Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: *Système d'Information et de Connaissances (S.I.C)*

Thème

Impact de la réduction de la dimension pour l'apprentissage supervisé. Application sur bases médicales

Réalisé par :

- **TALBI Youcef**
- **OUAFI Mohammed**

Présenté le 13 Juin 2013 devant le jury composé de MM.

- *BENMMAMAR Badr* *(Président)*
- *BENAZZOUZ Mourtada* *(Encadreur)*
- *BAGHLI Ismahan* *(Co-Encadreur)*
- *BENMOUNA Youcef* *(Examinateur)*
- *BELABED Amin* *(Examinateur)*

Année universitaire : 2012-2013

Remerciements

Nous aimerons remercier très sincèrement notre encadreur, Monsieur Mourtada BENAZZOUZ, pour le sujet, le temps et l'aide qu'il nous a accordés et ses multiples conseils tout au long de la réalisation de ce travail. Nous le remercie aussi pour son encouragement.

Un grand merci aussi pour notre co-encadreur, Melle BAGHLI Ismahan, pour le temps, l'aide et la patience tout au long de ce travail.

Nous tenons à exprimer remerciements aux membres du jury, qui ont accepté d'évaluer notre travail. Merci au président Mr BENMAMAR Badre, aux examinateurs Mr BEN MOUNA Youcef, Mr BELABED Youcef.

Enfin, nous adressons nos plus sincères remerciements à tous les personnes qui nous ont toujours soutenus et encouragés au cours de la réalisation de ce mémoire.

Merci

Dédicaces

Je dédie ce travail à mon Père, que je lui exprime mes chaleureux remerciements.

J’adresse mes dédicaces aussi à ma Mère, mais aucune dédicace n’est susceptible de vous exprimer ma profonde affection et mon immense gratitude.

Je dédie encore ce travail à mon chère frère RAFIK, mes sœurs KENZA, SABAH, SHARAF, et la princesse ASMAE, et aussi mes oncles Hassen et Mohammed (A.E.K), tout mes cousins surtout MONDJID et mes cousines.

Et enfin, j’adresse mes dédicaces à mon chère ami TALBI Youcef ainsi que les autres BENHAMED Mohammed et KHEDDAM Sidi mohammed el amine.

A tous qui m’aiment.

OUAFI Mohammed

Dédicaces

A ma très chère mère

Tu es l'exemple de dévouement qui n'a pas cessé de m'encourager et de prier pour moi. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mon père

Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A mes frères et sœurs; la petite ange

anfal,meriem ,hassan,chaimae,kawter,mohammed.

Vous vous êtes dépensés pour moi sans conter. En reconnaissances de tous les sacrifices consentis par tous et chacun pour me permettre d'atteindre cette étape de ma vie.

A mes oncles, tantes, cousin et cousines affectueuses reconnaissances.

A mes enseignants de l'écoule primaire jusqu'à l'université dont les conseils précieux m'ont guidée; qu'ils trouvent ici l'expression de ma reconnaissance.

A ma binôme ouafi qui a partagé avec moi les moments difficiles de ce travail et à sa famille.

A mes amis; abd razek, morad,miloud , reda, ismail, khadam, sidi mohammed, et à leurs familles.

Je vous remercie de votre patience vous m'a aidée toujours à avancer vous êtes tous des grandes amies si gentilles, merci d'être toujours près de moi, amies avec lesquelles je souris.

A mes camarades de la faculté des sciences étrangères de l'université et à leurs familles.

Je dédis ce travail

TALBI Youcef

Table des matières :

Table des Figures :

Table des Tableaux :

Introduction générale:

- *Problématique:*

  Dans de nombreux domaines (vision par ordinateur, reconnaissance de formes, etc.), la résolution des problèmes se base sur le traitement de données extraites à partir des données acquises dans le monde réel, et structurées sous forme de vecteurs. La qualité du système de traitement dépend directement du bon choix du contenu de ces vecteurs. Mais dans de nombreux cas, la résolution pratique du problème devient presque impossible à cause de la dimensionnalité trop importante de ces vecteurs.
  Par conséquent, il est souvent utile, et parfois nécessaire, de réduire celle-ci à une taille plus compatible avec les méthodes de résolution, même que souvent cette réduction peut conduire à une légère perte d'informations. Parfois, la résolution de phénomènes complexes avec des descripteurs de grande taille pourrait être gérée en utilisant peu de caractéristiques extraites des données initiales, car la technique de réduction a permis d'extraire les variables pertinentes pour le problème à résoudre.

  Le médecin est confronté à traiter un grand nombre de caractéristiques et établir un diagnostic, cela peut engendrer des faux positifs due à la subjectivité du processus. Les méthodes dites intelligentes, peuvent l'aider dans son diagnostic tout en augmentant sa précision. Pour des données de grande dimension, une étape de réduction est souvent nécessaire pour assurer un bon apprentissage.

- *Contribution:*

  Une méthode de réduction de la dimensionnalité est souvent définie comme un processus de prétraitement de données qui permet de supprimer les informations redondantes et bruitées.
  Les méthodes de réduction de la dimensionnalité sont généralement classées en deux catégories:

- ✓ L'extraction de caractéristiques qui permet de créer de nouveaux ensembles de caractéristiques, en utilisant une combinaison des caractéristiques de l'espace de départ ou plus généralement une transformation effectuant une réduction du nombre de dimensions.

- ✓ La sélection de caractéristiques qui regroupe les algorithmes permettant de sélectionner un sous-ensemble de caractéristiques parmi un ensemble de départ, en utilisant divers critères et différentes méthodes.

Dans ce contexte, nous désirons étudier l'influence de la réduction de la dimensionnalité sur des données médicales (Colon et Madelon), et comparer les performances des techniques d'extraction avec celles de sélection.

- *Organisation du memoire*

Ce manuscrit est organisé comme suit:

- ✓ Chapitre 1 : généralisation sur les réductions de dimension et le principe des méthodes utilisées (ACP, LDA, ReliefF et SFS).

- ✓ Chapitre 2 : présentation générale des réseaux de neurones et en particulier le PMC (Perceptron Multi Couches).

- ✓ Chapitre 3 : Application sur bases médicales

- ✓ Conclusion et perspectives

Chapitre 01

Réduction de la dimensionnalité

I. Introduction

Les algorithmes d'apprentissage artificiel requièrent typiquement peu de traits ou de variables (attributs) très significatifs caractérisant le phénomène étudié. Dans le domaine de la reconnaissance des formes et de la fouille de données, il pourrait encore être bénéfique d'incorporer un module de réduction de la dimension dans le système global avec comme objectif d'enlever toute information inconséquente et redondante. Cela a un effet important sur la performance du système. En effet le nombre de caractéristiques utilisées est directement lié à l'erreur finale. L'importance de chaque caractéristique dépend de la taille de la base d'apprentissage - pour un échantillon de petite taille, l'élimination d'une caractéristique importante peut diminuer l'erreur. Il faut aussi noter que des caractéristiques individuellement peu pertinentes peuvent être très informatives si on les utilise conjointement.

Dans ce chapitre, nous allons définir les objectifs de la réduction de dimension en détaillant les deux principales catégories à savoir l'extraction de caractéristiques et la sélection d'attributs, nous exposerons ensuite les techniques des 2 catégories utilisées dans ce travail et nous finirons par une conclusion.

II. État de l'art

De nos jours, les moyens mis en œuvre pour la caractérisation des données donnent naissance à des bases de grande dimension. Les attributs générés peuvent contenir des informations redondantes, contradictoires, et incohérentes. De ce fait, le besoin de réduire la dimension est né, tout en ayant l'objectif de réduire la redondance et éliminer les contradictions ainsi que les incohérences.

En 1997, et même que les capacités de calcul de l'époque ne permettaient de traiter que des bases de données comportant quelques dizaines ou quelques centaines d'attributs, nous pouvons citer les travaux de [1] et [2].

Guyon et al ont présenté plusieurs travaux sur la réduction de la dimension, en 2003, [3] proposent une vue d'ensemble de la sélection d'attributs. Ils ont conclut alors que les méthodes de sélection d'attributs venaient de franchir une marche quantitative, en se

confrontant à des bases de données comportant plusieurs milliers d'attributs (données génomiques ([4]). Afin de comparer les différentes approches existantes, un concours de sélection d'attributs (sur plusieurs bases, entre autres Madelon) est proposé à l'occasion de la conférence NIPS 2003 [5]. La sélection d'attributs reste un problème ouvert et la taille des bases de données ne cesse d'augmenter. Les données stockées peuvent contenir plusieurs dizaines de milliers d'attributs [6]. De ce constat, un nouveau concours de sélection d'attributs a eu lieu à l'occasion de la conférence KDD 2009 [7].

Tian Lan et al [8] ont travaillé sur la sélection de variable en utilisant l'analyse linéaire en composent indépendante et l'information mutuelle, ils ont testé leur approche sur le signal EGG avec le classiffieur Knn.

B.Chandra et al [9] ont introduit la méthode ERGS (Effective Range based Gene Sélection) son principe est que le meilleur poids est donné à la variable qui discrimine beaucoup plus la classe. L'évaluation a été faite avec Naive bayes et SVM.

Yuhang w. et al [10] ont présenté ReliefF pour la sélection des gênes sur plusieurs bases médicales avec les classifieurs SVM et Knn

Dans [11], les auteurs ont testé sur la base « colon » et ont introduit la méthode BEF (Bayes Error Filter) pour la sélection des variables. Dans [12], la base colon a aussi fait l'objet de test mais la sélection a été faite par les forets aléatoires.

