

Chapitre Quatre

SYSTEME HYBRIDE - ANFIS -

4.1. Introduction	81
4.2. Définition	81
4.3. Quelques types de combinaison Neuro-Floues	82
4.4 Le Modèle ANFIS	84
4.5 Conclusion	91

SYSTEME HYBRIDE -ANFIS-

4.1. Introduction

Les systèmes Neuro-Flous permettent de combiner les avantages de deux techniques complémentaires. Les systèmes flous fournissent une bonne représentation des connaissances. L'intégration de réseaux de neurones au sein de ces systèmes améliore leurs performances grâce à la capacité d'apprentissage de réseaux de neurones. Inversement, l'injection des règles floues dans les réseaux de neurones, souvent critiques pour leur manque de lisibilité, clarifie la signification des paramètres du réseau et facilite leur initialisation, ce qui représente un gain de temps de calcul considérable pour leur identification.

De nombreux types de systèmes Neuro-flous ont été définis et développés ces dernières années, loin d'être uniformisés, et parfois équivoques et confus. Afin de clarifier les définitions, nous proposons dans ce chapitre une présentation brève de quelques types de systèmes Neuro-flous et une présentation plus détaillée de L'ANFIS.

4.2. Définition

4.2.1 Réseaux de neurones flous

Le système Neuro-Flous se reporte à la manière d'appliquer de diverses techniques d'apprentissage développées à la littérature de réseaux de neurones aux systèmes d'inférence floue.

4.2.2. Les réseaux adaptatifs

Comme son nom implique, un réseau adaptatif est une structure de réseau dont le comportement global d'entrée-sortie est déterminé par les valeurs de la collection de paramètres modifiables. Plus spécifiquement, la configuration d'un réseau adaptatif se compose d'un ensemble de nœuds reliés par des liens orientés, où chaque nœud est un processus unitaire qui remplit une fonction statique sur ses signaux d'entrées pour générer un signal unique à la sortie de nœud et chaque nœud spécifie la direction de l'écoulement de signal d'un nœud à l'autre. Habituellement une fonction de nœud est paramétrée avec des paramètres modifiables; en changeant la fonction de nœud en plus de le comportement global du réseau additif. [25]

Dans le cas le plus générale, un réseau adaptatif est hétérogène et chaque nœud peut avoir une fonction différente des autres nœuds. Également il faut se rappeler que chaque lien

dans un réseau adaptatif sont juste employés pour indiquer la direction de propagation de signal de sortie d'un nœud : généralement il n'y a aucun poids ou paramètre lié aux liens.

4.3. Quelques types de combinaison Neuro-Floues

Il y a plusieurs types pour combiner les réseaux de neurone et les systèmes flous. Ces types peuvent être classés en fonctionnelle et structurelle, suivant leur architecture et la configuration recherche entre le système d'inférence flou et les réseaux de neurones.

4.3.1. Systèmes neuro-flou coopératifs et concurrents

Un système Neuro-flou coopératif peut être considéré comme préprocesseur où le mécanisme d'apprentissage de réseaux de neurones artificiels (RNA) détermine les fonctions d'appartenance de Système d'inférence flou (SIF) ou les règles floues à partir données d'apprentissage. Une fois que les paramètres de SIF sont déterminés, RNA va au fond. La règle basée est habituellement déterminée par un algorithme clustering flou. Les fonctions d'appartenance sont habituellement approximer à partir RNA par les données d'apprentissage. Dans un système Neuro-flou concurrent, RNA aide le SIF continûment pour déterminer les paramètres exiger particulièrement si les variables d'entrée du contrôleur ne peuvent pas être mesurées directement. Dans certains cas les sorties de SIF ne pourraient pas être directement applicables au processus. Les figures 3.1 et 3.2 représentent les modèles Neuro-Flou coopératifs et concurrents. [26]

