

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

Université Akli Mohand Oulhadj - Bouira -

ⵍⵓⵎⵓⵔ ⵏ ⵉⵏⵙⵉⵎⵏⵏⵉⵎ ⵙⵓ⵭ⵉⵔ
ⵏ ⵉⵔⵎⵉⵏⵏⵉⵔ ⵙⵉⵏⵜⵉⵎⵉⵏⵜ



وزارة التعليم العالي والبحث العلمي
جامعة أكلي محمد أولحاج
- البويرة -

Faculté des Sciences et des Sciences Appliquées

كلية العلوم والعلوم التطبيقية

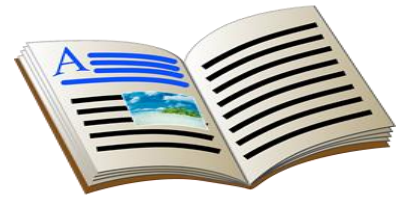
Département de génie électrique

Polycopié de cours

En : Electronique

Spécialité : Electronique des systèmes embarqués

Niveau : Master



Techniques de l'intelligence artificielle

Par : Dr. LADJOUZI Samir

Expertisé par:

Pr. Saoudi Kamel, Professeur, Université de Bouira

Pr. Khettab Khatir, Professeur, Université de M'sila

Année : 2022/2023

Table des matières

Introduction générale	1
Chapitre 1 : Les réseaux de neurones artificiels	
1- Introduction	2
2- Neurone biologique	2
3- Neurone artificiel (formel).....	3
4- Réseaux de neurones artificiels	5
4-1- Réseaux statiques.	5
4-2- Réseaux récurrents.	7
5- Apprentissage des RNA	8
5-1- Apprentissage supervisé.	8
5-2- Apprentissage non supervisé.	9
5-3- Apprentissage semi-supervisé.	9
5-4- Apprentissage profond.	9
6- Exemples de règles d'apprentissage des RNA	9
6-1- Règle de Hebb.	10
6-2- Règle de Widrow-Hoff(règle du Delta).	10
6-3- Algorithme de rétropropagation.....	11
6-3-1- Historique	11
6-3-2- Principe et procédure.	11
6-3-3- Exemple d'application sur le RMC	12
7- Domaines d'application des RNA	13
8- Avantages et inconvénients des RNA	13
8-1- Avantages.	14
8-2- Inconvénients.	14
Chapitre 2 : La logique floue	
1- Introduction	15
2- Historique	15
3- Notions d'ensemble flou et fonction d'appartenance	16
3-1-Définition.	16
3-2-Propriétés des ensembles flous.	19
a)- Le support.	19
b)- α -coupure	19
3-3-Opérations sur les ensembles flous.	19
4- Règles d'inférence	20
5- Systèmes flous.	21

5-1-Fuzzification.	21
5-2-Base de connaissances.	22
5-3-Engin d'inférence.	22
5-4-Déffuzzification.	22
6- Exemple d'application	23

Chapitre 3 : Les algorithmes génétiques

1- Introduction	27
2- Historique.	27
3- Présentation des algorithmes génétiques.	27
4- Structure d'un algorithme génétique	28
5- Les éléments d'un algorithme génétique.	30
5-1-Individu.	30
5-2-Population.	30
5-3-Fonction « Fitness ».	31
5-4-Sélection.	31
5-5-Reproduction.	31
6- Le codage dans un algorithme génétique.	32
6-1-Le codage binaire.	32
6-2-Le codage réel	33
7- Génération de la population initiale.	33
8- Fonction d'adaptation « Fitness »	33
9- Reproduction (sélection des parents)	34
9-1-Sélection par la roue de loterie biaisée.	34
9-2-Sélection élitiste.	35
9-3-Sélection par tournoi.	35
9-4-Sélection par rang.	35
10- Opérateurs de recombinaison génétique (croisement et mutation)	36
10-1-Croisement.	36
a- Croisement en un point.	36
b- Croisement en deux points.	37
c- Croisement uniforme.	37
d- Croisement barycentre.	37
10-2-Mutation	38
10-2-1-Mutation binaire.	38
10-2-2-Mutation réelle.	38
a- Mutation uniforme.	38
b- Mutation non uniforme.	39
11- Sélection des individus d'une nouvelle génération	39
11-1-Sélection par compétition.	39
11-2-Sélection élitiste.	39
11-3-Sélection par descendance.	39
12- Domaines d'application des AG.	40
Bibliographie.	41
Examen et corrigé 2021/2022	42
Rattrapage et corrigé 2021/2022	48

Liste des figures

Chapitre 1 : Les réseaux de neurones artificiels

Figure I-1 Le neurone biologique	2
Figure I-2 Le neurone artificiel	3
Figure I-3 Exemple de réseau RMC à une seule couche cachée	6
Figure I-4 Architecture du réseau d'Elman	7
Figure I-5 Représentation schématique de la règle de Hebb	10

Chapitre 2 : La logique floue

Figure II-1 Exemple de fonction d'appartenance d'ensemble flou qui représente les prix des PCs	17
Figure II-2 Représentation de la température en logique floue	18
Figure II-3 Les ensembles flous A et B	20
Figure II-4 L'ensemble flou $A \cup B$	20
Figure II-5 L'ensemble flou $A \cap B$	20
Figure II-6 L'ensemble flou \bar{A}	20
Figure II-7 Structure général d'un système flou	21
Figure II-8 Ensembles flous de la variable d'entrée : température (°) « x »	23
Figure II-9 Ensembles flous de la variable d'entrée : humidité (%) « y »	24
Figure II-10 Ensembles flous de la variable de sortie : vitesse du ventilateur (%) « z »	24
Figure II-11 Ensemble flou de sortie par la méthode de coupure	25

Chapitre 3 : Les algorithmes génétiques

Figure III-1 Différents niveaux dans un algorithme génétique	28
Figure III-2 Organigramme d'un AG standard.	29
Figure III-3 Chromosome avec codage binaire.	32
Figure III-4 Chromosome avec codage réel.	33
Figure III-5 Sélection par roulette.	35
Figure III-6 Croisement en un point.	36
Figure III-7 Croisement en deux points.	37
Figure III-8 Croisement uniforme.	37
Figure III-9 Mutation binaire.	38

Introduction générale

L'Intelligence Artificielle (IA) est une branche de l'informatique qui se consacre à la création de machines capables d'égaliser et même de dépasser les capacités cognitives des êtres vivants, en contraste avec l'intelligence naturelle humaine. Au fil du temps, le terme IA a connu une évolution majeure et englobe désormais toutes les idées visant à doter les machines de capacités cognitives similaires à celles de l'homme.

La mise en œuvre de l'intelligence artificielle nécessite une compréhension de la perception, du raisonnement, du traitement de l'information et de l'apprentissage. Elle conduit à des découvertes quotidiennes dans divers domaines, tant techniques que non techniques, civils et militaires, tels que l'économie, la sociologie, la médecine, l'agriculture, l'industrie et toutes les infrastructures militaires.

Le domaine de l'IA est extrêmement étendu et peut englober de nombreuses techniques différentes. Les capacités de calcul toujours croissantes des ordinateurs, une meilleure compréhension de certains processus naturels liés à l'intelligence, ainsi que les progrès des chercheurs dans les sciences fondamentales ont permis de réaliser d'importantes avancées dans ce domaine.

Ce cours constitue un aperçu et une initiation sur les techniques de l'Intelligence Artificielle (IA) les plus récentes. L'objectif est de faire découvrir aux étudiants au moins trois techniques d'IA, à savoir : les réseaux de neurones artificiels, la logique floue ainsi que les algorithmes génétiques.

Ce polycopié s'adresse spécialement aux étudiants de première année Master, option : Electronique des Systèmes Embarqués mais il peut être suivi par tous ceux qui s'intéressent aux techniques de l'IA.

L'objectif de ce cours est de permettre la compréhension et l'utilisation des outils fondamentaux à l'IA afin de les exploiter pour diverses applications aussi bien industrielles que pédagogiques tels que : la reconnaissance de formes, l'identification et le contrôle des systèmes dynamiques, détection de maladies,...etc.

Chapitre 1

Les réseaux de neurones artificiels

1- Introduction

Ces dernières décennies, l'Intelligence Artificielle (IA) a été présente dans beaucoup de domaines tel que : les sciences sociales, les sciences physiques, les probabilités et statistiques... etc, et ce dans le but de modéliser des problèmes complexes. Le principe de fonctionnement de l'IA repose sur des mécanismes intelligents inspirés de certaines capacités qu'un humain utilise pour résoudre des tâches compliqués.

Parmi ces mécanismes intelligents on peut citer les Réseaux de Neurones Artificiels (RNA) qui offrent de grands avantages par rapport à la modélisation conventionnelle. La théorie et la modélisation des RNA s'inspirent de la structure et du fonctionnement des systèmes nerveux biologiques où le neurone représente l'élément fondamental.

Nous allons aborder dans ce chapitre les notions et les fondements généraux de la théorie des RNA. Nous commencerons par la description d'un neurone biologique puis de son homologue le neurone formel et nous citerons ensuite quelques architectures les plus utilisées.

Nous parlerons de l'apprentissage des RNA, de ses types ainsi que quelques exemples des algorithmes utilisés pour l'apprentissage. Nous terminerons par quelques applications impliquant les RNA et enfin nous citerons quelques avantages et inconvénients des RNA.

2- Neurone biologique

Le cerveau humain est composé d'un grand nombre de cellules neuronales, estimées à plus d'un milliard qui traitent les informations. Chaque cellule fonctionne comme un simple processeur ou bien une unité de calcul. Un neurone biologique est composé de trois éléments principaux comme le montre la figure I-1 : les dendrites, le corps cellulaire et l'axone.

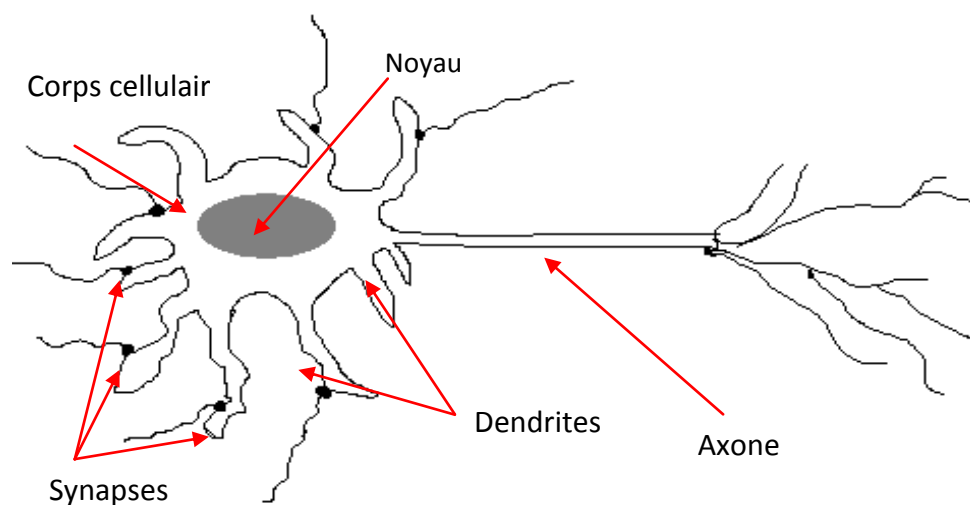


Figure I-1 Le neurone biologique

Le système nerveux envoie des informations électriques lesquelles transitent par les dendrites, qui représentent les signaux d'entrée du neurone, jusqu'au soma (corps cellulaire). Ce dernier accumule ces signaux et s'ils excitent suffisamment le neurone (en passant par un seuil d'excitation) alors ils sont transmis vers l'axone.

Les axones des neurones (sorties) sont connectés aux dendrites d'autres neurones par les synapses, ce qui fait de notre cerveau un réseau extrêmement complexe de neurones, compte tenu que nous possédons plusieurs milliards de neurones et que chaque neurone peut avoir plusieurs milliers de dendrites. Les synapses jouent un rôle important car elles sont le lien entre la sortie d'un neurone et l'entrée d'un autre neurone.

L'influx nerveux passe par ces synapses qui peuvent inhiber ou laisser passer le signal nerveux. Si l'amplitude du signal arrivant à la synapse est assez élevée, alors elle laissera passer le signal, sinon il sera inhibé. La synapse est capable de s'adapter pour laisser passer le signal ou empêcher son passage : ce mécanisme d'adaptation est à la base de l'apprentissage de notre cerveau.

3- Neurone artificiel (formel)

Le neurone artificiel est un modèle mathématique imitant le comportement d'un neurone biologique. Il est perçu comme un processeur élémentaire (une unité de calcul) qui reçoit un nombre variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associé un poids w abréviation de weight (poids en anglais) représentatif de la force de la connexion. Chaque processeur est doté d'une sortie unique, qui se ramifie ensuite pour être envoyé vers un nombre variable de neurones aval. La structure générale d'un neurone formel est représentée dans la figure I-2.

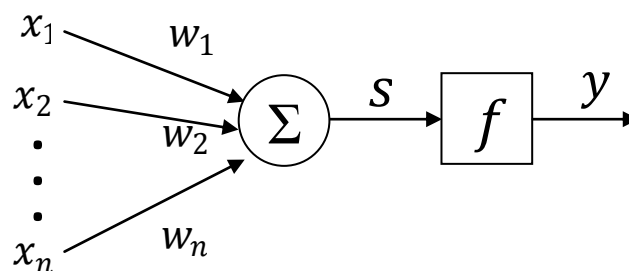


Figure I-2 Le neurone artificiel

Tel que :

n : le nombre d'entrées du neurone

x_i : la $i^{\text{ème}}$ entrée du neurone.

w_i : le $i^{\text{ème}}$ poids associé à la $i^{\text{ème}}$ entrée du neurone.

s : la somme pondérée.

f : la fonction d'activation.

y : la sortie du neurone.

Les équations mathématiques décrivant la dynamique du neurone artificiel sont données comme suit :

$$s = \sum_{i=1}^n w_i x_i \quad (I - 1)$$

$$y = f(s) \quad (I - 2)$$

Dans certains cas on ajoute une entrée additionnelle qui possède la valeur « 1 » pondérée par un poids b (appelé aussi le biais). Ainsi l'équation (I-1) deviendra :

$$s = \sum_{i=1}^n w_i x_i - b \quad (I - 3)$$

Le rôle de la fonction d'activation est d'introduire une non-linéarité dans la sortie du neurone et de décider ainsi si celui ci doit être activé ou non en calculant la somme pondérée et en ajoutant le biais. La non linéarité de la fonction d'activation rend le neurone capable d'apprendre et d'effectuer des tâches plus complexes au lieu qu'il soit juste un modèle de régression linéaire.

Plusieurs types de fonctions d'activation peuvent être envisagés, mais les plus utilisés sont :

➤ La fonction linéaire :

$$f(x) = x \quad (I - 4)$$

➤ La fonction échelon (tout ou rien) :

$$f(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases} \quad (I - 5)$$

➤ La fonction sigmoïde unipolaire :

$$f(x) = \frac{1}{1 + e^{-x}} \quad (I - 6)$$

➤ La fonction sigmoïde bipolaire :

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (I - 7)$$

4- Réseaux de neurones artificiels

Un réseau neuronal se compose d'un nombre variable d'éléments de traitement qui sont les neurones. Chaque neurone est connecté à d'autres neurones au moyen de liaisons de communication dirigées, chacune avec un poids associé. Ces poids représentent les informations utilisées par le réseau pour résoudre un problème donné.

Chaque neurone possède un état interne, appelé niveau d'activation ou d'activité, qui est en fonction des entrées qu'il a reçues. En règle générale, un neurone envoie son activation en tant que signal à plusieurs autres neurones.

La disposition des neurones dans un réseau se fait d'une manière ordonnée en respectant une certaine méthodologie. Généralement les neurones sont disposés sous forme de couches en commençant par la couche d'entrée, suivie d'une ou plusieurs couches cachées, et en terminant par une couche de sortie. Tous les neurones se trouvant sur une couche utilisent la même fonction d'activation. L'architecture neuronale varie selon la tâche que doit effectuer le réseau, mais on peut regrouper ces topologies neuronales en deux grandes classes : les réseaux statiques (non bouclés) et les réseaux récurrents (bouclés).

4-1- Réseaux statiques

Un réseau de neurones non bouclé peut être représenté par un ensemble de neurones reliés entre eux par des liaisons ou des connexions, où les données circulent à partir des entrées vers les sorties sans retour en arrière. Le terme « acyclique » est utilisé souvent pour désigner ce type de réseau car si on prend un neurone quelconque comme point de départ, alors en suivant n'importe quelle connexion on ne peut jamais revenir à ce neurone de départ. Le plus populaire des réseaux de neurones statiques est le Réseau Multi Couches (RMC).

Dans la topologie du RMC, les neurones sont disposés en différentes couches parallèles. Plusieurs entrées externes constituent la première couche appelée la couche d'entrée, tandis que la dernière couche est la couche de sortie qui délivre les sorties du réseau. Entre ces deux couches il existe plusieurs couches (une ou plusieurs) appelées couches cachées. Ces dernières assurent la liaison entre la couche d'entrée et celle de la sortie et effectuent les calculs

intermédiaires du réseau. Un exemple de réseau RMC à une seule couche cachée est illustré dans la figure I-3.

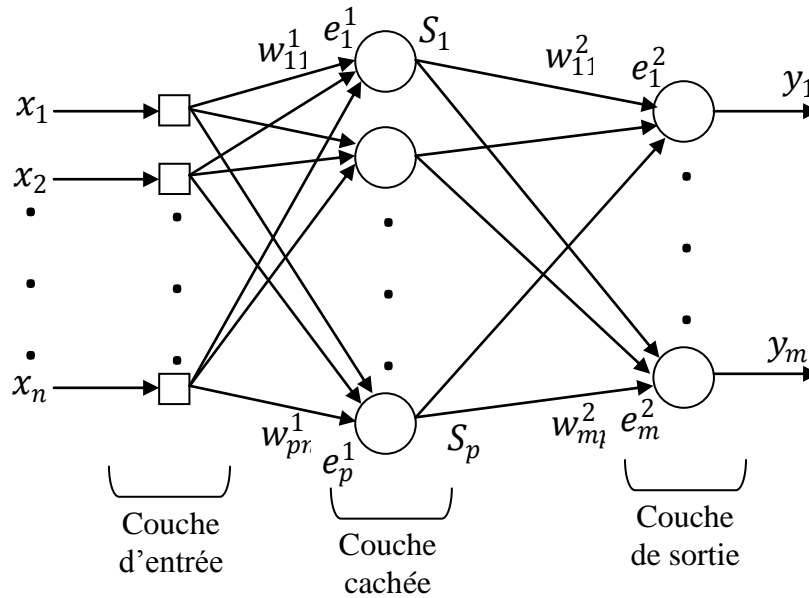


Figure I-3 Exemple de réseau RMC à une seule couche cachée

On définit :

n : le nombre d'entrées du réseau.

p : le nombre de neurones dans la couche cachée.

m : le nombre de neurones dans la couche de sortie.

$x_i(k)$: la $i^{\text{ème}}$ entrée du réseau à l'instant k .

$e_j^1(k)$: l'entrée du $j^{\text{ème}}$ neurone caché à l'instant k .

$e_h^2(k)$: l'entrée du $h^{\text{ème}}$ neurone de sortie à l'instant k .

$S_j(k)$: la sortie du $j^{\text{ème}}$ neurone caché à l'instant k .

$y_h(k)$: la sortie du $h^{\text{ème}}$ neurone dans la couche de sortie à l'instant k .

w_{ji}^1 : le poids de la connexion reliant la $i^{\text{ème}}$ entrée x_i au $j^{\text{ème}}$ neurone caché.

w_{hj}^2 : le poids de la connexion reliant le $h^{\text{ème}}$ neurone de sortie au $j^{\text{ème}}$ neurone caché.

b_j^1 : le biais du $j^{\text{ème}}$ neurone caché.

b_h^2 : le biais du $h^{\text{ème}}$ neurone de sortie.

f_1 : la fonction d'activation dans la couche cachée.

f_2 : la fonction d'activation dans la couche de sortie.

La dynamique du réseau RMC peut être décrite par les équations suivantes :

L'entrée du $j^{\text{ème}}$ neurone caché :

$$e_j^1(k) = \sum_{i=1}^n w_{ji}^1 x_i(k) + b_j^1, \quad j = 1, 2, \dots, p \quad (I - 8)$$

Sa sortie :

$$S_j(k) = f_1(e_j^1(k)) \quad (I - 9)$$

L'entrée du h^{ème} neurone de sortie :

$$e_h^2(k) = \sum_{j=1}^p w_{hj}^2 S_j(k) + b_h^2, \quad h = 1, 2, \dots, m \quad (I - 10)$$

Sa sortie :

$$y_h(k) = f_2(e_h^2(k)) \quad (I - 11)$$

4-2- Réseaux récurrents

Contrairement aux réseaux de neurones statiques, les réseaux de neurones récurrents (ou dynamique) peuvent avoir une topologie de connexions quelconque. Chaque neurone peut recevoir comme entrées les valeurs passées des autres neurones (que ce soit de la couche qui le précède ou bien même à partir des neurones de sa propre couche). Cette particularité d'interconnexion des neurones fait en sorte que ce type de réseaux puisse être utilisé dans la résolution de problèmes complexes. L'une des architectures récurrentes les plus simples et les plus populaires est celle du réseau d'Elman. Ce réseau proposé par J.L. Elman en 1990 ressemble à un réseau RMC à une seule couche cachée avec ajout d'une couche de contexte qui représente une copie de l'activation de la couche cachée calculée à l'instant précédent. L'architecture du réseau d'Elman est donnée dans la figure (I-4).

