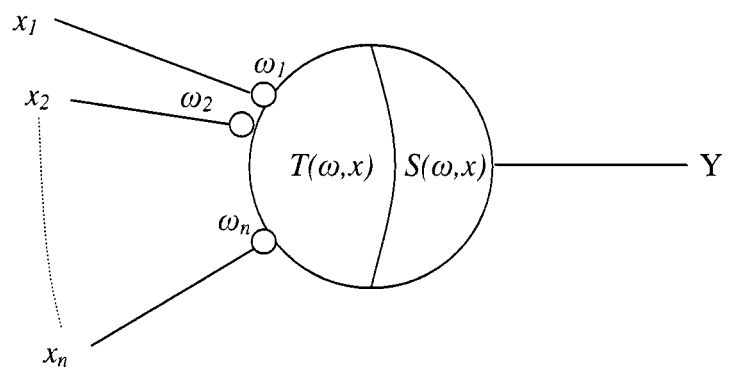


Figure 2.8 : Représentation du modèle simplifié

Trois modèles simplifiés sont testés. Ils sont obtenus à partir des différentes combinaisons de fonctions de transmission et de sortie qui sont présentées dans le tableau 2.3.

Tableau 2.3: Énumération des différents modèles utilisés

	Fonction pour la transmission du neurone $T(\omega,x)$	Fonction de sortie du neurone $S(\omega,x)$
Quadratique	$T(\omega,x)=\sum_{i=1}^{n}\omega_i^2 x_i$	$S(\omega,x)=\tanh\left(\sum_{i=1}^{n}\omega_i x_i\right)$
Linéaire	$T(\omega,x)=\sum_{i=1}^{n}\omega_i x_i$	$S(\omega,x)=\tanh\left(\sum_{i=1}^{n}\omega_i x_i\right)$
Sinusoïdal	$T(\omega,x)=\sin\left(\sum_{i=1}^{n}\omega_i x_i\right)$	$S(\omega,x)=\tanh\left(\sum_{i=1}^{n}\omega_i x_i\right)$

Ces trois modèles sont nommés selon leur fonction de transmission. Il en résulte le quantron simplifié quadratique, le quantron simplifié linéaire et le quantron simplifié sinusoïdal.

Maintenant que $T(\omega,x)$ et $S(\omega,x)$ sont définis, nous obtenons les gradients nécessaires pour l'inhibition ou l'excitation du neurone. Les différents gradients sont énumérés dans le tableau 2.4.

Tableau 2.4 : Énumération des différents gradients nécessaires pour l'inhibition ou l'excitation du neurone

	Gradient nécessaire pour inhiber ou pour exciter le neurone
Quadratique	$\frac{\delta y}{\delta \omega_i} = 2\omega_i x_i$
Linéaire	$\frac{\delta y}{\delta \omega_i} = x_i$
Sinusoïdale	$\frac{\delta y}{\delta \omega_i} = x_i \cos\left(\sum_{i=1}^{n} \omega_i x_i\right)$

La fonction $S(\omega,x)$ choisie est la tangente hyperbolique. Grâce à cette fonction de sortie, la rétropropagation de l'erreur revient à un cas classique avec un perceptron pour le réseau résultant, le développement de cet algorithme d'apprentissage de rétropropagation de l'erreur est détaillé dans le tableau 2.2.

2.2.2 Résultat

L'aspect que l'on veut tester avec le quantron simplifié est la convergence de l'apprentissage. La méthodologie de test choisie consiste à résoudre un problème de

classification de deux classes. Les deux classes ont été définies comme étant que la sortie du réseau conduise et l'autre comme étant que la sortie du réseau ne conduise pas. Pour classifier ces deux classes, un seul neurone de sortie est nécessaire.

Pour permettre une bonne visualisation du problème, les réseaux utilisés possèdent deux entrées donc deux coordonnées cartésiennes x et y, il est donc possible de montrer les résultats sur un graphique en deux dimensions. La figure 2.9 est un exemple de représentation graphique obtenue.

La première étape consiste à générer des données que l'on visualise sous la forme d'une image. Nous les générons en présentant les coordonnées cartésiennes (x,y) aux entrées d'un réseau multicouche de quantrons simplifiés. Le nombre de couche ainsi que le nombre de neurones par couche sont obtenus de manière aléatoire. Nous répétons cette opération pour toutes les coordonnées cartésiennes. Les coordonnées cartésiennes forment un quadrillage régulier. Lorsque le réseau conduit, nous marquons cette coordonnée par un cercle. Dans le cas contraire, nous laissons un espace blanc à cette coordonnée. Dans la figure 2.9, les cercles sont les valeurs que l'on désire qu'il y est conduction, l'espace en blanc sont les coordonnés où l'on désire qu'il n'y est pas de conduction. Maintenant que l'on détient les coordonnées des deux classes que l'on désire, il s'agit d'entraîner un autre réseau avec l'algorithme et d'observer comment l'algorithme d'apprentissage permet d'obtenir la bonne solution. Ce nouveau réseau est initialisé de manière aléatoire. Nous effectuons l'entraînement à partir de la valeur désirée et de la

valeur obtenue pour une coordonnée *(x,y)*. Nous répétons l'opération pour tous les points. Il s'agit d'une époque. À la fin de cette époque, nous superposons les deux images soit l'image désirée constituée des cercles et d'espaces blancs tandis que l'image obtenue avec le réseau entraîné est constituée de croix et d'espaces blancs. Les endroits où une croix n'est pas superposée à un cercle, il s'agit d'une erreur de classification, il est possible alors de connaître le nombre d'erreurs.

En premier lieu, on observe le comportement de l'algorithme d'apprentissage pour un exemple généré à partir du modèle de neurone simplifié quadratique. La figure 2.9 présente les images générées par les réseaux de neurones avant l'apprentissage. L'image formée de ronds correspond à l'image désirée tandis que l'image créée par les croix correspond à l'image obtenue.

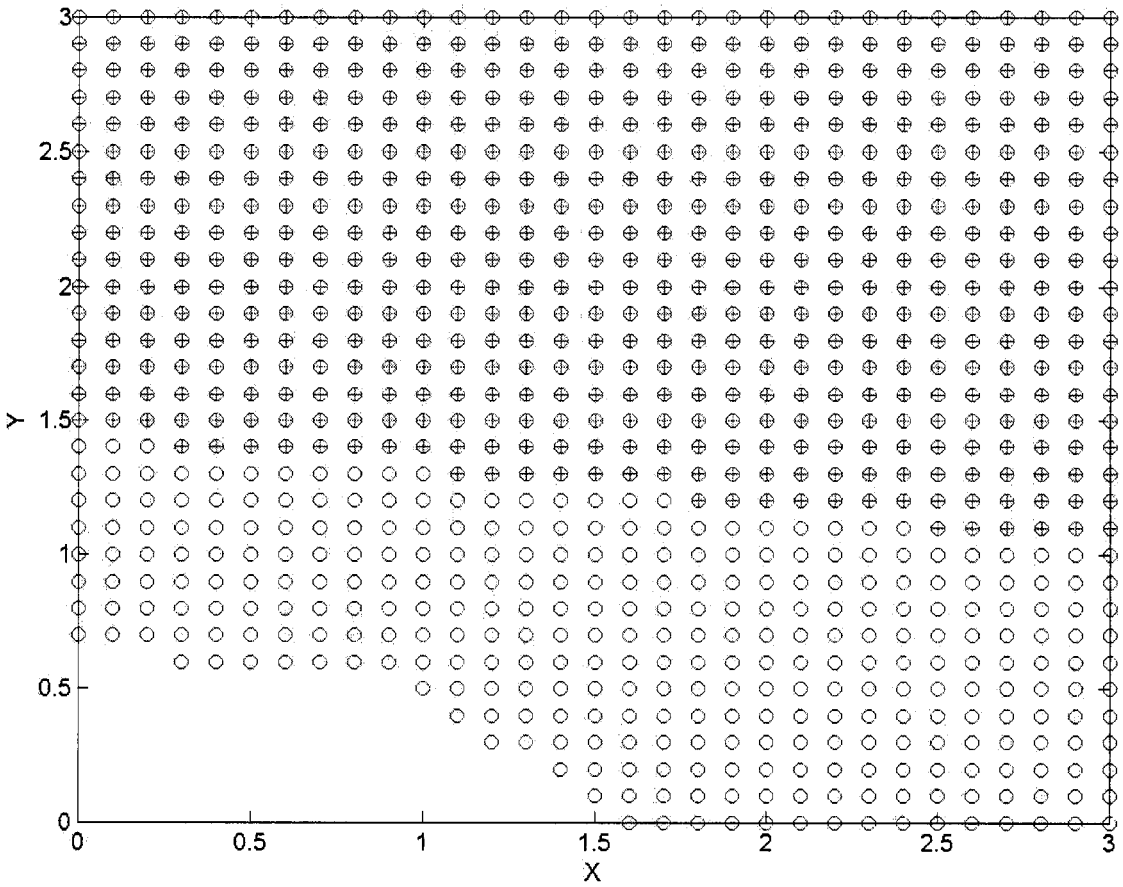


Figure 2.9 : Superposition de l'image désirée (rond) et de l'image obtenue (croix) avant l'apprentissage pour un réseau de quantrons simplifiés quadratiques

Ensuite, à la figure 2.10 on visualise l'image obtenue suite à un apprentissage de dix époques superposée à l'image désirée. Une époque correspond à la présentation de tous les points d'entrées durant l'apprentissage (Haykin (1999)). Ces images sont représentées respectivement par des croix et des ronds.

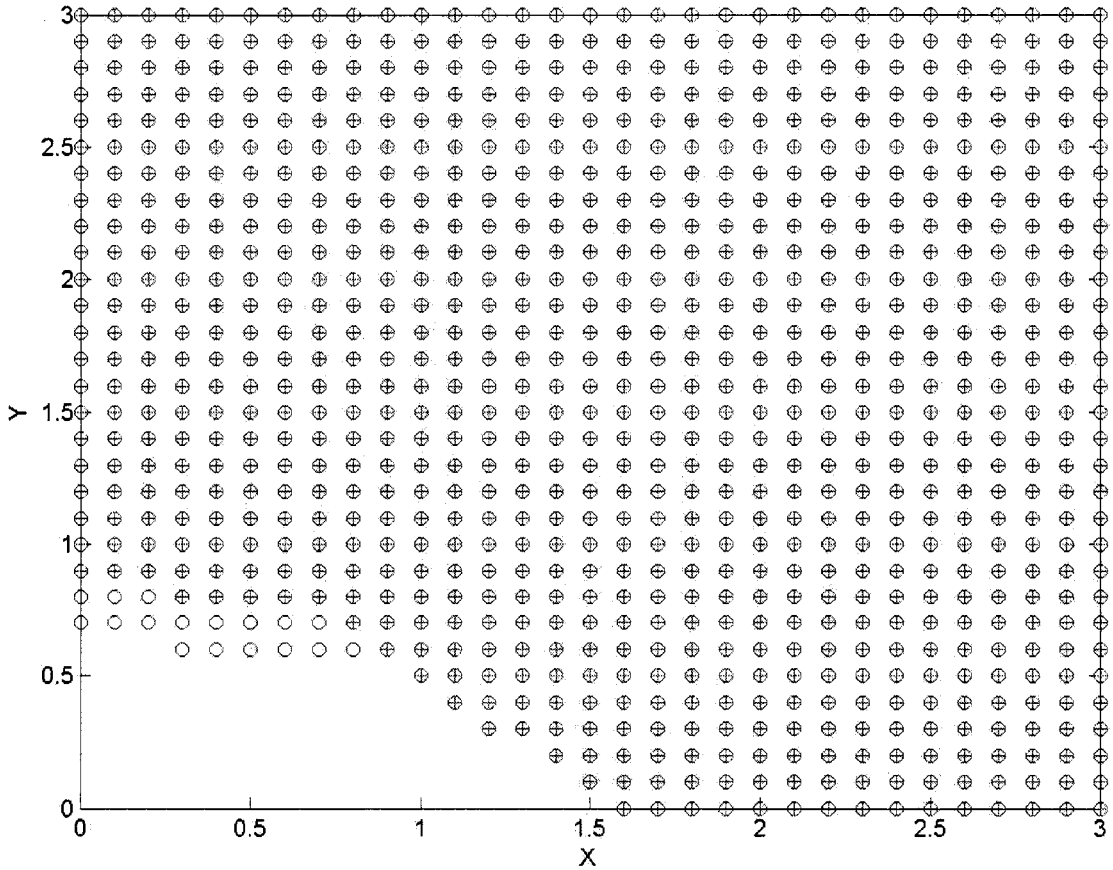


Figure 2.10 : Superposition de l'image désirée (rond) et de l'image obtenue (croix) après un apprentissage de dix époques

