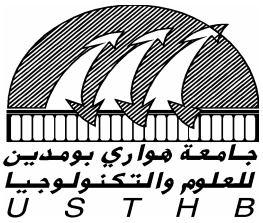
N° d'ordre : 06/2006-M/M T

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIR
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET LA RECHERCHE SCIETIFIQUE



Université des sciences et de la technologie Houari Boumediene
Faculté de Mathématiques

Mémoire

Présenté en vue de l'obtention du diplôme de Magister
En : Mathématiques

Spécialité : Probabilités et Statistiques

Par : **Mme MEBREK Fatma**

Thème

Apprentissage par les méthodes de segmentation

Soutenu publiquement, le 15 / 05 / 2006, devant le jury composé de :

M. K. BOUKHETALA,	Professeur U.S.T.H.B	Président.
M. M. DJEDOUR,	Professeur U.S.T.H.B	Directeur de thèse.
M. A. AISSANI,	Professeur U.S.T.H.B	Examinateur.
M D. CHAABANE,	Maître de Conférences U.S.T.H.B	Examinateur.
Mme. H.DIB,	Maître Assistante U.S.T.H.B	Invité.
Dr. M. BOUROUBA	Maître de Conférences C.H.U.BOLOGHINE.	Invité.

REMERCIEMENTS

Je tiens à remercier le bon DIEU, le tout puissant de m'avoir donnée la volonté, le courage, la patience et guidée vers les portes de savoir.

J'exprime ma gratitude et sincères remerciements à Monsieur M. Djedour, Professeur à L'U.S.T.H.B. de m'avoir proposée ce travail ainsi comme pour son aide précieuse et ses conseils et suggestions qui mon permis d'améliorer et de terminer mon mémoire.

J'exprime également ma reconnaissance à Monsieur K. Boukhetala. professeur à l'U.S.T.H.B, qui ma fait l'honneur de présider le jury.

Mes remerciements s'adressent aussi à M .A. Aissani, professeur à l'U.S.T.H.B, et M. D. Chaabane qui ont accepté de faire partie de mon jury. Egalement et plus particulièrement, je présente mes remerciements à Madame H. Dib, examinatrice, pour son aide et suggestions, qui m'ont beaucoup aidés.

J'adresse également ma reconnaissance et remerciements à docteur Bourouba Maître de Conférence au C.H.U de Bouloghine, pour son aide et collaboration.

Je tiens à manifester ma reconnaissance à mes enseignants : Mme.Djemai, M.Messaci et M. Astouati et M, Larjen durant ma première année de Poste graduation, pour l'apport bénéfique de leur cours.

J'adresse enfin ma reconnaissance à ma famille qui a toujours été présente pendant mes études.

Je tiens à exprimer mes plus grands remerciements à mon mari et mon frère pour leur encouragement, soutien moral, conseille et leur patience.

Sommaire

Introduction générale

Depuis plus de vingt ans, l'apprentissage par les méthodes non paramétriques ne cessent d'être un motif de recherche, aussi bien en analyse de données et statistique qu'en apprentissage et reconnaissance des formes. Dans notre travail, nous nous intéressons à deux de ces méthodes : arbre de décision par méthode CART, ainsi que les réseaux de neurones.

Pour un réseau de neurone, l'apprentissage peut être considéré comme un problème de mise à jour des poids de connexions au sein du réseau. Afin de réussir la tâche qui lui est demandée, il est donc la caractéristique principale des réseaux et il peut se faire de différentes manières et selon différentes règles. La capacité d'apprentissage des réseaux de neurones et leur propriété d'approximateur universel et parcimonieuse, leur donnent un intérêt très particulier dans plusieurs domaines.

Plusieurs types de réseaux de neurones ont été développés, leurs domaines d'application étant la robotique [29], l'optimisation [30], le traitement d'images et de la parole [30].

Le réseau de Hopfield [28] est une de ces généralisations, la machine de Bolzamann [30], repose sur le concept de la thermodynamique statistique. Ils ont été utilisés avec succès dans le traitement d'image, d'optimisation. Egalement les Cartes auto organisatrices (ou Carte de kohonen) [30] ont été utilisées pour générer des cartes phonétiques, des diagnostic de pannes, … etc.

Dans ce travail, on va utiliser pour la modélisation de notre problème les réseaux de neurones Multicouches.

Dans l'utilisation d'apprentissage par les arbres de décision, nous sommes confrontés à choisir de bons critères statistiques pour la construction de l'arbre. Cependant, les règles logiques régissant les arbres de décision vont nous permettre de choisir la variable la plus discrimination pour une représentation optimale de notre arbre [3].

On va construire un arbre de décision à partir d'un ensemble de variables descriptives, et une variable discriminante basées sur des données médicales.

Il existe plusieurs méthodes de segmentation, celle qui sera utilisée dans ce travail est la méthode CART (classification And Regression Tree), elle diffère des autres méthodes par le mode de construction de l'arbre et la technique d'élagage.

Le but de notre mémoire est l'utilisation des méthodes décisionnelles pour l'aide à la décision dans le domaine médical, dans une pathologie très fréquente en Algérie, le cancer de la thyroïde. L'aide à la prise de décision dans ce domaine constitue une étape très importante pour le diagnostique du malade. Ainsi on va élaborer un utilitaire dont le praticien disposera qu'il pourra utiliser comme moyen d'expertise de son diagnostic et de la future prise en charge thérapeutique.

Le mémoire est constitué en cinq chapitres:

Dans le premier chapitre, nous définissons les réseaux de neurones ainsi que leur mode de fonctionnement, notamment les règles d'apprentissage

Dans le second chapitre, nous présentons le Perceptron Multicouches, ainsi que l'algorithme de rétropagation du gradient.

Le chapitre trois sera consacré à un survol sur la théorie des graphes, aux arbres de décision et leurs modes de constructions et plus particulièrement la méthode CART.

Le quatrième chapitre est consacré à l'implémentation des arbres de décisions basée sur la méthode CART

Le dernier chapitre est consacré à la description du problème médical, interprétation et comparaison des résultats obtenus par et les arbres de décisions, et les réseaux de neurones

Nous terminons ce mémoire par une conclusion générale résumant le travail effectué et nous proposons des perspectives intéressantes.

Chapitre I
Réseaux de neurones artificiels

I.1. Introduction

Les réseaux de neurones inspiraient beaucoup de chercheurs, l'apparition des travaux Minsky et Papert [24], montraient l'impossibilité de réaliser les fonctions logiques avec les réseaux de neurones existants.

Au début des années 80 la recherche dans ce domaine a été relancée. La reprise des recherches est due a l'apparition de nouvelles architectures de réseaux de neurones. Entre autres la possibilité de construire des réseaux en plusieurs couches était importante, ils permettaient la résolution de plusieurs problèmes existants.

Les modèles neuronaux sons inspirés d'une représentation graphique ressemblant à la complexité des éléments du cerveau (KOHONEN et al. (1991)). Ces techniques sont très utiles pour représentation des modèles mathématiques, à un contexte non linéaire où il devient possible de traiter de nouvelles données et effectuer des modifications de données en temps réel**.**

Les applications principales se trouvent dans les domaines de la théorie à la décision, de la prévision, de la robotique, de la biologie, ainsi que plusieurs domaines médicaux. De telles applications comprennent souvent des modèles de type de régression qui sont améliorés par les méthodes neuronales.

Ce chapitre consiste en une introduction au modèle biologique, il permet d'accéder rapidement aux principes fondamentaux. Par la suite, nous représentons les différents réseaux de neurones artificiels utilisés dans diverses applications.

I.2. Le modèle biologique

Des cellules interconnectées appelées neurones sont les constituants du cerveau humain. Le neurone biologique se compose d'un corps cellulaire (soma) et de deux prolongements : les dendrites qui sont les entrées ou récepteurs du neurone, et l'axone qui est la sortie ou émetteur de la cellule tel que le montre la figure I.1. Il reçoit des signaux émanant d'autres neurones au niveau des dendrites et transmet l'information, générée dans son corps cellulaire, via l'axone. Chaque branche de cette arborisation se termine en un bouton synaptique autour duquel se trouvent les synapses. La synapse est une structure essentielle entre deux neurones permettant la transmission des signaux entre un axone et une dendrite.

Quand un signal arrive au niveau de la synapse, ces substances sont libérées, elles le traversent et se fixent sur les récepteurs du neurone récepteur [30].

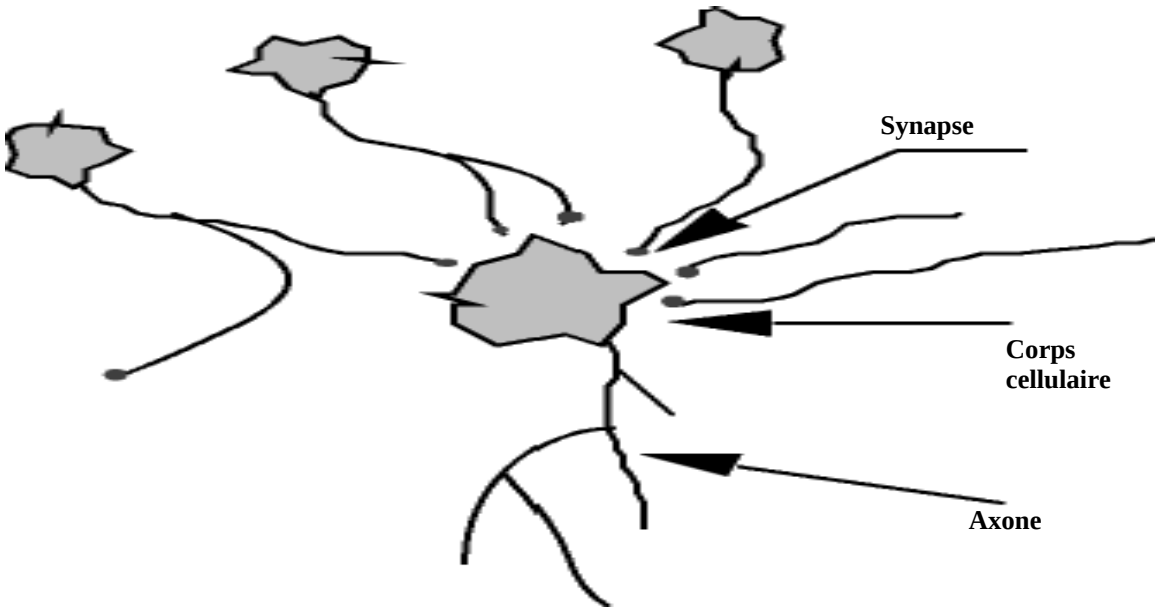


Figure I .1 : Modèle de neurone biologique

I.3. Le Neurone Formel

Il a été présenté par McCulloch et Pitts en 1943 [30-26], c'est une cellule ayant plusieurs entrées et une sortie. Pour chaque entrée, un poids synaptique lui est associé représentant ainsi la force de connexion entre cette entrée et le neurone. Ces entrées proviennent des neurones en amont et sa sortie alimente les neurones en aval [18]. La figure illustre un neurone formel

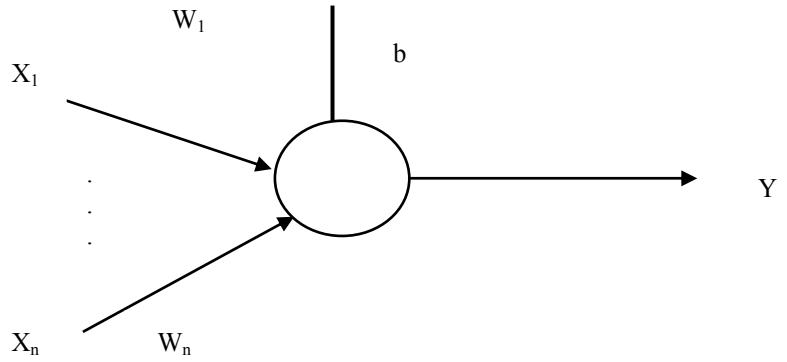


Figure I.2: Le neurone formel

Avec :

X_i : la $i^{éme}$entrée.

n : le nombre d'entrées.

W_i : le poids synaptique de la $i^{éme}$entrée.

b : la fonction seuil.

Y : la sortie du neurone.

Le neurone formel, est considéré comme une modélisation élémentaire du neurone réel, c'est un automate possédant ***n*** entrées réelles $X_1,\ldots,X_n$, et dont le traitement consiste à affecter à sa sortie Y, le résultat d'une fonction d'activation *f* de la somme pondérée de ses entrées.

Définition :

Un neurone est une fonction non linéaire, paramétrée à valeurs bornées.

On distingue deux types de réseaux de neurones ; les réseaux non bouclés et les réseaux bouclés. L'utilisation de chacun dépend de l'application envisagée.

I.4. Les réseaux de neurones non bouclés (Feed-Forward)

Un réseau de neurone non bouclé réalise une ou plusieurs fonction des ses entrées, par composition des fonctions réalisées par chacun des neurones [12]

Définition

C'est un réseau où les signaux ne peuvent se propager qu'en avant, c'est à dire ils traversent le réseau de l'entrée vers la sortie sans boucles de retour. Le temps nécessaire pour le calcul dans ce cas n'intervient pas car la somme pondérée des entrées se fait instantanément. Pour cette raison, les réseaux non bouclés sont souvent appelés « réseaux statique ».

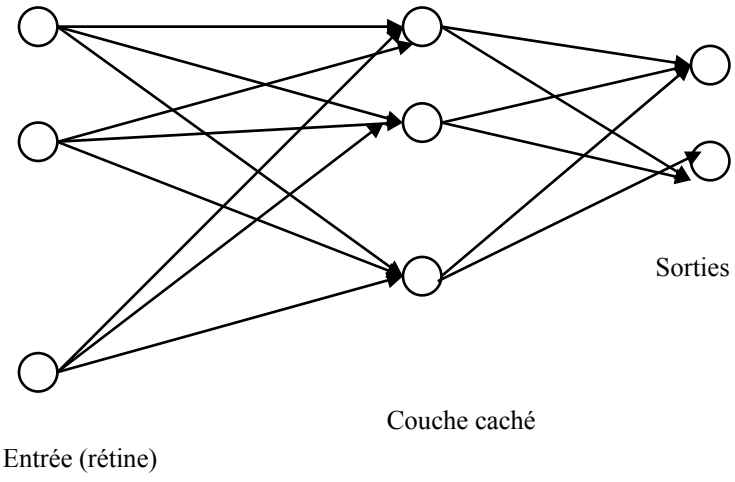


Figure I.3: Réseau de neurone à couche

I.5. Réseaux de neurones bouclés (Feed-back)

Définition

Le graphe des connexions dans ce types de réseaux est cyclique, c'est à dire lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de cycle). La sortie d'un neurone du réseau peut donc être fonction d'elle même. Cela n'est évidement concevable que si la notion de temps est explicitement prise en considération.

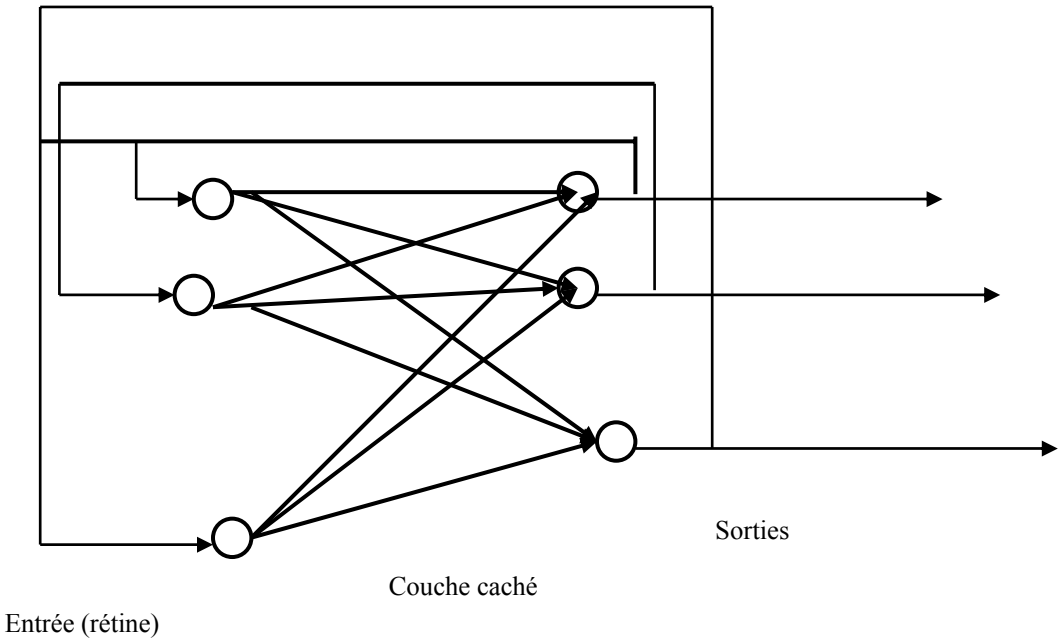


Figure I.4 : Réseau de neurone à couche

I.6. Les fonctions d'activation

Les fonctions d'activation utilisées dans les modèles connexionnistes d'aujourd'hui sont variées et certains modèles emploient même plusieurs fonctions différentes dans un même réseau, selon le rôle du neurone.

I.6.1 Les différentes types de fonction d'activation

La modalisation mathématique du neurone formel est la base des réseaux de neurones, c'est la succession de deux opérations tel que le montre la figure I.5 : une sommation des entrées pondérée par des coefficients appelés poids pour l'élaboration du potentiel du neurone et une opération non linéaire de feuillage qui calcule la sortie du neurone en fonction de ce potentiel.

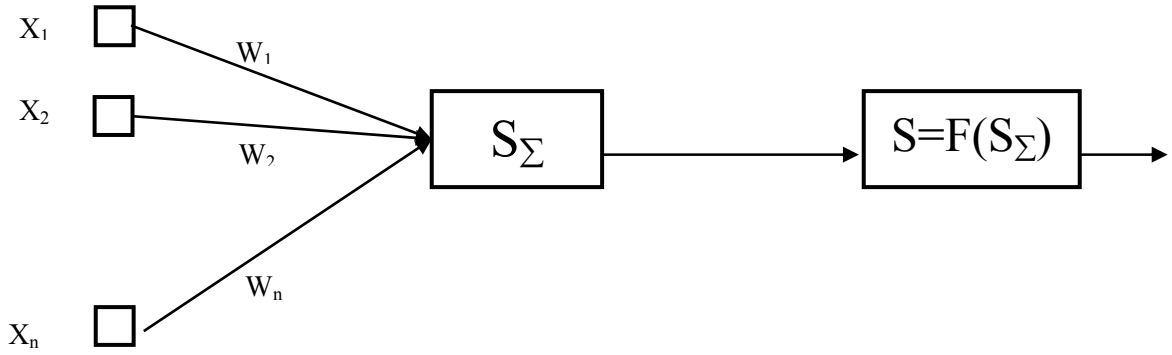


Figure I.5 : Modèle de neurone formel et de sa fonction d'activité

où

$$S=F(S_{\Sigma}) \tag{I.1}$$

$$S_{\Sigma} = \sum_{i=1}^{n} X_i w_i = \underline{W}^t \underline{X} \tag{I.2}$$

Avec S_{Σ} : le potentiel d'activation F(.) : la fonction d'activation

$\underline{X}$: Vecteur entrées du neurone $\underline{W}$: le vecteur poids.

L'utilisation d'une fonction d'activation permet au réseau de neurone de modéliser des équations dont la sortie n'est pas une combinaison linéaire des entrées. Cette caractéristique confère au réseau de neurone de grandes capacités de modélisation fortement appréciées pour la résolution de problèmes non linéaires.

