

5.1 Introduction

Malgré la constante augmentation de la puissance des calculateurs, plusieurs opérations restent toujours difficiles à effectuer. Parmi ces opérations on trouve la modélisation des systèmes, le contrôle des processus, le filtrage ou la reconnaissance des signaux [26]. Les techniques des réseaux de neurones relèvent d'une telle approche ; comprendre les principes selon les quels les systèmes biologiques traitent l'information et s'en inspirer pour élaborer de nouvelles techniques en sciences de l'ingénieur.

Tout à commencé en 1943, lorsque *Mc Culloch* et *Pitts* ont inventé le premier "neurone artificiel", quelques années plus tard, en 1949, *Hebb* propose une formulation du mécanisme d'apprentissage, sous la forme d'une règle de modification des connexions synaptiques. Le premier réseau de neurones artificiel apparaît en 1958, grâce aux travaux de *Rosenblatt* qui conçoit le fameux perceptron. En 1982 Hopfield démontre tout l'intérêt d'utiliser des réseaux récurrents (dits "feed-back"), et forme l'algorithme de rétro propagation de l'erreur, qui donne un mécanisme d'apprentissage pour les réseaux multicouches de type perceptron.

Ce chapitre se compose de deux parties principales :

Dans la première partie, nous présenterons les principales notions de base relatives aux réseaux de neurones telles que leurs architectures, les fonctions d'activation et l'apprentissage....etc.

Dans la seconde partie, nous appliquons la commande par un seul réseau de neurones formels sur notre système étudié.

A la fin, nous tirerons la conclusion nécessaire.

5.2 Neurone biologique

Le neurone est une brique du cerveau. Il s'agit d'une cellule nerveuse, de forme et de caractéristiques variables cette cellule est composée d'un corps cellulaire contenant un noyau appelé soma voir figure 5.1. Le soma présente des ramifications donnant au neurone des allures de chevelure ou d'arbre : ce sont les dendrites (fines extensions tubulaires qui se ramifient autour du neurone).

Les dendrites et le soma constituent la principale surface de réception des signaux envoyés par les autres neurones. Les signaux reçus par les dendrites sont acheminés vers le soma où le traitement de l'information a lieu. Une fois traité, le signal résultant chemine, sous forme d'influx nerveux, le long d'une protubérance unique, appelée l'axone, pour être transmise aux autres neurones. L'axone et les dendrites forment les fibres nerveuses.

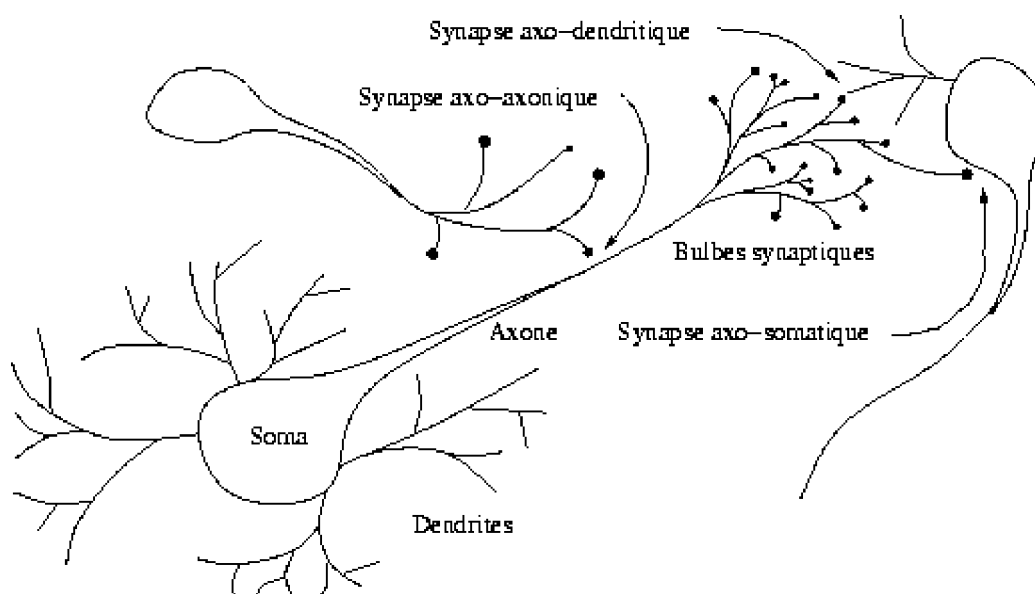


Fig.5.1 : Schéma du neurone biologique.

Chaque axone peut se ramifier en branches collatérales. Celles – ci se divisant à leur tour en petites fibres, plus fines que les dendrites, dont les terminaisons portent de minuscules boutons, appelés boutons synaptiques, le point de connexion entre deux neurones est appelé la synapse; c'est une jonction entre deux neurones, et généralement entre l'axone d'un neurone et un dendrite d'un autre neurone .

5.3 Neurone formel

Le premier neurone formel est apparu en 1943, on le doit à Mac Culloch et Pitts. Voici un schéma de leur modèle de neurone formel :

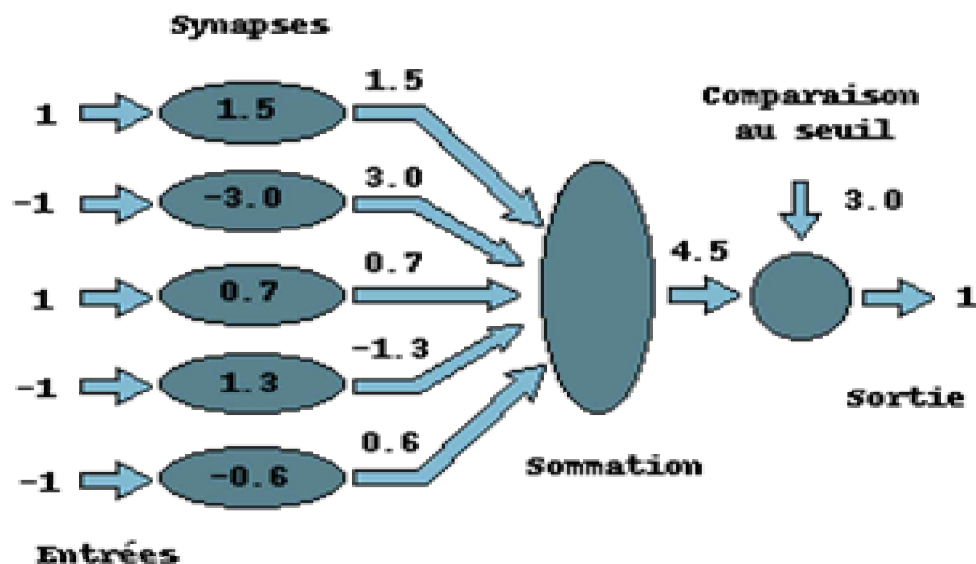


Fig.5.2 : Schéma du premier neurone formel.

Le neurone formel est donc une modélisation mathématique qui reprend les principes du fonctionnement du neurone biologique, en particulier la sommation des entrées. Sachant qu'au niveau biologique, les synapses n'ont pas toutes la même valeur (les connexions entre les neurones étant plus ou moins fortes), ils ont donc créé un algorithme qui pondère la somme de ses entrées par des poids synaptiques (coefficients de pondération). De plus, les 1 et les -1 en entrée sont là pour figurer une synapse excitatrice ou inhibitrice.

