Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique
Université **D**r. **Y**ahia **F**arès de **M**édéa
Faculté des **S**ciences et de la **T**echnologie

Mémoire en vue de l'obtention du diplôme de Magister
Spécialité : **Génie Mécanique**
Option : **Energétique et Transferts**

THÈME

Application de la méthode neuronale pour la détermination des échanges thermiques intensifs

Présenté par : **Mr. FARID HACHICHI**
Sous la direction de : **Pr. SALAH HANINI**
Pr. MOHAMED NADJIB BOUAZIZ

Devant le jury composé de :

Pr. **S. ABUDURA**	Professeur, U.Y.F. Médéa	Président
Dr. **A. KORICHI**	Maître de conférences A, U.Y.F. Médéa	Examinateur
Dr. **O. BENKORTBI**	Maître de conférences A, U.Y.F. Médéa	Examinateur

Soutenue le .. /.. / 2014

Remerciements

Tout d'abord je tiens à remercier le bon dieu ***ALLAH*** qui m'a donné la force, le courage et la patience de bien réaliser ce travail.

Je tiens aussi à exprimer mes remerciements et toute ma reconnaissance à mes directeurs de thèse, Monsieur ***S. HANINI***, Professeur à l'Université Dr. Yahia Farès de Médéa, et Monsieur ***M.N. BOUAZIZ***, Professeur à l'Université Dr. Yahia Farès de Médéa, qui se sont montrés d'une disponibilité sans faille tout au long de cette étude et qui ont su diriger mes travaux en restant parfaitement ouvert à mes initiatives. En plus de leurs rigueurs scientifiques et de leurs intérêts pour le sujet, ils m'ont apporté un soutien moral considérable.

Mes remerciements s'adressent également à Monsieur ***S. ABUDURA,*** Professeur à l'université Dr. Yahia Farès de Médéa, pour m'avoir fait l'honneur de présider le jury.

J'adresse mes sincères remerciements au Docteur ***A. KORICHI,*** Maître de conférences A à l'université Dr. Yahia Farès de Médéa et Docteur ***O.BENKORTBI,*** Maître de conférences A à l'Université Dr. Yahia Farès de Médéa, pour leur participation à mon jury et examiner ce travail.

Enfin, je terminerai par remercier toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce travail.

Dédicaces

À mes parents ;

À mes frères et sœurs ;

À tous mes proches ;

À tous mes amis ;

À tous ceux qui m'ont soutenu de près ou de loin.

Sommaire

Remerciement
Dédicace
Liste des figures
Liste des tableaux
Nomenclature
Abréviation
Résumé
Introduction générale …..……..1

Chapitre I Réseaux de neurones artificiels

I.1 Introduction ……..……3
I.2. Historique ……...4
I.3. Neurones biologiques ……5
I.4. Neurone formel ……...6
I.4.1. Entrées Xi ……7
I.4.2. Poids synaptiques Wi ……..7
I.4.3. Seuil θ ……..8
I.4.4. Fonction d'activation F ……..……..8
I.4.5. Sortie du neurone S ……...10
I.5. Architecture du réseau ……..10
I.5.1. Réseau de neurones non bouclés ……...13
I.5.1.1. Perceptron ……...13
I.5.1.2. Perceptron multicouche (MLP) ……..14
I.5.1.3. Réseaux RBF (Radial Basis Function) ……...14
I.5.2. Réseaux de neurones bouclés (récurrents) ……..……14
I.5.2.1. Modèle de Hopfield ……15
I.5.2.2. Modèle de Boltzmann ……16
I.5.2.3. Modèle de Kohonen ……16
I.5.2.4. Modèles ART ……..17
I.6. Apprentissage ……..……18
I.6.1. Apprentissage supervisé ……18
I.6.2. Apprentissage non supervisé ……18
I.7. Règles d'apprentissage .....…...19
I.7.1. Règle d'apprentissage de Hebb ……19
I.7.2. Règle d'apprentissage du perceptron de F. Rosenblatt ……..20
I.7.3. Règle d'apprentissage de Windrow-Hoff (ADALINE) ……21
I.7.4. Algorithme de rétropropagation du gradient .. ……..…….22
I.7.4.1. Méthode de Newton ……..…..26
I.7.4.2. Méthode de quasi-Newton ……..…28
I.7.4.3. Méthode Levenberg-Marquardt ……..28
I.8. Théorie de la généralisation ……..28
I.9. Critère d'arrêt pour l'apprentissage ……..……30
I.10. Procédure de modélisation neuronale ……....31
I.10.1. Sélection des données ……..32

I.10.2. Collecte des données ……32
I.10.3. Séparation de la base de données ……..…32
I.10.4. Normalisation des données ……..33
I.10.5. Choix du type de réseau de neurones ……33
I.10.6. Choix du nombre de neurones dans la couche cachée ……33
I.10.7. Choix de la fonction de transfert ……..…33
I.10.8. Estimation des paramètres ou apprentissage ……...33
I.10.9. Validation et généralisation ……33
I.11 Conclusion ……..….32

Chapitre II L'intensification des échanges thermiques

II.1 Introduction…….…..35
II.2 Les échanges thermiques ……..35
II.2.1 Définition …….…..35
II.2.2 Principe de fonctionnement ……..….35
II.2.3 Types de contact ……..…..35
II.2.4 Disposition des écoulements …….……35
II.3 Différents types d'échange ……36
II-4 Intensification des échanges thermiques ……..36
II.5 Différentes techniques d'intensification ……...37
II.5.1 Les ailettes ....…….37
II.5.2 Les inserts ……..39
II.5.3 Les Surfaces rugueuses ……..39
II-6 Écoulement laminaire et turbulent ……...40
II-7 Conclusion ……..…….44

Chapitre III Résultats et discussions

III.1 Introduction …… ……45
III.2 Base de données ……46
III.3 Sélection des variables d'entrées et de sorties ……..…….….48
III.4 Collecte des données …….…….48
III.5 Mise en forme des données ……....68
III.6 Normalisation des données ……69
III.7 Conception des réseaux ……70
III.7.1. Type du réseau ……..70
III.7.2. Nombre de couches cachées ……..70
III.7.3. Algorithme d'apprentissage …….…….…71
III.7.4. Fonction d'activation ……..…71
III.7.5. Nombre d'itération …….……72
III.7.6. Apprentissage …….……....…72
III.7.7. Validation ou Généralisation…….……...……73
III.8 discussion des résultats ……..…..……...77
III.9 Conclusion…….……..……...77
Conclusion générale…….……..……..……...78
Références bibliographiques
Annexe

Liste des figures

Chapitre I Réseaux de neurones artificiels

Fig. I.1 : Schéma d'un neurone biologique ……..…….……..……..... 6
Fig. I.2 : Mise en correspondance neurone biologique / neurone artificiel.……..... 6
Fig. I.3 : Neurone formel ……..... 7
Fig. I.4 : Neurone formel modèle étendu ……... 8
Fig. I.5 : Fonction de transfert : (a) du neurone "seuil" ; (b) du neurone "linéaire" et (c) du neurone "sigmoïde"…….……..... 9
Fig. I.6 : Couche de *M* neurones ……..……... 10
Fig. I.7 : Représentation matricielle d'une couche de *M* neurones ……... 11
Fig. I.8 : Représentation matricielle d'un réseau de trois couches …….…… 11
Fig. I.9 : Elément de retard ……..……..…… 12
Fig. I.10 : Taxinomie des architectures des réseaux non bouclés et récurrents …… 12
Fig. I.11 : Principe d'un perceptron ……. 13
Fig. I.12 : Réseau Radial Basis Function (RBF)……..... 14
Fig. I.13 : Réseau de neurone bouclé à deux entrées …….. 15
Fig. I.14 : Modèle de Hopfield ……..... 16
Fig. I.15 : Réseau de type machine de Boltzmann ……..... 16
Fig. I.16 : Architecture d'un modèle de Kohonen ……..... 17
Fig. I.17 : Architecture d'un modèle d'ART ……... 17
Fig. I.18 : Illustration de l'apprentissage supervisé ……. 18
Fig. I.19 : Illustration de l'apprentissage non supervisé ……..... 19
Fig. I.20 : Séparation des exemples d'apprentissage par la droite de décision…… 20
Fig. I.21 : Sens de propagation du calcul des sorties s_j..... 23
Fig. I.22 : Sens de rétropropagation des gradients δ_j ……... 24
Fig. I.23 : Graphe de la surface d'erreur ……. 26
Fig. I.24 : Extraction de règles ……..... 29
Fig. I.25: a) Apprentissage correct, b) Sur-apprentissage, c) Sous-apprentissage …….. 29
Fig. I.26: Evolution des erreurs d'apprentissage et de test au cours du temps ……..... 30
Fig. I.27: Optimisation des paramètres d'un réseau de neurone artificiel ……... 34

Chapitre II L'intensification des échanges thermiques

Fig. II.1: différents Modes de circulation …….. 36
Fig. II.2: plaques à ailettes …….. 38
Fig. II 3: Différents types des inserts ……..... 39
Fig. II.4: Différents types de rugosité ……..... 40
Fig. II.5: Représentation géométrique des tubes ……..... 41
Fig. II.5: Différents types de rugosité structurée et leurs profilés …….. 45

Chapitre III Résultats et discussions

Fig. III.1 : Procédure de modélisation par les réseaux de neurones ……..... 57
Fig. III.2 : Schéma simplifié de l'architecture du réseau neuronal…….. 57
Fig. III.3: Courbe de régression linéaire lors de la phase d'apprentissage…….. 75
Fig. III.4: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.1…….. 76
Fig. III.5: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.2…….. 76
Fig. III.6: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.3…….. 77
Fig. III.7: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.4…….. 77
Fig. III.8: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.5…….. 78
Fig. III.9: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.6…….. 78

Liste des tableaux

Tableau I.1 : Fonctions de transfert S = f(P)……..…09
Tableau III.2 : Structure du réseau neuronal (ANN)…….74

Nomenclature

b_i	: Biais du neurone
D	: Vecteur de sortie désirée
d_i	: Sortie désirée
E	: Fonction de coût (fonction d'erreur)
F	: Fonction d'activation du neurone
$H(w^t)$	: Matrice Hessienne
I, J	: Matrice identité, Exposants
i	: Neurone i
j	: Neurone j
Nu	: Nombre de Nusselt
N	: Nombre d'entrées du neurone
n	: Coefficient
Pr	: Nombre de Prandtl
Re	: Nombre de Reynolds
s_i	: Sortie du neurone
w_i	: Poids synaptiques du neurone
X	: Vecteur d'entrée
w_{ij}^{t+1}	: Nouveau poids
w^*	: Poids minimum
w_{ij}^t	: Ancien poids
x_i	: Entrées du neurone
Δ	: Différence
δ_i	: Gradient
ε	: Erreur admissible, Constante diélectrique
ε_i	: Variable aléatoire
η^t	: Coefficient d'adaptation
λ	: Conductivité thermique
σ	: Tension superficielle
μ^t	: Pas de la descente
μ	: de modification des poids (pas d'apprentissage), Coefficient de Joule-Thomson
t	: Itération ou transposé
$\dot{Q}$	: Débit
Δw^t	: Adaptation des poids à l'itérationt

Abréviation

AAE : Average Absolute Error
Adaline : Adaptative Linear Element
AE : Absolute Error
ANN : Artificiel Neural Network
ART : Adaptive Resonance Theory
MLP : Multi-Layer Perceptron
PMC : Perceptron MultiCouches
RBF : Radial Basis Function
RMSE : Root Mean Square Error
RNA : Réseau de Neurones Artificiels

Application de la méthode neuronale pour la détermination des échanges thermiques intensifs

Résumé:

L'objectif de cette étude est la mise en œuvre d'une méthode de calcul basée sur les réseaux de neurones artificiels de type multicouche (MLP) pour déterminer avec précision le coefficient d'échange thermique et/ou le nombre de Nusselt concernant les échanges thermiques intensifs.

Dans une première étape, les réseaux de neurones sont élaborés pour chaque méthode d'intensification permettant, d'une part, après apprentissage, de reproduire les résultats de travaux antérieurs similaires avec des erreurs relatives n'excédant pas le 1%, d'autre part, d'exploiter soit les données expérimentales, soit les résultats obtenus à l'aide des corrélations empiriques traditionnelles.

Mots-clés : Intensification, échanges thermiques, corrélations empiriques, modélisation, réseau de neurones.

Application of the method neuronal for determining intensive heat exchanges

Abstract:

The objective of this study is the implementation of a calculation method based on artificial neural networks type multilayer (MLP) to accurately determine the heat transfer coefficient and / or the Nusselt number on trade intensive heat.

In a first step, neural networks are developed for each method to reproduce, on the hand, after learning, the results of previous work with similar relative errors which not exceed 1%, and on the other hand, to use either the experimental data or the results obtained using traditional empirical correlations.

Keywords: Intensification, heat exchange, empirical correlations, modeling, neural network.

تطبيق طريقة الشبكات العصبية لتحديد التبادلات الحرارية المكثفة

الملخص:

الهدف من هذه الدراسة هو تنفيذ طريقة الحساب على أساس الشبكات العصبية الاصطناعية إلى نوع متعدد الطبقات (MLP) لتحديد بدقة معامل تبادل الحرارة و / أو عدد Nusselt الذي يخص التبادلات الحرارية المكثفة.
في الخطوة الأولى، يتم تطوير الشبكات العصبية لكل طريقة التي تسمح من جهة، بعد التلقين، باستنساخ نتائج أعمال سابقة مماثلة مع أخطاء نسبية لا تزيد عن 1٪، و من جهة أخرى، إما باستخدام المعطيات التجريبية أو النتائج التي تم الحصول عليها باستخدام العلاقات التجريبية التقليدية.

الكلمات المفاتيح: تكثيف، التبادلات الحرارية، العلاقات التجريبية ، النمذجة، الشبكة العصبية.

Introduction Générale

Introduction générale

Dans beaucoup de procédés de transformation de matière et de l'énergie, les performances, et par voie de conséquence, la "compacité" du procédé sont régies par les flux de transferts aux interfaces. L'intensification ou accélération des transferts consiste à augmenter ces flux. Pour ce faire, on peut agir sur les conductances de transfert, sur les modes d'écoulement, sur la gestion spatio-temporelle des forces motrices, sur le mode d'apport d'énergie, sur la texture de l'interface, sur la configuration globale du procédé.

Pour présenter la thématique de recherche sur l'intensification des transferts, en particulier des transferts thermiques, il est nécessaire de préciser cette notion. Pour cela, nous pouvons définir différents objectifs de l'intensification :

- **L'intensification de productivité**, c'est l'objectif qu'on se donne, lorsqu'on cherche à optimiser les paramètres de fonctionnement pour un équipement existant
- **L'intensification dimensionnelle, ou " compacification "**, où le but est de diminuer la taille pour une puissance donnée
- **L'intensification énergétique**, dont l'objectif est de diminuer le débit de caloporteur, ou le niveau thermique du caloporteur, ou encore la perte de charge
- **Pour des procédés transitoires ou cycliques**, on peut rechercher une dynamique rapide, et on parlera alors de l'intensification temporelle
- **Pour des procédés plus généraux**, on pourrait même parler d'intensification fonctionnelle pour avoir moins de composants, d'étapes, de flux auxiliaires et de traitement aval.

Diverses techniques pour l’intensification des échanges thermiques sont apparues dans les dernières années. Cependant, les corrélations obtenues expérimentalement ont une allure éparse et ne permettent pas d’accéder au coefficient d’échange thermique de façon sure.

Les réseaux de neurones sont devenus une technique de traitement de données bien maitrisée et un outil de modélisation puissant dont les domaines d’application sont multiples. Ils permettent de réaliser de manière simple et efficace, des modèles précis, statiques ou dynamiques, parcimonieux en nombre de paramètres.

Les réseaux de neurones, fabriqués de structures cellulaires artificielles, constituent une approche permettant d'aborder sous des angles nouveaux les problèmes de perception, de mémoire, d'apprentissage et de raisonnement au même titre que les algorithmes génétiques. Ils s'avèrent aussi des alternatifs très prometteurs pour contourner certaines des limitations des

méthodes numériques classiques. Grâce à leur traitement parallèle de l'information et à leur mécanismes inspirés des cellules nerveuses (neurones), ils infèrent des propriétés émergentes permettant de solutionner des problèmes jadis qualifiés de complexes [I. VINCENT, 2005].

L'objectif du présent mémoire est de développer une nouvelle approche basée sur les réseaux de neurones pour le calcul des propriétés thermodynamique et de transport de l'eau aux états liquide et vapeur. Cette approche inspirée du traitement de l'information effectuée par le cerveau, semble offrir un nouvel espace de modélisation [A. SÖZEN, 2010]. Dans ce contexte, on a organisé le présent mémoire en en trois chapitres

Le chapitre 1, comporte un aperçu historique, la notion de neurone formel et les fonctions d'activation, le perceptron et les règles d'apprentissage, les perceptrons multicouches, l'algorithme de retropropagation du gradient et ses principales variantes, les principaux modèles de réseaux de neurones statique et dynamique et la procédure de développement d'un réseau de neurone statique, dynamique et/ou génétique.

Le chapitre 2, sera consacré à la description des principaux formalises des 16 méthodes d'intensifications des échanges thermiques c-à-d une revue critique sur l'intensification des échanges thermiques, les principales corrélations proposées dans la littérature et l'élaboration d'une base de données concernant le coefficient d'échange de chaleur et/ou le nombre de Nusselt.

Le chapitre 3, comportera l'exploitation de la méthode neuronale (statique, dynamique et/ou génétique), pour la détermination du coefficient de transfert thermique et/ou le nombre de Nusselt ; une fois pour chaque méthode et une seconde pour l'assemblage des 16 méthodes. Une comparaison avec les études antérieures issues de la synthèse bibliographique sera effectuée dans le but de valider la méthode et d'énoncer les critiques et perspectives qui en découlent.

On trouvera en conclusion une synthèse des différents travaux menés et des apports de ce mémoire. Enfin diverses perspectives de recherche ouvertes par ce travail seront exposées.

Chapitre I

Réseaux de Neurones Artificiels

I.1 Introduction

Un réseau de neurones artificiel (ou Artificial Neural Network en anglais) est un modèle de calcul utilisé en intelligence artificielle, et dont la conception est schématiquement inspirée du fonctionnement de vrais neurones (humains ou non), pour prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement. Le cerveau humain contient environ 100 milliards de neurones. Ces neurones nous permettent entre autre, de lire un texte tout en maintenant une respiration régulière permettant d'oxygéner notre sang, en actionnant notre cœur qui assure une circulation efficace de ce sang pour nourrir nos cellules, etc. Ils nous permettent même de comprendre certaines idées [I. Vincent, 2005]. Chacun de ces neurones est par ailleurs fort complexe. Essentiellement, il s'agit de tissu vivant et de chimie. Les neurophysiciens commencent à peine à comprendre quelques uns de leurs mécanismes internes. On croit en général que leur différentes fonctions neuronales, y compris celle de la mémoire sont stockées au niveau des connexions (synapses) entre les neurones [W. Gerstner et W.M. Kistler, 2002].

En voulant prendre sa source d'inspiration dans le monde de la biologie, en n'en gardant que les principes directeurs, les chercheurs ont inventé un grand nombre d'algorithmes différents. Si aujourd'hui, le modèle de neurone que l'ingénieur considérait comme biologiquement "plausible" est devenu très approximatif pour le biologiste, la biologie reste pourtant une source d'inspiration inépuisable, ainsi qu'en témoigne l'importante littérature qui lui est dédiée. Si l'on admet qu'un réseau de neurones multicouches comportant un nombre de neurones et de couches cachées suffisant peut approximer n'importe quelle fonction continue avec une précision arbitraire, il est rare que ce nombre soit compatible avec la quantité de données disponibles dans le cas d'une application réelle. C'est en essayant de s'affranchir des contraintes pesant sur les données que l'on a assisté au développement d'un grand nombre de méthodes baptisées "neuronales" [M.T. Hagan et H.B. Demuth, 2002]. La dimension des données influence également le type de solution envisagée pour un problème. Parce que les problèmes de classification sont d'autant plus difficiles que la dimension de l'espace est grande (du fait d'une lacunarité statistiquement croissante), le comportement de différents types d'algorithmes pourra être fort différent d'un cas à l'autre [T.L. Fine, 1999].

On abordera ici les principales architectures des réseaux de neurones. Il ne s'agit pas de les étudier toutes, car elles sont trop nombreuses, mais plutôt d'en comprendre les mécanismes internes fondamentaux et de savoir comment et quant les utiliser.

I.2. Historique

L'histoire de doter une machine d'intelligence est une idée qui fascine l'humanité depuis très longtemps. Dès l'Antiquité, les Grecs anciens rêvaient de robots. Selon la légende, Héphaïstos, le dieu du feu, aurait construit des tables à trois pieds munies de roulettes qui pouvaient aller et venir toutes seules dans les palais des dieux. L'histoire des réseaux de neurones est jalonnée d'un certain nombre de publications clés, livres ou articles d'intérêt essentiellement historique, associées aux étapes décisives de leur développement.

- **Les premiers pas**

1890 : W. James, célèbre psychologue américain introduit le concept de mémoire associative, et propose ce qui deviendra une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones connue plus tard sous le nom de loi de Hebb [D.O. Hebb, 1949].

1943 : J. Mc Culloch et W. Pitts laissent leurs noms à une modélisation du neurone biologique (un neurone au comportement binaire). Ce sont les premiers à montrer que des réseaux de neurones formels simples peuvent réaliser des fonctions logiques, arithmétiques et symboliques complexes (tout au moins au niveau théorique) [W.S. Mc Culloch et W.H. Pitts, 1943].

1949 : D. Hebb, physiologiste américain explique le conditionnement chez l'animal par les propriétés des neurones eux-mêmes. Ainsi, un conditionnement de type pavlovien, tel que nourrir tous les jours à la même heure un chien, entraîne chez cet animal la sécrétion de salive à cette heure précise même en l'absence de nourriture. La loi de modification des propriétés des connexions entre neurones qu'il propose, explique en partie ce type de résultats expérimentaux [D.O. Hebb, 1949].

- **Les premiers succès**

1957 : F. Rosenblatt développe le modèle du Perceptron. Il construit le premier neuro-ordinateur basé sur ce modèle et l'applique au domaine de la reconnaissance de formes. Notons qu'à cette époque les moyens mis à sa disposition sont limités et c'est une prouesse technologique de réussir à faire fonctionner correctement cette machine pendant quelques minutes [F. Rosenblatt, 1958].

1960 : B. Widrow, un automaticien, développe le modèle Adaline (Adaptative Linear Element). Dans sa structure, le modèle ressemble au Perceptron. Cependant la loi d'apprentissage est différente. Celle-ci est à l'origine de l'algorithme de rétropropagation de gradient très utilisé aujourd'hui avec les Perceptrons multicouches. Les réseaux de type Adaline restent utilisés de

nos jours pour certaines applications particulières. B. Widrow a créé dès cette époque une des premières firmes proposant neuro-ordinateurs et neuro-composants, la "Memistor Corporation" [B. Widrow et M.E. Hoff, 1960].

1969 : M. Minsky et S. Papert publient un ouvrage qui met en exergue les limitations théoriques du perceptron. Limitations alors connues, notamment concernant l'impossibilité de traiter par ce modèle des problèmes non linéaires. Ils étendent ces limitations à tous les modèles de réseaux de neurones artificiels. Après, les recherches dans le domaine furent abandonnées par manque de financement et les chercheurs se tournèrent principalement vers les systèmes à bases de règles [M. Minsky et S. Papert, 1969].

- **L'ombre**

1967-1982 : Toutes les recherches ne sont, bien sûr, pas interrompues. Elles se poursuivent, mais sous le couvert de divers domaines comme : le traitement adaptatif du signal, la reconnaissance de formes, la modélisation en neurobiologie, etc. De grands noms travaillent durant cette période tels: S.Grosberg, T.Kohonen, etc. [S. Grosberg, 1977], [T. Kohonen, 1982].

- **Le renouveau**

1982 : J. J. Hopfield est un physicien reconnu, à qui l'on doit le renouveau d'intérêt pour les réseaux de neurones artificiels [J.J. Hopfield, 1982].

- **La levée des limitations**

1983 : La Machine de Boltzmann est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement lente (les temps de calcul sont considérables) [G.E.Hinton et T.J. Sejnowski, 1983], [D.E. Rumelhart et al., 1986].

1985 : La rétropropagation de gradient apparaît. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches (aussi appelés perceptrons multicouches). Sa découverte réalisée par trois groupes de chercheurs indépendants indique que "la chose était dans l'air". Dès cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables. De nos jours, les réseaux multicouches et la rétropropagation de gradient reste le modèle le plus étudié et le plus productif au niveau des applications [C. Touzet, 1992].

I.3. Neurones biologiques

Les cellules nerveuses, appelées neurones, sont les éléments de base du système nerveux central. Les neurones possèdent de nombreux points communs dans leur organisation générale et leur système biologique, avec les autres cellules. Ils présentent cependant des caractéristiques qui leur sont propres et se trouvent dans cinq fonctions spécialisées qu'ils assurent [R.L. Harvey 1994] :

- Recevoir des signaux en provenance de neurones voisins,
- Intégrer ces signaux,
- Engendrer un flux nerveux,
- Conduire le flux nerveux et le transmettre à un autre neurone capable de le recevoir.

Donc le neurone biologique est une cellule vivante spécialisée dans le traitement des signaux électriques. Les neurones sont reliés entre eux par des liaisons appelées axones. Ces axones vont eux-mêmes jouer un rôle important dans le comportement logique de l'ensemble. Ces axones conduisent les signaux électriques de la sortie d'un neurone vers l'entrée (synapse) d'un autre neurone.

Les neurones font une sommation des signaux reçus en entrée et en fonction du résultat obtenu vont fournir un courant en sortie. La structure d'un neurone se compose de trois parties (Figure I.1) :

- Le corps cellulaire : ou encore la somma, est la cellule d'activité nerveuse, au centre du neurone.
- L'axone : attaché au somma qui est électriquement actif, ce dernier conduit l'impulsion conduite par le neurone.
- Dendrites : électriquement passives, elles reçoivent les impulsions d'autres neurones.

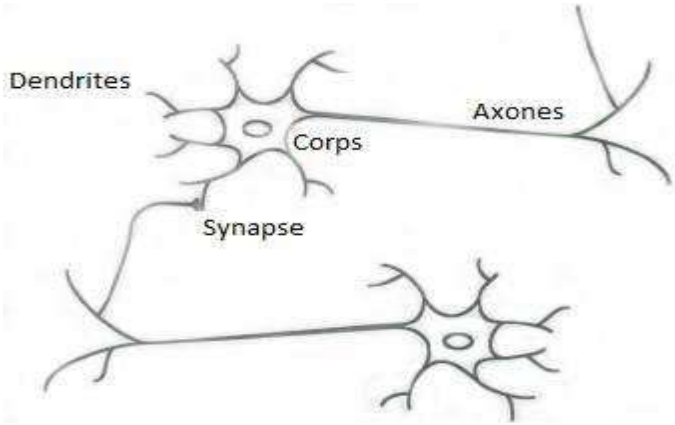


Fig. I.1 : Schéma d'un neurone biologique [M.T. Hagan et H.B. Demuth, 1996]

I.4. Neurone formel

La modélisation consiste à mettre en œuvre un système de réseaux neuronaux sous un aspect non pas biologique mais artificiel, cela suppose que d'après le principe biologique on aura une correspondance pour chaque élément composant le neurone biologique, donc une modélisation pour chacun d'entre eux (Figure I.2).

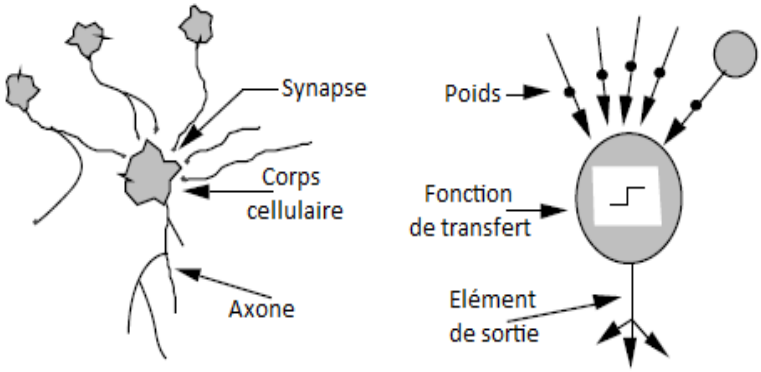


Fig. I.2 : Mise en correspondance neurone biologique / neurone artificiel [T.L. Fine, 1999]

Les réseaux de neurones formels sont à l'origine d'une tentative de modélisation mathématique du cerveau humain. Le premier neurone formel à été modélisé par W. Mc Culloch et W. Pitts, en 1943 [W.S. Mc Culloch et W.H. Pitts, 1943], [E. Davalo et P. Naïm, 1993]. Un neurone formel ou artificiel est un opérateur mathématique très simple. Un neurone possède des entrées qui peuvent être les sorties d'autres neurones, ou des entrées de signaux extérieures, et une sortie. La valeur de la sortie résulte du calcul de la somme des entrées pondérées par des coefficients (dits poids de connexions ou poids synaptiques) et du calcul d'une fonction non linéaire (dite fonction d'activation) de cette somme pondérée (Figure I.3). Un neurone formel peut être défini par les cinq éléments suivants [E. Davalo et P. Naïm, 1993] :

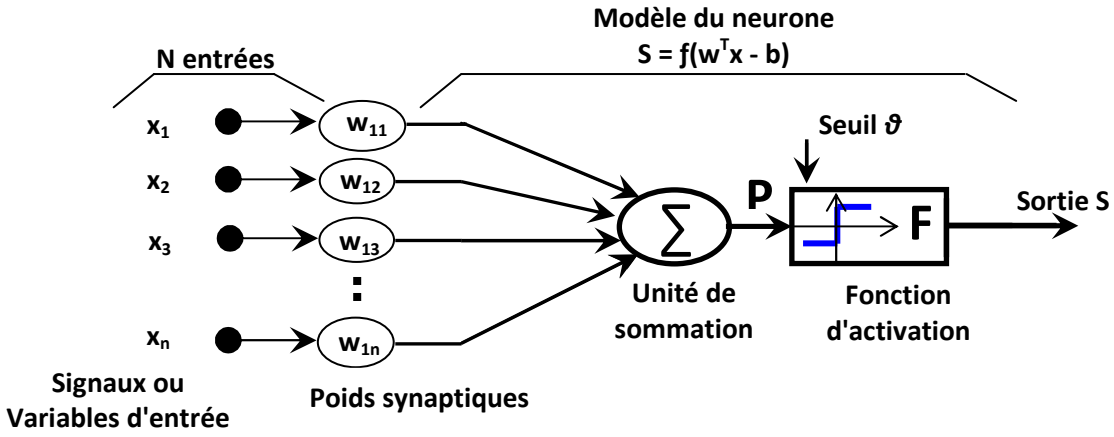


Fig. I.3 : Neurone formel [E. Davalo, 1993]

I.4.1. Entrées Xi

Les valeurs d'entrée sont des stimuli externes en provenance de l'environnement ou des sorties d'autres neurones artificiels. Elles peuvent être [D. Hassani, 2003] :

- Booléennes,
- Binaires (0, 1) ou bipolaires (-1, 1),
- Réelles.