Plusieurs fonctions d'activation peuvent être utilisées parmi les quelles :

a- La fonction linéaire

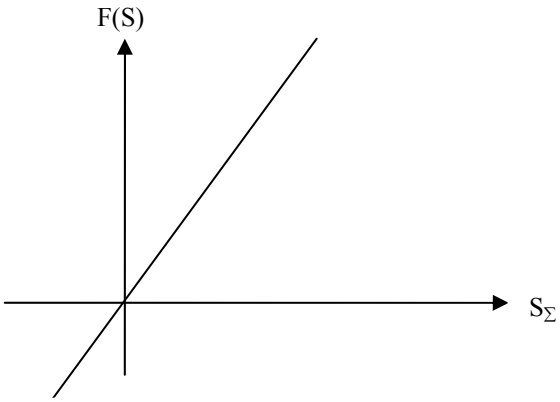


Figure I.6 : Fonction linéaire

$$F(S_\Sigma)=S_\Sigma \qquad (I.3)$$

b- la fonction sigmoïde

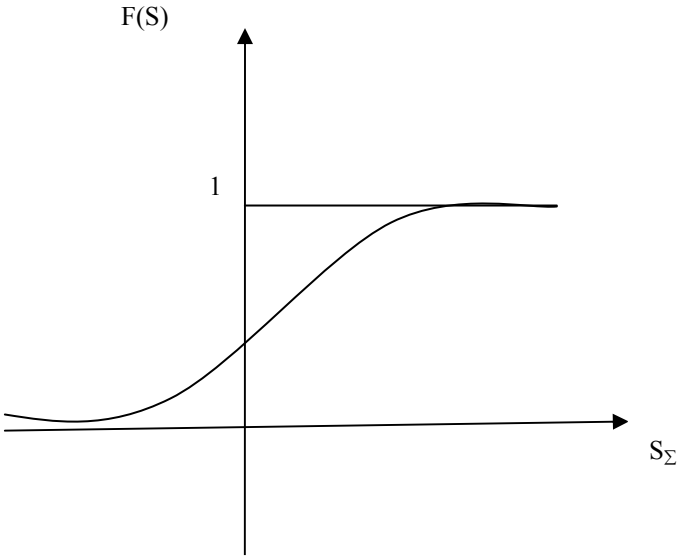


Figure I.7 : Fonction sigmoïde

La fonction sigmoïde est l'équivalent continu de la fonction linéaire. Ce type de fonction est généralement employé dans le Perceptron Multicouches.

$$F(S_\Sigma) = \frac{1}{1+\exp(-S_\Sigma)} \qquad (I.4)$$

La dérivée de cette fonction possède l'avantage d'être simple à calculer :

$$\frac{dF(S_\Sigma)}{dS_\Sigma} = F(S_\Sigma)(1-F(S_\Sigma)) \qquad (I.5)$$

c- la fonction tangente hyperbolique

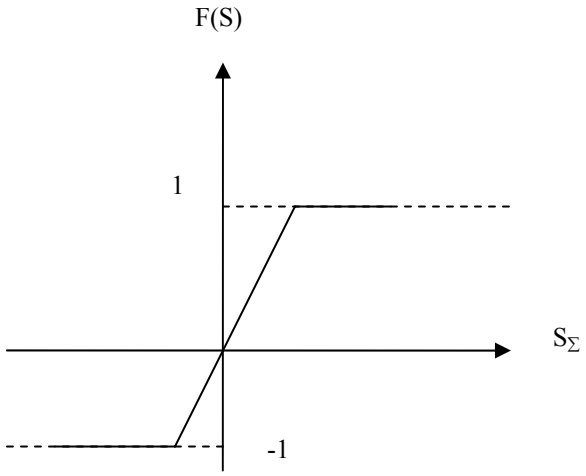


Figure I.8 : fonction tangente hyperbolique

Ce type de fonction est employé aussi dans les réseaux Multicouche.

$$F(S_\Sigma)= \tanh(S_\Sigma)= \frac{1-e^{-S_\Sigma}}{1+e^{-S_\Sigma}} \qquad (I.6)$$

La dérivée de cette fonction est simple à calculer :

$$\frac{dF(S_\Sigma)}{d(S_\Sigma)} = (1-(F(S_\Sigma))^2) \qquad (I.7)$$

d- Fonction gaussienne

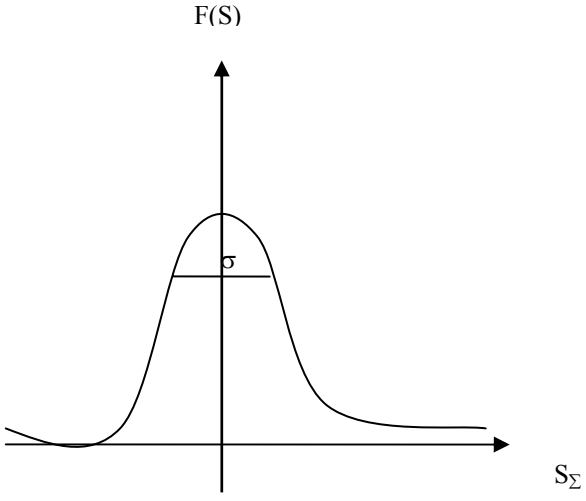


Figure I.9 : fonction gaussienne.

Cette fonction est souvent employée dans les réseaux RBF (Radial Basis Fonction)

$$F(S)= e^{-\frac{(S_\Sigma)^2}{2\sigma^2}} \qquad (I.8)$$

Le choix de la fonction d'activation dépend de l'application à traiter. Dans le cas de problème de classification, on utilise généralement des fonctions à seuil ou des sigmoïdes, car ces dernières ont l'avantage d'être dérivables, ce qui permet l'utilisation des algorithmes d'apprentissage qu'on va de type gradient.

I.7. Le nombre de neurones cachés

À l'heure actuelle, il n'existe pas de résultat théorique permettant de déterminer le nombre de neurones cachés, nécessaire pour obtenir une performance spécifiée du modèle. Pour remédier à cet obstacle, on élabore une procédure numérique de conception de modèle.

I.8. Le choix des poids synaptiques

Une fois l'architecture du réseau déterminé, il reste encore à choisir les valeurs des poids synaptiques des connexions. De ce fait, la plupart des modèles neurométique disposent de mécanismes capables de modifier leurs poids synaptiques automatiquement, ils sont dotés de règles d'apprentissage.

I.9. L'apprentissage des réseaux de neurones

On appelle apprentissage des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseaux, afin que celui-ci remplisse au mieux la tâche qui lui est affectée. On distingue deux types d'apprentissages principaux, l'apprentissage supervisé et l'apprentissage non supervisé [21].

I.9.1. l'apprentissage supervisé

Dans ce type d'apprentissage, le réseau s'adapte par comparaison entre les résultats qu'il a calculés, en fonction des entrées fournies et la réponse attendue en sortie. Ainsi le réseau va se modifier jusqu'à ce qu'il trouve la bonne sortie, c'est-à-dire celle attendue, correspondant à une entrée donnée.

I.9.2. l'apprentissage non supervisé

Ce type d'apprentissage est choisi lorsqu'il n'y pas de connaissances à priori des sorties désirés pour des entrées données. En fait, c'est de l'apprentissage par exploration où l'algorithme d'apprentissage ajuste les poids des liens entre neurones de façon à maximiser la qualité de classification des entrées.

I.10. Approximation de fonctions par les réseaux de neurones

Lors d'un apprentissage supervisé, on veut faire correspondre à chaque vecteur d'entrée, un vecteur de sortie. En d'autres termes, le réseau doit réaliser une fonction vectorielle voulue. Si cette fonction est connue analytiquement il s'agit d'une tâche d'approximation de fonction. Si on ne dispose que d'un certain nombre de mesures, il s'agit d'une tâche de régression.

I.10.1. Approximateurs universel

Définition :

N'importe quelle fonction suffisamment continue de plusieurs variables peut-être approchées avec une précision arbitraire par une fonction polynôme.

Théorème d'approximation universel :

Soit φ une fonction continue à valeurs réelles sur $[0,1]^n$ donnée, (ou tout autre sous ensemble compact sur R_n) et $\varepsilon<0$, il existe des vecteurs $W_1, W_2, \ldots, W_N$ et une fonction paramétrée

G : $[0,1]^n \rightarrow R$ telle que :

$$|G(X,W)-f(X)|< \varepsilon \qquad \text{pour tout } X \in [0,1]^n$$

Où

$$G(X,W) = \sum_{j=1}^{N} \alpha_j \varphi(W_j^T X + \theta_j) \tag{I.9}$$

avec : $W_j \in R^n$ et $\alpha_j, \theta \in R$ sont fixés.

I.10.2. Approximation parcimonieuse

Propriété 1

Tout fonction suffisamment régulière peut être approchée uniformément, avec une précision arbitraire dans un domaine fini de l'espace de ces variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire [19].

Propriété 2

Les réseaux de neurones non linéaires par rapport à leurs paramètres sont des approximateurs parcimonieux.

Dans la pratique, le nombre de fonctions nécessaires pour réaliser une approximation est un critère important dans le choix d'un approximateur.

Le concepteur du modèle doit toujours faire en sorte que le nombre de paramètres ajustables soit le plus faible possible : on dit que l'on cherche l'approximation la plus parcimonieuse.

Propriété 3

On montre (Barron (1993) [1]) que si l'approximation dépend des paramètres ajustables de manière non linéaire, elle est plus parcimonieuse que si elle dépend linéairement des paramètres. Plus précisément on montre que le nombre de paramètres, pour une précision donnée, croit exponentiellement avec le nombre de variables dans le cas des approximateurs linéaires par rapport à leurs paramètres, alors qu'il croit linéairement avec ce nombre pour les approximateurs non linéaire par rapport à leurs paramètres.

Remarque

Lorsqu'on cherche à modéliser un processus à partir des données, on s'efforce toujours d'obtenir les résultats les plus satisfaisants possibles avec un nombre minimum de paramètres ajustables.

Si la sortie de réseau de neurones est une fonction non linéaire des paramètres ajustables, elle est plus parcimonieuse que si elle est une fonction linéaire de ces paramètres. De plus pour des réseaux de neurones à fonction d'activation sigmoïdales, l'erreur commise dans l'approximation comme l'inverse du nombre de neurones cachés. Elle est indépendante du nombre de variables de la fonction à approcher. Par conséquent ce résultat s'applique aux réseaux de neurones à fonction d'activation sigmoïdale puisque la sortie de ces réseaux n'est pas linéaire par rapport aux poids synaptiques.

Ces propriétés montre l'intérêt des réseaux de neurones par rapport à d'autres approximateurs.

I.11. Fonction de coût

La fonction de coût permet de mesurer l'écart entre le modèle et les observations. Si cet écart est important, la fonction de coût doit être grande, et inversement. Il existe un grand nombre de fonctions.

I.11.1. Fonction coût quadratique

On entend par la fonction coût, la distance calculée entre les sorties observées et les sorties évaluées par la fonction d'activation qui dépend d'un vecteur paramètreθ. Ce paramètre est estimé on résolvent le problème suivant :

$$\begin{cases} \text{Min } J(\theta(= \frac{1}{N}\sum_{i=1}^{N}\sum_{q=1}^{N_i}(d_q^i - S_q^i)^2 \\ \text{t.q.} \\ \qquad \theta \in R^p \end{cases} \tag{I.10}$$

Sur un ensemble d'exemples E, la fonction de coût $J^E(\theta)$ est définie par la moyenne des carrées des écarts sur les N éléments de cet ensemble

$$J^E(\theta) = \frac{1}{N}\sum_{i=1,\, i\in E}^{N} J^i(\theta) \tag{I.11}$$

Cette fonction dépend du vecteur de paramètres θ et de l'ensemble E considéré. Ainsi; nous notons :

- EQMA : Écart Quadratique Moyenne sur l'exemple de l'ensemble d'apprentissage;

  EQMA=$J^A(\theta)$, A ensemble d'apprentissage
- EQMT : Écart Quadratique Moyen sur les exemples de l'ensemble test

  EQMT=$J^T(\theta)$, T ensemble test.

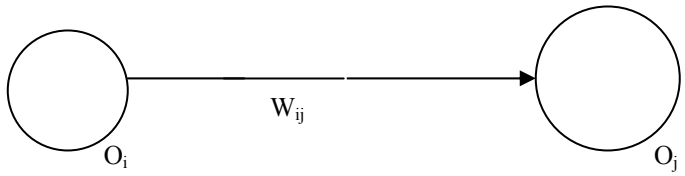
I.12. Les lois d'apprentissage

Une fois que la règle d'apprentissage est définie, l'étape suivante est l'ajustement des poids des liens entre les neurones, elle peut s'effectuer selon diverses équations mathématiques, dont la plus populaire est sans aucun doute la loi de Hebb.

Les autres équations sont souvent des dérivées de cette dernière. Par ailleurs, le choix de l'équation d'adaptation des poids dépend en grande partie de la topologie du réseau de neurones utilisé. Notons qu'il est aussi possible de faire évoluer l'architecture du réseau, soit le nombre de neurones et les interconnexions, mais avec d'autres types d'algorithmes d'apprentissage.

1- Loi de Hebb

Cette règle est très simple, elle émet l'hypothèse que lorsqu'un neurone « A » est excité par un neurone « B » de façon répétitive ou persistante, l'efficacité (ou le poids) de l'axone reliant ces deux neurones devrait alors être augmentée.



$$W_{ij}(t+1) = W_{ij}(t) + \eta O_i O_j \qquad \text{(I.12)}$$

Où : W est le poids, O_j la sortie théorique et O_i l'entrée et η un cœfficient d'apprentissage

2- Loi de Hopfield

Cette loi se base sur la même hypothèse que la loi de Hebb mais ajoute une variable supplémentaire pour contrôler le taux de variation du poids entre les neurones avec une constante d'apprentissage qui assure à la fois la vitesse de convergence et la stabilité du RN.

3- Loi Delta

Elle est aussi une version modifiée de la loi de Hebb. Les poids des liens entre les neurones sont continuellement modifiés de façon à réduire la différence (le delta) entre la sortie désirée et la valeur calculée de la sortie du neurone. Les poids sont modifiés de façon à minimiser l'erreur quadratique à la sortie du réseau de neurone. L'erreur est alors propagée des neurones de sortie vers les neurones des couches inférieures, une couche par couche, et on obtient le poids à l'instant t+1 par la formule :

$$W(t+1)=W(t)+\eta(T-O)\,E \qquad (I.13)$$

Où : W est le poids, T la sortie théorique et O la sortie réelle, E l'entrée et η un cœfficient d'apprentissage

I.13. Les différents types de réseaux de neurones

Plusieurs types de réseaux de neurones ont été développés dans des domaines d'application souvent très variés. Notamment trois types de réseaux sont bien connus:

- Le réseau de Hopfield est un réseau avec des sorties binaires, tous les neurones sont interconnectés avec des poids symétriques, c'est à dire que le poids du neurone Ni au neurone Nj est égal au poids du neurone Nj au neurone Ni. Les poids sont donnés par l'utilisateur [28].
- La machine de Boltzamann n'est autre q'un réseau de Hopfield, mais qui permet l'apprentissage grâce à la minimisation de cette énergie [30].
- Les cartes auto organisatrices de Kohonen sont utilisées pour faire des classifications automatiques des vecteurs d'entrée. Une application typique pour ce type de réseau de neurones est la reconnaissance de parole et texte [30].
- Les réseaux multicouches de type rétro propagation sont les réseaux les plus puissants des réseaux de neurones qui utilisent l'apprentissage supervisé.
- Les réseaux de neurones à fonction à base radiale ; ce type de réseau a été introduit par (Ardy (1971)), la théorie correspondante a été développée par Powell (1985). Ces réseaux étaient à l'origine appliqués aux problèmes d'interpolation.

Chapitre II
Le Perceptron Multicouches

II.1. Introduction

L'introduction de couches intermédiaires dans le réseau de neurone permet de résoudre des problèmes plus complexes que la simple séparation linéaire. Lorsqu'il existe au moins une couche cachée, les états internes du réseau ne peuvent plus être donnés directement par les exemples et les sorties désirées puisque les sorties des neurones appartenant aux couches intermédiaires sont inconnues.

La figure (II.1) représente un réseau de neurones multicouches avec comme entrée le vecteur de primitives et en sortie les classes où seront classées les formes.

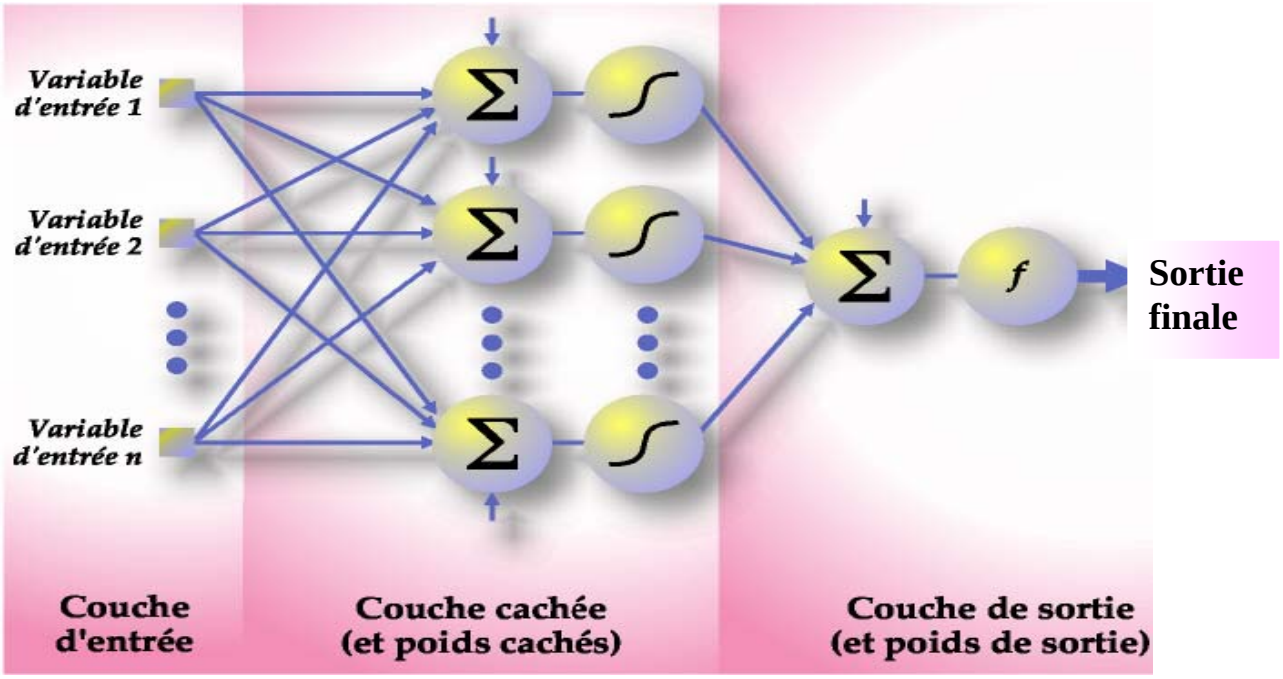


Figure II.1: Réseau de neurones Multicouches

Tel que :

$$S_i = \sum_{j=1}^{n} W_{ij} Z_j \qquad \text{(II.1)}$$

Et

$$Y_i = f(S_i) = \frac{1}{1 + e^{-S_i}} \qquad \text{(II.2)}$$

Avec :

Z_j : Valeur de l'entrée 'j' de la couche en question. Cette valeur peut être originaire d'une entrée du réseau ou de la sortie d'une autre couche.

W_{ij} : Poids associé à l'entrée 'j' de la couche 'i'.

n : Nombre d'entrées de la couche en question.

S_i : Valeur cumulée des entrées pondérées par les poids synaptiques de la couche i

Y_i : Valeur obtenue à la sortie de la couche 'i' après activation.

f(S) : Fonction d'activation, en général la sigmoïde.

Plusieurs types de réseaux de neurones multicouches ont été développés. L'apprentissage dans les réseaux Multicouche est l'adapter des poids des connexions entre les neurones de sorte que le réseau donne en sortie la classe d'appartenance des formes qui lui sont proposées en entrée. Ce qui revient à minimiser l'erreur commise par le réseau sur l'ensemble de formes de la base d'apprentissage.

Ce problème de minimisation de l'erreur a été résolu par l'algorithme de rétropropagation du gradient d'erreur [6,30]. Le terme «rétropropagation» est utilisé pour décrire l'apprentissage du réseau de neurones de type multicouche utilisant la descente du gradient appliquée à la fonction de la somme des erreurs quadratiques. Bishop [6] et autre ont démontré qu'un réseau de type multicouches à une couche cachée peut estimer n'importe quelle fonction dans R^n avec une précision arbitraire.

II.1.1. Théorème d'approximation universel

Soit *f*(.) une fonction continue non constante, bornée et monotone croissante. Soit $I^P=[0,1]^p$. $C(I^P)$ l'espace des fonctions continues sur I^P.

Soit $f \in C(I^P)$ et $\varepsilon > 0$ donné, il existe alors un entier M et des constantes réelles α_i ,β_i et w_{ij} pour i=1,…,M et j=1,…,p, et une fonction F définie par :

$$F(x_1,...,x_p) = \sum_{i=1}^{M} \alpha_i f\left(\sum_{i=1}^{P} w_{ij} x_i - \theta_i \right) \qquad \text{(II.3)}$$

est une approximation de la fonction G(.) telle que :

$$|F(x_1,\ldots,x_p) - f(x_1,\ldots,x_p)| < \varepsilon \qquad \text{(II.4)}$$

Pour tout $(x_1,\ldots,x_p) \in I^P$.

