

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار - عنابة

Année : 2018

Faculté: Sciences de l'ingénierat
Département: Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé

Commande d'un four électrique par réseau de neurone

Domaine : Sciences et Technologie.

Filière : Automatique.

Spécialité : Automatique et informatique industriel.

Par : M^{elle} SLIMI wafa.

DEVANT Le JURY

Président : M. BEKAIK.

Directeur de mémoire : CHAKER KARIMA.

Examineur : S. BENMOUSSA.

Examineur : N. DOUGHMANE.

Remercîment

Tout d'abord, louange à « **Allah** » qui m'a guide sur le droit chemin tout au long de ce travail et m'a inspiré les bons pas et les justes reflexes. Sans sa miséricorde, ce travail n'aura pas abouti.

Au terme de ce travail, je tiens à exprimer toute ma reconnaissance et remerciements à **chaker karima**, qui a fait preuve d'une grande patience et été un grande apport pour la réalisation de ce travail. Ses conseils, ces orientations ainsi que son soutien morale et scientifique m'ont permis de mener à terme ce projet.

Son encadrement était des plus exemplaires.

Je remercie également tous l'ensemble des enseignants pour la formation reçue d'eux et tous qui ma aidé et orienté pour élaboré ce travail.

Dédicace

Je dédie ce mémoire,

A mes parents ma Méré et mon père qui ont toujours été là pour moi ,je ne pourrais jamais les remercier assez pour leur soutien et les sacrifices qu'ils ont consentis pendant la durée de mes études en leurs souhaitant une longue vie pleine de joie et de santé.

A ma chère sœur

A mes frères

A tous mes cousins et mes amis

A mes chers amis de promo je n'oublierai jamais les bons moments que nous avons passés ensemble .

Table des matières

INTRODUCTION GENERALE

CHAPITRE 01

1.introduction	01
1.2historique	01
1.3 domaine d'application	02
1.4principe des réseaux de neurones artificiels (RNA).....	02
1.4.1définition	02
1.4.2elements de base	02
1.4.3neurone artificiel.....	03
1.4.3.1types de modeles de neurones	03
1.4.4fonction fe transfert	05
1.5perceptron	07
1.5.1 les différents types de perceptron.....	07
1.5.1.1reseau feed-forward.....	08
1.5.1.1.1perceptron monocouche	08
1.5.1.1.2perceptron multicouches.....	08
1.5.1.1.3réseau à fonction radiale.....	09
1.5.1.2réseau Feed-back.....	09

1.5.1.2.1	les cartes auto-organisatrices de kohonen.....	09
1.5.1.2.2	les réseaux de hopfilef	10
1.5.1.2.3	Les ART.....	10
1.6	Apprentissage des réseaux de neurones	10
1.6.1	Surapprentissage	11
1.6.2	types d'apprentissage.....	11
1.6.2.1	Apprentissage supervisé.....	11
1.6.2.2	Apprentissage renforcé	12
	Apprentissage non supervisé	12
	Algorithme d'apprentissage	12
	Conception d'un réseau de neurones	13
	Détermination des entrées /sorties du réseau de neurone	14
	Choix et préparation des échantillons	14
	Conclusion.....	15

Chapitre2

2.1	Introduction.....	16
2.2	Généralités	16
2.3	Puissance des fours a résistance	16
2.4	Les différents types de fours a résistance	17
2.5	Modulisation du four.....	19
	conclusion.....	21

Chapitre3

3.1 Introduction.....	23
3.2 Construction d'un réseaux de neurones à couches cachées.....	23
3.3 Algorithme de rétropropagation	24
3.4 Commande par modèle inverse neuronal.....	27

المخلص

يصنف هذا العمل في مجال التحكم الآلي بشكل عام ومجال الذكاء الاصطناعي على وجه الخصوص وهي التحكم في الأنظمة الغير خطية والتي. من خصائصها التي تتميز بها تقريبا "RN" مخصصة لاستخدام الشبكات العصبية نموذج أي وضيفة غير خطية إلى الحقيقة مع جودة درجة دقة اعتباطية وقد استخدم في هذا المشروع التحكم عن طريق نموذج معكوس العصبية .

والمستعملة بكثرة في مشاكل الانحدار غير الخطية وتسمح في الواقع بتمثيل جميع النظم الغير غير خطية إذا تم اختيار هيكلها بشكل مناسب

-يتم التحقيق من صحة العمل من خلال المحاكاة في MATLAB

Résumé

Ce travail est consacré à l'application des réseaux de neurones RN, dans le domaine de l'automatique de manière générale, et plus particulièrement pour l'identification du modèle inverse d'un processus modélisant un four électrique dans le but de l'utiliser pour la commande de la température de ce four. Ce qui a valu à ce type d'approche le nom de *commande neuronale par modèle inverse*. L'application des RN est due à leurs propriétés particulières qui les rend capables d'approximer avec un certain degré de précision, n'importe quelle fonction non linéaire, et d'autre part pour nous affranchir des difficultés inhérentes au calcul de l'inverse d'un modèle par les méthodes habituelles. Ce travail est appuyé par des résultats de simulations sous l'environnement MATLAB, obtenus par l'application des RN sur le modèle d'un four électrique. .

abstract

This work is in the field of automation in general, and artificial intelligence in particular, is devoted to the use of RN neural networks to control nonlinear dynamic systems. This is due to their universal approximation property which makes them able to approximate, with an arbitrary degree of precision fixed, any non-linear function, in this project, the control of a furnace by a neuronal inverse model, which is a widely used implantation, has been used. The work is validated by simulations under the MATLAB environment

Introduction générale

Les principales difficultés dans la théorie de la commande des systèmes dynamiques réels sont les non-linéarités et les incertitudes. Ceci étant, la commande est donc élaborée sur la base d'un modèle mathématique du système, en trouvant une relation entre les entrées et les sorties, ce qui suppose une bonne connaissance de la dynamique du système ainsi que de ses propriétés. Ainsi, l'étape de modélisation est très importante, car il en découle la garantie du bon fonctionnement du processus de command, c'est-à-dire la réalisation de performances optimales.[4, 5]

Comme la modélisation d'un processus se base sur la représentation du comportement dynamique d'un système à l'aide d'un modèle mathématique, et compte tenu des incertitudes et perturbations auxquelles le système peut être soumis, il n'est parfois pas suffisant de disposer d'une famille de modèles mathématiques très large pour obtenir des performances idéales réalisable par ce modèle. Ceci étant, le but de ce travail est d'utiliser une méthode de modélisation capable de prédire les paramètres du modèle sans passer par des représentations mathématiques laborieuse et complexe mais plutôt basée sur une approche statistique connu sous le nom de *réseau de neurone artificiel*. Est l'identification du modèle inverse d'un processus dans le but de utiliser pour se du commander.

Ceci étant, il faut savoir qu'un réseau de neurone est un ensemble d'opérateurs non linéaires interconnectés, recevant des signaux de l'extérieur par ses entrées, et délivrant des signaux de sorties qui sont des activités de certains neurones. L'idée de ce concept est de faire apprendre à cette base de données une fois constituée, la relation entre les entrées et sorties dans un processus appelé *apprentissage*, puis d'utiliser ces résultats pour l'identification des paramètres du modèle inverse du système à simuler d'où le nom de commande par modèle inverse neuronale [6]

Pour mettre en application cette méthode, nous avons opté pour la modélisation d'un processus très utilisé encore de nos jours en industrie, à savoir un *four électrique*. Les réseaux de neurones sont des outils très pratiques qui permettent, grâce à leur propriété d'approximation universelle, de reproduire le comportement des systèmes les plus complexes. Ils sont notamment utilisés aussi bien dans la modélisation que dans la commande des systèmes c'est pourquoi nous l'utilisons pour identifier le modèle inverse du four électrique dans le but de commander sa température.

L'ensemble de notre travail est regroupé dans trois chapitres.

✓ **Premier chapitre**

Dans cette partie, nous allons présenter les fondements théoriques et les éléments de base des réseaux de neurone

✓ **Deuxième chapitre**

Dans ce chapitre nous présentons les caractéristiques spécifiques des fours électriques à résistance

✓ **Troisième chapitre**

Ce chapitre concerne la présentation des résultats de l'application de la commande neuronale par inversion de modèle, sur le modèle du four électrique.

Enfin, nous terminons par une conclusion générale sur les performances de la commande appliquée.

1.1 Introduction :

Les réseaux de neurones artificiels, connus généralement sous l'acronyme **ANN (Artificiel Neural Network)**, sont des modèles mathématiques non linéaires de type « boîte noir » capables d'établir des relations entre les entres et les sorties d'un système

Les performances de ces derniers dans la modélisation non linéaire ont été prouvées dans plusieurs domaines.

1.2 Historique

Une première vague d'intérêt aux réseaux de neurones a vu le jour après l'introduction par Mc Culloch et Pitts en 1943 la notion des neurones simplifiés (perceptron) [1]. Ces neurones élément de base pour des circuits (réseaux) capables de réaliser des fonctions logique, arithmétiques et symboliques complexes. En 1949 Hebb, physiologiste américain propose une loi de modification des propriétés des connexions entre les neurones, qui explique en partie le conditionnement pavlovien chez l'animal.

