

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département de Génie Electrique
Filière Electrotechnique
Laboratoire de Haute Tension

Projet de Fin d'Etudes
Pour l'obtention du diplôme
D'Ingénieur d'Etat en Electrotechnique

THEME

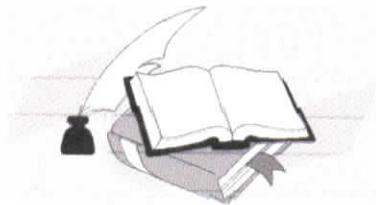
**UTILISATION DES RESEAUX DE NEURONES
ARTIFICIELS DANS LE VIEILLISSEMENT THERMIQUE
A 5000 HEURES DE L'HUILE DE TRANSFORMATEUR.**

Etudié par :

M^r : ALMI MOUSSA
M^r : GUEHRAR YUCEF

proposé et dirigé par :

M^r : A. BOUBAKEUR
M^r : A. MEKHALDI



E.N.P. 10, Avenue Hassen-Badi, El Harrach, ALGER
Promotion: Juin 2002

SUMMARY

This work presents the use of Artificial Neural Networks in the prediction of some dielectric properties of thermal ageing up to 5000 hours of the oil of transformer.

And for that, three calculation programmes based on the networks RBF was elaborated, first program RBF1 is only characterized by one input with training by Random Optimization Method ROM.

An entry also for second program RBF1 but in this case the training is done by Adaptive Centering Method ACM. Whereas last program RBF2 uses two inputs with ACM for the training.

Key words :

Artificial neural networks, thermal aging, learning, prediction.

ملخص

هذا العمل يتطرق إلى استعمال شبكة العصبونات الاصطناعية للتنبؤ ببعض خصائص الشيخوخة الحرارية لغاية 5000 ساعة لزيت المحول.

و من اجل هذا, استخدمنا ثلاثة برامج حسابية تتأسس على شبكات ال RBF, البرنامج الأول RBF1 يتميز بمدخل واحد و طريقة ROM في التعلم. و مدخل واحد أيضا من أجل البرنامج الثاني لكن هنا طريقة التعلم هي ACM. و برنامجا ثالثا RBF2 الذي يستعمل ACM كطريقة في التعلم.

الكلمات المفتاحية :

شبكة العصبونات الاصطناعية، الشيخوخة الحرارية، التعلم، التنبؤ.

RÉSUMÉ

Ce travail présent l'utilisation des Réseaux de Neurones Artificiels dans la prédiction de quelques propriétés diélectrique de vieillissement thermique jusqu'à 5000 heures d'huile de transformateur BORAC 22.

Et pour ça, on a élaboré trois programmes de calculé basé sur les réseaux RBF, le premier programme RBF1 se caractérise par une seul entré avec un apprentissage par la méthode d'optimisation aléatoire ROM.

Une entrée aussi pour le deuxième programme RBF1 mais dans ce cas l'apprentissage se fait par la méthode de centrage adaptatif ACM. Alors que le dernier programme RBF2 utilise deux entrées avec l'ACM pour la phase d'apprentissage.

Mots clés :

Réseaux de neurones artificiels, vieillissement thermique, apprentissage, prédiction.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

إهداء

أهدي هذا العمل المتواضع إلي

أمي أمي أمي

أبي

جدي الحاجة فميرة

خالتي العزيز إبراهيم

أخواتي عذبة و عبد الرحيم

أخواتي فاطمة، حورية، غنية، ليلى، زينب و أمينة

و إلي كل أصدقائي منهم موسى، رشيد، ياسين، عبد الحميد، محمد،

عبد العزيز، حسين، عبد المالك و خالد.

و إلي أعمز زملائي علي، عز الدين و أحمد إمين.

يوسف

إهداء

فالحمد لله أولا وأخيرا على تيسير لي في إتمام وإخراج هذا البحث وأشكرو على نعمه وأفضاله.

وبعد :

أهدي هذا البحث المتواضع إلى كل من الوالدين الكريمين

والى أخواني بشير و حسام

والى أختاي عقيلة و صباح

والى كل أفراد العائلة

والى كل الأصدقاء أخص بالذكر:

نذير، يوسف، عبد الحميد، ياسين، رشيد، جمال، حسين و خالد

والى كل زملائي في الدراسة منهم:

علي، أحمد، أمين، محمد و عز الدين

والى كل من ساهم في إتمام هذا البحث من قريب أو من بعيد.

موسى

REMERCIEMENTS

Nous remercions Dieu de nous avoir donné la force physique et morale pour accomplir ce travail.

Nous remercions nos encadrateurs M^{rs} : A.MEKHALDI et A.BOUBAKEUR, pour leurs aides, leurs encouragements, et leur patience avec laquelle ils ont bien voulu travaillé avec nous en nous guidant et nous corrigeant durant toute l'année.

Nous remercions également tous les membres du jury, pour avoir bien voulu évaluer notre travail.

Et finalement, nous adressons nos remerciements les plus profonds et les plus distingués à tous ceux qui ont contribué de proche ou de loin à ce modeste travail.

SOMMAIRE

INTRODUCTION GENRALE	1
CHAPITRE .I. LES RESEAUE NEURONES ARTIFICIEL	
I.1. INTRODUCTION.....	2
I.2. LES NEURONES.....	2
I.2.1. LE NEURONE BIOLOGIQUE.....	2
I.2.2. NEURONE ARTIFICIEL.....	3
I.2.2.1. DEFINITION	3
I.2.2.2. L'ETAT DES NEURONES.....	3
I.2.2.3. LES CONNEXIONS ENTRE NEURONES.....	4
I.2.2.3.1. LE VOISINAGE.....	4
I.2.2.3.2. LES CONNEXIONS.....	4
I.2.2.3.3. LES POIDS DES CONNEXIONS.....	4
I.2.2.4. DESCRIPTION D'UN NEURONE FORMEL.....	4
I.3. MODELE MATHEMATIQUE GENERAL DU NEURONE.....	5
I.3.1. FONCTION DISCRIMINANT (FONCTION DE BASE).....	5
I.3.1.1. FONCTION DE BASE LINEAIRE LBF	6
I.3.1.2. FONCTION DE BASE RADIALE RBF (RADIAL BASIS FUNCTION).....	6
I.3.1.3. FONCTION DE BASE ELLIPTIQUE EBF (ELLIPTIC BASIS FUNCTION).....	6
I.3.2. FONCTION DYNAMIQUE LINEAIRE.....	6
I.3.3. FONCTION NON DYNAMIQUE NON LINEAIRE	6
I.4. CLASSIFICATION D RESEAUX DE NEURONES.....	8
I.5.1. LES TYPES D'APPRENTISSAGE	8
I.6. RESEAUX DE NEURONES A APPRENTISSAGE NON-SUPERVISE	9
I.6.1. MEMOIRE ASSOCIATIVE.....	9
I.6.1.1. RESEAU DE HOPFIELD.....	10
I.6.1.2. MEMOIRES ASSOCIATIVES BIDIRECTIONNELLES (RAM)	11
I.6.1.3. MEMOIRES ASSOCIATIVES LINEAIRES OPTIMISEES (OLAM).....	11
I.6.2. RESEAUX COMPETITIFS.....	11
I.6.2.1. COUCHE DE GROSSBERG.....	11
I.6.2.2. ADAPTIVE RESONANCE THEORY ART.....	12
I.7. APPRENTISSAGE SUPERVISE	13
I.7.1. REGLES DU PERCEPTRON ET D'ADALINE, MADALINE	13
I.7.1.1. FONCTION DE COUT ET METHODE DE GRADIENT	15
I.7.1.2. LE GRADIENT STOCHASTIQUE : LA REGLE DE WIDROW-HOFF.....	16
I.7.2. RESEAUX MULTICOUCHES STATIQUES	17
I.7.3. RESEAUX A FONCTION DE BASE LINEAIRE LBF (LINEAR BASIS FUNCTION)	17
I.7.3.1. LA RETRO-PROPAGATION DU GRADIENT	18
I.7.3.2. L'ALGORITHME DE RETRO-PROPAGATION DU GRADIENT.....	18
I.7.3.2.1. PROPAGATION DES ETATS.....	18
I.7.3.2.2. RETRO-PROPAGATION DE L'ERREUR.....	19
I.7.4. UNE NOUVELLE ARCHITECTURE	22
I.7.4.1. LES UNITES.....	22
I.7.4.2. L'ORGANISATION DES CELLULES	22
I.7.5. METHODE D'OPTIMISATION ALEATOIRE ROM	23
I.7.6. RESEAUX A FONCTION DE BASE RADIALE RBF	24
I.7.6.1. ARCHITECTURE ET FONCTIONNEMENT DES RESEAUX RBF	24
I.7.6.2. APPRENTISSAGE DES RESEAUX RBF GENERALISES	25
I.7.6.3. RESEAU RBF ET APPROXIMATION DE FONCTION	26
I.8. CONCLUSION	27

CHAPITRE .II. SYNTHÈSE DES TRAVAUX EFFECTUES DANS LE DOMAINE DE HT EN UTILISANT LES RNA

II.1. APPLICATION DES RNA POUR LA DETECTION DE LA DEFORMATION DES ENROULEMENTS DE TRANSFORMATEUR.....	29
II.2. ANALYSES DE CONCEPTION DES SUBSTANCES DE HT UTILISANT LES RNA	30
II.3. OPTIMISATION DE CHAMP ELECTRIQUE D'UNE ELECTRODE DE HT BASE SUR LES RNA.....	31
II.4. OPTIMISATION DES SYSTEMES DES ELECTRODES DE HT PAR RNA	31
II.5. MODELISATION DE LA TENSION DE CONTOURNEMENT D'UN ISOLATEUR POLLUE EN UTILISANT LES RNA	32
II.6. INFLUENCE DU DEVELOPPEMENT DU DEFAT DE LA DP SUR LA RECONNAISSANCE DU TYPE DE DEFAT EN UTILISANT LE RNA	33
II.7. IDENTIFICATION DES DP EN UTILISANT LA MESURE DE L'EMISSION ACOUSTIQUE ET RNA.....	34
II.8. RECONNAISSANCE DES DECHARGES PARTIELLES BASEES SUR LES ESEAUX DE NEURONES ARTIFICIELS	35
II.9. UNE NOUVELLE APPROCHE POUR LA CONTROLE DE CONDITION DES LIGNES AERIENNES A HAUTE TENSION EN UTILISANT LES RESEAUX DE NEURONE.....	36
II.10. APPLICATION DE RNA POUR LA DETECTION DES DEFATS DANS L'ENROULEMENT DE TRANSFORMATEUR.....	38
II.11. APPLICATION DE RESEAU DE NEURONE DANS LE DOMAINE VEILLISSEMENT THERMIQUE D'ISOLATEUR DE CABLE DE HAUTE TENSION ...	38

CHAPITRE .III. LES HUILES DES TRANSFORMATEURS

III.1. INTRODUCTION	40
III.2. LES HUILES MINERALES ISOLANTES	40
III.2.1. LES ROLES D'UNE HUILE ISOLANTES.....	40
III.2.2. CHOIX D'UN DIELECTRIQUE LIQUIDE.....	40
III.2.3. CARACTERISTIQUES D'UNE HUILE MINERALE ISOLANTE.....	40
III.2.3.1. PROPRIETES ELECTRIQUES.....	40
III.2.3.2. PROPRIETES PHYSIQUE.....	42
III.2.3.3. PROPRIETES CHIMIQUES	42
III.2.4. VIEILLESSEMENT DES HUILES DE TRANSFORMATEUR.....	43
III.2.4.1. VIEILLESSEMENT THERMIQUE DES HUILES DE TRANSFORMATEUR	43
III.2.4.2. LOIS DU VIEILLESSEMENT DES ISOLANTS	43
III.3. LES MECANISMES DE CLAQUAGE DES DIELECTRIQUES LIQUIDES	44
III.3.1. MECANISME ELECTRONIQUE.....	44
III.3.1.1. DISSOCIATION RENFORCEE PAR LE CHAMP	44
III.3.1.2. PRODUCTION DES CHARGES AUX INTERFACES.....	45
III.3.1.3. CONSEQUENCE DU TRANSFERT DE CHARGE.....	45
III.3.2. MECANISME DE CLAQUAGE AVEC PHASE GAZEUSE	45
III.3.2.1. MECANISME THERMIQUE.....	45
III.3.2.2. MECANISME DE CAVITATION.....	45
III.3.3. MECANISME DE CLAQUAGE PAR PONT	46
III.3.4. EVOLUTION DES DECHARGES DANS LES DIELECTRIQUES LIQUIDES.....	46
III.4. LES ESSAIS SUR LES DIELECTRIQUES LIQUIDES	47

III.4. LES ESSAIS SUR LES DIELECTRIQUES LIQUIDES	47
III.4.1. ESSAIS DIELECTRIQUES	47
III.4.1.1. MESURE DU FACTEUR DE PERTES DIELECTRIQUES ($\text{tg}\delta$) ET DE LA PERMITTIVITE RELATIVE (ϵ_r)	47
III.4.1.2. MESURE DE LA RIGIDITE DIELECTRIQUE	47
III.4.1.3. MESURE DE LA RESISTIVITE (ρ)	47
III.4.2. MESURES PHYSICO-CHIMIQUES	47
III.4.2.1. MESURE DE LA VISCOSITE CINEMATIQUE (η)	47
III.4.2.2. LA COULEUR	47
III.4.2.3. L'ACIDITE	48
III.4.3. ESSAIS DE VIEILLISSEMENT DE L'HUILE	48
III.5. HUILE DE TRANSFORMATEUR BORAC22	48
III.5.1. PROPRIETES DE L'HUILE BORAC22	49
III.5.2. INFLUENCE DE LA TEMPERATURE SUR LES PROPRIETES DE L'HUILE	50
III.5.2.1. FACTEUR DE PERTES DIELECTRIQUES	50
III.5.3. INFLUENCE DU VIEILLISSEMENT THERMIQUE SUR LES PROPRIETES DE L'HUILE	52
III.5.3.1. FACTEUR DE PERTES DIELECTRIQUES	52
III.5.3.2. PERMITTIVITE RELATIVE ϵ_r :	53
III.7. CONCLUSION	54

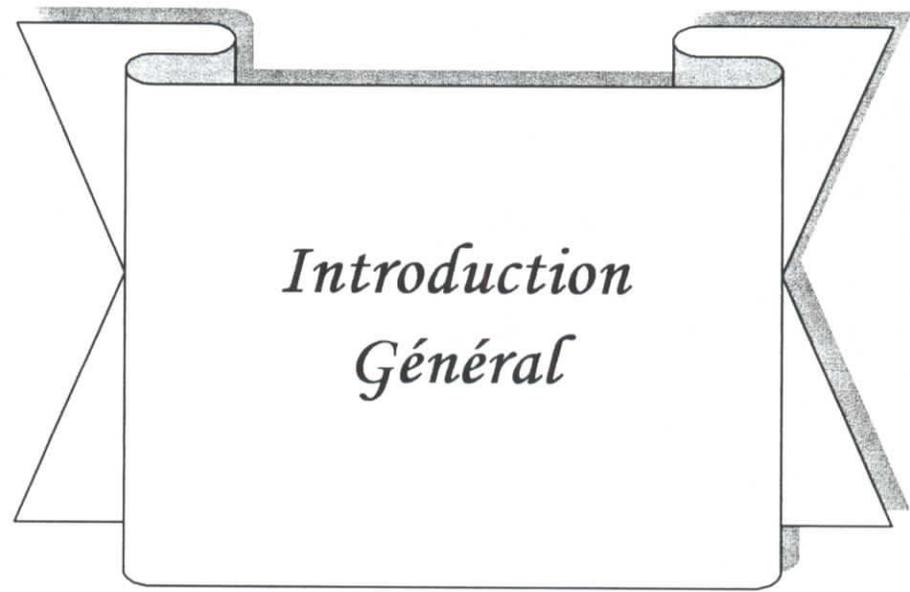
CHAPITER.II. ARCHITECTURE DES RESEAUX ELABORES

IV.1. INTRODUCTION	55
IV.2. ARCHITECTURE ET FONCTIONNEMENT DES RESEAUX RBF	55
IV.3. CODAGE DE L'INFORMATION	57
IV.4. PHASE D'APPRENTISSAGE DU RESEAU RBF	58
IV.5. PHASE DE PREDICTION	59
IV.6. EXEMPLE D' EXECUTION	62
IV.7. CONCLUSION	64

CHAPITRE V. RESULTATS ET INTERPRETATIONS

V.1. INTRODUCTION	65
V.2. RESULTATS A PARAMETRE CONSTANTE	65
V.2.1. RESULTATS LIES AU RESEAU RBF1	65
V.2.2. RESULTATS LIES AU RESEAU RBF1G1	68
V.2.3. RESULTATS LIES AU RESEAU RBF1G2	70
V.3. RESULTATS A PARAMETRES VARIABLES	72
V.3.1. RESULTATS LIES AU RESEAU RBF1	72
V.3.2. RESULTATS LIES AU RESEAU RBF1G1	78
V.3.3. RESULTATS LIES AU RESEAU RBF1G2	83
V.4. COMPARAISON ENTRE LES RESULTATS OBTENUS A PARTIR DES DIFFERENTES PROGRAMMES	86
V.5. INTERPRETATIONS	86
V.6. CONCLUSION	86

CONCLUSION GENERALE



Introduction
Général

INTRODUCTION GENERALE

L'utilisation du réseau de neurones artificiels (RNA) dans le domaine de technique de la haute tension (HT) est en cours de développement. A l'Ecole National Polytechnique (ENP), ce nouvel axe de recherche a été ouvert à partir de l'année universitaire 1997-1998.

Notre étude fait suite à d'autres travaux concernant l'utilisation des RNA dans le vieillissement thermique du PVC [1], le vieillissement thermique de l'huile de transformateur [2] et la détermination des caractéristiques des décharges électriques sur les surfaces polluées [3]. Ces travaux ont permis d'élaborer des programmes de calcul utilisant différentes techniques d'apprentissage. Le but est de réduire le temps d'expérimentation dans le cas de l'étude du vieillissement des matériaux isolants et développer des programmes permettant de bien définir l'évolution des décharges électriques sur des surfaces polluées, par exemples. [0].

Pour cette raison, nous nous intéressons dans notre travail à l'application des RNA au vieillissement thermique de l'huile de transformateur BORAK22 utilisée par SONELGAZ avec une durée de vieillissement de 5000h. Les résultats expérimentaux que nous avons utilisés, ont été réalisés par le Laboratoire de HT de l'ENP par d'autres chercheurs.

Ce travail est organisé dans cinq chapitres :

Dans le premier chapitre, nous présentons les généralités concernant les réseaux de neurones artificiels.

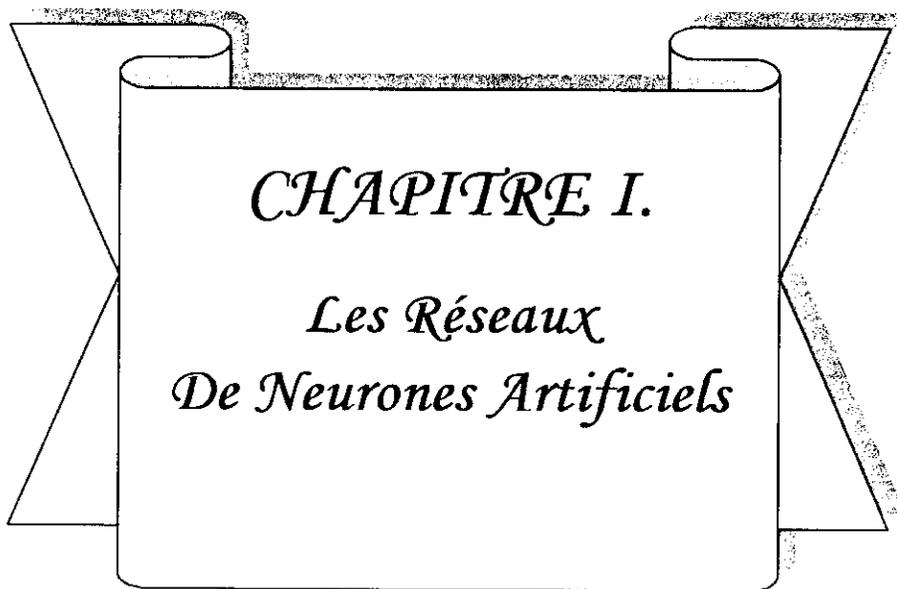
Le deuxième chapitre représente une synthèse de quelques travaux de recherche ayant utilisé les RNA dans le domaine de HT.

Le troisième chapitre présentes quelques caractéristiques d'une huile des transformateurs.

Dans le quatrième chapitre, nous détaillons les architectures des réseaux que nous avons étudiés, leurs organigrammes et un exemple d'application.

Enfin, dans le cinquième chapitre, nous présentons les différents résultats que nous obtenons ainsi que leurs interprétations.

Finalement, nous terminons ce travail par une conclusion générale.



CHAPITRE I.

*Les Réseaux
De Neurones Artificiels*

I.1. INTRODUCTION

Les réseaux de neurones artificiels ou réseaux connexionnistes sont fondés sur des modèles qui tentent d'expliquer comment les cellules du cerveau et leurs interconnexions parviennent, d'un point de vue globale, à exécuter des calculs complexes.

Ces systèmes qui stockent et retrouvent l'information de manière "similaire" au cerveau sont particulièrement adaptés aux traitements en parallèle de problèmes complexes comme la reconnaissance automatique de la parole, la reconnaissance de visages ou bien la simulation de fonctions de transfert. Ils offrent donc un nouveau moyen de traitement de l'information utilisé en reconnaissance de formes (vision, image, parole, etc.).

Les architectures connexionnistes s'inspirent de l'organisation neuronale du cerveau humain. Dans les réseaux de neurones artificiels de nombreux processeurs appelés cellules ou unités, capables de réaliser des calculs élémentaires, sont structurés en couches successives capables d'échanger des informations au moyen de connexions qui les relient. On dit de ces unités qu'elles miment les neurones biologiques.

Grâce à ce parallélisme massif, on peut espérer pouvoir surmonter les problèmes posés par des temps d'attente importants caractéristiques à la résolution de tâches complexes par des méthodes numériques.

I.2. LES NEURONES

Les Réseaux de Neurones Artificiels sont fondés sur une représentation schématique d'un neurone biologique. Les notions de mémorisation, d'apprentissage et de généralisation sont communes aux deux. La réponse du réseau de neurone à un signal d'entrée dépend de paramètres appelés poids synaptiques (mémoire). Une loi d'apprentissage modifie progressivement ces paramètres pour réduire un paramètre d'erreur. Si les exemples présentés forment un échantillon représentatif de l'ensemble des entrées, le réseau améliore son comportement pour n'importe quel signal (généralisation). L'élément fonctionnel du réseau de neurones artificiels est le neurone formel.

I.2.1. LE NEURONE BIOLOGIQUE

Les cellules nerveuses, appelées neurones, sont les éléments de base du système nerveux central. Le cerveau se compose d'environ 1012 neurones (mille milliards), avec 1000 à 10000 synapses (connexions) par neurone.

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone, l'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. en fait, il existe un espace intercellulaire de quelques dizaines d'angströms (10^{-9} m) entre l'axone du neurone afférent et les dendrites du neurone efférent, la jonction entre deux neurones est appelée la synapse.

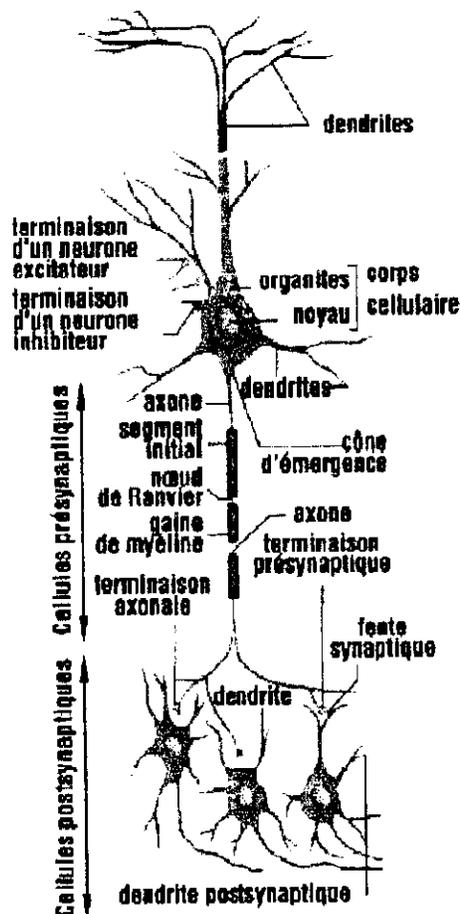


FIG. 1.1. Vue d'un neurone biologique

1.2.2. NEURONE ARTIFICIEL

1.2.2.1. DEFINITION

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau.

1.2.2.2. L'ETAT DES NEURONES

Un neurone artificiel est un élément qui possède un état interne. Il reçoit des signaux qui lui permettent éventuellement, de changer d'état. Nous noterons S l'ensemble des états possibles d'un neurone. S pourra être par exemple $\{0, 1\}$ où 0 sera interprété comme l'état inactif et 1 l'état actif. S pourra également prendre un nombre plus grand de valeurs $\{0, 1, \dots, \%P\}$ pour une image en $P + 1$ niveaux de gris ou même, par extension, un continuum de

valeurs $[-1, 1]$ ou \mathbb{R} tout entier. Dans une application correspondant aux signaux écho-radar peuvent être représentées continues. L'état d'un neurone peut alors être défini dans $[-1, 1]$ représente la valeur minimum du signal, et 1 le maximum.

L'état d'un neurone est fonction des états des neurones au. l'état d'un neurone il faut donc considérer les connexions entre neurones. Nous définirons par la suite les connexions entre neurones e .

I.2.2.3. LES CONNEXIONS ENTRE NEURONES

Architecture est le terme le plus général pour désigner la façon dont sont connectés les différents neurones qui composent un réseau. On parle également de voisinage (terme emprunté de la théorie des graphes). Au niveau des neurones on parle plutôt de voisinage. Ce terme fait allusion à la façon dont un neurone est connecté à d'autres neurones. Il est donc en rapport direct avec l'architecture du réseau. Voyons de plus près la signification du mot voisinage dans une architecture de réseaux de neurones.

I.2.2.3.1. LE VOISINAGE

Le voisinage d'un neurone est l'ensemble des neurones connectés à ce neurone. On parle de voisinage d'ordre n pour un neurone i , s'il y a n neurones connectés à ce neurone. Les connexions entre neurones ont souvent un sens.

I.2.2.3.2. LES CONNEXIONS

Une connexion est un lien établi explicitement entre deux neurones. Les connexions sont aussi appelées synapses, en analogie avec le nom des connecteurs des neurones réels. Une connexion entre deux neurones a une valeur numérique associée appelé poids de connexion.

I.2.2.3.3. LES POIDS DES CONNEXIONS

Le poids de connexion w_{ij} entre deux neurones j et i peut prendre des valeurs discrètes dans \mathbb{Z} ou bien continues dans \mathbb{R} . L'information qui traverse la connexion sera affectée par la valeur du poids correspond. Une connexion avec un poids $w_{ij} = 0$ est équivalente à l'absence de connexion.

On définit une matrice des poids de connexions \mathbf{W} où les lignes et les colonnes correspondent aux neurones et chaque valeur w_{ij} représente le poids de la connexion entre la cellule j et la cellule i du réseau.

I.2.2.4. DESCRIPTION D'UN NEURONE FORMEL

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. L'état du neurone, appelé aussi activité, est défini comme la somme pondérée de ses entrées. Son schéma de fonctionnement est donné en Figure 1-2. L'information est ainsi transmise de manière unidirectionnelle. Un neurone se caractérise par trois concepts : son état, ses connexions avec d'autres neurones et sa fonction d'activation. Nous utiliserons par la suite les notations suivantes.

- S_i : l'état à la sortie du neurone i .
- e_j ($j=1,2,\dots,p$): les entrées de neurone
- f_i : la fonction d'activation associée au neurone i .
- w_{ij} : le poids de la connexion entre les neurones j et i .
- w_{i0} : le poids de la connexion entre le neurone biais (+1) et les neurones i .

Ainsi, le neurone i recevant les informations de p neurones effectue l'opération suivante :

$$S_i = f\left(\sum_{j=1}^p w_{ij}e_j - w_{i0}\right) \quad (1.1)$$

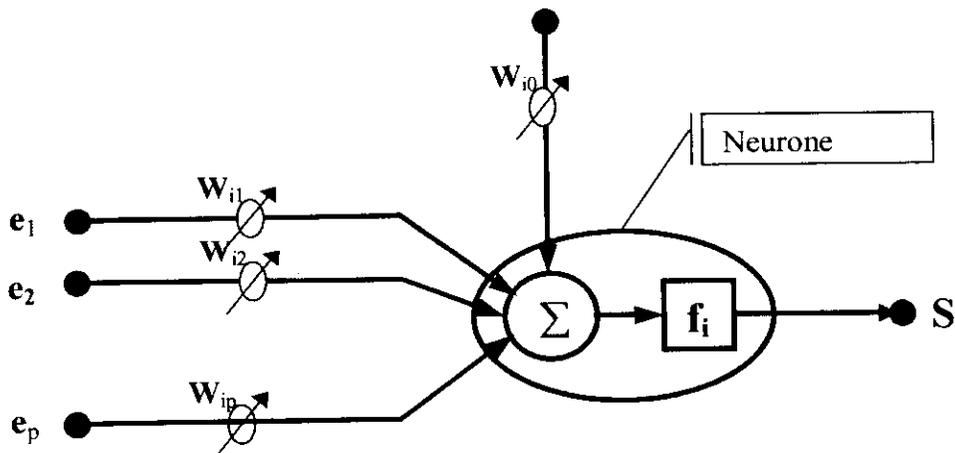


Fig. 1.2. Schéma de fonctionnement d'un neurone formel.

1.3. MODELE MATHEMATIQUE GENERAL DU NEURONE

1.3.1. FONCTION DISCRIMINANT (FONCTION DE BASE)

Elle définit l'activité du neurone. Dans le cas de Perceptron, c'est la fonction linéaire qui est utilisée. Cette fonction est depuis longtemps la plus usitée. C'est le cas pour les réseaux multicouches [4] et de manière générale les réseaux LBF (Linear Basis Function).

D'autres formes de fonctions discriminants non linéaires ont été élaborées ces dernières années afin d'améliorer les performances des réseaux.

Nous citons ci-dessous les fonctions de base les plus utilisées.

I.3.1.1. FONCTION DE BASE LINEAIRE LBF

Cette fonction fait la somme pondérée des entrées vers le neurone. Sa forme est en générale définie par :

$$\Phi(z, w) = z^T \cdot w \quad (1.2)$$

Où W représente la matrice des poids et z les entrées qui viennent de l'extérieur ou de la part des autres neurones vers le neurone en question. En plus de son efficacité, cette fonction est plus simple à implémenter.

I.3.1.2. FONCTION DE BASE RADIALE RBF (RADIAL BASIS FUNCTION)

La forme de cette fonction est définie par :

$$\Phi(\lambda, w) = -\frac{\|x - w\|^2}{2} \quad (1.3)$$

Où les poids W sont représentés comme étant les centroïdes de chaque classe de l'espace d'entrée. Le discriminant calcule le rayon entre les entrées x et le centre de chaque classe.

I.3.1.3. FONCTION DE BASE ELLIPTIQUE EBF (ELLIPTIC BASIS FUNCTION)

La forme de ce discriminant est une généralisation de la fonction de base radiale

$$\Phi(x, w) = \sum_{k=1}^n \alpha_k (x - w)^2 + \theta \quad (1.4)$$

Celle ci a une forme modulée par des paramètres α_k et dotée d'un Biais θ . Ces paramètres font qu'elle ne soit pas symétrique

I.3.2. FONCTION DYNAMIQUE LINEAIRE

Ce bloc détermine la dynamique du réseau. Cette fonction est régie par l'équation différentielle de premier ordre suivante

$$\alpha_0 \dot{u}_i(t) + \alpha_1 u_i(t) = v_i \quad (1.5)$$

Où u_i représente l'activité du $i^{\text{ème}}$ neurone et v_i l'entrée du système dynamique décrit par cette équation.

I.3.3. FONCTION NON DYNAMIQUE NON LINEAIRE

Elle a pour objectif de rendre l'activité du neurone bornée, la fonction d'activation peut être une fonction de seuil (Simple Threshold Function), sigmoïde (Sigmoidal), tangente hyperbolique (Hyperbolic Tangent) ou fonction de base Radiale [5].

Tableau 1- les fonctions d'activation.

nom de fonction	La forme mathématique
Linéaire	$f(s) = s$
Seuil	$f(s) = \begin{cases} +1, & \text{si } s > s_t \\ -1, & \text{si non} \end{cases}$
Sigmoïde	$f(s) = a \frac{e^{ks} - 1}{e^{ks} + 1}$
Tangente hyperbolique	$f(s) = \frac{1 - e^{-2s}}{1 + e^{-2s}}$
Fonction de base radiale	$f(s) = e^{-s^2 / \beta^2}$

La fonction d'activation doit être continue, dérivable et monotone.

Il existe une autre fonction d'activation qui n'est pas monotone, s'est la fonction Gaussienne; elle se distingue des autres, par sa linéarité significative et son comportement impulsionnelle. La caractéristique de comportement local rend cette fonction très puissante en approximation de fonction et en classification.

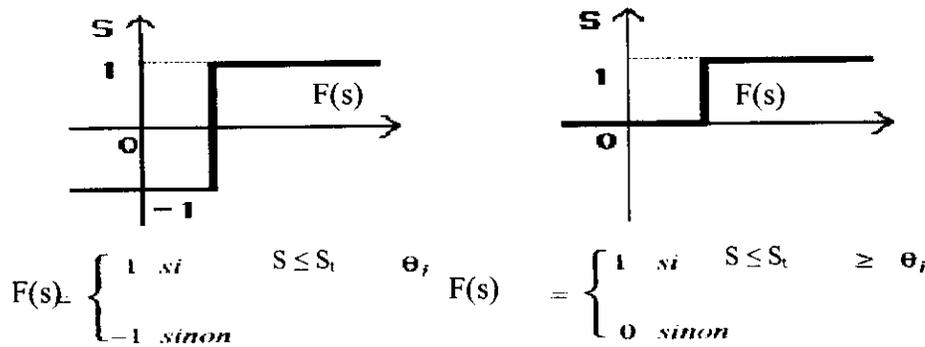


Fig. I. 3. Fonction à seuil

En générale, toutes les fonctions citées ont un fondement biologique [5] par exemple les neurone du système visuel sont connus ayant un comportement Gaussien. [6]. La recherche de l'efficacité a permis à d'autres fonctions de voir le jour ces dernières années comme les fonctions logarithmique, cubiques ou exponentielles logarithmiques.

En générale, toutes les fonctions citées ont un fondement biologique [6] par exemple les neurone du système visuel sont connus ayant un comportement Gaussien. [7]. La recherche de L'efficacité a permis à d'autres fonctions de voir le jour ces dernières années comme les fonctions logarithmique, cubiques ou exponentielles logarithmiques.

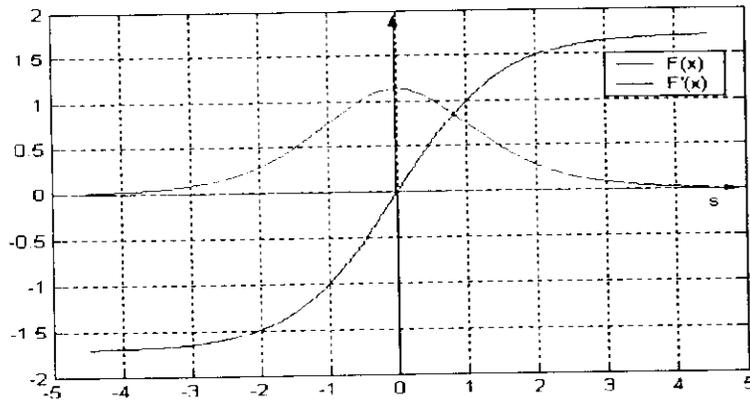


FIG. I.4. Fonction sigmoïde f et dérivée première f' . f est une fonction Quasi-linéaire, pour $a=1.72$ et $k=1.33$ (voir tableau 1)

I.4. CLASSIFICATION D RESEAUX DE NEURONES

La figure suivante (Fig.I.5.) résume les trois différentes possibilités suivant lesquelles on peut classifier les réseaux de neurones. Dans ce qui suit, nous utilisons ces différentes classifications pour faire l'étude de plusieurs points liés aux réseaux de neurones.

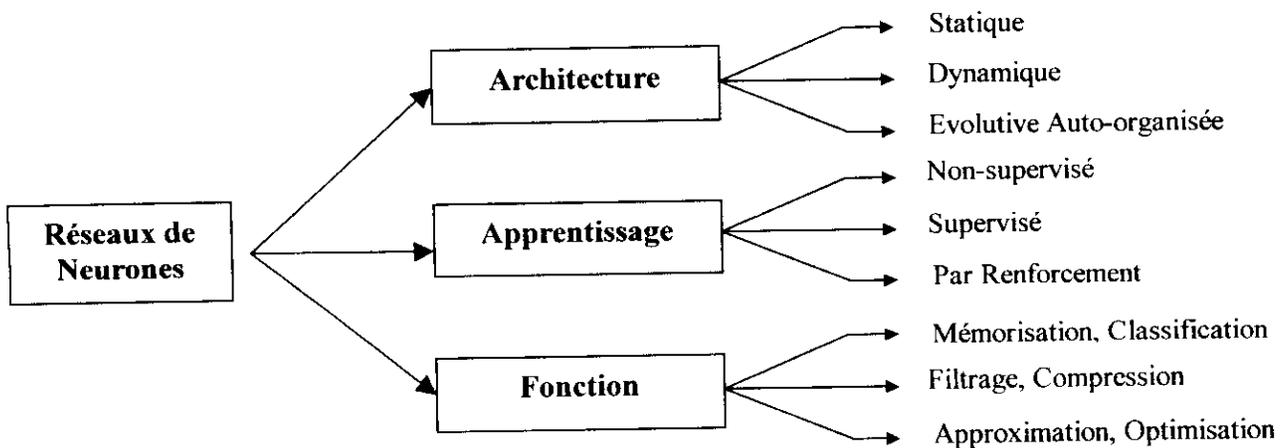


Fig.I.5. Les différentes possibilités de classification des réseaux de neurones

I.5.1. LES TYPES D'APPRENTISSAGE

Les techniques d'apprentissage des réseaux de neurones se repartissent en deux grandes familles:

• L'apprentissage est dit supervisé lorsque les exemples sont constitués de couples de valeurs du type : (valeur d'entrée, valeur de sortie désirée). Tout le problème de l'apprentissage supervisé consiste, étant donné un ensemble d'apprentissage E de N couples (entrée-sortie désirée) $(x_i, y_i)_{i = 1, 2, \dots, n}$, à déterminer le vecteur des poids W d'un réseau F_w capable de mettre ces informations en correspondance, c'est à dire un réseau tel que

$$F_w(x_i) = y_i, \quad i = 1, 2, \dots, n. \tag{1.6}$$

• L'apprentissage est qualifié de non-supervisé lorsque seules les valeurs d'entrée sont disponibles. Dans ce cas, les exemples présentés à l'entrée provoquent une autoadaptation du réseau afin de produire des valeurs de sortie qui soient proches en réponse à des valeurs d'entrée similaires [9].

