

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة البليدة 1
Université BLIDA 1

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Étude

réalisé par

BRAHIM Walid

&

BENKOUIDER Mounir

pour l'obtention du Master en Électronique option automatique

Thème

Classification neuro-floue pour l'identification des défaillances

Proposé par : *SALHI Hassen & KHENTOUT Nourddine*

Soutenu le : 21/06/2015

devant le jury :

Mr. K. KARA	Pr	Université Blida 1	Président
Mr. A. MADDI	MCB	Université Blida 1	Membre
Mr. H. SALHI	Pr	Université Blida 1	Promoteur
Mr. N. KHENTOUT	Chercheur	C.R.N.Birine	Encadreur

Année Universitaire 2014-2015



Dédicace

Je tiens à dédier ce modeste travail avant tout :

A mon très cher père.

A ma très chère mère, ma fierté et mon bonheur.

A mes très chers frères et ma sœur.

A toute la famille.

A tous mes amis et mes collègues de promos

A mon binôme « Mounir ».

et à tous ceux qui m'ont aidés de près ou de loin à accomplir

ce travail.

W. Brahim





Dédicace

A maman, que dieu me la garde éternellement heureuse.

A papa, Allah yarehmou.

A mes sœurs et mes frères.

A tous mes amis.

A tous ceux qui me sont chers.

M. Benkouider



Remerciements

Nous tenons à exprimer nos vifs remerciements à Allah le tout puissant, pour la volonté, la santé et la patience qu'il nous a donné durant toutes ces longues années d'étude et qui nous ont permis d'arriver là.

Que tous ceux qui nous ont assistés et encouragés pour la réalisation de ce mémoire, trouvent ici l'expression de notre profonde gratitude.

Nous tenons à remercier particulièrement :

Notre promoteur monsieur H.Salhi pour son aide et son suivi durant l'élaboration de ce mémoire.

Notre encadreur Monsieur N.Khentout chercheur au centre de recherche nucléaire de Birine qui nous a accueillie dans son laboratoire d'électronique nucléaire et nous a guidé et encouragé durant ce travail.

Tous ceux qui ont contribué de près ou de loin à notre formation trouvent ici nos sincères remerciements.

Grâce à eux, ce stage nous a permis de nous insérer avec facilité dans le monde professionnel et de disposer désormais d'une expérience concrète à présenter auprès de nos futurs employeurs.

Résumé

Lorsque la surveillance des défauts des systèmes industriels est réalisée avec efficacité. Elle permet de détecter de façon précoce une dégradation, qui est un moyen important pour contribuer à un gain de productivité. Sa vocation principale est de détecter et de classer une éventuelle défaillance du processus.

L'utilisation des techniques d'intelligence artificielle, ramène la classification, partie du diagnostic, à un niveau d'automatisation très élevé. La combinaison entre les différentes sortes de ces techniques offre des possibilités d'amélioration des performances de ces classificateurs de données en général.

Mots clés : Classification, Réseaux de neurones artificiels, Logique floue, Neuro-flou, NEFCLASS.

Abstract

When fault monitoring of industrial systems, are effectively realized and allow the detection of possible degradation. It is considered as an important tool to contribute to better productivity. Its principal vocation is to detect and diagnose in order to prevent failure in the process.

The use of artificial intelligence technics has brought the fault classification, a part of diagnostics, to a very high level of automation. The combination between different types of these techniques offer better possibilities of performances improvement of the data classifiers in general.

Keywords: Classification, Artificial neural network, Fuzzy logic, Neuro-fuzzy, NEFCLASS.

ملخص

تحليل الأعطال في الأنظمة الصناعية، عندما يؤدي بكفاءة و اذا كان يسمح بالتعرف على تدهور أو عطب، يعتبر وسيلة لتحقيق الزيادة في نسبة الانتاج. هدفه الرئيسي هو التعرف على الأعطال المحتملة في الأجهزة الصناعية وتحديد مكانها. استعمال الذكاء الاصطناعي سمح بدفع مجال تحليل الأعطال الى مستوى متقدم من التطور، وذلك بتوفير الأدوات من أجل عمليات التصنيف واتخاذ القرارات. الدمج بين هذه الأدوات، أدى الى خلق فرص كبيرة من أجل تحسين قدرات هذه المصنفات.

الكلمات المفتاحية: شبكة الأعصاب الاصطناعية، التصنيف، الأنظمة الهجينة، تحليل الأعطال.

Sommaire

Introduction générale	1
I - Chapitre I : Réseaux de neurones artificiels	3
1.1 - Introduction	3
1.2 - Définition	4
1.3 - Architecture	5
1.3.1 - Simple/multicouche	5
1.3.2 - Non-récurrent/récurrent	6
1.4 - Etapes de création	8
1.4.1 - Apprentissage	10
1.4.2 - Test	12
1.5 - Applications	12
1.6 - Conclusion	12
II - Chapitre II : Logique floue	14
II.1 - Introduction	14
II.2 - Principe	14
II.3 - Opérateurs de la logique floue	15
II.4 - Fonctions d'appartenance	17
II.5 - Système d'inférence floue	17
II.5.1 - Fuzzification	19
II.5.2 - Inférence	19
II.5.3 - Défuzzification	20
II.6 - Types de système d'inférence	21
II.7 – Création du système flou	23
II.8 - Application de la logique floue	24
II.9 - Conclusion	24
III - Chapitre III : Systèmes neuro - flous	25
III.1 - Introduction	25
III.2 - Définition	25
III.3 - Combinaison des réseaux de neurones et de la logique floue	25
III.3.1 - Systèmes neuro-flous coopératifs	25
III.3.2 - Systèmes neuro-flous concurrents	26
III.3.3 - Systèmes neuro-flous hybrides	27
III.4 - Apprentissage des systèmes neuro-flous	28
III.5 - Modèles des systèmes neuro-flous hybride	29

III.5.1 - ANFIS.....	30
II.5.2 - NEFCLASS.....	33
III.6 - Création du système neuro-flou.....	33
III.7 - Applications des systèmes neuro-flous	34
III.8 - Conclusion.....	36
IV - Chapitre IV : Classification des données	37
IV.1 - Introduction	37
IV.2 – Définition	37
IV.3 - Catégorisation	38
IV.4 - Méthodes.....	39
IV.4.1 - Méthodes statistiques	39
IV.4.2 - Intelligence artificielle	43
IV.5 - Etapes de Création	44
IV.6 - Chaîne de classification.....	45
IV.7 - Performances.....	46
IV.8 - Conclusion	46
V - Chapitre V : Application du système neuro-flou pour la classification.....	47
V.1 - Introduction	47
V.2 - Classification par réseau de neurones	47
V.2.1 – Base de données IRIS.....	47
V.2.2 – Base de données du cancer de sein (Breast Cancer).....	49
V.2.3 – Base de données Wine	50
V.3 - Classification par NEFCLASS.....	52
V.3.1 – Base de données IRIS	55
V.3.2 – Base de données du Cancer de sein (Breast Cancer).....	56
V.3.3 – Base de données Wine	57
V.4 – Conclusion	57
Conclusion générale	60
Bibliographie	61
Annexe I	A.I-1
Annexe II	A.II-1
Annexe III	A.III-1

Liste des figures

Chapitre I

Fig. I.1 - La structure du neurone biologique.....	3
Fig. I.2 - La structure d'un neurone formel.....	4
Fig. I.3 - Les fonctions d'activations les plus utilisées.....	5
Fig. I.4 - Différentes architectures des réseaux de neurones.....	6
Fig. I.5 - Réseau de neurones PMC.....	7
Fig. I.6 - Réseau de neurones RBF.....	7
Fig. I.7 - Réseau de neurones récurrent.....	8
Fig. I.8 - Etapes de création du réseau de neurones.....	9
Fig. I.9 - Apprentissage Supervisé.....	10
Fig. I.10 - Apprentissage non-supervisé.....	12

Chapitre II

Fig. II.1 - Opérateurs Flous.....	15
Fig. II.2 - Les fonctions d'appartenance les plus utilisées.....	17
Fig. II.3 - Système d'inférence floue.....	18
Fig. II.4 - Exemple de fuzzification.....	20
Fig. II.5 - Méthodes de défuzzification.....	21
Fig. II.6 - Types des systèmes flous.....	22
Fig. II.7 - Etapes de création d'un système d'inférence floue (SIF).....	23

Chapitre III

Fig. III.1 - Système neuro-flou coopératif.....	26
Fig. III.2 - Système neuro-flou concurrent.....	26
Fig. III.3 - Structure générale d'un système flou neuronal (hybride).....	27
Fig. III.4 - Synoptique de l'apprentissage des systèmes neuro-flous.....	28
Fig. III.5 - Ajustement des intervalles des ensembles flous.....	29
Fig. III.6 - Réglage des paramètres de la fonction d'appartenance triangulaire.....	30
Fig. III.7 - Architecture du modèle ANFIS.....	32
Fig. III.8 - Architecture du modèle NEFCLASS.....	34
Fig. III.9 - Etapes de création d'un système neuro-flou.....	35

Chapitre IV

Fig. IV.1 – Classification linéaire et non-linéaire	39
Fig. IV.2 - Exemple de classification k-ppv.....	40
Fig. IV.3 - Arbre de décision de reconnaissance des fruits.....	41
Fig. IV.4 - L'hyperplan optimal (en rouge) avec la marge maximale. Les échantillons pleins sont des vecteurs supports.....	42
Fig. IV.5 – Synoptique du système de classification.....	45

Chapitre V

Fig. V.1 – RN crée par Matlab pour la base de données iris.....	47
Fig. V.2 – Les performance de la phase d'apprentissage du RN pour les données iris..	48
Fig. V.3 – Représentation graphique des résultats de classification pour les données iris.....	48
Fig. V.4 – RN crée par Matlab pour la base de données du cancer de sein.....	49
Fig. V.5 – Les performance de la phase d'apprentissage du RN pour les données du cancer du sein.	49
Fig. V.6 – Représentation graphique des résultats de classification pour les données du cancer du sein.....	50
Fig. V.7 – RN créer par Matlab pour la base de données Wine.....	50
Fig. V.8 – Les performance de la phase d'apprentissage du RN pour les données du wine.....	51
Fig. V.9 – Représentation graphique des résultats de classification pour les données du wine.....	52
Fig. V.10 – Synoptique de création du classificateur NEFCLASS.....	54
Fig. V. 11 - Résultats de classification des fleurs iris par NEFCLASS.....	55
Fig. V. 12 - Résultats de classification des données du cancer par NEFCLASS.....	56
Fig. V. 13 - Résultats de classification des données du wine par NEFCLASS.....	58

Annexe I

Fig. A.I.1 – Système neuro-flou de Mamdani	A.I-1
Fig. A.I.2 – Système neuro-flou de Sugeno.....	A.I-2
Fig. A.I.3 – Synoptique du réseau neuro-flou (Tsukamoto).....	A.I-3
Fig. A.I.4 – Structure générale de l'apprentissage des systèmes neuro-flous.....	A.I.4

Annexe II

Fig. A.II.1 – Résumé de la phase d'apprentissage du RN pour la base de données iris...	A.II-1
Fig. A.II.2 – Résumé de la phase d'apprentissage du RN pour la base de données du cancer du sein.....	A.II-2
Fig. A.II.3 – <i>Résumé de la phase d'apprentissage du RN pour la base de données wine</i>	A.II-3
Fig. A.II.4 – Modification des ensembles flous pour l'entrée Petal Width.....	A.II.4
Fig. A.II.5 – Modification des ensembles flous pour l'entrée Clump thickness.....	A.II.4
Fig. A.II.6 – Modification des ensembles flous pour l'entrée Flavonoids.....	A.II.5
Fig. A.II.7 – Modification des ensembles flous pour l'entrée Color intensity.....	A.II.5
Fig. A.II.8 – Modification des ensembles flous pour l'entrée Proline.....	A.II.6

Introduction
Générale

Les installations nucléaires, particulièrement les réacteurs, sont larges en échelle et en complexité, alors l'information recueillie à partir des différents systèmes et processus est excessive, ce qui fait que les opérateurs humains ne peuvent pas la traiter convenablement. En fait, quand une défaillance s'effectue dans l'installation, un afflux de données aura lieu, et la cause de la défaillance ne peut pas être identifiée facilement et proprement.

Le rôle principal de la surveillance industrielle est d'augmenter la disponibilité des installations industrielles afin de réduire les coûts directs et indirects de la maintenance des équipements de production. Les coûts directs de cette maintenance sont ceux relatifs aux diverses pièces de rechange, main d'œuvre, etc. Par contre, les coûts indirects sont essentiellement dûs au manque à gagner engendré par un arrêt de production. Alors l'enjeu d'une bonne politique de surveillance est très important pour les entreprises soucieuses d'avoir une meilleure maîtrise des coûts de maintenance [1].

La surveillance des défauts dans les divers systèmes est basée principalement sur la détection du défaut et en cas de présence, il est nécessaire de faire un diagnostic qui consiste à identifier, isoler et localiser l'anomalie. La classification ou généralement la reconnaissance de forme sont des termes utilisés souvent dans le domaine du diagnostic et parfois pour signifier le diagnostic lui-même.

Les réseaux de neurones artificiels et la logique floue, sont deux techniques largement utilisées dans le domaine du diagnostic et de surveillance des systèmes industriels. Leur usage a donné des résultats positifs, ce qui nous a incités à les introduire dans les systèmes de classification des défauts. Leur combinaison permet de réunir les avantages de chaque technique et de pallier les inconvénients qu'elles présentent.

La combinaison des réseaux de neurones artificiels et la logique floue, connue par la méthode neuro-floue, est un réseau de neurones qui est équivalent à la structure de la logique floue. Il est considérablement pratique parce qu'il intègre tous les avantages des deux algorithmes et rend le système entier plus robuste.

L'objectif de ce travail est le développement d'un module de classification basé sur la reconnaissance de formes des signaux de défaillances en utilisant l'approche neuro floue.

Ce mémoire est divisé en cinq chapitres. Le *premier chapitre* comporte une introduction au concept de base des réseaux de neurones.

Le *deuxième chapitre* est réservé à la logique floue, sa définition, les caractéristiques essentielles de raisonnement approximatif, et le système d'inférence floue. La logique floue permet en effet de faire le lien entre modélisation numérique et modélisation symbolique.

Le *troisième chapitre* traite la combinaison des deux techniques précédentes, soit l'approche neuro-floue. Les systèmes neuro-flous nés de l'association des réseaux de neurones avec la logique floue, de manière à tirer profits des avantages de chaque technique, leur principale propriété des systèmes est leur capacité à traiter dans un même outil des connaissances numériques et symboliques d'un système. Ces derniers permettent donc d'exploiter les capacités d'apprentissage des réseaux de neurones d'une part et les capacités de raisonnement de la logique floue d'autre part.

Dans le *quatrième chapitre*, nous allons aborder le problème de la classification, sa définition, ses principales étapes, et les différentes méthodes utilisés avec quelques exemples d'application.

Le *cinquième chapitre* contient les résultats obtenus, à savoir une étude comparative entre les performances des classificateurs : réseau de neurones et neuro-flou (NEFCLASS).

Enfin, nous terminons par une *conclusion et perspectives* qui contient un récapitulatif des résultats obtenus et les perspectives de recherche à poursuivre.

Chapitre I

***Réseaux de
Neurones Artificiels***

1.1 - Introduction

Les réseaux de neurones artificiels (RNA) constituent une nouvelle approche de traitement d'information. Ils offrent des solutions compactes et rapides pour une large gamme de problèmes, en particulier ceux ayant des contraintes en temps réel tel le cas de la plupart des applications de surveillance de défaillance des systèmes (nucléaires, robotiques, aériennes, spatiales, maritimes, etc.). Ceci est davantage concrétisé par l'utilisation des émulations et des implantations matérielles.

Les RNA ont été évoqués pour la première fois en 1943, dans un article publié par *MC-Culloch* et *Pitts*. Entre 1970 et 1980 peu d'articles furent publiés en raison de la faible quantité de chercheurs intéressés. Au début des années quatre-vingt, les recherches dans ce domaine ont repris surtout après la publication du travail de *J.J. Hopfield* en 1982 [2].

La structure d'un neurone artificiel est inspirée du modèle du neurone biologique (*Fig. 1.1*). Ce dernier est constitué de trois parties principales: La zone d'entrées (dendrites et corps cellulaire), l'axone et la zone de sortie représenté par les terminaisons synaptiques.

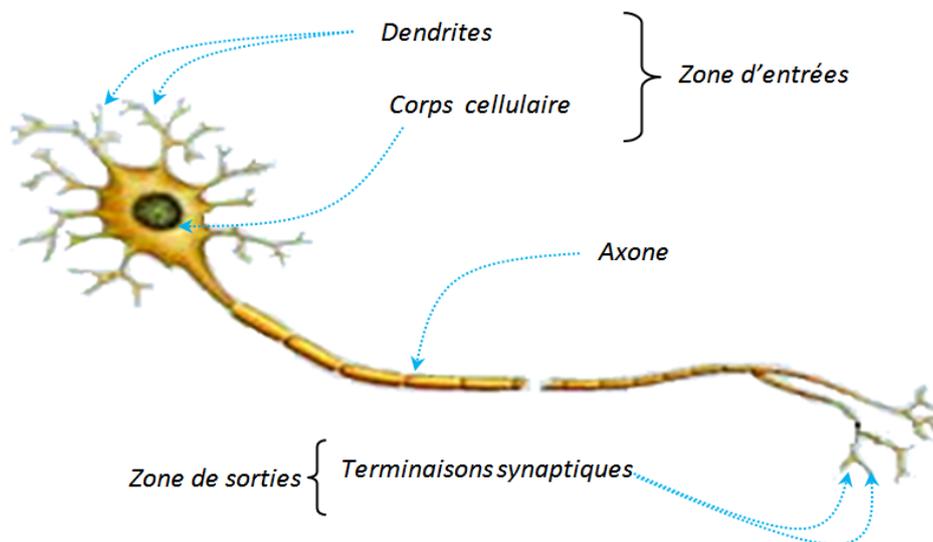


Fig. 1.1 - La structure du neurone biologique.

1.2 - Définition

Les réseaux de neurones artificiels sont des systèmes constitués d'un ensemble de neurones formels connectés selon une architecture bien définie. Sachant, qu'un neurone formel (*Fig. 1.2*), est une cellule de communication et de traitement de l'information.

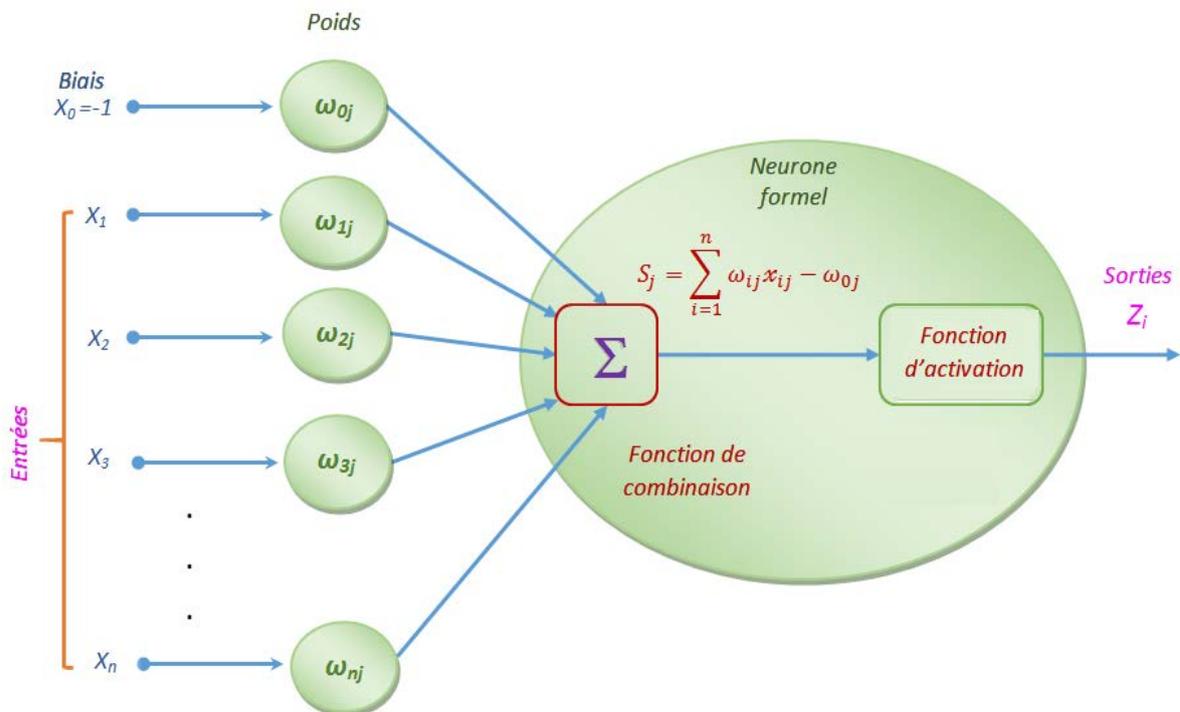


Fig. 1.2 - La structure d'un neurone formel.

Ses principaux composants sont: [2]

- 1- Les entrées et les sorties qui peuvent prendre des valeurs binaires ou réelles;
- 2- La fonction de combinaison, nommée aussi d'agrégation ou de sommation, est une fonction de pondération des entrées par les poids synaptique ;
- 3- La fonction d'activation ou de seuillage est une fonction linéaire ou non linéaire qui prend comme paramètres l'entrée totale S . Cette fonction peut prendre plusieurs formes (*Fig. 1.3*) ;
- 4- Enfin la fonction de sortie (qui n'est pas présentée sur *Fig. 1.2*) est considérée généralement comme une fonction identité.

Les réseaux de neurones sont de puissants calculateurs, tolérants aux pannes, et leurs abilité d'apprendre et généraliser leurs connaissances les rendent des outils très efficace [3].

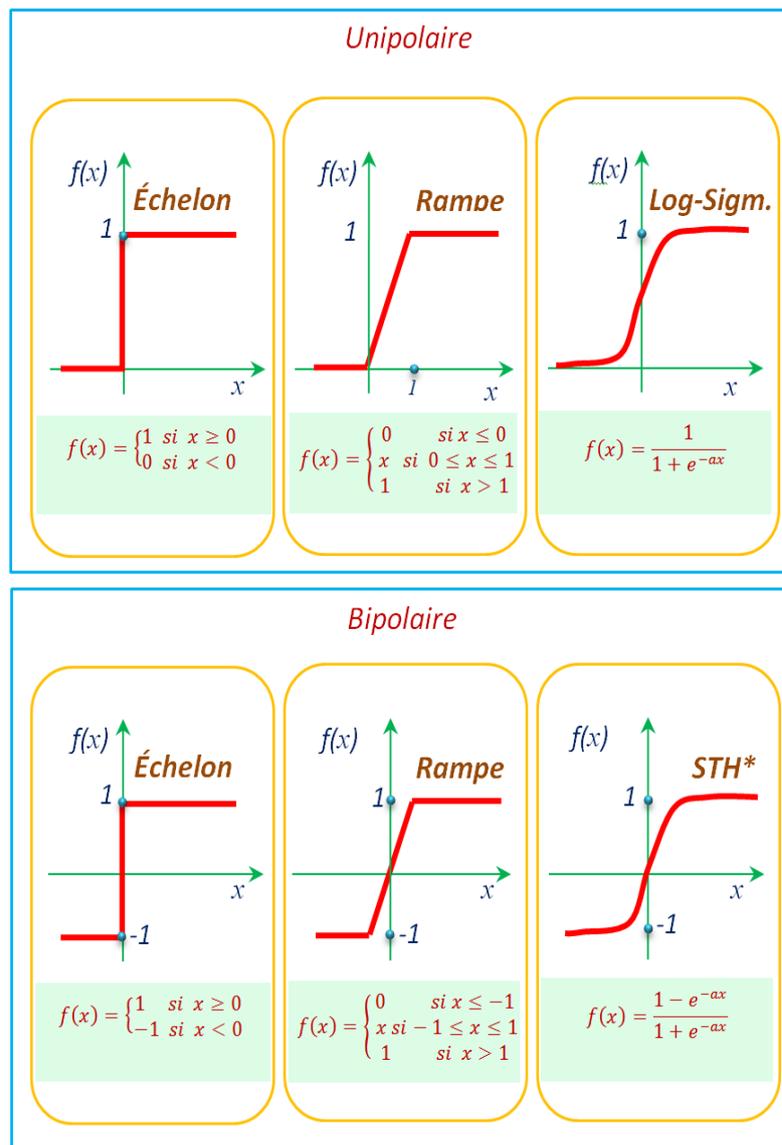


Fig. I.3 - Les fonctions d'activations les plus utilisées

*STH : sigmoïde, tangente hyperbolique.

1.3 - Architecture

Il existe plusieurs classifications des architectures des réseaux de neurones. Les plus connues sont basées sur le nombre de couches internes ; simples ou multicouches ou sur le type de connexion entre les neurones du réseau ; récurrents ou non-récurrents.

1.3.1 - Simple/multicouche

Les réseaux de neurones peuvent être classés selon le nombre de couche qui les constituent, simple-couche ou multicouche.

a - Simple couche

Le réseau de neurones simple-couche ou perceptron est un neurone formel avec une fonction d'activation de type échelon.

b - Multicouche

Les réseaux de neurones multicouches sont la classe la plus utilisée. Elle est caractérisée par la présence d'une ou plusieurs couches cachées. Elle comporte plusieurs modèles comme le perceptron multicouche, les réseaux de neurones de fonction à base radiale, les réseaux de Hopfield, les cartes de Kohonen.

1.3.2 - Non-récurrent/récurrent

Une autre classification des architectures des réseaux de neurones se base sur la direction de circulation de l'information dans le réseau, soit non-récurrent ou récurrent (Fig. 1.4).

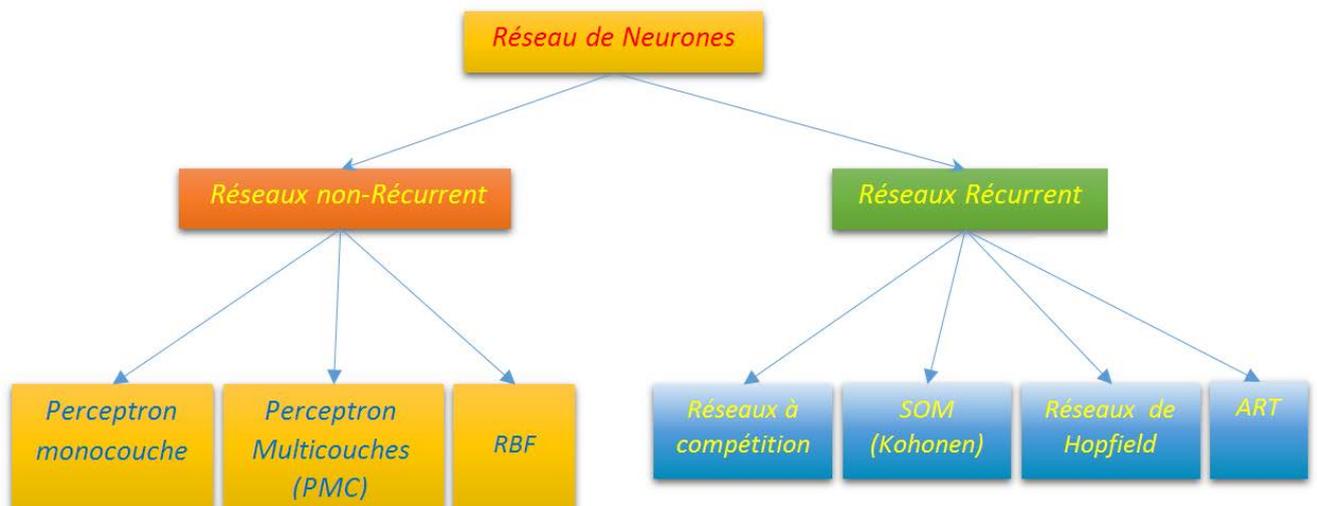


Fig. 1.4 - Différentes architectures des réseaux de neurones.