Ce théorème s'applique directement au perceptron multicouche.

La fonction logistique utilisée $\frac{1}{1+e^{s_i}}$ vérifie les conditions de *f*.

L'équation II.4 ci-dessus décrit la sortie d'un réseau Multicouches.

De ce fait, pour une application donnée, la construction du réseaux multicouches nécessite un certain nombre d'essais afin d'obtenir des performances de généralisation intéressantes.

La détermination du nombre de couches cachées dans un réseau dépend du problème à résoudre, et pour les poids de toutes les connexions du réseau, l'utilisation des algorithmes d'apprentissage sont obligatoires.

II.2. Méthode du gradient

Soit *f* une fonction d'une variable réelle à valeurs réelles, suffisamment dérivable dont on recherche un minimum. La méthode du gradient construit une suite x_n qui doit en principe s'approcher du minimum. Pour cela, on part d'une valeur quelconque x_0 et l'on construit la suite récurrente par :

Pour tout $n > 0$

$$x_{n+1} = x_n + D\,x_n \quad \text{avec} \quad D\,x_n = -\,\xi\, f'(x_n)$$

Où ξ est une valeur choisie.
On a :

$$f(x_{n+1}) = f(x_n - \xi\, f'(x_n)) \approx f(x_n) - \xi\,(f'(x_n))^2$$

D'après le théorème des approximations finies si $\xi\, f'(x_n)$ est petit. On voit que, sous réserve de la correction de l'approximation, $f(x_{n+1})$ est inférieur à $f(x_n)$.

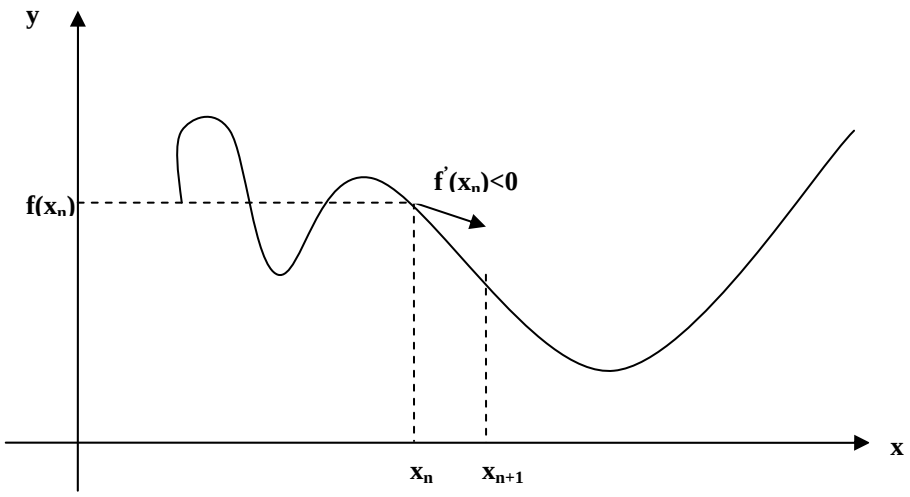


Figure II.2: La méthode du gradient

On remarque que x_{n+1} est d'autant plus éloigné de x_n que la pente de la courbe en x_n est grande. On peut décider d'arrêter l'itération lorsque cette pente est suffisamment faible. Les inconvénients bien connus de cette méthode sont :

1. le choix de ξ est empirique,
2. si ξ est trop petit, le nombre d'itérations peut être très élevé,
3. si e est trop grand, les valeurs de la suite risquent d'osciller autour du minimum sans converger,
4. rien ne garantit que le minimum trouvé soit un minimum global.

II.3. Algorithme d'apprentissage

L'objectif des algorithmes d'apprentissage est de minimiser l'erreur de décision effectuée par le réseau de neurone en ajustant les poids à chaque présentation d'un vecteur d'entraînement. Pour ce qui est de l'évaluation de la qualité d'apprentissage du réseau, l'erreur cumulée par tous les vecteurs de l'ensemble d'entraînement est évaluée. Cette erreur cumulée est calculée pour tous les cycles de la phase d'entraînement et elle est définie à partir de l'erreur quadratique. Cette mesure de l'erreur illustre la précision obtenue après *P* cycles d'apprentissage.

II.3.1. Algorithme d'apprentissage par descente de gradient

Le principe de l'algorithme est de minimiser une fonction d'erreur. Il s'agit ensuite de calculer la contribution à cette erreur de chacun des poids synaptiques. En effet, chacun des poids influe sur le neurone correspondant, mais, la modification pour ce neurone va influer sur tous les neurones des couches suivantes.

Soit un Perceptron multicouches défini par une architecture à *n* entrées et à *p* sorties, soit $\vec{W}$ le vecteur des poids synaptiques associés à tous les liens du réseau.

L'erreur du Perceptron multicouche sur un échantillon d'apprentissage *S* d'exemples $(\vec{O}^S, Z^S)$ est définie par :

$$E(\vec{W}) = \frac{1}{2} \sum_{(\vec{Z}^S, O^S) \in S} \sum_{k=1}^{p} (O_k^s - Y_k^s)^2 \qquad \text{(II.5)}$$

L'erreur mesure donc l'écart entre les sorties attendue et calculée sur l'échantillon complet. On suppose *S* fixé, le problème est donc de déterminer un vecteur $\vec{W}$ qui minimise $E(\vec{W})$. Cependant, on cherche à minimiser l'erreur sur chaque présentation individuelle d'exemple. L'erreur pour un exemple est :

$$E_{(\vec{z},\vec{o})}(\vec{W}) = \frac{1}{2} \sum_{k=1}^{p} (O_k^s - Y_k^s)^2 \qquad \text{(II.6)}$$

E est une fonction des *n* variables w_i. La méthode du gradient a été rappelée dans le cas d'une variable réelle, mais cette méthode peut être étendue au cas de fonctions de plusieurs variables réelles.

Pour mettre en oeuvre la méthode appliquée à la fonction erreur quadratique E, nous allons, tout d'abord, évaluer la dérivée partielle de E par rapport à w_i, pour tout *i*.

On a :

$$\frac{\partial \mathrm{E}}{\partial \mathrm{w_{ij}}} = \frac{\partial \mathrm{E}}{\partial \mathrm{S_i}} \frac{\partial \mathrm{S_i}}{\partial \mathrm{w_{ij}}} = \frac{\partial \mathrm{E}}{\partial \mathrm{S}} \mathrm{Z_{ij}} \tag{II.7}$$

Il suffit de calculer $\partial E / \partial S_i$, pour cela nous allons distinguer deux cas : le cas où la couche i est une couche de sortie et le cas où c'est une couche interne. si i est une couche de sortie, dans ce cas la quantité S_i ne peut influencer la sortie du réseau que par le calcule de Y_i.

Nous avons donc :

$$\frac{\partial \mathrm{E}}{\partial \mathrm{S_i}} = \frac{\partial \mathrm{E}}{\partial \mathrm{Y_i}} \frac{\partial \mathrm{Y_i}}{\partial \mathrm{S_i}} \tag{II.8}$$

On a :

$$\frac{\partial \mathrm{E}}{\partial \mathrm{Y_i}} = \frac{\partial}{\partial \mathrm{Y}} \frac{1}{2} \sum_{\mathrm{k=1}}^{\mathrm{p}} (\mathrm{O_k} - \mathrm{Y_k})^2 \tag{II.9}$$

Seul le terme correspondant à k=i a une dérivée non nulle, ce qui nous donne finalement :

$$\frac{\partial \mathrm{E}}{\partial \mathrm{Y_i}} = \frac{\partial}{\partial \mathrm{Y}} \frac{1}{2} (\mathrm{O_i} - \mathrm{Y_i})^2 = -(\mathrm{O_i} - \mathrm{Y_i}) \tag{II.10}$$

Pour le second terme de la dérivée de l'équation (II.6), on utilisant la formule de calcule de la dérivée de la fonction sigmoïde donnée en (I.5), nous avons :

$$\frac{\partial \mathrm{Y_i}}{\partial \mathrm{S_i}} = \mathrm{Y_i}(1 - \mathrm{Y_i}) \tag{II.11}$$

En substituant les résultats obtenus par les équation (II.10) et (II.11) dans l'équation (II.8), nous obtenons :

$$\frac{\partial \mathrm{E}}{\partial \mathrm{S_i}} = -(\mathrm{O_i} - \mathrm{Y_i})\mathrm{Y_i}(1 - \mathrm{Y_i}) \tag{II.12}$$

Si i est une couche caché, dans ce cas S_i va influencer le réseau par tout les calcule des neurones de la couche de sortie Nous avons alors :

$$\frac{\partial E}{\partial S_i} = \sum_k \frac{\partial E}{\partial S_k}\frac{\partial S_k}{\partial S_i} = \sum_k \frac{\partial \mathrm{E}}{\partial \mathrm{S_k}}\frac{\partial \mathrm{S_k}}{\partial \mathrm{Y_i}}\frac{\partial \mathrm{Y_k}}{\partial \mathrm{S_i}} = \sum_k \frac{\partial \mathrm{E}}{\partial \mathrm{S_k}} \times \mathrm{w_{ij}} \times \mathrm{Y_i}(1-\mathrm{Y_i})$$

On a donc :

$$\frac{\partial \mathrm{E}}{\partial \mathrm{S_i}} = \mathrm{Y_i}(1-\mathrm{Y_i})\sum_k \frac{\partial \mathrm{E}}{\partial \mathrm{S_k}} \times \mathrm{w_{ki}} \qquad \text{(II.13)}$$

Enfin, pour en déduire la modification à effectuer sue les poids synaptiques, il nous reste simplement à rappeler qu la méthode du gradient nous indique que :

$$\Delta \mathrm{w_{ij}} = -\varepsilon \frac{\partial \mathrm{E}(\vec{\mathrm{W}})}{\partial \mathrm{w_{ij}}} \qquad \text{(II.14)}$$

II.4.Critère d'arrêt

Il existe plusieurs critères d'arrêts qui peuvent être combinés entre eux :

- Le premier critère est basé sur l'amplitude du gradient de la fonction d'activation, puisque par définition le gradient sera à zéro au minimum. L'apprentissage du réseau de type Multicouches utilise la technique de recherche du gradient pour déterminer les poids du réseau.

- Le second critère d'arrêt est de fixer un seuil que l'erreur quadratique ne doit pas dépasser. Toutefois ceci exige une connaissance préalable de la valeur minimale de l'erreur qui n'est pas toujours disponible.

Ces critères sont sensibles aux choix des paramètres (par exemple : le nombre de neurones dans la couche cachée, le seuil d'erreur, …etc), si le choix n'est pas bon alors les résultats obtenus seront mauvais ou le temps de calcul de la performance du système de reconnaissance sera plus lent.

II.5. L'algorithme de rétropropagation

Pour écrire l'algorithme, nous allons simplifier quelques notations. Nous appelons δ_i la quantité $-\partial E/\partial S_i$. En utilisant les équations (II12), (II.13) et (II.14) nous obtenons les formules suivantes

Pour une cellule i de sortie, nous avons :

$$\delta_i = (O_i - Y_i) Y_i (1 - Y_i) \qquad \text{(II.15)}$$

Pour une cellule i de la couche cachée, nous avons :

$$\delta_i = Y_i (1 - Y_i) \sum_k \delta_k \times w_{ki} \qquad \text{(II.16)}$$

La modification du poids w_{ij} est alors définie par :

$$\Delta w_{ij} = -\varepsilon Z_{ij} \delta_i \qquad \text{(II.17)}$$

II.5.1. le déroulement de l'algorithme de rétro propagation du gradient

Entrée : un échantillon S; ε

Un Perceptron multicouches avec une couche d'entrée C_0, q-1 couches cachées C_1, ..., C_{q-1}, une couche de sortie C_q, n cellules.

Initialisation aléatoire des poids w_i pour i entre 1 et n.

Répéter

Prendre un exemple $(\overrightarrow{O}^{S}, Z^{S})$ de S et calculer

$$S_i = \sum_{j=1}^{n} W_{ij} Z_j$$

$$Y_{ii} = f(S_i) = \frac{1}{1+e^{-S_{ii}}}$$

Calcul des δ_i par rétropropagation

Pour toute cellule de sortie i

$$\delta_i \leftarrow (O_i - Y_i) Y_i (1 - Y_i)$$

Fin Pour

Pour chaque couche de q-1 à 1

Pour chaque cellule i de la couche courante

$$\delta_i = Y_i (1 - Y_i) \sum_k \delta_k \times w_{ki}$$

Fin Pour

Fin Pour

Mise à jour des poids

Pour tout poids

$$w_{ij} \leftarrow w_{ij} + \varepsilon\, \delta_i\, x_{ij}$$

Fin Pour

Fin Répéter

Sortie : Un PMC défini par la structure initiale choisie et les w_{ij} modifies.

Chapitre III
Méthode de Discrimination Basée Sur La Construction D'un Arbre de Décision Binaire

Les travaux de [Morgan et Sonquist 1963], [Sonquist et Morgan, 1964] ont été le point de départ pour les développements des techniques de segmentation ou de discrimination par arbre. Mais les premiers développements dans ce domaine on été lancer par [Hunt, Marin et Stone, 1966] [Messenger et Mandell, 1972], les modifications introduites par Quinlan en 1979, 1983 et 1986 et l'approche de [Breiman et al. 84] ont beaucoup contribué à la grande popularité de cette technique, et celle des arbres de décision. En fait, les arbres de décisions n'ont pas surgi par hasard ou d'une façon abstraire, ils sont apparus pour résoudre des problèmes complexes, qui ne pouvaient pas être résolus par les autres méthodes de discrimination existantes.

III.1 Théorie des graphes

Introduction

L'histoire de la théorie des graphes débute peut-être avec les travaux d'Euler au XVIII^{e} siècle et trouve son origine dans l'étude de certains problèmes, tels que celui des ponts de Königsberg. La théorie des graphes s'est alors développée dans diverses disciplines telles que la chimie, la biologie, les sciences sociales. Depuis le début du XX^{e} siècle, elle constitue une branche à part entière des mathématiques, grâce aux travaux de König, Menger, Cayley puis de Berge et d'Erdös. De manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations entre ses éléments : réseau de communication, réseaux routiers, interaction de diverses espèces animales, circuits électriques.

Les graphes constituent donc une méthode de pensée qui permet de modéliser une grande variété de problèmes en se ramenant à l'étude de sommets et d'arcs. Les derniers travaux en théorie des graphes sont souvent effectués par des informaticiens, du fait de l'importance qu'y revêt l'aspect algorithmique.

III.1.1 Généralité sur les graphes [11]

Soit G=[X, U] un graphe dont l'ensemble des sommets est X et U l'ensemble de ses arêtes.

Définition : Une chaîne de longueur q est une séquence de q arêtes.

$$L= \{u_1,u_2,\ldots,u_q\}.$$

Telle que chaque arcs u_r de la séquence ($2\leq r\leq q-1$) ait une extrémité commune avec l'arc $u_{r-1}(u_{r-1}\neq u_r)$ et l'autre extrémité commune avec l'arc $u_{r+1}(u_{r+1}\neq u_r)$.

Une chaîne qui n'utilise pas deux fois la même arête est dite simple.

Définition : Un cycle est une chaîne simple.

Définition d'un graphe connexe : c'est un graphe tel que pour toute paire x, y de deux sommets distincts,il existe une chaîne reliant ces deux points.

Définition : Un arbre est un graphe connexe sans cycles.

Définition : un sommet pendant est un sommet qui n'est adjacent qua un seul sommet

Définition : dans un graphe on appelle racine un point (sommet) « a » tel que tout autre sommet du graphe puisse être atteint par un chemin issue de « a ».

Un arbre de décision est un graphe ou un diagramme représente un système de classification ou un modèle de prédiction.

Théorème

Soit G=(X, U) un graphe d'ordre $|X|=n\geq 2$, les propriétés suivantes sont équivalentes pour caractériser un arbre :

a) G est un connexe et sans cycles.
b) G est sans cycle et admet n-1 arêtes.
c) G est connexe et admet n –1 arêtes.
d) G est sans cycles et on ajoutant une arête, on crée un cycle (et un seul).
e) G est connexe, et si on supprime une arête quelconques, il n'est plus connexe.
f) Tout couple de sommets est relié par une chaîne et une seule.

III.2. Arbre de décision binaire

La segmentation couvre un ensemble de méthode permettant la construction d'un graphe d'induction. Elle explique une variable qualitative ou quantitative à l'aide de tout type de critère. Elle cherche à résoudre les problèmes de discrimination et de régression en segmentant de façon progressive l'échantillon pour obtenir un arbre de décision.

Il existe différentes méthodes de segmentation, AID (Automatique Interaction Detector) CAID, CART (Classification And Régression Trees) ; SIPINA, on va s'intéresser à la méthode de CART.

Propriété

- Les arbres permettent le traitement des cas où les variables sont nombreuses, grâce à leur sélection automatique.

- Un autre aspect très important concerne les règles de décision résultante; celles-ci sont très simples, d'utilisation d'interprétation et ont à la fois un pouvoir explicatif et décisionnel, ce qui les rend très recherchées par les praticiens. Les arbres de décision sont capables de résoudre des problèmes difficiles, même complexes.

III.2.1. construction des arbres de décision binaires

L'utilisation des arbres binaires remontre au programme 'A.I.D' (Automatic Interaction Detection) proposé par J.A. MORGAN et J.N. SONQUIST dans les années 60. Les importants développements théoriques récents sont dû à L. BREIMAN et Al. qui proposent de construire un arbre binaire sans s'imposer de règle d'arrêt de la procédure de division des nœuds.

Dans la méthode présentée, les variables explicatives peuvent être de nature quelconque, Mais toute variable quantitative doit être préalablement transformée en une variable qualitative ayant suffisamment de modalités de sorte que la perte d'information entraînée par ce codage soit négligeable

III.2.1.1 principe

Dans la construction d'un arbre, la seule connaissance a priori se réduit à une description de l'ensemble d'apprentissage. Les données sont constituées de l'observation de p variables qualitatives ou quantitatives explicatives X^j et d'une variable à expliquer y qualitative a m modalités $\{\tau_1; l = 1,..., m\}$ ou quantitative réelle observée sur un échantillon de n individus.

La construction d'un arbre de discrimination binaire consiste à déterminer une séquence de nœuds. L'idée de base pour la construction d'un arbre de décision binaire est d'effectuer la division d'un nœud, cette procédure nécessite de définir un critère permettant de sélectionner la meilleure division d'un nœud.

Dans notre travail on s'intéresse au cas de la discrimination car la variable y est nominale et répartie en K classes, la sélection d'une division doit être telle que les segments descendants soient plus purs que le nœud parent. Autrement dit, il faut que le mélange des classes soit moins important dans le segment descendant que dans le nœud parent.

- Un nœud est défini par le choix conjoint d'une variable parmi les explicatives et d'une division qui induit une partition en deux classes. Implicitement, à chaque nœud correspond donc un sous-ensemble de l'échantillon.
- Une division est elle-même définie par une valeur seuil de la variable quantitative sélectionnée ou un partage en deux groupes des modalités si la variable est qualitative.
- A la racine ou nœud initial correspond l'ensemble de l'échantillon.