Amel Hafa [13] a expérimenté les méthodes de sélection de variables de nature probabiliste et statistique sur la base « colon »

III. Réduction de dimensionnalité

Les principaux objectifs de la réduction de dimensionnalité sont :

- faciliter la visualisation et la compréhension des données,
- réduire l'espace de stockage nécessaire,

• réduire le temps d'apprentissage et d'utilisation,

• identifier les facteurs pertinents.

Selon que la réduction combine l'espace d'attributs initiale pour avoir un espace plus réduit, ou sélectionne quelques attributs pertinents selon un ou plusieurs critères, deux principales catégories de réduction de la dimension ont vu le jour : l'extraction de caractéristique et la sélection d'attributs.

IV. Extraction de caractéristiques

L'extraction vise à sélectionner des caractéristiques dans un espace transformé dans un espace de projection (figure 1)

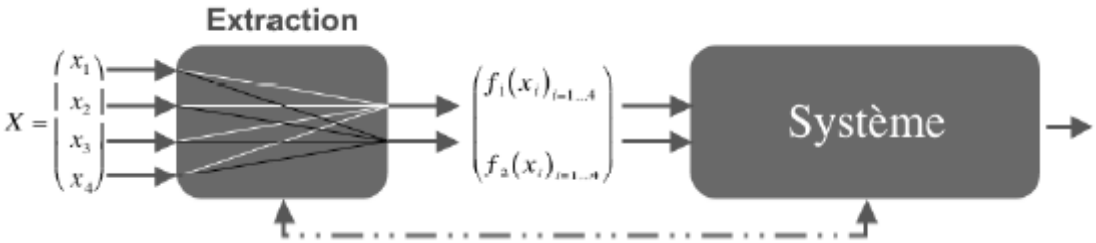

Figure 1. : Schéma d'extraction de caractéristiques

Théoriquement, une méthode d'extraction consiste en la recherche de *m* paramètres en fonction des *n* paramètres initiaux ($m << n$) . Ces *m* paramètres seront en général calculés à partir de combinaisons linéaires des *n* paramètres initiaux. Les points sont alors projetés dans un sous-espace R^m. Cependant, le nombre de paramètres à calculer pour caractériser chaque observation sera toujours *n* mais l'interprétation des observations se fera alors dans ce sous-espace R^m. Par ailleurs, le sens des paramètres peut être perdu.

Dans ce présent mémoire, nous nous sommes intéressés à deux techniques l'une non supervisée l'ACP (Analyse Composante Principale), et l'autre supervisée l'ADL ou LDA (Analyse Discriminante Linéaire).

1. Analyse Composante principale

Définition : L'Analyse en Composantes Principales (ACP) est l'une des méthodes d'analyse de données multi-variées les plus utilisées. Dès lors que l'on dispose d'un tableau de données quantitatives (continues ou discrètes) dans lequel n observations (des individus, des produits, …) sont décrites par p variables (des descripteurs, attributs, mesures, …), si p est assez élevé, il est impossible d'appréhender la structure des données et la proximité entre les observations en se contentant d'analyser des statistiques descriptives uni-variées ou même une matrice de corrélation.

Mathématiquement, l'analyse en composantes principales est un simple changement de base : passer d'une représentation dans la base canonique des variables initiales à une représentation dans la base des facteurs définis par les vecteurs propres de la matrice des corrélations.

Principe de l'ACP : L'ACP consiste à remplacer une famille de variables par de nouvelles variables de variance maximale, non corrélées deux à deux et qui sont des combinaisons linéaires des variables d'origine. Ces nouvelles variables, appelées *composantes principales*,

*Le calcul d'une matrice de covariance à partir du tableau des données

*La diagonalisation de cette matrice symétrique positive et l'extraction des vecteurs propres : u1;u2;…

Nous considérons une transformation linéaire d un vecteur M $\in R^n$ avec la moyenne et la matrice de covariance$\sum m$ à un vecteur de dimension inferieur W$\in R^q$, q$< n$

$$W=A_q^T M \qquad (1)$$

Avec $A_q^T A_q = I_q$ ou I_q est la matrice d'identité q×q .

En ACP, A_q est une matrice $n \times$ q : les colonnes sont les q vecteurs propres orthonormaux correspondant aux premiers q valeur propres larges de la matrice de covariance $\sum m$

2. Analyse Discriminante Linéaire

L'analyse linéaire discriminante, est une méthode de réduction du nombre de dimensions proposée. Cette méthode s'applique lorsque les classes des individus sont connues. L'idée a été de créer une méthode pour choisir entre les combinaisons linéaires des variables celles qui maximisent l'homogénéité de chaque classe. En d'autres termes, cette méthode consiste à chercher un espace vectoriel de faible dimension qui maximise la variance interclasse (pour une description complète de la méthode) cette technique est supervisée, contrairement à l'ACP.

L'approche supervisée pour une réduction linéaire de la dimensionnalité est basée sur l'ADL. Cette approche définit la matrice de transformation optimale L qui maximise le critèrede Fisher J ([14], [15], [16]):

$$L=argmax_A\, J(A) \qquad (2)$$

Avec

$$J(A)=tr((AS_WA^t)^{-1}AS_BA^t) \qquad (3)$$

S_W est la matrice de covariance moyenne intra-classe et S_B est la matrice de covariance inter-classe La matrice S_W $(n \times n)$ est une matrice moyenne pondérée des matrices de covariance des classes et décrit la (co)variance qui est (en moyenne) présente dans chaque classe. Le matrice S_B $(n \times n)$ décrit la covariance entre plusieurs classes.

Dans l'´equation 3, AS_WA^t et AS_BA^t sont des matrices $d \times d$ de covariance intra-classe et inter-classe des vecteurs caractéristiques après la réduction de la dimensionnalité des données (**d**) à l'aide de la transformée linéaire $\boldsymbol{A}$. Lors de la maximisation de l'équation 2, on minimise simultanément la covariance intra-classe et maximise la covariance inter-classe dans un espace de dimension inferieure qui est engendrée par les lignes de $\boldsymbol{A}$. Le critère cherche à déterminer une transformation L dont la projection des vecteurs caractéristiques appartenant à une même classe soit la plus proche possible entre eux, tout en essayant de maintenir aussi loin que possible les vecteurs qui ne font pas partie de la même classe. La matrice optimale, telle que définie par l'équation 3, est la transformée associée à l'ADL. Une fois les matrices de covariances S_W et S_B ont été estimées à partir des données d'apprentissage, le problème de maximisation de l'équation 3, peut être résolu au moyen d'une décomposition aux valeurs propres, avec

la maximisation du quotient Rayleigh des matrices S_B et S_W ([14], [15], [16],[17]). Le problème des valeurs propres à résoudre est :

$$\boldsymbol{S_B V = S_W V \Lambda} \qquad (4)$$

Ou équivalente à :

$$\boldsymbol{S_W^{-1} S_B V = V \Lambda} \qquad (5)$$

Dans laquelle V est une matrice $n \times n$ des vecteurs propres (**n** comme vecteurs de colonnes) et Λ est une matrice diagonale $n \times n$ avec **n** valeurs propres $\succ_i$ associe aux vecteurs propres v_i de V. La matrice de transformation **L** de dimension $d \times n$ qui maximise le critère de Fisher est obtenue en définissant les lignes de **L** comme la transposée des vecteurs propres v_i^t correspondant aux plus grandes valeurs propres.

3. Comparaison entre ACP et ADL

Toutes ces techniques permettent de passer outre la malédiction de la dimension. Cependant, si le nombre de dimensions est très grand (plus de 512), il se peut que la réduction ne permette pas de réduire suffisamment le nombre de dimensions de l'espace sans perdre de l'information. De plus, ces techniques nécessitent de recalculer les matrices, si l'on ajoute de nouvelles données. Ensuite, elles sont très sensibles aux valeurs aberrantes, c'est pourquoi il est préférable de détecter ces erreurs, puis de normaliser les données avant d'effectuer ces opérations. Lorsque l'ensemble d'apprentissage est petit, l'ACP donne de meilleurs résultats que la ADL, et est moins sensible aux données d'apprentissage [18]. Lorsque le nombre de dimensions est très grand, l'ADL nécessite de longs calculs [19]. Pour éviter ce problème, une technique appelée «ACP plus ADL» [20] propose de combiner les deux opérations en utilisant certaines de leurs caractéristiques.

V. Sélection d'attributs

La sélection consiste à choisir des attributs selon un ou plusieurs critères de pertinence dans l'espace de mesure (figure 2)

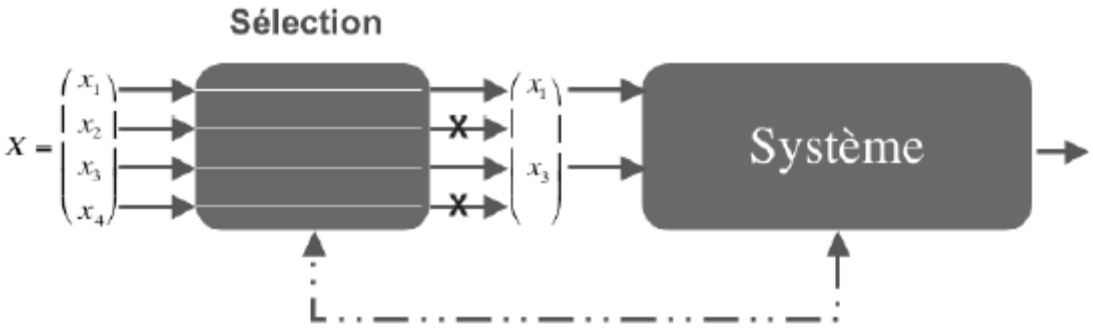

Figure 2 Schéma de la sélection d'attributs

On distingue, deux catégories de sélection les filtres et les wrapper :

1) Filter :

Le modèle "filter" a été le premier à utiliser pour la sélection de caractéristiques. Dans celui-ci, le critère d'évaluation utilise évalue la pertinence d'une caractéristique selon des mesures qui reposent sur les propriétés des données d'apprentissage. Cette méthode est considérée, davantage comme une étape de prétraitement (filtrage) avant la phase d'apprentissage.