1.3.2.1 - Non-récurrent

Un réseau de neurones non-récurrent, nommé parfois non-bouclé, statique, ou *'feedforward'* en anglais, est présenté par un ensemble de neurones connectés entre eux tel que l'information circule depuis les entrées vers les sorties sans retour [1].

On distingue plusieurs modèles du genre non-récurrent, principalement: réseaux de neurones multicouches (Perceptron MultiCouche 'PMC') et les réseaux de neurones de fonction à base radiale (FBR).

a - Réseaux de neurones PMC

Le perceptron multicouche (PMC) est un réseau de neurones artificiels non-récurrents organisé en plusieurs couches (*Fig. 1.5*). Les algorithmes de descente du gradient et rétro-propagation sont utilisés pour l'apprentissage.

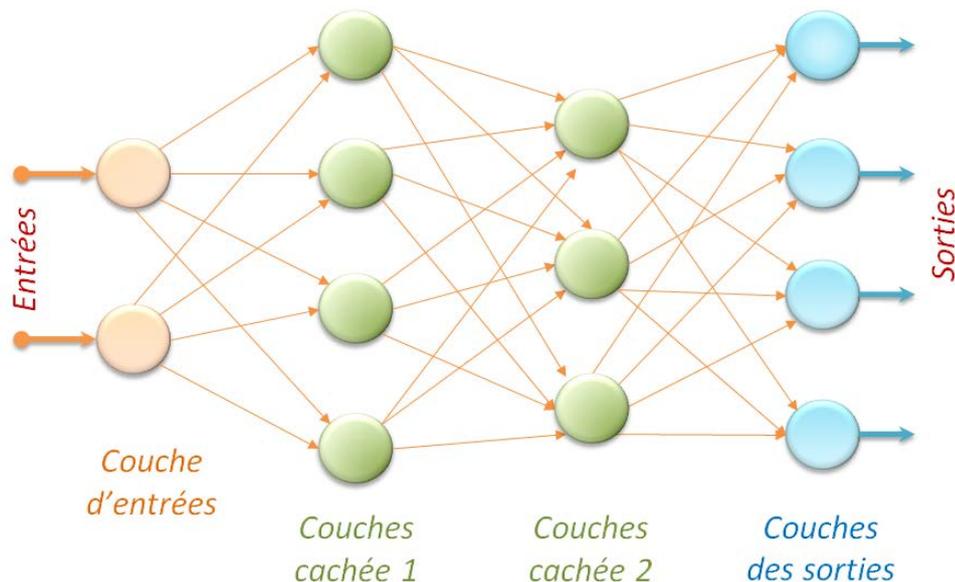


Fig. 1.5 - Réseau de neurones PMC.

b - Réseaux de neurones RBF

Les réseaux de neurones à fonction de base radiale (*Fig. 1.6*) sont des réseaux de neurones à une seule couche cachée dont les fonctions d'activation sont des fonctions à base radiale, le plus souvent des gaussiennes (définies par leur centre et leur écart type) [4].

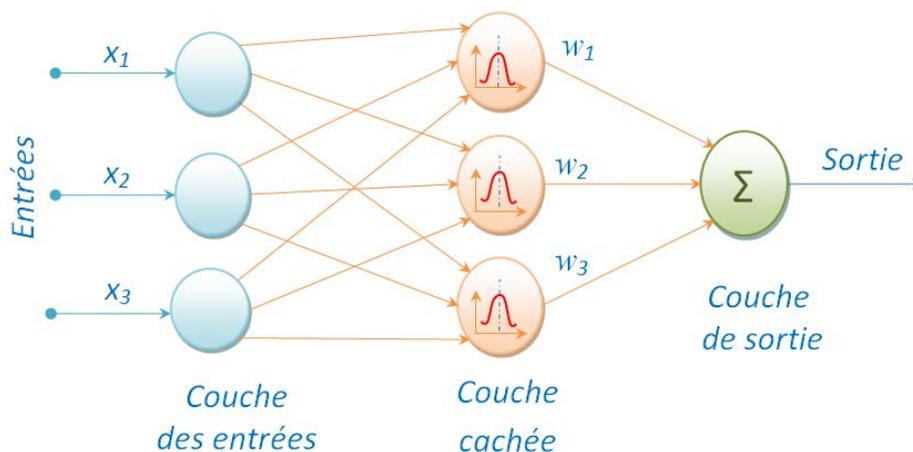


Fig.1.6 - Réseau de neurones RBF.

1.3.2.2 - Récurrents

Les réseaux de neurones récurrents (*Fig. 1.7*) sont caractérisés par la prise en compte de l'aspect temporel du phénomène qui se traduit par la présence d'au moins une *boucle* de rétroaction au niveau des neurones ou entre les couches. Ces modèles présentent un inconvénient majeur qui est les difficultés à les mettre en œuvre.

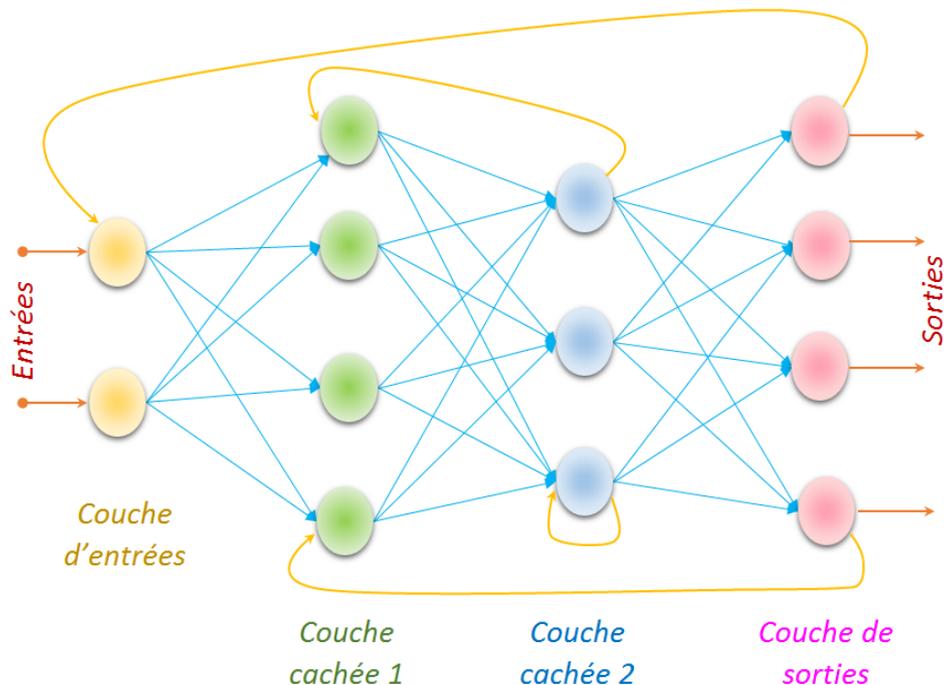


Fig. 1.7 - Réseau de neurones récurrent.

Il existe plusieurs modèles dont les plus connus sont les cartes de Kohonen et les réseaux de Hopfield. Les réseaux de Hopfield sont entièrement connectés, c'est-à-dire que chaque neurone est connecté à tous les autres neurones. Dans ce genre de réseaux, il n'y a aucune différenciation entre les neurones d'entrée et de sortie. Les cartes de Kohonen sont des réseaux à apprentissage non-supervisé qui établissent une carte discrète, ordonnée topologiquement, en fonction des formes d'entrées [1], [2].

1.4 - Etapes de création

La création d'un réseau de neurones passe par plusieurs étapes (*Fig. 1.8*) à savoir:

- 1- Le choix de l'*architecture* ;
- 2- La *configuration* qui consiste à définir les fonctions d'activation par couche et les valeurs initiales des poids ainsi que le nombre d'entrées et sorties, de couches, et de neurones sur chaque couche ;

- 3- La phase d'apprentissage ;
- 4- Le test.

Si les résultats du test ne sont pas satisfaisants, il est toujours possible de reconfigurer le réseau et refaire l'apprentissage jusqu'à en avoir les résultats désirés.

Afin d'améliorer les performances du réseau, certaines améliorations peuvent être introduites sur le nombre des couches cachées et de neurones dans chaque couche, sur la fonction de performance, fonction d'apprentissage et les dimensions de la base de données.

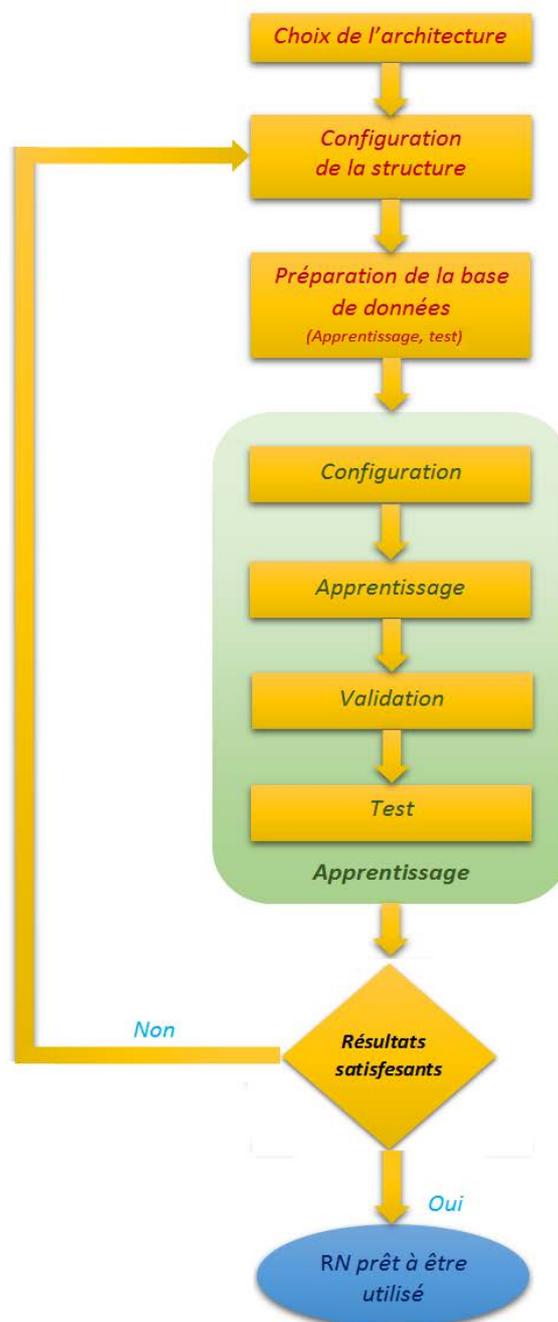


Fig. I.8 - Etapes de création du réseau de neurones.

1.4.1 - Apprentissage

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.

Il consiste à choisir la fonction de performance (MSE, MAE, etc.), les conditions d'arrêt (nombre maximal d'itérations, d'erreur de validation, gradient minimal, valeur de performance et temps maximal), l'algorithme qui est spécifique à chaque architecture (retro-propagation, etc.), le type d'apprentissages (batch ou incrémentale).

On distingue deux grandes classes d'algorithmes d'apprentissage :

- Supervisé ;
- Non-supervisé.

1.4.1.1 - Supervisé

Dans l'apprentissage supervisé, les couples d'entrées avec leurs sorties désirées sont introduits. Les performances du réseau sont déterminées par la suite, par l'intermédiaire d'un critère à optimiser. L'intérêt de cet apprentissage est de trouver les poids synaptiques optimaux afin de ramener au réseau un comportement désiré [2].

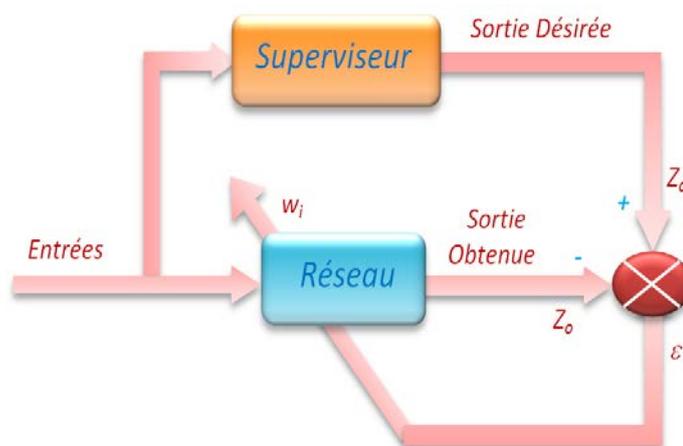


Fig I.9 - Apprentissage Supervisé.

Les algorithmes les plus utilisés pour l'apprentissage supervisé sont : descente du gradient et Widrow-Hoff.

a - Algorithme de la descente du gradient

Pour comprendre cette méthode, il faut définir :

- L'erreur quadratique E ;
- Le couple (X_k, Y_k) 'entrée – sortie désiré' pour le $k^{\text{ième}}$ échantillon;
- La sortie obtenue pour le $k^{\text{ième}}$ échantillon notée S_k ;
- Le pas d'apprentissage α (dans l'intervall $[0, 1]$).

L'erreur quadratique est définie par

$$E = \frac{1}{2} \sum_{k=1}^N (Y_k - S_k)^2 \quad (1.1)$$

L'erreur est nulle si le réseau de neurones ne se trompe sur aucun des exemples, c'est à dire s'il parvient à calculer la bonne sortie (par exemple classifier) pour chacun des exemples correctement. C'est rarement le cas car souvent on démarre avec un vecteur des poids $W = \{w_1, w_2, \dots, w_n\}$ tirés aléatoirement. Il s'agit donc de minimiser, pour un ensemble de N exemples donné, cette erreur quadratique.

La méthode de descente du gradient passe par les étapes suivantes:

- Créer n variables dw_i , pour $1 \leq i \leq n$, égales à 0 ;
- Prendre un exemple e_k , pour $1 \leq k \leq N$;
- Calculer la sortie obtenue avec les poids actuels, notée S_k (1) ;
- Rajouter à dw_i , pour tout $1 \leq i \leq n$, le nombre $\alpha (Y_k - S_k) X_i$ (2) ;
- Répéter (1) et (2) sur chacun des exemples ;
- Pour $1 \leq i \leq n$, remplacer w_i par $w_i + dw_i$.

b - Algorithme de Widrow-Hoff

Dans l'algorithme précédent, les erreurs commises sur chaque échantillon sont sommés puis un changement de poids sera effectué. Cette méthode met beaucoup de temps avant de tendre vers le bon coefficient.

L'algorithme de Widrow-Hoff, ou encore "la règle delta", n'est qu'une variante de l'algorithme précédent. Il consiste à modifier les poids après chaque exemple, et non pas après que tous les exemples aient défilé. Ceci va donc minimiser l'erreur de manière précise, et ce sur chaque exemple.

1.4.1.2 - Non-supervisé

L'apprentissage non-supervisé, dont le synoptique est donné sur *Fig. 1.10*, est indépendant de l'utilisateur, les données ne contiennent pas d'informations sur une sortie désirée, c'est-à-dire, il n'y a pas de superviseur 'ou expert humain' pour le traitement de ces données. il est plutôt une méthode de groupement statistique qui trie les objets selon leurs propriétés et forment des ensembles ayant des caractéristiques similaires [5].

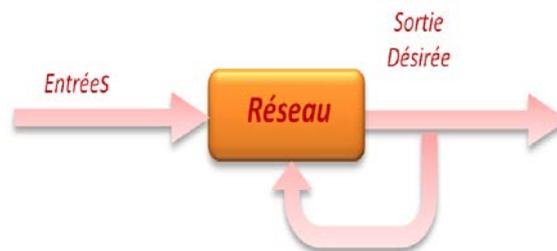


Fig. 1.10 - Apprentissage non-supervisé.

Ce type d'apprentissage est utilisé pour les modèles bouclés, qui sont souvent nommés auto-organiseurs ou bien à apprentissage compétitif où la readaptation des poids d'éléments d'entrée est autonome.

1.4.2 - Test

La phase de test consiste à évaluer les performances du réseau de neurones face à des données différentes aux données d'apprentissage. Cette phase a pour but de voir si le système a pu généraliser les connaissances acquises durant l'apprentissage.

1.5 - Applications

Les réseaux de neurones sont utilisés dans plusieurs domaines tels que le contrôle et la surveillance des différents instruments et systèmes industriels. Parmi les plus importantes applications, on trouve la détection, le diagnostic et la classification des défauts.

1.6 - Conclusion

Dans ce chapitre, un aperçu a été présenté sur les réseaux de neurones, leurs architectures, les types d'apprentissage et leurs différentes applications.

L'utilisation des réseaux de neurones a connu une grande augmentation à cause de ses avantages, à savoir:

- 1- Ne nécessite pas un modèle mathématique;
- 2- Apprentissage ;
- 3- Tolérance aux fautes (mesure de leur aptitude à exécuter la tâche demandée en présence d'informations erronées);
- 4- Leur capacité de calcul même si une partie du réseau est endommagée.

La limitation la plus importante des réseaux de neurones est dans l'interprétation, car ils sont considérés comme des boîtes noires, présentent des difficultés à comprendre ses manières de traitement de données.

Chapitre II

Logique

Floue

II.1 - Introduction

Les prémices de la logique floue sont apparues dans les années 20, mais sa vraie apparition était en 1965, où le professeur *Lotfi A. Zadeh* a publié son article qui a introduit le terme 'flou' dans la littérature technique et donne naissance au concept d'ensemble flou. Plusieurs chercheurs se sont intéressés à la logique floue comme *Mamdani* (1974), qui a introduit le contrôleur flou dans la commande des processus industriels. A partir des années 80, les recherches ont été lancées au Japon par le chercheur *M. Sugeno* en 1985.

La théorie des ensembles flous a donné naissance à un traitement original de l'incertitude, fondée sur l'idée d'ordre, et qui permet de formaliser le traitement de l'ignorance, et de formaliser les systèmes d'informations avancés. Les ensembles flous ont également un impact sur les techniques de classification automatique, et ont contribué à un certain renouvellement des approches existantes de l'aide à la décision [1].

Dans ce chapitre, le principe et les concepts fondamentaux de la logique floue seront traités.

II.2 - Principe

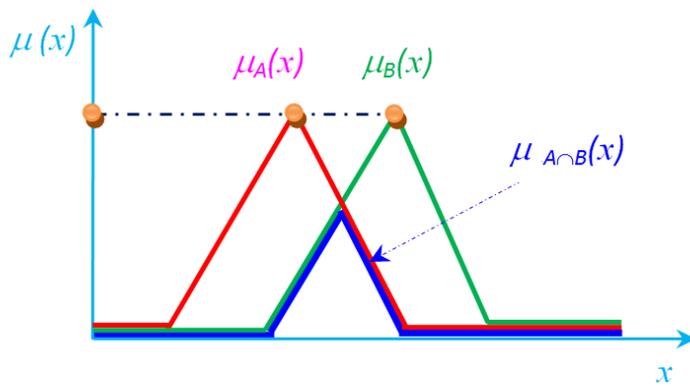
La logique floue constitue une généralisation de la logique booléenne classique, et ajoute cependant une fonctionnalité déterminante: la possibilité de calculer un paramètre en disant simplement dans quelle mesure il doit se trouver dans un interval de valeurs.

La notion d'ensemble est une des notions fondamentales des mathématiques mais elle ne permet cependant pas de rendre compte des situations simples et assez fréquentes, car dans la théorie des ensembles classiques, un élément appartient ou n'appartient pas à un ensemble donné [2]. Par exemple, il est aisé de définir parmi les capteurs, l'ensemble de ceux qui sont défectueux ou opérationnels. Par contre, il est impossible de définir le degré de vieillissement des capteurs puisqu'ils ne deviennent pas vieux du jour au lendemain. Ils vieillissent graduellement.

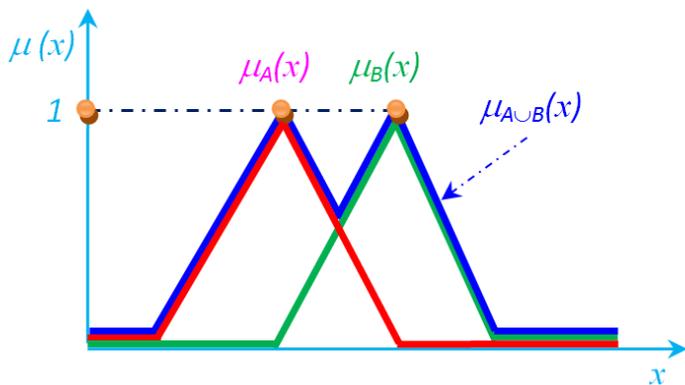
C'est pour tenir compte de ce genre de situation que *Zadeh* a introduit la notion d'ensemble flou. Elle repose sur le principe d'appartenance partielle. Chaque élément x peut: ne pas appartenir à un ensemble A , appartenir un peu à A , appartenir assez à A , appartenir fortement à A ou appartenir totalement à A . Le contour de chaque ensemble flou n'est pas "rigide" ou "net" mais "graduel" ou "flou" [2], [6], [7].

II.3 - Opérateurs de la logique floue

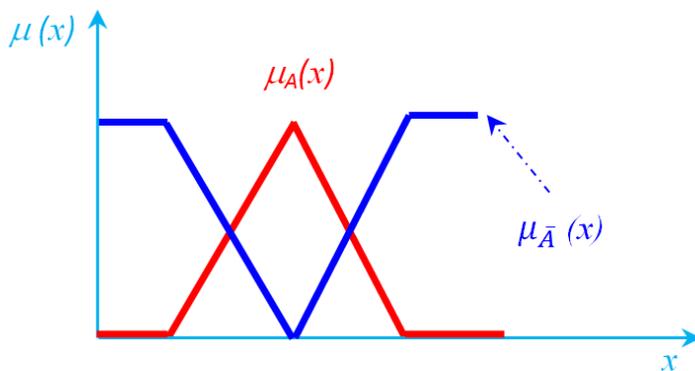
Comme dans la théorie des ensembles classiques, on définit l'intersection, l'union des ensembles flous, le complémentaire d'un ensemble flou ainsi que l'opérateur d'implication. Ces relations sont traduites par les opérateurs "ET", "OU", "NON", présenté par la Fig.II.1 et "→" pour les implications.



Intersection :
 $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$



Union:
 $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$



Complément:
 $\mu_{\bar{A}}(x) = 1 - \mu(x)$

Fig. II.1 - Opérateurs Flous.

- **ET**: opérateur de conjonction : les plus employés sont le “*minimum*” et le “*produit*”.

$$x \text{ appartient à } A \text{ ET } B \Leftrightarrow x \in A \cap B \Leftrightarrow \mu_{A \cap B}(x)$$

L’opérateur “ET” est connu par la norme triangulaire (norme T).

- **OU**: opérateur de disjonction, les plus employées sont le “*maximum*” et la “*somme*”.

$$x \text{ appartient à } A \text{ OU } B \Leftrightarrow x \in A \cup B \Leftrightarrow \mu_{A \cup B}(x)$$

L’opérateur “OU” est connu par la co-norme triangulaire qu’on appelle aussi la norme S.

- **NON**: opérateur qui désigne le complémentaire d’un ensemble flou :

$$x \text{ appartient au complément de } A \Leftrightarrow x \in \bar{A} \Leftrightarrow \mu_{\bar{A}}(x)$$

De même, l’implication floue peut être définie comme étant un opérateur qui permet d’évaluer le degré de vérité d’une règle floue. Les normes d’implication les plus utilisées sont :

[8]

- *Forme générale*

$$(x = A) \rightarrow (y = B) \quad \text{IF } x \text{ is } A \text{ THEN } y \text{ is } B$$

- *Zadeh*

$$\mu_R(x, y) = \max \{ \min [\mu_A(x), \mu_B(y)], 1 - \mu_A(x) \}$$

- *Mamdani*

$$\mu_R(x, y) = \min [\mu_A(x), \mu_B(y)]$$

- *Lukasiewicz*:

$$\mu_R(x, y) = \min \{ 1, [1 - \mu_A(x) + \mu_B(y)] \}$$

- *Godel*:

$$\mu_R(x, y) = \begin{cases} 1 & \mu_A(x) \leq \mu_B(y) \\ \mu_B(y) & \text{sinon} \end{cases}$$

II.4 - Fonctions d'appartenance

La fonction d'appartenance $\mu_A(x)$ fait correspondre à toute valeur d'entrée x son degré d'appartenance à l'ensemble A . Cette valeur graduelle est comprise entre 0 et 1. Elle peut théoriquement prendre n'importe quelle forme, toutefois les plus utilisées (Fig. II.1) sont les fonctions : triangulaire, trapézoïdale, gaussienne et sigmoïde.

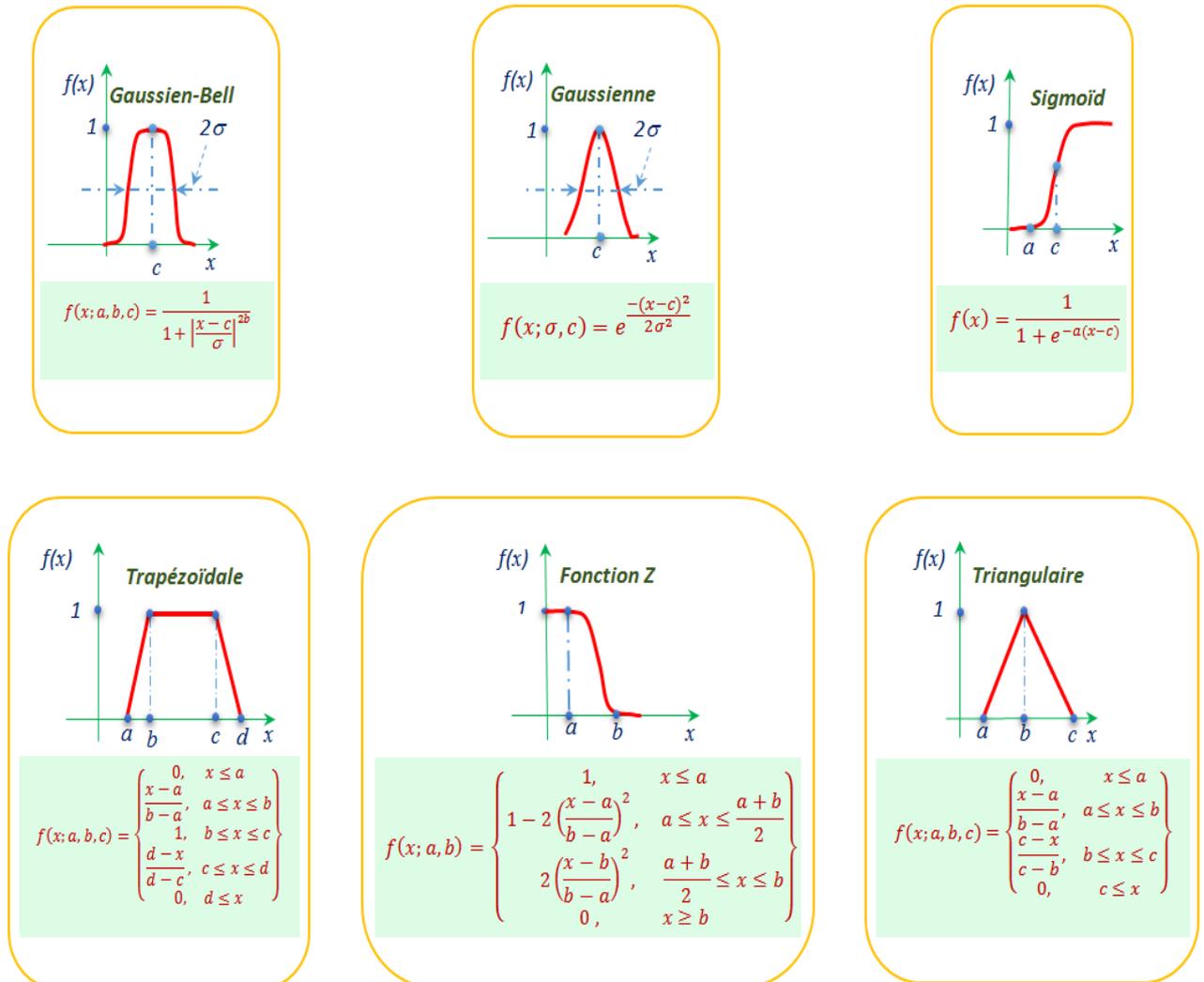


Fig. II.2 - Les fonctions d'appartenance les plus utilisées [9].