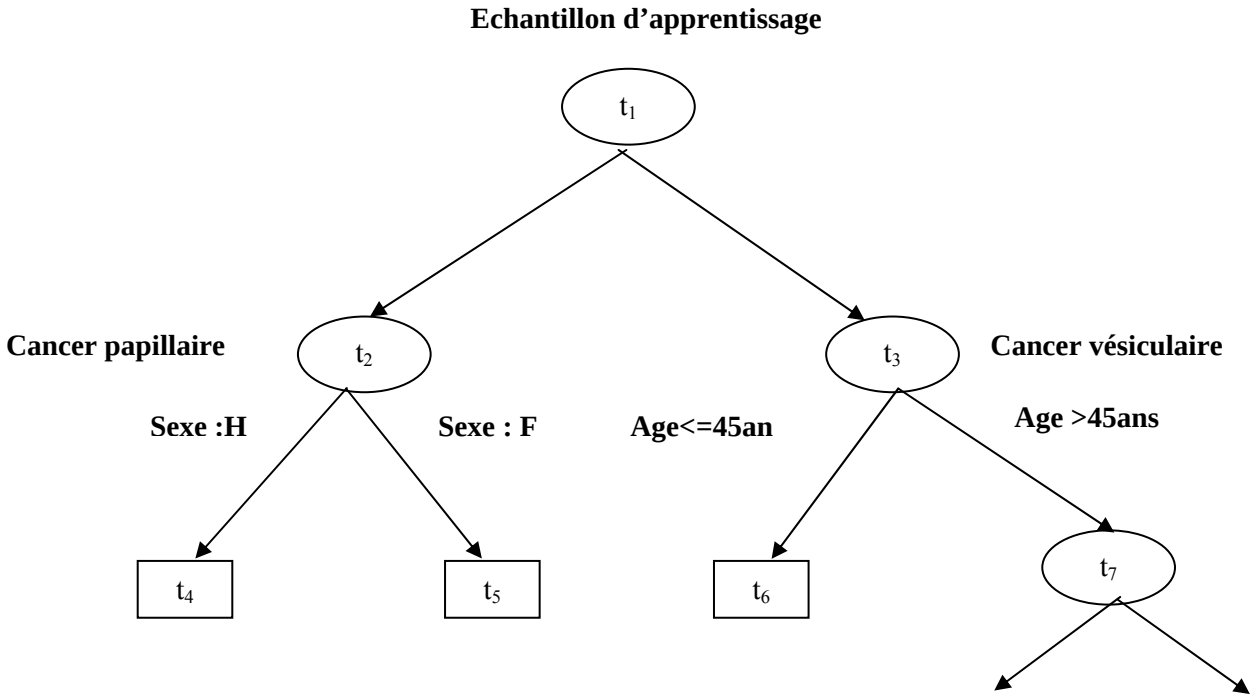


Figure III.1: arbre de décision binaire

La figure (III.1) représente un arbre de décision binaire illustratif où l'on distingue deux types de nœuds :

- Les nœuds intermédiaires (nœuds non terminaux) entourés d'un cercle : ce sont des nœuds qui fournissent deux descendants immédiats : par exemple t_3 qui se divise en t_6, t_7 ;
- Les nœuds terminaux (feuilles de l'arbre) entourés d'un carré: ce sont les nœuds qui ne sont plus divisés.

III.2.1.2. Critère de division

Une division est dite admissible si aucun des deux nœuds descendants qui en découlent n'est vide. Si la variable explicative est qualitative ordinale avec m modalités, elle fournit (m-1) divisions binaires admissibles, si elle est seulement nominale le nombre de divisions passe à $2^{(m-1)}$-1.

Le critère de division repose sur la définition d'une fonction d'hétérogénéité ou de désordre. L'objectif étant de partager les individus en deux groupes les plus homogènes au sens de la variable à expliquer. L'hétérogénéité d'un nœud se mesure par une fonction non négative qui doit être :

1- Nulle si et seulement si, le nœud est homogène c'est-à-dire que tout les individus appartiennent à la même valeur de Y.
2- Maximale lorsque les valeurs de Y sont équiprobable ou très dispersées.

III.2.1.3. Réduction de L'impureté d'un nœud t par la division [22]

A chaque nœud t de l'arbre est associé une mesure de l'impureté i(t) qui représente le degré de mélange des groupes dans t. Dans le cas général de k groupes, i(t) a la forme suivante :

$$i(t) = 2\{\sum[P(r/t).P(s/t); r > s = 1,2,.....,k]\} \quad \text{(III.1)}$$

où P(r/t) est la proportion de sujets du groupe G_r dans le nœud t.
en utilisant le résultat :

$$1 = [\sum_r P(r/t)]^2 = \sum_r [P^2(r/t)] + 2\{\sum_{r,s} [P(r/t).P(r/t); r>s]\} \quad \text{(III.2)}$$

il vient :

$$i(t) = 1 - \sum_r [P^2(r/t); r=1,2,....,k] \quad \text{(III.3)}$$

Un nœud est dit « pur » s'il ne contient que des sujets d'un seul groupe, dans ce cas :

$$i(t)=o$$

Plus le mélange des groupes dans t est important, plus l'impureté i (t) est élevée.
Dans le cas particulier de deux groupes ($(k = 2)$:

$$i(t) = 2P(1/t).P(2/t) \quad \text{(III.4)}$$

Chaque division 'd' d'un nœud t par la variable x_j entraîne une réduction de l'impureté, l'expression est :

$$\Delta_j^m = i(t) - \left[p_g i(t_g) + p_d i(t_d)\right] \qquad \text{(III.5)}$$

Où : t_g est Le segment gauche qui contient les sujets vérifiant $\{X_j \in m_1\}$ et t_d le segment droite qui contient les sujets qui vérifient $\{X_j \in m_2\}$. p_g et p_d sont les proportions des sujets de t allant respectivement dans les descendants t_g et t_d.

Δ_j^m Représente la différence entre l'impureté du nœud parent et la moyenne pondérée des impuretés de ses nœuds descendants immédiats.

III.2.1.4.Critère de la pureté maximale [22]

P(j/t) est la proportion de la classe j dans le segment t : c'est le nombre d'individus appartenant à la classe j dans le segment t.

L'impureté i(t) d'un nœud t est une fonction non négative f de p(1/t),…,p(k/t) qui vérifie les conditions suivantes :

1- f est maximale pour p(r/t), (r=1,…,k) : l'impureté d'un nœud est maximale quand pour ce nœud les probabilités d'appartenance aux différents groupes sont égales entre elles.

2- f est nulle pour {p(r/a)=1 et p(s/a)=0 pour r≠s, (r=1,…,k ;s=1,..,k) ; l'impureté est nulle dés que le nœud t ne contient que des observations d'un seul groupe.

3- f est une fonction symétrique des probabilités p(r/a), (r=1,…,k).

Propriétés :

Si la fonction f définissant l'impureté est strictement concave, alors toute division d'un nœud conduit à une réduction positive ou nulle de l'impureté.

$$\Delta_j^m \geq 0 . \qquad \text{(III.6)}$$

La nullité de Δ_j^m étant obtenue si et seulement si l'égalité $p(r/t_g)=p(r/t_d)=p(r/t)$ est vraie pour tout r variant de 1 à k.

$$\Delta_j^m = 0 \Leftrightarrow p(r/t_g)=p(r/t_d)=p(r/t) \qquad \forall r. \tag{III.7}$$

En effet, la stricte concavité de la fonction f implique :

$P_g I(t_g)+p_d I(t_d)=p_g f(p(1/t_g),\ldots,p(k/ t_g)+ p_d f(p(1/t_d),\ldots,p(k/ t_d))$

$\Leftarrow f (p_g(p(1/t_g)+p_d(1/t_d),\ldots,P_g p(k/ t_g)+p_d(p(k/ t_d)).$

L'égalité se produisant si et seulement si :

$$p_d(p(r/ t_d)= p_g(p(r/ t_g)$$

pour tout r variant de 1 à k, La relation est vraie.

Par conséquent pour chaque variable x_j, la meilleure division d_j^* est telle que la réduction de l'impureté Δ_j^* est maximale :

$$\Delta_j^* = \max_{m \in d_j} \{\Delta_j^m\} \tag{III.8}$$

où d_j est l'ensemble des p variables, la division du nœud t est effectuée à l'aide de la variable qui assure :

$$\Delta^* = \max_{j=1,\ldots,p} \{\Delta_j^*\} \tag{III.9}$$

Choisir la division Δ^* sur le nœud t qui maximise la réduction de l'impureté de l'arbre est équivalent à choisir la division qui donne le maximale de l'impureté du nœud t.

III.2.1.5. critère d'arrêt

La croissance de l'arbre s'arrête à un nœud donné, qui devient terminal ou feuille, lorsqu'il est homogène c'est-à-dire lorsqu'il n'existe plus de partition admissible ou, pour éviter un découpage inutilement fin, si le nombre d'observations qu'il contient est très faible (généralement inférieur à une valeur seuil comprise en général entre 1et 5).

III.2.2. Algorithme général de la segmentation

Les étapes de l'algorithme sont les suivantes :

1. au départ, on dispose d'un seul segment contenant l'ensemble des individus.

2. A la première étape, la procédure de construction de l'arbre examine une par une toutes les variables explicatives.

Chaque division scinde l'échantillon en segments descendants :

Le segment gauche t_g contient les sujets vérifiants $\{X_j \in m_1\}$ et le segment droite contient les sujets qui vérifient $\{X_j \in m_2\}$.
La procédure sélectionne la meilleure division parmi toutes les variables explicatives, au sens d'un critère de division adopté.
Ainsi pour chaque variable on obtient la meilleure division qui fournit les deux segments les plus purs vis à vis de y.

3. A l'étape suivante, on applique la même procédure à chacun des deux segments descendants obtenus.

4. La procédure s'arrête lorsque tous les segments sont déclarés terminaux, soit parce qu'ils ne nécessitent plus de division soit parce que leur taille est inférieure à un effectif fixé au préalable.

5. Pour un nouvel individu, on définit une règle d'affectation simple.

III.3. Méthode CART

La méthode CART (Classification And Régression Trees) est proposée par Breiman et Al (1984), elle est le produit de dix années de travaux et de recherche, ce qui lui assure des performances stables, elle présente des avantages importants dont le premier est la stabilité des règles d'affectation, l'interprétation des résultats étant directe et intuitive.

Par ailleurs la méthode CART, contrairement aux autres méthodes de segmentation, n'impose aucune règle d'arrêt de division de segment. Elle fournie à partir de l'arbre binaire complet la séquence des sous arbres obtenues en utilisant une procédure d'élagage.

III.3.1. Principe général de la méthode CART

La méthode de discrimination par arbre binaire CART, proposée par Breiman, Friedman, Olshen et Stone (1984), inclut des solutions qui répondent aux critiques les plus importants faîtes aux arbres de décision.

La construction d'un arbre de décision binaire s'effectue d'une façon récursive à travers une division successive de l'ensemble d'états qui correspond à la racine de l'arbre en sous-ensembles correspondants aux nœuds descendants.

III.3.2. Les critères de segmentations

L'idée fondamentale dans la construction d'un arbre de décision binaire par la méthode CART est la sélection dans chaque nœud t, d'un attribut binaire de façon à ce que les nœuds descendants t_g et t_d soient purs que le nœud parent t.

La méthode CART comporte divers critères de choix de l'attribut binaire : Gini, Symgini, Twoing, les moindres carrés, Shannon, critère de χ^2,…ect.

III.3.2.1 indice de Gini

Le coefficient de Gini consiste à choisir la variable binaire W qui coupe t en t_g,t_d qui maximise la diminution de l'impureté, donnée par :

$$f(p_1^t, p_2^t, ..., p_K^t) = \sum_{1 \le i \ne j \le K} p_i^t p_j^t = 1 - \sum_{1 \le i \le K} (p_i^t)^2 \qquad \text{(III.10)}$$

Ce coefficient se met sous la forme :

$$G(t, W) = f(t) - p_d^t f(t_d) - p_g^t f(t_g) \quad \text{(III.11)}$$

$$G(t, W) = p_d^t \sum_{j=1}^{K} (p_j^{t_d})^2 + p_g^t \sum_{j=1}^{K} (p_j^{t_g})^2 - \sum_{j=1}^{K} (p_j^t)^2 \quad \text{(III.12)}$$

Telle que :

$$p_g^t = \frac{n_g}{n}, \qquad p_j^{t_g} = \frac{n_{ig}}{n_g},$$

$$p_d^t = \frac{n_d}{n}, \qquad p_j^{t_d} = \frac{n_{id}}{n_g},$$

$$G(t_g, t_d) = \frac{n_g}{n} \sum_{i=1}^{m} \frac{n_{ig}}{n_g} \left(1 - \frac{n_{ig}}{n_g}\right) + \frac{n_d}{n} \sum_{i=1}^{m} \frac{n_{id}}{n_d} \left(1 - \frac{n_{id}}{n_d}\right) \quad \text{(III.13)}$$

Il est équivalent à la bipartition qui maximise la variation d'impureté ou le gain informationnel $I_G(t_g,t_d)$, dont l'expression est donnée ci-dessus :

$$G(t, W) = \sum_{i=1}^{m} \frac{n_{ig} + n_{id}}{n} \left(1 - \frac{n_{ig} + n_{id}}{n}\right) - G(t_g, t_d) \quad \text{(III.14)}$$

Ou encore :

$$G(t, W) = \frac{n_g n_d}{nn} \sum_{i=1}^{m} \left(\frac{n_{ig}}{n_g} - \frac{n_{id}}{n_d}\right)^2 \quad \text{(III.15)}$$

L'impureté est maximale quand toutes les classes sont mélangées dans le nœud et minimale quand le nœud ne contient qu'une classe.

III.3.2.2. Indice de Twoing

Ce critère qui apparaît dans [Breiman et all, 1984], correspond à un des deux coefficients adoptés par ces auteurs dans le cas de plusieurs classes à prédire. Il permet de réduire la complexité en contraignant l'espace de recherche des attributs binaires. Considérons alors une variable prédictive ayant L modalités.

- La première étape consiste à binariser cette variable sur la base de l'indice de Gini, en l'optimisant par rapport à toutes les bipartitions de l'ensemble des k classes à prédire.
- La deuxième étape consiste à considérer toutes les variables binaires issues de la variable prédictive. Ensuite, on évalue chacune d'entre elles par rapport à la variable binaire à prédire w qui lui est le mieux associée.

Les auteurs précités démontrent que cette bipartition doit être choisie comme suit :

$$c_1(W) = \left\{ j : p_j^{t_g} \geq p_j^{t_d} \right\} \qquad \text{et} \qquad c_2(W) = \left\{ j : p_j^{t_g} < p_j^{t_d} \right\} \tag{III.16}$$

Le critère optimisé prend la forme dite « Twoing » [ref]

$$T(t, w) = \frac{p_g^t p_d^t}{4} \left[\sum_{j=1}^{k} \left| p_j^{t_g} p_j^{t_d} \right| \right]^2 \tag{III.17}$$

et :

$$p_g^t = \frac{n_g}{n}, \qquad p_g^{t_g} = \frac{n_{i_g}}{n_g}$$

$$p_d^t = \frac{n_d}{n}, \qquad p_d^{t_d} = \frac{n_{i_d}}{n_d}$$

le terme entre parenthèse représente la distance entre p_j^{tg} p_j^{td}.

Le critère de Gini et Twoing sont ainsi équivalents pour un nombre K de classes à prédire égal à 2. Le critère de Twoing est utilisé lorsque le nombre de classes K est supérieur à deux.

Définitions

1- On définit le taux d'erreur sur le sommet t par la quantité ε(t) comme suit :

$$\varepsilon(t) = 1 - \max_{r=1}^{k} p(r/t) \tag{III.18}$$

2- Le coût d'une erreur de classement est une fonction non symétrique sur les classes, définit comme suit :

$$\gamma : \{1,2,\ldots,k\}^2 \rightarrow R_+$$

$$(r,s) \in \{1,2,\ldots,k\}^2 \rightarrow \gamma(r,s) = \gamma_{rs} = \begin{cases} 0 & \text{si } r = s \\ \succ 0 & \text{si } r \neq s \end{cases}$$

γ_{rs} est le coût de l'erreur consistant à classer un élément de la classe r dans la classe s.

Soit un sommet t auquel est associée une distribution de classes p(r/t) pour r=1,…,k, on définit alors le coût moyen d'un classement dans la classe s sur le sommet t :

$$\overline{\gamma_s(t)} = \sum_{r=1}^{k} p(r/t)\gamma_{rs} \qquad \text{(III.19)}$$

3- On définit le coût de mauvais classement associé aux erreurs sur un sommet t :

$$\xi(t) = \min_{s=1}^{k} \overline{\gamma_s(t)} \qquad \text{(III.20)}$$

4- On définit le coût de mauvais classement associé aux erreurs sur une partition $t=(t_1,\ldots,t_j,\ldots,t_k)$ par la quantité $\xi(t)$, comme suit :

$$\xi(t) = \sum_{j=1}^{k} \xi(t_j)p(t_j) \qquad \text{(III.21)}$$

si $\gamma_s(t)=1$ pour tout les couples de classe (r,s) $r \neq s$ alors :

$$\overline{\gamma_s(t)} = \sum_{r=1}^{k} p(r/t)\gamma_{rs} = 1\text{-}p(s/t) \qquad \text{(III.22)}$$

d'où :

$$\xi(t) = \min_{s=1}^{k} \overline{\gamma_s(t)} \qquad \text{(III.23)}$$

$$\xi(t) = 1 - \max_{r=1}^{k}(p(r/t)) \qquad \text{(III.24)}$$

ce qui donne :

$$\xi(t)=\varepsilon(t) \qquad \text{(III.25)}$$

III.3.3. procédure de sélection

Il est nécessaire de diviser l'échantillon de base en deux parties, l'échantillon d'apprentissage constitué par 2/3 de l'échantillon de base et l'échantillon de test le tiers restant.

La recherche du meilleur sous arbre G^* se fait de la façon suivante :

A partir de l'échantillon d'apprentissage, on construit l'arbre complet G_{max} ou chaque segment terminal contient un nombre restreint d'individus.
Puis l'opération d'élagage de l'arbre G_{max} consiste à construire une séquence optimale de sous arbres emboîtés $\{G_H,\ldots,G_h,\ldots,G_1\}$, où G_H coïncide avec G_{max}, ,G_h est le sous arbre ayant h segment terminaux et G_1est l'échantillon total.

Chaque sous arbre G_h de cette séquence est optimal au sens suivant : son erreur apparente est minimale parmi les sous arbres ayant le même nombre de segment terminaux, si S_h est l'ensemble des sous arbres de G_{max} ayant h segments terminaux alors :

$$\xi(G_h) = \min_{G \in S_h} \xi(G) \qquad \text{(III.26)}$$

A partir de l'échantillon test, on sélectionne parmi les sous arbres de la séquence optimale, le meilleur sous arbre G^*, qui présente la plus petite erreur théorique :

$$\xi(G^*) = \min_{1 \le h \le H} \xi(G_h) \qquad \text{(III.27)}$$

Les individus de l'échantillon test parcourent chacun des sous arbres de la séquence optimale et atterrissent dans un segment terminal, ce qui entraîne une estimation de l'erreur théorique pour le sous arbre.

III.3.4. L'élagage de l'arbre

Définition

L'élagage est l'étape qui consiste a supprimer les parties de l'arbre qui ne semblent pas performantes pour prédire la classe de nouveaux cas et remplacées la par un noeud terminal (associé à la classe majoritaire)

III.3.4.1. Principe général

De façon générale, considérons le coût de mauvais classement $\xi(t)$, qui correspond usuellement au taux d'erreur moyen calculé sur l'échantillon d'apprentissage.
Soit G_{max} l'arbre le plus fin que nous puissions obtenir par l'algorithme de construction précédent. On notera S_{max} la partition terminale associée à G_{max} et ξ (S_{max}) le taux d'erreur correspondant.

$\xi(S_{max})$étant une estimation optimiste du taux d'erreur théorique ξ^* inconnu, généralement $\xi(S_{max}) < \xi^*$.

Le principe de l'élagage par CART repose sur un critère « coût complexité », qui vise à pénaliser la prolifération de sommets.

III.3.4.2. Coût de complexité d'un arbre

Considérons un arbre binaire G_{max} dont la partition terminale est notée $t=\{t_1,\ldots,t_k\}$.
Soit $\alpha \geq 0$; le coût de complexité de l'arbre G_{max} est définit par :

$$C_\alpha(G_{max}) = \xi(t) + \alpha \, card(t) \qquad \text{(III.28)}$$

Or card (t)=K,

Et

$\xi(t) = \sum_{j=1}^{k} \xi(t_j)$ le taux d'erreur sur chaque sommet terminal.

Par conséquent :

$$C_{\alpha}(G_{max}) = \sum_{j=1}^{k}[\xi(t_j) + \alpha] = \sum_{j=1}^{k}[C_{\alpha}] \quad \text{(III.29)}$$

Soit t un sommet non terminal de l'arbre G, le sous arbre descendant de t noté $G(t^{+})$, est le sous arbre de G dont la racine est s.

III.3.5.Algorithme d'élagage

Considérons l'arbre de taille maximum G_{max} , on notera alors S l'ensemble des sommets, S_{ter} l'ensemble des sommets terminaux, S_{int} l'ensemble des sommets intermédiaire.

$$S=S_{ter} \cup S_{int}$$

L'algorithme de l'élagage se déroule en deux étapes :

La première consiste à construire une liste ordonnée de sous arbres imbriqués, allant de l'arbre maximum jusqu'au sous arbre G_R qui ne contient que la racine :

$$G_{max}<G_1<G_2<\ldots.<G_t<\ldots<G_R$$

Cet ordre résulte du critère « coût de complexité ».

La seconde étape consiste à déterminer, dans cette liste le sous arbre optimal au sens d'un second critère qui est le taux d'erreur en validation. Le sou arbre G fournit le taux minimum d'erreur en validation. L'élagage s'effectue par le bas en remontant jusqu'à la racine.