D'un point de vue mathématique, le neurone formel peut être représenté de la manière suivante :

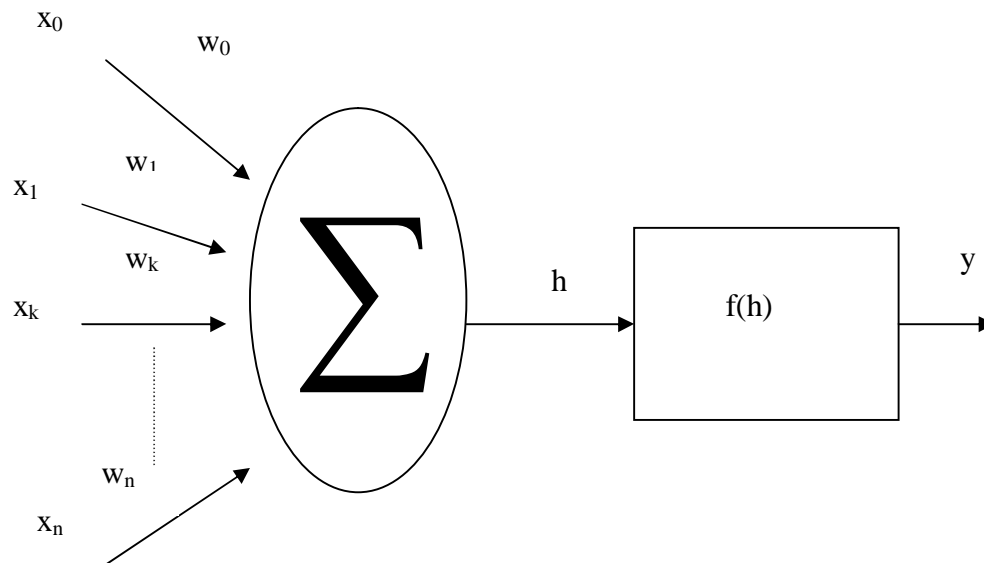


Fig.5 .3 : Schéma général d'un neurone formel.

Un premier neurone reçoit $n+1$ signaux d'entrée, avec $i=0, \dots, n$ pour élaborer l'activation h . Chaque signal d'entrée est pondéré par un coefficient (ou poids) w_i déterminé au cours de la phase d'apprentissage.

De sorte que l'activation est donnée par l'expression:

$$h = b + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n = b + \sum_{i=1}^n w_i \cdot x_i \quad (5.1)$$

Avec :

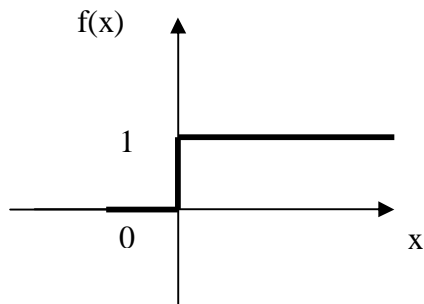
$b = w_0 \cdot x_0$: est appelé le biais de neurone.

Un second élément élabore le signal de sortie y , par application de la fonction d'activation f , de sorte que [22] :

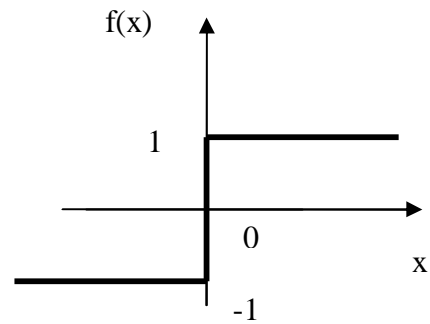
$$y = f(h) \quad (5.2)$$

Dans sa première version, le neurone formel était donc implémenté avec une fonction à seuil, mais de nombreuses versions existent. Ainsi le neurone de McCulloch et Pitts a été généralisé de différentes manières, en choisissant d'autres fonctions d'activations, comme les fonctions linéaires par morceaux, des sigmoïdes, ou des gaussiennes par exemples [9].

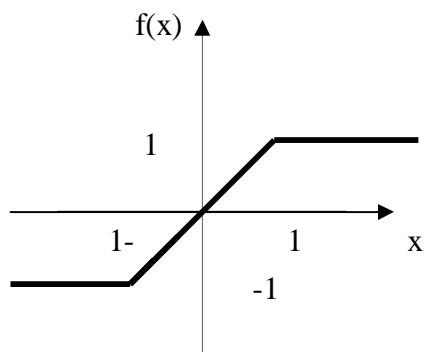
Voici les fonctions d'activation les plus utilisées :

**Fig. 5.4 :** Fonction de Heaviside.

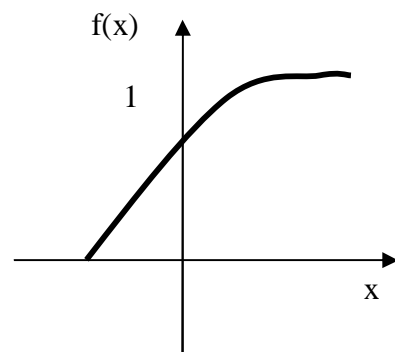
$$\begin{aligned} f(x) &= 0 & \text{si} & \quad x \leq 0 \\ f(x) &= 1 & \text{si} & \quad x > 0 \end{aligned}$$

**Fig.5.5 :** Fonction signe.

$$\begin{aligned} f(x) &= -1 & \text{si} & \quad x < 0 \\ f(x) &= 1 & \text{si} & \quad x > 0 \end{aligned}$$

**Fig. 5.6 :** Fonction linéaire bornée.

$$\begin{aligned} f(x) &= 1 & \text{si} & \quad x > 1 \\ f(x) &= x & \text{si} & \quad -1 \leq x \leq 1 \\ f(x) &= -1 & \text{si} & \quad x < -1 \end{aligned}$$

**Fig. 5.7 :** Fonction sigmoïde unipolaire.

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

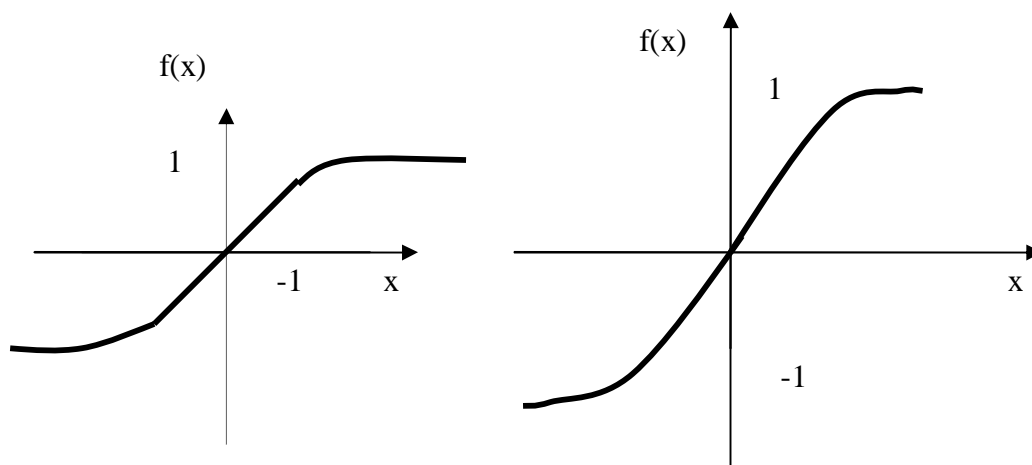


Fig. 5.8 : Fonction tangente hyperbolique. **Fig. 5.9 :** Fonction sigmoïde bipolaire.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