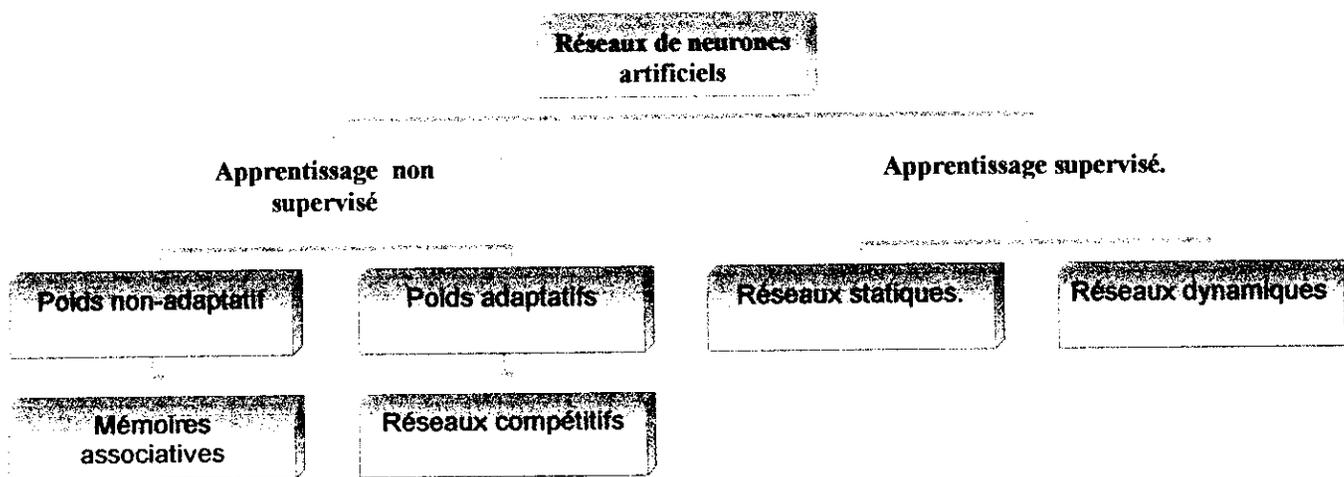


Fig. I.6. Classification des réseaux de neurones suivant leur apprentissage

1.6. RESEAUX DE NEURONES A APPRENTISSAGE NON-SUPERVISE

Ces réseaux traitent des échantillons en entrée sans avoir d'informations sur les sorties que ceux ci doivent générer. L'objectif de ses réseaux n'est pas de faire une approximation. Le rôle de ces réseaux est basé sur la mémorisation et la classification de leurs entrées.

1.6.1. MEMOIRE ASSOCIATIVE

La mémoire associative est une des premières applications issues des réseaux de neurones adaptatifs. Il s'agit de réaliser un dispositif capable de mémoriser des informations que l'on peut ensuite retrouver, non par leur adresse comme dans une mémoire classique, mais en fournissant des données même incomplètes ou bruitées relatives aux informations stockées [8].

Les premiers modèles de mémoires associatives sont ceux de Widrow et de Rosenblatt, suivis par les modèles de Kohonen et Anderson. [10]

La mise en mémoire des informations à l'intérieur de ces réseaux se fait grâce à un apprentissage non-supervisé. Pendant cette étape, appelée **Encoding Step**, le réseau calcule les poids synaptiques W , qui représente un codage de l'information à stocker. Grâce à la présentation interne déduite de l'apprentissage, les neurones calculent leurs activations. A la sortie, il doit se rappeler de ce qu'il appris. Cette étape est appelée **Recal Step**. [8]

On distingue deux types de mémoires associatives suivant le mode de fonctionnement :

a) Mémoires auto-associatives :

Elles sont destinées à stocker des vecteurs prototypes

$$P^k = p_1^k, p_2^k, \dots, p_n^k. \tag{I.7}$$

Tel que $k = 1, 2, \dots, M$ représente les différentes classes.

b) Mémoires hétéro-associatives :

Elles stockent les couples Prototypes (X^k, Y^k) , où :

$$Y^k = y_1^k, y_2^k, \dots, y_n^k \quad \text{avec } k = 1, 2, \dots, M \tag{I.8}$$

A chaque vecteur X^k en entrée, cette mémoire associe le vecteur Y^k en sortie et vice versa.

I.6.1.1. RESEAU DE HOPFIELD

Ce modèle a été développé en 1982 selon une analogie formelle avec des réseaux de particules orientées en interaction. Ce réseau est une mémoire auto-associative constituée par une couche de neurones entièrement interconnectés (**Fig.I. 7.**)

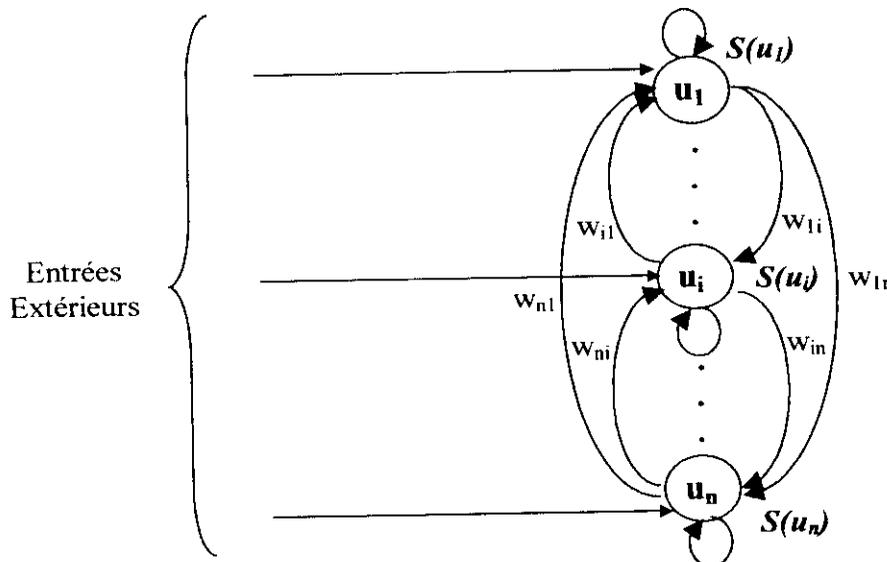


Fig.I.7. Architecture du réseau de Hopfield

L'originalité du réseau de Hopfield réside notamment dans son modèle électrique très connu, ce modèle traduit le comportement dynamique intégrateur des neurones, celui-ci est défini par le système d'équation suivant :

$$\dot{u}_i = \sum_{j=1}^n w_{ij} S_j(u_j) - b_i u_i + I_i \quad i = 1, 2, \dots, n \tag{I.9}$$

Où I_i représentent les entrées extérieures, w_{ij} des paramètres définissant les forces des liens synaptiques représentées par les conductances de ces connections, b_i représente la conductance équivalente à l'entrée du neurone, et $S_j(u_i)$ représente la sortie du neurone [8]. La sortie du neurone est binaire comme son entrée, donc la fonction de sortie est une fonction seuil définie par l'équation (I. 9).

1.6.1.2. MEMOIRES ASSOCIATIVES BIDIRECTIONNELLES (RAM)

Elles représentent une extension du réseau de Hopfield vers une mémoire hétéro - associative, ces réseaux sont composés de deux couches de neurones, l'information à l'intérieur de ce réseau peut circuler dans les deux sens (de la première couche vers la deuxième couche et vice versa).

1.6.1.3. MEMOIRES ASSOCIATIVES LINEAIRES OPTIMISEES (OLAM)

Le réseau de Hopfield ne garantit un bon fonctionnement que si les exemples sont mutuellement orthogonaux, ce qui n'est pas toujours vérifié en pratique.

Les mémoires associatives optimisées constituent une solution pour les problèmes d'orthogonalité, ces réseaux utilisent la règle de projection pour leur apprentissage, mais leur architecture est la même que celle de Hopfield [8].

En injectant une entrée égale à un des prototypes stockés ou proche de lui, on doit retrouver ce prototype en sortie. Ceci revient à résoudre l'équation suivante :

$$W X^k = X^k \quad \text{avec } k = 1, 2, \dots, M \quad (\text{I. 10})$$

Une solution de cette équation est :

$$W = X \cdot X^{\dagger} \quad (\text{I.11})$$

Où X est une matrice regroupant tout les vecteurs prototypes à mémoriser et X^{\dagger} représente le pseudo-inverse de cette matrice.

1.6.2. RESEAUX COMPETITIFS

Ces réseaux s'adaptent au fur et à mesure de leur utilisation, contrairement aux réseaux que nous avons étudié jusqu'à maintenant, qui sont à poids fixes. Ces réseaux contiennent, en plus de la couche d'entrée, une couche compétitive, où les neurones entrent en compétition entre eux, afin de déterminer lequel est le plus représentatif de la classe de l'espace d'entrée. Nous présentons maintenant les modèles des principaux réseaux compétitifs.

1.6.2.1. COUCHE DE GROSSBERG

Elle est constituée de neurones entièrement interconnectés, sa dynamique est régie par l'équation différentielle suivante :

$$\dot{u}_i = -\mu \cdot u_i + \delta \sum_{j=1}^n S(u_j) w_{ij} + I_i \quad i = 1, 2, \dots, n \quad (\text{I.12})$$

Où :

- u_i : représente l'activité du $i^{\text{ème}}$ neurone.
- I_i : l'entrée extérieure vers ce neurone.

$S(u_i)$: représente la sortie du neurone.

w_{ij} : les poids synaptiques.

δ et μ : constantes positives.

Lors de la présentation d'une entrée X^k au réseau, la dynamique de l'équation (I.12) fait que le neurone de la classe à laquelle cet exemple est le plus proche est activé au maximum. A l'équilibre c'est le seul qui reste activé et le seul à se voir réadapter ses poids. L'apprentissage se fait de la manière suivante :

$$\dot{w}_{ij} = S(u_i)[- \alpha w_{ij} + \beta S(u_i)] \quad (I.13)$$

Où :

α : un paramètre contrôlant le régime transitoire.

β : le taux d'apprentissage.

Le défaut de ce modèle est qu'il traite de la même manière que le signal. En effet l'équation (I.13), qui est linéaire, ne permet aucun filtrage du bruit. Ceci conduit le modèle rapidement vers la saturation.

I.6.2.2. ADAPTIVE RESONANCE THEORY ART

Le fonctionnement de ce réseau constitue une analogie avec la résonance physique. A partir d'un faible signal, l'information fait des 'aller-retour' entre la couche d'entrée et celle de sortie, constituant des vibrations, qui continuent jusqu'à ce que celle-ci entre en résonance soit en adhérant à une classe déjà existante, soit en créant une nouvelle classe, suivant la finesse de la classification [8].

Ce réseau est constitué de deux couches ; une couche d'entrée ordinaire et une couche de sortie appelée Maxnet, dont les neurones sont interconnectées (**Fig.1.8**)

Chaque neurone de la couche de sortie représente une classe. Lorsque une entrée x est présentée au réseau, celle-ci est transmise vers la couche supérieure en passant par des liens synaptiques allant dans le sens direct appelés Bottom-Up « bas-haut ». Chaque neurone du Maxnet calcule son activation. Le neurone ayant ses poids les plus corrélés avec l'exemple en entrée, présente le plus grand résultat en sortie, celui-ci sera candidat pour être désigné comme étant la classe à laquelle cet exemple appartiendrait. Avant de prendre cette décision, l'exemple x est envoyé vers la couche d'entrée, en passant par les liens synaptiques qui vont dans le sens inverse appelés Top-Down « haut-bas », là, un test dit de vigilance, est appliqué afin d'estimer à quel point cet exemple ressemble à ceux appartenant à la classe gagnant. Si celui-ci réussit le test, l'exemple est accepté. Sinon le neurone gagnant est disqualifié et une autre classe est recherchée [8].

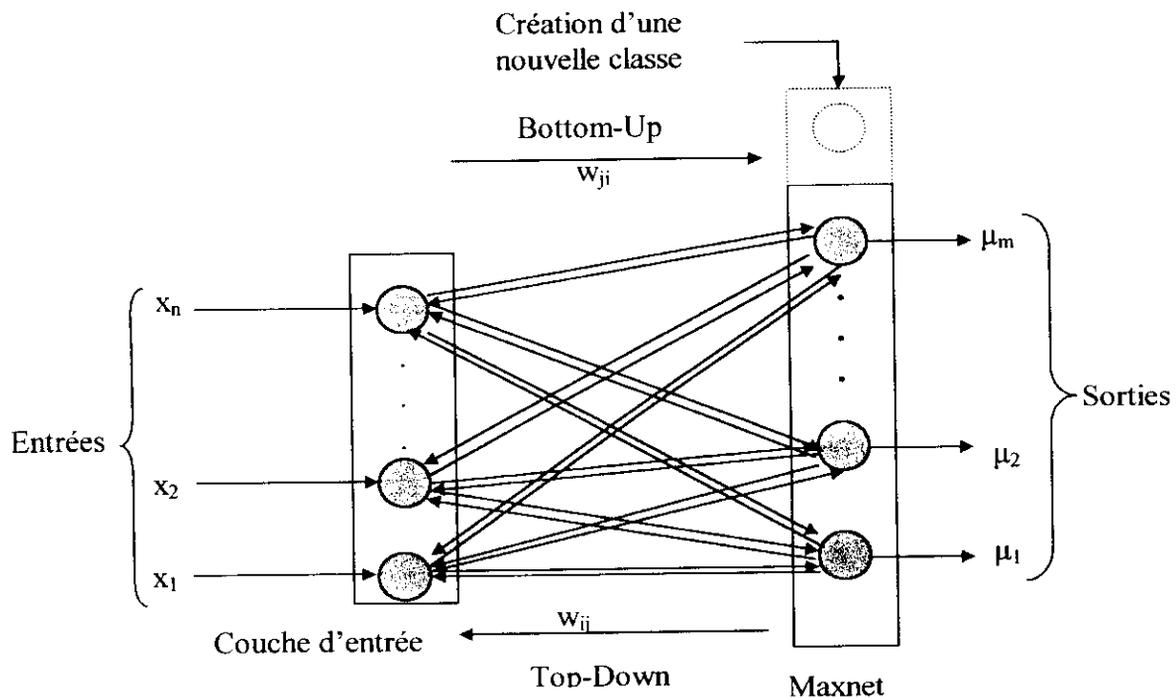


Fig.I.8. Architecture du ART

I.7. APPRENTISSAGE SUPERVISE

Nous considérerons par la suite un réseau comme un associer défini par son architecture et un ensemble de paramètres à ajuster. Pour simplifier la présentation, nous considérerons que les seuls paramètres à ajuster sont les poids des connexions.

Il existe deux approches qui peuvent être faites pour ce type d'apprentissage. La première est basée seulement sur la validité de la décision du réseau. Ce type de réseau est appelé (DBNN) réseaux basés sur les décisions, la deuxième catégorie est basée sur l'optimisation d'un critère.

I.7.1. REGLES DU PERCEPTRON ET D'ADALINE, MADALINE

Les règles d'apprentissages utilisées dans le Perceptron de Rosenblatt et ADALINE (ADAPtive LINEar NEuron) de Widrow et Hoff reposent toutes deux sur la minimisation d'un critère d'erreur quadratique instantanée.

Les architectures des réseaux proposés dans les deux cas sont très voisines (Fig.I.9).

A savoir une couche d'entrée connectée à une où plusieurs sorties, dans le cas de plusieurs sorties, il est alors appelé MADALINE (Multiple-Adaline).

La caractéristique du neurone dans le Perceptron est une fonction Echelon : $F(p) = \text{Ech}(p - \theta)$, où θ est un seuil.

L'apprentissage se fait avec superviseur, c'est-à-dire que l'on présente un vecteur à l'entrée du réseau et en même temps on fournit la sortie désirée : dans le cas d'Adaline, il s'agit du potentiel désiré $P_{\text{désiré}}$, dans le cas du Perceptron, il s'agit de la classe désirée $y_{\text{désirée}}$.

Pour un seul « neurone », à chaque pas d'apprentissage, l'erreur quadratique E est minimale est donc :

$E = (P - P_{\text{désiré}})^2$, erreur analogique, dans le cas d'Adaline,

$E = (y - y_{\text{désirée}})^2$, erreur binaire, dans le cas du Perceptron.

Le principe de minimisation d'erreur n'est autre qu'un cas particulier de l'application de la méthode de descente de gradient.

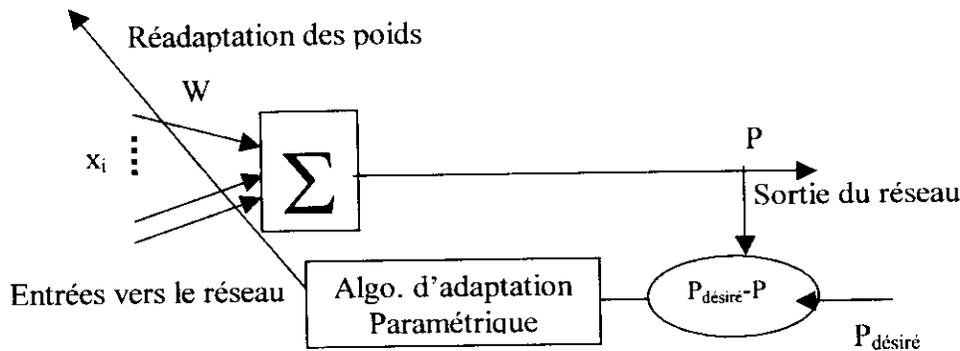
$$\frac{\partial E}{\partial w_k} = 2(P - P_{\text{désiré}}) \frac{\partial P}{\partial w_k} \tag{I.14}$$

Or $P = \sum_j w_j x_j$,

Donc $\frac{\partial P}{\partial w_k} = x_k$,

Et finalement

$$\frac{\partial E}{\partial w_k} = 2(P - P_{\text{désiré}}) x_k.$$



ADALINE

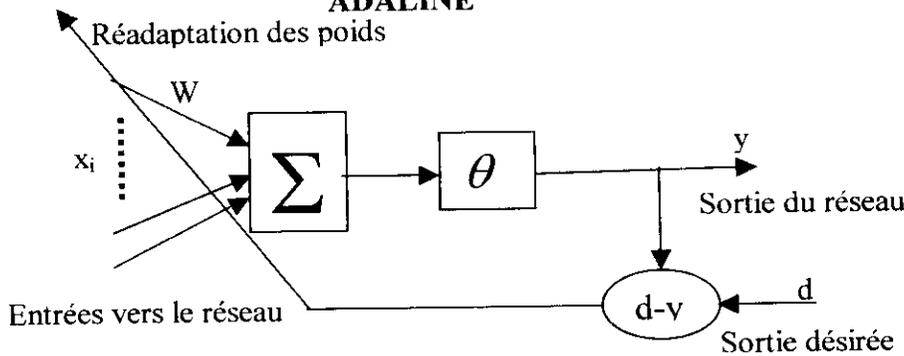


Fig. I.9. Neurones D'Adaline et Perceptron

D'où :

$$\Delta w_k = -\alpha (P - P_{\text{désiré}}) x_k, \tag{I.15}$$

Où α est un gain d'adaptation positif.

Le même type d'équation que (I.15) est utilisé dans l'algorithme du Perceptron, les poids sont modifiés selon la règle suivante :

$$\Delta w_k = -\alpha(y - y_{désirée})x_k \quad (\text{I.16})$$

Cet algorithme est très simple, mais il présente l'inconvénient des algorithmes de type gradient stochastique en ce qui concerne la convergence. Par ailleurs, il est évident que minimiser l'erreur quadratique instantanée n'est pas identique à minimiser l'erreur quadratique moyenne calculée sur toute la base d'apprentissage, et que ce dernier critère est le plus performant [8].

1.7.1.1. FONCTION DE COUT ET METHODE DE GRADIENT

Les techniques de gradient permettent l'estimation des paramètres W par l'optimisation d'une fonction de coût qui caractérise les performances du réseau. Cette fonction de coût sera calculée entre les vecteurs désirés et les vecteurs calculés par le réseau.

Soit l'ensemble des vecteurs d'entrée X_1, X_2, \dots, X_m , définis dans R_n , les vecteurs réponses y_1, y_2, \dots, y_m , définis dans R_p et W la matrice de poids de connexions définie dans R_{pn} . Alors, on a :

$$x_k = [x_1^k, x_2^k, \dots, x_n^k]^T, y_k = [y_1^k, y_2^k, \dots, y_p^k]^T \text{ et}$$

$$W = [w_1, w_2, \dots, w_p]^T = [w_{ij}]_{p \times n}$$

Où $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ est le vecteur colonne composé par les poids des connexions entre les n cellules d'entrée et l'automate i en sortie. Définissons la fonction de coût suivante :

$$C(W) = \frac{1}{2} \sum_{k=1}^m \|w^T \cdot x_k - y_k\|^2 \quad (\text{I.17})$$

Où $\|\cdot\|^2$ est la norme quadratique. Le problème est alors de trouver un W^* Pour W tel qu'il minimise $C(W)$. Une condition nécessaire est la suivante :

$$\nabla C(w)_{w=w^*} = \left\{ \frac{\partial C(w)^T}{\partial w_{ij}} \right\}_{p \times n} \Big|_{w=w^*} = 0 \quad (\text{I.18})$$

Les techniques de gradient sont des méthodes d'optimisation itératives qui permettent de trouver des solutions vérifiant la condition (3.8). La version la plus simple de ces techniques est celle dite du *gradient de la plus grande pente* :

Définir la configuration initiale $W = W_0$
 $\forall k, k = 1, 2, \dots$

$$W_{k+1}^T = W_k^T - \varepsilon \nabla C(W_k) \quad (\text{I.19})$$

Où ε est un scalaire positif, souvent dépendant de k –auquel cas on écrit ε^k –et qui est appelé le pas d'itération. Ce coefficient est incorporé dans le terme Δw_{ij}^k .

Dans le cas quadratique, en remplaçant (I.17) dans (I.19) on obtient l'expression :

$$W_{k+1}^T = W_k^T - \varepsilon^k \sum_{k=1}^m (W^T x_k - y_k) \cdot x_k^T \quad (I.20)$$

Dans (I.20) les changements de chaque $w_{ij} \in \mathbf{W}$ sont fonction des toutes les données $(\mathbf{x}_k, \mathbf{y}_k)$, $k = 1, 2, \dots, m$, m étant le nombre d'exemples.

I.7.1.2. LE GRADIENT STOCHASTIQUE : LA REGLE DE WIDROW-HOFF

Ces techniques de gradient ne sont pas adaptées au cas des réseaux connexionnistes où l'on désire mettre en oeuvre des systèmes adaptatifs capables de modifier leurs paramètres en continu en fonction d'un environnement qui peut changer. Widrow et Hoff ont proposé en 1960 la règle adaptative qui porte leur nom et qui est également connue sous le nom de règle LMSE [8]. Il s'agit tout simplement d'une règle de gradient suivant la plus grande pente où les paramètres w sont modifiés non pas en tenant compte à chaque itération de l'erreur globale réalisée sur un ensemble de formes, mais après chaque présentation de forme on modifie le vecteur de paramètres. La formulation mathématique de la règle est la suivante :

$$W_{k+1}^T = W_k^T - \varepsilon^k (W^T \cdot x_k - y_k) \cdot x_k^T \quad (I.21)$$

Où $(\mathbf{x}_k, \mathbf{y}_k)$ est le couple présenté à l'instant de temps t_k . Il a été démontré que, dans le cas d'une erreur quadratique, (I.21) converge avec une probabilité 1 vers \mathbf{W}^* , la solution de \mathbf{W} que minimise la fonction de coût $C(\mathbf{W})$.

L'algorithme de l'Adaline est très général puisqu'il permet d'apprendre des associations quelconques. Quelque soit la nature du problème étudié, cet algorithme fournira une solution qui est la meilleure que l'on puisse avoir avec cette architecture au sens de l'erreur aux moindres carrés. Toutefois, l'architecture utilisée étant extrêmement simple, il est évident que la qualité de l'approximation trouvée ne sera pas toujours excellente.

L'algorithme de l'Adaline est extrêmement employé, c'est notamment l'algorithme de base en traitement adaptatif du signal. Toutefois, à cause des limitations évoquées plus haut, on a cherché des architectures et des algorithmes permettant de réaliser des tâches plus complexes ou avec une meilleure approximation que celle fournie par une simple Adaline. Ces recherches ont donné lieu à de très nombreuses publications dans les années 60-70. Toutefois, aucun algorithme général n'a été formulé permettant d'entraîner des réseaux plus complexes et d'usage général. Widrow par exemple a lui même proposé la notion de Madaline [3] où plusieurs Adalines sont utilisées simultanément ou séquentiellement pour résoudre des problèmes non solubles de façon exacte par une seule Adaline. Les Madalines, comme tous les algorithmes présentés à la même époque, nécessitent une importante connaissance a priori du problème qui permette de définir l'architecture à employer. Ce type de connaissance n'est bien sûr quasiment jamais disponible quand on s'attaque à un problème réel. Il ne s'agit donc pas d'une machine d'apprentissage général.

Il faut attendre le milieu des années 80 pour voir apparaître des machines beaucoup plus puissantes que celle des années 60 avec des procédures d'apprentissage générales et efficaces. Le modèle le plus célèbre qui est également celui que nous avons employé pour nos applications est celui des Perceptrons multicouches entraînés par l'algorithme de

rétro-propagation *du gradient*. Toutefois, cet algorithme n'est qu'une généralisation à un type d'architecture plus complexe de celui de l'Adaline.

I.7.2. RESEAUX MULTICOUCHES STATIQUES

L'entrée dans ces réseaux est suivie par une succession de couches dites couches cachées, pour finalement aboutir à la sortie (**Fig.I.10**) ; ce passage par ces couches contribue à la richesse du traitement de l'information à l'intérieur du réseau afin d'obtenir le résultat désiré en sortie [8].

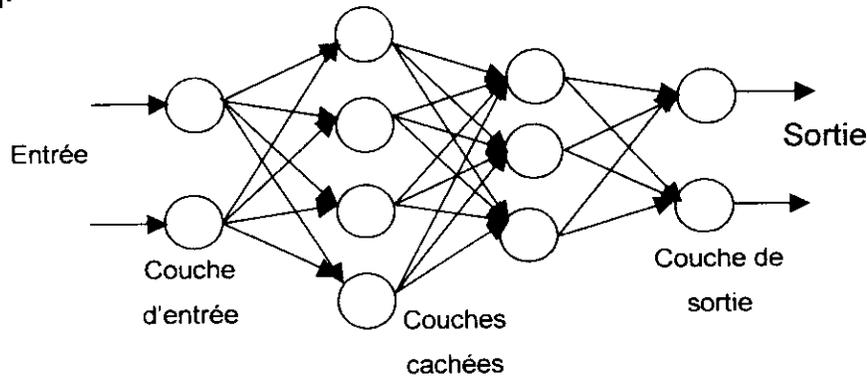


Fig.I.10. Réseau statique multicouche

I.7.3. RESEAUX A FONCTION DE BASE LINEAIRE LBF (LINEAR BASIS FUNCTION)

Le réseau LBF a un discriminant linéaire, il est régi par l'équation dynamique suivante :

$$u_i^l(t) = \sum_{j=1}^{N_{l-1}} w_{ij}^l(t) y_j^{l-1}(t) + \theta_i^l(t) \tag{I.22}$$

$$y_i^l(t) = f(u_i^l(t)) \tag{I.23}$$

Tel que :

l : le nombre de couches.

$u_i^l(t)$: représente l'activité du $j^{\text{ème}}$ neurone dans la $l^{\text{ème}}$ couche.

$y_i^l(t)$: la sortie du $i^{\text{ème}}$ neurone dans la $l^{\text{ème}}$ couche, en provenance du $j^{\text{ème}}$ neurone de la couche précédente.

$\theta_i^l(t)$: est un paramètre appelé Bias, il a pour objectif de donner de plus de souplesse à la fonction de sortie du neurone en approximation de fonctions.

La fonction d'activation f du neurone est monotone non décroissante. Elle a pour rôle de limiter l'activité du neurone. Pour les neurones de la couche d'entrée, cette fonction n'est autre que l'identité. Ainsi dans la couche d'entrée, chaque neurone fait passer une composante du vecteur entrée vers sa sortie sans aucun traitement, c'est à dire $y_i^1(t) = x_i(t)$ où $x_i(t)$ représente la $i^{\text{ème}}$ entrée du réseau à l'instant t .

I.7.3.1. LA RETRO-PROPAGATION DU GRADIENT

Les limitations des modèles neuronaux des années 60 comme l'Adaline ou le Perceptron [2], ont conduit les chercheurs à abandonner progressivement cette ligne de méthodes d'apprentissage au profit des approches symboliques de ce que l'on appelle aujourd'hui l'Intelligence Artificielle classique.

Dans les années 80, grâce aux travaux de Teuvo Kohonen et de J. Hopfield [8], cette voie a été remise au goût du jour et a suscité un intérêt croissant de la part de nombreux chercheurs issus de différentes disciplines.

C'est ainsi que fut proposé simultanément par plusieurs équipes travaillant indépendamment l'algorithme de la rétro-propagation du gradient. David E. Rumelhart, Geoffrey E. Hinton et Ronald J. Williams en 1986 [5], ont présenté la méthode, qu'ils ont appelé la Règle Delta Généralisée [5]. Presque simultanément, Yan le Cun [8] et David B. Parker [8] présentaient chacun séparément des algorithmes d'apprentissage similaires. Des dérivations formelles de la méthode de rétro-propagation du gradient peuvent être trouvées dans [19] et [17].

I.7.3.2. L'ALGORITHME DE RETRO-PROPAGATION DU GRADIENT

Notons x_i l'état d'un neurone i , $x_i \in S$ qui est l'ensemble des états possibles. l'état x_i du neurone i est calculé par :

$$x_i = f(A_i - \theta_i) \quad (I.24)$$

Où, pour les MLP, f est une fonction sigmoïde, A_i l'activité du neurone i et θ_i le seuil associé au neurone i . L'activité A_i est donnée par :

$$A_i = \sum_j W_{ij} x_j \quad (I.25)$$

Où j est indice des neurones "en amont" du neurone i , et w_{ij} est le poids de la connexion du neurone j au neurone i .

I.7.3.2.1. PROPAGATION DES ETATS

On considère encore une fois l'ensemble des vecteurs d'entrée X_1, X_2, \dots, X_m , définis dans \mathbb{R}^n et les vecteurs réponses Y_1, Y_2, \dots, Y_m , définis dans \mathbb{R}^p et W la matrice de poids de connexions cette fois-ci définie dans $\mathbb{R}^{nc \times nc}$, nc étant le nombre total de cellules. Alors,

$$x_k = [x_1^k, x_2^k, \dots, x_n^k]^T, y_k = [y_1^k, y_2^k, \dots, y_p^k]^T \text{ et}$$

$$W = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ w_1 & 0 & \dots & 0 & 0 \\ 0 & w_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & w_p & 0 \end{bmatrix} = \{ W_{ij} \}_{nc \times nc}, \text{ où } w_i = \{ w_{ki} \}_{nc(i-1) \times nc(i)} \quad (I.26)$$

Chaque sous-matrice W_i contient les poids de connexions entre la couche $i-1$ –avec $nc_{(i-1)}$ cellules– et la couche i –avec $nc_{(i)}$ cellules–.

A chaque instant k , un vecteur X_k est présenté dans la couche d’entrée du réseau. Les états des n cellules en entrée prennent les valeurs $(x_1^k, x_2^k, \dots, x_n^k)$

Ces états sont propagés selon les équations (I.24) et (I.25), vers des unités se trouvant dans les couches en aval de la couche d’entrée jusqu’à arriver à la couche de sortie.

Appelons $S_k = (s_1^k, s_2^k, \dots, s_p^k)$ la réponse du réseau pour une entrée X_k donnée. S_k est le vecteur composé par les états de p cellules de la couche de sortie. Leurs états ont été calculés par les équations (I.24) et (I.25) en fonction des cellules j connectées à la couche de sortie

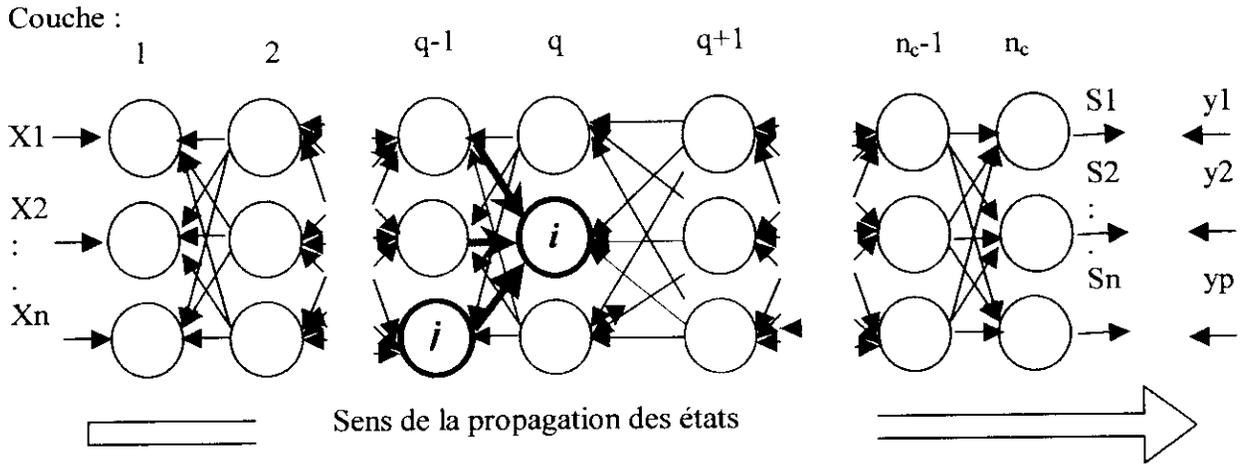


Fig.I.11. Propagation des états dans un réseau multicouche.

La propagation des états se fait “en aval” dès la couche d’entrée vers la couche de sortie. L’état de la cellule i dans la couche q est fonction des états des cellules j des couches précédentes $q-1$, etc.

1.7.3.2.2. RETRO-PROPAGATION DE L’ERREUR

Le but étant d’obtenir pour une entrée X_k une réponse S_k la plus proche possible du vecteur y^k désiré correspondant, on est amené à optimiser notre réseau afin de réduire la différence entre sortie désirée et calculée comme pour l’Adaline.

Définissons la fonction de coût suivante :

$$C^k(W) = \sum_{i=1}^p (s_i^k - y_i^k)^2 \tag{I.27}$$

qui est l’erreur ou distance quadratique entre la sortie calculée s^k et la sortie désirée y^k . La totalité des poids de connexions du réseau doit être ajustée en fonction de cette erreur. Cette fonction de coût dépendra bien sûr de l’état du système et de l’exemple qui lui est présenté. Pour minimiser la fonction de coût $C^k(W)$ on doit calculer son gradient par rapport à W .

$$\frac{\partial C^k(W)}{\partial w_{ij}} = \frac{\partial C^k(W)}{\partial A_i} \frac{\partial A_i}{\partial w_{ij}} = \frac{\partial C^k(W)}{\partial A_i} x_j = \delta_i^k x_j \tag{I.28}$$

Ceci en posant
$$\delta_i^k = \frac{\partial C^k(W)}{\partial A_i}$$

- Pour les cellules i de la couche de sortie on peut réécrire δ_i^k de la manière suivante :

$$\delta_i^k = \frac{\partial C^k(W)}{\partial A_i} = \frac{\partial C^k(W)}{\partial s_i} f'(A_i) = 2(s_i^k - y_i^k) f'(A_i) \tag{I.29}$$

- Pour les cellules i ne faisant pas partie des cellules de la couche de sortie on a :

$$\delta_i^k = \frac{\partial C^k(W)}{\partial A_i} = \frac{\partial C^k(W)}{\partial x_i} f'(A_i)$$

Mais

$$\frac{\partial C^k(W)}{\partial x_i} = \sum_l \frac{\partial C^k(W)}{\partial A_l} \frac{\partial A_l}{\partial x_i} = \sum_l \delta_l^k w_{ij} \tag{I.30}$$

Où les cellules l sont "en aval" de la cellule i . On obtient alors :

$$\delta_i^k = f'(A_i) \sum_l \delta_l^k w_{ij} \tag{I.31}$$

Pour les neurones i dans les couches intermédiaires du réseau.

Enfin, selon l'équation générale d'adaptation des coefficients w_{ij} (I. 32) :

$$\Delta w_{ij}^{k+1} = w_{ij}^k + \Delta w_{ij}^k. \tag{I. 32}$$

Dont les Δw_{ij} sont données par l'expression

$$\Delta w_{ij} = \varepsilon^k \frac{\partial C^k(W)}{\partial w_{ij}} = \varepsilon^k \delta_i^k x_j \tag{I. 33}$$

La valeur du pas d'apprentissage ε^k est typiquement assez petite et décroît dans le temps. Les gradients δ_i^k sont donnés par les équations (I. 29) et (I. 31).

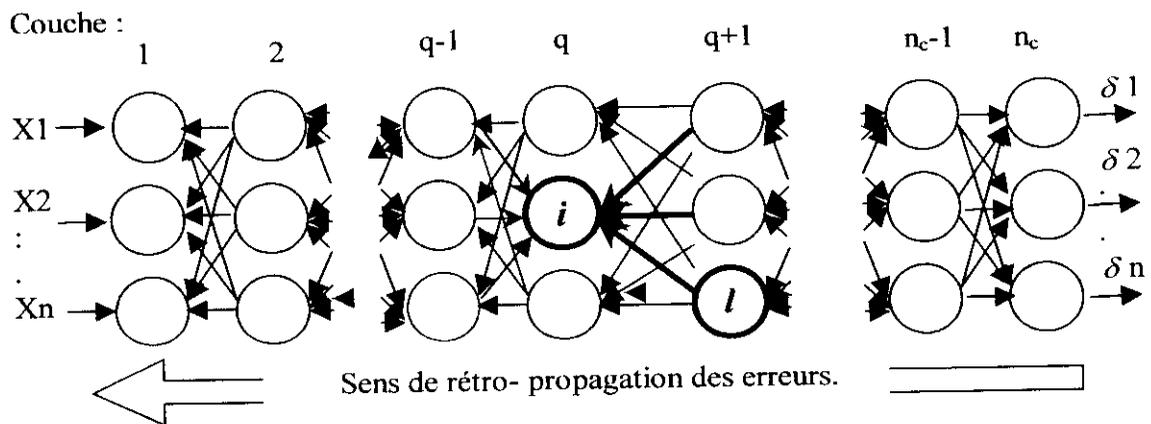


Fig.I.12. Rétro-propagation des gradients.

La rétro-propagation des gradients se fait “en amont” dès la couche de sortie vers la couche d’entrée. Le gradient de la cellule i dans la couche q est fonction des gradients des cellules l des couches supérieures $q+1$, etc.

I.7.3.2.3. BACKPROPAGATION AVEC MOMENTIUM

D.E.Rumelhart [5] a proposé une solution très efficace pour accélérer la convergence, qui consiste à utiliser les changements précédents des poids pour la réadaptation des poids actuels.

L’équation d’adaptation devient :

$$w_{ij}^l(t+1) = w_{ij}^l + \eta \Delta w_{ij}^l(t) + \alpha \Delta w_{ij}^l(t-1) \quad (I.34)$$

Le terme ajouté est appelé “momentum” (élan), en analogie avec le système mécanique classique, où un objet en mouvement garde l’élan acquis grâce à la quantité de mouvement qui lui a été communiquée précédemment pour accélérer son mouvement (Fig.I.13).

Le paramètre α est utilisé pour pondérer l’effet de ce terme. Sa valeur est généralement prise entre 0.8 et 0.9.

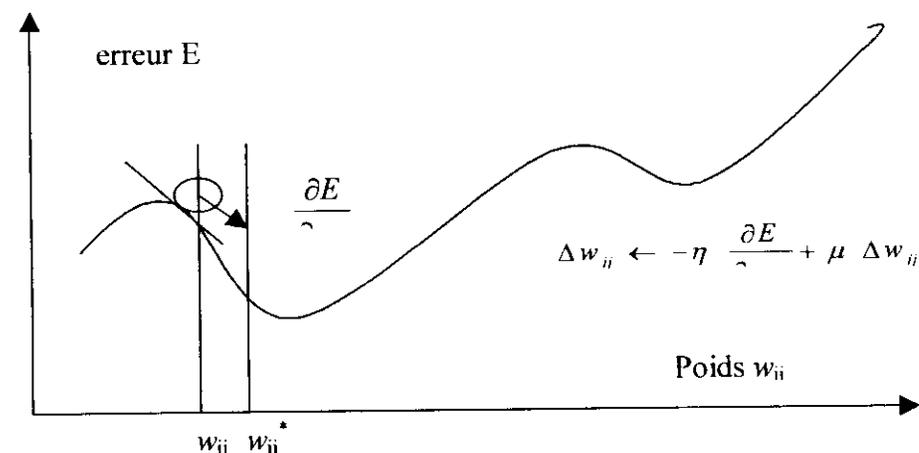


Fig.I.13. Effet du momentum pour échapper d'un minimum

I.7.4. UNE NOUVELLE ARCHITECTURE :

L’apport fondamental de ce nouveau type de réseau a été de proposer une méthode d’apprentissage efficace pour entraîner des réseaux composés d’unités non linéaires assemblés dans des architectures complexes.