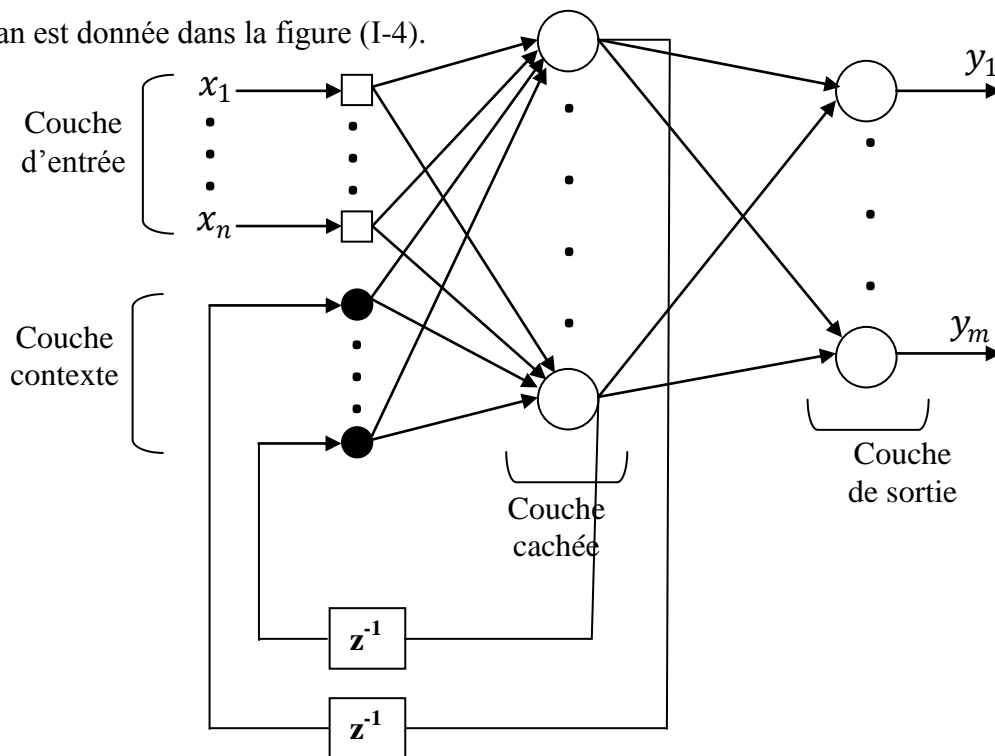


Figure I-4 Architecture du réseau d'Elman

On utilise les mêmes notations que pour le réseau RMC, de plus on définit :

w_{ji}^3 : le poids de la connexion reliant la $i^{\text{ème}}$ entrée de la couche contexte au $j^{\text{ème}}$ neurone caché.

Les mêmes équations décrivant le comportement du réseau RMC sont utilisées aussi pour le réseau d'Elman sauf pour l'entrée du $j^{\text{ème}}$ neurone caché qui sera réécrite comme suit:

$$e_j^1(k) = \sum_{i=1}^n w_{ji}^1 x_i(k) + b_j^1 + \sum_{i=1}^p w_{ji}^3 S_i(k-1) , j = 1, 2, \dots, p \quad (I - 12)$$

5- Apprentissage des RNA

L'apprentissage (ou l'entraînement) représente la propriété la plus intéressante des RNA. Par définition l'apprentissage est décrit comme une étape d'évolution d'un réseau de neurones durant laquelle le comportement du réseau neuronal est modifié jusqu'à obtenir le comportement souhaité. L'étape d'apprentissage vient juste après le choix de l'architecture du réseau de neurones. On associe pour cette étape un algorithme qui représente la démarche ou le processus utilisé pour mener à bien la tâche d'apprentissage.

Dans la plupart des algorithmes actuels, les variables adaptées ou modifiées pendant l'apprentissage sont les poids des connexions entre les neurones. On parle alors d'adaptation ou d'ajustement des poids. Cette opération d'adaptation des poids permet d'assurer que la réponse du réseau soit la plus proche possible d'une réponse désirée prédéfinie

. Après cette phase d'apprentissage, les valeurs des poids sont fixées et on procède ensuite à l'étape de validation du réseau qui consiste à tester, à partir de la réponse du réseau, si les poids obtenus sont optimaux sinon on refait la phase d'apprentissage.

On distingue plusieurs types d'apprentissage :

5-1 Apprentissage supervisé

L'apprentissage supervisé, comme son nom l'indique, consiste à superviser (surveiller) l'entraînement du réseau en lui donnant des **exemples** (des données) d'entrée/sortie de la tâche qu'il doit réaliser. L'apprentissage est effectué en variant les poids du réseau afin de minimiser un certain critère d'erreur. Cet entraînement est réitéré plusieurs fois jusqu'à ce qu'il n'y ait plus de changement considérable dans les poids du réseau.

5-2 Apprentissage non supervisé

Contrairement au type supervisé, l'apprentissage non supervisé (ou l'apprentissage auto-organisé) n'a pas besoin des entrées et des sorties pour entraîner le réseau. Ce dernier exploite les données qu'on lui présente et cherche à les regrouper en des catégories ou des classes. Dans ce type d'apprentissage la sortie désirée du réseau n'est pas fixée, par contre dans l'apprentissage supervisé le vecteur de sortie désirée est connu pour chaque forme d'entrée et il est emmagasiné dans la base d'apprentissage.

5-3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé combine les propriétés des deux types d'apprentissage supervisé et non supervisé. Dans ce type d'apprentissage, on utilise un ensemble de données étiquetées (entrée avec étiquette) et non étiquetées (l'entrée seule est fournie). L'intérêt de l'apprentissage semi-supervisé résulte du fait que lorsque les données présentées au réseau sont très grands, alors leur étiquetage nécessite l'intervention d'un utilisateur humain.

5-4 Apprentissage profond

Lorsqu'on présente au réseau de neurones des données brutes il est nécessaire des fois de passer par une étape d'extraction des caractéristiques de ces données afin de mener à bien la tâche d'apprentissage. L'utilisation des techniques d'apprentissage classiques peut engendrer des mauvais résultats dans ce cas là, sans parler du coût élevé en terme de mémorisation et la possibilité d'une mauvaise extraction des caractéristiques.

L'apprentissage profond (Deep learning) est une nouvelle « branche » d'apprentissage dont l'idée est apparue en 2006 regroupant une catégorie d'algorithmes d'apprentissage qui utilisent plusieurs couches d'unité de traitement non linéaire pour l'extraction et la transformation des caractéristiques. Pour les RNA, l'apprentissage profond a une relation avec le nombre de couches du réseau car plus le réseau possède de couches cachées et mieux sera l'apprentissage.

6- Exemples de règles d'apprentissage des RNA

Plusieurs règles et algorithmes ont été proposés par les chercheurs pour les différents types d'apprentissage. Nous citerons quelques une de ces règles ainsi que l'algorithme de rétropropagation qui est très utilisé dans les RNA.

6-1 Règle de Hebb

La règle de Hebb a été formulée par D. Hebb en 1949 sous l'énoncé suivant : « Quand une cellule A est assez proche d'une cellule B pour l'exciter, et quand cette excitation engendrée par A soit répétée et de manière persistante, alors un processus de croissance ou un changement

métabolique s'opère dans l'une (ou les deux cellules) de telle sorte que l'efficacité de A s'accroît à cet effet».

Cet énoncé peut être résumé à l'idée de consolider les connexions entre deux neurones lorsque ceux-ci sont activés simultanément.

Considérons, dans la figure I-5, deux neurones « i » et « j » avec leurs sorties respectives y_i et y_j ainsi que le poids de connexion noté w_{ji}

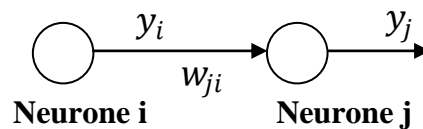


Figure I-5 Représentation schématique de la règle de Hebb

La modification des poids suivant la règle de Hebb est donnée par l'équation suivante :

$$w_{ji}(k+1) = w_{ji}(k) + \mu y_i(k) y_j(k) \quad (I-13)$$

Tel que μ représente le facteur d'apprentissage.

6-2 Règle de Widrow-Hoff (règle du Delta)

La règle de Widrow-Hoff (ou règle Delta) a été proposée par Bernard Widrow et Marcian Edward Hoff en 1960 comme règle d'apprentissage pour l'Adaline (un réseau neuronal à une simple couche). Le principe de cette méthode consiste en la minimisation d'une erreur quadratique calculée en fonction des réponses désirées et des sorties obtenues par le réseau neuronal. Les poids de connexions sont choisis au début aléatoirement puis ils sont adaptés jusqu'à ce que le réseau soit capable de fournir les sorties désirées.

La modification des poids utilisant la règle de Widrow-Hoff est donnée par la formule suivante :

$$w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k) \quad (I-14)$$

Tel que :

$$\Delta w_{ji}(k) = \mu (y_{dj}(k) - y_j(k)) y_i(k) \quad (I-15)$$

Avec :

$y_{dj}(k)$: les sorties désirées à l'instant k .

6-3 Algorithme de rétropropagation

Cette technique d'apprentissage est la plus connue et la plus utilisée parmi les algorithmes d'apprentissage des RNA. Il est basé sur la méthode de descente du gradient étendue à chacune des couches du réseau par la règle de chaînage.

6.3.1 Historique

- 1961 : le concept de base de la rétropropagation fut proposé dans le contexte de la théorie du contrôle par J. Kelly, H. Arthur et E. Bryson.
- 1962 : dérivation de l'algorithme par la règle de chaînage élaborée par S. Dreyfus
- 1970 : mise en œuvre de l'algorithme pour fonctionner sur des ordinateurs par S. Linnainmaa
- 1974 : P. Werbos mentionne la possibilité d'appliquer le principe de rétropropagation aux RNA.
- 1986 : les travaux de D. E. Rumelhart , G. E. Hinton , R. J. Williams et J. McClelland ont permis de populariser l'algorithme.
- 1987 : Y. LeCun propose la forme moderne de l'algorithme d'apprentissage de rétropropagation dans sa thèse de doctorat.
- 1993 : Wan a pu remporter un prix international dans le concours de reconnaissance de formes à l'aide de l'algorithme de rétropropagation

Au cours des années 2000 l'intérêt pour l'algorithme de rétropropagation a connu un léger recul mais à partir des années 2010, et grâce au développement et l'accessibilité des unités de traitement graphique, l'algorithme a repris de l'attention chez les chercheurs.

6.3.2 Principe et procédure

La rétropropagation est une généralisation de la règle du Delta pour les réseaux à plusieurs couches. Le but de cet algorithme est la modification des poids du réseau neuronal dans le sens contraire du gradient de l'erreur afin de satisfaire un certain critère de performances. Cette méthode est basée sur l'algorithme de descente du gradient et elle utilise les règles de calcul des dérivées ce qui implique que les fonctions d'activation des neurones doivent être dérivables.

L'erreur commise en sortie du réseau sera rétropropagée vers les couches cachées d'où le nom de rétropropagation.

La procédure de l'algorithme de rétropropagation peut être résumée dans ces étapes :

1. Initialiser les poids du réseau à de petites valeurs.

2. Injecter au réseau des entrées et calculer les sorties correspondantes.
3. Calculer les erreurs entre les sorties du réseau et les sorties désirées.
4. Rétropager les erreurs à toutes les couches du réseau.
5. Ajuster les poids du réseau.
6. Répéter les étapes (2) à (5) pour l'ensemble des exemples d'entrée ou bien jusqu'à atteindre une valeur prédéfinie du critère d'erreur.

6-3-3 Exemple d'application sur le RMC

Pour montrer l'utilisation de l'algorithme de rétropropagation, nous allons l'appliquer sur un réseau RMC.

Considérons un réseau RMC à une seule couche cachée contenant « p » neurones avec « n » entrées et « m » sorties.

Soit le critère suivant à minimiser :

$$E = \frac{1}{2} \sum_{k=1}^N \sum_{h=1}^m (y_h(k) - y_h^d(k))^2 \quad (I - 16)$$

Tel que :

N : le nombre d'exemples présentés au réseau.

$y_h(k)$: la $h^{\text{ème}}$ sortie du réseau à l'instant k.

$y_h^d(k)$: la $h^{\text{ème}}$ sortie désirée à l'instant k.

L'adaptation des poids du réseau se fera avec les équations suivantes :

$$w_{ji}^1(k+1) = w_{ji}^1(k) - \mu \frac{\partial E}{\partial w_{ji}^1(k)} \quad (I - 17)$$

$$w_{hj}^2(k+1) = w_{hj}^2(k) - \mu \frac{\partial E}{\partial w_{hj}^2(k)} \quad (I - 18)$$

$$b_j^1(k+1) = b_j^1(k) - \mu \frac{\partial E}{\partial b_j^1(k)} \quad (I - 19)$$

$$b_h^2(k+1) = b_h^2(k) - \mu \frac{\partial E}{\partial b_h^2(k)} \quad (I - 20)$$

En utilisant la dérivation par chaîne on obtient les expressions :

$$\frac{\partial E}{\partial w_{ji}^1(k)} = \frac{\partial E}{\partial y_h(k)} \frac{\partial y_h(k)}{\partial e_h^2(k)} \frac{\partial e_h^2(k)}{\partial S_j(k)} \frac{\partial S_j(k)}{\partial e_j^1(k)} \frac{\partial e_j^1(k)}{\partial w_{ji}^1(k)} \quad (I - 21)$$

$$\frac{\partial E}{\partial w_{hj}^2(k)} = \frac{\partial E}{\partial y_h(k)} \frac{\partial y_h(k)}{\partial e_h^2(k)} \frac{\partial e_h^2(k)}{\partial w_{hj}^2(k)} \quad (I - 22)$$

$$\frac{\partial E}{\partial b_j^1(k)} = \frac{\partial E}{\partial y_h(k)} \frac{\partial y_h(k)}{\partial e_h^2(k)} \frac{\partial e_h^2(k)}{\partial S_j(k)} \frac{\partial S_j(k)}{\partial e_j^1(k)} \frac{\partial e_j^1(k)}{\partial b_j^1(k)} \quad (I - 23)$$

$$\frac{\partial E}{\partial b_h^2(k)} = \frac{\partial E}{\partial y_h(k)} \frac{\partial y_h(k)}{\partial e_h^2(k)} \frac{\partial e_h^2(k)}{\partial b_h^2(k)} \quad (I - 24)$$

Et à partir des équations (I-8) jusqu'à (I-11), on calcule les différentes dérivées partielles :

$$\frac{\partial E}{\partial w_{ji}^1(k)} = (y_h(k) - y_h^d(k)) f_2'(e_h^2(k)) w_{hj}^2(k) f_1'(e_j^1(k)) x_i(k) \quad (I - 25)$$

$$\frac{\partial E}{\partial w_{hj}^2(k)} = (y_h(k) - y_h^d(k)) f_2'(e_h^2(k)) S_j(k) \quad (I - 26)$$

$$\frac{\partial E}{\partial b_j^1(k)} = (y_h(k) - y_h^d(k)) f_2'(e_h^2(k)) w_{hj}^2(k) f_1'(e_j^1(k)) \quad (I - 27)$$

$$\frac{\partial E}{\partial b_h^2(k)} = (y_h(k) - y_h^d(k)) f_2'(e_h^2(k)) \quad (I - 28)$$

7- Domaines d'application des RNA

Les réseaux de neurones ont su attirer l'attention de beaucoup de chercheurs grâce aux propriétés intéressantes qu'ils possèdent, et l'intérêt porté pour ces réseaux les a rendu utilisables dans plusieurs domaines, parmi lesquels nous pouvons citer :

- Identification des systèmes non linéaires.
- Commande des robots.
- Reconnaissance d'images.
- Reconnaissance de la parole.
- Reconnaissance de caractères.
- Prévisions financières.
- Diagnostics médicaux.

8- Avantages et inconvénients des RNA

Bien que les réseaux de neurones ont su prouver leur efficacité dans plusieurs domaines, néanmoins ils possèdent des avantages ainsi que des inconvénients qui sont résumés par [25] :

8.1 Avantages

- Les réseaux de neurones sont capables d'approximer des fonctions complexes non linéaires.
- Les réseaux de neurones sont flexibles vis-à-vis des données incomplètes et/ou bruitées.
- Les réseaux de neurones peuvent être adaptables avec de nouvelles données, ce qui les rend utile pour les environnements dynamiques.
- Les réseaux de neurones surmontent certaines limitations de quelques méthodes statistiques en les généralisant.
- Les réseaux de neurones peuvent être implémentés sur un support matériel.
- Les performances des réseaux neuronaux peuvent être automatisées, minimisant l'implication humaine.
- Les réseaux de neurones sont adaptés pour résoudre différents problèmes dans beaucoup de domaines.

8.2 Inconvénients

- Les réseaux de neurones fonctionnent comme des «boîtes noires».
- Le choix de la topologie du réseau ainsi que les paramètres de l'algorithme d'entraînement est heuristique
- L'estimation des poids du réseau nécessite de grandes quantités de données,
- Il n'y a pas de règles prédéfinies pour sélectionner l'algorithme le plus approprié pour l'apprentissage.
- Les réseaux de neurones peuvent converger vers un minimum local.
- Aucune garantie sur la convergence des algorithmes d'apprentissage

Chapitre 2

La logique floue

1- Introduction

Un intérêt considérable a été suscité ces dernières années par la logique floue au sein de la communauté scientifique. L'une des principales raisons est le succès étonnant qu'a connu la fabrication des équipements domestiques destinés au grand public issus de l'industrie japonaise. Les systèmes flous peuvent être classés dans la catégorie des « systèmes à base de connaissance » dont l'objectif principal est la mise en œuvre d'un savoir-faire humain ou bien proposer des règles heuristiques sous forme de programme informatique.

Des règles linguistiques de la forme 'if...then...' sont utilisées pour modéliser l'expérience humaine et l'interpréter sous forme de système flou. Les méthodes ou les procédés fondés sur la logique floue représentent une solution prometteuse dans certaines tâches telles que la commande des systèmes possédant des non linéarités ou bien encore des systèmes dont on ne trouve pas de modèle mathématique pour les modéliser.

Les opinions varient largement en ce qui concerne l'utilité de la théorie floue. Certains pensent que cette théorie constitue uniquement une extension de la logique classique. D'autres chercheurs, au contraire, travaillent à l'amélioration de cette théorie et en l'utilisant contribuent dans divers domaines scientifiques, à savoir : le diagnostic médical, la robotique,...etc.

L'utilisation vaste de la logique floue dans de nombreux domaines est due à plusieurs facteurs dont les principaux sont :

- a) L'introduction directe des informations floues et linguistiques, issues d'un expert humain, dans le système flou.
- b) La non nécessité d'un modèle mathématique pour la mise en œuvre d'un système flou.
- c) Facilité de compréhension des règles de la logique floue

2- Historique

La logique floue prend ses origines durant les années 20 avec l'introduction, par des physiciens et en utilisant le principe d'incertitude de Heisenberg, d'une 3^{ème} valeur « 0.5 » dans le système binaire {0,1}.

Durant la même période, le logicien polonais Jan Lukasiewicz a proposé des systèmes logiques basés sur trois propositions {vraie, fausse, possible}, puis cette logique a été élargie à tous les nombres rationnels compris entre 0 et 1.

Le philosophe Max Black, durant les années 30, a étendu le concept de logique floue sur des ensembles d'éléments ou de symboles. L'incertitude de ces ensembles a été appelée par lui « imprécision ». C'est à lui aussi que revient la proposition de la première fonction d'appartenance d'un ensemble flou.

Un article intitulé “*Fuzzy sets*” a paru en 1965 dont l'auteur est un enseignant chercheur de nationalité iranienne Lotfi Zadeh. Dans cet article ce chercheur a incorporé la théorie des ensembles flous dans la littérature technique, ce qui a provoqué le commencement des essais de modélisation des systèmes en utilisant les relations floues. Zadeh a été le premier qui a formalisé les règles floues et à proposer la notion de « système expert » appliquée au réglage automatique dans laquelle la loi de commande est substituée par des règles floues.

En 1975 les deux chercheurs Mandani et Assilian ont réussi à obtenir les premiers résultats sur la commande floue encourageant ainsi plusieurs recherches dans différents pays tels que le Danemark, l'Angleterre et la France.

A partir des années 80, les japonais ont poussé les recherches dans ce domaine et ils ont réussi à lancer plusieurs produits basé sur la logique floue spécialement les appareils électroménagers et des appareils audio.

3- Notions d'ensemble flou et fonction d'appartenance

3.1. Définition

Dans la théorie des ensembles classiques, la fonction d'appartenance $\mu_F(x)$ est définie comme suit :

$$\mu_F(x) = \begin{cases} 1 & \text{si } x \in F \\ 0 & \text{si } x \notin F \end{cases} \quad (II - 1)$$

Cette définition signifie que l'élément x a deux possibilités : est un élément de F si $\mu_F(x) = 1$, et non si $\mu_F(x) = 0$. Donc l'ensemble classique est un ensemble dans lequel l'appartenance est brusque. Cependant, dans plusieurs cas il n'est pas clair qu'un élément x appartient à un ensemble F ou non, par exemple, F représente les PC trop chers pour le budget des étudiants, le prix du PC est fixé à 30000 DA, mais les bornes de cet ensemble ne sont pas bien définies : qu'est ce qu'on peut dire sur les PC de prix 29900 DA et 30100 DA ? Ces deux PC sont trop chers ou non ? Il est clair qu'on doit déterminer la limite au-dessus de laquelle le prix du PC est trop cher pour le budget moyen de l'étudiant.

Dans cet intervalle, une classification graduelle doit être utilisée pour les prix partiellement trop chers. Cette classification est appelée un *ensemble flou* dans lequel l'appartenance est graduelle dans l'intervalle $[0,1]$.