L'algorithme d'apprentissage converge vers l'image désirée, les résultats sont affichés à la figure 2.11. La courbe de l'erreur en fonction des époques correspond au même type de courbe obtenue pour l'apprentissage du perceptron, c'est-à-dire une courbe asymptotique.

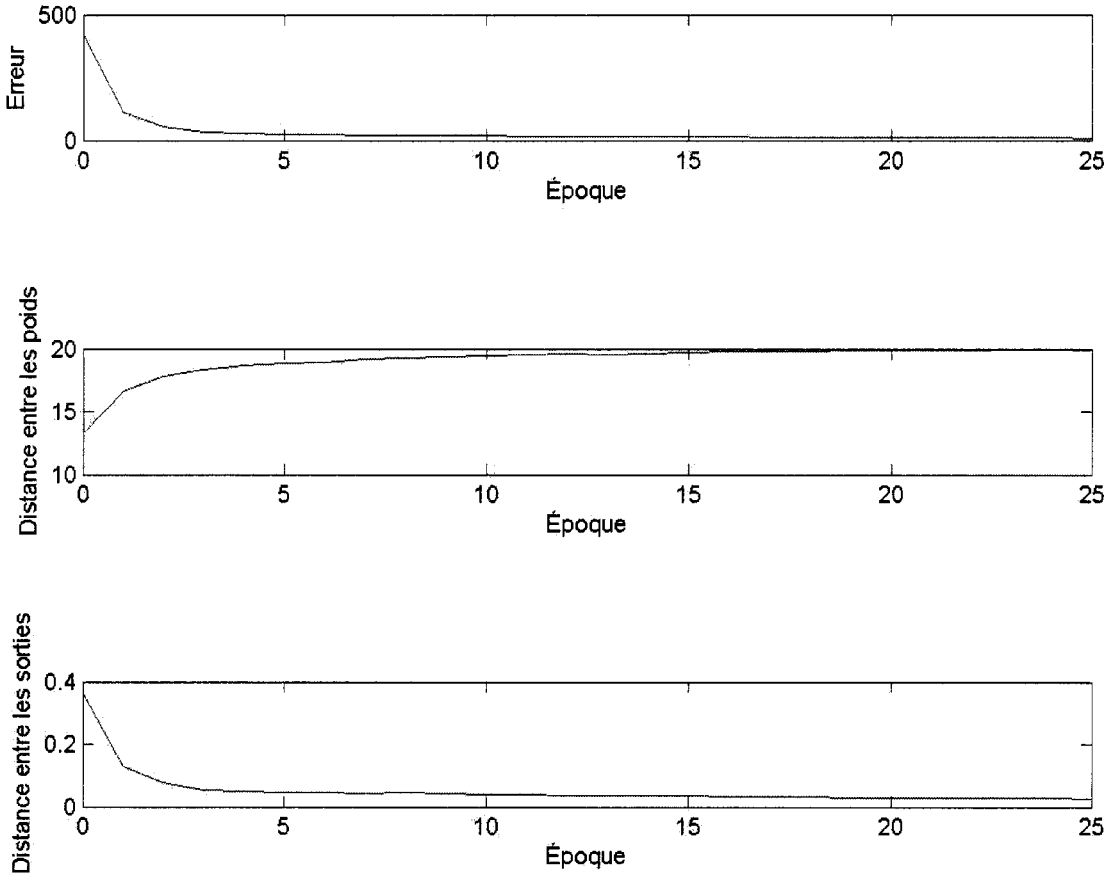


Figure 2.11 : Résultat de la simulation avec le quantron simplifié quadratique

L'algorithme d'apprentissage a convergé dans le cas illustré ci-dessus, mais qu'en sera-t-il lors de plusieurs essais? Puisque un seul cas ne peut être concluant, l'opération a été répétée pour une centaine de réseaux différents générés aléatoirement pour les trois modèles. Le critère d'arrêt de l'apprentissage est défini par une cinquantaine d'époques maximales ou par cinq pour cent d'erreur par rapport au nombre d'erreurs initiales. Les résultats sont regroupés au tableau 2.5 pour les trois types de quantrons simplifiés qui ont été testés. Les résultats observés correspondent au nombre moyen d'époques avant

d'atteindre l'un des deux critères d'arrêt ainsi que le pourcentage d'erreurs après l'apprentissage.

Tableau 2.5 : Tableau de résultats pour une simulation avec 100 réseaux

	Moyenne du nombre d'époques	Réduction d'erreurs par rapport aux erreurs initiales
Linéaire	29,97	86,2%
Quadratique	25,42	89,3%
Sinusoïdal	Ne converge pas	

2.3.3 Discussion

En étudiant le tableau 2.5, on conclut que l'algorithme converge pour le modèle linéaire et quadratique du neurone tandis que le modèle sinusoïdal ne converge pas. La différence entre les résultats de convergence s'explique par une différence majeure entre les modèles. Les modèles linéaire et quadratique sont des fonctions croissantes dans l'intervalle de zéro à l'infini positif tandis que le modèle sinusoïdal ne l'est pas. Dans ce dernier cas, l'algorithme d'apprentissage est incapable de modifier les poids synaptiques convenablement à cause des minimums et des maximums locaux. Le modèle sinusoïdal est testé car le paramètre θ dans le quantron entraîne une certaine périodicité dans la réponse en décalant les signaux. Il est évident que l'algorithme d'apprentissage a des

difficultés avec le modèle sinusoïdal, une attention particulière doit être portée sur le paramètre θ lors de l'apprentissage du quantron.

CHAPITRE 3 : IMPLANTATION DE L'ALGORITHME D'APPRENTISSAGE DANS UN RÉSEAU DE QUANTRONS

Au chapitre précédent, l'algorithme d'apprentissage a été exposé. Nous avons testé son comportement avec un modèle de quantron simplifié. Puisque les résultats sont concluants, il s'agit maintenant d'implanter l'algorithme d'apprentissage au réseau multicouche de quantrons. L'implantation a été effectué avec MATLAB 7.0.1 en utilisant une programmation objet.

La tâche sera plus ardue pour le quantron qu'elle ne l'a été pour le quantron simplifié. Le quantron simplifié possède une fonction analytique entre les entrées et la sortie, mais ce n'est pas le cas pour le quantron. En effet, cette fonction est plutôt définie par la jonction entre la réponse totale et le seuil critique. Étant donné cette différence, les trois étapes suivantes sont nécessaires à l'implantation de l'algorithme d'apprentissage dans le réseau multicouche de quantrons. La première étape d'implantation de l'algorithme d'apprentissage est de trouver une méthode permettant la propagation avant du signal à travers le réseau multicouche. La seconde est d'obtenir toutes les équations de mise à jour des paramètres pour chacun des quatre cas de l'algorithme d'apprentissage. La dernière et troisième étape est de tester l'algorithme d'apprentissage.

3.1 Propagation avant du signal d'entrée

La propagation avant du signal d'entrée pour obtenir la valeur de sortie est la première étape du processus d'apprentissage. La valeur obtenue est comparée avec la valeur désirée et il en résulte une erreur. À ce moment, l'algorithme d'apprentissage modifie les paramètres synaptiques selon l'erreur obtenue. C'est pourquoi avant d'effectuer l'apprentissage, il est nécessaire d'obtenir la valeur de sortie du réseau de quantrons. Pour le quantron simplifié, cette étape est simple puisque la fonction entre les entrées et la sortie du neurone est définie analytiquement, ce qui n'est pas le cas pour le quantron.

L'une des méthodes simples pour déterminer la sortie du quantron consiste à observer la réponse totale à plusieurs instants t pour déterminer le moment où sa valeur dépasse le seuil critique. Cette méthode est inefficace et peu instructive à propos du neurone. Alors, il a plutôt été choisi d'effectuer une approximation. Cette approximation s'effectue par une série de Fourier tronquée. D'abord, une fenêtre de la réponse totale est sélectionnée, tel qu'illustré à la figure 3.1.

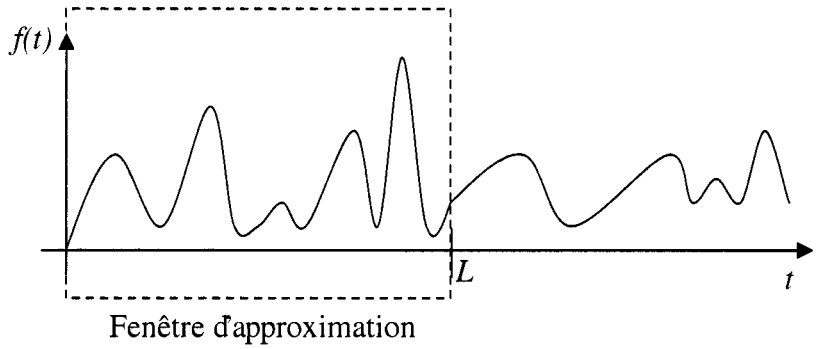


Figure 3.1 : Fenêtre de la fonction à approximer

Après avoir effectué la sélection de la fenêtre d'approximation, une extension paire est utilisée pour définir une nouvelle fonction, tel qu'illustré à la figure 3.2.

