N° d'ordre : 2501

Université des Sciences et Technologies de Lille

Thèse préparée au

Laboratoire d'Automatique I3D

Pour obtenir le titre de Docteur
en Automatique et Informatique Industrielle

Présentée par

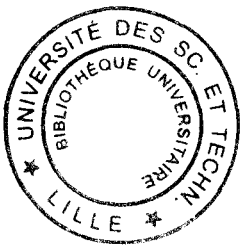
Mohamed BETROUNI

Réseaux de neurones pour la projection plane de données multidimensionnelles et pour le suivi de procédés industriels

Soutenue le 03 Mars 1999 devant le Jury composé de MM :

Alain FAURE	Professeur à l'IUT de l'Université du Havre, Rapporteur
Stéphane CANU	Professeur à l'INSA Rouen, Rapporteur
Denis HAMAD	Professeur à l'Université de Picardie Jules Verne, Co-directeur
Jack-Gérard POSTAIRE	Professeur à l'USTL, Co-directeur
Christian VASSEUR	Professeur à l'USTL, Président du jury
Mohamed DAOUDI	Maître de conférences à l'ENIC/INT, Examinateur
Jean Michel GRAVE	Chef du département Techniques et Systèmes de Norelec, Examinateur

Avant propos



Ce travail a été effectué au Laboratoire d'Automatique I3D (Interaction, Image et Ingénierie de la Décision) de l'Université des Sciences et Technologies de Lille USTL. Le Laboratoire I3D (anciennement CAL) est dirigé par le Professeur Christian VASSEUR.

Le travail de thèse débute par une recherche fondamentale sur les réseaux de neurones à apprentissage non supervisé pour la projection plane. Ce thème de recherche a été développé par Monsieur Denis HAMAD dans le cadre de son HDR soutenue en 1997. Cet axe de recherche a été initié par la thèse de Monsieur Mohamed DAOUDI soutenue en 1993 et poursuivi par la thèse de Monsieur Stéphane DELSERT soutenue en 1996.

Le travail de recherche a évolué vers l'application des réseaux de neurones dans le cadre de deux contrats industriels négociés par le Professeur Jack-Gérard POSTAIRE et dont le suivi scientifique a été assuré par Monsieur Denis HAMAD.

Le premier contrat, avec MSC-BSN, concerne la détection de défauts de type glaçures sur des bouteilles en verre. Ce contrat a été mené en collaboration avec Monsieur Christian FIRMIN qui a soutenu sa thèse en Mars 1997.

Le second est financé par la Région du Nord Pas de Calais en partenariat avec des entreprises régionales et porte sur le suivi de fonctionnement des aéro-générateurs. Deux chercheurs, Messieurs Noureddine EL HOR et Régis BERTRAND, sont financés pour implanter le système de suivi de fonctionnement à base de réseaux de neurones sur l'une des éoliennes du parc éolien de Dunkerque.

Je tiens à remercier le professeur Jack-Gérard POSTAIRE pour m'avoir accueilli au sein de son équipe Image & Décision et particulièrement le Professeur Denis HAMAD qui m'a toujours soutenu et sans qui ce travail n'aurait pas vu le jour. Je remercie également Messieurs les Professeurs Alain FAURE et Stéphane Canu pour avoir accepté de rapporter sur ce travail.

Enfin, mes vifs remerciements vont à mes parents, mes frères et soeurs et à ma femme qui a fait preuve de beaucoup de courage et de patience tout au long de ce travail.

Sommaire

Introduction générale

Introduction générale

L'acquisition de nos connaissances s'effectue à partir de notre reconnaissance de l'environnement selon le principe "observer pour comprendre et classer pour expliquer". Ainsi, lors de l'observation d'un objet, nous recherchons des éléments suggestifs pour le reconnaître puis le classer par association avec des objets semblables dans un mode de représentations relationnelles hiérarchiques.

Le but de la classification automatique est donc de découvrir, dans une population d'objets, la présence de classes au sein desquelles se regroupent des objets semblables. En général, les objets sont caractérisés par un ensemble d'attributs qu'il est commode de représenter par des points dans un espace multidimensionnel.

Lorsqu'on ne dispose d'aucune information a priori sur les objets à classer, c'est à dire quand on se place dans un contexte non supervisé, on peut faire appel à des techniques de classification automatique pour découvrir la structure des données, c'est à dire la recherche de groupements ou classes [Dud 73]. Parmi ces techniques, certaines consistent à optimiser des critères qui reflètent les distances interclasses et intraclasses des objets [Dev 82]. D'autres consistent à supposer que les classes sont caractérisées par des domaines de l'espace où se concentrent les observations, séparés par des domaines avec peu d'observations. Il s'agit alors d'analyser les propriétés de la fonction de densité sous-jacente à la distribution des observations à classer pour en déterminer les maxima et les minima locaux [Pos 89]. Avec ces techniques totalement automatiques, le rôle de l'opérateur humain est juste réduit au réglage de certains paramètres, sans pouvoir contrôler les résultats de la classification.

Une approche alternative à ces techniques dites "presse-bouton" consiste à impliquer davantage l'opérateur dans le processus de classification en lui présentant les données dans un espace de dimension inférieure ou égale à trois et à les afficher sur un écran graphique [Fuk 82] [Sam 69]. L'être humain est en effet capable de reconnaître et d'analyser, presque instantanément, une représentation plane ou spatiale des données pour en extraire les informations utiles, alors qu'il est vite rebuté face à un tableau de nombres. Présenter un

volume important de données sous une forme graphique facilement analysable par l'opérateur est peut être la démarche la plus efficace pour découvrir visuellement la présence de groupements ou classes au sein d'une population.