Donc par définition, un ensemble flou est un ensemble dans lequel l'appartenance d'une variable à un ensemble est partielle dans l'intervalle $[0,1]$.

L'ensemble flou F , est un sous ensemble de U , défini par l'ensemble des paires ordonnées :

$$F = \{ (x, \mu_F(x)) / x \in U \}$$

Tel que :

$\mu_F(x)$: est une valeur réelle qui prend des valeurs comprises entre 0 et 1, appelée la *fonction d'appartenance*. Elle caractérise l'ensemble flou F et fournit une mesure du degré d'appartenance d'un objet x de U dans F .

U est appelé *l'univers de discours*, il peut contenir des valeurs discrètes ou continues

La figure (II-1) montre un exemple d'un ensemble flou qui représente les PCs cher pour le budget de l'étudiant.

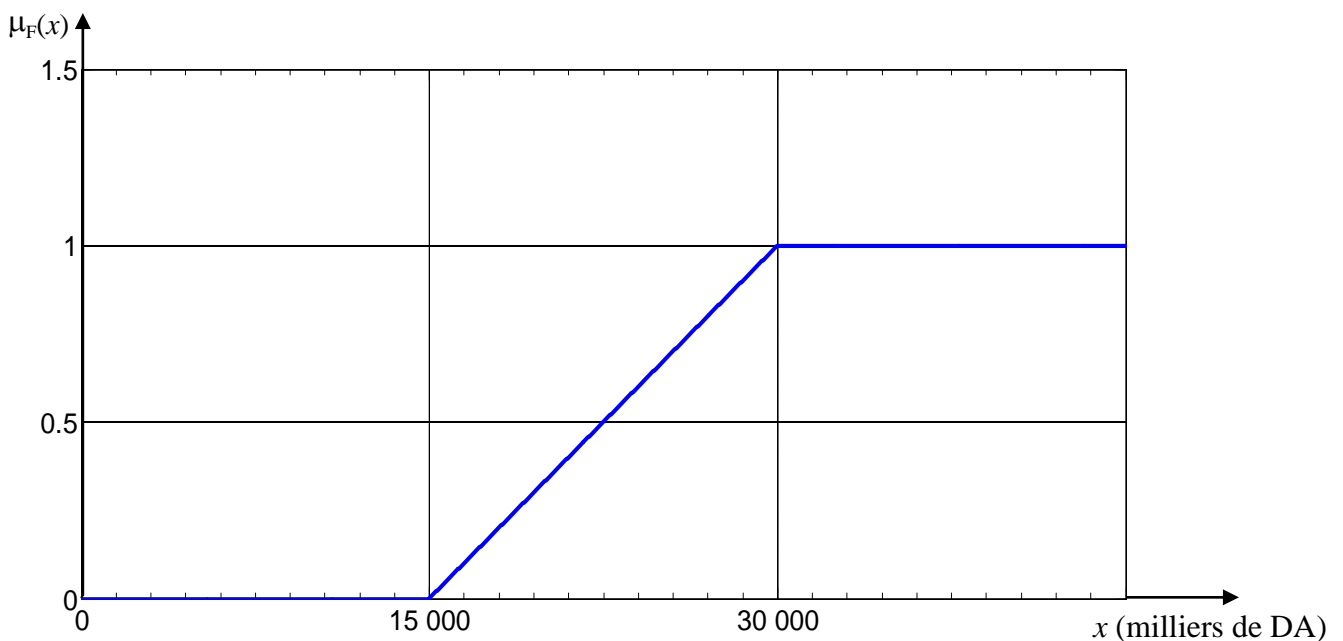


Figure II-1 Exemple de fonction d'appartenance d'ensemble flou qui représente les prix des PCs

La figure en bleue représente la fonction d'appartenance de la variable x

On peut observer que le prix des PCs au-dessus de 15000 DA n'est pas trop cher, et les prix des PCs au-dessus de 30000 DA sont considérés comme des prix trop chers.

Pour les autres prix, on peut observer une augmentation graduelle du degré d'appartenance dans l'ensemble trop cher.

Pour comprendre la différence entre la logique classique et la logique booléenne, considérons les figures (II-2) et (II-3) qui montrent deux exemples de représentation de la température, une en logique classique, et l'autre en logique floue.

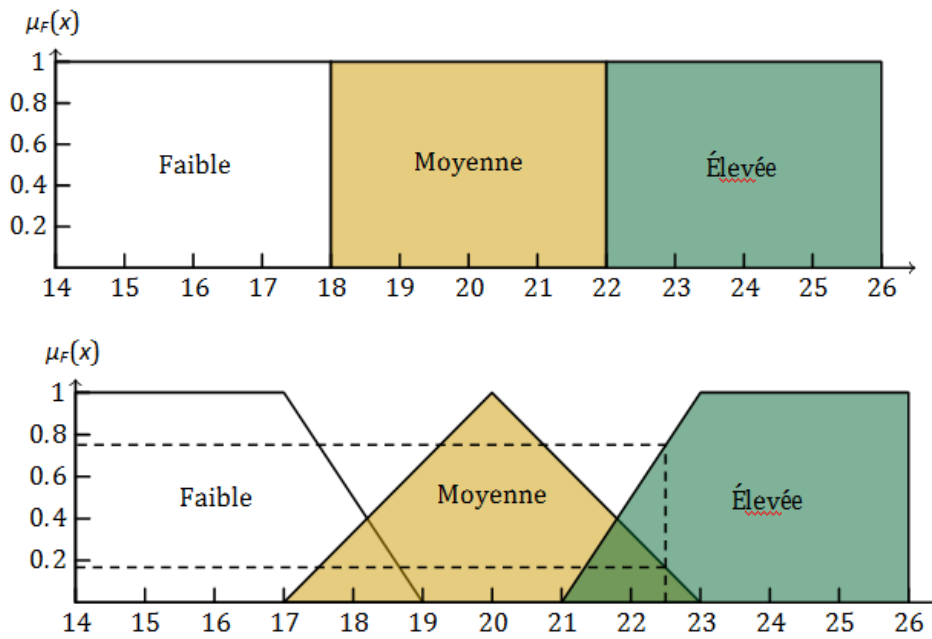


Figure II-2 Représentation de la température en logique floue

Selon les deux figures, en logique classique, une température de 22.5° est considérée comme élevée. En logique floue, une température de 22.5° appartient au groupe "moyenne" avec un degré d'appartenance de 0.167, et appartient au groupe "élevée" avec un degré d'appartenance de 0.75

Les variables floues *Faible*, *Moyenne* et *Elevée* sont représentées par des fonctions linéaires.

D'autres fonction peuvent être utilisées, comme des trapézoïdes, des paraboles, etc. Cependant, les fonction linéaires sont beaucoup plus faciles à implémenter de façon pratique, et donnent de bons résultats.

On utilise souvent une notation vectorielle pour représenter les fonctions. Pour les fonctions d'appartenance de la figure (II-3), pour la logique floue, on peut utiliser la notation suivante :

- Température faible : (1/17, 0/19)
- Température moyenne : (0/17, 1/20, 0/23)

- Température élevée : (0/21, 1/23)

3.2. Propriétés des ensembles flous

Les propriétés généralement utilisées dans la théorie des ensembles flous sont :

La hauteur :

La hauteur d'un ensemble flou F est la plus grande valeur du degré d'appartenance $\mu_F(x)$.

Elle est notée par $h(F)$ et elle est définie par :

$$h(F) = \max\{\mu_F(x)/x \in U\}$$

Un ensemble flou est appelé normal si $h(F)=1$, et sous normal si $h(F) < 1$.

Le noyau :

Le noyau d'un ensemble flou F est l'ensemble des éléments qui appartiennent totalement à F c'est-à-dire dont le degré d'appartenance à F vaut 1. Il est noté par « n » et défini par :

$$n(F) = \{x \in U/\mu_F(x) = 1\}$$

S'il y a un seul point x de degré égal à 1, alors ce point est dit *le point pic* de F

a) Le support :

On appelle support d'un ensemble flou F dans U l'ensemble des valeurs x dans U tel que $\mu_F(x) > 0$. il est noté par $S(F)$, et il est défini par :

$$S(F) = \{x \in U/\mu_F(x) > 0\}$$

b) α -coupure :

Une α -coupure d'un ensemble flou F est l'ensemble des éléments x ayant un degré d'appartenance supérieur ou égal à α . On le note par :

$$\alpha - \text{coupe}(F) = \{x \in U/\mu_F(x) \geq \alpha\}$$

3.3. Opérations sur les ensembles flous

Les opérations appliquées sur les ensembles classiques tel que le complément, l'union et l'intersection peuvent être aussi appliquées sur les ensembles flous puisque les valeurs de la fonction d'appartenance ne se limitent pas aux valeurs de l'ensemble $\{0,1\}$ uniquement mais elles s'étendent à toutes les valeurs réelles de l'intervalle $[0, 1]$

Seulement, les opérations floues ne sont pas représentées de la même façon que celles appliquées aux ensembles classiques.

Soient A , B et C des ensembles flous dans U décrits, respectivement, par leurs fonctions d'appartenances : $\mu_A(x)$, $\mu_B(x)$ et $\mu_C(x)$.

L'union de A et B , notée $A \cup B$, est définie par : $\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$

L'intersection de A et B , notée $A \cap B$, est définie par : $\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$

Le complément de A , noté \bar{A} , est défini par : $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

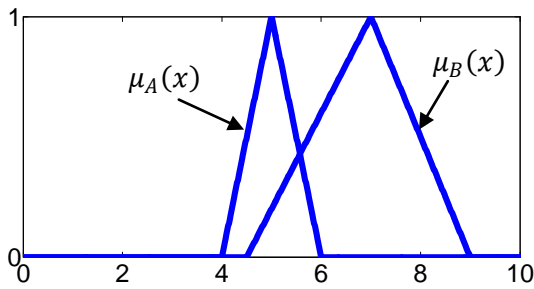


Figure II-3 Les ensembles flous A et B

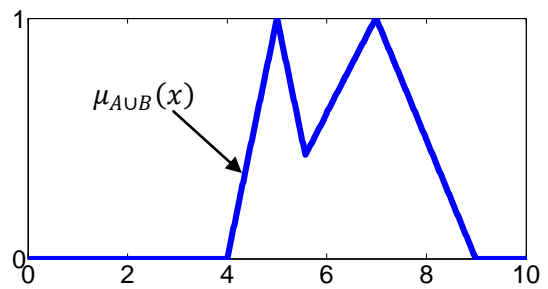


Figure II-4 L'ensemble flou $A \cup B$

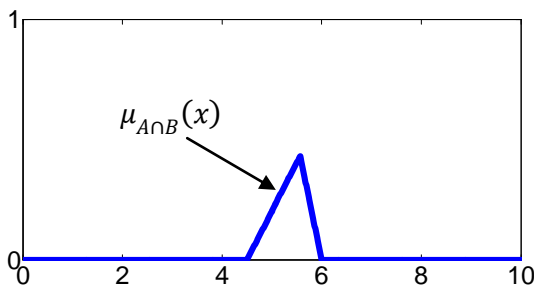


Figure II-5 L'ensemble flou $A \cap B$

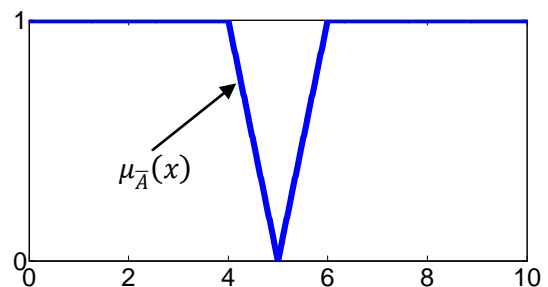


Figure II-6 L'ensemble flou \bar{A}

4- Règles d'inférence

On appelle règles d'inférence (ou règles floues), l'ensemble des différentes règles reliant les variables floues d'entrée d'un système aux variables floues de sortie de ce système. Ces règles se présentent sous la forme :

SI x est A **ALORS** y est B

Tel que x et y sont des variables linguistiques, et A et B sont des valeurs linguistiques, déterminées par les ensembles flous sur les ensembles X et Y . Une variable linguistique est une variable floue.

La partie « x est A » est appelée « prémisses » ou « antécédent »

La partie « y est B » est appelée « conséquence » ou « action »

Par exemple : La tension est élevée. La variable linguistique *tension* prend la valeur linguistique *élevée*. La plage de valeurs linguistiques possibles d'une règle représente l'univers de cette variable. Un exemple de règle floue est :

SI vitesse est lente ALORS arrêt est court

La variable *vitesse* peut avoir une plage de valeurs entre 0 et 220 km/h. On peut inclure des sous-ensembles flous (très lent, lent, moyenne, rapide, très rapide) pour modifier cette règle. Chaque sous-ensemble flou représente une valeur linguistique pour la variable.

5- Système flou

Un système flou est composé de quatre blocs essentiels comme le montre la figure (II-7).

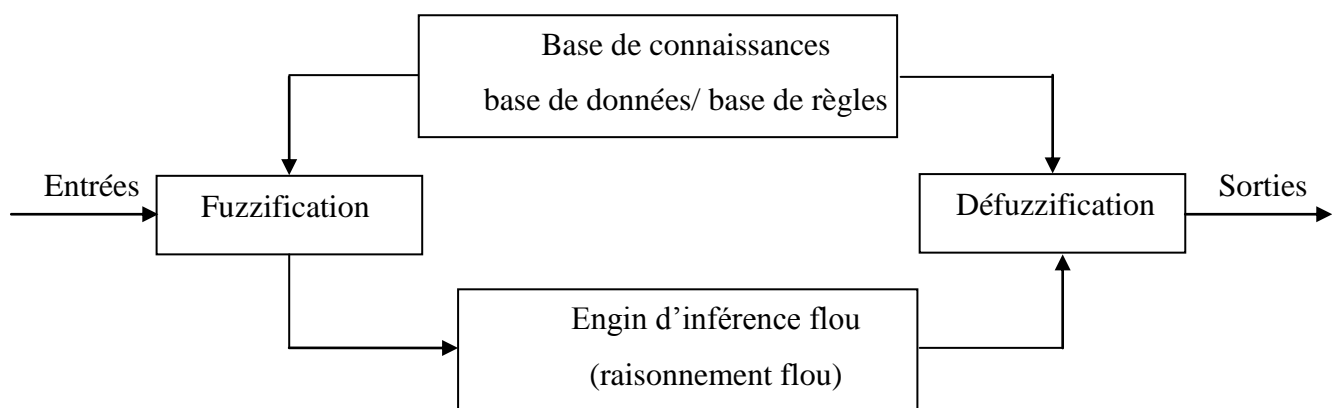


Figure II-7 Structure général d'un système flou

5.1. Fuzzification

La fuzzification est une opération qui transforme une entrée numérique observée en une variable linguistique. On effectue cette transformation par attribution de degrés d'appartenance à chaque valeur d'entrée. Les variables linguistiques sont définies par leurs valeurs linguistiques. En général, les fonctions d'appartenance qui représentent les valeurs linguistiques sont définies en forme triangulaire, trapézoïdale ou en forme gaussienne.

Il n'y a pas de règle précise pour la définition de fonction d'appartenance. On peut introduire pour une variable x trois, cinq ou sept (en général un nombre impair) valeurs linguistiques. Un exemple de désignation d'ensembles flous :

- **NG** : Négatif Grand
- **NM** : Négatif Moyen
- **NP** : Négatif Petit
- **Z** : Zéro

- **PP** : Positif Petit
- **PM** : Positif Moyen
- **PG** : Positif Grand

5.2. Base de connaissances

L'engin d'inférence est composé de deux bases : la première formée de données et la seconde de règles floues. Les fonctions d'appartenance des ensembles flous se trouvent dans la base de données, tandis que la base de règles regroupe un ensemble de règles floues de type **SI-ALORS (IF-THEN)** définissant la relation entre les antécédents décrivant l'état du système et les conséquences formulent l'action de l'opérateur qui commande le système

5.3. Engin d'inférence

Il consiste à associer à chacune des conclusions des règles une valeur de sortie. Cette dernière dépend de la partie conclusion des règles qui peut prendre plusieurs formes. Généralement, deux règles importantes sont utilisées :

Règle de Mamdani

La conclusion est une proposition floue :

SI.....ALORS y est B, tel que B est un ensemble flou

Règle de Sugeno

La conclusion est une fonction réelle :

SI.....ALORS y est $f(x_1, x_2, \dots, x_n)$, tel que x_1, x_2, \dots, x_n sont les entrées du système flou.

Remarque 1

On utilise beaucoup plus la règle de Mamdani car on obtient de meilleurs résultats avec elle. L'ensemble flou de sortie par cette règle s'obtient par plusieurs méthodes dont la plus utilisée est la méthode de coupure qui consiste à déduire une nouvelle forme de l'ensemble flou de sortie à partir des différentes fonctions d'appartenance associées aux différentes conclusions des règles. La hauteur de ce nouveau ensemble flou est déterminé à partir des valeurs d'appartenance des diverses fonctions d'appartenance de sortie.

5.4. Déffuzzification

La défuzzification est le processus de convertir une valeur floue en valeur nette. Plusieurs méthodes existent dont la plus utilisée est la méthode du centre de gravité.

La méthode du centre de gravité (CG)

Elle consiste à calculer l'abscisse du centre de gravité de la fonction d'appartenance résultante par la méthode de coupure par exemple. Le calcul se fait selon cette expression :

$$y = CG = \frac{\sum_{x=a}^b x * \mu_F(x)}{\sum_{x=a}^b \mu_F(x)}$$

Tel que :

x : valeurs de l'axe des abscisses

« a » et « b » sont les valeurs minimales et maximales des valeurs de x

Remarque 2

Le centre de gravité n'a pas besoin d'être calculé de façon précise, il suffit de prendre un certain nombre d'échantillons compris entre « a » et « b ».

6- Exemple d'application

Soit l'exemple d'un système de contrôle d'un ventilateur de maison, ayant 2 entrées (température et humidité) et une sortie (vitesse du ventilateur). On propose les règles suivantes :

SI x est A₁ OU y est B₁ ALORS z est C₁

SI x est A₂ ET y est B₂ ALORS z est C₂

SI x est A₃ ALORS z est C₃

Où « x » est la température, « y » est l'humidité et « z » est la vitesse du ventilateur. Les ensembles flous sont :

- A₁ = faible, A₂ = moyenne, A₃ = élevée
- B₁ = sec, B₂ = humide
- C₁ = lent, C₂ = moyenne, C₃ = rapide

Les ensembles flous sont donnés dans les figures suivantes :

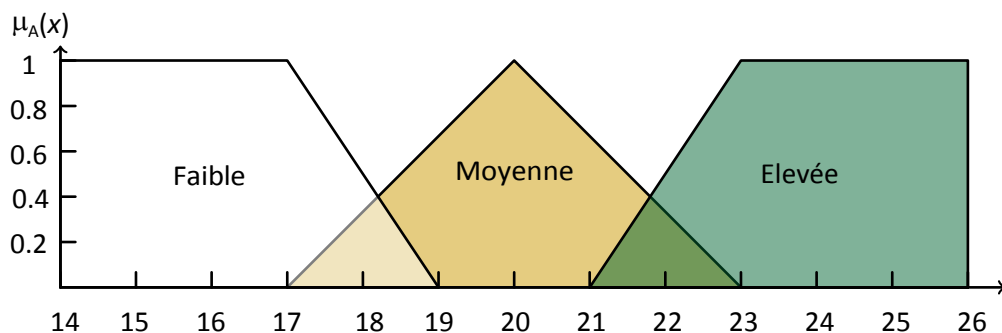


Figure II-8 Ensembles flous de la variable d'entrée : température (°) « x »

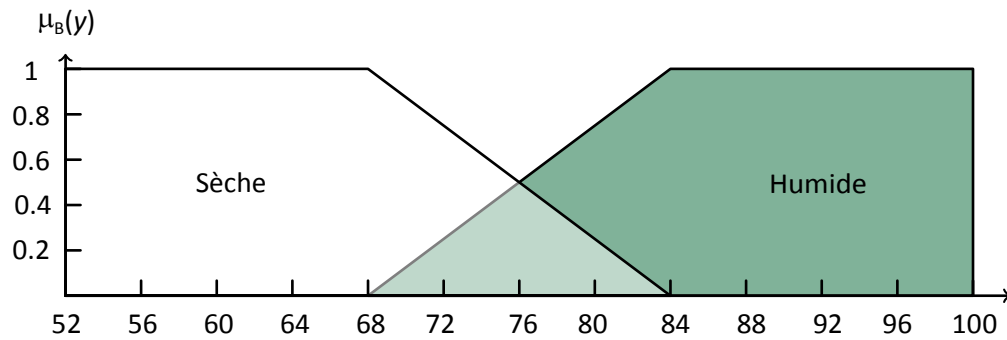


Figure II-9 Ensembles flous de la variable d'entrée : humidité (%) « y »

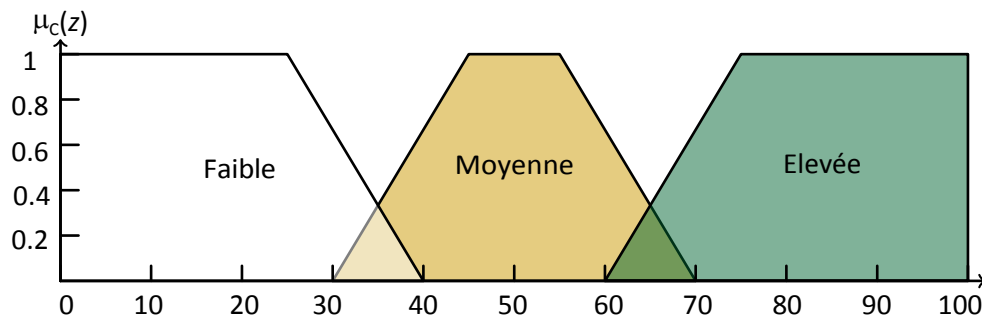


Figure II-10 Ensembles flous de la variable de sortie : vitesse du ventilateur (%) « z »

Supposons qu'il fait actuellement 18°C ($x = 18$), et que l'humidité est de 80% ($y = 80$). On applique ces entrées sur les fonctions d'appartenance pour déterminer l'appartenance à chaque variable. Une température de 18°C correspond à une appartenance de 0.5 à l'ensemble *faible* ($\mu_{A1} = 0.5$) et une appartenance de 0.33 à l'ensemble *moyenne* ($\mu_{A2} = 0.33$).