II.5 - Système d'inférence floue

La notion de règle floue permet de définir un système expert flou comme une extension d'un système expert classique, manipulant la proposition floue. Un système d'inférence floue (SIF) est formé de trois blocs principaux : *fuzzification*, *moteur d'inférence* et *défuzzification* donnée par Fig. II.3.

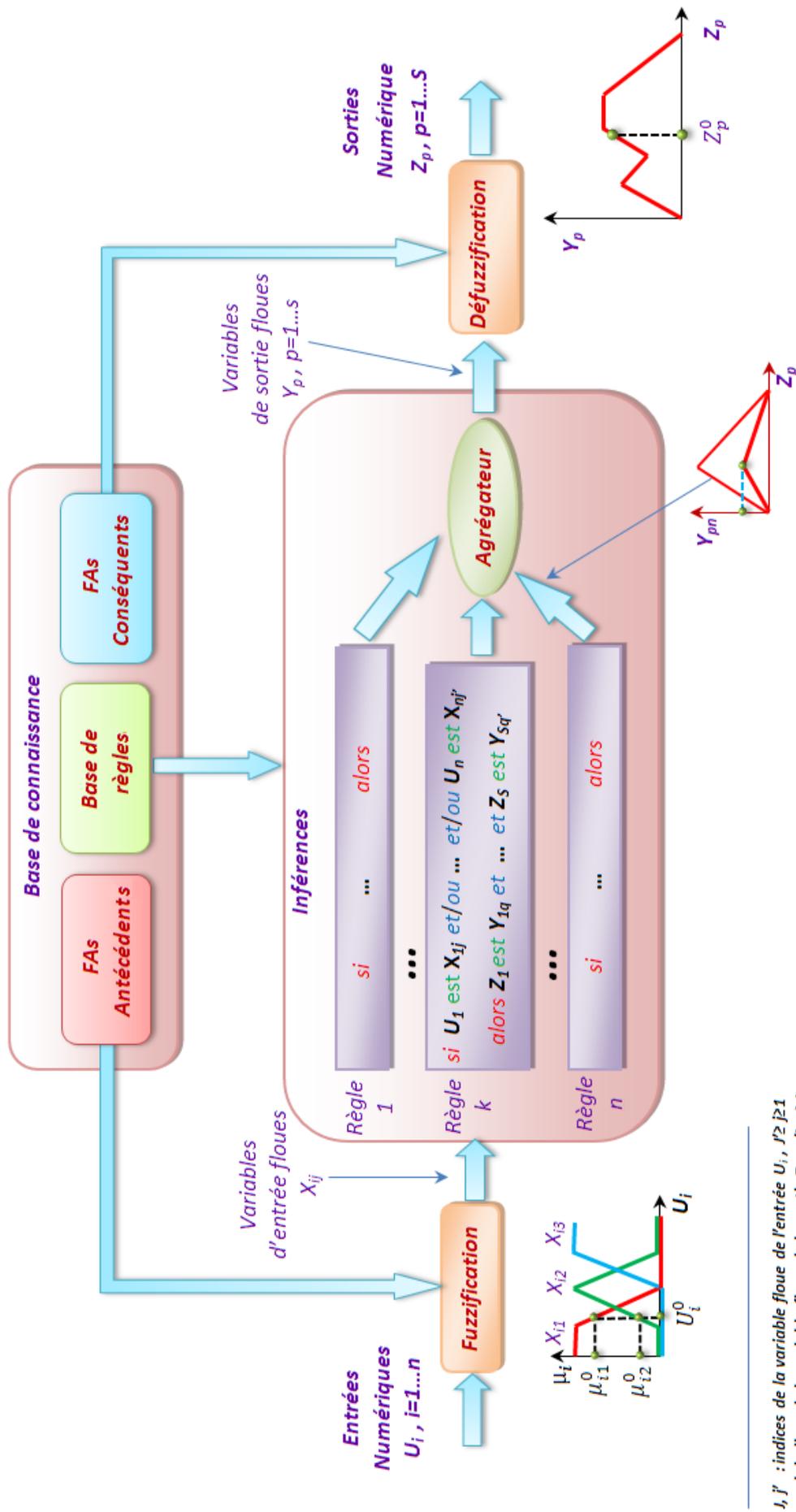


Fig. II.3 - Système d'inférence floue.

II.5.1 - Fuzzification

La fuzzification consiste à attribuer à la valeur réelle de chaque entrée, au temps t , son degré d'appartenance à chacune des valeurs floues préalablement définies, elle réalise ainsi un passage des grandeurs physiques aux variables linguistiques à traiter par la suite [2].

Le choix du nombre d'ensemble flou, de la forme des fonctions d'appartenance, du recouvrement de ces fonctions et de leur répartition sur l'univers de discours est primordial et un ensemble de règle doit être suivi.

Les fonctions triangulaires et trapézoïdales sont les plus répandues car leurs implémentation est facile et l'étape de fuzzification ne requiert alors que peu de temps de calcul lors de son évolution en temps réel.

Pour fuzzifier, il faut donner:

- L'univers du discours, c.-à-d. : Plage de variation possible de l'entrée considérée.
- Une partition en valeurs floues de cet univers.
- Les fonctions d'appartenance de chacune de ces valeurs.

Un exemple de fuzzification est illustré sur *Fig. II.4.*

II.5.2 - Inférence

Les inférences floues relient les variables d'entrée à une variable de sortie par des règles linguistiques combinées en utilisant les opérateurs logiques ET et OU. L'opérateur ET est utilisé pour les variables à l'intérieur d'une même règle, alors que l'opérateur OU relie les différentes règles entre elles [2].

Les règles d'inférences peuvent être décrites de façons *linguistique* ou *symbolique* ou par *matrice d'inférence*.

- *Linguistique* - on écrit les règles d'une façon explicite :

Si la formation est bonne et l'expérience est longue alors la qualification est très bonne

- *Symbolique* - il s'agit d'une description linguistique où la désignation des fonctions d'appartenance est remplacée par des observations :

Si F est B et E est L alors Q est TB

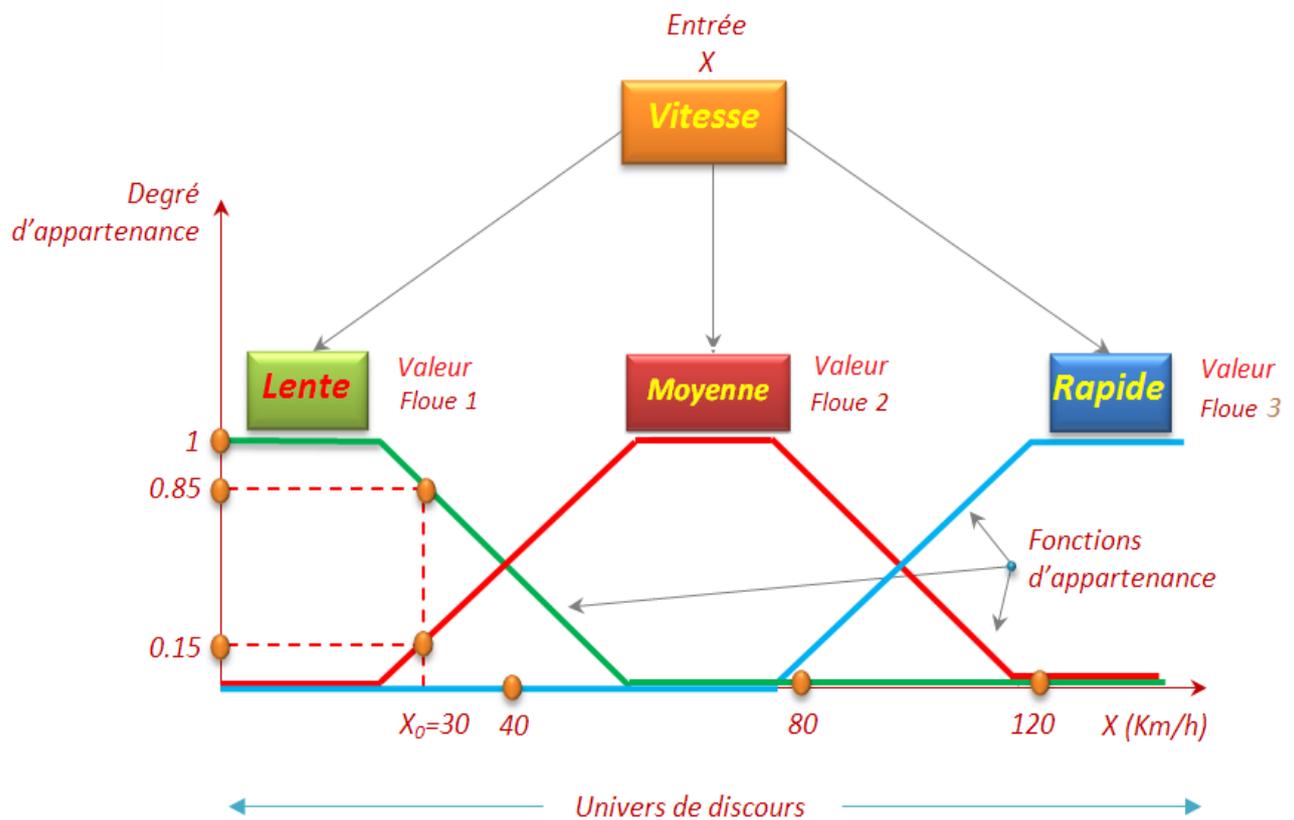


Fig. II.4 - Exemple de fuzzification.

- *Par matrice d'inférence* - elle rassemble toutes les règles d'inférence sous forme de tableau à deux dimensions. L'intersection d'une colonne et d'une ligne donne la valeur floue de la sortie définie par la règle. Il y a autant de cases que de règles [1].

II.5.3 - Défuzzification

A la fin de l'inférence, le résultat est une fonction d'appartenance qui n'est pas directement exploitable pour donner une information précise à l'opérateur ou commander un actionneur. La défuzzification permet alors de transformer ces fonctions d'appartenance en informations déterministes et donc de définir des variables de sortie numériques précises.

Plusieurs méthodes sont proposées dans la littérature et aucune ne peut être jugée meilleure que les autres. Le choix de la méthode à utiliser dépend essentiellement de l'expérience du concepteur et du cahier des charges. La priorité peut donc être donnée d'un côté à la réduction du temps de calcul par la diminution de la complexité du système et de l'autre à la compatibilité de l'application et la crédibilité des résultats du point de vue expert [2].

Les différentes méthodes de défuzzification, comme est indiqué sur *Fig. II.5*, sont: centre de gravité (COG), moyenne des maxima (MOM), bissectrice de la zone (BIS), plus grand maximum (LOM), plus petit maximum (SOM). En plus, la moyenne pondérée est utilisée dans certains types de systèmes flous [1], [2], [9].

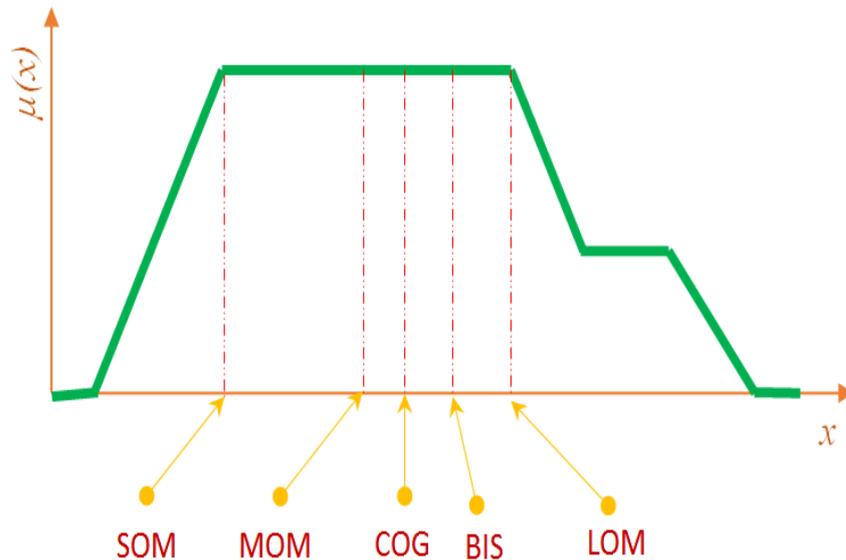


Fig. II.5 - Méthodes de défuzzification.

II.6 - Types de système d'inférence

Il existe plusieurs types de système d'inférence dépendant de la méthode de défuzzification utilisée.

Dans la méthode de *MAMDANI* (*Fig. II.6*), les prémisses et les conclusions des règles sont floues. L'inférence floue est réalisée à l'aide de l'opérateur '*min*' (minimum) ou '*prod*' (produit) et l'agrégation des règles à l'aide de l'opérateur '*max*' (maximum). La défuzzification est généralement effectuée par la méthode du centre de gravité. Pour *Tsukamoto*, la fonction d'appartenance de chaque conclusion est une fonction monotone croissante. L'inférence floue est réalisée par l'opérateur '*min*' ou '*prod*'. La sortie numérique est une moyenne pondérée des valeurs issues des fonctions monotones. La méthode de *Takagi-Sugeno* est très efficace du point de vue calcul, car elle privilégie l'analyse mathématique, elle fonctionne bien avec les techniques linéaires, l'optimisation et les techniques dites adaptatives. L'inférence floue est réalisée avec l'opérateur '*min*' ou '*prod*'. Les conclusions des règles sont des valeurs numériques calculées à partir des entrées et la valeur finale est obtenue en effectuant une moyenne pondérée des conclusions [10].

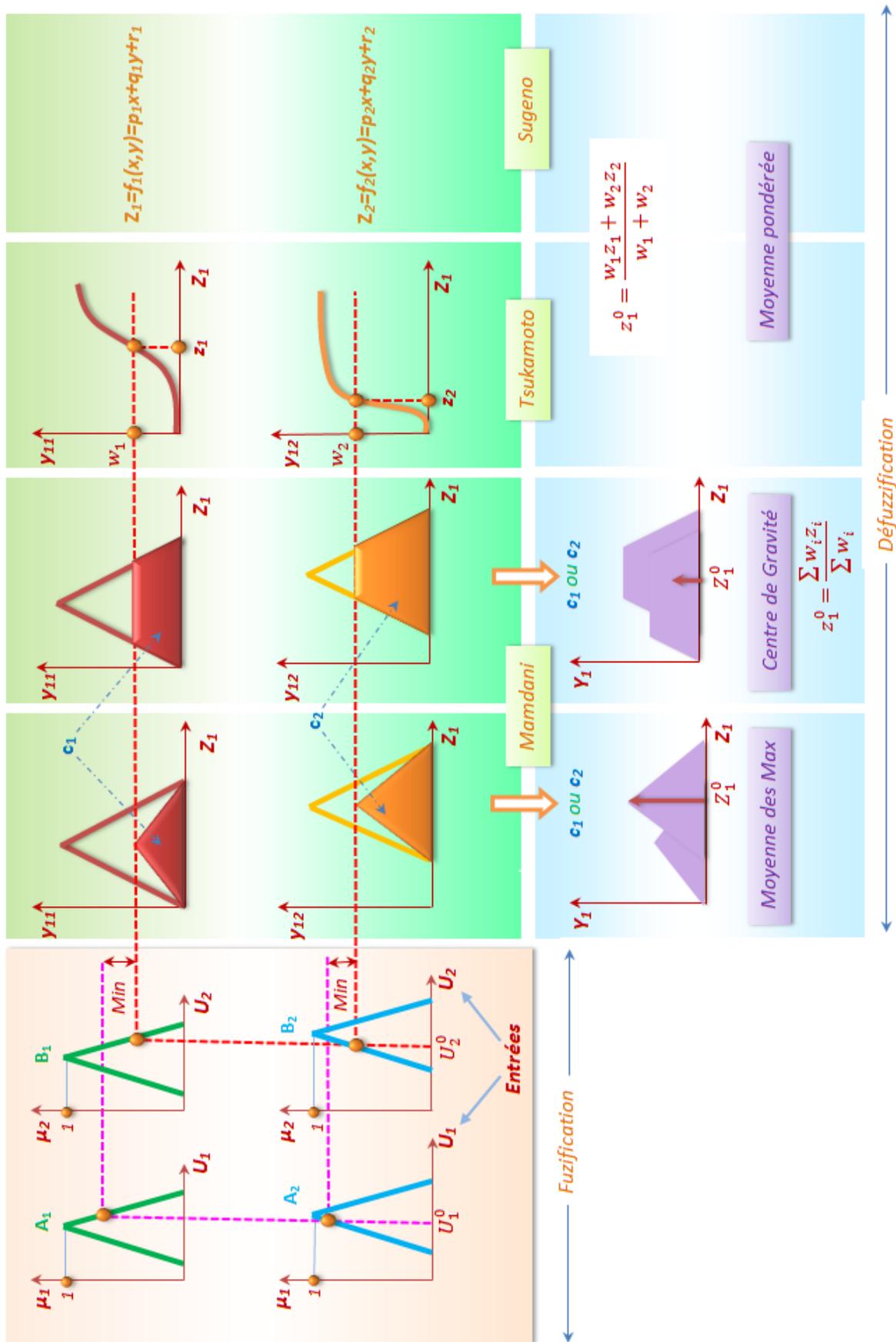


Fig. II.6 - Types des systèmes flous.

La méthode de *Sugeno* offre un calcul simple, un bon fonctionnement avec les techniques linéaires et les techniques d'optimisation et d'adaptation, tandis que la méthode de *Mamdani* est simple à comprendre et bien faite pour interagir avec l'humain [9].

II.7 – Création du système flou

La création d'un système d'inférence, comme est présenté sur *Fig. II.7* passe par les étapes:

- 1- *Choix* de la structure à adopter (*Mamdani*, *Tsukamoto* ou *Sugeno*), les entrées/sorties, l'univers de discours, les opérateurs flous, la méthode de défuzzification, les valeurs floues et leurs fonctions d'appartenance pour chaque entrée ;
- 2- Présentation de la base des règles ;
- 3- Test du fonctionnement du système.

Le logiciel Matlab offre une boîte à outils très riche pour le développement des systèmes flous, avec une application pour le développement graphique qui simplifie la tâche de conception.

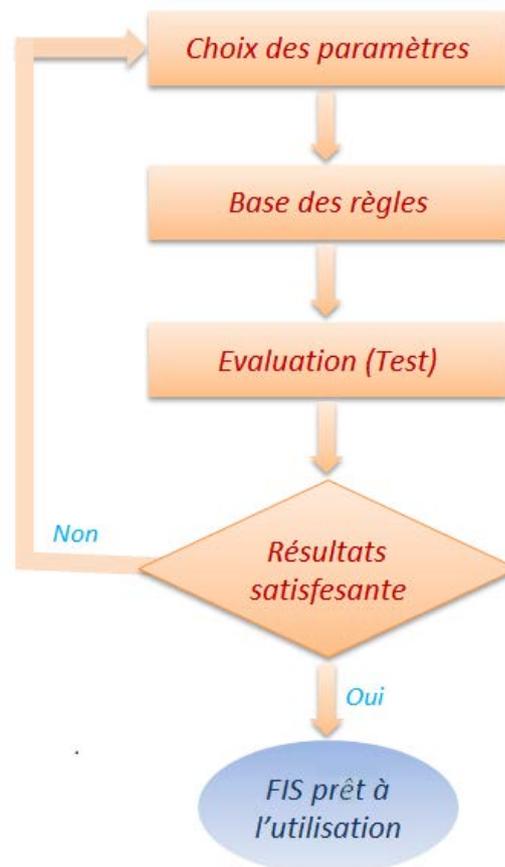


Fig. II.7 - Etapes de création d'un système d'inférence floue (SIF).

11.8 - Application de la logique floue

La logique floue est utilisée dans plusieurs domaines très différents, comme les sciences médicales, la commande des systèmes, le diagnostic industriel et l'économie.

11.9 - Conclusion

Les outils fournis par la logique floue permettent une modélisation des phénomènes pouvant en un certain sens s'approcher du raisonnement humain. Le fait de transcender le tout ou rien des ordinateurs introduit une souplesse faisant la puissance des outils flous dans de nombreux domaines.

La logique floue présente les avantages suivants:

- Le modèle mathématique non requis;
- La connaissance antérieure sur les règles peut être utilisée;
- Une interprétation et implémentation simple.

D'un autre côté, ses inconvénients sont:

- Les règles doivent être disponibles;
- Ne peut pas apprendre (Adaptation difficile au changement de l'environnement);
- Aucune méthode formelle pour l'ajustement.

Pour pallier à ces inconvénients, les réseaux de neurones qui sont complémentaires (puissance de calcul et apprentissage des RNA et l'interprétabilité de la logique floue) sont utilisés. Le chapitre suivant traite la combinaison de ces deux techniques.

Chapitre III

Systemes

Neuro - Flous

III.1 - Introduction

Les systèmes hybrides qui combinent la logique floue, les réseaux neuronaux, les algorithmes génétiques, et les systèmes experts ont prouvé leur efficacité dans une variété de problèmes du monde réel et dans l'industrie. Par exemple, pendant que les réseaux neuronaux sont intéressants pour reconnaître des modèles, ils ne peuvent pas expliquer la manière comment ils atteignent leurs décisions. De même pour les systèmes de la logique floue qui utilisent l'information imprécise dans son raisonnement, ils exigent la disponibilité auparavant des règles qui sont élaborées, généralement, par les connaisseurs (experts) du processus qui est une tâche difficile à fournir. Ces limites ont été une raison derrière la création de systèmes hybrides intelligents où deux ou plus de techniques sont combinées dans une manière à vaincre les limitations d'une seule technique [1].

Dans ce chapitre, les principaux aspects des systèmes neuro-flous seront présentés.

III.2 - Définition

Les réseaux neuro-flous sont nés de l'association des réseaux de neurones avec la logique floue, de manière à tirer profit des avantages de chacune de ces deux techniques. Leur principale propriété est la capacité à traiter dans un même outil des connaissances numériques et symboliques d'un système [12]. Ce sont des réseaux neuronaux dont l'élément de base est un opérateur flou qui est souvent appelé le neurone flou. Un neurone flou (NF) est un système mathématique possédant la même architecture que son homologue le neurone formel. La différence est que quelques-uns ou tous ses paramètres sont décrits par la logique floue.

III.3 - Combinaison des réseaux de neurones et de la logique floue

Les systèmes neuro-flous peuvent être classés en trois catégories: systèmes neuro-flous coopératifs, systèmes neuro-flous concurrents et systèmes neuro-flous hybrides.

III.3.1 - Systèmes neuro-flous coopératifs

L'un des deux systèmes, flou ou neuronal, intervient une seule fois pour effectuer une tâche spécifique. Les résultats obtenus du premier sont injectés et utilisés par le second système [2].

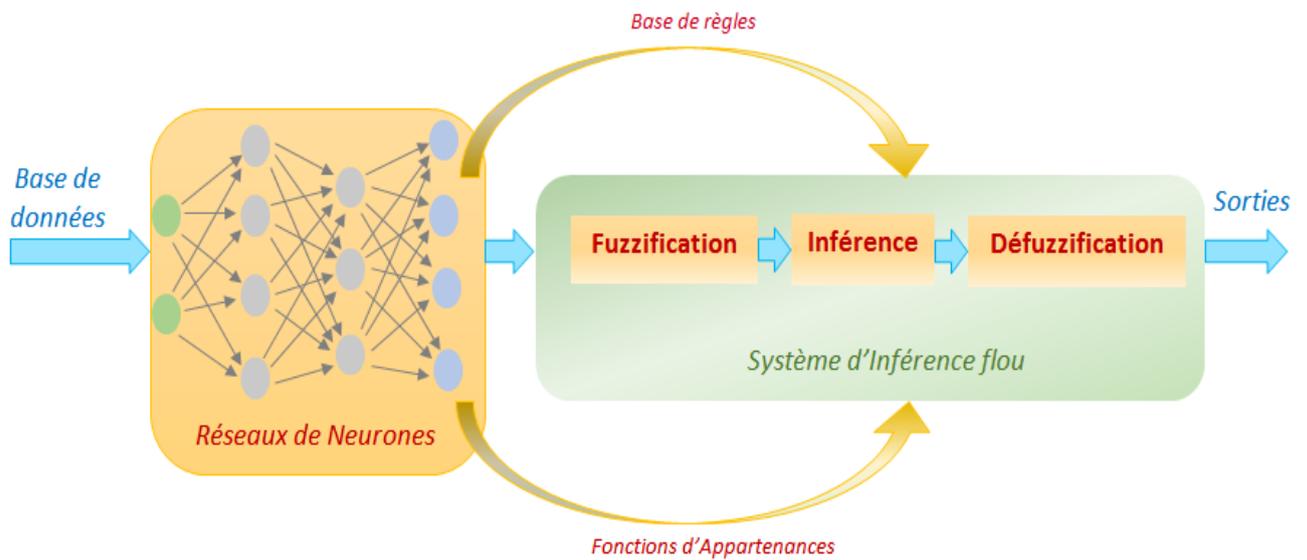


Fig. III.1 - Système neuro-flou coopératif.

III.3.2 - Systèmes neuro-flous concurrents

L'un des deux systèmes, flou ou neuronal, intervient continuellement pour effectuer une tâche spécifique. Les résultats obtenus du premier sont injectés et utilisés par le second système [2].

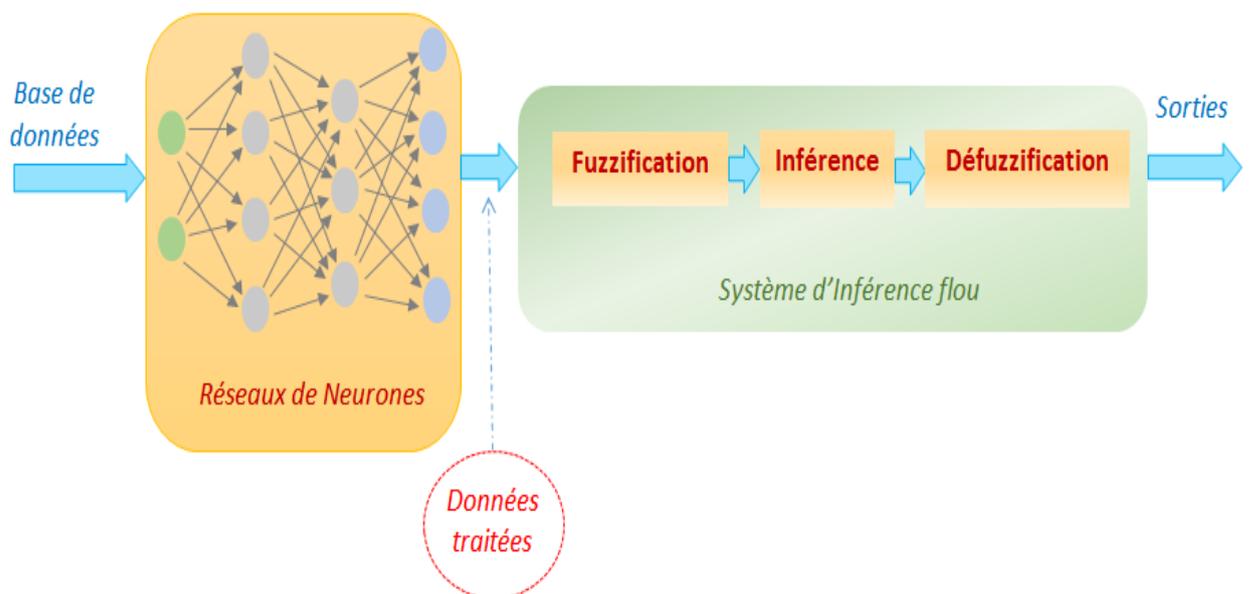


Fig. III.2 - Système neuro-flou concurrent.

En général, les réseaux de neurones se placent soit en amont pour le traitement des entrées, soit en aval pour la génération des sorties.

III.3.3 - Systèmes neuro-flous hybrides

Un système neuro-flou hybride est une combinaison homogène d'un réseau neuronal et un système d'inférence flou. Il peut être soit, un réseau neuronal mené des paramètres flous nommé *réseau de neurones flou* ; ou un système flou dont les paramètres sont ajustés par apprentissage grâce au réseau de neurone nommé *système flou neuronal* indiqué par Fig. III.3 [1].