En d'autres termes, l'évaluation se fait généralement indépendamment d'un classificateur [21]. Les méthodes qui se basent sur ce modèle pour l'évaluation des caractéristiques, utilisent souvent une approche heuristique comme stratégie de recherche

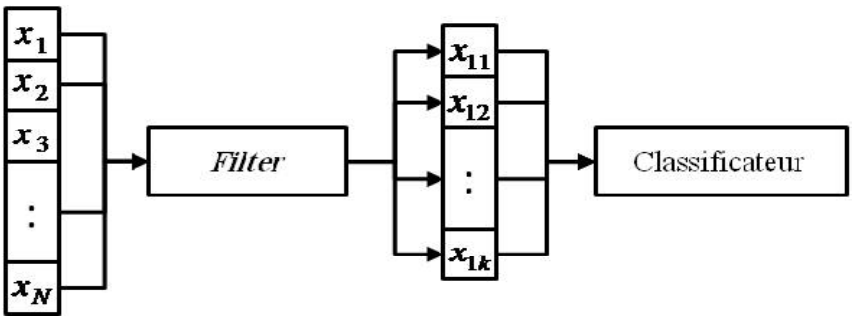

Figure 3. La procédure du modèle "filter"

2) Wrapper :

Le principal inconvénient des approches "filter" est le fait qu'elles ignorent l'influence des Caractéristiques sélectionnées sur la performance du classificateur à utiliser par la suite.

Pour résoudre ce problème, Kohavi et John ont introduit le concept "wrapper" pour la sélection de caractéristiques [22]. Les méthodes "wrapper", appelées aussi méthodes enveloppantes, évaluent un sous-ensemble de caractéristiques par sa performance de classification en utilisant un algorithme d'apprentissage.

L'évaluation se fait à l'aide d'un classificateur qui estime la pertinence d'un sous-ensemble donné de caractéristiques.

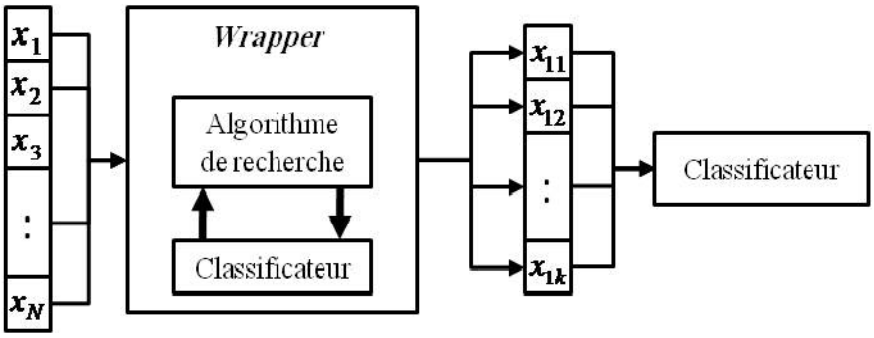

Figure 4. La procédure du modèle "wrapper"

- **Pertinence d'attributs**

Les auteurs définissent la pertinence dans le cas d'attributs et de fonctions booléennes, et en supposant que les données sont non bruitées. Une définition plus large définit les attributs pertinents comme ceux dont les valeurs varient systématiquement avec les valeurs de classe. Autrement dit, un attribut Fi est pertinent si connaître sa valeur change les probabilités sur les valeurs de la classe C

1. ReliefF

Une des méthodes de filtrage les plus connues pour la sélection des attributs est la méthode Relief. Cette méthode fut proposée en 1992 par Kira et Rendell [23]. Son principe est de calculer une mesure globale de la pertinence des caractéristiques en accumulant la différence des distances entre des exemples d'apprentissage choisis aléatoirement et leurs plus proches voisins de la même classe et de l'autre classe. La simplicité, la facilite de la mise en ouvre ainsi que la précision même sur des données bruitées, représentent les avantages de cette méthode. En revanche, sa technique aléatoire ne peut pas garantir la cohérence des résultats lorsqu'on applique plusieurs fois la méthode sur les mêmes données. Par ailleurs, cette méthode ne prend pas en compte la corrélation éventuelle entre les caractéristiques. Afin d'éviter le caractère aléatoire de l'algorithme, John et al. [24] ont proposé une version déterministe appelée ReliefD. D'autres variantes de cet algorithme, ReliefF, pour améliorer sa performance, sa vitesse ou les deux, ont été proposées dans [25] et [26].

Algorithm1 Algorithme de sélection RelifF

1 : initialiser les poids

2 : tirer aléatoirement une donnée X_i

3 : Trouver les K plus proche voisin de X_i ayant les même étiquettes (hits),

4 :T rouver les K plus proche voisin de X_i ayant une étiquette différente de la classe de X_i (misses)

5 : Pour chaque caractéristiques mettre à jour les poids

$$W_d = w_d - \sum_{j=1}^{K} \frac{diff(x_i, d_i, hits_j)}{m*k} + \sum_{cclas(x_i)} \left(\frac{p(c)}{1-p(class(x_i))}\right) \sum_{j=1}^{k} \frac{diff\,(x_i, d_i, misses_j)}{m*k}$$

6: La distance utilisée est définie par:

$$diff(x_i, d_i, x_j) = \frac{\|x_i d\ x_j d\|}{\max(d)\ \min(d)}$$

2. Sequential Forward Selection (SFS)

SFS (Sequential Forward Selection) ou (sélection séquentielle croissante) est la première méthode proposée pour la sélection de caractéristiques. Cette méthode a été proposée en 1963 par Marill et Green [27]. Une approche heuristique de recherche est utilisée dans cette méthode, en commençant par un ensemble vide de caractéristiques chaque itération, la meilleure caractéristique parmi celles qui restent sera sélectionnée, supprimée de l'ensemble de départ et ajoutée au sous-ensemble des caractéristiques sélectionnées (Algorithme 2.1). Le processus de sélection continue jusqu'a un critère d'arrêt

Algo2.1 : Algorithme SFS

Entrées:

$F = \{f_1, f_2, .., f_n\}$

M: taille de l ensemble final

Sorties : $\boldsymbol{E = \{f_{s1}, f_{s2}, .., f_{sM}\}}$

$E = \emptyset$

Pour $\boldsymbol{i = 1}$ à M **Faire**

Evaluer $f_j \cup E$

Fin pour

$\boldsymbol{f_{max} = meilleur\ f_j}$

$\boldsymbol{E = E \cup f_{max}, F = F \setminus j_{max}}$

Fin Pour

Retourner E

VI. Conclusion

Dans ce chapitre, nous avons abordé le domaine de la réduction de dimension. Nous avons exposé les méthodes que nous avons utilisé dans nos expérimentations ; ACP et ADL (méthodes d'extraction) et ReliefF et SFS (méthodes de sélection). Nous passerons dans le chapitre suivant aux réseaux de neurones et particulièrement le PMC pour ensuite voir l'impact de la réduction sur son apprentissage.

.

Chapitre 02

Les Réseaux de Neurones

I. Introduction :

Depuis les années 40, les savants reconnaissent et voient de plus prés la perfection du bon Dieu, en déchiffrant le cerveau humain et en essayant de comprendre sa grande complexité et ses mécanismes afin de construire une machine intelligente capable de reproduire les comportements humains par une imitation de son cerveau.

Grace à l'obstination des scientifiques, la communauté scientifique assiste à une mutation conceptuelle dans ce domaine. Cette dernière fut possible grâce aux efforts conjugués de la biologie, des sciences cognitives et des sciences de l'ingénieur.

Dans ce chapitre, nous allons tracer un historique, la définition biologique et formel du neurone et les points essentiels sur les réseaux de neurones afin de montrer les capacités d'apprentissages d'une architecture de référence: le Perceptron Multicouche 'PMC', puisque nous l'avons utilisé dans notre travail de comparaison de techniques de réduction de la dimension.

II. Historique :

Le champ des réseaux de neurones remonte à 1943 avec les travaux de W.McCulloch et W.Pitts [28], qui ont donné naissance au neurone formel (une abstraction du neurone physiologique). Ils veulent démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques.

- ✓ En 1949, D. Hebb [29] présente dans son ouvrage « The Organisation of Bihavior » une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb.

- ✓ En 1958, F. Rosenblatt [30] développe le modèle du perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perceptron et une couche liée à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience.

- ✓ Dans la même période ; le modèle de l'Adaline (ADAptive LINar Element) a été présenté par B. Windrow et Hoff [31]. Ce modèle sera par la suite le modèle de base des réseaux de multicouches.

- ✓ En 1969, M. Minsky et S. Papert [32] publient une critique des propriétés du perceptron. Cela a freiné la recherche dans ce domaine jusqu'en 1972, où T. kohonen [33] présente ses travaux sur les mémoires associatives et propose des applications à la reconnaissance de formes.

- ✓ C'est en 1982 que J. Hopfield [34] présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique.