Une humidité de 80% correspond à une appartenance de 0.25 à l'ensemble *sèche* ($\mu_{B1} = 0.25$) et une appartenance de 0.75 à l'ensemble *humide* ($\mu_{B2} = 0.75$).

Évaluation des règles

Il faut maintenant évaluer les règles en fonction des entrées obtenues. On applique les opérateurs flous correspondants pour combiner les règles. Si une règle possède plusieurs antécédents, un opérateur flou est utilisé pour obtenir un seul chiffre qui représente le résultat. Ce dernier est ensuite appliqué à la fonction d'appartenance de la conséquence. On utilisera dans cet exemple la méthode de coupure pour produire le résultat.

On applique les valeurs d'appartenance à la règle 1 :

SI x est A_1 (0.5) OU y est B_1 (0.25) ALORS z est C_1 (?)

On peut utiliser deux méthodes pour combiner ces antécédents :

1. **Méthode 1** : maximum

$$\mu_{C1} = \max[\mu_{A1}, \mu_{B1}] = 0.5$$

2. **Méthode 2** : probor

$$\begin{aligned} \mu_{C1} &= \text{probor}[\mu_{A1}, \mu_{B1}] = (\mu_{A1} + \mu_{B1}) - (\mu_{A1} \cdot \mu_{B1}) \\ &= (0.5 + 0.25) - (0.5 * 0.25) = 0.625 \end{aligned}$$

On utilise la méthode 1, l'appartenance est 0.5

Pour la règle 2 :

SI x est A_2 (0.33) ET y est B_2 (0.75) ALORS z est C_2 (?)

On peut utiliser deux méthodes pour combiner ces antécédents :

1. **Méthode 1** : minimum

$$\mu_{C2} = \min[\mu_{A2}, \mu_{B2}] = 0.33$$

2. **Méthode 2** : prod

$$\mu_{C2} = \text{prod}[\mu_{A2}, \mu_{B2}] = \mu_{A2} * \mu_{B2} = (0.33)(0.75) = 0.25$$

On utilise la méthode 1, l'appartenance est 0.33

Finalement on obtient :

SI x est A_1 (0.5) OU y est B_1 (0.25) ALORS z est C_1 (0.5)

SI x est A_2 (0.33) ET y est B_2 (0.75) ALORS z est C_2 (0.33)

SI x est A_3 (0.0) ALORS z est C_3 (0.0)

Inférence

Si on utilise la méthode de coupure pour combiner les règles, il faut générer un nouvel ensemble flou à partir des trois fonctions d'appartenance de la conséquence C. La hauteur de cet ensemble flou est déterminée à partir des différentes valeurs d'appartenance calculées auparavant. Le nouvel ensemble flou est illustré dans la figure (II-12).

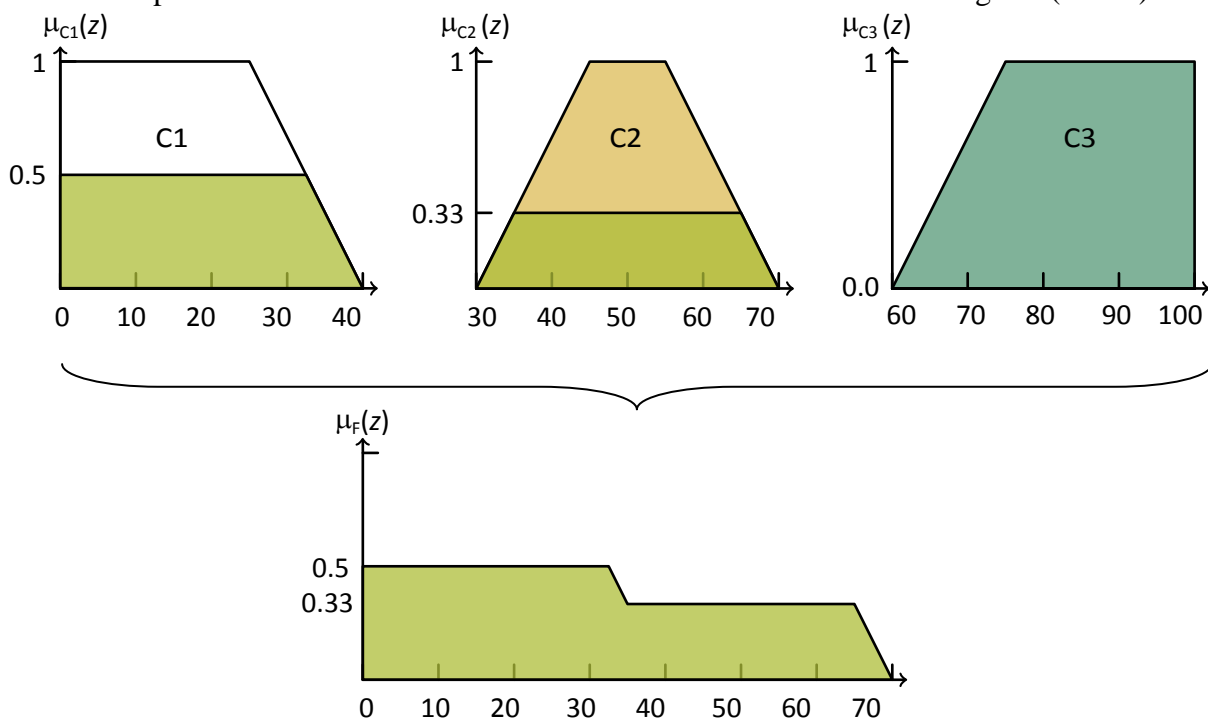


Figure II-11 Ensemble flou de sortie par la méthode de coupure

Défuzzification

Après avoir combiné les règles, il faut maintenant produire un chiffre net comme sortie. Dans ce cas ci, la sortie doit être la vitesse du ventilateur. La technique la plus populaire est la méthode du centre de gravité :

$$y = CG = \frac{\sum_{z=a}^b z * \mu_F(z)}{\sum_{z=a}^b \mu_F(z)}$$

On prend $a = 0$, $b = 70$ et le pas de progression de a vers b est pris égal à 10

$$y = CG = \frac{(0 + 10 + 20 + 30) * 0.5 + (40 + 50 + 60) * 0.33 + 70 * 0}{0.5 + 0.5 + 0.5 + 0.5 + 0.33 + 0.33 + 0.33 + 0.0} \approx 26.59$$

Le ventilateur doit donc être à 26.59% de sa vitesse maximale.

Chapitre 3

Les algorithmes génétiques

1- Introduction

Les algorithmes génétiques (AG) appartiennent à la classe des techniques d'optimisation utilisés surtout dans le cas des applications non linéaires. Cette méthode repose sur le principe de l'évolution naturelle énoncé par Charles Darwin qui stipule que les individus les mieux adaptés à survivre se reproduisent plus que les autres et auront plus de descendants.

Grace à ce principe de sélection, la qualité de l'ensemble génétique de la population va être accrue, les gènes les plus efficaces seront plus nombreux et ainsi la qualité de la population va s'améliorer. Le déroulement d'un algorithme génétique débutera par une population de base formée de solutions initiales, qui se reproduiront tout en s'assurant que la reproduction affectera surtout les meilleures solutions, et ainsi une nouvelle population de solutions sera générée. Ce processus se répétera plusieurs fois en obtenant à chaque fois une population actualisée formée des meilleures solutions.

2- Historique

C'est au physicien John Holland à qui l'on doit les premières recherches effectuées en 1975 sur les algorithmes génétiques. Ce physicien a développé les fondements principaux des algorithmes génétiques dans le cadre de l'optimisation mathématique. Malheureusement, avec les limites des outils informatiques disponibles à cette époque, les travaux de Holland n'ont pas été capables de résoudre des problèmes de taille complexe. Ce n'est que plus tard que les AG ont pu être développés par certains chercheurs tel que : Goldberg, Jong et Michalewicz.

En 1989, est apparu un livre décrivant l'utilisation des AG pour la résolution de problèmes réels. Ce livre, dont l'auteur est D.E. Goldberg, représente un ouvrage de référence dans ce domaine et a permis de valoriser l'intérêt de ces algorithmes au sein de la communauté scientifique surtout grâce au progrès notables de l'informatique.

De nos jours, les algorithmes génétiques demeurent un outil puissant et largement répandu pour la résolution de divers problèmes d'optimisation et dans de nombreux domaines d'applications.

3- Présentation des algorithmes génétiques

Un algorithme génétique peut être vu comme une technique de programmation partant du principe de l'évolution et visant à trouver la solution optimale pour un problème donné. Les notions et le vocabulaire utilisé dans les AG correspondent à ceux de la théorie de l'évolution et

de la génétique, en utilisant des termes tels que : population d'individus, gènes, chromosomes, parents, descendants, reproduction, croisement, mutation, ...etc.

Voici quelques termes de base de l'algorithme génétique :

- **Population** : ensemble fini d'individus (de solutions).
- **Individu** : solution potentielle du problème ou expression des chromosomes.
- **Chromosome** : solution potentielle du problème sous une forme codée (forme de chaîne de caractères) ou ensemble des gènes.
- **Gène** : partie élémentaire (caractère) non divisible d'un chromosome.
- **Fitness** : terme qui désigne la fonction d'évaluation d'un individu. Cette Fonction est liée à la fonction à optimiser et permet de définir le degré de performance d'un individu (son adaptation au problème posé donc d'une solution).

Une population est un ensemble d'individus, chaque individu est composé de plusieurs chromosomes, chaque chromosome est un ensemble de gènes et enfin chaque gène est un ensemble de valeurs généralement données sous forme binaire (des bits).

On aura ainsi une structure à niveaux comme illustré à la figure (III-1)

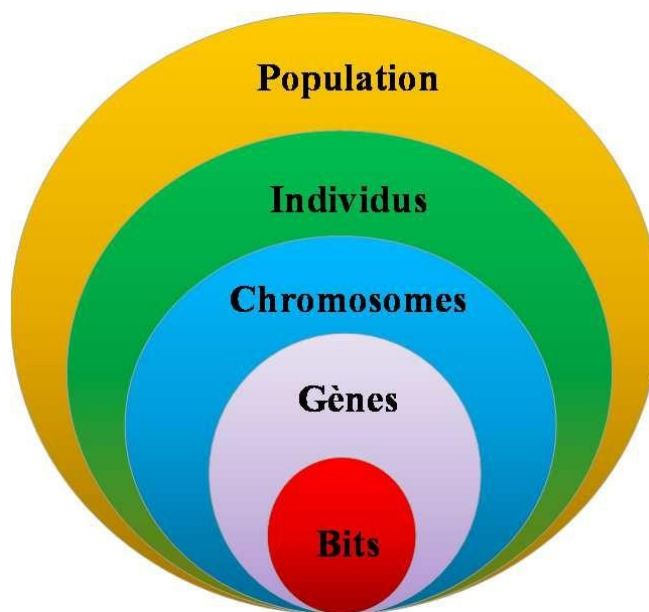


Figure III-1 Différents niveaux dans un algorithme génétique

4- Structure d'un algorithme génétique

L'implémentation d'un AG est spécifique au problème à résoudre. Pour l'utiliser, il faut disposer des cinq éléments suivants:

- ✓ Le codage de l'individu : La qualité du codage des données conditionne le succès des algorithmes génétiques. Les codages binaires ont été très utilisés à l'origine. Les codages réels sont désormais largement utilisés, notamment dans les domaines applicatifs pour l'optimisation de problèmes à variables réelles.
- ✓ La génération de la population initiale : Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures.
- ✓ La fonction à optimiser : Celle-ci retourne une valeur appelée fitness ou fonction d'évaluation de l'individu.
- ✓ Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population ainsi que l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'états.
- ✓ Des paramètres de dimensionnement: la taille de la population, nombre total de générations, probabilités d'application des opérateurs de croisement et de mutation.

L'organigramme fonctionnel de la figure (III-2) illustre la structure de l'algorithme génétique.

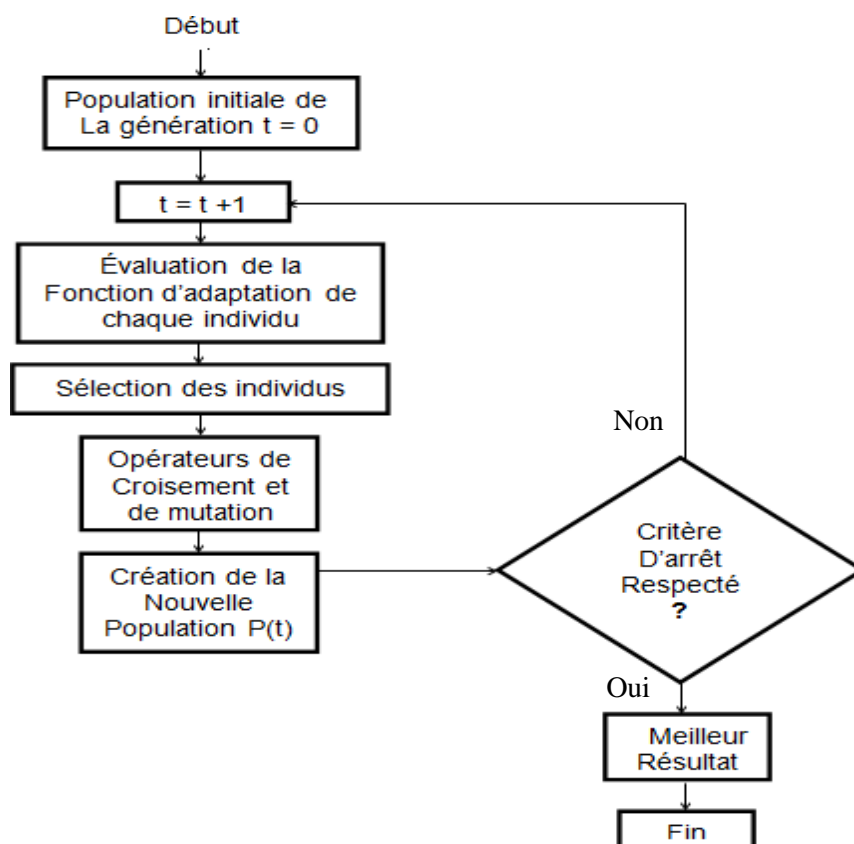


Figure III-2 Organigramme d'un AG standard

5- Les éléments d'un algorithme génétique

5-1- Individu

On désigne par « individus » les solutions au problème à résoudre. Avant que l'AG puisse être appliqué, il y a nécessité de coder ces solutions. On appelle une solution codée un « chromosome » composé à son tour par des « gènes » dont chacune d'elles représente une variable du problème, une éventuelle solution ou bien une notion abstraite.

Généralement, on utilise un codage en vecteurs tel que chaque vecteur correspond à une certaine solution. Le type du vecteur peut prendre la forme binaire (ne possédant que deux valeurs 0 et 1), ou bien un autre type discret quantifiable tel que des entiers, des caractères, ...etc. Le type continu (nombres réels) peut être aussi utilisé, mais il faudra adapter les opérations qui permettent de modifier la nature des chromosomes en tenant compte de la fonction qui génère les chromosomes ainsi que les opérateurs génétiques.

Afin de s'assurer de la simplicité de la mise en œuvre de l'AG, les chromosomes doivent être uniformes (tous les gènes soient de même type). Dans certains cas on peut utiliser des chromosomes de types différents en tenant compte des opérations modifiant la nature des chromosomes. Habituellement, il est recommandé de bien choisir les chromosomes de sorte qu'ils aient la même longueur et les gènes similaires doivent être aux mêmes endroits sur leurs chromosomes respectifs.

D'autres formes de codage peuvent être utilisés mais ils ne sont pas très fréquents, mais on leur préfère le codage par vecteurs vu sa grande simplicité.

5-2- Population

La population est un ensemble d'individus (ou bien de chromosomes) de la même génération. La taille de la population est fixe et ne change pas durant tous le processus de l'AG.

Généralement, au départ d'un algorithme génétique, il faut créer une population d'individus. Ces individus sont générés par une fonction simple. Cette fonction affecte à chaque individu qu'elle crée une valeur aléatoire pour chacun de ses gènes. L'algorithme génétique peut également utiliser comme population de départ une population déjà créée à priori.

5-3- Fonction «Fitness »

Le calcul de la qualité d'un individu est essentiel aux algorithmes génétiques. Cette fonction donne, en valeur numérique (habituellement réelle), la qualité d'un individu. C'est

selon cette valeur numérique que sont calculées les chances de sélection de cet individu. La fonction de fitness doit avoir 0 comme plancher, pour ne pas fausser le calcul des pourcentages.

Les algorithmes génétiques étant une technique d'optimisation, ils cherchent la qualité maximale, donc l'optimisation de la fonction de qualité. Si on cherche plutôt à minimiser une fonction, il faudra la modifier de sorte que la fonction de qualité se maximise. Il serait bien entendu possible de conserver une fonction de qualité qui fonctionne à l'envers et de modifier à la place le calcul des probabilités, mais ceci rendrait l'algorithme beaucoup plus difficile à décoder pour les utilisateurs externes.

5-4- Sélection

Selon la qualité des individus, chacun se voit attribuer un pourcentage de chances d'être choisi pour la reproduction, qui correspond à l'importance relative de la qualité de l'individu par rapport à la qualité totale de la population.

5-5- Reproduction

La reproduction est effectuée en faisant le croisement de deux individus (les parents) qui à leur tour produiront deux autres nouveaux individus (les enfants) qui seront placés dans la nouvelle population. La reproduction s'effectue en appliquant des opérateurs génétiques sur les deux chromosomes sélectionnés.. Les opérateurs classiques de l'AG sont le croisement et la mutation. Environ la moitié des gènes de chaque parent est transmise au chromosome enfant.

On prend généralement 50 % de probabilité pour l'opérateur de croisement, ce qui permet d'avoir un mélange plus efficace entre les différents chromosomes. En prenant cette valeur de probabilité, il y aura un échange au niveau des gènes semblables des chromosomes. Grace à ce choix on obtiendra deux enfants qui seront complémentaires par rapport à leurs parents et si un enfant reçoit un certain gène d'un parent, l'autre enfant aura exactement le même gène de l'autre parent. En ce qui concerne l'opérateur de mutation, on prend généralement une probabilité variant entre 0.5 % et 5 %. L'opérateur de mutation joue un rôle important dans la procédure d'un AG puisque sans lui l'algorithme risque de tomber sur un minima local (solution non optimale)

Puisque les chromosomes enfants sont issus de leurs parents ils sont donc limités aux caractéristiques de leurs parents. Si un gène par exemple ne se trouve pas dans la population

initiale (ou s'il disparaît au cours des générations) il ne pourrait pas se développer chez les descendants. L'opérateur de mutation sert justement à éviter ce problème puisque chaque gène

dispose d'une petite probabilité de muter (de changer de valeur) et ainsi d'être substitué par une autre transformation de ce gène. L'utilisation de l'opérateur de mutation permet de veiller à la continuité de la diversité génétique. Cependant, en général la mutation génère des individus faibles peu capables de survivre.

Toutefois, une mutation adéquate permet d'accroître l'évolution de la population. Les opérateurs génétiques peuvent généralement prendre en compte les différentes modifications survenues dans l'AG. Cependant, suivant la tâche à accomplir, on doit modifier ou changer totalement les opérateurs génétiques. D'une façon générale, un opérateur génétique est une fonction quelconque qui permet la modification génétique d'un ou plusieurs chromosomes.

6- Le codage dans un algorithme génétique

Le codage indique la représentation des différents états des variables tels qu'elles puissent être utilisables dans l'AG. Le codage établit une correspondance entre les valeurs des variables et les individus de la population de sorte à reproduire l'interprétation existante entre les deux sortes de variables (réelles et individus). On distingue divers types de codage tel que : **binaire, réel, codage de gray et codage dynamique.**

Chaque codage est utilisé pour une tâche spécifique mais le plus usuel est le codage binaire.

6-1- Le codage binaire

Le codage binaire est représenté sous forme d'une chaîne de bits qui contient toute l'information nécessaire pour représenter un point de l'espace de recherche.

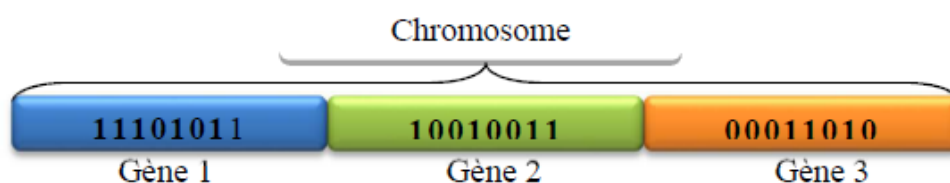


Figure III-3 Chromosome avec codage binaire

Pour chaque paramètre x_i situé dans l'intervalle $[x_{imin}, x_{imax}]$, on associe une chaîne binaire $b_0 b_1 \dots b_{l_{xi}-1}$, définie sur l_{xi} bits. A cette chaîne correspond un entier naturel y_i [] :

$$y_i = \sum_{j=0}^{l_{xi}-1} b_j 2^{l_{xi}-j-1}$$

Cette chaîne doit être décodée pour pouvoir calculer la valeur de la fonction d'évaluation qui lui est associée. Le paramètre réel x_i de l'espace de recherche relatif à y_i est obtenu par l'interpolation linéaire suivante :

$$x_i = x_{imin} + \frac{x_{imax} - x_{imin}}{2^{lxi} - 1} y_i \quad i$$

6-2- Le codage réel

Le codage réel est plus simple que le codage binaire, chaque chromosome est en fait un vecteur dont les composantes sont les paramètres du processus d'optimisation. Par exemple, si on recherche l'optimum d'une fonction de n variables $f(x_1, x_2, \dots, x_{n-1}, x_n)$, on peut utiliser tout simplement un chromosome contenant les n variables.