I.4.2. Poids synaptiques Wi

Les paramètres w_n (Figure I.3) sont souvent désignés sous le nom de "poids" ou, en raison de l'inspiration biologique des réseaux de neurones, "poids synaptiques". Ce sont des valeurs réelles qui déterminent la contribution de chaque entrée. Un algorithme d'apprentissage permet de déterminer le meilleur ensemble de poids pour le problème considéré. Trouver l'ensemble optimal est souvent un compromis entre temps de calcul, minimisation de l'erreur et conservation de la capacité de généralisation pour le réseau.

La première chose que fait un neurone formel est de calculer la somme pondérée des poids et des entrées. Conformément à l'usage (également inspiré par la biologie), cette somme pondérée est appelée potentiel (niveau d'activation), à laquelle se soustrait souvent un terme constant b appelé biais (seuil d'activation).

$$P = \sum_{j=1}^{n-1} w_{1j}\, x_j - b \qquad \text{(I.1)}$$

Les N entrées du neurones correspondent au vecteur $X = [x_1, x_2,\ldots x_n]^T$, alors que $w = [w_{1,1} w_{1,2} \cdots w_{1,n}]^T$ représente le vecteur des poids du neurone. La sortie P de l'intégrateur est donnée par l'équation (I. 1) que l'on peut aussi écrire sous la forme matricielle suivante :

$$P = W^T \qquad \text{(I. 2)}$$

I.4.3. Seuil θ

Le seuil est un nombre réel qui est soustrait de la somme pondérée des valeurs d'entrée. Pour des raisons de simplicité, le seuil peut être vu comme une entrée supplémentaire associée à un poids, où (w = θ) et (x = -1), tel qu'il est illustré par la Figure suivante :

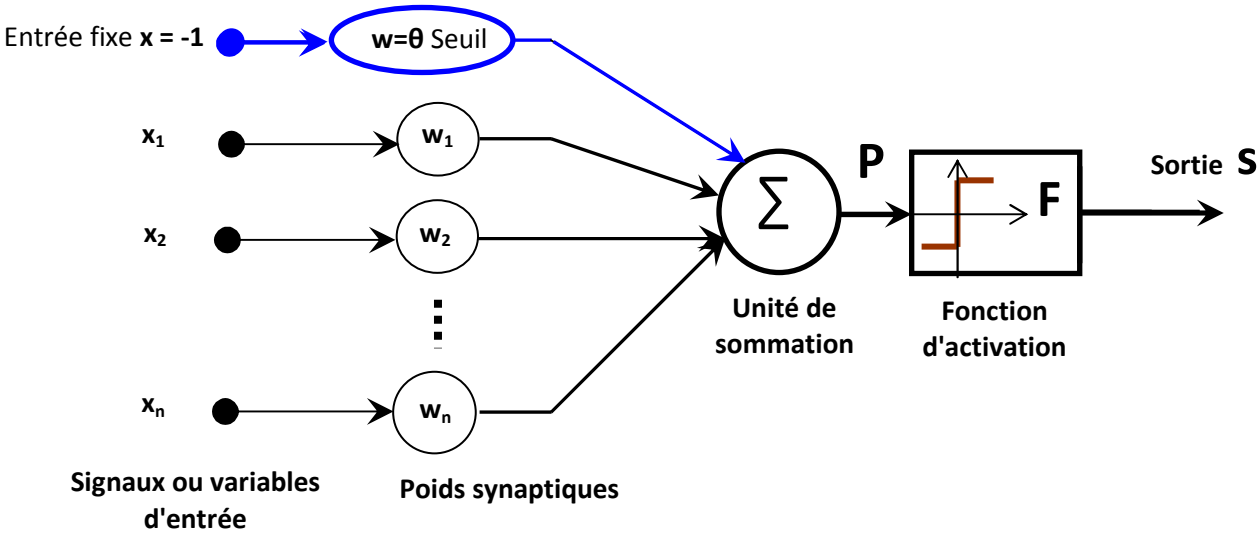


Fig. I.4 : Neurone formel modèle étendu [A. Bouzidi, 2006]

I.4.4. Fonction d'activation F

La fonction d'activation ou de transfert calcule la valeur de l'état du neurone. C'est cette valeur qui sera transmise aux neurones avals. Il existe de nombreuses formes possibles pour la fonction de transfert. A la différence des neurones biologiques dont l'état est binaire, la plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprises dans l'intervalle [0, +1] ou [-1, +1].

Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone sont énumérées au tableau I.1. Les trois les plus utilisées sont les fonctions "seuil" (en anglais "hard limit"), "linéaire" et "sigmoïde".

Comme son nom l'indique, la fonction seuil applique un seuil sur son entrée. Plus précisément, une entrée négative ne passe pas le seuil, la fonction retourne alors la valeur 0 (on peut interpréter ce 0 comme signifiant faux), alors qu'une entrée positive ou nulle dépasse le seuil, et la fonction retourne 1 (vrai). Utilisée dans le contexte d'un neurone, cette fonction est illustrée à la figure I.5a. On remarque alors que le biais *b* dans l'expression de s = hardlim($w^T x - b$) (équation I.2) détermine l'emplacement du seuil sur l'axe $w^T x$, où la fonction passe de 0 à 1. Cette fonction permet de prendre des décisions binaires. La fonction linéaire est très simple, elle affecte directement son entrée à sa sortie :

$$S = P \tag{I.3}$$

Appliquée dans le contexte d'un neurone, cette fonction est illustrée à la figure I.5b. Dans ce cas, la sortie du neurone correspond à son niveau d'activation dont le passage à zéro se produit lorsque $w^T x = b$. La fonction de transfert sigmoïde est quant à elle illustrée à la figure I.5c. Son équation est donnée par :

$$s = \frac{1}{1 + e^{-p}} \tag{I.4}$$

Tableau I.1 : Fonctions de transfert S = f(P) [M. Parizeau, 2004]

Nom de la fonction	**Relation d'entrée/sortie**	**Icône**	**Nom MATLAB**
Seuil	$s = 0$ *si* $p < 0$ $s = 1$ *si* $p \geq 0$		hardlim
Seuil symétrique	$s = -1$ *si* $p < 0$ $s = 1$ *si* $p \geq 0$		hardlims
Linéaire	$s = n$		purelin
Linéaire saturée	$s = 0$ *si* $p < 0$ $s = p$ *si* $0 \leq p \leq 1$ $s = 1$ *si* $p > 1$		satlin
Linéaire saturée symétrique	$s = -1$ *si* $p\, n < -1$ $s = p$ *si* $-1 \leq p \leq 1$ $s = 1$ *si* $p > 1$		satlins

Linéaire positive	$s = 0$ *si* $p < 0$ $s = p$ *si* $p \geq 0$		poslin
Sigmoïde	$s = \frac{1}{1+e^{-p}}$		logsig
Tangente hyperbolique	$s = \frac{e^{p}-e^{-p}}{e^{p}+e^{-p}}$		tansig
Compétitive	$s = 1$ *si n maximum* $s = 0$ *autrement*		compet

Elle ressemble soit à la fonction seuil, soit à la fonction linéaire, selon que l'on est loin ou près de b, respectivement. La fonction seuil est très non-linéaire car il y a une discontinuité lorsque $w^T x = b$. De son côté, la fonction linéaire est tout à fait linéaire. Elle ne comporte aucun changement de pente. La sigmoïde est un compromis intéressant entre les deux précédentes. Notons finalement, que la fonction "tangente hyperbolique" est une version symétrique de la sigmoïde.

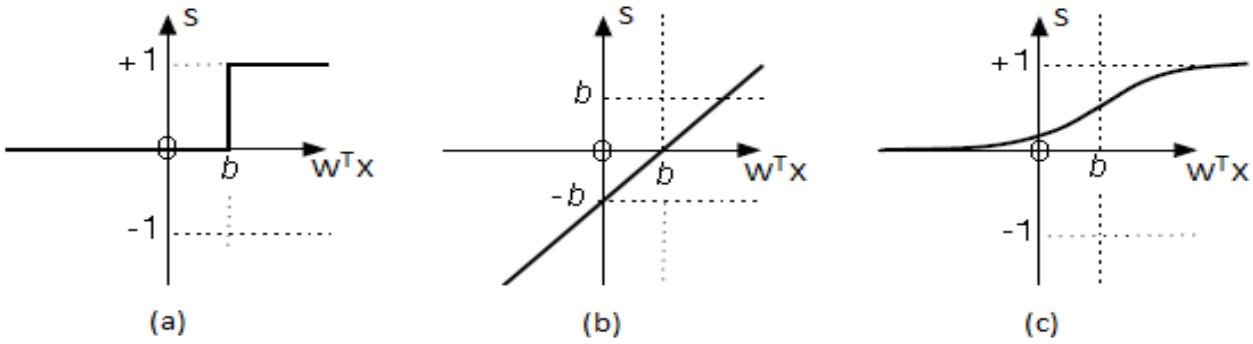


Fig. I.5 : Fonction de transfert : (a) du neurone "seuil" ; (b) du neurone "linéaire" et (c) du neurone "sigmoïde" [I. Vincent, 2005]

I.4.5. Sortie du neurone S

La sortie du neurone n_i n'est autre que l'application de la fonction de transfert à la somme pondérée de ces entrées selon l'équation suivante :

$$S = F\left(\sum_{j=1}^{n-1} w_{1j}\, x_j - b\right) \qquad \text{(I.5)}$$

La sortie calculée, qui peut être discrète ou réelle selon la fonction d'activation utilisée, passe aux autres neurones ou directement à l'environnement extérieur.

I.5. Architecture du réseau

Par définition, un réseau de neurones est un maillage de plusieurs neurones, généralement organisés en couches. Pour construire une couche de M neurones, il s'agit simplement de les assembler comme à la figure ci-dessous [M. Parizeau, 2004] :

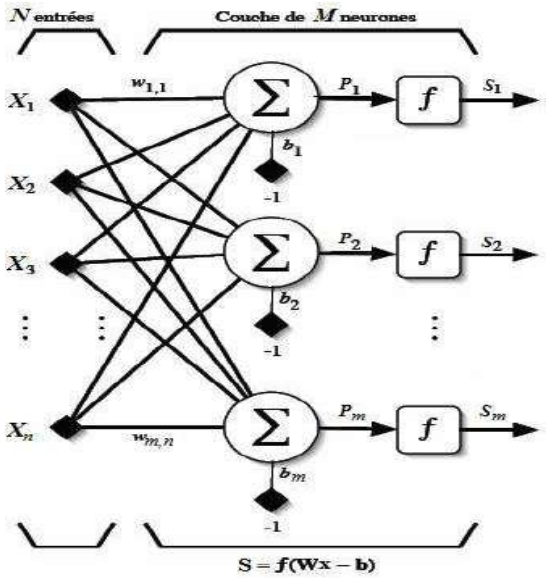


Fig. I.6 : Couche de *M* neurones [H. Demuth et al., 2009]

Les *M* neurones d'une même couche sont tous branchés aux *N* entrées. Nous disons alors que la couche est "totalement connectée". Un poids $W_{i,j}$ est associé à chacune des connexions. Nous noterons toujours le premier indice par *i* et le deuxième par *j* (jamais l'inverse). Le premier indice (rangée) désigne toujours le numéro de neurone sur la couche, alors que le deuxième indice (colonne) spécifie le numéro de l'entrée. Ainsi, désigne le poids de la connexion qui relie le neurone *i* à son entrée *j*. L'ensemble des poids d'une couche forme donc une matrice *W* de dimension :

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w_{1,1}} & \boldsymbol{w_{1,2}} & \boldsymbol{w_{1,n}} \\ \boldsymbol{w_{2,1}} & \boldsymbol{w_{2,2}} & \boldsymbol{w_{2,n}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{w_{m,1}} & \boldsymbol{w_{m,2}} & \boldsymbol{w_{m,n}} \end{bmatrix} \qquad \text{(I.6)}$$

Il faut bien sûr prendre en compte que nous n'avons pas nécessairement *M=N* dans le cas général (les nombres de neurones et d'entrées sont indépendants). Si nous considérons que les *M* neurones forment un vecteur de neurones, alors nous pouvons créer les vecteurs :

$$\vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \quad \vec{P} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{pmatrix} \quad \vec{S} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix} \tag{I.7}$$

Ceci nous amène à la représentation simplifiée illustrée ci-dessous :

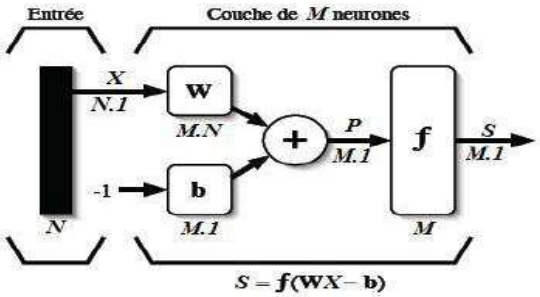


Fig. I.7 : Représentation matricielle d'une couche de *M* neurones [H. Demuth et al., 2009]

Pour construire un réseau de neurones (ou PMC pour "Perceptron Multicouches"), il ne suffit plus que de combiner des couches comme ci-dessous :

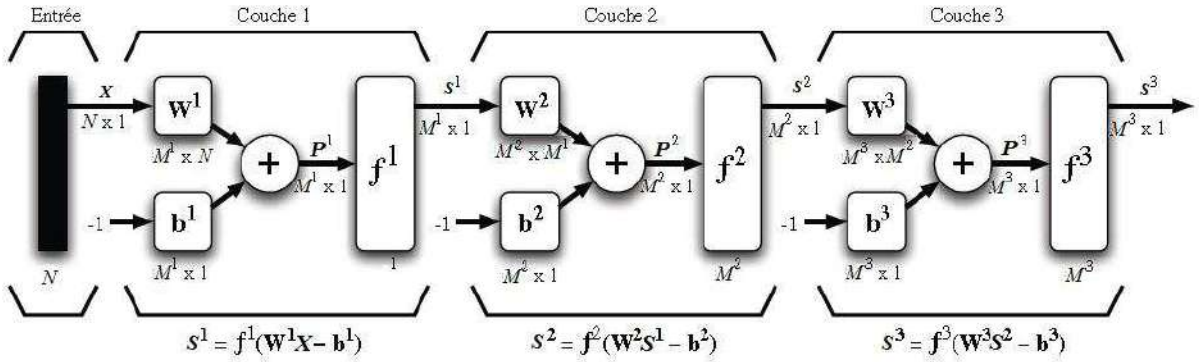


Fig. I.8 : Représentation matricielle d'un réseau de trois couches [H. Demuth et al., 2009]

Cet exemple comporte *N* entrées et trois couches de neurones comptant respectivement M^1, M^2, M^3 neurones. Dans le cas général, de nouveau ces nombres ne sont pas nécessairement égaux. Chaque couche possède aussi sa propre matrice de poids W^K, où *k* désigne l'indice de couche. Dans le contexte des vecteurs et des matrices relatives à une couche, nous emploierons toujours un exposant pour désigner cet indice. Ainsi, $\overrightarrow{b^k}$, $\overrightarrow{p^k}$, $\overrightarrow{s^k}$ les vecteurs sont aussi associés à la couche *k*. Il importe de remarquer dans cet exemple que les couches qui suivent la première ont comme entrée la sortie de la couche précédente. Ainsi, nous pouvons enfiler autant de

couches que nous voulons, du moins en théorie. On peut fixer un nombre quelconque de neurones sur chaque couche. En pratique, on verra plus tard qu'il n'est cependant pas souhaitable d'utiliser trop de neurones.

Noter aussi que rien n'empêche de changer de fonction de transfert d'une couche à l'autre. Ainsi, dans le cas général nous n'avons pas nécessairement $f^1=f^2=f^3$.

La dernière couche est nommée "couche de sortie". Les couches qui précèdent la couche de sortie son nommées "couches cachées". Le réseau de la figure I.8 possède donc deux couches cachées et une couche de sortie.

Les réseaux multicouches sont beaucoup plus puissants que les réseaux simples à une seule couche bien évidemment. En utilisant deux couches, à condition d'employer une fonction d'activation sigmoïde sur la couche cachée, nous pouvons "entraîner" un réseau à produire une approximation de la plupart des fonctions, avec une précision arbitraire. Sauf dans de rares cas, les réseaux de neurones formels exploitent deux ou trois couches. Entraîner un réseau de neurones signifie modifier la valeur de ses poids et de ses biais pour qu'il réalise la fonction d'entrée sortie (I/O).

Finalement, la figure I.9 illustre le dernier élément de construction que nous emploierons pour bâtir des réseaux dit "récurrents". Il s'agit d'un registre à décalage qui permet d'introduire un retard dans une donnée que l'on veut acheminer dans un réseau. La sortie retardée s (t) prend la valeur de l'entrée u au temps t-1. Cet élément de retard présuppose que l'on peut initialiser la sortie au temps t = 0 avec la valeur s(0). Cette condition initiale est indiquée à la figure I.9 par une flèche qui entre par le bas de l'élément.

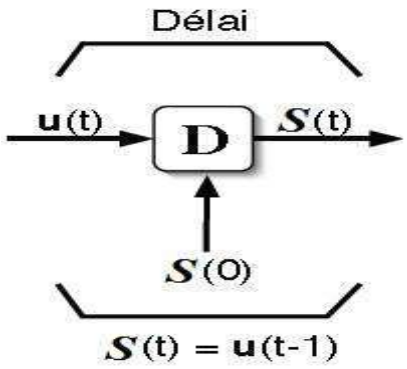


Fig. I.9 : Elément de retard [H. Demuth et al., 2009]

Selon le mode de connexion des neurones, On distingue les réseaux suivants (Figure I.10) [A.K. Jain et J. Mao, 1996]:

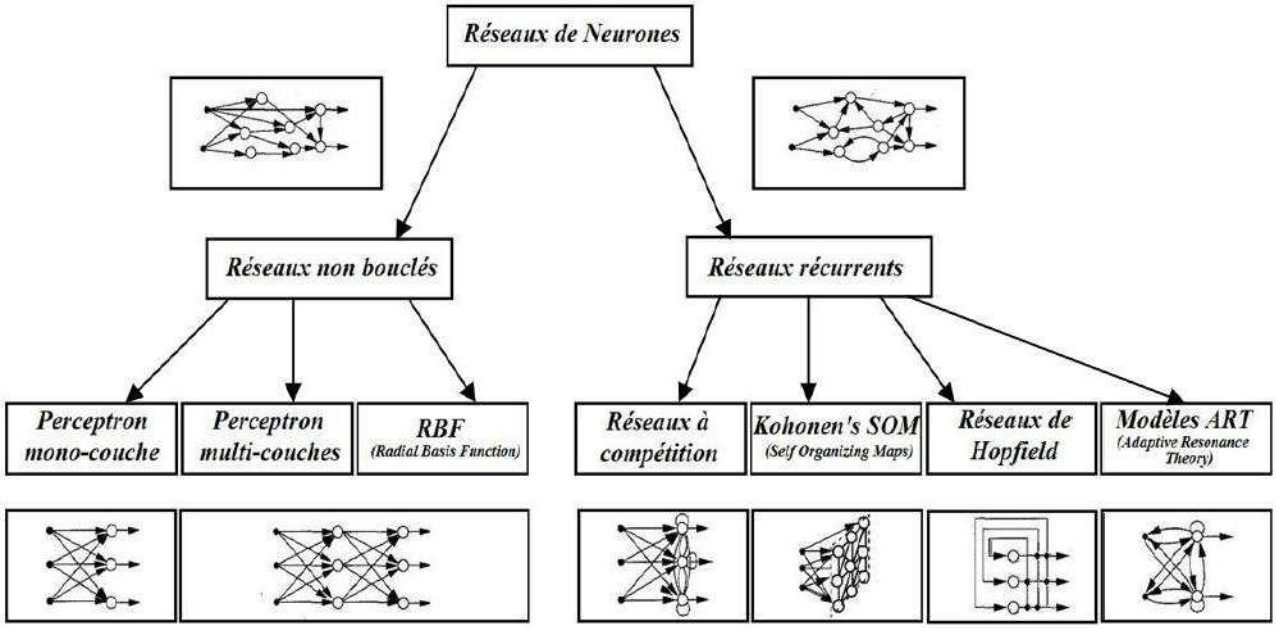


Fig. I.10 : Taxinomie des architectures des réseaux non bouclés et récurrents [A.K. Jain et J. Mao, 1996]

I.5.1. Réseau de neurones non bouclés

La définition de ce type de réseaux selon G. Dreyfus [G. Dreyfus et al., 2002] est : "Un réseau de neurones non bouclé réalise une (ou plusieurs) fonction algébriques de ses entrées, par composition des fonctions réalisées par chacun des neurones".

Un réseau de neurones non bouclé est donc représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans retour en arrière : si l'on se déplace dans le réseau, à partir d'un neurone quelconque, en suivant les connexions, on ne peut pas revenir au neurone de départ. Les neurones qui effectuent le dernier calcul de la composition des fonctions sont les neurones de sortie ; ceux qui effectuent des calculs intermédiaires sont les neurones cachés.

Il convient d'insister sur le fait que le temps ne joue aucun rôle fonctionnel dans le réseau de neurones non bouclé : si les entrées sont constantes, les sorties le sont également. Le temps nécessaire pour le calcul de la fonction réalisée par chaque neurone est négligeable et, fonctionnellement, on peut considérer ce calcul comme instantané. Pour cette raison, les réseaux non bouclés sont souvent appelés "réseau statique", par opposition aux réseaux bouclés (récurrents) ou dynamique que nous introduirons plus loin.

Le domaine d'application par excellence des réseaux non bouclé, dont l'apprentissage est souvent supervisé, est la modélisation statique type boite noire. En effet, si on dispose d'un

ensemble de variables mesurées et d'un ensemble de mesures d'une grandeur relative à un processus de nature quelconque (physique, chimique, économique, financier…) et en supposant qu'il existe une relation (suffisamment régulière) entre le vecteur des variables et la grandeur en question, alors il est possible d'approximer cette relation par un réseau de neurone non bouclé. Selon le nombre de couches dans un réseau et la fonction d'activation, on distingue :

I.5.1.1. Perceptron

C'est l'un des premiers réseaux de neurones, conçu en 1958 par F.Rosenblatt [F. Rosenblatt, 1958]. Il est linéaire et représente l'unité élémentaire d'un réseau de neurones.

Le perceptron est inspiré du système visuel. La première couche ou couche d'entrée représente la rétine, et la couche de sortie est équivalente aux cellules de décision.

Un Perceptron calcule (Equation I.5) à partir d'un vecteur de n entrées réelles [x1 x2… xn], une combinaison linéaire de ces entrées ; le résultat P de cette combinaison est ensuite utilisé comme argument d'une fonction F d'une variable pour délivrer la valeur de sortie s (Figure I.11).

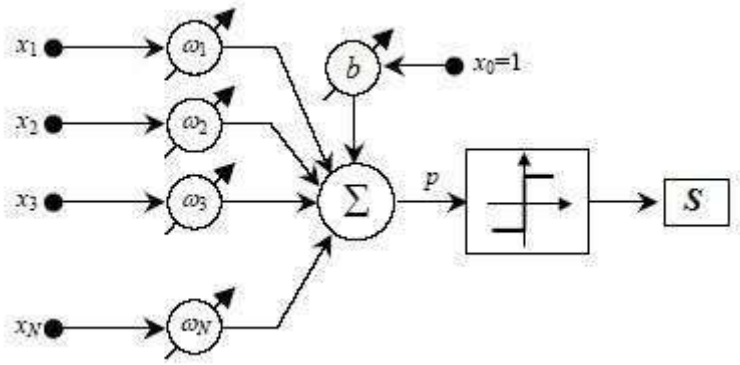


Fig. I.11 : Principe d'un perceptron [G. Dreyfus et al., 2002]

I.5.1.2. Perceptron multicouche (MLP)

Les réseaux neurones multicouches contiennent en plus des couches d'entrée et de sortie, un certain nombre (supérieur ou égal à un) de couches intermédiaires ou cachées dont les neurones ont une fonction d'activation sigmoïde. Les liens n'existent évidemment qu'entre les cellules d'une couche avec les cellules de la couche suivante (réseau non bouclé). De telles architectures sont habituellement désignées dans la littérature sous le nom "Multi-Layer Perceptron" MLP (figure I.8) [G. Dreyfus et al., 2002].

Les MLP possédant un seul neurone de sortie linéaire, peuvent approcher toute fonction bornée suffisamment régulière, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables [K. Hornik, 1991]. Néanmoins, cette propriété, qui n'est qu'un théorème d'existence et

ne donne pas la méthode de trouver les paramètres du réseau, n'est pas spécifique aux réseaux de neurones. C'est la propriété de parcimonie qui leur est particulière et fait tout leur intérêt.

Les MLP sont, en effet, des approximateurs parcimonieux, c'est-à-dire ; que le nombre des paramètres de pondération ou poids, pour une précision donnée, croit linéairement avec le nombre des variables d'entrées du modèle. La parcimonie, qui est une propriété fondamentale des MLP, est donc d'autant plus importante que le nombre des entées d'un modèle est grand [G. Dreyfus et al., 2008].

I.5.1.3. Réseaux RBF (Radial Basis Function)

Ces réseaux ont été introduits par M. Powel [M. Powel, 1985] et D. Broomhead [D. Broomhead et D. Lowe, 1988]. Il s'agit d'une "spécialisation" d'un perceptron multicouche (MLP). Les neurones de ces réseaux sont actifs pour une zone restreinte de l'espace d'entrée qui est définie par des fonctions radiales. Cette technique consiste à utiliser la non linéarité d'une fonction (par exemple une gaussienne) pour ponctuer la réponse des neurones des couches cachées (figure I.12) [A. Bouzidi, 2006], [R. Zemouri et al., 2002], [M.R. Berthold et J. Diamond, 1995].

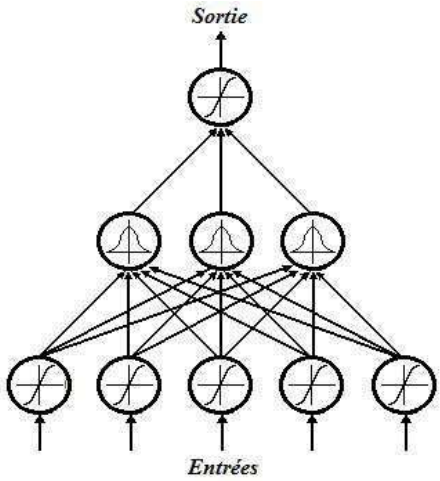


Fig. I.12 : Réseau Radial Basis Function (RBF) [R. Zemouri et al., 2002]

I.5.2. Réseaux de neurones bouclés (récurrents)

Un réseau de neurone est dit bouclé ou récurrent lorsque des sorties de neurones d'une couche sont connectées aux entrées d'une couche d'indice inférieur. Autrement dit, lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ.

La sortie d'un neurone du réseau peut donc être fonction d'elle-même ; cela n'est évidemment concevable que si la notion de temps est explicitement prise en considération. Raison pour la quelle ce type de réseaux sont appelés dynamique.

Ainsi, à chaque connexion d'un réseau de neurones bouclé est attaché, outre un poids comme pour les réseaux non bouclé, un retard, multiple entier de l'unité de temps choisie [G. Dreyfus et al., 2008]. Une grandeur, à un instant donné, ne pouvant pas être fonction de sa propre valeur au même instant, tout cycle du graphe du réseau doit avoir un retard non nul.

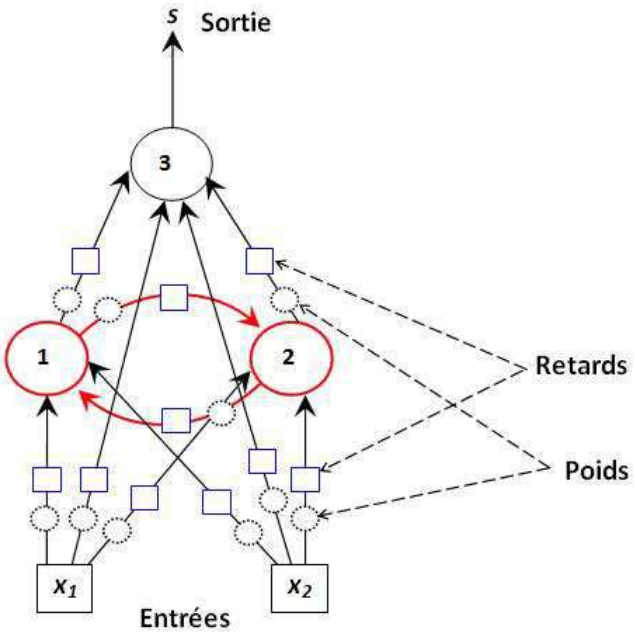


Fig. I.13 : Réseau de neurone bouclé à deux entrées [G. Dreyfus et al., 2008]

Les réseaux de neurones bouclés sont mise en œuvre essentiellement pour concevoir des modelés dynamiques. Ils peuvent être utilisés à titre d'exemples ; comme modèle de simulation pour prévoir l'évolution d'un processus quelconque dans le temps ou comme prédicteur à très court terme (une période d'échantillonnage) afin de l'intégrer à un système de commande [G. Dreyfus et al., 2008]. Selon la complexité de son connexionnisme, on distingue :

I.5.2.1. Modèle de Hopfield

Le modèle élaboré par J.Hopfield [J.J. Hopfield, 1982], est totalement interconnecté c'est le réseau bouclé par excellence (figure I.14). Il n'y a plus de notions de couche, du fait que chaque neurone du réseau est a la fois une entrée et une sortie, le réseau implantant ainsi une mémoire associative (adressables par le contenu), c'est-à-dire qu'on lui donne des exemples et il doit être capable de les reconnaître lorsqu'ils sont bruités. La question qui se pose alors est : quelle est la

capacité de stockage d'un tel réseau? Des recherches ont prouvé que le rapport optimal entre le nombre d'exemples et le nombre de neurones est de 0,14 [A.I. Galushkin, 2007].

Dès que le nombre d'exemples devient trop grand, le réseau oublie tout ce qu'il a appris. Le processus d'apprentissage fait appel à la règle de Hebb [J. Sarangapani, 2006], [D.O. Hebb, 1949] qui renforce la connexion entre deux neurones s'ils sont actifs simultanément. Ce type de réseau est utilisé essentiellement dans la reconnaissance de la parole et la reconnaissance des formes [L. Bottou, 1991].