I.7.4.1. LES UNITES :

Dans les Perceptrons multicouches, les unités peuvent éventuellement être linéaires comme dans l’Adaline; elles peuvent être aussi non linéaires et c’est l’intérêt de ce type d’unités. Pour des raisons à la fois historiques et pratiques, les unités employées dans la plupart des modèles neuronaux sont du type quasi-linéaire au sens où on l’a défini au paravent.

Ainsi, l’état d’une cellule i est calculé par :

$$x_i = f\left(\sum_{j=1}^n w_{ij} x_j - \theta_0\right) \quad (I.35)$$

Où f est une fonction différentiable, la plus communément employée étant celle originellement proposée dans le modèle des MLP, il s'agit de la *fonction sigmoïde* (cf. §2.2.5%); x_j représente les états des cellules connectées à la cellule i ; w_{ij} le poids de connexion entre les cellules j et i ; et θ_0 le seuil associé à la cellule i .

I.7.4.2. L'ORGANISATION DES CELLULES

L'Adaline possède des unités d'entrée et des unités de sortie. Les deux types de cellules étant accessibles par le monde extérieur et sont dites unités externes. Elles constituent respectivement l'interface d'entrée et celle de sortie du système.

Les réseaux MLP possèdent le même type d'interface, mais ont également des cellules qui ne sont ni des unités d'entrée ni des unités de sortie et qui sont appelées *cellules cachées*. L'état des cellules cachées n'étant pas accessible de l'extérieur, il est modifié en fonction des états des autres cellules du réseau. Une donnée présentée en entrée du réseau est propagée à travers

Les connexions jusqu'aux cellules cachées. De même, les états de cellules cachées sont propagés à travers, éventuellement d'autres *couches cachées*, jusqu'à la couche de sortie du réseau. La rétro-propagation du gradient permet de travailler avec un réseau à plusieurs couches de neurones, ou réseau multicouches; on parle souvent de Perceptron Multicouches. L'utilisation des automates quasi-linéaires dans l'algorithme de rétro-propagation du gradient conjuguée aux architectures multicouches va permettre à cette méthode de résoudre certains types de problèmes qu'il n'était pas possible de résoudre efficacement par les réseaux plus simples comme l'Adaline et le Perceptron.

I.7.5. METHODE D'OPTIMISATION ALEATOIRE ROM

En 1965, J. Mathias a présenté cette méthode, elle a été développée par F.J.Sollis et J.B.Wells en 1981 [5], qui ont démontré la convergence de l'algorithme vers un minimum global [8].

Nous présentons ci-dessous l'algorithme de cette méthode:

1. initialiser aléatoirement tous les poids W du réseau. Initialiser la variance v et la moyenne de la séquence aléatoire $b(o) = 0$.
2. Générer une nouvelle séquence aléatoire $\xi(k)$ de moyenne $b(k)$ et variance $v(k)$.
3. Calculer les erreurs $E = \sum_{p=1}^n E_p$ à la sortie du réseau pour chaque cas:
 - $E(w(k)) \rightarrow E_1$,
 - $E(w(k) + \xi(k)) \rightarrow E_2$,
 - $E(w(k) - \xi(k)) \rightarrow E_3$,

$$4. \text{ si } E_2 < E_1 \text{ alors } \begin{cases} w(k+1) = w(k) + \xi(k) \\ b(k+1) = 0.2b(k) + 0.4\xi(k) \end{cases}$$

$$\text{si } E_2 > E_1 \text{ et } E_3 < E_1 \text{ alors } \begin{cases} w(k+1) = w(k) - \xi(k) \\ b(k+1) = 0b(k) - 0.4\xi(k) \end{cases}$$

$$\text{Sinon } \begin{cases} w(k+1) = w(k) \\ b(k+1) = 0.5b(k) \end{cases}$$

5. poser $k=k+1$ et refaire les étapes de 2 à 4 jusqu'à l'obtention de l'erreur désirée.

Contrairement à la méthode de backpropagation où le calcul du Jacobien est nécessaire, le grand avantage de cette méthode est son indépendance du critère à minimiser, son utilisation dépend seulement de la variance du vecteur gaussien. Une séquence avec une grande variance génère de grands changements dans l'erreur mais jamais des oscillations. Il est intéressant de commencer l'entraînement avec une variance importante et de la faire baisser au fur et à mesure que l'entraînement avance.

1.7.6. RESEAUX A FONCTION DE BASE RADIALE RBF

Les réseaux de neurones à fonction de base radiale, outre les couches d'entrée et de sortie, ils comprennent une couche cachée de n unités qui ne réagissent significativement qu'à une partie restreinte dans l'espace d'entrée suivant une fonction d'activation de type gaussien[8]. A travers une combinaison linéaire de fonctions non linéaires de base radiale, le fonctionnement de ces réseaux repose sur le principe des estimateurs à noyau, et le généralise du monovarié vers le multivarié. Un estimateur à noyau considère des fonctions de R^+ vers R et sont de la forme :

$$\phi_i(x) = \phi(\|x - \xi_i\|) \quad (I.36)$$

L'idée principale a été introduite par M.J.D.Powell [20%], toute fonction $f(x)$ d'une variable x de R^n peut être approchée par une interpolation composée par la somme de P fonctions noyaux de forme fixée $\phi(\|x - \xi_i\|)$:

$$f(x) = \sum_{i=1}^P \lambda_j \phi(\|x - \xi_i\|) \quad (I.37)$$

Où ξ_i représente le nœud d'interpolation pour $i=1, \dots, n$.

λ_j sont les paramètres que l'on déterminera à partir des exemples connus x_i, γ_j en résolvant le système :

$$f(x) = \gamma_j \quad i = 1, n \quad (I.38)$$

$\phi(\|x - \xi_i\|)$ est une fonction assurant la continuité aux nœuds et la dérivabilité d'ordre supérieur en ces points [8].

I.7.6.1. ARCHITECTURE ET FONCTIONNEMENT DES RESEAUX RBF

Parmi les fonctions noyaux qui sont utilisées dans les réseaux RBF, on peut trouver les formes suivantes :

Formes cubiques : $\phi(r) = r^3$

Formes multi-quadratiques : $\phi(r) = (r^2 + k^2)^{-1}$

Formes gaussiennes : $\phi(r) = \exp\left(-\frac{r^2}{\beta^2}\right)$

Où r est un réel quelconque, remplaçant le discriminant des neurones. Mais la fonction gaussienne est la plus utilisée.

L'architecture d'un réseau RBF est représentée par la figure I. 14.

Il est composé de trois couches. La première reçoit les entrées injectées au réseau, la couche cachée est constituée de neurones, dont le discriminant est à fonction de base radiale.

Chacun de ces neurones est doté d'un vecteur C_i appelé centre, les valeurs de ces vecteurs représente chacune le barycentre de la classe qui lui correspond.

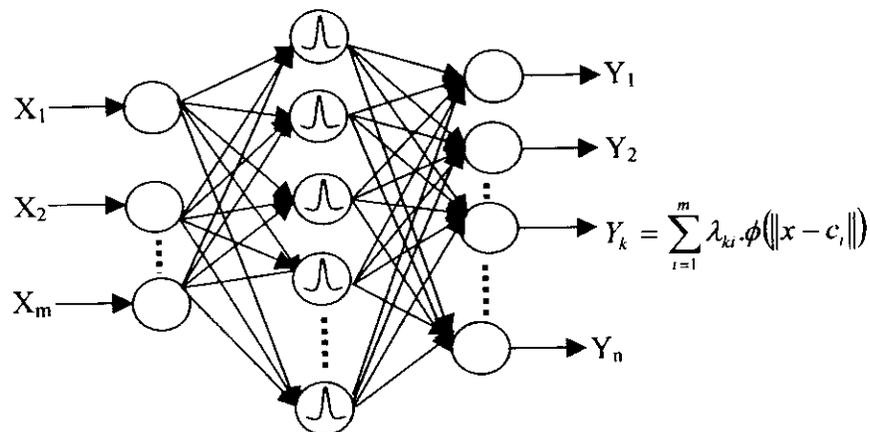


Fig.I.14. Architecture d'un réseau RBF.

Lorsqu'un vecteur X est présenté à l'entrée du réseau, chaque neurone calcule, à travers son discriminant, la distance euclidienne de cet exemple par rapport aux centres de la classe qui le représente :

$$d_i = \|X - C_i\| \quad (I. 39)$$

La sortie de chaque neurone de cette couche est donnée comme suit :

$$s_i = \exp\left(\frac{(X - C_i)^T (X - C_i)}{2\sigma_i}\right) \quad (\text{I. 40})$$

Où σ_i est un paramètre qui représente une mesure de la dispersion des données associées à chaque nœud. La couche de sortie est constituée de neurones linéaires, effectuée l'approximation définie par la somme pondérée suivante :

$$y_k(X) = \sum_{i=1}^m \lambda_{ki} \phi(\|X - \xi_i\|) \quad 1 \leq k \leq n \quad (\text{I. 50})$$

Avec n le nombre de neurones de sortie.

I.7.6.2. APPRENTISSAGE DES RESEAUX RBF GENERALISES

Ces réseaux permettent non seulement l'apprentissage des poids des connexion w , mais aussi celui des centres et des rayons de ces réseaux à base radiale généralisés. L'ajustement des centres et des rayons de ces réseaux peut être aisément effectué au moyen d'un algorithme de type descente de gradient.

Nous présentons ci-dessous les étapes de cette technique d'apprentissage supervisé :

1. initialiser les centres $c_i(0)$, les paramètres de la sortie de chaque neurone de la couche cachée $\sigma_i(0)$, $i=1, \dots, m$, et les poids synaptique $\lambda_{ij}(0)$, $j=1, \dots, n$.
2. choisir un taux d'entraînement $\alpha < 1$ variable ou fixe.
3. présentation successive de P exemples d'entraînement.

Calcul de l'erreur quadratique en sortie E entre la réponse désirée et celle fournie par le réseau.

$$E = \sum_{p=1}^M \sum_{j=1}^n (d_j^p - y_j^p)^2 \quad (\text{I. 51})$$

avec :

M représentant le nombre d'exemples d'entraînement

y_j^p , d_j^p représente respectivement la $j^{\text{ème}}$ sortie du réseau et celle désirée pour le $p^{\text{ème}}$ vecteur d'entrées.

4. Réajuster les centres et les paramètres des fonctions noyau :

$$\begin{aligned} c_i(t+1) &= c_i(t) + \alpha \Delta c_i(t) \\ \sigma_i(t+1) &= \sigma_i(t) + \alpha \Delta \sigma_i(t) \end{aligned} \quad (\text{I. 52})$$

Avec :

$$\begin{aligned} \Delta c_i(t) &= - \frac{\partial E}{\partial \lambda_{ij}(t)} \\ \Delta \sigma_i(t) &= - \frac{\partial E}{\partial \sigma_i(t)} \end{aligned} \quad 1 \leq i \leq m \quad (\text{I. 53})$$

6. réajuster les poids synaptiques :

$$\lambda_{ij}(t+1) = \lambda_{ij}(t) + \alpha \Delta\lambda_{ij}(t) \quad (\alpha \text{ est le taux d'apprentissage}) \quad (I.54)$$

Avec :

$$\Delta\lambda_{ij}(t) = - \frac{\partial E}{\partial \lambda_{ij}(t)}, \quad 1 \leq j \leq n, \quad 1 \leq i \leq m \quad (I.55)$$

6. répéter les étapes de 3. à 5, le nombre de fois nécessaire jusqu'à l'obtention de la précision désirée en sortie.

1.7.6.3. RESEAU RBF ET APPROXIMATION DE FONCTION

Les réseaux RBF constituent des modèles de réseaux très efficaces notamment pour l'approximation des fonctions. L'utilisation de la fonction Gaussienne permet de bénéficier de sa caractéristique locale pour faciliter l'apprentissage et améliorer l'approximation. Par ailleurs, la procédure d'entraînement, basée sur le fonctionnement de ce réseau, qui consiste en une classification suivie d'une optimisation rend l'apprentissage beaucoup moins difficile et plus rapide que celui des réseaux LBF. Par ailleurs, ces réseaux sont utilisés avec une seule couche cachée, ceci libère l'utilisateur du choix du nombre de couches [8].

Néanmoins, ces réseaux présentent quelques inconvénients. En effet, lorsque le nombre d'exemples d'apprentissage est important, la couche cachée risque d'être surchargée en neurones, compliquant ainsi les calculs et ralentissant l'apprentissage.

1.8. CONCLUSION

Les réseaux multicouches entraînés par l'algorithme de rétro propagation du gradient sont les modèles connexionnistes les plus étudiés et utilisés à ce jour.

Les champs d'application de cette méthode sont très vastes tels que la prédiction, classification, l'identification de processus de sire temporelles, la commande de processus et de robot, le traitement d'image et de parole.

Il est clair que le réseau apprend les exemples de base d'apprentissage, dont les réponses désirées sont élaborées par le modèle que constitue le superviseur. Si les exemples de la base de données concernent un fonctionnement dans un domaine réduit, le réseau ne saura pas répondre correctement en dehors de ce domaine. Si le superviseur donne des informations erronées le réseau apprendra un modèle erroné.

Si le processus que l'on cherche à modéliser et non stationnaire ou change brutalement, le réseau doit répondre au bon modèle.

D'un point de vue théorique, on sait qu'il existe toujours un réseau à trois couches capable d'approcher toute fonction continue.

Cependant, pour un réseau donné, on ne connaît pas le nombre optimum de neurones par couches. De plus, on ne sait pas quel est le nombre de couches du réseau de complexité minimale, en nombre de paramètres (connections).

Un nombre de neurones trop petit induit une modélisation insuffisante, mais un trop grand nombre de neurones entraîne une sur paramétrisation du modèle, qui nuira aux performances en généralisation. La taille du réseau et celle de la base d'apprentissage son liées.

D'un point de vue pratique, la principale difficulté est l'optimisation de la phase d'apprentissage. Le choix de l'architecture adéquat ou la détermination du « pas d'apprentissage » se fait par essais successifs. L'utilisation d'une base indépendante de celle d'apprentissage, appelée base de test permet de déterminer le réseau optimal. On détermine les poids du réseau à partir de la base d'apprentissage et on calcule les performances sur la base de test. Le réseau « optimal » est celui qui minimise l'erreur commise sur la base de test.

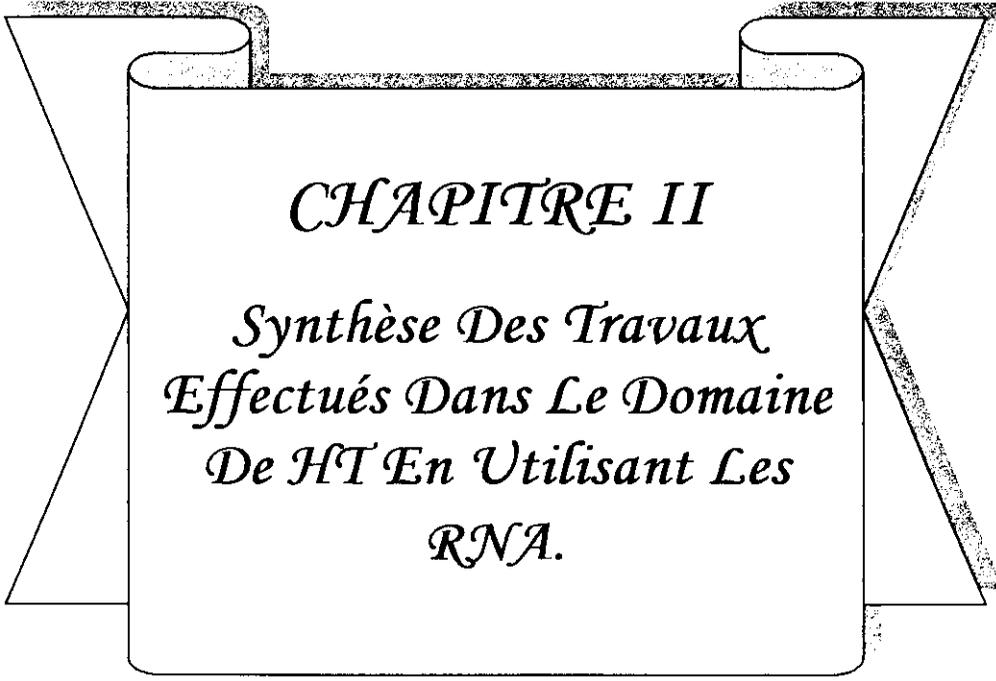
Une autre difficulté est liée aux caractéristiques de la base d'apprentissage, aussi bien en terme de taille et de représentativité que de répartition des exemples. Le nombre d'exemples doit être suffisamment grand devant les paramètres (les poids) à déterminer. Le domaine de validité de l'algorithme neuronal est directement lié à la représentativité des exemples de la base d'apprentissage. L'irrégularité des exemples dans certaines régions de l'espace des données peut conduire à une mauvaise optimisation du réseau. Dans ces conditions, un soin particulier doit être appliqué lors de la création des bases de données; une méthode d'échantillonnage apparaît primordiale.

L'algorithme d'optimisation aléatoire ROM, constitue une alternative très intéressante. En effet, cet algorithme ne dépend pas du système pour lequel le réseau est utilisé. De plus, sa technique de recherche permet d'atteindre un minimum global.

Les réseaux LBF souffrent de l'indétermination du nombre optimal d'éléments dans chaque couche. Mais concernant la dimension du réseau, il est clair qu'un réseau à deux couches constitue un approximateur universel.

Les réseaux RBF se distinguent par leur caractéristique de meilleurs approximateurs.

En effet, ces réseaux sont dotés de la caractéristique de calcul et d'adaptation paramétrique locaux. Ceci fait que ces réseaux soient à l'abri de plusieurs problèmes que connaissent les LBF, comme le sur-apprentissage, et la difficulté d'introduction de nouveaux exemples sans toucher à la précision d'apprentissage des exemples précédemment appris.



CHAPITRE II

*Synthèse Des Travaux
Effectués Dans Le Domaine
De HT En Utilisant Les
RNA.*

Dans ce chapitre nous présentons une synthèse sur quelques travaux ayant utilisé les réseaux de neurones artificiels dans le domaine de la haute tension. Ici, les réseaux de neurone ont été utilisés pour résoudre plusieurs problèmes de différents types : identification, classification, modélisation, etc.

II.1. APPLICATION DES RNA POUR LA DETECTION DE LA DEFORMATION DES ENROULEMENTS DE TRANSFORMATEUR

Dans ce travail [10] l'auteur a présenté l'application des RNA à la méthode de réponse fréquentielle FRP (Frequency Response Analysis) pour la détection de déformation des enroulements de transformateur.

Le coefficient de corrélation et l'écart type sont choisis comme des paramètres propres de chaque section de fréquence. A cause de la structure des enroulements, la distribution des pôles de la fonction de transfert est différente. Et dans le but de refléter l'état des enroulements par ces paramètres propres, la division de la gamme de fréquence doit être raisonnable.

Dans de cette méthode, huit paramètres propres sont choisis comme modèle d'entrée des RNA, donc le vecteur d'entrée est un vecteur à huit dimensions et la valeur des neurones dans la couche d'entrée est donc huit. Le choix de sortie dans ce problème est vraiment simple, une sortie binaire est suffisante pour indiquer que la bobine est dans un bon état ou non, la valeur 0 indique 'NORMAL' et 1 indique 'ALARM', donc le nombre de neurones dans la couche de sortie égale un neurone. La sélection de la couche cachée est importante à partir des essais et des erreurs. Cependant, il a observé que dans la plupart des applications une couche cachée est suffisante. Pour obtenir le nombre convenable des neurones dans la couche cachée, le réseau est entraîné avec différents nombres des neurones dans cette couche avec une erreur de 0.5 % durant l'apprentissage. Le nombre de neurones dans la couche cachée est choisi égal à 5.

Pendant que le réseau est entraîné aussi bien par les anciennes entrées que par les nouvelles entrées à la fois, le pourcentage des réponses correctes est 95%, c'est à dire que ce réseau est capable de détecter la déformation des enroulements avec un grand degré de précision, et il est important de rassembler les types des entrées pour entraîner continuellement le réseau, ainsi le réseau peut posséder de plus en plus d'expériences pour améliorer la capacité de reconnaissance.

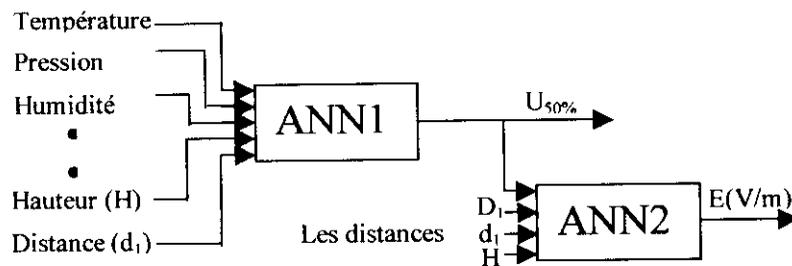
Les résultats montrent cet algorithme est capable de distinguer les enroulements en bon état des enroulements défectueux avec différents types et dans différents positions, et ainsi à établir avec succès l'efficacité de la méthode proposée. Donc pour les enroulements des transformateurs modernes avec déformation cachée, la méthode FRA serait plus effective. Cette méthode est simple, robuste et facile pour mettre en œuvre.

II.2. ANALYSES DE CONCEPTION DES SUBSTANCES DE HT UTILISANT LES RNA

L'intensité du champ électrique et la tension disruptive critique dans les substances sont en fonction de plusieurs facteurs atmosphériques et géométriques, comme la pression, la température, l'humidité, distance entre phases, hauteurs des jeux de barres et la forme d'onde. Dans ce travail [11], les RNA multicouches statiques sont utilisés pour représenter des relations entre des variables associées avec le processus de spécification des substances.

La structure de base contient une seule couche cachée, elle apparaît assez puissante pour faire une organisation arbitraire entre les variables. Le RNA est entraîné par la représentation successive des couples entrées-sorties. Durant l'apprentissage, l'information se propage en avance dans le réseau qui va ajuster ses poids pour minimiser une fonction coût (l'erreur entre la sortie produite par le réseau et la sortie désirée). On a utilisé pour cette minimisation l'algorithme de la Backpropagation.

L'architecture générale du réseau de neurones utilisée est donnée dans la figure II.1.



FigII.1. Architecture du réseau

Cette architecture est composée de deux réseaux de neurones. Le premier RNA, ANN1, calcule $U_{50\%}$, tension disruptive. On note que ce réseau prend en considération plusieurs facteurs géométriques et atmosphériques.

Le deuxième RNA, ANN2, calcule l'intensité du champ électrique entre les jeux de barres. Comme le montre la figure II.1, la sortie produite par le réseau ANN1 est utilisée comme paramètre d'entrée pour le réseau ANN2. Cependant, tous les facteurs atmosphériques et géométriques sont aussi pris en considération dans le calcul de l'intensité du champ électrique.

Donc, dans ce travail, une nouvelle méthodologie pour la conception des substances en utilisant les RNA a été présentée. Les RNA sont applicables dans le contexte de l'identification dans le domaine de la haute tension. L'apprentissage du réseau de neurone est effectué en utilisant les informations concernant les conditions atmosphériques et les paramètres géométriques tirés des travaux expérimentaux.

Après l'apprentissage, les réseaux de neurones deviennent capables de généraliser ce qu'ils avaient appris pour de nouvelles entrées qui n'ont pu être effectuées au laboratoire, ce qui permet de diminuer le travail expérimental et donc gagner du temps.

II.3. OPTIMISATION DE CHAMP ELECTRIQUE D'UNE ELECTRODE DE HT BASE SUR LES RNA

Dans ce travail [12], une technique très efficace d'optimisation du champ électrique basée sur les réseaux de neurones artificiels (RNA) est présentée. On a introduit la méthode de backpropagation pour effectuer l'apprentissage avec la fonction sigmoïde comme fonction d'activation pour diminuer l'écart entre la sortie désirée et celle fournie par le réseau de neurones. Pour avoir de meilleurs résultats, les données sont normalisées par rapport à la valeur moyenne et l'écart type.

Le RNA dans cette étude doit apprendre la relation entre la valeur du champ électrique, le contour et la distance inter électrode. Le réseau responsable de cette première étape est nommé NN1. Cependant, l'objectif de cette étude est d'avoir une configuration optimale du contour de l'électrode, d'où la nécessité d'un autre réseau de neurones chargé d'apprendre la relation entre le contour et la distance inter électrodes et le contour optimal, c'est le réseau NN2. A la fin de l'apprentissage, par l'introduction de la distribution désirée du champ électrique à l'entrée du réseau, on peut obtenir une distance inter électrodes et une configuration du contour de l'électrode.

Donc, deux réseaux de neurones NN1 et NN2 sont préparés. NN1 apprend la relation entre le champ et le contour de l'électrode, NN2 apprend la relation entre le contour et les coordonnées de points. Une fois l'apprentissage de NN1 terminé, la relation entre le champ et la courbe est obtenue. Après, on passe à l'apprentissage de NN2.

En faisant introduire la distribution désirée du champ à NN1, on peut obtenir le contour et la distance inter électrodes du contour optimal. En présentant ces deux derniers paramètres comme entrées pour NN2, on obtient la configuration du contour optimal.

Ce travail a montré que l'introduction de la méthode des réseaux de neurones a prouvé une efficacité remarquable pour la technique d'optimisation du champ électrique. Donc avec cette technique, nous obtenons une distribution du champ électrique uniforme, une fois le réseau de neurones a appris les données que nous avons sur le contour de champ. La méthode de normalisation de données a augmenté l'efficacité de l'approche proposée dans cette étude.

II.4. OPTIMISATION DES SYSTEMES DES ELECTRODES DE HT PAR RNA

L'étude d'électrode et de contour de l'isolateur dans un appareil ou un système de H.T est principalement basée sur l'approche de l'itérative. Dans ces méthodes, le contour est changé itérativement pour obtenir la distribution désirée du champ, et le temps de calcul est habituellement grand parce que le champ doit être calculé à chaque étape d'itération.

Dans l'article [13], l'auteur a étudié l'optimisation de deux systèmes d'électrodes de HT en utilisant les RNA. Ces réseaux sont basés sur l'algorithme de backpropagation, mais

l'algorithme de backpropagation classique est lent. Dans cet article, une nouvelle méthode d'apprentissage a été appliquée pour l'optimisation de champ électrique qu'elle est plus rapide que la méthode classique. Les contours des électrodes ont été pris comme étant un quart ou un demi ellipse, parce que la forme elliptique est plus flexible que la forme circulaire. Bien que ce dernier soit changé par trois paramètres, les coordonnées du centre et le rayon, la forme peut changer avec l'emplacement de centre et les deux distances axiales.

Les deux exemples qui ont été optimisés sont :

- La borne de GIS monophasé
- L'anneau de protection d'un transformateur triphasé

La méthode de simulation de charge CSM (the Charge Simulation Method) a été utilisée pour le calcul des champs, pour la préparation des ensembles d'apprentissage ainsi que pour contrôler les sorties désirées du réseau.

Le RPROP (Resilient PROPagation) est un nouvel algorithme d'apprentissage qui considère la topologie locale de la fonction erreur E . L'adaptation des poids ou leurs modifications dépend des dérivées de E et pas de sa valeur. L'adaptation des poids se base sur ce qu'on appelle la règle d'apprentissage de Manhattan ;

Puisque les variables d'entrées et de sorties du RNA ont différentes gammes, si les données originales sont fournies au réseau on peut avoir un problème de convergence, il est évident que le sortie du RNA doit faire partie de l'intervalle $(0...0,1)$. En outre, des signaux d'entrées devraient être maintenus petits afin d'éviter l'effet de saturation de la fonction sigmoïdale. Ainsi les modèles d'entrée- sortie sont normalisés avant l'apprentissage du réseau. La normalisation par rapport au valeur moyen et l'écart type est trouvé plus approprié que la normalisation par rapport à la valeur maximale, parce que l'ancien donne une distribution plus diverse et la formation est plus efficace .

Dans ce travail, les contours des électrodes ont été portés comme quart ou demi ellipse à cause de sa flexibilité que les formes circulaires. Les dimensions optimales, la position du centre et les longueurs de deux rayons ont été déterminés par le RNA basé sur l'algorithme RPROP. Dans les deux exemples, le réseau a deux couches cachés avec 8 neurones chacune. L'utilisation de RNA réduit le nombre de calculs dans l'optimisation. L'algorithme RPROP a été trouvé pour être plus rapide que l'algorithme de Backpropagation standard et le nombre d'itération peut être réduit de 60 à 70% moins que le nombre d'itérations avec la méthode classique, mais le temps d'exécution peut être considéré le même pour les deux méthodes. Cependant dans un RNA, on doit choisir d'une manière judicieuse les entrées et les sorties.

II.5. MODELISATION DE LA TENSION DE CONTOURNEMENT D'UN ISOLATEUR POLLUE EN UTILISANT LES RNA

Dans un autre travail [14], il a étudié la modélisation de la tension de contournement d'un isolateur pollué en utilisant les RNA avec l'algorithme de RPROP. Les modèles théoriques qui calculent la tension de contournement d'un isolateur pollué sont utiles pour au moins deux raisons :

- 1) Une meilleure compréhension du processus physique responsable du contournement.
- 2) Si le modèle peut être indiqué comme étant mieux correspondent le mieux aux modèles expérimentaux de laboratoire, ceci peut être utilisé pour la prédiction des tensions de

contournement avec une large gamme de condition de champ et de cette façon réduire le nombre des essais nécessaires au laboratoire, il est bien connu que les considérations théoriques et les recherches expérimentales dans le contournement avec pollution est cependant à un bon modèle mathématique qui peut déterminer de façon précise la tension de contournement critique V_{ec} , mais le but de cette étude est de modéliser V_{ec} de manière précise en se basant sur un RNA, en remplacement de toute approche empirique. En considérant l'effet de la résistance par unité de longueur R_p , longueur de la couche polluée L , et l'épaisseur de la surface électrolytique w .

Dans Ce travail, il a utilisé un RNA comme un estimateur de fonction pour modéliser de façon exacte la relation de V_{ec} avec les paramètres R_p , L et w de la couche polluée.

La modélisation de $V_{ec} = f(R_p, L, w)$ à l'aide des données expérimentales a été fait en utilisant l'algorithme RPROP au lieu de quelques approches théoriques. Durant le processus d'apprentissage, 44 vecteurs donnés d'entrée-sortie vont être présentés. Chaque vecteur d'apprentissage contient 3 composantes (3 neurones d'entrée), caractérisant les paramètres physiques : R_p, L, w , et un neurone de sortie correspondant à la valeur de V_{ec} . Toutes les variables d'entrée et de sortie sont normalisées avant d'être utilisées à l'apprentissage et tester le réseau. Une fois l'apprentissage du réseau est effectué en utilisant 44 vecteurs, on le teste avec 4 vecteurs sélectionnés aléatoirement.

L'auteur de ce rapport a trouvé que la modélisation de $V_{ec} = f(R_p, L, w)$ est effectuée en utilisant les RNA multicouches avec algorithme d'apprentissage RPROP. L'utilisation de cette méthode a donné de meilleurs résultats que d'autres approches analytiques et empiriques.

II.6. INFLUENCE DU DEVELOPPEMENT DU DEFAUT DE LA DP SUR LA RECONNAISSANCE DU TYPE DE DEFAUT EN UTILISANT LE RNA

Les mesures des Décharges Partielles ont une grande importance dans l'assurance de la qualité des isolations électriques. Les résultats de mesure montrent des courbes typiques pour des défauts individuels de la DP. De cette manière, il doit être considéré que les changements de l'activité de la DP durant le vieillissement causés par exemple par la formation d'une conductivité de surface, dégradation de l'isolation, arborescence électrique, déterminent les données mesurées et leur classification. Pour analyser ces influences sur la classification de la DP le modèle de Kohonen est utilisé dans ce travail [15] pour classifier les défauts de la DP, qui sont partiellement influencés par des paramètres tels que la pression, la conductivité de la surface et la dégradation de l'isolation.

Dans un examen récent, des réseaux de neurones sont utilisés avec succès pour la discrimination automatique du développement des dommages et la reconnaissance de la source de la DP. Pendant la première application, le changement de l'activité de la DP est l'exigence de base pour la description des dommages, dans la deuxième la reconnaissance du défaut de la DP peut être obstruée par cette variation. Par conséquent, la classification des différents défauts de la DP est analysée par la carte d'auto-organisation de Kohonen en prenant en considération l'influence de la dégradation de l'isolation, la formation de la conductivité de surface et la pression de gaz.

L'applicabilité de la carte d'auto- organisation de Kohonen pour distinguer les sources de DP est prouvée.

La compression des données est faite en utilisant des procédures statistiques de post traitement; les quantités déduites de DP sont les composants des vecteurs de classification, formant la matrice d'entrée pour le réseau de neurones.

Les résultats montrent le changement d'activité DP par arborescence, conductivité de la surface et variation de la pression. L'investigation séparée de chaque paramètre prouve son influence sur la classification qui utilise la carte d'auto- organisation de Kohonen. Néanmoins la classification entre les différents défauts de DP paraît encore être possible, même si ces défauts sont influencés par paramètres mentionnés au-dessus.

II.7.IDENTIFICATION DES DP EN UTILISANT LA MESURE DE L'EMISSION ACOUSTIQUE ET RNA

Les décharges partiels (DP) son l'un des principaux facteurs de la dégradation d'isolation et qui parfois conduit au claquage. Dans ce travail [16], l'auteur a étudié l'identification des décharges partielles en utilisant la mesure de l'émission acoustique et les réseaux de neurones artificiels (RNA). Il est vraiment nécessaire de maîtriser les conditions d'apparition des DP et de localiser ces décharges. Les techniques de l'émission acoustique (EA) peuvent êtres utilisées pour la détection des DP, et elle a prouvé son efficacité. EA est basée sur la détection des ondes produites par la déformation brusque dans le matériel sous tension. Ces ondes sont transmises de la source vers le(s) sonde(s) qui ont le rôle de les convertir en signaux électriques, cette approche n'est pas touchée par le phénomène d'interférence électrique. La technique de l'EA est capable non seulement de définir l'emplacement de la décharge mais aussi le type de DP dans les câbles de haute tension. L'identification des DP par la mesure de AE, peut être obtenir par l'utilisation des réseaux de neurones artificiels, cette méthode a prouvé son efficacité dans cette application.

Une étude préliminaire sur l'identification de deux types de DP est présentée. Il s'agit des décharges internes et des décharges superficielles en utilisant trois configurations de RNA utilisés pour l'identification des décharges partielles : avec rétro propagation (BP), Kohonen et LVQ. Tous ces réseaux sont constitués d'une seule couche d'entrée, et une seule couche cachée, et une couche de sortie. Le processus d'identification nécessite deux étapes : une phase d'apprentissage ; et une phase d'identification en utilisant les signaux de l'émission acoustique et leur spectre de fréquence comme informations d'apprentissage des ces réseaux RNA.

Les taux d'identification pour chaque réseau en utilisant les deux domaines, temporelle et fréquentielle comme entrées respectivement sont déterminés. Les entrées du réseau se regroupent en 10 ensembles de signaux de l'EA pour les décharges internes, et 10 ensembles de signaux pour les décharges superficielles. Donc les entrées de réseau englobe 20 ensembles en total pour chaque type (temporelle ou fréquentielle).

Le spectre de fréquences est obtenu par la méthode de FFT (Fast Fourier Transform). Le signal origine est échantillonné avec 1MHz par période de 20 ms. La couche cachée du réseau de neurones est constituée de 10 éléments dans le cas du rétro propagation, la fonction d'activation est la fonction sigmoïde. Dans le cas du LVQ nous avons 4 neurones cachés,

l'utilisation de spectre de fréquence comme une entrée de réseau a abouti à un meilleur taux d'identification.

A la fin, l'auteur a présenté une comparaison entre 4 procédures d'apprentissages ; avec 10 ensembles des signaux d'EA (5 pour les décharges internes et 5 pour les décharges superficielles), et avec 20 ensembles des signaux d'EA (10 pour les décharges internes et 10 pour les décharges superficielles) puis 30 et 40. Les résultats sont présentés dans le tableau II.14.

Les résultats montrent l'efficacité d'utiliser le spectre d'énergie pour distinguer les différents types des décharges partielles. Les résultats montrent aussi que les meilleurs résultats d'identification sont obtenus en utilisant le spectre d'énergie comme type de donnée d'entrée, ça augmente la vitesse d'apprentissage du réseau mais peut diminuer le taux d'identification. D'autre part on a constaté que le taux d'identification est lié au nombre de neurones dans la couche cachée du réseau, le nombre de cycles d'apprentissage et la valeur du critère d'arrêt de l'algorithme d'apprentissage. Pour le temps d'apprentissage il est influencé par le type du modèle du réseau de neurones choisi d'une part et de l'algorithme et le critère d'apprentissage d'arrêt d'autre part. Après cette étude on peut confirmer l'efficacité d'utilisation des réseaux de neurones artificiels pour l'identification des DP.

II.8. RECONNAISSANCE DES DECHARGES PARTIELLES BASEES SUR LES RESEAUX DE NEURONES ARTIFICIELS

La caractéristique de décharge partielle (DP) est l'une des propriétés importante pour un appareil électrique, la mesure de données (quantité de décharge q , angle de phase Φ et le taux de répétition d'impulsion n) contient beaucoup d'informations sur la DP.

L'auteur [17] a étudié l'effet de la structure des RNA et les données de sur la dispersion de DP, il a utilisé la méthode de Monte-Carlo pour la simulation de DP. Cette méthode a été utilisée par Hikit et al en 1990 pour la simulation de DP dans les gaz.

Quatre modèles expérimentaux sont utilisés pour simuler les 4 types de DP. Ces modèles sont soumis à la tension alternative entre 3.5-25kv de 50Hz.

L'auteur a étudié aussi l'effet de plusieurs paramètres sur la sûreté de reconnaissance de DP, ces paramètres sont :

- 1 L'intervalle de donnée pris.
- 2 Nombre de neurones dans la couche d'entrée.
- 3 Nombre de neurones dans la couche cachée.

Il a abouti aux résultats suivants :

- 1) Les données simulées de DP réalisées par la méthode de Monte-Carlo pourraient être produites autant que nécessaire et sans compter que le dispersité statistique, la répétition d'elles était assez. Ces données ont pu être employées d'une manière satisfaisante pour étudier les capacités d'identifier la reconnaissance d'ANN.
- 2) si les données de DP de courbe en 3D sont utilisées comme vecteurs d'entrée de RAN, il est préférable que le nombre de neurones L de la couche d'entrée est entre $12 \times 10 - 18 \times 20$, le nombre de neurones N de la couche de sortie est égal à celui du nombre de prototypes et le nombre de neurones dans la couche cachée doit être respecté $L < M < N$; et si $N=4$ alors M sera entre 10-20.
- 3) si P cycles de tension d'acquittions de donnée de DP était inférieur à 10, la prédiction de réseau sera manquée, et quand P est plus grand que 20-100 le prédiction du réseau est

satisfaisante.

4) quand une donnée réglée pour chaque type de 4 modèles est prise comme échantillon d'apprentissage la confiance de RNA est qualifiée à identifier le modèle obtenu dans le même état que des prototypes était plus de 0,92.

5) Si deux données réglées de chaque type de 4 modèles sont prises comme échantillon d'apprentissage, quand les tensions appliquées sont compris entre les tensions d'apprentissage, la sûreté d'apprentissage de RNA pour reconnaître le même type du modèle sont plus que 0.17 et quand les tensions appliquées sont à l'extérieur des tensions d'apprentissage la sûreté de reconnaissance le même type du modèle est moins de 0.87.