III.3.5.1. Construction de la liste des sous arbres

Elle s'effectue en deux phases :

La première permet d'élaguer les sous arbres qui ont deux sommet terminaux et dont la suppression ne provoque aucune augmentation du critère de segmentation.

La seconde est répétitive, elle consiste à évaluer « le coût de complexité » de chacun des sous arbres restant afin de déterminer celui qu'il supprimer. Nous allons décrire de manière plus explicite de ces deux phases :

Phase initiale : soit S_{ter}, l'ensemble des sommets terminaux de la partition, et soient t_{ig} et t_{id} deux d'entre eux, ils ont le même sommet père t_i.

On sait que la valeur du critère de segmentation est toujours positive ou nulle $\Delta i(d,t) \geq 0$.

Si elle est nulle, cela signifie que la segmentation de t_i en deux segments n'apporte aucune information sur les classes à discriminer.

Plus généralement, considérons que nous somme à l'itération V, dans laquelle la partition terminale est S_{ter}^{v} est soit t_u un sommet intermédiaire dont les fils sont terminaux : $t_{u,g}$,$t_{u,d}$.

Si $\Delta i(d,t) = 0$ alors on peut supprimer de la partition terminale les deux sommets ainsi t_u deviendras un sommet terminal.

Phase courante : à l'issue de la phase initiale nous obtenons une partition terminale que nous noterons S_{ter}^{1} le tout, formant un premier sous arbre binaire noté G_0, il s'agit maintenant de passer à un nouveau sous arbre binaire G_1. Pour le déterminer nous utiliserons une procédure différente.

Le coût de complexité d'un arbre binaire est une combinaison linéaire entre le taux d'erreur de la partition et le nombre de sommets qui la composent. L'objectif est de déterminer un sous arbre G_1 qui a le plus faible « coût complexité » :

$$C\alpha(G)=\xi(S_{ter})+\alpha card(S_{ter}) \quad \text{pour } \alpha \in [0,1]$$

Pour chaque valeur de α un sous arbre qui a le plus faible « coût de complexité »sera déterminer.

Plaçons nous dans le cas du passage de G_0 à G_1. Considérons un sommet $t \in S_{ter}^{1}$ non terminal de G_0 nous calculerons alors deux quantités :

1. le coût de complexité en t est $C\alpha(t)=\xi(t)+\alpha$ qui n'est rien d'autre que le taux d'erreur au sommet t :

$$\xi(t) = 1 - \max_{r=1}^{k} p(r/t) \qquad \text{(III.30)}$$

2. le coût de complexité du sous arbre $G(t^+)$qui a pour racine t :

$$C\alpha(G)=\xi(S_{t+})+\alpha\ card(S_{t+}) \quad \text{(III.31)}$$

Où S_{t+} est la partition terminale associée au sous arbre $G(t^+)$

Il est évident que si :

$$C\alpha(t)< C\alpha(G(t^+)).$$

Alors, il vaut mieux d'élaguer le sous arbre $G(t^+)$, mais tant que :

$$C\alpha(t)>C\alpha(G(t^+)).$$

Il est préférable de garder $G(t^+)$

En faisant varier α par valeur croissante, il est clair que l'inégalité $C\alpha(t)< C\alpha(G(t^+))$va à une valeur précise de α, elle sera: $C\alpha(t)>C\alpha(G(t^+))$.

Ainsi nous obtenons :

$$\alpha=\frac{\xi(t)-\xi(S_{t^+})}{card(S_{(t^+)})-1} \quad \text{(III.32)}$$

Pour tout $t \in S^1_{\text{int}}$ nous déterminons $\alpha(t)$.

Puisque nous cherchons la seconde valeur de α qui nous permet d'avoir le sous arbre G_1, il faut alors prendre :

$$\alpha_1=\min_{t\in S^1_{int}}\alpha_i \quad \text{(III.33)}$$

Soit $t \in S^1_{int}$ tel que $\alpha(t)=\alpha_1$, le sous arbre G_1est obtenu en retirerons à G_0 le sous arbre $G(t^+)$.

Dans le cas où il y a plusieurs sommet t_k k={1,…,k} qui vérifient $\alpha(t_k)=\alpha_1$, nous retirons tous les sous arbres $G(t^+)$ dont ils sont racine. Nous recommençons le processus en cherchant cette fois-ci, G_2 à partir de G_1 et ainsi de suite, jusqu'à la racine.

Nous obtenons ainsi une liste de couples (α_i, G_i) que nous ordonnons suivant les valeurs de α_i :

$$G_0 < G_1 < \ldots < G_t < \ldots < G_R.$$

III.3.5.2. Détermination du meilleur sous arbre G

Nous choisirons parmi les sous arbres de la liste ordonnée le meilleur selon un critère. La procédure la plus simple consiste à disposer d'un échantillon test Ω_t , et évaluer le taux d'erreur lors du classement des individus de cet échantillon à partir de chaque sous arbre. Soit $\xi_t(G_i)$, le taux d'erreur sur l'échantillon de test Ω_t lorsqu'on applique le modèle de décision donnée par G_i.le meilleur sous arbre G* est tel que :

$$\xi_t(G_i) = \min_{i=0}^{R} \xi_t(G_i) \qquad \text{(III.34)}$$

III.3.6. Spécialisation de l'arbre

La spécialisation de l'arbre consiste a étiqueter chacun de ses sommets par l'une des classes $c_1, c_2, \ldots, c_m$. La stratégie la plus couramment utilisée revient à affecter un sommet à la classe majoritaire. Si on désigne par n_{is}/n_s la proportion d'individus de l'échantillon d'apprentissage qui appartiennent à la classe c_i du sommets s, celui-ci sera alors associé à la classe c_k, si :

$$\frac{n_{ks}}{n_s} = \max_{i=1}^{m}\left(\frac{n_{is}}{n_s}\right)$$

Cette spécialisation permet d'affecter un nouvel individu, n'ayant pas servi l'ors de la phase d'apprentissage, à l'un des classes $C_1, C_2, \ldots, C_m$.

Chapitre IV
Implémentation des méthodes

IV.1. Traitement de la structure de données

La structure de données utilisées pour construire l'arbre de décision est la structure des arbres (ici les arbres m-aire) et la récursivité (la construction de l'arbre se fait de manière récursive)

Un nœud a la structure suivante :

```
Nœud = Record
INFO : typeinfo ;
FG, FD, PERE : ^Nœud ;    // ^ : un pointeur vers noeud
END;

Typeinfo =Record
Variable: string;
Entropy: extended;
Gain: extended;
END;
```

Chaque nœud contient l'information (INFO) et trois adresses pour pointer le fils gauche (FG), le fils droit (FD) et le père (PERE).

INFO : est un enregistrement contient :

Variable : est une chaîne de caractères contient le nom par exemple : IODE, AGE,…

Entropie : valeur numérique pour stocker l'entropie calculée de la variable de nœud.

Gain : valeur numérique pour stocker le gain de la variable de nœud.

IV.2. Description de fonctionnement du programme

IV.2.1. Chargement des données

Pour charger les données, le programme se connecte automatiquement à une base de données (MS ACCESS) ou (PARADOX7) où le nom de la table est «datarbre».
Le programme parcourt toute la table est charge les données dans une grille pour permettre le traitement et le calcul des entropies.

IV.2.2. Calcul des entropies

On effectue le calcul de l'entropie de la variable décision Y, puis l'entropie de chaque variable Xi par rapport à la variable Y et on prend la variable qui nous donne le meilleur Gain en information.

IV.2.2.1. L'entropie de Y

$$I(Y) = -\sum p(u)\log_2(p(u)) \qquad \text{(IV.1)}$$

Où u appartient à Dy

Dans le programme : le calcul se fait comme suit :

$$P^{+}(Y) = \frac{nb(1)}{N} \qquad \text{(IV.2)}$$

N est le nombre total d'individus de Y, tel que Y est la variable décision.
nb(1) : est le nombre d'individus qui appartient a la classe 1 dans la variable Y

$$P^{-}(Y) = \frac{nb(0)}{N} \qquad \text{(VI.3)}$$

nb(0) : est le nombre d'individus qui appartient a la classe 0 dans la variable Y

$$I(Y) = -(P^{+}(Y)\log_2(p^{+}(Y)) + P^{-}(Y)\log_2 p^{-}(Y)) \qquad \text{(VI.4)}$$

IV.2.2.2. L'entropie des Xi

On désigne par N le nombre d'individus total.

$$p^{+}(X) = \frac{nb(1)}{N}$$

nb(1) : le nombre d'individus qui appartient a la classe « 1 » dans la variable X_i

$$P^{-}(X) = \frac{nb(0)}{N}$$

nb(0) *:* : le nombre d'individus qui appartient a la classe « 0 » dans la variable X_i

$$P^{+}(X_1) = \frac{nb'(1)}{nb(1)}$$

$nb'(1)$: Le nombre d'individus qui appartient à la classe « 1 » de la variable Xi, qui correspond au « 1 » de la variable Y.

$$P^{-}(X_1) = \frac{nb'(0)}{nb(1)}$$

$nb'(0)$*:* Le nombre d'individus qui appartient à la classe « 0 » de la variable Xi qui correspond au « 1 » de la variable Y.

$$P^{+}(X_0) = \frac{nb'(1)}{nb(0)}$$

$nb'(1)$: Le nombre d'individus dans la classe « 1 » de la variable X_i qui correspond au 0 de la variable Y

$$p^{-}(X_0) = \frac{nb'(0)}{nb(0)}$$

$nb'(0)$: Le nombre d'individus qui appartient à la classe «0 » de la variable X_i , qui correspond au 0 de Y

On pose :

P= $P^{+}(X)$; désigne le nombre des 1 sur le nombre d'individus N de X_i

M = $P^{-}(X)$; désigne le nombre des 0 sur le nombre d'individus N de X_i

$P_1 = P^+(X_1)$; désigne le nombre des 1 sur le nombre d'individus N de X_i qui correspond à la valeur 1 de Y

$P_0 = P^+(X_0)$; désigne le nombre des 1 sur le nombre d'individus N de X_i qui correspond à la valeur 0 de Y

$M_1 = P^-(X_1)$; désigne le nombre des 0 sur le nombre d'individus N de X_i qui correspond à la valeur 1 de Y

$M_0 = P^-(X_0)$; désigne le nombre des 0 sur le nombre d'individus N de X_i qui correspond à la valeur 0 de Y

$$I(X_i(1)) = -[P_1 * \log_2(P_1) + M_1 * \log_2(M_1)] \quad \text{(IV.5)}$$

$$I(X_i(0)) = -[P_0 * \log_2(P_0) + M_0 * \log_2(M_0)] \quad \text{(IV.6)}$$

Pour trouver la valeur de l'entropie de X_i on calcul la quantité suivante

$$I(X_i) = P * I(X_i(1)) + M * I(X_i(0)) \quad \text{(IV.7)}$$

Par conséquent pour choisir la variable qui possède la plus grande information mutuelle revient à trouver celle qui minimise le gain d'entropie

$$\Delta_{X_i}^{Y} = I(Y) - I(X_i) \quad \text{(IV.8)}$$

IV.2.3. La construction de l'arbre

La construction de l'arbre se fait par un algorithme récursif.
Soit R la racine de l'arbre, F_G le fils gauche de et F_D le fils droit :

Algorithme :

Si $F_G(R)$ =NULL alors
Si $F_D(R)$ =NULL alors *Fin de traitement*
Sinon *parcourir ($F_D(R)$) ;*
Fin si ;
Sinon parcourir ($F_G(R)$) ;
Fin si ;

Parcours d'une branche gauche ou droite se fait comme suit :

On divise la racine en deux branches selon la variable Y (0 et 1) une branche gauche $F_G(R)$ et une branche droite $F_D(R)$

Algorithme :

SI Y contient que des '1' alors *$F_G(R)$ est une feuille a la valeur '1' ;*
SINON
SI Y contient que des '0' alors *$F_G(R)$ est une feuille a la valeur '0' ;*
SINON *// ie : (Y contient des '0' et des '1')*
Parcours ($F_G(R)$)
Parcours ($F_D(R)$)
Fin SI ;
Fin SI ;

Parcours F_G (R)

Le fils gauche $F_G(R)$) est la branche gauche correspond à la valeur '0' de père.

Algorithme :

SI (Y/Xi=0) contient que des '0' alors *F_G(R) est une feuille a la valeur '0'*
SINON
SI (Y/Xi=0) contient que des '1' alors *F_G(R) est une feuille a la valeur '1'*
SINON (algorithme récursif)

On procède au choix de la variable qui divise la branche ($F_G(R)$), pour cela on calcul les entropies de toutes les variables restantes. Après le calcul des entropies et les gains corresponds on prend la variable qui nous donne le meilleur gain en quantité d'information et on divise cette branche ($F_G(R)$) selon les valeurs d'individus de cette variable choisie.

Puis on refait le même travail pour cette nouvelle sous arbre et ainsi de suite jusqu'à ce que il ne reste aucune variable à prendre. Puis on suit le même algorithme pour le fils droit ($F_D(R)$).

Après le traitement des deux fils (branches FG et Fd), l'arbre sera complètement construit.

Fin SI ;
Fin SI ;

IV.2.4. Elagage de l'arbre

***Principe* :** Scinder le jeu de données en deux parties (3/4 et 1/4), Utiliser la première pour l'apprentissage (3/4), utiliser la deuxième en test de généralisation (1/4), Supprimer les branches ayant trop d'erreurs.

Objectif* :** Supprimer les parties de l'arbre qui ne ***semblent pas performantes pour prédire la classe de nouveaux cas et remplacées la par un noeud terminal (associé à la classe majoritaire). Processus de type "*bottom-up*" (du bas vers le haut: des extrémités vers la racine), basé sur une estimation du taux d'erreur de classification: un arbre est élagué à un certain noeud si le taux d'erreur estimé à ce noeud (en y allouant la classe majoritaire) est inférieur au taux d'erreur obtenu en considérant les sous arbres terminaux.

Élagages successifs (au départ des extrémités) jusqu'à ce que tous les sous arbres restants satisfassent la condition sur les taux d'erreur de classification.

IV.3. Exemple illustratif

La variable aléatoire w possède une entropie H(w) qui se définit par :

$$H(w) = \sum_{u \in D_w} P(w = u)\log(P(w = u)) \quad \text{(IV.9)}$$

De même, on peut définir l'entropie de w conditionnée par a comme

$$H(w \setminus a) = - \sum_{u,v \in D_w \times D_a} P(w = u, a = v)\log(P(w = u \setminus a = v)) \quad \text{(IV.10)}$$

Un résultat classique de théorie de l'information nous dit que :

$$I(w,a) = H(w) - H(w \setminus a) \quad \text{(IV.11)}$$

Dans notre cas les variables sont binaire par exemple D_X={VRAI, FAUX}

Dans l'exemple sui suit, le problème d'apprentissage consiste à trouver une règle de décision binaire à partir de huit exemples sur quatre paramètres binaires.
Le problème qui se pose à un enfant qui revient de l'école est le suivant : peut-il aller jouer chez son voisin ou pas ?
L'expérience, qu'il a acquise par punition récompense sur les huit jours d'école précédents, est résumée dans le tableau des huit exemples d'apprentissage.

	Mes devoirs sont-ils Finis?	**Maman est elle De Bonne Humeur?**	**Est-ce qu'il Fait Beau?**	**Mon Goûter est-il Pris?**	**Décision**
1	Vrai	Faux	vrai	Faux	Oui
2	Faux	vrai	Faux	Vrai	Oui
3	Vrai	vrai	vrai	Faux	Oui
4	Vrai	Faux	vrai	Vrai	Oui
5	Faux	vrai	vrai	Vrai	Non
6	Faux	vrai	Faux	Faux	Non
7	Vrai	Faux	Faux	Vrai	Non
8	vrai	vrai	Faux	Faux	Non

$H(M/DF)=\frac{5}{8}P(DF=vrai)+\frac{3}{8}P(DF=Faux)$

Avec :

$P(DF=Vrai)= -\frac{3}{5}\log_2(\frac{3}{5})-\frac{2}{5}\log_2(\frac{2}{5})\approx 0.971$

$P(DF=Faux)= -\frac{1}{3}\log_2(\frac{1}{3})-\frac{2}{3}\log_2(\frac{2}{3})\approx 0.918$

$H(M/DF)\approx 0.93$

De même pour les autres variables on trouve :

$H(M/FB)\approx 0.8$, $H(M/MBH)\approx 0.93$, $H(M/GP)\approx 1$

On choisit donc pour racine de l'arbre la variable Faut-il Beau ? Au bout de la branche gauche portant la valeur Vrai se trouve le tableau suivant des exemples corrects pour cette variable :

	Mes devoirs Sont-ils Finis ?	**Maman est-elle de Bonne Humeur ?**	**Mon Goûter est-il Pris ?**	**Décision**
1	Vrai	Faux	Faux	Oui
3	Vrai	Vrai	Faux	Oui
4	Vrai	Faux	Vrai	Oui
5	Faux	Vrai	Vrai	Non

Sous la branche droite, portant la valeur Faux se trouve le tableau suivant :

	Mes devoirs Sont-ils Finis ?	**Maman est- elle de Bonne Humeur ?**	**Mon Goûter est-il Pris ?**	**Décision**
2	Faux	Vrai	Vrai	Oui
6	Faux	Vrai	Faux	Non
7	Vrai	Faux	Vrai	Non
8	Vrai	Faux	Faux	Non

Et la procédure se refait de la même manière avec les deux tableaux, on aura l'arbre final suivante :

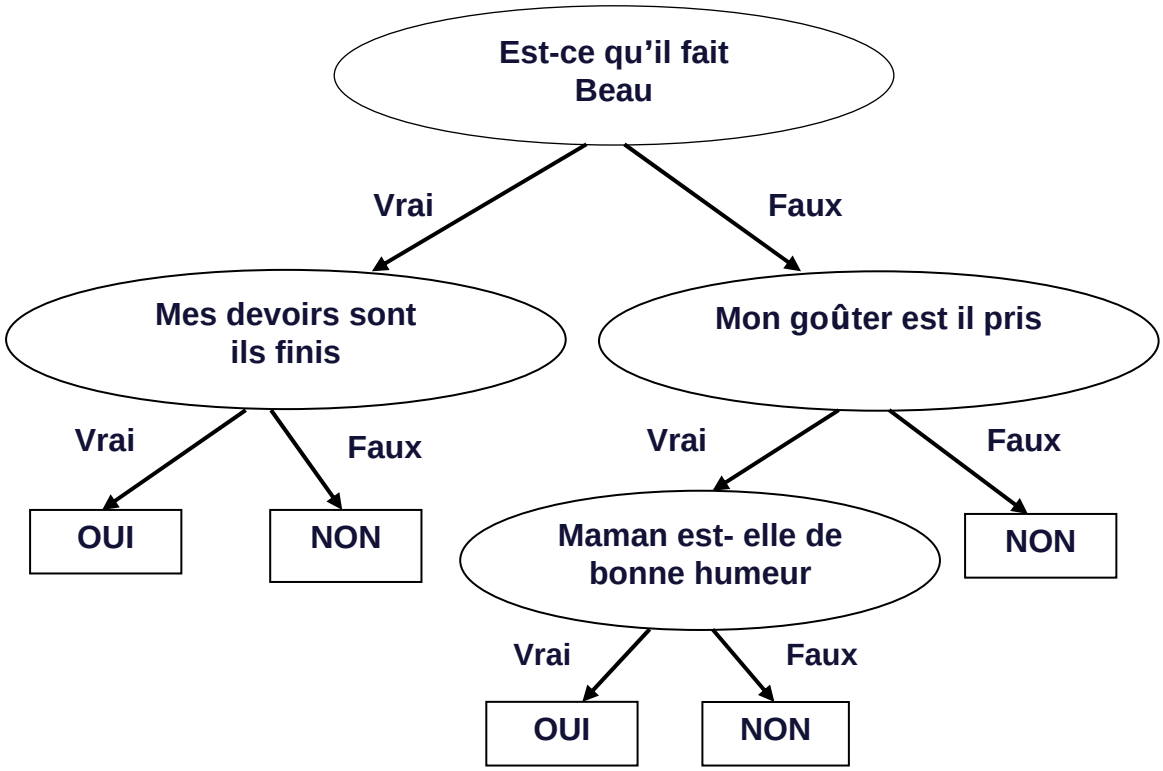


Figure IV.1 : l'arbre final

Conclusion :

Un logiciel de la méthode CART est élaboré dans le cadre de notre étude sont guide est donné à l'annexe 3.