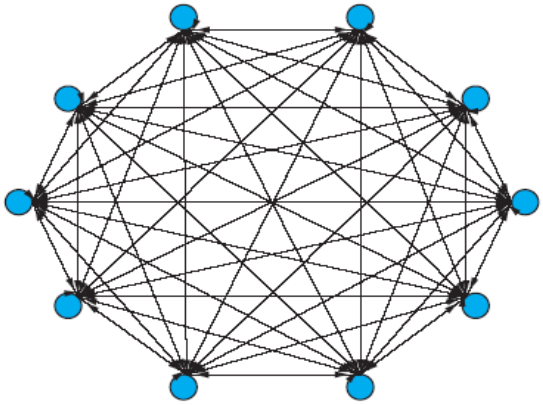


Fig. I.14 : Modèle de Hopfield [L. Bottou, 1991]

I.5.2.2. Modèle de Boltzmann

La machine de Boltzmann qui a été introduite en 1984 par G.E.Hinton et al. [G.E. Hinton et al., 1984], est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Il s'agit d'un réseau fortement récurrent au même titre que les réseaux de Hopfield. Mais à la différence de ces derniers, le réseau est divisé en trois parties distinctes et les entrées et sorties sont clairement séparées et identifiées [H. Abdi, 1994].

L'utilisation pratique de ces réseaux s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables). Il est souvent employé dans le domaine de reconnaissance de la parole ou de la vision [L. Bottou, 1991].

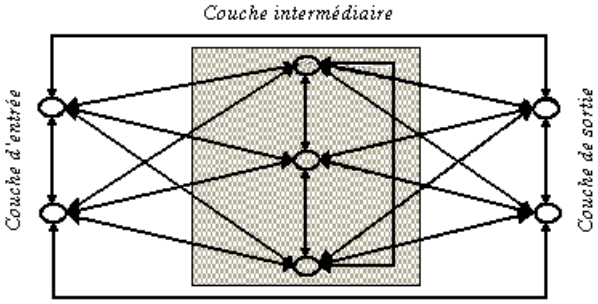


Fig. I.15 : Réseau de type machine de Boltzmann [H. Abdi, 1994]

I.5.2.3. Modèle de Kohonen

La carte de Kohonen [T. Kohonen, 1984] est en général à deux dimensions. Chaque neurone de la couche d'entrée est relié à chaque neurone de la carte de Kohonen (lien 1 vers tous). Chaque neurone de la carte de Kohonen est relié à tous les neurones de la carte. Les poids sont déterminés en fonction de la distance selon une fonction "chapeau mexicain" ou DOG (Difference of Gaussians). Ces réseaux fonctionnent dans leur très grande majorité dans un mode non supervisé. Il est donc relativement difficile de lui faire apprendre nos bases [E. Davalo et P. Naïm, 1993].

Les réseaux de Kohonen trouvent leurs champs d'application dans plusieurs domaine tels : la classification, le traitement de l'image, la reconnaissance de la parole et l'optimisation [L. Bottou, 1991].

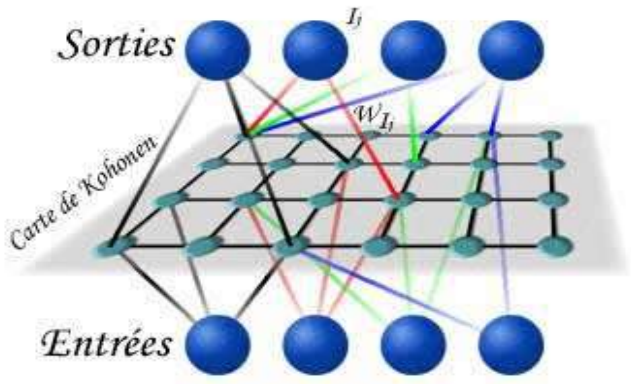


Fig. I.16 : Architecture d'un modèle de Kohonen [E. Davalo et P. Naïm, 1993]

I.5.2.4. Modèles ART

ART (Adaptive Resonance Theory) est un modèle de réseau de neurones à architecture évolutive développé en 1987 par Carpenter et Grossberg [B. Krose et P. Vander Smagt, 1996]. Dans la plupart des réseaux de neurones, deux étapes sont considérées. La première est la phase d'apprentissage : les poids des connexions sont modifiés selon une règle d'apprentissage, la deuxième est la phase d'exécution où les poids ne sont plus modifiés. Avec le réseau ART, ces deux étapes sont réalisées simultanément. Le réseau en phase de test, s'adapte à des entrées inconnues en construisant de nouvelles classes (ajout de neurones) tout en dégradant au minimum les informations déjà mémorisées. Il existe plusieurs versions de réseaux (ART1, ART2, ART3).

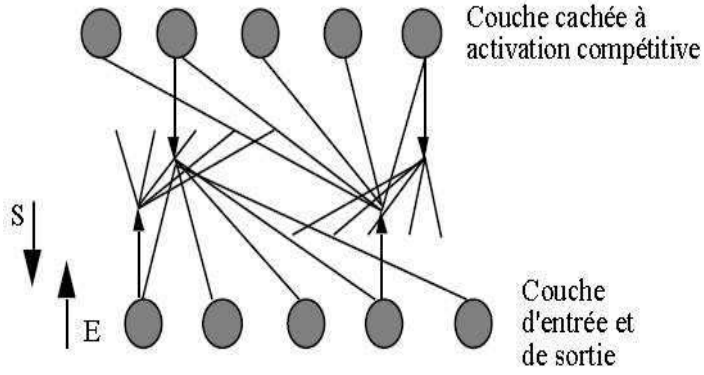


Fig. I.17 : Architecture d'un modèle d'ART [C. Touzet, 1992]

Le réseau ART1 (figure I.17) est un réseau à entrées binaires. Il est formé d'une couche d'entrée qui est aussi la couche de sortie et d'une couche cachée. Le terme de couche cachée est emprunté au réseau multicouche, il souligne le fait que cette couche n'est pas directement observable par l'utilisateur à la différence de l'entrée ou de la sortie. Il n'y a pas de connexion entre les neurones d'entrés. Par contre, la couche cachée est une couche d'activation compétitive, tous les neurones sont reliés les uns aux autres par des connexions inhibitrices de poids fixes. Chaque neurone de la couche d'entrée est relié à tous les neurones de la couche cachée et, réciproquement, chaque neurone de la couche cachée est relié à tous les neurones de la couche de sortie. A chaque connexion est associé un poids. La couche d'entrée est aussi celle de sortie. Tous les neurones de la couche d'entrée sont reliés à tous les neurones de la couche cachée et tous les neurones de la couche cachée à chacun de ceux de la couche de sortie. Il n'y a pas de relation entre les neurones d'entrés alors que la couche cachée est à activation compétitive.

III.6. Apprentissage

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux. Elle ne concerne cependant pas tous les modèles, mais les plus utilisés. L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage fait appel à des exemples de comportement du processus à modéliser [G. Hinton, 1989].

On distingue deux types d'apprentissage : l'apprentissage supervisé et l'apprentissage non supervisé, selon qu'on fournit ou non au modèle les sorties désirées :

I.6.1. Apprentissage supervisé

L'apprentissage supervisé admet l'existence d'un expert qui possède une connaissance innée de l'environnement. Le rôle de l'expert est de fournir les informations relatives à l'environnement nécessaires à l'apprentissage du réseau, sous forme d'un ensemble d'exemples composés de stimuli auxquels sont associées des réponses désirées.

L'apprentissage supervisé exige donc que les exemples ou les données soient constitués de couples de valeurs du type : (valeur d'entrée, valeur de sortie désirée). Tout le problème de l'apprentissage supervisé consiste, pour un vecteur d'apprentissage de n couples (x, d) i = 1, 2, …, n, à déterminer le vecteur des poids des neurones capables de prédire le même vecteur de sortie à partir du même vecteur d'entrée.

Un apprentissage supervisé force donc le réseau à converger vers un état final précis, en même temps qu'on lui présente un motif (données).

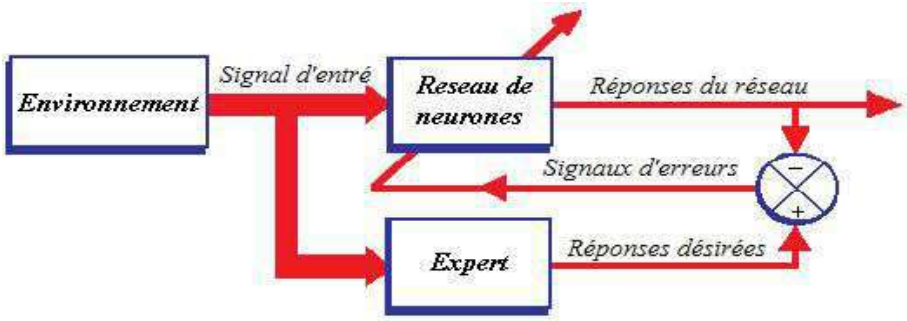


Fig. I.18 : Illustration de l'apprentissage supervisé [D. Hassani, 2003]

I.6.2. Apprentissage non supervisé

Contrairement à l'apprentissage supervisé, lors d'un apprentissage non-supervisé, le réseau est laissé libre de converger vers n'importe quel état final lorsqu'on lui présente un motif. Il n'y a donc pas de professeur qui dirige le réseau vers une situation souhaitée, c'est au réseau de découvrir les ressemblances entre les éléments de la base de données, et de les traduire par des réponses adéquates.

L'apprentissage est qualifié donc de non supervisé lorsque seules les valeurs d'entrée sont disponibles. Dans ce cas, les exemples présentés à l'entrée provoquent une auto adaptation du réseau afin de produire des valeurs de sortie qui soient proches en réponse pour des valeurs d'entrées similaires.

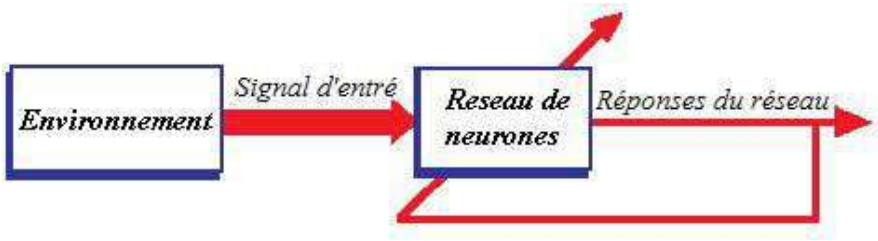


Fig. I.19 : Illustration de l'apprentissage non supervisé [D. Hassani, 2003]

I.7. Règles d'apprentissage

I.7.1. Règle d'apprentissage de Hebb

Les réseaux neuronaux se sont développés grâce à leur capacité d'apprentissage. Le psychologue Hebb fût le premier à présenter un apprentissage à partir des neurones artificiels. "Lorsqu'un axone de la cellule A est suffisamment proche pour pouvoir exciter la cellule B et qu'il prend part de manière répétitive ou persistante à cette excitation, alors on doit trouver soit un phénomène de croissance, soit un changement métabolique dans l'une ou l'autre des cellules tel que l'efficacité de la cellule A pour exciter B doit être accrue." [D.O. Hebb, 1949].

En d'autres termes, la synapse entre la cellule A et la cellule B doit faciliter la transmission de l'influx nerveux lorsque les deux sont excités en même temps, et au contraire si les cellules A et B ne sont pas activés sur la même excitation, la synapse doit inhiber cette transmission.

La règle de Hebb est utilisée dans un apprentissage supervisé. Son algorithme modifie de façon itérative les poids pour adapter la réponse obtenue à la réponse désirée. Il comprend les points suivants [C. Touzet, 1992] :

1. Présentation des matrices d'entrée X et des vecteurs de sortie désirée D de la base d'apprentissage;
2. Initialisation des poids w_i et du seuil θ à des valeurs petites, choisies au hasard, ainsi que μ le pas de modification des poids;
3. Calcul de la sortie S obtenue pour cette entrée :

$$S = F(x, w) = F\left(\sum_j w_j x_j - \theta\right) \qquad \text{(I.8)}$$

$$F(x,w) = f(x) = \begin{cases} +1, & si \left(\sum_j w_j x_j - \theta\right) > 0 \\ -1, & sinon \end{cases} \quad \text{(I.9)}$$

4. Si la sortie *S* est différente de la sortie désirée *D* alors modification des poids :

$$w_j^{t+1} = w_j^t + \mu\left(x_j d_j\right) \quad \text{(I.10)}$$

5. Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement, retour à l'étape 3.

Le perceptron est donc capable de réaliser une partition de son espace d'entrée en deux classes (1 et 2), selon la valeur de sa sortie (+1 ou -1).

La séparation entre ces deux classes étant donnée par un hyperplan (figure I.20). L'équation de la droite séparatrice appelée encore droite de décision est [C. Touzet, 1992] :

$$\sum_j \left(w_j x_j\right) - \theta = 0 \quad \text{(I.11)}$$

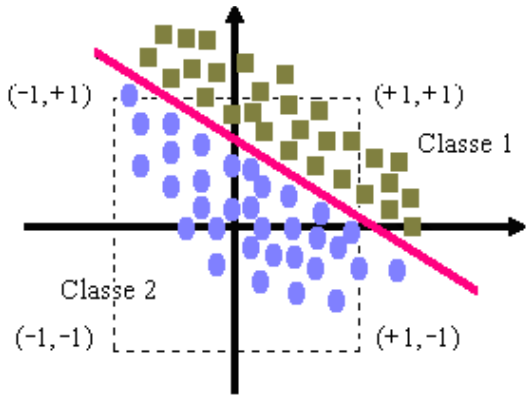


Fig. I.20 : Séparation des exemples d'apprentissage par la droite de décision [C. Touzet, 1992]

I.7.2. Règle d'apprentissage du perceptron de F. Rosenblatt

Elle consiste à présenter au réseau une série d'exemples d'apprentissage, puis à minimiser l'erreur entre la sortie désirée et la sortie effectivement obtenue. L'algorithme est le suivant [C. Touzet, 1992] :

1. Présentation des matrices des vecteurs d'entrée X et de la sortie désirée D de la base d'apprentissage;

2. Initialisation des poids w_i et du seuil θ à des valeurs petites choisies au hasard, ainsi que μ le pas de modification des poids et ε l'erreur admissible;
3. Calcul de la sortie S obtenue pour cette entrée :

$$S = F(x, w) = F\left(\sum_j w_j x_j - \theta\right) \tag{I.12}$$

$$F(x, w) = f(x) = \begin{cases} +1, & si \left(\sum_j w_j x_j - \theta\right) > 0 \\ -1, & sinon \end{cases} \tag{I.13}$$

4. Calcul de l'erreur :

$$e = d - s \tag{I.14}$$

5. Si $e < \varepsilon$ le calcul s'arrête;
6. Si non, on ajuste les poids synaptiques :

$$w_j^{t+1} = w_j^t + \mu\left[\left(d_j - s_j\right)x_j\right] \tag{I.15}$$

7. Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement, retour à l'étape 3.

I.7.3. Règle d'apprentissage de Windrow-Hoff (ADALINE)

Le modèle de L'adaline (ADAptive LINear Element) a été présenté par B. Widrow, [B. Widrow et M.E. Hoff, 1960] chercheur américain à Stanford, dans les années 1960. Dans la règle de Widrow-hoff (appelé aussi règle delta), l'apprentissage est réalisé par itération. Les poids à la phase initiale sont générés aléatoirement. L'idée est de remplacer la fonction seuil d'activation dans les cellules de sortie par la fonction identité (figure I.20). L'avantage de cette approche est de ne plus être restreinte à une réponse binaire, et d'offrir tout l'éventail de valeurs possibles. Cependant, l'apprentissage va être défini comme la minimisation de l'erreur globale du système. La fonction d'erreur ou fonction de coût repose sur la méthode des moindres carrés [J.A. Freeman et D.M. Skapura, 1992] :

$$E = \frac{1}{2}\sum_{j}\left(d_j - s_j\right)^2 \tag{I.16}$$

L'appellation delta vient du fait qu'on exprime la modification d'une connexion par :

$$w_j^{t+1} = w_j^t + \Delta w_j \tag{I.17}$$

Avec :

$$\Delta w_j = -\mu \frac{\partial E}{\partial w_j} \tag{I.18}$$

$$\frac{\partial E}{\partial w_j} = \frac{\partial}{\partial w_j}\left[\frac{1}{2}\sum_{j}\left(d_j - s_j\right)^2\right] \tag{I.19}$$

$$\frac{\partial E}{\partial w_j} = -\left(d_j - s_j\right)\frac{\partial s_j}{\partial w_j} \tag{I.20}$$

$$\frac{\partial E}{\partial w_j} = -\left(d_j - s_j\right)x_j \tag{I.21}$$

D'où

$$w_j = w_j + \mu\left(d_j - s_j\right)x_j \tag{I.22}$$

Son algorithme d'apprentissage peut être décrit par les points suivants [J.A. Freeman et D.M. Skapura, 1992] :

1. Présentation des matrices des vecteurs d'entrée X et de la sortie désirée D de la base d'apprentissage;
2. Initialisation des paramètres initiaux;
3. Calcul de la sortie S

$$S = F(P) = P \tag{I.23}$$

Avec

$$P = \sum_{j} w_j x_j - \theta \tag{I.24}$$

4. Phase de test : calcul de l'erreur quadratique E

$$E = \frac{1}{2}\sum_{j}\left(d_j - s_j\right)^2 \tag{I.25}$$

5. Si E < ε (erreur admissible), le calcul s'arrête
6. Si non, on ajuste les poids synaptiques :
7.

$$w_j^{t+1} = w_j^t + \mu\left[\left(d_j - s_j\right)x_j\right] \tag{I.26}$$

8. Retour à l'étape 3.

I.7.4. Algorithme de rétropropagation du gradient

L'algorithme de rétropropagation du gradient est à la base des premiers succès des réseaux de neurones multicouches. Il figure parmi les algorithmes d'apprentissage les plus utilisés. Il s'agit de minimiser l'erreur quadratique, ce qui est assez simple quand on utilise une fonction d'activation dérivable (la sigmoïde par exemple). On propage la modification des poids de la couche de sortie jusqu'à la couche d'entrée [F. Elie, 1997]. L'apprentissage par rétropropagation du gradient d'un perceptron multicouche peut donc se décomposer principalement en quatre phases [F. Elie, 1997] :

- la propagation du signal de la couche d'entrée jusqu'à la couche de sortie,
- le calcul de l'erreur en sortie,
- la rétropropagation de l'erreur;
- correction des poids synaptiques.

Avant tout apprentissage, il est indispensable de normaliser et de centrer toutes les variables d'entrées. Il faut aussi, qu'au lancement de l'apprentissage, les valeurs initiales des poids doivent être différentes de zéro pour que l'algorithme de rétropropagation puisse fonctionner.

D'autre part, l'utilisation de valeurs élevées peut provoquer un phénomène de saturation prématurée qui contribue à diminuer la vitesse de convergence de l'apprentissage [Y. Lee et al.,

1991], [G. Burel, 1991]. La fonction de coût la plus utilisée dont on cherche à évaluer le gradient est de la forme [H. Abdi, 1994] :

$$E = \sum_j (d_j - s_j)^2 \qquad \text{(I.27)}$$

Elle exprime l'erreur ou la distance quadratique entre la sortie réelle du modèle s_j et la sortie désirée d_j. Cette fonction a tendance à amplifier les erreurs les plus importantes. Par conséquent, au cours de l'apprentissage, la mise à jour des poids est largement déterminée par la correction des grandes erreurs, ce qui est recherché en général [B. Hassibi et D.G. Stork, 1993].

L'apprentissage comporte une première phase de calcul dans le sens direct (figure I.21), où chaque neurone effectue la somme pondérée de ses entrées et applique en suite la fonction d'activation F (fonction dérivable), pour obtenir la mise à jour du neurone. L'équation suivante correspond à cette mise à jour :

$$s_j = F\left(P_j = \sum_{j=0}^{n} w_{ij} x_j\right) \qquad \text{(I.28)}$$

Avec :

- P_i : le potentiel post-synaptique du neurone i;
- x_j : l'état du neurone de la couche précédente;
- w_{ij} : le poids de la connexion du neurone *j* vers le neurone *i.*

Cette phase, dite de propagation, permet de calculer la sortie du réseau en fonction de l'entrée.

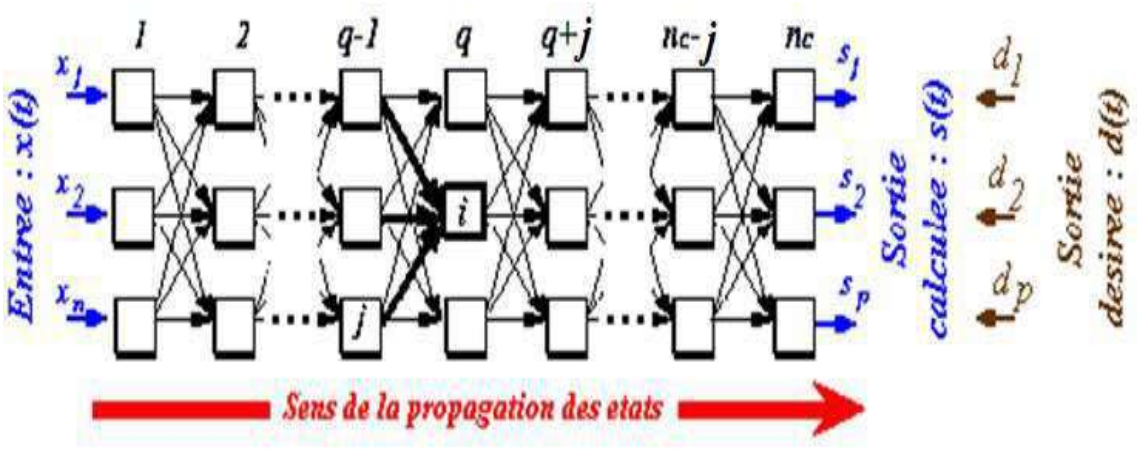


Fig. I.21 : Sens de propagation du calcul des sorties s_j [F. Elie, 1997]

Pour pouvoir minimiser la fonction de coût, on doit calculer son gradient pour tous les poids de la manière suivante [F. Elie, 1997] :

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial P_j}\frac{\partial P_j}{\partial w_{ij}} = \frac{\partial E}{\partial P_j}x_j \qquad \text{(I.29)}$$

Le gradient δ_j sera ensuite :

$$\delta_j = -\frac{\partial E}{\partial P_j} \qquad \text{(I. 30)}$$

On distingue alors deux cas, suivant que le neurone d'indice *j* est un neurone de sortie ou non. Dans le cas de la couche de sortie, le gradient attaché aux cellules de sortie est alors obtenu par l'équation suivante [F. Elie, 1997] :

$$\delta_j = -\frac{\partial E}{\partial P_j} = -\frac{\partial E}{\partial s_j}\frac{\partial s_j}{\partial P_j} = -\frac{\partial E}{\partial s_j}\frac{\partial F(P_j)}{\partial P_j} = 2(d_j - s_j)F'^{(P_j)} \qquad \text{(I. 31)}$$

Car seul s_j dépend de P_j et $s_j = F(P_j)$.

Pour les neurones des couches cachées, l'ordre de calcul des gradients est l'inverse de celui utilisé pour la mise à jour des états dans le réseau. Il s'effectue d'une manière récursive en menant les calculs depuis la (ou les) sortie(s) du réseau vers ses entrées (figure I.22); on parle alors de rétropropagation. L'expression du gradient prend la forme suivante [F. Elie, 1997] :

$$\delta_j = -\frac{\partial E}{\partial P_j} = -\sum_l \frac{\partial E}{\partial P_l}\frac{\partial P_l}{\partial P_j} = \sum_l \delta_l \frac{\partial P_l}{\partial P_j} = \sum_l \delta_l \frac{\partial P_l}{\partial s_j}\frac{\partial s_j}{\partial P_j} \qquad \text{(I.32)}$$

Soit encore :

$$\delta_j = -F'(P_j)\sum_l \delta_l w_{lj} \qquad \text{(I.33)}$$

Où δ_l : le gradient du neurone l de la couche suivante (en aval) de la cellule j.

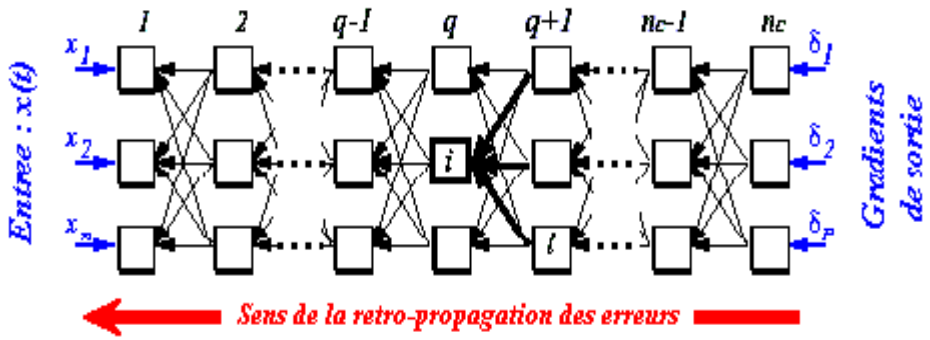


Fig. I.22 : Sens de rétropropagation des gradients δ_j [F. Elie, 1997]

Dans le cas de l'algorithme de gradient total, ils s'agit d'un apprentissage non adaptatif c'est-à-dire tous les exemples de la base d'apprentissage sont présentés successivement au réseau, la modification des poids n'intervient qu'après présentation de tous les exemples. Par opposition au gradient stochastique, qui s'effectue au moyen d'un apprentissage adaptatif, la modification des poids est effectuée au fur et à mesure que les données ou les exemples sont disponibles [A. Tsutsumi et W. Chen, 2001].

Enfin, selon l'équation générale de modification des poids w_{ij} [F. Elie, 1997] :

$$w_{ij}^{t+1} = w_{ij}^{t} + \Delta w_{ij}^{t} \tag{I.34}$$

$$\Delta w_{ij} = -\mu \frac{\partial E}{\partial w_{ij}} = \mu \delta_i w_j \tag{I.35}$$

D'où :

$$w_{ij}^{t+1} = w_{ij}^{t} + \mu \delta_i x_j \tag{I.36}$$

μ : est un petit nombre positif qui représente le pas de déplacement en direction du minimum le plus proche (pas d'apprentissage ou pas du gradient).

La méthode d'évaluation des paramètres w du modèle (l'équation I.36), dite méthode du gradient simple, est commode, mais elle présente de nombreux inconvénients:

- si le pas du gradient est trop petit, la décroissance du coût est très lente ; si le pas est trop grand, le coût peut augmenter ou osciller ;
- au voisinage d'un minimum de la fonction du coût, le gradient de cette dernière tend vers zéro, donc l'évolution du vecteur des coefficients devient très lente ; il en va de même si

la fonction de coût présente des plateaux où sa pente est très faible ; ces plateaux peuvent être près d'un minimum, et, dans la pratique, il est impossible de savoir si une évolution très lente du gradient est due au fait que l'on est au voisinage d'un minimum, ou que l'on se trouve sur un plateau de la fonction du coût ;

- si la courbure de la fonction du coût varie beaucoup, la direction du gradient peut être très différente de la direction qui mènerait vers un minimum.

Un autre problème surgi fréquemment lors de l'apprentissage des réseaux de neurones est que certains neurones sont écartés lors de la phase d'apprentissage vu que leur vecteur de poids ne subi pas de modification au cour de l'apprentissage ce qui rend la contribution du neurone au quel sont associés dans l'ensemble du réseau, insignifiante. Ces neurones sont souvent désignés de neurones morts ou inactifs.

Pour résoudre ce problème un nombre appelé biais est souvent ajouté à chaque neurone du réseau. Le résultat est que le biais des neurones qui ne répondent pas fréquemment, augmente au cours de l'apprentissage pour permettre à ces derniers de contribuer d'une manière significative dans l'élaboration du modèle. Par contre, le biais des neurones très actifs diminue. Ce terme joue, en effet, le rôle de régulateur. La rétropropagation du gradient permet donc de minimiser la fonction de coût $E(w_{ij})$. Toutefois, cette méthode ne garantit pas l'obtention du minimum global de la fonction (blocage dans un minimum local). Dans ce cas la minimisation de l'erreur n'est pas optimale. Cette méthode consiste en fait à se placer en un point et à se laisser descendre le long des pentes de la courbe $Z = E(w_{ij})$, tel qu'elle est illustrée par la figure ci-après :

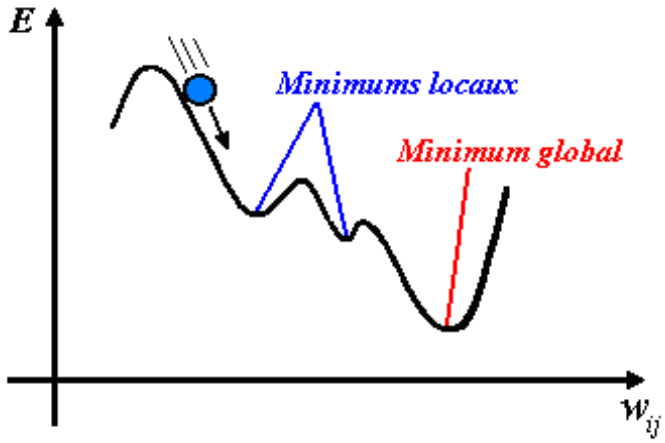


Fig. I.23 : Graphe de la surface d'erreur [A. Tsutsumi et W. Chen, 2001]

En résumé, l'algorithme de rétropropagation du gradient s'articule autour des points suivants :

1. Initialisation des poids à des petites valeurs aléatoires;
2. Présentation d'une entrée X_k et de la sortie désirée d_k;
3. Calcul de la sortie actuelle par propagation à travers les couches en utilisant l'équation (I.28);
4. Accumulation des erreurs en sortie, en utilisant l'équation (I.27);
5. Rétropropagation du gradient de l'erreur depuis la dernière couche vers la première couche :
    - Pour chaque cellule de sortie en utilisant l'équation (I.31);
    - Pour chaque cellule cachée en utilisant l'équation (I.33);
6. Mise à jour des poids en utilisant l'équation (I.36);
7. Répéter les étapes de 2 à 6 pour l'entrée suivante, jusqu'à ce que l'erreur en sortie soit inférieure à la limite fixée ou que le nombre maximum d'itérations soit atteint.

Ils existent d'autres variantes pour la modification des paramètres d'un modèle (gradient du second ordre, BFGS, Levenberg-Marquardt,…) dont nous présentons respectivement les principes ci-après.

I.7.4.1. Méthode de Newton

Pour trouver une solution qui minimise la fonction de coût, la méthode de rétropropagation du gradient utilise uniquement l'information donnée par la dérivée de premier ordre. Il est connu en optimisation que l'utilisation des informations données par les dérivées d'ordre supérieur peut accélérer le processus de recherche d'une solution optimale. De nombreux chercheurs ont tenté d'étendre cette approche aux techniques adaptatives dans le cadre des réseaux de neurones [R.L. Watrous, 1987].