Les méthodes de projection sont, d'un point de vue mathématique, des cas particuliers des méthodes de réduction de la dimension de l'espace des observations. Cependant, les méthodes de projection et les méthodes de réduction de la dimension diffèrent dans leurs applications. En effet, le but des méthodes de réduction de la dimension est de résoudre le problème de la "malédiction de la dimension" ou "curse of dimensionality" défini par Bellman et bien connu en classification [Dud 73], alors que les méthodes de projection réduisent l'espace de projection à une ou deux dimensions dans le but de visualiser l'ensemble des projections sur un support graphique. Cette visualisation plane mise à la disposition de l'opérateur pour qu'il découvre la structure géométrique des données multidimensionnelles est la base fondamentale de la classification interactive.

Pour élaborer une visualisation plane, la littérature statistique est riche en méthodes de projection. En effet, la variété des critères d'optimisation joue un rôle fondamental dans cette diversité [Sie 88a] [Sie 88b].

Le travail présenté dans cette thèse, constituée de six chapitres, peut être décomposé en deux parties. La première concerne les méthodes de projection plane et couvre les quatre premiers chapitres alors que la deuxième, constituée de deux chapitres, présente notre contribution dans le cadre de deux contrats industriels.

Dans le chapitre 1, nous présentons les méthodes de projection analytiques faisant appel à des fonctions linéaires, telles que la méthode d'analyse en composantes principales, l'algorithme des projections révélatrices et les méthodes de projection non analytiques se référant à des procédures algorithmiques, telles que l'algorithme de Sammon.

Le chapitre 2 expose des réseaux de neurones à apprentissage non supervisé par la règle de Hebb. Ces réseaux sont considérés comme une implantation neuronale de la méthode d'analyse en composantes principales.

Le chapitre 3 est consacré aux réseaux de neurones à fonctions de transfert non linéaires. Le principal avantage de ces réseaux réside dans leur faculté à élaborer une transformation analytique entre l'espace des observations multidimensionnelles et l'espace des projections. Différentes architectures de réseaux multicouches sont présentées. Notre contribution se situe,

d'une part au niveau de la reformulation du critère de minimisation de l'erreur des moindres carrés comme un problème de maximisation de la vraisemblance et, d'autre part, dans l'optimisation de l'architecture des réseaux multicouches par des critères informationnels.

Dans le chapitre 4 nous montrons sur deux exemples les performances des réseaux de neurones en projection plane en fonction de leurs paramètres d'apprentissage d'une part et de leurs architectures d'autre part.

Au cours de notre recherche nous avons appliqué les réseaux de neurones artificiels à apprentissage supervisé et non supervisé dans le cadre de deux contrats industriels pour le contrôle qualité et pour le diagnostic des machines.

Le premier contrat, financé par MSC, filiale du groupe BSN Danone, a porté sur la détection de glaçures sur les goulots des bouteilles en verre par vision artificielle. Dans le chapitre 5, nous présentons l'étude qui a été menée depuis la phase d'acquisition des images et leurs traitements jusqu'à la phase de décision par réseaux de neurones.

Le dernier chapitre présente la seconde application industrielle qui a aussi fait l'objet d'un contrat, financé par la Région Nord-Pas de Calais en collaboration avec NORELEC. L'étude a consisté à exploiter les signaux provenant des vibrations de certains éléments d'une éolienne pilote installée à Dunkerque en vue de la prédiction de dysfonctionnements éventuels. Ce travail est actuellement poursuivi par deux chercheurs financés dans le cadre de ce contrat.

Chapitre 1

Méthodes classiques de projection de données multidimensionnelles pour la classification interactive

Chapitre 1 Méthodes classiques de projection de données multidimensionnelles pour la classification interactive

1.1 Introduction

L'idée de base de la classification interactive est d'exploiter les capacités exceptionnelles de perception et d'analyse de l'opérateur humain en lui fournissant une visualisation des données sur un écran couleur et des outils interactifs d'aide à la décision. Les questions qui se posent alors sont:

- Comment présenter les données sur un plan ?
- Quels outils fournir à l'opérateur pour lui permettre d'agir sur les données pour les manipuler ?

La réponse à la seconde question relève du domaine de la communication Homme/Machine qui n'est pas le sujet de notre travail. Cependant, avant de répondre à la première question qui fait l'objet principal de cette thèse, il est important de préciser le cadre de notre étude.

On considère un ensemble de N objets $\{O_1, O_2, ..., O_n, ..., O_N\}$ caractérisés chacun par D attributs regroupés sous la forme d'un vecteur d'attributs $X = (x_1, x_2, ..., x_d, ... x_D)^T$, où x_d est la d-ème composante du vecteur X. Soient :

- $Xn = (xn1, xn2, ...xnd, ..., xnD)^T$ la réalisation particulière du vecteur X associée à l'objet On. Dans la suite de notre travail, on appellera Xn indifféremment le "vecteur observation" ou "l'observation" associé à l'objet On. xnd est la valeur, supposée réelle, prise par l'attribut xd pour l'objet On.
- {X1, X2, ..., Xn, ..., XN} l'ensemble des observations recueillies sur les N objets {O1, O2, ..., On, ..., ON}.

D'un point de vue géométrique, chaque observation peut être représentée par un point dans un espace multidimensionnel R^D appelé *espace des observations*. L'ensemble des observations définit alors un nuage de points dans cet espace. Si l'espace d'observation est de dimension

inférieure ou égale à trois, il est facile de représenter l'ensemble des observations par un nuage de points sur un écran graphique. L'analyste peut aisément examiner ce nuage pour découvrir la présence de groupements ou classes ou pour repérer des points isolés. Malheureusement, les observations à analyser ont souvent une dimension supérieure à trois et il convient de les projeter sur des espaces de dimension réduite dans le but de faciliter leur analyse et leur interprétation.

D'une manière générale, les méthodes de projection consistent à définir une transformation de *l'ensemble des observations* $\{X_1, X_2, ..., X_n, ..., X_N\}$ défini dans l'espace d'observation R^D vers *l'ensemble des projections* $\{Y_1, Y_2, ..., Y_n, ..., Y_N\}$, $Y_n = (y_{n1}, y_{n2}, ..., ..., y_{nm}, ..., y_{nM})^T$, défini dans l'espace de projection R^M, où $M = 2$ si l'on désire bénéficier des avantages de la visualisation plane. Soit Φ cette transformation qui, de l'ensemble des observations $\{X_1, X_2, ..., X_n, ..., X_N\}$ conduit au nouvel *ensemble des projections* $\{Y_1, Y_2, ..., Y_n, ..., Y_N\}$. Les transformations Φ peuvent être soit analytiques quand elles font appel à des fonctions linéaires ou non linéaires, soit non-analytiques quand elles résultent de procédures algorithmiques.