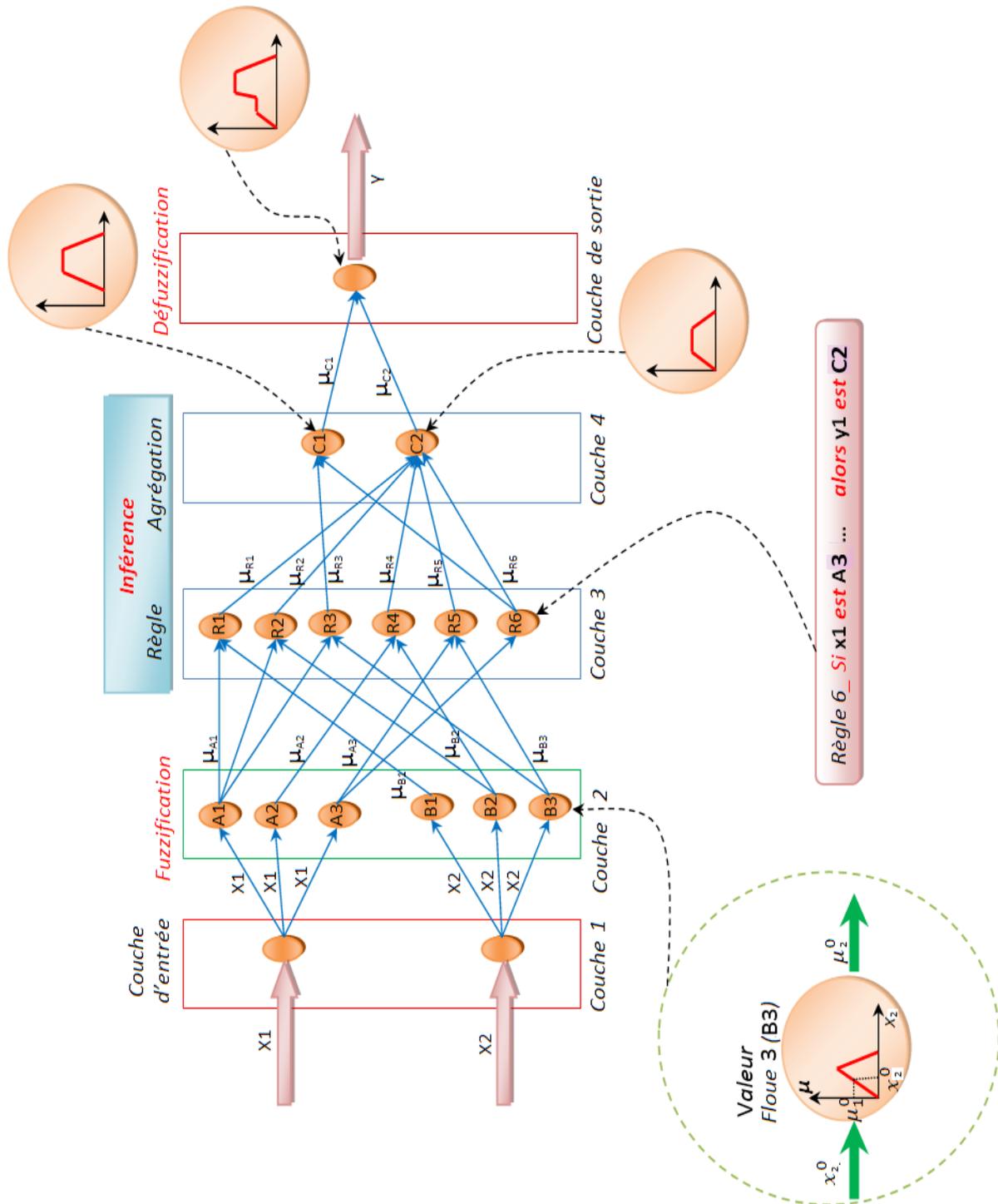


Fig III.3 - Structure générale d'un système flou neuronal (hybride).

Pour les systèmes flous neuronaux, l'approche de *Takagi-Sugeno* (Fig.A.I.2) est appliquée pour la génération des résidus, par contre l'approche de *Mamdani* (Fig.A.I.1) est plus convenable pour la classification [13].

De plus, Il existe une variété d'architectures liées à des applications spécifiques, tels que l'ANFIS pour l'approximation des fonctions, FALCON pour le contrôle et NEFCLASS pour la classification.

III.4 - Apprentissage des systèmes neuro-flous

L'apprentissage des neurones flous implique le changement des relations entre les synapses d'entrées et la sortie en agissant sur le poids des entrées. L'adaptation suit en général, la méthode de descente de gradient [2].

Le synoptique de l'apprentissage des systèmes neuro-flous est illustré sur Fig. III.4 et Fig. A.I.4.

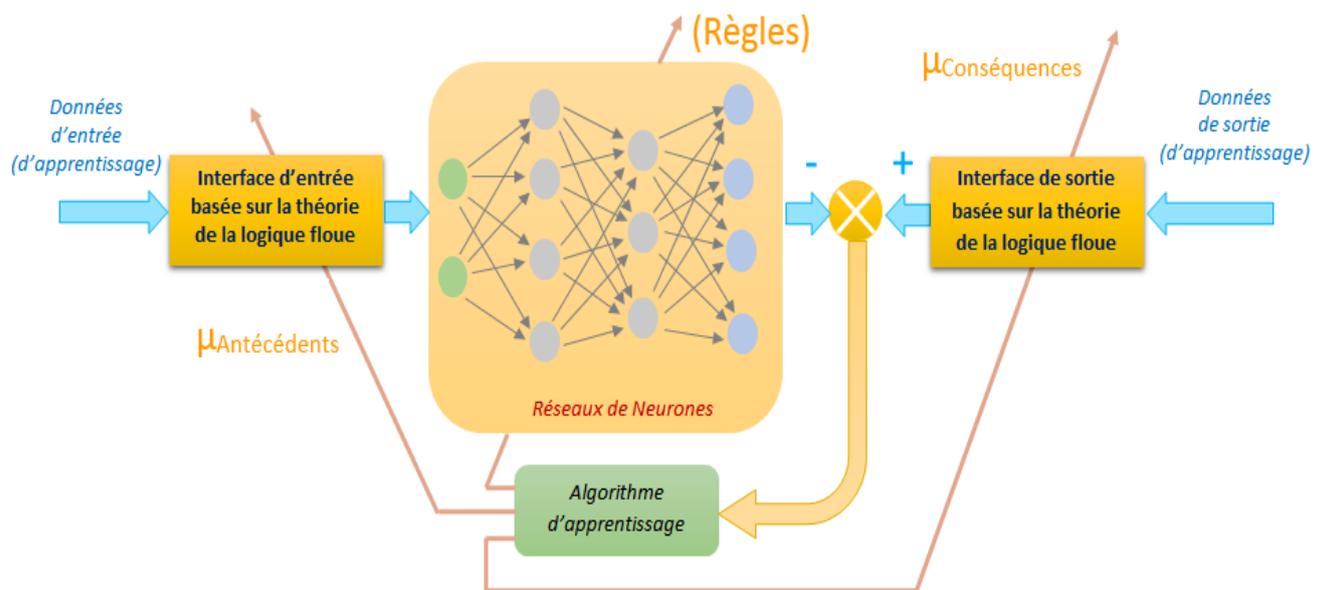


Fig. III.4 - Synoptique de l'apprentissage des systèmes neuro-flous.

En général, l'apprentissage se divise en deux phases :

- 1- Le réglage des règles floues ;
- 2- Le réglage des paramètres des fonctions d'appartenance.

Le réglage des règles se fait par l'ajustement des intervalles des ensembles flous afin d'avoir la conclusion désirée (Fig. III.5).

Le réglage des paramètres des fonctions d'appartenances veut dire la modification de ses dimensions, la *Fig. III.6* illustre les modifications possibles pour une fonction d'appartenance de type triangulaire.

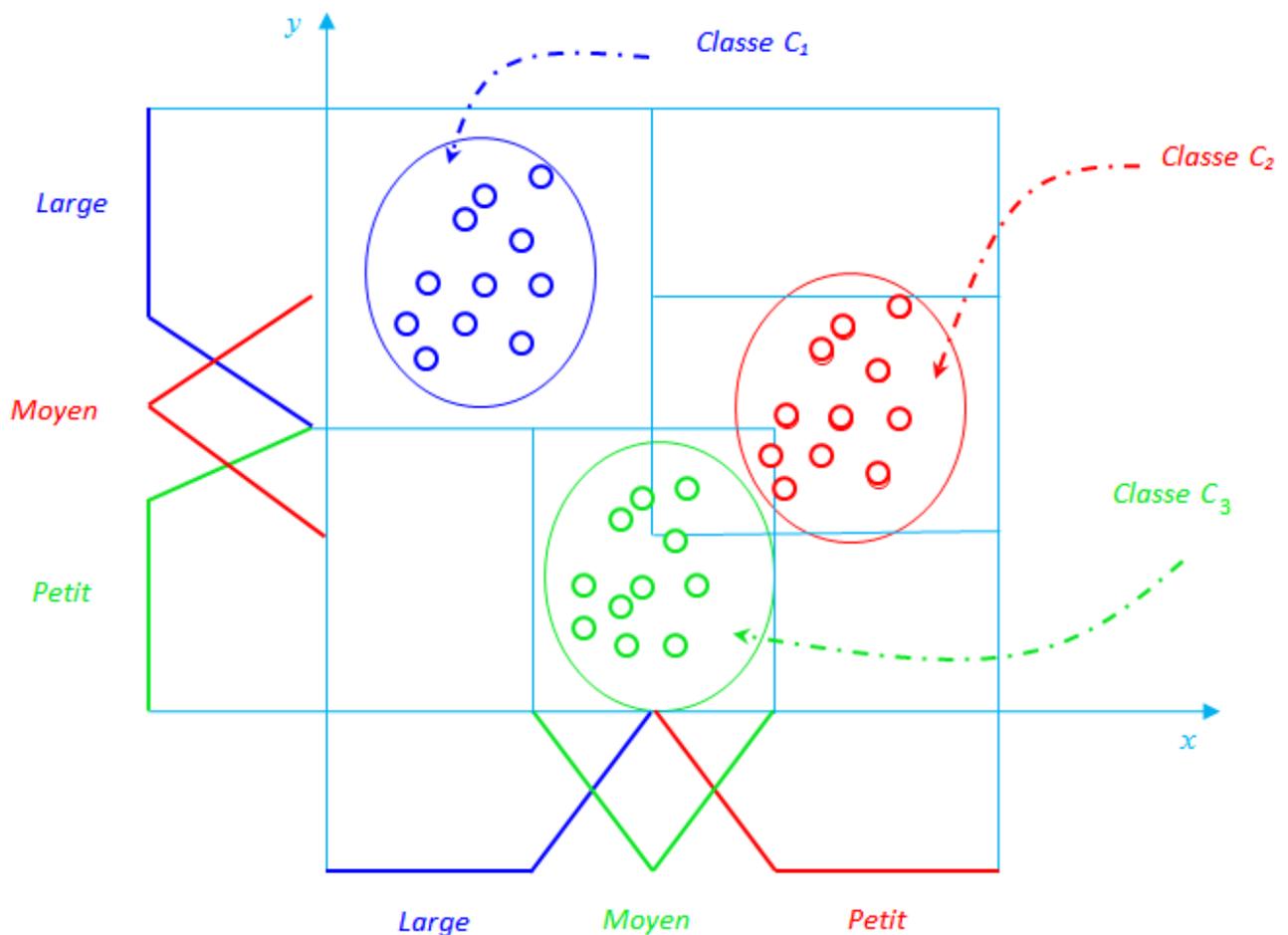


Fig.III.5 - Ajustement des intervalles des ensembles flous.

III.5 - Modèles des systèmes neuro-flous hybride

Il existe un grand nombre de modèles [14]. Chacune possède des caractéristiques bien déterminées et est orientée vers un domaine d'application spécifique (ANFIS, NEFCLASS). Ils existent d'autres modèles des systèmes neuro-flous tels que :

- NEFCON (Neuro-Fuzzy Controller)
- NEFPROX (Neuro Fuzzy function apPROXimator)
- GARIC (Generalized Approximate Reasoning based Intelligent Control)
- EFuNN (Evolving Fuzzy Neural Network)
- FALCON (The Fuzzy Adaptive Learning Control Network)

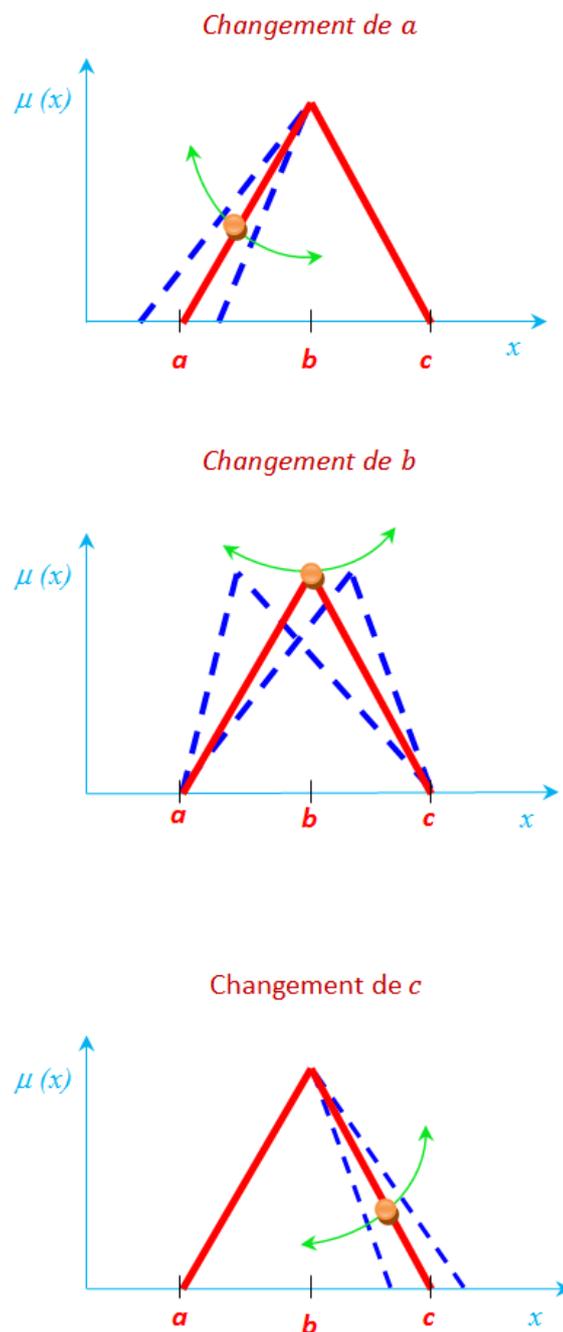


Fig. III.6 - Réglage des paramètres de la fonction d'appartenance triangulaire.

III.5.1 - ANFIS

a. Définition

Le système hybride ANFIS (Adaptive Network Based Inference System) est un système d'inférence neuro-flou adaptatif qui consiste à utiliser un réseau de neurone de

type CMP à 5 couches pour laquelle chaque couche est destinée à la réalisation d'une étape d'un système d'inférence floue de type *Takagi Sugeno*. Il est proposé par *Jang [3]*.

b. Architecture

Il est constitué de cinq couches comme illustré sur *Fig. III.7* :

1. *La première couche* : Assigne chaque variable à ses fonctions d'appartenance (Fuzzification), par la fonction 'gaussienne Bell'

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x-c_i}{a_i} \right|^{2b_i}} \quad (III.1)$$

Sortie : degré d'appartenance de l'entrée

2. *La deuxième couche* : Utilise la norme *T* (généralement 'prod') pour combiné les antécédents des règles.

Sortie : Force de la règle (*Firing Strength*)

3. *La troisième couche* : Fait la *normalisation* des forces de chaque règle.

Sortie : Force de la règle normalisée.

4. *La quatrième couche* : Détermine la conséquence de chaque règle par une *fonction polynomiale*.

Sortie : Evaluation du polynôme.

5. *La cinquième couche* : Calcule la sortie globale (*Défuzzification par la moyenne pondérée*).

Sortie : Evaluation de la sortie.

c. Apprentissage

Pour la détermination de ses paramètres, l'ANFIS utilise l'apprentissage par *rétro-propagation* pour les fonctions d'appartenance de chaque entrée, et la *méthode des moindres carrés* pour les conséquences. Le rôle du réseau de neurones est de déterminer les paramètres des fonctions d'appartenance des antécédents et conséquences.

Le *Tab III.1* donne le nombre de paramètres à calculer par couche.

où *P* est le nombre de fonctions d'appartenance pour chaque variable d'entrée, le nombre 3 indique la fonction d'appartenance de Gauss-Bell (a_i, b_i, c_i).

Ce modèle donne de très bons résultats dans les domaines tels que le traitement du signal, l'approximation non linéaire et la commande dynamique.

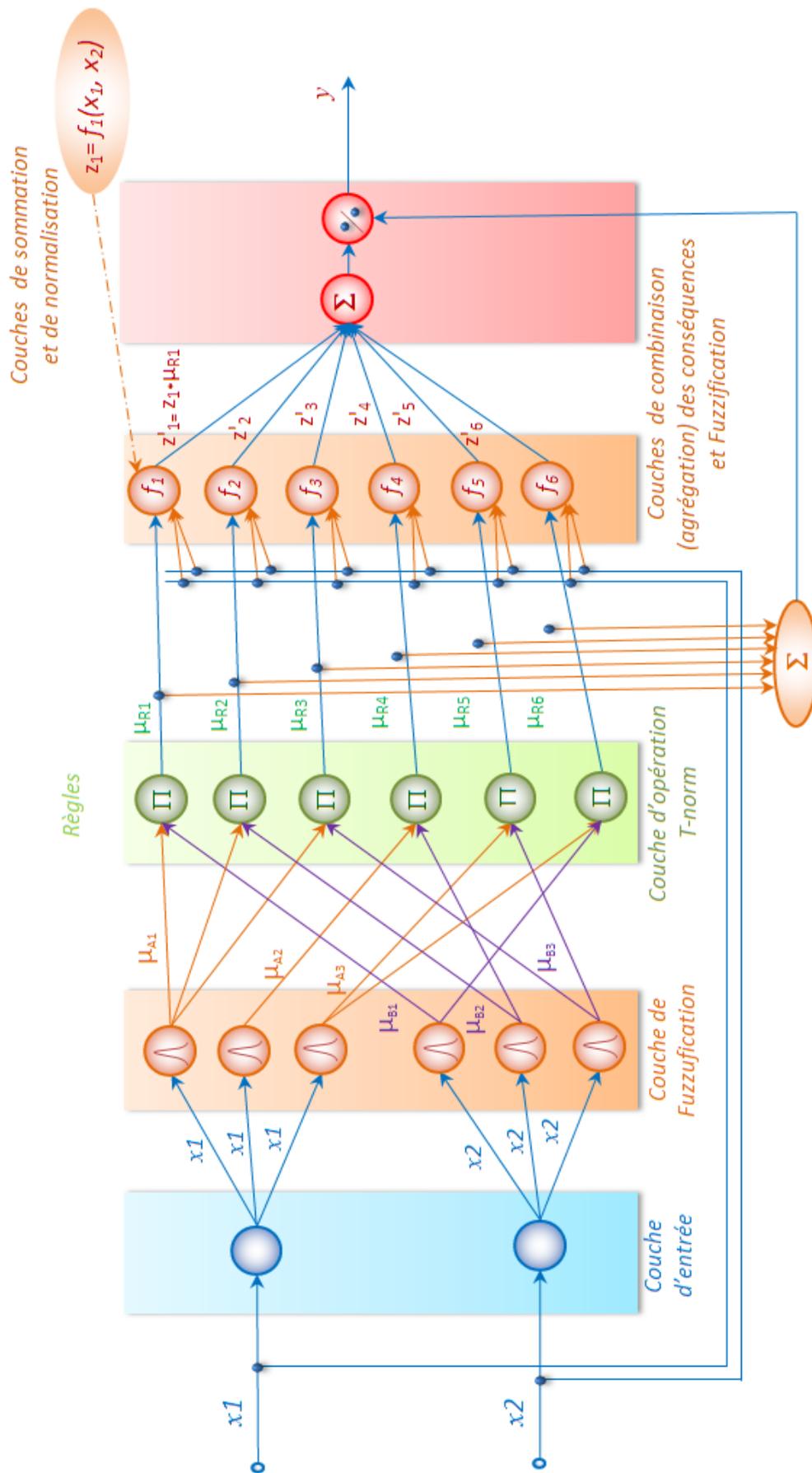


Fig. III.7 - Architecture du modèle ANFIS.

<i>Couche N°</i>	<i>Type de la couche</i>	<i>Nombre de neurones</i>	<i>Nombre de paramètres</i>
<i>L0</i>	<i>Entrées</i>	<i>n</i>	<i>0</i>
<i>L1</i>	<i>Valeurs floues</i>	<i>p*n</i>	3 <i>*(p*n)</i>
<i>L2</i>	<i>Règles</i>	<i>pⁿ</i>	<i>0</i>
<i>L3</i>	<i>Normalisation</i>	<i>pⁿ</i>	<i>0</i>
<i>L4</i>	<i>Fonction Linéaire</i>	<i>pⁿ</i>	<i>(n+1)* pⁿ</i>
<i>L5</i>	<i>Somme</i>	<i>1</i>	<i>0</i>

Tab III.1. Nombre de paramètres par couche.

III.5.2 - NEFCLASS

Ce modèle est utilisé généralement en classification. Il est constitué de 3 couches : une couche d'entrée avec les fonctions d'appartenance, une couche cachée représentée par des règles et une couche de sortie définissant les classes. Ce modèle est facile à mettre en application. Il évite l'étape de défuzzification, tout en étant précis dans le résultat final, avec une rapidité bien supérieure aux autres modèles [1], [3].

Ce modèle va être détaillé dans le dernier chapitre.

III.6 - Création du système neuro-flou

Les étapes de création d'un système neuro-flous (Fig. III.9) diffèrent selon l'application et les connaissances présentes, mais de façon générale, on peut les résumer en cinq étapes :

- Définition des formes initiales des fonctions d'appartenance pour les entrées et les sorties ;
- Détermination de la base initiale des règles floues ;

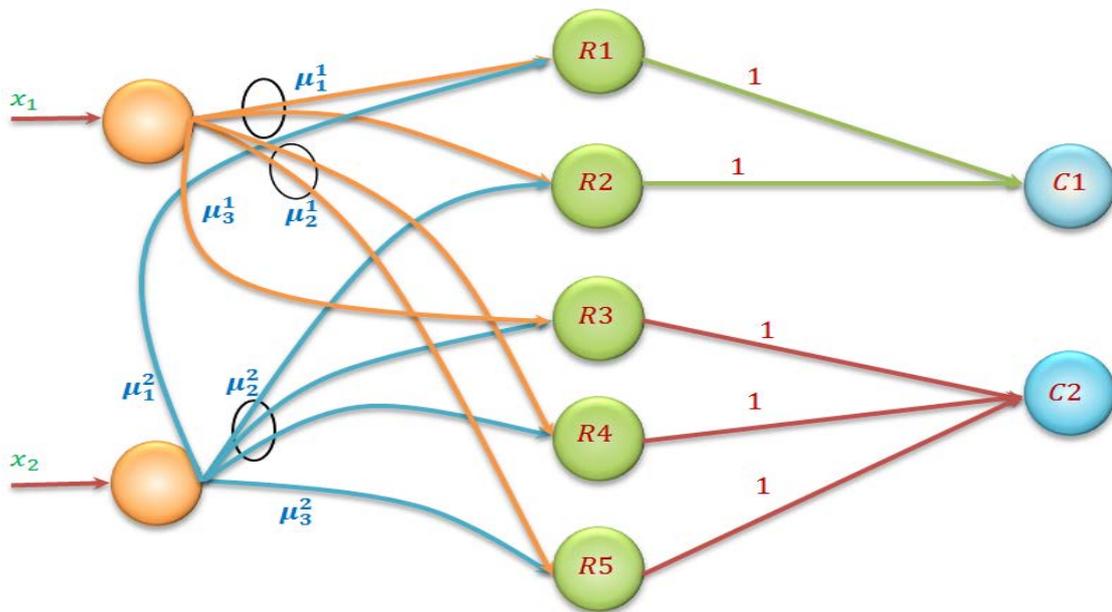


Fig. III.8 - Architecture du modèle NEFCLASS.

- Réglage des règles et les fonctions d'appartenance de manière à avoir la meilleure approximation des règles d'après les données d'apprentissage, et aussi la meilleure généralisation ;
- Tester le système face à des données de test ;
- Elimination des règles faibles dans le but d'améliorer l'interprétation du système, en préservant ses performances en matière de tolérance.

Si les résultats ne sont pas satisfaisants, il est toujours possible de refaire l'apprentissage en modifiant les paramètres de l'algorithme.

III.7 - Applications des systèmes neuro-flous

Les premières applications des systèmes neuro-flous sont apparues sur le marché (Japonais et Coréen) en 1991. Quelques exemples incluent des appareils à conditionner de l'air, moquettes, ventilateurs électriques, et traitements de texte [1].

Les systèmes neuro-flous sont utilisés aussi dans l'industrie pour le diagnostic et la prise de décision, où ils sont employés comme des détecteurs ou des classificateurs de pannes.

De plus, ils sont utilisés aussi dans les domaines (médecine, économie, commande des systèmes, mécanique, physique, chimie, etc ...).

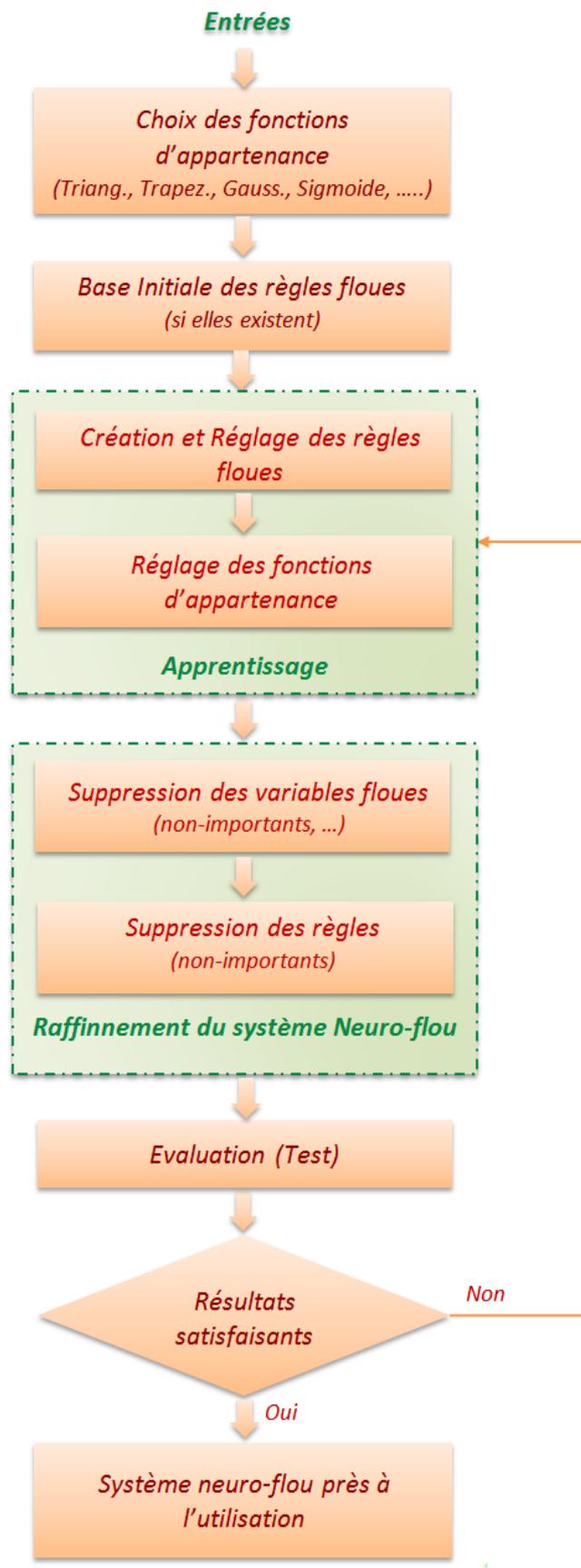


Fig. III.9 - Etapes de création d'un système neuro-flou

III.8 - Conclusion

En guise de conclusion, les systèmes neuro-flous présentent plusieurs avantages qui les rendent attractifs, à savoir :

- possibilité d'utiliser la base des règles ;
- l'apprentissage peut démarrer de zéro ou avec une base de règles initiale ;
- simples à interpréter et implémenter ;
- flexibilité et rapidité de calcul et généralisation des connaissances.