Les méthodes du second ordre opèrent une recherche itérative d'un point minimum w^* à partir d'approximations locales quadratiques de la fonction du coût E. Supposons l'adaptation ($\Delta w^t = w^{t+1} - w^t$) des paramètres du réseau relativement faible à chaque itération. Il est alors possible d'estimer la valeur $E\left(w^{t+1}\right)$ de la fonction du coût E à l'instant *(t+1)*, par un développement en série de *Taylor* multi-variable, limité à l'ordre 2 au voisinage du point w_t :

$$E\left(w^{t+1}\right) = E\left(w^t + \Delta w^t\right) \cong E\left(w^t\right) + \nabla_{(w)} E\left(w^t\right)\Delta w^t + \tfrac{1}{2}\Delta w^t H\left(w^t\right)\Delta w^t \qquad \text{(I.37)}$$

Où $H(w^t)=\nabla^2_{(w)}E(w^t)$ représente la matrice Hessienne des dérivées partielles du second ordre de la fonction de coût E, évaluée au point w_t. Les éléments de cette matrice sont définis par :

$$H(w)_{ij}=\frac{\partial^2 E(w)}{\partial w_i \partial w_j} \tag{I.38}$$

$$H(w)=\begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_1 \partial w_n} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_n \partial w_1} & \frac{\partial^2 E}{\partial w_n \partial w_2} & \cdots & \frac{\partial^2 E}{\partial w_n^2} \end{bmatrix} \tag{I.39}$$

Un point minimum *w** de la fonction de coût est approximativement localisé, à partir de l'expression approchée (I.37), par annulation du vecteur gradient $\nabla_{(\Delta w)}E(w^{t+1})$ selon :

$$\nabla_{(\Delta w)}E(w_{t+1})=\nabla_{(w)}E(w_t)+H(w^t)\Delta w_t=0 \tag{I.40}$$

Soit :

$$\Delta w^t=-H(w^t)^{-1}\nabla_{(w)}E(w^t) \tag{I.41}$$

La règle d'adaptation des paramètres du réseau du second ordre se déduit alors comme suivant :

$$w^{t+1}=w^t-\mu^t A \tag{I.42}$$

$A=-H(w^t)^{-1}\nabla_{(w)}E(w^t)$; représente la direction de minimisation du second ordre empruntée à l'instant t, et μ^t constitue le pas de la descente (coefficient d'adaptation positif $0<\mu^t<1$).

La méthode d'optimisation paramétrique décrite par la règle d'adaptation, est connue sous la dénomination de méthode d'optimisation de Newton. Elle peut localiser le minimum d'une fonction de coût quadratique en une unique itération (en ce cas le pas de la descente est constant fixé à un $\mu^t=1$).

En pratique, cette méthode est peu employée car elle nécessite le calcul du Hessien et surtout de son inverse à chaque itération qui est complexe. On lui préfère une méthode plus économique dite de quasi-Newton.

I.7.4.2. Méthode de quasi-Newton

La méthode de quasi-Newton [R.L. Watrous, 1987] consiste à approcher l'inverse du Hessien plutôt que de calculer sa valeur exacte. L'inverse du Hessien est approché par une matrice M^k définie positive modifiée à chaque itération. La suite des matrices $\{M^k\}$ est construite de manière à converger vers l'inverse du Hessien lorsque la fonction de coût est quadratique. La modification des paramètres est donnée par :

$$w^{t+1} = w^t - \mu^t M^t \nabla E^t \qquad \text{(I.43)}$$

A la première itération, la matrice M^0 est prise égale à la matrice identité. Cette suite est construite grâce à la méthode dite BFGS, développée indépendamment par *C.G.Broyden*, *R.Fletcher*, *D.Goldfarb* et *D.F.Shanno* [A. Duprat et al., 1998], dont la vitesse de convergence est beaucoup plus grande que celle de la méthode du gradient.

I.7.4.3. Méthode Levenberg-Marquardt

La méthode de *Levenberg-Marquardt* [K. Levenberg, 1944] consiste à modifier les paramètres selon la relation suivante :

$$w^{t+1} = w^t - \left(H^t + \eta^t I\right)^{-1} \nabla E^t \qquad \text{(I.44)}$$

Avec I : matrice identité.

Cette méthode est particulièrement astucieuse car elle s'adapte d'elle-même à la forme de la fonction de coût. Elle effectue un compromis entre la direction du gradient et la direction donnée par la méthode de Newton. En effet, si η^t est grand, on reconnaît la méthode du gradient simple (dans ce cas la valeur du pas est donnée par $1/\eta^t$) et si η^t est petit, la modification des paramètres correspond à celle de la méthode de Newton [A. Duprat et al., 1998].

I.8. Théorie de la généralisation

La généralisation concerne la tâche accomplie par le réseau une fois son apprentissage achevé [D.W. Marquardt, 1963]. Elle peut être évaluée en testant le réseau sur des données qui n'ont pas servi à l'apprentissage. Elle est influencée essentiellement par quatre facteurs : la complexité du problème, l'algorithme d'apprentissage, la complexité de l'échantillon (le nombre d'exemples et la manière dont ils représentent le problème) et enfin la complexité du réseau (nombre de poids). La complexité du problème est déterminée en partie par sa nature même : on peut parler de "complexité intrinsèque". Par ailleurs, l'algorithme d'apprentissage influe sur la généralisation par son aptitude à trouver un minimum local assez profond, sinon le minimum global [A. Duprat et al., 1998].

La complexité de l'échantillon est définie par la figure I.24. Supposons que l'on cherche à extraire une règle à partir d'exemples, la zone *U* représente l'ensemble des couples entrée-sortie possibles. La partie *R* représente la région dans laquelle la règle qu'on cherche à extraire est valable. On dit que *R* est consistante avec la règle.

On sélectionne un certain nombre d'exemples pour l'apprentissage du réseau, qui forment l'ensemble d'apprentissage Z_A. On sélectionne un autre ensemble distinct Z_T pour tester le réseau et estimer sa généralisation. Les deux ensembles Z_A et Z_T doivent être représentatifs de la région R.

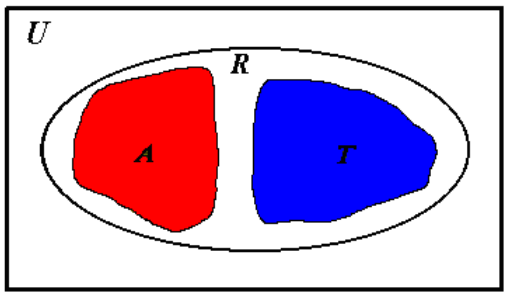


Fig. I.24 : Extraction de règles [A. Duprat et al., 1998]

L'autre facteur influant sur la généralisation est la complexité du réseau. Pour bien cerner cet aspect, on fait l'analogie avec un problème de régression polynomiale classique. Supposons que l'on dispose d'un nuage de points issus d'une fonction *g* d'une variable réelle inconnue. Les exemples à notre disposition sont des couples (x_i, y_i) bruités de la forme [A. Bouzidi, 2006] :

$$y_i = g(x_i) + \varepsilon_i \tag{I.45}$$

Où les ε_i sont des réalisations d'une variable aléatoire. L'objectif est de modéliser la fonction g par un modèle polynomial en utilisant les exemples d'apprentissage. Les courbes de la figure III.25 représentent la modélisation de g par trois modèles qui différent par leur nombre de paramètres. On peut constater que le modèle ayant très peu de paramètres n'a pas assez de flexibilité pour réaliser un apprentissage correct des exemples d'apprentissage. Les erreurs d'apprentissage et de test sont toutes deux importantes : c'est la situation de sous-apprentissage (figure I.25c). En revanche, le modèle constitué de nombreux paramètres, lisse parfaitement les exemples d'apprentissage. Il commet donc une erreur faible sur ces données, mais probablement une erreur plus importante sur les données de test. C'est la situation de sur-apprentissage (figure I.25b). Finalement, le modèle possédant un nombre de paramètres modérés réalise un bon compromis entre précision d'apprentissage et bonne généralisation (figure I.25a).

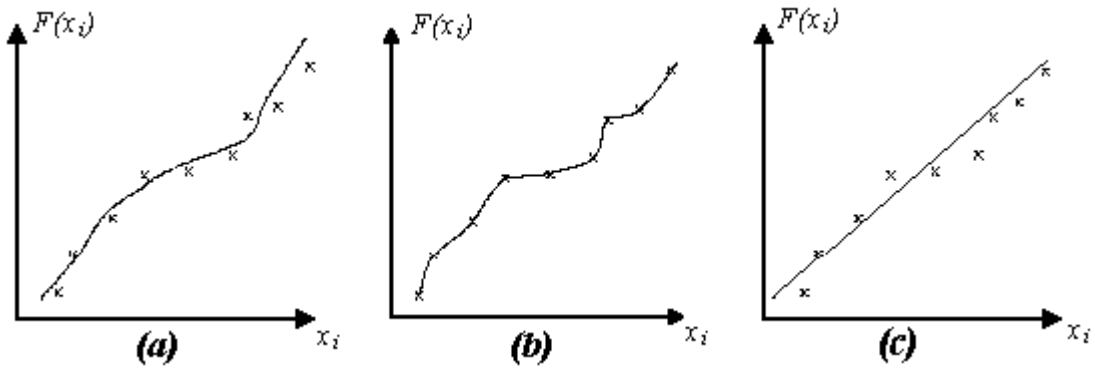


Fig. I.25: a) Apprentissage correct, b) Sur-apprentissage, c) Sous-apprentissage [A. Bouzidi, 2006]

Le problème de la généralisation est souvent vu sous deux perspectives différentes. Dans la première, la taille du réseau est fixée (en accord avec la complexité du problème) et la question est : combien d'exemples d'apprentissage sont nécessaires pour atteindre une bonne généralisation ? Cette perspective est intéressante dans les applications où l'on a la possibilité d'acquérir autant d'exemples que l'on veut. Dans le second cas, le nombre d'exemples d'apprentissage est fixé et la question est : quelle taille du réseau donne la meilleure généralisation pour ces données ? On est conduit à adopter ce point de vue lorsque l'on est limité dans la possibilité d'acquérir des données d'apprentissage : il importe alors de déterminer quelle est la taille permettant au réseau de décrire au mieux les données en notre possession.

Cependant une manière d'évaluer la faculté de généraliser la représentation construite par le réseau de neurones à toutes les données, y compris celles n'appartenant pas à l'ensemble

d'apprentissage, consiste à mesurer les performances du réseau sur des données représentatives du problème non apprises. Il s'agit d'une évaluation de l'erreur de généralisation.

La différence entre l'erreur d'apprentissage et l'erreur de généralisation (figure I.26) représente une mesure de la qualité de l'apprentissage effectué [S. Thiria et al., 1997].

I.9. Critère d'arrêt pour l'apprentissage

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

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

Les méthodes qui évaluent l'erreur de généralisation sont presque toutes basées sur la partition de l'ensemble des données qu'on possède en plusieurs sous-ensembles. Par exemple un ensemble utilisé pour l'apprentissage et un ensemble de validation. L'ensemble de validation est utilisé pour contrôler et mesurer la généralisation du réseau. Pendant l'apprentissage l'erreur d'apprentissage décroît continuellement, tandis que sur l'ensemble de validation elle ne diminue que jusqu'à un certain point au-delà duquel elle augmente [D.W. Marquardt, 1963]. A partir de ce point le réseau apprend par cœur les données de l'ensemble d'apprentissage et l'apprentissage doit être stoppé (figure I.26). Ces deux ensembles servent à déterminer l'architecture la plus appropriée pour le problème à traiter : pour différentes architectures (nombre de neurones cachés variable), on contrôle l'erreur de validation et on choisit l'architecture pour laquelle elle est minimale.

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

- La technique nommée ``split-samples'' réserve un troisième ensemble de données appelé ensemble de test, pour tester le réseau sur des données qui n'ont jamais été utilisées ni pour l'apprentissage ni pour la validation.
- Pour la cross-validation l'ensemble des données de départ est découpé en *k* parties de taille égale. Le réseau est entraîné *k* fois, chaque fois en utilisant *(k-1)* parties

pour l'apprentissage et la dernière pour la validation et le calcul des erreurs commises en généralisation. L'ouvrage de *R.Tibshirani* [S. Boucheron, 1992] donne une comparaison de ces méthodes.

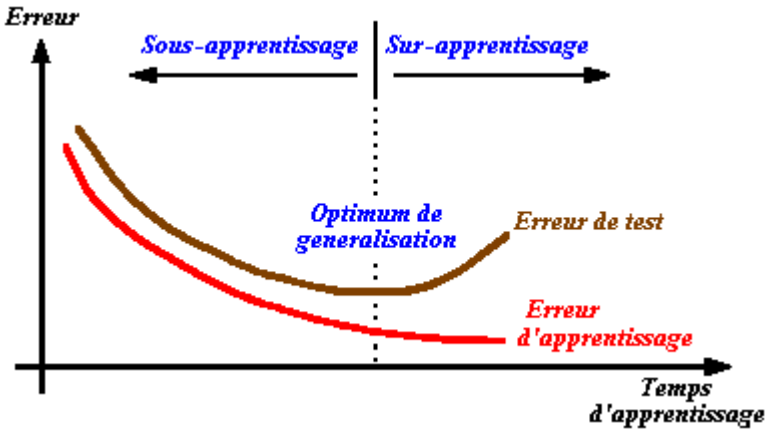


Fig. I.26: Evolution des erreurs d'apprentissage et de test au cours du temps [S. Thiria et al., 1997]

I.10. Procédure de modélisation neuronale

Le modèle boite noire constitue la forme la plus primitive de modèle mathématique : il est réalisé uniquement à partir de données expérimentales ou d'observations ; il peut avoir une valeur prédictive, dans un certain domaine de validité, mais il n'a aucune valeur explicative [G. Dreyfus et al., 2008]. A l'opposé, un modèle de connaissance, que l'on pourrait qualifier de boite blanche, est issu de l'analyse des phénomènes physiques, chimiques, biologiques, etc., dont résulte la grandeur que l'on chercher à modéliser. Ces phénomènes sont mis en équation à l'aide des connaissances théoriques disponibles au moment de l'élaboration du modèle. La démarche scientifique s'efforce, dans la mesure du possible, de construire des modèles de connaissance : l'élaboration d'un modèle de connaissance nécessite l'existence d'une théorie, alors qu'un modèle boite noire requiert seulement la disponibilité de mesures.

Entre la boite noire et le modèle de connaissance se situe le modèle semi- empirique, ou modèle boite grise, qui contient à la fois des équations résultant d'une théorie, et des équations purement empiriques, résultant d'une modélisation de type boite noire [G. Dreyfus et al., 2008].

Les réseaux de neurones constituent le plus souvent des modèles boite noire, néanmoins dans certaines conditions, ils peuvent être utilisés avec profit dans le cadre de modèles boite grises. Un modèle statique est régi exclusivement par des équations algébriques ; un modèle dynamique, en revanche, obéit à des équations différentielles et à des équations algébriques [G. Dreyfus et al., 2008].

On examinera en bref, exclusivement, dans ce qui suit la conception d'un modèle statique type boite noire, réalisé à l'aide de réseaux de neurones bouclés (MLP) du fait qu'il a un lien direct avec le thème de ce travail.

L'objectif de la modélisation boite noire est de trouver, à partir des données ou des mesures disponibles, une relation déterministe, si elle existe, entre les variables d'entrées du modèle *x* et la grandeur à modéliser *y*.

La procédure de développement d'un réseau de neurone qui répond à une modélisation statique peut se résumer aux points suivants:

1. Sélection des données
2. Collecte des données
3. Séparation des bases de données,
4. Normalisation des données,
5. Choix du type de réseau de neurones,
6. Choix du nombre de neurones dans la couche cachée
7. Choix de la fonction de transfert
8. Estimation des paramètres ou apprentissage,
9. Validation et généralisation.

I.10.1. Sélection des données

Lorsqu'on conçoit un modèle à partir des mesures, il est indispensable que l'ensemble des variables d'entrée soit aussi réduit que possible ; en effet, à chaque entrée du modèle correspond un nombre de paramètres égal au nombre de neurones cachés. Il est donc nécessaire.

- que la représentation des entrées soit aussi compacte que possible,
- que la représentation des entrées contienne tous les facteurs pertinents par rapport à la grandeur à modéliser, mais seulement ceux-ci. En effet, la présence de variables qui ont, sur la grandeur à modéliser, moins d'influence que les perturbations ou le bruit, entraîne celle de paramètres superflus et introduit des variations non significatives sur les variables d'entrées, qui constituent autant de sources d'erreur pour le modèle [G. Dreyfus et al., 2008].

Avant la conception d'un modèle neuronale, il convient donc de procéder à une sélection des entrées. Cette sélection revêt deux aspects :

- réduction de la dimension de l'espace de représentation des variables du modèle,
- élimination des entrées qui n'ont qu'une faible influence sur la sortie.

I.10.2. Collecte des données

Une fois les entrées pertinentes du modèle sont biens connues, l'objectif de cette étape est de les recueillir, à la fois pour développer (l'apprentissage) le réseau de neurones et pour le tester. La base de données reconstituée doit être représentative des données susceptibles d'intervenir en phase d'utilisation du système neuronal.

I.10.3. Séparation de la base de données

Afin de développer un modèle neuronal, il est nécessaire de disposer d'au moins deux bases de données : une base pour effectuer l'apprentissage et une autre pour valider le réseau obtenu et déterminer ses performances. On peut utiliser une troisième base de données, si elle est disponible, pour tester le réseau. Il n'y a pas de règle précise pour déterminer le partage de manière quantitative des données.

Il résulte souvent d'un compromis tenant compte du nombre de données dont on dispose, l'essentiel, est que la base de données consacrées à l'apprentissage doit être la plus importante. Il est important de souligner que les sous-ensembles formés à partir de la base de données de départ devront être représentatifs de celle-ci.

I.10.4. Normalisation des données

Avant tout apprentissage, il est indispensable de normaliser et de centrer toutes les variables d'entrée. En effet, si des entrées ont des grandeurs très différentes, celles qui sont petites n'ont pas d'influence sur l'apprentissage. Il existe plusieurs approches de normalisation des données dont le choix dépend de la nature des données et de la fonction à approximer.

I.10.5. Choix du type de réseau de neurones

Comme il s'agit d'une modélisation statique type boite noire pour approcher une fonction bornée et continue, la meilleure approche neuronale envisageable est celle des MLP à une seule couche cachée.

I.10.6. Choix du nombre de neurones dans la couche cachée

Il n'existe pas de règle permettant de prévoir à l'avance le nombre de neurones dans la couche cachée. On pourrait penser qu'en plaçant un grand nombre de neurones dans la couche cachée, on obtiendrait systématiquement de meilleurs résultats.

Mais il convient de prendre garde d'une part au temps de calcul, d'autre part au risque de sur apprentissage ou de sous apprentissage lorsque leur nombre est insuffisant.

D'autre part, on sait que le nombre des paramètres croit linéairement avec le nombre des variables d'entrées du modèle (parcimonie). Il va falloir donc trouver un compromis entre la parcimonie et le sur ou le sous apprentissage.

I.10.7. Choix de la fonction de transfert

La fonction de transfert neuronale, que l'on appelle aussi fonction d'activation, est la fonction mathématique appliquée à la sortie de chaque neurone. Là encore, il n'existe pas de méthode permettant de prévoir a priori quelle sera la fonction qui donnera les meilleurs résultats. Mais pour les MLP, il faut des fonctions de type sigmoïde pour les neurones de la couche cachée et généralement une fonction linéaire pour le neurone de la couche de sortie.

I.10.8. Estimation des paramètres ou apprentissage

Pour les modélisations statiques au moyen des MLP, il s'agit de faire un apprentissage supervisé. Le critère d'arrêt de l'apprentissage est souvent calculé à partir d'une fonction de coût, caractérisant l'écart entre les valeurs de sortie obtenues et les valeurs désirées. Ce qui permet un arrêt adéquat de l'apprentissage pour obtenir de bonnes performances de généralisation.

L'obtention des paramètres architecturaux optimaux se fera par comparaison des performances obtenues pour différentes architectures de réseaux de neurones [F. Elie, 1997].

I.10.9. Validation et généralisation

La validation consiste à tester le réseau après ou au cours de l'apprentissage avec des données proches des données utilisées lors de la phase d'apprentissage. Les résultats de ce test permettront au modèle d'éviter les situations de sous ou de sur apprentissage. En revanche la généralisation s'effectue avec des données tout à fait nouvelles et différentes de la base de données de départ. Elle permet de savoir si le réseau a développé une capacité de généralisation assez performante, l'organigramme de la figure ci-dessous indique les étapes pour l'optimisation

des paramètres d'un réseau de neurone artificiel [S. Lolas et O.A. Olatunbosun, 2008], [David M. Himmelblau, 2000].

I.10.10. Conclusion :

Dans ce chapitre, on vient de voir les applications possibles aux réseaux de neurones. Actuellement, cette technique se développe dans l'industrie sortant du cadre de recherche pure. Dans certains cas précis, les réseaux de neurones apportent en effet une bien meilleure solution que d'anciennes techniques. Mais ce qui caractérise avant tous ces réseaux, c'est leur comportement de boite noire. Les exemples nécessaires sont récoltés en nombre, puis l'apprentissage fourni au bout d'un certain temps un système avec certaines sorties dont le comportement interne n'est pas connu avec exactitude. L'organisation des liaisons et des poids d'un réseau est alors difficilement explicable concrètement et on ne peut en comprendre toute les subtilités.

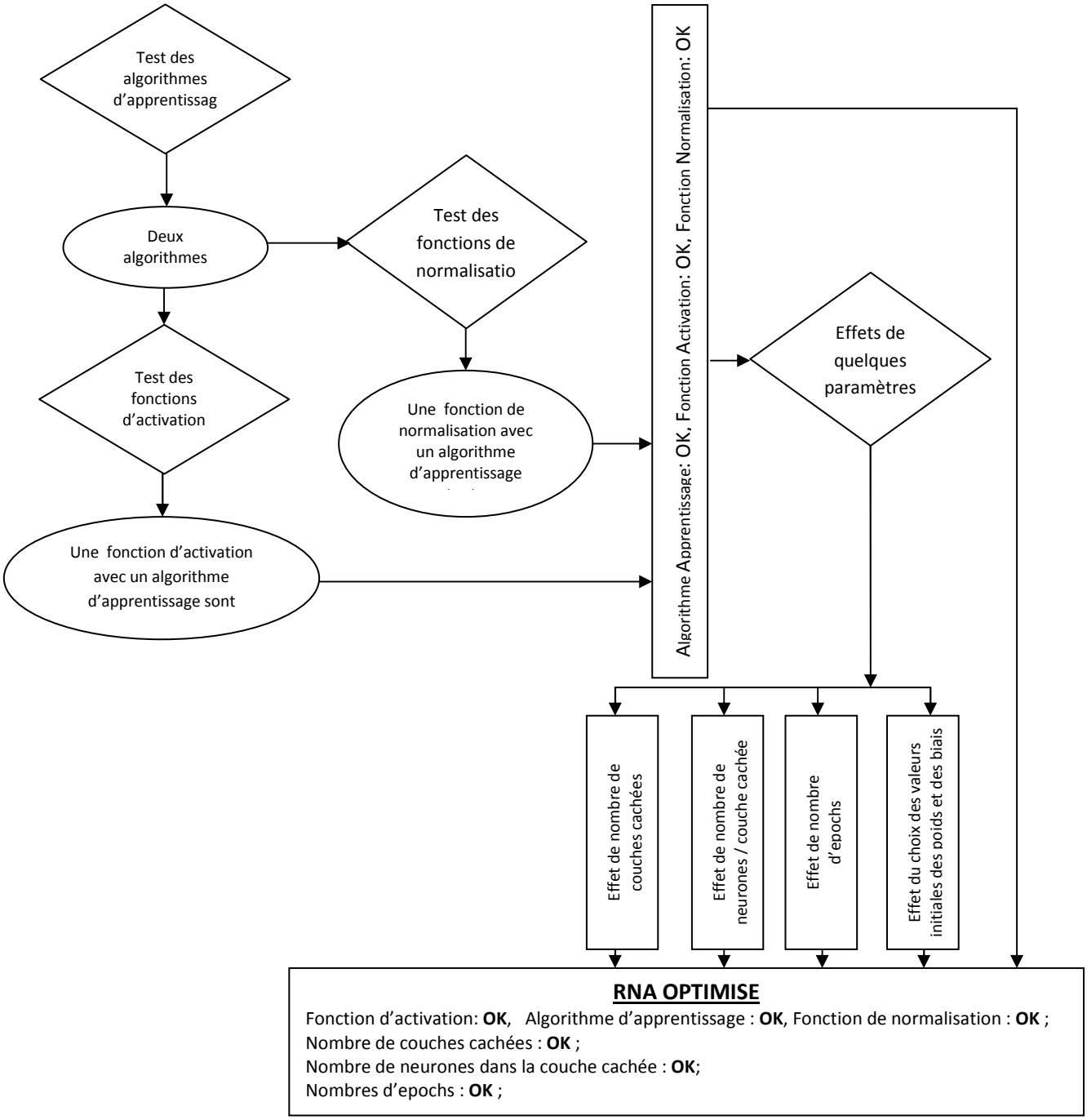


Fig. I.27: Optimisation des paramètres d'un réseau de neurone artificiel [S. Lolas et O.A. Olatunbosun, 2008]

Chapitre II

Intensification des Échanges Thermiques

II.1. Introduction

Le domaine de l'intensification des échangeurs de chaleur a depuis de nombreuses années dépassées le stade de laboratoire et a été largement pris en compte dans les applications industrielles. Nombre d'échangeurs, dans des procédés très divers, sont équipés de surface d'échange (tubes ou plaques) spécialement conçue pour présenter des coefficients de chaleur élevés. La relation bien connue qui relie le flux ou la puissance thermique échangée Φ et l'écart de température logarithmique moyen *DTLM* s'exprime par:

$$\Phi = K \times S \times DTLM \qquad \text{(II.1)}$$

L'industrie utilise des techniques d'intensification qui permettent d'augmenter le terme *KS* de façon significative. Ainsi pour une puissance thermique constante, l'augmentation de *K* permet :

- de réduire la surface d'échange, donc réduction du coût de l'appareil
- de réduire l'écart de température, donc diminution du coût de fonctionnement. Cette intensification implique la modification de la géométrie de la paroi d'échange, qui augmente le coefficient d'échange local *h*.

II.2 Les échangeurs de chaleur

II.2.1 Définition

Un échangeur de chaleur est un système qui permet d'échanger la chaleur entre deux fluides sans se mélanger. Dans un échangeur thermique, le fluide chaud et le fluide froid sont séparés par une paroi (plane ou tubulaire). La transmission de la chaleur se fait du fluide chaud vers le fluide froid. Les fluides, chauffant et chauffé sont des fluides caloporteurs.

Le fluide caloporteur est un fluide à l'état liquide qui est utilisé sans changement de phase pour le transfert d'un lieu à un autre une certaine quantité de chaleur générée par des moyens étranges. Il est caractérisé par sa chaleur massique C_p. étant donné que la puissance d'échange thermique *Q* est proportionnelle à la chaleur massique.

II.2.2 Principe de fonctionnement

Dans un échangeur, la chaleur est transmise d'un fluide à un autre. Le même fluide peut être conservé son état physique (liquide ou gazeux) ou se présenter successivement sous les deux phases (cas des condenseurs et des vaporiseurs).

II.2.3 Types de contact

a) Échangeurs à contact direct :

Le type le plus simple comprend un récipient (ou canalisation) dans lequel les deux fluides sont directement mélangés et atteignent la même température finale, exemples :

- les réchauffeurs d'eau (injection de vapeur d'eau sous pression).
- les dés réchauffeurs industriels et les condenseurs à injection (centrale thermique).

b) Échangeurs à contact indirect :

En principe, pour les échangeurs les plus répondus dans l'industrie, les deux fluides, désignés par 1 et 2, s'écoulent dans des espaces séparés par une paroi ou cloison, à faible inertie thermique. La chaleur que le fluide 1 cède à la paroi, par convection le long de la surface de contact, traverse l'épaisseur de la paroi par conduction et est cédé au fluide 2 par convection le long de l'autre face. Les processus de transfert présentent une réciprocité: convection (1) conduction (paroi), convection (2). Le flux de chaleur échangé s'exprime par le produit d'une conductance globale constante le long de l'échangeur, et de la différence entre les températures moyennes T_1 et T_2 des fluides.

II.2.4 Disposition des écoulements

Ce classement est réalisé à partir de la géométrie de l'échangeur et du sens relatif de l'écoulement des deux fluides. Dans ce qui suit, on représente uniquement les circulations des fluides les plus souvent rencontrées. Dans les échangeurs à fluide séparés, les modes de circulation des fluides peuvent se ranger en deux catégories : Ou bien les vecteurs vitesses des écoulements sont approximativement parallèles : S'ils sont de même sens les écoulements sont « co-courant ». S'ils sont de sens contraire, les écoulements sont «à contre-courant ». Ou bien les vecteurs vitesses sont en moyenne perpendiculaire l'un à l'autre ; il s'agit cette fois de «courant croisés ».

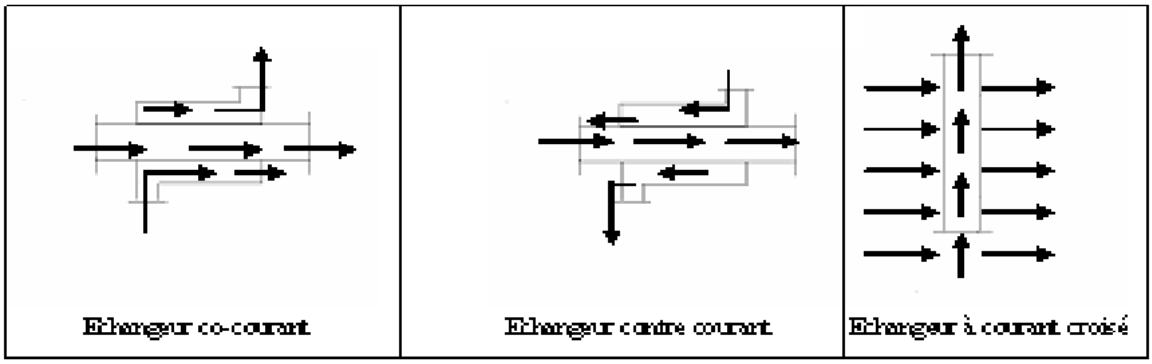


Fig. II.1: différents Modes de circulation [J.Padet, 1994]

II.3 Différents types d'échange

Les échangeurs de chaleur sont diphasiques ou monophasique.

a) Échange sans changement de phase :

Les échangeurs de chaleur sans changement de phase correspondent aux échangeurs dans lesquels l'un des fluides se refroidit pour réchauffer le deuxième fluide sans qu'il y ait changement de phase .les températures des fluides sont donc variables, tout le long de l'échangeur. Parmi ces échangeurs on cite les échangeurs : (liquide –liquide, gaz -liquide et gaz - gaz...).

b) Échange avec changement de phase:

Les échanges avec changement de phase sont caractérisés par trois cas différents:

1. l'un des fluides se condense alors que l'autre se vaporise. Ces échangeurs sont rencontrés dans les machines frigorifiques installées en cascade. Ces sont les évaporateurs - condenseurs.
2. le fluide secondaire se vaporise en recevant de la chaleur du fluide primaire, lequel ne subit pas de changement d'état. Ils sont appelés évaporateurs.
3. le fluide primaire se condense en cédant sa chaleur latente au fluide secondaire plus froid, lequel ne subit pas de transformation d'état .A titre d'exemple; ce cas est rencontré dans les condenseurs des machines frigorifiques.