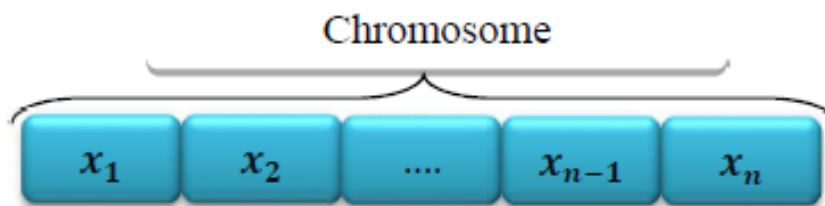


Figure III-4 Chromosome avec codage réel

Avec ce type de codage, la procédure d'évaluation des chromosomes est plus rapide vu l'absence de l'étape de transcodage (du binaire vers le réel).

7- Génération de la population initiale

Le choix de la population initiale conditionne fortement la rapidité de convergence de l'algorithme. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement les individus de la population en faisant des tirages au hasard selon une distribution uniforme. Si des informations à priori sur le problème sont disponibles, il est naturel de générer les individus dans un sous domaine particulier afin d'accélérer la convergence.

La taille de la population doit être choisie de façon à réaliser un bon compromis entre temps de calcul et qualité du résultat. En effet, une taille de population trop grande augmente le temps de calcul et nécessite un espace mémoire considérable, alors qu'une taille de population trop petite conduit à l'obtention d'un optimum local.

8- Fonction d'adaptation « Fitness »

En raison de l'analogie avec la théorie de l'évolution (survie des individus les mieux adaptés à leur environnement), l'Algorithme Génétique est formulé en terme de

maximisation. Etant donné une fonction f réelle à une ou plusieurs variables, le problème d'optimisation sur l'espace de recherche S s'écrit de la manière suivante :

$$\max_{x \in S} (f(x))$$

Dans beaucoup de problèmes d'optimisation, l'objectif est exprimé sous forme de minimisation d'une fonction coût g :

$$\min_{x \in S} (g(x))$$

Le passage du problème de minimisation à un problème de maximisation est obtenu par transformation de la fonction g selon la relation suivante :

$$f(x) = \frac{1}{1 + g(x)}$$

9- Reproduction (sélection des parents)

La sélection est un mécanisme qui fixe à partir de la génération Précédente, quels individus pourront se reproduire pour former la génération suivante. Lors de cette phase, les individus les plus forts sont généralement dupliqués et forment les parents de la génération en cours.

On trouve essentiellement quatre types de méthodes de sélection :

- *Sélection par la roue de loterie biaisée.*
- *Sélection élitiste.*
- *Sélection par tournois.*
- *Sélection par rang.*

9-1- Sélection Par La Roue De Loterie Biaisée (Roulette Wheel) De Goldberg

Avec cette méthode chaque individu a une chance d'être sélectionné proportionnellement à sa performance 'fitness'. Plus la performance d'un individu est élevée par rapport à celle des autres, plus il a une chance d'être reproduit dans la population. Ainsi, dans le cas d'un codage binaire, la fonction d'évaluation d'un chromosome particulier chr_i étant $f(chr_i)$, la probabilité avec laquelle il sera réintroduit dans la nouvelle population de taille N est :

$$P_s = \frac{f(chr_i)}{\sum_{j=1}^N f(chr_j)}$$

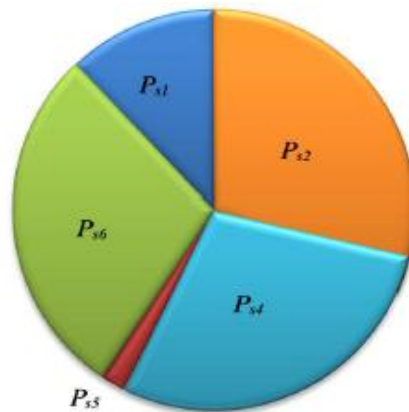


Figure III-5 Sélection par roulette

9-2- Sélection élitiste

La stratégie élitiste consiste à conserver le meilleur individu à chaque génération. Ainsi l'élitisme empêche l'individu le plus performant de disparaître au cours de la sélection ou que ses bonnes combinaisons soient affectées par les opérateurs de croisement et de mutation. Après chaque évaluation de la performance des individus à une génération k donnée, le meilleur individu de la génération précédente $k - 1$ est réintroduit dans la population si aucun des individus de la génération k n'est meilleur que lui. Par cette approche, la performance du meilleur individu de la population courante est monotone de génération en génération

9-3- Sélection par tournoi

On effectue un tirage avec remise de deux individus de la population courante, et on le fait "combattre", celui qui a la fitness la plus élevée l'emporte. On répète ce processus N fois de manière à obtenir les N individus de la nouvelle population qui serviront de parents.

9-4- Sélection par rang

La sélection par rang trie d'abord la population par fitness. Chaque individu se voit associé un rang en fonction de sa position. La valeur du rang est calculée selon le total des adaptations des individus divisé par la taille de la population.

$$rang = \frac{\sum_{i=1}^N f(i)}{N}$$

Le plus mauvais individu aura le premier rang, le suivant le deuxième rang, ainsi de suite. La sélection par rang d'un individu est la même que par roulette, mais les proportions sont en relation avec le rang plutôt qu'avec la valeur de l'évaluation.

10- Opérateurs de recombinaison génétique (croisement et mutation)

Les opérateurs de recombinaison génétique garantissent la diversification des individus dans la population, en assurant le brassage des matériels génétiques au cours des générations.

Les principaux opérateurs de recombinaison génétique sont le croisement et la mutation.

10-1- Croisement

La naissance d'un nouvel individu, nécessite la prise aléatoire d'une partie des gènes de chacun des deux parents. Ce phénomène, issu de la nature est appelé croisement (crossover).

L'application de l'opérateur de croisement se fait avec une probabilité notée P_c sur un couple de parents choisis aléatoirement afin d'engendrer deux enfants. La valeur de P_c est prise en général entre les deux valeurs 0.5 et 0.9, et plus cette valeur augmente plus il y aura des changements importants au sein de la population.

La littérature définit plusieurs opérateurs de croisement, ils diffèrent selon le type de codage adapté.

On définit le croisement binaire comme étant un échange entre certains bits qui se fait avec deux chromosomes. On distingue divers types de croisement :

a- Croisement en un point

C'est le croisement le plus simple et le plus connu. Il consiste à choisir au hasard un point de croisement pour chaque couple de chromosomes parents (chr_1 et chr_2) constitués de l gènes, ensuite on échange les deux sous chaînes terminales de chacun des chromosomes, ce qui produit deux enfants chr'_1 et chr'_2 comme indiqué sur la figure (III-6)

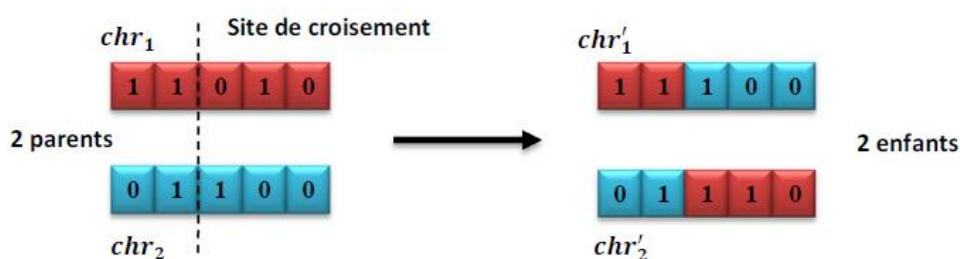


Figure III-6 Croisement en un point

b- Croisement en deux points

Dans ce type de croisement, on prend arbitrairement deux points de coupure puis on inter-échange les valeurs des bits correspondant à ces points pour générer deux enfants.

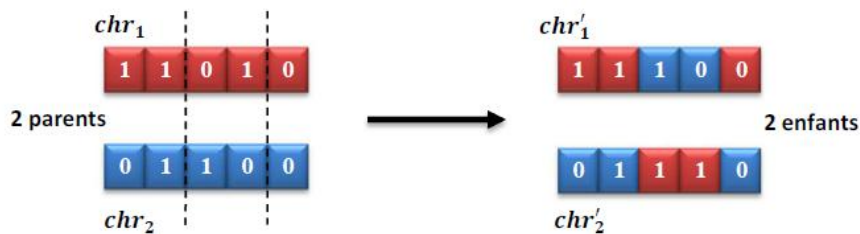


Figure III-7 Croisement en deux points

c- Croisement uniforme

Pour chaque couple d'individus, on leur génère d'une façon aléatoire un masque de croisement dont les valeurs sont binaires et sa taille est la même que celle du chromosome. Suivant la valeur du bit du masque il y aura transmission de la valeur du gène de l'un des parents vers un enfant. En effet, si cette valeur vaut « 1 » alors on aura la valeur du gène qui se transmet du *parent1* vers l'*enfant1* et aussi du *parent2* vers l'*enfant2*. Si au contraire cette valeur vaut « 0 » on aura la valeur du gène qui se transmet du *parent2* vers l'*enfant1* et du *parent1* vers l'*enfant2*.

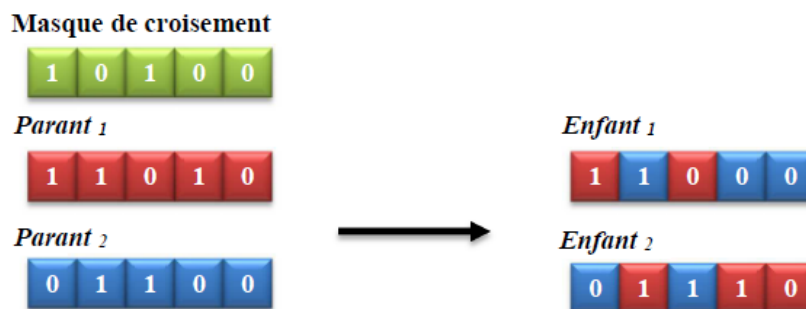


Figure III-8 Croisement uniforme

d- Croisement barycentre

Le croisement barycentre consiste en la sélection de deux gènes $chr_1(i)$ et $chr_2(i)$ à partir des parents à la même $i^{ème}$ position. On définit ainsi deux nouveaux gènes $chr'_1(i)$ et $chr'_2(i)$ obtenus par combinaison linéaire.

$$chr'_1(i) = r * chr_1(i) + (1 - r) * chr_2(i)$$

$$chr'_2(i) = (1 - r) * chr_1(i) + r * chr_2(i)$$

Tel que le coefficient r représente un facteur de pondération dont la valeur de trouve dans l'intervalle $[-0.5, 1.5]$

10-2- Mutation

La mutation est une modification qui intervient de manière aléatoire sur un chromosome, les individus de la population issue du croisement vont subir un processus de mutation avec une probabilité p_m . La valeur de cette probabilité est faible, généralement fixée entre 0,001 et 0,01.

L'opérateur de mutation permet de maintenir une certaine diversité dans la population. Il existe plusieurs méthodes de mutation parmi lesquels on trouve :

10-2-1 Mutation binaire

Une mutation en codage binaire consiste simplement en l'inversion d'un bit se trouvant en un locus bien particulier, bit est donc remplacé par son complément. La position du bit qui doit subir la mutation est déterminée aléatoirement

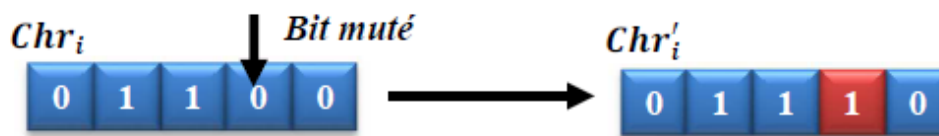


Figure III-9 Mutation binaire

10-2-2 Mutation réelle

La différence entre les deux types de mutation réelle et binaire réside dans la nature de l'élément sur lequel il y a changement. Dans la mutation réelle, on n'inverse pas un bit mais on prend une variable réelle aléatoirement sur son intervalle de définition. Pour un codage réel, deux opérateurs de mutation sont généralement utilisés : la mutation uniforme et la mutation non uniforme.

a- Mutation uniforme

La mutation uniforme est une extension de la mutation binaire en fixant la valeur de la probabilité de mutation P_m . Pour chaque gène $chr(i)$ d'un chromosome, un tirage au sort est

effectué qui décidera de la modification ou non de ce gène. Le gène choisi est alors remplacé par une valeur $chr'(i)$ sélectionnée arbitrairement dans l'intervalle $[chr(i)_{\min}, chr(i)_{\max}]$.

b- Mutation non uniforme

Pendant les premières générations, des changements majeurs surviennent au niveau du gène $chr(i)$. Mais ces modifications tendent à décroître en progressant dans la procédure d'optimisation. Lors de la $k^{\text{ème}}$ génération, un tirage au sort est effectué pour avoir une valeur binaire laquelle va décider si la modification sera positive ou négative.

La nouvelle valeur $chr'(i)$ du gène $chr(i)$ est donnée par :

$$chr'(i) = \begin{cases} \Delta(k, chr(i)_{\max} - chr(i)) & \text{si } r = 0 \\ \Delta(k, chr(i) - chr(i)_{\min}) & \text{si } r = 1 \end{cases}$$

Où $\Delta(k, y)$ représente une fonction définissant l'écart entre la valeur initiale et la nouvelle valeur à la $k^{\text{ème}}$ génération. r est un nombre binaire pouvant prendre les valeurs 0 ou 1.

11- Sélection des individus d'une nouvelle génération

Après avoir appliqué les opérateurs de croisement et de mutation, la population d'individus comportera $2N$ individus (N enfants et N parents). Mais puisque chaque génération ne doit avoir que N individus, il est indispensable de supprimer la moitié des $2N$ individus et ne conserver que les individus les mieux adaptés (les solutions les plus optimales). Diverses méthodes peuvent être utilisées pour choisir les meilleurs individus, à savoir :

11-1 Sélection par compétition

Afin de déterminer quels sont les meilleurs individus entre les enfants et leurs parents, une compétition survient entre eux pour choisir les meilleurs qui seront les individus de la nouvelle génération. Ce choix est déterminé suivant les performances de chaque individu enfants ou parents.

11-2 Sélection élitiste

Dans ce type de sélection, un « concours » est organisé entre les parents et leurs enfants en même temps afin de déterminer les meilleurs parmi les quatre.

11-3 Sélection par descendance

Les parents sont remplacés directement par leurs enfants et ceci quels que soient leurs performances. Le point faible de ce type de sélection est le risque de voir disparaître des caractéristiques génétiques intéressantes des parents dans la nouvelle population.

12- Domaines d'application des AG

Les applications des AG sont multiples :

- optimisation de fonctions numériques difficiles (discontinues...).
- traitement d'image (alignement de photos satellites, reconnaissance de suspects...).
- optimisation d'emplois du temps.
- optimisation de design.
- contrôle de systèmes industriels.
- apprentissage des réseaux de neurones
- Les AG peuvent être utilisées pour contrôler un système évoluant dans le temps (chaîne de production, centrale nucléaire...).
- Les AG sont également utilisées pour optimiser des réseaux (câbles, fibres optiques, mais aussi eau, gaz...), des antennes
- TSP (voyageur de commerce)
- Bin Packing - Remplissage de boîtes rectangulaires.
- Construction de réseaux de communication.
- Contrôle de pipe-lines et d'autres systèmes complexes.

Bibliographie

- [1] Dreyfus G., Martinez J.M., Samuelides M., Gordon M._B., Badran F., Thiria S., Herault L., « Réseaux de neurones: méthodologies et applications », édition EYROLLES, 2004.
- [2] Wiem M. C., « Apprentissage autonome de réseaux de neurones pour le pilotage en temps réel des systèmes de production basé sur l'optimisation via simulation, PhD Thesis, Université Blaise Pascal - Clermont-Ferrand II, 2009.
- [3] Haque R., "Transmission Loss Allocation Using Artificial Neural Networks", PhD Thesis, University of Saskatchewan Saskatoon, Saskatchewan, 2006
- [4] Rivals I., « Modélisation et commande de processus par réseaux de neurones: application au pilotage d'un véhicule autonome », thèse de doctorat de l'université Pierre et Marie Curie, Paris VI, France, 1995.
- [5] Dubreil H., « Méthodes d'optimisation de contrôleurs de logique floue pour le paramétrage automatique des réseaux mobiles UMTS », thèse de doctorat de l'école nationale supérieure des télécommunications, 2005.
- [6] Bustince H., Herrera F. Montero J., « Fuzzy Sets and Their Extensions: Representation Aggregation and Models », Germany, Berlin: Springer-Verlag, 2007.
- [7] Ta Cao M., « Commande Numérique de Machines Asynchrones par Logique Floue », thèse de doctorat, Université Laval Québec 1997.
- [8] Robyns B., Francois B., Degobert P., Hautier J-P., « Commande vectorielle de la machine asynchrone: Désensibilisation et optimisation par la logique floue », Paris: Editions TECHNIP, 2007.
- [9] Baghli L., « Contribution a 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 Faculté des sciences France 14 janvier, 1999.*
- [10] Golberg D.E., « Genetic algorithms in search, optimization, and machine learning », Boston: Addison Wesley; 1989.
- [11] Durand N., « Algorithmes Génétiques et autres méthodes d'optimisation appliqués à la gestion de trafic aérien », Ph.D. Dissertation, institut national polytechnique de Toulouse, 2004
- [12] Djouada M., Ziani R., Felkaoui A., Zegadi R., « Diagnostic des défauts par un Couplage réseaux de neurones artificiels - algorithmes génétiques », 4th International Conference on Computer Integrated Manufacturing CIP'2007.

Examen et corrigé 2021/2022

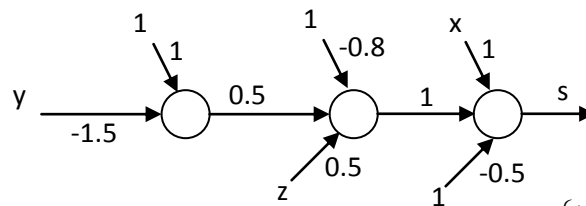
Examen du module : « Techniques d'intelligence artificielle »

Questions de cours (06.00 pts):

1. Citer deux avantages et deux inconvénients des réseaux de neurones artificiels ?
2. Citer deux applications des réseaux de neurones artificiels?
3. Donner le schéma général d'un système flou ?

Exercice 1 (06.00 pts):

Soit le réseau neuronal suivant :

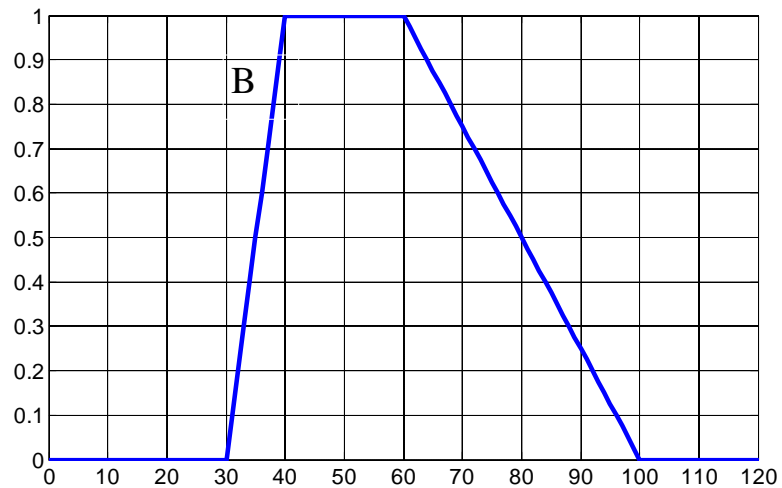
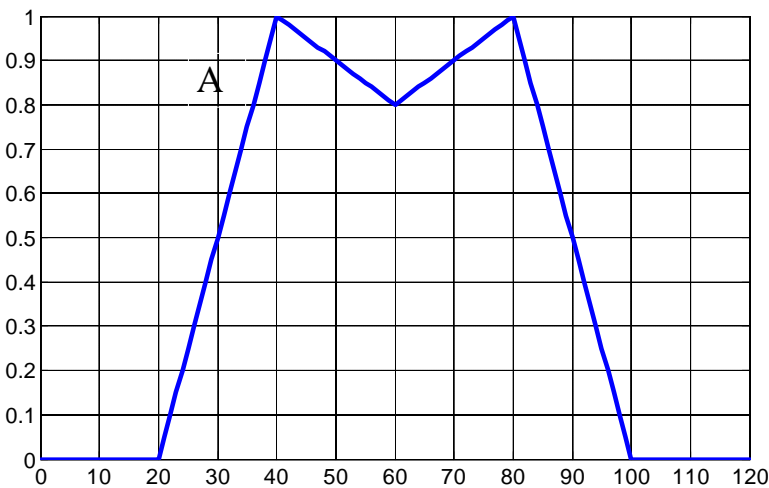


Sachant que les 03 neurones possèdent la même fonction d'activation $f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$

Trouver la fonction booléenne la plus simplifiée réalisée par ce réseau ?