II.9. UNE NOUVELLE APPROCHE POUR LA CONTROLE DE CONDITION DES LIGNES AERIENNES A HAUTE TENSION EN UTILISANT LES RESEAUX DE NEURONE

Dans ce rapport, l'auteur [18] a présenté une nouvelle approche de contrôleur de condition "Condition Monitoring" (CM) d'une ligne arienne de haute tension en utilisant les réseaux de neurones artificiels. Les lignes ariennes sont les plus efficaces à la vue de leurs prix, et elles sont très utilisées dans les réseaux électriques. La plus importante quantité d'électricité consommée en Angleterre est transportée par les lignes ariennes de 275 kV et 400 kV. Elles sont exposées aux détériorations de l'environnement, et si on n'intervient pas rapidement (détecter et localiser les détériorations), on abouti à de problèmes graves. Donc il faut chercher des méthodes efficaces pour identifier les pannes sous conditions de service.

Dans ce sens, plusieurs travaux ont montré que les DP associées aux conducteurs défectueux dans les lignes ariennes de transport peuvent être modélisées en utilisant une technique mathématique basée sur une approche empirique. Les résultats de cette technique mathématique sont comparés aux tests effectués au laboratoire faits avec n conducteurs typiques de 400 kV défectueux, les résultats montrent une corrélation entre les deux familles de résultats.

Dans cette étude, une nouvelle technique de CM pour les lignes de transport ariennes basée sur le bruit fréquentiel associé au phénomène de DP dans les lignes est présentée. Ainsi, une récente technique, pour le développement d'un nouveau plan de protection, est utilisée pour la mesure du bruit de haute fréquence dans les lignes de transport soumises à des décharges partielles.

L'étude des DP dans les lignes ariennes défectueuses a prouvé que ce phénomène dépend dans un premier lieu de la protubérance que présente la surface défectueuse. Le niveau de décharges partielles augmente en conduisant à des régions de tension à gradient élevé. On abouti à une configuration en pointe, et un courant de fuite avec une grande valeur et un taux de réapparition de pulsations de DP un peu grand. Une telle panne dans la ligne peut conduire à une forme d'onde simple, double ou multiple, tout dépend de la nature de la panne.

En se basant sur les trois types d'ondes, un programme de mesure des signaux des décharges partielles à la fin de la ligne a été développé. Le réseau considéré est une ligne typique de 400kV.

Le rôle d'une MC peut se répartir en trois sections qui représentent les fonctions de base; détection, classification et la localisation des DP. La première section détermine quand une décharge partielle peut se présenter dans le réseau. Une fois on a détecté une DP, la deuxième

section détermine le type de cette décharge. La dernière section intervient pour localiser cette décharge.

Les entrées de la section de détection sont les tensions de trois phases; va, vb, vc après une petite combinaison pour avoir va- vb, vb- vc, vc- va. Cette configuration des entrées élimine l'effet des signaux supplémentaires dans la ligne. Le taux échantillonné utilisé est de 200 kHz qui va supprimer toutes les composantes fréquentielles supérieures par un filtre. Les signaux sont digités à travers un convertisseur analogique/digital (A/D). L'extraction de caractéristiques est extrêmement importante pour n'importe quelle technique de reconnaissance.

Sachant que les phénomènes transitoires présents dans le réseau dus aux DP contiennent une large échelle de composantes fréquentielles, il est très pratique d'utiliser les formes d'ondes en domaine du temps directement comme entrées du RNA. D'où certaines caractéristiques parmi celles identifiées devant être extraites pour représenter entièrement l'état de la propagation des DP sur la ligne de transport. Néanmoins, un nombre d'algorithmes de sélection de caractéristiques existe mais qui ne sont pas applicables dans cette étude.

Une méthode basée sur la décomposition des fréquences est utilisée ici. Dans ce sens, la bande de fréquence est coupée en petits intervalles pour commencer à extraire la caractéristique et tester le réseau de neurones. Si ça devient inefficace, la bande de fréquence est recoupée en grand nombre d'intervalles avec de petite taille. Après une série d'études en utilisant l'approche de décomposition, on choisit le domaine de fréquence pour chaque cas. Après l'approche d'extraction on fixe le nombre des entrées du réseau de neurones à 18 entrées correspondantes aux 3 signaux : va- vb, vb- vc, vc- va.

L'aspect le plus important dans l'apprentissage du RNA est d'établir la meilleure topologie pour la reconnaissance d'une DP et localiser son apparition sur la ligne. Dans cette étude on considère 9 types de DP. Après la détection d'une telle DP, on la reconnaît (on donne son type), puis on la localise (sur quelle phase elle se trouve). Donc les deux problèmes de classification et de la localisation sont indépendantes l'un de l'autre, d'où un seul réseau de neurones ne suffit pas pour résoudre le problème complètement. Afin de construire un réseau de neurones, les entrées et les sorties doivent être définies pour les vecteurs d'apprentissage. Les sorties du réseau représentent les variables: polarisation simple (SP), polarisation multiple (MP), A, B, et C qui indiquent le type de DP et à quelle phase elle s'applique.

Le RNA choisi pour effectuer la classification dans cette étude est le Perceptron multicouches. Ce réseau est constitué de 18 entrées et six sorties. Le nombre de neurones dans la couche cachée ne peut être fixé dès le début. Après un nombre suffisant de tests, on a abouti à choisir un nombre de neurones cachés égal à 12. Comme algorithme d'apprentissage on a sélectionné la rétro-propagation du gradient pour la réadaptation des poids de connexions, en minimisant l'erreur entre les valeurs désirées et les valeurs calculées par le réseau pour tous les vecteurs d'apprentissage.

Il faut un réseau de neurones séparé pour chaque type de DP. Ils dérivent tous du réseau de la classification des DP et les signaux d'entrée sont générés de la même façon. On a 9 réseaux, pour chaque type de DP, un réseau de neurones associé pour localiser l'apparition de la DP. Les 9 réseaux ont la même architecture ; 18 entrées et une sortie pour indiquer la phase où elle apparaît la DP. La différence entre les architectures des 9 réseaux destinés à la localisation réside dans le nombre de neurones cachés, il change d'un réseau à l'autre.

Les données d'apprentissage sont les types des DP et leur emplacement. Terminé l'apprentissage, on teste les performances de notre réseau avec des données de la même façon que celles de l'apprentissage mais qu'on n'a pas précédé l'apprentissage.

Comme résultat principal dans ce travail, une nouvelle technique de Conditionnement Momentané (CM) pour les lignes aériennes de transport est présentée, une méthode basée sur les réseaux neurones artificiels. Les résultats obtenus du réseau de classification sont très satisfaisants et le réseau est capable de classifier divers types de DP et localiser leurs apparitions. Après la classification, les réseaux de localisation ont effectué leur rôle avec succès, on a réussi à localiser tous les types de décharges partielles.

II.10. APPLICATION DE RNA POUR LA DETECTION DES DEFAUTS DANS L'ENROULEMENT DE TRANSFORMATEUR

Dans ce rapport, l'auteur [19] a fait l'application des réseaux de neurones pour localiser les défauts dans l'enroulement de transformateur. Il a proposé un modèle expérimental de 1/5 du modèle actuel du transformateur 275kV 300MVA, avec 14 sections de bobinage. Le défaut artificiel est simulé par un interrupteur à transistor fluteur pour le court-circuit entre les 14 voisines sections.

Ici, l'auteur a utilisé un Perceptron comme un réseau de neurone, qui se compose de 14 neurones pour la couche d'entrée, 4 pour la sortie et 28 pour la couche cachée, 14 points sont tirés à partir de spectre de fréquence.

Le nombre de neurones dans la couche cachée n'est pas défini sans aucun doute théoriquement, mais il a trouvé que 28 neurones sont bien convenables pour l'apprentissage, et d'atteindre une erreur de 0.0001, avec l'algorithme de rétro-propagation d'erreur pour un nombre maximum de 10 000 itérations.

II.11. APPLICATION DE RESEAU DE NEURONE DANS LE DOMAINE VEILLISSEMENT THERMIQUE D'ISOLATEUR DE CABLE DE HAUTE TENSION

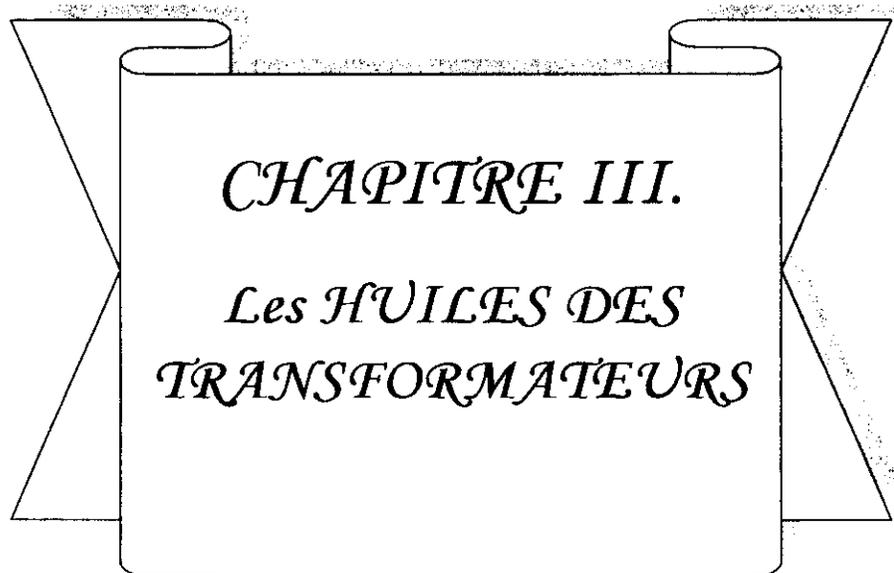
Dans ce rapport, l'auteur [1] a étudié l'application des réseaux de neurone dans le vieillissement thermique des câbles de haute tension isolés au Polychlorure de Vinyle (PVC), les températures utilisées pour le vieillissement du câble sont 80°C, 100°C, 120°C et 14°C et le temps maximal de vieillissement est 5000 heures. Cette durée est suffisante pour obtenir la dégradation des propriétés diélectrique pour les hautes températures appliquées, mais pas pour les températures inférieures qui nécessite un plus long temps.

Les expériences ont été effectuées en respectant ce qui concerne la recommandation CEI. Ces objectifs est de réduire le temps d'expérience du vieillissement, mais d'obtenir les mêmes résultats comme dans le cas de longue durée d'essais, on utilisant les réseaux de neurones artificiels.

L'auteur et après le teste de différents types des réseaux de neurones a abouti à des excellents résultats, il a choisi un réseau de neurone à base radiale comme fonction d'activation et pour l'apprentissage, l'algorithme de optimisation aléatoire (Random

Optimisation Method) avec un temps d'apprentissage inférieur de 2000 h, le réseau a prédit les mêmes caractéristiques non linéaire obtenus dans les essais avec un erreur relative de 5%.

Il a conclut que les réseaux de neurone artificiel à base radiale a apprentissage d'optimisation aléatoire consiste une très performance outil pour la prédiction des fonction non linéaire et en particulière celle de PVC en fonction de temps de vieillissement.



CHAPITRE III.

*Les HUILES DES
TRANSFORMATEURS*

III.1. INTRODUCTION

Les huiles isolantes sont largement utilisées comme diélectrique dans l'industrie d'électrotechnique à haute tension, et plus précisément les huiles minérales qui sont utilisées dans l'élaboration de l'isolation et le refroidissement du matériel électrique en service comme les transformateurs, les cabales, les disjoncteurs et les condensateurs.

Durant l'exploitation, les huiles se détériorent progressivement à cause des différentes contraintes. Cette dégradation au cours du temps est connue sous le nom de vieillissement.

Dans cette partie, nous présentons quelques notions et définitions qui concernent les diélectriques liquides. Ainsi que quelques caractéristiques d'une des huiles de transformateur utilisée par la SONELGAZ et dénommée BORAK22.

III.2. LES HUILES MINERALES ISOLANTES

Depuis 1938, les bruts naphthéniques paraissent bien adaptés à la production des huiles minérales isolantes. Elles représentent la fraction lourde de pétrole brut, et sont constituées donc de mélanges très complexes de composés appelés hydrocarbures [2].

III.2.1. LES ROLES D'UNE HUILE ISOLANTES

Les principaux rôles d'une huile isolante sont le refroidissement (éviter la surchauffe et évacuer les calories vers les parois) et l'isolement électrique (prévenir les arcs, éviter la reprise d'humidité et éviter les fines bulles d'air et de gaz piégés dans les enroulements). L'huile de transformateur a aussi une très bonne aptitude à résister au vieillissement.

III.2.2. CHOIX D'UN DIELECTRIQUE LIQUIDE

Le choix d'une huile isolante est déterminé en fonction de ses caractéristiques physiques et chimiques que l'on désire lui conférer. Comme la viscosité, la densité, la stabilité ..., ainsi qu'en fonction de la nature chimique recherchée (paraffinique, naphthénique ou mixte).

III.2.3. CARACTERISTIQUES D'UNE HUILE MINERALE ISOLANTE

Comme on a dit, le choix d'une huile isolante est lié à certaines propriétés physiques, chimiques et électriques.

III.2.3.1. PROPRIETES ELECTRIQUES

- La **permittivité** définit la possibilité sous l'action du champ électrique, de libérer des charges. Elle résulte de diverses contributions de déplacement de charges [9bija]. la permittivité relative « ϵ_r » est définie comme le rapport « ϵ / ϵ_0 » de la permittivité absolue

« ϵ » du liquide à celle de vide « ϵ_0 » ($\epsilon_0 = (1/36\pi)10^{-9}$ F/m). la permittivité relative des huiles est comprise entre 2.1 et 2.5 à 90°C [20].

• **La conductivité** σ d'un milieu qui renferme n porteurs de mobilité μ_i et de densité q_i exprime le rapport constant de la densité de courant \vec{j} au champ \vec{E} , selon l'équation suivante [21] :

$$\vec{J} = \left(\sum_{i=1}^n \mu_i q_i \right) \vec{E}$$

• **La résistivité** ρ est définie comme l'inverse de la conductivité. Dans le cas d'une huile isolante la résistivité est comprise entre 20 et 2000G Ω m à 90°C [20].

• En pratique, pour mieux caractériser un diélectrique du point de vue pertes, il est usuel d'utiliser la tangente de l'angle de pertes δ (**facteur de pertes diélectrique**). C'est un facteur qui augmente avec l'augmentation de la température et dépend de la tension appliquée [10bija]. Le milieu du diélectrique sous tension alternative sinusoïdale de pulsation ω , il obéit à la loi d'Ohm et il peut être représenté par un schéma équivalent comportant une résistance R et une capacité C en parallèle [21].

Donc :

$$\text{tg}(\delta) = 1 / RC\omega = \sigma / \epsilon\omega = 1 / \rho\epsilon\omega.$$

• **La rigidité diélectrique** d'une huile est la valeur maximale du champ électrique que l'on peut lui appliquer sans décharge.

On peut représenter quelques facteurs qui peuvent influencer sur la rigidité diélectrique :

- **L'humidité** : la présence d'un peu d'humidité dans le liquide diminue considérablement la rigidité diélectrique de ce dernier.
- **La pression hydrostatique** : la tension de claquage croît de façon remarquable quand la pression augmente sans indication de saturation en polarité négative, cette croissance est bien moins rapide et tend à disparaître en polarité positive [22]. Cette forte dépendance de pression en polarité négative indique que le mécanisme des bulles est dominant, sans pour autant exclure les mécanismes gazeux [23].
- **Distance entre électrodes** : en augmentant la distance entre électrodes la tension de claquage augmente, mais la rigidité diélectrique du liquide diminue.
- **Température** : l'influence de la température est double. En augmentant la température, la concentration de gaz et de l'humidité diminue dans le liquide. Cependant les conditions pour avoir un claquage purement thermoélectrique s'améliorent [24].
- **Durée d'application de la tension** : plus la durée d'application augmente plus l'effet des différentes impuretés se fait ressentir et la tension de claquage diminuent [20].

III.2.3.2. PROPRIETES PHYSIQUE

- **La viscosité dynamique** ou **La viscosité absolue** η est le coefficient de proportionnalité entre les réactions du milieu (qui sont provoquées par les contraintes de cisaillement ou de glissement dans un liquide) et le gradient de vitesse [25].

La viscosité diminue avec la température et augmente avec la pression. L'expérience montre que dans le cas d'une huile minérale, la viscosité double quand la pression passe de la pression atmosphérique à 350 bar.

- **Le point d'écoulement** est la température à laquelle les liquides se figent, il nous permet d'évaluer les performances d'une huile à basse température [25], En particulier, dans les transformateurs avec circulation forcée. Ainsi, la température à laquelle la viscosité permet le fonctionnement des pompes est de l'ordre de -25°C à 35°C pour des points d'écoulement de -45°C [27][24].

- **La conductivité thermique** est mesurée en Watt par mètre Kelvin, elle exprime le flux thermique s'écoulant, en régime permanent sous l'effet d'un gradient thermique entre deux isothermes du liquide. Elle décroît lorsque la température et la masse volumique augmentent [27].

- **Le point d'éclair** indique la température de l'huile pour laquelle les vapeurs émises au contact de l'air et approchées d'une flamme prennent feu sans atteindre la combustion totale. Sa détermination renseigne quant à la présence d'hydrocarbures gazeux dissous dans l'huile par la suite de défauts électrique ou thermique [2].

- **Le point de feu** est le point d'inflammation de huile. Il indique la température minimale pour laquelle la combustion des vapeurs d'un liquide sera entretenue [27].

- **La couleur** est aussi un paramètre souvent considéré. En générale, les liquides isolants sont incolores ; certains, comme les huiles minérales ont une couleur ambrée ou une fluorescence caractéristique. La variation de la couleur dans des livraisons d'huile neuve peut indiquer un changement dans l'origine de huile ou dans la fabrication, sans que cela soit absolument certain. Dans le cas d'une huile usagée, une coloration foncée indique sa détérioration [2].

III.2.3.3. PROPRIETES CHIMIQUES

- **L'acidité** d'une huile en service résulte des produits d'oxydation. Pour une huile neuve l'acidité est de l'ordre de $0,5 \text{ mgKOH/g}$ [24].

- Les propriétés électriques des liquides sont affectées par **leur teneur en eau** ou, mieux, par l'association eau-particules. La quantité d'eau dissoute dans un liquide à saturation dépend de sa nature et s'exprime en parties par million en masse [p.p.m., mg/kg ou 10^{-6}]. L'eau dissoute dans un liquide, avant saturation, est à l'état de vapeur. Sa teneur dépend de la température et de la pression partielle d'eau dans l'atmosphère au-dessus du liquide, c'est-à-dire de l'humidité relative extérieure. Si cette dernière est de cent pour cent, le liquide se sature en eau et, à une température donnée, cette saturation correspond aux premières gouttes d'eau condensée précipitables, généralement sous l'aspect d'un trouble [25].

- **La stabilité chimique** concerne la stabilité à l'oxydation, la stabilité thermique et, dans certains cas, la stabilité à l'hydrolyse, ensemble de processus réactionnels désignés sous le terme de vieillissement et responsables de la durée de vie des liquides en service, qui devrait correspondre à celle des matériels.

La stabilité à l'oxydation des liquides repose sur l'expérience acquise avec les huiles minérales. L'oxydation de ces huiles se traduit par un accroissement de l'indice d'acidité totale, en abrégé TAN (de l'anglais Total Acidity Number). Cet indice est exprimé en [mg KOH/g d'huile] et il est la somme de l'acidité soluble ou indice de neutralisation et de l'acidité volatile, tout deux exprimés également en [mg KOH/g d'huile]. L'acidité soluble a une influence sur la dégradation thermique de l'isolation cellulosique. Les huiles, de couleur claire à jaune, deviennent de plus en plus foncées et, à partir d'une certaine valeur de TAN forment, des dépôts insolubles ou boues d'oxydation qui se déposent sur l'isolant solide, réduisant l'évacuation des pertes ; l'huile est à régénérer. Les liquides de synthèse ne forment généralement pas de dépôt [25].

III.2.4. VIEILLESSEMENT DES HUILES DE TRANSFORMATEUR

Pendant leurs utilisations, les huiles minérales isolantes sont soumises à plusieurs contraintes électriques, thermique et chimiques. Sous l'action combinée ou séparée de ces contraintes, les huiles isolantes se dégradent au cours de temps, ce qui est connue sous le nom de vieillissement. Donc, on appelle vieillissement tous phénomènes se traduisant par une évolution lente et irréversible des propriétés du matériau.

III.2.4.1. VIEILLISSEMENT THERMIQUE DES HUILES DE TRANSFORMATEUR

On convient d'appeler vieillissement thermique tous phénomènes traduisant une évolution qui ne fait intervenir que [28] la température, l'atmosphère et le matériaux. Il y a deux buts principaux pour l'étude du phénomène de vieillissement thermique d'un isolant liquide qui sont :

- Evaluer la durée de vie de cet isolant sous l'effet des contraintes thermiques.
- Trouver une éventuelle corrélation entre le processus de vieillissement et les contraintes qui le provoquent.

III.2.4.2. LOIS DU VIEILLISSEMENT DES ISOLANTS

Les premiers travaux ont porté sur le vieillissement des matériaux isolants solides. En 1930 V.M.Montsinger, après avoir étudié la diminution des caractéristiques mécaniques des papiers isolant imprégnés d'huiles minérales sous l'influence de la température, propose une loi générale pour traduire le vieillissement qui s'exprime par la relation suivante [29] :

$$L(h) = C \cdot \exp(-b \cdot \theta)$$

Avec :

C et b sont des constantes.

L(h) est la durée de vie.

θ (°C) est la température.

Des études expérimentales sont venues confirmer que, dans un très grand nombre de réaction chimiques et tout au moins en première approximation, les constantes de vitesse k varient avec la température conformément à l'équation d'Arrhenius :

$$k = k_0 \exp\left(-\frac{W}{RT}\right)$$

Avec :

k_0 est un constante.

R est la constante de gaz parfaits.

T est la température.

W est l'énergie d'activation.

En s'appuyant su cette équation, W.Dakin(1948) à montré que la durée de vie L d'un matériaux isolant ou d'un système d'isolation en fonction de la température T peut se mettre sous la forme simple [2] :

$$\log(L) = A + (B / T) .$$

A et B sont des constantes.

III.3. LES MECANISMES DE CLAQUAGE DES DIELECTRIQUES LIQUIDES

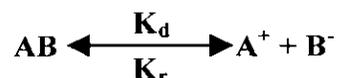
Les processus fondamentaux conduisant au claquage des isolants liquides sont encore mal connus. Ceci est lié en grande partie au fait que la connaissance des propriétés physique de l'état liquide est encore moins bien développée que dans les gaz et les solides. Ainsi, de nombreux chercheurs ont tenté d'appliquer aux liquides les théories proposées pour les gaz et les solides.

III.3.1. MECANISME ELECTRONIQUE

La conduction dans les diélectriques liquides, même sous champ intense (≈ 10 MV/cm) est presque toujours assurée par des ions de mobilité faible, qui préexistent dans le milieu ou qui se créent aux interfaces [26].

III.3.1.1. DISSOCIATION RENFORCEE PAR LE CHAMP

Il existe une réaction d'équilibre entre les molécules neutres AB d'un liquide et les ions A^+ et B^- , qui s'écrit :



Avec :

K_d : constante de vitesse de dissociation.

K_r : constante de vitesse de recombinaison.

L'application d'un champ électrique au liquide, abaisse l'énergie de liaison de molécules neutres AB . En conséquence de quoi, le nombre des ions dissociés devient supérieur aux nombres des ions recombines.

III.3.1.2. PRODUCTION DES CHARGES AUX INTERFACES

La création des ions aux électrodes a pour origine des réactions électrochimiques résultant de la mise en contact de l'électrode et du diélectrique. Des échanges électroniques ont lieu lorsque l'ion pénètre dans une zone de 10^{-7} cm d'épaisseur, où règne un champ très intense ($\sim 10^7$ V/cm). Cette région est formée par des ions non déchargés fixés à la surface du métal en permanence [30].

III.3.1.3. CONSEQUENCE DU TRANSFERT DE CHARGE

Le transfert de charge s'accompagne d'une augmentation ou d'une réduction de la résistivité du liquide selon que l'élément qui réagit est un ion ou une espèce neutre [26].

III.3.2. MECANISME DE CLAQUAGE AVEC PHASE GAZEUSE

Dans les liquides ultra purs, et en géométrie pointe- plan, sous tension continue, il a été observé, qu'à partir d'un certain seuil de tension, des bulles gazeuses se forment au voisinage de la pointe, puis sont violemment chassées vers le plan. Cette phase gazeuse résulterait soit de la vaporisation et de la nucléation des bulles, soit de la cavitation [26][20].

III.3.2.1. MECANISME THERMIQUE

D'après Watson et Sharbaugh, une bulle de vapeur est générée dans le liquide par une injection de courant à partir d'aspérités sur la cathode. Ce courant serait limité par la charge d'espace et à la forme suivante :

$$I = A \cdot V^n$$

Avec :

A : est une constante.

V : est la tension appliquée, $1.5 \leq n \leq 2$.

Un modèle plus élaboré et tenant compte du régime transitoire a été proposé par KAO. Si l'intensité de courant est suffisante pour ramener la température du liquide à son point d'ébullition, il y a claquage [2].

III.3.2.2. MECANISME DE CAVITATION

Le phénomène de cavitation dans un liquide se manifeste lorsque la pression de ce liquide à température constante tombe au dessous de sa vapeur saturante [2].

D'après Krasucki, les bulles se forment en des points où le champ est élevé (particules solides de très petites dimensions dans le liquide ou attachées aux électrodes) pour qu'il y ait formation d'une cavité, le critère de pression nulle est :

$$P_{em} = P_h + P_s.$$

Avec :

P_{em} : Pression électromécanique.

P_h : Pression hydrostatique.

P_s : Pression due à la tension superficielle du liquide.

Un critère analogue a été proposé par Thomas, en supposant que la pression d'électrostriction s'oppose au décollement du liquide.

III.3.3. MECANISME DE CLAQUAGE PAR PONT

Les impuretés se trouvent dans les isolants liquides sous différentes natures (bulles de gaz, gouttelettes d'eau, particules solides isolantes ou conductrices).

La présence de ces impuretés renforce le champ localement. La déformation du champ dépend de plusieurs paramètres comme :

- La forme et la dimension des impuretés.
- L'intervalle et la forme des électrodes.
- La permittivité et la conductivité des impuretés.
- La concentration des impuretés entre les électrodes.
- La valeur des charges libres existant à la surface des particules.

L'élévation du champ peut conduire à des claquages partiels dans le cas des impuretés solide conductrices et des impuretés liquides.

Dans le cas des impuretés solides isolantes, celle-ci se polarisent sous l'effet du champ électrique et tendent à se déplacer vers les zones où le champ est le plus intense. Pour qu'un pont de particules se forme entre les électrodes, il faut que le temps d'application soit relativement lent. Quand ce pont relie les électrodes, il s'établit un fort courant à travers le pont qui est plus conducteur que le liquide. On aura ainsi, un échauffement local important qui peut engendrer le claquage [20].

Dans le cas des l'eau condensé, les gouttes d'eau sont initialement sphériques et tendent à se déformer sous l'influence du champ électrique. Elles prennent la forme d'ellipsoïdes qui s'allongent dans la direction du champ. Lorsque le volume de la goutte est suffisant, il peut ainsi se former un canal d'eau reliant les électrodes. Ce dernier engendre un fort courant qui provoquerait la vaporisation de l'eau et par suite le claquage dans le canal « vapeur - eau » en cours de formation [20].

III.3.4. EVOLUTION DES DECHARGES DANS LES DIELECTRIQUES LIQUIDES

De nombreux travaux ont portés sur la propagation de streamers, dans des systèmes d'électrodes à champ divergent. Il est généralement admis que le claquage est précédé d'une phase de pré-claquage pouvant elle-même se séparer en deux phases [2] :

- Une phase de génération caractérisée par un temps t_g dit de génération, pendant lequel apparaît au voisinage d'une électrode une perturbation révélant la forme d'arborescence.
- Une phase de propagation durant laquelle se développe la perturbation précédemment créée. Cette phase est caractérisée par un temps t_p .

Les streamers sont caractérisés par leur indice de réfraction qui est différent de ce lui du liquide. Ils produisent des courants, et émettent de la lumière.

III.4. LES ESSAIS SUR LES DIELECTRIQUES LIQUIDES

III.4.1. ESSAIS DIELECTRIQUES

III.4.1.1. MESURE DU FACTEUR DE PERTES DIELECTRIQUES ($\text{tg}\delta$) ET DE LA PERMITTIVITE RELATIVE (ϵ_r)

Le facteur de pertes diélectriques ($\text{tg}\delta$) et la permittivité relative (ϵ_r), sont mesurés à l'aide d'un pont de Schering, selon la norme CEI250. La cellule d'essai contenant l'échantillon d'huile est constituée de deux cylindres coaxiaux de distance entre électrodes de 5mm [31].

Les essais sont effectués sous une tension de 2kV, 50Hz, et sous une température variant de 30 °C à 120 °C, la sensibilité peut aller jusqu'à 10^{-6} .

III.4.1.2. MESURE DE LA RIGIDITE DIELECTRIQUE

La mesure de la tension de claquage est effectuée conformément à la norme CEI 156.

L'appareil utilisé est un Spintermètre. C'est une cellule d'essais ayant des électrodes sphériques de diamètre $d=12.5\text{mm}$ et un écartement $e=2.5\text{mm}$ [31].

III.4.1.3. MESURE DE LA RESISTIVITE (ρ)

La résistivité (ρ) est mesurée à l'aide d'un Meghommètre selon la norme CEI 93.

Les mesures sont effectuées sous une tension continue de 500V, appliquée durant 10mn, et sous une température variant de 30°C à 120°C [31].

III.4.2. MESURES PHYSICO-CHIMIQUES

III.4.2.1. MESURE DE LA VISCOSITE CINEMATIQUE (η)

La viscosité cinématique (η) est mesurée selon la norme NFT60-100. On procède à la mesure du temps d'écoulement d'un volume du liquide dans un viscosimètre à capillaire en verre.

La viscosité cinématique est le produit de ce temps par la constante de temps d'étalonnage de l'appareil [31].

III.4.2.2. LA COULEUR

La couleur d'une huile isolante est déterminée à l'aide d'un colorimètre en lumière transmise. Elle est exprimée par un indice obtenu par comparaison à une série de couleurs de verres étalons. L'étalon de couleur correspondant, ou celui dont la couleur est juste supérieure, est pris comme valeur de couleur [31].

III.4.2.3. L'ACIDITE

La mesure de l'acidité est réalisée par un procédé chimique simple : la neutralisation.

L'acidité est exprimée en mg de KOH par gramme d'huile. Elle correspond au nombre de mg d'une solution de Potasse alcoolique de normalité 0.1, nécessaire pour neutraliser une gamme d'huile à essayer [31].

Elle se calcule par l'équation suivante :

$$IA=0.1M_1.V/M$$

Où :

- 0.1 : normalité de la solution KOH (g/l).
- M_1 : masse moléculaire de la solution KOH (g/moles).
- M : masse d'huile (g).
- V : volume en ml de KOH nécessaire pour avoir la neutralisation.

III.4.3. ESSAIS DE VIEILLISSEMENT DE L'HUILE

Ces essais ont pour objectif de définir les variations des propriétés de l'huile sous l'effet de contraintes appliquées durant une longue période, de préférence jusqu'à la dégradation définissant la limite de durée de vie.

Les principaux types de vieillissement de l'huile de transformateur sont le vieillissement thermique, le vieillissement électrochimique et le vieillissement électrique.

Le vieillissement thermique correspond à l'échauffement cyclique répété ou continu de longue durée, à des températures relativement élevées.

Le vieillissement électrochimique est caractérisé surtout par les conséquences, sous l'effet du champ électrique, de l'action de longue durée de certains agents chimiques accidentellement introduits dans l'huile ou provenant de la propre dégradation de cette dernière.

Le vieillissement électrique est surtout caractérisé par l'action de longue durée des décharges partielles ou tout phénomène d'ionisation dû au champ électrique [31].

III.5. HUILE DE TRANSFORMATEUR BORAC22

Dans ce suit, nous présentons Les résultats d'essais effectués par des chercheurs du laboratoire de H.T. dans le cadre d'un projet de recherche en collaboration avec SONAELGAZ

III.5.1. PROPRIETES DE L'UILE BORAC22

- **rigidité diélectrique** : La valeur de la tension de claquage de l'huile BORAK 22 neuve est de 51kV, elle correspond a une valeur de la rigidité diélectrique de 235 kV /cm [32]. Le tableau (III.1) représente la tension de claquage de l'huile avant et après vieillissement

Tableau III.1.

Etat de l'huile	Tension de claquage (kV)	Rigidité diélectrique (kV/cm)
Neuve	51	235
Vieillie à 80°C	51	235
Vieillie à 120°C	49	225

- **L'acidité** : L'indice d'acidité de l'huile neuve utilisée est très faible, à savoir 0.026mg KOH/g d'huile. Cette acidité reflète l'existence de composés organiques acides inhérents à la constitution chimique de l'huile [32].

Le tableau (III.2) représente l'indice d'acidité de l'huile avant et après vieillissement

Tableau III.2.

Etat de l'huile	Indice d'acidité (mg KOH/g)
Neuve	0.026
Vieillie à 80°C	0.046
Vieillie à 120°C	0.108

- **La teneur en eau** La mesure de la teneur en eau de l'huile neuve étudiée a donné une quantité d'eau de 30 p.p.m, (valeur acceptable) [32].

Le tableau (III.3) représente la teneur en eau de l'huile avant et après vieillissement.

Tableau III.3.

Etat de l'huile	Teneur en eau (p.p.m)
Neuve	30
Vieillie à 80°C	31
Vieillie à 120°C	35

- **La couleur** : L'indice de couleur de l'huile neuve est 1, cette huile contient une fraction minime de la fonction aromatique [32].

Le tableau (III.4) représente l'indice de couleur de l'huile avant et après vieillissement.

Tableau III.4.

États de l'huile	Indice de couleur
Neuve	1
Vieillie à 80°C	1
Vieillie à 120°C	3

1 : blanc pur

3 : citron pâle

- **La viscosité cinématique** : La viscosité cinématique, correspondant à une température fixe, est donnée par :

$$V = C. t$$

Avec :

C : constante de temps d'étalonnage de l'appareille.

t : temps d'écoulement en secondes.

Pour La BORAK22 : C = 0.04787, t = 262 s.

D'où :

$$V = C \cdot t = 12.54 \text{ cSt}$$

Le tableau (III.5) représente la viscosité cinématique de l'huile avant et après vieillissement [32].

Tableau III.5.

États de l'huile	Viscosité cinématique (CST)
Neuve	12.54
Vieille à 80°C	12.54
Vieille à 120°C	12.49

III.5.2. INFLUENCE DE LA TEMPERATURE SUR LES PROPRIETES DE L'HUILE

III.5.2.1. FACTEUR DE PERTES DIELECTRIQUES

Le facteur de pertes diélectrique augmente en exponentiel en fonction de la température (Fig.III.1). Cette évolution est due en même temps à la réduction de la viscosité causant une augmentation de la mobilité des porteurs de charges et à l'augmentation du nombre des porteurs à cause d'une plus grande dissociation thermique.

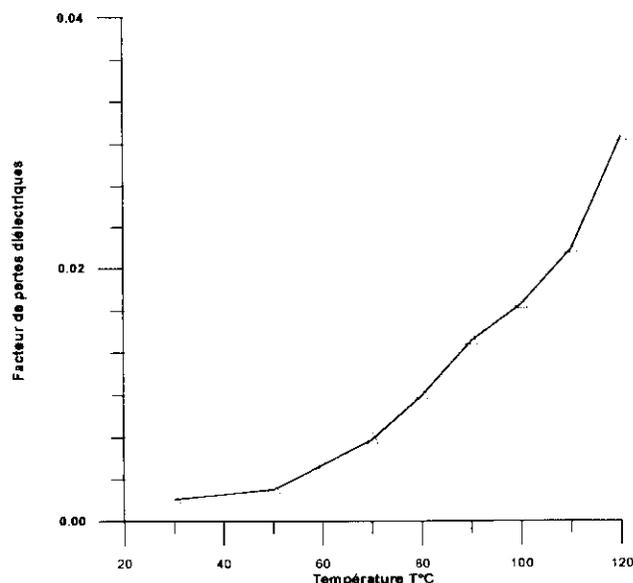


Fig.III.1. Variation du facteur de pertes diélectriques en fonction de la température [33]

PERMITTIVITE RELATIVE ϵ_r

La permittivité relative décroît presque linéairement avec la température (Fig.III.2.).

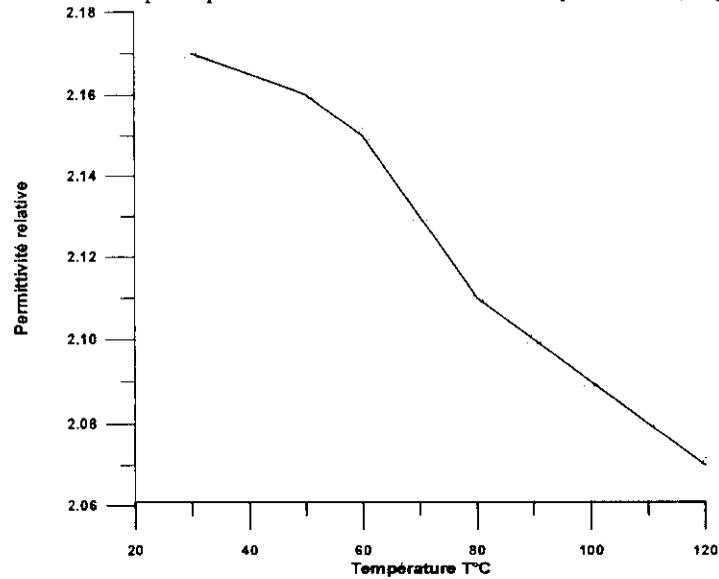


Fig.III.2.Variation de la permittivité relative en fonction de la température [33]

RESISTIVITE

La résistivité décroît en fonction de la température (Fig.III.3.).

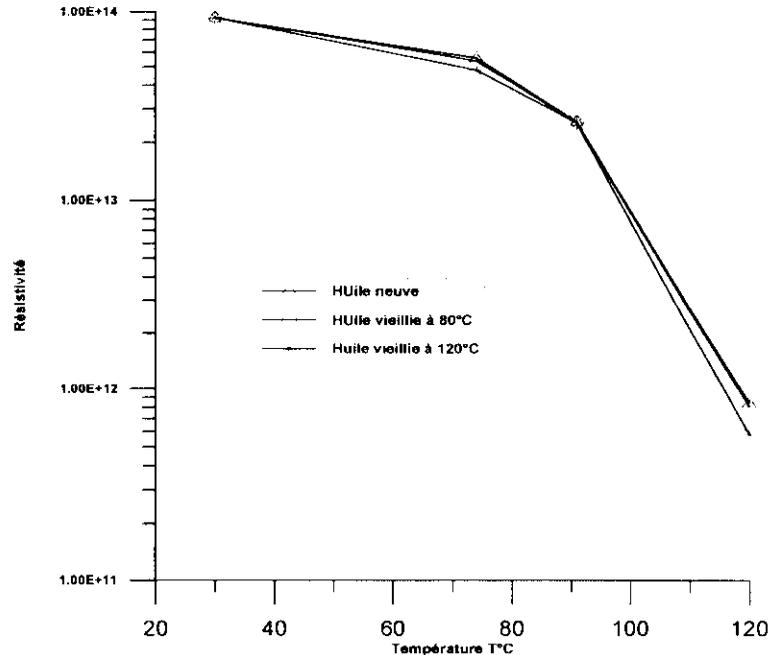


Fig.III.3.résistivité en fonction de la température avant et après vieillissement [33].