II.4 Différentes techniques d’intensification

Les différentes techniques d’intensification peuvent être classées en six catégories :

- La modification de la nature de la surface d’échange par revêtement ou dépôts de couches minces continues ou non, par exemple, une amélioration de la condensation de vapeur d’eau, en favorisant la condensation en goutte.
- La modification de l’état de surface des parois d’échange (porosité, rugosité) : la rugosité, uniforme du type grain de sable, ou non uniforme, est prometteur efficace de turbulence qui permet un transfert de chaleur accru à proximité de la paroi d’échange.
- L’extension de surface : avec l’utilisation d’ailettes lisses, ondulées, discontinues…
- La mise en place des dispositifs créant un écoulement rotatif ou secondaire par des différents inserts dans des tubes.
- La mise en place de dispositifs favorisant le mélange des filets fluides.

- La modification de la géométrie de la paroi d'échange avec des ondulations ou des rainures, ces surfaces sont utilisées avec des écoulements en double phase.

II.4.1 Les ailettes

Le coefficient d'échange local entre une paroi et un gaz en circulation forcée étant médiocre, il peut être de 10 à 50 plus faible que celui d'un liquide en circulation forcée .il est donc avantageux dans les échangeurs gaz ou gaz liquide d'équiper la paroi en contact avec la Phase gazeuse d'ailettes lisses ou géométrie performante. l a paroi en contact avec la phase gazeuse d'ailettes lisses ou géométrie performante. Dans le cas des échangeurs à plaques et à ailettes .différentes géométries équipent les appareils actuels :

- ailettes lisses de formes rectangulaires ou triangulaires
- ailettes ondulées
- ailettes perforées
- ailettes discontinues
- ailettes à persiennes

Dans le cas des échangeurs à tubes et ailettes :

- tubes à section circulaire avec ailettes circulaires
- tubes à section circulaire avec ailettes planes lisses
- tubes à section quasi-rectangulaire avec ailettes planes ou lisses ou améliorées

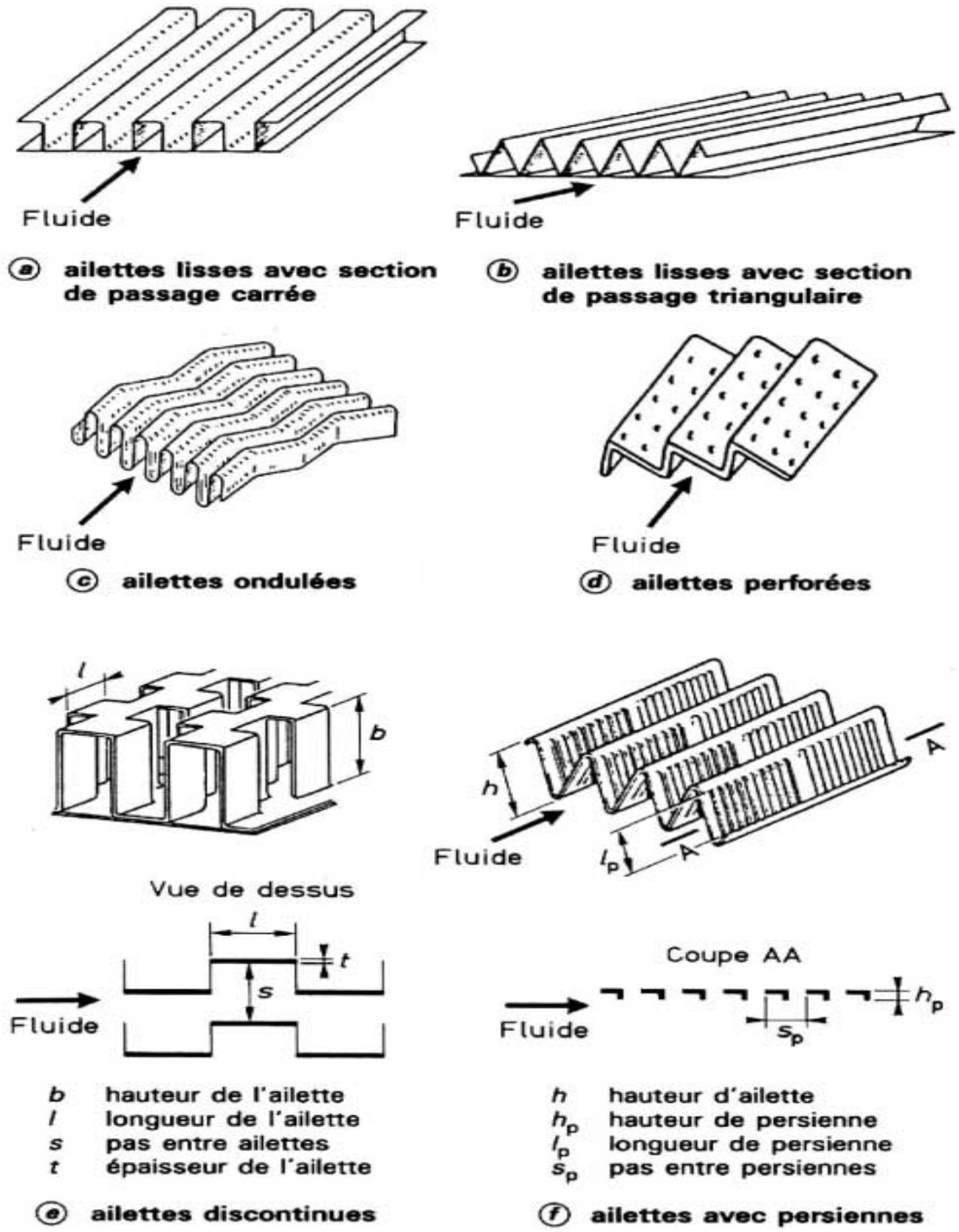


Fig. II.2: plaques à ailettes [H.John Linhard,2005]

II.4.2 Les inserts

Ce type de dispositif est inséré dans un tube lisse, favorise l'amélioration du transfert de chaleur : certains jouent un simple rôle d'ailettes, d'autres favorisent les écoulements rotatifs, ou des mélanges des filets fluides :

- inserts à disques ou à billes
- insert à ressorts
- insert à étoiles
- insert à rubans torsadés

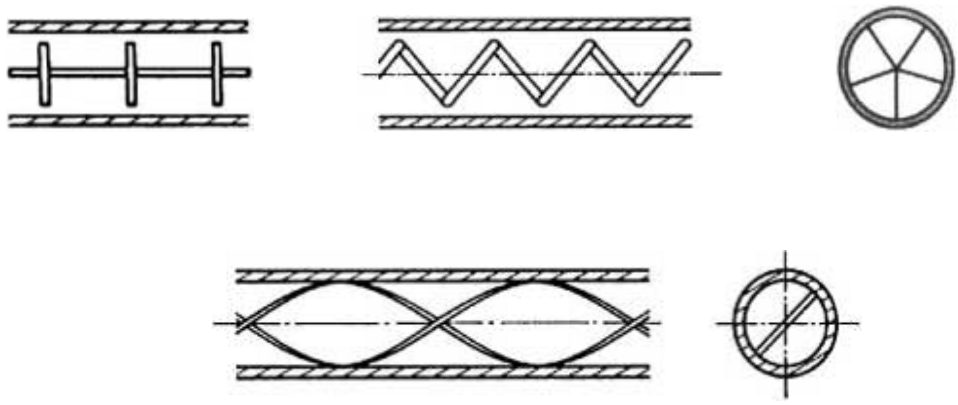


Fig. II 3: Différents types des inserts [H.John Linhard, 2005]

II.4.3 Les Surfaces rugueuses

Il existe de nombreuses applications de surfaces présentant des rugosités dans les échangeurs de chaleur que la surface d'échange soit une plaque, la paroi interne d'un tube, la paroi externe d'un tube. Il existe également une masse considérable de données relatives à ce type d'intensification de l'échange. On peut classer les rugosités en trois familles :

- Les rugosités à trois dimensions dont l'exemple le plus commun est la paroi passée à la toile émeri,
- Les corrugations à deux dimensions caractérisées par des obstacles à la paroi répartis régulièrement,
- Les rainures à deux dimensions réparties régulièrement à la paroi**.**

II-5 Écoulement laminaire et turbulent

Le comportement de l'écoulement monophasé, thermo-hydraulique a des performances et des applications divers par exemple les tubes courbés et spiralé circulaire et non-circulaire de section transversale ont été présentés dans plusieurs revues de la littérature (Nandakumar

Masliyah, 1986; Shah et Joshi, 1987; Bergles et al ., 1991; Ebadian et Dong, 1998). Les caractéristiques de courbure induite les flux tourbillon pour les tubes en spirale ou en hélice courbes sont fortement dépendants de leurs caractéristiques géométries, une représentation typique des tubes de section transversale circulaire est donnée dans la Fig. II.5. Les paramètres sans dimension conséquente (nombre Dean, le rayon de courbure, et le nombre hélicoïdale) qui décrivent le comportement de l'écoulement et la géométrie sont généralement définis comme suit:

$$H_e = R_e\sqrt{\frac{d}{2R_c}} = D_e\left[1+\left(\frac{H}{2\pi R}\right)^2\right]^{1/2} \quad \text{(II.2)}$$

$$R_c = R\left[1+\left(\frac{H}{2\pi R}\right)^2\right] \quad \text{(II.3)}$$

$$D_e = R_e\sqrt{\frac{d}{2R}} \quad \text{(II.4)}$$

On peut noter que le nombre hélicoïdale réduit au nombre Dean lorsque H = 0.

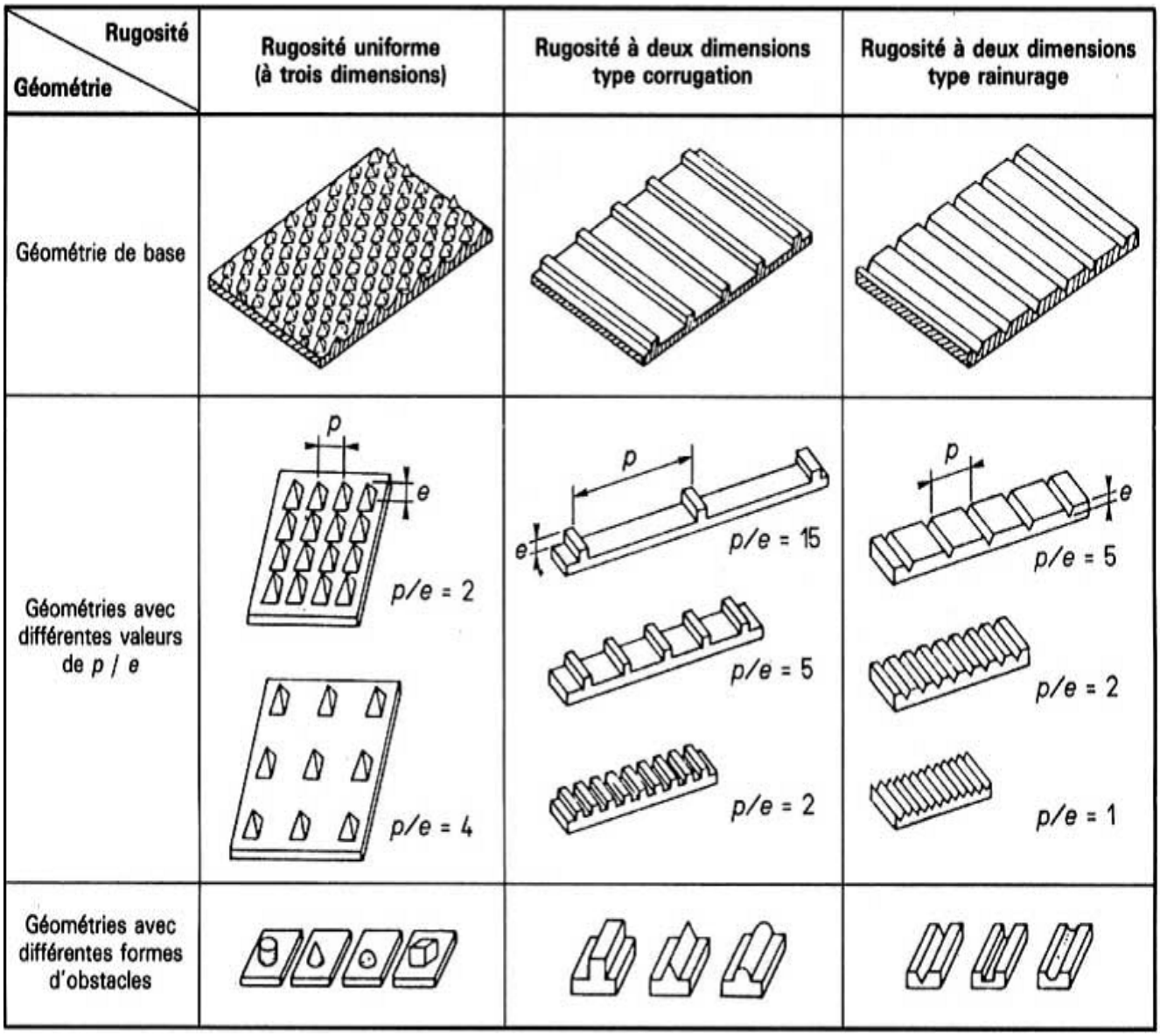


Fig. II.4: Différents types de rugosité [A.Bontenps, et al., 1994]

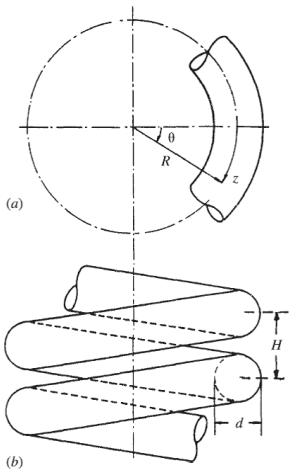


Fig. II.5: Représentation géométrique des tubes [Ebadian et Dong, 1998]

Les corrélations pour un écoulement laminaire et de transfert de chaleur dans les tubes circulaires courbé et spiralé, ont été présentés par B. Manlapaz et Churchill (1980) ont également donné deux expressions distinctes, mais similaires pour supposer le nombre de Nusselt moyen pour les écoulements entièrement développé avec les deux types des conditions aux limites le plus utilisées Dirichlet (UWT) et Newman (UHF) sont respectivement:

$$Nu_{m,F} = \left[\left\{4.364 + \frac{4.636}{[1+(1342/PrHe^2)]^2}\right\}^3 + 1.816\left\{\frac{He}{[1+(1.15/Pr)]}\right\}^{3/2}\right]^{1.3} \quad \text{(II.5)}$$

$$Nu_{m,T} = \left[\left\{3.657 + \frac{4.343}{[1+(957/PrHe^2)]^2}\right\}^3 + 1.158\left\{\frac{He}{[1+(0.477/Pr)]}\right\}^{3/2}\right]^{1.3} \quad \text{(II.6)}$$

Parmi les techniques pour la génération des écoulements tourbillons l'insertion des bandes enroulées. Dans ces cas plusieurs expressions différentes ont été développées pour déterminer le nombre de Reynolds critique de transition (Nandakumar et Masliyah, 1986), et la corrélation suivante donnée par Srinivasan et al. (1968) est plus largement cité:

$$Re_{tr} = 2100\left(1 + 12\sqrt{\frac{d}{2R}}\right) \qquad 10 < \frac{2R}{d} < \infty \quad \text{(II.7)}$$

Pour le transfert de chaleur et les écoulements turbulent pleinement développé, Mori et Nakayama (1967) suggèrent les corrélations suivantes:

$$Nu = \begin{cases} \dfrac{Pr}{26.2\left(Pr^{2/3} - 0.074\right)} Re^{4/5}\left(\dfrac{d}{2R}\right)^{1/10}\left[1 + 0.098\left\{Re\left(\dfrac{d}{2R}\right)^2\right\}^{1/5}\right] & \text{(II.8}a\text{)} \\ Pour\ Pr \approx 1 \\ \dfrac{Pr^{0.4}}{41.0} Re^{5/6}\left(\dfrac{d}{2R}\right)^{1/12}\left[1 + 0.061\left\{Re\left(\dfrac{d}{2R}\right)^{2.5}\right\}^{1/6}\right] & \text{(II.8}b\text{)} \\ Pour\ Pr > 1 \end{cases}$$

Ces corrélations sont valable pour :

$$Re_{tr} < Re < 10^{05},\ \ 6.7 < 2R/d < 346, et\ 0 < H/2R < 25.4$$

Dans une étude expérimentale plus récente sur les écoulements turbulents de l'eau et l'huile dans les tubes en spirale ondulée, Dong et al. (2001) ont élaboré un nouvel ensemble

d'analogie basée sur le coefficient de frottement et les corrélations nombre de Nusselt. Adopter une approche empirique, combinée à une analyse statistique d'une base de données assez importante pour les coefficients de transfert thermique et les facteurs de friction pour les différents types de rugosité Fig. II.6, et Ravigururajan Bergles (1996) ont proposé les corrélations du nombre de Nusselt suivants:

$$Nu = Nu_0 \left\{ 1 + \left[2.64 Re^{0.036} \left(\frac{e}{d}\right)^{0.212} \left(\frac{p}{d}\right)^{-0.21} \left(\frac{\alpha}{90}\right)^{0.29} Pr^{-0.024} \right]^7 \right\}^{1/7} \quad \text{(II.9)}$$

$$Nu_0 = \frac{Re.Pr(f_0/2)}{1+12.7\sqrt{f_0/2\left(Pr^{2/3}-1\right)}} \quad \text{(II.10)}$$

$$f_0 = (1.58 lnRe - 3.28)^{-2} \quad \text{(II.11)}$$

Pour les tubes à ailettes en interne soit des ailettes droites ou en spirale et le régime d'écoulement laminaire, Watkinson et al. (1975) donnent suivant les diamètres hydrauliques deux corrélations différentes pour les nombres de Nusselt respectifs:

- Pour des tubes à ailettes droites :

$$Nu_h = \frac{1.08 logRe_h}{n^{0.5}\left(1+0.01 Gr_h^{1/3}\right)} Re_h^{0.46} Pr^{1/3} \left(\frac{L}{d_h}\right)^{1/3} \left(\frac{\mu_w}{\mu_b}\right)^{0.14} \quad \text{(II.12)}$$

- Pour des tubes à ailettes en spirale :

$$Nu_h = \frac{8.533 logRe_h}{1+0.01 Gr_h^{1/3}} Re_h^{0.26} Pr^{1/3} \left(\frac{t}{p}\right)^{0.5} \left(\frac{L}{d_h}\right)^{1/3} \left(\frac{\mu_w}{\mu_b}\right)^{0.14} \quad \text{(II.13)}$$

Carnavos (1979) a recommandé pour les écoulements turbulents pour les mêmes types de Watkinson et al. (1975) la corrélation suivant :

$$Nu_h = 0.023 Re_h^{0.8} Pr^{0.4} \left(\frac{A_f}{A_{fi}}\right)^{0.1} \left(\frac{A_i}{A}\right)^{0.1} (sec\alpha)^3 \quad \text{(II.14)}$$

Ces corrélations ont été montré pour décrire les données pour l'air, l'eau, l'huile, et de l'éthylène glycol (Watkinson et al, 1975; Carnavos, 1979; Marner et Bergles, 1989).

Pour un écoulement traverser faisceaux de tube à ailettes, un ensemble assez important de données expérimentales peuvent être trouvés dans la littérature (Zhukauskas, 1989), et les corrélations recommandé pour les tubes à ailettes circulaires ou hélicoïdales sont donnés ci-dessous :

- Pour des tubes à ailettes circulaires ou hélicoïdales en ligne :

$$Nu = 0.303\varepsilon^{-0.375} Re_h^{0.625} Pr^{0.36} \left(\frac{Pr}{Pr_w}\right)^{0.25} \quad \text{(II.15)}$$

Cette corrélation est valable pour :

$$5 \times 10^{03} \leq Re \leq 10^{05}, \quad 5 \leq \varepsilon \leq 12\,, \qquad 1.72 \leq S_T/D \leq 3\,, \qquad 1.8 \leq S_L/D \leq 4$$

- Pour les tubes en quinconce avec ailettes circulaires ou hélicoïdales :

$$Nu = C_2 Re^a Pr^b \left(\frac{S_T}{S_L}\right)^{0.2} \left(\frac{P_f}{D}\right)^{0.18} \left(\frac{h_f}{D}\right)^{-0.14} \left(\frac{Pr}{Pr_w}\right)^{0.25} \quad \text{(II.16)}$$

où :

$$C_2 = \begin{cases} 0.192, & a = 0.65, \quad b = 0.36 & for\ 10^{02} \leq Re \leq 2 \times 10^{04} \\ 0.0507, & a = 0.8, \quad b = 0.4 & for\ 2 \times 10^{04} \leq Re \leq 2 \times 10^{05} \\ 0.0081, & a = 0.95, \quad b = 0.4 & for\ 2 \times 10^{05} \leq Re \leq 1.4 \times 10^{06} \end{cases}$$

et la gamme des paramètres sont:

$$0.06 \leq \frac{p_f}{D} \leq 0.36, 0.07 \leq \frac{h_f}{D} \leq 0.715, \qquad 1.1 \leq \frac{S_T}{D} \leq 4.2, \qquad 1.03 \leq \frac{S_L}{D} \leq 2.5$$

Pour le transfert de chaleur d'un écoulement laminaire dans les tubes maintenus à une température de paroi uniforme (UWT), la corrélation suivante développée par Manglik et Bergles (1993) est recommandée:

$$Nu_m = 4.612(\mu_b/\mu_w)^{0.14}\{[(1 + 0.0951Gz^{0.894})^{2.5} + 6.413 \times 10^{-09}(SwPr^{0.391})^{3.835}]^2 + 2.132 \times 10^{-14}(Re_a.Ra)^{2.23}\}^{0.1} \quad \text{(II.17)}$$

Pour les tubes avec l'état de flux de chaleur uniforme (UHF), la corrélation conçu par Hong et Bergles (1976) pour les écoulements turbulents pleinement développée avec des variations des propriétés visqueuses comme suit:

$$Nu_z = 5.172\left[1 + 5.484 \times 10^{-3}Pr^{0.7}\left(\frac{Re_a}{y}\right)^{1.25}\right]^{0.5}\left(\frac{\mu_b}{\mu_w}\right)^{0.14} \quad \text{(II.18)}$$

Par ailleurs, pour tenir compte l'influence de la convection naturelle sur les écoulements turbulents en UHF, Bandyopadhyay et al. (1991) ont étendu la corrélation de Hong et Bergles d'inclure convection mixte en tant :

$$Nu_z = \left[Nu_{z,HB}^9 + 1.17\left(Ra^{*0.181}\right)^9\right]^{1/9} \quad \text{(II.19)}$$

Pour le transfert de flux de chaleur turbulent avec Re ≥ 1004, Manglik et Bergles (1993b) ont proposé une corrélation nombre de Nusselt qui est exprimée en :

$$Nu = 0.023Re^{0.8}Pr^{0.4}\left(1 + \frac{0.769}{y}\right)\left[\frac{\pi+2-(2\delta/d)}{\pi-(4\delta/d)}\right]^{0.2}\left[\frac{\pi}{\pi-(4\delta/d)}\right]^{0.8}\emptyset \quad \text{(II.20 a)}$$

Où le facteur de correction de la propriété φ est donnée par :

$$\emptyset = \left(\frac{\mu_b}{\mu_w}\right)^n \text{ Ou bien } \emptyset = \left(\frac{T_b}{T_w}\right)^m \quad \text{(II.20 b)}$$

Avec :

$$n = \begin{cases} 0.18 & \text{Refroidissment} \\ 0.30 & \text{Chauffage} \end{cases} \quad \text{et } m = \begin{cases} 0.45 & \text{Refroidissment} \\ 0.15 & \text{Chauffage} \end{cases}$$

Les signes de l'équation. (II.20) ont été trouvés (Manglik et Bergles, 1993; 2002) pour décrire la majorité des données expérimentales pour une très large gamme de rapport de torsion

de bandes ($2 \leq y \leq \infty$) rapportés dans la littérature à la fois pour le gaz et liquides dans des tubes de flux turbulents circulaires avec torsadée bandes inserts.

Royal et Bergles (1978) semblent avoir déclaré la seule étude sur la condensation de la vapeur dans des tubes équipés de bandes torsadées inserts. Ils ont trouvé que le coefficient moyenne de transfert de chaleur a été augmenté de 30% sur les valeurs de tube vide ajusté au bandes de y = 3,3 et 7,0, où l'amélioration a été plus élevé pour la torsion serré bande. Basé sur ces données, la corrélation suivante a été donnée:

$$Nu_{h,m} = 0.0265\left[\frac{Gd_h}{\mu_t}\left(1 - x_m + \frac{x_m\rho_t}{\rho_g}\right)\right]^{0.8} Pr_t^{0.33}\left[1 + 160\left(\frac{H^2}{Ld}\right)^{1.91}\right] \quad \text{(II.21)}$$

Dans de nombreux cas, il est pratique d'avoir de simples équations pour l'estimation des coefficients de transfert thermique. Ci-dessous sont des corrélations recommandées pour un écoulement laminaire de convection dans des géométries différentes et avec les deux types des conditions aux limites le plus utilisées Driclet (UWT) et newuman (UHF) sont respectivement:

$$Nu_0 = 3.657 + \frac{0.0677\left(RePr\frac{D}{L}\right)^{1.33}}{1+0.1Pr\left(Re\frac{D}{L}\right)^{0.3}} \quad \text{(II.22)}$$

$$Nu_0 = 4.364 + \frac{0.086\left(RePr\frac{D}{L}\right)^{1.33}}{1+0.1Pr\left(Re\frac{D}{L}\right)^{0.83}} \quad \text{(II.23)}$$

qui est valable sur toute la gamme $0.7 < Pr < 7$.

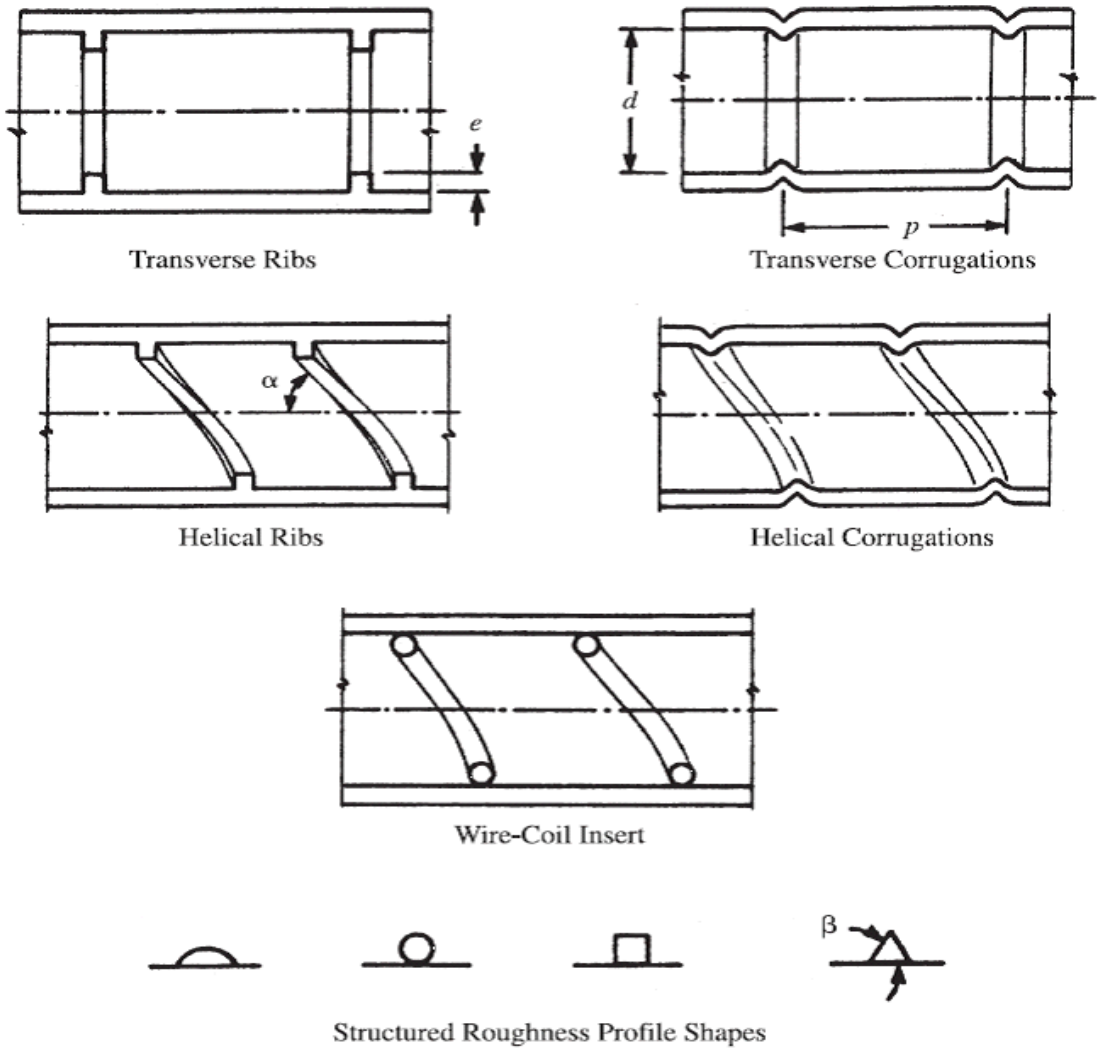


Fig. II.5: Différents types de rugosité structurée et leurs profilés [Ravigururajan, T.S., and Bergles, A.E. 1996]

II-6 Conclusion

Face à la diversité des techniques d'intensification, le concepteur d'un échangeur de chaleur est amené à prendre en compte de nombreux éléments à savoir :

- l'aspect constructif : tenue aux vibrations, tenue aux corrosions et liaison entre les tubes et les plaques tubulaires ;
- l'aspect économique : coût de la surface et disponibilité sur le marché du tube avec le matériau qu'impose éventuellement le procédé ;
- l'aspect thermique : performances thermohydrauliques et comportement thermique à l'encrassement.

On conçoit que le choix de géométries adaptées à un procédé particulier soit le résultat d'un compromis sur les différents aspects évoqués ci-dessus.

Chapitre III

Résultats & Discussions

III.1 Introduction

Une approche algorithmique nécessite l'écriture (avant la transcription dans un quelconque langage de programmation) du processus à suivre pour résoudre le problème. C'est un avantage lorsque tous les cas ont été prévus à l'avance par l'algorithmicien.

La seconde approche possible est celle de l'intelligence artificielle (IA), avec pour applications les plus connues les systèmes experts. Ici, la résolution du problème est confiée à un ensemble de règles données par l'expert humain du domaine. Il n'en demeure pas moins que toutes les règles doivent avoir été exprimées préalablement au traitement, et que le programme demeure binaire dans son exécution. Les cas qui n'ont pas été prévus par l'expert ne seront pas correctement traités. L'exécution reste totalement déterministe.