Ce chapitre contient un rappel sur les systèmes neuro-flous, leurs définitions, les combinaisons neuro-floues possibles, les modèles les plus utilisées ainsi que leurs applications.

Chapitre IV

***Classification
des données***

IV.1 - Introduction

Auparavant, la classification des événements se faisait par analyse manuelle basée sur l'inspection visuelle et le calcul des caractéristiques. Ce processus est subjectif et demande un travail dur, un temps significatif et une expérience considérable.

En plus, vu le grand volume de données reçues continuellement, le considérable traitement quotidien fait par les analystes est stressant et laborieux. Donc, construire des classificateurs automatiques et fiables est crucial afin de réduire considérablement l'effort demandé aussi bien de faire une classification rapide et plus objective.

La classification de formes est un élément essentiel dans beaucoup de techniques d'ingénierie telle que les applications des sonars, des radars, cardiologiques, sismiques et de diagnostic, qui exigent le pouvoir de classifier exactement les différentes sortes de situations. Le contrôle, le dépistage (*tracking*) et des systèmes de prédiction emploient souvent des classificateurs pour déterminer des rapports d'entrée-sortie.

À cause du grand nombre de domaines d'application, la classification de forme a été traitée dans beaucoup d'articles et ouvrages [3], [13].

La flexibilité du modèle de classification, adaptabilité aux différents problèmes et un paramétrage facile sont des caractéristiques nécessaires pour ces algorithmes. Cependant dans quelques applications, comme les systèmes d'aide à la décision médicale, la possibilité d'expliquer les raisons et les règles utilisées par le classificateur est très importante.

Dans ce chapitre, la classification sera définie comme étant une forme de la reconnaissance de forme, en citant les méthodes de classification les plus utilisées.

IV.2 – Définition

La classification est l'une des tâches de prise de décision les plus employées. Elle constitue le principe de base de plusieurs systèmes d'aide au diagnostic.

La classification est une technique d'apprentissage supervisée qui consiste à assigner des objets d'un ensemble à des catégories, groupes ou classes prédéterminés(es) basée sur un nombre d'attributs observés, liés à ces objets. Les objets sont alors classifiés suivant qu'ils vérifient ou non ces conditions ou propriétés [13].

L'algorithme qui implémente une classification est connu sous le nom de classificateur. Le terme "classificateur" parfois se réfère aussi à la fonction mathématique, mise en œuvre par l'algorithme de classification, qui décrit la liaison d'une donnée d'entrée à une catégorie.

Les méthodes de classification sont appropriées pour résoudre le problème de diagnostic de défauts, particulièrement si on ne connaît pas le modèle du processus. Ils sont utilisés afin d'instaurer des interactions logiques entre les symptômes observés (effets) et les défauts inconnus (causes).

IV.3 - Catégorisation

Il existe deux principales catégories de classificateurs : linéaire ou non-linéaire ; Une classification linéaire, nommée parfois discrimination linéaire, est basée sur la combinaison linéaire des caractéristiques. Cette combinaison est une droite en cas de deux caractéristiques. Dans le cas où le nombre de caractéristiques dépasse deux, on aura un hyperplan d'équation *Eq.IV.1*.

$$w_1 f_1 + w_2 f_2 + w_3 f_3 + \dots + w_d f_d + w_0 = 0 \quad (IV.1)$$

où d est le nombre des caractéristiques f .

Chaque coefficient w_i est considéré comme poids de la caractéristique correspondante. Les caractéristiques importantes sont liées avec des poids élevés; par contre les caractéristiques les moins importantes ne sont pas utilisées.

Géométriquement, le vecteur des poids W est normal à l'hyperplan qui sépare les classes. La pente est déterminée par w_1, w_2, \dots, w_d et la position par le biais w_0 .

Les classificateurs linéaires sont souvent employés dans les situations où une faible complexité est souhaitée, car ce sont les classificateurs les plus simples et donc les plus rapides.

Sur *Fig. IV.2*, il existe deux classes d'objets séparées par deux classificateurs c_1 et c_2 utilisant deux caractéristiques, f_1 et f_2 . c_1 est linéaire tandis que c_2 est non-linéaire et l'effet de f_2 s'affaiblit lorsque f_1 augmente.

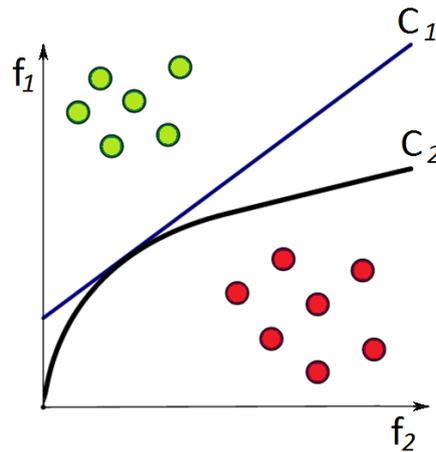


Fig. IV.1 – Classification linéaire et non-linéaire

IV.4 - Méthodes

En général, suivant les objets considérés (les maladies en médecine, les défauts dans un système, etc.), deux principales catégories de méthodes existent : *intelligence artificielle* et *statistiques* nommées parfois *classiques*.

IV.4.1 - Méthodes statistiques

Les méthodes traditionnelles ou statistiques de la classification sont basées sur la théorie de décision bayésienne. Un modèle de probabilité est assumé pour calculer la probabilité à posteriori sur laquelle la décision de classification est faite. La limitation principale de ces méthodes statistiques réside dans le fait qu'ils travaillent bien seulement quand les suppositions sont correctes. L'efficacité de ces méthodes dépend en grande partie sur les suppositions diverses ou les conditions sous lesquelles les modèles sont développés [13].

Les modèles statistiques les plus utilisés sont : les *plus proches voisins*, *arbre de décision* et les *séparateurs à vaste marge*.

IV.4.1.1 - Plus proches voisins (PPV)

La méthode de plus proches voisins (*ppv*), est la plus simple des méthodes de classification statistique. Elle est non-paramétrique et de type géométrique, qui classifie les objets en se basant sur les exemples d'apprentissage les plus proches en distance dans l'espace des paramètres.

Contrairement à d'autres méthodes statistiques, cette approche ne nécessite aucun apprentissage (c.-à-d. pas de modèle à ajuster) [13].

Le classificateur k -plus proches voisins (k -ppv), assigne un point x à une classe particulière basé sur un vote de majorité parmi les classes des k -plus proches points à x . La distance euclidienne est généralement la plus employée. La valeur de k dépend de l'ensemble de données et d'habitude choisie expérimentalement. Une valeur trop basse de k rend le classificateur sensible au bruit des données et une valeur trop élevée permet au voisinage de s'étendre parmi des objets de différentes classes.

Au niveau de l'exemple donné sur *Fig. IV.3*, le cercle vert doit être classé, soit dans la première classe des carrés bleus ou bien dans la deuxième classe des triangles rouges. Si $k = 3$ (le cercle en ligne continue) le cercle vert est assigné à la deuxième classe parce qu'il y a 2 triangles contre un carré à l'intérieur. Si $k = 5$ (le cercle en ligne discontinue) le cercle vert est assigné à la première classe (3 carrés contre 2 triangles à l'intérieur du cercle).

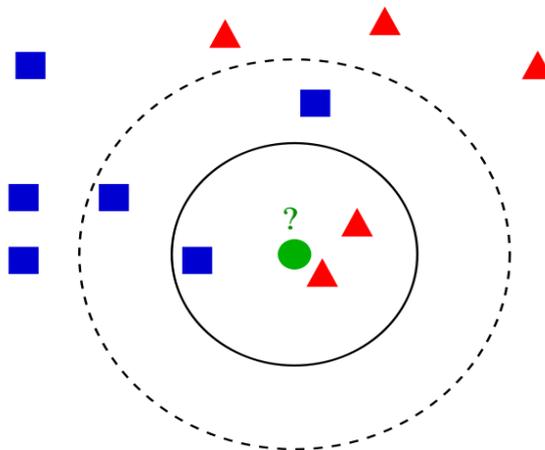


Fig.IV.2 - Exemple de classification k -ppv.

Cette méthode possède l'avantage d'être plus facile à implémenter. Elle peut donner de bons résultats si les caractéristiques sont attentivement choisies. Cependant, il existe plusieurs inconvénients :

- cette méthode ne simplifie pas la distribution des objets dans l'espace de paramètres vers un ensemble de paramètres compréhensible.
- Elle est plus lente lorsque l'ensemble d'apprentissage possède plus d'exemples.

IV.4.1.2 - Arbres de décision

L'arbre de décision est un outil d'aide à la décision qui fait partie des méthodes d'apprentissage supervisé.

Il permet de classer les objets par division en sous-classes. Il est basé sur le regroupement de fonctions locales structurées en forme graphique d'un arbre de façon à faire apparaître à l'extrémité de chaque branche (ou feuille) les différents résultats possibles en fonction des décisions prises à chaque étape. A chaque nœud, un seul paramètre est comparé à une certaine constante. Dans les arbres de décision binaires, si la valeur de la caractéristique est supérieure au seuil, la branche de droite de l'arbre est prise ; si la valeur est plus petite la branche de gauche est prise. Après une série de tests, on atteint à ce que tous les objets sont étiquetés selon leurs classes d'appartenance. *L'avantage* de cette méthode par rapport aux autres approches est l'efficacité, la souplesse aux modifications (rajout ou élimination de branches), la lisibilité (interprétable) et la simplicité en structure qui utilise seulement peu de paramètres pour classer les objets. Toutes ces performances avec un peu d'inconvénients (ex. la sensibilité au bruit) a fait de cette technique de classification la plus populaire [15].

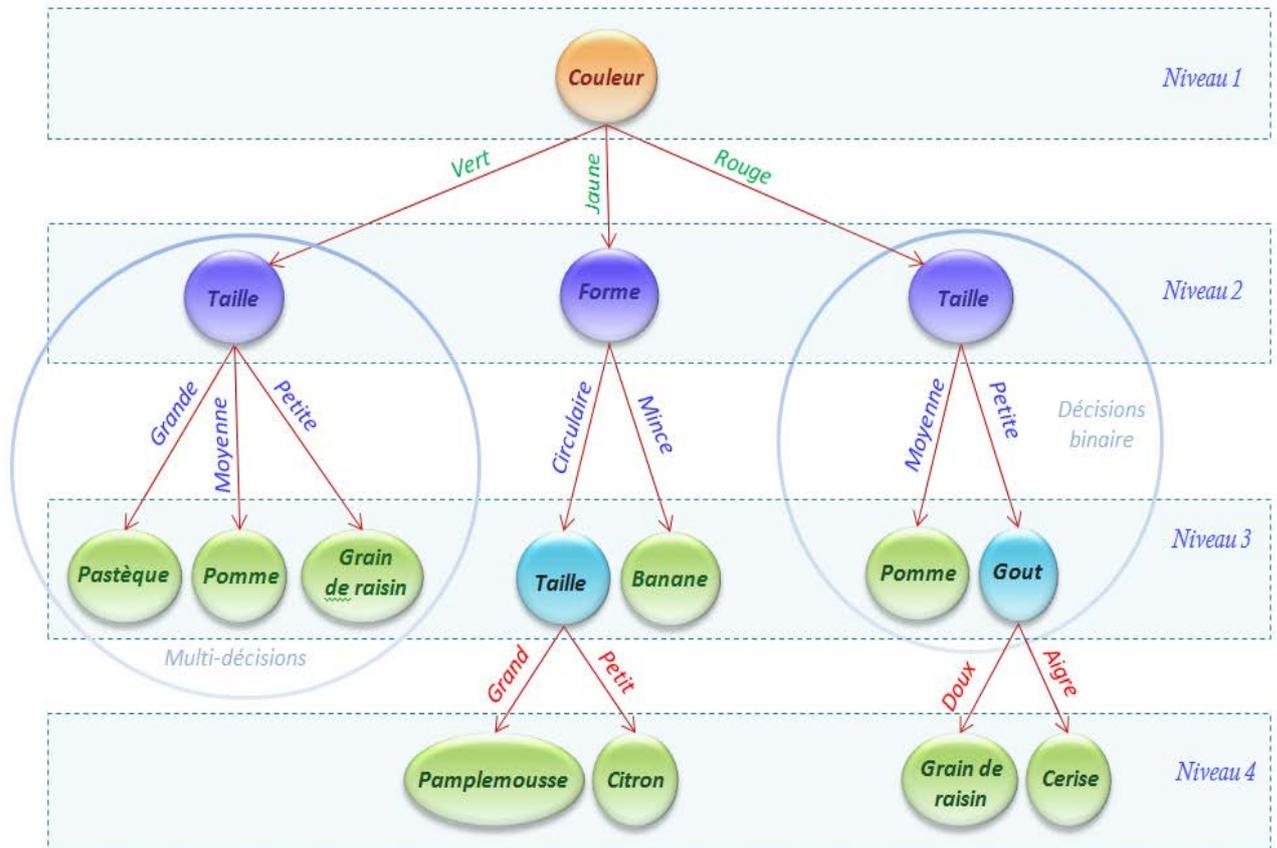


Fig.IV.3 - Arbre de décision de reconnaissance des fruits

IV.4.1.3 - Séparateurs à vaste marge (SVM)

Les séparateurs à vaste marge (SVM), ou les machines à vecteurs de support ont été développés dans les années 1990 à partir des considérations théoriques de *Vladimir Vapnik* sur le développement d'une théorie statistique de l'apprentissage.

L'objectif de cette méthode est de rechercher une manière de tri qui sépare les données de différentes classes et maximise la distance entre elles.

Pour des problèmes plus compliqués, il n'existe en général pas de séparateur linéaire. Alors les SVM dans ces conditions permettent d'effectuer la classification en construisant des règles (frontières) de décision non-linéaires.

Les classificateurs SVM reposent sur deux idées clés ; la notion de marge maximale et la notion de fonction noyau.

La *marge*, comme est indiquée sur *Fig. IV.5*, est la plus petite distance entre la frontière de séparation (hyperplan séparateur) et les échantillons d'apprentissage les plus proches. Ces derniers sont appelés vecteurs supports. La frontière de séparation est choisie de façon qu'elle maximise la marge. L'objectif alors, est de trouver cette frontière séparatrice optimale (quadratique), à partir d'un ensemble d'apprentissage.

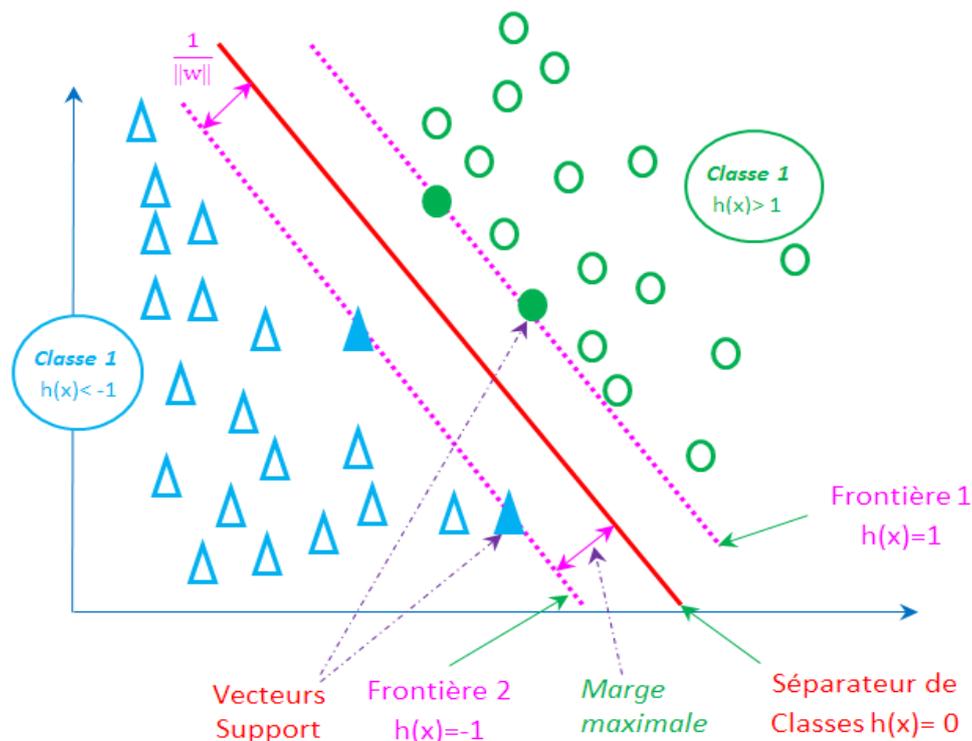


Fig. IV.4 - L'hyperplan optimal (en rouge) avec la marge maximale. Les échantillons pleins sont des vecteurs supports.

Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables, la *deuxième* idée clé des SVM est de projeter les données dans un espace de dimension supérieure, dans lequel il est probable qu'il existe un séparateur linéaire.

Ceci est réalisé grâce à une *fonction noyau* qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace.

La capacité des classes d'hyperplans séparateurs diminue lorsque leur marge augmente, et comme la marge vaut désormais $\frac{1}{\|w\|}$, il s'agit donc de maximiser $\|w\|^{-1}$.

Cette méthode possède plusieurs *avantages* :

- possède des fondements mathématiques solides ;
- les exemples de test sont comparés juste avec les supports vecteur et non pas avec tous les exemples d'apprentissage ;
- décision rapide ;
- la classification d'un nouvel exemple consiste à voir le signe de la fonction de décision $h(x)$.

D'autre part, Les SVM présentent les inconvénients suivants :

- classification binaire d'où la nécessité d'utiliser l'approche un-contre-un ;
- grande quantité d'exemples en entrées implique un calcul matriciel important et Temps de calcul élevé lors d'une régularisation des paramètres de la fonction noyau.

IV.4.2 - Intelligence artificielle

IV.4.2.1 - Réseaux de neurones

Les réseaux de neurones ont montré depuis de nombreuses années de grandes capacités à résoudre des problèmes de classification dans des différents domaines.

Dans les réseaux multicouches utilisé dans la classification, le nombre de neurones sur la couche de sortie correspond au nombre de classes attendues.

IV.4.2.2 - Classificateur floue

La logique floue est une technique efficace pour la classification. En effet, un classificateur supervisé est basé sur les concepts de base de la logique floue.

Un système de classification floue utilise les connaissances modélisées sous forme de règles floues du type « si-alors ».

IV.4.2.3 - Classificateur neuro-flou

Un classificateur neuro-flou permet de générer automatiquement une base de connaissances, pour justifier la classification. Cette base constitue un plus pour ce classificateur par rapport aux autres techniques, en effet elle permet l'interprétation des résultats après la classification.

IV.5 - Etapes de Création

La création d'un *classificateur* qui utilise les méthodes d'intelligence artificielle est essentiellement formée de *deux étapes* : *l'apprentissage* et *la généralisation*.

IV.5.1 - Apprentissage

L'apprentissage est l'étape la plus importante dans la création des classificateurs. Elle consiste à élaborer des règles de classification à partir des connaissances disponibles a priori. Autrement dit, à partir d'un ensemble de données, le classificateur apprend à faire la partition en classes de l'univers de description. Cet ensemble de données nommé parfois ensemble d'apprentissage utilisé en intelligence artificielle, programmation génétique, systèmes intelligents et statistiques. Dans tous ces domaines, l'ensemble d'apprentissage a le même rôle et souvent utilisé en conjonction avec l'ensemble de test.

L'ensemble d'apprentissage doit contenir la liste d'objets avec des classifications connues. Idéalement, cet ensemble contient beaucoup d'échantillons incluant les types communs et rares d'objets.

Si un problème possède seulement quelques importantes caractéristiques, alors la classification est souvent une simple tâche. Mais cette tâche devient très difficile lorsqu'il s'agit de plusieurs paramètres à considérer.

IV.5.2 - Généralisation

Dans la *deuxième étape*, une procédure de *généralisation* est introduite. Il s'agit en effet d'extraire une règle générale qui devra classer correctement les futurs objets. Il est conseillé d'éviter d'effectuer l'apprentissage et le test avec le même ensemble d'objets.

IV.6 - Chaîne de classification

Le synoptique de la chaîne de classification est donnée sur *Fig. IV.6*. Il est composé dans la plupart des cas de trois étages ; *prétraitement*, *extraction de caractéristiques* et *classification*. Notons que l'ensemble de ces trois étages existent dans les phases de création du classificateur (*apprentissage et test*).

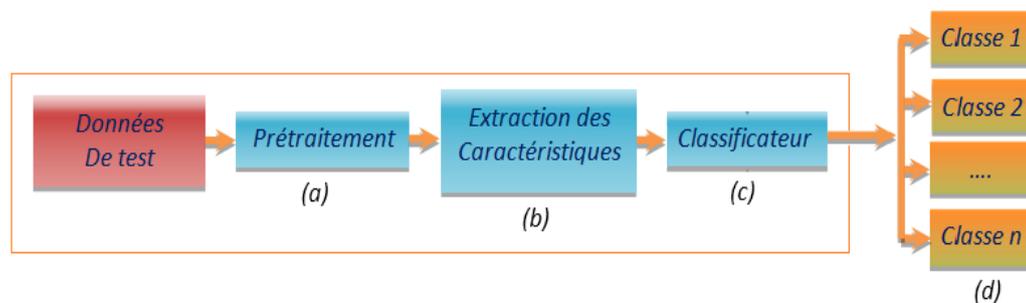


Fig. IV.5 – Synoptique du système de classification

a- Le *prétraitement* consiste à réduire l'effet du bruit et à la normalisation des données.

b- L'*extraction* d'un nombre minimal de *caractéristiques* où l'information la plus appropriée pour la classification est présentée. Ces caractéristiques sont généralement indépendantes, discriminantes, optimales et fiables. Les principales techniques d'extraction de caractéristiques sont à base du traitement du signal (ex. analyse temporelle, fréquentielle ou temps-fréquence) ou à base de modèles (ex. estimation de paramètres). Cet étage est *important* car la qualité des caractéristiques extraites détermine les performances de tout système de classification. Toutes les méthodes de classification utilisent les *caractéristiques* ou les *paramètres* qui définissent chaque objet. Plus de caractéristiques non-appropriées ou non-pertinentes rend la classification plus difficile pour toutes les méthodes. Augmenter les dimensions par le rajout de paramètres rend l'optimisation plus difficile. Alors, il est toujours mieux de donner à l'algorithme les paramètres nécessaires au lieu de s'attendre qu'il apprend à ignorer la caractéristique non-appropriée. Non seulement

les paramètres non-appropriés soient omis, mais les paramètres hautement corrélés doivent être combinés.

c- La *classification* utilise les caractéristiques déjà extraites pour identifier la classe des données.

d- La partie *décision* inclue les données de sortie classifiées qui montrent le résultat de la classification.

IV.7 - Performances

Une fois un classificateur est défini, il est nécessaire de mesurer sa précision. La précision peut être déterminée par l'application du classificateur à un ensemble d'objet d'apprentissage indépendant avec des classifications connues.

IV.8 - Conclusion

De ce qui précède, la classification consiste à réaliser attentivement l'algorithme qui traite l'ensemble des objets à trier afin de les assigner dans l'ensemble des classes. Au départ, Il faut bien choisir la méthode qui convient pour mettre en œuvre l'application.

En terme de comparaison entre les méthodes de classification, les méthodes d'intelligence artificielle, grâce à leurs hautes performances, ont pu s'imposer dans les différentes applications industrielles, surtout dans les systèmes complexes tel que la surveillance des défauts des installations nucléaires.

Chapitre V

***Application du système
neuro-flou pour la
classification***

V.1 - Introduction

Ce chapitre porte sur l'application des techniques d'intelligence artificielle pour la classification des données iris, cancer du sein et wine.

Dans ce mémoire, le modèle NEFLCLASS est utilisé pour la classification des données. Ce dernier sera traité en détail dans ce chapitre.

L'évaluation des performances de la classification est basée sur le taux d'erreur et l'interprétation des résultats des classificateurs utilisés ; réseau de neurones et NEFLCLASS.

V.2 - Classification par réseau de neurones

Dans cette partie, le réseau de neurones sera utilisé pour faire la classification des données iris, cancer du sein et wine.

Le réseau de neurones utilisé est de type PMC, constitué d'une couche cachée de dix neurones et une couche de sortie d'un seul neurone qui représente la classe.

Les données ont été divisées sur deux groupes : 66% pour l'apprentissage et 34% pour le test. Pour l'apprentissage, la fonction de rétro-propagation (*Levenberg - Marquardt*) est choisie. La fonction de performance sera l'erreur selon les moindres carrés (Mean Square Error MSE).

V.2.1 – Base de données iris

L'ensemble de données iris contient trois classes « Setosa, Versicolor, Verginica » de cinquante instances pour chacune. L'ensemble contient 150 échantillons, de quatre attributs (*Sepal Length, Sepal Width, Petal Length* et *Petal Width*).

100 échantillons seront utilisés pour l'apprentissage et 50 pour le test. Le réseau de neurones créé pour cette base de données est présenté sur *Fig. V.1*

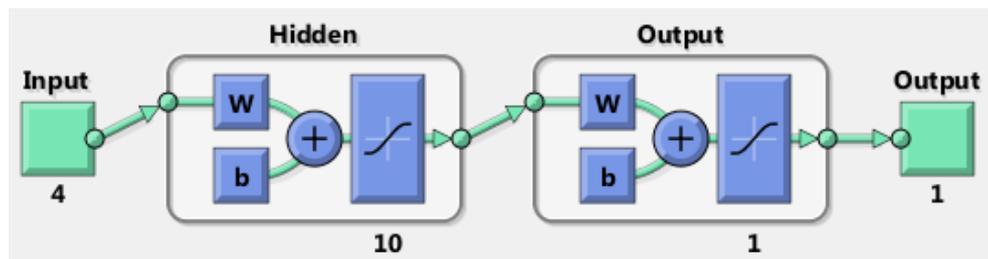


Fig. V.1 – RN créée par Matlab pour la base de données iris.

Les résultats d'apprentissage de la classification des données iris par réseau de neurones sont présentés sur *Fig. A.II.1* (Annexe II).

Après 13 itérations, l'apprentissage est terminé sous le critère de maximum d'erreur de validation avec une erreur de 8.45×10^{-4} . Les performances de la phase d'apprentissage du réseau de neurones sont présentées sur *Fig. V.2*.

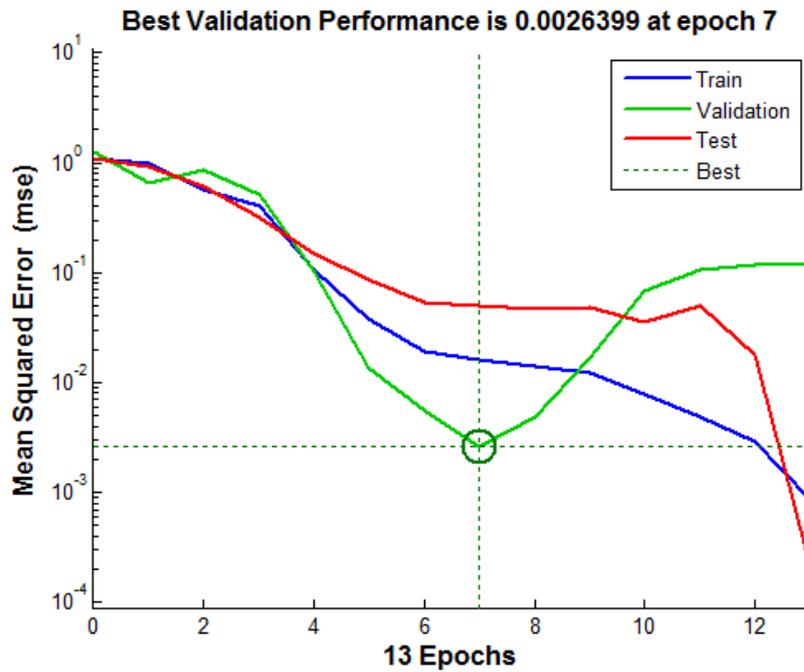


Fig. V.2 – Les performance de la phase d'apprentissage du RN pour les données iris.