Une transformation analytique, Φ, établit une correspondance entre l'espace d'observation R^D et l'espace de projection R^M.

$$\begin{aligned} &\Phi : R^D \rightarrow R^M \quad ; \quad M < D \\ &\quad X \rightarrow Y \end{aligned} \tag{1}$$

A toute observation X de l'espace d'observation R^D, on peut associer, grâce à cette transformation, une projection $Y = \Phi(X)$ dans l'espace R^M.

A l'inverse des transformations analytiques, les procédures algorithmiques établissent des correspondances entre l'ensemble des observations et l'ensemble des projections.

$$\Psi : \{X_1, X_2, ..., X_n, ..., X_N\} \rightarrow \{Y_1, Y_2, ..., Y_n, ..., Y_N\}. \tag{2}$$

Ces procédures algorithmiques ne fournissent pas de relations explicites entre les deux ensembles pour projeter une nouvelle observation. Il faut donc les réexecuter intégralement en prenant en compte l'ensemble des observations, dès qu'on en ajoute une nouvelle.

D'une manière générale, les méthodes de projection résultent de l'optimisation de critères de telle sorte que l'ensemble des projections ne représente fidèlement l'ensemble des observations qu'au sens de ce critère. La littérature en classification est riche en méthodes de projection car

la variété des critères d'optimisation employés joue un rôle fondamental dans cette diversité. Les papiers de Siedlecki et al. offrent une revue détaillée des techniques classiques de projection pour l'analyse exploratoire des données multidimensionnelles [Sie 88a], [Sie 88b].

Globalement, on distingue quatre familles de méthodes de projection, selon qu'elles opèrent dans un contexte supervisé ou non et quelles sont analytiques ou non (cf. figure 1.1).

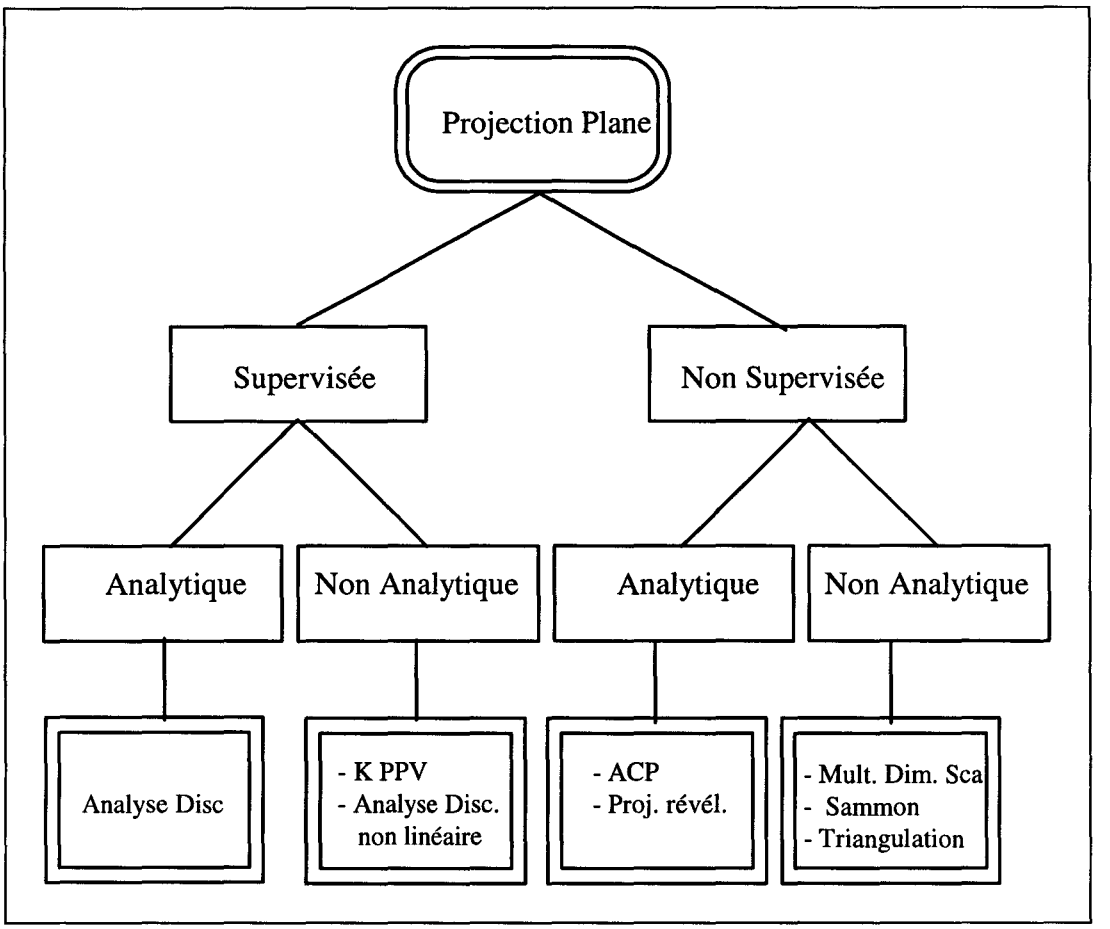


Figure 1.1: Familles de méthodes de projection classiques

L'analyste dispose ainsi de [Ham 97]:

- Méthodes supervisées et analytiques telle que:

  - l'analyse discriminante [Seb 71].

- Méthodes supervisées et non analytiques telles que:

  - la projection par l'algorithme des K plus proches voisins [Fuk 82],

- l'analyse discriminante non linéaire [Fuk 90].