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

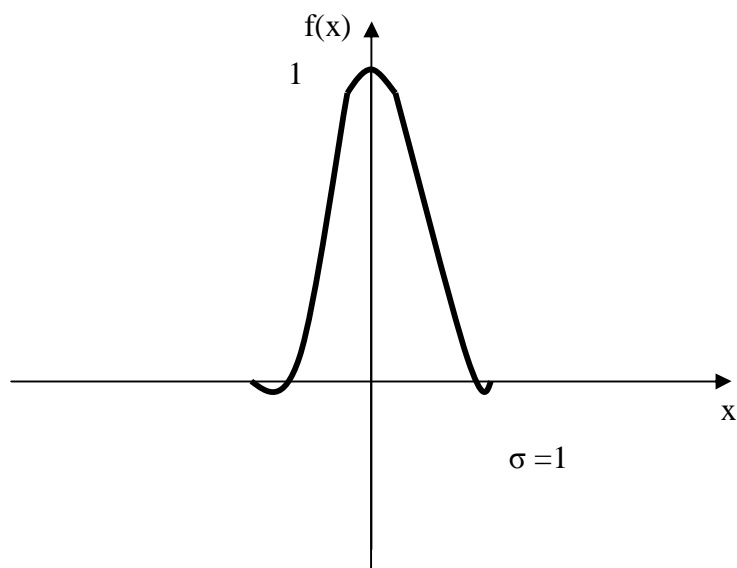


Fig. 5.10 : Fonction gaussienne.

$$f(x) = e^{-\frac{x^2}{2\sigma^2}}$$

5.4 Champs de neurones

Le neurone lui-même, en tant qu'unité autonome élémentaire n'a aucun pouvoir. La force et l'efficacité du cerveau résident en effet, dans le regroupement de ces neurones et le partage des tâches entre eux. Ce regroupement est appelé un champ de neurones où chaque élément reçoit et envoie de l'information. L'organisation de plusieurs champs entre eux, constitue un réseau de neurones, dont les unités à l'intérieur doivent travailler ensemble pour remplir une certaine tâche bien déterminée. Ces neurones sont reliés entre eux par des connexions. Les mécanismes de cette organisation déterminent l'architecture du réseau [20].

5.5 L'architecture des réseaux de neurones

C'est la manière dont les neurones sont interconnectés. Il existe deux types de réseaux de neurones :

- ▶ Les réseaux statiques (ou les réseaux non bouclés, appelés aussi "réseaux de type perceptron", "feed-forward").
- ▶ Les réseaux dynamiques (ou les réseaux bouclés, appelés aussi "réseaux récurrents", "feed-back") .

5.5.1 Les réseaux de neurones statiques :

Un réseau de neurones statique est un réseau dont la topologie de la connexion ne contient pas de boucle fermée, autrement dit dans lequel l'information se propage de couche en couche sans retour en arrière.

Les réseaux statiques les plus courants sont :

- ▶ Le réseau monocouche (perceptron monocouche);
- ▶ Le réseau multicouche (perceptron multicouches PMC).

5.5.1.1 Le réseau monocouche (perceptron monocouche)

C'est historiquement le premier réseau de neurones, c'est le perceptron de Rosenblatt. C'est un réseau simple, puisqu'il ne se compose que d'une couche d'entrée et d'une couche de sortie. Le perceptron monocouche est utilisé pour la connaissance des formes, pour faire de la classification et pour résoudre des opérations logiques simples. Sa principale limite est qu'il ne peut résoudre que des problèmes linéaires, il suit généralement un apprentissage supervisé selon la règle de correction de l'erreur.

La couche d'entrée formée de plusieurs neurones est reliée directement à la couche de sortie par des connexions modifiables figure 5.11.

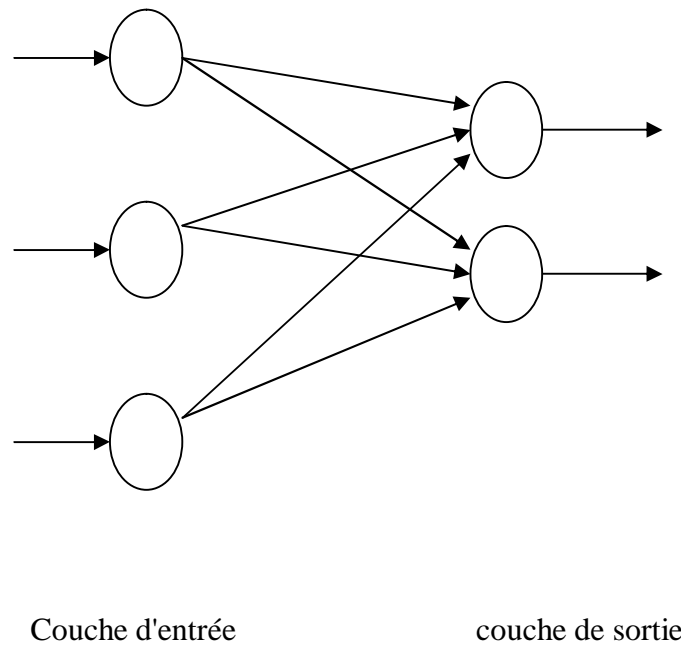


Fig.5.11 : Réseau monocouche

On appelle couche de m neurones à n entrées figure 5.12 un ensemble de m neurones possédant chacun $n+1$ entrées (n entrées + biais) [33].