La meilleure performance pour la validation a été enregistrée pour l'époque 7 avec une valeur de 2.6399×10^{-3} . Les résultats de la classification des données iris sont indiqués sur *Fig. V.3*.

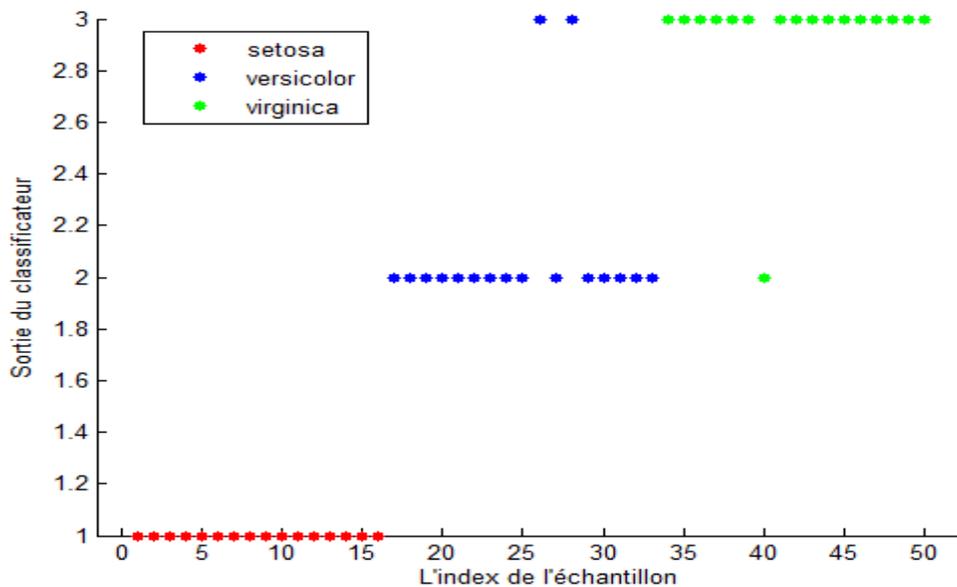


Fig. V.3 – Représentation graphique des résultats de classification pour les données iris.

Les résultats de classification montrent que le classificateur a pu classer 94% des fleurs correctement, soit trois fausses classifications.

V.2.2 – Base de données du cancer de sein (Breast Cancer)

L'ensemble de données du cancer du sein contient deux classes « Benign, Malignant ». L'ensemble contient 699 échantillons, de neuf attributs (Clump thickness, Uniformity of cell size, Uniformity of cell shape, Marginal Adhesion, Single epithelial cell size, Bare nuclei, Bland chromatin, Normal nucleoli et Mitoses).

469 échantillons seront utilisés pour l'apprentissage et 230 pour le test, le réseau de neurones créé pour cette base de données est présenté sur *Fig. V.4*

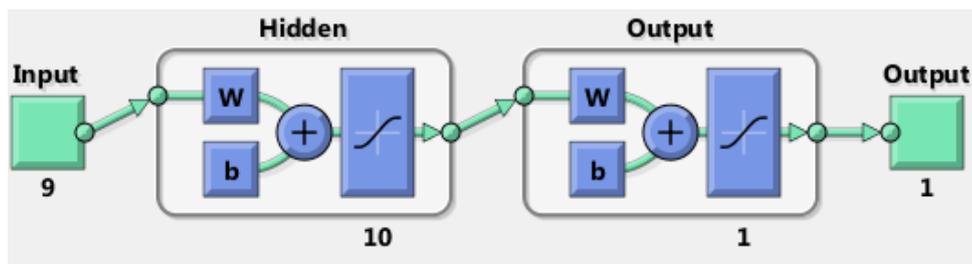


Fig. V.4 – RN créé par Matlab pour la base de données du cancer de sein.

Les résultats d'apprentissage de la classification des données iris par réseau de neurones sont présentés sur *Fig. A.II.2*.

Après 7 itérations, l'apprentissage est terminé sous le critère de maximum d'erreur de validation avec une erreur de 3.1×10^{-3} . Les performances de la phase d'apprentissage du réseau de neurones sont présentées sur *Fig. V.5*.

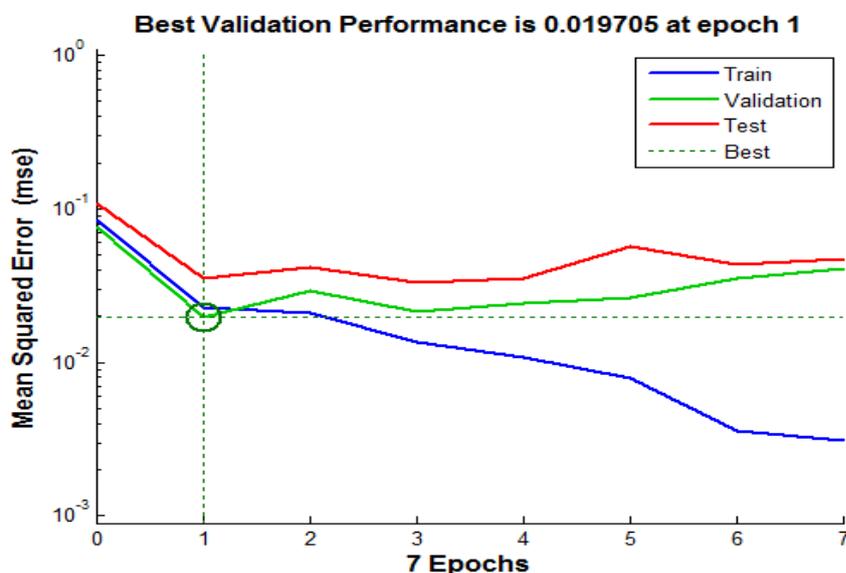


Fig. V.5 – Les performances de la phase d'apprentissage du RN pour les données du cancer du sein.

La meilleure performance pour la validation a été enregistrée pour l'époque 1 avec une valeur de 1.9705×10^{-2} .

Les résultats de la classification des données du cancer sont indiqués sur *Fig. V.6* :

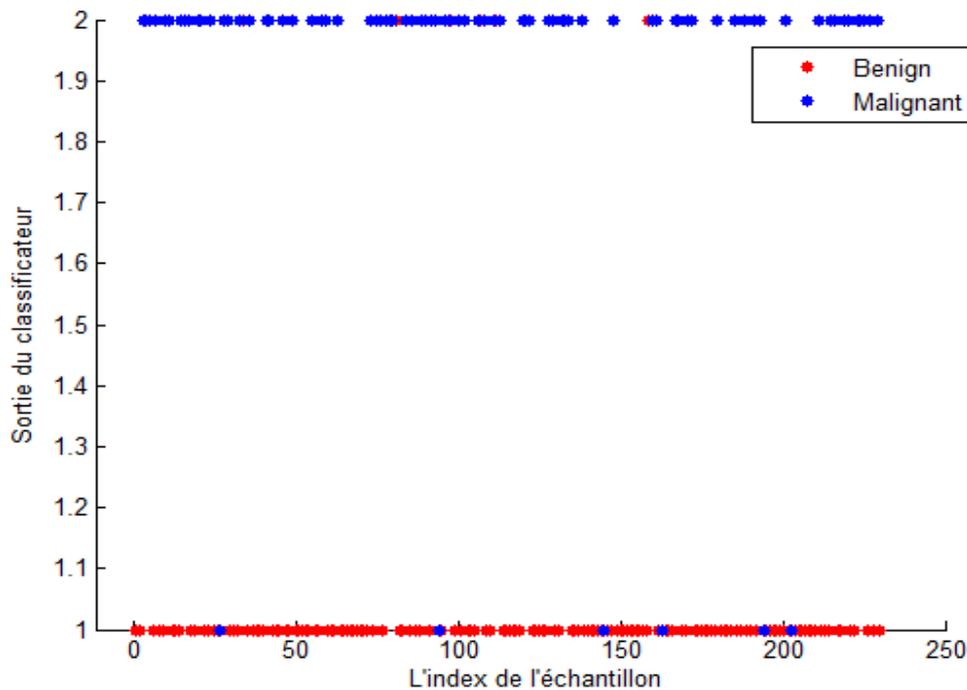


Fig. V.6 – Représentation graphique des résultats de classification pour les données du cancer du sein.

Les résultats de classification montrent que le classificateur a pu classer 94,78% des patients correctement, soit douze cas de fausse classification.

V.2.3 – Base de données wine

La base de données Wine contient trois classes « Winery 1, Winery 2 et Winery 3 ». L'ensemble contient 178 échantillons, de neuf attributs (*Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline*).

119 échantillons pour l'apprentissage et 59 pour le test, le réseau de neurone créé pour cette base de données est présenté sur *Fig. V.7*

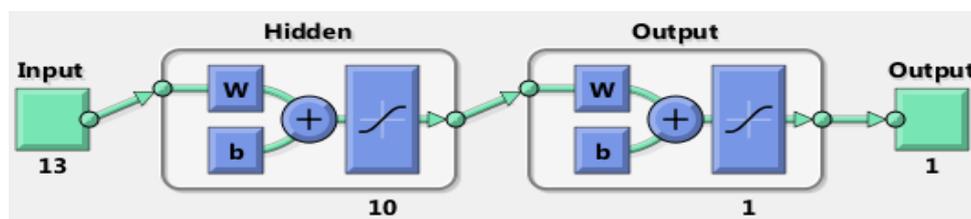


Fig. V.7 – RN créé par Matlab pour la base de données Wine.

Les résultats d'apprentissage de la classification des données wine par réseau de neurones sont présentés sur *Fig. A.II.3*.

Après 12 itérations, l'apprentissage est terminé sous le critère de maximum d'erreur de validation avec une erreur de 5.66×10^{-7} . Les performances de la phase d'apprentissage du réseau de neurones sont présentées sur *Fig. V.8*.

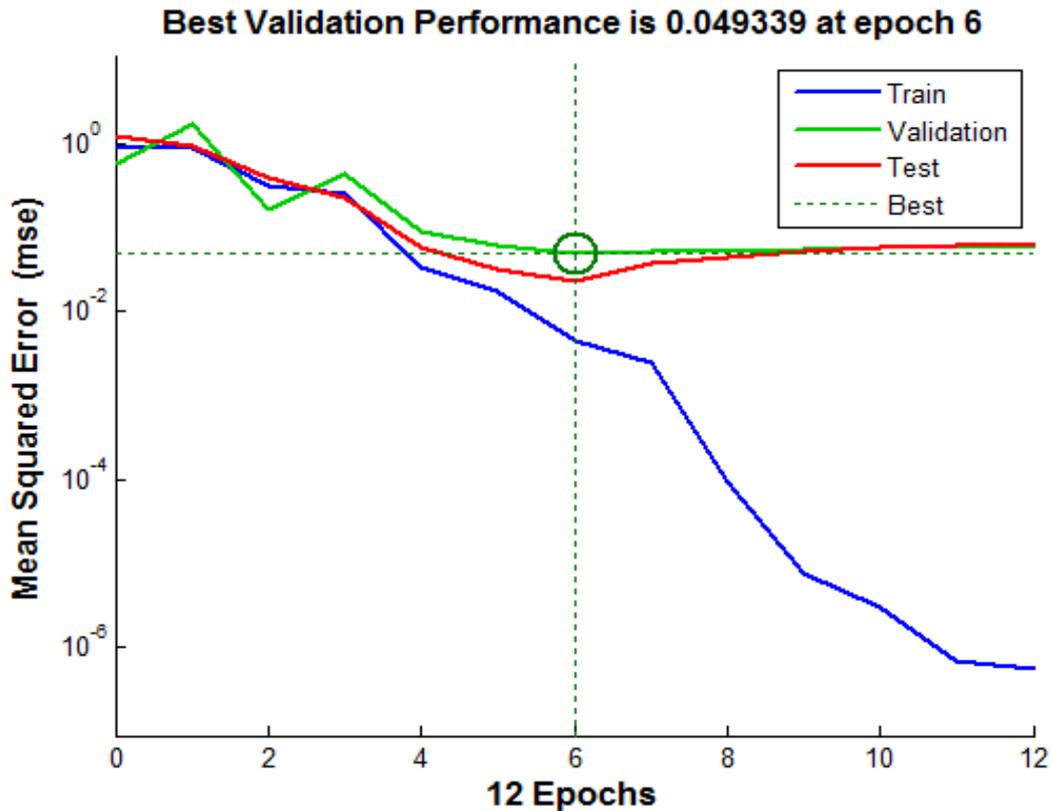


Fig. V.8 – Les performance de la phase d'apprentissage du RN pour les données du wine.

La meilleure performance pour la validation a été enregistrée pour l'époque 6 avec une valeur de 4.9339×10^{-2} .

Les résultats de la classification des données iris sont indiqués sur *Fig. V.9*.

Les résultats de classification montrent que le classificateur a pu classer 93.22% des données correctement, soit quatre cas de fausse classification.

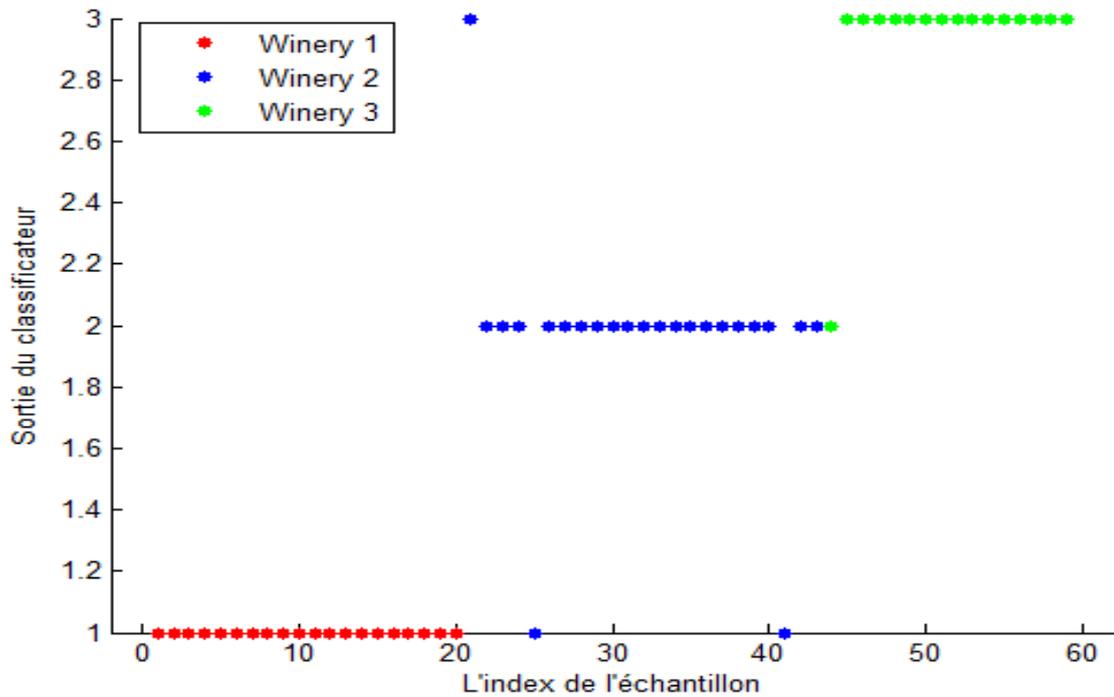


Fig. V.9 – Représentation graphique des résultats de classification pour les données du wine.

V.3 - Classification par NEFCLASS

Le modèle NEFCLASS est constitué de 3 couches (Entrée, Règle, Sortie), il peut être traité comme un réseau de neurones. Le modèle NEFCLASS est définie comme un système neuro-flou de type Mamdani.

La structure du NEFCLASS est présentée sur *Fig. III.8*. Elle illustre la structure parallèle du modèle et la circulation de l'information ; en direct (de l'entrée vers la sortie) pour la classification et par retour (de la sortie vers l'entrée) pour l'apprentissage. Le modèle NEFCLASS utilise les *poids partagés*, qui signifie que des sorties d'un neurone rassemblées par un cercle ont le même degré d'appartenance. Les neurones de ce modèle utilisent les normes T et S comme fonction d'activation.

Nous avons choisi ce modèle à cause de ses capacités qui combinent la représentation à base des règles et la généralisation par réseau de neurones. L'apprentissage du modèle NEFCLASS passe par deux phases :

- 1- *Apprentissage structurel* qui crée la base des règles (*A.III.1*) ;
- 2- *Apprentissage des paramètres* des fonctions d'appartenances (*A.III.2*).

Le but de la *première phase* est de créer une base de règles qui peut être réglée par la deuxième phase. La première phase est constituée de trois étapes :

- 1- Un ensemble de règles candidats créé à partir des données d'apprentissage ;
- 2- Vérification de la validité des conclusions de chaque règle. Sachant qu'une règle sélectionne la classe qui possède la plus grande somme des degrés d'activation ;
- 3- Calcul des degrés de performance de chaque règle.

L'apprentissage structurel se termine par la sélection des règles selon les valeurs de performances calculées. Alors, on distingue trois méthodes : sélection simple des règles, sélection des meilleures règles et sélection des meilleures règles par classe.

- *Sélection Simple des règles* - Consiste à prendre les premiers K_{\max} règles de l'ensemble des candidats, cette méthode est utilisée lorsque les échantillons sont sélectionnés de façon aléatoire et ils sont distribués de façons égales sur les différentes classes ;
- *Sélection des meilleurs règles* - Prendre les meilleures K_{\max} règles selon les performances calculées. Cette méthode est utilisée lorsque il s'agit de classes qui nécessitent d'être présentées par un nombre plus large de règles ;
- *Sélection des meilleurs règles par classe* - Prendre les meilleures règles pour chaque classe. Cette méthode est utilisée lorsque les classes ne sont pas représentées de façon égale.

La *deuxième phase* concerne l'apprentissage des paramètres des fonctions d'appartenance, qui est similaire à l'algorithme de rétro-propagation. Les algorithmes basés sur le calcul du gradient ne peuvent pas être utilisés car les fonctions d'appartenances utilisées ne sont pas dérivables. Durant l'exécution de l'algorithme, Les fonctions d'appartenance subissent des modifications telles que la translation latérale, l'augmentation ou la diminution de ses supports [16], [17], [18].

Le classificateur NEFLCLASS offre les avantages suivants:

- Une génération rapide du classificateur
- Un raffinement automatique de règles
- Traite les données numériques et symboliques
- Traitement des données incomplètes
- Possibilité d'introduire des connaissances a priori et la modification du classificateur manuellement.

Fig. V.10 représente la synoptique des phases de création d'un classificateur NEFCLASS. Dans notre application, nous avons choisi les paramètres suivants : $K_{\max} = 20$ Règles ; $\text{erreur}_{\max} = 0.1$; $\text{sigma initial} = 0.7$; nombre maximum d'itération = 100. Les fonctions d'appartenance sont de types *triangulaires*.

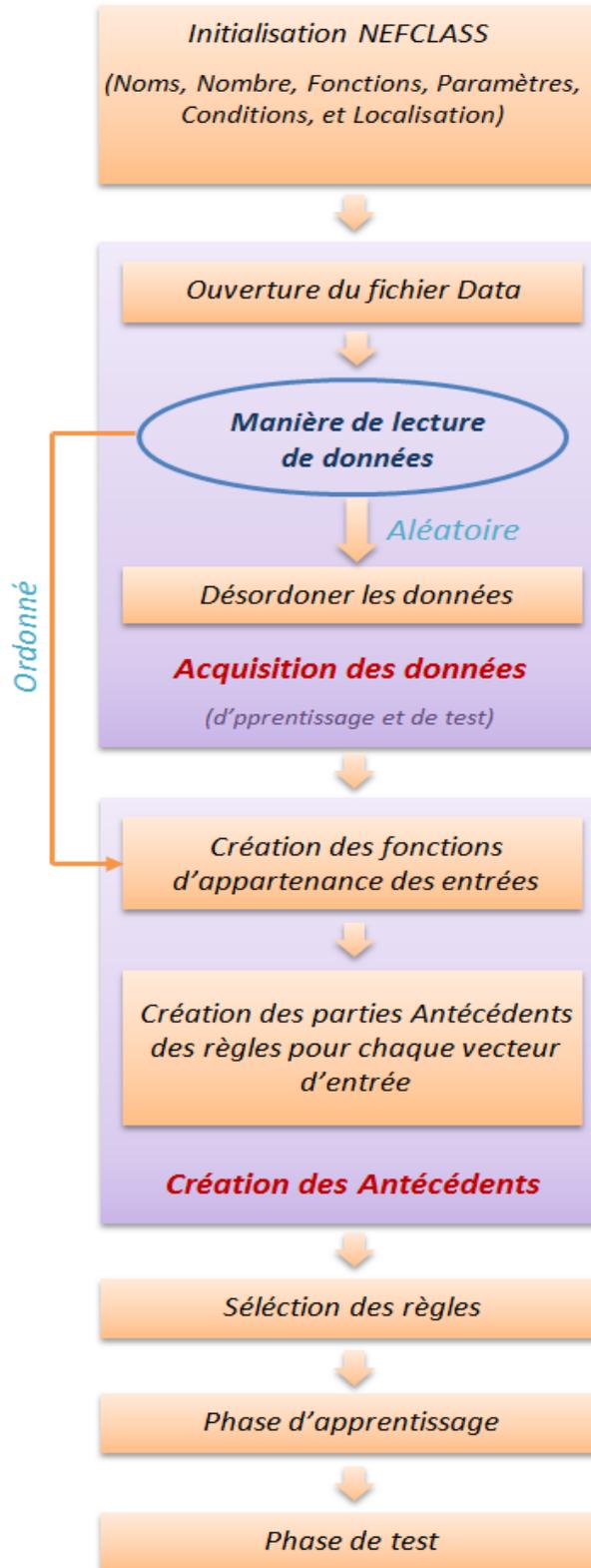


Fig. V.10 – Synoptique de création du classificateur NEFCLASS

V.3.1 – Base de données iris

Après plusieurs tests, les meilleurs résultats ont été obtenus pour le cas où nous avons considéré que la quatrième entrée (*Petal Width*).

Les résultats d'apprentissage de la classification des données iris par NEFCLASS, après 1 itération, sont présentés comme suit :

- La base de règles créées est la suivante :

Petal Width is small, then setosa

Petal Width is Medium, then versicolor

Petal Width is large, then virginica

- Les résultats du test de la classification des données iris sont représentés sur *Fig. V.11* :

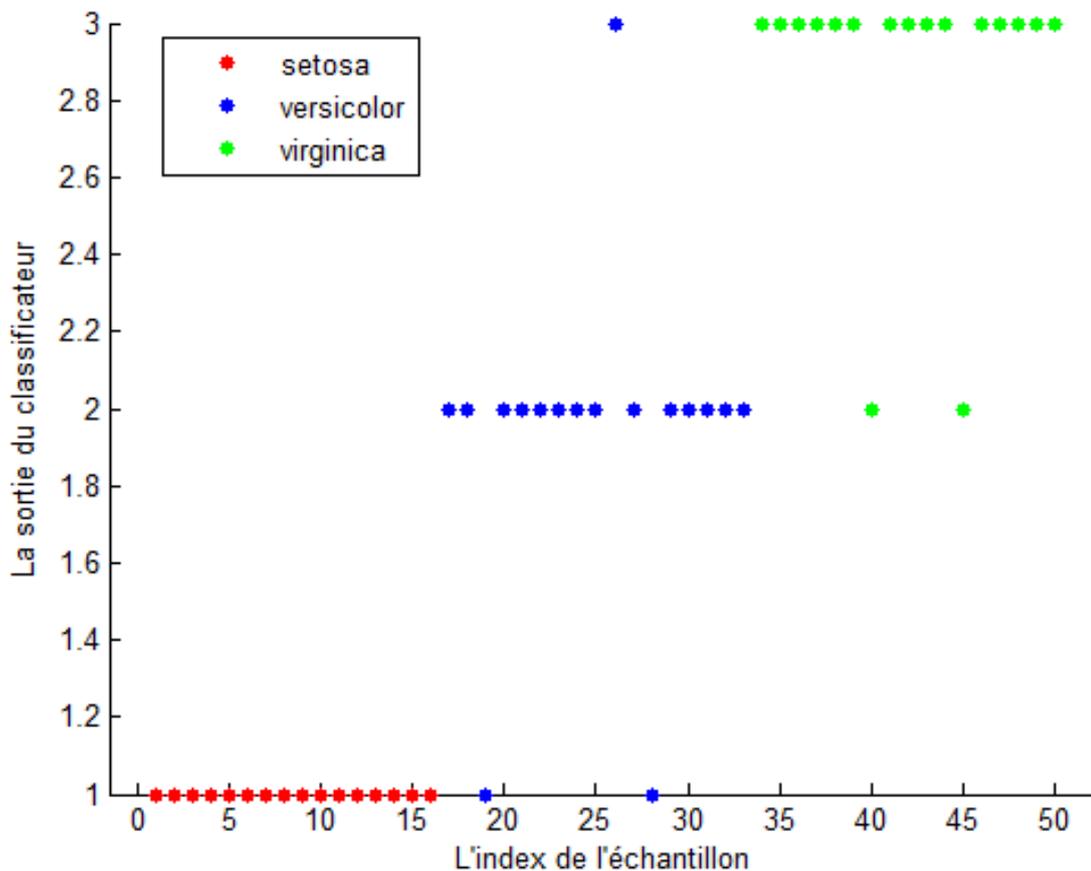


Fig. V. 11 - Résultats de classification des fleurs iris par NEFCLASS

Les résultats de classification montrent que le classificateur a pu classer 90% des fleurs, soit 5 cas de fausse classification.

V.3.2 – Base de données du cancer de sein (Breast Cancer)

Après plusieurs tests, les meilleurs résultats ont été obtenus pour le cas où nous avons considéré que la caractéristique *Clump thickness*.

Les résultats d'apprentissage de la classification des données du cancer par NEFCLASS, après 1 itération, sont présentés comme suit :

- La base de règles créées est la suivante :

Clump thickness is small, then Benign

Clump thickness is Medium, then Malignant

- Les résultats du test de la classification des données du cancer sont représentés sur *Fig. V.12* :

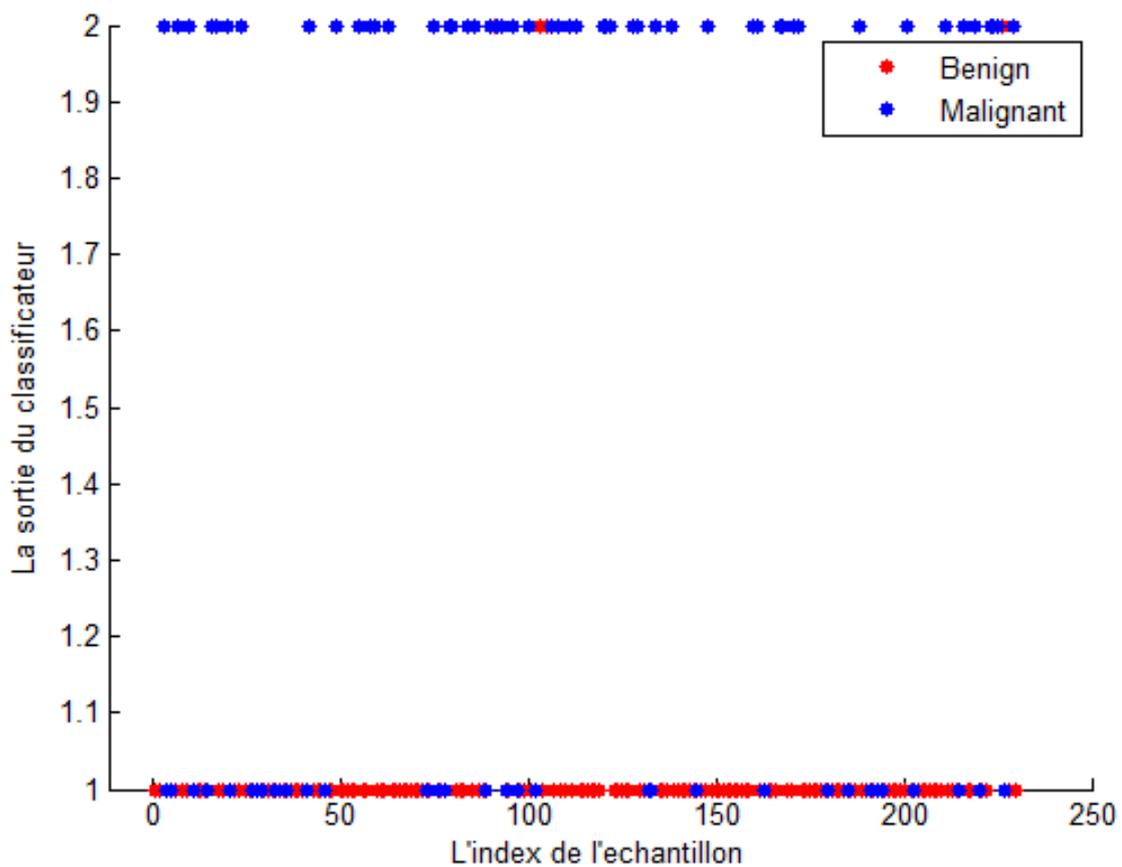


Fig. V. 12 - Résultats de classification des données du cancer par NEFCLASS

Les résultats de classification montrent que le classificateur a pu classer 84.35 % des données, soit 36 cas de fausse classification.