- ✓ En 1983, le chercheurOakley [35] développe la machine de Boltzmann, qui est le premier modèle connu apte à traiter de manière satisfaisante les limitations recensées dans le cas de perceptron.

- ✓ En 1984, les cartes auto-organisatrices de Kohennen [36].

- ✓ En 1985, apparait la rétro-propagation du gradient proposé entre autre par fogelman [37], c'est un algorithme d'apprentissage adapté au réseau de neurones multicouches MLP. Dés cette découverte, nous avons la possibilité de réaliser une fonction non linéaire d'entrée/sortie sur un réseau en décomposant cette fonction en une suite d'étapes linéairement séparables.

- ✓ En 1986, David Rumelhart, Hinton et William [38], généralisèrent la loi delta et développèrent une méthode efficace d'entrainement des réseaux multicouches.

- ✓ En 1996, Steve Lawrence et Andrew D.Back [39] introduisirent un nouveau type de réseau de neurone appelé Gamma MLP.

III. Le neurone biologique

III-1 Définition du neurone :

Le neurone est un type de cellule constituant l'unité fonctionnelle du système nerveux. Les neurones sont 10 à 50 fois moins nombreux que les cellules gliales, seconds composants du tissu nerveux assurant plusieurs fonctions dont le soutien et la nutrition des neurones.

On estime que le cerveau humain comprend environ 100 milliards (10^{11}) de neurones. Les neurones assurent la transmission d'un signal que l'on nomme influx nerveux.

Le neurone est composé d'un corps cellulaire, et de deux types de prolongements : l'axone qui conduit « l'influx nerveux » de manière centrifuge et le ou les dendrites [40].

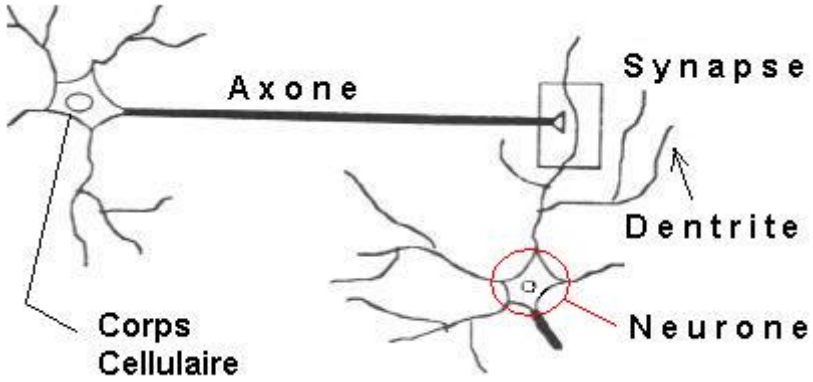

Figure 5 : Neurone Biologique.

- **Les dendrites** : sont nombreuses (environ 100 000), courtes et ramifiées dés leur origine. Elles sont parfois recouvertes d'épines dendritiques. Elles forment ma porte d'entrée des informations des neurones en collectant les informations en provenance des autres neurones. C'est au niveau de ces prolongements que viennent majoritairement se connecter les axones des autres cellules nerveuses.
- **Le corps cellulaire ou « soma »** : il contient le noyau et la plupart des organites cytoplasmique nécessaire à la survie et au fonctionnement du neurone, il traite ces entrées et renvoie une impulsion en sortie.

- **L'axone** : le neurone ne possède qu'un seul axone et il correspond à la porte de sortie des informations. Il décrit un trajet log avant de se terminer en se ramifiant. Ces derniers se connectent aux dendrites d'autres cellules, la connexion entre deux neurones est appelée 'synapse'.
- **Les synapses** : ils permettent aux cellules nerveuses de communiquer entre elles.

IV. Le neurone formel

IV-1 Définition :

Un neurone formel est une représentation mathématique et informatique du neurone biologique (voire fig. 1). Il produit certaine caractéristique biologique, en particulier les dendrites, axone et synapse, au moyen de fonctions et de valeurs numériques. Les neurones formels sont regroupés en réseaux de neurones.

Grace à des algorithmes d'apprentissage automatique, il est possible de régler un réseau de neurone pour lui faire accomplir des taches qui relèvent de l'intelligence artificielle.
La figure ci-dessous montre la structure d'un neurone artificiel. Chaque neurone artificiel est un neurone élémentaire [41].

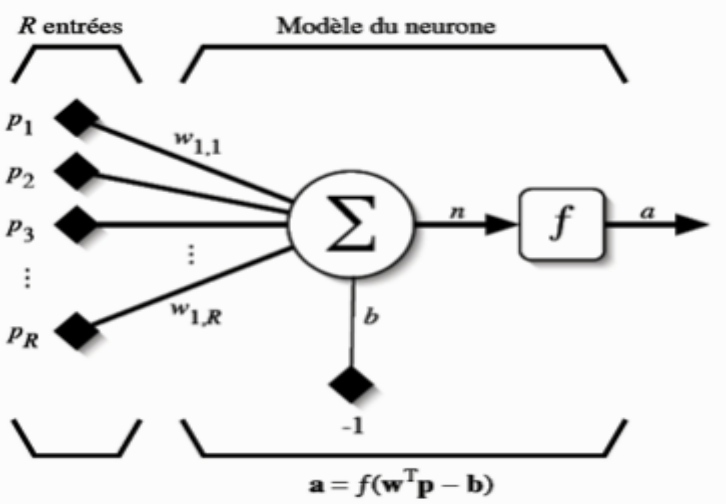

Figure 6 : Le Neurone Formel

Il reçoit un nombre variable « d'entrées R » en provenance de neurone amonts.

A chaque une de ces entrées est associé un « poids W » représentatif de la force de la connexion.

Un neurone est essentiellement constitué d'un intégrateur qui effectue la somme pondérée de ses entrées. Le résultat « résultat n » de cette somme est ensuite transformé par une « fonction de transfert f » qui produit la « sortie a » du neurone.

Les « R » entrées du neurone correspondent au vecteur P=$[p_1, p_{2,\dots,} p_R]^T$ alors que W=$[w_{1.1}, w_{1.2,\dots,} w_{1.R}]^T$ représente le vecteur des poids du neurone. La sortie n de l'intégrateur est donnée par l'équation suivante :

$$n = \sum_{j=1}^{R} w_{1.j} p_1 - b \quad (1)$$

Cette sortie correspondent à une somme pondérée des poids et des entrées moins ce qu'on nomme le « biais b » du neurone. Le résultat « n » de la somme pondérée s'appelle le « niveau d'activation du neurone ». Lorsque le niveau d'activation atteint ou dépasse le seuil « b », alors l'argument de « f » devient positif (ou nul). Sinon, il est négatif.

Il existe plusieurs possibilités pour les différentes fonctions de transfert (fonction d'activation du neurone). Le tableau suivant énumère les principales fonctions [42]:

Nom de la fonction	Relation d'entrée/sortie	Icône
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
linéaire	$a = n$	
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
sigmoïde	$a = \frac{1}{1+\exp^{-n}}$	
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
compétitive	$a = 1$ si n maximum $a = 0$ autrement	

Tableau 1 : Tableau représentatif des fonctions d'activations les plus utilisées.

V. Le mécanisme de l'apprentissage

L'apprentissage comprend généralement quatre étapes de calcul :

1. Initialisation des poids synaptique du réseau ; souvent ces poids sont initialisés à des valeurs aléatoires.
2. Présentation du « patron d'entrée » (vecteur caractéristique des informations à traiter) et la propagation d'activation.
3. Calcul de l'erreur : elle est calculée à partir de l'activation d'un neurone et de celles des neurones qui lui sont reliés. Dans le cas d'un apprentissage supervisé ; cette erreur sera la différence entre le « patron de sortie » et le « patron de référence ».
4. Calcul du vecteur de correction ; à partir des valeurs d'erreurs, on détermine alors la correction à apporter aux poids synaptiques des connexions.

Ce mécanisme d'apprentissage sera répété plusieurs fois jusqu'à l'obtention du comportement désiré.

VI. Calcule des poids synaptiques :

La rétro-propagation est une méthode de calcul des poids (aussi appelés 'poids synaptiques') pour un réseau à apprentissage supervisé qui consiste à minimiser l'erreur quadratique de sortie (somme des carrés de l'erreur de chaque composante entre la sortie réelle et la sortie désirée).

L'optimisation de l'algorithme de rétro-propagation du gradient rencontre les difficultés suivantes :

- ✓ La sélection d'une architecture de modèle de réseau adapté aux problèmes à résoudre en termes de complexité. Autrement dit le manque d'éléments théoriques permettant de relier d'une part le nombre de couches cachées et le nombre de neurones par couches (paramètres architecturaux) et d'autre part le type et la complexité du problème à traiter.
- ✓ La tache de trouver un nombre suffisant de données et de leurs associées à chacune une valeur désirée sont des taches qui sont souvent difficiles.
- ✓ Le comportement des réseaux est gouverné par un ensemble de paramètres d'apprentissage, pour lesquels un mauvais choix peut compromettre l'apprentissage.
- ✓ La plupart des applications utilisant les techniques connexionnistes montrent que cette approche est très couteuse en temps pendant l'apprentissage. Ainsi, plus le réseau possède de poids et de neurones, plus le temps de calcul, est grand lors de son utilisation.
- ✓ Les réseaux multicouches forment des surfaces d'erreurs complexes qui contiennent souvent plusieurs minimums locaux : La rétro-propagation de base utilise le gradient de l'erreur global obtenue avec tous les exemples, cette méthode ne garantit pas l'obtention du minimum global de la fonction à optimiser, car elle présente l'inconvénient de s'arrêter au premier minimum local rencontré.