III.5.3. INFLUENCE DU VIEILLISSEMENT THERMIQUE SUR LES PROPRIETES DE L'HUILE

III.5.3.1. FACTEUR DE PERTES DIELECTRIQUES

Le facteur de pertes diélectrique $\text{tg}(\delta)$ pour des différentes valeurs de température augmente au cours de vieillissement thermique, et plus que la température augmente plus il augmente rapidement (Fig.III.4. et Fig.III.5.). On peut remarquer que pour $T=50^{\circ}\text{C}$ le facteur de pertes diélectrique augmente linéairement.

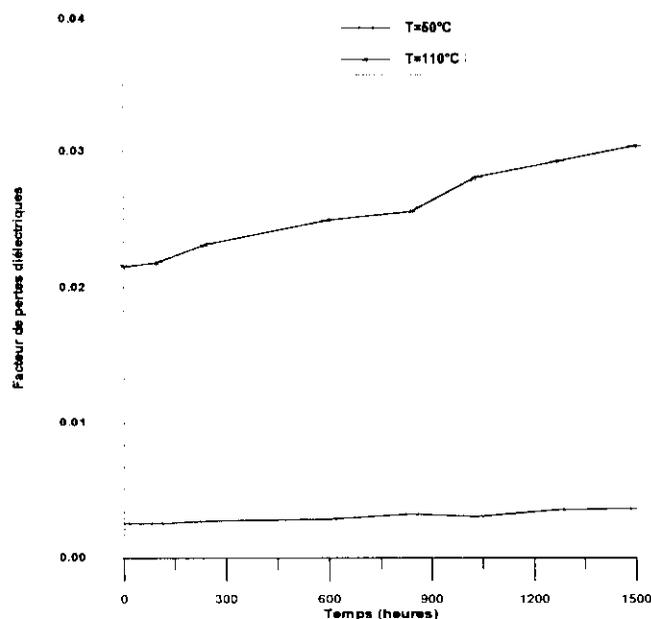


Fig.III.4. facteur de pertes diélectriques durant le vieillissement à $T=80^{\circ}\text{C}$ [33].

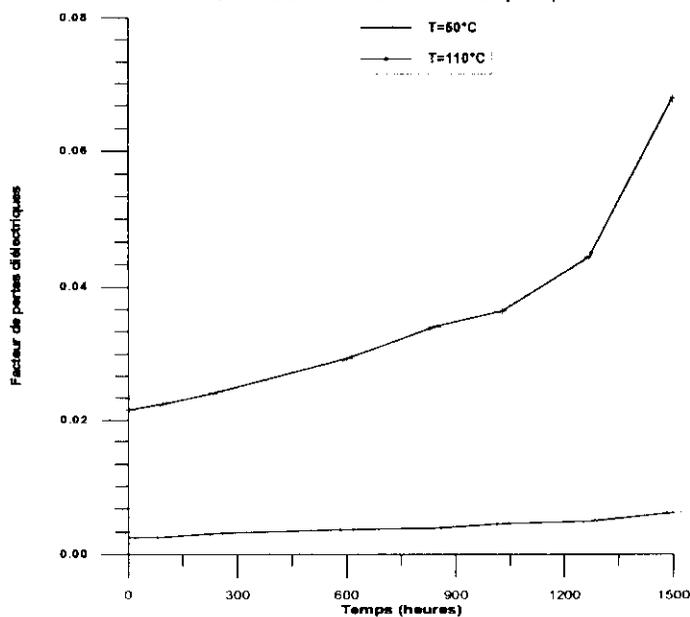


Fig.III.5. facteur de pertes diélectriques durant le vieillissement à $T=120^{\circ}\text{C}$ [33].

III.5.3.2. PERMITTIVITE RELATIVE ϵ_r :

Les figures suivantes (**Fig.III.6.et Fig.III.7**) représentent la variation de la permittivité relative en fonction de temps de vieillissement thermique à $T=80^\circ\text{C}$ et à $T=120^\circ\text{C}$. On remarque que la permittivité décroît avec une très faibles variation durant le vieillissement.

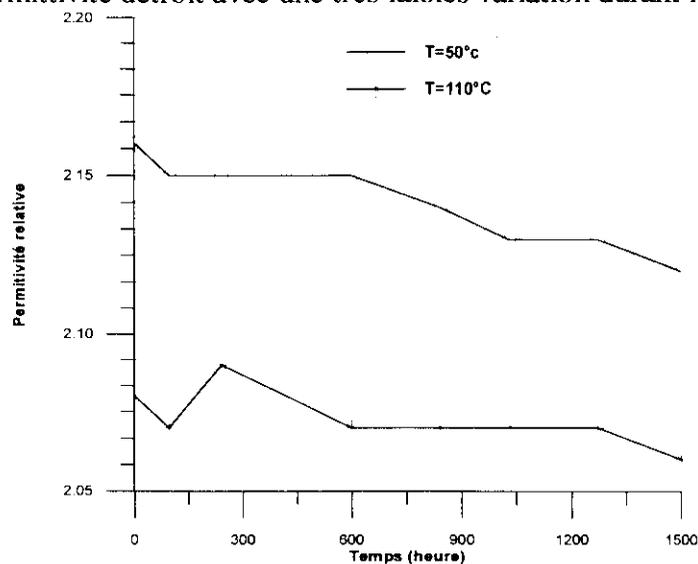


Fig.III.5.permittivité relative ϵ_r durant le vieillissement à $T=80^\circ\text{C}$ [33].

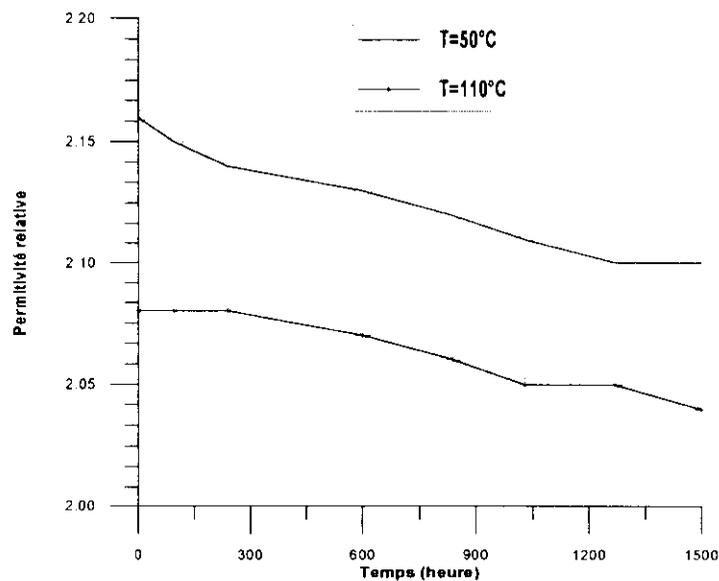
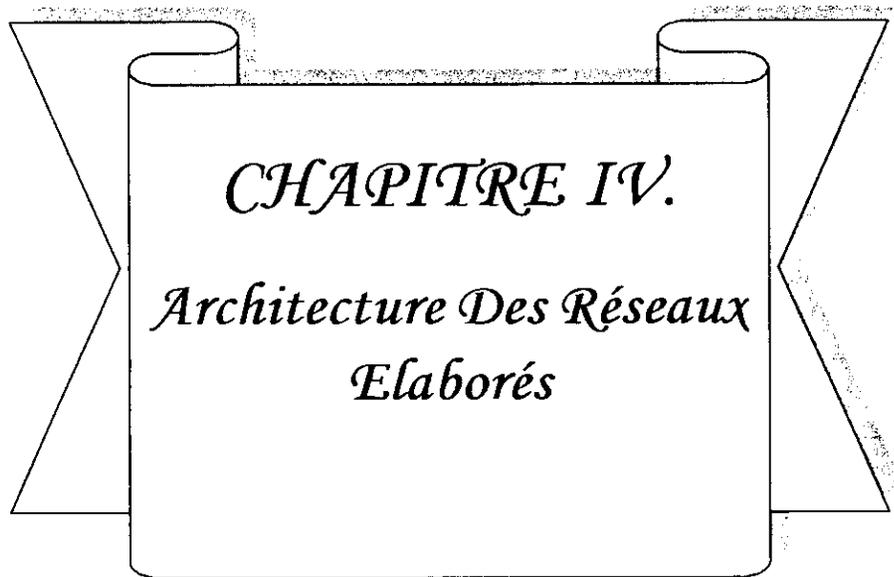


Fig.III.6.permittivité relative ϵ_r durant le vieillissement à $T=120^\circ\text{C}$ [33].

III.7. CONCLUSION

- Le facteur de pertes diélectrique augmente avec la température.
- La permittivité relative et la viscosité diminuent avec l'augmentation de la température, mais la permittivité subit à une très faible variation.
- pour le vieillissement thermique de l'huile, les propriétés électriques, les propriétés physiques et les propriétés chimiques varient en fonction de la durée de vieillissement, et surtout pour celle vieillie à température élevée.



CHAPITRE IV.
Architecture Des Réseaux
Elaborés

IV.1. INTRODUCTION

L'objectif de ce chapitre est de réaliser un programme de calcul nous permettant de prédire certaines propriétés électriques d'une huile isolante (BORAC 22) en fonction du temps de vieillissement sous une température constante.

Pour une bonne prédiction, l'élaboration de la meilleure architecture possible du réseau est très importante. Ceci fait l'objet de la première partie de ce chapitre.

Comment savoir que notre réseau donnera une bonne prédiction ?

Si on arrive à prédire la caractéristique donnée avec une erreur acceptable par rapport à la caractéristique expérimentale, avec une durée d'apprentissage relativement courte par rapport au temps global de vieillissement et d'un pas d'apprentissage le plus grand possible. Donc on dira que la prédiction de notre réseau est bonne.

IV.2. ARCHITECTURE ET FONCTIONNEMENT DES RESEAUX RBF

Parmi toutes les architectures que nous avons citées, nous avons opté pour l'architecture d'un réseau RBF composé de trois couches, une seule couche de sortie Y_1 , et une couche cachée de nombre de neurones N_c variable de fonction d'activation gaussienne (l'utilisation de cette fonction permet de bénéficier de sa caractéristique locale pour faciliter l'apprentissage et améliorer la généralisation). Chacun de ces neurones est doté d'un vecteur C_i appelé centre, les valeurs de ces vecteurs représentent chacune le barycentre de la classe qui lui correspond.

Les centres des fonctions gaussiennes sont choisis par plusieurs techniques afin de les mettre comme points représentatifs dans les zones de grandes concentrations de données.

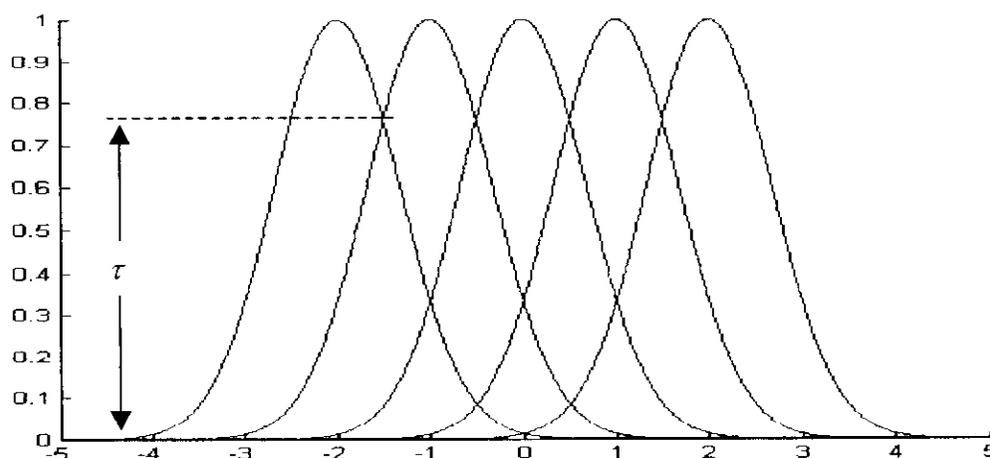


Fig.IV.1. Schéma d'un treillis régulier unidimensionnel

Mais il existe une méthode plus simple qui consiste à disposer les centres en treillis réguliers pour couvrir uniformément la partie utile de l'espace d'entrée. Les rayons des gaussiennes dans ce cas, sont fixés par les distances entre centres voisins afin d'assurer un certain degré de recouvrement entre les différentes gaussiennes. La figure IV.1. définit le taux

de recouvrement τ d'un réseau à base radial à treillis régulier unidimensionnel, en supposant les gaussiennes d'écart type constant et une distance entre centres de δ , le taux de recouvrement est donné par :

$$\tau = \exp\left(-\frac{1}{2} \frac{\left(\frac{\delta}{\sigma}\right)^2}{\sigma^2}\right) \tag{IV.1.}$$

Pour un réseau à treillis et uniforme sur le domaine d'approximation, le nombre de centres (L) et le taux de recouvrement τ définissent totalement la topologie du réseau. Mais quelquefois, la disposition des centres a priori s'avère impraticable, car, le nombre de centres nécessaires est parfois trop élevé (dans le cas où l'on décide d'adopter une disposition en treillis régulier). Il y a une méthode très efficace qui consiste à utiliser un nombre beaucoup plus restreint de centres et de laisser libre la disposition de ces centres ; l'algorithme d'apprentissage ajuste, outre les poids w_j , les positions des centres ainsi que les valeurs des rayons associées. Le réseau s'auto-organisera pour placer un nombre élevé de centres là où la fonction est accidentée, et espacer les centres dans les régions où la fonction est plus lisse. Ces réseaux nous permettent non seulement l'apprentissage des poids des connexions [9], mais aussi celui des centres et des rayons, ils sont appelés Réseaux à Base Radial Généralisés c'est la deuxième méthode la deuxième méthode d'apprentissage que nous avons utilisée.

Mais en ce qui concerne le nombre d'entrées, nous avons utilisé deux modèles, le premier avec une seule entrée (modèle1), et un deuxième avec deux entrées (modèle2).

Donc l'architecture de notre réseau est comme suit :

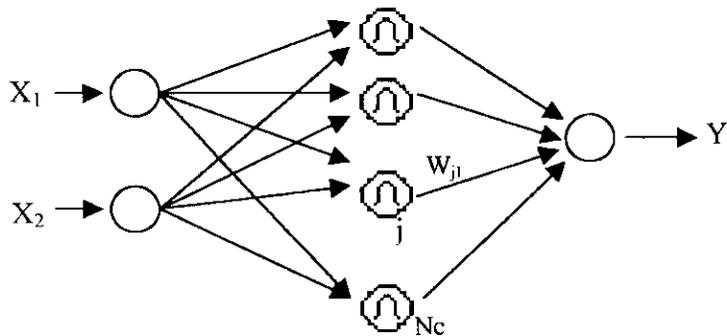


Fig.IV.2. Architecture de réseau RBF.

$$y_i(X) = \sum_j^{Nc} w_{j1} \rho_j(\|X_k - C_j^k\|) \quad 1 \leq i \leq n + f \tag{IV.2.}$$

Où :

$$\rho_j(\|X_i^k - C_j^k\|) = \exp\left(-\frac{1}{2} \sum_{k=1}^K \frac{(X_i^k - C_j^k)^2}{\sigma_j^{k2}}\right) \quad 1 \leq j \leq Nc \tag{IV.3.}$$

Tel que :

- K : est le nombre de neurones dans la couche d'entrée (modèle1, modèle2).
- Nc : est le nombre de neurones dans la couche cachée.

n : est le dimension de vecteur Y_1 au début de l'apprentissage.

f : est le nombre de valeurs à prédire.

C_j^k : est le centre du $j^{\text{ème}}$ neurone de la couche cachée de la $K^{\text{ème}}$ entée.

ρ_{ij} : est l'élément de la matrice de réponse de fonction d'activation du $j^{\text{ème}}$ neurone.

IV.3. CODAGE DE L'INFORMATION

Afin d'améliorer les performances de notre réseau, les données d'une telle caractéristique ne sont plus appliquées directement, une phase de prétraitement consiste à normaliser ces données.

Au début, nous avons tenté de normaliser le signal par rapport à sa valeur moyenne \bar{Y} et l'écart type σ comme suit :

$$Y_i^* = \frac{Y_i - \bar{Y}}{\sigma} \quad (\text{IV.4.})$$

$$\text{Avec} \quad \sigma = \sqrt{\sum \frac{(Y_i - \bar{Y})^2}{n}} \quad i=1,2,\dots,n. \quad (\text{IV.5.})$$

Le nouveau signal obtenu est donc centré par rapport à l'axe des abscisses, d'où l'apparition des valeurs négatives.

Dans le deuxième cas, la normalisation du signal est faite par rapport à sa valeur maximale comme suit :

$$Y_i^* = \frac{Y_i}{\max [Y]} \quad (\text{IV.6.})$$

De plus, pour que notre réseau reste souple durant le test des différents signaux nous avons préféré de laisser quelques paramètres variables afin de choisir le réseau convenable pour une telle caractéristique. Ces paramètres sont :

- La durée globale du signal (pour notre application, c'est le temps global de vieillissement).
- La durée relative d'apprentissage par rapport à la durée globale (en pour cent).
- Le pas d'apprentissage relatif par rapport à la durée d'apprentissage (en pour cent).
- Le nombre de neurones dans la couche cachée.

Cette phase est résumée ci-dessous :

- Lire les Données :
 1. Les deux vecteurs X , $Y_{\text{désirée}}$ de la courbe à prédire.
 2. La durée globale TG.
 3. La durée d'apprentissage relatif (T_AR) en pour cent par rapport à TG.
 4. le Pas d'Apprentissage relatif (P_AR) en pour cent par rapport à T_A.
 5. l'erreur désirée entre les $Y_{\text{désirées}}$ et $Y_{\text{calculées}}$ du réseau.
- Calculer le vecteur Temps, puis sélectionner le vecteur Y_d^* .

IV.4. PHASE D'APPRENTISSAGE DU RESEAU RBF

1) Apprentissage avec ROM :

Nous avons essayé de tester les performances des deux réseaux, nommés RBF_1 et RBF_2 (l'indice 1 est pour une seule entrée et l'indice 2 pour deux entrées) avec la méthode d'optimisation aléatoire dont nous présentons ci-dessous l'algorithme.

1. Initialiser aléatoirement tous les poids W_j du réseau. Initialiser la variance v et la moyenne de la séquence aléatoire $b_j(0) = 0$.
2. Générer une nouvelle séquence aléatoire $\xi^j(k)$ de moyenne $b_j(k)$ et variance $v(k)$.
3. Calculer les erreurs $E = \sum_{p=1}^m E_p$ à la sortie du réseau pour chaque cas :

$$\begin{aligned} - E(w(k)) &\rightarrow E_1 \\ - E(w(k) + \xi(k)) &\rightarrow E_2 \\ - E(w(k) - \xi(k)) &\rightarrow E_3 \end{aligned}$$

$$\text{a. Si } E_2 < E_1 \text{ alors } \begin{cases} w_j(k+1) = w_j(k) + \xi_j(k) \\ b_j(k+1) = 0.2b_j(k) + 0.4\xi_j(k) \end{cases} \quad (\text{IV.8.})$$

$$\text{b. Si } E_2 > E_1 \text{ et } E_3 < E_1 \text{ alors } \begin{cases} w_j(k+1) = w_j(k) - \xi_j(k) \\ b_j(k+1) = 0b_j(k) - 0.4\xi_j(k) \end{cases} \quad (\text{IV.9.})$$

$$\text{Si non } \begin{cases} w_j(k+1) = w_j(k) \\ b_j(k+1) = 0.5b_j(k) \end{cases} \quad (\text{IV.10.})$$

4. Poser $k=k+1$ et refaire les étapes de 2 à 4 jusqu'à l'obtention de l'erreur désirée.

2) Méthode de Centrage Adaptatif AC M (Adaptive Centring Method)

Ce mode d'apprentissage permettant non seulement l'adaptation des poids des connexion w_j , mais aussi celui des centres C_j et les rayons du réseau σ_j .

Nous présentons ci-dessous les étapes de cette technique d'apprentissage supervisé :

1. initialiser les centres $c_j(0)$, les paramètres de la sortie de chaque neurone de la couche cachée $\sigma_j(0)$, et les poids synaptique $w_j(0)$, $j = 1, \dots, N_c$.
2. Choisir un taux d'entraînement $\alpha < 1$ variable ou fixe.
3. Adapter les poids comme suit :
 - Calcul de l'erreur quadratique en sortie E entre la réponse désirée et celle fournie par le réseau.

$$E = \sum_{p=1}^M \sum_{j=1}^n (d_j^p - y_j^p)^2 \quad (\text{IV.11.})$$

Où y_j^l d_j^l représente respectivement la $j^{\text{ème}}$ sortie du réseau et celle désirée.

- Réajuster les centres et les paramètres de fonction noyau :

$$\begin{cases} c_j(t+1) = c_j(t) + \alpha \Delta c_j(t) \\ \sigma_j(t+1) = \sigma_j(t) + \alpha \Delta \sigma_j(t) \end{cases} \quad (\text{IV.12.})$$

Avec :

$$\Delta c_j(t) = - \frac{\partial E}{\partial C_j^k(t)} \quad (\text{IV.13.})$$

$$\Delta \sigma_j(t) = - \frac{\partial E}{\partial \sigma_j^k(t)} \quad (\text{IV.14.})$$

- réajuster les poids synaptiques :

$$w_j(t+1) = w_j(t) + \alpha_w \Delta w_j(t) \quad (\alpha_w \text{ est le taux d'apprentissage des poids } w_j) \quad (\text{IV.15.})$$

Avec :

$$\Delta w_j(t) = - \frac{\partial E}{\partial w_j(t)}, \quad 1 \leq j \leq Nc; \quad 1 \leq i \leq n+f \quad (\text{IV.16.})$$

4. répéter l'étape 3 le nombre de fois d'itérations nécessaires jusqu'à l'obtention de la précision désirée en sortie.

$$E \leq E_{\text{Désirée.}}$$

Nous n'allons pas commenter en détails chacune de ces dérivés. Le calcul détaillé est donné dans l'annexe 1.

IV.5. PHASE DE PREDICTION

Cette phase est identique pour les deux modèles de différents algorithmes d'apprentissages ; après l'adaptation des paramètres du réseau, on présente à l'entrée du réseau le vecteur X_{n+1}^1 (ou X^1, X^2 en même temps), dont on ne connaît pas la sortie désirée Y_{n+1} . Le réseau calcule sa sortie Y_{n+1} et on la considère comme valeur exacte, la dimension de Y sera d'ordre $n+1$, on le traite comme étant un nouvel échantillon, c'est-à-dire qu'on refait l'apprentissage dès le début avec ces nouvelles valeurs.

Cette procédure est répétée jusqu'à ce que l'on arrive à prédire les m valeurs. A la fin, la dimension de Y sera $n+m$.

Nous présentons ci-dessous les organigrammes des différents modèles :

Modèle1 avec ROM :

Début

Donnés

- X, Y_d, TG, T_AR et T_A.
- Calculer T.

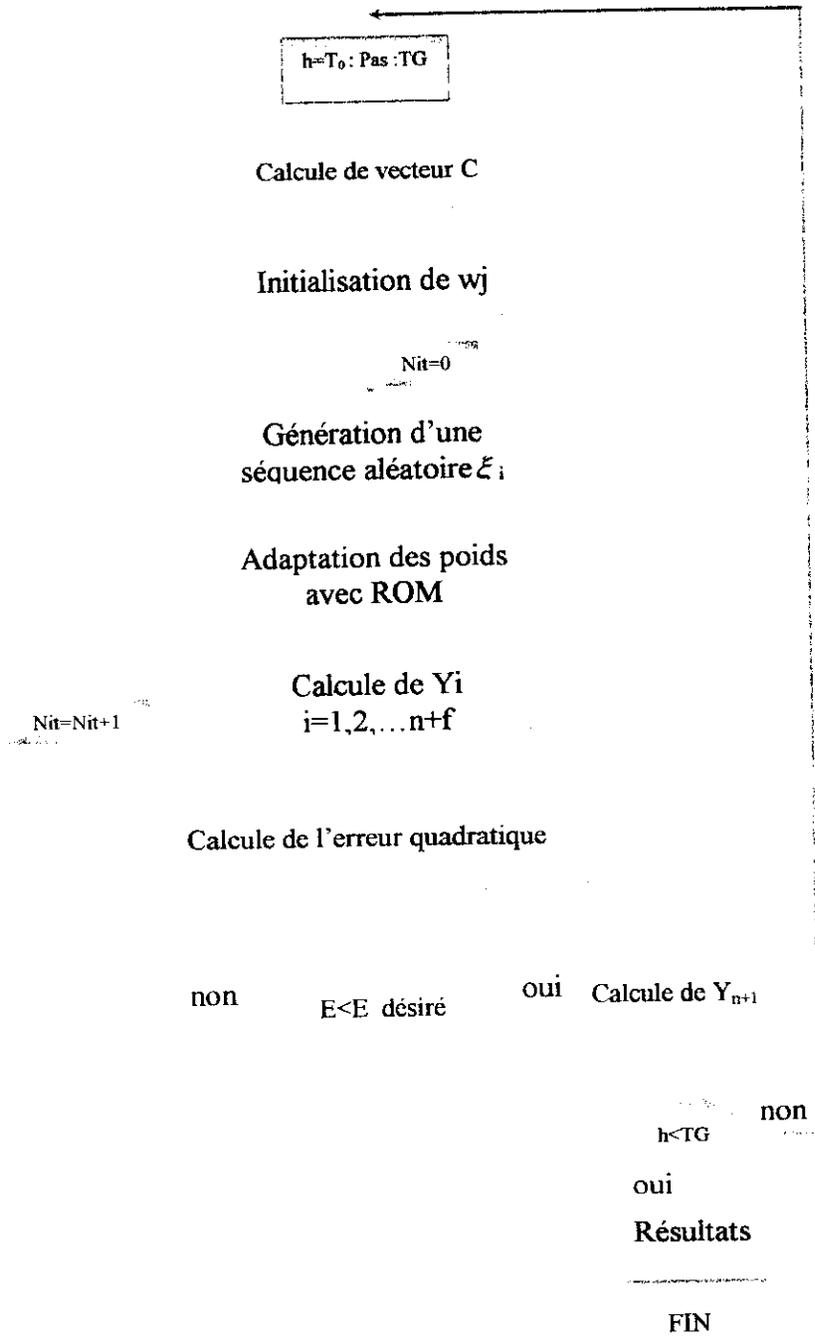


Fig.IV.3. Organigramme du réseau RBF

Modèle1 avec ACM :

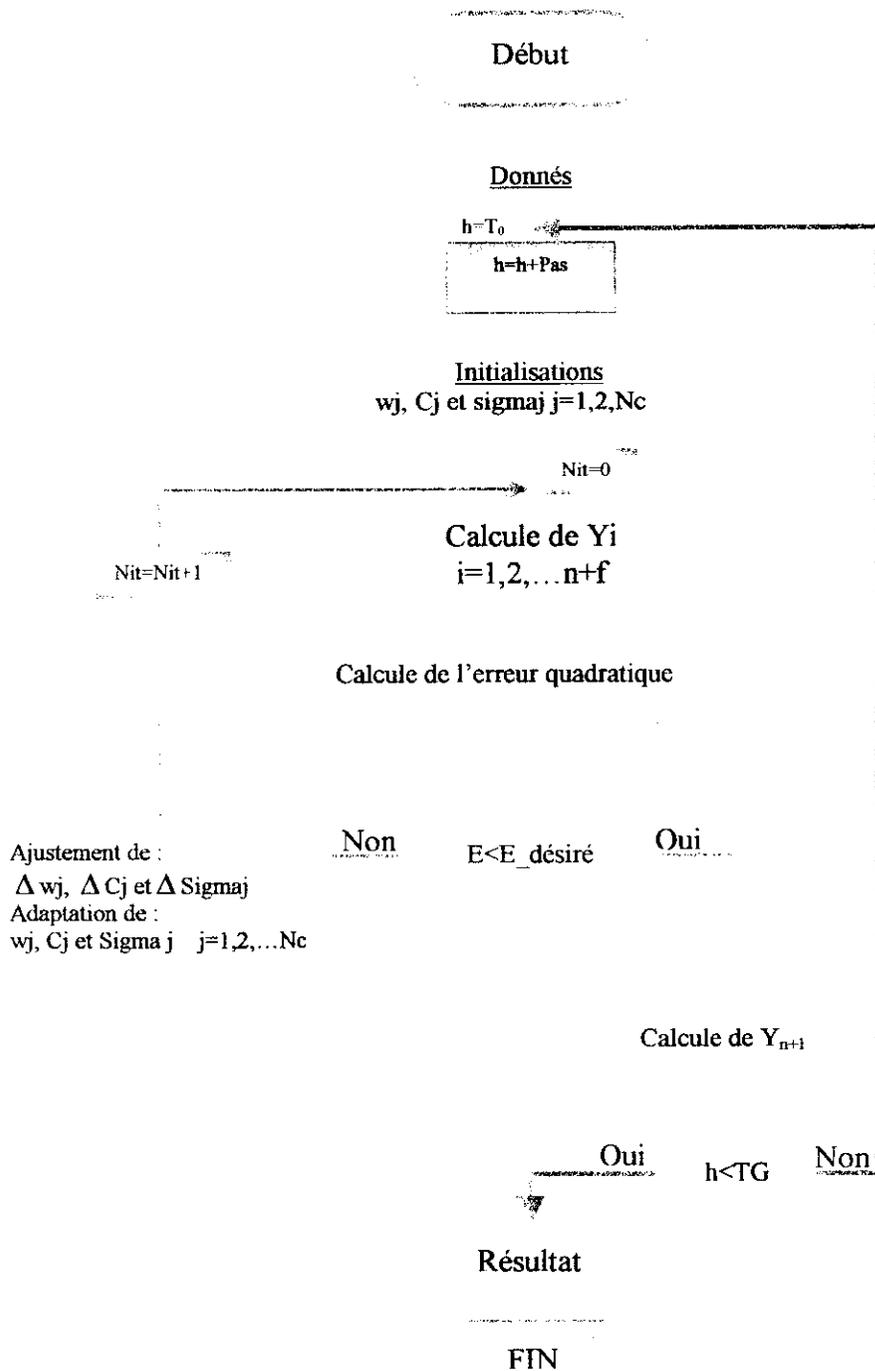


Fig.IV.4.Organigramme du réseau RBF_G

IV.6. EXEMPLE D' EXECUTION

Afin d'illustrer les possibilités de prédiction de nos programmes, nous présentons un exemple du réseau RBF1 :

La façon de dérouler le programme est la suivante :

1. Lancer l'exécution du programme, avec double clique sur « RNA.exe », une fenêtre principale apparaît pour nous aider à exécuter les différents programmes, de faire entrer et enregistrer nos données sous le nom approprié ainsi que visualiser les résultats, c'est l'interface graphique de l'application. Cette fenêtre est plus ou moins comme suit (elle dépend de système d'exploitation utilisé) :

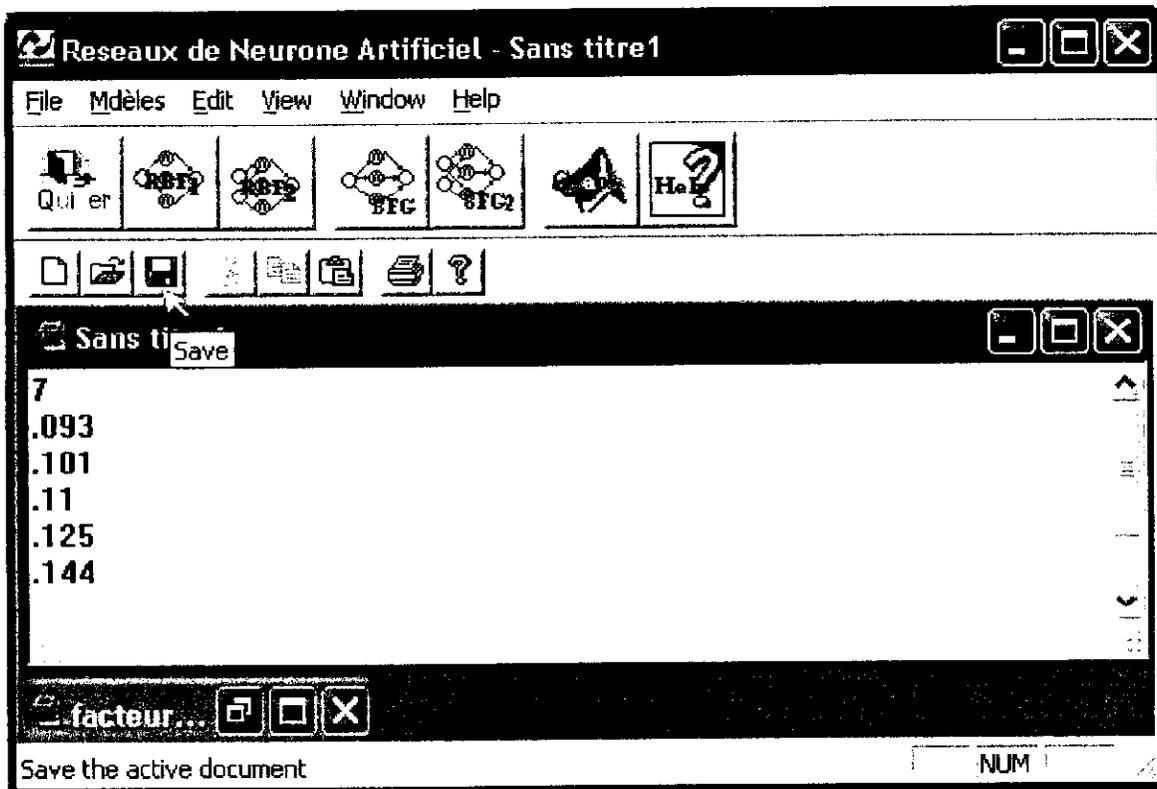


Fig.IV.5. L'interface graphique de l'application

2. Sauvegarder ce fichier en donnant un nom approprié, l'extension est « * .rbf ». Remarque la forme des données que nous sauvegardons, D'abord le nombre de points puis le vecteur Y.

3. Ensuite, choisir l'un des trois programmes (une simple clique sur la barre d'outil) par exemple RBF1. Donc le programme RBF1 s'exécute. Après quelques lignes de présentation, il vous propose le choix suivant :

Voulez vous utiliser les valeurs enregistrées auparavant ?

Taper (O)ui ou (N)on.

Selon le choix, le programme demande d'autres informations.

-Si oui alors, le programme va demander le nom de ce fichier.

Donner le nom de ce fichier Nom : _

Editer un nom en spécifiant l'extension rbf, notons que nous pouvons utilisé d'autres formats (*.txt, *.dat, *.Doc...).

Finalement, il vous demande les paramètres de l'apprentissage que nous avons présenté dans le paragraphe concernant le codage de l'information codage de l'information.

-Si vous choisi non vous avez encore la chance d'éditer nos donnés mais dans ce cas, elles sont enregistrées sous le nom par défaut de « Donnee.rbf ».

La fenêtre DOS pour le premier choix est comme suis :

```

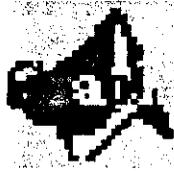
G:\RBF_GVRbFG1.exe
Voulez vous utilies les valeurs enregistres au paravant
Taper<O>ui ou <N>on
oui
Donner le Nom de ce fichier Nom:  facteur.rbf
Quel est le pas d'abcisse
Pas= 50
Donner l'abscisse initiale
X[1]= 0
Choisir l'Abcisse_Globale
A_G= 5000
Fixer le Temps d'Apprentissage_Relatif par rapport a A_G
A_R(<%)= 30
Fixer le pas d'Apprentissage<Mx pas,M=1,2,...>
Deltat= 500
Choisir le taux de recouvrement
Taux(<%)= 98
Choisir le nombre de neurones dans la couche cachee
N_c= 5
Confermer les donnes ...Taper <O>ui
o
Erreur=9.999855e-003
Le point predit de Y[5]est:.... 0.101733
- Erreur=2.134121e-002
  
```

FIG .IV.6. La fenêtre DOS dans le cas oui pour RBFG1

Répondre aux les questions, une fois que cela est fait, après la fin de l'exécution de programme, ouvrir le fichier « resultat.rbf » c'est le nom par défaut qu'on a proposé, il

contient les résultats de l'exécution (n'oubliez pas de le sauvegarder sous un autre nom car à la prochaine exécution les résultats seront écrasés).

4- D'autres possibilités nous permettent de tracer directement la courbe de prédiction en cliquant sur « graphe » dans la barre d'outil.



Là encore, nous pouvons sauvegardé la figure en spécifiant l'extension *.fig, comme nous pouvons également manipuler la courbe, par exemple agrandir une zone pour bien voir l'erreur commise, ou bien le représenter en 3D, imprimer ou ouvrir une autre figure.

(L'installation du logiciel MATLAB6.0 est indispensable pour l'exécution de ce programme).

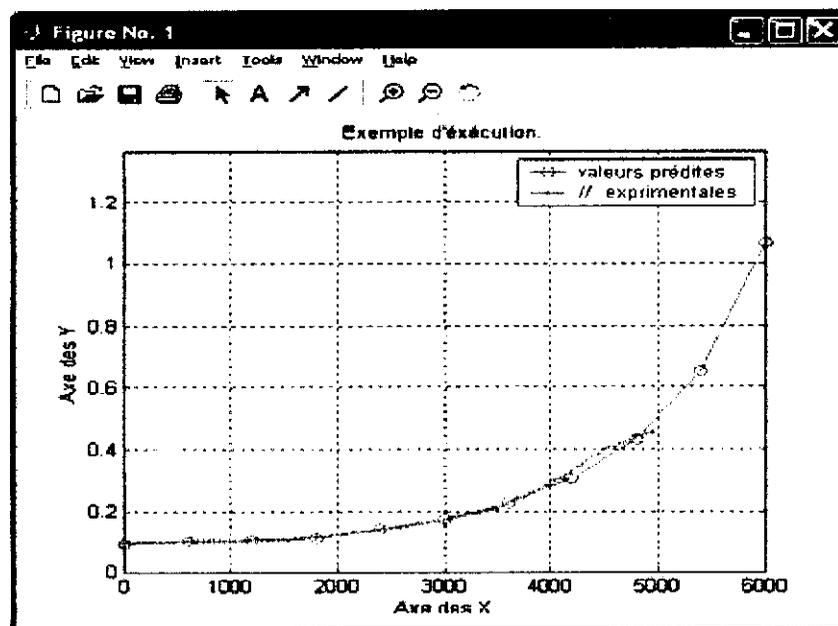
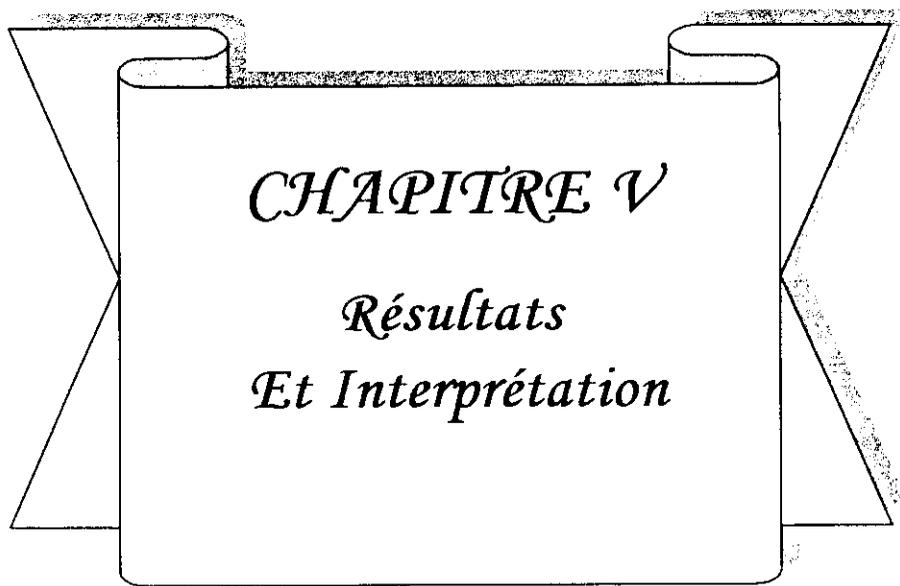


FIG. IV.7. Exemple de exécution

Les courbes du chapitre suivant sont tracées de cette manière mais avec quelques traitements en spécifiant les paramètres utilisés.