- Méthodes non supervisées et analytiques telles que:

  - l'analyse en composantes principales [Seb 71],

  - l'algorithme des projections révélatrices, plus connu sous le vocable anglais de "Projection pursuit algorithm" [Fri 74].

- Méthodes non supervisées et non analytiques telles que:

  - l'algorithme MDS "Multidimensional scaling " [Kru 64],

  - l'algorithme de Sammon [Sam 69],

  - la projection par triangulation [Lee 79].

Dans la suite de ce chapitre, nous présentons les méthodes de projection non supervisées analytiques et non analytiques les plus connues.

1.2 Méthode d'analyse en composantes principales

La méthode d'analyse en composantes principales (ACP), appelée aussi transformation de Karhunen-Loeve, est une méthode statistique bien connue ayant des applications dans différents domaines des sciences de l'ingénieur, en particulier, en analyse de données, en traitements d'image et du signal. L'ACP est une transformation linéaire orthogonale de l'espace d'observation R^D vers l'espace de projection R^M, $M \leq D$ [Seb 71].

Le principe de l'ACP consiste à supposer que l'on dispose d'un vecteur aléatoire $X=(x_1, x_2, ..., x_d, ..., x_D)^T$ dans R^D de moyenne $\overline{X} = E(X)$ et de matrice de variance-covariance $S = E((X-\overline{X})(X-\overline{X})^T)$, et à rechercher un vecteur aléatoire Y de dimension M de composantes orthogonales non-corrélées et de variances maximales.

Soient $W_1, W_2, ..., W_m, ..., W_M$, avec $W_m = (w_{m1}, w_{m2}, ..., w_{md}, ..., w_{mD})^T$, les M vecteurs propres normés de la matrice de variance-covariance S associés aux M valeurs propres, ordonnées en valeurs décroissantes : $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq \cdots \geq \lambda_M$.

L'espace de projection est défini par les M vecteurs propres W_m, m=1, ..., M. Les composantes y_m du vecteur de projection Y sont obtenues par:

$$y_m = W_m^T X \quad , \quad m = 1, \cdots, M . \tag{3}$$

Les coordonnées y_m, m=1, ..., M du vecteur Y, appelées composantes principales, ne sont pas corrélées entre-elles et leurs variances sont les valeurs propres λ_m, m=1, ..., M, de la matrice S.

Dans la pratique on dispose d'un ensemble fini d'observations $\{X_1, X_2, ..., X_n, ..., X_N\}$, réalisations particulières du vecteur aléatoire X. Le vecteur moyenne $\overline{X}$ de la distribution de X est estimé par $\mu = \frac{1}{N}\sum_{n=1}^{N} X_n$ et la matrice de covariance S est estimée par $\Sigma = \frac{1}{N-1}\sum_{n=1}^{N} (X_n - \mu)(X_n - \mu)^T$.

Les algorithmes d'implantation de l'ACP sont faciles à mettre en oeuvre et rapides, le temps de calcul est de l'ordre de NxD^2. Cependant, les différentes versions de l'ACP sont basées sur l'optimisation d'un critère faisant intervenir des moyennes et des variances. Par conséquent, l'espace de projection obtenu est celui dans lequel les composantes des observations sont les mieux dispersées. Cette méthode de projection selon les axes de fortes dispersions ne tient pas toujours compte de la répartition locale des observations dans l'espace.

Dans la suite nous présentons deux exemples qui seront utilisés pour comparer les résultats obtenus par les différentes méthodes de projection.

Exemple 1: Quatre classes Gaussiennes

On considère un ensemble de quatre classes générées artificiellement, dans un espace à trois dimensions. Chacune est constituée de 100 observations. Il s'agit de classes Gaussiennes dont les centres sont disposés sur les coins d'un cube de côté 1, (cf. Figure 1.2). Les caractéristiques statistiques des classes sont résumées dans le tableau 1.1.

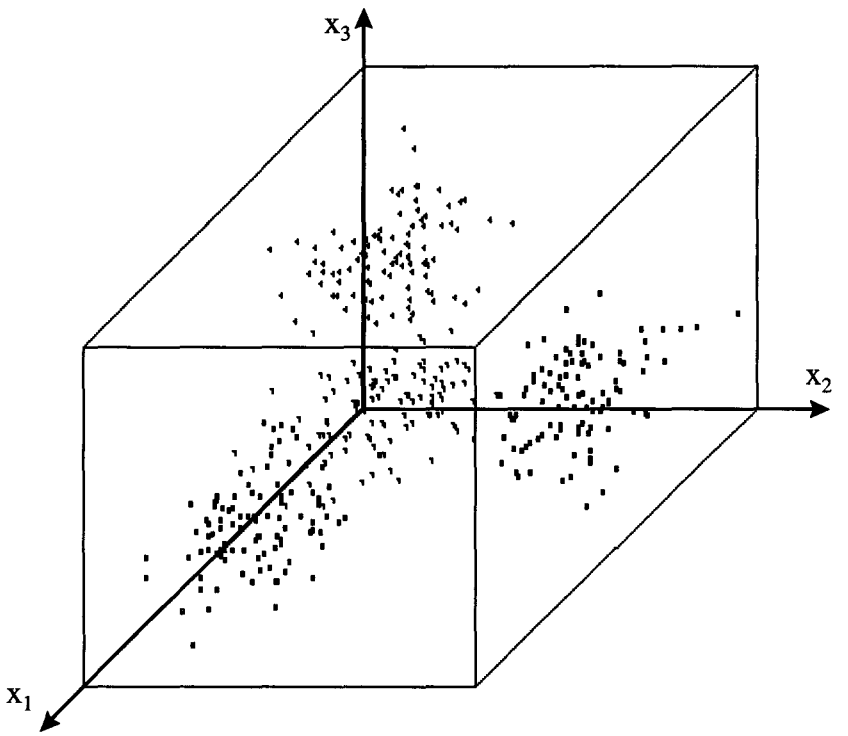


Figure 1.2 Représentation des quatre classes Gaussiennes dans leur espace d'origine

	Classe 1	Classe 2	Classe 3	Classe 4
Taille	100	100	100	100
Vecteurs moyennes	0 0 0	1 0 0	0 1 0	0 0 1
Matrices de variance-covariance	0,05 0 0 0 0,05 0 0 0 0,05	0,05 0 0 0 0,05 0 0 0 0,05	0,05 0 0 0 0,05 0 0 0 0,05	0,05 0 0 0 0,05 0 0 0 0,05

Tableau 1.1. Caractéristiques statistiques de 4 classes Gaussiennes.