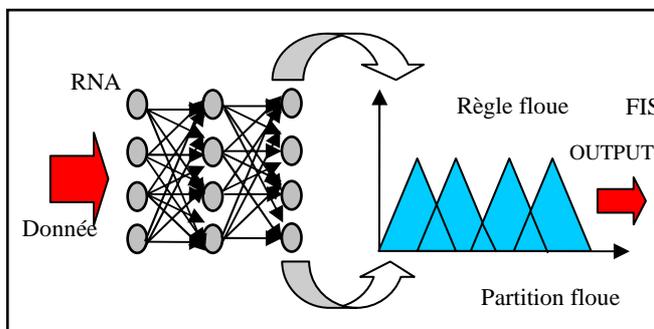


Figure 4.1- Système neuro-flou Coopératif

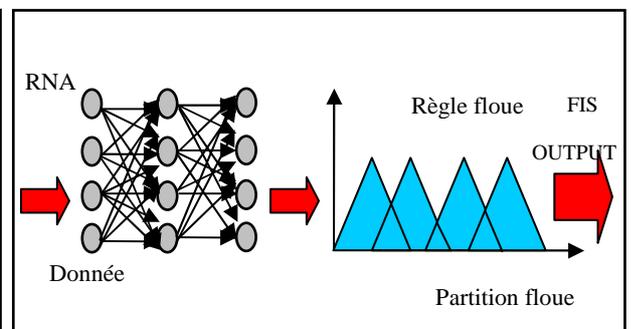


Figure 4.2- Système neuro-flou concurrent

4.3.2. Les systèmes neuro- flou fondus

Dans une architecture Neuro-Flou fondue, les RNA sont utilisées pour déterminer les paramètres de SIF. Les systèmes Neuro-Flou fondus partagent les structures de données et la représentation de connaissance. Une manière habituel d'appliquer un algorithme d'apprentissage à un système flou est à représentez-la dans une architecture spéciale

De quelque manière que l'algorithme d'apprentissage de l'RNA ne peut pas être appliqués directement à un système d'inférence comme une fonction parce que les fonctions utilisées dans le système d'inférence sont habituellement non différentiables. Ce problème peut être abordé en employant des fonctions différentiables dans le système d'inférence ou près ne pas utiliser l'algorithme d'apprentissage de RNA standard. Certains types de systèmes Neuro-Flou principaux GARIC, FAUCON, ANFIS, NEFCON AMUSEMENT, SONFIN, et beaucoup d'autres. [26]

4.3.3. Falcon (Fuzzy Adaptive Learning Control Network)

FALCON à une architecture cinq couches, comme il est représenté dans la figure 3.3. Il y a deux neurones pour chaque variable de sortie. Une pour les données d'apprentissage (sortie désirée) et l'autre est pour la sortie de FALCON. La première couche cachée sert à fuzzifier les variables d'entrées. Chaque neurone dans cette couche représentant une fonction d'appartenance à un ensemble flou. La deuxième couche cachée définit les parties antécédentes des règles floues suivie par les parties conséquences des règles dans la troisième couche cachée. FALCON emploie un algorithme d'apprentissage hybride comportant l'apprentissage non supervisée pour localiser des fonctions d'appartenance et base des règles initiale et l'apprentissage supervisé pour optimiser l'ajustement des paramètres du MF pour générer les sorties désirées.

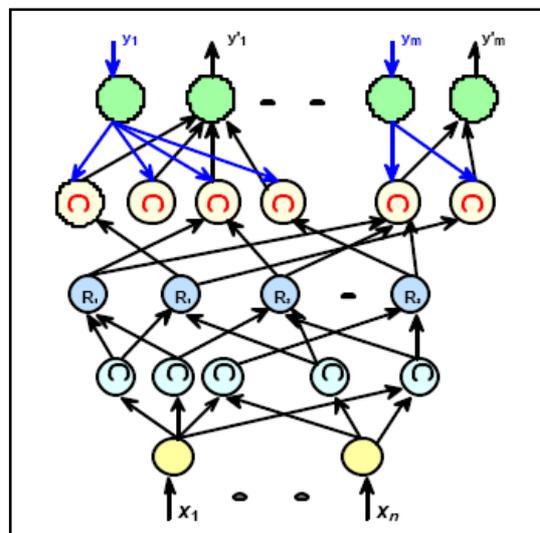


Figure 4.3- architecture de FALCON

4.3.4. Le NEFCON (NEuro-Fuzzy CONtrol)

NEFCON est conçu pour mettre en application le système d'inférence flou type Mandani. Il est constitué de 2 couches dont les poids sont les ensembles flous et les règles floues. Avec la même utilisation antécédente supposée ont partagé les poids, qui sont

représentés par des ellipses dessinées autour des raccordements. Elles assurent l'intégrité de la base de règle. La couche d'entrée assure la tâche de l'interface de fuzzification, la logique d'inférence est représentée par les fonctions de propagation, et la couche de sortie est l'interface de défuzzification. L'apprentissage du modèle de NEFCON est basé sur un mélange de l'apprentissage non supervisé et supervisée (rétropropagation). NEFCON peut être employé pour apprendre des règles initiales, si aucune connaissance du système n'est disponible ou même pour optimiser une base manuellement définie de règle. [26]