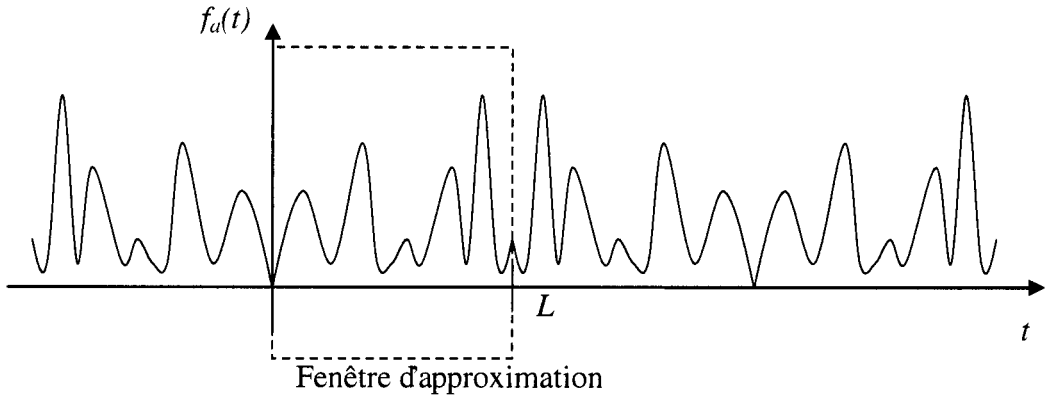


Figure 3.2 : Extension paire de la fenêtre d'approximation

L'utilisation d'une extension paire permet d'obtenir une approximation des séries de Fourier ne comportant que des fonctions cosinus, tel que montrées à l'équation 3.1.

$$f_a(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{n\pi t}{L}\right) \tag{3.1}$$

Les coefficients de la fonction d'approximation $f_a(t)$ sont trouvés par l'équation suivante :

$$\begin{aligned} A_0 &= \frac{1}{P}\int_0^P R(t)dt \\ A_n &= \frac{2}{P}\int_0^P R(t)\cos\left(\frac{n\pi t}{L}\right)dt \end{aligned} \tag{3.2}$$

Il est important de noter que les coefficients A_0 jusqu'à A_n dépendent des paramètres ω, θ et S du quantron ainsi que de la valeur d'entrée x. Cette relation est définie par un lemme.

<u>Lemme</u>

Soit une fonction $f_a(t)$ représentant l'approximation de la réponse totale du quantron

$$f_a(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos\left(\frac{n\pi t}{L}\right)$$

Les coefficients A_0 à A_n d'un quantron à noyau rectangulaire sont donnés par

$$A_0 = \frac{1}{L}\sum_{j=1}^{n}\sum_{i=0}^{m}\omega_j S_j$$

$$A_n = \sum_{j=1}^{n}\frac{2\omega_j}{n\pi}\sum_{i=0}^{m}\sin\left(\frac{n\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{n\pi(\theta_j + xi)}{L}\right)$$

Preuve

Le noyau utilisé pour l'expérimentation est un noyau rectangulaire tel qu'illustré à la figure 3.3. L'utilisation de ce noyau permet l'obtention de coefficients de la série de Fourier beaucoup plus simples. En simplifiant le noyau, l'implantation du quantron et de l'algorithme d'apprentissage est facilitée. Nous avons effectué cette simplification pour observer plus rapidement et facilement la réaction de notre réseau multicouche de quantrons face à notre algorithme d'apprentissage. Si les résultats permettent une minimisation de l'erreur, les objectifs seront atteints et l'étape ultérieure consistera à implanter le quantron avec son noyau original.

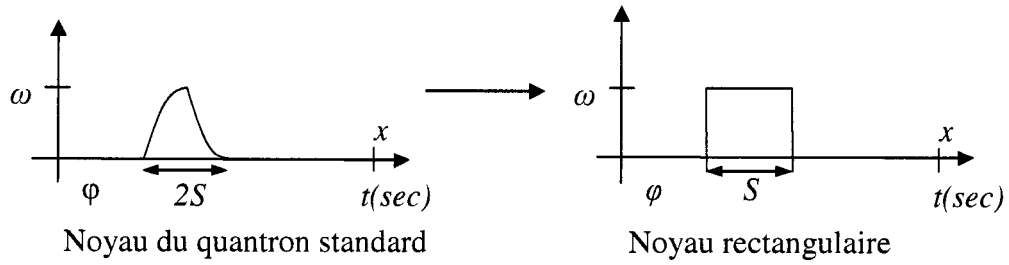


Figure 3.3 : Modification du noyau du quantron

Ce noyau s'écrit comme suit,

$$\begin{aligned} \varphi(t) &= \omega(u(t-S)-u(t)) \\ R(t) &= \sum_{i=0}^{\infty} \varphi(t-\theta-ix) \end{aligned} \tag{3.3}$$

En appliquant l'équation 3.2, on obtient les coefficients de la série de Fourier suivants pour un noyau rectangulaire,

$$\begin{aligned} A_0 &= \frac{1}{L}\sum_{j=1}^{n}\sum_{i=0}^{m}\omega_j S_j \\ A_n &= \sum_{j=1}^{n}\frac{2\omega_j}{n\pi}\sum_{i=0}^{m}\sin\left(\frac{n\pi(\theta_j+xi+S_j)}{L}\right)-\sin\left(\frac{n\pi(\theta_j+xi)}{L}\right) \end{aligned} \tag{3.4}$$

où n représente le nombre d'entrées du quantron et m représente le nombre de périodes de la fonction contenue dans la période L. □

Maintenant que l'on connaît les coefficients de l'équation $f_a(t)$, nous devons trouver la valeur de sortie du quantron. La valeur de sortie correspond à l'intersection de l'approximation de la réponse totale et du seuil critique, mais il est impossible de résoudre la fonction d'approximation actuelle puisque sa somme tend à l'infini. C'est pourquoi la fonction doit être tronquée à l'ordre i. Évidemment, une erreur est introduite, mais elle sera négligeable si l'on choisit un ordre assez élevé. L'équation de la fonction d'approximation devient alors :

$$f_{at}(t) = A_0 + A_1\cos(wt) + A_2\cos(2wt) + \ldots + A_{i-1}\cos((i-1)wt) + A_i\cos(iwt)$$
$$où \quad w = \pi/L \tag{3.5}$$

L'objectif est de résoudre cette dernière équation pour déterminer l'instant de conduction. À l'aide d'une transformation, nous modifierons les harmoniques en un polynôme de la fréquence fondamentale. Nous effectuerons cette transformation à l'aide du polynôme de Tchebychev :

$$\cos(nwt) = T_n(\cos(wt)) \tag{3.6}$$

Le polynôme de Tchebychev se trouve facilement de manière récurrente, soit par :

$$T_0(x) = 1, \qquad T_1(x) = x, \qquad T_{n+2}(x) = 2xT_{n+1}(x) - T_n(x) \qquad n \geq 0 \tag{3.7}$$

L'équation de la fonction d'approximation devient :

$$f_{at}(t) = A_0 + A_1T_1(\cos(wt)) + \ldots + A_{i-1}T_{i-1}(\cos(wt)) + A_iT_{i1}(\cos(wt)) \tag{3.8}$$

En organisant l'équation, nous obtenons :

$$f_{at}(t) = B_0 + B_1 \cos(wt) + B_2 \cos(wt)^2 + \ldots + B_{i-1} \cos(wt)^{i-1} + B_i \cos(wt)^i \tag{3.9}$$

où les coefficients B_0 à B_i sont des combinaisons linéaires de A_0 à A_i. En effectuant un changement de variables,

$$x = \cos(wt) \tag{3.10}$$

nous obtenons un polynôme de degré i :

$$\begin{aligned} x &= \cos(wt) \\ f_{at}(x) &= B_0 + B_1 x + B_2 x^2 + \ldots + B_{i-1} x^{i-1} + B_i x^i \end{aligned} \tag{3.11}$$

Pour connaître l'intersection entre la réponse totale et le seuil critique Γ, on doit soustraire le seuil critique à l'équation 3.11 et l'on obtient :

$$f_{at}(x) = B_0 + B_1 x + B_2 x^2 + \ldots + B_{i-1} x^{i-1} + B_i x^i = \Gamma \tag{3.12}$$

Les intersections de la réponse totale et du seuil critique correspondent aux zéros de l'équation 3.12 et les racines sont obtenues par une méthode de résolution numérique. La sortie du quantron correspond au zéro réel qui est supérieur à zéro, tout en étant le plus petit possible.

$$y = [y_1, y_2, ..., y_i] \tag{3.13}$$

Soit Z un sous-ensemble Y, où les élément du sous-ensemble sont les valeurs positives et réelles.

$$\alpha = \min[z_1, z_2, ..., z_n] \tag{3.14}$$

Cette méthode permet d'obtenir très rapidement la valeur de sortie du quantron, en plus d'obtenir une équation de la réponse totale qui contient plusieurs informations précieuses. Par exemple, à partir de l'approximation, il est possible de déterminer les maximums.

3.2 Implantation des quatre cas d'apprentissage spécifiques au quantron

Dans le chapitre précédent, nous avons présenté notre algorithme d'apprentissage et avons prouvé qu'il est en mesure d'effectuer l'apprentissage du quantron simplifié. Nous devons maintenant vérifier si notre algorithme d'apprentissage peut effectuer l'apprentissage du quantron. Pour ce faire, les quatre cas d'apprentissage doivent être implantés au quantron avant qu'il ne soit possible d'y appliquer l'algorithme d'apprentissage. Tel que nous le savons, dans le premier cas il y a absence de

modification. Pour l'implantation des trois autres cas, nous devons surmonter les difficultés suivantes. La mise à jour des paramètres lors de l'excitation ou de l'inhibition du réseau doit être effectuée pour les cas numéros 2 et 3 et les gradients d'erreur nécessaires à l'implantation de la rétropropagation de l'erreur doivent être obtenus pour le cas numéro 4.

3.2.1 Excitation du quantron

Ce cas survient lorsque l'on désire qu'il y ait conduction, mais que le circuit ne propage pas l'influx. Il s'agit alors de permettre la conduction du réseau en stimulant les neurones qui ne conduisent pas.

Stimuler un quantron signifie permettre à sa réponse totale de dépasser le seuil critique. Le moment de la réponse totale le plus propice pour atteindre le seuil critique est la valeur maximale de la réponse totale dans les limites de la fenêtre, ce qui est illustré à la figure 3.4 En stimulant le neurone pour que la valeur augmente, le neurone tend à conduire. L'équation 3.11 représente le polynôme de la réponse totale qui nous permet de trouver la valeur maximale de la réponse totale. Il est possible de trouver les maximums et les minimums du polynôme en le dérivant pour trouver ses zéros, c'est-à-dire l'équation suivante :

$$f'_{at}(x) = B_1 + 2B_2x + 3B_3x^2 + \ldots + (i-1)B_{i-1}x^{i-2} + iB_ix^{i-1} = 0 \qquad (3.15)$$

En résolvant l'équation 3.15 de manière analytique ou numérique, les maximums et les minimums sont obtenus. La prochaine étape est de déterminer le moment de la valeur maximale globale de la fenêtre d'approximation qui est définie par t_{max}.