Les observations sont projetées sur un plan par la méthode d'analyse en composantes principales (cf. Figure 1.3). L'information apportée par le premier axe principal est de 42,8% et celle apportée par le 2ème axe est 41,05%. Un total de 83,85% de l'information est donc expliquée par le plan principal.

La figure 1.3 fait apparaître deux classes projetées l'une sur l'autre. Cette méthode de projection selon les axes de fortes dispersions ne tient pas compte de la répartition locale des observations.

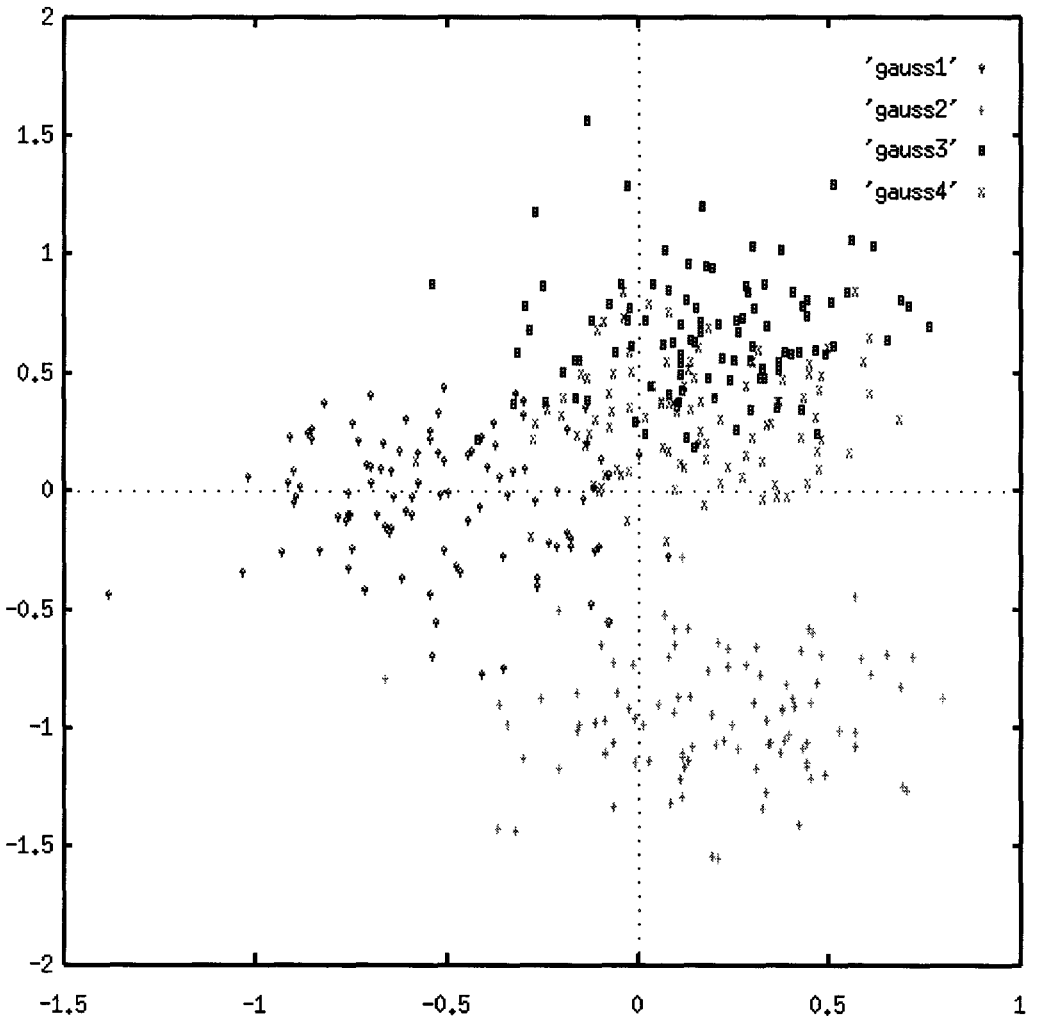


Figure 1.3 Projection des quatre classes Gaussiennes par ACP. 83,85% de l'information totale est expliquée par le plan principal.

Exemple 2: Iris de Fisher

L'exemple des Iris de Fisher est un exemple bien connu dans la littérature statistique. Il s'agit de 3 variétés de fleurs d'Iris (Setosa, Versicolor et Verginica) caractérisées par la longueur et la largeur des pétales ainsi que la longueur et la largeur des sépales exprimées en centimètres. Pour chaque type de fleur on dispose d'un échantillon de taille 50. La population totale est donc constituée de 150 observations réparties en 3 classes de 50 observations chacune,

représentées dans un espace de dimension 4. Le tableau 1.2 indique les caractéristiques statistiques des classes, à savoir leurs vecteurs moyennes et leurs matrices de variance-covariance. On peut remarquer que les éléments sur les diagonales ne sont pas identiques, donc les classes ne sont pas sphériques et que les éléments hors-diagonales sont non nuls.

	Classe 1 (Setosa)	**Classe 2 (Versicolor)**	**Classe 3 (Verginica)**
Vecteurs moyennes	2,46 14,62 34,28 50,1	13,26 43,22 27,64 59,36	20,06 55,52 29,74 65,88
Matrices de variance-covariance	0,124 0,099 0,016 0,010 0,099 0,144 0,012 0,009 0,016 0,012 0,030 0,006 0,010 0,009 0,006 0,011	0,266 0,085 0,183 0,056 0,085 0,098 0,083 0,041 0,183 0,083 0,221 0,073 0,056 0,041 0,073 0,039	0,404 0,094 0,303 0,049 0,094 0,104 0,071 0,048 0,303 0,071 0,304 0,049 0,049 0,048 0,049 0,075

Tableau 1.2. Caractéristiques statistiques des Iris.