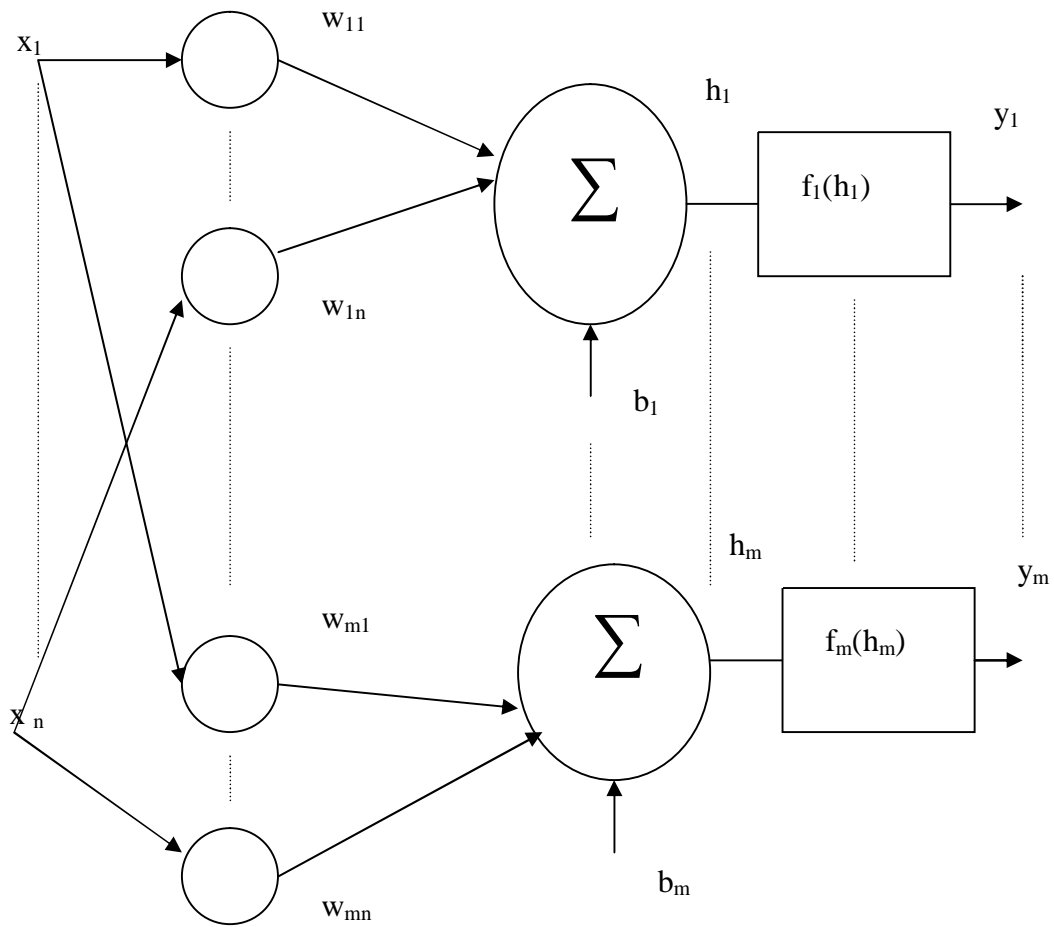


Fig.5.12 : Couche de neurones.

L'activation de la cellule n^oi de sortie est calculée comme suit :

$$a_i = \sum_{j=1}^n w_{ij} \cdot x_j \quad (5.3)$$

Le transfert entrée-sortie d'une couche peut s'exprimer comme suit [33]:

$$h_i = \sum_{j=1}^n w_{ij} \cdot x_j + b_i \quad (5.4)$$

$$i = 1, \dots, m$$

$$y_i = f_i(h_i) \quad (5.5)$$

Avec :

w_{ij} : représente le poids reliant l'entrée j au neurone i ,

x_j : cellule d'entrée ($j=1,2,\dots,n$),

b_i : biais du i ème neurone,

a_i : activation du i ème neurone.

La représentation vectorielle est donnée par la figure 5.13 :

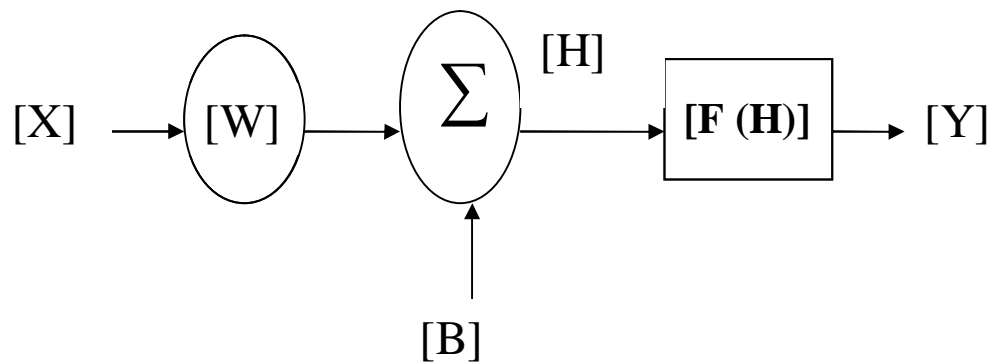


Fig.5.13 : Représentation vectorielle d'une couche neuronale.

Le comportement de la couche des neurones se décrit également de manière vectorielle :

$$[H] = [W] \cdot [X] + [B] \quad (5.6)$$

$$[Y] = [F(H)] \quad (5.7)$$

Avec :

$$[W] = \begin{bmatrix} w_{11} & w_{12} & \backslash & w_{1n} \\ w_{21} & w_{22} & \backslash & w_{2n} \\ \wedge & \wedge & \backslash & \wedge \\ w_{m1} & w_{m2} & \backslash & w_{mn} \end{bmatrix} : \text{Matrice des poids synaptiques,}$$

$$[X] = [x_1, x_2, \dots, x_n]^T : \text{Vecteur d'entrée,}$$

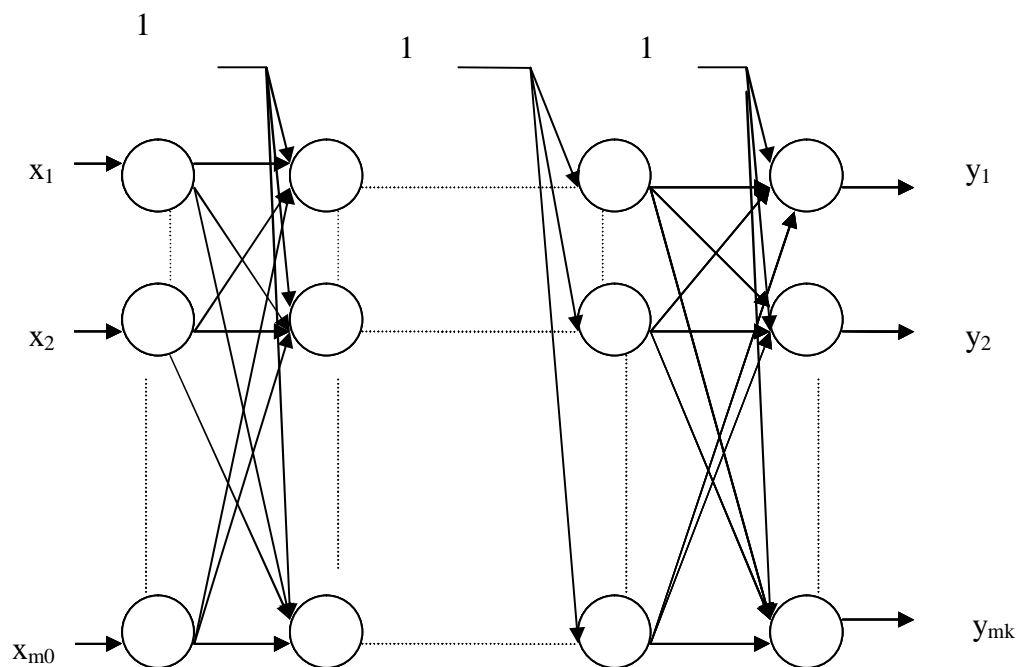
$$[Y] = [y_1, y_2, \dots, y_m]^T : \text{Vecteur de sortie,}$$

$$[H] = [h_1, h_2, \dots, h_m]^T : \text{Vecteur d'état,}$$

$$[B] = [b_1, b_2, \dots, b_m]^T : \text{Vecteur biais.}$$

5.5.1.2 Le réseau multicouche (perceptron multicouche)

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 d'entrée et de sortie, et il n'y a pas de connexions entre les cellules d'une même couche. Les fonctions d'activations 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. Il suit aussi un apprentissage supervisé selon la règle de correction de l'erreur, la figure 5.14 illustre de manière générique un réseau multicouches, qui est formé de k couches m_0 à m_k neurones respectivement [33].



Couche d'entrée Couche cachée n°1 Couche cachée n°(k-1) Couche de sortie
à m_0 neurones à m_1 neurones à m_{k-1} neurones à m_k neurones

Fig. 5.14 : Réseau multicouches statique.