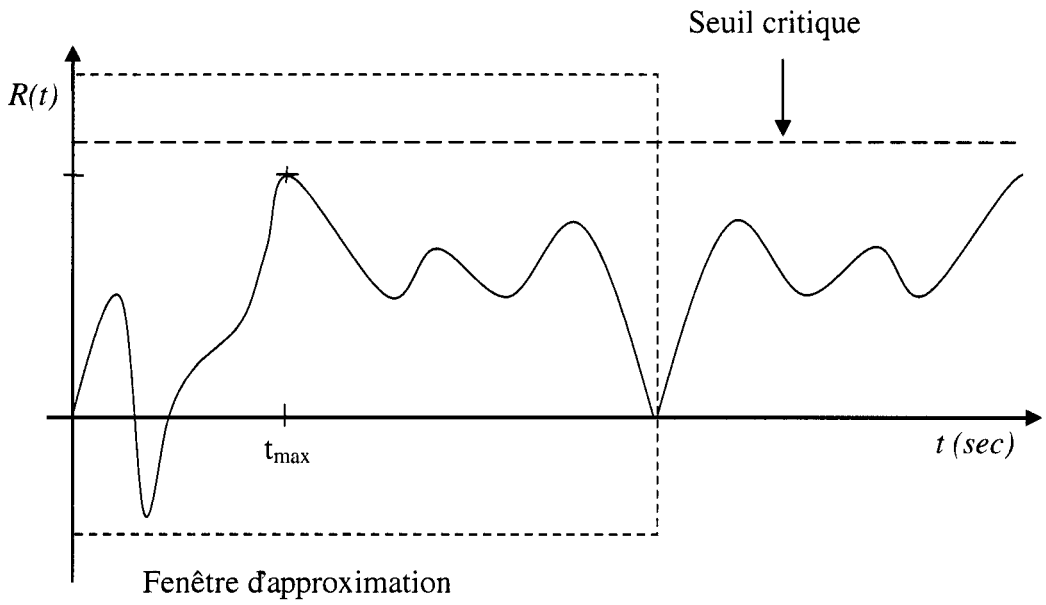


Figure 3.4 : Représentation de la réponse totale à exciter au moment le plus propice

En substituant t par t_{max} dans l'équation 3.5, on obtient une équation étant seulement fonction des paramètres ω_i, θ_i et S_i puisque les coefficients A_0 à A_n dépendent de ces trois paramètres.

$$f_{at}(\omega,\theta,S) = A_0 + A_1\cos(wt_{\max}) + \ldots + A_{i-1}\cos((i-1)wt_{\max}) + A_i\cos(iwt_{\max}) \tag{3.16}$$

où tous les coefficients de A_0 à A_n dépendent de ω, θ et S. Il est maintenant possible d'effectuer les dérivées partielles par rapport à tous les paramètres d'entrées ω, θ et S. La prochaine équation montre cette dérivation, il est à noter que la variable P représente l'un des trois paramètres.

$$\frac{\delta f_{at}(\omega,\theta,S)}{\delta P} = \frac{\delta A_0}{\delta P} + \frac{\delta A_1}{\delta P}\cos(wt_{\max}) + \ldots + \frac{\delta A_{i-1}}{\delta P}\cos((i-1)wt_{\max}) + \frac{\delta A_i}{\delta P}\cos(iwt_{\max}) \tag{3.17}$$

Les dérivées partielles correspondent aux directions des modifications pour permettre à la valeur maximale d'atteindre le seuil critique. On obtient la règle d'apprentissage suivante :

$$\begin{aligned} \omega_{i,k} &= \omega_{i,k-1} + \eta\,\frac{\delta f_{at}(\omega,\theta,S)}{\delta\omega_{i,k-1}} \\ \theta_{i,k} &= \theta_{i,k-1} + \eta\,\frac{\delta f_{at}(\omega,\theta,S)}{\delta\theta_{i,k-1}} \\ S_{i,k} &= S_{i,k-1} + \eta\,\frac{\delta f_{at}(\omega,\theta,S)}{\delta S_{i,k-1}} \end{aligned} \tag{3.18}$$

où η est le taux d'apprentissage. Cette dernière équation signifie que l'on modifie les poids du neurone pour que la valeur maximale de la réponse totale puisse augmenter et finalement conduire. Le tableau 3.1 montre les différentes dérivées partielles selon le coefficient.

Tableau 3.1 : Dérivées partielles des coefficients par rapport aux paramètres synaptiques pour le cas de l'excitation du quantron

	Coefficients	
Paramètre	A_0	A_n
$\frac{\delta f_{at}(\omega,\theta,S)}{\delta\omega_j}$	$\frac{\delta A_0}{\delta\omega_j} = \frac{1}{L}\sum_{i=0}^{m} S_j$	$\frac{2}{\pi}\sum_{i=0}^{m}\sin\left(\frac{n\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{n\pi(\theta_j + xi)}{L}\right)$
$\frac{\delta f_{at}(\omega,\theta,S)}{\delta\theta_j}$	$\frac{\delta A_0}{\delta\theta_j} = 0$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(n\frac{\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{n\pi(\theta_j + xi)}{L}\right)$
$\frac{\delta f_{at}(\omega,\theta,S)}{\delta S_j}$	$\frac{\delta A_0}{\delta S_j} = \frac{1}{L}\sum_{i=0}^{m}\omega_j$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{n\pi(\theta_j + xi + S_j)}{L}\right)$

3.2.2 Inhibition du quantron

Ce cas est l'opposé du cas précédent. Il survient lorsque l'on désire qu'il n'y ait pas de conduction, mais que le circuit propage l'influx. Il s'agit alors d'arrêter la conduction du réseau en inhibant les neurones qui conduisent.

Inhiber un neurone signifie réduire sa réponse totale pour qu'elle devienne inférieure au seuil critique. Pour être en mesure de réduire la réponse totale, il est nécessaire d'identifier le dernier moment où elle dépasse le seuil critique pour la réduire par la suite. Ce moment est identifié par la variable t_{max} que l'on détermine de la même manière que dans le cas précédent. Ce cas est illustré à la figure 3.5.

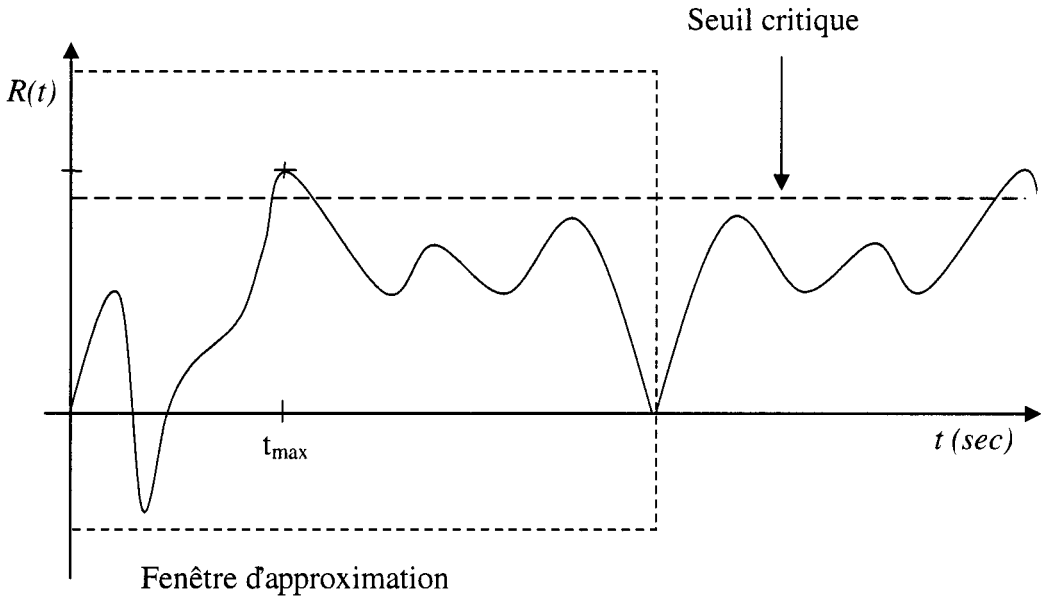


Figure 3.5 : Représentation de la réponse totale à inhiber au moment le plus propice

En inhibant le neurone pour que la valeur diminue, le neurone tend à ne plus conduire. Le raisonnement demeure le même que dans le cas précédent pour l'obtention des dérivées partielles. Cependant au lieu d'exciter les neurones qui ne conduisent pas, on inhibe les neurones qui conduisent l'influx. La règle d'apprentissage devient :

$$\begin{aligned} \omega_{i,k} &= \omega_{i,k-1} - \eta \frac{\delta f_{at}(\omega,\theta,S)}{\delta \omega_{i,k-1}} \\ \theta_{i,k} &= \theta_{i,k-1} - \eta \frac{\delta f_{at}(\omega,\theta,S)}{\delta \theta_{i,k-1}} \\ S_{i,k} &= S_{i,k-1} - \eta \frac{\delta f_{at}(\omega,\theta,S)}{\delta S_{i,k-1}} \end{aligned} \tag{3.19}$$

La seule différence est le signe de la mise à jour, c'est-à-dire que l'on inhibe le réseau au lieu de l'exciter.

3.2.3 Rétropropagation de l'erreur pour le quantron

Le rétropropagation de l'erreur nécessite l'obtention des dérivées partielles des paramètres synaptiques. Le théorème suivant est énoncé pour obtenir les dérivées partielles, d'ordre deux, des paramètres synaptiques :

Théorème

Soit une fonction f(t) qui représente l'approximation d'ordre deux de la réponse totale du quantron ayant un noyau rectangulaire

$$f(t) = A_0 + A_1 \cos(wt) + A_2 \cos(2wt)$$

où les coefficients A_2, A_1, A_0 sont les coefficients de la série de Fourier tronquée.

$$A_0 = \frac{1}{L}\sum_{j=1}^{n}\sum_{i=0}^{m}\omega_j S_j$$

$$A_1 = \sum_{j=1}^{n}\frac{2\omega_j}{\pi}\sum_{i=0}^{m}\sin\left(\frac{\pi(\theta + xi + S_j)}{L}\right) - \sin\left(\frac{\pi(\theta_j + xi)}{L}\right)$$

$$A_2 = \sum_{j=1}^{n}\frac{\omega_j}{\pi}\sum_{i=0}^{m}\sin\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{2\pi(\theta_j + xi)}{L}\right)$$

Alors le gradient du paramètre P, qui représente un des quatre suivant x, ω, θ et S, est obtenu par

$$\frac{\delta y}{\delta P} = \frac{-L\left[2a\left(-b \pm \dfrac{12b\dfrac{\delta A_1}{\delta P} - 8c\dfrac{\delta A_2}{\delta P} - 4a\left(\dfrac{\delta A_0}{\delta P} - \dfrac{\delta A_2}{\delta P}\right)}{2\sqrt{b^2 - 4ac}}\right) - 4\dfrac{\delta A_2}{\delta P}\Delta\right]}{4a^2\pi\sqrt{1-\Delta^2}}$$

où

$$\Delta = \left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right)$$

$$a = 2A_2$$

$$b = A_1$$

$$c = A_0 - A_2 - \Gamma$$

où les différentes dérivées partielles des coefficients A_2, A_1, A_0 par rapport au paramètre désiré sont résumées dans le tableau suivant.