Les premiers succès de cette méthode furent présentés par Rosenblatt en 1957 qui a construit le premier neuro-ordinateur basé sur le modèle du perceptron et l'appliqué au domaine de la reconnaissance des formes. En 1960 Widrow un automaticien, a développé le modèle Adaline qui a amélioré la loi d'apprentissage.

Quand Minsky et Papert publièrent en 1969 leur ouvrage qui en exergue les limitations théoriques du perceptron, il y a abandon financier des recherches dans le domaine, les chercheurs se tournèrent principalement vers l'intelligence artificielle (IA). Quelques chercheurs ont continué leurs efforts parmi lesquels on note Teuvo Kohonen, Stephen Grossberg, James Anderson et Kunihiko Fukushima.

En 1982, Hopfield, un physicien reconnu à qui l'on doit le renouveau d'intérêt pour les réseaux de neurones artificiels, au travers d'un article court, clair et bien écrit, il présente une théorie du fonctionnement et des possibilités des réseaux de neurones. Hopfield fixe préalablement le comportement à atteindre pour son modèle et construit à partir de la structure et la loi d'apprentissage correspondant au résultat escompté. Notons qu'à cette date l'IA est l'objet d'une certaine désillusion, elle n'a pas répondu

à toutes les attentes et s'est même heurtée à de sérieuses limitations. Aussi, bien que les limitations du perceptron mises en avant par Minsky ne soient pas levées par le modèle d'Hopfield, les recherches sont relancées. La machine de Boltzmann (1983) est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas du perceptron. Mais l'utilisation pratique s'avère difficile, la convergence de l'algorithme étant extrêmement longue (les temps de calcul sont considérables).

En 1985, la rétropropagation du gradient apparaît. C'est un algorithme d'apprentissage adapté aux réseaux de neurones multicouches. Dès cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables. De nos jours, les réseaux multicouches et la rétropropagation de gradient restent le modèle le plus étudié au niveau des applications.

Depuis, les réseaux neuronaux ont connu un grand essor vu les applications potentielles et la disponibilité de grand calculateur. En fait les réseaux de neurone sont utilisés dans de nombreux domaines à cause de leurs propriétés et en particulier, leur capacité d'apprentissage et de généralisation.

Néanmoins les réseaux de neurones artificiels ont besoin de cas réels servant d'exemples pour leur apprentissage. Ces cas doivent être d'autant plus nombreux que le problème est complexe et que sa topologie est peu structurée. Sur un plan pratique. Cela n'est pas toujours facile car les exemples peuvent être soit en quantité absolument limitée ou trop onéreux à collecter en nombre suffisants.

1.3 Domaines d'application

Grâce à leur capacité dans la classification, approximation, et généralisation les réseaux de neurones ont été utilisés dans plusieurs domaines :

- Classification
- Approximation
- Commande robotique
- Prédiction
- Mémoire associative

1.4 Principe des Réseau de Neurones Artificiels (RNA)

1.4.1 Définition

Le réseau de neurones, fabriquée de structures cellulaires artificielles, est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des neurones biologiques.

1.4.2 Eléments de base

Les éléments de base du système nerveux centrale (cerveau) sont les neurones[2], leur structure est composée particulièrement de trois parties : le corps cellulaire, les dendrites et l'axone voir (fig.1.1).

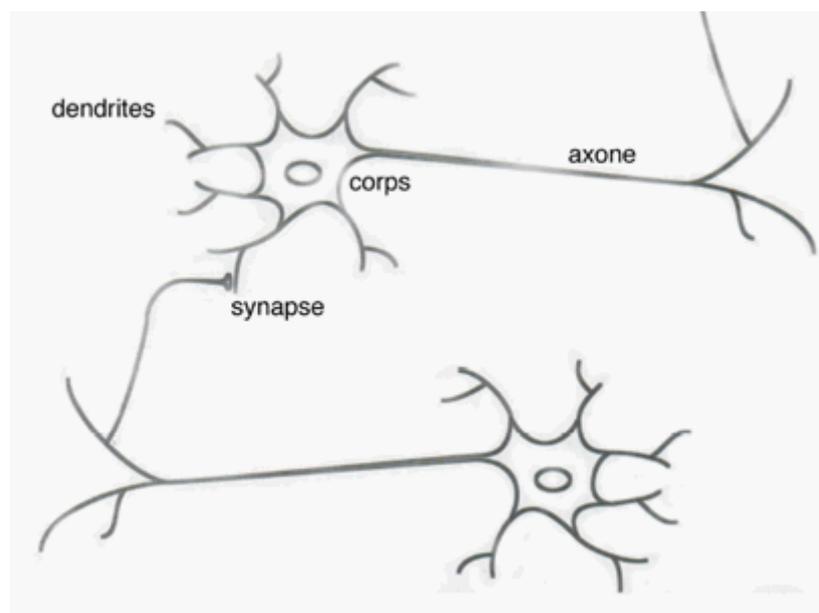


Figure1.1. la cellule nerveuse et ces composantes

1.4.3 Neurone artificiel

Neurone informatique : c'est un petit modèle statistique à lui tout seul, il doit accomplir deux tâches suivantes :

* La synthèse des informations qui lui sont fournies.

* La transformation, (souvent non linéaire), de cette information en quelque chose de nouveau.

Ce fonctionnement est expliqué ci-dessous par les différents modèles neuronaux. [9]

1.4.3.1 Types de modèles de neurones

a) **Modèle d'un neurone à entrée simple** (voir figure 1.2)

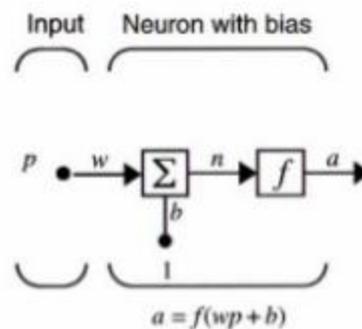


Figure 1.2 neurone à simple entrée

Le neurone de sortie est calculé par l'équation : $a=f(wp+b)$. Si par exemple :

$$\begin{cases} W=3 \\ P=2 \\ B=-1.5 \end{cases} \implies a=f(3*2-1.5) = f(4.5)$$

La sortie dépend de la fonction de transfert choisie. Le biais est quasi similaire au poids sauf qu'il possède une valeur d'entrée constante égale à 1. Il peut cependant être omis selon les conditions choisies par l'utilisateur. A noter que w et b sont des paramètres scalaires ajustables de neurone.

Typiquement, la fonction de transfert f est choisie par l'utilisateur et les paramètres w et b sont, ajustés par des lois d'apprentissage de façon à adapter le neurone entrée/sortie à un but spécifique.

b) **Neurone à entrées multiples**

Typiquement, un neurone possède plus d'une entrée, comme le montre la figure (1.3) ci-dessous.

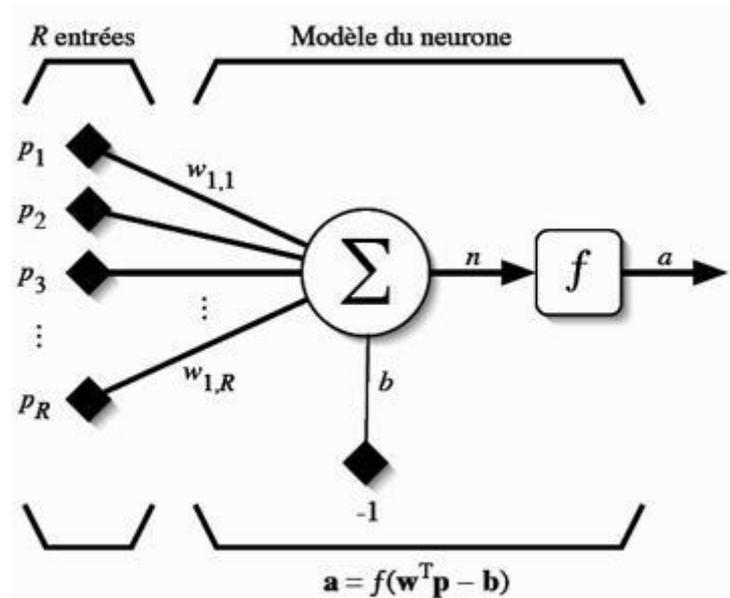


Figure 1.3 Neurone à entrées multiples

Les entrées individuelles p_1, p_2, \dots, p_r sont pondérées par les éléments $w_{1,1}, w_{1,2}, \dots, w_{1,r}$ de la matrice \mathbf{W} . Le neurone possède un biais b , sommé avec les entrées pondérées pour former l'entrée n du réseau

$$n = w_{1,1} * p_1 + w_{1,2} * p_2 + \dots + w_{1,r} * p_r + b \quad (1.2)$$

L'équation peut être formulée sous forme matricielle

$$n = \mathbf{W}\mathbf{p} + b \quad (1.3)$$

1.4.2 Fonction de transfert

Cette fonction peut être linéaire ou non. Une fonction particulière est choisie pour satisfaire les spécificités d'un problème que le neurone est appelé à résoudre.[10]

Il existe plusieurs types de fonction les plus utilisées sont présentées comme suite :

- **Fonction « hard limit »** : Elle donne la valeur 0 au neurone de sortie si l'argument de la fonction est négatif, et s'il est positif ou nul, comme l'illustre la figure 1.4.