V.3.3 – Base de données wine

Après plusieurs test, les meilleurs résultats ont été obtenue pour le cas où nous avons considéré que les trois entrées (*Flavanoids*, *Color intensity* et *Proline*).

Les résultats d'apprentissage de la classification des données Wine par NEFCLASS, après 1 itération, sont présentés comme suit :

- La base de règles créées est la suivante :

Flavanoids is large, Color intensity is large, Proline is large, then Winery 1

Flavanoids is large, Color intensity is medium, Proline is large, then Winery 1

Flavanoids is medium, Color intensity is medium, Proline is large, then Winery 1

Flavanoids is large, Color intensity is medium, Proline is medium, then Winery 1

Flavanoids is medium, Color intensity is small, Proline is small, then Winery 2

Flavanoids is medium, Color intensity is small, Proline is medium, then Winery 2

Flavanoids is large, Color intensity is small, Proline is small, then Winery 2

Flavanoids is large, Color intensity is medium, Proline is small, then Winery 2

Flavanoids is small, Color intensity is large, Proline is small, then Winery 3

Flavanoids is small, Color intensity is large, Proline is medium, then Winery 3

Flavanoids is small, Color intensity is medium, Proline is small, then Winery 3

Flavanoids is small, Color intensity is medium, Proline is medium, then Winery 3

- Les résultats du test de la classification des données du wine sont représentés sur *Fig. V.12* ;

Les résultats de classification montrent que le classificateur a pu classer 83.05 % des données, soit 10 cas de fausse classification.

V.4 – Conclusion

D'après les résultats obtenus (*Tab. V.1*), nous pouvons faire les commentaires qui suivent :

Le classificateur NEFCLASS a obtenu une meilleure performance pour la base de données iris, mais les résultats pour les autres bases de données sont acceptables.

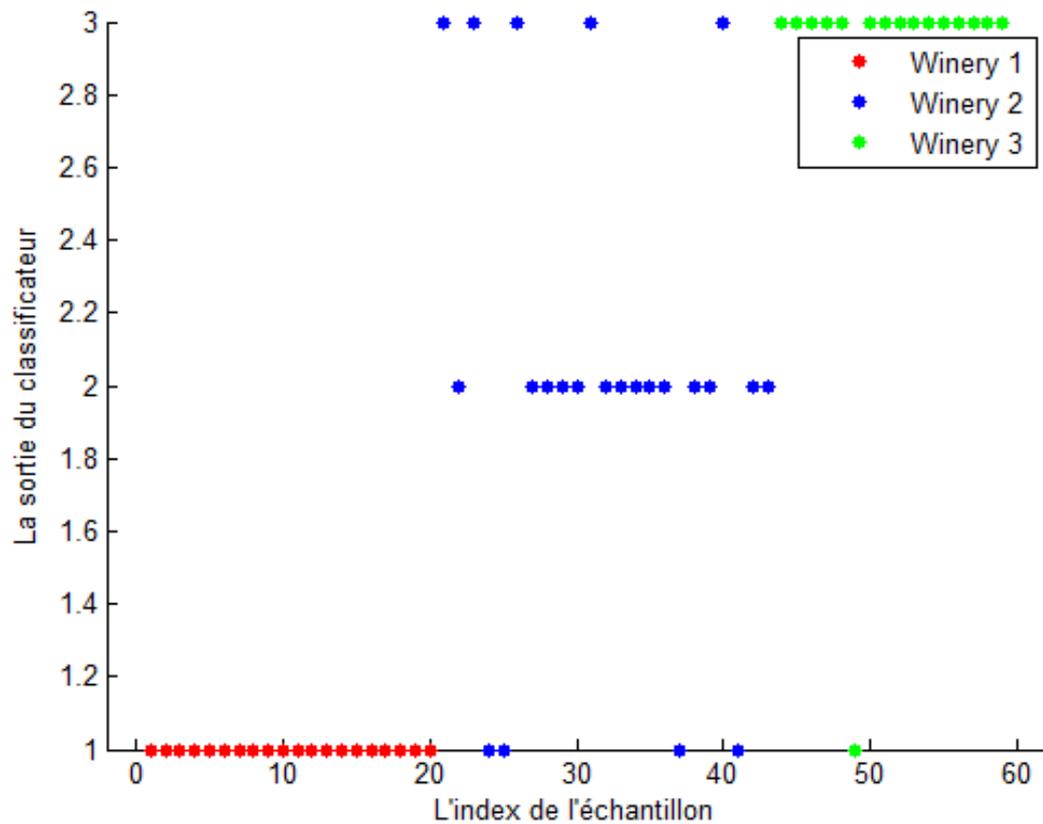


Fig. V. 13 - Résultats de classification des données du wine par NEFCLASS

La composition de la base de données influe beaucoup sur les résultats d'apprentissage et par conséquent sur les résultats de classification. Pour la base de données iris, l'utilisation de la caractéristique *Petal Width seulement* à améliorer les résultats obtenus. De même, l'utilisation des caractéristiques *Bare nuclei* parmi les 9 présente dans la base de données du cancer du sein a donné de meilleurs résultats. Pour la base de données du wine, l'utilisation des caractéristiques *Flavanoids, Color intensity et Proline* parmi les 13 caractéristiques a montré de meilleurs résultats.

Le réseau de neurones a montré une excellente performance pour la classification des trois bases de données. Le classificateur NEFCLASS de son côté n'a pas pu atteindre les performances du réseau de neurones, mais ses résultats restent dans la marge tolérée par les applications industrielles.

Malgré ses performances réduites par rapport au réseau de neurones, l'interprétation des résultats obtenus par le classificateur NEFCLASS le rend plus favorable pour les applications industrielles.

<i>Base de données</i>	<i>Classificateur</i>	<i>Nombre d'itération</i>	<i>Pourcentage de fausse classification</i>	<i>Temps d'exécution (s)</i>	<i>Nombre de règles</i>
<i>Données iris</i>	<i>Réseau de neurones</i>	<i>13</i>	<i>6 %</i>	<i>7.53</i>	<i>/</i>
	<i>NEFCLASS</i>	<i>1</i>	<i>10%</i>	<i>1.59</i>	<i>3</i>
<i>Données Cancer du sein</i>	<i>Réseau de neurones</i>	<i>7</i>	<i>4.78 %</i>	<i>5.86</i>	<i>/</i>
	<i>NEFCLASS</i>	<i>1</i>	<i>15.65%</i>	<i>1.18</i>	<i>2</i>
<i>Données Wine</i>	<i>Réseau de neurones</i>	<i>12</i>	<i>6.77 %</i>	<i>6.14</i>	<i>/</i>
	<i>NEFCLASS</i>	<i>1</i>	<i>16.95 %</i>	<i>3.09</i>	<i>12</i>

Tab. V.1 – Comparaison des résultats obtenus de classification par réseau de neurones et NEFCLASS.

Conclusion
Générale

L'objet de l'étude que nous venons de présenter concerne le domaine de la classification de données qui fait partie du diagnostic de défaillance, qu'il soit industriel ou médical. Un bon diagnostic est primordial, pour assurer la sécurité des biens et des personnes et la continuité du service. Le recours aux systèmes intelligents permet de valoriser les connaissances et l'expérience humaine. Partant du constat que l'homme est un acteur incontournable dans l'activité du diagnostic, nous nous sommes penchés sur le développement de techniques d'aide au diagnostic. L'objectif du travail présenté dans cette thèse est de proposer des éléments de réponse au problème d'intégration des connaissances d'experts et des données d'acquisition dans un processus permettant de faire la classification d'une anomalie.

Les techniques de la logique floue et des réseaux de neurones artificiels sont de bons outils d'aide à la décision (dans le domaine de la reconnaissance de forme) mais chacune présente des lacunes qui peuvent fausser les résultats. En fait, leurs fiabilités n'est pas absolue. La combinaison de ces deux approches permet de surmonter les difficultés présentées par chacune des deux et de rassembler leurs avantages. Le résultat est un système expert qui utilise le raisonnement humain tout en étant capable d'apprendre.

Les trois premiers chapitres ont été consacrés à un rappel sur les principes de base de ces trois techniques, à savoir : les réseaux de neurones, la logique floue et les systèmes neuro-flous. Dans le quatrième chapitre, nous avons détaillé le processus de classification, sa définition, ses étapes, ses méthodes et ses performances.

Le dernier chapitre comporte la mise en œuvre de la structure NEFCLASS pour la classification des données iris, cancer du sein et wine qui sont des bases de données pédagogiques disponibles dans Matlab. Les performances de ce classificateur ont été comparées à celles d'un réseau de neurones de type perceptron multicouche. Les résultats obtenus montrent que l'interprétation des résultats du classificateur NEFCLASS le rend plus favorable pour les applications industrielles, malgré ses performances réduites par rapport aux réseaux de neurones.

L'ensemble des résultats obtenus lors de notre travail durant cette période de formation de fin des études est considéré une plate forme intéressante pour la poursuite dans le futur des travaux d'amélioration, de développement et de recherche, particulièrement dans le domaine de la classification et diagnostic des défauts.

Enfin, en raison d'insuffisance de temps, nous n'avons pas pu intégrer la technique des algorithmes génétiques qui peuvent apporter des améliorations dans les performances du système de classification et participer dans l'optimisation de ces paramètres.

Bibliographie

- [1]. R. Mahadou, "*Diagnostic industriel par neuro-flou - Application à un système de production*", Thèse de magistère, Université El-hadj Lakhdar, Batna, 2007 ;
- [2]. N. Aouchar, C. Bekhaled, "*Application des systèmes hybrides neuro-flous au diagnostic des huiles de transformateurs*", Thèse de fin d'études d'ingénieur d'état, Ecole National Polytechnique, Alger, 2005 ;
- [3]. B. Gorzalczany, "*Computational Intelligence Systems and Applications*", Springer, 2002;
- [4]. L.Baghli, "*Contribution à la commande de la machine asynchrone, utilisation de la logique floue, des réseaux de neurones et des algorithmes génétiques*", Thèse de doctorat, Université Henri Poincaré, Nancy-I, France, 1999 ;
- [5]. A. Messaoud, M. Ben Messaoud, A. Kachouri, F. Sellami, "*Classification des arythmies cardiaques par logique floue à partir de signaux ECG*", LETI, Ecole Nationale d'Ingénieurs de Sfax, 2005 ;
- [6]. M.N. Cirstea, A. Dinu, J.G. Khor, M. McCormick, "*Neural and Fuzzy Logic Control of Drives and Power Systems*", Newnes, 2002;
- [7]. F. Chevrie, F. Guély, "*Cahier technique n°191 - La logique floue*", Groupe schneider, 1998 ;
- [8]. F.Lahouazi, "*Mise en œuvre d'une stratégie de commande neuro- floue : Application à un pendule inversé*", Thèse de Magistère, Université Mouloud Mammeri, Tizi-Ouzou, 2011 ;
- [9]. "*Fuzzy Logic Toolbox™ User's Guide R2014b*", Mathworks, 2014;
- [10]. N.Palluat, "*Méthodologie de surveillance dynamique à l'aide des réseaux neuro-flous temporels*", Thèse de doctorat, Université de Franche – Comté, France, 2006;
- [11]. A.Abraham, "*Neuro Fuzzy Systems: State-of- the - art Modeling Techniques*", Article, Université Monash, Australia;
- [12]. Syed Azhar Syed AB. Rahman, Fahmi Asyadi MD Yusof, Mohd Zailani Abu Bakar, "*The Method Review of Neuro-Fuzzy Applications in Fault Detection and Diagnosis System*", International Journal of Engineering & Technology IJET-IJENS Vol.10, No,03;
- [13]. S. Theodoridis, K. Koutroumbas, "*Pattern recognition*", Academic Press, 3rd edition, 2006;
- [14]. J.Vieira, F.M.Dias, A.Mota, "*Neuro-Fuzzy Systems : A Survey*", Article, Escola Superior de Tecnologia de Castelo Branco, Portugal ;
- [15]. F.Moutard, "*Brève introduction aux arbres de décision*", Mines ParisTech, 2013 ;
- [16]. U.Nauck, "*Design and Implementation of a Neuro-Fuzzy Data Analysis Tool in Java*", Université de Braunschweig, Allemagne, 1999;
- [17]. B.Gliwa, A.Byrski, "*Hybrid neuro-fuzzy classifier based on NEFCLASS model*", Computer Science, Vol. 12, 2011;

- [18]. D.D. Nauck, "*Fuzzy data analysis with NEFCLASS*", International Journal of Approximate Reasoning 32(2003) 103–130;

Annexe I

Architectures

Neuro - Flous

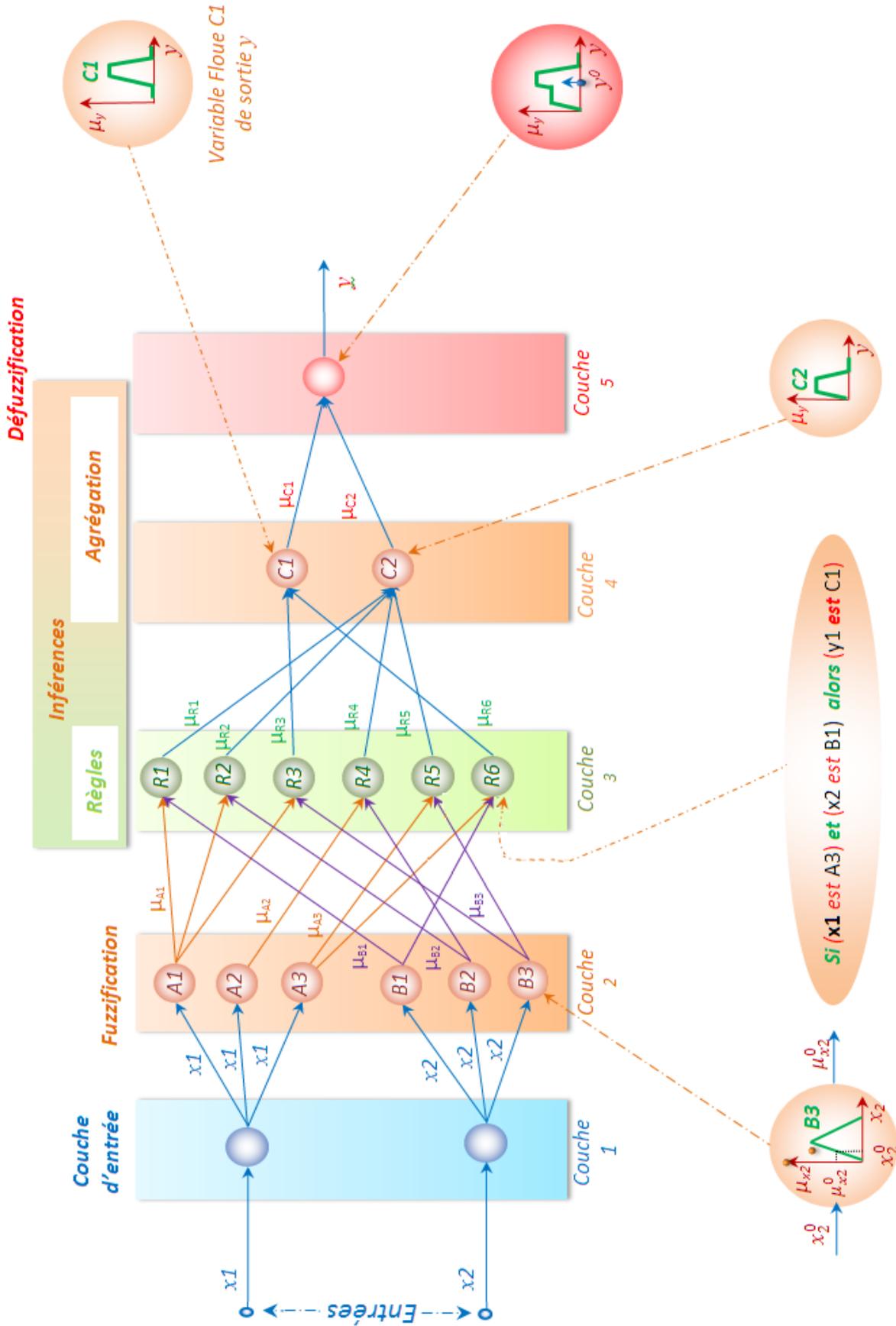


Fig. A.I.1 - Système neuro-flou de Mamdani

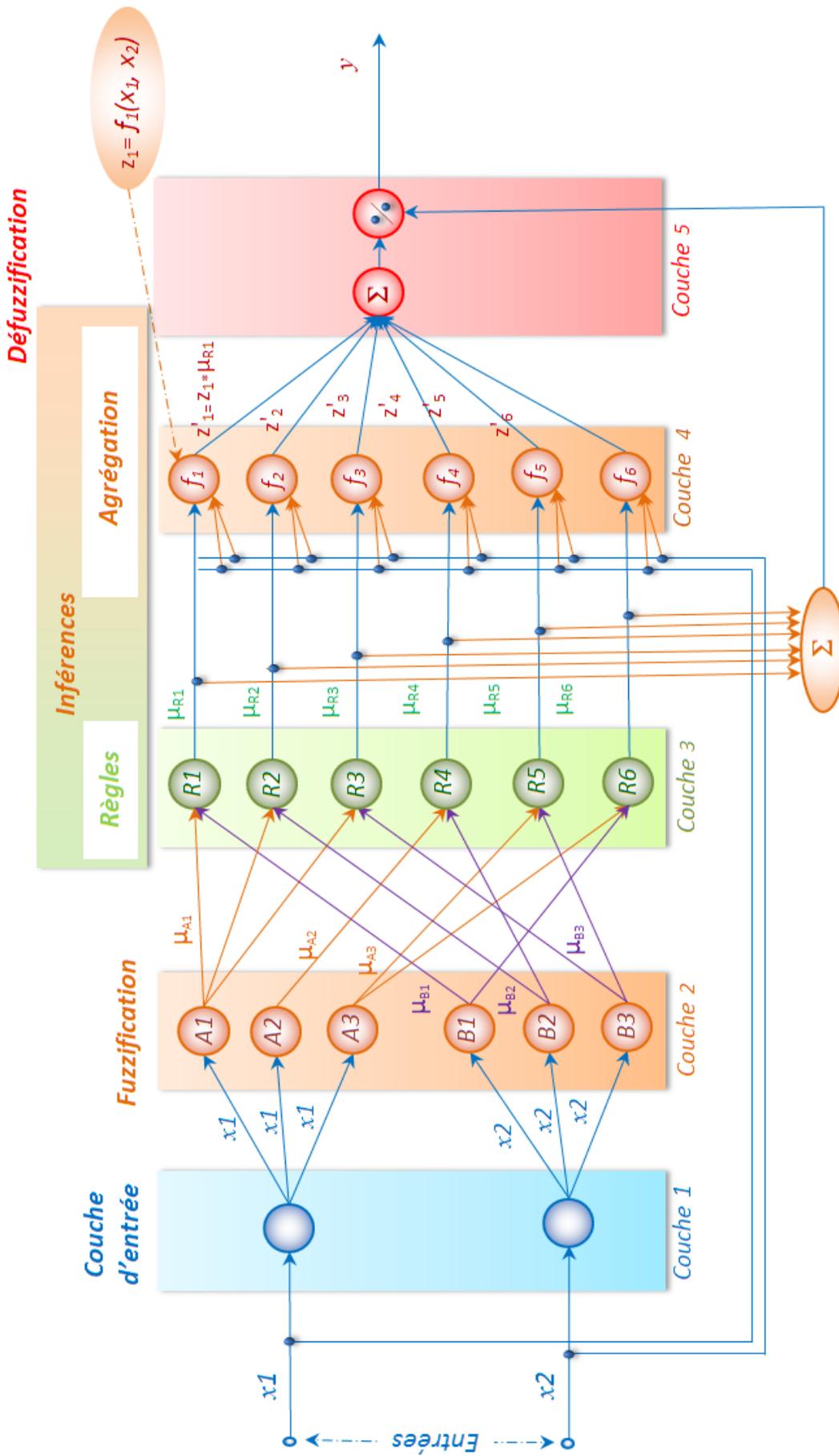


Fig. A.I.2 - Système neuro-flou de Sugeno

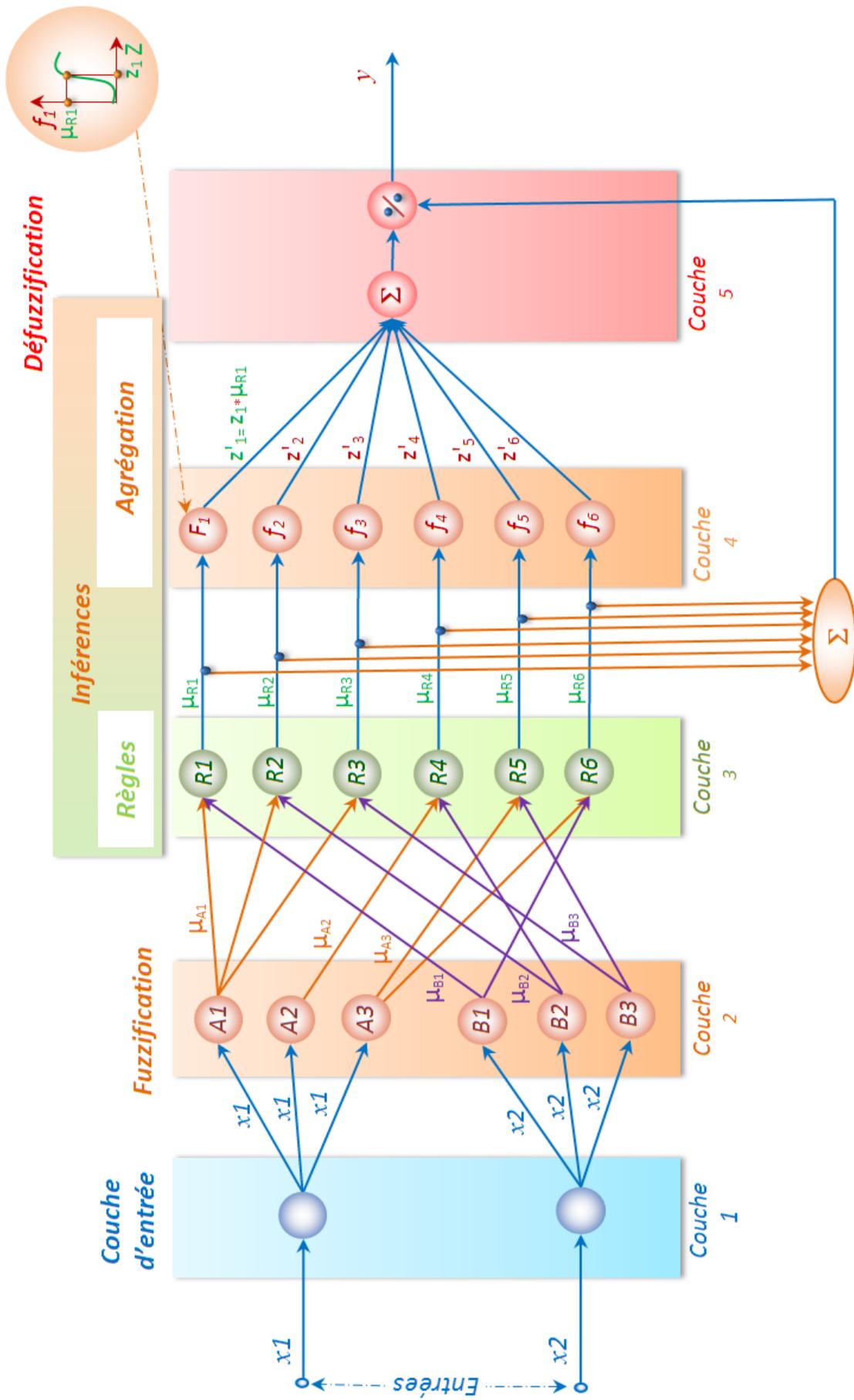


Fig. A.I.3 - Synoptique du réseau Neuro-Flou (Tsukamoto)