IV.7. CONCLUSION

Nous avons proposé dans ce chapitre différentes architectures des réseaux de neurones. Ces réseaux ont été utilisés dans nos applications, pour prédire quelques propriétés de l'huile minérale (facteur des pertes diélectriques et permittivité relative), les résultats ainsi que leurs intégrations sont présentés dans le chapitre suivant.



CHAPITRE V

Résultats

Et Interprétation

V.1. INTRODUCTION

Dans ce chapitre, nous présentons quelques résultats de prédiction des trois programmes que nous avons élaborés et ce, en jouant sur les paramètres suivants :

- L'intervalle d'apprentissage.
- Le pas d'apprentissage.
- Le nombre de neurones dans la couche cachée.

Notre intérêt est porté sur la recherche des paramètres communs qui nous donnent une bonne prédiction pour toutes les courbes à utiliser en s'approchant de ces courbes pratiques, c'est-à-dire, choisir un pas en heures où le manipulateur peut prendre des valeurs, et d'essayer d'optimiser le temps d'apprentissage de telle sorte qu'il soit le plus court possible, cela pour voir le comportement du réseau dans un intervalle élevé.

Les résultats prédits vont être comparés aux valeurs de la courbe expérimentale. S'ils coïncident avec une certaine marge d'erreur, on juge la prédiction comme bonne, et que notre réseau a une certaine fiabilité de prédire des points à l'extérieur de la courbe expérimentale.

Les résultats de prédiction présentés dans ce chapitre, concernent deux propriétés, facteur des pertes diélectriques et la permittivité relative, à différentes températures pour différentes températures d'essai.

Une comparaison entre les trois programmes est effectuée et à partir de laquelle nous sélectionnons le meilleur.

V.2. RESULTATS A PARAMETRE CONSTANTE

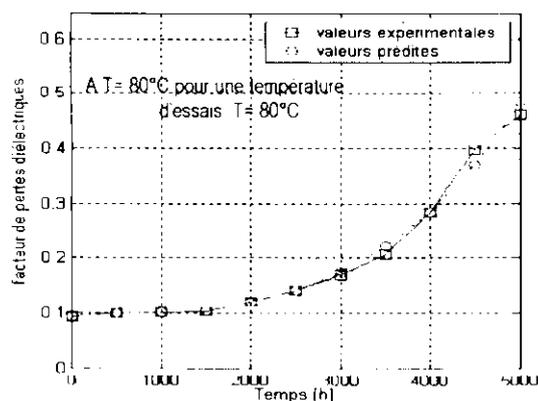
Après la recherche de la bonne configuration de notre réseau (taux de recouvrement et nombre de neurones dans la couche cachée pour un temps d'apprentissage 2500 heures et un pas de 500 heures), nous avons obtenus des paramètres qui donnent une bonne prédiction pour la majorité des propriétés utilisées. Ces paramètres sont fixés pour tous les réseaux, c'est-à-dire un taux de recouvrement $\tau = 98\%$ et un nombre de neurones de la couche cachée $N_c = 5$.

V.2.1. RESULTATS LIES AU RESEAU RBF1

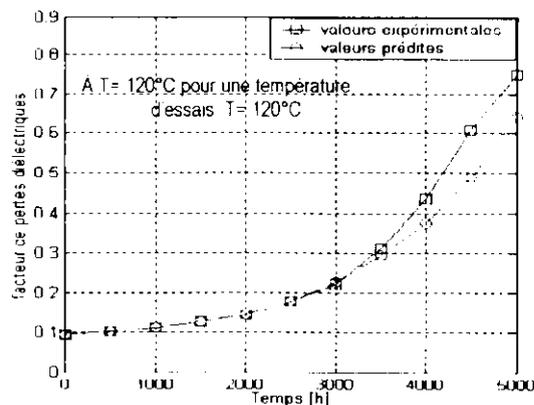
Voici les résultats obtenus par le réseau RBF1 avec un pas constant de 500h et un temps d'apprentissage de 2500h, pour tous les propriétés qui concernent le facteur des pertes diélectriques et la permittivité relative.

D'après les résultats représentés ci-dessous, nous pouvons dire que ce réseau à bien prédit la courbe (a) pour le facteur des pertes diélectrique avec un erreur moyenne inférieur à 7% (Tableau V.1). Ainsi que pour toutes les propriétés de la permittivité relative avec un erreur moyenne acceptable inférieur à 2.5% (Tableau V.2).

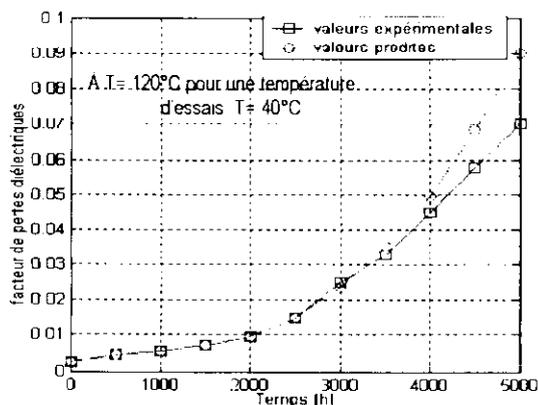
Malgré que la marge d'erreur soit relativement importante pour le reste des courbes (Fig.V.1. b, c et d) de facteur des pertes diélectriques, la sortie du réseau suit l'allure de ces courbes expérimentales.



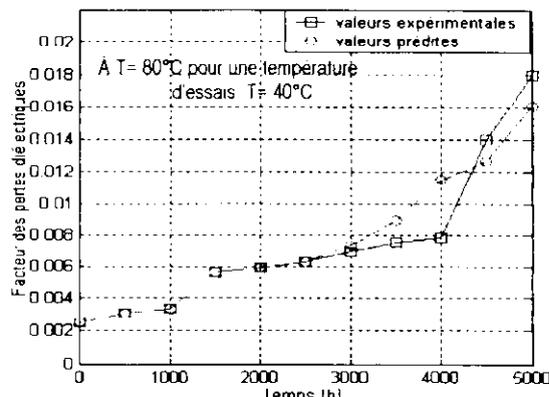
(a)



(b)



(c)

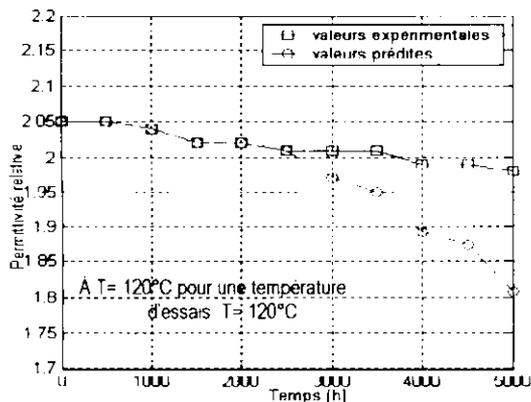


(d)

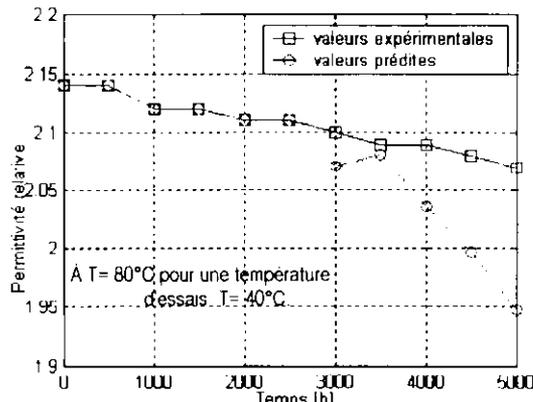
Fig.V.1. Facteur des pertes diélectriques en fonction du temps de vieillissement

	(a)	(b)	(c)	(d)
L'erreur relative minimale	0.0335	2.8420	5.7046	6.6282
L'erreur relative maximale	6.8870	19.5154	28.2763	45.2672
L'erreur relative moyenne	4.1335	11.282	14.0025	18.2070

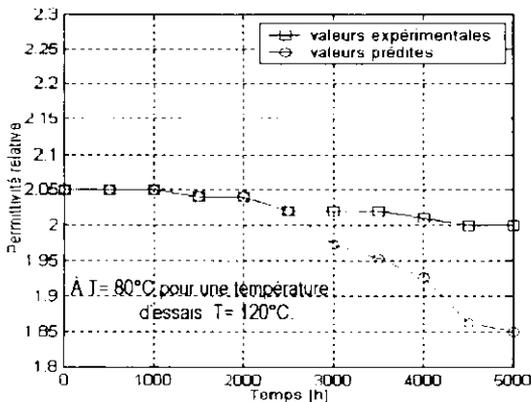
Tableau V.1. L'erreur relative concernant le facteur des pertes diélectriques.



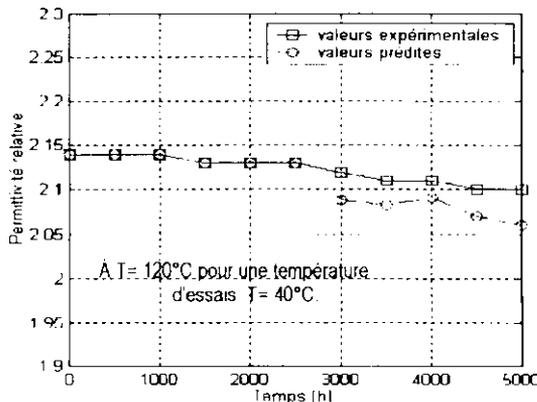
(a)



(b)



(c)



(d)

Fig.V.2. La permittivité relative en fonction du temps de vieillissement.

	(a)	(b)	(c)	(d)
L'erreur relative minimale	1.9793	0.4160	2.2846	0.8986
L'erreur relative maximale	8.6649	5.8976	7.5413	1.8712
L'erreur relative moyenne	4.8996	2.8551	4.8025	1.4035

Tableau V.2. L'erreur relative concernant la permittivité relative.

V.2.2. RESULTATS LIES AU RESEAU RBFG1

Avec le réseau RBFG1, nous ferons la même procédure et avec les mêmes paramètres que RBF1.

La sortie du réseau en générale, suite l'allure des courbes expérimentales mais avec une erreur appréciable pour les courbes (c) et (d) de la figure V.3., contrairement à la deuxième propriété, où nous remarquons que le réseau donne une bonne prédiction avec une erreur moyenne inférieure à 3.5% (Tableau V.4.), mais dans ce cas l'allure des courbes prédites est erronée.

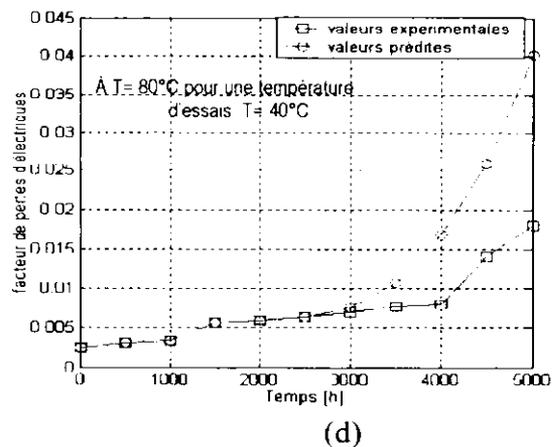
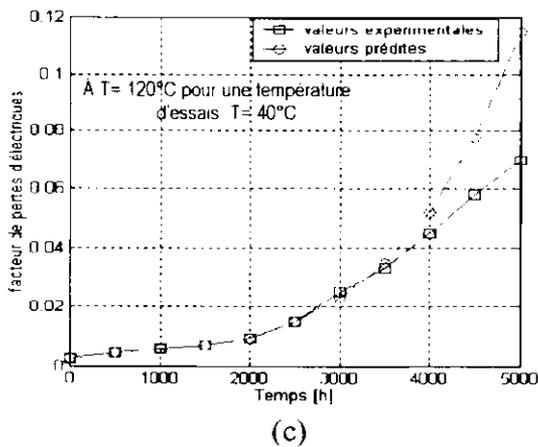
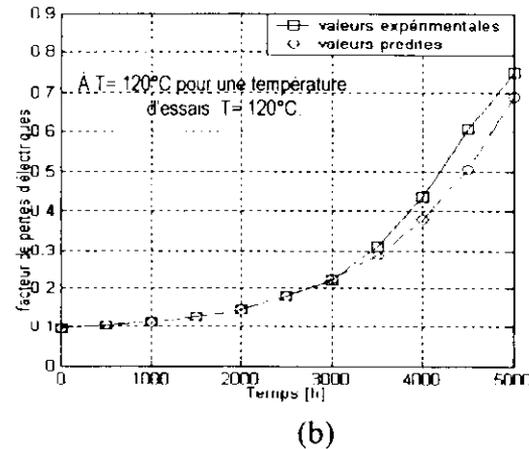
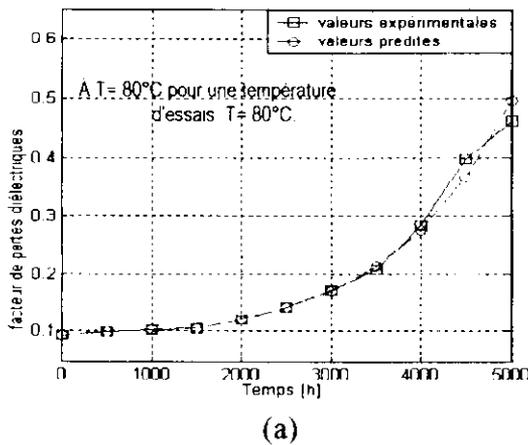


Fig.V.3.Facteur des pertes diélectriques en fonction du temps de vieillissement

	(a)	(b)	(c)	(d)
L'erreur relative minimale	1.0391	1.5707	5.0277	8.4005
L'erreur relative maximale	8.0462	16.7299	63.60966	123.3011
L'erreur relative moyenne	4.5484	9.35371	25.26580	74.3059

Tableau V.3. L'erreur relative concernant le facteur des pertes diélectriques.

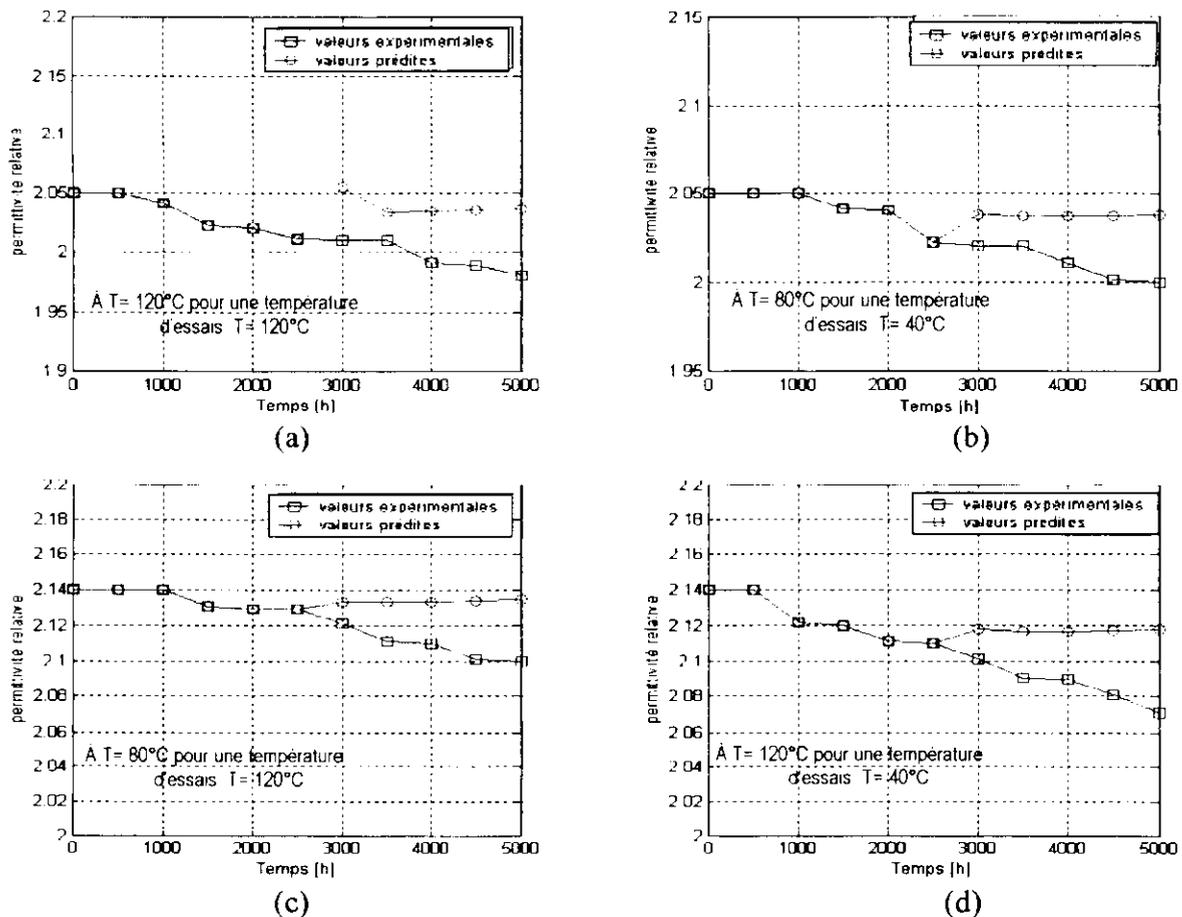


Fig.V.4. La permittivité en fonction du temps de vieillissement

	(a)	(b)	(c)	(d)
L'erreur relative minimale	1.2434	0.6184	0.3682	0.1556
L'erreur relative maximale	6.7372	5.9843	5.3049	4.9439
L'erreur relative moyenne	3.4714	3.1321	2.3601	2.1805

Tableau V.4. L'erreur relative concernant la permittivité relative.

V.2.3. RESULTATS LIES AU RESEAU RBFG2

Pour ce réseau les résultats sont représentés dans les figures V.5. et V.6 avec un pas égale 500h et un temps d'apprentissage égale 2500h.

Dans ce cas, les différentes valeurs des erreurs relatives ne donnent pas une expression réelle de ce qui se passe, car les derniers points divergent fortement, ce qui augmente l'erreur moyenne. Dans ce cas (Fig. V.6.) nous jugeons que la prédiction est bonne si nous limitons la prédiction à 4300h.

Nous remarquons aussi que le réseau a atteint une erreur minimale relative inférieure à 1%
Les résultats des autres réseaux restent toujours les meilleurs.

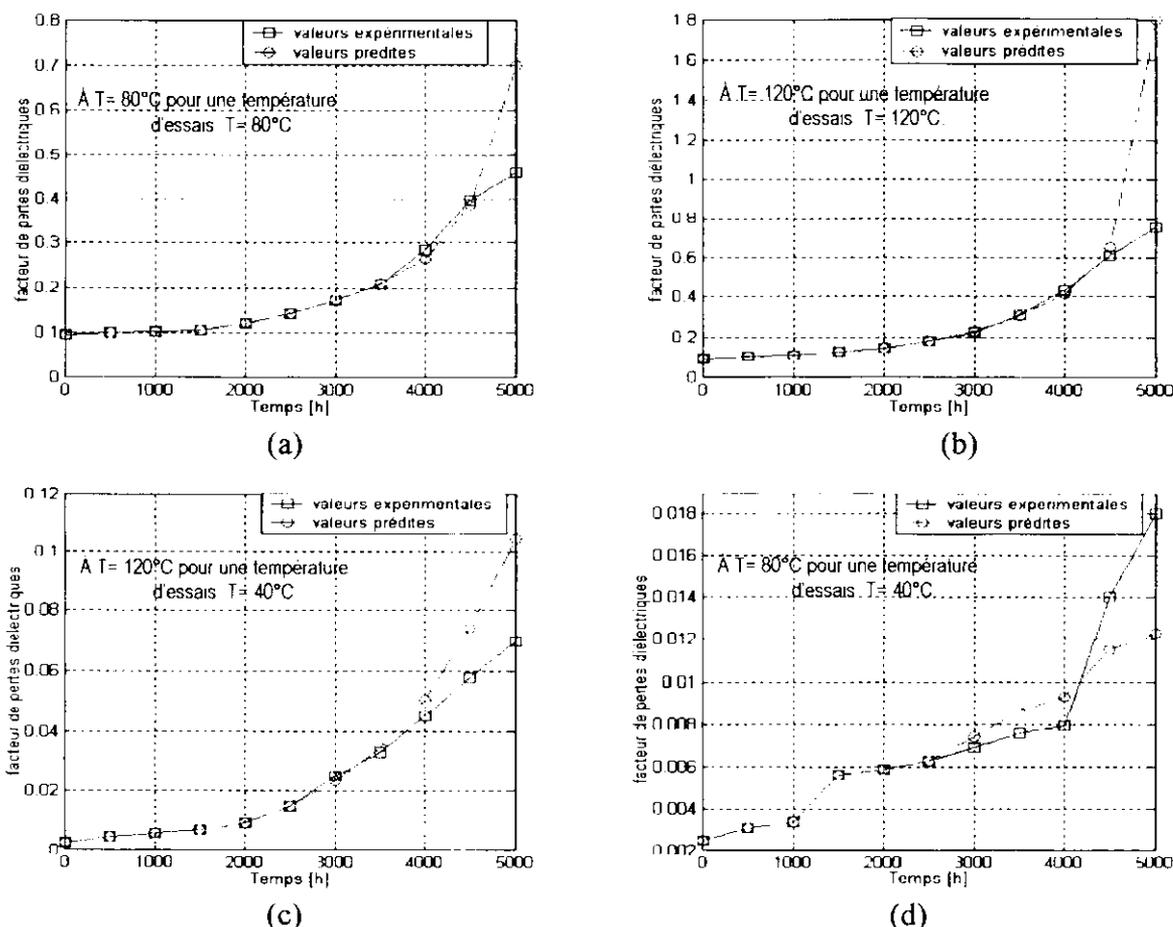
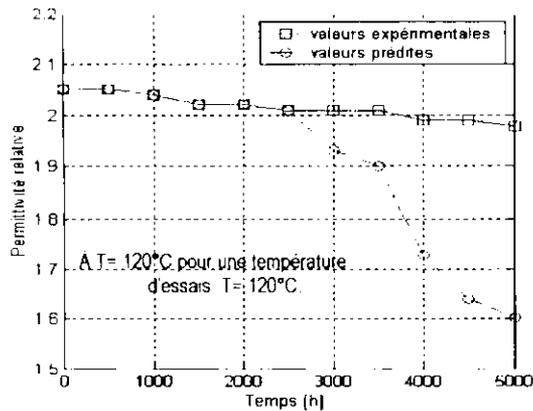


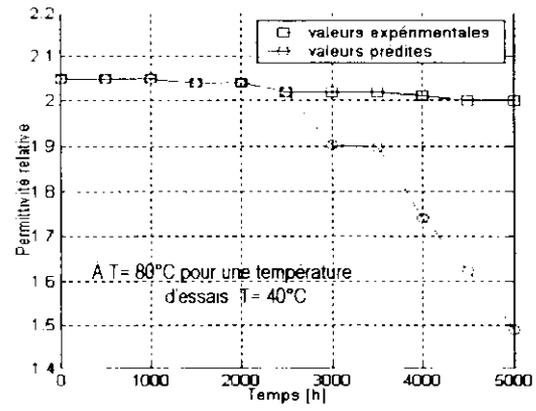
Fig.V.5. Facteur des pertes diélectriques en fonction du temps de vieillissement

	(a)	(b)	(c)	(d)
L'erreur relative moyenne	12.9476	20.8749	20.2224	17.3065
L'erreur relative minimale	0.9251	0.3340	5.5975	7.0164
L'erreur relative maximale	52.2643	54.8957	48.8349	31.5642

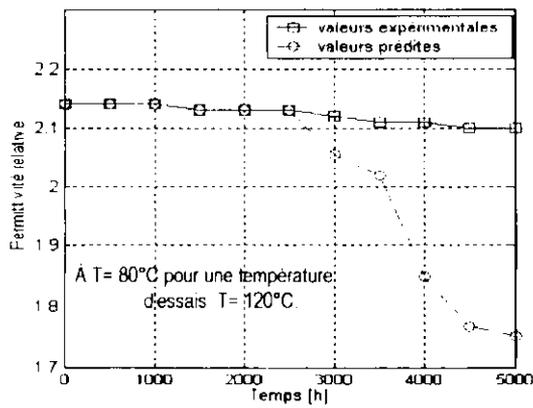
Tableau V.5. L'erreur relative concernant le facteur des pertes diélectriques.



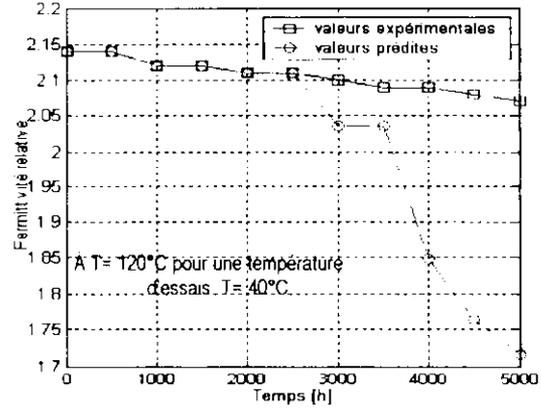
(a)



(b)



(c)



(d)

Fig.V.6. La permittivité en fonction du temps de vieillissement

	(a)	(b)	(c)	(d)
L'erreur relative moyenne	11.8000	9.8976	13.9208	10.445
L'erreur relative minimale	3.7720	2.5683	5.8382	1.9804
L'erreur relative maximale	19.0288	17.0926	25.5310	16.6737

Tableau V.6. L'erreur relative concernant la permittivité relative.

V.3. RESULTATS A PARAMETRES VARIABLES

Pour tester nos réseaux, nous avons pris trois pas pour chaqu'un des trois temps d'apprentissage, pour deux propriétés que nous avons choisie (la propriété -a- de facteur des pertes diélectriques et la propriété -a- de permittivité diélectrique cité ci-dessus)

V.3.1. RESULTATS LIES AU RESEAU RBF1

Dans ce cas, nous avons pris trois temps d'apprentissages en heures différents (1500h, 2500h et 3500h). Pour chaque temps, nous avons pris trois pas d'apprentissages (250h, 500h et 750h).

- Temps d'aprantissage de 1500h (30%)

Pour les deux propriétés (Fig.V.7 et Fig.V.8), il n'y a pas un pas d'apprentissage permettant d'avoir une convergence entre les résultats prédits et les résultats expérimentaux. Donc le temps d'apprentissage 1500h n'a pas donné de résultats encourageants.

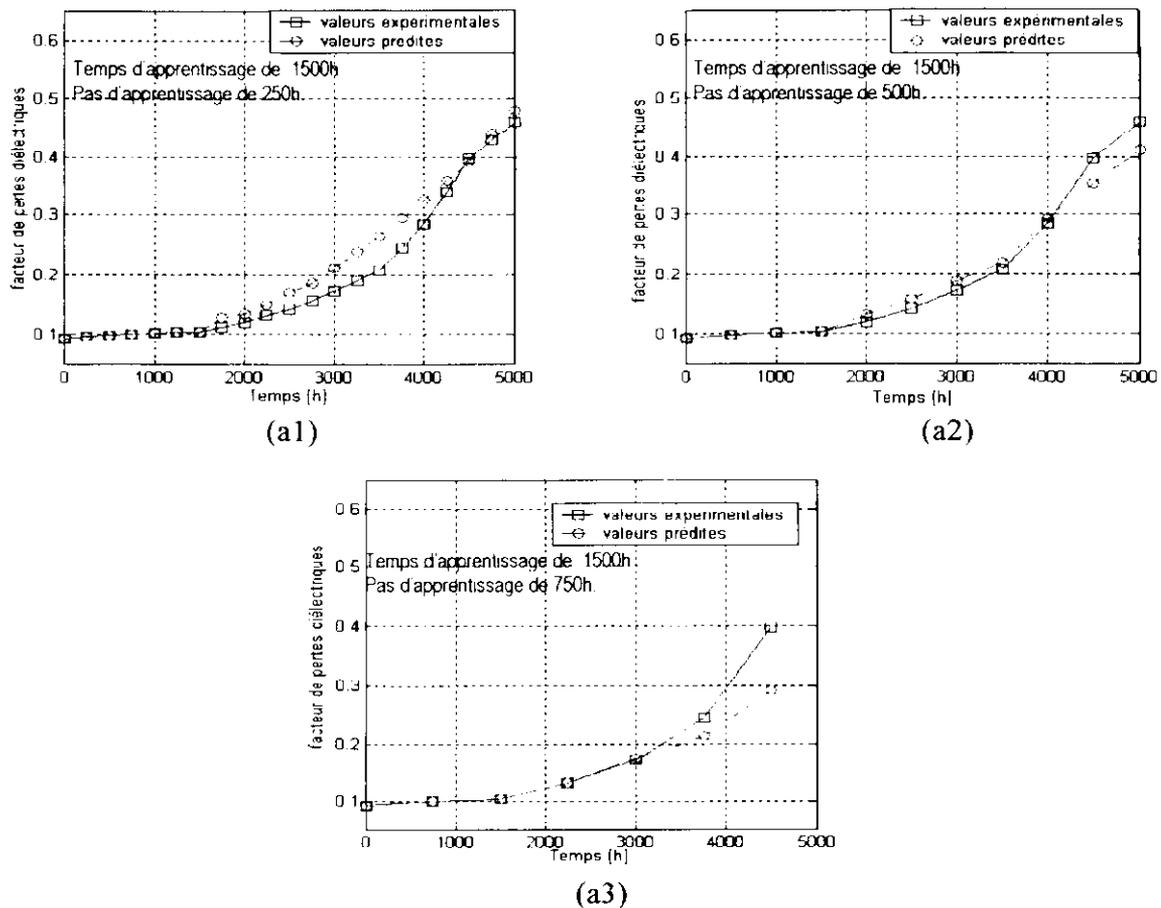


Fig.V.7.Facteur de pertes diélectriques en fonction du temps de vieillissement

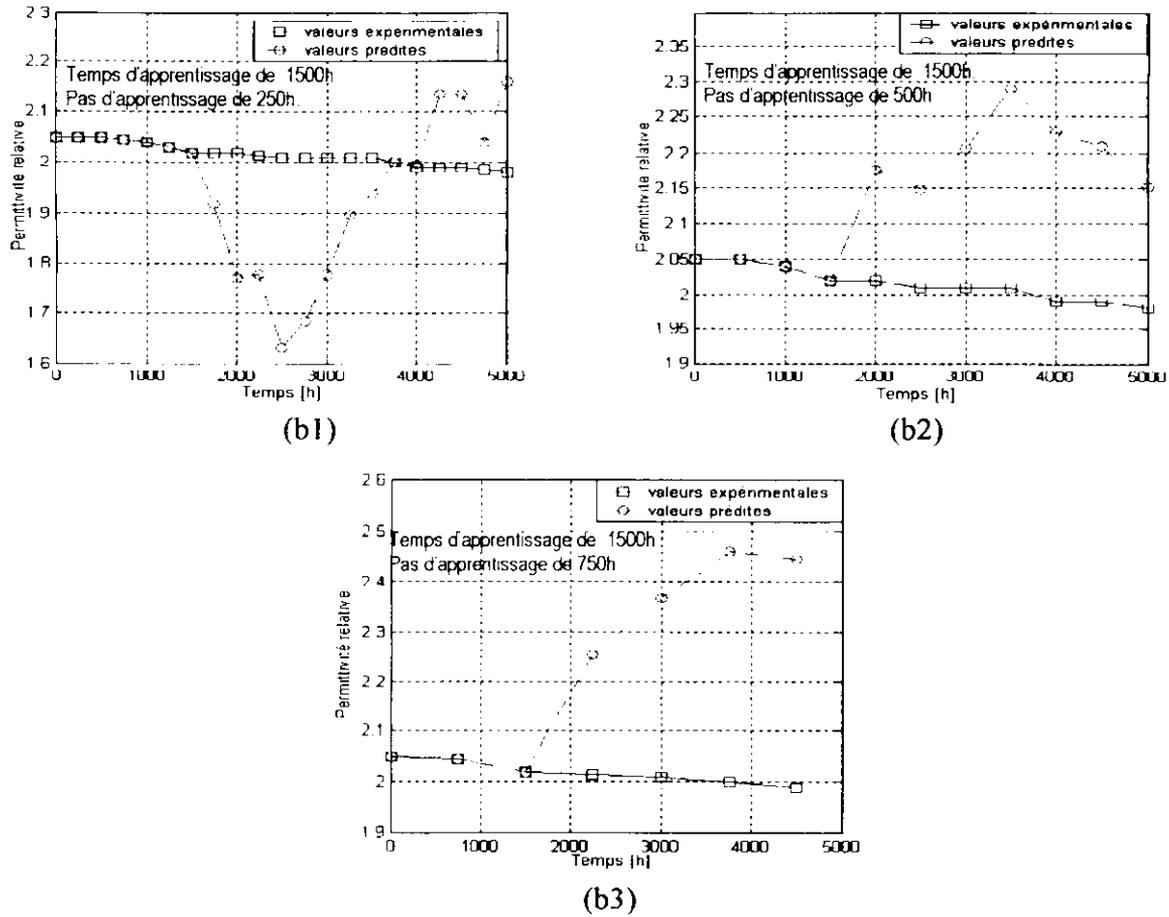


Fig.V.8. La permittivité relative en fonction du temps de vieillissement

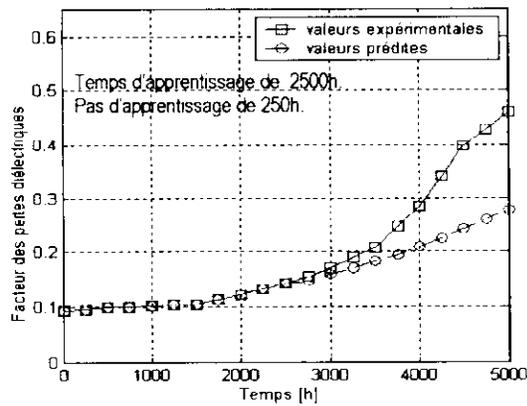
	(a1)	(a2)	(a3)	(b1)	(b2)	(b3)
Le pas relatif	16.6667	33.3333	50	16.6667	33.3333	50
L'erreur relative moyenne	34.6834	8.6735	10.7592	10.6544	12.5275	16.8702
L'erreur relative minimale	16.2656	2.9622	1.0655	1.4915	6.8507	5.2657
L'erreur relative maximale	55.5380	10.9206	26.8895	18.4122	23.6822	37.2747

Tableau V.7. L'erreur relative et le pas relatif concernant les deux propriétés.

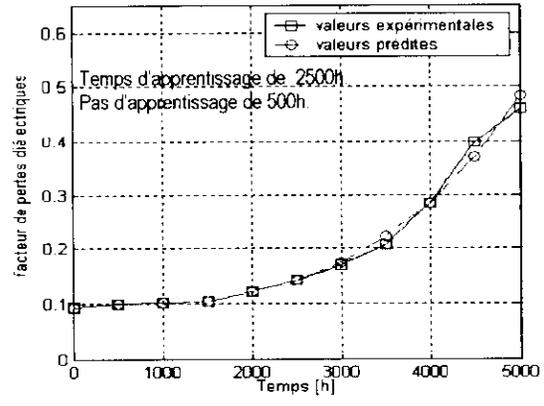
• Temps d'apprentissage de 2500h (50%)

La courbe (a2) de la figure V.9 coïncide avec celle de la courbe expérimentale. Mais pour la courbe (a1) de la même figure, nous constatons le phénomène de sur apprentissage.

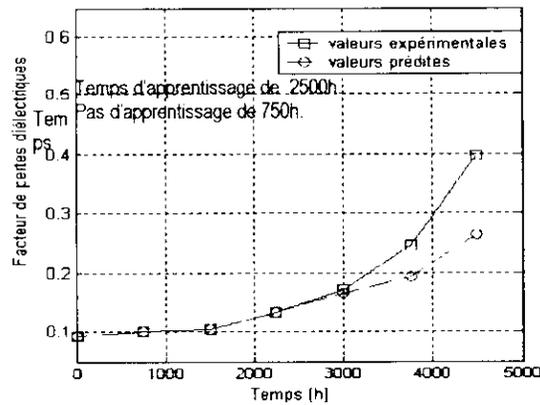
Et pour la permittivité relative, le réseau a bien prédit cette propriété avec une erreur moyenne voisine de 5% pour tous les pas considérés.



(a1)



(a2)



(a3)

Fig.V.9.Facteur de pertes diélectriques en fonction du temps de vieillissement

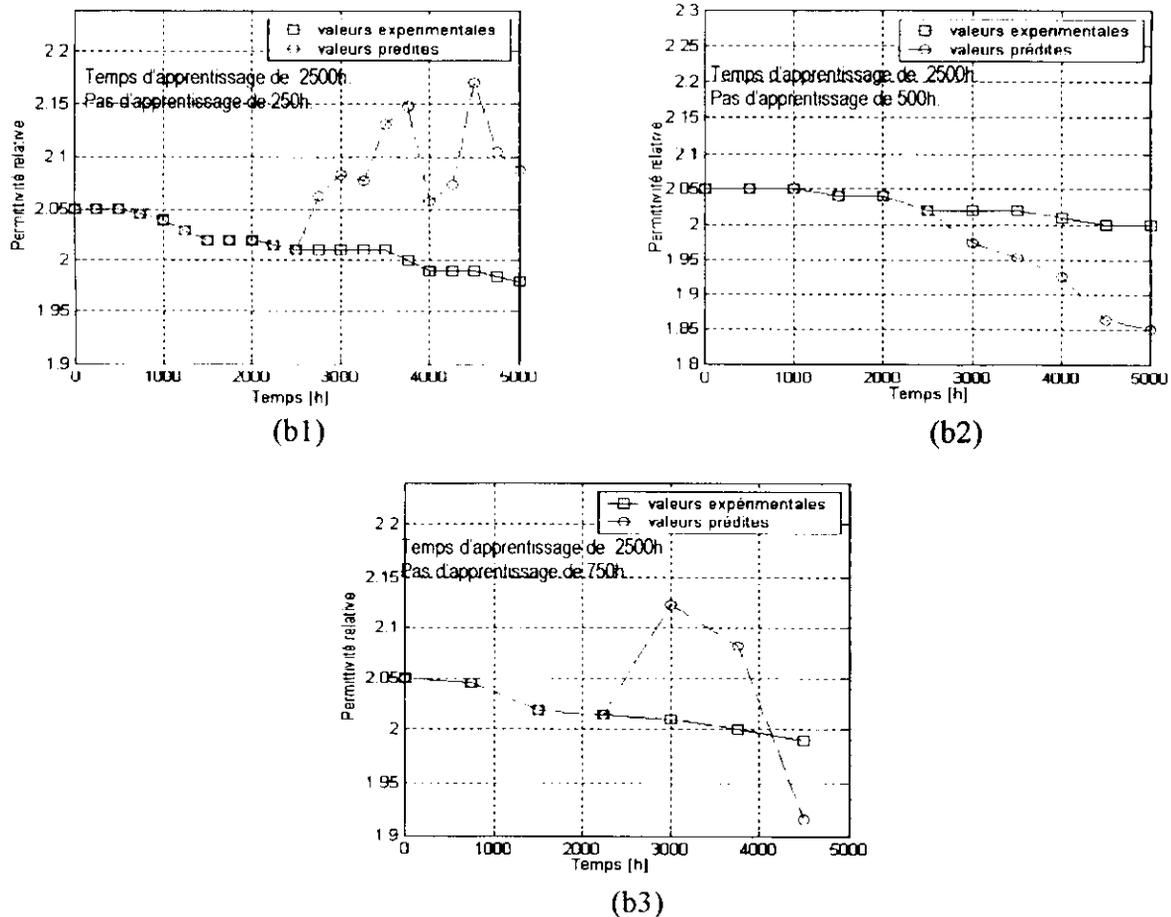


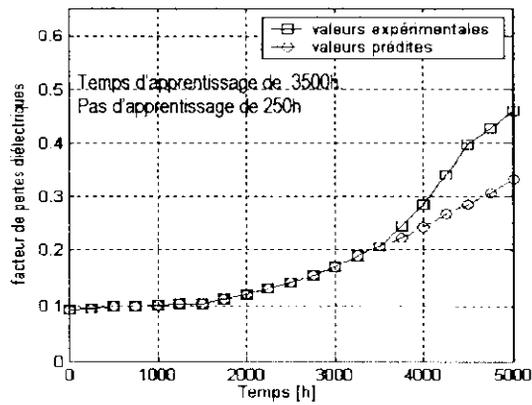
Fig.V.10. La permittivité relative en fonction du temps de vieillissement

	(a1)	(a2)	(a3)	(b1)	(b2)	(b3)
Le pas relative	10	20	30	10	20	30
L'erreur relative moyenne	23.7362	4.1335	19.6555	5.1040	4.8996	4.5046
L'erreur relative minimale	5.7858	0.0335	3.4579	2.6102	1.9793	3.7637
L'erreur relative maximale	39.772	6.8870	33.4930	9.0829	8.6649	5.6238

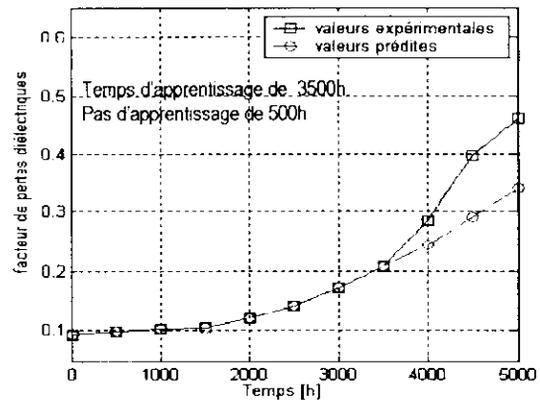
Tableau V.8. L'erreur relative concernant le facteur des pertes diélectriques.