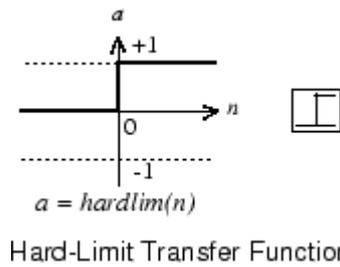


Figure 1.4 Fonction de transfert « hard limit »

Cette fonction est utilisée pour créer des neurones capables de classifier des entrées en deux catégories distinctes.

- **Fonction linéaire** : la valeur de sortie est égale à la valeur d'entrée $a=n$, comme l'illustre la figure 1.5.

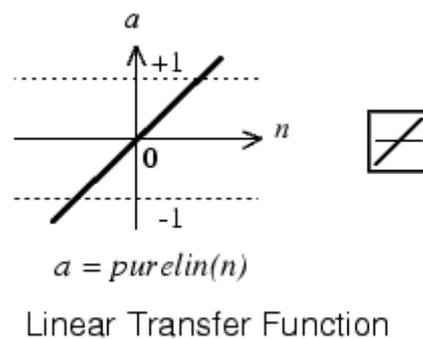


Figure 1.5. fonction de transfert linéaire

- **Fonction log-sigmoïd** : Elle est communément utilisée dans les réseaux de neurones multicouche éduqués avec un algorithme de rétropropagation .

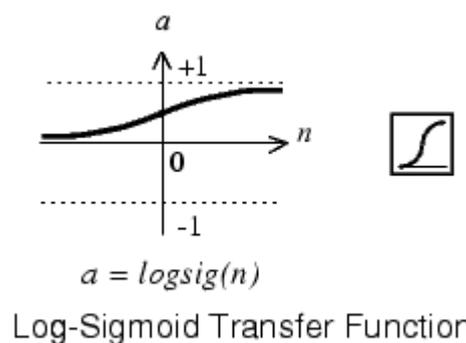


Figure 1.6. Fonction de transfert log-sigmoïd

La valeur d'entrée est comprise entre $]-\infty ; +\infty [$. La fonction aplatit la valeur de sortie entre $[0 ; 1]$ selon l'expression

1.4 Les perceptrons

De manière générale, un neurone, même avec de nombreuses entrées, n'est pas suffisant. Il en faut plusieurs, opérant en parallèle. Ce type de RNA est appelé perceptron. Ce dernier peut avoir une seule couche comme indiqué à la figure (1.7) ou plusieurs couches. Dans ce cas, il sera appelé réseaux de neurones multicouches, autrement dit perceptron multicouches

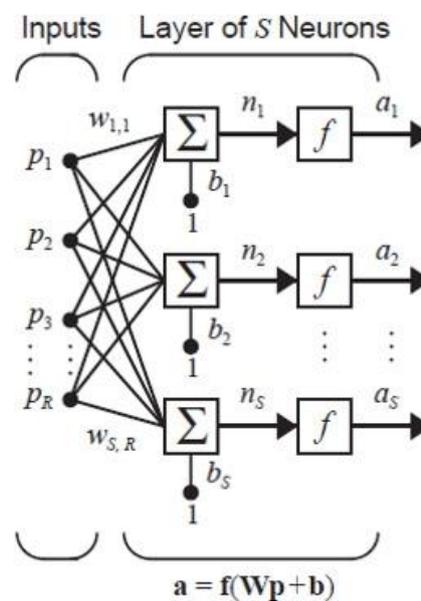


Figure 1.7 Couche de S-neurones

Les réseaux multicouches sont plus puissants que les réseaux à simple couche. Par exemple, un réseau à 2 couches avec une couche 1 sigmoïde et une couche linéaire peut être éduqué pour approximer convenablement la plupart des fonctions. Un réseau à simple couche ne le peut pas.

Dans la pratique, la plupart des réseaux ne possèdent que 2 ou 3 couches. Une fois créé, le réseau de neurone est utilisé selon le protocole suivant :

- Initialisation des poids et des biais.
- Apprentissage avec une base de données connue.

- Simulation avec des nouvelles données

1.5.1. Différents types de perceptrons

On peut classer les réseaux de neurones artificiels (RNA) type perceptron en deux grandes catégories :

- * Réseaux Feed-Forward
- * Réseaux Feed-Back

1.5.1.1- Les réseaux Feed-Forward

Appelés aussi « réseaux de type perceptron », ce sont des réseaux dans lesquels l'information se propage de couche sans arrière possible.

1.5.1.1.1 perceptron monocouche

Historiquement le première RNA, c'est le perceptron de Rosenblatt[3]. C'est un réseau simple, puisqu'il ne se compose d'une couche d'entrée et d'une couche de sortie, voir figure (1.8). Il est calqué, à la base, sur le système visuel et de ce fait été conçu dans un but premier de reconnaissance des formes. Cependant, il peut aussi être utilisé pour faire de la classification et pour résoudre des opérations logiques simples (telle « et » ou « ou »). sa principale limite est qu'il ne peut résoudre que des problèmes linéairement séparables. Il suit généralement un apprentissage supervisé selon la règle de correction de l'erreur (ou selon la règle de Hebb).

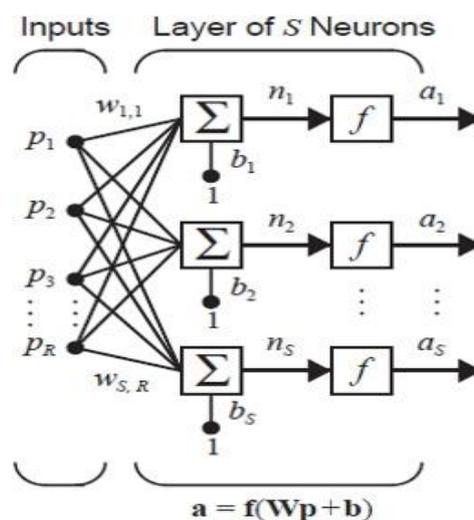


Figure 1.8. Perceptron monocouche

1.5.1.1.2 perceptron multicouches (PMC)

C'est une extension du précédent, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante (excepté pour les couches d'entrée et de sortie) et il n'y a pas de connexions entre les cellules d'une même couche. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions à seuil ou sigmoïdes. Il peut résoudre des problèmes non-linéairement séparables et des problèmes logiques plus compliqués, et notamment les fameux problèmes de XOR. Il suit aussi un apprentissage supervisé selon la règle de correction de l'erreur, voir figure (1.9).

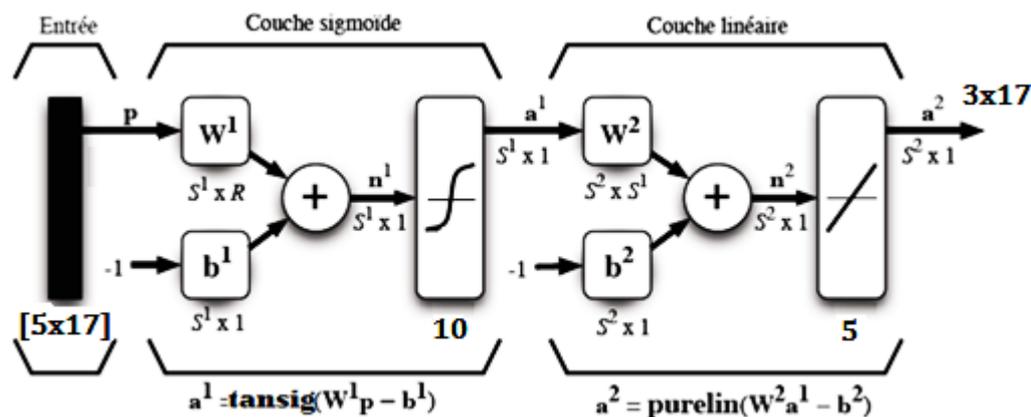


Figure 1.9 perceptron multicouches

1.5.1.1.3 Les réseaux à fonction radiale

Ce sont les réseaux que l'on nomme aussi RBF (« radial basic fonction »). L'architecture est la même que pour les PMC cependant, les fonctions de base utilisées ici sont des fonctions Gaussiennes. Les RBF seront donc employés dans les mêmes types de problèmes que les PMC à savoir, en classification et en approximation de fonctions, particulièrement. L'apprentissage le plus utilisé pour les RBF est le mode hybride et les règles sont soit, la règle de correction de l'erreur soit, la règle d'apprentissage par compétition.

1.5.1.2 Les réseaux feed-back

Appelés aussi réseaux récurrents ce sont des réseaux dans lesquels il y a retour en arrière de l'information, voir figure (1.10)

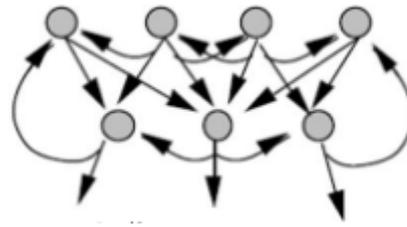


Figure 1.10. Réseau à connexions récurrentes

1.5.1.2.1 Cartes auto-organisatrices de Kohonen

Ce sont des réseaux à apprentissage non-supervisé qui établissent une carte discrète, ordonnée typologiquement, en fonction de patterns d'entrée. Le réseau forme ainsi une sorte de treillis dont chaque nœud est un neurone associé à un vecteur de poids. La correspondance Entre chaque vecteur de poids est calculée pour chaque entrée. Par la suite, le vecteur de poids ayant la meilleur corrélation, ainsi que certains de ses voisins, vont être modifiés afin d'augmenter encore cette corrélation.