Soit $[X]$ le vecteur d'entrée possédant m_0 composantes, les notations précédentes équations (5.4) et (5.5) complétées par les indices de couche q en exposant, permettent d'exprimer la relation de transfert entrée/ sortie du réseau comme suit:

$$h_i^q = \sum_{j=1}^{m_{q-1}} w_{ij}^q \cdot y_j^{q-1} + b_i \quad (5.8)$$

$$i = 1, \dots, m$$

$$q = 1, \dots, k$$

$$y_i^q = f_i^q(h_i^q) \quad (5.9)$$

Les équations 5.8 et 5.9 sont données sous forme vectorielle comme suit :

$$[H^q] = [W^q][Y^{q-1}] + [B^q] \quad (5.10)$$

$$[Y^q] = [F^q(H^q)] \quad q = 1, \dots, k \quad (5.11)$$

Avec :

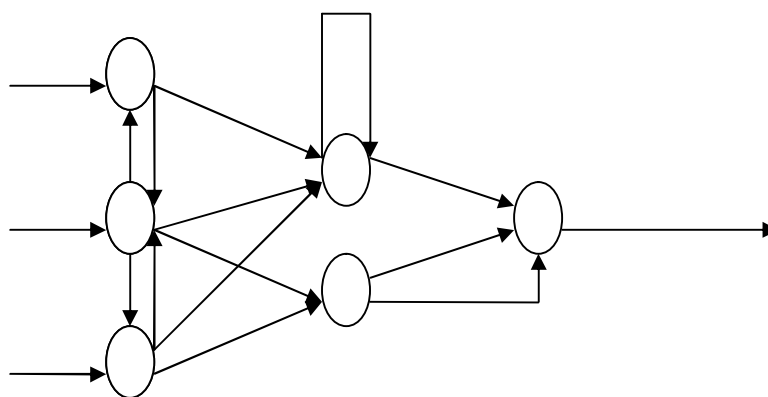
$[Y^0] = [X]$: Vecteur entrée,

$[Y^k] = [Y]$: Vecteur sortie [11] .

5.5.2 Les réseaux de neurones dynamiques (réseaux bouclés)

Les réseaux de neurones dynamiques ou bouclés, sont des réseaux dans les quels il y'a retour en arrière de l'information.

Chaque neurone peut être connecté à n'importe quel neurone du réseau et généralement à lui-même [19].



Couche d'entrée

Couche cachée

couche de sortie

Fig.5.15 : Représentation du réseau de neurones dynamique

5.6 L'apprentissage des réseaux de neurones

L'apprentissage est défini comme étant un changement opéré dans la mémoire du réseau. Cette modification affecte les poids synaptiques et les biais des neurones.

L'apprentissage a comme objectif l'amélioration des performances futures du réseau sur la base d'une connaissance acquise au fur et à mesure des expériences passées.

Le mécanisme d'apprentissage diffère suivant la tâche pour la quelle ce réseau est utilisé. Il existe principalement deux types d'apprentissage différents :

- L'apprentissage supervisé.
- L'apprentissage non supervisé [20].

5.6.1 L'apprentissage supervisé

Il se fait en présence d'un superviseur qui dirige le comportement du réseau en lui présentant les couples d'entrées et leurs sorties désirées. Donc l'apprentissage consiste ici, à comparer le résultat obtenu avec le résultat désiré, puis ajuster les poids des connexions pour minimiser la différence entre les deux, figure 5.16 [20].

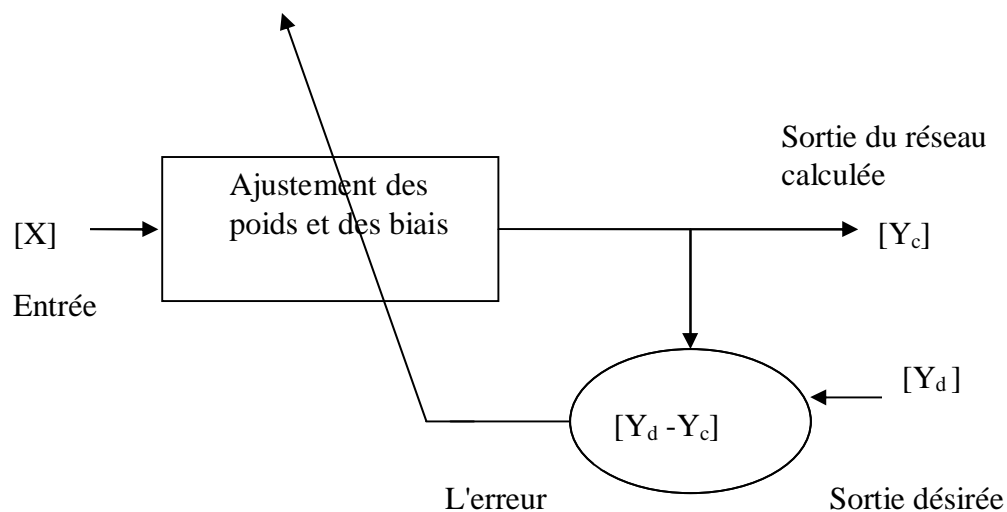


Fig.5.16 : Apprentissage supervisé

5.6.2 L'apprentissage non supervisé

L'apprentissage non supervisé nécessite la présence des entrées seulement sans l'intervention d'un superviseur. Cet entraînement se fait sur la base d'informations locales existant aux niveaux des neurones et découvre les propriétés collectives qui existent entre les données sur la base desquelles le réseau doit s'organiser pour produire une sortie désirée, figure5.17 [20].

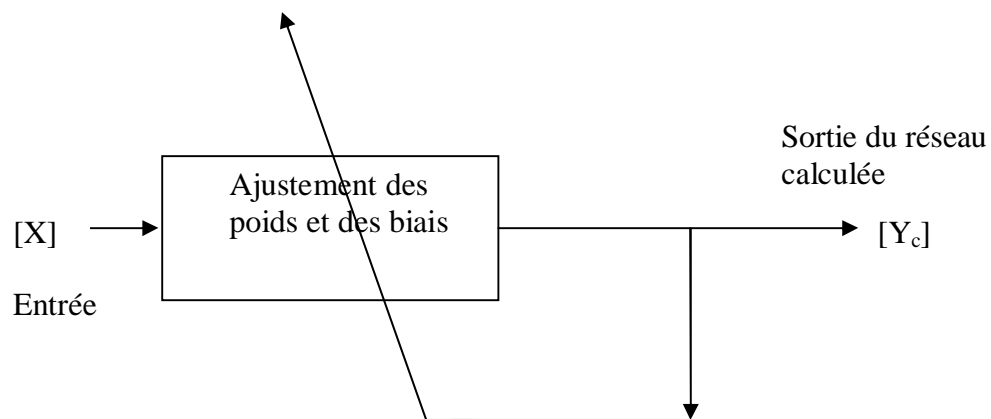


Fig.5.17 : Apprentissage non supervisé

5.6.3 Algorithme d'apprentissage

L'algorithme le plus utilisé pour l'apprentissage supervisé des réseaux de neurones multicouches est celui de rétro propagation (Back propagation). Cet algorithme change les poids et les biais d'un réseau à chaque itération, ce changement est réalisé de telle sorte de minimiser l'erreur entre la sortie désirée Y_d et celle calculée Y_c .

L'algorithme de rétro propagation se résume par les étapes suivantes :

Etape 1 : Initialisation aléatoire des poids et des biais.

Etape 2 : Appliquer le vecteur d'entrée de la couche d'entrée.