Cette application, première en son genre, traitant des variables binaires à l'aide des arbre de décision, s'adresse à différents secteurs (tel que la médecine, Marketing,…) l'acquisition d'un module segmentation (d'SPPS) étant très onéreux non disponible sur le marché Algérien, c'est dans cette ligne de pensée que cette thèse à été réalisée.

Chapitre V

Aide à la décision pour le diagnostic médical

V.1. Introduction

Chaque jour, les médecins sont amenés à prendre des dizaines de décisions : prescrire un examen, déclencher ou interrompre un traitement, en changer ou faire admettre un patient à l'hôpital. La plupart des décisions sont prises facilement, le plus souvent automatiquement, lorsque le diagnostic est évident, le traitement choisi devient efficace et les risques nuls.

Néanmoins, dans bien d'autres cas, la bonne décision à prendre n'est pas évidente par exemple, lorsque le diagnostic est particulièrement incertain, lorsque les investigations ou traitements présentent des risques, lorsque l'efficacité des traitements disponibles n'est pas clairement établie ou lorsque l'éventail des choix possibles implique des arbitrages significatifs entre des objectifs contradictoires. En pareil cas, les avis des médecins sont souvent divergents, même chez ceux qui sont consultés comme experts quant à une démarche à suivre. Dans un tel contexte, l'analyse de décision peut offrir un éclairage complémentaire.

Ainsi le but de notre travail est d’élaborer un utilitaire dont le praticien disposera qu’il pourra utiliser comme moyen d’expertise de son diagnostic et de la future prise en charge thérapeutique.

L'Algérie est un pays à forte endémie goitreuse ceci est dû à plusieurs facteurs, dont l'un le déséquilibre de la ration iodée alimentaire ce qui entraîne une augmentation de l'incidence des dysthyroïdes ; c'est à dire des maladies du fonctionnement de la thyroïde.

Parmi cette pathologie, le cancer de la thyroïde est original par son caractère hormono-dépendant c'est à dire qu'il reste sous l'influences des hormones.

Dans ce travail on dispose d'une base de données regroupant les descriptions de patients atteints de cancer de la thyroïde. Ces patients proviennent de toute L'Algérie et sont décrits par des paramètres concernant le malade par les symptômes (année de déclaration de la maladie, volume du goitre, ..), Le traitement suivi (chirurgical, isotopique, ..) Sur l'état général du malade (récidives, état au dernier contrôle,..).

Une fiche de renseignement nous informe sur les différentes variables choisies. Ces variables peuvent être classées en cinq groupes :

1- Renseignement généraux sur le patient : ce groupe de variables caractérise les données sociales sur le malade (âge lors du diagnostic, sexe, origine, antécédent,etc.)

2- Caractéristique de la thyroïde : ces variables définissent les caractéristiques de la tumeur thyroïdienne ainsi que son évolution (volume du goitre, ancienneté, signes de maligne de

3- Particularité des adénopathies : cette classe de variables, précise les métastases compression, etc.)découvertes chez les patients lors du diagnostic souvent après exploration isotopique (métastases, cervicales, etc.).

4- Examen anatomo-pathologique de la thyroïde : c'est une partie très importante qui permet de classer les malades selon la nature de leur tumeur (analyse macroscopique et microscopique, ganglions).

5- Traitement des malades : Ce groupe de variables nous informe sur les traitements des la tumeur thyroïdienne et sur d'éventuelles rechutes des malades après traitement médical, il comporte aussi une variable très importante qui nous renseigne sur l'état du patient au dernier contrôle.

Cette base de données comme elle nous a été présentée, a nécessité certaines transformations pour être exploitable. A partir de cette fiche de renseignements des patients, seuls certains variables ont été retenus en collaboration avec le spécialiste de la maladie, et la classification histologique de l'Organisation Mondiale de la Santé O.M.S.

V.2. Au sujet de la thyroïde

La thyroïde est une glande en forme de papillon situé à la base du cou. Elle se compose de deux lobes, le gauche et le droit, reliés en leur centre. La glande thyroïde sécrète, emmagasine et libère des hormones (T_3,T_4) qui influent sur pratiquement toutes les cellule de l'organisme et contribuent à réguler le métabolisme.

Qu'est-ce que le cancer de la thyroïde ?

On parle de cancer de la thyroïde lorsqu'une tumeur ou une masse cancéreuse se développe dans cette glande. Normalement, les cellules thyroïdiennes se renouvellent dans certaines cellules, perturbant ainsi le cycle normal de croissance cellulaire. Les cellules anormales qui continuent de croître et de se reproduire de façon anarchique par former une tumeur.

Il existe quatre principaux types de cancer de la thyroïde, mais dans cette étude on ne s'intéresse qu'a deux types :

- Papillaire
- Vésiculaire

Les cancers papillaires sont les plus fréquents des tumeurs thyroïdiennes.

Le cancer de la thyroïde est presque trois fois plus fréquent chez la femme que chez l'homme, contrairement à la plupart des autres cancers il frappe à un âge précoce la majorité des patients sont en effet âgé de 20-54ans. Les chances de guérison sont fonction du type de cancer, de son siège anatomique, de l'âge du patient et de son état de santé en général.

V.2.1. Traitement

La prise en charge du cancer de la thyroïde comporte de nombreux volets. Le traitement le plus courant consiste en l'ablation de la tumeur cancéreuse suivie d'une radiothérapie destinée à détruire tant les cellules saines que les cellules cancéreuses de la thyroïde.

Au cours de la radiothérapie, on administre au patient par voie orale de l'iode 131 radioactif.

V.3. Caractéristique des variables retenues

La base de données telle qu'elle a été établie ne permet pas souvent de faire des traitements statistiques adéquats. Nous avons été conduit à mettre en œuvre des procédures de transformation, d'extraction et de recodification des données en vue de la construction de tableaux de données répondant à des codifications précises.

1- classification TNM

Elle permet de classer en fonction de son caractère plus ou moins agressif le cancer de la thyroïde.

T : Définit le diamètre maximal de la tumeur.

N : Définit le nombre de métastases ganglionnaires.

M : Définit la présence ou non de métastase viscérale à distance.

2- Risque et Récidives

- Envahissement de la capsule thyroïdienne.
- Age du malade au moment du diagnostic (16<age<45).
- Présence de métastases ganglionnaires ou viscérales à distance.
- Types histologique : les cancers folliculaires sont de moins bon pronostic que papillaire.

3- Traitement des cancers différenciés de la thyroïde

La prise en charge du cancer de la thyroïde comporte de nombreux volets, le traitement le plus courant consiste à l'ablation de la tumeur suivie d'une radiothérapie métabolique.

Chirurgie

Consiste en une ablation totale de la glande thyroïde.

Irathérapie

Irathérapie ou traitement isotopique par l'iode 131 radioactif est un traitement complémentaire. L'iode 131^{*} est administré au patient par voie orale sous forme de capsule.

V. 4. Résultats et Interprétation

V.4.1. les résultats obtenus par les arbres de décision

Comme nous l'avons déjà évoqué dans le chapitre 3 la segmentation est une méthode explicative qui sert à expliquer une variable qualitative en fonction d'autres variables.
Comme toutes les méthodes d'apprentissage nous avons subdivisé l'échantillon de base en deux échantillons, un échantillon d'apprentissage qui constitue le 2/3 de l'échantillon total et un échantillon test qui permet la sélection du meilleur sous arbre. La subdivision s'est effectuée de manière aléatoire.

Dans notre travail nous avons utilisée la segmentation par la méthode CART, alors pour pouvoir appliquer le logiciel élaboré pour notre étude nous avons besoin d'un échantillon où les individus sont répartis en deux classes.

V.4.1.1. Interprétation des résultas obtenus

Dans notre base de donnée, le problème d'apprentissage consiste à trouver une règle de décision binaire à partir de 509 malades sur 11 paramètres binaires.

Variables / Malade	Sexe	Age	Classe T	Pas D'anop	un seul ganglion	plusieurs ganglion	pas methas	methas unique	plusieurs methas	Microscop	iode	état Actuel
01	1	0	0	1	0	0	1	0	0	1	1	1
02	1	1	1	0	0	1	1	0	0	0	1	1
03	1	1	0	1	0	0	1	0	0	0	0	1
04	1	0	1	0	0	1	1	0	0	1	1	0
05	1	1	1	1	0	0	1	0	0	0	1	1
06	1	1	0	0	0	1	1	0	0	1	1	1
07	0	0	0	0	0	1	0	1	0	1	1	1
08	1	1	1	0	0	1	1	0	0	1	1	1
09	1	0	0	0	1	0	0	0	1	1	1	1
10	1	1	0	0	0	1	0	1	0	1	1	1
11	1	0	0	0	1	0	0	0	1	1	1	1
12	1	0	0	0	1	0	0	0	1	1	1	1
13	0	0	0	0	0	1	0	1	0	0	1	0
14	1	0	0	1	0	0	1	0	0	0	1	0
15	1	0	0	1	0	0	1	0	0	1	1	1
16	1	1	0	0	0	1	1	0	0	0	1	1
"	"	"	"	"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"	"	"	"	"
509	1	1	1	1	0	0	1	0	0	0	1	1

La première étape dans la construction d'un arbre binaire, est de choisir parmi les11 variables celles qui a la plus petite valeur d'entropie ou bien le meilleur gain d'entropie.

Variables	Sexe	Age	Classe_T	Pas D'anopath	Un seul ganglion	Plusieur ganglion	Pas Methas	unique Methas	Plusieur Methas	Micros-copique	iode	Etat Actuel
Entropie	0,7257	0,6960	0,7089	0,6892	0,7061	0,6881	0,6927	0,7111	0,7108	0,7252	0,6287	0,6900

D'après les résultats obtenus on constate que la variable qui a la plus petite valeur d'entropie est la variable « Iode ».

On choisit donc pour racine de l'arbre la variable « Iode », On développe chaque branche de la même manière avec un nouvel ensemble de n-uplets.

Le coût de mauvais classement dans l'arbre vaux 0,017 ce qui implique une bonne segmentation, et les segments ont une représentation significative.

Puit vient l'étape de l'élagage dans cette étape en utilisant l'échantillon test, l'arbre final retenu se réduit à 12 segments terminaux, avec un taux d'erreur égale à 0,14.

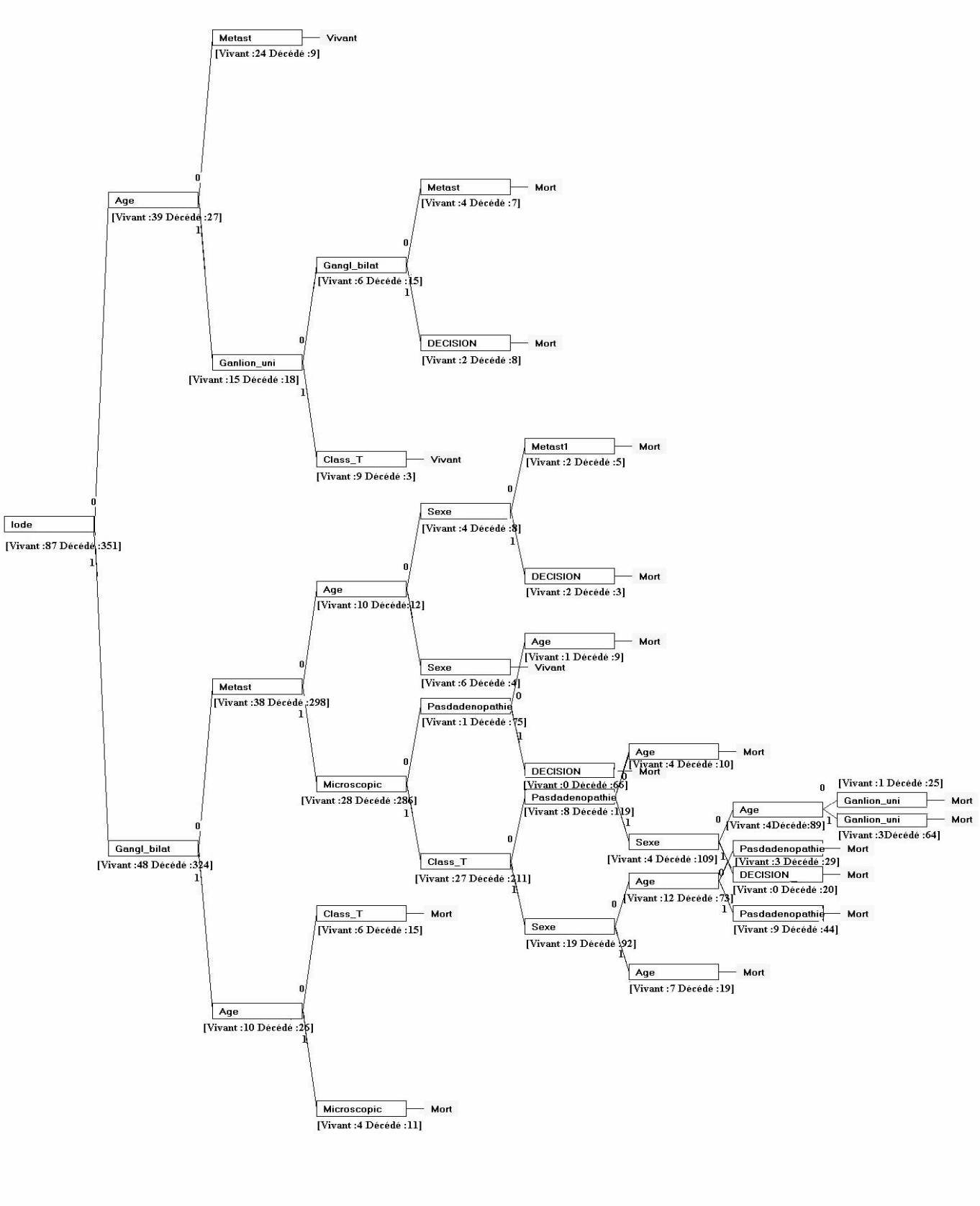


Figure V.1 : Arbre obtenue avant élagage

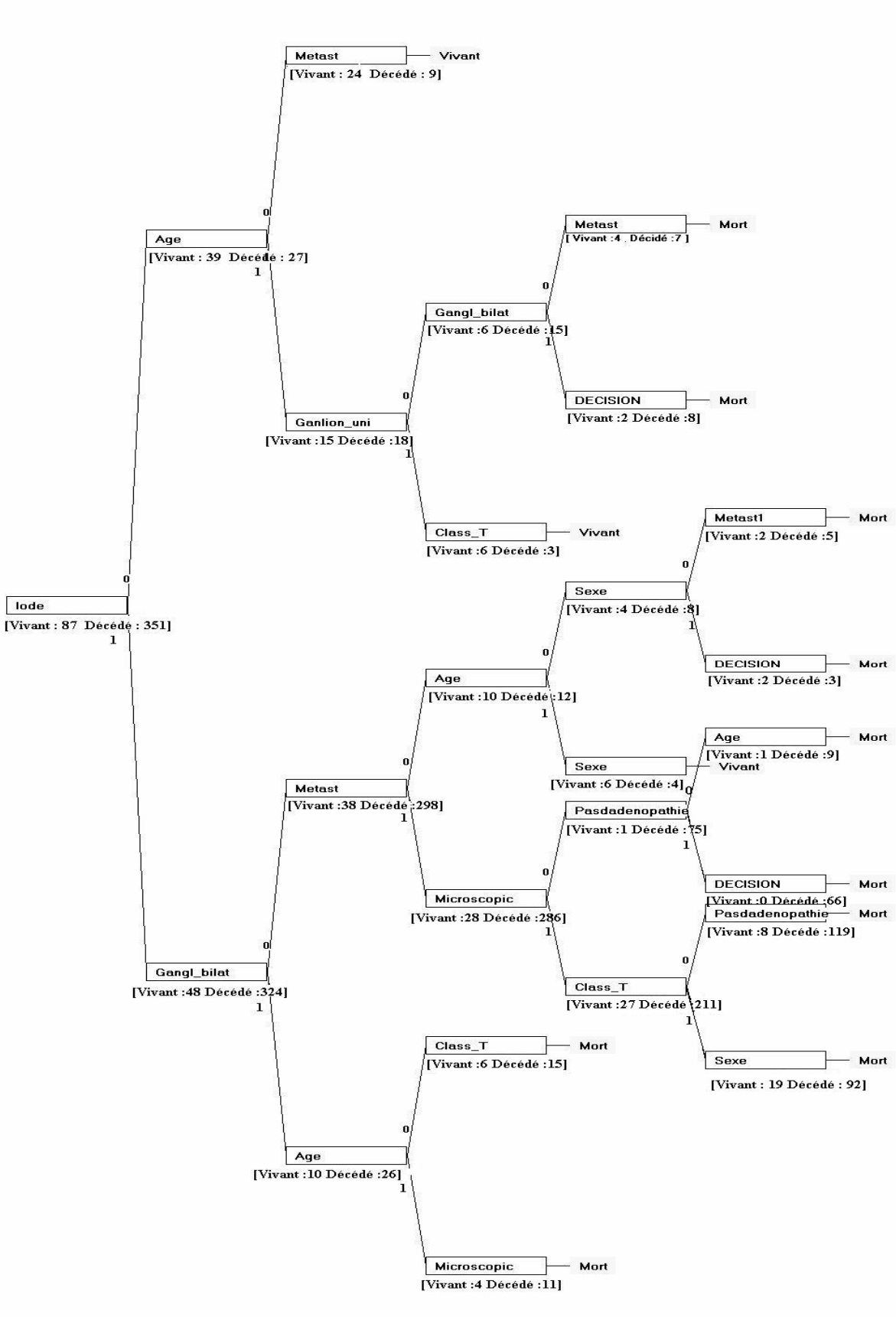


Figure V.2 : Arbre obtenue après élagage

Une règle sera en fait un chemin racine feuille

Si on prend le premier nœud on a :

Un patient qui n'a pas pris sa dose d'iode ayant plus de 45ans a plus de chance de décéder. D'après ce chemin on constate que la prise d'iode et l'âge du patient sont des variables fortement corrélés avec la survie du patient. De plus l'âge de malade constitue un facteur de risque, en effet plus le malade est âgé plus ses chances de survie sont faibles.

Si on prend le nœud «pas d'adénopathie », on constate qu'un patient qui n'a pas de ganglions a plus de chance de survivre.

D'après les résultats obtenus par l'étude des données, la variable sexe apparaît comme un facteur de risque, on remarque que les femmes sont les plus touchées par le cancer de la thyroïde mais un fort taux de décès caractérise les hommes.

Discussion :

Ces résultats faites sur 760 patients recensés en Algérie ont montré que l'âge, le sexe et la pathologie et la prise d'iode sont fortement corrélés avec la survie du patient.

On peut dire que les chances de survie décroissent avec l'âge avancé des patients.

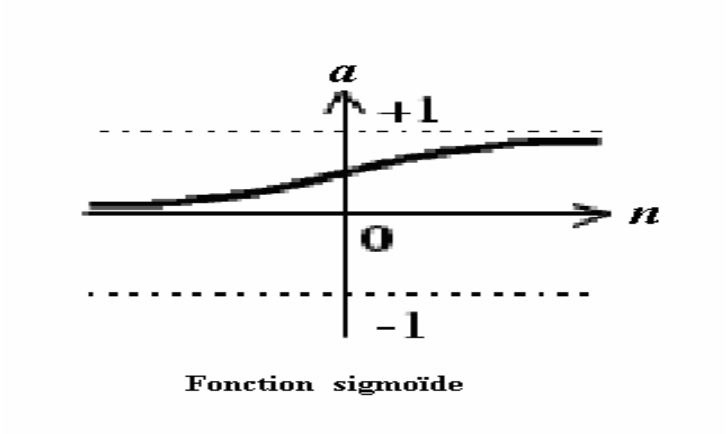
Les résultats obtenus par le logiciel « Arbre de décision » sont confirmés par les médecins spécialistes.

V.4.2. Les résultats obtenus par les réseaux de neurones

Nous avons abordé le même problème médical à l'aide des réseaux Multicouches.

V.4.2.1. Architecture de réseaux de neurones

Pour modéliser donc notre problème nous avons utilisé un réseau avec 11noeurones dans la couche d'entrés et 7 dans la couche cachée, et la fonction sigmoïde pour calculer les valeurs d'activation de réseau entre les couches.