1.5.1.2.2 Réseaux Hopfield

Les réseaux de Hopfield sont des réseaux récurrents et entièrement connectés. Dans ce type de réseau, chaque neurone est connecte a chaque autre neurone et il n'y a aucune différenciation entre les neurones d'entrée et de sortie .ils fonctionnent comme une mémoire associative non-linéaire et sont capables de trouver un objet stocké en fonction de représentation partielles ou bruitées. L'application principale des réseaux des réseaux de Hopfield est l'entrepôt de connaissances mais aussi la résolution de problèmes d'optimisation. Le mode d'apprentissage utilisé ici est le mode non supervisé.

1.5.1.2.3ART

Les réseaux ART ("Adaptative Résonance Théorie") sont des réseaux à apprentissage par compétition. Le problème majeur qui ce pose dans ce type de réseaux est le dilemme " stabilité/plasticité". En effet, dans un apprentissage par compétition, rien ne garantit que les catégories formées aillent rester stables. LA

Seule possibilité, pour assurer la stabilité, serait que le coefficient d'apprentissage tende vers zéro, mais le réseau perdrait alors sa plasticité.

Les ART ont été conçus spécifiquement pour contourner ce problème. Dans ce genre de réseau, Les vecteur de poids ne seront adaptés que si l'entrée fournie est suffisamment proche, d'un prototype déjà connu par le réseau. On parlera alors de résonance. A l'inverse, si l'entrée s'éloigne trop des prototypes existants, une nouvelle catégorie va alors se carreaudier, avec pour prototype, l'entrée qui a engendrée sa création. Il est à noter qu'il existe deux principaux types de réseaux ART : les ART-1 pour des entrées binaires et les ART-2 pour des entrées continues. Le mode d'apprentissage des ART peut être supervisé ou non.[8]

1.6 Apprentissage des réseaux de neurones

Le point crucial du développement d'un réseau de neurones est sont apprentissage. Il s'agit d'une procédure adaptative par laquelle les connexions des neurones sont ajustées face a une source d'information (Hebb ; 1949 ; Grossberg, 1982 ; Rumelhart et al, 1986). Dans le cas des réseaux neurones artificiels, on ajoute souvent à la description du modèle l'algorithme d'apprentissage. Le modèle sans apprentissage présente en effet peu d'intérêt.

Dans la majorité des algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions. L'apprentissage est la modification des poids du réseau dans l'optique d'accorder la réponse du réseau aux exemples et l'expérience. Les poids sont initialisés avec des valeurs aléatoires. Puis des données représentatives du fonctionnement du procédé dans un domaine donné, sont présentées au réseau de neurones. Ces exemples sont constitués de couples de vecteurs d'entrée et de sortie. Une méthode d'optimisation modifie les pois au fur et à mesure des itérations pendant lesquelles on présente la totalité des exemples, afin de minimiser l'écart entre les sorties simulées et les sorties cibles. Afin d'éviter les problèmes de sur-apprentissage, la base d'exemples est divisée en deux parties : la base d'apprentissage et la base de test. L'optimisation des poids se fait sur la base d'apprentissage, mais les poids retenus sont ceux pour lesquels l'erreur obtenue sur la base de test est la plus faible. En effet, si les poids sont optimisés sur tous les exemples de l'apprentissage, on obtient une précision très satisfaisante sur ces

exemples mais on risque de ne pas pouvoir généraliser le modèle à des données nouvelles. A partir d'un certain nombre d'itérations, le réseau ne cherche plus l'allure générale de la relation entre les entrées et les sorties du système, mais s'approche trop près des points et "apprend" le bruit (Pollard et al.92).

1.6.1 Sur-apprentissage

Il arrive qu'à faire apprendre un réseau de neurones toujours sur le même échantillon, celui-ci devient inapte à reconnaître autre chose que les éléments présents dans l'échantillon. Le réseau ne cherche plus l'allure générale de la relation entre les entrées et les sorties du système, mais cherche à reproduire les allures de l'échantillon. On parle alors de sur-apprentissage : le réseau est devenu trop spécialisé et ne généralise plus correctement.

Ce phénomène apparaît aussi lorsqu'on utilise trop d'unités cachées (de connexion), la phase d'apprentissage devient alors trop longue (trop de paramètres réglables dans le système) et les performances du réseau en généralisation deviennent médiocres.

1.6.2 Type d'apprentissage :

Il existe de nombreux types de règles d'apprentissage qui peuvent être regroupées en trois catégories (Hassoun, 1995) : les règles d'apprentissage supervisé, non supervisé, et renforcé.

Mais l'objectif fondamental de l'apprentissage reste le même : soit la classification, l'approximation de fonction ou encore la prévision (Weiss et Kulikowski, 1991). Dans l'optique de la prévision, l'apprentissage consiste à extraire des régularités (à partir des exemples) qui peuvent être transférées à des nouveaux exemples.

1.6.2.1 Apprentissage supervisé

Un apprentissage est dit supervisé lorsque l'on force le réseau à converger vers un état final précis, en même temps qu'on lui présente un motif. Ce genre d'apprentissage est réalisé à l'aide d'une base d'apprentissage, constitué de plusieurs exemples de type entrée-sorties (les entrées du réseau et les sorties désirées ou encore les solutions souhaitées pour l'ensemble des sorties du réseau).

La procédure usuelle dans le cadre de la prévision est l'apprentissage supervisé (ou à partir d'exemples) qui consiste à associer une réponse spécifique désirée à chaque signal d'entrée. La modification des poids s'effectue progressivement jusqu'à ce que l'erreur (ou l'écart) entre les sorties du réseau (ou résultats calculés) et les résultats désirés soient minimisés.

Cet apprentissage n'est possible que si un large jeu de données est disponible et si les solutions sont connues pour les exemples de la base d'apprentissage.

1.6.2.2 Apprentissage renforcé

L'apprentissage renforcé est une technique similaire à l'apprentissage supervisé à la différence qu'au lieu de fournir des résultats désirés au réseau, on lui accorde plutôt un grade (ou score) qui est une mesure de degré de performance du réseau après quelques itérations. Les algorithmes utilisant la procédure d'apprentissage renforcé sont surtout utilisés dans le domaine des systèmes de contrôle (White et Sofge, 1992 ; Sutton ; 1992).

1.6.2.3 Apprentissage non supervisé

L'apprentissage non supervisé consiste à ajuster les poids à partir d'un seul ensemble d'apprentissage formé uniquement de données. Aucun résultat désiré n'est fourni au réseau.

Qu'est-ce que le réseau apprend exactement dans ce cas ? L'apprentissage consiste à détecter les similarités et les différences dans l'ensemble d'apprentissage. Les poids et les sorties du réseau convergent, en théorie, vers les représentations qui capturent les régularités statistiques des données (Fukushima, 1988 ; Hinton, 1992). Ce type d'apprentissage est dit compétitif et (ou) coopératif (Grossberg, 1988). L'avantage de ce type d'apprentissage réside dans sa grande capacité d'adaptation reconnue comme une auto-organisation, " self-organizing" (Kohonen, 1987). L'apprentissage non supervisé est surtout utilisé pour le traitement du signal et l'analyse factorielle.

1.6.3 Algorithme d'apprentissage

L'algorithme d'apprentissage est la méthode mathématique qui va modifier les poids de connexions afin de converger vers une solution qui permettra au réseau d'accomplir la tâche d'optimiser les valeurs des poids du réseau.

Plusieurs algorithmes itératifs peuvent être mis en œuvre, parmi lesquels on note :

L'algorithme de rétro propagation, la méthode Quasi-Newton, Algorithme de BFGS, etc....

Dans ce qui suit, nous allons aborder que celui que nous avons utilisé dans ce mémoire.

1.6.3 Algorithme de rétropropagation (ARP)

L'algorithme de rétropropagation (ARP) ou de propagation arrière "backpropagation" est l'exemple d'apprentissage supervisé le plus utilisé à cause de l'écho médiatique de certaines applications spectaculaire telles que la démonstration de Sejnowski et Rosenberg (1987) dans laquelle l'ART est utilisé dans un système qui apprend à lire un texte. Un autre succès fut la prédiction des cours du marché boursier (Refenes et al, 1994 ; Lee et al, 1996) et plus récemment la détection de la fraude dans les opérations par cartes de crédit (Dorrnsoro et al, 1997).

La technique de rétropropagation du gradient (Backpropagation en anglais) est une méthode qui permet de calculer le gradient de l'erreur pour chaque neurone du réseau, de la dernière couche vers la première. L'historique de la publication montre que l'ARP a été découvert indépendamment par différents auteurs mais sous différentes appellations (Grossberg, 1998). Le principe de la rétropropagation peut être décrit en trois étapes fondamentales : acheminement de l'information à travers le réseau ; rétropropagation des sensibilités et calcul du gradient ; ajustement du paramètre par la règle du gradient approximé. Il est important de noter que l'ARP souffre des limitations inhérentes à la technique du gradient à cause du risque d'être piégé dans un minimum local. Il suffit que les gradients ou leurs dérivées soient nuls pour que le réseau se retrouve bloqué dans un minimum local. Ajoutons à cela la lenteur de convergence surtout lorsqu'on traite des réseaux de grande taille (c.à.d. pour lesquels le nombre de poids de connexion à déterminer est important).