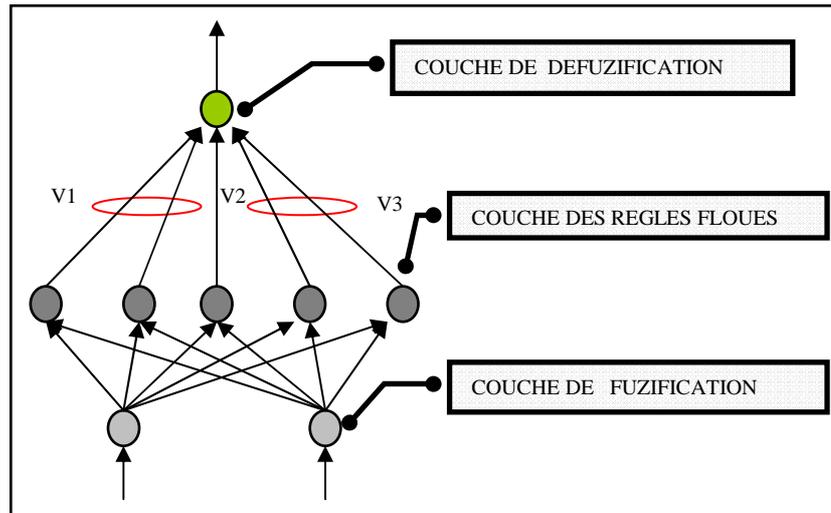


Figure 4.4- L'architecture de NEFCON

4.4. Le Modèle ANFIS

4.4.1. architecture de l'ANFIS

ANFIS (*Adaptive Network Based Fuzzy Inference System*) c'est un système d'inférence adaptatif neuro-flou qui consiste à utiliser un réseau neurone de type MLP à 5 couches pour lequel chaque couche correspond à la réalisation d'une étape d'un système d'inférence floue de type Takagi Sugeno. Pour la simplicité, nous supposons que le système d'inférence flou à deux entrées x et y , et à comme une sortie f . Supposer que la base de règle contient deux règles floues de type Takagi-Sugeno.

$$\text{Règle1 : SI } x \text{ est } A_1 \text{ et } y \text{ est } B_1 \text{ ALORS } f_1 = p_1 x + q_1 y + r_1 \quad (4.1)$$

$$\text{Règle2 : SI } x \text{ est } A_2 \text{ et } y \text{ est } B_2 \text{ ALORS } f_2 = p_2 x + q_2 y + r_2 \quad (4.2)$$

L'ANFIS à une architecture posée par cinq couches comme représenté sur la figure 4.5.