VII. Le Perceptron Multi Couche (PMC) :

De tous les réseaux de neurones artificiels qui réalisent un apprentissage supervisé des connaissances, le Perceptron Multicouches est le classifieur par excellence, le plus puissant et le plus populaire.

Le perceptron est un réseau à la base des méthodes connexionnistes, c'est un réseau orienté, apparu en 1985 [43], bien qu'il est souvent qualifié de boite noire qui s'avère

difficile à interpréter, implémenter et à régler, il a pris une place indéniable dans le domaine de la reconnaissance, grâce à la découverte de l'algorithme rétro-propagation de gradient.

VII-1 L'architecture :

Comme son nom l'indique, les neurones du PMC sont réparties en couches, chaque neurone est connecté à toutes les sorties des neurones de la couches précédente, et nourrit de sa sortie tous les neurones de la couche suivante. On distingue les neurones de la première couche (entrée), de la dernière couche (sortie) et des couches intermédiaires (cachées)[43]. (sur la figure 4, I: couche d'entrée, J: couche cachée et K: couche de sortie)

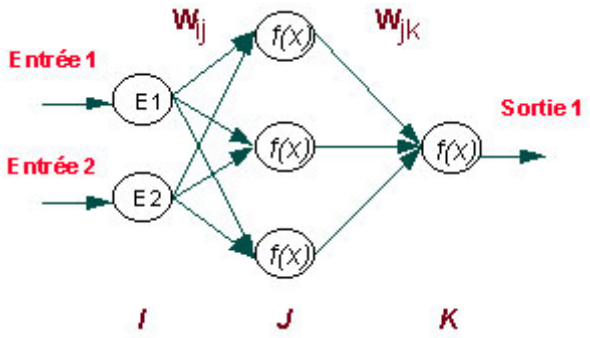

Figure 7 : exemple d'un perceptron multicouche.

- ✓ Les fonctions d'entrées et de transferts sont les mêmes pour les neurones d'une même couche, mais peuvent différer selon la couche. Ainsi la fonction de transfert de la couche de sortie est généralement identité.
- ✓ Le nombre de couche cachés détermine la complexité des frontières des différents sous-espaces que le réseau pourra représenter.
- ✓ La complexité de l'approximation est également détermine le nombre maximal d'information que le réseau peut extraire du signal traité.

VII -2 L'apprentissage du PMC:

L'apprentissage du perceptron multicouche se fait en deux étapes:

1. **Propagation :** qui consiste à présenter une configuration d'entrée au réseau, puis à la propager à celle de sortie en passant par les couches cachées.
2. **Rétro-propagation** : qui consiste, après le processus de propagation, à minimiser l'erreur commise sur l'ensemble des exemples présentés, erreur considérée comme une fonction des poids synaptiques. Cette erreur représente la somme des différences au carré entre les réponses calculées et celles désirées pour tous les exemples contenus dans l'ensemble de l'apprentissage.

L'apprentissage d'un PMC [44] est défini comme un problème d'optimisation qui consiste à trouver les coefficients du réseau minimisant une fonction d'erreur globale (fonction de cout). La définition de cette fonction de cout est primordiale, car celle-ci, sert à mesurer l'écart entre les sorties désirées du modèle et les sorties du réseau observées. La fonction la plus couramment utilisée est la fonction dite fonction d'erreur quadratique, dont la définition est :

Pour chaque exemple 'n'on calcule une fonction d'erreur quadratique :

$$e(n) = \frac{1}{2}\sum_{j=1}^{x}\left[d_j(n) - y_j(n)\right]^2 \quad (2)$$

Pour tout l'ensemble d'apprentissage 'N' on peut définir la fonction de cout (appelé aussi l'erreur quadratique moyenne EQM) :

$$E(n) = \frac{1}{N}\sum_{n=1}^{N} e(n) \qquad (3)$$

VII -3 Rétro-propagation de gradient :

La création d'un perceptron multicouche pour résoudre un problème donné passe donc par l'inférence de la meilleure application possible telle que définie par un ensemble de données d'apprentissage constituées de paires de vecteurs d'entrées et de sorties désirées.

Cette inférence peut se faire, entre autre, par l'algorithme dit de rétro-propagation [45].

Soit le couple ($\vec{x}$(n), $\vec{d}$(n)) désignant la « $n^{ème}$ » donnée d'entrainement du réseau ou :

$$\vec{x}(n)=< x_1(n), \dots, x_p(n) > \text{ et } \vec{d}(n)=< d_1(n), \dots, d_q(n) > \qquad (4)$$

Correspondent respectivement aux « p » entrées et aux « q » sorties désirées du système. L'algorithme de rétro-propagation consiste alors à mesurer l'erreur entre les sorties désirées « $\vec{d}$(n) » et les sorties observées « $\vec{y}$(n) » :

$$\vec{y}(n)=< y_1(n), \dots, y_q(n) > \qquad (5)$$

Résultant de propagation vers l'avant des entrées « $\vec{x}$(n) », et à rétropropager cette erreur à travers les couches du réseau en allant des sorties vers entrées.

a) Cas de la couche de sortie :

L'algorithme de rétro-propagation procède à l'adaptation des poids neurone par neurone en commençant par la couche de sortie. Soit l'erreur observée « $e_j(n)$ » pour le neurone de sortie « j » et la donnée d'entrainement « n » :

$$e_j(n) = d_j(n) - y_j(n) \quad (6)$$

Ou « $d_j(n)$ » correspond à la sortie désirée du neurone « j » et « $y_j(n)$ » à sa sortie observée.

- ✓ La variable 'n' représentera toujours la donnée d'entrainement c'est-à-dire le couple contenant un vecteur d'entrées et un vecteur de sorties désirées.
- ✓ L'objectif de l'algorithme est d'adapter les poids des connexions du réseau de manière à minimiser la somme des erreurs sur tous les neurones de sortie.
- ✓ L'indice 'j' représentera toujours les neurones pour lequel on veut adapter les poids.

Soit « E(n) » la somme des erreurs quadratiques observées sur l'ensemble « C » des neurones de sortie :

$$E(n) = \frac{1}{2}\sum j \in c\, e_j^2(n) \quad (7)$$

La sortie « $y_j(n)$ » du neurone « j » est définie par :

$$y_j(n) = p\left[v_j(n)\right] = p[\sum_{i=0}^{r} w_{ji}(n)\, y_j(n)] \quad (8)$$

Ou « p » est la fonction d'activation du neurone, « $v_j(n)$ » est la somme pondérée des entrées du neurone « j », « $w_{ji}(n)$ » est le poids de la connexion entre le neurone « i » de la couche précédente et le neurone « j » de la couche courante, et «$y_i(n)$ » est la sortie du neurone « i ».

Il est convenu ici que la couche précédente contient « r » neurone numérotés de « 0 » à « r », que le poids « $w_{j0}(n)$» correspond au biais du neurone « j » et que l'entrée $y_0(n) = -1$

L'indice « i » représentera toujours un neurone sur la couche précédente par rapport au neurone « j » ; on suppose par ailleurs que cette couche contient « r » neurones.

Pour corriger l'erreur observée, il s'agit de modifier le poids «$w_{ji}(n)$» dans le sens opposé au gradient « $\frac{\partial E(n)}{\partial w_{ji}(n)}$ » de l'erreur ; comme le montre la figure ci-dessous.

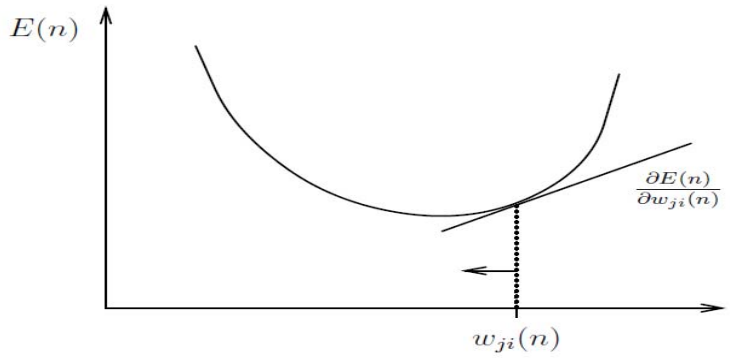

Figure 8 : graphique du gradient de l'erreur totale.

b) Cas d'une couche cachée :

Considérons maintenant le cas des neurones sur la dernière couche cachée (le cas des autres couches cachées est semblable).

- ✓ La variable « n » désignera toujours la donnée d'entrainement c'est-à-dire un couple de vecteurs d'entrées et de sorties désirées.
- ✓ L'objectif sera toujours d'adapter les poids de la couche courante en minimisant la somme des erreurs sur les neurones de la couche de sortie.
- ✓ Les indices « i » et « j » désigneront respectivement (comme précédemment) un neurone sur la couche précédente et un neurone sur la couche courante.
- ✓ L'indice « k » servira maintenant à désigner un neurone sur la couche suivante.

VII –4 L'algorithme de la rétro-propagation:

1. Initialisation aléatoire des pondérations «w_{ji}» dans l'intervalle [0,+1]
2. Introduction du vecteur de la n^{eme} données d'entrainement $\vec{x}$(n) =< x_1(n),…, x_p(n)> à la couche d'entrée
3. Calcul des entrées des neurones de la couche cachée :

$$v_j^c(n) = \sum_{i=0}^{p} w_{ji}^c(n) x_i(n) \quad (9)$$

On a suppose ici que la couche d'entrée contient « p » neurone numérotés de « 1 » à « p », que le poids «$w_{j0}(n)$» correspond au biais du neurone « j » et que l'entrée x_0(n)=-1 ce qui revient à dire que le premier terme de la sommation va représenter le biais.