Etape 3 : Calculer le vecteur d'entrée de chaque couche cachée :

$$[H^q] = [W^q]. [Y^{q-1}] + [B^q], q=1,2, \dots, k-1.$$

Etape 4 : Calculer le vecteur de sortie de chaque couche cachée :

$$[Y^q] = [F^q (H^q)] \quad q=1, \dots, k-1.$$

Etape 5 : Calculer le vecteur d'entrée de la couche de sortie :

$$[H^k] = [W^k]. [Y^{k-1}] + [B^k].$$

Etape 6 : Calculer le vecteur de sortie du réseau :

$$[Y_c] = [Y^k] = [F^k (H^k)].$$

Etape 7 : Ajuster les poids et les biais de la couche de sortie :

$$[W^k] = [W^k] + \eta [\delta^k] [Y^{k-1}]$$

$$[B^k] = [B^k] + \eta [\delta^k]$$

Avec : η pas d'apprentissage

$$[\delta^k] = [Y_d - Y_c]. [F'^k (H^k)].$$

F' : Représente la dérivée de la fonction d'activation vectorielle.

Etape 8 : Modifier les poids et les biais de chaque couche cachée :

$$[W^q] = [W^q] + \eta [\delta^q] [Y^{q-1}]$$

$$[B^q] = [B^q] + \eta [\delta^q]$$

Avec :

$$[\delta^q] = [W^q]. [\delta^{q+1}]. [F'^q (H^q)]$$

Etape 9 : Calculer la sortie du réseau en utilisant les nouveaux poids.

L'organigramme de l'apprentissage pour les réseaux multicouches est présenté par la figure 5.18 [33].

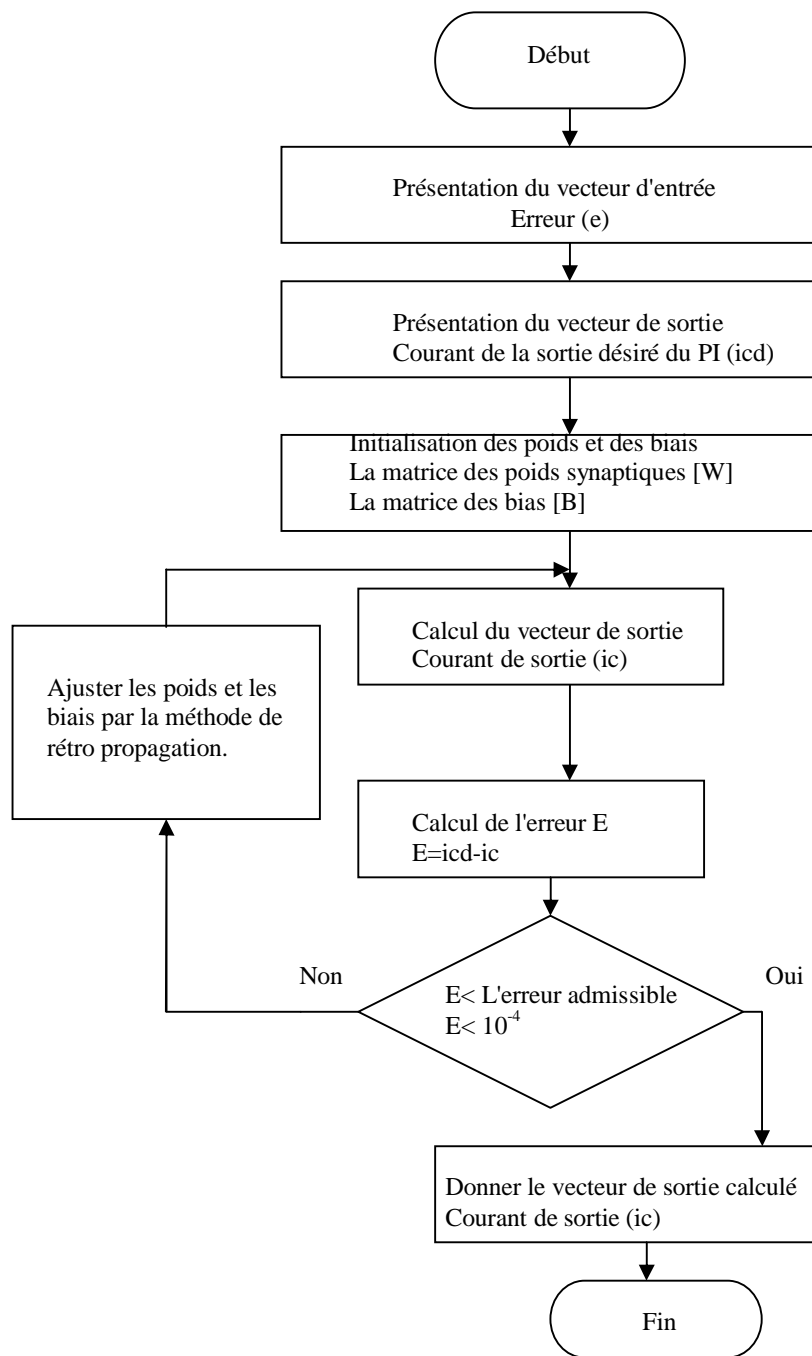


Fig.5.18 : Organigramme de l'apprentissage pour les réseaux multicouches.

5.7 Application de la commande supervisée sur la table à induction

Les réseaux de neurones ont été fortement sollicités ces dernières années pour la commande des systèmes non linéaires. Cette seconde partie consiste à utiliser les caractéristiques des réseaux de neurones pour identifier le comportement du système (PI), et le remplacer par un seul réseau de neurone formel.

5.7.1 Choix de la structure du réseau de neurones

Chaque réseau de neurones remplir une fonction bien définie dépendant de l'architecture choisie (nombre de couches cachées, le nombre de neurones dans chaque couche cachée). Le problème est de trouver celle qui donne de meilleur résultats. Pour cela, plusieurs essais doivent être effectués afin de déterminer l'architecture optimale de réseau.

► L'entrée du réseau est :

e : l'erreur entre le courant de l'inducteur i_m et celle de la consigne i_{ref} .

► La sortie désirée de commande est :

icd : courant sortant du régulateur PI.

► La sortie de commande calculée par le réseau est :

ic : courant sortant du régulateur.

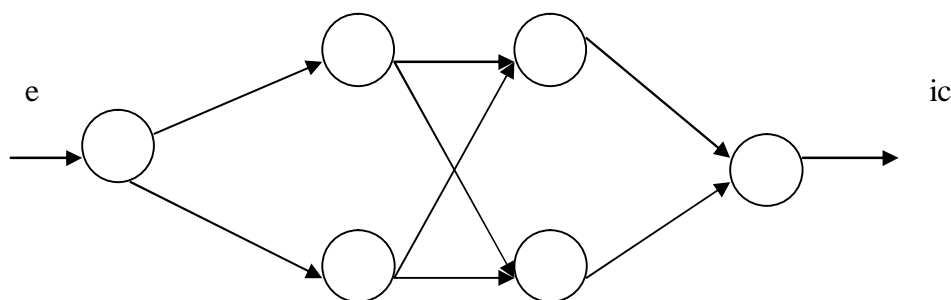


Fig.5.19 : Exemple de réseau de neurones : structure [1-2-2-1].