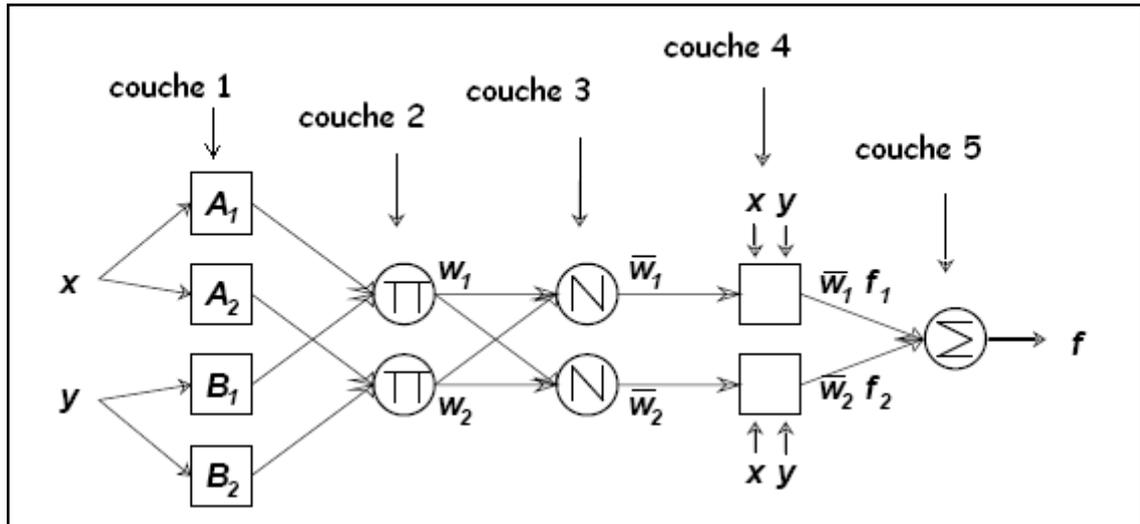


Figure 4.5- L'Architecture de l' ANFIS.

Une architecture classique peut être décrite de la manière suivante :

1. La première couche d'une architecture de type ANFIS comporte autant de neurones qu'il y'a de sous ensembles flous dans le système d'inférence représenté. Chaque neurone calcule le degré de vérité d'un sous ensemble flou particulier par sa fonction de transfert. La seule restriction sur le choix de cette fonction concerne sa dérivabilité. En retrouve dans la littérature, l'utilisation, de fonctions gaussiennes et les paramètres modifiables sont le centre et la pente de la gaussienne (variance).

La fonction d'activation des neurones i de la première couche :

$$f_i^1 = \mu_{A_i}(x) \quad (4.3)$$

Tel que x est l'entrée au neurone i , et A_i est un sous ensemble flou correspondant au variable x . En d'autres termes, f_i^1 est la fonction d'appartenance du A_i et il indique le degré auquel donné x satisfait le quantifier A_i . Nous choisissons $\mu_{A_i}(x)$ pour être en forme de (Gaussien, triangle, trapézoïdal) avec le maximum égal à 1 et le minimum égal à 0, tel que les fonctions généralisée de ces formes est :

$$\text{Triangle :} \quad \mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (4.4)$$

$$\text{Trapézoïdale :} \quad \mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (4.5)$$

$$\text{Gaussienne :} \quad \mu(x) = \exp\left(-\frac{(x-c)^2}{\sigma^2}\right) \quad (4.6)$$

Où $\{a, b, c, \sigma\}$ est l'ensemble des paramètres. Pendant que les valeurs de ces paramètres changent, les fonctions en forme précédente changent en conséquence, de ce fait présenter de diverses formes de fonction d'appartenance sur la variable linguistique A_i . Les paramètres dans cette couche désigner sous le nom des paramètres de fonction d'appartenance [25].

2. La deuxième couche cachée sert à calculer le degré d'activation des prémisses. Les neurones de la cet couche représentent chacun la prémisse d'une règle [25]. Ils reçoivent en entrée le degré de vérité des différents sous-ensembles flous composant cette prémisse et ont en charge le calcul de son propre degré de vérité. Les fonctions d'activation utilisées pour ces neurones dépendant des opérateurs présents dans les règles (ET ou OU).

La fonction d'activation des neurones i de la première couche :

$$W_k = \mu_{A_i}(x) \times \mu_{B_j}(y) \quad (4.7)$$

Où k : représente le nombre de règle, i : représente le nombre de partition de x , et j : le nombre de partition de y .

3. La troisième couche cachée normalise de degré d'activation des règles.

Chaque neurone dans cette couche est un neurone de cercle noté N . le $i^{ème}$ neurone calcule le rapport entre $i^{ème}$ poids de règles et la somme de toutes les poids des règles. Cette opération est appelée **la normalisation des poids**.

$$\overline{W}_k = \frac{W_k}{\sum W_i} \quad (4.8)$$

L'ensemble des sorties de cette couche seront appelées les poids normalisé.

4. La quatrième couche cachée sert à déterminer les paramètres la partie conséquence des règles (p, q, r). La fonction de chaque neurone dans cette couche est la suivante

$$f_k^4 = \overline{W}_k \times f_k = \overline{W}_k \times (p_k x + q_k y + r_k) \quad (4.9)$$

Où \overline{W}_k est la sortie de la troisième couche, et $\{p_i, q_i, r_i\}$ est l'ensemble des paramètres. Ces paramètres désigner sous le nom les paramètres conséquents. [25]

5. La couche de sortie contient un seul neurone dans cette couche, est un neurone de cercle noté S qui calcule la sortie globale comme addition de tous les signaux entrants, c'est-à-dire :

$$f^5 = \sum_k \overline{W}_k \times f_k^4 \quad (4.10)$$

La figure 4.6 représente un système ANFIS, à 2 entrées chaque entrée repartie en trois sous ensemble floue et 9 règles.