	Coefficient		
Paramètre	A_0	A_1	A_2
$\frac{\delta}{\delta \omega_j}$	$\frac{\delta A_0}{\delta \omega_j} = \frac{1}{L}\sum_{i=0}^{m} S_j$	$\frac{2}{\pi}\sum_{i=0}^{m}\sin\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{\pi(\theta_j + xi)}{L}\right)$	$\frac{2}{\pi}\sum_{i=0}^{m}\sin\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{2\pi(\theta_j + xi)}{L}\right)$
$\frac{\delta}{\delta \theta_j}$	$\frac{\delta A_0}{\delta \theta_j} = 0$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{\pi(\theta_j + xi)}{L}\right)$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{2\pi(\theta_j + xi)}{L}\right)$
$\frac{\delta}{\delta S_j}$	$\frac{\delta A_0}{\delta S_j} = \frac{1}{L}\sum_{i=0}^{m}\omega_j$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right)$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right)$
$\frac{\delta}{\delta x}$	$\frac{\delta A_0}{\delta x} = 0$	$\frac{2i\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{\pi(\theta_j + xi)}{L}\right)$	$\frac{2i\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{2\pi(\theta_j + xi)}{L}\right)$

Preuve

L'implantation de l'algorithme d'apprentissage de rétropropagation pour le quantron nécessite une fonction entre les entrées et la sortie du quantron. Jusqu'à présent, les caractéristiques du quantron furent obtenues à partir d'approximation. C'est aussi le cas pour la fonction analytique d'entrées et de sortie du quantron, cependant l'ordre de la fonction d'approximation est tronqué à un ordre deux. Cette troncature permet d'obtenir une fonction analytique entre les entrées et la sortie. L'erreur introduite par cette troncature est importante. Il est important de noter que la réduction à un ordre deux a été effectuée dans le but de simplifier la formule et qu'une réduction jusqu'à un ordre quatre est possible puisqu'il s'agit du degré maximum de polynôme pouvant être résolu analytiquement. En tronquant l'équation 3.5, on obtient la fonction d'approximation d'ordre deux de la réponse totale :

$$f(t) = A_0 + A_1 \cos(wt) + A_2 \cos(2wt) \tag{3.20}$$

où A_2, A_1, A_0 sont les coefficients de la série de Fourier tronquée. Ces coefficients dépendent des paramètres ω, θ et S et ils sont obtenus à partir de l'équation 3.2.

$$
\begin{aligned}
A_0 &= \frac{1}{L}\sum_{j=1}^{n}\sum_{i=0}^{m}\omega_j S_j \\
A_1 &= \sum_{j=1}^{n}\frac{2\omega_j}{\pi}\sum_{i=0}^{m}\sin\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{\pi(\theta_j + xi)}{L}\right) \\
A_2 &= \sum_{j=1}^{n}\frac{\omega_j}{\pi}\sum_{i=0}^{m}\sin\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{2\pi(\theta_j + xi)}{L}\right)
\end{aligned}
\tag{3.21}
$$

Les coefficients T_2 à T_0 sont nécessaires pour transformer les fréquences harmoniques en fréquence fondamentale.

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1
\end{aligned}
\tag{3.22}
$$

L'équation devient alors :

$$f(t) = A_0 + A_1\cos(wt) + A_2(2\cos(wt)^2 - 1) \qquad 3.23)$$

En organisant l'équation, on obtient :

$$f(t) = A_0 - A_2 + A_1\cos(wt) + 2A_2\cos(wt)^2 \tag{3.24}$$

Par un changement de variable,

$$x = \cos(\ wt\ ) \tag{3.25}$$

nous obtenons le polynôme de degré deux.

$$f(x) = A_0 - A_2 + A_1 x + 2A_2 x^2 \tag{3.26}$$

On soustrait le seuil critique à la fonction d'approximation pour déterminer le moment de la jonction entre le seuil critique Γ et la réponse totale.

$$f(x) = A_0 - A_2 - \Gamma + A_1 x + 2A_2 x^2 \tag{3.27}$$

Réécrivons le tout plus simplement :

$$\begin{aligned} & f(x) = ax^2 + bx + c \\ & où \\ & a = 2A_2 \\ & b = A_1 \\ & c = A_0 - A_2 - \Gamma \end{aligned} \tag{3.28}$$

Les racines de l'équation quadratique sont obtenues par l'équation suivante :

$$r_1, r_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{3.29}$$

Les racines de l'équation 3.25 s'obtiennent par:

$$y_1, y_2 = L/\pi \, ar\cos\left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\right) \tag{3.30}$$

La sortie correspond à la racine réelle la plus petite, supérieure à zéro. Dans ce cas, ce n'est pas la racine qui nous intéresse, mais plutôt la formule qui mène à la racine. Cette formule correspond à la fonction entre les entrées et la sortie du quantron. Avec cette fonction, il est possible de dériver la formule par rapport aux paramètres synaptiques. En assumant que l'une des deux est la bonne racine, la fonction entre les entrées et la sortie d'ordre deux correspond à :

$$y = L/\pi \, ar\cos\left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\right) \tag{3.31}$$

L'objectif est de déterminer les dérivées partielles de notre fonction d'entrées et de sortie. Nous appliquerons la dérivée en chaîne de l'équation 3.31. On obtient :

$$\frac{\delta y}{\delta P} = \frac{-L\left[2a\left(-b \pm \frac{12b\frac{\delta b}{\delta P} - 4\frac{\delta a}{\delta P}c - 4\frac{\delta c}{\delta P}a}{2\sqrt{b^2-4ac}}\right) - 2\frac{\delta a}{\delta P}\left(\frac{-b \pm \sqrt{b^2-4ac}}{2a}\right)\right]}{4a^2\pi\sqrt{1-\left(\frac{-b \pm \sqrt{b^2-4ac}}{2a}\right)^2}} \tag{3.32}$$

où P est un des paramètres synaptiques. Cette dernière équation permet de connaître la formule des dérivées partielles par rapport aux paramètres synaptiques. Il reste à connaître les dérivées partielles par rapport aux coefficients a, b et c.

$$\begin{aligned} \frac{\delta a}{\delta P} &= 2\frac{\delta A_2}{\delta P} \\ \frac{\delta b}{\delta P} &= \frac{\delta A_1}{\delta P} \\ \frac{\delta c}{\delta P} &= \frac{\delta A_0}{\delta P} - \frac{\delta A_2}{\delta P} \end{aligned} \tag{3.33}$$

En effectuant la substitution, nous obtenons l'équation 3.34 qui présente la dérivée partielle désirée pour l'implantation de l'algorithme d'apprentissage de rétropropagation de l'erreur dans un réseau multicouche de quantrons.

$$\frac{\delta y}{\delta P} = \frac{-L\left[2a\left(-b \pm \frac{12b\frac{\delta A_1}{\delta P} - 8c\frac{\delta A_2}{\delta P} - 4a\left(\frac{\delta A_0}{\delta P} - \frac{\delta A_2}{\delta P}\right)}{2\sqrt{b^2 - 4ac}}\right) - 4\frac{\delta A_2}{\delta P}\Delta\right]}{4a^2\pi\sqrt{1-\Delta^2}} \tag{3.34}$$

$$\Delta = \left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\right)$$

Nous devons maintenant substituer les différentes dérivées partielles du tableau 3.2 dans l'équation précédente selon le paramètre synaptique à dérivée.

Tableau 3.2 : Dérivées partielles des coefficients par rapport aux paramètres synaptiques pour le cas de la rétropropagation de l'erreur

	Coefficient		
Paramètre	A_0	A_1	A_2
$\frac{\delta}{\delta\omega_j}$	$\frac{\delta A_0}{\delta\omega_j} = \frac{1}{L}\sum_{i=0}^{m} S_j$	$\frac{2}{\pi}\sum_{i=0}^{m}\sin\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{\pi(\theta_j + xi)}{L}\right)$	$\frac{2}{\pi}\sum_{i=0}^{m}\sin\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right) - \sin\left(\frac{2\pi(\theta_j + xi)}{L}\right)$
$\frac{\delta}{\delta\theta_j}$	$\frac{\delta A_0}{\delta\theta_j} = 0$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{\pi(\theta_j + xi)}{L}\right)$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{2\pi(\theta_j + xi)}{L}\right)$
$\frac{\delta}{\delta S_j}$	$\frac{\delta A_0}{\delta S_j} = \frac{1}{L}\sum_{i=0}^{m}\omega_j$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right)$	$\frac{2\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right)$
$\frac{\delta}{\delta x}$	$\frac{\delta A_0}{\delta x} = 0$	$\frac{2i\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{\pi(\theta_j + xi)}{L}\right)$	$\frac{2i\omega_j}{L}\sum_{i=0}^{m}\cos\left(\frac{2\pi(\theta_j + xi + S_j)}{L}\right) - \cos\left(\frac{2\pi(\theta_j + xi)}{L}\right)$

Nous possédons maintenant toutes les valeurs nécessaires à l'implantation de l'algorithme d'apprentissage de la rétropropagation de l'erreur dans notre quatrième cas d'apprentissage. □

À partir des différents gradients obtenus, nous pouvons obtenir les gradients d'erreurs nécessaires à l'implantation de l'algorithme d'apprentissage de rétropropagation de l'erreur

pour le réseau multicouche de quantrons tel que résumé dans le tableau 3.3 où la variable P représente un des quatre paramètres x, θ, ω et S.

Tableau 3.3 : Gradients d'erreurs nécessaires à l'algorithme d'apprentissage

	Paramètre ω	Paramètre θ	Paramètre S
Couche de sortie k	$\frac{\delta E}{\delta \omega_k} = -(d_k - y_k)\frac{\delta y_k}{\delta \omega_k}$ $\delta_{\omega k} = -(d_k - y_k)$	$\frac{\delta E}{\delta P_k} = -(d_k - y_k)\frac{\delta y_k}{\delta P_k}$ $\delta_k = -(d_k - y_k)$	$\frac{\delta E}{\delta P_k} = -(d_k - y_k)\frac{\delta y_k}{\delta P_k}$ $\delta_k = -(d_k - y_k)$
Couche cachée j	$\frac{\delta E}{\delta \omega_j} = -\sum_k \left(\delta_{\omega k}\frac{\delta y_k}{\delta x_k}\right)\frac{\delta y_j}{\delta \omega_j}$ $\delta_{\omega j} = -\sum_k \left(\delta_{\omega k}\frac{\delta y_k}{\delta x_k}\right)$	$\frac{\delta E}{\delta P_j} = -\sum_k \left(\delta_k\frac{\delta y_k}{\delta x_k}\right)\frac{\delta y_j}{\delta P_j}$ $\delta_j = -\sum_k \left(\delta_k\frac{\delta y_k}{\delta x_k}\right)$	$\frac{\delta E}{\delta P_j} = -\sum_k \left(\delta_k\frac{\delta y_k}{\delta x_k}\right)\frac{\delta y_j}{\delta P_j}$ $\delta_j = -\sum_k \left(\delta_k\frac{\delta y_k}{\delta x_k}\right)$
Couche d'entrée i	$\frac{\delta E}{\delta \omega_i} = -\sum_j \left(\delta_{\omega j}\frac{\delta y_j}{\delta x_j}\right)\frac{\delta y_i}{\delta \omega_i}$ $\delta_{\omega i} = -\sum_j \left(\delta_{\omega j}\frac{\delta y_j}{\delta x_j}\right)$	$\frac{\delta E}{\delta P_i} = -\sum_j \left(\delta_j\frac{\delta y_j}{\delta x_j}\right)\frac{\delta y_i}{\delta P_i}$ $\delta_i = -\sum_j \left(\delta_j\frac{\delta y_j}{\delta x_j}\right)$	$\frac{\delta E}{\delta P_i} = -\sum_j \left(\delta_j\frac{\delta y_j}{\delta x_j}\right)\frac{\delta y_i}{\delta P_i}$ $\delta_i = -\sum_j \left(\delta_j\frac{\delta y_j}{\delta x_j}\right)$

3.3 Expérimentation avec le quantron

Les différentes équations permettant l'apprentissage ont été obtenues dans la section précédente. Il s'agit maintenant de tester l'algorithme d'apprentissage. Cette expérimentation a pour but d'examiner le comportement et la convergence de l'algorithme d'apprentissage. Trois différents cas sont testés. Le premier consiste à générer une image à partir d'un réseau où les poids synaptiques sont générés de manière aléatoire et

d'effectuer l'apprentissage à partir d'un autre réseau généré aléatoirement. Le second test consiste à résoudre le problème classique du « ou exclusif » et le troisième à résoudre un problème linéairement séparable.