Exercice 2 (08.00 pts):

Soient A et B deux ensembles flous décrits, respectivement, par leurs fonctions d'appartenances : $\mu_A(x)$ et $\mu_B(x)$ et représentés sur la figure suivante :



1- Déterminer graphiquement les ensembles flous suivants : \bar{A} , \bar{B} , $A \cap B$, $A \cup B$, $\bar{A} \cup B$, $A \cap \bar{B}$

2- Calculer la hauteur, le support, le noyau et 0.4-coupe pour chaque ensemble flou A et B ?

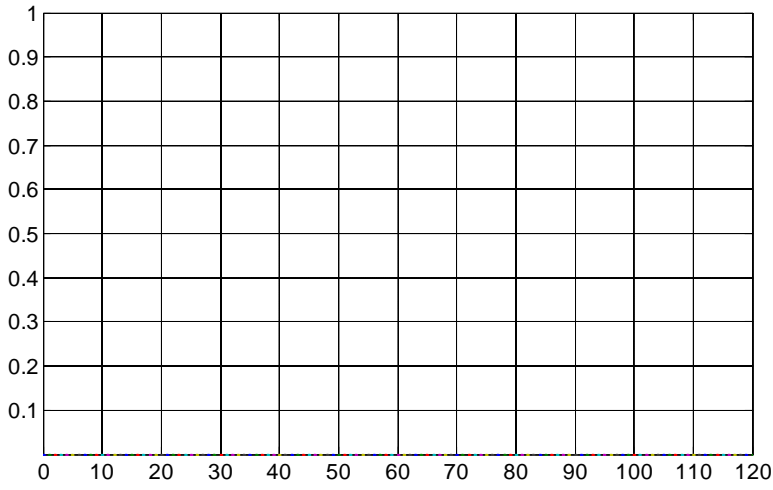
(prendre 04 chiffres après la virgule).

3- Soit l'ensemble flou **F** défini par : $\mu_F(x) = \frac{1}{1+0.5(x-2)^2}$, déterminer le α -coupe de F en fonction de α ?

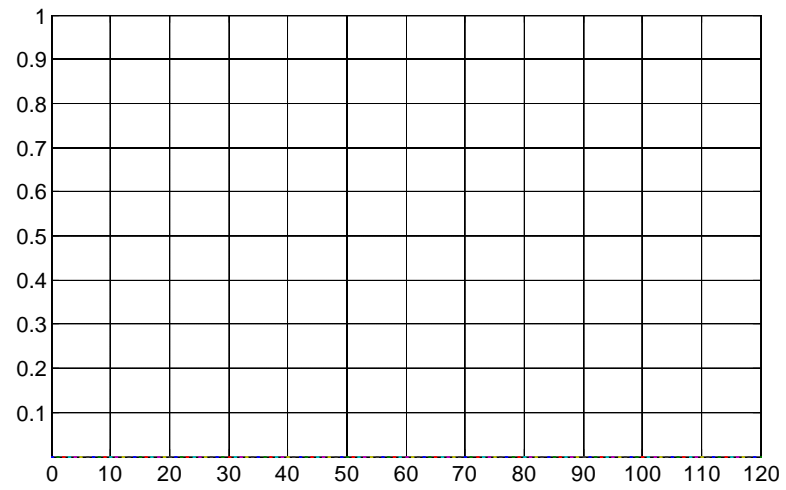
Nom :

Prénom :

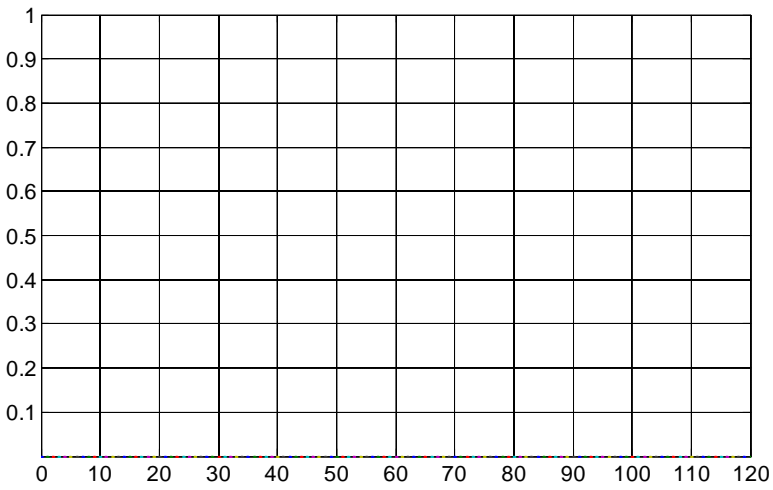
Groupe :



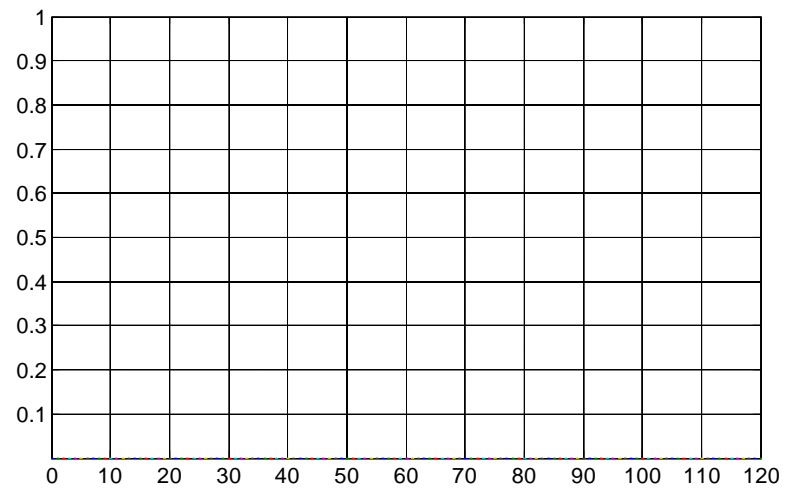
L'ensemble flou \bar{A}



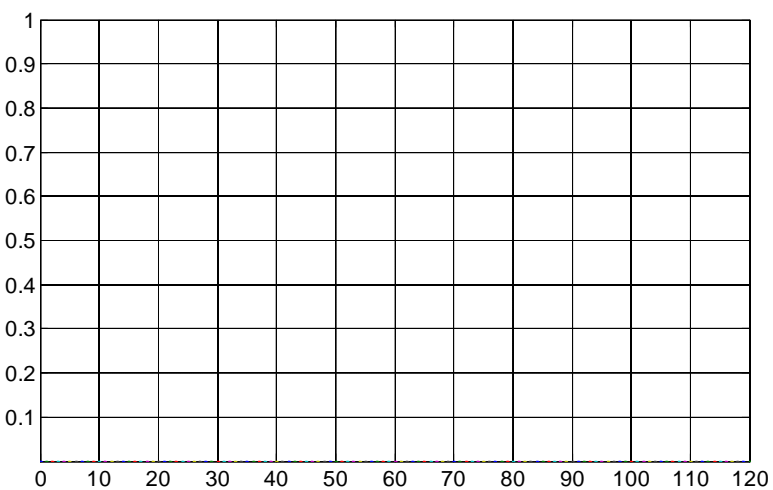
L'ensemble flou \bar{B}



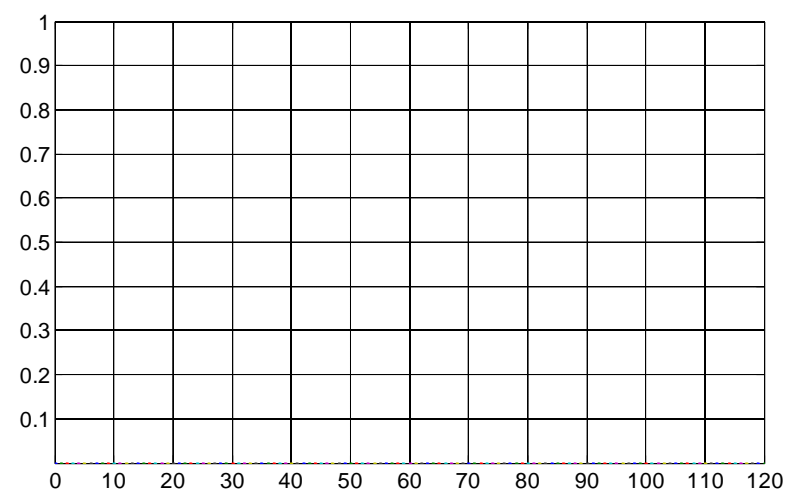
L'ensemble flou $A \cap B$



L'ensemble flou $A \cup B$



L'ensemble flou $\bar{A} \cup B$



L'ensemble flou $A \cap \bar{B}$

Corrigé examen :« Techniques d'intelligence artificielle»

Questions de cours (06.00 pts):

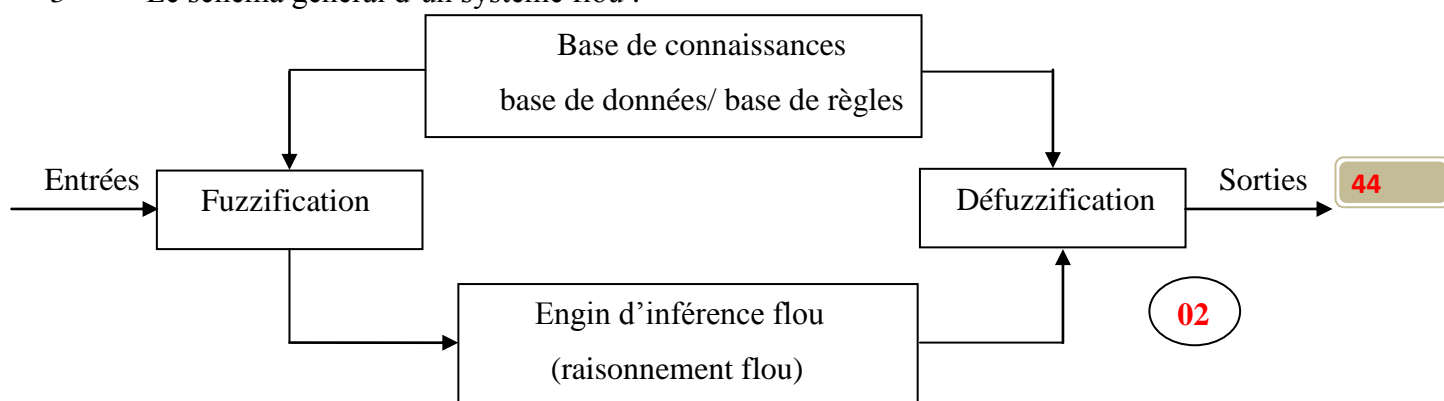
1- Il suffit de citer deux avantages et deux inconvénients parmi les suivants :

Avantages	inconvénients
<ul style="list-style-type: none">○ Les réseaux de neurones sont capables d'approximer des fonctions complexes non linéaires.○ Les réseaux de neurones sont flexibles vis-à-vis des données incomplètes et/ou bruitées.○ Les réseaux de neurones peuvent être adaptables avec de nouvelles données.○ Les réseaux de neurones surmontent certaines limitations de quelques méthodes statistiques en les généralisant.○ Les réseaux de neurones peuvent être implémentés sur un support matériel.○ Les performances des réseaux neuronaux peuvent être automatisées.○ Les réseaux de neurones sont adaptés pour résoudre différents problèmes dans beaucoup de domaines. <p style="text-align: right;">01</p>	<ul style="list-style-type: none">○ Les réseaux de neurones fonctionnent comme des «boîtes noires».○ Le choix de la topologie du réseau ainsi que les paramètres de l'algorithme d'entraînement est heuristique○ L'estimation des poids du réseau nécessite de grandes quantités de données.○ Il n'y a pas de règles prédéfinies pour sélectionner l'algorithme le plus approprié pour l'apprentissage.○ Les réseaux de neurones peuvent converger vers un minimum local.○ L'adaptation aux environnements différents est difficile et le réapprentissage est souvent obligatoire○ Aucune garantie sur la convergence des algorithmes d'apprentissage. <p style="text-align: right;">01</p>

2- Il suffit de citer deux applications parmi les suivantes :

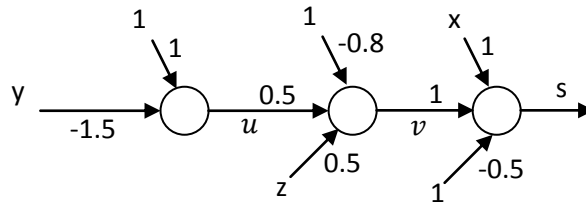
- Identification des systèmes non linéaires.
- Commande des robots.
- Reconnaissance d'images. **02**
- Reconnaissance de la parole.
- Reconnaissance de caractères.
- Prévisions financières.
- Diagnostics médicaux.

3- Le schéma général d'un système flou :



Exercice 1 (06.00 pts):

Soit le réseau neuronal suivant qui possède trois entrées « x », « y » et « z » avec une seule sortie « s » :



On note : « u » la sortie du 1^{er} neurone et « v » la sortie du 2^{ème} neurone

Tous les neurones possèdent la même fonction d'activation $f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$

Les équations du réseau sont données comme suit :

- Pour le 1^{er} neurone (de sortie u) :

$$S_u = y * (-1.5) + 1 * (1) = -1.5y + 1$$

$$u = f(S_u) = f(-1.5y + 1)$$

- Pour le 2^{ème} neurone (de sortie v) :

$$S_v = u * (0.5) + z * (0.5) + 1 * (-0.8) = 0.5u + 0.5z - 0.8$$

$$v = f(S_v) = f(0.5u + 0.5z - 0.8)$$

- Pour le 3^{ème} neurone (de sortie s) :

$$S_s = v * (1) + x * (1) + 1 * (-0.5) = v + x - 0.5$$

$$s = f(S_s) = f(v + x - 0.5)$$

A l'aide de ces équations on peut remplir la table de vérité ci-dessous :

x	y	z	S_u	u	S_v	v	S_s	s
0	0	0	1	1	-0.8	0	-0.5	0
0	0	1	1	1	0.2	1	0.5	1
0	1	0	-0.5	0	-0.8	0	-0.5	0
0	1	1	-0.5	0	-0.8	0	-0.5	0
1	0	0	1	1	-0.8	0	0.5	1
1	0	1	1	1	0.2	1	1.5	1
1	1	0	-0.5	0	-0.8	0	0.5	1
1	1	1	-0.5	0	-0.8	0	0.5	1

0.5 0.5 0.5 0.5 0.5 0.5

La sortie du réseau est active ($s = 1$) pour les valeurs suivantes :

$$(x, y, z) = \{(0,0,1); (1,0,0); (1,0,1); (1,1,0); (1,1,1)\}$$

La fonction logique réalisée par ce réseau est : $s = f(x, y, z) = \bar{x}\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + xy\bar{z} + xyz$ **1.5**

Pour simplifier cette fonction, nous allons utiliser le tableau de Karnaugh :

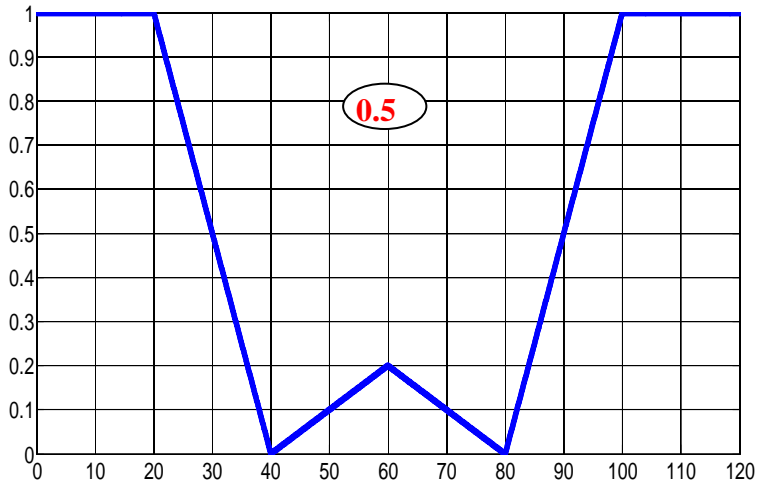
xy \ z	00	01	11	10
0			1 1	
1	1 1		1 1	

La relation simplifiée est donnée par : $s = x + \bar{y}z$

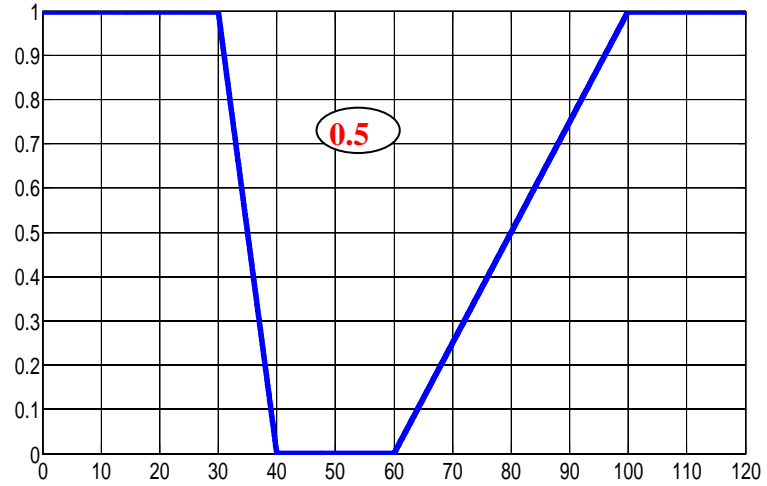
1.5

Exercice 2 (08.00 pts):

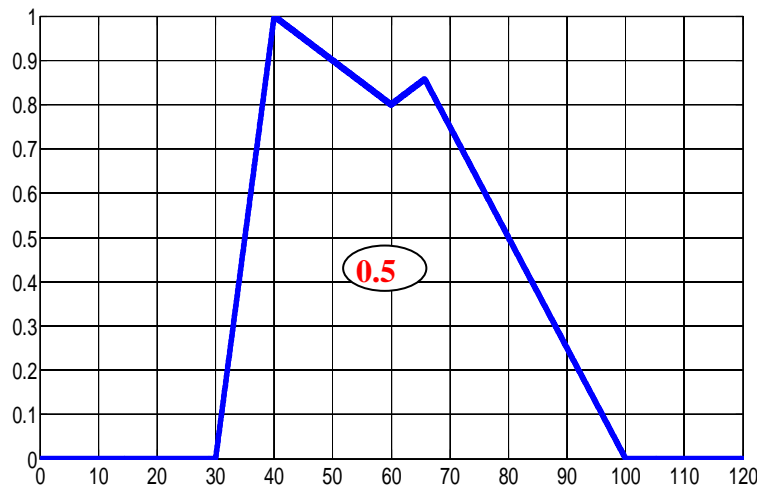
1- Déterminer graphiquement les ensembles flous suivants : \bar{A} , \bar{B} , $A \cap B$, $A \cup B$, $\bar{A} \cup B$, $A \cap \bar{B}$



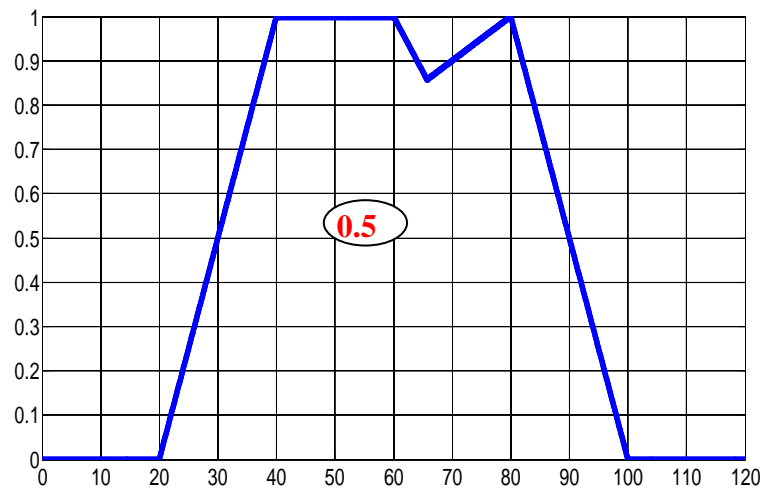
L'ensemble flou \bar{A}



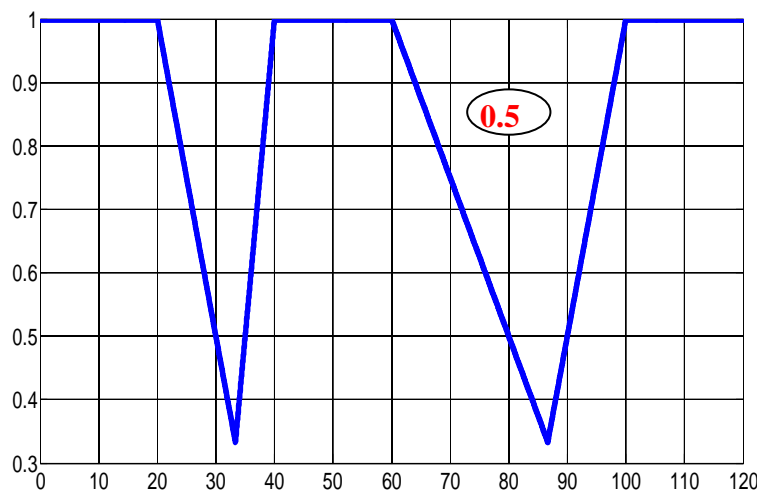
L'ensemble flou \bar{B}



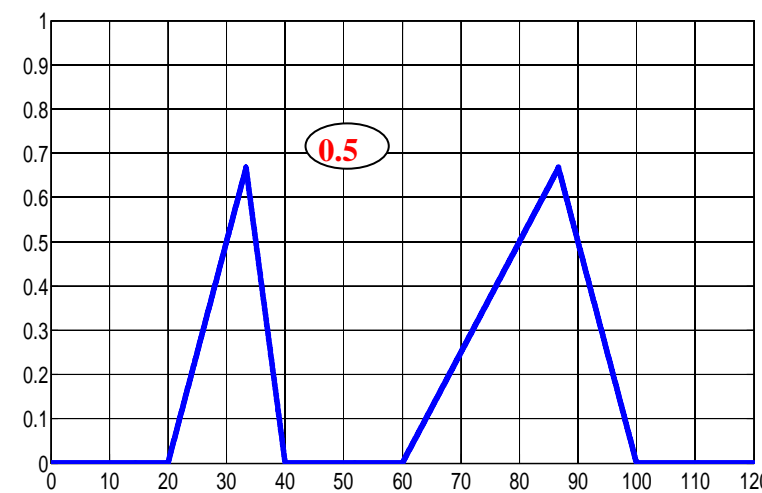
L'ensemble flou $A \cap B$



L'ensemble flou $A \cup B$



L'ensemble flou $\bar{A} \cup B$



L'ensemble flou $A \cap \bar{B}$

2- Calculer la hauteur, le support, le noyau et 0.4-coupe pour chaque ensemble flou A et B ?