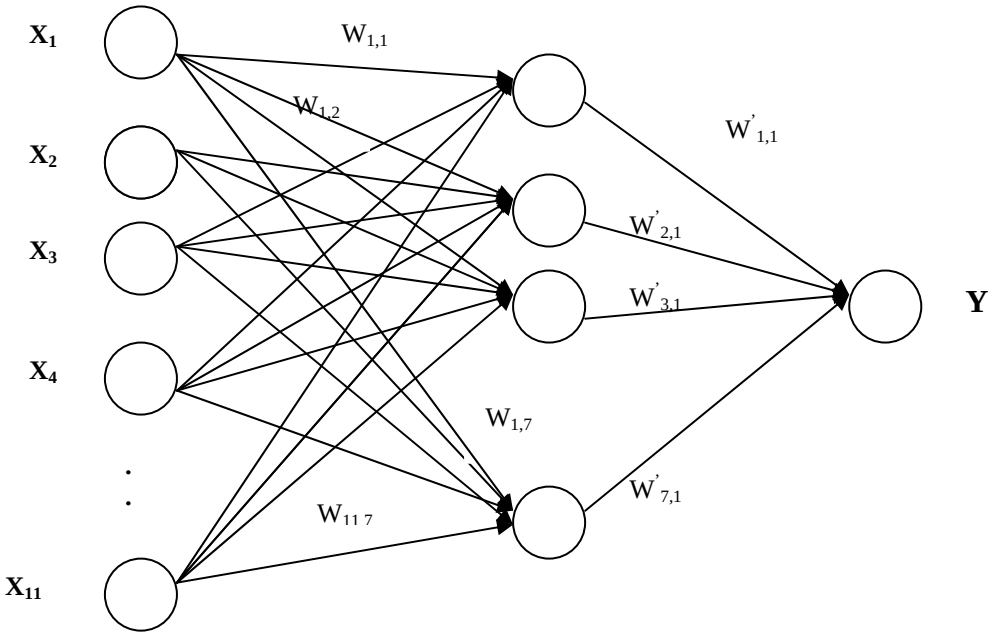


Fonction sigmoïde

La première étape dans les réseaux Multicouches consiste à calcules les valeurs d'activations ensuite en calcul la fonction d'activation, qui va nous donner la sortie du réseau.

$$S_i = \sum_{j=1}^{11} W_{ij} X_j$$

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



V.4.2.2. Interprétation des résultas obtenus

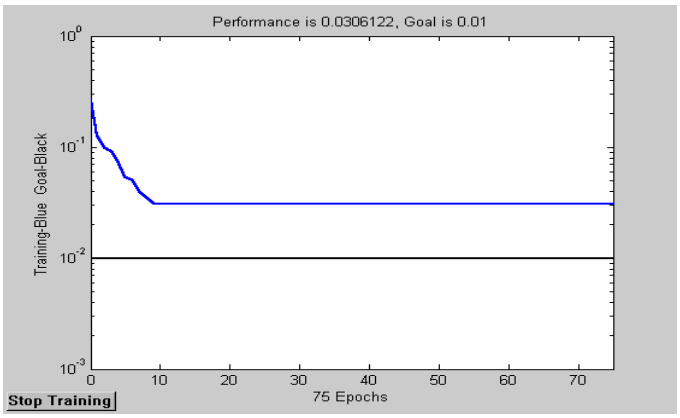
La fonction a approché étant une fonction à deux valeurs {0,1} sur son domaine de définition, donc elle n'est pas régulière. Les résultats obtenus par les réseaux multicouches s'interprète comme une approximation de cette fonction, qui prend des valeurs proche de 0 et de 1 sur le domaine de définition de la fonction initiale, elle peut être interpréter comme une fonction discriminante de (0,1) des deux classes (Vivant, Décédé).

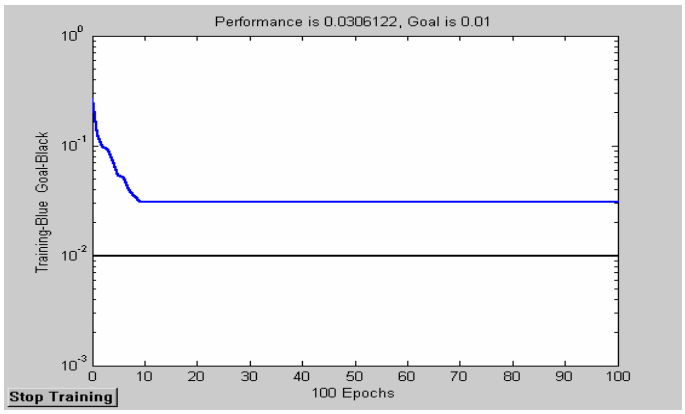
Les poids $W_{i,j}$

0,5623	-3,4583	0,9798	1,4846	-1,8253	1,0497	-1,0026	-2,5258	2,3719	-0,5761	-3,6805
-0,4070	0,9423	1,8230	1,9056	3,0630	-2,5890	-0,8944	3,1232	-0,5842	2,9825	-1,6957
-3,3249	0,7912	1,4037	-0,1576	-1,9513	2,5064	0,6675	-1,9113	2,8450	-0,0713	-3,2866
-2,9608	-3,0438	-2,6050	0,3451	1,9099	-2,0427	-2,3813	0,9144	1,4714	1,9800	1,2100
-1,1783	-3,0438	-0,2870	-2,3844	2,5704	-2,0719	-2,9072	2,9396	1,1791	-0,2463	0,9450
-3,4307	-2,1818	-0,4098	-0,3467	-1,8875	3,4799	-0,2915	1,1613	-1,0829	-0,3003	3,4019
-1,0726	0,8010	-1,3518	1,9851	-2,4001	-0,5563	3,4047	3,4074	-3,0752	-0,4554	0,4853
1,0993	-2,6544	-2,0804	2,3581	-2,7022	-0,9608	2,6082	-2,9447	-2,0665	-0,5270	-0,5987
-2,7652	-0,9005	1,1925	-1,5390	-2,8657	-1,2628	-1,6003	-2,4647	-2,0998	2,7282	-2,0500

Les poids $W'_{j,k}$

0,5593	-3,4583	0,9783	1,4827	-1,8252	1,0480	-1,0042	-2,5283	2,3724	-0,5772	-3,6798
-0,4069	0,9415	1,8231	1,9058	3,0625	-2,5893	-0,8943	3,1233	-0,5850	2,9819	-1,6971
-3,3246	0,7912	1,4044	-0,1576	-1,9509	2,5065	0,6671	-1,9106	2,8451	-0,0711	-3,2862
-2,9620	-3,0335	-2,6056	0,3436	1,9099	-2,0429	-2,3821	0,9136	1,4716	1,9792	1,2102
-1,1790	-3,0441	-0,2873	-2,3852	2,5705	-2,0721	-2,9071	2,9387	1,1790	-0,2469	0,9448
-3,4302	-2,1823	-0,4096	-0,3468	-1,8879	3,4806	-0,2915	1,1614	-1,0836	-0,3003	3,4010
-1,0685	0,8004	-1,3513	1,9879	-2,4005	-0,5544	3,4058	3,4106	-3,0752	-0,4541	0,4850
1,0995	-2,6565	-2,0799	2,3588	-2,7035	-0,9604	2,6074	-2,9427	-2,0679	-0,5271	-0,6000
-2,7668	-0,8999	1,1936	-1,5419	-2,8651	-1,2625	-1,6007	-2,4667	-2,0990	2,7291	-2,0469





Les résultats obtenus dans notre travail montre que les réseaux multicouches à une seule couche cachée convient à notre modélisation. De plus en changent le nombre de neurones d'entrée et le nombre de neurones cachée, ou le nombres d'itération l'erreur n'a pas diminué. Ainsi pour notre travail plusieurs simulations ont été effectuées, conduisant au fait qu'un réseau à 07 neurones dans la couche cachée suffisent à la modélisation du diagnostic d'un malade.

V.5. Comparaisons entre les arbres de décision et les réseaux de neurones

Les méthodes	Arbre de décision	Réseaux de neurones
Le taux d'erreurs	0,14	0,0306

A partir des résultats obtenus par les deux méthodes on constate que le taux d'erreur dans la méthode CART est moins important que celui obtenu par les réseaux multicouches.

V. 5.1. Conclusion

Les résultats sont conformes aux résultats émis par les spécialistes en cancérologie thyroïdienne, basés sur d'autres méthodes statistiques :

- L'âge est un facteur prédictif de la survie.
- Le sexe permet de séparer de façon discriminantes deux types de survie importants chez la femme moindre que chez l'homme.

- La forme histologique qui montre que le cancer papillaire est plus « bénin » que le cancer vésiculaire.
- La taille de la tumeur : plus la tumeur est « gros » moins la survie est importante.
- La présence d'adénopathies « ganglion » est péjorative quant à la survie.
- Enfin la présence de métastase traduit une mauvaise évolution du cancer et donc réduit la survie.

Toutes ces informations connues par l'expérience des médecins spécialistes deviennent donc quantifiables grâce au logiciel que nous avons développé, qui permet un diagnostic rapide et une orientation fiable, et donc vous permettre des analyses statistiques objectives.

Conclusion générale

Nous avons présenté dans notre mémoire des systèmes d'apprentissage basés sur les méthodes de segmentation : arbre de décision et réseaux de neurones ; il nous semble important de rappeler que pour l'analyse de données, les méthodes statistiques restent très importantes et sont largement utilisées dans plusieurs domaines.

L'apprentissage par les méthodes de segmentation fait partie de la classe des méthodes non-paramétriques et sont issues de l'intelligence artificielle.

Dans notre travail, nous avons utilisés cette technique de prise de décision dans une pathologie fréquente en Algérie, à savoir le cancer de la thyroïde.

A partir d'une base de donnée initiale très documentée, nous avons appliqué des méthodes de segmentation. Ceci à permis la conception du logiciel arbre de décision basé sur la méthode CART. IL a tracé rapidement le meilleur chemin proposé par la suite aux médecins praticiens soucieux d'une prise en charge effective du cancéreux. Les résultats obtenus par le logiciel sont extrêmement faciles à interpréter, le diagramme de l'arbre permet d'identifier rapidement les plus importants prédicteurs. La réalisation de ce travail nous a permis la maîtrise du langage évolué Delphi.

Ainsi, le praticien disposant de cet outil d'aide, pourra l'utiliser comme moyen d'expertise de son diagnostic et de la future prise en charge thérapeutique.

Enfin les résultats obtenus au cours de cette étude mérite d'être poursuivis, et d'être appliqués pour d'autres domaines.

Il existe plusieurs types de réseaux, et pour chaque réseau quelques paramètres sont modifiés afin d'améliorer la précision. Les résultats obtenus dans notre travail montre que les réseaux multicouches ayant une seule couche cachée et quelques neurones à l'intérieure de chaque couche est un approximateur universelle parcimonieux pour toute fonction suffisamment régulière dans son domaine de définition.

Dans ce travail nous avons voulus aborder le même problème médical à l'aide des réseaux de neurones. De plus en changent le nombre de neurone d'entrée et le nombre de neurones dans la couche cachée, le nombre d'itération l'erreur n'a pas diminué, ainsi pour notre travail on a trouvé qu'un réseau à 11 entrées et 07 neurones de la couche cachée suffisent à la modélisation du diagnostic d'un malade atteint du cancer de la thyroïde.

Cette thèse n'a pas la prétention de résoudre le problème de l'application de la segmentation, mais traite de l'application des réseaux de neurones aux données binaires, ce qui fait son originalité. Elle propose un outil d'aide à la décision, voire de modélisation des données binaires.

Les perspectives à envisager concernent la nature des données. Ainsi l'application pourrait être élargie aux autres composantes du diagnostic d'un malade. La diversité des données pourrait contribuer à l'affinement de la modélisation.

Il est à signaler que divers méthodes peuvent être associées aux réseaux de neurones : la logique floue, les chaînes de Markov cachées, les algorithmes génétiques permettant ainsi L'amélioration de la précision des modèles.

Nous pensons étendre cette étude au cas des arbres de décision non binaire. Cela va nous permettre en outre l'introduction d'autre coefficient d'association entre variables et faire une étude comparative avec des données médicales

Dans un autre cadre de recherche, nous pensons et aimerions étendre notre travail à un arbre neuronal.

Les différents types de critères et indices d'association

Dans le processus de construction d'un arbre de décision, il faut à chaque nœud t, choisir une variable w pour la segmenter en deux nœuds descendants, t_d et t_g. Il s'agit de tirer, parmi un ensemble de variables, celle qui est la plus associée avec la variable à discriminer qu'on désigne par Y.

$$W : O \rightarrow \{t_g, t_d\} \text{ et } y : O \rightarrow \{y_1, y_2, \ldots, y_K\}.$$

Cependant l'utilité de ces coefficients est plus générale et concerne le croisement de deux variables qualitatives, ayant chacune un nombre quelconque de modalités ou valeurs. Il s'agit d'évaluer le croisement de la variable explicative et la variable a expliqué, sur un ensemble test. Les biologistes [Rost &sander, 1993] ont introduit un coefficient appelé « Info », qu'ils utilise pour évaluer la qualité de la prédiction, il s'agit d'un coefficient d'association entre variables qualitatives nominales, cette qualité est très généralement mesurée par la proportion (ou pourcentage) de bon classement.

Dans la littérature de nombreux critères ou indices d'association .des coefficients qui sont construis à partir du tableau de contingence croisant les deux variables qualitatives nominales.

Très souvent, la construction d'un coefficient d'association est purement intuitive, pour les coefficients contingenciaux, cette construction repose généralement sur des statistique simples. Pour les coefficients dont la conception se situe au niveau de O×O, la relation qui existe entre les graphes des relations, d'équivalence, associées aux deux partitions à comparer.

Quel que soit le niveau de conception, O ou O×O, on peut toujours chercher à, étudier la distribution statistique du coefficient dans l'hypothèse d'indépendance entre les deux variables. En outre, quel que soit le degré du lien entre les deux variables, mesuré au moyen du coefficient au niveau de population-mére P, on peut aussi étudier la distribution d'échantillonnage d'un tel coefficient].

Enfin, quelle que soit la conception du coefficient d'association, le calcul s'effectue au niveau du tableau de contingence 2×K, croisant la variable prédictive et celle à prédire.

1.1. L'influence des coefficients d'association sur l'arbre de décision

John Mingers a utilisé dans ses expériences dix coefficients, dont six sont liés à la qualité d'information, trois à la statistique du χ^2, et le coefficient de Gini utilisé par [Breiman &al 84]. La seule différence que John Mingers a trouvé dans ce expériences concernera la taille de l'arbre maximal avant l'élagage, $T_{max.}$.

Pour étudier l'influence des coefficients sur les arbres de décision, plusieurs coefficients sont proposés : Gini, Twoing, (cf. §II), Shannon, χ^2, Matusita, Affinité W, Affinité Δ ; les coefficients de Lerman s, Q_1, R, le coefficient introduit par Lerman et Pinto da Costa (1996) ; plus un choix aléatoire. Ces coefficients seront décrits ci-dessous :

1.2. Le coefficient de Shannon :

La mesure d'impureté du nœud t donnée par la quantité d'information de Shannon est :

$$I(t) = -\sum_{1 \le j \le K} p_j^t \log_2 (P_j^t) \qquad (1.1)$$

L'utilisation de cette formule classique de la théorie de l'information a été proposée par Quinlan (1979, 1983, 1986). La fonction I(t) mesure la quantité d'information nécessaire pour classifier les individus.

Propriétés :

- Elle est d'autant plus petite que le sous-ensemble d'apprentissage correspondant au nœud t apporte une information certaine au niveau des probabilités P_j^t ;
- L'information nécessaire est nulle quant un des événement est certain (il n'y a qu'un P_j^t non nul) ;
- Si tous les événements sont équiprobables, c'est à dire, si les P_j^t sont tous les mêmes, alors l'incertitude est maximale et l'information nécessaire, I(t), est maximal ; auquel cas, elle vaut $\log_2 (K)$.
- La valeur de I(T) dépend aussi du nombre K de classes à prédire, c'est à dire ; plus il y a de classe et plus elles sont équiprobables, plus il est difficile (I(t) grande) de classer un individu.

On cherche donc toujours à maximiser la diminution d'impureté (grain d'information), donnée par la formule suivante :

$$S(t,w) = I(t) - P_l^t I(t_l) - P_r^t I(t_r) \qquad (1.2)$$

où $P_l^t I(t_l) + P_r^t I(t_r)$ est une mesure globale pour l'attribut prédictif w. On démontre que I(t), qui se met sous la forme :

$$I(t) = -\sum_{1 \le j \le K} (p_l^t P_j^{t_l} + p_r^t P_j^{t_r}) \log_2 (P_j^t) \qquad (1.3)$$

est plus grand que $P_l^t I(t_l) + P_r^t I(t_r)$. D'autre part, la différence S(t, w), qui est donc positive, peut s'écrire sous la forme symétrique

$$S(t,w) = \sum_{1 \le j \le k} [P_l P_j^{t_l} \log_2 (\frac{P_j^{t_l}}{P_j^t}) + P_r P_j^{t_r} \log_2 (\frac{P_j^{t_r}}{P_j^t})] \qquad (1.4)$$

Cette expression représente l'information entre les deux variables qualitatives w et c. Le choix de l'attribut w doit être celui qui maximise la dernière quantité critère.
[Quinlan 86] a aussi proposé d'utiliser une variante à S(t,w)

$$GR(t,w) = \frac{S(t,w)}{I(w)} \quad \text{(Gain ration)} \qquad (1.5)$$

La mesure GR a tendance à produire des nœuds avec un nombre faible d'individus, ce qui doit produire des arbres plus petits.

1.3. Le critère du χ^2

Pour introduire ce critère, revenons à l'ensemble des objets, et au couple de partition sur l'ensemble, respectivement définis par deux variables à modalités, u et v. On notera ces partition par

$$\pi = \{E_i / 1 \le i \le p \ \}$$
et (1.6)
$$\chi = \{F_j / 1 \le j \le q\}$$

Considérons maintenant le tableau de contingence croisant ces deux partitions et soit

- $n_{ij} = \text{card}(E_i \cap F_j)$; $1 \le i \le p, 1 \le j \le q$
- $n_{i.} = \text{card}(E_i)$; $1 \le i \le p$
- $n_{.j} = \text{card}(F_j)$; $1 \le j \le q$
- $n = \text{card}(O)$

L'expression de la statistique du χ^2 attachée au tableau de contingence

$$\chi^2 = \sum_{i=1}^{p} \sum_{j=1}^{q} \frac{n\left[(n_{ij} - \frac{n_{i.} n_{.j}}{n} \right]^2}{n_{i.} n_{.j}} \qquad (1.7)$$

Proposé par Pearson en 1904, représente une distance traditionnelle entre deux tableaux de contingence. La statistique qui en résulte suit, approximativement, une distribution du χ^2.

La distance du χ^2 est fonction de p, q et N. il en existe deux versions « indices d'association », indépendantes de p, q et de N : l'indice de Tchuprov

$$\tau = \chi^2 / N\sqrt{(p-1)(q-1)} \qquad (1.8)$$

et la version donnée par Gramer en 1964,

$$\chi^2_{norm} = \chi^2 / (N.\min[(p-1),(q-1)]). \qquad (1.9)$$

$$\chi^2/n_t = p_l^t p_r^t \sum_{l\le j\le k} \frac{1}{p_j^t}(p_j^{t_l} - p_j^{t_r})^2 \qquad (1.10)$$

Cette expression n'est autre que χ^2/n_t, où χ^2 est le coefficient usuel du Chi-deux, et n_t =card(O_t).

1.4. Le critère de Hellinger

Comme c'est le cas pour l'indice de Gini, Twoing et χ^2, ce critère est défini au moyen d'une distance entre les deux distributions empiriques de probabilité

$$\left\{p_j^{t_l}/1 \le j \le K\right\} \quad \text{et} \quad \left\{p_j^{t_r}/1 \le j \le K\right\} \qquad (1.11)$$

En adoptant la métrique euclidienne ordinaire, on a pour l'expression de ce critère

$$H(t_l, t_r) = \sum_{j=1}^{k}\left(\sqrt{p_j^{t_l}} - \sqrt{p_j^{t_r}}\right)^2 = 2\left(1 - A(t_l, t_r)\right);$$

où (1.12)

$$A(t_l, t_r) = \sum_{j=1}^{K} \sqrt{p_j^{t_l} p_j^{t_r}}$$

La méthode ARCADE

2.1. Introduction

La méthode ARCADE (Arbre de Classification et de Décision) comprend la méthode CART et notamment son aspect le plus original, qui concerne l'élagage de l'arbre. Elle s'en distingue néanmoins par trois compléments :

1. Pour la construction de l'arbre une nouvelle famille de coefficients d'association entre variables qualitatives [lerman 1970, 1981, 1992a,1992 b], [Bacelar-Nicolau 1980,1988] issus de la méthode de classification AVL (Analyse de Vraisemblance des liens) [lerman1970] cette nouvelle famille permettra, en outre, de construire de façon plus riche des arbre de décision non binaire.
2. Pour la validation statistique de l'arbre, une nouvelle méthode est introduite, qui est la méthode Hybride de la validation croisés et de l'ensemble test. Cette méthode, au lieu de minimiser un certain critère pour trouver le sous-arbre optimal, cherche plutôt à estimer par une moyenne ce sous-arbre.
3. La distinction la plus importante entre la méthode CART et ARCADE cancers la manière de binariser les attributs qualitatifs prédictifs.