Une troisième approche au traitement de l'information semble donc s'offrir à nous, où l'on cherche à s'inspirer du traitement de l'information effectué par le cerveau. Les réseaux de neurones artificiels sont dotés de propriétés qui, comme l'apprentissage à partir d'exemples, semblent prometteuses dans certains domaines d'applications. En fait, est considéré comme éligible, tout problème qui se représente sous la forme d'une fonction de mise en correspondance entre un espace d'entrée et un espace de sortie, dès lors que l'on ne dispose que d'exemples de comportement de cette fonction [T.L. Fine, 1999]. Les réseaux de neurones utilisés dans ce chapitre, les réseaux supervisés, sont appliqués pour prédire les paramètres énergétiques influentes au nombre de Nusselt.

Enfin la performance du réseau peut être, éventuellement, comparée avec celles des autres types de modélisation. La figure III.1 résume les étapes de la méthodologie utilisée dans ce travail pour la modélisation neuronale pour notre system des grandeurs mesurables.

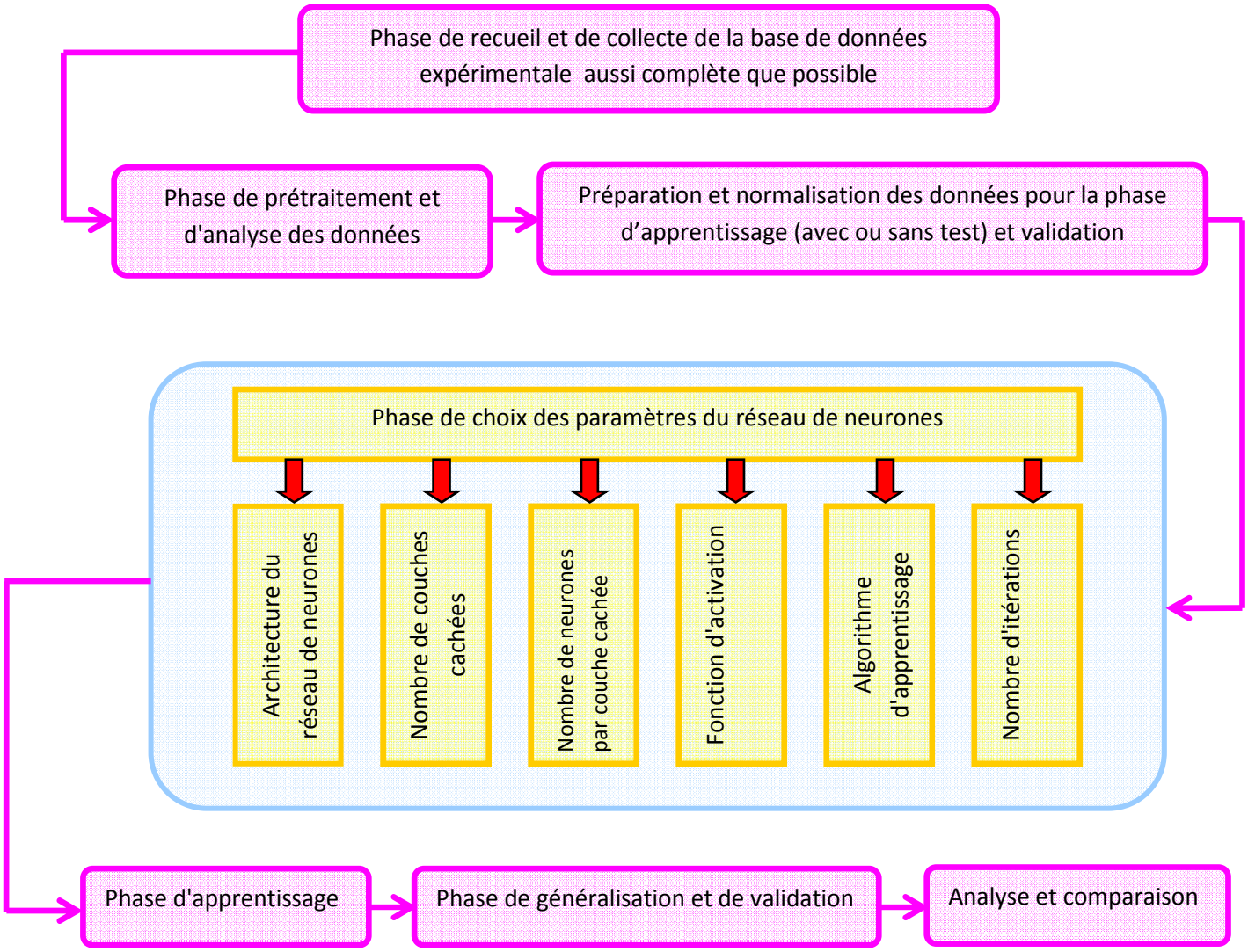


Fig. III.1 : Procédure de modélisation par les réseaux de neurones [C. SI-Moussa et al., 2008]

III.2. Base de données

L'élaboration d'une base de données est une phase décisive dans la conception d'un modèle neuronal. Elle doit être exécutée avec la plus grande précaution car toute erreur à ce stade va connaître des influences importantes tant sur la convergence du modèle que sur la généralisation.

Dans le but de constituer une base de données fiable et représentative, nous avons suivi une méthodologie, inspirée de l'étude bibliographique [G. Dreyfus et al, 2002], qui revêt les étapes suivantes :

- sélection des entrées du modèle neuronal,
- collecte des données,

- mise en forme des données,
- normalisation des données.

Parmi les corrélations empiriques présentées dans le chapitre précédent, on a sélectionné ceux dont tous les paramètres qui entrent dans le calcul du nombre de Nusselt sont connus et qui sont données comme suit :

$$Nu_{m,F} = \left[\left\{4.364 + \frac{4.636}{[1 + (1342/PrHe^2)]^2}\right\}^3 + 1.816\left\{\frac{He}{[1 + (1.15/Pr)]}\right\}^{3/2}\right]^{1.3} \quad \text{(III. 1)}$$

$$Nu_{m,T} = \left[\left\{3.657 + \frac{4.343}{[1 + (957/PrHe^2)]^2}\right\}^3 + 1.158\left\{\frac{He}{[1 + (0.477/Pr)]}\right\}^{3/2}\right]^{1.3} \quad \text{(III. 2)}$$

$$Re_{tr} = 2100\left(1 + 12\sqrt{\frac{d}{2R}}\right) \qquad 10 < \frac{2R}{d} < \infty$$

$$Nu = \begin{cases} \dfrac{Pr}{26.2\left(Pr^{2/3} - 0.074\right)} Re^{4/5}\left(\dfrac{d}{2R}\right)^{1/10}\left[1 + 0.098\left\{Re\left(\dfrac{d}{2R}\right)^2\right\}^{1/5}\right] & \text{(III. 3}a\text{)} \\ Pour\ Pr \approx 1 \\ \dfrac{Pr^{0.4}}{41.0} Re^{5/6}\left(\dfrac{d}{2R}\right)^{1/12}\left[1 + 0.061\left\{Re\left(\dfrac{d}{2R}\right)^{2.5}\right\}^{1/6}\right] & \text{(III. 3}b\text{)} \\ Pour\ Pr > 1 \end{cases}$$

$$Nu = Nu_0\left\{1 + \left[2.64Re^{0.036}\left(\frac{e}{d}\right)^{0.212}\left(\frac{p}{d}\right)^{-0.21}\left(\frac{\alpha}{90}\right)^{0.29} Pr^{-0.024}\right]^7\right\}^{1/7} \quad \text{(III. 4)}$$

$$Nu_0 = \frac{Re.Pr(f_0/2)}{1 + 12.7\sqrt{f_0/2\,(Pr^{2/3} - 1)}}$$

$$f_0 = (1.58lnRe - 3.28)^{-2}$$

$$Nu = 3.657 + \frac{0.0677\left(RePr\frac{D}{L}\right)^{1.33}}{1 + 0.1Pr\left(Re\frac{D}{L}\right)^{0.3}} \quad \text{(III. 5)}$$

$$Nu = 4.364 + \frac{0.086\left(RePr\frac{D}{L}\right)^{1.33}}{1 + 0.1Pr\left(Re\frac{D}{L}\right)^{0.83}} \quad \text{(III. 6)}$$

III.3. Sélection des variables d'entrées et de sorties

Lorsqu'on conçoit un modèle à partir de mesures, il est indispensable que l'ensemble des variables d'entrée soit aussi réduit que possible ; en effet, à chaque entrée du modèle correspond un nombre de paramètres égal au nombre de neurones cachés. Il est donc nécessaire :

- que la représentation des entrées soit aussi compacte que possible,
- que la représentation des entrées contienne tous les facteurs pertinents par rapport à la grandeur à modéliser, mais seulement ceux-ci. En effet, la présence de variables qui ont, sur la grandeur à modéliser, moins d'influence que les perturbations ou le bruit, entraîne celle de paramètres superflus et introduit des variations non significatives sur les variables d'entrées, qui constituent autant de sources d'erreur pour le modèle.

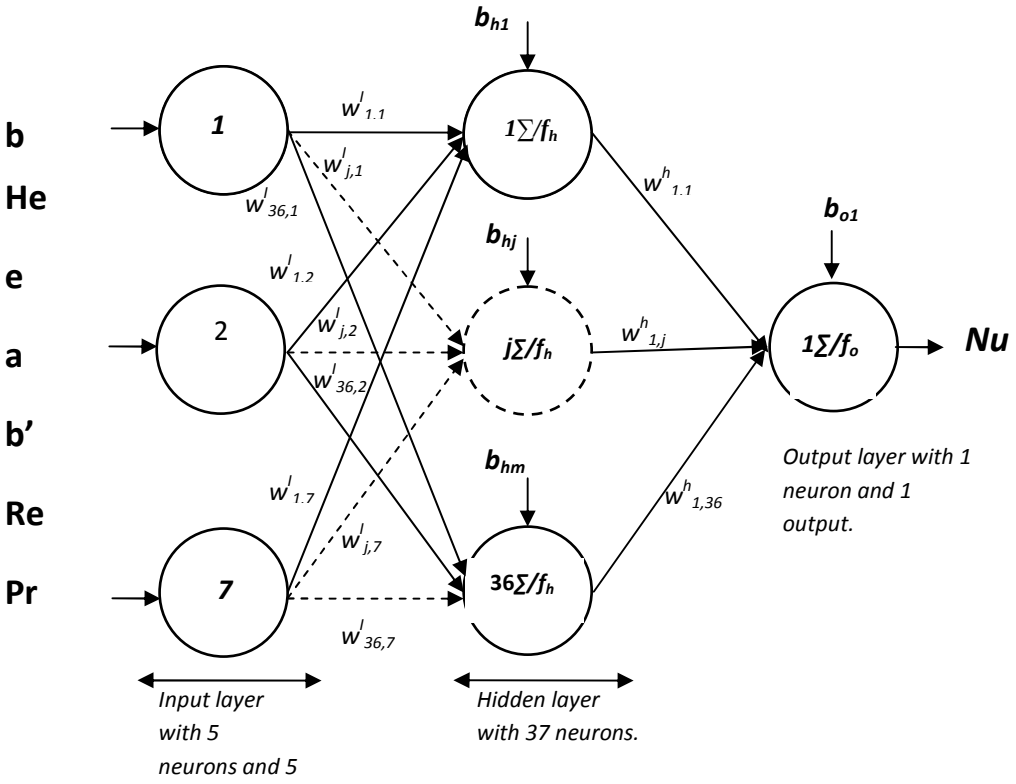


Fig. III.2 : Schéma simplifié de l'architecture du réseau neuronal

III.4. Collecte des données

Une fois les entrées pertinentes du modèle sont sélectionnées, l'objectif de cette étape est de les recueillir. La base de données ainsi constituée doit répondre à deux exigences [G. Dreyfus al, 2002] :

- contenir le maximum d'échantillons ou d'exemples (génériques et/ou expérimentaux),
- être représentative des données susceptibles d'intervenir en phase d'utilisation du modèle neuronal.

Les bases de données utilisées (Annexe 2) dans le présent travail ont été obtenu grâce à la programmation numérique sous Matlab (Annexe 1) des équations retenues.

III.5. Mise en forme des données

Dans cette étape les données sont d'abord subdivisées, au moins, en deux sous ensembles ; un ensemble pour l'apprentissage et l'autre est réservé à la validation. L'échantillonnage des deux ensembles est fait de façon qu'ils reflètent parfaitement la base de données de départ. C'est-à-dire, on choisit toujours, à partir du tableau des données, deux lignes pour l'apprentissage et la troisième pour la validation et ainsi de suite jusqu'à la fin, soit 75% de la totalité de la base de données pour l'apprentissage et les 25% pour la validation [H. Demuth et al., 2009]. Ensuite, l'ensemble des données est écrit sous forme matricielle. Comme il s'agit là d'un apprentissage supervisé, on a donc des matrices représentants l'entrée du modèle et des matrices représentants des sorties désirées selon le cas.

III.6. Normalisation des données

La normalisation est une étape importante dans le processus de l'élaboration des données. Elle devient dans la plus part des cas, une exigence pour les matrices des inputs, du fait qu'on utilise souvent des fonctions de transfert bornées de type sigmoïdes dans les modélisations statiques.

Dans ce travail, on a normalisé la sortie qui comprend à une seule paramètre par la fonction logarithmique et l'entrée est normalisé par l'instruction Matlab "mapminmax", qui effectue une normalisation de la valeur maximale et minimale de chaque rangée [H. Demuth et al., 2009].

$$y = \frac{(y_{max} - y_{min})(x - x_{min})}{(x_{max} - x_{min})} + y_{min} \qquad \text{(III.7)}$$

Tableau III.1 : Poids et biais du RNA optimale pour le système

Poids entre la couche d'entré et la couche cachée Wh,i							Bais couche cachée bh	Poids entre la couche cachée et la couche de sortie, Wo,h	Bais couche de sortie bo
1,30	0,99	1,10	-0,34	-0,16	0,56	1,06	-2,34	0,18	
1,10	-1,02	0,57	0,80	0,37	0,92	-1,12	-2,21	0,85	
0,85	1,19	0,16	0,57	-1,69	-0,01	-0,34	-2,07	0,70	
1,30	-0,27	0,88	-1,52	0,61	0,46	-0,09	-1,93	0,35	
0,64	-0,63	0,94	-1,39	0,02	1,39	0,01	-1,80	0,50	
0,82	-0,46	1,77	0,13	-1,00	-0,10	-0,66	-1,67	-0,22	
0,90	-0,46	0,62	1,29	-0,69	-1,12	-0,78	-1,54	-0,41	
-0,46	1,14	0,36	0,18	-0,65	1,27	1,25	1,43	0,45	
0,29	0,85	0,35	-0,04	1,46	-0,43	1,49	-1,29	-0,72	
1,23	-0,72	-0,63	-0,38	1,59	0,37	0,19	-1,04	-0,17	
-1,01	-1,36	-1,10	0,85	-0,53	0,56	0,27	1,01	-0,69	
0,54	-0,37	1,06	0,72	1,03	-1,90	0,42	-0,93	-1,10	
0,51	1,30	-1,23	-0,29	0,20	0,47	1,28	-0,73	0,83	
0,05	-0,25	0,93	1,11	-0,55	-1,47	0,85	-0,60	-0,57	
0,24	0,38	1,18	-1,49	-0,30	0,08	1,25	-0,47	0,15	
1,43	-0,60	-1,16	-1,01	-0,91	-0,03	-0,21	-0,32	-0,83	
1,08	-0,49	-1,09	0,96	0,49	-0,77	-1,16	-0,15	-0,65	
0,18	1,23	-0,46	1,36	0,77	1,02	0,49	-0,08	0,90	
0,11	-0,76	-1,43	-0,13	1,21	0,79	0,17	0,07	0,82	
1,10	-0,51	0,45	1,37	0,03	0,59	-1,25	0,20	-0,29	
0,05	0,16	1,06	0,65	0,52	-0,98	-1,71	0,49	0,50	
-1,09	1,17	0,85	-0,54	1,11	-0,47	-0,66	-0,44	0,03	
-0,27	-1,60	-1,45	0,53	-0,49	-0,08	-0,43	-0,60	-0,13	
-1,25	0,11	-1,00	0,94	0,97	0,74	0,74	-0,67	1,08	
0,78	1,21	-0,49	-1,20	0,03	-1,40	0,78	0,85	0,84	
1,12	1,77	-1,01	-0,08	0,07	0,24	-0,18	0,89	-0,63	-0.23
0,04	-0,26	1,16	-1,21	1,05	-1,26	0,01	1,14	0,49	
-1,17	0,95	-0,70	-0,48	0,81	-0,91	1,13	-1,21	0,43	
0,26	0,46	-1,24	1,03	-1,23	0,04	0,48	1,34	0,65	
-0,92	-0,24	-1,20	0,96	1,32	0,66	-0,11	-1,50	0,32	
-0,76	-1,22	0,95	-1,01	-0,60	0,64	0,87	-1,71	0,61	
1,06	1,54	0,87	-0,05	0,15	-1,12	0,32	1,83	-0,58	
-0,79	-1,10	0,49	1,00	0,40	1,00	1,11	-1,93	-0,75	
0,33	-0,69	0,15	-0,40	-0,98	-1,79	-0,72	2,07	-0,01	
0,39	0,79	-1,48	1,04	-0,15	0,99	-0,14	2,31	0,27	
1,33	1,38	1,14	-0,47	0,47	0,09	0,63	2,27	-0,46	

III.7. Conception des réseaux

La conception d'un modèle neuronal consiste à faire une étude d'évaluation des éléments constitutifs du réseau en fonction de la performance de modélisation souhaitée. Elle vise à fixer les éléments suivants :

- le type du réseau,
- le nombre de couches cachées,
- le nombre de neurones dans les couches,
- la fonction de transfert des neurones,
- l'algorithme d'apprentissage,
- le nombre d'itérations (cycles).

Dans notre cas, il s'agit d'approximer les grandeurs physiques, qui sont des fonctions suffisamment régulières, par un réseau de neurones. Il est donc question, de faire une modélisation statique indépendamment du temps, le réseau adéquat ne peut être en général qu'un réseau non bouclé type MLP (Multi Layer Perceptron). La propriété fondamentale de ces réseaux est l'approximation universelle, qui s'énonce de la façon suivante :
"Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire" [G. Dreyfus et al., 2002], [A. Bouzidi, 2006].

C'est cette propriété appelée encore théorème d'existence [G. Dreyfus et al., 2002] qui a justifié notre choix de l'architecture de réseaux de neurones à une couche cachée, type MLP, avec des tangentes sigmoïdes comme fonctions de transfert des neurones cachés et une fonction linéaire (identité) pour le neurone de sortie. L'algorithme d'apprentissage le mieux adapté et après confirmation par une étude évaluative, est celui de Levenberg-Marquardt.

III.7.1. Type du réseau

Le réseau de type feedforward est le réseau le plus utilisé pour l'approximation des fonctions des données statiques. Il est d'ailleurs le réseau par défaut pour les interfaces graphiques de réseaux de neurones, ***nntool*** et ***nftool***, de MATLAB®. Par conséquent, le réseau de type feedforward, fonction ***newff*** de MATLAB®, a été notre choix.

III.7.2. Nombre de couches cachées

Comme nous l'avons mentionné plus haut, il n'y a pas de règles générales qui fixent le nombre de couches cachées d'un MLP pour un problème donné. Par conséquent la stratégie d'essai et erreur (trial and error) est la plus raisonnable (Curry et Morgan (2006)). Le critère d'arrêt est la capacité de généralisation (phase de validation du réseau) et une fonction de performance donnée (Plumb et al. (2005)).

La capacité de généralisation est évaluée en calculant un vecteur du meilleur ajustement linéaire [α, β, R] de la sortie calculée par le RNA en fonction de la sortie cible pour un ensemble de données (égal à un tiers de la base de données et bien distribuées sur l'espace de données) non utilisées dans la phase d'apprentissage.

Ce vecteur est constitué de la pente de la droite de régression linéaire, α, de l'ordonné à l'origine, β, et du coefficient de corrélation, R. Idéalement ce vecteur doit être égal à [α, β, R] = [1, 0, 1]. Ce vecteur est aisément obtenu à l'aide de la fonction ***postreg*** de MATLAB®. Pour la fonction de performance nous avons utilisé l'erreur relative absolue moyenne (ERAM) ou l'erreur absolue moyenne (EAM), entre la sortie calculée et la sortie cible, à fixer pour chaque sortie.

Nous avons implanté cette stratégie dans ce travail en commençant par une seule couche et on fait varier le nombre de neurones en commençant par un nombre réduit (égal au nombre des neurones de la couche d'entrée) et en ajoutant chaque fois un neurone jusqu'à un nombre maximum (égal à cinq fois le nombre des neurones de la couche d'entrée). Cette variation se fait à l'intérieur d'une boucle externe qui fait varier l'algorithme d'apprentissage, elle même à l'intérieur d'une autre boucle externe qui fait varier la fonction d'activation.

Si aucune topologie à une seule couche, qui vérifie le critère d'arrêt, n'est pas obtenue on considère alors l'utilisation d'une deuxième couche cachée.

$$ERAM(\%) = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{Nu^{calculé} - Nu^{estimé}}{Nu^{calculé}}\right|_i \qquad \text{(III.8)}$$

Erreur Relative Absolue Moyenne =0.8416.

III.7.3. Algorithme d'apprentissage

En ce qui concerne le choix de l'algorithme d'apprentissage, le programme commence par l'algorithme par défaut de MATLAB®, à savoir l'algorithme de Levenberg-Marquardt **(*trainlm*)**. Il utilise ensuite l'algorithme de la régularisation Bayésienne **(trainbr).** L'adoption de ce choix, bien qu'il ne soit pas probablement le meilleur, est basée sur les conclusions du travail de Plumb et al. (2005). Puisqu'une étude récente (Torrecilla et al. 2009) stipule que l'algorithme de descente des gradients conjugués avec la méthode de mise à jour de Polak-Ribiére **(*traincgp*)** est le meilleur pour la modélisation des problèmes considérés par les auteurs de cette étude. Il faut remarquer que ce genre de résultats qui apparaissent souvent contradictoires fait en sorte que la modélisation neuronale soit considérée par certains plus de l'art que de la science.

III.7.4. Fonction d'activation

Pour le choix de la fonction d'activation nous avons opté pour les fonctions sigmoïdes pour les couches cachées et la fonction linéaire simple pour la couche de sortie. On commence par la fonction tangente hyperbolique (fonction ***tansig*** de MATLAB®) ensuite la fonction logistique (fonction ***logsig*** de MATLAB®). Pour les réseaux avec plusieurs couches cachées nous avons utilisé la même fonction d'activation.

Tableau III.2 : Structure du réseau neuronal (ANN)

Type of network	Training Algorithm	Input layer	Hidden layer		Output layer	
		No. of neurons	No. of neurones	Activation function	No. of neurones	Activation function
Algorithme : FFBP NN	BRBP using Levenberg-Marquardt optimisation.	7	36	TANSIG	1	TANSIG

III.7.5. Nombre d'itération

Le nombre d'itération est lié à un autre facteur, très souvent négligé par certains, qui est l'initialisation des poids synaptiques et les biais des neurones cachés et neurones de sortie. L'utilisation de différentes initialisation aide à éviter le problème se convergence vers le minimum local.

$$(Nu)_{estimé} = 0.99\ (Nu)_{calculé} + 0.0033 \qquad \text{(III.9)}$$

Dans ce travail nous avons opté pour l'initialisation aléatoire par défaut de MATLAB® et l'exécution de l'apprentissage de la même topologie d'un réseau avec un nombre suffisant (20 à 30) de différentes initialisations.

III.7.6. Apprentissage

Pour réduire le temps de recherche et éviter le problème de sur apprentissage nous avons adopté l'apprentissage atténué avec tous les algorithmes d'apprentissage à l'exception de l'algorithme de régularisation Bayésienne.

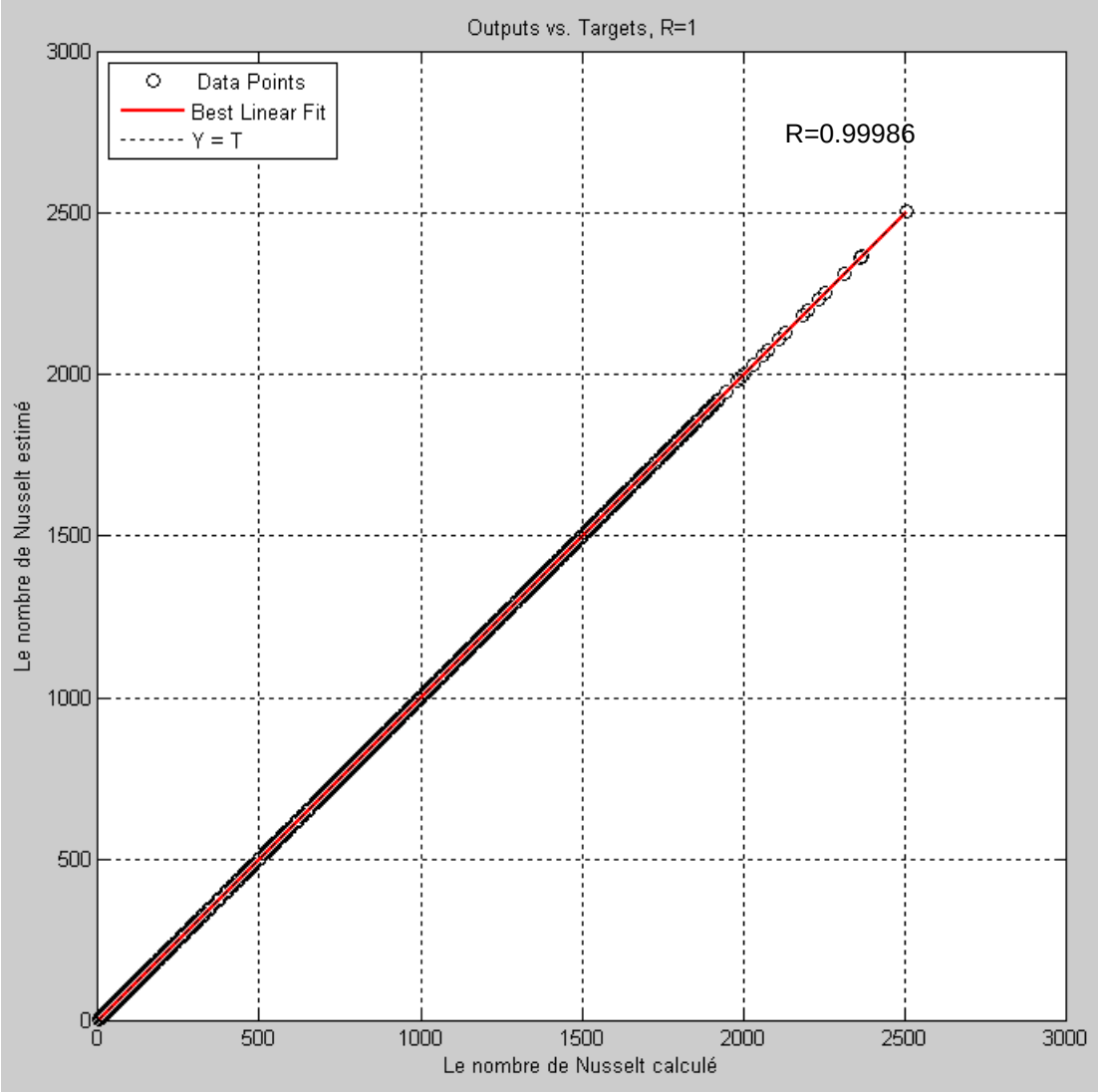


Fig. III.3: Courbe de régression linéaire lors de la phase d'apprentissage

III.7.7. Validation ou Généralisation

Après l'obtention de la meilleure structure du RN qui vérifie la fonction de performance choisie pour chaque sortie, on présente au réseau un ensemble de données de validation pour tester sa capacité de généralisation. L'évaluation est réalisée par le calcul de vecteur du meilleur ajustement linéaire cité plus haut est une évaluation visuelle de la distribution des sortie cible par rapport à la droite de l'ajustement linéaire idéal.

$$Nu = g\left[\sum_{h=1}^{36}\left(g\left[\sum_{i=1}^{7}(X_i \,.\, W_{h,i}) + b_h\right]\right) \times W_{1,h} + b_1\right] \quad \text{(III.10)}$$

$$Nu = g\left[\sum_{h=1}^{36}\left(g\left[W_{h,1} \times \boldsymbol{e} + W_{h,2} \times \boldsymbol{a} + W_{h,3} \times \boldsymbol{b} + W_{h,4} \times \boldsymbol{b1} + W_{h,5} \times \boldsymbol{He} + W_{h,6} \times \boldsymbol{Pr} + Wh,7 \times \boldsymbol{Re} + bh\right) \times W1,h + b1\right.\right. \quad \text{(III.11)}$$