1.7 Conception d'un réseau de neurones

Les réseaux de neurones réalisent des fonctions non linéaires paramétrées. Leurs mises en œuvre nécessitent :

- * la détermination des entrées et des sorties pertinentes, c'est ç dire les grandeurs qui ont une influence significative dur le phénomène que l'on cherche à modéliser.

- * la collecte des données nécessaire à l'apprentissage et à l'évaluation des performances du réseau de neurones.

- * La détermination du nombre de neurones cachés nécessaires pour obtenir une approximation satisfaisante.

- * la réalisation de l'apprentissage.

- * L'évaluation des performances du réseau de neurones à l'issue de l'apprentissage.

1.7.1 Détermination des entrées/sorties du réseau de neurones

Pour toute conception de modèle, la sélection des entrées doit prendre en compte deux points essentiels :

a- Premièrement, la dimension intrinsèque du vecteur des entrées doit être aussi petite que possible, e, d'autre terme, la représentation des entrées doit être la plus compacte possible, tout en conservant pour l'essentiel la même quantité d'information, er en gardant à l'esprit que les différentes entrées doivent être indépendantes.

b- En second lieu, toutes les informations présentées dans les entrées doivent être pertinentes pour la grandeur que l'on cherche à modéliser : elles doivent donc avoir une influence réelle sur la valeur de la sortie.

1.7.2 choix et présentation des échantillons

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix et la préparation des échantillons de données. La façon dont se présente l'échantillon conditionne le type de réseau, le nombre de cellules d'entrées, le nombre de cellules de sorties et la façon dont il faudra mener l'apprentissage, les tests et la validation (Bishop, 1995). Il faut donc déterminer les grandeurs qui en une influence significative sur le phénomène que l'on cherche à modéliser.

Lorsque la grandeur que l'on modéliser dépend de nombreux facteurs, c'est-à-dire lorsque le modèle possède de nombreuses entrées, il n'est pas possible de réaliser un " pavage " régulier dans tout le domaine de variation des entrées : il faut donc trouver une méthode permettant de réaliser uniquement des expériences qui apportent une information significative pour l'apprentissage du modèle. Cet objectif peut être obtenu en mettant en œuvre un plan d'expériences. Pour les modèles linéaire, l'élaboration de plans d'expérience est bien maîtrisée, par ailleurs, ce n'est pas le cas pour les modèles non linéaires.

Afin de développer une application à base de réseaux neurones, il est nécessaire de disposer de deux bases de données, une pour effectuer l'apprentissage et l'autre pour tester le réseau obtenu et déterminer ses performances.

Notons qu'il n'y a pas de règle pour déterminer ce partage d'une manière quantitative, néanmoins chaque base doit satisfaire aux contraintes de représentativité de chaque classe de données et doit généralement refléter la distribution réelle, c'est à dire la probabilité d'occurrence des diverses classes (Nascimento, 2000).

1.8. Conclusion :

Dans cette partie, nous avons essayé de présenter les réseaux neurones artificiels afin de mieux comprendre pourquoi ils ont pu susciter un engouement si fort. L'élément essentiel des réseaux de neurone est qu'ils peuvent capter les dépendances non-linéaires de haut niveau entre les variables, ce qui possible grâce à la présence d'une transformation, elle-même non-linéaire, dans le calcul de valeur prédite.

2.1 Introduction

Le four électrique à résistances est sans doute le plus connu des appareils électrothermiques. Son utilisation industrielle remonte en effet au début des années 1920 et sa technique n'a cessé de se perfectionner depuis. Le principe de ce type de four est extrêmement simple : il est constitué d'une enceinte chauffée à l'aide de résistances électriques, très bien calorifugée pour réduire le plus possible les déperditions thermiques. La charge à chauffer est placée dans cette enceinte. Le four à résistances est un équipement à chauffage indirect, la chaleur produite par effet Joule par les résistances est transmise à la charge par rayonnement et convection. Dans cet article, l'accent a été mis surtout sur les caractéristiques spécifiques des fours à résistances, en particulier la nature et les performances des résistances comme corps de chauffe, la mise en œuvre des résistances, le transfert thermique entre les éléments chauffants et la charge, et l'exploitation et la conduite de ces fours. La robustesse et la fiabilité de ces équipements, leur facilité de mise en œuvre, la simplicité de maintenance et leur efficacité énergétique élevée font que les applications industrielles sont fort nombreuses, il n'est guère de secteur industriel qui n'y fasse appel. [4]

2.2. Généralités

Une des propriétés les plus remarquables de l'électricité comme source d'énergie thermique est son caractère protéiforme : elle est, en effet, mobilisable sous des formes très diverses en fonction du mode de transmission de l'énergie électrique. Les procédés électrothermiques peuvent être divisés en deux grandes catégories :

- Le chauffage indirect où la transmission d'énergie de la source vers le récepteur (corps à chauffer) obéit aux lois usuelles de la thermique.
- Le chauffage direct où le récepteur, parcouru par un courant électrique, est lui-même le siège du dégagement de chaleur, les échanges ultérieurs de chaleur s'effectuant selon les lois du thermique classique.

les fours spécifiques à une industrie notamment sont :

- les fours de l'industrie sidérurgique ;
 - les fours de traitement thermique des métaux
 - les fours de raffinage et de l'industrie chimique ;
 - les fours de cimenterie ;
 - les fours de l'industrie verrière ;
- les fours des industries céramiques et des produits réfractaires

2.3. Puissance des fours à résistances

L'évaluation du bilan thermique, en tenant compte des caractéristiques de chaque composant et des configurations, amène à chiffrer pour un cycle de chauffe ou en fonctionnement continu :

- L'énergie emmagasinée dans la charge.
- L'énergie emmagasinée dans les supports.
- L'énergie emmagasinée et le flux thermique transmis par les parois, la voûte et la sole
- Le flux thermique et l'énergie perdus par les joints, les entrées/sorties d'air et les ouvertures de portes.
- Le flux thermique perdu par les chauffages de l'atmosphère et les ponts thermiques (dont les passages froids des éléments chauffants).
- Le flux thermique perdu par l'eau de refroidissement.[3]

On arrive ainsi à considérer un bilan thermique évalué soit en énergie dans le cas d'un cycle de chauffe, soit en puissance pour un régime permanent. Cela permet de déterminer la puissance moyenne nécessaire et, partant de là, la puissance à installer compte tenu d'un coefficient de sécurité dépendant, entre autres, du type d'éléments chauffants, de la régulation, etc.

2.4. Classification des fours à résistance électrique

La classification des fours à résistances repose sur les critères suivants :

- Les différents types de fours à résistance électrique.

- La nature des résistances Fours électriques à résistances- Technologies de mise en œuvre et leur position par rapport à la charge (rayonnement direct ou indirect, convection forcée, ventilation de la chambre de chauffe)
- Le mode d'utilisation du four : continu ou chauffage au défilé, discontinu, utilisant des charges fixes ou statiques
- Le mode de manutention (tapis continu, vibrations, vis hélicoïdale, logerons mobiles...).
- La nature de l'atmosphère et du milieu : ambiance (air), atmosphère contrôlée, vide, milieu liquide, bain fluidisé.
- Le mode de transmission de la chaleur : rayonnement, convection.

La chaleur à l'intérieur du four électrique ventilé à modéliser.

Cette chaleur est produite par une résistance chauffante, commandé en tension V_c par un amplificateur de puissance. La mesure de la température se fait à partir d'un thermocouple placé dans une cavité de mesure et d'un amplificateur d'instrumentation produisant une tension V_m , image de la température q_m . L'ensemble capteur+amplificateur d'instrumentation sont supposés linéaires dans la gamme de température du four. Les différents paramètres constituant ce processus sont :

$Q=K_1 V_c$ est la quantité de chaleur produite,

R_a : la résistance thermique freinant la circulation de la chaleur du conduit jusqu'à l'enceinte du four,

C_a : la capacité calorifique de l'enceinte du four,

R_m : La résistance thermique freinant la circulation de la chaleur du four à l'intérieur de la cavité de mesure,

C_m : La capacité calorifique de la cavité de mesure,

R_f : La résistance de fuite freinant la circulation de la chaleur vers l'extérieur du four,

C_e : La capacité calorifique extérieur du four, considérée comme infinie,

$\Theta_a, \Theta_m, \Theta_e$: température respectives de l'enceinte du four, de la cavité de mesure, et de l'extérieur du four,

Θ_m : La température mesurée son image est la tension V_m ($V_m=K_2 q_m$).