La figure 1.4 montre la projection plane obtenue par l'ACP. L'information apportée par le premier axe principal est de 72,9% et celle apportée par le deuxième axe est de 22,85%. Au total 95,75% de l'information est expliquée par le plan principal. On peut remarquer la présence d'une classe nettement séparée des deux autres et correspondant à la classe des Iris Setosa. Les 2 classes qui se chevauchent fortement correspondent aux Iris Versicolor et Verginica qui son difficilement différentiables sur cette projection plane.

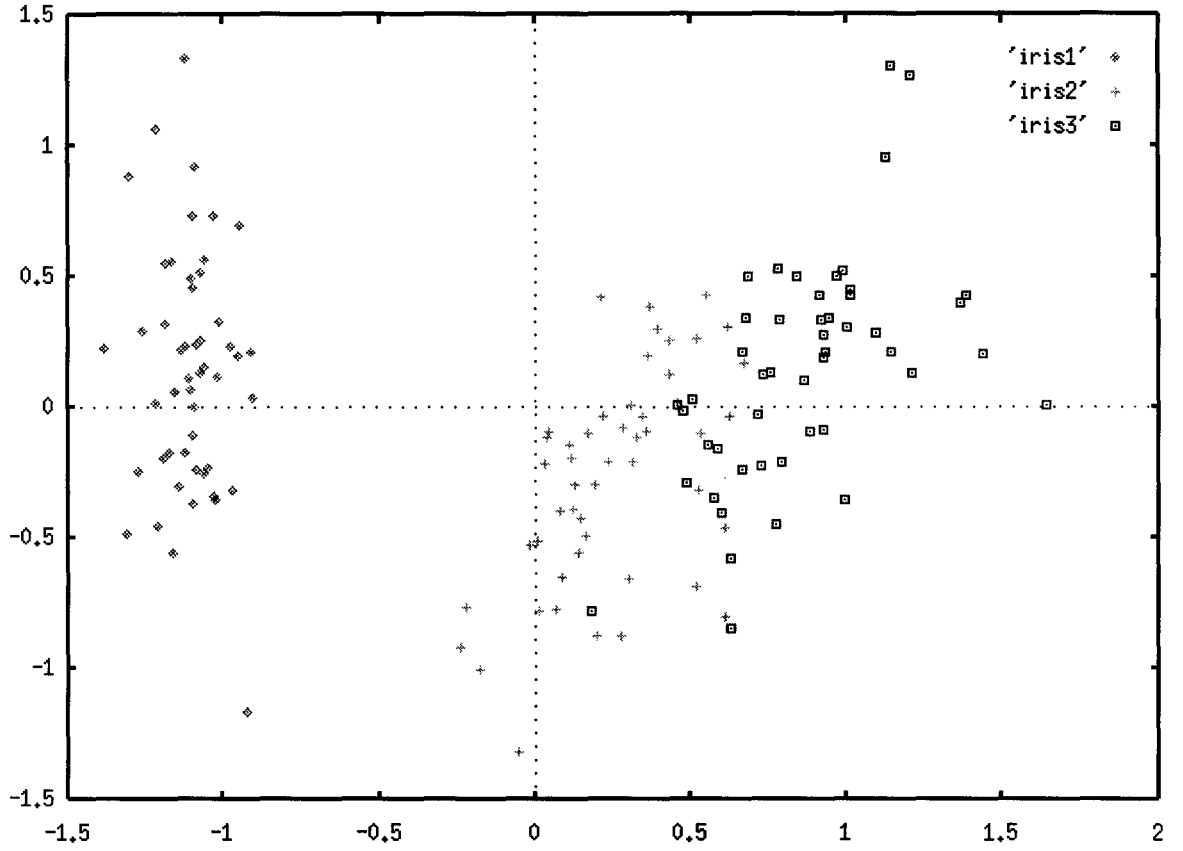


Figure 1.4 Projection des Iris de Fisher par ACP. 95,75% de l'information totale est expliquée par le plan principal.

1.3 Algorithme des projections révélatrices (Projection Pursuit Algorithm)

A l'inverse de l'ACP, l'algorithme des projections révélatrices définit l'espace de projection en maximisant un indice qui tient compte non seulement de la dispersion des observations projetées mais aussi de leurs densités locales dans cet espace [Fri 74].

Recherche d'un axe de projection

Soit U le vecteur directeur dans R^D d'un axe de projection. A chaque point X_n, n=1, ..., N, de l'espace des observations correspond un point y_n sur l'axe de projection défini par:

$$y_n = U^T Xn. \qquad (4)$$

Si on suppose que les points projetés sont ordonnés par ordre croissant de leurs abscisses $\{y_1, y_2, ..., y_n, ..., y_N\}$, un indice I(U) défini par:

$$I(U) = \sigma_p(U).\delta_R(U)\,, \tag{5}$$

peut être associé à l'axe de projection où $\sigma_p(U)$ est un facteur de contribution de la variance et $\delta_R(U)$ est un facteur de contribution de la densité locale des observations projetées sur cet axe. Précisons chacun de ces facteurs:

- ***Facteur de contribution de la variance***

Le facteur de contribution de la variance $\sigma_p(U)$ défini par:

$$\sigma_p(U) = \sqrt{\frac{1}{(1-2p)N}\sum_{n=pN}^{(1-p)N}(y_n - \overline{y}_p)^2} \tag{6}$$

est une version robuste de la variance puisqu'il ne tient pas compte des points situés sur les extrémités de l'axe de projection. En effet, une fraction p des points situés sur chaque extrémité de l'axe de projection est omise du calcul de cette variance.