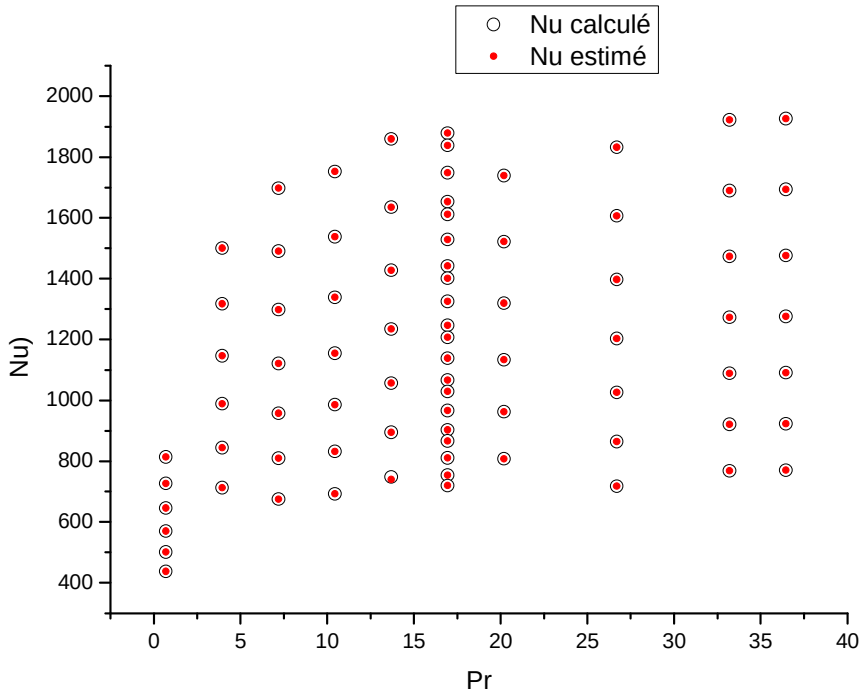


Fig. III.4: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.1

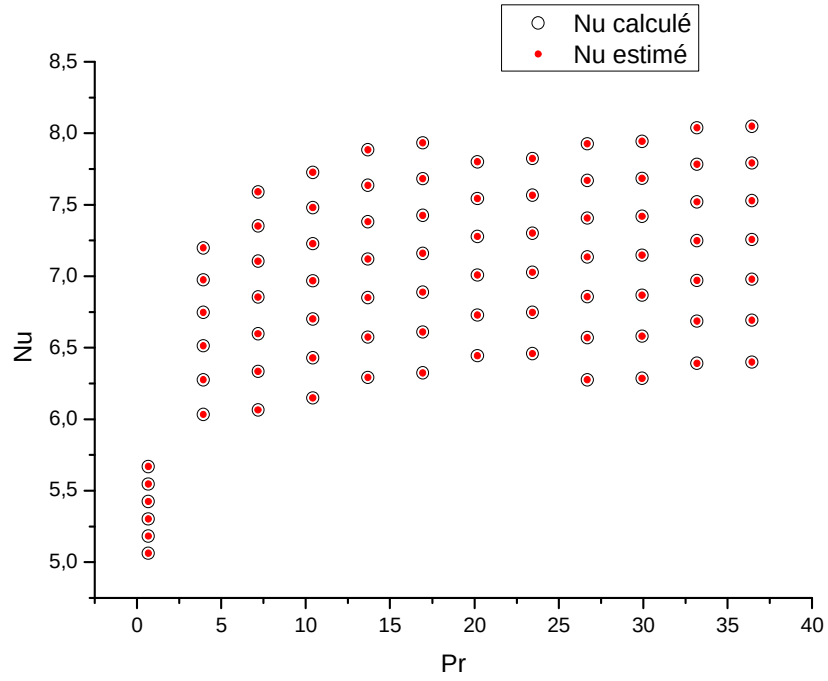


Fig. III.5: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.2

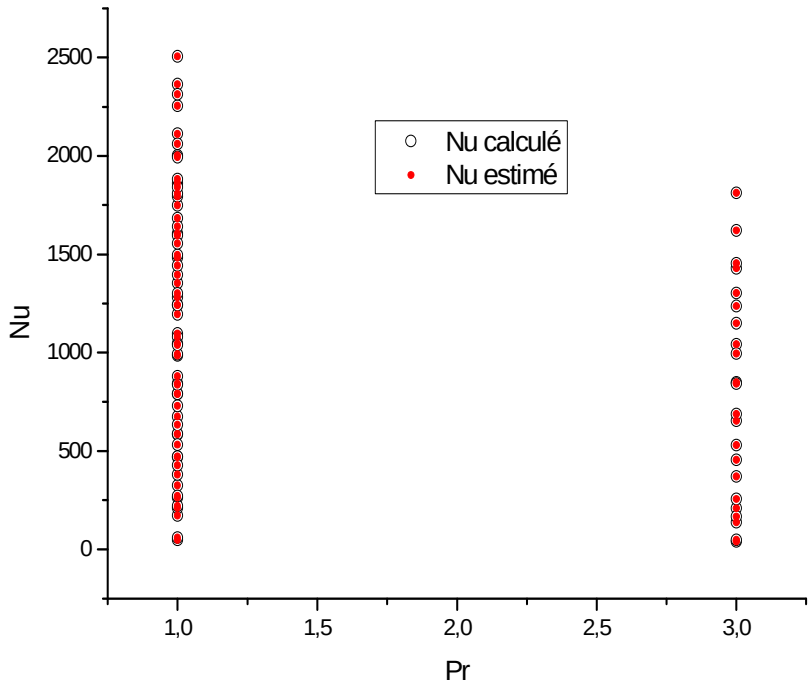


Fig. III.6: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.3

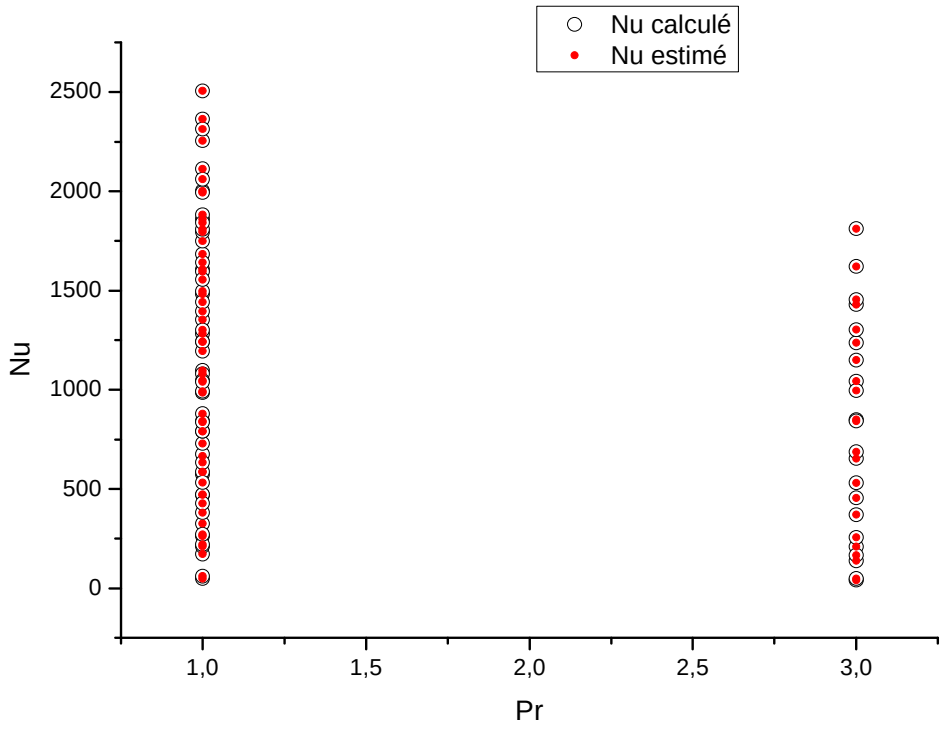


Fig. III.7: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.4

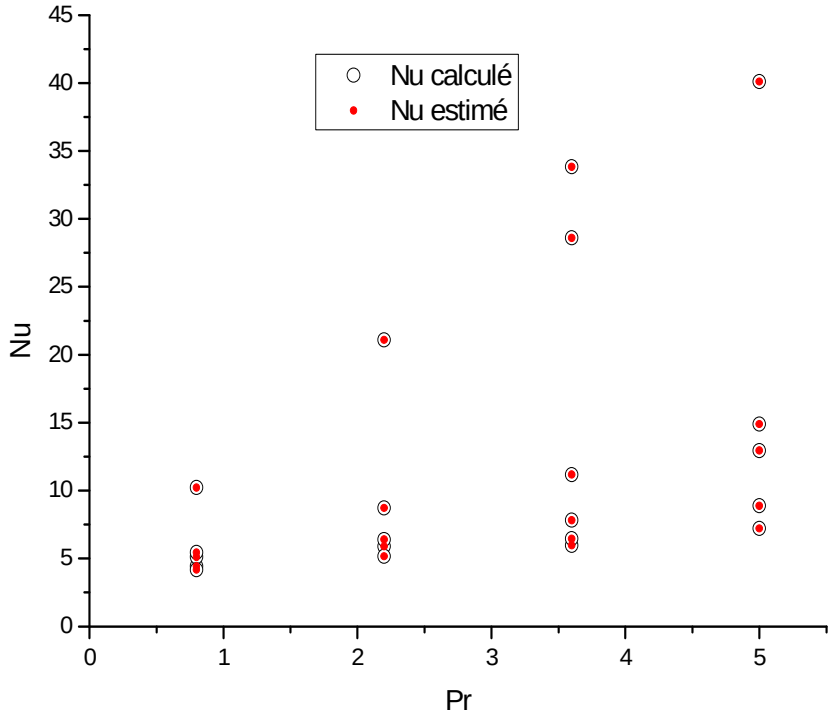


Fig. III.8: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.5

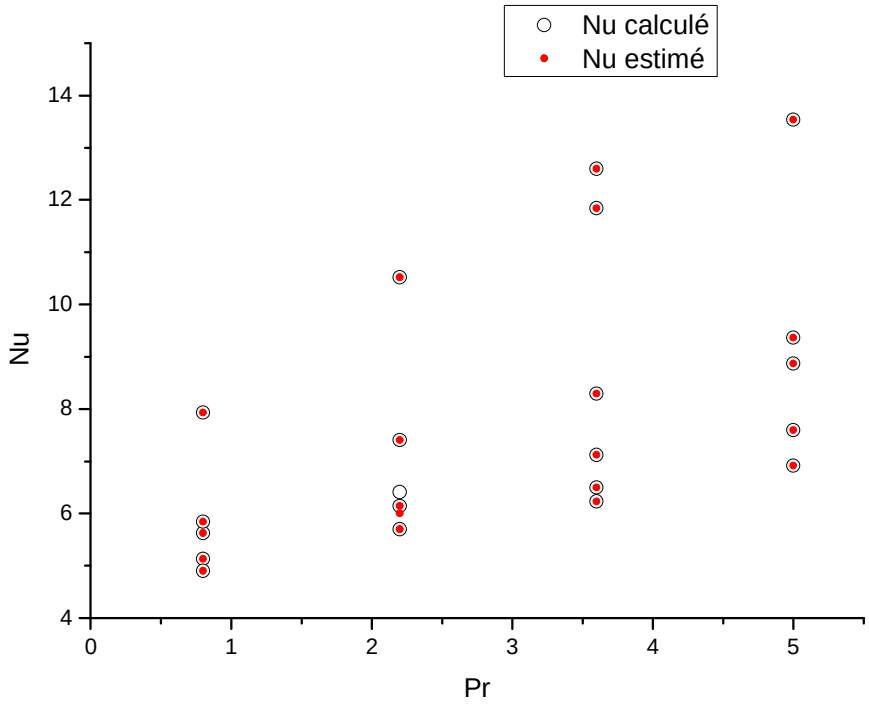


Fig. III.9: Comparaison entre les valeurs de Nu calculé et Nu estimé à partir de l'équation III.6

III.8 Discussion des résultats

La courbe de régression linéaire de RNA pour toute la base de données est visualisée dans la figure (III.3). la juxtaposition de la première bissectrice et de la droite du meilleur ajustement linéaire de la sortie avec la cible, ainsi que la distribution des points obtenus par calcul numérique montrent l'excellent accord entre la valeur de Nusselt calculé par la technique neuronale et la valeur empirique calculé par le modèle numérique.

L'erreur relative absolue moyenne dans ce cas est de l'ordre de 0.8416%

Les figures III.4 à III.9 montrent les variations du nombre de Nusselt par rapport à celui de Prandtl. Elles visualisent une comparaison entre les données numériques obtenues à partir des corrélations empiriques et les résultats de prédiction du réseau de neurone artificiel optimisé.

Les figures montrent un excellent accord entre les données analytiques de littérature (indiqué par des cercles vides) et les résultats de prédiction de modèle neuronale (indiqué par des cercles pleins).

III.9. Analyse de la sensibilité des entrées par rapport à la sortie

Le RNSO a fourni une matrice de poids de connexion, les poids sont des coefficients relatifs aux neurones artificiels, ils sont analogues aux forces synapses entre les axones et les dendrites dans de vrais neurones biologiques. La matrice des poids issue du RNSO peut être utilisée pour analyser la sensibilité des entrées par rapport à la sortie désirée. Garson (Garson. 1991) a proposé une équation basée sur le partitionnement de poids de connexion:

$$I_j = \frac{\sum_{m=1}^{m=N_h} \left(\left(\frac{\left|W_{jm}^{ih}\right|}{\sum_{k=1}^{N_i} \left|W_{km}^{ih}\right|} \right) \times \left|W_{mn}^{h0}\right| \right)}{\sum_{k=1}^{k=N_i} \left\{ \sum_{m=1}^{m=N_h} \left(\frac{\left|W_{km}^{ih}\right|}{\sum_{k=1}^{N_i} \left|W_{km}^{ih}\right|} \right) \times \left|W_{mn}^{h0}\right| \right\}} \qquad \text{(III.12)}$$

Où Ij est sensibilité relative de la j-ième variable d'entrée sur la variable de sortie; Ni et Nh représentent le nombre de neurones d'entrée et cachées, respectivement; Ws sont des poids de connexion, l'exposant i, h et o réfèrent à la couche d'entrée, cachée et de sortie respectivement, et les indices k, m, n et réfèrent à la couche d'entrée, cachée et de sortie respectivement.

La sensibilité relative des différentes variables calculées par l'équation III.12 est représentée sur la figure III.10. Comme on peut le constater, toutes les variables ont des effets importants sur le nombre de Nusselt. Par conséquent, il peut être observé que la contribution la plus élevée a été obtenue avec Re (35%), en suite Pr a une contribution de 19 % et e, b et b'ont presque la même influence sur les valeurs de Nusselt, tandis que le a et He ont moins d'influence sur les valeurs de Nusselt. Chose qui confirme l'importance des toutes les entrées sur la sortie de RNAO.

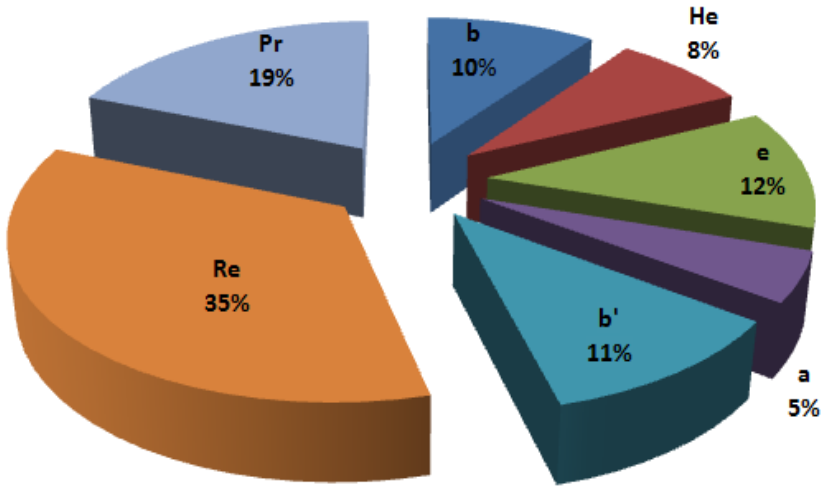


Fig. III.10: Importance relative des variables d'entrées sur la prédiction du Nu.

III.10 Conclusion

Dans ce chapitre on a présenté les résultats de réseau de neurone artificiel du nombre de Nusselt qui caractérise les échanges thermiques intensifs par comparaison avec les résultats analytiques de littérature par modélisation numérique en utilisant un programme Matlab élaborer dans le cadre de ce travail. Nous avons également décrit la méthode d'évaluation de la performance du RNA optimisé sous forme de calcul de vecteur du meilleur ajustement linéaire entre les données expérimentales de la solubilité et celles calculées par le RNA optimisé, et sous forme de calcul d'erreur des valeur calculées par rapport aux valeurs numériques par le réseau de neurones statique optimisé (RNSO).

De ce fait la modélisation neuronale pour Nu s'avère une alternative, fiable, facile. Les conclusions relatives à la stratégie à adopter pour arriver au modèle neuronal avec une capacité de généralisation acceptable et une performance (en termes d'erreur) fixée à priori, sont comme suit :

- Cette étude confirme les résultats de travaux antérieurs indiquant que les règles heuristiques, pour le choix du nombre de couches cachées et du nombre de neurones par couche cachées, n'ont aucune crédibilité, du moins pour les réseaux de type feedforward. La règle la plus raisonnable est la règle d'essai et erreur en allant du réseau à structure simple (une seule couche cachée et un nombre faible de neurones) vers des structures à plusieurs couches cachées, en fixant comme objectif une bonne capacité de généralisation et une performance fixée.

- Pour atténuer le problème de convergence vers les minimums locaux, il est conseillé de réaliser l'apprentissage du RN ayant une structure donnée avec plusieurs initialisations différentes des matrices des poids et biais synaptiques.
- Le RNSO dont la conception de la topologie est optimisée en terme:
    - 07 neurones dans la couche d'entrée;
    - 36 neurones dans la couche cachée avec la même fonction d'activation (Tansig);
    - 01 neurone de sortie avec la même fonction d'activation (Purline);
    - 01 algorithme d'apprentissage (Levenberg-Marquardt).
    - 01 nombre d'itération de 1000;
- Le RNSO ainsi obtenu présente une capacité d'estimation intéressante pour Nu avec erreur relative absolue moyenne (ERAM) égale à 0.84% et un coefficient de corrélation R2> 0.999.
- Une étude de sensibilité des entrées par rapport à la sortie a été faite à partir des poids synaptiques du RNSO élaboré.
- Cette comparaison montre que le modèle neuronale obtenu a une capacité de généralisation acceptable et une performance (en terme d'erreur) fixé à priori de se faite le réseau de neurone artificiel s'avère une alternative fiable, facile et complémentaire à la méthode conventionnelle.

Conclusion Générale

Conclusion générale

Ce mémoire s'inscrit dans une démarche novatrice en proposant d'appliquer les techniques neuronales qui se sont imposées par leurs simplicités relatives et leurs efficacités irréfutables. En effet l'étude qu'on a effectuée sur l'intensification des échanges thermiques par une technique basée sur les réseaux de neurones multicouches, a amplement montrée ses performances. Les différents tests effectués sur les RNA ont permis de mettre en évidence une très bonne concordance des résultats obtenus à ceux analytiques de littérature obtenus par modélisation numérique sous MATLAB. Les erreurs relatives commises sont très acceptables. Des études comparatives effectuées entre ces derniers et les modèles proposés dans la littérature ont également apporté des preuves tangibles quant à leurs performances.

Les conclusions relatives à la stratégie à adopter pour arriver au modèle neuronal avec une capacité de généralisation acceptable et une performance (en termes d'erreur) fixée à priori, sont comme suit :

- Cette étude confirme les résultats de travaux antérieurs indiquant que les règles heuristiques, pour le choix du nombre de couches cachées et du nombre de neurones par couche cachées, n'ont aucune crédibilité. La règle la plus raisonnable est la règle d'essai et erreur en allant du réseau à structure simple (une seule couche cachée et un nombre faible de neurones) vers des structures à plusieurs couches cachées, en fixant comme objectif une bonne capacité de généralisation et une performance fixée.
- On confirme aussi les résultats de la littérature qui suggèrent que l'algorithme d'apprentissage utilisant la régularisation Bayésienne a la tendance à donner des RNA ayant une meilleure capacité de généralisation que les algorithmes de Levenberg-Marquardt, du gradient conjugués, et de quasi-Newton.
- Pour éviter le surapprentissage et réduire le temps de modélisation neuronale, il est vivement conseillé d'utiliser l'apprentissage atténué avec les algorithmes de Levenberg-Marquardt, gradient conjugués, quasi-Newton.
- Pour atténuer le problème de convergence vers les minimums locaux, il est conseillé de réaliser l'apprentissage du RNA ayant une structure donnée avec plusieurs initialisations différentes des matrices des poids synaptiques.

Enfin, du moins à partir des points énumérés ci-dessus, il est facile de constater pourquoi la modélisation neuronale, pour certains, relève plus de l'art que de la science.

En perspective, il serait intéressant d'envisager la validation de la technique neuronale par des résultats expérimentaux ce qui suppose de toute évidence l'acquisition d'une base de données et d'envisager ainsi l'application de cette méthode pour d'autres problèmes de transfert thermique tels que la conduction et le rayonnement, etc.

Références Bibliographiques

Références bibliographiques

1. **A.** BOUZIDI, S. HANINI, F. SOUAHI, B. MOHAMMEDI et M. TOUIZA, Viscosity Calculation at Moderate Pressure for Nonpolar Gases Via Neural Network, Journal of Applied Sciences 7(17) 2450-2455, 2007.
2. **A.** BOUZIDI, Méthode des réseaux de neurones pour le calcul des coefficients pertinents des phénomènes de transport : Viscosité, Conductivité et Diffusion Moléculaire des Gaz, Mémoire de Thèse de Magister, Centre Universitaire de Médéa, 2006.
3. **A.**I. GALUSHKIN, Neural networks theory, Springer-Verlag Berlin Heidelberg, 2007.
4. **A.**K. JAIN et J. MAO, Artificial neural networks : A tutorial, Cybernetics IEEE, 1996.
5. **A.** DUPRAT, T. HUYNH et G. DREYFUS, Towards a Principled Methodology for Neural Network Design and Performance Evaluatioon in QSAR ; Application to the Prediction of LogP, Journal of Chemical Information an Computer Sciences, 38, 586-594, 1998.
6. **B.** KROSE et P. VAN DER SMAGT, An introduction to neural network, The University of Amsterdam, 1996.
7. **B.** WIDROW et M.E. HOFF, Adaptive switching circuits, 1960 IRE Western Electric Show and Convention Record, part 4, p. 96-104, 1960.
8. **C.** SI-MOUSSA S. HANINI, R. DERRICHE, M. BOUHEDDA et A. BOUZIDI, Prediction of high-pressure vapor liquid equilibrium of six binary systems, carbon dioxide with six esters, using an artificial neural network model, Brazilian Journal of Chemical Engineering, Vol. 25, No. 01, pp. 183 - 199, 2008.
9. **D.** HASSANI, Méthodes des Réseaux de Neurones pour le Calcul de la Dispersion Axiale dans les Lits Fixes, Mémoire de Thèse de Magister, Centre Universitaire de Médéa, 2003.
10. **D.**O. HEBB, The organization of behavior, a neuropsychological theory, New York: John Wiley, 1949.
11. **D.**W. MARQUARDT, an Algorithm for Least-Squares Estimation of Non-linear Parameters, Journal of the Society of Industrial and Applied Mathematics 11, 431-441, 1963.
12. **E.** DAVALO et P. NAÏM, Réseaux de neurones, Eyrolles, 2ème Edition, Mars, 1993.
13. **F.** ELIE, Conception et Réalisation d'un Système Utilisant des Réseaux de Neurones pour l'Identification et la Caractérisation, à Bord de Satellites, de Signaux Transitoires de type Sifflement", Thèse de Doctorat, Université d'Orléans, 1997.
14. **G.** DREYFUS et al., Réseaux de Neurones Méthodologie et Applications, Editions Eyrolles, Paris, 2002.
15. **G.** DREYFUS et al., Apprentissage statistique, Editions Eyrolles, Paris, 2008.
16. **G.** THEZE, Réalisation et évaluation d'un réseau de neurones comme outil d'aide à l'analyse du pilotage d'un cœur de réacteur nucléaire, Université René Descartes Paris 5/CEA, France, 2004.
17. **G.** HINTON, Connectionnist Learning Procedures, Artificial Intelligence, 1989.
18. **H.** DEMUTH, M. BEALE et M. HAGAN, Matlab, Neural Network Toolbox 6, User's Guide, MathWorks, Inc., 2009.
19. **H.** WEI et al., Applications of genetic neural network for prediction of critical heat flux, International Journal of Thermal Sciences, Vol. 49, 143–152, 2010.
20. **H.** ABDI, Les réseaux de Neurones. Sciences et Technologies de la Connaissance, PUG, 1994.
21. **I.** VINCENT, Eléments de mathématiques appliquées, Sciences.ch, 2005.
22. **I.** RIVALS, Les réseaux de neurones formels pour le pilotage de robots mobiles FLUX, revue de l'Association amicale Les Ingénieurs SUPÉLEC, N° 178, 1996.
23. **J.** SARANGAPANI, Neural Network Control of Nonlinear Discrete-Time Systems, Taylor & Francis Group, LLC, 2006.
24. **J.**A. FREEMAN et D.M. SKAPURA, Neural Networks Algorithms, Applications and Programming Techniques, Addison-Wesley Publishing Company, New York, 1992.
25. **K.** LEVENBERG, A Method for the Solution of Certain Non-linear Problems in Least Squares, Quartely Journal of Applied Mathematics II, 164-168, 1944.

26. **L.** ALLEN et C.M. BlSHOP, Neural network approach to energy confinement scaling in TOKAMAKS, Plasma Physics and Controlled Fusion, Vol.34, N° 7, pp. 1291-1302,1992.
27. **M.**J. LEE, S.M. HWANG et J.T. CHEN, Density and Viscosity Calculations for Polar Solutions via Neural Networks, Journal of Chemical Engineering of Japan, 27, 6, 1994.
28. **M.** PARIZEAU, Réseaux de neurones, Université LAVAL, 2004.
29. **M.**T. HAGAN et H.B. DEMUTH, Neural network design, PWS publishing company, USA, 1996.
30. **P.**D. McNELIS, Neural Networks in Finance: Gaining Predictive Edge in the Market, Elsevier Academic Press, 2005.
31. **R.** ZEMOURI, D. RACOCEANU, et N. ZERHOUNI, Application des Réseaux de Neurones Récurrents (RBF) à la Détection Dynamique de Dégradations, Les Journées de travail du GRP GRENOBLE, LAG – ENSIEG, 2002.
32. **R.**L. HARVEY, Neural network principles, Printice Hall International INC, 1994.
33. **S.** LOLAS et O.A. OLATUNBOSUN, Prediction of vehicle reliability performance using artificial neural networks, science direct: Expert Systems with Applications, 34 p. 2360–2369, 2008.
34. **S.** GROSBERG, Classical and instrumental learning by neural networks, in progress theoretical biology, New York academic press, Vol.3, p.51-141, 1977.
35. **T.**L. FINE, Feedforward neural network methodology, Statistics for engineering and information science, Springer-verlag New York, 1999.
36. **W.** GERSTNER et W.M. KISTLER, Spiking Neuron Models Single Neurons, Populations, Plasticity, Cambridge University Press, 2002.
37. **W.**S. McCULLOCH et W.H. PITTS, A logical calculus of the ideas imminent in nervous activity, Bulletin of Mathematical Biophysics, 5, p.115-133, 1943.
38. **Y.** LEE, S.H. OH et M.W. KIM, The Effect of Initial Weights on Premature Saturation in Back-Propagation Learning, International Joint Conference on Neural Networks, 1, 765-770, 1991.

Annexe

Annexe A : Programme de calcul base de données

```
clear all,clc
%**************************************
%       equation 14.51 Feuil 1         *
%**************************************
% Retr (nombre de Reynolds transitoir)
%pour les écoulements turbulents
format short g
a=0:0.05:0.1;  % a = d/2*R (10<d/2*R< infini)
for i=1:length(a)
Retr(i)=2100*(1+12*sqrt(a(i)));
end
Retr
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all,clc
format short g
%*************************************************
%       equation 14.53a et 14.53b Feuil 1         *
%*************************************************
Retr= input('Retr=')
 % pour nombre de PRANDTEL = 1
Re=Retr:10000:1e5;
b=6.7:67.86:346; % b=2*R/d
Pr=1;
for i=1:length(Re)
    for j=1:length(b)
Nu1(i,j)=Pr*Re(i)^(4/5)*(1/b(j))^(1/10)*(1+0.098*(Re(i)/(1/b(j))^2)^(1/5))/(26
.2*((Pr^(2/3))-0.074));
end
end
Nu1
% pour nombre de PRANDTEL > 1
Re=Retr:10000:1e5;
b=6.7:67.86:346; % b=2*R/d
Pr=3;
for i=1:length(Re)
    for j=1:length(b)
Nu2(i,j)=(Pr^(0.4))*Re(i)^(5/6)*(1/b(j))^(1/12)*(1+0.061*(Re(i)/(1/b(j))^2.5)^
(1/6))/41;
end
end
Nu2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all,clc
format short g
%*************************************************
%       equation 14.47 et 14.48 Feuil 2 et 3      *
%*************************************************
He=21:1:40;
Pr=0.7:3.25:38;
%Temperature uniforme (TMU)1447
for i=1:length(He)
    for j=1:length(Pr)

Nu1(i,j)=((3.657+4.343/(1+(957/Pr(j)*(He(i)^2)))^2)^3+1.158*(He(i)/(1+0.477/Pr
(j)))^(3/2))^(1.3);
  end
```

```
end
Nu1
%Flux de chaleur uniforme (UHF) 1448
for i=1:length(He)
    for j=1:length(Pr)

Nu2(i,j)=((4.364+4.636/(1+(1342/Pr(j)*(He(i)^2)))^2)^3+1.816*(He(i)/(1+1.15/Pr
(j)))^(3/2))^(1/3);
  end
end
Nu2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc
format short g
%***********************************************
%      equation de stephan TUW Feuil 4          *
%***********************************************
% regime laminaire (Re<2300)
%1****Temperature constante***
%e=D/L%%%%   Re*Pr*e=33
Re=1700:100:2000
Pr=0.8:1.4:6
e=33./(Re.*Pr);
for i=1:length(Re)
    for j=1:length(Pr)
        for k=1:length(e)
Nu(i,j,k)=3.657+(0.0677*(Re(i)*Pr(j)*e(k))^1.33)/(1+0.1*Pr(j)*(Re(i)*e(k))^0.3
);
    end
    end
end
Nu
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc
format short g
%***********************************************
%      equation de stephan FUH Feuil 5          *
%***********************************************
% regime laminaire (Re<2300)
%1****Temperature constante***
%e=D/L%%%%   Re*Pr*e=33
Re=1700:100:2000
Pr=0.8:1.4:6
e=33./(Re.*Pr);
for i=1:length(Re)
    for j=1:length(Pr)
        for k=1:length(e)
Nu(i,j,k)=4.364+(0.086*(Re(i)*Pr(j)*e(k))^1.33)/(1+0.1*Pr(j)*(Re(i)*e(k))^0.83
);
    end
    end
end
Nu
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all; clc
format short g
%************************************************************
%      equation 14.13 et 14.12 et 14.10 FUH Feuil 6          *
%************************************************************
%%%%%Regime turbulent%%%%%%%
Re=3800:400:5000;
```

```
Pr=0.7:6:38;
for i=1:length(Re)
f0(i)=(1.58*(log(Re(i)))-3.28)^(-2);
end
m=f0/2
for i=1:length(Re)
    for j=1:length(Pr)
        for k=1:length(m)
Nu0(i,j,k)=(Re(i)*Pr(j)*m(k))/(1+12.7*sqrt(m(k))*(((Pr(j))^(2/3))-1));
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%a=e/d
%%%%%%%%%%%%%%%%%%%%%%%%%b=p/e
a=0.01:0.01:0.04
b=10:2:15
alf=60
for i=1:length(Re)
    for j=1:length(Pr)
        for k=1:length(a)
         for n=1:length(b)
Nu(i,j,k,n)=Nu0(i,j,k)*((1+(2.64*((Re(i))^0.036)*((a(k))^0.002)*((b(n))^(-
0.21))*((alf/90)^(0.29))*((Pr(j))^(-0.024)))^7)^(1/7));
         end
        end
    end
end
Nu
```

Annexe B : Base des données

Tableau B.1 : Valeurs du nombre de *Nusselt* en fonction de *He* et *Pr* données par l'équation (III.1) valable pour une condition aux limites de type Dirichlet

Pr	He	Nu
0.7	21	4.9838
	22	5.0229
	23	5.0623
	24	5.102
	25	5.1419
	26	5.1819
	27	5.2221
	28	5.2624
	29	5.3029
	30	5.3434
	31	5.3839
	32	5.4246
	33	5.4652
	34	5.5058
	35	5.5465
	36	5.5871
	37	5.6277
	38	5.6683
	39	5.7088
	40	5.7493

Pr	He	Nu
3.95	21	5.8697
	22	5.9518
	23	6.0336
	24	6.1149
	25	6.1957
	26	6.2761
	27	6.356
	28	6.4353
	29	6.5141
	30	6.5924
	31	6.6701
	32	6.7473
	33	6.8239
	34	6.9
	35	6.9755
	36	7.0505
	37	7.1249
	38	7.1988
	39	7.2722
	40	7.345