4. Calcul des sorties des neurones de la couche cachée :

$$y_j^c(n) = \varphi(v_j^c(n)) = \sum_{i=0}^{p} w_{ji}^c(n) x_i(n) \quad (10)$$

5. Passer à la couche de sortie et calculer les entrées de chaque neurone :

$$v_K^s(n) = \sum_{i=0}^{r} w_{Ki}^s(n) y_{Kj}^s(n) \quad (11)$$

6. Calcul des sorties des neurones de la couche de sortie :

$$y_K^S(n) = \varphi\left(v_K^S(n)\right) = \varphi\left(\sum_{i=0}^{r} w_{Ki}^S(n) y_{Kj}^S(n) \right) \quad (12)$$

7. Calcul de l'erreur de la couche de sortie :

$$\delta_K^S(n) = [d_K(n) - y_K^S(n)]\varphi'[v_K^S(n)] \quad (13)$$

Tel que $\vec{d}$(n) =$< d_1(n), \dots, d_q(n) >$ correspond à la sortie désirée.

8. Calcul de l'erreur de la couche cachée :

$$\delta_j^c(n) = \varphi'\left[v_j^c(n)\right] \sum_{k=1}^{q} w_{kj}^s(n) \delta_k^s(n) \quad (14)$$

L'erreur de la couche cachée est calculée avant la modification des poids de la couche de sortie.

9. Modification des poids de la couche de sortie :

$$w_{kj}^s(n+1) = w_{kj}^s(n) + \mu \delta_k^s(n) y_j^c(n) \quad (15)$$

10. Modification des poids de la couche cachée :

$$w_{kj}^s(n+1) = w_{ji}^c(n) + \mu \delta_j^c(n) x_i(n) \quad (16)$$

L'ordre de modification des poids d'une même couche n'est pas important

11. Calcul de l'erreur :

$$E(n) = \frac{1}{2} \sum_{k=1}^{q} [d_k(n) - y_k^s(n)]^2 \quad (17)$$

12. Itérez l'algorithme jusqu'à l'obtention de :

$$E(n) \leq E_{seuil}(n) \qquad (18)$$

VII -5 Avanatges et inconvénients du PMC :

✓ **Avantages**

⇒ Un classifieur très précis (s'il est bien paramétré).

⇒ Apprentissage automatique des poids.

⇒ Possibilité de faire le parallélisme (les éléments de chaque couche peuvent fonctionner en parallèle).

⇒ Résistance aux pannes (si un neurone ne fonctionne plus, le réseau ne se perturbe pas).

✓ **Inconvénients**

⇒ Détermination de l'architecture du réseau est complexe.

⇒ Paramètres difficiles à interpréter (boite noire).

⇒ Difficulté de paramétrage surtout pour le nombre de neurone dans la couche cachée.

VIII. Conclusion :

Dans ce chapitre, nous avons abordé des généralités sur les réseaux de neurones et nous nous sommes particulièrement intéressés par le Perceptron Multi-Couche (PMC), vu que nous l'avons choisit comme classifieur pour voir l'impact de la réduction sur son pouvoir d'apprentissage.

Chapitre 03

Application sur des bases médicales

I. Préliminaire

Après avoir donné les notions théoriques sur la réduction de la dimension et les réseaux de neurones dans le chapitre 1 et le chapitre 2 respectivement. Nous passerons dans ce chapitre aux expérimentations réalisées.

Nous commencerons, dans la section suivante, par décrire les bases médicales utilisées à savoir la base « Colon » et la base « Madelon », nous présenterons après, dans la section III, les performances obtenues des différentes réalisations, nous discuterons les résultats dans la section IV et nous terminerons par une conclusion.

II. Description des Bases

1. Colon

C'est une base médicale représente les cas des personnes qu'ils ont un cancer de colon ou pas, elle contient 2000 attributs et 62 exemples [46].

2. Madelon

C'est une base représente les cas des personnes qu'ils ont un cancer de prostate. Il s'agit d'un problème de classification à deux classes avec entrée continue Variables, elle contient 500 attributs et 2000 exemples [47].

III. Expérimentation

- ✓ Pour tester nos techniques de réduction, nous avons construit l'ensemble d'apprentissage et l'ensemble de test comme suit :
  1. Base ***colon*** : nous l'avons divisé en 2 ensembles, l'un pour l'apprentissage et l'autre pour le test (soit 31 exemples pour chaque ensemble).
  2. Base ***Madelon*** : nous avons 2000 exemples pour l'ensemble d'apprentissage et 600 exemples pour le test.

- ✓ Les différents PMCs ont été construits via Matlab (2011b) à l'aide de la commande «feedforwardnet » avec une seule couche cachée de 10 neurones.

- ✓ Après la préparation des données, nous avons réduit les ensembles et lancer les différents PMCs comme suit :

a) ACP

1. Pour la base Colon

- Calculer les composantes principales (obtention d'une matrice de 2000X2000)
- Définir la matrice de passage (matrice de transformation=2000Xn) (n indique la nouvelle dimension, dans ce travail, nous avons pris n=15, 30 et 45)
- Lancer l'apprentissage de 3 PMCs sur les 3 nouveaux ensembles réduits.
- Tester les architectures PMCs obtenus sur les ensembles de tests (les 3 réductions réalisées)

Cela, nous a permis d'obtenir les performances suivantes pour les 3 réductions (dans les tableaux de performances, Taux=taux de reconnaissance, Se=sensibilité, Sp=spécificité et Préc= précision) :

composantes principales		Taux	Se	Sp	Préc
15	Apprentissage	90,32%	90,91%	90,00%	83,33%
	Test	74,19%	81,82%	70,00%	60,15%

Composantes principales		Taux	Se	Sp	Préc
30	Apprentissage	93,55%	90,91%	95,00%	90,91%
	Test	83,87%	72,73%	90,00%	80,00%

Composantes principales		Taux	Se	Sp	Préc
45	Apprentissage	90,32%	90,91%	90,00%	83,33%
	Test	64,52%	63,64%	65,00%	50,00%

Tableau 2 : Performances de la réduction ACP sur la base colon

Considérons les performances sur l'ensemble d'apprentissage et l'ensemble de test, nous avons remarqué que pour la réduction en 15 et 45 composantes, les résultats d'apprentissage sont restés les mêmes mais pour le test la réduction en 15 a donné de meilleurs résultats que la réduction en 45. Les meilleures performances ont été enregistrées sur la réduction en 30 composantes.

2. Pour la base Madelon

- Calculer les composantes principales (obtention d'une matrice de 500X500)
- Définir la matrice de passage (matrice de transformation=500Xn) (nous avons pris n=15, 30 et 45)
- Lancer l'apprentissage de 3 PMCs sur les 3 nouveaux ensembles réduits.
- Tester les architectures PMCs obtenus sur les ensembles de tests

Cela, nous a permis d'obtenir les performances suivantes pour les 3 réductions:

composantes principales		Taux	Se	Sp	Préc
15	Apprentissage	82.10%	81.70%	82.50%	81.53%
	Test	78.00%	80.67%	75.33%	76.58%

Composantes principales		Taux	Se	Sp	Préc
30	Apprentissage	79.10%	79.70%	78.50%	78.75%
	Test	72.50%	73.33%	71.67%	72.13%

Composantes principales		Taux	Se	Sp	Préc
45	Apprentissage	71.50%	69.70%	73.30%	72.30%
	Test	61.83%	60.00%	63.67%	62.28%

Tableau 3 : Performances de la réduction ACP sur la base Madelon

Nous avons remarqué que pour la réduction en 15,30et 45 composantes, les résultats d'apprentissage et de tests ont diminués en augmentant la dimension. Donc la réduction en 15 a donné les meilleurs.

b) ADL

1. Pour la base Colon

- Calculer la matrice de transformation à l'aide de la covariance intra-classe et la covariance interclasse (obtention d'une matrice de 2000X2000)
- Définir la matrice de passage (matrice de transformation=2000Xn) (n indique la nouvelle dimension, dans ce travail, nous avons pris n=15, 30 et 45)
- Lancer l'apprentissage de 3 PMCs sur les 3 nouveaux ensembles réduits.