3.3.1 Résultats

Pour tester l'algorithme d'apprentissage du quantron, nous utiliser la même procédure que pour le quantron simplifié au chapitre 2.2.2. Le premier test effectué consiste à générer une image à partir d'un à partir d'un réseau où les poids synaptiques ainsi que la configuration du réseau sont générés de manière aléatoire. La structure généré aléatoirement varie de deux couches à cinq couches et chaque couche varie de deux neurones à quatre Cette image sert à l'entraînement d'un second réseau initialisé aléatoirement. La figure 3.6 compare l'image de sortie du réseau entraîné à celle de l'image désirée. L'erreur entre les deux images est petite, ce qui signifie que l'algorithme d'apprentissage fonctionne adéquatement dans le présent exemple. La figure 3.7, présente l'évolution de l'erreur en fonction de l'époque et montre que l'erreur diminue.

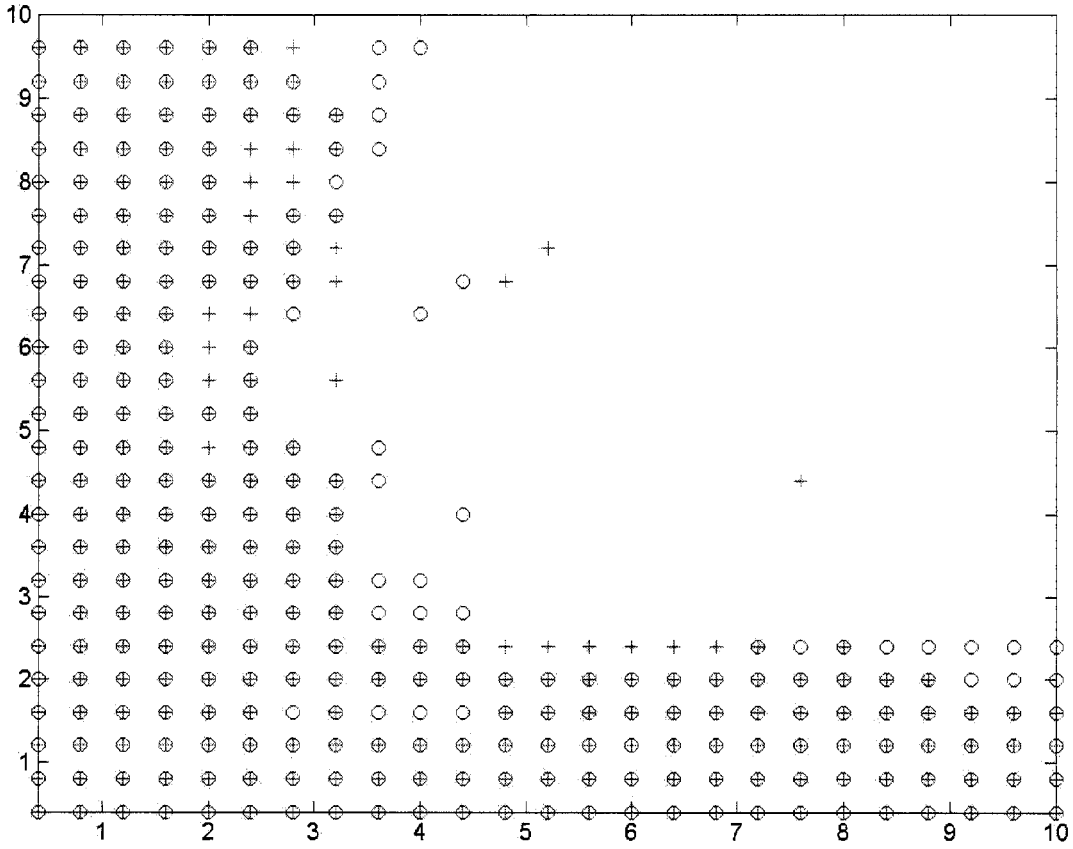


Figure 3.6 : Superposition des valeurs obtenues (croix) et des valeurs désirées (cercles) pour l'apprentissage d'un réseau multicouche de quantrons

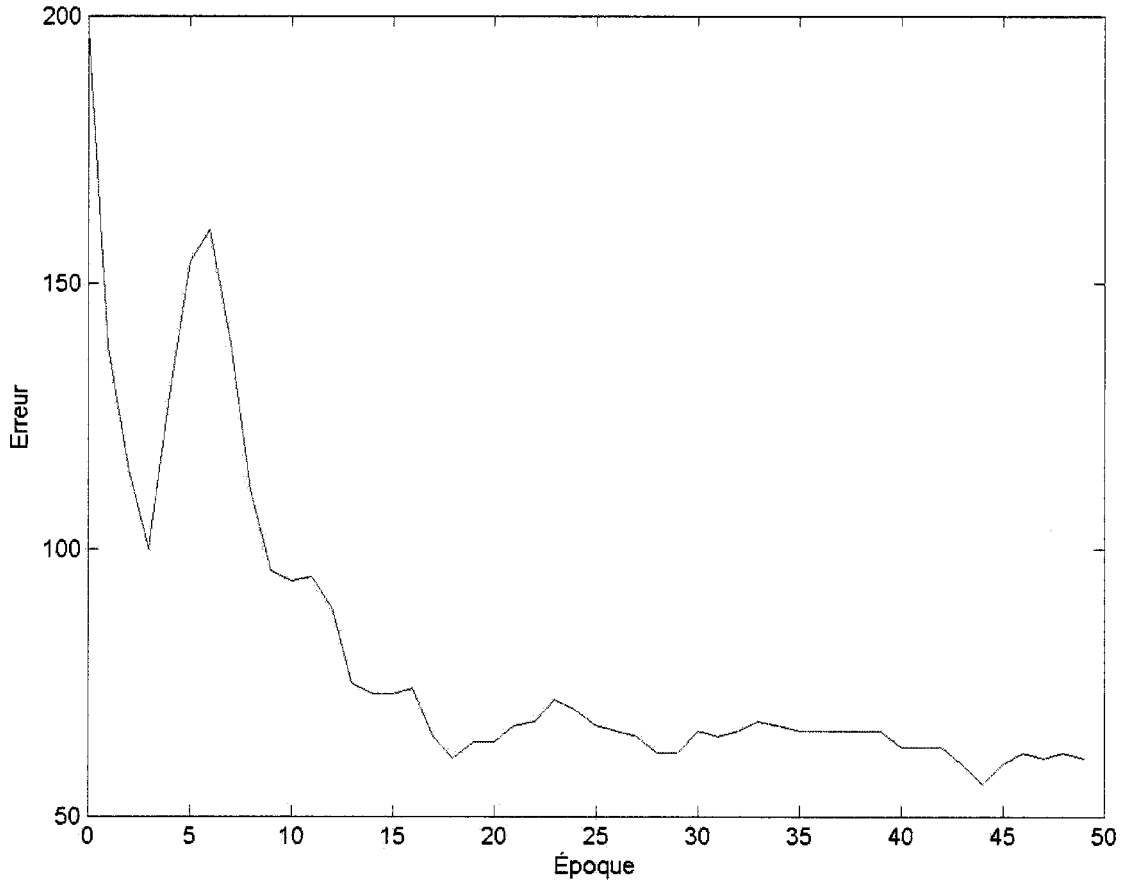


Figure 3.7 : Graphique de l'erreur en fonction de l'époque

Pour observer le comportement général de l'algorithme d'apprentissage, nous avons effectué l'apprentissage sur une centaine de réseaux différents. Les résultats de l'apprentissage de 100 réseaux sont présentés dans le tableau 3.4.

Tableau 3.4 : Résultat pour la convergence de l'algorithme

	Moyenne du nombre d'époques	Réduction d'erreurs par rapport aux erreurs initiales	Réduction d'erreurs maximale
Apprentissage sans rétropropagation	48,61 époques	53,4 %	60,4 %
Apprentissage avec rétropropagation	32,16 époques	65,0 %	72,3 %

Le rapport entre l'erreur minimale et l'erreur initiale au cours de l'apprentissage ainsi que le rapport entre l'erreur finale et l'erreur initiale sont inférieurs lorsque l'apprentissage s'effectue avec rétropropagation. Voilà la preuve que l'algorithme d'apprentissage fonctionne mieux avec l'utilisation de la rétropropagation. Ces résultats sont très satisfaisants car tel qu'illustré à la figure 3.6, les problèmes à résoudre sont très non linéaire.

Le second test effectué consiste à résoudre le problème du « ou exclusif ». La procédure de test reste la même sauf que l'image désirée correspond à celle du « ou exclusif ». Le problème est énoncé dans le tableau 3.5

Tableau 3.5 : Énoncé du problème du « ou exclusif » pour le quantron

Entrée X	Entrée Y	Sortie
3	3	Conduction
3	5	Non Conduction
5	3	Non Conduction
5	5	Conduction

Nous avons tenté de résoudre le problème avec un seul quantron. Les critères d'arrêt sont d'obtenir une erreur nulle ou que le nombre d'époque atteigne cinq cent époques. Nous avons résolu le problème cent fois, les résultats obtenus sont résumés dans le tableau 3.6.

Tableau 3.6 : Résultat de la résolution du problème du « ou exclusif » avec un seul quantron

	Taux de réussite	Nombre moyen d'époques pour atteindre la solution
Algorithme d'apprentissage sans rétropropagation	71%	243,3 époques
Algorithme d'apprentissage avec rétropropagation	75%	85,6 époques

Cette fois encore, l'algorithme d'apprentissage avec rétropropagation obtient de meilleurs résultats que celui sans rétropropagation. Les résultats nous satisfont.

Le troisième et dernier test consiste à l'apprentissage d'un problème linéairement séparable par un réseau possédant huit neurones à la couche cachée et un neurone de sortie. La figure 3.8 constitue la meilleure époque lors de l'apprentissage. L'algorithme d'apprentissage peine à résoudre un problème très simple De plus, l'apprentissage suit une courbe asymptotique, tel qu'illustré à la figure 3.9.