Pr	He	Nu
7.2	21	6.0645
	22	6.1548
	23	6.2445
	24	6.3336
	25	6.4221
	26	6.5099
	27	6.597
	28	6.6835
	29	6.7693
	30	6.8544
	31	6.9388
	32	7.0225
	33	7.1056
	34	7.188
	35	7.2697
	36	7.3508
	37	7.4312
	38	7.5109
	39	7.5901
	40	7.6686

Pr	He	Nu
10.45	21	6.1494
	22	6.2431
	23	6.3362
	24	6.4286
	25	6.5203
	26	6.6112
	27	6.7014
	28	6.7909
	29	6.8796
	30	6.9676
	31	7.0548
	32	7.1413
	33	7.227
	34	7.3121
	35	7.3964
	36	7.48
	37	7.5629
	38	7.6451
	39	7.7267
	40	7.8075

Pr	He	Nu
13.7	21	6.1969
	22	6.2926
	23	6.3875
	24	6.4817
	25	6.5751
	26	6.6678
	27	6.7597
	28	6.8508
	29	6.9411
	30	7.0307
	31	7.1194
	32	7.2074
	33	7.2947
	34	7.3812
	35	7.4669
	36	7.5519
	37	7.6362
	38	7.7198
	39	7.8027
	40	7.8848

Pr	He	Nu
16.95	21	6.2272
	22	6.3241
	23	6.4202
	24	6.5156
	25	6.6101
	26	6.7039
	27	6.7969
	28	6.889
	29	6.9804
	30	7.0709
	31	7.1606
	32	7.2496
	33	7.3378
	34	7.4252
	35	7.5118
	36	7.5977
	37	7.6829
	38	7.7673
	39	7.851
	40	7.9341

Pr	He	Nu
20.2	21	6.2483
	22	6.346
	23	6.443
	24	6.5391
	25	6.6344
	26	6.7289
	27	6.8226
	28	6.9155
	29	7.0076
	30	7.0988
	31	7.1892
	32	7.2788
	33	7.3677
	34	7.4557
	35	7.543
	36	7.6295
	37	7.7152
	38	7.8003
	39	7.8846
	40	7.9682

Pr	He	Nu
23.45	21	6.2638
	22	6.3621
	23	6.4596
	24	6.5564
	25	6.6523
	26	6.7473
	27	6.8416
	28	6.935
	29	7.0275
	30	7.1193
	31	7.2102
	32	7.3003
	33	7.3896
	34	7.4781
	35	7.5658
	36	7.6528
	37	7.739
	38	7.8244
	39	7.9092
	40	7.9932

Pr	He	Nu
26.7	21	6.2757
	22	6.3744
	23	6.4724
	24	6.5696
	25	6.6659
	26	6.7614
	27	6.856
	28	6.9498
	29	7.0428
	30	7.1349
	31	7.2262
	32	7.3167
	33	7.4064
	34	7.4952
	35	7.5833
	36	7.6706
	37	7.7571
	38	7.8429
	39	7.928
	40	8.0123

Pr	He	Nu
29.95	21	6.285
	22	6.3842
	23	6.4825
	24	6.58
	25	6.6767
	26	6.7725
	27	6.8675
	28	6.9616
	29	7.0549
	30	7.1473
	31	7.2389
	32	7.3297
	33	7.4196
	34	7.5087
	35	7.5971
	36	7.6847
	37	7.7715
	38	7.8575
	39	7.9428
	40	8.0274

Pr	He	Nu
33.2	21	6.2926
	22	6.3921
	23	6.4907
	24	6.5885
	25	6.6854
	26	6.7815
	27	6.8768
	28	6.9711
	29	7.0647
	30	7.1573
	31	7.2492
	32	7.3402
	33	7.4303
	34	7.5197
	35	7.6083
	36	7.6961
	37	7.7831
	38	7.8693
	39	7.9549
	40	8.0396

Pr	He	Nu
36.45	21	6.2989
	22	6.3986
	23	6.4975
	24	6.5955
	25	6.6927
	26	6.789
	27	6.8844
	28	6.979
	29	7.0727
	30	7.1656
	31	7.2576
	32	7.3488
	33	7.4392
	34	7.5288
	35	7.6175
	36	7.7055
	37	7.7927
	38	7.8791
	39	7.9648
	40	8.0497

Tableau B.2 : Valeurs du nombre de *Nusselt* en fonction de *He* et *Pr* données par l'équation (III.2) valable pour la condition aux limites de type Newman

Pr	He	Nu
0.7	21	398.21
	22	417.43
	23	437.31
	24	457.84
	25	479.03
	26	500.87
	27	523.37
	28	546.51
	29	570.31
	30	594.76
	31	619.86
	32	645.61
	33	672.02
	34	699.07
	35	726.77
	36	755.12
	37	784.13
	38	813.78
	39	844.08
	40	875.03

Pr	He	Nu
3.95	21	632.8
	22	672.17
	23	713
	24	755.3
	25	799.06
	26	844.28
	27	890.97
	28	939.11
	29	988.71
	30	1039.8
	31	1092.3
	32	1146.3
	33	1201.7
	34	1258.6
	35	1316.9
	36	1376.7
	37	1438
	38	1500.7
	39	1564.9
	40	1630.5

Pr	He	Nu
7.2	21	675.12
	22	718.2
	23	762.89
	24	809.2
	25	857.12
	26	906.65
	27	957.8
	28	1010.6
	29	1064.9
	30	1120.9
	31	1178.5
	32	1237.7
	33	1298.5
	34	1360.9
	35	1425
	36	1490.6
	37	1557.9
	38	1626.7
	39	1697.2
	40	1769.2

Pr	He	Nu
10.45	21	692.75
	22	737.37
	23	783.68
	24	831.66
	25	881.32
	26	932.65
	27	985.67
	28	1040.4
	29	1096.7
	30	1154.8
	31	1214.5
	32	1275.9
	33	1338.9
	34	1403.7
	35	1470.1
	36	1538.1
	37	1607.9
	38	1679.3
	39	1752.4
	40	1827.1

Pr	He	Nu
13.7	21	702.41
	22	747.88
	23	795.07
	24	843.97
	25	894.58
	26	946.91
	27	1001
	28	1056.7
	29	1114.2
	30	1173.3
	31	1234.2
	32	1296.8
	33	1361.1
	34	1427.1
	35	1494.8
	36	1564.2
	37	1635.3
	38	1708.2
	39	1782.7
	40	1858.9

Pr	He	Nu
16.95	21	708.51
	22	754.52
	23	802.26
	24	851.75
	25	902.96
	26	955.91
	27	1010.6
	28	1067
	29	1125.2
	30	1185.1
	31	1246.7
	32	1310
	33	1375.1
	34	1441.9
	35	1510.4
	36	1580.7
	37	1652.7
	38	1726.4
	39	1801.8
	40	1879

Pr	He	Nu
20.2	21	712.71
	22	759.09
	23	807.22
	24	857.1
	25	908.73
	26	962.12
	27	1017.2
	28	1074.1
	29	1132.8
	30	1193.1
	31	1255.3
	32	1319.1
	33	1384.7
	34	1452.1
	35	1521.2
	36	1592
	37	1664.6
	38	1739
	39	1815
	40	1892.8

Pr	He	Nu
23.45	21	715.78
	22	762.43
	23	810.84
	24	861.01
	25	912.95
	26	966.65
	27	1022.1
	28	1079.3
	29	1138.3
	30	1199
	31	1261.5
	32	1325.8
	33	1391.8
	34	1459.5
	35	1529.1
	36	1600.3
	37	1673.4
	38	1748.1
	39	1824.7
	40	1902.9

Pr	He	Nu
26.7	21	718.12
	22	764.97
	23	813.6
	24	864
	25	916.16
	26	970.1
	27	1025.8
	28	1083.3
	29	1142.5
	30	1203.5
	31	1266.3
	32	1330.9
	33	1397.2
	34	1465.2
	35	1535.1
	36	1606.7
	37	1680
	38	1755.1
	39	1832
	40	1910.6

Pr	He	Nu
29.95	21	719.96
	22	766.98
	23	815.77
	24	866.35
	25	918.7
	26	972.82
	27	1028.7
	28	1086.4
	29	1145.9
	30	1207.1
	31	1270.1
	32	1334.9
	33	1401.4
	34	1469.7
	35	1539.8
	36	1611.6
	37	1685.3
	38	1760.6
	39	1837.8
	40	1916.7

Pr	He	Nu
33.2	21	721.45
	22	768.6
	23	817.53
	24	868.25
	25	920.74
	26	975.02
	27	1031.1
	28	1088.9
	29	1148.5
	30	1210
	31	1273.1
	32	1338.1
	33	1404.8
	34	1473.3
	35	1543.6
	36	1615.7
	37	1689.5
	38	1765.1
	39	1842.5
	40	1921.6

Pr	He	Nu
36.45	21	722.68
	22	769.94
	23	818.98
	24	869.81
	25	922.43
	26	976.84
	27	1033
	28	1091
	29	1150.8
	30	1212.3
	31	1275.6
	32	1340.8
	33	1407.6
	34	1476.3
	35	1546.8
	36	1619
	37	1693
	38	1768.8
	39	1846.3
	40	1925.7

Tableau B.3 : Valeurs du nombre de *Nusselt* en fonction de *Re*, Re_{tr}, *b* et *Pr* données par les équations (III.3a) pour *Pr=3* et (III.3b) et pour *Pr=3*

Pr	Re_{tr}	Re	b	Nu	Pr	Re_{tr}	Re	b	Nu
1	2100	2100	6.7	30.506	3	2100	2100	6.7	28.0963
			74.56	43.102				74.56	35.9145
			142.42	48.965				142.42	40.0136
			210.28	53.185				210.28	43.0561
			278.14	56.577				278.14	45.5457
			346	59.453				346	47.6842
		12100	6.7	149.39			12100	6.7	134.2399
			74.56	227.62				74.56	184.3175
			142.42	262.7				142.42	209.1236
			210.28	287.76				210.28	227.3387
			278.14	307.81				278.14	242.1614
			346	324.78				346	254.8489
		22100	6.7	259.81			22100	6.7	230.9505
			74.56	405.48				74.56	325.0036
			142.42	470.2				142.42	370.9217
			210.28	516.33				210.28	404.5432
			278.14	553.22				278.14	431.8621
			346	584.4				346	455.2231
		32100	6.7	366.73			32100	6.7	323.6469
			74.56	580.59				74.56	462.4056
			142.42	675.11				142.42	529.6083
			210.28	742.42				210.28	578.7355
			278.14	796.21				278.14	618.6200
			346	841.65				346	652.7076
		42100	6.7	471.47			42100	6.7	413.8112
			74.56	753.99				74.56	597.7362
			142.42	878.43				142.42	686.3322
			210.28	966.97				210.28	751.0277
			278.14	1037.7				278.14	803.5216
			346	1097.4				346	848.3694
		52100	6.7	574.65			52100	6.7	502.15
			74.56	926.16				74.56	731.57
			142.42	1080.6				142.42	841.64
			210.28	1190.4				210.28	921.95
			278.14	1278.1				278.14	987.08
			346	1352.2				346	1042.7
		62100	6.7	676.63			62100	6.7	589.06
			74.56	1097.4				74.56	864.24
			142.42	1281.9				142.42	995.84
			210.28	1413				210.28	1091.8

			278.14	1517.7
			346	1606.1
		72100	6.7	777.63
			74.56	1267.8
			142.42	1482.4
			210.28	1634.9
			278.14	1756.6
			346	1859.4
		82100	6.7	877.83
			74.56	1437.7
			142.42	1682.4
			210.28	1856.2
			278.14	1995
			346	2112.1
		92100	6.7	977.33
			74.56	1607
			142.42	1881.9
			210.28	2077.1
			278.14	2232.9
			346	2364.4
1	7734.9	7734.9	6.7	99.262
			74.56	148.46
			142.42	170.7
			210.28	186.61
			278.14	199.36
			346	210.15
		17735	6.7	212.17
			74.56	328.28
			142.42	380.03
			210.28	416.95
			278.14	446.48
			346	471.45
		27735	6.7	320.38
			74.56	504.4
			142.42	585.9
			210.28	643.96
			278.14	690.36
			346	729.58
		37735	6.7	425.97
			74.56	678.47
			142.42	789.84
			210.28	869.11
			278.14	932.44
			346	985.94
			6.7	529.77

			278.14	1169.6
			346	1236
		72100	6.7	674.82
			74.56	995.97
			142.42	1149.2
			210.28	1260.8
			278.14	1351.3
			346	1428.5
		82100	6.7	759.62
			74.56	1126.9
			142.42	1301.7
			210.28	1429.1
			278.14	1532.2
			346	1620.3
		92100	6.7	843.58
			74.56	1257.2
			142.42	1453.6
			210.28	1596.7
			278.14	1712.6
			346	1811.5
3	7734.9	7734.9	6.7	89.85
			74.56	121.13
			142.42	136.81
			210.28	148.36
			278.14	157.76
			346	165.82
		17735	6.7	189.37
			74.56	264.11
			142.42	300.79
			210.28	327.67
			278.14	349.52
			346	368.22
		27735	6.7	283.55
			74.56	402.73
			142.42	460.63
			210.28	502.98
			278.14	537.37
			346	566.77
		37735	6.7	374.71
			74.56	538.87
			142.42	618.12
			210.28	676.01
			278.14	723
			346	763.15
			6.7	463.78

			74.56	851.13				74.56	673.31
			142.42	992.47				142.42	774
		47735	210.28	1093			47735	210.28	847.49
			278.14	1173.3				278.14	907.1
			346	1241.1				346	958.02
			6.7	632.24				6.7	551.28
			74.56	1022.7				74.56	806.46
			142.42	1194.1				142.42	928.65
		57735	210.28	1315.9			57735	210.28	1017.8
			278.14	1413.2				278.14	1090
			346	1495.4				346	1151.8
			6.7	733.65				6.7	637.52
			74.56	1193.5				74.56	938.58
			142.42	1395				142.42	1082.3
		67735	210.28	1538.1			67735	210.28	1187.1
			278.14	1652.4				278.14	1272.1
			346	1748.9				346	1344.6
			6.7	834.18				6.7	722.71
			74.56	1363.6				74.56	1069.8
			142.42	1595.2				142.42	1235.2
		77735	210.28	1759.7			77735	210.28	1355.7
			278.14	1891				278.14	1453.3
			346	2001.9				346	1536.7
			6.7	933.97				6.7	807.02
			74.56	1533.1				74.56	1200.4
			142.42	1794.8				142.42	1387.4
		87735	210.28	1980.7			87735	210.28	1523.6
			278.14	2129.1				278.14	1633.9
			346	2254.3				346	1728.1
			6.7	1033.1				6.7	890.56
			74.56	1702.1				74.56	1330.3
			142.42	1994				142.42	1539
		97735	210.28	2201.3			97735	210.28	1690.9
			278.14	2366.7				278.14	1814
			346	2506.3				346	1919
1			6.7	126.27	3			6.7	113.82
			74.56	190.95				74.56	155.11
			142.42	220.05				142.42	175.66
		10069	210.28	240.84			10069	210.28	190.76
	10069		278.14	257.49		10069		278.14	203.06
			346	271.58				346	213.59
			6.7	237.73				6.7	211.71
			74.56	369.63				74.56	296.76
			142.42	428.31				142.42	338.37
		20069	210.28	470.16			20069	210.28	368.85

			278.14	503.62				278.14	393.63
			346	531.9				346	414.81
			6.7	345.22				6.7	305.06
			74.56	545.18				74.56	434.69
			142.42	633.65				142.42	497.56
		30069	210.28	696.65			30069	210.28	543.53
			278.14	747				278.14	580.86
			346	789.54				346	612.77
			6.7	450.34				6.7	395.66
			74.56	718.88				74.56	570.39
			142.42	837.24				142.42	654.63
		40069	210.28	921.47			40069	210.28	716.16
			278.14	988.75				278.14	766.09
			346	1045.6				346	808.75
			6.7	553.8				6.7	484.33
			74.56	891.27				74.56	704.49
			142.42	1039.6				142.42	810.19
		50069	210.28	1145.1			50069	210.28	887.33
			278.14	1229.4				278.14	949.89
			346	1300.5				346	1003.3
			6.7	656				6.7	571.51
			74.56	1062.7				74.56	837.38
			142.42	1241.1				142.42	964.6
		60069	210.28	1367.9			60069	210.28	1057.4
			278.14	1469.1				278.14	1132.6
			346	1554.6				346	1196.8
			6.7	757.19				6.7	657.49
			74.56	1233.3				74.56	969.29
			142.42	1441.7				142.42	1118.1
		70069	210.28	1589.9			70069	210.28	1226.5
			278.14	1708.1				278.14	1314.4
			346	1808				346	1389.5
			6.7	857.54				6.7	742.47
			74.56	1403.2				74.56	1100.4
			142.42	1641.8				142.42	1270.8
		80069	210.28	1811.3			80069	210.28	1394.9
			278.14	1946.6				278.14	1495.5
			346	2060.8				346	1581.4
			6.7	957.17				6.7	826.59
			74.56	1572.6				74.56	1230.8
			142.42	1841.4				142.42	1422.8
		90069	210.28	2032.3			90069	210.28	1562.7
			278.14	2184.6				278.14	1676
			346	2313.2				346	1772.7

Tableau B.4 : Valeurs du nombre de *Nusselt* en fonction de *Re, Pr, e, p* et *d* données par l'équation (III.4) avec : $a = \left(\frac{e}{d}\right)$ et $b = \left(\frac{p}{d}\right)$

a	b	Re	Pr	Nu
	10	3I800	0.7000	33.9124
			6.7000	73.8376
			12.7000	90.7444
			18.7000	102.6558
			24.7000	112.1227
			30.7000	120.0959
			36.7000	127.0464
		4200	0.7000	37.6162
			6.7000	81.9006
			12.7000	100.6530
			18.7000	113.8646
			24.7000	124.3649
			30.7000	133.2084
			36.7000	140.9175
		4600	0.7000	41.3326
			6.7000	89.9910
			12.7000	110.5953
			18.7000	125.1116
			24.7000	136.6487
			30.7000	146.3654
			36.7000	154.8356
		5000	0.7000	45.0606
			6.7000	98.1066
			12.7000	120.5685
			18.7000	136.3934
			24.7000	148.9705
			30.7000	159.5632
			36.7000	168.7969
	12	3800	0.7000	32.6519
			6.7000	71.1065
			12.7000	87.3937
			18.7000	98.8696
			24.7000	107.9910
			30.7000	115.6735
			36.7000	122.3709
		4200	0.7000	36.2177
			6.7000	78.8701
			12.7000	96.9349
			18.7000	109.6632
			24.7000	119.7799
			30.7000	128.3008
			36.7000	135.7288

0.0100		4600	0.7000	39.7956
			6.7000	86.6600
			12.7000	106.5084
			18.7000	120.4933
			24.7000	131.6087
			30.7000	140.9707
			36.7000	149.1320
		5000	0.7000	43.3846
			6.7000	94.4741
			12.7000	116.1115
			18.7000	131.3568
			24.7000	143.4740
			30.7000	153.6797
			36.7000	162.5765
	14	3800	0.7000	31.6259
			6.7000	68.8858
			12.7000	84.6705
			18.7000	95.7933
			24.7000	104.6345
			30.7000	112.0816
			36.7000	118.5738
		4200	0.7000	35.0792
			6.7000	76.4058
			12.7000	93.9127
			18.7000	106.2490
			24.7000	116.0548
			30.7000	124.3142
			36.7000	131.5146
		4600	0.7000	38.5442
			6.7000	83.9511
			12.7000	103.1862
			18.7000	116.7400
			24.7000	127.5135
			30.7000	136.5880
			36.7000	144.4990
		5000	0.7000	42.0200
			6.7000	91.5198
			12.7000	112.4882
			18.7000	127.2633
			24.7000	139.0075
			30.7000	148.8995
			36.7000	157.5232

a	b	Re	Pr	Nu
0.0200	10	3800	0.7000	32.7562
			6.7000	72.4074
			12.7000	89.1333
			18.7000	100.9050
			24.7000	110.2559
			30.7000	118.1287
			36.7000	124.9900
		4200	0.7000	36.3337
			6.7000	80.3143
			12.7000	98.8660
			18.7000	111.9228
			24.7000	122.2943
			30.7000	131.0265
			36.7000	138.6366
		4600	0.7000	39.9235
			6.7000	88.2481
			12.7000	108.6319
			18.7000	122.9780
			24.7000	134.3737
			30.7000	143.9680
			36.7000	152.3296
		5000	0.7000	43.5244
			6.7000	96.2065
			12.7000	118.4280
			18.7000	134.0675
			24.7000	146.4905
			30.7000	156.9497
			36.7000	166.0650
	12	3800	0.7000	31.5386
			6.7000	69.7288
			12.7000	85.8415
			18.7000	97.1828
			24.7000	106.1922
			30.7000	113.7779
			36.7000	120.3892
		4200	0.7000	34.9828
			6.7000	77.3420
			12.7000	95.2133
			18.7000	107.7923
			24.7000	117.7849
			30.7000	126.1983
			36.7000	133.5310
		4600	0.7000	38.4387
			6.7000	84.9811
			12.7000	104.6169
			18.7000	118.4377

			24.7000	129.4167
			30.7000	138.6607
			36.7000	146.7172
		5000	0.7000	41.9054
			6.7000	92.6438
			12.7000	114.0495
			18.7000	129.1160
			24.7000	141.0845
			30.7000	151.1616
			36.7000	159.9441
	14	3800	0.7000	30.5474
			6.7000	67.5508
			12.7000	83.1661
			18.7000	94.1583
			24.7000	102.8909
			30.7000	110.2440
			36.7000	116.6528
		4200	0.7000	33.8830
			6.7000	74.9251
			12.7000	92.2442
			18.7000	104.4356
			24.7000	114.1210
			30.7000	122.2762
			36.7000	129.3840
		4600	0.7000	37.2299
			6.7000	82.3242
			12.7000	101.3530
			18.7000	114.7477
			24.7000	125.3889
			30.7000	134.3489
			36.7000	142.1581
		5000	0.7000	40.5872
			6.7000	89.7463
			12.7000	110.4898
			18.7000	125.0915
			24.7000	136.6914
			30.7000	146.4587
			36.7000	154.9714

a	b	Re	Pr	Nu
	10	3800	0.7000	31.7433
			6.7000	71.1172
			12.7000	87.6751
			18.7000	99.3182
			24.7000	108.5626
			30.7000	116.3433
			36.7000	123.1229
		4200	0.7000	35.2101
			6.7000	78.8832
			12.7000	97.2486
			18.7000	110.1627
			24.7000	120.4161
			30.7000	129.0462
			36.7000	136.5658
		4600	0.7000	38.6889
			6.7000	86.6756
			12.7000	106.8547
			18.7000	121.0441
			24.7000	132.3100
			30.7000	141.7922
			36.7000	150.0542
		5000	0.7000	42.1784
			6.7000	94.4923
			12.7000	116.4907
			18.7000	131.9593
			24.7000	144.2408
			30.7000	154.5777
			36.7000	163.5845
	12	3800	0.7000	30.5632
			6.7000	68.4861
			12.7000	84.4369
			18.7000	95.6542
			24.7000	104.5609
			30.7000	112.0578
			36.7000	118.5904
		4200	0.7000	33.9009
			6.7000	75.9637
			12.7000	93.6554
			18.7000	106.0968
			24.7000	115.9755
			30.7000	124.2905
			36.7000	131.5358
			0.7000	37.2499
			6.7000	83.4666
			12.7000	102.9051

0.0300		4600	18.7000	116.5748
			24.7000	127.4287
			30.7000	136.5645
			36.7000	144.5251
		5000	0.7000	40.6094
			6.7000	90.9928
			12.7000	112.1834
			18.7000	127.0852
			24.7000	138.9173
			30.7000	148.8765
			36.7000	157.5544
	14	3800	0.7000	29.6026
			6.7000	66.3467
			12.7000	81.8049
			18.7000	92.6768
			24.7000	101.3099
			30.7000	108.5768
			36.7000	114.9092
		4200	0.7000	32.8350
			6.7000	73.5895
			12.7000	90.7345
			18.7000	102.7925
			24.7000	112.3674
			30.7000	120.4271
			36.7000	127.4502
		4600	0.7000	36.0784
			6.7000	80.8568
			12.7000	99.6942
			18.7000	112.9424
			24.7000	123.4622
			30.7000	132.3173
			36.7000	140.0335
		5000	0.7000	39.3319
			6.7000	88.1466
			12.7000	108.6816
			18.7000	123.1235
			24.7000	134.5912
			30.7000	144.2440
			36.7000	152.6554

a	b	Re	Pr	Nu
	10	3800	0.7000	30.8512
			6.7000	69.9570
			12.7000	86.3614
			18.7000	97.8876
			24.7000	107.0353
			30.7000	114.7326
			36.7000	121.4382
		4200	0.7000	34.2207
			6.7000	77.5963
			12.7000	95.7915
			18.7000	108.5760
			24.7000	118.7221
			30.7000	127.2596
			36.7000	134.6971
		4600	0.7000	37.6016
			6.7000	85.2616
			12.7000	105.2537
			18.7000	119.3006
			24.7000	130.4487
			30.7000	139.8292
			36.7000	148.0011
		5000	0.7000	40.9931
			6.7000	92.9508
			12.7000	114.7453
			18.7000	130.0586
			24.7000	142.2116
			30.7000	152.4378
			36.7000	161.3462
	12	3800	0.7000	29.7043
			6.7000	67.3687
			12.7000	83.1715
			18.7000	94.2761
			24.7000	103.0896
			30.7000	110.5061
			36.7000	116.9673
		4200	0.7000	32.9482
			6.7000	74.7243
			12.7000	92.2519
			18.7000	104.5683
			24.7000	114.3436
			30.7000	122.5694
			36.7000	129.7356
			0.7000	36.2031
			6.7000	82.1048
			12.7000	101.3630

0.0400		4600	18.7000	114.8954
			24.7000	125.6357
			30.7000	134.6735
			36.7000	142.5471
		5000	0.7000	39.4681
			6.7000	89.5082
			12.7000	110.5023
			18.7000	125.2544
			24.7000	136.9627
			30.7000	146.8150
			36.7000	155.3982
	14	3800	0.7000	28.7706
			6.7000	65.2640
			12.7000	80.5788
			18.7000	91.3414
			24.7000	99.8840
			30.7000	107.0730
			36.7000	113.3361
		4200	0.7000	31.9122
			6.7000	72.3886
			12.7000	89.3745
			18.7000	101.3114
			24.7000	110.7860
			30.7000	118.7592
			36.7000	125.7056
		4600	0.7000	35.0644
			6.7000	79.5374
			12.7000	98.2000
			18.7000	111.3150
			24.7000	121.7247
			30.7000	130.4847
			36.7000	138.1166
		5000	0.7000	38.2265
			6.7000	86.7083
			12.7000	107.0527
			18.7000	121.3494
			24.7000	132.6971
			30.7000	142.2463
			36.7000	150.5658

Tableau B.5 : Valeurs du nombre de *Nusselt* en fonction de *Re, Pr, e, p* et *d* données par l'équation (III.5)

e	Re	Pr	Nu
0.0243	1700	0.8000	9.3498
		2.2000	19.9281
		3.6000	28.6034
		5.0000	35.7399
	1800	0.8000	9.7787
		2.2000	21.0921
		3.6000	30.3323
		5.0000	37.9160
	1900	0.8000	10.2138
		2.2000	22.2685
		3.6000	32.0758
		5.0000	40.1077
	2000	0.8000	10.6548
		2.2000	23.4567
		3.6000	33.8333
		5.0000	42.3143

e	Re	Pr	Nu
0.0083	1700	0.8000	5.1092
		2.2000	8.0706
		3.6000	10.6879
		5.0000	12.9388
	1800	0.8000	5.2199
		2.2000	8.3924
		3.6000	11.1854
		5.0000	13.5818
	1900	0.8000	5.3322
		2.2000	8.7180
		3.6000	11.6879
		5.0000	14.2301
	2000	0.8000	5.4462
		2.2000	9.0473
		3.6000	12.1951
		5.0000	14.8837

e	Re	Pr	Nu
0.0048	1700	0.8000	4.3754
		2.2000	5.9018
		3.6000	7.3002
		5.0000	8.5312
	1800	0.8000	4.4304
		2.2000	6.0669
		3.6000	7.5607
		5.0000	8.8725
	1900	0.8000	4.4862
		2.2000	6.2341
		3.6000	7.8239
		5.0000	9.2169
	2000	0.8000	4.5429
		2.2000	6.4033
		3.6000	8.0898
		5.0000	9.5643

e	Re	Pr	Nu
0.0033	1700	0.8000	4.0967
		2.2000	5.0556
		3.6000	5.9553
		5.0000	6.7602
	1800	0.8000	4.1304
		2.2000	5.1591
		3.6000	6.1208
		5.0000	6.9791
	1900	0.8000	4.1647
		2.2000	5.2639
		3.6000	6.2881
		5.0000	7.2001
	2000	0.8000	4.1995
		2.2000	5.3700
		3.6000	6.4572
		5.0000	7.4231

Tableau B.6 : Valeurs du nombre de *Nusselt* en fonction de *Re, Pr, e, p* et *d* données par l'équation (III.6)

e	Re	Pr	Nu
0.0243	1700	0.8000	7.6317
		2.2000	10.298
		3.6000	11.845
		5.0000	12.973
	1800	0.8000	7.784
		2.2000	10.519
		3.6000	12.102
		5.0000	13.257
	1900	0.8000	7.9329
		2.2000	10.734
		3.6000	12.352
		5.0000	13.533
	2000	0.8000	8.0786
		2.2000	10.943
		3.6000	12.595
		5.0000	13.801

e	Re	Pr	Nu
0.0083	1700	0.8000	5.625
		2.2000	7.1567
		3.6000	8.1415
		5.0000	8.8709
	1800	0.8000	5.6974
		2.2000	7.2829
		3.6000	8.2938
		5.0000	9.0409
	1900	0.8000	5.7691
		2.2000	7.4062
		3.6000	8.4421
		5.0000	9.2061
	2000	0.8000	5.8399
		2.2000	7.5267
		3.6000	8.5866
		5.0000	9.3669

e	Re	Pr	Nu
0.0048	1700	0.8000	5.0836
		2.2000	6.1463
		3.6000	6.8962
		5.0000	7.4699
	1800	0.8000	5.1288
		2.2000	6.2363
		3.6000	7.0096
		5.0000	7.5987
	1900	0.8000	5.1737
		2.2000	6.3247
		3.6000	7.1203
		5.0000	7.7241
	2000	0.8000	5.2185
		2.2000	6.4114
		3.6000	7.2285
		5.0000	7.8464

e	Re	Pr	Nu
0.0033	1700	0.8000	4.8386
		2.2000	5.6305
		3.6000	6.2323
		5.0000	6.7081
	1800	0.8000	4.8699
		2.2000	5.6994
		3.6000	6.3227
		5.0000	6.8128
	1900	0.8000	4.9012
		2.2000	5.7674
		3.6000	6.4112
		5.0000	6.9149
	2000	0.8000	4.9325
		2.2000	5.8343
		3.6000	6.4979
		5.0000	7.0147