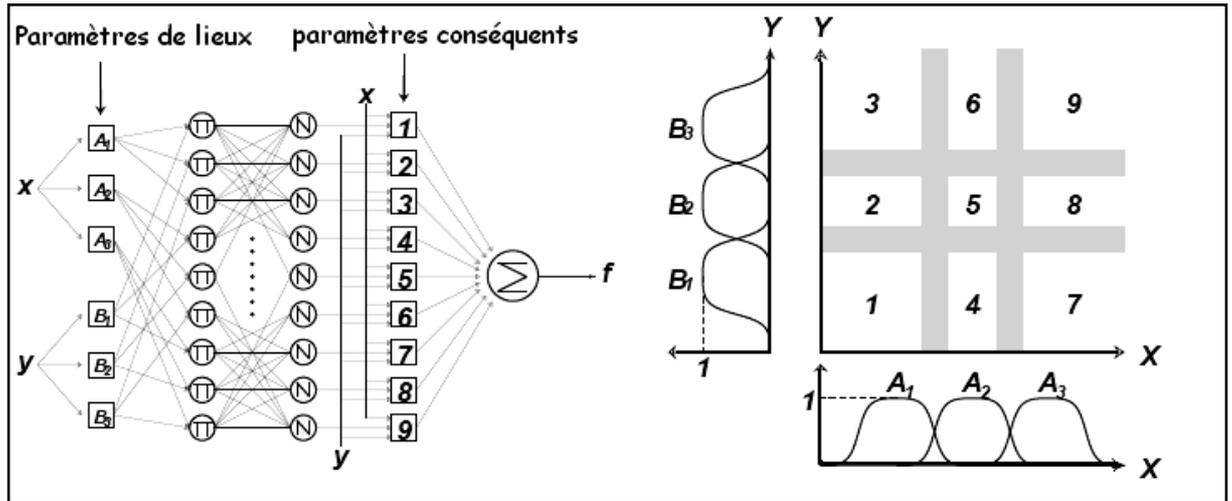


Figure 4.6- Exemple ANFIS à 2 entées avec 9 règles.

Les différentes couches	Type des couches	Le nombre de neurone dans la couche
Couche 0	Les entrées	n
Couche 1	Les valeurs	(p.n)
Couche 2	Les règles	p^n
Couche 3	La normalisation	p^n
Couche 4	Linéarisation des fonctions	p^n
Couche 5	Somme	1

Tableau 4.1 les différentes couches d'un système ANFIS

Tel que :

n : le nombre des entrées.

p : le nombre des sous ensembles flous d'entrée (partition flou).