L'apport principal de la méthode ARCADE consiste alors à choisir, parmi touts les variables binaires engendrées par un attribut qualitatif à L catégories, celle qui sont les plus prédictives. La sélection de variables a des objectifs très importantes. D'une part, il s'agit de réduire la complexité de recherche.

2.2. Coefficient d'association par la méthode AVL

La méthode AVL utilise la notion statistique de vraisemblance pour calculer le lien entre les variables. Ainsi, les indices de similarité utilisés dans AVL sont probabilistes.

2.3. Choix de l'arbre optimal par la méthode Hybride

Considérons alors, la division de l'ensemble d'apprentissage O en U sous-ensembles

Construisons alors, sur chaque $O^{u,i}$, l'arbre de décision maximal $T_{max,u,i}$ et la séquence correspondante de sous-arbre emboîtés à laquelle correspond la séquence croissante de paramètres d'élagage

$$T_1^{u,i} > T_2^{u,i} > .... > T_1^{u,i}$$

$$\alpha_1^{u,i} \leq \alpha_2^{u,i} \leq .... \leq \alpha_{s(u,i)}^{u,i}$$

O_i^u est alors utilisé pour choisir l'arbre $T_{s(u,i)}^{u,i}$ qui minimise le coût de mauvais classement.

Cet arbre correspond un paramètre d'élagage, que nous désignons simplement par $\alpha(u,i)$.

l'idée consiste à prendre la moyenne arithmétique des ces $\alpha(u,i)$

$$\alpha(u) = \frac{1}{n}\sum_{i=1}^{n} \alpha(u,i)$$

on construit maintenant sur O^u un arbre maximal. Après élagage, on trouve le sous arbre, T^u, correspondant à $\alpha(u)$. Le coût de ce sous-arbre, $C(T^u)$, est estimé avec l'ensemble O_u, qui n'a pas encore été utilisé.

Nous finissons par trouver un ensemble de U sous-arbre optimaux,

$$\{ T^u / 1 \leq u \leq U \}$$

Dont les coût constituent l'ensemble

$$\{ C(T^u) / 1 \leq u \leq U \}$$

Et auquel correspond l'ensemble des paramètres d'élagage optimaux

$$\{ \alpha(u) / 1 \leq u \leq U \}$$

Considérons maintenant la moyenne arithmétique de ce dernier ensemble ;

$$\overline{\alpha} = \frac{1}{U}\sum_{u=1}^{U} \alpha(u)$$

Sur l'ensemble total O construisons l'arbre maximal T_{max}, et son sous-arbre correspondant à $\overline{\alpha}$, $T_{\overline{\alpha}}$. A ce sous-arbre, qu'on utilise pour prédire, on attribue le coût moyen

$$C(T_{\overline{\alpha}}) = \frac{1}{U}\sum_{u=1}^{U} C(T^u)$$

au contraire de la validation utiliser par la méthode CART, qui cherche le sous-arbre optimal en minimisant un certain critère, cette méthode « hybride » cherche plutôt à estimer le paramètre d'élagage par une moyenne.

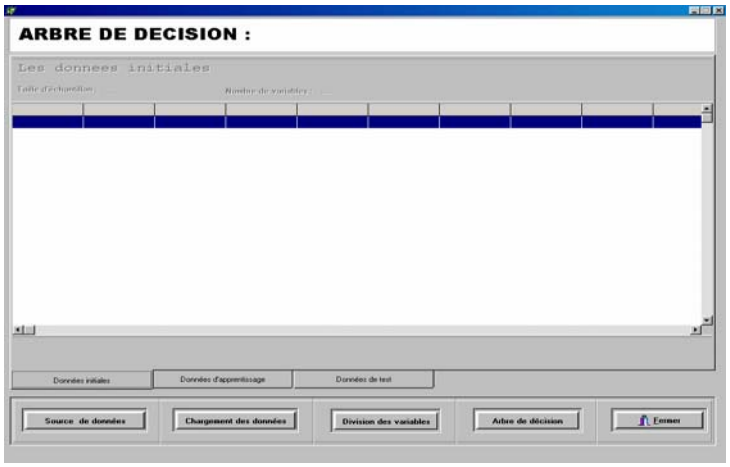
Le logiciel : Arbre de décision par la Méthode CART

3.1. Présentation de logiciel élaboré

La fenêtre principale du logiciel est une fenêtre d'authentification (autorisation d'accès) où l'utilisateur doit obligatoirement entrer son nom (utilisateur) et son mot de passe (mot de passe) pour exploiter le logiciel.

Après avoir saisie le nom de l'utilisateur et le mot de passe, on clique sur le bouton (OK) le système vérifie si les informations tapées sont correctes : si oui le système donne à l'utilisateur l'autorisation d'accès, sinon l'utilisateur doit retenter une nouvelle fois l'opération d'accès au système.

Si l'utilisateur a une autorisation pour travailler avec le logiciel, le système active certains boutons d'activité (les boutons de navigation dans le logiciel) de la fenêtre principale.



Le bouton « source de données » est utilisé pour charger les données à partir de fichiers type Excel ou base de données Access. Ce bouton ouvre une fenêtre de chargement de données.

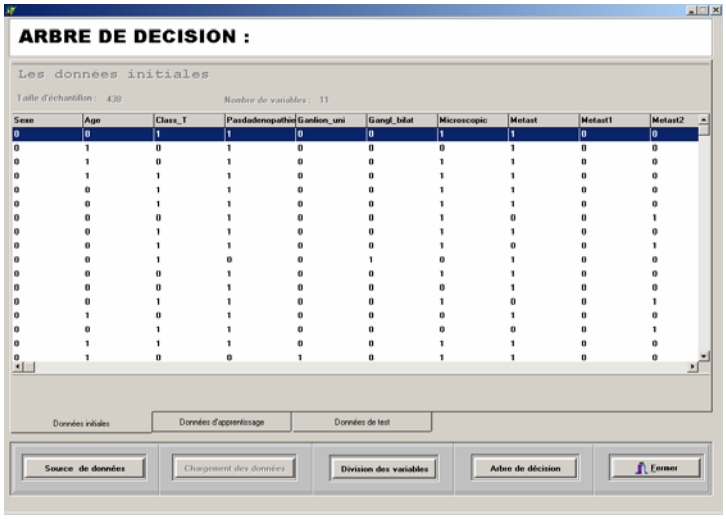
Le bouton « arbre de décision » pour le calcul des entropies et l'affichage de l'arbre de décision et même l'arbre élaguée.

Le bouton « quitter » pour sortir et fermer le logiciel.

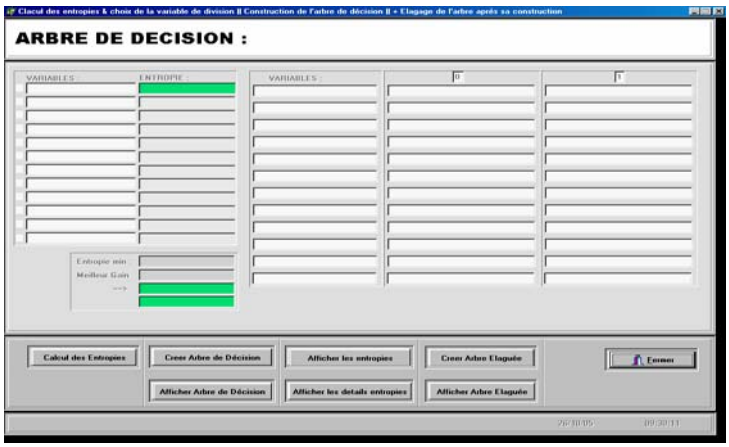
Le fait de cliquer sur le bouton « arbre de décision », active une fenêtre contenant trois onglets :

- le premier onglet contient toutes les données initiales (pour visualiser les donnés il faut d'abord appuyer sur le bouton au dessous « chargement de données ».
- le deuxième onglet contient les données d'apprentissage (3/4 des données chargées)
- le troisième onglet contient les données test (1/4 des données chargées)

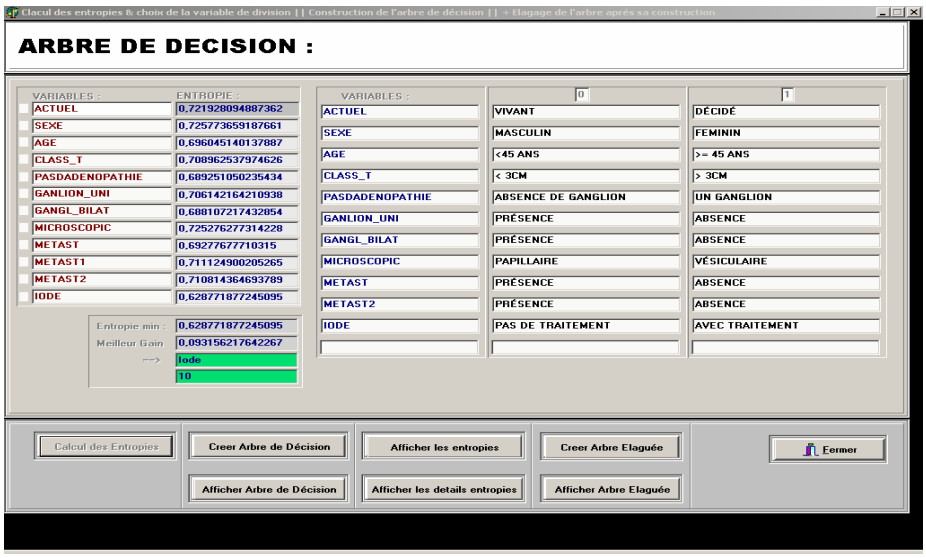
N.B : le choix des données d'apprentissage se fait de manière aléatoire en utilisant bien sur la fonction aléatoire (random)



Après le chargement des données, on clique sur le bouton « division des variables » pour le choix des données d'apprentissage et les données de test. Puis on passe à l'étape de construction de l'arbre de décision en appuyant sur le bouton « arbre de décision ». Ceci donnera la fenêtre suivante :



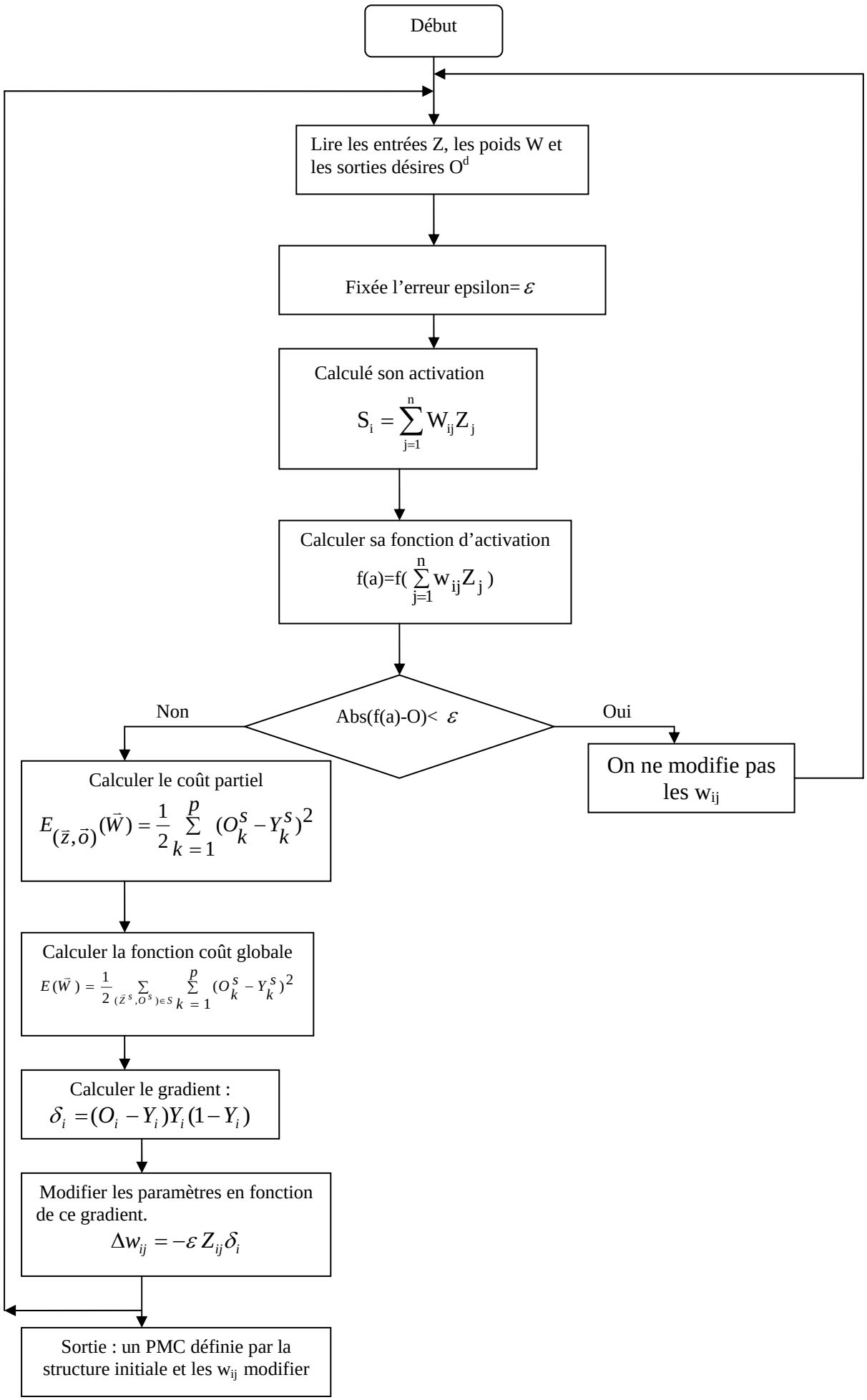
Le bouton « calcul des entropies » fait le calcul d'une part d'entropies de la variable décision, et les entropies de chaque variable ensuite il choisira la variable de meilleur gain (la plus petite entropie).



Une fois l'étape précédente est terminer, on crée l'arbre de décision (appuyez sur le bouton « créer arbre de décision ») et on a la possibilité de l'afficher (par le bouton « afficher l'arbre de décision »).

Le « bouton arbre élagué » fait l'élagage de l'arbre déjà construit, le programme analyse toutes les branches de l'arbre de manière ascendante (de la feuille à la racine) et test le taux d'erreur. Puis réaffiche l'arbre élagué dans une autre fenêtre.

Organigramme de réseau Multicouches



Références Bibliographiques

[1] R. A. Barron. (1993). Universal approximation bounds for superpositions of a sigmoide function. IEEE Transactions on Information Theory, 39, 930- 945.

[2] Assadi Réza & Khattar Karim. (1 mai 2002). L'utilisation d'un réseau de neurones pour optimiser la gestion d'un firewall. École Polytechnique de Montréal. Québec, Canada.

[3] Breiman & al. (1984). Classification and Regression Tree. Wadsworth, Belmont.

[4] Bakiri.F&Benmiloude.M. (1991). Maladies des Glandes Endocrines. INESSM .O.P.U. Alger.

[5] Benahmed Nadia. (Mars 2002). Optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isoles : sélection et pondération des primitives par l'algorithmes Génétiques. Ecole de technologie supérieure. Université de Québec. Canada.

[6] Bishop Christopher M. (1995). *Neural networks for pattern recognition*. Birmingham: Clarendon press Oxford.

[7] Christine Largeron – Leténo Algorithme de comparaison d'arbres : application au classement de séquences. Université Jean Monnet Saint-Etienne. (France).

[8] Claude TOUZET.(Septembre 1998). Réseaux de neurones artificiels a la robotique coopérative. Université de Droit, d'Economie et des Sciences d'Aix-Marseille III Faculté des Sciences et Techniques de Saint-Jérôme.(France)

[9] Culloch W.S.Mc et Pitts W.(1943) A logical calculs of the idea immanent in Nervous activity. Bulletin of Mathematical Biophysics,5, p 115-133 al.

[10] Crucianu Mihail. (Avril 1997). Algorithmes d'évolution pour les réseaux de neurones. Laboratoire d'Informatique ; Ecole d'Ingénieurs en Informatique pour l'industrie (France)

[11] C. Berge. (1973). Graphes et hypergraphes, Bordas, deuxième édition

[12] Dreyfus G., Martinez J.M., Samueliesc M. (2004). Réseaux de neurones méthodologie et application. Edition Eyrolles.

[13] Duchesne Thierry (2004). Département de mathématiques et de statistique Université Laval. Canada.

[14] F-C.chen, (1990). Back-propagation neural network for nonlinear self-tuning adaptative control, IEEE control system Magazine, pp.44-48

[15] Guyon I., Poujaud I., Personnaz L., Dreyfus G., Denker J., & Le Cun Y. (1989). Comparing different neural networks architectures for classifying handwritten digits2. Int. J. Conf. on Neural Networks, 2, 127-132. Washington, USA

[16] Hoie.J, collaborateurs. (1988). Distant Metastases in Papillary Thyoïd Cancer. Cancer 61 :1-6

[17] Jodouin Jean-François. (1994). Les réseaux neuromimétiques : modèles et applications. Édition Hermes.

[18] Jodouin Jean-François. (1994). Les réseaux de neurones: principes et définitions. Edition Hermes.

[19] K. Hornik. (1988). Approximation capabilities of multilayer feeforward networks. Neural networks, 4 (2), p 251-257.

[20] Kary FRÄMLING. (Juillet 1992). Les réseaux de neurones comme outils d'aide à la décision floue Ecole Nationale Supérieure des Mines de Saint-Etienne. France

[21] Krranowsky.w.j. (25-27.April1995) Discrimination and Classification Using Both binary tree hypothesis. Proc. Of the ECML 95, 8th European Conference on ML, Heraklion, Grete, Greece.

[22] L.L., Alin Morineau, M.P. (juillet1995).Statistique exploratoire multidimensionnelle.

Paris.

[23] Lerman.I.C &Ph. Peter. (juillet 1985) Elaboration et logiciel d'un indice de similarité entre objets d'un type quelconque. Application au problème du consensus en classification. Publication Interne Irisa n.262, Rennes.

[24] Medsker Larry, Liebowitz Jay (1994). Design and Development of Expert Systems and Neural Networks. Édition MacMillan Publishing Company.

[25] Morgan. J.N &J.A. Sonquist. (1963).Problems in analysis of survy data and a prosal. J. Am. Statist.Assoc,58,pp.415-434.

[26] O.Kinouchi.M.H.R. (1996). Tragtenberg, « Modeling neurons by simple maps » international journal of bifurcation and chos, vol.6, N12 A,pp.150-1560

[27] Remm Jean-François *Probabilités et réseaux de neurones*. CRIN-CNRS et INRIA-Lorraine Rapport Technique RR n°2909.

[28]Richard O. Duda, Peter E. Hart , & David G. Stork. (2001). *Pattern Classification.* (Second edition). New York: Wiley-Interscience.

[29] Rival Isabelle. LES Réseaux de Neurones Formels pour le pilotage de Robots Mobiles**,** Laboratoire d'Électronique de l'ESPCI (École Supérieure de Physique et de Chimie Industrielles).

[30] Rivals I., Personnaz L., Dreyfus G., Ploix J.-L. (1995). Modélisation, classification et commande par réseaux de neurones : principes fondamentaux, méthodologie de conception, et illustrations industrielles , in Récents progrès en génie des procédés 9, Lavoisier technique et documentation, Paris.

[31] Rosenblatt,R. (1958).Principles of Neurodynamics. Spartan Books.New-York

[32] S.Haykin.(1994). Neuronal Network. A comprehensive fondation, Macmillan, New York

[33] Tubiana M., collaborators.(1985). Long-term Results and Prognostic Factors in Patient with Differentiated Thyroid carcinoma. Cancer55:794-804.

[34] Zapranis Achilleas, Refenes Apostolos-Paul (1999). Principles of Neural Model Identification, Selection and Adequacy with applications to financial Econometrics. Edition Springer- Verlag London Berlin Heidelberg.

[35] Zurada J.M. (1992). Introduction to artificial neural systems. Minnesota: West publishing company.