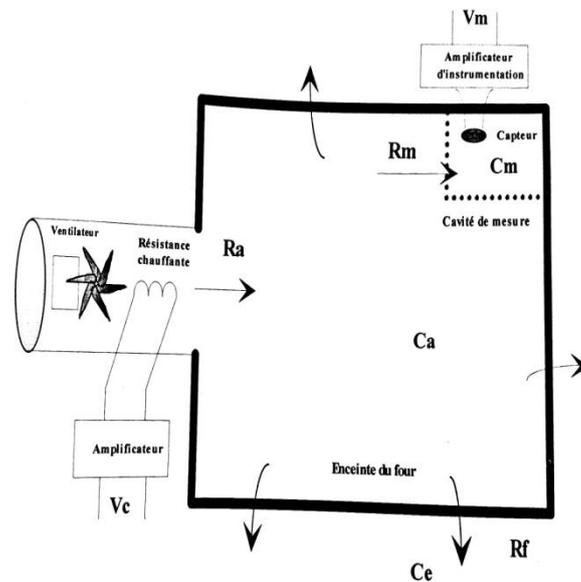


Figure 2.1 processus d'un four ventilé

2.5 Modélisation du four

La figure (2.1) montre le schéma électrique d'un four ventilé. Ce dernier est résistif – capacitif. [7]

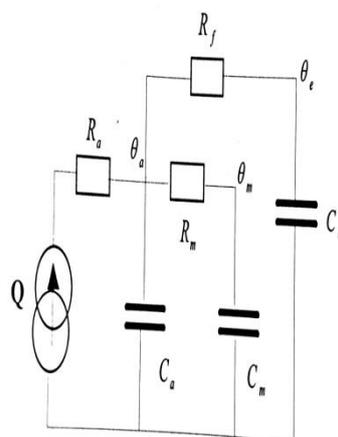


Figure 2.2 schéma électrique du four

Afin de donner un aspect plus explicite à notre étude nous avons trouvé nécessaire de donner le modèle mathématique du four ventile

Par identification entre les paramètres du four et le schéma électrique et en considérant la quantité de chaleur Q comme étant le courant, on obtient les équations suivantes :

$$Q = C_a \frac{d\theta_a}{dt} + C \frac{d\theta_m}{dt} + \frac{\theta_a - \theta_e}{R_f} \quad (2.1)$$

$C \frac{d\theta}{dt}$ étant le courant qui traverse la capacité, la température θ_m est exprimée par (2) comme suit :

$$\theta_m = \theta_a - R_m C_m \frac{d\theta_m}{dt} \quad (2.2)$$

Les paramètres θ_a , θ_m et θ_e ont été choisis comme étant respectivement les tensions (images des températures) des capacités C_a , C_m et C_e .

En utilisant la transformée de Laplace des expressions suivantes (1) et (2)

Nous obtenons les équations suivantes :

$$\theta_a = \left(Q + \frac{\theta_e}{R_f} \right) \cdot \left(\frac{R_f(1+R_m C_m p)}{1+(R_m C_m + R_f C_m + R_f C_a)p + R_f R_m C_m C_a p^2} \right) \quad (2.3)$$

$$\frac{\theta_m}{\theta_a} = \frac{1}{1+R_m C_m p} \quad (2.4)$$

Compte tenu des équations (3) et (4), le schéma fonctionnel du processus complet peut être décrit comme suit :

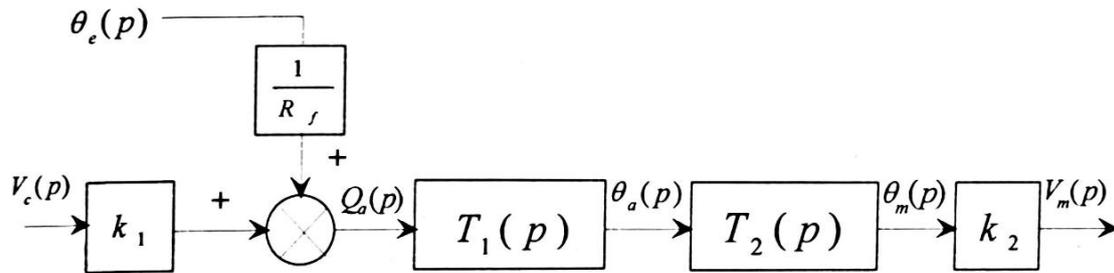


figure 2.3 schéma fonctionnel du four

avec

$$T1(p) = \frac{Rf(1+RmCmp)}{1+(RmCm+RfCm+RfCa)p+RfRmCm p^2} \quad (2.5)$$

$$T2(p) = \frac{1}{1+RmCmp} \quad (2.6)$$

Conclusion

Les fours électriques à résistances sont les équipements électrothermiques industriels les plus connus et les plus répandus, présents dans un grand nombre de procédés. Encore de nos jours, ils font l'objet de perfectionnements, comme l'amélioration de la durée de vie des résistances et l'élaboration de nouveaux isolants thermiques. Dans ce chapitre nous présentons détaillé les caractéristiques spécifiques des fours à résistance, dont la nature et les performances des résistances, le transfert entre les éléments chauffants et la charge. Ces équipements sont robustes et fiables, faciles à mettre en œuvre, ils présentent de plus une efficacité énergétique élevée.

3.1 Introduction

Nous allons appliquer la commande neuronale par modèle inverse pour commander un four électrique, dont la théorie de base de la commande appliquée ainsi que le modèle et la représentation d'état du système électrique équivalent au four, ont été présenté dans les deux premiers chapitres. En fait, nous allons détailler les étapes par lesquelles nous sommes passés pour construire la structure du réseau de neurone qui a fourni la variation de la commande par identification du modèle inverse du système, que l'on injectera par la suite au système du four électrique pour contrôler sa température, d'où le nom de '*commande par modèle inverse neuronal*'. Des résultats de simulation ont permis de confirmer l'efficacité de ce type de contrôle à commander la température du four.

3.2 Construction d'un réseaux de neurones à couches cachées

Les cellules de couche d'entrée propagent le stimulus vers la couche cachée via des connexions. Le vecteur d'entrée de la couche cachée est donné, sous forme matricielle, par

$$h_{in} = wx + w0 \quad (3.1)$$

Où w et $w0$ représentent respectivement la matrice de poids et du biais.

Chaque cellule de la couche cachée transforme son entrée en réponse à travers une fonction de transfert.

$$h = f(h_{in}) = f(wx + w0) \quad (3.2)$$

De même, chaque cellule de la couche de sortie transforme son entrée, en sortie par une fonction de transfert, la réponse de la couche de sortie est donnée par :

$$o = g(Zh + z0) \quad (3.3)$$

Les matrices de poids w , z et les vecteurs biais w_0, z_0 sont initialisés à des valeurs aléatoires, le vecteur de sortie du réseau est ensuite comparé au vecteur cible t que l'on cherche à obtenir et l'on déduit l'erreur $(t-o)$ en sortie du réseau.

Cette erreur sera rétro-propagée dans le réseau afin de mettre à jour les matrices de poids et les vecteurs de biais suivant un algorithme dit de rétropropagation de l'erreur qui minimise la somme des carrés des erreurs commises sur l'ensemble des stimuli.

La fonction de transfert ou d'activation utilisée est :

- Sigmoides unipolaires $\frac{1}{1+e^{-x}}$

$$f'(x) = f(x)[1 - f(x)] \quad (3.4)$$

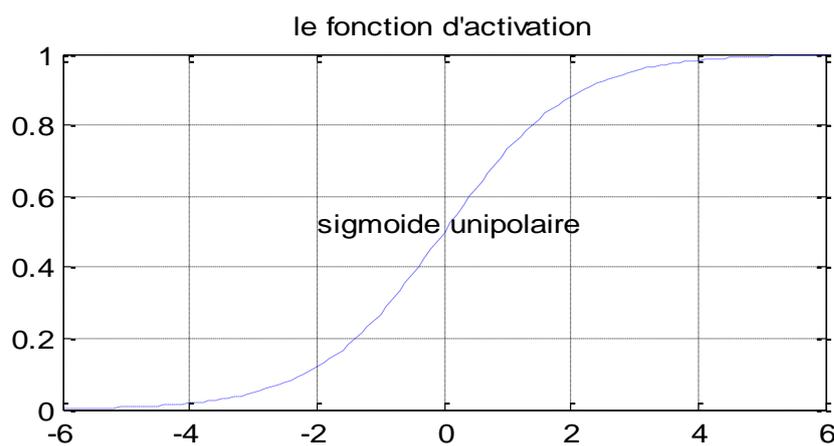


Figure 3.1 Fonction d'activation sigmoïde unipolaire

L'utilisation de cette fonction impose des conditions sur l'amplitude de signaux d'entrée et de sortie du réseau (une normalisation). Pour cela, on définit une sigmoïde dont on peut modifier la pente et les valeurs extrêmes.

Si l'on modifie seulement la pente, on considère la fonction

$$f(x) = f(\sigma x) = \frac{1}{1+e^{-\sigma x}} \quad (3.5)$$

Dont la dérivée est donnée par :

$$f'(x) = \sigma f(\sigma x)[1 - f(\sigma x)] \quad (3.6)$$

Si l'on veut, modifier les valeurs extrêmes, on définit la fonction

$$g(x) = \alpha f(\sigma x) - \beta \quad (3.7)$$

La dérivée, comme pour les autres fonctions sigmoïdes, possède l'expression de la fonction de transfert suivante :

$$g'(x) = \frac{\sigma}{\alpha} [\beta + g(x)][[\alpha - \beta + g(x)]] \quad (3.8)$$

3.3 Algorithme de rétropropagation

A chaque présentation d'un stimulus n°k, on réalise les 2étapes successive suivantes :

1. le passage du stimulus de l'entrée vers la sortie du réseau,
2. la rétropropagation de l'erreur dans le réseau pour la mise à jour des matrices de poids et biais afin de réduire l'erreur entre la sortie obtenue et la cible recherchée

- **Passage du stimulus à travers le réseau**

Chaque cellule de la couche cachée somme toutes ses entrées et fait passer le signal obtenu à travers sa fonction de transfert. De même pour les cellules de la couche sortie. Sous forme matricielle, le vecteur de sortie n°k a pour expression :

$$o_k = g[Z(f(wkx + w_0) + z_0)] \quad (3.9)$$

avec f et g, les fonctions de transfert, des couches cachée et de sortie qui peuvent être identiques. Le vecteur de sortie est comparé à celui que l'on désire obtenir, le vecteur t_k , pour déduire l'erreur.