- Tester les architectures PMCs obtenus sur les ensembles de tests (les 3 réductions réalisées)

Ce qui nous a permis d'obtenir les performances suivantes :

n		Taux	Se	Sp	Préc
15	Apprentissage	100%	100%	100%	100%
	Test	100%	100%	100%	100%

n		Taux	Se	Sp	Préc
30	Apprentissage	100%	100%	100%	100%
	Test	100%	100%	100%	100%

n		Taux	Se	Sp	Préc
45	Apprentissage	100%	100%	100%	100%
	Test	100%	100%	100%	100%

Tableau 4 : Performances de la réduction ADL sur la base colon

Que la réduction soit en 15, 30 ou 45, les performances ont atteint les 100% pour l'apprentissage ainsi que pour le test. Nous justifions ça par le nombre réduit d'exemples (31 patients seulement)

2. Pour la base Madelon

- Calculer la matrice de transformation à l'aide de la covariance intra-classe et la covariance interclasse (obtention d'une matrice de 500X500)
- Définir la matrice de passage (matrice de transformation=500Xn) (nous avons pris n=15, 30 et 45)
- Lancer l'apprentissage de 3 PMCs sur les 3 nouveaux ensembles réduits.
- Tester les architectures PMCs obtenus sur les ensembles de tests (les 3 réductions réalisées)

Cela, nous a permis d'obtenir les performances suivantes pour les 3 réductions (:

N		Taux	Se	Sp	Préc
15	Apprentissage	56.30%	63.30%	49.30%	55.53%
	Test	52.17%	61.67%	42.67%	51.82%

N		Taux	Se	Sp	Préc
30	Apprentissage	53.65%	55.70%	51.60%	53.51%
	Test	51.17%	50.33%	52.00%	51.19%

N		Taux	Se	Sp	Préc
45	Apprentissage	55.50%	60.80%	50.20%	54.97%
	Test	51.17%	53.00%	49.33%	51.13%

Tableau 5 : Performances de la réduction LDA sur la base madelon

Contrairement aux résultats sur la base colon, ceux de Madelon sont moins intéressants, vu que les performances sont aux alentours de 50%.

c) ReliefF

1. Pour la base Colon

- Fixer le nombre de voisin k (nous avons pris 5,7 et 9)
- Prendre les n premiers attributs ayant les poids les plus élevés (nous avons choisit n=30)
- Lancer l'apprentissage de 3 PMCs sur les 3 nouveaux ensembles réduits (les poids obtenus selon les 3 k choisis).
- Tester les architectures PMCs obtenus sur les ensembles de tests (les 3 réductions réalisées)

Cela, nous a permis d'obtenir les performances suivantes:

k		Taux	Se	Sp	Préc
05	Apprentissage	100%	100%	100%	100%
	Test	77.42%	63.64%	85.00%	70.00%

k		Taux	Se	Sp	Préc
07	Apprentissage	100%	100%	100%	100%
	Test	70.79%	72.73%	70.00%	57.14%

k		Taux	Se	Sp	Préc
09	Apprentissage	90.32%	72.73%	100%	100%
	Test	77.42%	54.55%	90.00%	75.00%

Tableau 6 : Performances de la réduction ReliefF sur la base colon

Les performances sur les ensembles d'apprentissages ont été satisfaisantes, cependant ils n'ont pas atteint les 80% pour les ensembles de tests. Les meilleurs résultats ont été enregistrés avec k=5.

2. Pour la base Madelon

- Fixer le nombre de voisin k (nous avons pris 5,7 et 9)
- Prendre les n premiers attributs ayant les poids les plus élevés (nous avons choisit n=30)
- Lancer l'apprentissage de 3 PMCs sur les 3 nouveaux ensembles réduits (les poids obtenus selon les 3 k choisis).
- Tester les architectures PMCs obtenus sur les ensembles de tests (les 3 réductions réalisées)

Ce qui nous a permis d'atteindre les résultats suivants:

k		Taux	Se	Sp	Préc
05	Apprentissage	80.65%	82.60%	78.70%	79.50%
	Test	78.67%	78.00%	79.33%	79.05%

k		Taux	Se	Sp	Préc
07	Apprentissage	84.25%	85.00%	83.50%	83.74%
	Test	78.50%	75.33%	81.67%	80.43%

k		Taux	Se	Sp	Préc
09	Apprentissage	82.60%	81.60%	83.60%	83.27%
	Test	79.17%	75.00%	83.30%	81.82%

Tableau 7 : Performances de la réduction ReliefF sur la base Madelon

Les meilleurs résultats ont été obtenus avec un nombre de voisins k=9.

d) SFS

1. Pour la base Colon

- Lancer le SFS (avec le Naive bayes), ce qui abouti aux attributs 513 et 734.
- Lancer l'apprentissage de PMC sur le nouvel ensemble réduit.
- Tester l'architecture PMC obtenue sur l'ensemble de test.

Ce qui nous a permis d'obtenir ces résultats :

Attributs		Taux	Se	Sp	Préc
513 et 734	APP	90,32%	81.82%	95.00%	90.00%
	TEST	70.97%	45.45%	85.00%	62.50%

Les résultats d'apprentissage et de test de la réduction sont acceptables.

2. Pour la base Madelon

- Lancer le SFS (avec le Naive bayes), ce qui abouti aux attributs 14, 29, 106, 242, 259, 282, 325, 392, 434 et 468.
- Lancer l'apprentissage de PMC sur le nouvel ensemble réduit.
- Tester l'architecture PMC obtenue sur l'ensemble de test.

Cela, nous a permis d'obtenir les performances suivantes:

Attributs		Taux	se	sp	préc
14, 29, 106, 242, 259, 282, 325, 392, 434 et 468	APP	76.20%	77.40%	75.00%	75.59%
	TEST	72.17%	71.67%	72.67%	72.39%

Tableau 8 : Performances de la réduction SFS sur la base Madelon

Les résultats sont dans la moyenne par rapport aux autres méthodes testées.

IV. Discussion des résultats

Afin de voir l'impact de la réduction, nous avons construit des modèles PMC sur les bases non réduites, ce qui nous a menés aux résultats suivants :

Base COLON					
Méthode		Taux	Se	Sp	Préc
Sans réduction	APP	93.55 %	90.91%	95%	90.91%
	TEST	83.87%	81.82%	85%	75%
ACP	APP	93,55%	90,91%	95,00%	90,91%
	TEST	83,87%	72,73%	90,00%	80,00%
LDA	APP	100%	100%	100%	100%
	TEST	100%	100%	100%	100%
RELIEF F	APP	90,32%	72,73%	100%	100%
	TEST	77,42%	54,55%	90,00%	75,00%
SFS	APP	90,32%	81,82%	95,00%	90,00%
	TEST	70,97%	45,45%	85,00%	62,50%

Tableau 9 : Performances de la réduction ACP,LDA,ReliefF,SFS sur la base colon

Base MADELON					
Méthode		Taux	Se	Sp	Préc
Sans réduction	APP	63.55%	64.70%	62%	63%
	TEST	58.5%	61.33%	55.67%	58.04%
ACP	APP	82,10%	81,70%	82,50%	81,53%
	TEST	78,00%	80,67%	75,33%	76,58%
LDA	APP	56,30%	63,30%	49,30%	55,53%
	TEST	52,17%	61,67%	42,67%	51,82%
RELIEF F	APP	82,60%	81,60%	83,60%	83,27%
	TEST	79,17%	75,00%	83,30%	81,82%
SFS	APP	76,20%	77,40%	75,00%	75,59%
	TEST	72,17%	71,67%	72,67%	72,39%

Tableau 10 : Performances de la réduction ACP,LDA,ReliefF,SFS sur la base Madelon

Nous avons remarqué que :

- Pour la base Colon : les résultats sont similaires entre les méthodes de réductions et sans réduction, la différence réside dans le temps d'apprentissage du PMC, alors que ça dure quelques secondes pour les ensembles réduits, ça prend aux alentours de 10 minutes pour l'ensemble non réduit. Cependant cette base demeure petite par rapport aux nombres d'exemples même qu'elle est riche du point de vue nombre d'attributs.
- La base madelon est considérée comme une base Challenge pour les scientifiques (2000 exemples de 500 attributs), les méthodes de réductions ont montrés leurs intérêts sauf l'ADL.

V. Conclusion

Nous avons vu l'impact de la réduction sur l'apprentissage du PMC, en testant les performances sur 2 bases médicales différentes à savoir la base Colon et la base Madelon. Les résultats nous ont montré que le nombre d'exemples de la base influe sur les performances.

Conclusion générale et perspectives:

De nos jours, les moyens mis en œuvre pour la caractérisation des données donnent naissance à des bases de grande dimension. Les attributs générés peuvent contenir des informations redondantes, contradictoires, et incohérentes comme on dit « trop d'information tue l'information ». De ce fait, le besoin de réduire la dimension est né, tout en ayant l'objectif de réduire la redondance et éliminer les contradictions ainsi que les incohérences.

Dans notre travail, nous avons utilisé deux méthodes de réduction de dimension (l'extraction et la sélection) sur deux bases médicales (Colon et Madelon) pour construire l'ensemble réduit de chaque base.

Pour la base colon nous avons appliqués deux méthodes d'extraction (ACP et LDA) et deux méthodes de sélection (Relief F et SFS) sur un ensemble de 31 exemples pour l'apprentissage et le test, nous avons obtenus un meilleur résultat sur LDA avec des performances de 100%.

Pour la base Madelon nous avons fait la même chose sur un ensemble d'apprentissage de 2000 exemples et sur un ensemble de test de 600 exemples, les meilleurs résultats ont été obtenus avec la réduction ReliefF avec des performances aux alentours des 80%.

Comme perspectives pour ce travail :

- Tester d'autres algorithmes d'apprentissages, tels que SVM et les arbres de décisions
- Elargir les expérimentations sur la base Madelon, vu qu'elle est considérée comme une base Challenge pour les techniques de réductions.

Référence bibliographiques:

[1] Avrim L. Blum & Pat Langley. Selection of relevant features and examples in machine learning. Arti_cial Intelligence, vol. 97, no. 1-2, pages 245_271, 1997

[2] Ron Kohavi & George H. John. Wrappers for Feature Subset Selection. Artifficial Intelligence, vol. 97, no. 1-2, pages 273_324, 1997.

[3] Isabelle Guyon. Design of experiments for the NIPS 2003 variable selection benchmark, July 2003.

[4] Isabelle Guyon, Jason Weston, Stephen Barnhill & Vladimir Vapnik. Gene selection for cancer classi_cation using Support Vector Machines. Machine Learning, vol. 46, no. 1-3, pages 389-422, 2002.