Le terme $\overline{y}_p$ est une version robuste de la moyenne qui ne tient pas compte des points extrêmes:

$$\overline{y}_p = \frac{1}{(1-2p)N}\sum_{n=pN}^{(1-p)N} y_n\,. \tag{7}$$

- ***Facteur de contribution de la densité locale***

Le facteur de contribution de la densité locale défini par:

$$\delta_R(U) = \begin{cases} \sum_{u=1}^{N}\sum_{v=1}^{N} f(d_{uv}) & \text{si } d_{uv} \le R \\ 0 & \text{sinon} \end{cases} \tag{8}$$

fait intervenir une fonction monotone décroissante $f(d_{uv})$ où:

$$d_{uv} = |y_u - y_v| \tag{9}$$

est la distance entre deux points y_u et y_v sur l'axe U. On peut choisir, par exemple, une fonction linéaire $f(d_{uv})=R-d_{uv}$, ou bien une fonction quadratique $f(d_{uv}) = R^2 - d_{uv}^2$ où R est un

rayon définissant une fenêtre qui contrôle la contribution des points projetés intervenant dans le calcul de $\delta_R(U)$.

Recherche d'un plan de projection

La projection sur un plan est caractérisée par deux vecteurs de directions U et V choisis orthogonaux. Dans ce cas l'indice à maximiser s'écrit:

$$I(U,V) = \sigma_p(U).\sigma_p(V).\delta_R(U,V) \tag{10}$$

où $\sigma_p(U)$ et $\sigma_p(V)$ sont donnés par l'équation (6) et $\delta_R(U,V)$ est définie par:

$$\delta_R(U,V) = \begin{cases} \sum_{u=1}^{N}\sum_{v=1}^{N} f(d_{uv}) & \text{si } d_{uv} \le R \\ 0 & \text{sinon} \end{cases} \tag{11}$$

d_{uv} est la distance Euclidienne entre les points Y_u et Y_v du plan de projection:

$$d_{uv} = \sqrt{(y_{u1} - y_{v1})^2 + (y_{u2} - y_{v2})^2}\,. \tag{12}$$

L'algorithme des projections révélatrices est une généralisation de l'ACP. Comme l'ACP, il définit une transformation linéaire de l'espace des observations vers l'espace des projections. On peut remarquer que pour p = 0 et pour $\delta_R = 1$, l'indice I à maximiser devient celui de l'ACP (cf. équations (5) et (10)).

Le principal inconvénient de cette méthode de projection est que l'indice de projection I(U, V) n'est pas différentiable en tout point et par conséquent ne peut pas être maximisé par une méthode analytique d'optimisation. De plus, la mise en oeuvre pratique nécessite le réglage du paramètre p définissant la fraction des points extrêmes rejetés dans les calculs et du paramètre R qui module l'influence de la densité locale.

1.4 Algorithme d'analyse de proximité et algorithme de projection plane de Sammon

Les algorithmes d'analyse de proximité (MDS pour Multi-Dimensional Scaling) regroupent un ensemble de méthodes statistiques de projection non linéaire. Dans ces méthodes, au lieu de disposer d'un tableau de données croisé (Observations)x(Attributs), on dispose d'une matrice

de proximité (Observations)x(Observations) définissant la similarité (dissimilarité) entre les observations prises deux à deux.

Le but des algorithmes MDS est de rechercher, à partir d'une matrice de proximité de dimension NxN, une configuration de points dans un espace de projection de dimension M telles que les points projetés représentent au mieux les similarités (dissimilarités) entres les observations. Les algorithmes MDS sont dits métriques si les éléments de la matrice de proximité sont des distances entre les observations. Il sont dits non métriques si les éléments de la matrice de proximité ne sont pas des distances [Kru 64].

L'algorithme de Sammon est un cas particulier des méthodes MDS où les éléments de la matrice de proximité représentent les distances entre les observations [Sam 69]. L'idée de base de l'algorithme de Sammon est de chercher, pour un ensemble d'observations $\{X_1, X_2, ..., X_u, ..., X_v, ..., X_N\}$ dans l'espace R^D, un ensemble de projections $\{Y_1, Y_2, ..., Y_u, ..., Y_v, ..., Y_N\}$, dans l'espace R^M, telles que les structures géométriques locales des observations soient préservées au mieux par leurs projections.

Plus précisément, soient deux points X_u et X_v de l'ensemble des observations et Y_u et Y_v leurs projections respectives.

Posons:

- $d^*_{uv} = d(X_u, X_v)$, la distance entre les points X_u et X_v.

- $d_{uv} = d(Y_u, Y_v)$, la distance entre les points Y_u et Y_v.

La structure est dite préservée si la distance entre deux observations est assez proche de la distance entre leurs projections, soit :

$$d^*_{uv} \approx d_{uv} \qquad 1 \le u < v \le N \tag{13}$$

Les algorithmes de projection diffèrent par le choix du type de distance et de la fonction erreur qui représente la qualité de l'ajustement de la configuration de l'ensemble des projections par rapport à l'ensemble des observations.

La distance la plus employée est la distance Euclidienne:

$$d^*_{uv} = \sqrt{\sum_{d=1}^{D} (x_{ud} - x_{vd})^2} \text{ et } d_{uv} = \sqrt{\sum_{m=1}^{M} (y_{um} - y_{vm})^2} . \tag{14}$$

Pour comparer la structure des observations dans leur espace d'origine de dimension D à leur structure dans l'espace de projection de dimension M, Sammon propose la fonction erreur E, appelée "stress de Sammon", définie par:

$$E = \frac{1}{C} \sum_{1 \le u < v \le N} \frac{(d^*_{uv} - d_{uv})^2}{d^*_{uv}}, \tag{15}$$

où

$$C = \sum_{1 \le u < v \le N} d^*_{uv} . \tag{16}$$

Cette fonction s'écrit sous une forme plus générale :

$$E_\alpha = \frac{1}{C_\alpha} \sum_{1 \le u < v \le N} (d^*_{uv})^\alpha (d^*_{uv} - d_{uv})^2 , \tag{17}$$

où C_α est défini par [Sie 88]:

$$C_\alpha = \sum_{1 \le u < v \le N} (d^*_{uv})^{\alpha+2} . \tag{18}$$

On retrouve le critère de Sammon pour $\alpha = -1$.