- Temps d'apprentissage de 3500h (70%)

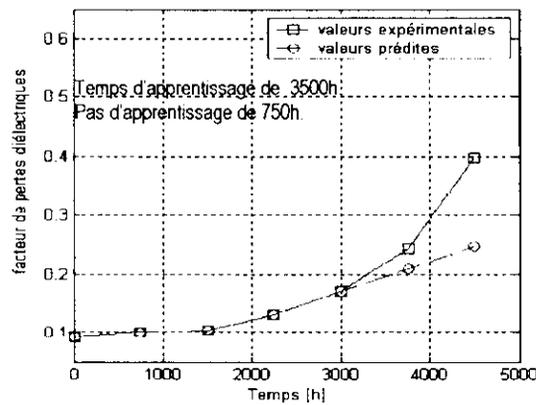
On remarque que le réseau est totalement divergeant pour la première propriété. Mais il donne de bons résultats pour la deuxième.



(a1)



(a2)



(a3)

Fig.V.11. Facteur des pertes diélectriques en fonction du temps de vieillissement

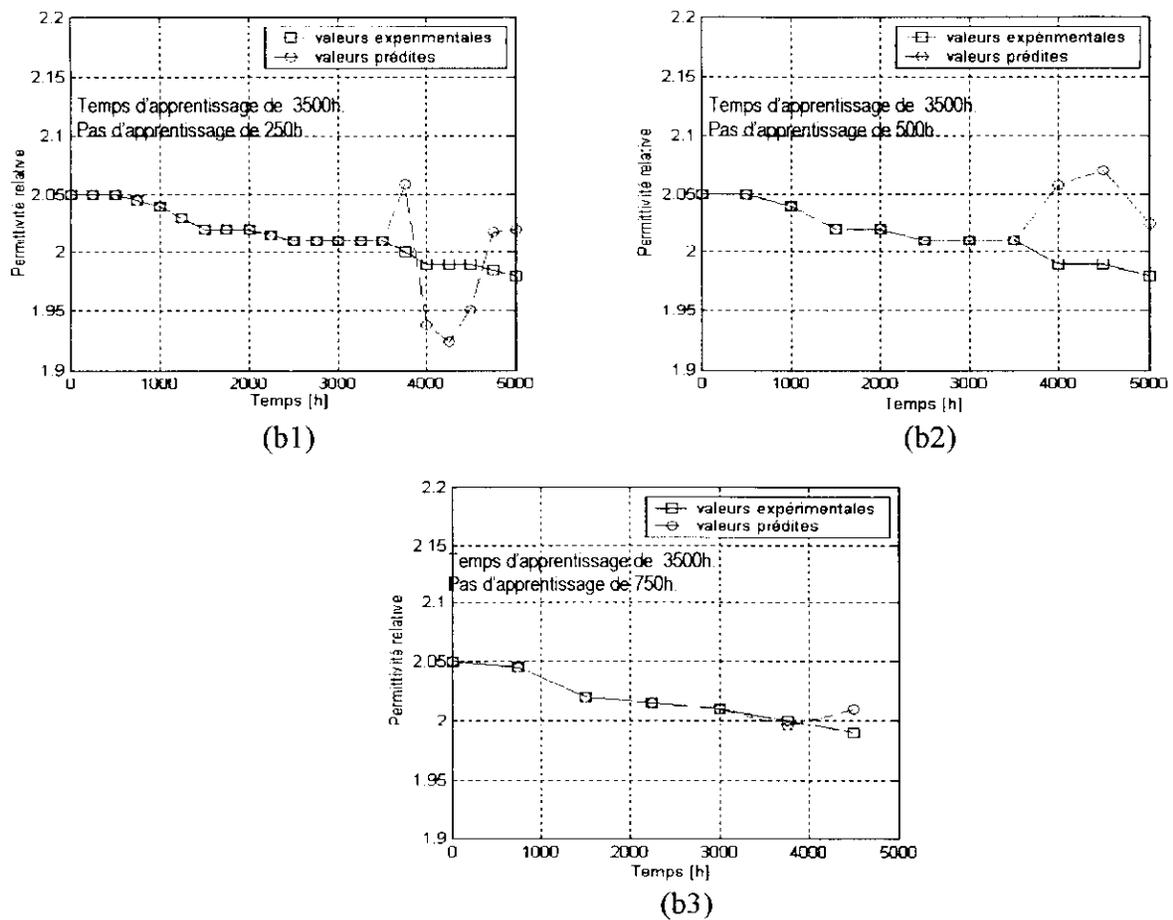


Fig.V.12. La permittivité en fonction de temps du vieillissement

	(a1)	(a2)	(a3)	(b1)	(b2)	(b3)
Le pas relatif	7.1429	14.2857	21.2857	7.1429	14.2857	21.2857
L'erreur relative moyenne	21.7093	22.6168	26.0555	2.4175	3.2510	0.6132
L'erreur relative minimale	8.7636	14.6811	14.9539	1.6693	2.2729	0.2181
L'erreur relative maximale	28.8475	26.8508	37.1572	3.2934	4.0654	1.008

Tableau V.9. L'erreur relative concernant le facteur des pertes diélectriques.

V.3.2. RESULTATS LIES AU RESEAU RBFG1

Toujours avec les trois pas 250h, 500h et 750h qui sont utilisés dans le réseau RBF1, le réseau RBFG1 nous donne ces résultats.

- Temps d'apprentissage de 1500h (30%)

Toujours nous remarquons que pour la première propriété, le réseau suit la même allure mais avec une certaine convergence pour la courbe (a2) de la figure V.13 et un sur- apprentissage de celle de (a1) de la même figure.

Et pour la deuxième propriété le réseau donne des bons résultats (Tableau.V.10. et Fig.V.14).

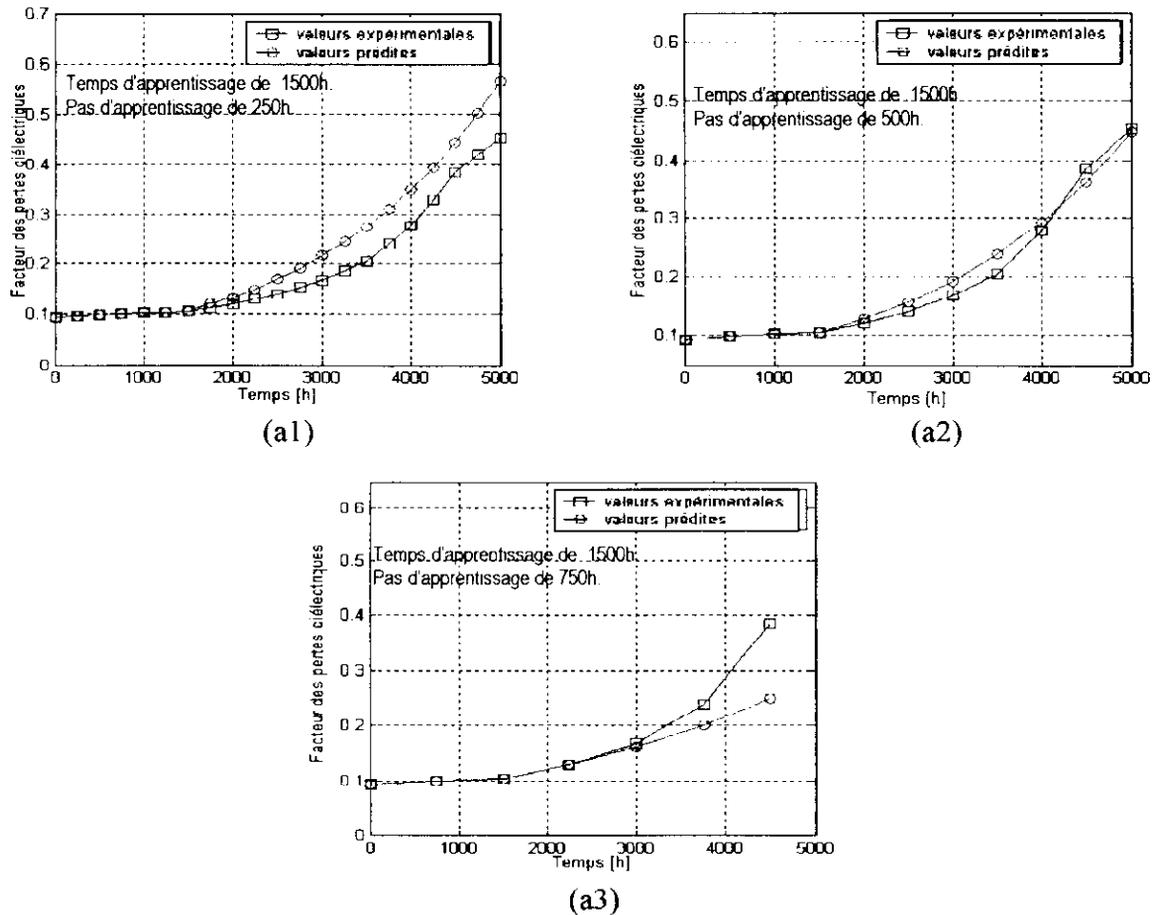
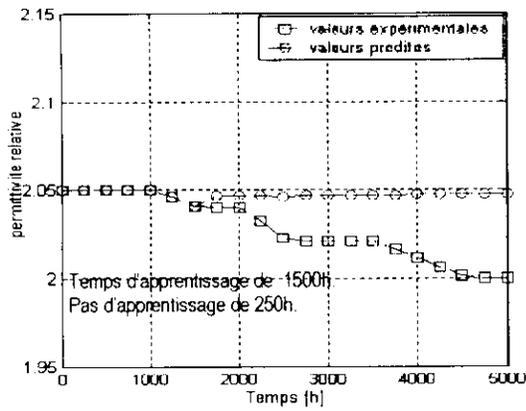
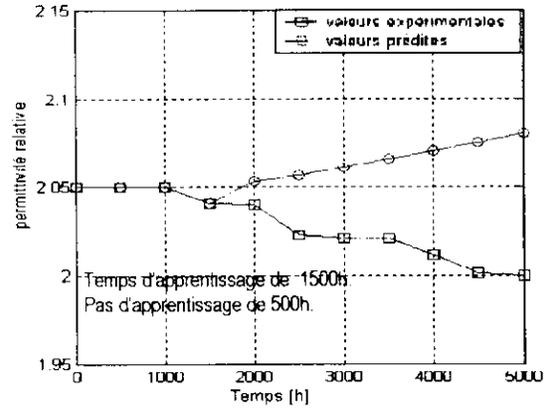


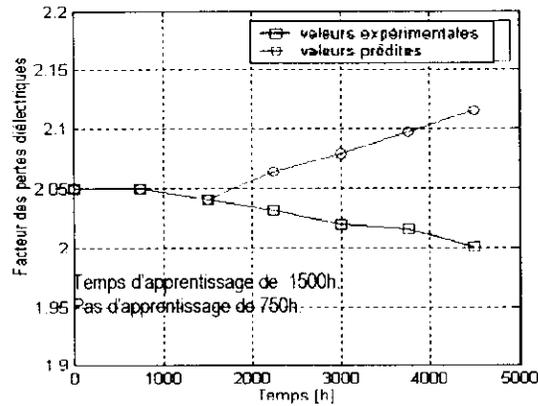
Fig.V.13. Facteur des pertes diélectriques en fonction du temps de vieillissement



(b1)



(b2)



(b3)

Fig.V.14. La permittivité relative on fonction du temps de vieillissement

	(a1)	(a2)	(a3)	(b1)	(b2)	(b3)
Le pas relatif	16.6667	33.3333	50	16.6667	33.3333	50
L'erreur relative moyenne	16.7799	9.1250	56.7773	1.4484	2.4952	3.5876
L'erreur relative minimale	0.1565	1.3538	9.1080	0.3231	0.6475	1.6283
L'erreur relative maximale	36.1335	16.8651	99.5977	2.3765	4.0425	5.6967

Tableau V.10. L'erreur relative concernant le facteur des pertes diélectriques.

• Temps d'apprentissage de 2500h (50%)

Nous remarquons que ce réseau donne de bons résultats pour toutes les propriétés et pour tous les pas (Fig.V.15 et Fig.V.16). Seule la caractéristique (a3), diverge de la figure V.15.

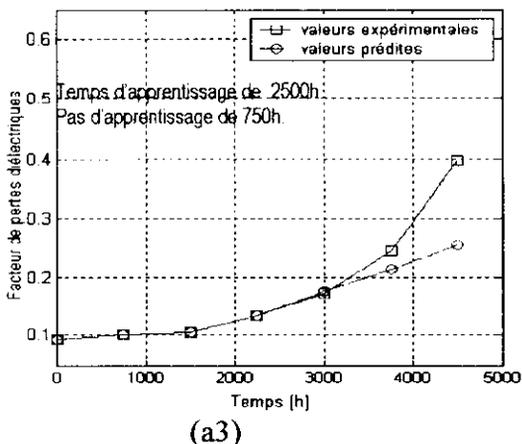
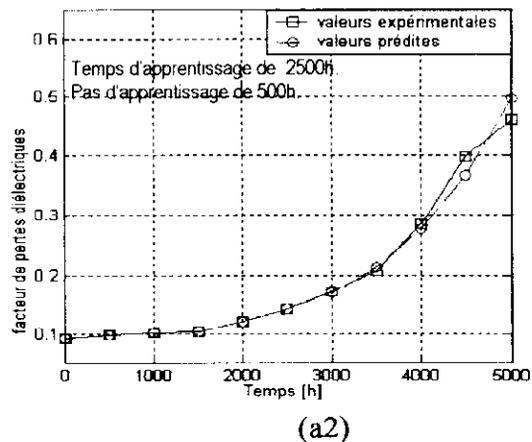
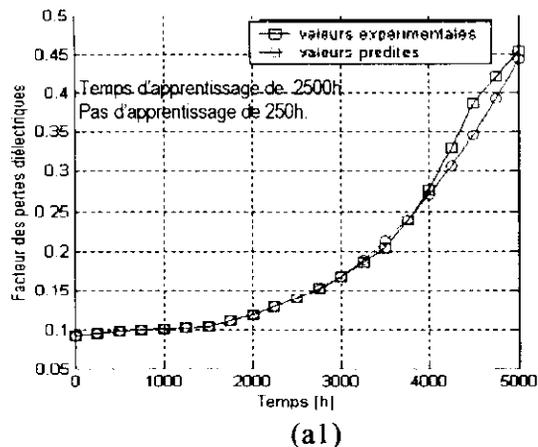
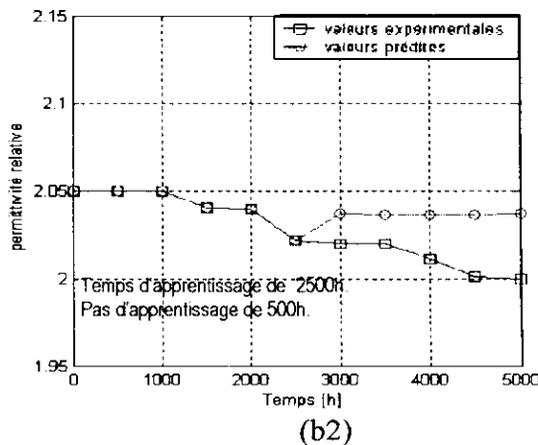
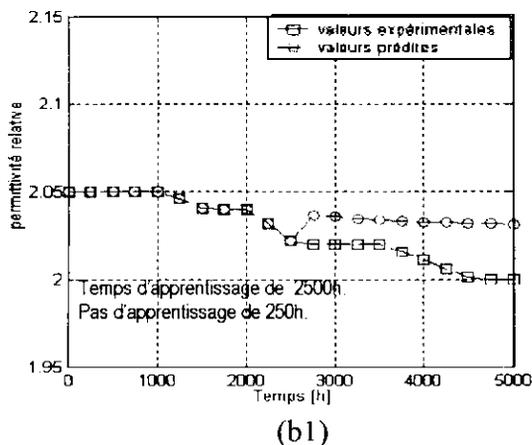
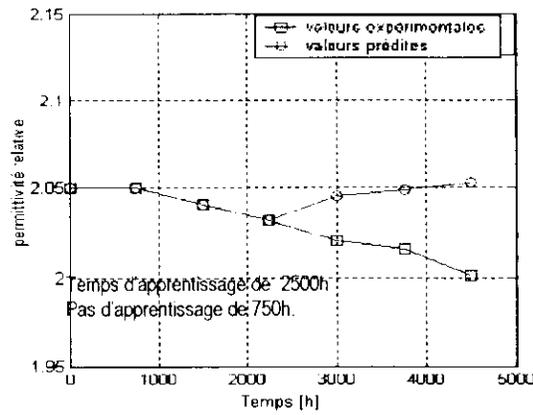


Fig.V.15. Facteur des pertes diélectriques en fonction du temps de vieillissement





(b3)

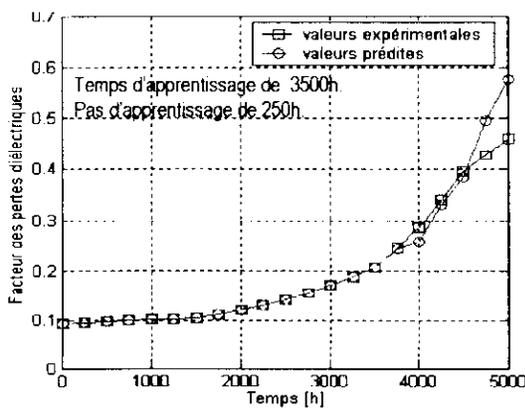
Fig.V.16. La permittivité relative en fonction de temps du vieillissement

	(a1)	(a2)	(a3)	(b1)	(b2)	(b3)
Le pas relatif	10	20	30	10	20	30
L'erreur relative moyenne	3.6715	4.5484	17.3209	1.1123	3.4714	1.8225
L'erreur relative minimale	0.0987	.0391	2.6154	0.7105	1.2434	1.2646
L'erreur relative maximale	10.0558	8.04621	35.8357	1.5940	6.7372	2.5964

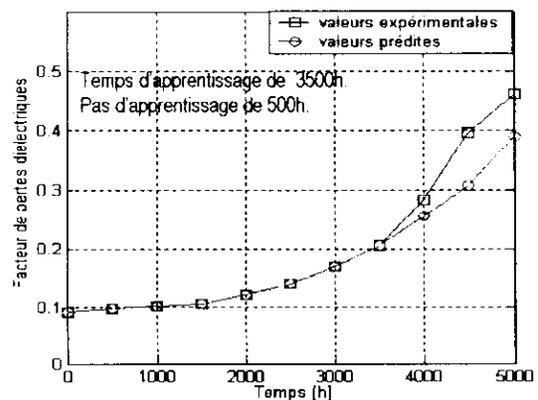
Tableau V.11. L'erreur relative concernant le facteur des pertes diélectriques.

- Temps d'apprentissage de 3500h (70%)

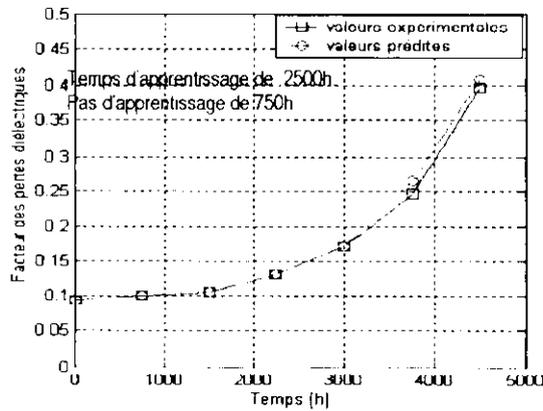
Avec un temps d'apprentissage pareil, presque tous les résultats sont bons (Fig.V.17 et Fig.V.18).



(a1)

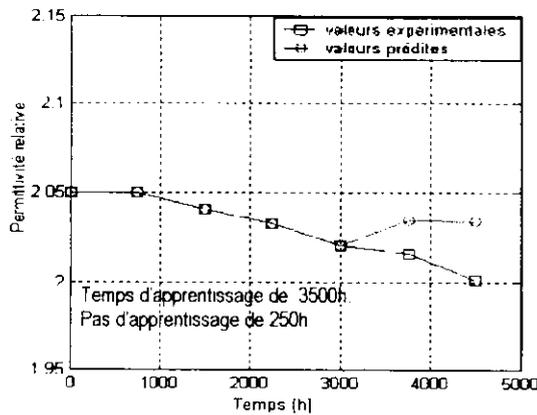


(a2)

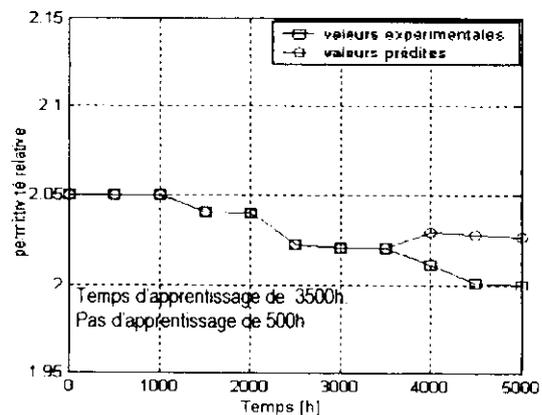


(a3)

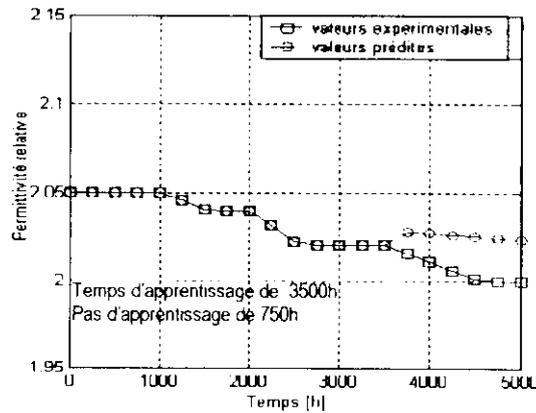
Fig.V.17.Facteur des pertes diélectriques en fonction du temps de vieillissement



(b1)



(b2)



(b3)

Fig.V.18. La permittivité relative en fonction du temps de vieillissement

	(a1)	(a2)	(a3)	(b1)	(b2)	(b3)
Le pas relatif	7.1429	14.2857	21.2857	7.1429	14.2857	21.2857
L'erreur relative moyenne	9.5674	15.7574	5.3669	1.0112	1.1754	1.2699
L'erreur relative minimale	1.1948	9.9509	2.7558	0.6090	0.8880	0.9005
L'erreur relative maximale	24.8486	22.3457	7.9780	1.2285	1.3206	1.6393

Tableau V.12. L'erreur relative et le pas relatif concernant les deux propriétés.

V.3.3. RESULTATS LIES AU RESEAU RBF2

Nous allons essayer d'utiliser les mêmes paramètres que dans le cas des deux réseaux RBF1 et RBF1, cela pour pouvoir comparer entre les trois programmes. Mais nous n'avons pas présentés les résultats concernant le pas 250h car le temps d'exécution est très lent.

- Temps d'apprentissage de 1500h (30%)

Seule la figure V.20.(b2) donne de résultats satisfaisants

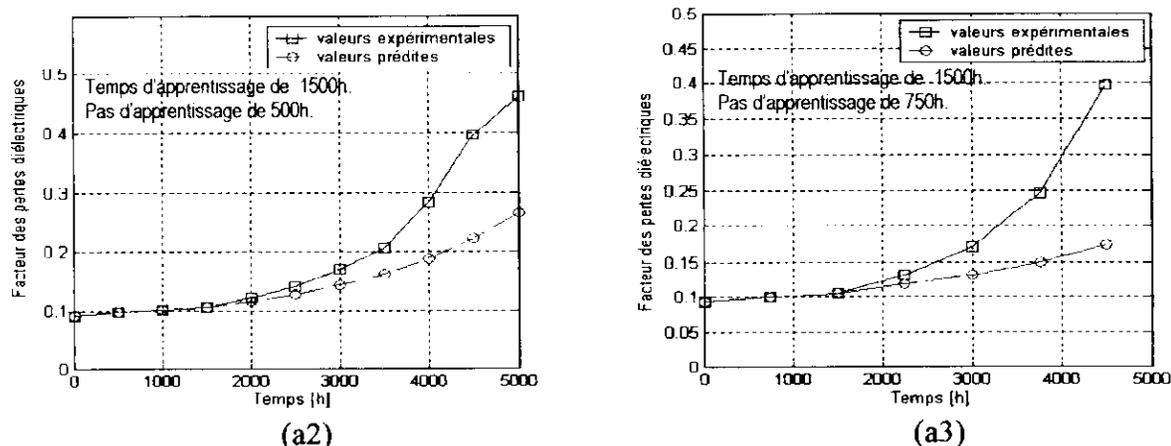


Fig.V.19.Facteur de pertes diélectriques en fonction du temps de vieillissement

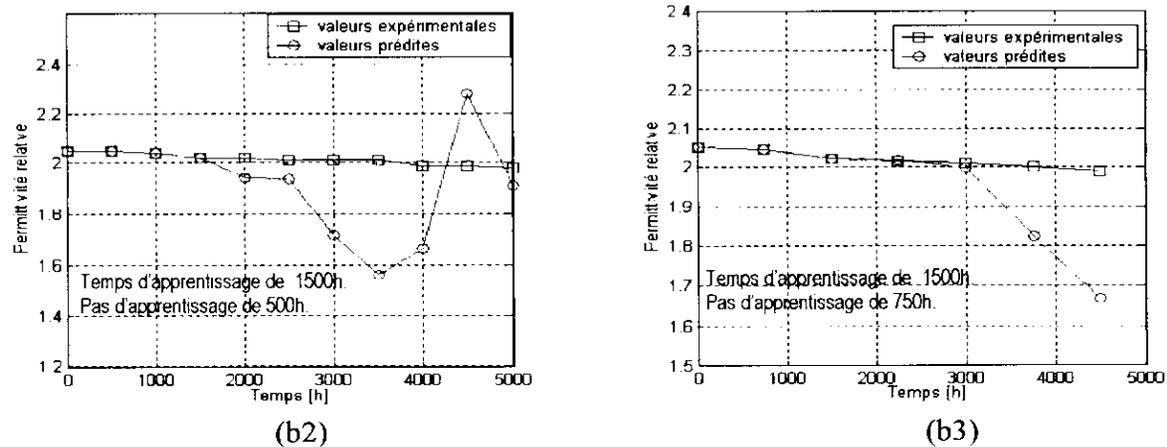


Fig. V.20. La permittivité relative on fonction du temps de vieillissement

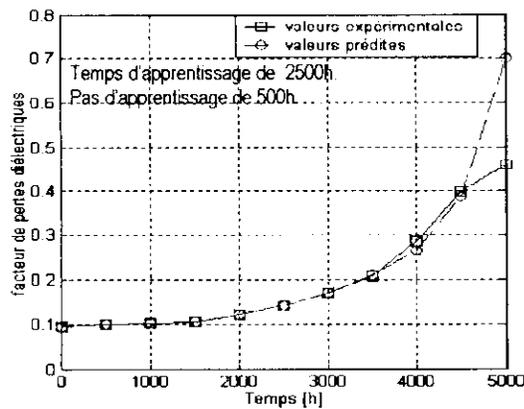
	(a2)	(a3)	(b2)	(b3)
Le pas relatif	33.3333	50	33.3333	50
L'erreur relative moyenne	24.4995	32.1452	11.2520	6.3862
L'erreur relative minimale	4.0422	10.6329	3.5901	0.1082

L'erreur relative maximale	43.8012	56.0997	22.2154	16.1869
----------------------------	---------	---------	---------	---------

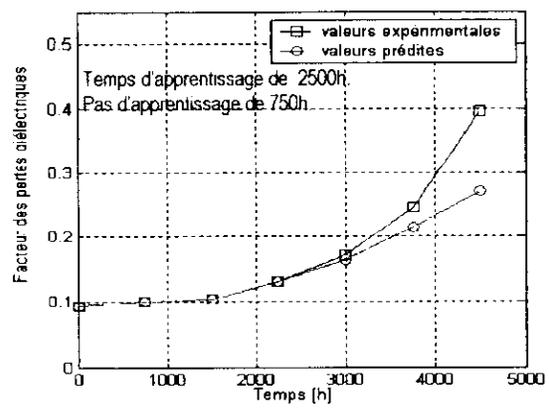
Tableau V.13. L'erreur relative et le pas relatif concernant les deux propriétés.

• Temps d'apprentissage de 2500h (50%)

On remarque que toutes les courbes divergent (Fig.V.21. et Fig.V.22.).

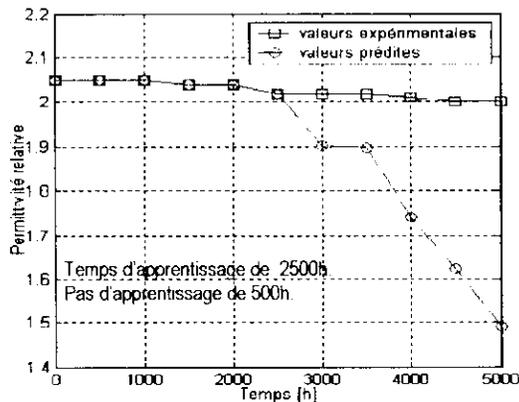


(a1)

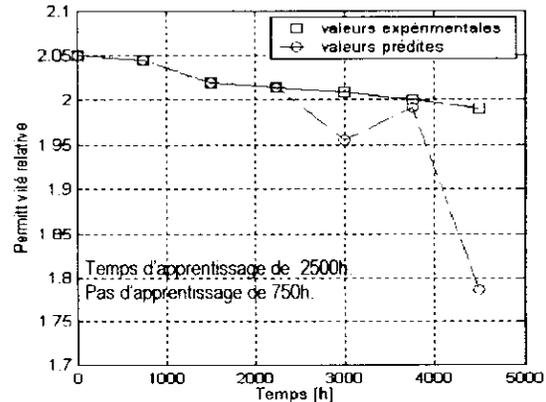


(a2)

Fig.V.21.Facteur de pertes diélectriques en fonction du temps de vieillissement



(b1)



(b2)

Fig.V.22. La permittivité relative on fonction du temps de vieillissement

	(a1)	(a2)	(b1)	(b2)
Le pas relatif	20	30	20	30
L'erreur relative moyenne	12.9476	16.0334	11.8000	16.0334
L'erreur relative minimale	0.9251	3.8466	3.7720	3.8466
L'erreur relative maximale	52.2643	31.4302	9.0288	31.4302

Tableau V.14. L'erreur relative et le pas relatif concernant les deux propriétés.

• Temps d'apprentissage de 3500h (70%)

Pour le facteur des pertes diélectriques (Fig.V.23), les résultats divergent. Mais pour la permittivité relative (Fig.V.24), nous obtenons de bons résultats.

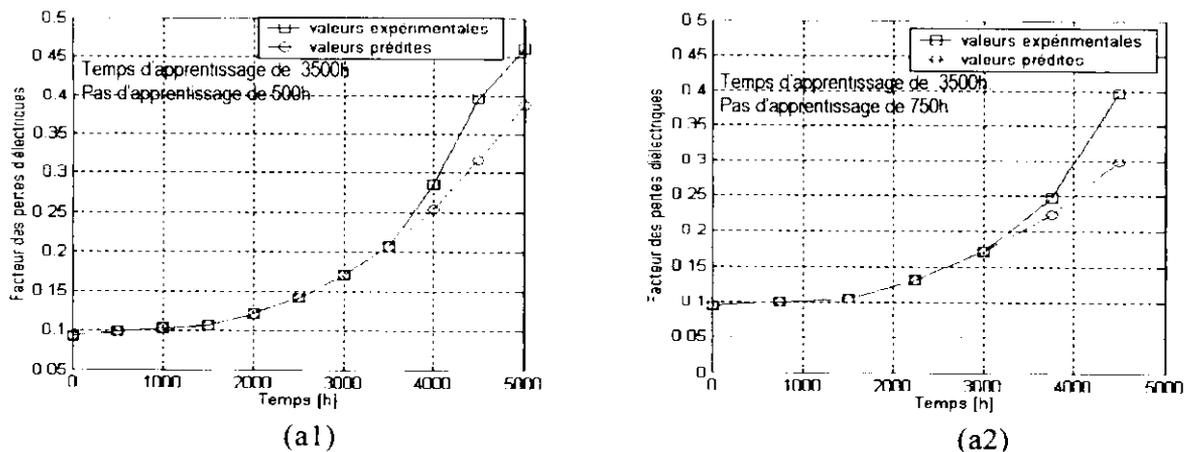


Fig.V.23. Facteur des pertes diélectriques en fonction du temps de vieillissement

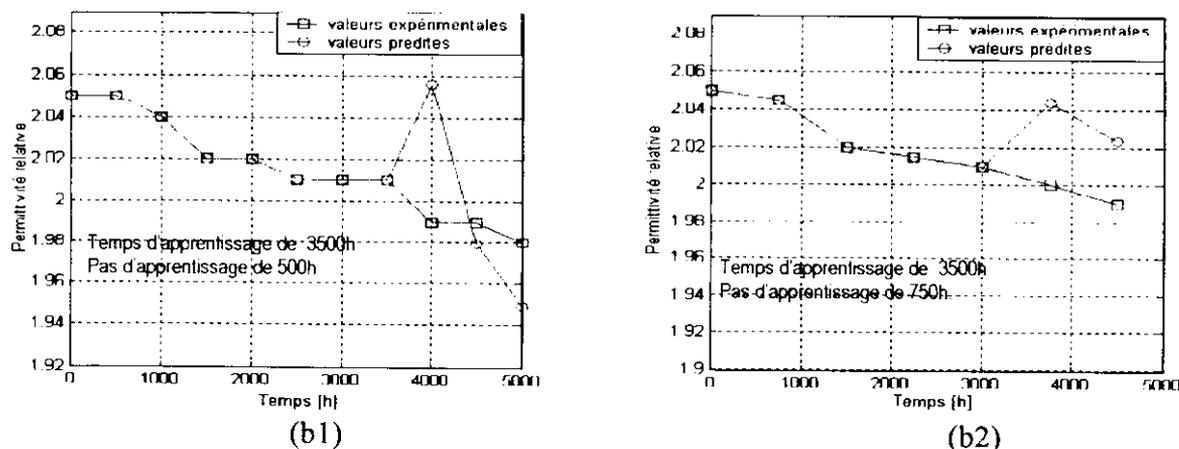


Fig.V.24. La permittivité relative en fonction du temps de vieillissement

	(a1)	(a2)	(b1)	(b2)
Le pas relatif	14.2857	21.2857	14.2857	21.2857
L'erreur relative moyenne	15.7574	16.6104	1.8106	1.9531
L'erreur relative minimale	9.9509	9.0810	0.5393	1.6997
L'erreur relative maximale	22.3457	24.8410	3.3297	2.2064

Tableau V.15. L'erreur relative et le pas relatif concernant les deux propriétés.

V.4. COMPARAISON ENTRE LES RESULTATS OBTENUS A PARTIR DED DIFFERENTS PROGRAMMES

D'après les résultats obtenus, nous pouvons dire que le programme RBFG1 est le meilleur par rapport aux deux autres. Il faut choisir un pas d'apprentissage relatif convenables pour l'adaptation des poids et pour éviter le sur- apprentissage du réseau.

Les réseaux RBF1 et RBFG2 donnent de bons résultats pour la permittivité relative, et pour le facteur des pertes diélectriques. Ces réseaux ne donnent pas de bons résultats dans le cas où nous diminués le temps et le pas d'apprentissage.

V.5. INTERPRETATIONS

En effet, en testant le réseau avec seulement les valeurs des propriétés, en ignorant le paramètre temps, nous obtenons une divergence des données. C'est pour cette raison, que nous avons rajouté le paramètre temps. Pendant l'exécution de ce programme.

A l'exécution de ce programme, le nouveau point à prédire dépend des valeurs prédites précédemment, si nous commettons une erreur au niveau de ces valeur, celle-ci de s'accumuler au cours de la prédiction.

Dans tous les programmes, les nouveaux points prédits dépendent de l'initialisation du vecteur de poids et du vecteur de centres au début de l'apprentissage, ce dernier donne l'information au réseau soit pour qu'il soit croissant ou bien décroissant. C'est-à-dire, selon que la disposition des centres est concentrée au début ou bien à la fin du vecteur de sortie désirée. Et c'est ici la grande différence entre le vecteur centre fixe pour RBF et le vecteur centre adaptatif pour RBFG. L'inconvénient le plus important que nous ayons relevé au cours de l'utilisation de ce dernier programme (RBFG) est la durée de recherche du minimum qui peut augmenter parfois et où des oscillations peuvent être générées.

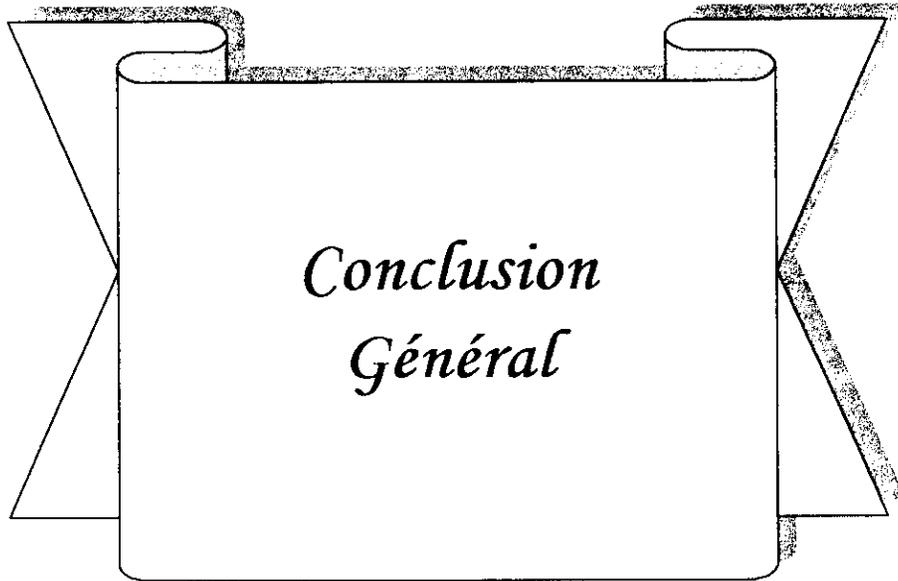
V.6. CONCLUSION

Les réseaux RBF normalisés se distinguent par leur caractéristique de meilleurs approximateurs, et sont à l'abri de plusieurs problèmes que connaissent les autres réseaux.