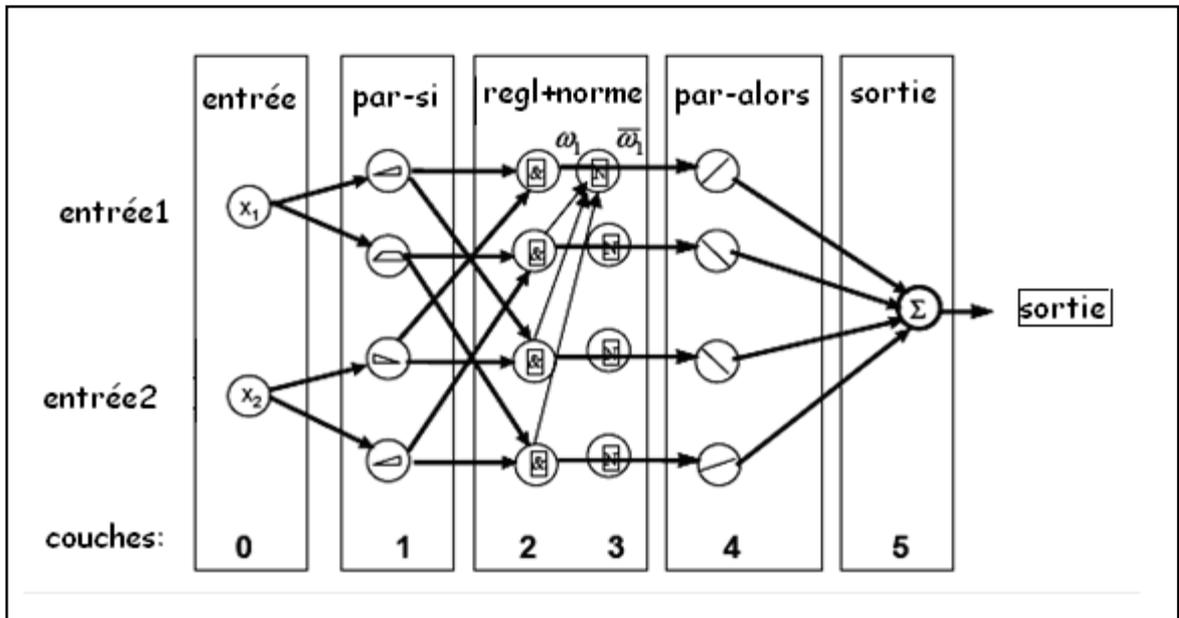


Figure 4.7- Réseau d'ANFIS

Noter que les neurones dans ANFIS ont différentes structures:

- Valeurs [fonction d'appartenance définie par différentes formes].
- Règles [habituellement produit].
- Normalisation [division de somme et d'arithmétique].
- Fonctions [régressions linéaires et multiplication avec \bar{w} , tel que \bar{w} est la normalisation du poids w].
- la sortie [Somme Algébrique].

4.4.2. Algorithme d'apprentissage de l'ANFIS

Le système ANFIS applique le mécanisme d'apprentissage des réseaux neurone sur des techniques d'inférence floues. D'un autre terme L'ANFIS est un système d'inférence floue (SIF) dont les paramètres des fonctions d'appartenances sont ajustés en utilisant l'algorithme d'apprentissage rétropropagation, ou en combinaison avec un autre type d'algorithmes comme le moindre carré.

Dans l'architecture ANFIS proposée dans la figure 4.5, la sortie globale peuvent être exprimé en tant que des combinaisons linéaires des paramètres **conséquents** [25]. Avec plus précision, la conclusion (la sortie) sur la figure 4.5 peut être récrit comme :

$$\begin{aligned}
 f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \\
 &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2
 \end{aligned}
 \tag{4.11}$$

La sortie est une fonction linéaire des paramètres conséquences (p , q , r). ANFIS est représentation paramétrique deux ensembles de paramètres: S_1 et S_2 tel que :

- *S1* représente les paramètres des ensembles flous utilisés pour la fuzzification dans la première couche de système ANFIS

$$S1 = \{ \{a_{11}, b_{11}, c_{11}\}, \{a_{12}, b_{12}, c_{12}\}, \dots, \{a_{1p}, b_{1p}, c_{1p}\}, \dots, \{a_{np}, b_{np}, c_{np}\} \} \quad (4.12)$$

Ou *p* est le nombre de partition floue de chacun des variables d'entrées et *n* est le nombre de variables d'entrées.

- *S2* représente les coefficients des fonctions linéaires (les paramètres conséquents)

$$S2 = \{ p_1, p_2, p_3, \dots, q_1, q_2, q_3, \dots, r_1, r_2, r_3, \dots \} \quad (4.13)$$