- **Rétropropagation de l'erreur**

L'apprentissage du réseau consiste à modifier, à chaque pas d'apprentissage, les poids et les biais afin de minimiser la somme des carrés des erreurs en sortie. La méthode de rétropropagation est basée sur la technique du gradient.

La quantité à minimiser, à chaque pas d'apprentissage k, est la variance de l'erreur en sortie du réseau.

$$e_k = e_k^t e_k = \frac{1}{2} (t_k^t t_k + o_k^t o_k - 2o_k^t t_k) \quad (3.10)$$

$$\nabla E_{k/z} = \frac{1}{2} \nabla [o_k^t e_k - 2o_k^t t_k] / z \quad (3.11)$$

Si l'on considère des fonctions de transfert identiques pour les 2 couches ($f \equiv g$), la sortie du réseau est donnée par :

$$\nabla E_k / z = \frac{\partial E_k}{\partial z} = \frac{\partial E_k}{\partial o_k} \frac{\partial o_k}{\partial (zh_k + z_0)} \frac{\partial (zh_k + z_0)}{\partial z} \quad (3.12)$$

D'après l'expression de E_k , la première dérivée partielle est :

$$\frac{\partial E_k}{\partial o_k} = o_k - t_k \quad (3.13)$$

La deuxième dérivée partielle dépend du type de fonction de transfert utilisée. Dans le cas de la sigmoïde unipolaire, la deuxième dérivée partielle a pour expression :

$$\frac{\partial o_k}{\partial (zh_k + z_0)} = f(zh_k + z_0) * [I - f(zh_k + z_0)] = o_k * (I - o_k) \quad (3.14)$$

Avec I , représentant un vecteur unitaire de même taille que le vecteur de sortie o_k .

La troisième dérivée partielle est :

$$\frac{\partial (zh_k + z_0)}{\partial z} = h_k^t \quad (3.15)$$

La mise à jour se faisant dans le sens inverse du gradient, la matrice de poids Z de l'étape future ($k+1$) est donné par:

$$Z(k + 1) = Z(k) - \eta \nabla E_{K/Z} = Z(k) + \eta (t_k - o_k) * o_k * (1 - o_k) h_k^t = Z(k) + \partial s h_k^t \quad (3.16)$$

Avec η représente le gain d'apprentissage.

La même méthode sera utilisée pour la mise à jour du vecteur des biais z_0 . En considérant que le vecteur d'entrée passant par les biais est unitaire, on obtient facilement :

$$Z_0(k + 1) = Z_0(k) - \eta \nabla E_{K/Z_0} = Z_0(k) + \eta (t_k - o_k) * o_k * (1 - o_k) h_k^t = z_0(k) + \eta \partial s \quad (3.17)$$

L'erreur en sortie ∂s est rétropropagée à la sortie de la couche cachée à travers la transposée de la matrice Z

Par analogie à l'expression précédente de remise à jour de la matrice Z , avec h_k le vecteur de sortie de la couche cachée, on obtient :

$$w(k + 1) = w(k) - \eta \nabla E_{\frac{k}{w}} = w(k) + \eta Z(k)^t \partial s * h_k * (1 - h_k) x_k^t = w(k) + \eta \partial_h x^t \quad (3.18)$$

et

$$\begin{aligned} w_0(k+1) &= w_0(k) - \eta \nabla E_{k/w_0} = w_0(k) + \eta Z(k)^t \partial s \cdot h_k \cdot (1 - h_k) x_k^t \\ &= w_0(k) + \partial_h \end{aligned} \quad (3.19)$$

Dans notre programme, nous considérons les variables suivantes :

X : matrice de stimuli, dimensions (I, N) ou N représente le nombre de stimuli à présenter au réseau,

X_k : stimulus n°k (colonne n°k de la matrice X),

T : matrice de cibles ou réponses théoriques, dimensions (I, N),

T_k : cible n°k (colonne n°k de la matrice T),

H_k : sortie de la couche cachée, dimensions (L, I)

O_k : réponse du réseau, vecteur colonne à J composantes,

W : matrice de poids des connexions liant la couche d'entrée à la couche cachée, dimensions (L, I)

W_0 : vecteur de biais des cellules de la couche cachée, dimensions (L, I),

Z : matrice de poids des connexions liant la couche cachée à la couche de sortie, dimensions (J, I)

Z_0 : vecteur de biais des cellules de la couche de sortie, dimensions (J, I)

3.4 Commande par modèle inverse neuronal

Le modèle inverse neuronal d'un processus donné est un réseau de neurones qui permet de calculer l'entrée (ou le signal de commande) $u(t)$ de l'instant discret $t=KT$, à partir des valeurs antérieures de ses entrées-sorties.

L'étape de modélisation nécessite l'acquisition d'un fichier d'entrée-sortie du processus. Pour que les différents modes du processus soient excités, on applique un signal riche en fréquences : une séquence binaire pseudo-aléatoire autour du point de fonctionnement désiré.

Si l'on réalise N acquisitions, on obtient un fichier sous la forme d'une matrice à N lignes et 2 colonnes représentant respectivement l'entrée discrète $u(t)$ et la sortie $y(t)$.

Le modèle de transmittance

$$\frac{H(z)=0.065046z+0.0326}{z^2-1.0219z+0.11955} \quad (3.20)$$

Auquel on appliquera un signal de commande $u(t)$, formé d'une séquence binaire pseudo-aléatoire de hauteur 1 et longueur 1023 que l'on superpose à une valeur constante égale à 5 correspondant à un point de fonctionnement du processus.

A partir du fichier contenant $u(t)$ et $y(t)$, on calcule les variations $\Delta u(t) = u(t) - u(t-1)$ et $\Delta y(t) = y(t) - y(t-1)$.

Avec l'utilisation de la sigmoïde unipolaire, on doit normaliser les différents signaux $u(t)$, $y(t)$, $\Delta u(t)$ et $\Delta y(t)$ entre les valeurs limites 0.1 et 0.9.

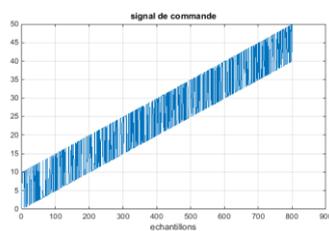


Figure 3.2 Signal de commande formé par la séquence binaire pseudo-aléatoire.

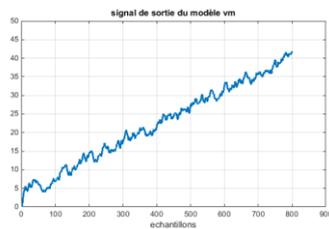


Figure3 .3 Signal de sortie du modèle : Tension Vm

D'après les figures 3.2 et 3.3, on voit bien que nous avons besoin de normaliser les signaux d'entrée et de sortie ainsi que leurs variations.

L'apprentissage du réseau, se fait par la méthode de rétropropagation de l'erreur où les poids et les biais sont initialisés à des valeurs aléatoires, de distribution uniforme entre -0.5 et 0.5, le gain d'adaptation est choisi égal à 0.8. Tout le fichier des entrées-sorties est présenté au réseau pendant un nombre d'itération égale à 1000

Nous avons choisi d'utiliser 5 cellules dans la couche cachée. L'évolution des poids et des biais à chaque itération de l'apprentissage est donnée par les courbes suivantes : figure 3.4 et 3.5

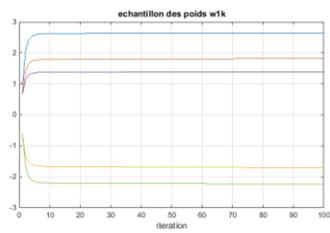


Figure3.4 Echantillons des poids

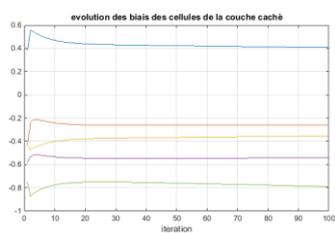


Figure3.5 Evolution des biais de la couche caché

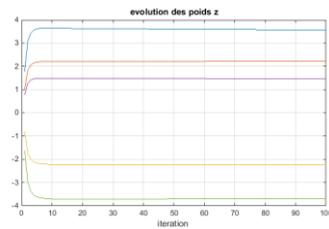


Figure 3.6 Evolution des poids Z de la couche cachée.

On peut clairement voir dans ces figure, qu'au bout d'une cinquantaine d'itération, on obtient une bonne convergence des poids et des biais. Malgré la valeur assez élevée du gain d'adaptation, les différents poids et biais ne varient plus au-delà de la 100^{ème} itération.