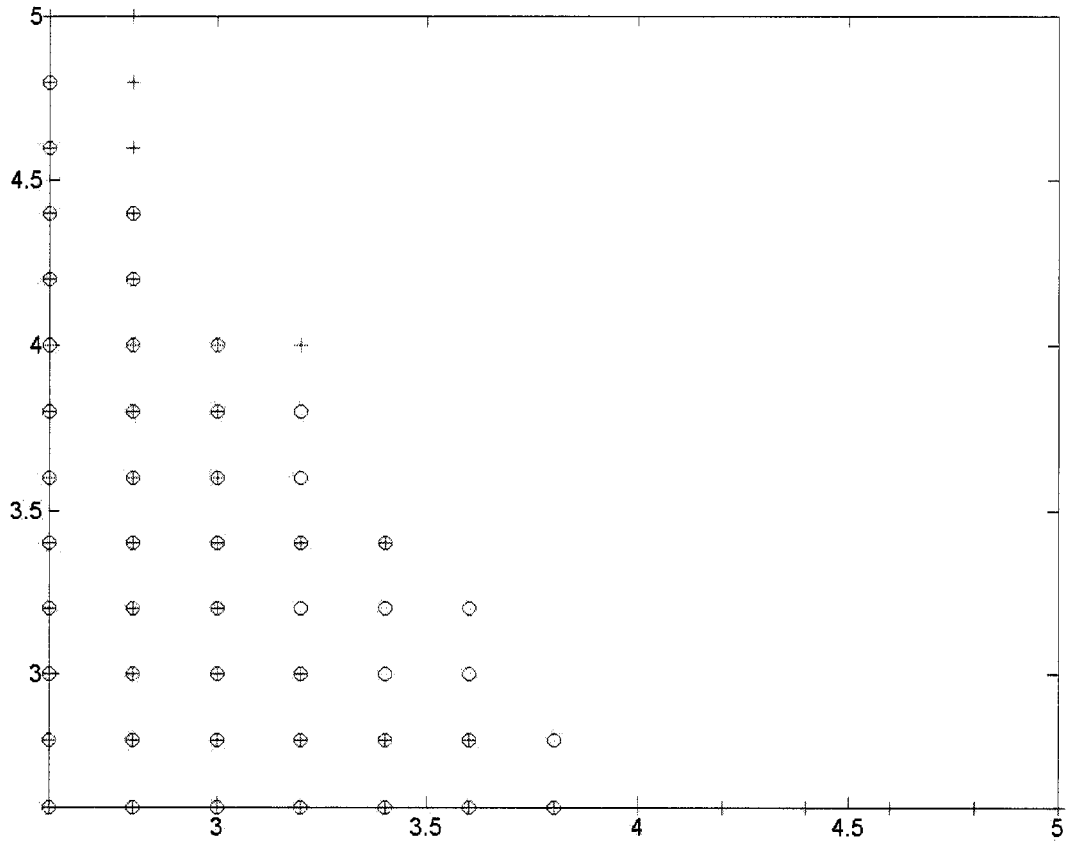


Figure 3.8 : Fonction discriminante obtenue lors de la résolution d'un problème linéairement séparable

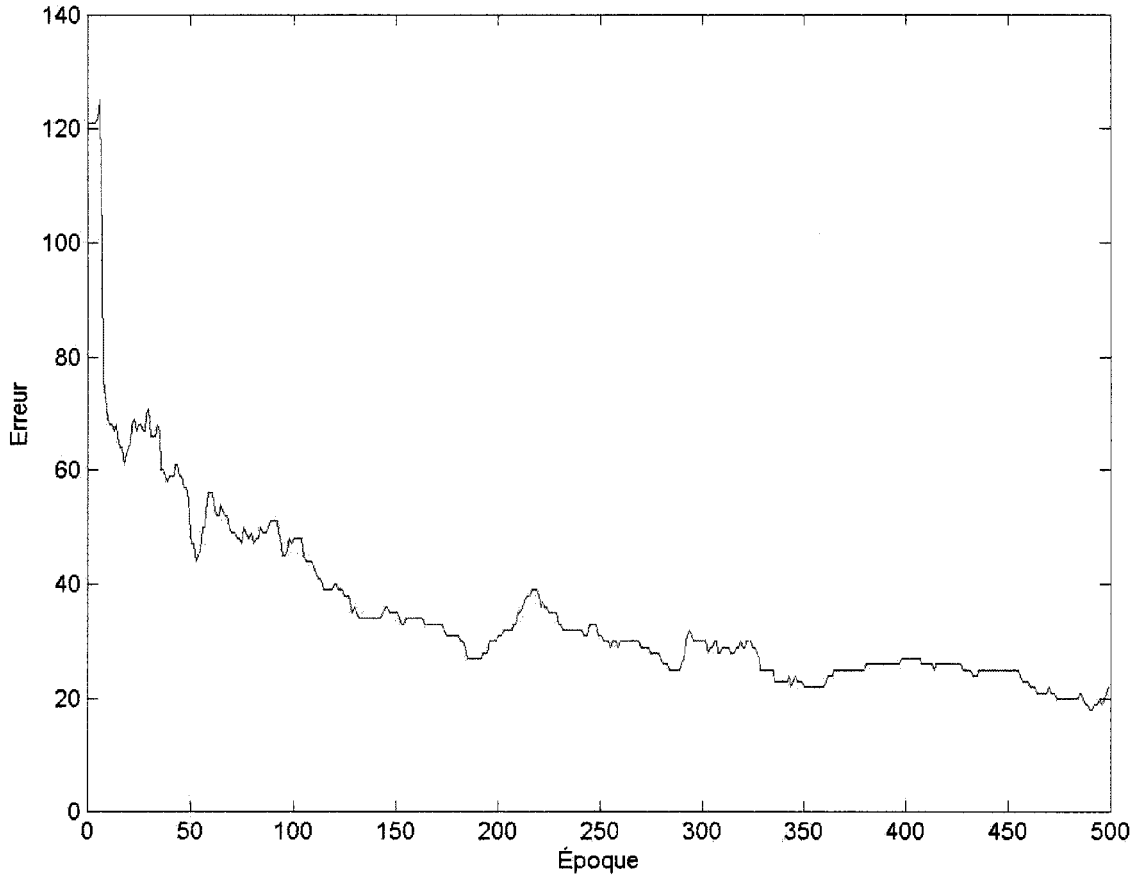


Figure 3.9 : Graphique de l'erreur en fonction de l'époque de l'apprentissage

Pour observer le comportement de l'algorithme d'apprentissage sur ce problème, nous avons simulé la réponse pour une centaine de réseaux. Les résultats sont compilés dans le tableau 3.7. Les résultats obtenus sont décevants puisque le problème à résoudre est assez simple.

Tableau 3.7 : Résultat pour la résolution d'un problème linéairement séparable

	Moyenne du nombre d'époque	Réduction d'erreurs par rapport aux erreurs initiales	Réduction d'erreurs maximale
Apprentissage sans rétropropagation	43,4 époques	75,8 %	80 %
Apprentissage avec rétropropagation	45,5 époques	76,0 %	80,6 %

3.2.3 Discussion

Notre premier objectif de recherche était d'élaborer et d'implanter un algorithme d'apprentissage pour le quantron. Nous cherchions un algorithme d'apprentissage permettant de minimiser l'erreur. Trois tests ont été effectués pour observer le comportement de l'algorithme d'apprentissage. Les trois tests ont présenté une diminution de l'erreur, ce qui montre que l'algorithme d'apprentissage permet de converger vers une solution se rapprochant de la solution désirée. Également, nous avons constaté que l'ajout de la rétropropagation de l'erreur dans l'algorithme d'apprentissage améliore les performances dans les trois tests.

Nous avons observé plusieurs images de sortie et elles possèdent toutes certains points en communs qui sont présentés à la figure 3.10. Un de ces points communs est la fonction discriminante qui sépare la zone de conduction du réseau multicouche de

quantrons de la zone de la non conduction. Cette fonction discriminante est non linéaire. Dans la zone de conduction, les rectangles formés de lignes pointillées correspondent à la zone de « sur excitation ». Seule l'excitation d'une seule des deux entrées entraîne la conduction du réseau entier, peu importe les valeurs de l'autre entrée. Graphiquement, nous observons ce phénomène par des bandes rectangulaires débutant à l'origine et tendant vers l'infini.

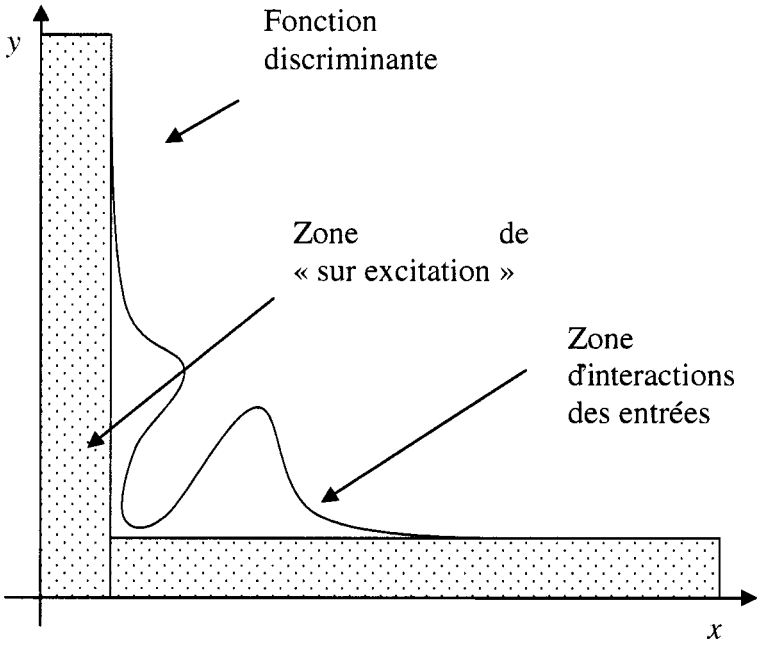


Image typique de sortie du quantron

Figure 3.10 : Caractéristiques de l'image de sortie du quantron pour deux entrées

La zone d'interactions des entrées correspond à une zone où les différentes entrées interagissent. Cette zone forme des fonctions discriminantes non linéaires très variées. Elle constitue notre zone d'intérêt, c'est-à-dire celle à contrôler par l'algorithme d'apprentissage. Afin d'observer le comportement de l'algorithme d'apprentissage, dans cette zone d'interaction, nous avons effectué l'apprentissage du problème du « ou exclusif » et du problème linéairement séparable. Les résultats sont encourageants pour la résolution du problème du « ou exclusif » puisque l'algorithme d'apprentissage avec rétropropagation le résout 75% du temps. Pour la résolution du problème linéairement séparable, l'algorithme d'apprentissage ne le résout pas entièrement, mais il minimise l'erreur initiale de 76 % et la réduction de l'erreur à sa meilleure époque est de 80,6 %. Cependant ce dernier problème peut être résolu complètement par un simple perceptron. L'une des hypothèses que l'on avance pour ce résultat décevant est que cette zone d'interactions n'est pas contrôlée parfaitement. Nous supposons que ce manque de contrôle est du à une certaine périodicité dans la fonction entre les entrée et la sortie. Cette périodicité amène des minimums et des maximums locaux, comme il a été montré avec le modèle du quantron simplifié sinusoïdal, l'algorithme peine dans ce cas là. Cette caractéristique est en majeure partie due au paramètre θ. Nous croyons qu'un meilleur contrôle de ce paramètre apportera des améliorations de performances significatives.

Le quantron nous apparaît alors peu efficace pour résoudre des problèmes de classification linéaire cependant il est plus performant pour résoudre des problèmes complexe et non linéaire comme le montre la résolution par un quantron du problème du

« ou exclusif ». Cette inefficacité indique que l'algorithme d'apprentissage ne contrôle pas parfaitement cette zone d'interaction. Il est très difficile de contrôler la zone d'interaction, car la forme de sortie varie énormément en fonction des paramètres synaptiques, l'une des causes possible peut être la périodicité du paramètre θ que l'on n'arrive pas à contrôler avec le quantron simplifié sinusoïdale.