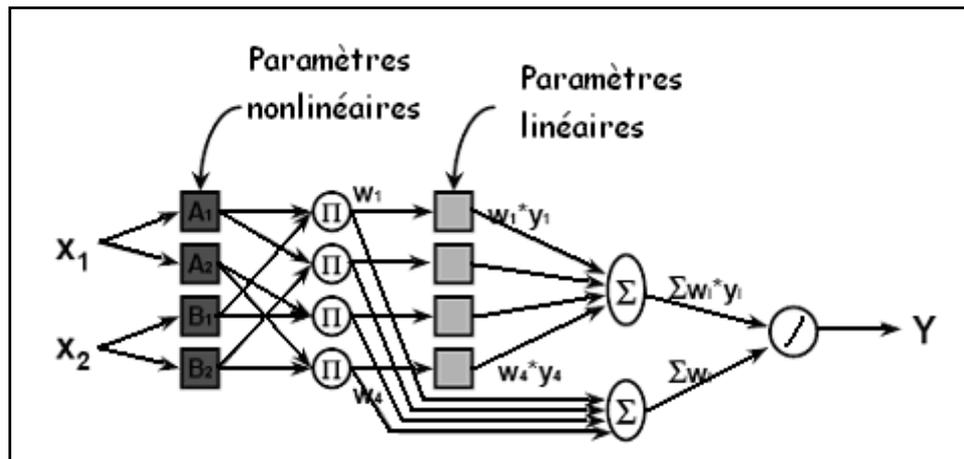


Figure 4.8 : Méthode d'apprentissage Hybride

	Passage Vers l'avant	Passage En arrière
Paramètre de fonction d'appartenance (non linéaire a_i, b_i, c_i)	fixe	Rétro propagation
Paramètre de coefficient (Linéaire p, q, r)	moindres carrés	fixe

Tableau 4.2- Les paramètres à ajuster d'un système ANFIS
ANFIS utilise un cycle d'apprentissage de deux passages :

- **le passage en avant:** *S1* est fixe et *S2* est calculé en utilisant l'algorithme de **moindre carré** de l'erreur (LSE). (Le LSE est appliqué seulement une fois lorsque commencer à obtenir les valeurs initiales des paramètres conséquents)
- **Passage en arrière:** *S2* est fixe et *S1* est calculé en utilisant l'algorithme de **Rétro-Propagation** [27] .

a. Rétro Propagation de l'ANFIS :

Mesure d'erreur E_k pour l'entrée k^{ieme} donné d'apprentissage:

$$E_k = \sum_{i=1}^{N(L)} (d_i - x_{L,i})^2 \quad (4.14)$$

$N(L)$: Nombre de neurones dans la couche L

d_i : Iemme composant de vecteur de sortie désiré

x_{Lj} : Iemme composant de vecteur de sortie de réel (de l'ANFIS)

Mesure d'erreur globale E:

$$E = \sum_{k=1}^K E_k \quad (4.15)$$

Pour chaque paramètre α_i ($\alpha_i \in S_1$) sont modifiées en suivant l'équation :

$$\Delta\alpha_i = -\eta \frac{dE}{d\alpha_i} \quad (4.16)$$

Où η est une constante positive appelée *pas d'apprentissage*)

Et le calcul de la quantité $\frac{\partial E}{\partial \alpha_i}$ est déjà expliqué dans chapitre 3.

4.5. Les avantages de l'ANFIS

Les avantages de cette technique stratégie sont :

- Exploitation de la connaissance disponible, grâce à la base de règles.
- Réduction de la taille de la base de règles : il suffit d'avoir des règles générales, les détails seront fournis par le RN.
- Réduction de la complexité de l'apprentissage : le RN doit simplement apprendre les cas particuliers ou les exceptions, pas le problème complet.
- Efficacité immédiate dès le début de l'apprentissage et possibilité d'éviter des comportements initiaux erratiques. [28]

4.6. Conclusion

Dans ce chapitre nous avons introduit et analysé une méthodologie permettant d'évaluer une perte de performance d'un système par l'évaluation des indicateurs de performance. Cette méthodologie s'appuie sur l'utilisation d'un réseau de neurones flou de type ANFIS. Cet outil permet d'intégrer une connaissance partielle issue de l'expertise et une connaissance issue de données. La connaissance experte est utilisée avec les relations causales dans une première modélisation assez simple et grossière mais suffisante. La connaissance experte est ensuite exprimée sous formes de règles floues et de contraintes sur la fuzzyfication des entrées, alors que la phase d'apprentissage permet d'ajuster les paramètres non définis (paramètres conséquents) à l'aide de données.

L'utilisation de connaissances expertes permet ainsi de résoudre en partie les problèmes d'exhaustivité des comportements représentés par un ensemble de données. En revanche, les capacités d'apprentissage de l'ANFIS permettent de combler le manque de précision issue de l'expertise. C'est donc un système gagnant où les deux sources de connaissance (règles et données) permettent de combler des lacunes de l'autre.

Ce système d'inférence adaptatif est très performant et largement utilisé dans les réalisations pratiques comme il est évident dans le chapitre suivant.