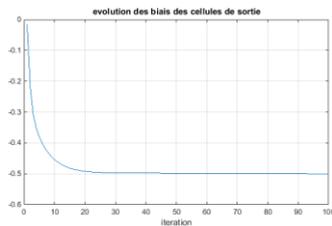


Figure 3.7 Evolution des biais des cellules de sortie

Pour vérifier la qualité de l'apprentissage, on utilise les mêmes signaux normalisés que l'on applique au réseau dont les poids et les biais sont figés aux valeurs finales.

La comparaison de la variation de commande normalisée en sortie du réseau et celle du modèle du système permet de montrer la qualité d'identification du modèle inverse

La courbe de la figure 3.8, représente l'erreur d'apprentissage :

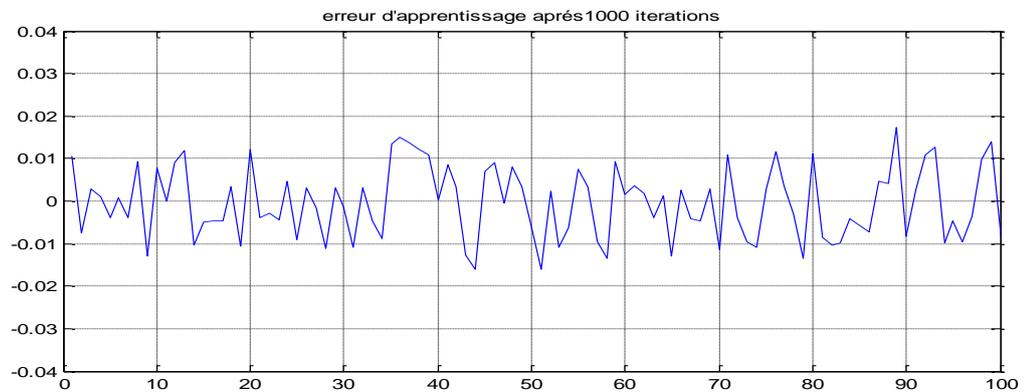


Figure 3.8 Erreur d'apprentissages après 1000 itérations

On peut voir que l'erreur maximale entre la sortie du réseau, représentant la variation de commande et celle du système, atteint un maximum de 0.018787, où la variance de l'erreur est de $7.0511e-005$.

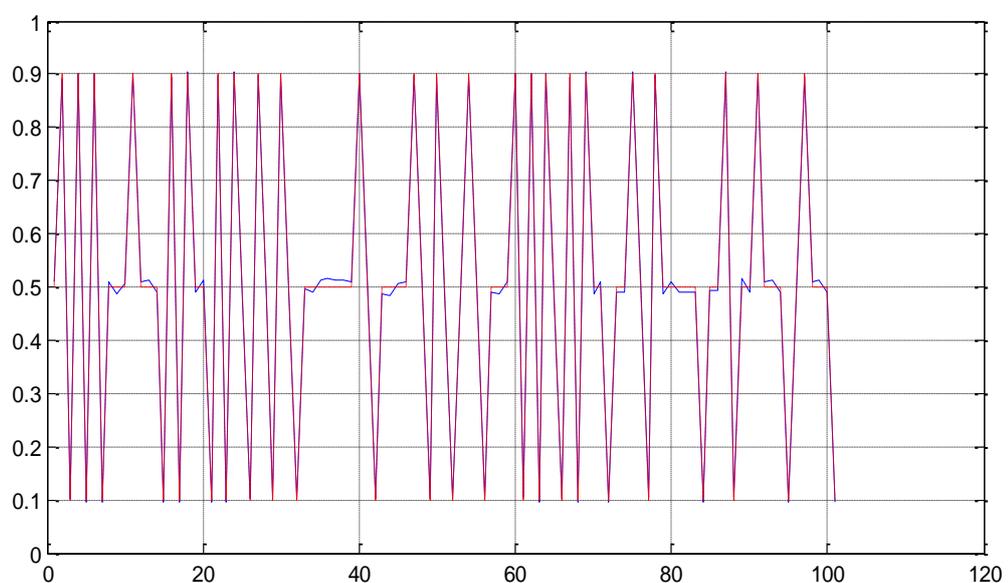


Figure 3.9 Réponse réelle et désirée du réseau après 300 itérations

Au bout des 300 itérations, on obtient une similitude quasi parfaite entre la réponse du réseau et le signal réel à apprendre

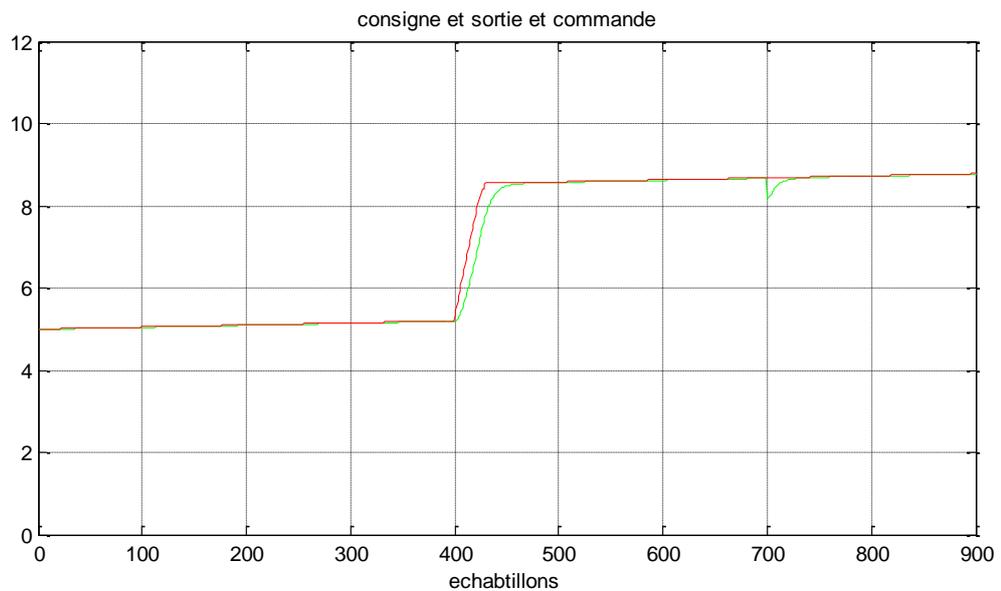


Figure 3.10 Commande neuronale par modèle inverse.

A l'instant discret $t=700$, on crée une perturbation sur la sortie du modèle. Comme pour la poursuite de l'échelon, cette perturbation est rejetée avec un léger déplacement.

Conclusion générale

En conclusion on peut dire que notre travail a été fructueux car il nous a permis d'aborder le domaine de l'identification ainsi que de la commande des systèmes en utilisant les techniques heuristiques, notamment les réseaux de neurones.

En effet, ce projet a été consacré à l'application des réseaux de neurones RN, dans le domaine de l'automatique de manière générale, et plus particulièrement pour l'identification du modèle inverse d'un processus modélisant un four électrique dans le but de l'utiliser pour la commande de la température de ce four. Ce qui a valu à ce type d'approche le nom de *commande neuronale par modèle inverse*.

Ceci étant, nous avons appuyé ce travail par des résultats de simulations sous l'environnement MATLAB, obtenus par l'application des RN sur le modèle d'un four électrique. Après le choix de la structure du réseau, des entrées ainsi que de l'algorithme d'optimisation, les simulations obtenus ont confirmé la capacité des réseaux de neurones à identifier le modèle inverse du four et à commander sa température.

Notons donc que notre objectif de se familiariser avec une approche de commande de système par les réseaux de neurones RN par modèle inverse est atteint, permettant de contourner les problèmes liés au calcul de l'inverse d'un modèle par les méthodes habituelles et de l'utiliser pour la commande du système choisi avec succès.

Références

- [1] **marc parzeau 2004** : » Réseau de neurone, université Lavel. »
- [2] **ali bouafia 2010** : » Introduction à la dynamique des sols, calcul dynamique des ouvrages géotechniques tome 1 tome 2 »
- [3] **Jean-François BOURGEOIS** et al, Fours électriques à résistances - Présentation générale, technique de l'ingénieur Février 2005
- [4] **Mohammed Salem , Djamel eddine Chaouch, M.faycel khelfi** –commande neuronale inverse des système non linéaire-Centre universitaire Mustapha Stambouli, Mascara ,Algérie et Université d'Oran Es-,Algérie .novembre 2007
- [5] **LANDAU I.D-** Identification et commande des systèmes-Automatique, 2ème édition .Ed.Hermés.(1993)
- [6] **semar benthami**-modélisation numérique d'un four électrique à convection forcée
- [7] **M.MARIE** application de **MATLAB** et **SIMULINK2**
- [8] **Goudjil kamel** –application d'un réseau de neurones pour analyser un problème d'interaction sol-structure
- [9] **lamine thiaw**-identification de systèmes dynamique non linéaire par réseaux de neurones et multimodèles –Automatique Université Paris val de Marne **2008.france.**
- [10] **B.Irie et s Miyake**
Capabilities of tree-layered perceptrons
Proceedings of the IEE second international conférence on neural networks
(San diego) 1 :641-647,1988.