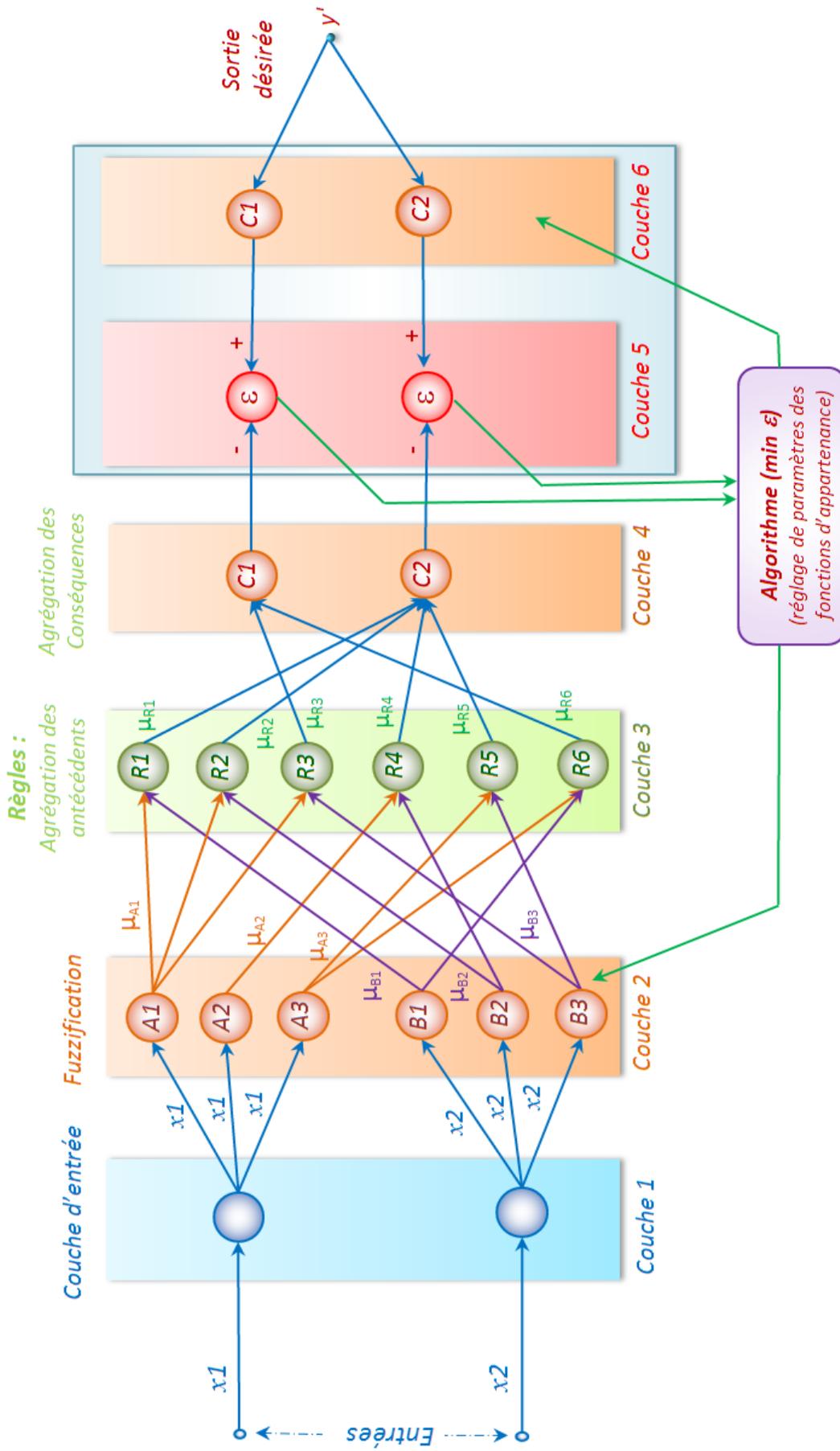


Fig. A.I.4 - Structure générale de l'apprentissage des systèmes neuro-flous

Annexe II

Apprentissages

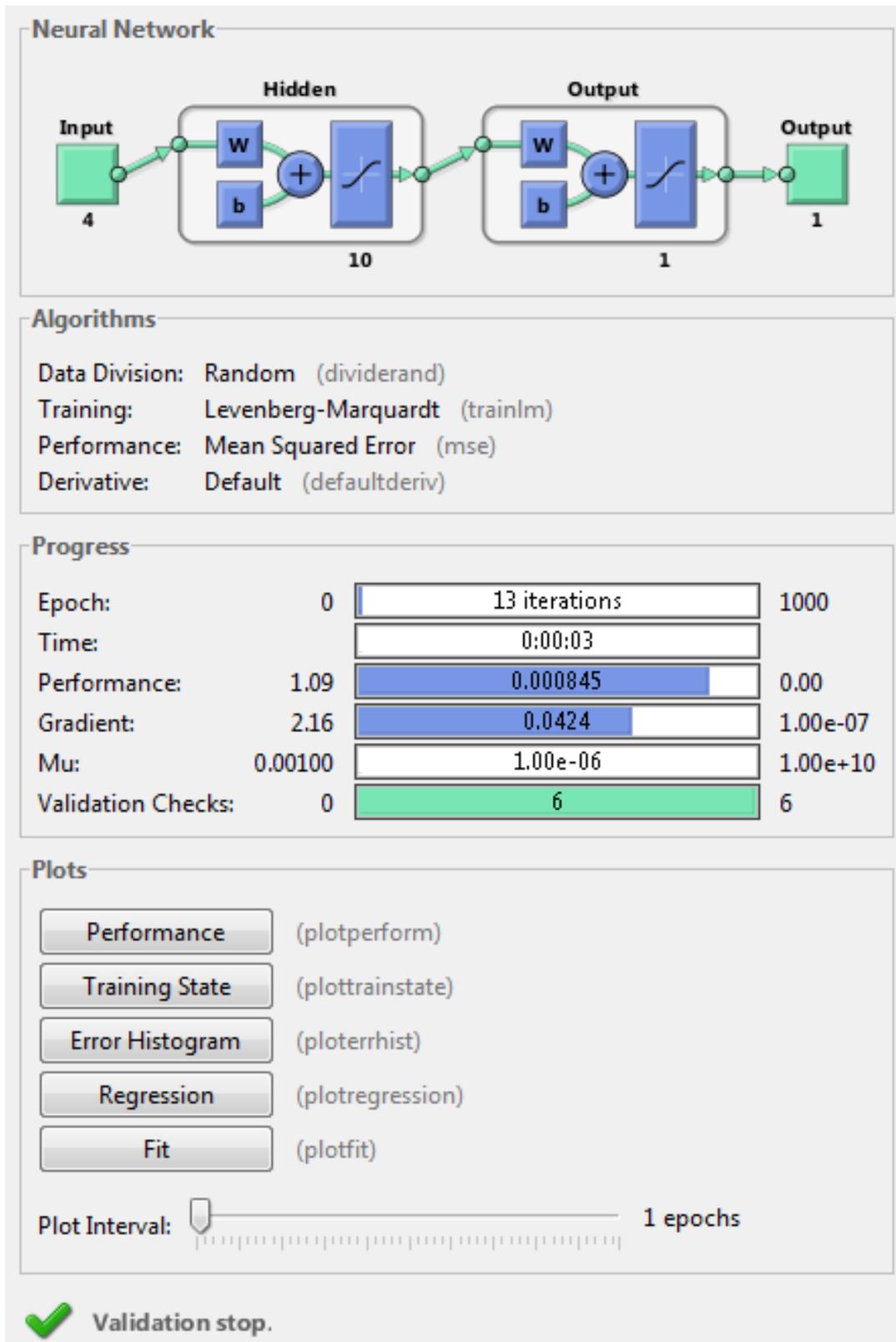


Fig. A.II.1 – Résumé de la phase d'apprentissage du RN pour la base de données iris.

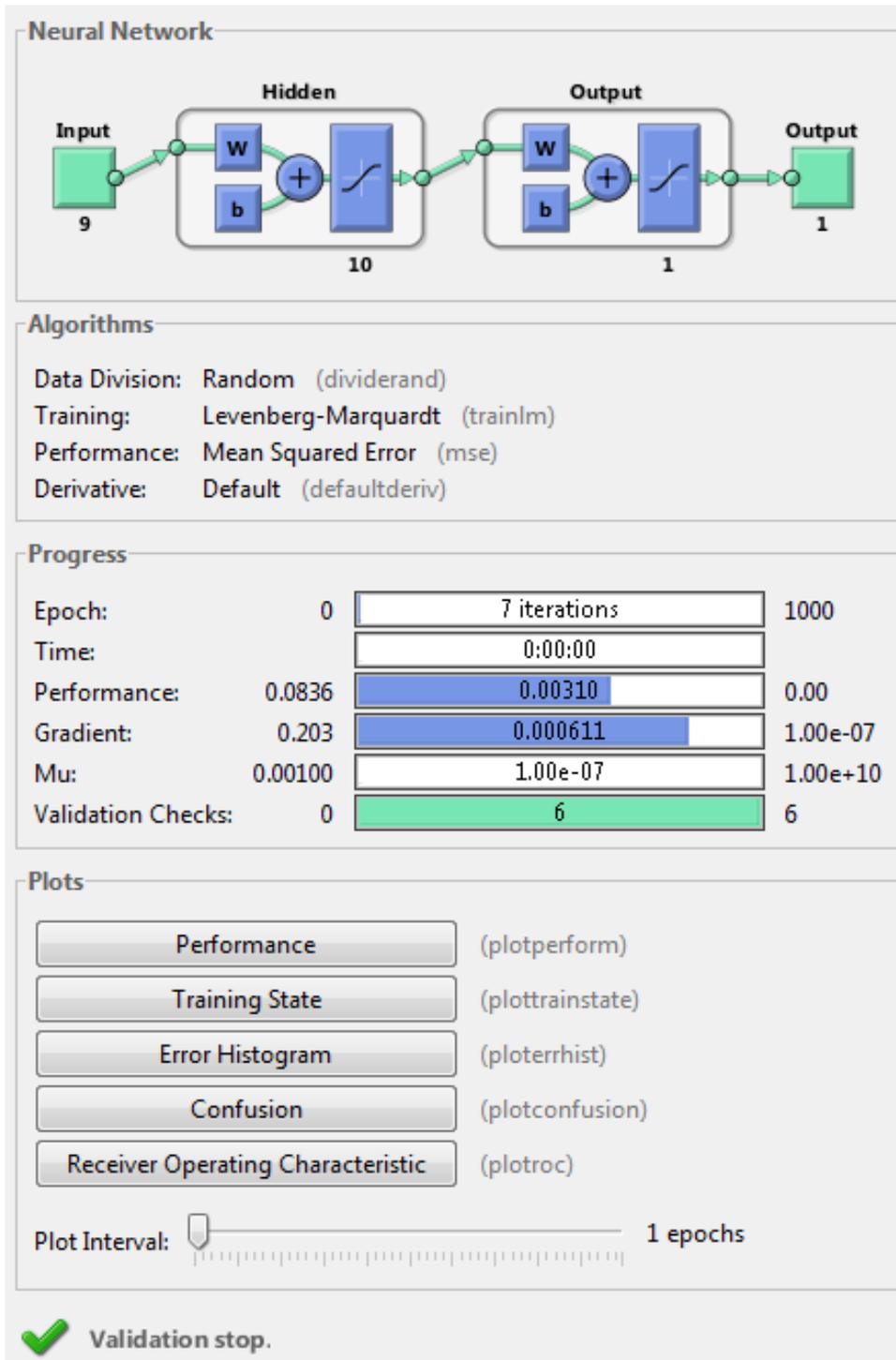


Fig. A.II.2 – Résumé de la phase d'apprentissage du RN pour la base de données du cancer du sein.

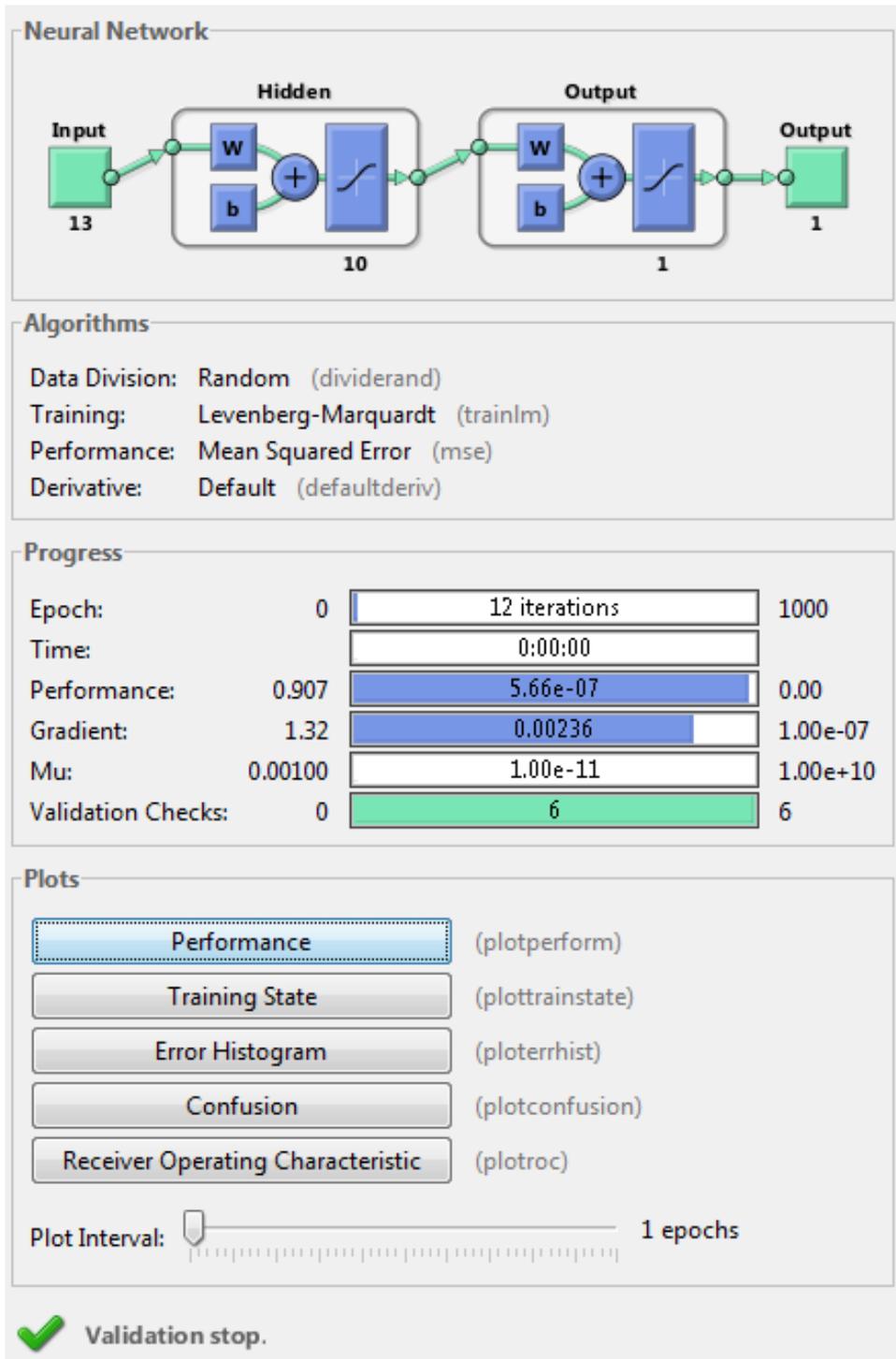


Fig. A.II.3 – Résumé de la phase d'apprentissage du RN pour la base de données wine.

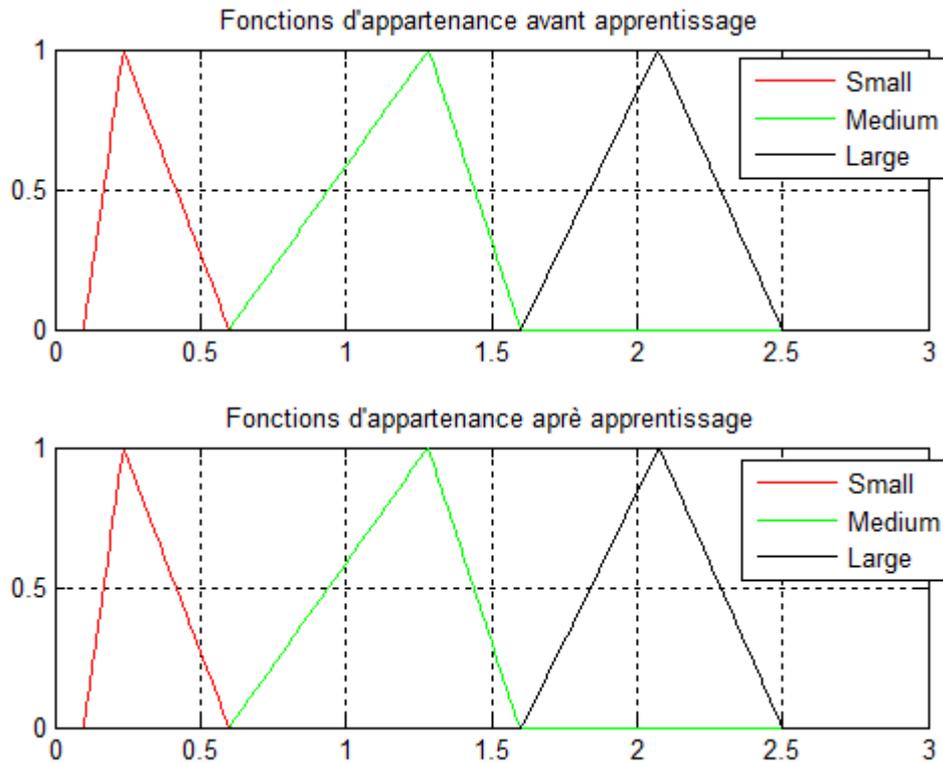


Fig. A.II.4 – Modification des ensembles flous pour l'entrée *Petal Width*
(Base de données iris)

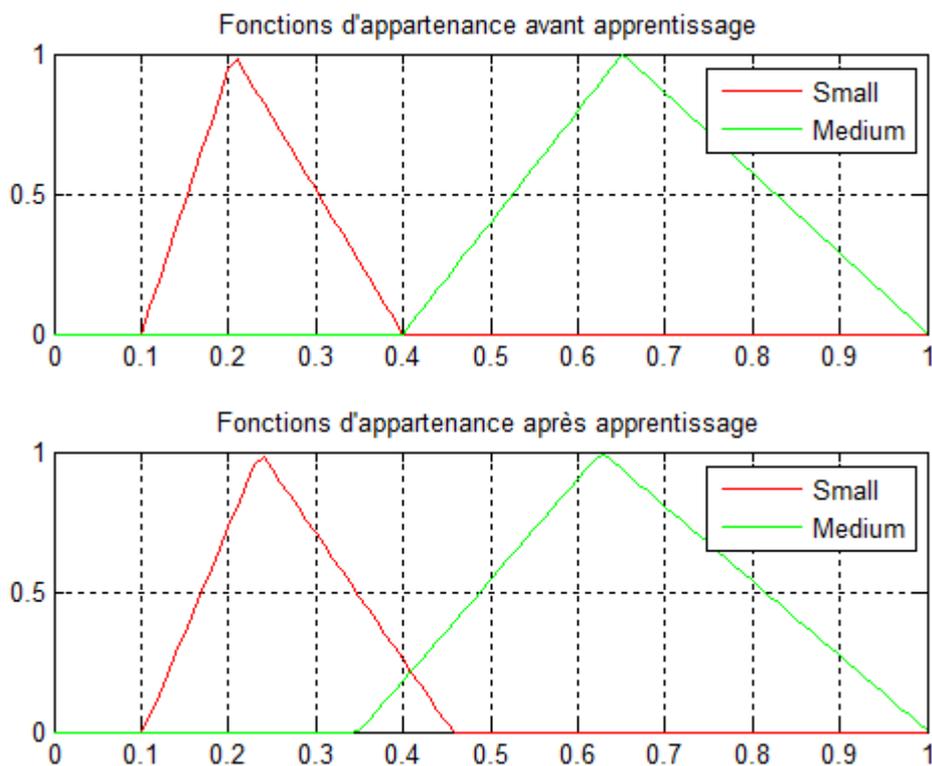


Fig. A.II.5 – Modification des ensembles flous pour l'entrée *Clump thickness*
(Base de données du cancer du sein)

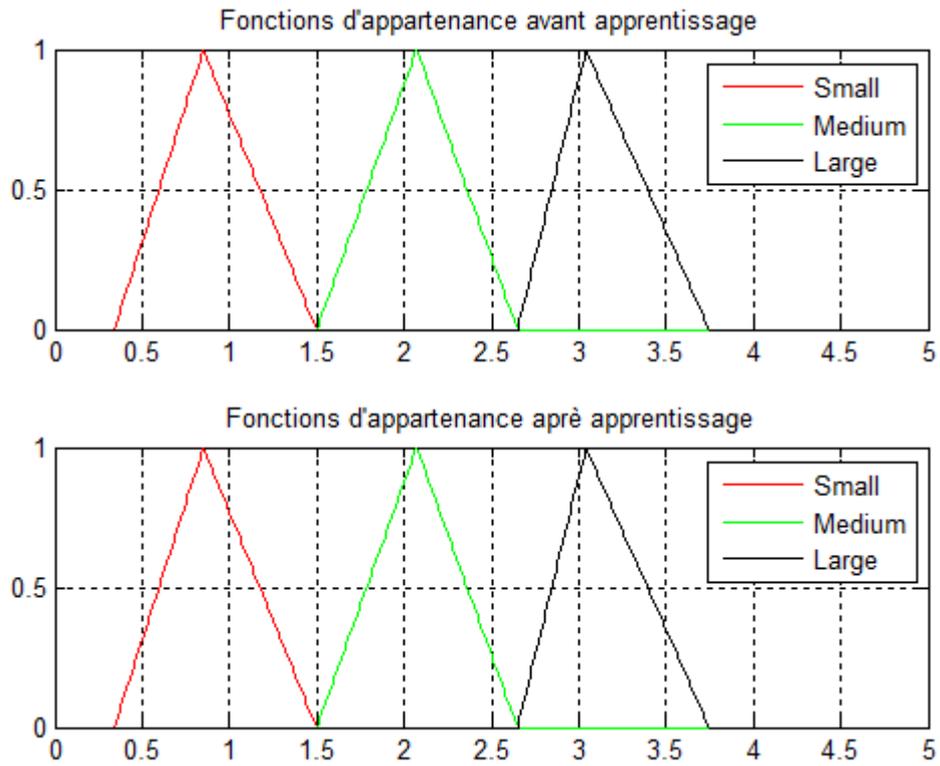


Fig. A.II.6 – Modification des ensembles flous pour l'entrée Flavanoids
(Base de données wine)

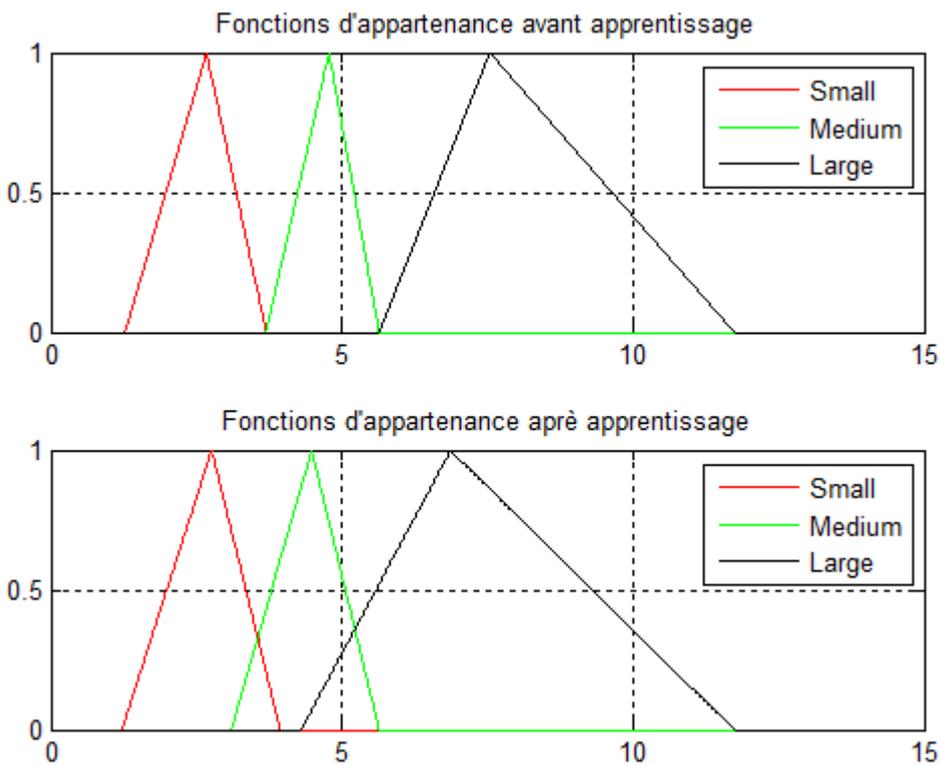


Fig. A.II.7 – Modification des ensembles flous pour l'entrée Color intensity
(Base de données wine)

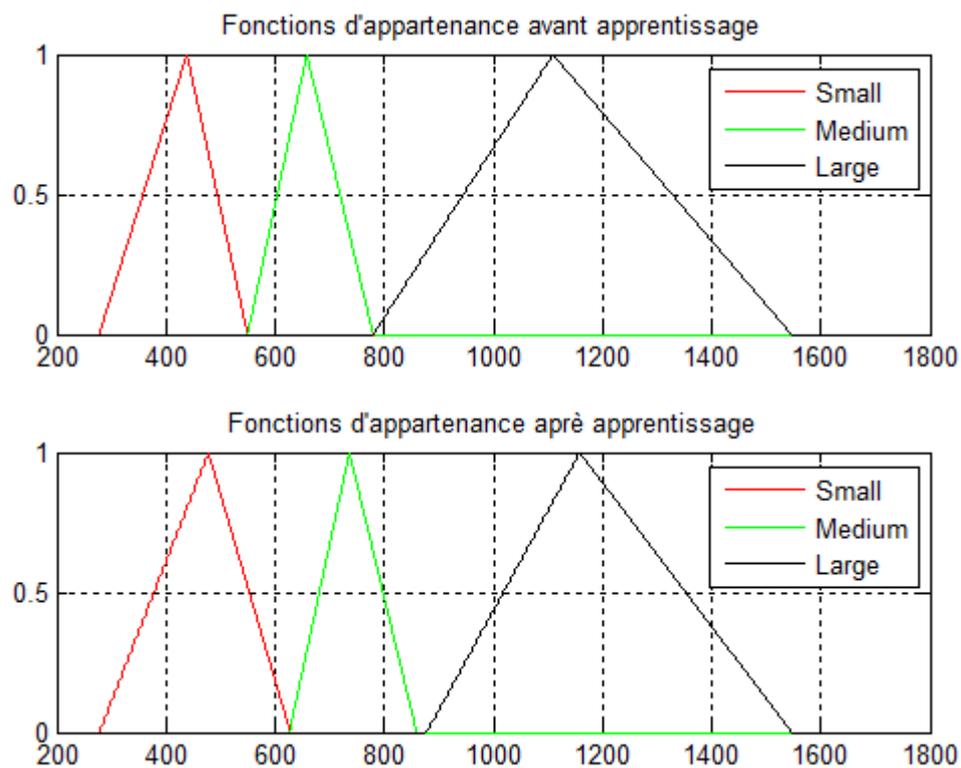


Fig. A.II.8 – Modification des ensembles flous pour l'entrée Proline
(Base de données wine)

Annexe III

Algorithme NEFCLASS

Pour chaque échantillon (p, t) de L fait

Commence

Pour chaque caractéristique d'entrée

Trouve $\mu_{ji}^{(i)}$ tel que $\mu_{ji}^{(i)} = \max \{\mu_j^{(i)}(p_i)\}$; $j \in \{1, \dots, q_i\}$

Créer l'antécédent $A = (\mu_{ji}^{(1)}, \dots, \mu_{ji}^{(n)})$;

Si A n'est pas dans la list des antécédents

Alors ajouter A à la liste des antécédents

Fin

Pour chaque échantillon (p, t) de L fait

Pour chaque antécédent A_j fait

Commence

$c = \text{index de la classe de } p \text{ fournit par } t$;

$C_j(c) = C_j(c) + A_j(p)$;

Fin

Pour chaque antécédent A_j fait

Commence

$c = \text{argmax} \{C_j(i)\}; i \in \{1, \dots, m\}$

Créer la règle R_i avec l'antécédent A et la consequence c ;

Ajouter R_i à la liste des règles candidates ;

$\text{performance}_j = C_j(c) - \sum_{i \in \{1, \dots, m\}} C_j(i)$;

Fin

Si "best"

Alors pour $i = 1$ à k_{\max} fait

$R = \text{argmax}_{R_j} \{\text{performance}_j\}$;

Ajouter R à la base des règles ;

Supprimer R de la liste des règles candidates

Si "best per class"

Alors spour chaque classe c fait

Pour $i = 1$ à $\frac{k_{\max}}{m}$ fait

$R = \text{argmax} \{\text{performance}_j\}$;

Ajouter R à la base des règles ;

Supprimer R de la liste des règles candidates ;

Fig. A.III.1 – Algorithme d'apprentissage des règles du modèle NEFCLASS [16].

Répéter

Propager l'échantillon suivant (p,t)

Pour chaque unité de sortie c_i fait

$$ec_i = t_i - \text{activation}(c_i);$$

Pour chaque règle R avec activation (R) > 0 fait

Commence

$$e_R = \text{activation}(R) \cdot (1 - \text{activation}(R)) \cdot \sum_{c_i} (W(R, c) \cdot ec_i);$$

$$j = \operatorname{argmin} \{W(x_i, R)(p_i); \quad i \in \{1, \dots, n\}\}$$

$$\mu = W(x_j, R);$$

(a_μ , b_μ , et c_μ sont les paramètres de l'ensemble flou μ *)*

$$\delta_b = \sigma \cdot e_R \cdot (c_\mu - a_\mu) \cdot \operatorname{sgn}(p_i - b_\mu);$$

$$\delta_a = -\sigma \cdot e_R \cdot (c_\mu - a_\mu) \cdot \delta_b;$$

$$\delta_c = \sigma \cdot e_R \cdot (c_\mu - a_\mu) \cdot \delta_b;$$

Fin

Jusqu'à un critère d'arrêt.

Fig. A.III.2 – Algorithme d'apprentissage des fonctions d'appartenance du modèle NEFCLASS [16].