$$h(A) = \sup\{\mu_A(x) / x \in U\} = 1 \quad (0.25)$$

$$h(B) = 1 \quad (0.25)$$

$$\text{supp}(A) = \{x \in U / \mu_A(x) > 0\} = [20, 100] \quad (0.25)$$

$$\text{supp}(B) = [30, 100] \quad (0.25)$$

$$\text{noy}(A) = \{x \in U / \mu_A(x) = 1\} = \{40, 80\} \quad (0.25)$$

$$\text{noy}(B) = [40, 60] \quad (0.25)$$

$$0.4\text{-coupe}(A) = \{x \in U / \mu_A(x) \geq 0.4\}$$

On définit l'équation de la droite pour l'ensemble A dans l'intervalle [20, 40] : $y = 0.05x - 1$

$$y = 0.4 \Rightarrow x = \frac{y+1}{0.05} = \frac{1.4}{0.05} = 28$$

On définit l'équation de la droite pour l'ensemble A dans l'intervalle [80, 100] : $y = -0.05x + 5$

$$y = 0.4 \Rightarrow x = \frac{5-y}{0.05} = \frac{4.6}{0.05} = 92 \Rightarrow 0.4\text{-coupe}(A) = [28, 92] \quad (0.5)$$

On définit l'équation de la droite pour l'ensemble B dans l'intervalle [30, 40] : $y = 0.1x - 3$

$$y = 0.4 \Rightarrow x = \frac{y+3}{0.1} = \frac{3.4}{0.1} = 34$$

On définit l'équation de la droite pour l'ensemble B dans l'intervalle [60, 100] : $y = -0.025x + 2.5$

$$y = 0.4 \Rightarrow x = \frac{2.5-y}{0.025} = \frac{2.1}{0.025} = 84 \Rightarrow 0.4\text{-coupe}(B) = [34, 84] \quad (0.5)$$

3- Soit l'ensemble flou F défini par : $\mu_F(x) = \frac{1}{1+0.5(x-2)^2}$

$$0.4\text{-coupe}(F) = \{x \in U / \mu_F(x) \geq \alpha\} \Rightarrow \frac{1}{1+0.5(x-2)^2} \geq \alpha$$

$$1+0.5(x-2)^2 \leq \frac{1}{\alpha}$$

$$(x-2)^2 \leq 2 \frac{1-\alpha}{\alpha} \quad (01)$$

$$(x-2)^2 - 2 \frac{1-\alpha}{\alpha} \leq 0 \Rightarrow \left(x-2 - \sqrt{\frac{2(1-\alpha)}{\alpha}}\right) \left(x-2 + \sqrt{\frac{2(1-\alpha)}{\alpha}}\right) \leq 0$$

$$x \in \left[2 - \sqrt{\frac{2(1-\alpha)}{\alpha}}, 2 + \sqrt{\frac{2(1-\alpha)}{\alpha}}\right] \Rightarrow \alpha\text{-coupe} = \left[2 - \sqrt{\frac{2(1-\alpha)}{\alpha}}, 2 + \sqrt{\frac{2(1-\alpha)}{\alpha}}\right]$$

(1.5)

**Rattrapage et
corrigé
2021/2022**

Rattrapage du module : « Techniques d'intelligence artificielle »

Questions de cours (04.50 pts):

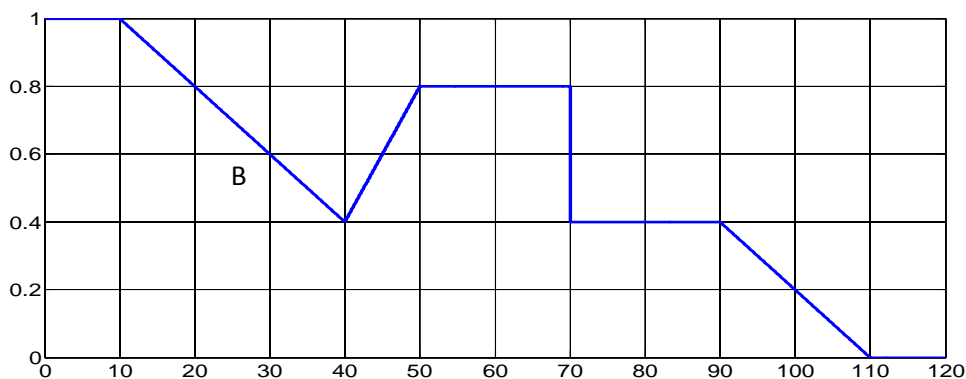
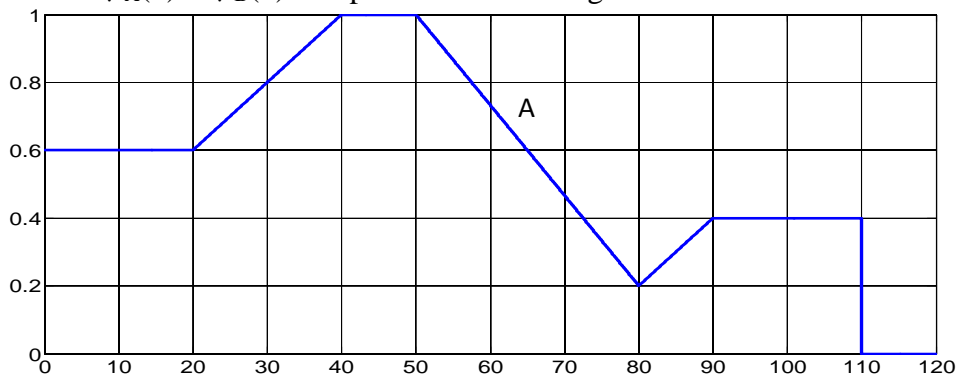
- 1- Quel est le rôle des synapses et donner leur principale particularité ?
- 2- Quelle est l'utilité de la fonction d'activation ?
- 3- Quelle est l'autre notation utilisée pour représenter les fonctions d'appartenance et donner sa représentation?

Exercice 1 (07.50 pts): On définit un ensemble flou A représenté par l'expression mathématique suivante :

$$\mu_A(x) = \begin{cases} 1 - \frac{4}{\pi} \left| x - \frac{\pi}{2} \right| & \text{si } x \in \left[\frac{\pi}{4}, \frac{3\pi}{4} \right] \\ 0 & \text{ailleurs} \end{cases}$$

- 1- Représenter graphiquement l'ensemble flou A sur l'intervalle $[0, 2\pi]$?
- 2- Calculer la hauteur, le support, le noyau et 0.25-coupe pour l'ensemble flou A ?
- 3- Représenter graphiquement l'ensemble flou \bar{A} et déterminer son équation mathématique ?

Exercice 2 (08.00 pts): Soient A et B deux ensembles flous décrits, respectivement, par leurs fonctions d'appartenances : $\mu_A(x)$ et $\mu_B(x)$ et représentés sur les figures suivantes :

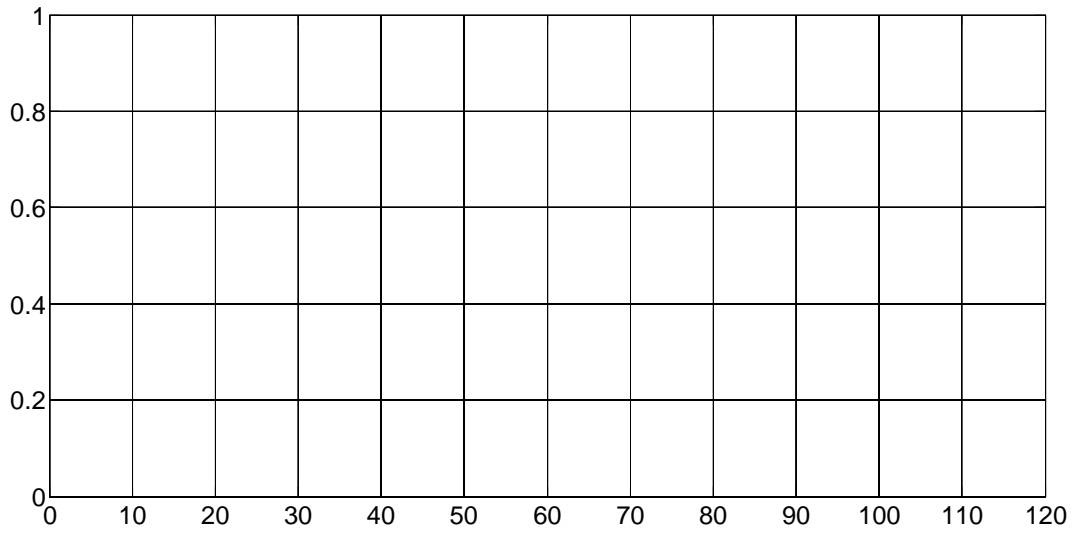


- 1- Déterminer graphiquement les ensembles flous suivants : \bar{A} , \bar{B} , $A \cap B$, $A \cup B$, $\bar{A} \cup B$, $A \cap \bar{B}$
- 2- Calculer la hauteur, le support, le noyau et 0.7-coupe pour chaque ensemble flou A et B ?
- 3- Soit l'ensemble flou **F** défini par : $\mu_F(x) = \frac{1}{2(3x-1)^2}$, déterminer le α -coupe de F en fonction de α ?

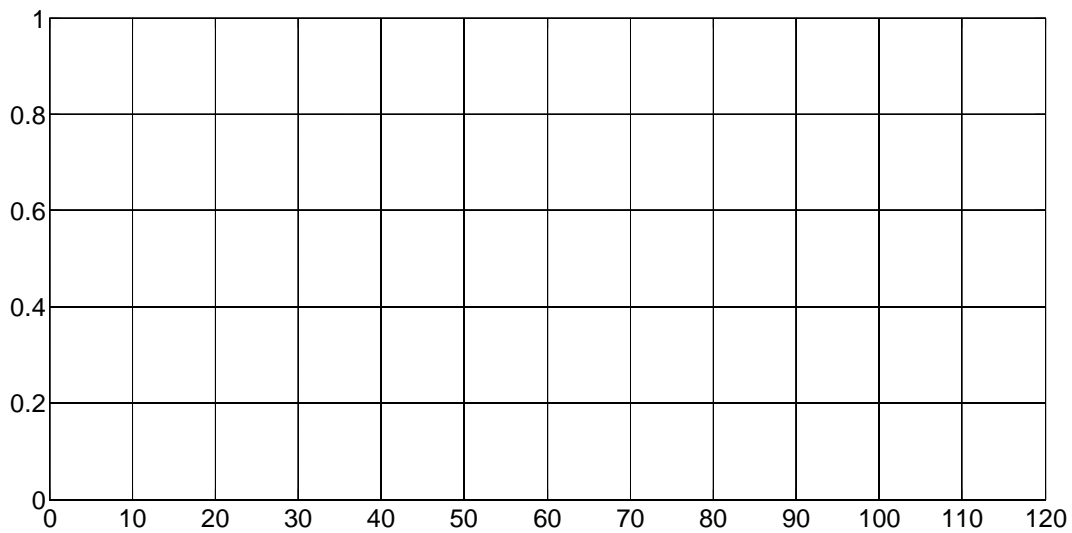
Nom :

Prénom :

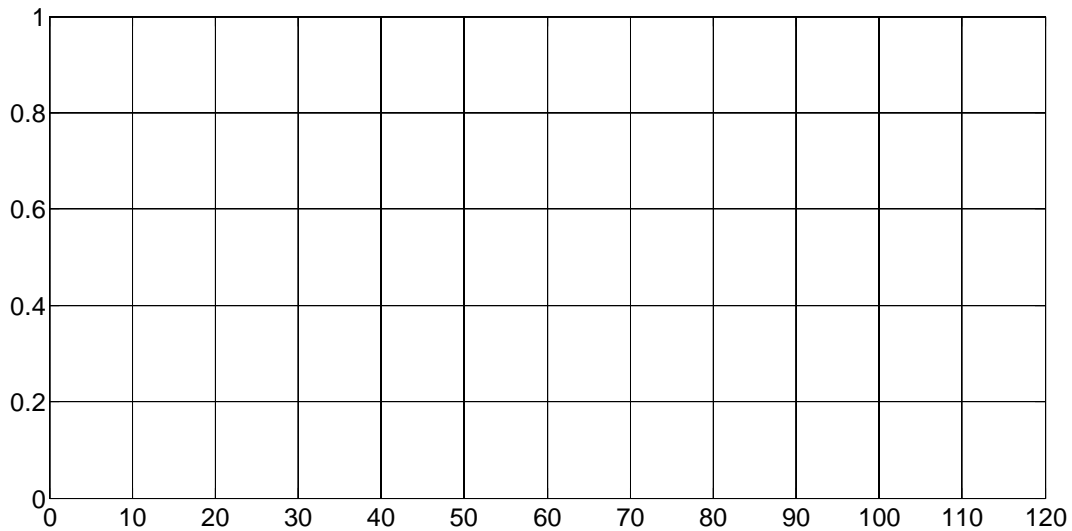
Groupe :



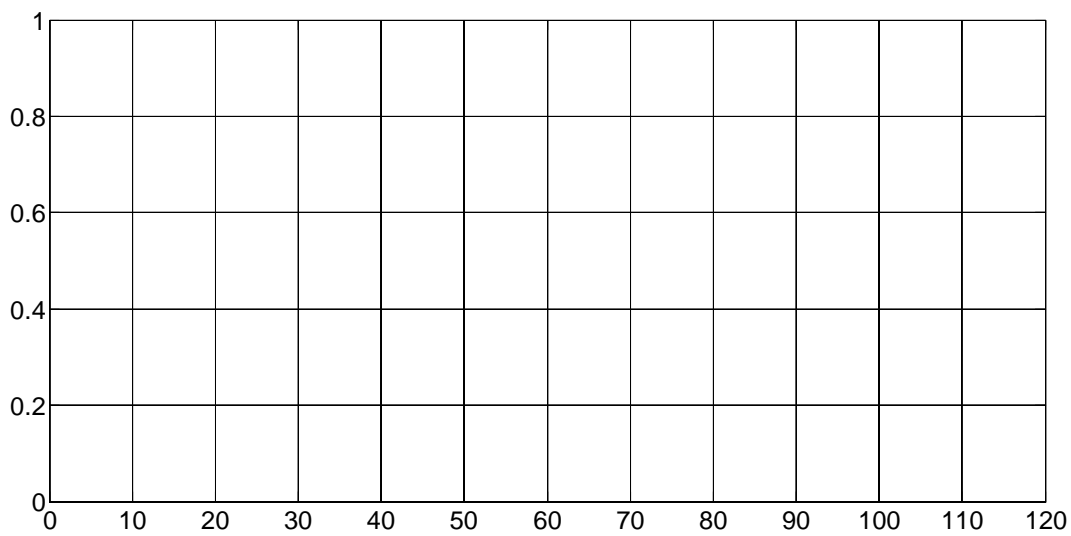
L'ensemble flou \bar{A}



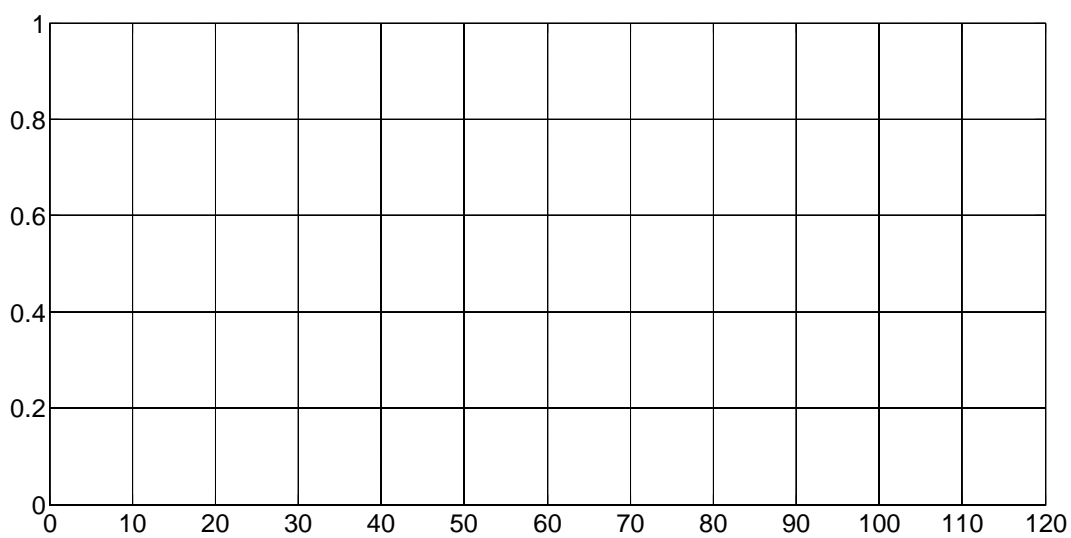
L'ensemble flou \bar{B}



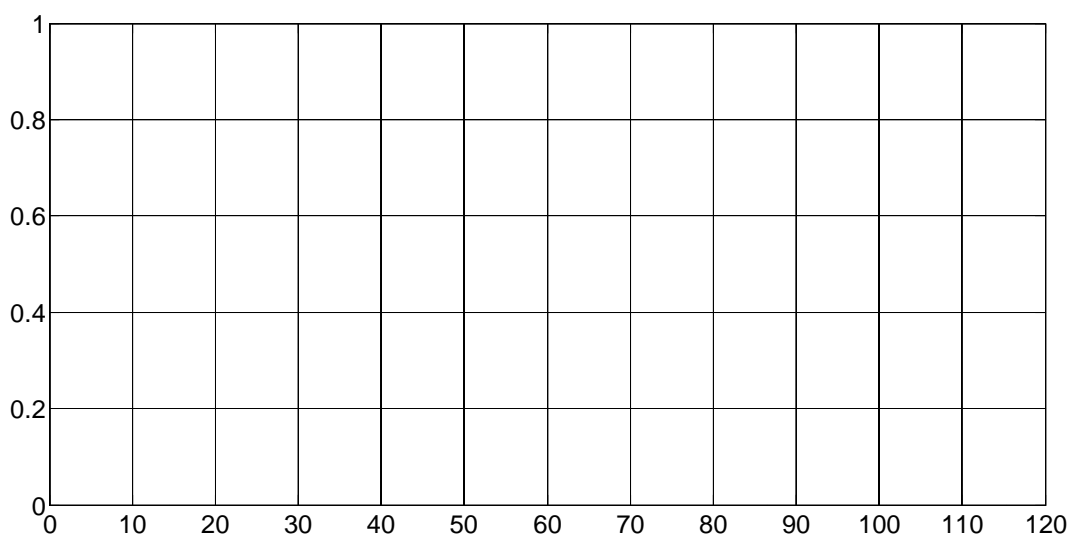
L'ensemble flou $A \cap B$



L'ensemble flou $A \cup B$



L'ensemble flou $\bar{A} \cup B$



L'ensemble flou $A \cap \bar{B}$

Corrigé rattrapage : « Techniques d'intelligence artificielle »

Questions de cours (04.50 pts):

1- Les synapses jouent un rôle important car elles sont le lien entre la sortie d'un neurone et l'entrée d'un autre neurone. (0.75)

Les synapses sont capables de s'adapter pour laisser passer le signal ou empêcher son passage : c'est le mécanisme d'adaptation (0.75)

2- La fonction d'activation permet d'introduire une non-linéarité dans la sortie du neurone et de décider ainsi si celui ci doit être activé ou non en calculant la somme pondérée et en ajoutant le biais. (1.5)

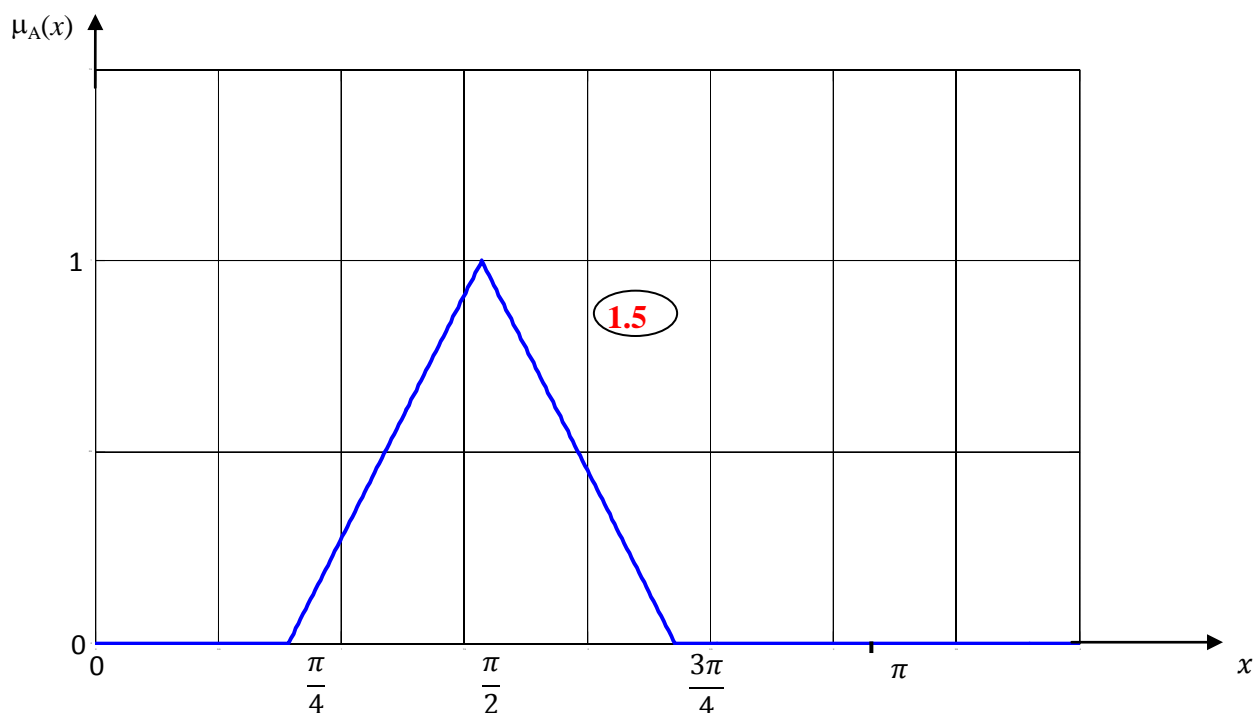
3- On utilise souvent une notation vectorielle pour représenter les fonctions d'appartenance, comme exemple : (1/17, 0/19) (0.75) (0.75)

Exercice 1 (07.50 pts):

On définit un ensemble flou A représenté par l'expression mathématique suivante :

$$\mu_A(x) = \begin{cases} 1 - \frac{4}{\pi} \left| x - \frac{\pi}{2} \right| & \text{si } x \in \left[\frac{\pi}{4}, \frac{3\pi}{4} \right] \\ 0 & \text{ailleurs} \end{cases}$$

1- Représenter graphiquement l'ensemble flou A ?