Dans tous les essais, l'apprentissage se déroulera sous les conditions initiale nulles. L'apprentissage est obtenu en présentant plusieurs exemples au réseau avec une erreur admissible de 10^{-4} et le nombre d'itérations maximale égal à 10^4 .

Le tableau 5.1 présente les résultats d'apprentissage effectués sur les différentes structures du réseau de neurones.

Essai	Structure	Nombre d'itération	Erreur
1	1-1-1	10^3	5.62×10^{-5}
2	1-5-1	10^3	5.30×10^{-5}
3	1-10-1	10^3	5.06×10^{-5}
4	1-2-2-1	745	4.50×10^{-5}
5	1-2-1	10^4	5.40×10^{-5}
6	1-1-1-1	10^4	4.80×10^{-5}

Tableau.5.1 : Résultats d'apprentissages pour différentes structures

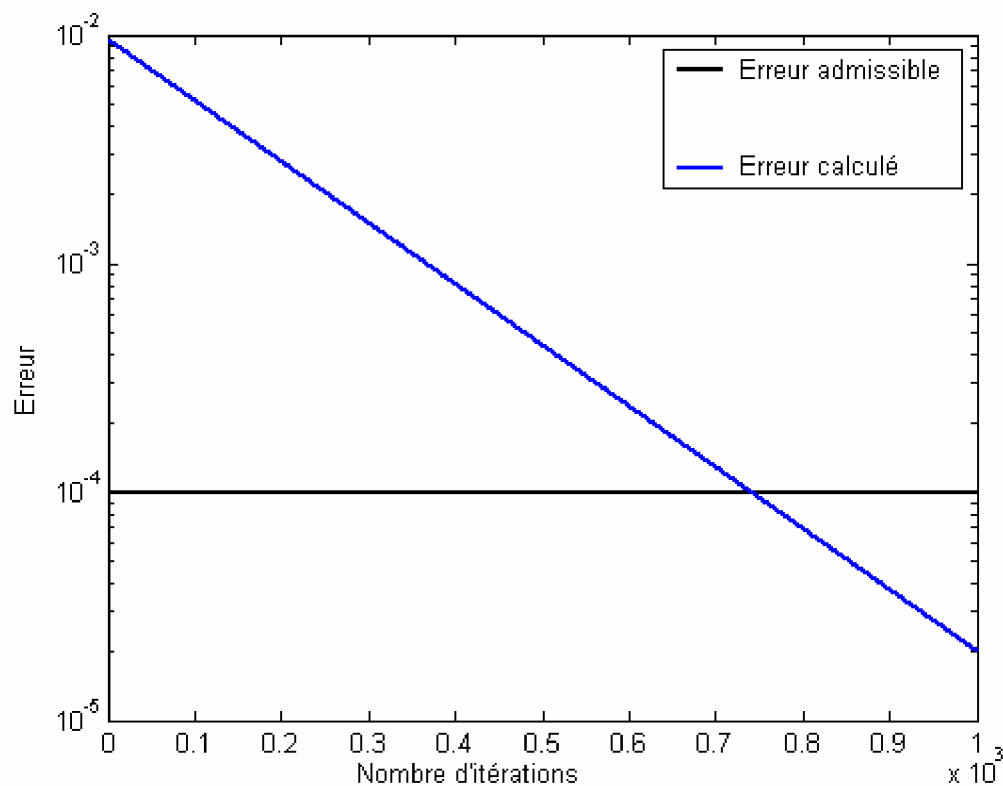


Fig.5.20 : Evolution de l'erreur lors de l'apprentissage

D'après les essais effectués sur les différentes structures , nous avons opté pour un réseau de neurones complètement connecté, à un seul neurone dans la couche d'entrée et un neurone dans la couche de sortie ,avec deux couches cachées de deux neurones chacune (essai n° 4).

5.7.2 Simulation et commentaire

L'application de résultat d'apprentissage supervisé avec la structure [1-2-2-1] permet d'obtenir des bonnes performances sur le courant de l'inducteur, qui a une forme très proche de la sinusoïde et ne présente pas d'ondulations et d'oscillations sur l'amplitude, et la valeur efficace du courant est constante figures 5.21, 5.22, 5.23.

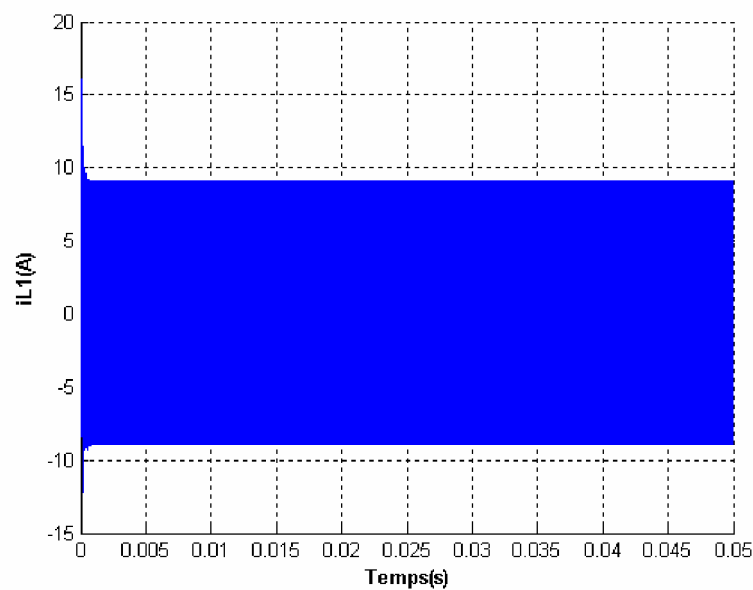


Fig.5.21 : Courant contrôlé dans l'inducteur avec un temps de 0.05 s.

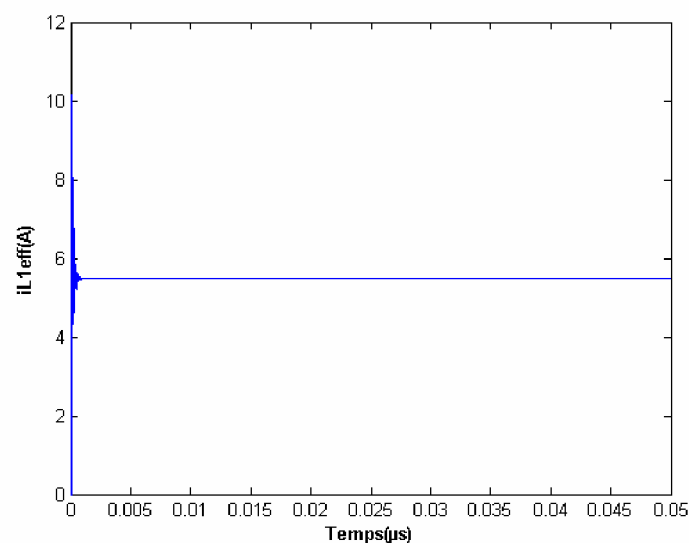


Fig.5.22 : La valeur efficace du courant contrôlé dans l'inducteur.