Après la discussion de tous les résultats obtenus par ces programmes, nous pouvons sélectionner le programme RBFG1 comme meilleur approximateur de propriété. Avec l'injection du temps à l'entrée du réseau et l'adaptation de la sortie de ce dernier aux valeurs expérimentales, on pourra bien avoir une approche de la caractéristique expérimentale en effectuant l'apprentissage avec un intervalle de temps relativement court.

Les résultats obtenus par les réseaux RBF1 et RBFG2 sont aussi acceptables, mais ils ont nécessite un temps d'apprentissage moyen.

Le choix du nombre de neurones dans la couche cachée et le taux de recouvrement peuvent influencer sur le temps d'exécution et les valeurs à prédire.



*Conclusion
Général*

CONCLUSION GENERALE

Le travail que nous avons effectué consiste à utiliser les réseaux de neurones artificiels dans la prédiction de l'évolution de certaines propriétés d'une huile isolante, soumise à un vieillissement thermique. Ce travail est l'amélioration d'un autre [2] sur le même thème, qui prolonge la durée de prédiction de 1500h à 5000h, et traite deux propriétés. C'est deux travaux sont basés sur la même famille de réseau (réseau RBF)

Nous avons élaboré sur trois programmes de calcul, le premier RBF1 utilise le temps comme entrée et la méthode ROM pour la phase d'apprentissage, et les deux autres RBF1 et RBF2 utilisent la méthode ACM pour la phase d'apprentissage, RBF1 est comme RBF1 utilise une seule entrée et RBF2 deux entrées (le temps et les valeurs de la propriété)

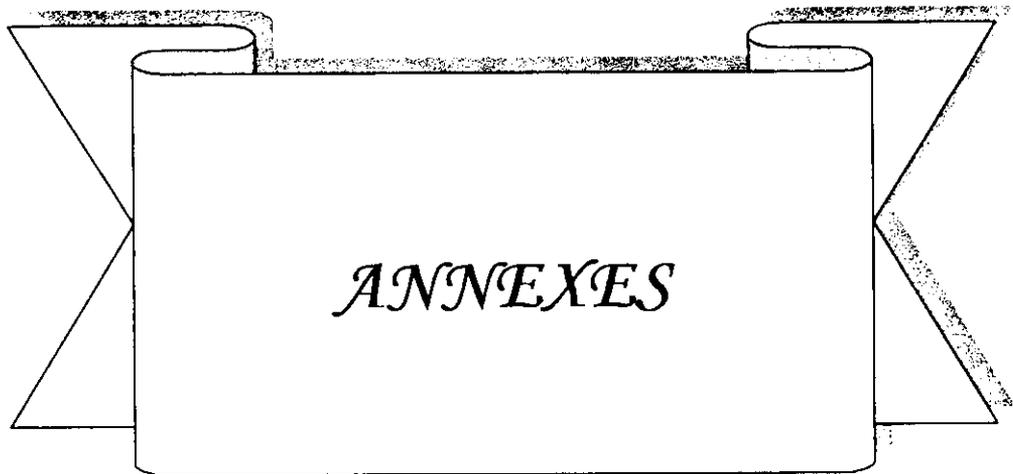
Durant les premiers tests, nous avons constaté que si on augmente le nombre de neurones N_c de la couche cachée, la durée d'exécution diminue mais la prédiction est mauvaise. Pour cela nous avons fixé ce nombre à 5 neurones pour tous les réseaux, bien que, pour certaines propriétés, d'autres nombres de neurones N_c conviennent mieux. Ceci pour que notre réseau soit fiable pour les deux propriétés

Les résultats obtenus nous ont permis d'aboutir aux conclusions suivantes :

- Les réseaux à base radiale normalisés se distinguent par leur caractéristique de meilleurs approximateurs des fonctions.
- La méthode de centrage adaptative ACM est la meilleure technique d'apprentissage. En effet les centres et les noyaux sont adaptatifs, ils conviennent le mieux pour les fonctions accidentées.
- Pour les caractéristiques concernant les permittivités relatives, tous les types de réseaux donnent de bons résultats même si nous changeons le pas et le temps d'apprentissage.
- Avec un pas d'apprentissage petit, la prédiction de réseau RBF2 est très lente quand on essaye de diminuer le temps d'apprentissage, ceci nous oblige d'entraîner le réseau sur des intervalles de temps élevés pour que le réseau puisse trouver une dépendance entre les valeurs qu'il a apprises.
- Les résultats obtenus par les réseaux RBF1 et RBF2 sont aussi acceptables, mais ils ont nécessité un temps d'apprentissage relativement moyen.

Enfin, nous terminons avec quelques idées représentant les perspectives auxquelles ce travail peut s'ouvrir.

- Dans notre étude, le nombre de neurones dans la couche cachée et le taux de recouvrement sont fixés, nous proposons d'optimiser ces paramètres avec une méthode convenable pour avoir des bons résultats.
- Nous mentionnons qu'à travers ce travail, nous avons constaté que les réseaux de neurones sont une famille d'outils très diversifiés. Chaque modèle a ces propres caractéristiques et peut être destiné à des opérations spécifiques. Nous pensons qu'il serait intéressant de construire différents réseaux qui peuvent être assemblés pour travailler ensemble dans une structure organisée.



Annexe 1 : RBF Généralisé

Soit $E = \sum (Y_d - Y_i)^2$ le critère à minimiser, telle que Y_i est donnée par l'équation suivante :

$$Y_i = \frac{\sum_j w_j \rho_j}{\sum_j \rho_j} \quad j = 1, 2, \dots, n \text{ et } i = 1, 2, \dots, n + f.$$

$$\frac{\partial E}{\partial w_j} = - \sum_i (Y_d - Y_i) \cdot \frac{\partial Y_i}{\partial w_j}$$

- $\frac{\partial Y_i}{\partial w_j} = \frac{\rho_j}{\sum_j \rho_j}$

$$\Rightarrow \frac{\partial E}{\partial w_j} = - \sum_i (Y_d - Y_i) \cdot \frac{\rho_j}{\sum_j \rho_j}$$

$$\frac{\partial E}{\partial C_{kj}} = - \sum_i (Y_d - Y_i) \cdot \frac{\partial Y_i}{\partial C_{kj}}$$

- $\frac{\partial Y_i}{\partial C_{kj}} = \frac{\partial Y_i}{\partial \rho_j} \frac{\partial \rho_j}{\partial C_{kj}}$

$$\frac{\partial Y_i}{\partial \rho_j} = \frac{w_j \cdot \sum_j \rho_j - \sum_j w_j \rho_j}{(\sum_j \rho_j)^2}$$

Avec

$$\frac{\partial \rho_j}{\partial C_{kj}} = \frac{1}{\sigma_{kj}^2} \cdot (X_i - C_{kj}) \cdot \rho_j$$

$$\frac{\partial E}{\partial \sigma_{kj}} = - \sum_i (Y_d - Y_i) \cdot \frac{\partial Y_i}{\partial \sigma_{kj}} \frac{\partial \rho_j}{\partial C_{kj}}$$

- $\frac{\partial Y_i}{\partial \rho_j} = \frac{\partial Y_i}{\partial \rho_j} \frac{\partial \rho_j}{\partial \sigma_{kj}}$

avec

$$\frac{\partial \rho_j}{\partial \sigma_{kj}} = \frac{1}{\sigma_{kj}^3} \cdot (X_i - C_{kj})^2 \cdot \rho_j$$

$$\Delta w_j = -f w_j \cdot \frac{\partial E}{\partial w_j}$$

$$\Delta C_{kj} = -f c \frac{\partial E}{\partial C_{kj}}$$

$$\Delta \sigma_{kj} = -f \sigma \frac{\partial E}{\partial \sigma_{kj}}$$

Annexe 2 : PROGRAMMES

"Header file" fichier d'entête inclus les fonctions suivant :
 Normalisation, random, Donnee, prediction, resultat.
 Il est comun pour RBF1 et RBF1.

GestionE_S1.h

```
#include "math.h"
#include "iostream.h"
#include "stdio.h"
#include "stdlib.h"
FILE *Res, *Gra, *NOM, *Don;

char C, Nom[14];
int n, Dim, N_c, f, i, j, k, p;
float Deltat, h, TG, T_App, Pas;
float Yd[200], X[200]; //Declarations.
double Temps[200], ND, A, B;
double Cent[200], DeltaS, Deltay, Er, Er_max=.0, Er_min=100., Er_moy=0, maxY=.0;
double ry[200], Ro[200][200], Sigma[200];
double Taux ,W[200], Y[200], Yd[200], X[200], Err_d=0.03, e;
double dRo_sigmy, dRo_ceny, Ceny[200], Sigmay[200];

//fonctions utilisés.

double random(void)
{
    return(rand()/(RAND_MAX+1.0));
}

void Normlisation(void)
{
    for (i=0; i<n+f+2; i++)
        {if (Yd[i]>maxY)
            maxY=Yd[i];}

    for (i=0; i<n+f+2; i++)
        Yd[i]/=maxY;
}

void Donnee(void)
{
    printf("\nVoulez vous utiliser les valeurs enregistres auparavant");
    printf("\n Taper (O)ui ou (N)on\n");
    scanf("%c", &C);
    switch (C)
    {
    case 'O':
    case 'o':
        {
        cout<<"\nDonner le Nom de ce fichier Nom: ";
        cin>>Nom;
        NOM=fopen(Nom, "r");
        fscanf(NOM, "%d\n", &Dim);
        }
    }
}
```

```

for(i=0;i<Dim;i++)
    fscanf(NOM,"%f\n",&_Yd[i]);
fclose(NOM);
break;
}
case 'N':
case 'n':
    {Don=fopen("Donnee.RBF","w+");
    printf("\nDonner le nombre de points que vous avez\n\n : ");
    cin>>Dim;
    fprintf(Don,"%d\n",Dim);
    cout<<"\nDonner les valeurs de Ydi\n";
    for(i=0;i<Dim;i++)
        {cout<<"\nYd["<<i+1<<"] est ";
        cin>>_Yd[i];
        fprintf(Don,"%f\n",_Yd[i]);}
    fclose(Don);
    break;
}
default :
    {
    exit(-1);
    break;
}
}
do
{
cout<<"\nQuel est le pas d'abscisse \n\t Pas= ";
cin>>Pas;
cout<<"\nDonner l'abscisse initiale\n\t X[1]= ";
cin>>_X[0];
cout<<"\nChoisir l'Abscisse_Globale\nA_G= ";
cin>>TG;
cout<<"\nFixer le Temps d'Apprentissage_Reltif par rapport a A_G\nA_R(%)=
";
cin>>T_App;T_App*=(TG/100);
cout<<"\nFixer le pas d'Apprentissage(Mx pas,M=1,2,...)\n Deltat= ";
cin>>Deltat;
cout<<"\nChoisir le taux de recouvrement\nTaux(%)= ";
cin>>Taux;Taux/=100;
cout<<"\nchoisir le nombre de neurones dans la couche cachee\n\tN_c= ";
cin>>N_c;
cout<<"\n Confermer les donnees ...Taper (O)ui\n\t";
cin>>C;
}while(C!='o');

for(i=1;i<Dim;i++)
    _X[i]=_X[0]+Pas*i;

for (h=_X[0];h<=TG;k++,h=h+Deltat)
    Temps[k]=h;

//Selections des donnees.
p=Deltat/Pas;Yd[0]=_Yd[0];
for (h=_X[0];h<=T_App;n++,h=h+Deltat)
    Yd[n+1]=_Yd[n*p];
}
void prediction(void)
{

```

```

X[n+f]=0;ry[n+f]=0; // La prédiction de Yn+f
for (j=0;j<N_c;j++)
{
    A=- (Temps[n+f]-Cent[j]) * (Temps[n+f]-
Cent[j]) / (2*Sigma[j]*Sigma[j]);
    B=- (Yd[n+f]-Ceny[j]) * (Yd[n+f]-Ceny[j]) / (2*Sigmay[j]*Sigmay[j]);
    Ro[n+f][j]=exp(A+B);
    ry[n+f]+=Ro[n+f][j];
    X[n+f]+=W[j]*Ro[n+f][j];
}
Yd[n+f+1]=Y[n+f]=X[n+f]/ry[n+f];
printf("\nLe point predict de Y[%d]est:.... %f\n",n+f+1,Yd[n+f+1]*maxY);
}

void Resultat(void)
{
fprintf(Gra,"%d\n",n+f);
for(i=0;i<n+f;i++)
fprintf(Gra,"% .4f\n",Temps[i]);
for(i=0;i<n+f;i++)
fprintf(Gra,"% .4f\n",_Yd[i*p]);
fprintf(Gra,"%d\n", (n+f));
fprintf(Res,"\n\t##### RESULTATS #####\n");
fprintf(Res,"\nLe Temps_Globale\n\tT_G= %.0f",TG);
fprintf(Res,"\nLe Temps d'Apprentissage\n\tT_App= %.1f",T_App);
fprintf(Res,"\nLe Pas d'Apprentissage\n\tPas= %.1f",Deltat);
fprintf(Res,"\nLe taux de recouvrement\n\tTaux= %.3f",Taux);
fprintf(Res,"\nLe nombre de neurones dans la couche cachee\n\tN_c=
%d",N_c);
fprintf(Res,"\nL'erreur de la sortie Ydi\n\tErreur= %.3f\n",Err_d);
for (i=0;i<n+f;i++)
{fprintf(Res,"\n\t T[%d]= %.1f ..... Y[%d]=
%.3f\n",i+1,Temps[i],i+1,Yd[i+1]);
fprintf(Gra,"% .4f\n",Temps[i]);}
for (i=0;i<n+f;i++)
fprintf(Gra,"% .4f\n",Yd[i+1]);

for (i=n;i<n+f;i++)
{
Er=fabs((_Yd[i*p]-Yd[i+1])*100/_Yd[i*p]);
Er_moy+=Er/f;
if(Er>Er_max)
Er_max=Er;
if(Er<Er_min)
Er_min=Er;
}
fprintf(Res,"\n Erreur relative moyenne : \n\t %e",Er_moy);
fprintf(Res,"\n Erreur relative minimal : \n\t %e",Er_min);
fprintf(Res,"\n Erreur relative maximal : \n\t %e",Er_max);
printf("\a\a");

fclose(Res);
fclose(Gra);
}

```

LE PROGRAMME RBF1

RBF1.CPP

```

#include "GestionE_S1.h"
double ksi[40],v, moy[40];
double ek=0,e_Plus, e_Moins;
double Y_Plus[40],Y_Moins[40];
double W_Plus[40],W_Moins[40];
double X_Plus[40],X_Moins[40];

void main()
{
Res=fopen("Resultat.RBF","w+");
Gra=fopen("graph.RBF","w+");
Donnee();
for (f=0;h<=TG;h+=Deltat,f++)
    {/$$$$$Le debut de la boucle de pr diction.$$$$$
    Normlisation();

    DeltaS=(Temps[n+f-1]-Temps[f])/N_c;

    v=0.3;
for (j=0;j<N_c;j++)
    {Cent[j]=Deltat*(j+f);
    Sigma[j]=sqrt(-(DeltaS)*(DeltaS)/(8*log(Taux)));
    W[j]=random(); //Initialisations.
    moy[j]=.0;
    }
do
    {/** Le debut de la RoM "Apprentissage"**//
    for (j=0;j<N_c;j++)
    {ksi[j]=v*random()+moy[j];
    W_Plus[j]=W[j]+ksi[j];
    W_Moins[j]=W[j]-ksi[j];}

for (i=f;i<(n+f);i++)
    {
    ry[i]=0;
    for (j=0;j<N_c;j++)
    {Ro[i][j]=exp(-(Temps[i]-Cent[j])*(Temps[i]-
Cent[j]))/(2*Sigma[j]*Sigma[j]));
    ry[i]+=Ro[i][j];}
    }
for (i=f;i<(n+f);i++)
    {
    X[i]=.0;X_Plus[i]=.0; X_Moins[i]=.0;
for(j=0;j<N_c;j++)
    {X[i]+=W[j]*Ro[i][j];
    X_Plus[i]+=W_Plus[j]*Ro[i][j];
    X_Moins[i]+=W_Moins[j]*Ro[i][j];}

    Y[i]=X[i]/ry[i];
    Y_Plus[i]=X_Plus[i]/ry[i];
    Y_Moins[i]=X_Moins[i]/ry[i];
    }
    }
}

```

```

    }
    e=.0;e_Plus=.0;e_Moins=.0;//Initailisations
    for (i=f;i<(n+f);i++)
    {
        e+=(Y[i]-Yd[i])*(Y[i]-Yd[i])/n;
        e_Plus+=(Y_Plus[i]-Yd[i])*(Y_Plus[i]-Yd[i])/n;
        e_Moins+=(Y_Moins[i]-Yd[i])*(Y_Moins[i]-Yd[i])/n;
    }

    e=sqrt(e);e_Plus=sqrt(e_Plus);e_Moins=sqrt(e_Moins);
    if (e_Plus<e)
    {
        for (j=0;j<N_c;j++)
        {W[j]+=ksi[j];
        moy[j]-=.2*moy[j]+.4*ksi[j];}
    }
    else if((e_Plus>=e)&&(e_Moins<e))
    {
        for (j=0;j<N_c;j++)
        W[j]-=ksi[j];moy[j]-=.4*ksi[j];
    }
    else
    {
        for (j=0;j<N_c;j++)
        {moy[j]*=.5;}
    }
    printf("\tErreur- %e \r",e);
    if ((ek-e) ==0)
    {if (v>=.0)
        v-=0.1;
        else
        v=random();}
    ek=e;
    }while (e>=Err_d);//****La fin de la boucle ROM.

    // La prédiction de Y(n+f). //
    prediction();
    for (i=0;i<n+f+1;i++)
        Yd[i]*=maxY;

} //%%%%%%/ Fin de boucle de prédiction.

    // **Résultats** //
    Resultat();

}

```

LE PROGRAMME RBFG1

RBFG1.CPP

```

#include "GestionE_S1.h"

double dE_w[50],dE_cent[50],dE_sigm[50];
double dRo_sigm,dRo_cent,dY_Ro;

void main()
{
Res=fopen("Resulta.RBF","w");// contient les résultats de l'execution.
Gra=fopen("Graph.RBF","w");// contient les valeurs a tracer.
Donnee();

for (f=0;h<=TG;h+=Deltat,f++)
{ //$$$$$Le debut de la boucle de prédiction.$$$$$
  Normlisation();
  DeltaS=(Temps[n+f-1]-Temps[0])/N_c;
  for (j=0;j<N_c;j++)
  {
    W[j]=random();
    Cent[j]=DeltaS*(j+f+1);
    Sigma[j]=sqrt(-(DeltaS)*(DeltaS)/(8*log(Taux)));
  }
  for(int N_it=0;;N_it++)
  { //*****Le debut de la boucle d'apprentissage****//
    for (i=f;i<(n+f);i++)
    {
      ry[i]=.0;X[i]=.0;//ry=Somme(Sij).
      for (j=0;j<N_c;j++)
      {Ro[i][j]=exp(-(Temps[i]-Cent[j])*(Temps[i]-
Cent[j]))/(2*Sigma[j]*Sigma[j]));
      ry[i]+=Ro[i][j];
      X[i]+=-W[j]*Ro[i][j];};//X=somme(Wj.Sij)
    }
    e=.0; //Initailisations pour chaque itérations.
    for (i=f;i<(n+f);i++)
    {
      Y[i]=X[i]/ry[i];
      e+=((Y[i]-Yd[i])*(Y[i]-Yd[i])/n);
    }
    e=sqrt(e);
    printf("\tErreur=%e\r",e);
    if (e<=Err_d)
    {
      break;
    }
    //*****La fin de la boucle d'apprentissage.****//

//Calcule préliminaire pour l'adaptation des poids.
  for (j=0;j<N_c;j++)
    { dE_w[j]=0;dE_cent[j]=.0;dE_sigm[j]=.0;//Initialisations

```

```
for (i=f;i<(n+f);i++)
    {dRo_cent=(Temps[i]-
Cent[j])*Ro[i][j]/(Sigma[j]*Sigma[j]);
dRo_sigm=(Temps[i]-Cent[j])*(Temps[i]-
Cent[j])*Ro[i][j]/(pow(Sigma[j],3));
dY_Ro=(W[j]*ry[i]-X[i])/(ry[i]*ry[i]);
dE_cent[j]-=((Yd[i]-Y[i])*dY_Ro*dRo_cent);
dE_sigm[j]-=((Yd[i]-Y[i])*dY_Ro*dRo_sigm);
dE_w[j]-=((Yd[i]-Y[i])*Ro[i][j]/ry[i]);}
}
for (j=0;j<N_c;j++)
    {//Adaptations des poids.
W[j]-=(1.05*dE_w[j]);
Cent[j]-=(.85*dE_cent[j]);
Sigma[j]-=(.85*dE_sigm[j]);
}
} //**** end de N_it ****.
prediction();
for (i=0;i<n+f+1;i++)
    Yd[i]*=maxY;

} //$$$$$$$Fin de boucle de prédiction.$$$$$$$

Resultat();
}
```

LE PROGRAMME RBFG2

GestionE_S1.h

```

#include "math.h"
#include "iostream.h"
#include "stdio.h"
#include "stdlib.h"
FILE *Res, *Gra, *NOM, *Don;

char C, Nom[14];
int n, Dim, N_c, f, i, j, k, p;
float Deltat, h, TG, T_App, Pas;
float _Yd[200], _X[200]; //Declarations.
double Temps[200], ND, A, B, fc;
double Cent[200], DeltaS, Deltay, Er, Er_max=.0, Er_min=100., Er_moy=0, maxY=.0;
double ry[200], Ro[200][200], Sigma[200];
double Taux, W[200], Y[200], Yd[200], X[200], Err_d=0.03, e;
double dRo_sigmaty, dRo_ceny, Ceny[200], Sigmay[200];

//fonctions utiliseés.

double random(void)
{
    return(rand()/(RAND_MAX+1.0));
}

void Normlisation(void)
{
    for (i=0; i<n+f+2; i++)
    {
        if (Yd[i]>maxY)
            maxY=Yd[i];
    }

    for (i=0; i<n+f+2; i++)
        Yd[i]/=maxY;
}

void Donnee(void)
{
    printf("\nVoulez vous utiliser les valeurs enregistres auparavant");
    printf("\n Taper (O)ui ou (N)on\n");
    scanf("%c", &C);
    switch (C)
    {
    case 'O':
    case 'o':
    {
        cout<<"\nDonner le Nom de ce fichier Nom: ";
        cin>>Nom;
        NOM=fopen(Nom, "r");
        fscanf(NOM, "%d\n", &Dim);
        for(i=0; i<Dim; i++)
            fscanf(NOM, "%f\n", &_Yd[i]);
        fclose(NOM);
    }
    }
}

```

```

        break;
    }
case 'N':
case 'n':
    {Don=fopen("Donnee.RBF","w+");
    printf("\nDonner le nombre de points que vous avez\nn : ");
    cin>>Dim;
    fprintf(Don,"%d\n",Dim);
    cout<<"\nDonner les valeurs de Ydi\n";
    for(i=0;i<Dim;i++)
        {cout<<"\nYd["<<i+1<<"] est ";
        cin>>_Yd[i];
        fprintf(Don,"%f\n",_Yd[i]);}
    fclose(Don);
    break;
    }
default :
    {
    exit(-1);
    break;
    }
}
do
{
cout<<"\nQuel est le pas d'abscisse \n\t Pas= ";
cin>>Pas;
cout<<"\nDonner l'abscisse initiale\n\t X[1]= ";
cin>>_X[0];
cout<<"\nChoisir l'Abscisse_Globale\nA_G= ";
cin>>TG;
cout<<"\nFixer le Temps d'Apprentissage_Reltif par rapport a A_G\nA_R(%)=
";
cin>>T_App;T_App*=(TG/100);
cout<<"\nFixer le pas d'Apprentissage(Mx pas,M=1,2,..)\n Deltat= ";
cin>>Deltat;
cout<<"\nChoisir le taux de recouvrement\nTaux(%)= ";
cin>>Taux;Taux/=100;
cout<<"\nchoisir le nombre de neurones dans la couche cachee\n\tN_c= ";
cin>>N_c;
cout<<"\n Confermer les donnees ...Taper (0)ui\n\t";
cin>>C;
}while(C!='o');

for(i=1;i<Dim;i++)
    _X[i]=_X[0]+Pas*i;

for (h=_X[0];h<=TG;k++,h=h+Deltat)
    Temps[k]=h;

p=Deltat/Pas;Yd[0]=_Yd[0]; //Selections des donnees.

for (h=_X[0];h<=T_App;n++,h=h+Deltat)
    Yd[n+1]=_Yd[n*p];

}
void prediction(void)
{
    X[n+f]=0;ry[n+f]=0; // La prédition de Yn+f
    for (j=0;j<N_c;j++)
        {

```

```

        A=- (Temps[n+f]-Cent[j]) * (Temps[n+f]-
Cent[j]) / (2*Sigma[j]*Sigma[j]);
        B=- (Yd[n+f]-Ceny[j]) * (Yd[n+f]-Ceny[j]) / (2*Sigmay[j]*Sigmay[j]);
        Ro[n+f][j]=exp(A+B);
        ry[n+f]+=Ro[n+f][j];
        X[n+f]+=W[j]*Ro[n+f][j];
    }
    Yd[n+f+1]=Y[n+f]-X[n+f]/ry[n+f];
printf("\nLe point predit de Y[%d]est:.... %f\n",n+f+1,Yd[n+f+1]*maxY);
}

void Resultat(void)
{
fprintf(Gra,"%d\n",n+f);
for(i=0;i<n+f;i++)
fprintf(Gra,"% .4f\n",Temps[i]);
for(i=0;i<n+f;i++)
fprintf(Gra,"% .4f\n",_Yd[i*p]);
fprintf(Gra,"%d\n", (n+f));
fprintf(Res,"\n\t##### RESULTATS #####\n");
fprintf(Res,"\nLe Temps_Globale\n\tT_G= %.0f",TG);
fprintf(Res,"\nLe Temps_d'Apprentissage\n\tT_App= %.1f",T_App);
fprintf(Res,"\nLe Pas_d'Apprentissage\n\tPas= %.1f",Deltat);
fprintf(Res,"\nLe taux de recouvrement\n\tTaux= %.3f",Taux);
fprintf(Res,"\nLe nombre de neurones dans la couche cachee\n\tN_c=
%d",N_c);
fprintf(Res,"\nL'erreur de la sortie Ydi\n\tErreur= %.3f\n",Err_d);
for (i=0;i<n+f;i++)
    {fprintf(Res,"\n\t T[%d]= %.1f ..... Y[%d]=
%.3f\n",i+1,Temps[i],i+1,Yd[i+1]);
    fprintf(Gra,"% .4f\n",Temps[i]);}
for (i=0;i<n+f;i++)
    fprintf(Gra,"% .4f\n",Yd[i+1]);

for (i=n;i<n+f;i++)
    {
    Er=fabs((_Yd[i*p]-Yd[i+1])*100/_Yd[i*p]);
    Er_moy+=Er/f;
    if(Er>Er_max)
        Er_max=Er;
    if(Er<Er_min)
        Er_min=Er;
    }
fprintf(Res,"\n Erreur relative moyenne : \n\t %e",Er_moy);
fprintf(Res,"\n Erreur relative minimal : \n\t %e",Er_min);
fprintf(Res,"\n Erreur relative maximal : \n\t %e",Er_max);
printf("\a\a");

fclose(Res);
fclose(Gra);
}

```

RBF2.CPP

```

#include "GestionE_S2.h"
double dE_w[200],dE_cent[200],dE_sigm[200];
double dE_ceny[200],dE_sigmy[200];
double dRo_sigm,dRo_cent,dY_Ro;
void main()
{
Res=fopen("Resultat.RBF","w+");// contient les résultats de l'execution.
Gra=fopen("Graph.RBF","w+"); // contient les valeurs a tracer.
Donnee();

for (f=0;h<=TG;h+=Deltat,f++)
{ //$$$$$Le début de la boucle de prédiction$$$$$
  Normlisation();
  DeltaS=(Temps[n+f-1]-Temps[0])/N_c;
  Deltay=(Yd[n+f-1]-Yd[0])*maxY/N_c;
  for (j=0;j<N_c;j++)
  {
  W[j]=random();
  Cent[j]=Temps[0]+DeltaS*(j+f);
  Ceny[j]=Yd[0]*maxY+Deltay*(j+f);
  Sigma[j]=sqrt(-(DeltaS)*(DeltaS)/(8*log(Taux)));
  Sigmay[j]=sqrt(-(Deltay)*(Deltay)/(8*log(Taux)));
  }
  for(int N_it=0;;N_it++)
  { //*****Le début de la boucle d'apprentissage*****/
    for (i=f;i<(n+f);i++)
    {
      ry[i]=.0;X[i]=.0;//ry=Somme ($ij).
      for (j=0;j<N_c;j++)
        {A=-(Temps[i]-Cent[j])*(Temps[i]-
Cent[j])/(2*Sigma[j]*Sigma[j]);
        B=-(Yd[i]*maxY-Ceny[j])*(Yd[i]*maxY-
Ceny[j])/(2*Sigmay[j]*Sigmay[j]);
        Ro[i][j]=exp(A+B);
        ry[i]+=Ro[i][j];
        X[i]+=W[j]*Ro[i][j];} //X=somme(Wj.$ij)
    }
    e=.0; //Initialisations pour chaque itération.
    for (i=f;i<(n+f);i++)
    {
      Y[i]=X[i]/ry[i];
      e+=((Y[i]-Yd[i+1])*(Y[i]-Yd[i+1])/n);
    }
  }
  e=sqrt(e);
  printf("\tErreur= %e\r",e);

  if (e<=Err_d)
  {
    break;
  } //*****La fin de la boucle d'apprentissage*****/

  //Calcule préliminaire pour l'adaptation des poids.
  for (j=0;j<N_c;j++)

```

```

    { dE_w[j]=0;dE_cent[j]=.0;dE_sigm[j]=.0;
    dE_ceny[j]=.0;dE_sigmy[j]=.0;//Initialisations

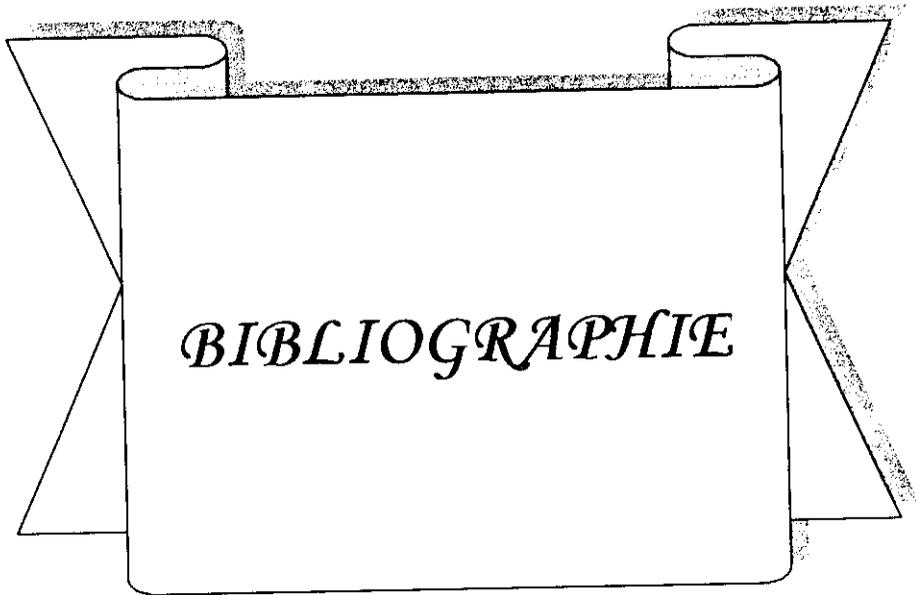
    for (i=f;i<(n+f);i++)
        {dRo_cent=(Temps[i]-
    Cent[j])*Ro[i][j]/(Sigma[j]*Sigma[j]);
    dRo_ceny=(Yd[i]-Ceny[j])*Ro[i][j]/(Sigmay[j]*Sigmay[j]);
    dRo_sigm=(Temps[i]-Cent[j])*(Temps[i]-
    Cent[j])*Ro[i][j]/(pow(Sigma[j],3));
    dRo_sigmy=(Yd[i]-Ceny[j])*(Yd[i]-
    Ceny[j])*Ro[i][j]/(pow(Sigmay[j],3));
    dY_Ro=(W[j]*ry[i]-X[i])/(ry[i]*ry[i]);
    dE_cent[j]-=((Yd[i+1]-Y[i])*dY_Ro*dRo_cent);
    dE_sigm[j]-=((Yd[i+1]-Y[i])*dY_Ro*dRo_sigm);
    dE_ceny[j]-=((Yd[i+1]-Y[i])*dY_Ro*dRo_ceny);
    dE_sigmy[j]-=((Yd[i+1]-Y[i])*dY_Ro*dRo_sigmy);
    dE_w[j]-=((Yd[i+1]-Y[i])*Ro[i][j]/ry[i]);}
    }
    for (j=0;j<N_c;j++)
        { //Adaptations des poids.
        W[j]-=(.95*dE_w[j]);
        Cent[j]-=(.95*dE_cent[j]);
        Ceny[j]-=(.95*dE_ceny[j]);
        Sigma[j]-=(.95*dE_sigm[j]);
        Sigmay[j]-=(.95*dE_sigmy[j]);
        }
    } //**** end de N_it ****.
prediction();

for (i=0;i<n+f+2;i++)
    Yd[i]*=maxY;

} //$$$$$$$ Fin de boucle de prédiction. $$$$$$
Resultat();

}

```



BIBLIOGRAPHIE

BIBLIOGRAPHIE

- [01] A.Boubakeur, N.Mokhnache, A.Khelifi, A.L.Nemmour, « Neural Networks Application In Th Field Of High Voltage Cable Insulation Thermal Ageing », 3rd International conferece QRM, Oxford 2000.
- [02] Bedja. A/Rahim et O.N.M^{ed} Elboukhary. « *Utilisation Des Réseaux De Neurones Artificiels Dans Les Vieillissement Thermique De L'huile De Transformateur* », thèse de PFE, ENP, Juin 1999.
- [03] H.Ailam. « *Utilisation des Réseaux de Neurones dans le Domaine de la Pollution des Isolateurs* », thèse de Magister, ENP, décembre 2001.
- [04] Rumelhart D.E, G.E.Hinton and R.J.Wiliams, « *Learning Internal Representation by Error Propagation* », 1986
- [05] Liu, « *Neural Networks For Identification, Prediction And Control* », Springer, 1994.
- [06] Kosko.B, « *Neural Networks and Fuzzy Systems. a Dynamical Machine Approach to Machines Systems* », Printece Hall International, 1992
- [07] . Freeman,J.D. and M.Skapura. « *Neural Networks, Algorithms, Applications and Programming Technique* », Adison Wesley Company, 1992
- [08] M.Yazid M'hamed Yeddou, « *Etude de Synthèse sur les Réseaux de Neurones et leur Applications* », thèse de Magister, ENP, juin 1998
- [09] J.Hérault, C.Jutten,«Réseaux Neuronaux et Traitement du Signale», Traité des Nouvelles Technologies, HERMES, 1994
- [10] D.K.Xu, C.Z.Fu,Y.M.Li, « *Application of Artificial Neural Network to the Detection of the Transformer Winding Deformation* », International Symposium on High-Voltage Engineering, London, 1999.
- [11] I.N.da Silva, A.N. de Souza, « *Analysis of High-Voltage Substations Design Using Artificial Neural Networks* », International Symposium on High-Voltage Engineering, London, 1999.
- [12] H.Okubo, T.Otsuka, K.Kato, N.Hayakawa, M.Hikita, « *Electric Field Optimization of High Voltage Electrode Based on Neural Network* », IEEE Transactions on Power Systems. Vol.12, No.4, November 1997.
- [13] P.K.Mukherjee, C.Trinitis, H.Steinbigler, « *Optimisation of HV Electrode Systems by Neural Networks Using a New Learning Method* », IEEE Transactions on Dielectrics and Electrical Insulation, Vol.3, No.6, December 1996.
- [14] P.S.Ghosh, S.Chakravorti, N.Chattergee, « *ANN With RPROP Learning Algorithm Applied to Modelling of Flashover Voltage Characteristics of Polluted Insulators* », Ninth International Symposium On High Voltage Engineering, Gratz, 1995.

- [15] K.Engel, D.Peier, « *Influence of PD Fault Development on Fault Type Recognition Using an Artificial Neural Network* », Ninth International Symposium On High Voltage Engineering, Gratz, 1995.
- [16] Y.Tian, P.L.Lewin, A.E.Davies, Z.Richardson, « *PD Pattern Identification Using Acoustic Emission Measurement And Neural Networks* », International Symposium on High-Voltage Engineering, London, 1999.
- [17] T.Kexiong, Z.Deheng, W.Zhenyuan, Z.Dongsong, L.Fuqi, « *Recognition of Partial Discharge Based on Neural Network* », 10th International Symposium on High-Voltage Engineering 1997.
- [18] J.A.S.B.Jayasinghe, R.K.Aggarwal, A.T.Johns, J.Downes, « *a New Approach to Condition Monitoring of High Voltage Overhead Lines Using Neural Networks* », International Symposium on High-Voltage Engineering, London, 1999.
- [19] Y.Kawaguchi, T.Shimizu, « *Neural Networks Application to The Fault Location in THE Transformer Winding* », International Symposium on High-Voltage Engineering, London, 1999.
- [20] A.Boubakeur, « *Claquage des Diélectriques Liquides* », cours de technique de HT, quatrième année électrotechnique.
- [21] R.Tobazéon, « *Conduction Electrique dans les Liquides* », techniques de l'ingénieur, D2 430, 1986.
- [22] R.Tobazéon, « *Liquides Diélectriques, Claquage Et Préclaquage* », techniques de l'ingénieur, D226, 1986.
- [23] H.Moulai, « *Contribution a L'étude des Phénomènes de Streamers dans les Diélectriques Liquides en Géométrie Pointe-Plan Sous Tension De Choc De Foudre* », thèse de magister, ENP, 1990.
- [24] D.Ouakli, H.Brouri, « *Propriétés Diélectriques de L'huile de Transformateur en Présence De Gaz Dissous* », thèse de PFE, ENP, Juillet 1997.
- [25] R.Tobazéon, « *Huiles et Liquides Isolants* », techniques de l'ingénieur, D230, 1986.
- [26] S.Abdi, S.Zelmat, « *Influence du Vieillissement Thermiques sur les Propriétés de L'huile de Transformateur* », thèse de PFE, ENP, Juin 1998.
- [27] B.Chalabi, S.Ouabdesselam, « *Analyse des Gaz Dissous dans les Huiles Isolantes* », thèse de PFE, ENP, Juillet 1997.
- [28] M.Mameri, « *Contribution a la Caractéristiques Physico-Chimique et Electrique d'une Huile Isolante de Transformateur Neuve et Usagée* », thèse de PFE, ENP, Juillet 1997.
- [29] SM.sekhri, K.Nouasri, « *Influence de la Température sur Certaines Propriétés de L'huile de Transformateur Utilisée par la SONELGAZ* », deuxième conférence nationale sur la la haute tension, Tizi-ouzou 1997.

- [30] R.Fournié, « *Les Isolants en Electrotechnique, Essais, Mécanismes de Dégradation. Application Industrielles* », Ed Eyrolles, Paris 1990.
- [31] H.A.Nadhra, « *Contribution a l'étude des Phénomènes de Préclaquage dans une Huile de Transformateur sous Champs Alternatif Divergent* », thèse de magister,ENP, Octobre 1997.
- [32] A.Boubakeur, « *Les Essais sur les Diélectriques Liquides* », cours de technique de HT, quatrième année électrotechnique.
- [33] A.Boubakeur, et autres, « *Influence du Vieillissement Thermique sur les Propriétés de BORAC22* », rapport interne, ENP.