2- Calculer la hauteur, le support, le noyau et 0.25-coupe pour l'ensemble flou A :

$$h(\mathbf{A}) = \sup\{\mu_A(x) / x \in U\} = 1 \quad (0.5)$$

$$\text{supp}(\mathbf{A}) = \{x \in U / \mu_{\mathbf{A}}(x) > 0\} = \left[\frac{\pi}{4}, \frac{3\pi}{4} \right] \quad (0.5)$$

$$\text{noy}(\mathbf{A}) = \{x \in U / \mu_{\mathbf{A}}(x) = 1\} = \frac{\pi}{2} \quad (0.5)$$

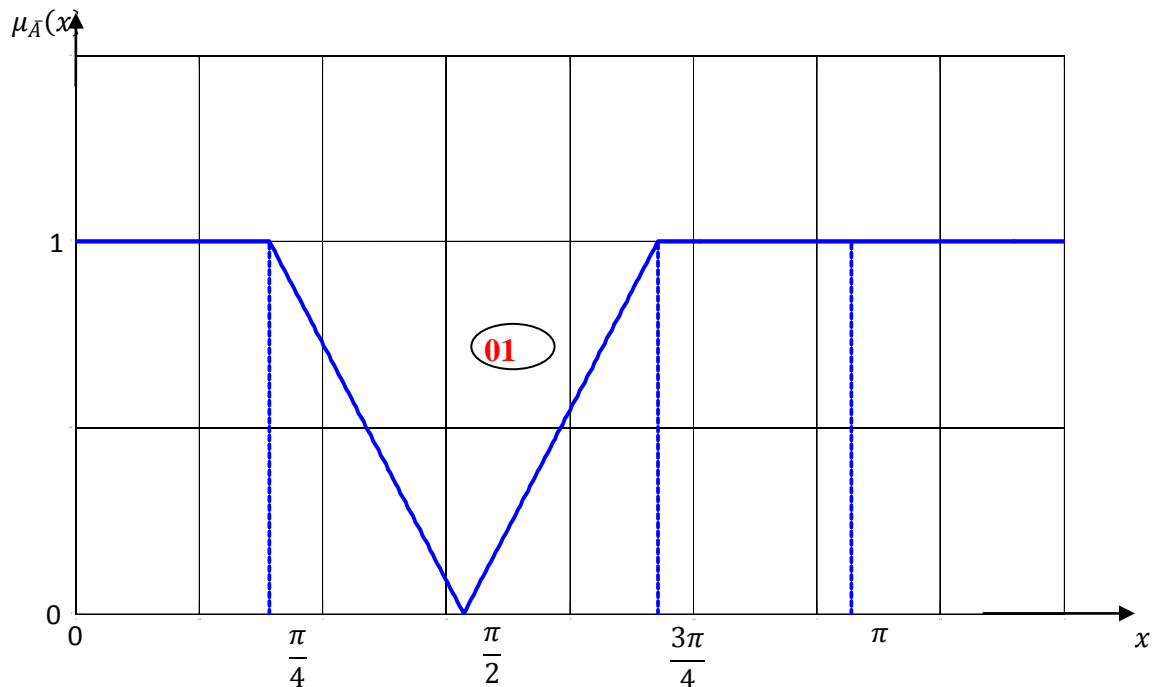
$$0.25\text{-coupe}(\mathbf{A}) = \{x \in U / \mu_{\mathbf{A}}(x) \geq 0.25\} \Rightarrow 1 - \frac{4}{\pi} \left| x - \frac{\pi}{2} \right| \geq 0.25 = \frac{1}{4} \quad (01)$$

$$\frac{4}{\pi} \left| x - \frac{\pi}{2} \right| \leq 1 - \frac{1}{4} = \frac{3}{4} \Rightarrow \left| x - \frac{\pi}{2} \right| \leq \frac{3\pi}{16} \Rightarrow -\frac{3\pi}{16} \leq x - \frac{\pi}{2} \leq \frac{3\pi}{16}$$

$$\frac{\pi}{2} - \frac{3\pi}{16} \leq x \leq \frac{\pi}{2} + \frac{3\pi}{16} \Rightarrow \frac{5\pi}{16} \leq x \leq \frac{11\pi}{16}$$

$$0.25\text{-coupe}(\mathbf{A}) = \left[\frac{5\pi}{16}, \frac{11\pi}{16} \right] \quad (01)$$

3- Représenter graphiquement l'ensemble flou $\bar{\mathbf{A}}$ et déterminer son équation mathématique :



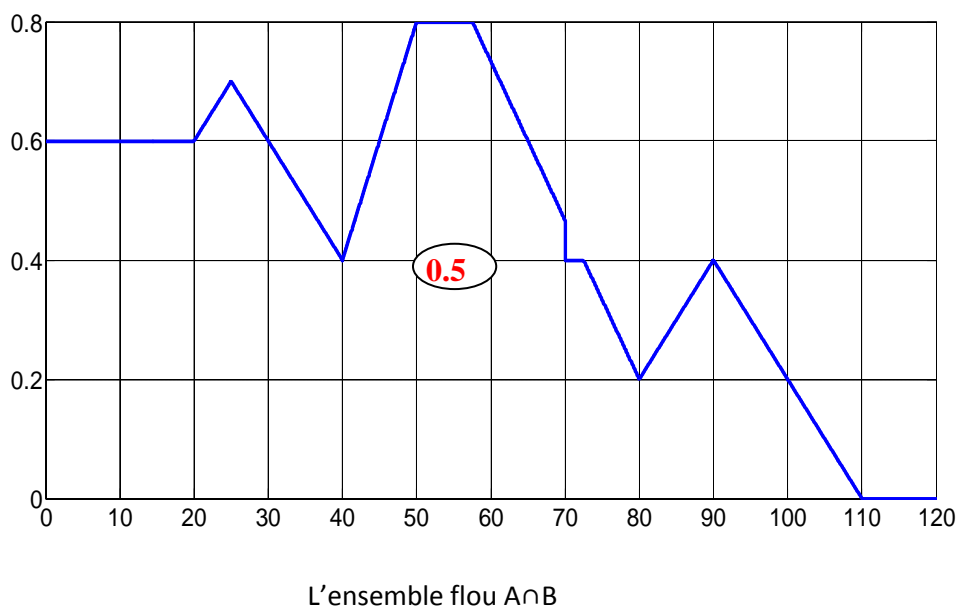
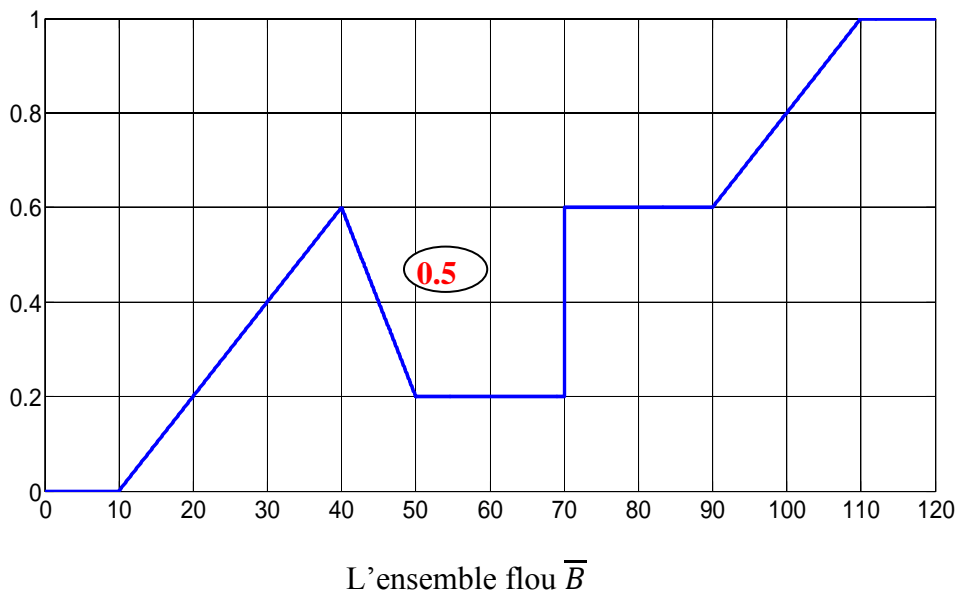
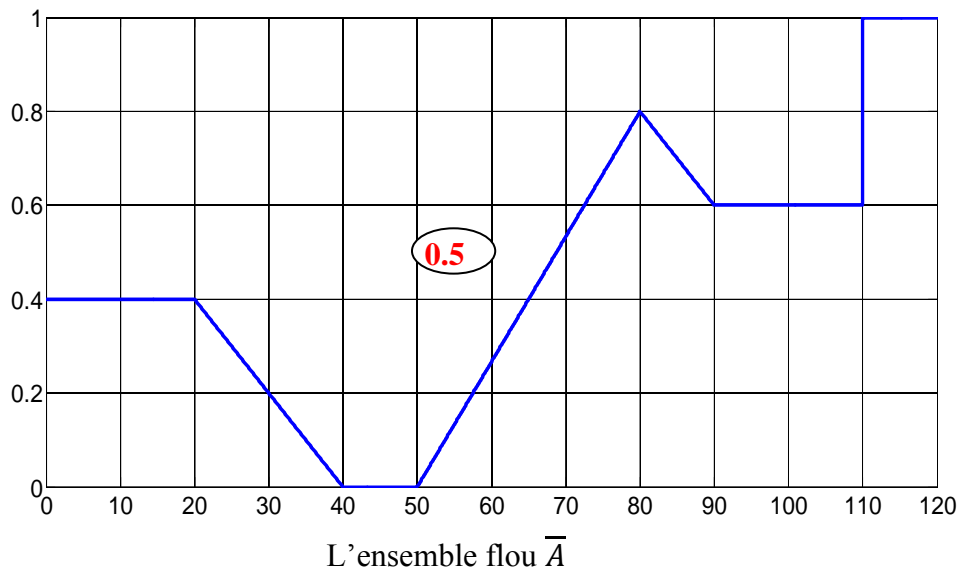
$$\mu_{\bar{\mathbf{A}}}(x) = \begin{cases} \frac{4}{\pi} \left| x - \frac{\pi}{2} \right| & \text{si } x \in \left[\frac{\pi}{4}, \frac{3\pi}{4} \right] \\ 1 & \text{ailleurs} \end{cases}$$

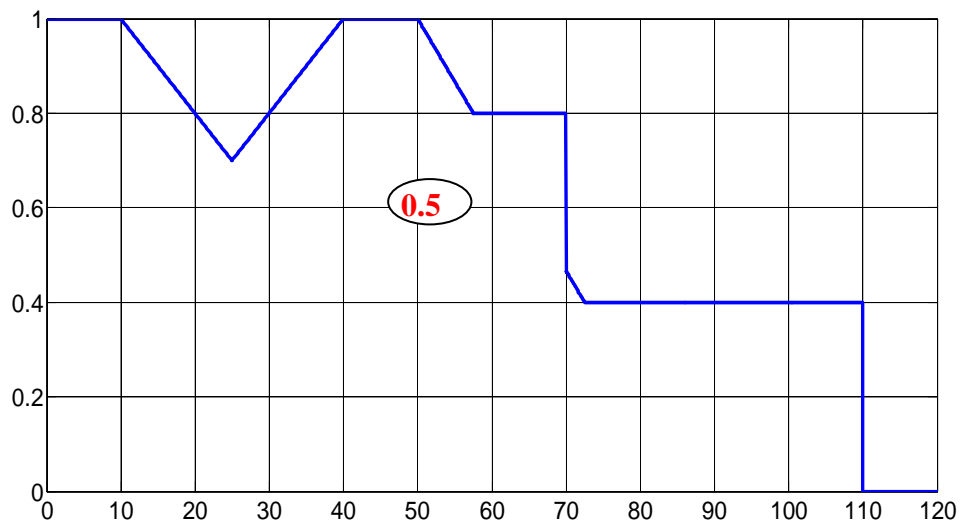
Ou bien :

$$\mu_{\bar{\mathbf{A}}}(x) = \begin{cases} -\frac{4}{\pi}x + 2 & \text{si } x \in \left[\frac{\pi}{4}, \frac{\pi}{2} \right] \\ \frac{4}{\pi}x - 2 & \text{si } x \in \left[\frac{\pi}{2}, \frac{3\pi}{4} \right] \\ 1 & \text{ailleurs} \end{cases} \quad (1.5)$$

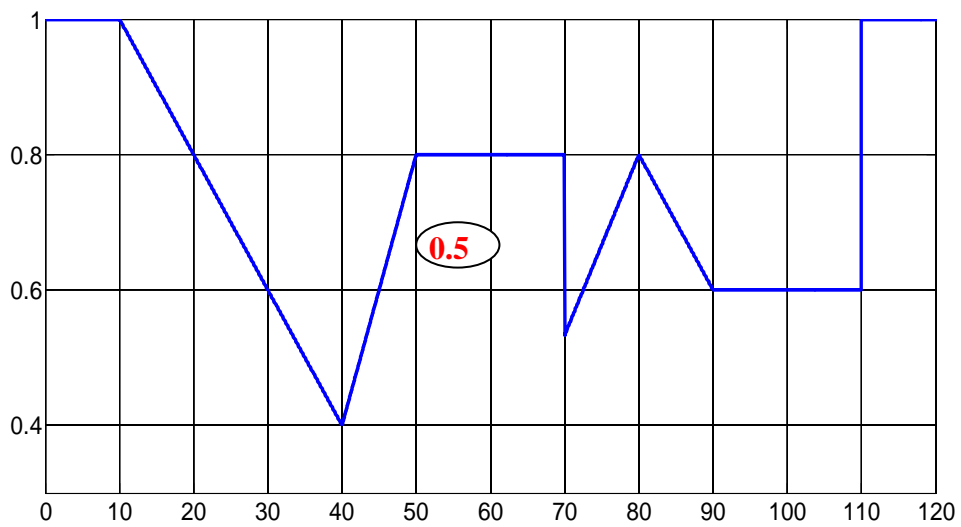
Exercice 2 (08.00 pts):

1- Déterminer graphiquement les ensembles flous suivants : \bar{A} , \bar{B} , $A \cap B$, $A \cup B$, $\bar{A} \cup B$, $A \cap \bar{B}$

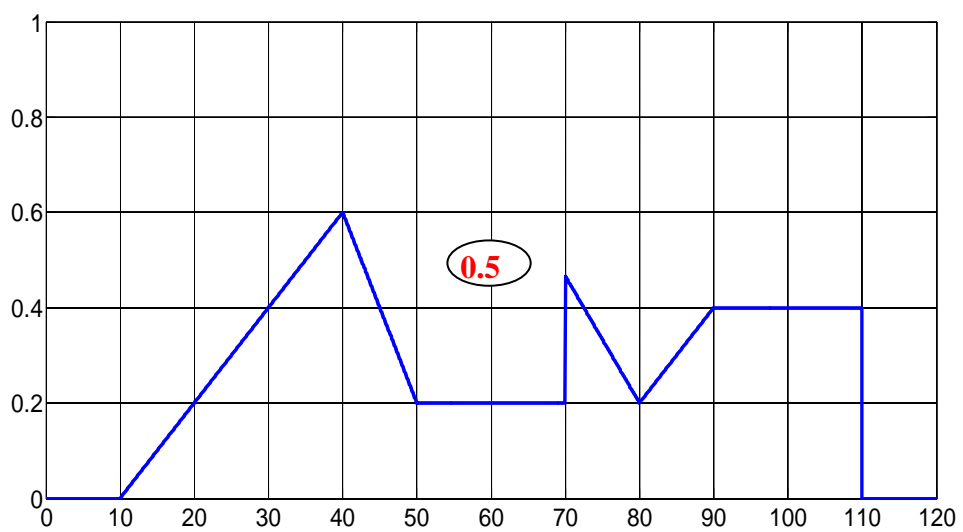




L'ensemble flou $A \cup B$



L'ensemble flou $\bar{A} \cup B$



L'ensemble flou $A \cap \bar{B}$

2- Calculer la hauteur, le support, le noyau et 0.4-coupe pour chaque ensemble flou A et B ?

$$h(A) = \sup\{\mu_A(x) / x \in U\} = 1 \quad (0.25)$$

$$h(B) = 1 \quad (0.25)$$

$$\text{supp}(A) = \{x \in U / \mu_A(x) > 0\} = [0, 110] \quad (0.25)$$

$$\text{supp}(B) = [0, 110] \quad (0.25)$$

$$\text{noy}(A) = \{x \in U / \mu_A(x) = 1\} = [40, 50] \quad (0.25)$$

$$\text{noy}(B) = [0, 10] \quad (0.25)$$

$$0.7\text{-coupe}(A) = \{x \in U / \mu_A(x) \geq 0.7\}$$

On définit l'équation de la droite pour l'ensemble A dans l'intervalle [20, 40] : $y = 0.02x + 0.2$

$$y = 0.7 \Rightarrow x = \frac{y - 0.2}{0.02} = \frac{0.5}{0.02} = 25$$

On définit l'équation de la droite pour l'ensemble A dans l'intervalle [50, 80] : $y = -\frac{0.8}{30}x + \frac{7}{3}$

$$y = 0.7 \Rightarrow x = \frac{\frac{7}{3} - y}{\frac{0.8}{30}} = \frac{\frac{4.9}{3}}{\frac{0.8}{30}} = \frac{49}{0.8} = 61.25 \Rightarrow 0.7\text{-coupe}(A) = [25, 61.25] \quad (0.5)$$

On définit l'équation de la droite pour l'ensemble B dans l'intervalle [10, 40] : $y = -0.02x + 1.2$

$$y = 0.7 \Rightarrow x = \frac{1.2 - y}{0.02} = \frac{0.5}{0.02} = 25 \Rightarrow y \geq 0.7 \text{ pour } x \in [0, 25] \quad (I)$$

On définit l'équation de la droite pour l'ensemble B dans l'intervalle [40, 50] : $y = 0.04x - 1.2$

$$y = 0.7 \Rightarrow x = \frac{y + 1.2}{0.04} = \frac{1.9}{0.04} = 47.5 \Rightarrow y \geq 0.7 \text{ pour } x \in [47.5, 70] \quad (II)$$

De (I) et (II) on déduit que : $0.7\text{-coupe}(B) = [0, 25] \cup [47.5, 70] \quad (01)$

3- Soit l'ensemble flou F défini par : $\mu_F(x) = \frac{1}{2(3x-1)^2}$

$$\alpha\text{-coupe}(F) = \{x \in U / \mu_F(x) \geq \alpha\} \Rightarrow \frac{1}{2(3x-1)^2} \geq \alpha$$

$$2(3x-1)^2 \leq \frac{1}{\alpha}$$

$$(3x-1)^2 \leq \frac{0.5}{\alpha} \quad (01)$$

$$(3x-1)^2 - \frac{0.5}{\alpha} \leq 0 \Rightarrow \left(3x-1 - \sqrt{\frac{0.5}{\alpha}}\right) \left(3x-1 + \sqrt{\frac{0.5}{\alpha}}\right) \leq 0$$

$$x \in \left[\frac{1 - \sqrt{\frac{0.5}{\alpha}}}{3}, \frac{1 + \sqrt{\frac{0.5}{\alpha}}}{3} \right] \Rightarrow \alpha - \text{coupe} = \left[\frac{1 - \sqrt{\frac{0.5}{\alpha}}}{3}, \frac{1 + \sqrt{\frac{0.5}{\alpha}}}{3} \right] \quad \text{01}$$