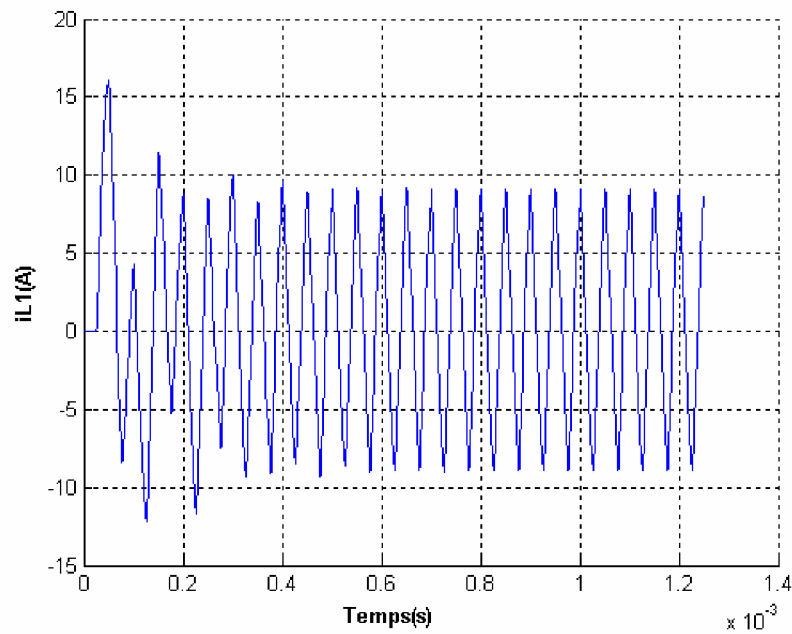


Fig.5.23 : Agrandissement du courant contrôlé dans l'inducteur.

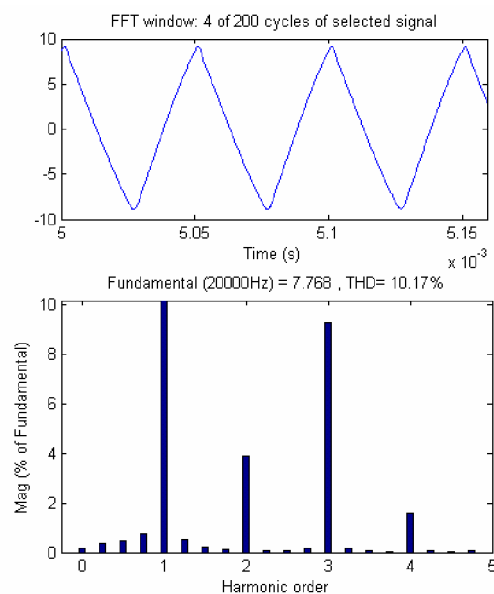


Fig.5.24 : Spectres d'harmoniques.

La qualité de forme d'onde (THD=10.17%) est légèrement améliorée par rapport à celle du régulateur classique.

5.8 Test de robustesse

La robustesse d'une commande est sa capacité à surmonter l'incertitude sur le modèle à contrôler. Les performances des correcteurs ne doivent pas être affectées par les variations de paramètres ni par les perturbations.

Nous proposons dans cette partie l'étude de la robustesse de la commande par réseaux de neurones vis-à-vis des perturbations affectant le courant inducteur, qui revient à l'utilisation de plusieurs foyers placés à proximité dans une table à induction.

5.8.1 Test de robustesse vis-à-vis d'une perturbation affectant le courant inducteur

Pour le test de robustesse, on est arrivé jusqu'à 50% sans que le système ne soit dystabilisé figure 5.25.

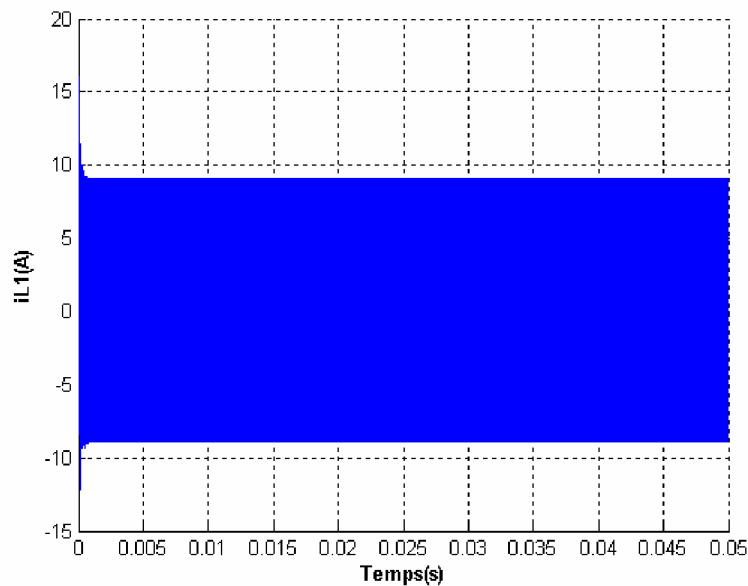


Fig.5.25 : Test de robustesse vis-à-vis d'une perturbation
(de 50% à $t = 0.025$ s)

5.8.2 Interprétation

Le graphe de la figure 5.25 illustre le courant inducteur dans la table à induction commandé par réseau de neurones, avec présence d'une perturbation de 50% à $t = 0.025$ s.

Nous constatons que une perturbation de 10% jusqu'à 50% n'influe pas sur les performances de la table à induction, la grandeur à commander suit leur valeur de consigne. Donc, cette commande est robuste.

5.9 Comparaison entre les deux commandes

L'analyse des deux réponses du système figure 5.26 nous montre, qu'il y a une petite amélioration en qualité de forme d'onde par la régulation neuronale (le THD=10.19% pour la régulation classique et THD=10.17% pour la régulation par

réseaux de neurones). Aussi cette stratégie de contrôle est très robuste (comparée avec celle du régulateur PI).

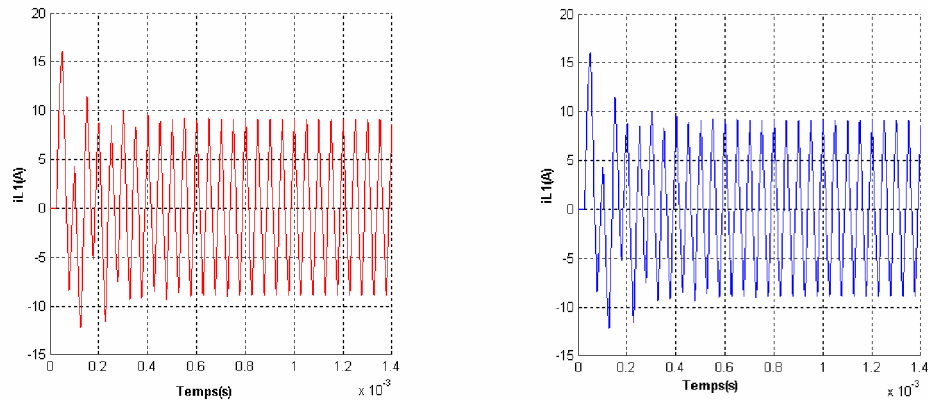


Fig.5.26: Comparaison entre les deux réponses du système obtenues respectivement avec la commande classique et la commande par réseaux de neurones.

5.10 Conclusion

Après une étude sur les concepts fondamentaux sur les réseaux de neurones, nous avons utilisé une des stratégies de commande neurales (commande supervisée).

On a remplacé le bloc PI pour la régulation de la table à induction par un contrôleur neuronal, le bon choix était de prendre une structure du réseau de neurones à deux couches cachées contenant deux neurones. Les résultats obtenus sont améliorés de vu performances dynamiques et statiques à ceux apportés par la régulation classique. La régulation est devenue plus simple et plus souple avec les réseaux de neurones.