[5] Isabelle Guyon, Steve R. Gunn, Asa Ben-Hur & Gideon Dror. Result Analysis of the NIPS 2003 Feature Selection challenge. In Proceedings of the 18th Annual Conference on Neural Information Processing Systems, pages 545_552, 2004.

[6] Raphaël Feraud, Marc Boullé, Fabrice Clérot, Françoise Fessant & Vincent Lemaire. The Orange Customer Analysis Platform. In Proceedings of the 10th International Conference on Data Mining, volume 6171 of Lecture Notes in Computer Science, pages 584_594. Springer, 2010.

[7] Isabelle Guyon, Vincent Lemaire, Marc Boullé, Gideon Dror & David Vogel. Analysis of the KDD Cup 2009 : Fast Scoring on a Large Orange Customer Database. In Proceedings of KDD-Cup 2009 competition, volume 7 of JMLR Workshop and Conference Proceedings, pages 1-22. JMLR.org, 2010.

[8] Tian Lan, Deniz Erdogmus, Andre Adami, and Michael Pavel. Feature selection by independent component analysis and mutual information maximization in eeg signal classi_cation. An International Journal, 170 :409_418, 2005.

[9] B. Chandra and Manish Gupta. An efficient statistical feature selection approch for classification of gene expression data. Journal of biomedical informatics, 3 :3-7, 2010

[10] Yuhang Wing and Fillia Makedon. Application of relief feature fitering algorithm to selecting informative genes for cancer classiffication using microarray data. IEEE Computational Systems, pages 497_498, 2004.

[11] Hong-Wen Deng Ji-Gang Zhang. Gene selection for classiffication ofmicroarray data based on the bayes error. BMC Bioinformatics, pages1-9, 2007.

[12] Badih Ghattas and Anis BenIshak. Selection de variable pour la clasiffication binaire en grande dimension aux donn en biopuces. Journal de la soci Franse de Statistique, 3 :44-66, 2008.

[13] Amel Hafa, "Sélection de Variables Biologiques par l'approche FILTER", mémoire de fin d'étude, université Abou-Bekr Belkaid, Tlemcen 2012.

[14] T. Hastie, R. Tibshirani and J. Friedman: The Elements of Statistical Learning. Springer Series in Statistics, Springer, New York, Berlin, Heidelberg, (2001)

[15] K. Fukunaga: Introduction to Statistical Pattern Recognition. Academic Press, New York, (1990)

[16] P.A. Devijver and J. Kittler: Pattern Recognition: a Statistical Approach. Prentice-Hall, London, (1982)

[17] G. Strang: Linear Algebra and its Applications. third ed., Harcourt, Brace and Jovanovich, New York, (1988)

[18] Martinez & Kak, 2001 Martinez, A. M., & Kak, A. C. 2001. PCA versus ADL. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), 228-233

[19]Yang & Yang, 2003 Yang, Jian, & Yang, Jing-Yu. 2003. Why can ADL be performed in PCA transformed space? *Pattern Recognition*, 36(2), 563-566.

[20] Belhumeur *et al.*, 1997 Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. 1997. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 711-720.

.

[21] John, G. H., Kohavi, R. et Pfleger, K. (1994). Irrelevant features and the subset selection problem. In MACHINE LEARNING : PROCEEDINGS OF THE ELEVENTH INTERNATIONAL, pages 121-129. Morgan Kaufmann

[22] Kohavi, R. et John, G. H. (1997). Wrappers for feature subset selection. Artif. Intell., 97:273-324.

[23] Kira, K. et Rendell, L. A. (1992). The feature selection problem : Traditional methods and a new algorithm. In AAAI, pages 129{134, Cambridge, MA, USA. AAAI Press and MIT Press

[24] John, G. H., Kohavi, R. et Pfleger, K. (1994). Irrelevant features and the subset selection problem. In MACHINE LEARNING : PROCEEDINGS OF THE ELEVENTH INTERNATIONAL, pages 121{129. Morgan Kaufmann

[25] Koller, D. et Sahami, M. (1996). Toward optimal feature selection. pages 284-292.

[26]Liu, H., Motoda, H. et Yu, L. (2002). Feature selection with selective sampling. In Proceedings of the Nineteenth International Conference on Machine Learning, ICML '02, pages 395-402, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc

[27] Marill, T. et Green, D. M. (1963). On the effectiveness of receptors in recognition systems. IEEE transactions on Information Theory, 9:11-17.

[28] : En particulier dans McCulloch, W. S., Pitts, W., *A Logical Calculus of the Ideas Immanent in Nervous Activity*, Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943

[29]: Donald Olding Hebb, *The Organization of Behavior : A Neuropsychological Theory*, Wiley, New York, 1949.

[30]: F. Rosenblatt (1958), "The perceptron: a probabilistic model for information storage and organization in the brain",- repris dans J.A. Anderson & E. Rosenfeld (1988), Neurocomputing. Foundations of Research, MIT Press

[31]: Widrow, B., and Hoff, M. E., Jr., 1960, Adaptive switching circuits, in *1960 IRE WESCON Convention Record*, Part 4, New York: IRE, pp. 96–104.

[32]: McCorduck, Pamela (2004), *Machines Who Think* (2nd ed.), Natick, MA: A. K. Peters, Ltd., ISBN 1-56881-205-1, pp. 104−107

[33]: T. Kohonen, *Self-Organized Formation of Topologically Correct Feature Maps*, Biological Cybernetics, vol. 46, pp. 59–69, 1982.

[34]: Le modèle de J. Hopfield est décrit dans "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *P.N.A.S. USA*, vol. 79 (1982)

[35] : R.M Downs .B.D.Stea.Essai sur la cartographie mental tr.J.Rondel.Paris.Edisem,1981,p.103-104

[36]: T. Kohonen, *Self-Organizing Maps*, vol. 30, Springer Verlag, 1995.

[37]: Patrick van der Smagt, *An Introduction to Neural Networks*, 1996, page 33

[38]: Rumelhart, D. E., Hinton, G. E. & Williams, R. J. in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Vol. 1: *Foundations* (eds Rumelhart, D. E. & McClelland, J. L.) 318−362 (MIT, Cambridge, 1986).

[39]: S. Lawrence, I. Burns, A.D. Back, A.C. Tsoi and C. Lee Giles ``Neural Network Classification and Prior Class Probabilities, Tricks of the Trade, Lecture Notes in Computer Science State-of-the-Art Surveys, G. Orr, K-R. Mueller, R. Caruana (Eds), Springer-Verlag. pp. 299-314, 1998.

[40]: Janlou Chaput, « En bref : le cerveau aurait-il perdu 14 milliards de neurones ? » [archive], sur *Futura-Sciences*, 15 mars 2012

[41]: Williams, R and Herrup, K (2001). "The Control of Neuron Number." Originally published in *The Annual Review of Neuroscience* **11**:423–453 (1988). Last revised Sept 28, 2001.

[42]: Marieb, Helen, Hoen, Katja. Anatomie et physiologie humaines, ERPI éditions, 2010, p.442

[43] : Marc Parizeau, *Réseaux de Neurones* Université Laval, Laval, 2004, 272 p.

[44] : Patrick van der Smagt, *An Introduction to Neural Networks*, 1996, page 33

[45] : Marc Parizeau Département de génie électrique et de génie informatique Université Laval 10 septembre 2004

[46] : Mohamed Gadiri. Tlemcen : les cancer occupent une place importante dans les préoccupations sanitaires. 2009.

[47] : http://archive.ics.uci.edu/ml/datasets/Madelon

Résumé :

La caractérisation des informations a donné naissance aux bases de données de grande dimension. Ces bases contiennent souvent des informations redondantes et/ou contradictoires. De ce fait, la réduction de dimension a pris place. Dans notre mémoire, nous avons abordé l'impact de différentes méthodes de réduction sur l'apprentissage du PMC (Perceptron Multi-Couche). Nous avons appliqué les méthodes d'extractions (ACP et LDA) et les méthodes de sélections (Relief F et SFS) sur deux bases médicales Colon et Madelon. La LDA a donné les meilleurs résultats sur la base Colon en atteignant les 100%, et le ReliefF a atteint des performances aux alentours des 80% sur la base Madelon.

Mot clés : PMC, extraction, sélection, Colon, Madelon.

ملخص

خاصة المعلومات أعطت ولادة قواعد البيانات من الحجم الكبير, غالبا ما تحتوي على قواعد البيانات هذه المعلومات الزائدة عن الحاجة و / أو متناقضة, إختصار المعلومات أخد مكانه.في مدكرتنا هده أخدنا مختلف أساليب إختصار المعلومات للتعلم في المستقبلات المتعددة الطبقات.

ناقشنا تأثير أساليب مختلفة للحد من التعلم وPMC (متعدد الطبقات المتعرف). طبقنا أساليب الاستخراج (PCA وLDA) وطرق اختيار (Relief F وFSS) على قاعدتين طبيتين colon وMadelon. أعطى LDA على أفضل النتائج حول colonتصل إلى 100٪ ووصلت إلى أداء ReliefF حول Madelon على 80٪.

مفاتيح PMC ,Madelon, Colon, استخراج،الاختيار

Abstract :

Characterization information gave birth to databases of large size. These databases often contain redundant and / or contradictory information. Therefore, dimension reduction has taken place. In our work, we have discussed the impact of different methods of reducing for the MLP learning (Multi-Layer Perceptron). We applied the extraction methods (PCA and LDA) and selection methods (Relief F and FSS) on both Colon and Madelon medical database. The LDA gave the best results based on Colon reaching 100% and ReliefF reached performance around 80% on Madelon.

Key words: MLP, extraction, selection, Colon, Madelon