Alors, pour améliorer notre algorithme d'apprentissage, deux solutions s'offrent à nous. La première consiste à implanter les gradients d'ordre quatre à la rétropropagation de l'erreur plutôt que des gradients d'ordre deux. Ce résultat est déjà étonnant puisque la troncature est d'ordre deux donc les performances devraient être nettement améliorée. La deuxième est d'implanter le noyau original développé par Labib (1999) plutôt que d'utiliser un noyau rectangulaire.

Notre deuxième objectif de recherche était d'obtenir un algorithme possédant une ressemblance biologique. Parmi nos quatre cas, nous savons que la rétropropagation de l'erreur n'est pas plausible biologiquement, cependant, pour les cas d'inhibition et d'excitation du quantron leur plausibilité biologique n'a pas été vérifiée; la question est donc ouverte.

Le futur de la recherche réside dans l'amélioration de l'algorithme d'apprentissage et l'ajout de techniques heuristiques d'apprentissage. Tel que nous l'avons mentionné, les premières améliorations possibles sont l'implantation au quantron du noyau découvert par Labib

(1999) et l'augmentation de l'ordre des gradients à quatre pour la rétropropagation de l'erreur. Ensuite, des techniques heuristiques pourront être empruntées du perceptron dans le but d'améliorer les performances de l'apprentissage. L'une des techniques possibles serait l'ajout d'un moment tel qu'expliqué par Yu et Chen (1997). Une autre serait l'implantation d'un taux d'apprentissage particulier à chacun des paramètres synaptiques. Cette dernière technique a déjà été généralisée par les algorithmes *SuperSab* et *Bar-Delta-Bar* qui permettent d'obtenir des taux d'apprentissage variables (Hagan et al.(1996)). Finalement, l'une des améliorations les plus intéressantes serait la construction d'un réseau mixte de neurones, c'est-à-dire un réseau où il y a des couches de perceptrons et de couches de quantrons. Pour effectuer cette dernière amélioration, notre algorithme d'apprentissage pourrait être implanté sans aucune modification.

CONCLUSION

Au cours de cette recherche, nous avons élaboré un algorithme d'apprentissage supervisé pour l'implanter dans un réseau multicouche de quantrons, ce modèle de neurone artificiel tout à fait nouveau. Le but était de créer un algorithme d'apprentissage se rapprochant davantage de la réalité biologique, tout comme le modèle du quantron.

Dans un premier temps, les différents modèles de neurones ressemblant au quantron ont été explorés pour nous permettre de connaître leur algorithme d'apprentissage. Le premier modèle étudié est le perceptron. Ce neurone utilise un algorithme d'apprentissage très intéressant : la rétropropagation de l'erreur. Cependant, il existe des différences fondamentales entre le modèle du quantron et du perceptron qui entraînent l'impossibilité d'implanter directement l'algorithme d'apprentissage. La différence entre la notion de potentiel synaptique variant dans le temps et le modèle atemporel du perceptron démarque ces deux modèles. Le second modèle étudié est le neurone à impulsion. Ce modèle possède de grandes similarités avec le quantron puisqu'il se base sur les nouvelles connaissances de la neurobiologie. Cependant, sa sortie et ses entrées sont constituées d'un schéma d'impulsions temporelles, ce qui est totalement différent de la sortie et des entrées du quantron qui ont une valeur numérique représentant la période d'émission des potentiels synaptiques. Ces conclusions nous ont permis de comprendre que le nouvel algorithme d'apprentissage se devait d'être original puisque aucun

algorithme d'apprentissage ne satisfaisait toutes les caractéristiques du quantron. Celles-ci ont alors été étudiées en détail pour nous permettre de découvrir l'algorithme d'apprentissage répondant au besoin du quantron.

Dans un deuxième temps, l'algorithme d'apprentissage proposé a été détaillé. La principale difficulté a consisté à gérer la conduction ou la non conduction du quantron. Il en a résulté quatre cas différents : absence de modification, excitation du réseau, inhibition du réseau et rétropropagation de l'erreur. Un apprentissage a été introduit pour chacun des cas. L'algorithme d'apprentissage que nous avons créé ne s'applique pas qu'au quantron. La preuve est que, avant son implantation au quantron, nous avons observé son comportement sur un modèle de quantron simplifié. Nos tests avec le quantron simplifié ont consisté à effectuer une reconnaissance d'image. Les résultats ont été excellents, puisque notre algorithme d'apprentissage a permis de minimiser l'erreur. Cette partie de la recherche est novatrice puisque jusqu'à ce jour aucun algorithme d'apprentissage ne permettait l'apprentissage de ce type de neurone.

Dans un troisième temps, encouragé par les résultats avec le quantron simplifié, nous avons implanté l'algorithme d'apprentissage à un réseau multicouche de quantrons. Le noyau que l'on a choisi d'implanter aux quantrons est un noyau rectangulaire. L'implantation de l'algorithme d'apprentissage au réseau multicouche de quantrons s'est avérée beaucoup moins simple qu'avec le modèle simplifié. Premièrement, nous avons dû propager le signal vers la sortie. Pour déterminer la sortie du quantron, nous avons

effectué une approximation de la réponse totale par les séries de Fourier. En tronquant et effectuant un changement de variable, nous avons obtenu un polynôme permettant de trouver le moment de la jonction du seuil critique et de la réponse totale qui correspond à la sortie du quantron. Deuxièmement, nous avons dû pallier les particularités de chacun des quatre cas d'apprentissage. L'implantation des trois premiers cas n'ont pas posé de grandes difficultés. Par contre, l'implantation de la rétropropagation de l'erreur a été plus complexe. Une formule analytique a été nécessaire pour obtenir les différents gradients d'erreurs. Ensuite, les gradients d'erreurs obtenus ont été implantés à partir du polynôme d'ordre deux, mais il est possible de les implanter jusqu'à un ordre quatre. Par la suite, nous avons testé l'algorithme d'apprentissage. Le premier test a consisté à observer l'évolution de l'apprentissage lors d'une reconnaissance d'image. Le second test a consisté à résoudre le problème du « ou exclusif ». Le troisième test, et non le moindre, a consisté à résoudre un problème linéairement séparable. Les résultats obtenus sont encourageants. Malgré l'utilisation de la rétropropagation de l'erreur, qui est biologiquement improbable, notre algorithme d'apprentissage est plus fidèle aux découvertes de la neurobiologie. Nous croyons qu'il permettra au réseau de neurones de résoudre des problèmes plus complexes de classification et de régression.

À notre avis, les développements futurs associés à l'algorithme d'apprentissage graviteront autour de trois points principaux. Le premier point sera l'implantation du noyau original et l'amélioration des gradients d'erreurs du quantron. Le second point sera l'ajout de techniques heuristiques d'apprentissage tel que l'ajout du moment et

l'implantation de taux d'apprentissage variables. Le dernier point consistera à élaborer un réseau mixte de perceptrons et de quantrons en exploitant les avantages de chacun des modèles de neurone artificiel.

BIBLIOGRAPHIE

BARLETT, P.L. et DOWNS, T. (1992). « Using random weights to train multilayer networks of hard-limiting units ». *IEEE Transactions on Neural Networks*. 3: 2 , P.202–210.

CHAN, M., ANDRE, B., HERRERA, A. et CELSIS, P. (1993). « Incremental learning in a multilayer neural network as an aid to Alzheimer's disease diagnosis ». Systems, Man and Cybernetics. *Systems Engineering in the Service of Humans*, Conference Proceedings., International Conference on 17-20 Oct. 1993 P:1 - 4 vol.4

DE KAMPS, M. et VAN DER VELDE, F. (2001). «From artificial neural networks to spiking neuron populations and back again», *Neural Networks*.14:6/7, p. 941.

GERSTNER, W. et KISTLER, W.M. 2002. *Spiking Neuron Models* [En ligne]. Cambridge : Cambridge University Press. 494 p. www.epfl.ch/~gerstner/SPNM/SPNM.html (Page consultée le 2 mars 2004)

GOODMAN, R.M. et ZHENG ZENG. (1994). « A learning algorithm for multi-layer perceptrons with hard-limiting threshold units ». *IEEE World Congress on Computational Intelligence*.1, P.193–197.

HAGAN, M.T. et MENHAJ, M.B. (1994). « Training feedforward networks with the Marquardt algorithm ». IEEE Transactions on Neural Networks. 5:6, P.989–993.

HAYKIN S. 1999. Neural Networks, A Comprehensive Foundation, Second edition, P.842.

HODGKIN, A. L. et HUXLEY, A. F. (1952). A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes. *J. Physiol. (Lond.)*, 117, P.500-544.

JEONG, D.-G. et LEE, S.-Y. (1996). «Merging back-propagation and Hebbian learning rules for robust classifications_». *Neural Networks*.9, p. 1213.

LABIB, R. (1999). «New single neuron structure for solving nonlinear problems». *IJCNN '99. International Joint Conference on Neural Networks*.1, P. 617–620.

LABIB, R. (2000). Processus de diffusion: outils de modélisation, de prévision et de contrôle. 107 p. Thèse de doctorat en mathématique appliquée, École Polytechnique de Montréal.

MAASS, W. et BISHOP, C.M. (1999). Pulsed Neural Networks. Cambridge : MIT Press. 408 p.

MINSKY, M.L. et PAPERT, S.A. (1988), *Perceptorns*, Expand Edition, MIT Press, Cambridge MA.

NG, V., FANG, R., BERT, J., BAND, P., SUIRCHEV, L. et KEEFE, A. (1993). « Pattern recognition of occupational cancer using neural networks ». *Communications, Computers and Signal Processing*, IEEE Pacific Rim Conference on Volume 1, 19-21 P:296 - 300 vol.1

O'REILLY, R.C. (1998).« Six Principles for Biologically-Based Computational Models of Cortical Cognition ». Trends in Cognitive Sciences, 2, p.455-462.

RIEDMILLER, M. et BRAUN, H. (1993).« A direct adaptive method for faster backpropagation learning: the RPROP algorithm ». *IEEE International Conference on Neural Networks*. 1, P.586–591.

RUMELHART, D.E, HINTON, G.E. et WILLIAMS, R.J. (1986). Learning internal representations by error propagation, in Parallel Distributed Processing (Vol. 1): Foundations (Rumelhart, D.E., McClelland, J.L. and PDP Research Group, eds), 318-362 p., MIT Press

STORK, D.G. (1989). « Is backpropagation biologically plausible*?*». *International Joint Conference on Neural Networks*. 2, P.241–246.

TOMS, D.J. (1990). « Training binary node feedforward neural networks by back propagation of error »,*IEE Electronics Letters* .26: 21, P.1745–1746.

Upper Saddle River, NJ : Prentice-Hall. 842 p.

WHITE, R.H. (1990).« The learning rate in back-propagation systems: an application of Newton's method ». *International Joint Conference on Neural Networks*. 1, P.679–684.

YU, X.-H. et CHEN, G.-A.. (1997). «Efficient backpropagation learning using optimal learning rate and momentum_». Neural Networks. 10, p. 517.