Contrairement à l'ACP, qui est une méthode de projection linéaire analytique, l'algorithme de Sammon effectue une projection non linéaire et non analytique.

Cependant, l'algorithme de Sammon présente deux limitations majeures. La première est relative au temps de calcul qui est de l'ordre de N^3 alors que le temps de calcul de l'ACP est de l'ordre N. Lorsque le nombre d'observations est très grand, il est cependant possible de réduire ce nombre en utilisant l'algorithme des nuées dynamiques [Cel 89], ou tout autre algorithme de réduction de données et d'appliquer l'algorithme de Sammon sur ces données réduites pour se ramener à un temps de calcul raisonnable. La deuxième limitation de l'algorithme de Sammon provient du fait que si on désire projeter une nouvelle observation, il faut reexécuter l'algorithme en incluant cette nouvelle observation dans l'ensemble des observations.

Dans la suite nous montrons les résultats obtenus par l'algorithme de Sammon appliqué sur l'exemple de quatre classes Gaussiennes et celui des Iris de Fisher.

Algorithme de Sammon

1. Initialisation:

   - Fixer le nombre d'itérations maximum T_f.

   - Choisir aléatoirement une configuration de vecteurs $Y_1, Y_2, ..., Y_N$. (On peut choisir comme configuration initiale, la projection obtenue par l'ACP).

2. Itérations:

   - Calculer les distances d^*_{uv} et d_{uv}, u=1, ..., N-1 et v = u+1, ..., N,

   - Calculer E (cf. équation (15)).

3. Actualisation de la configuration dans le plan à l'itération de rang t+1,

   - Pour n = 1, ..., N, calculer:

$$Y_n(t+1) = Y_n(t) - \eta(t)\Delta_n(t) \text{ ; où } \Delta_n(t) = \left(\frac{\partial E}{\partial Y_n(t)}\right) \Big/ \left(\frac{\partial^2 E}{\partial\left(Y_n^2(t)\right)}\right).$$

4. Incrémentation de t (t=t+1) et aller en 2 jusqu'à ce que $t = T_f$.

Exemple 1: Quatre classes Gaussiennes

La figure 1.5 montre le plan de projection obtenu par cet algorithme. L'erreur E associée à cette procédure est égale à 0,02 (cf. équation 15). On peut remarquer que les classes sont projetées sur le plan en respectant leurs structures dans l'espace d'origine.

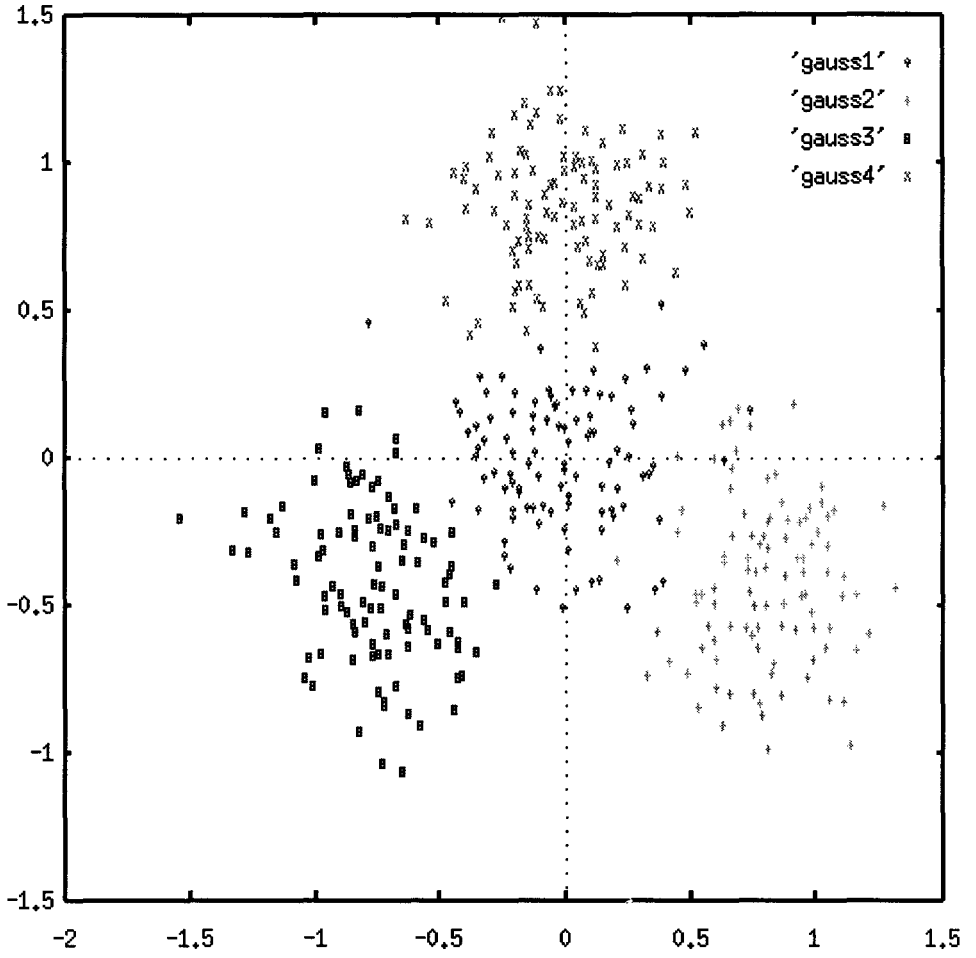


Figure 1.5. Plan de projection des quatre classes Gaussiennes obtenu par l'algorithme de Sammon, E = 0,02.

Exemple 2: Iris de Fisher

La figure 1.6 montre la projection plane des observations obtenue par l'algorithme de Sammon. L'erreur E est ici égale à 0,004. Notons que la classe des iris Setosa apparaît bien séparée des classes Versicolor et Verginica. Ces deux dernières présentent un taux de chevauchement moins important que celui obtenu par l'ACP (voir figure 1.4).

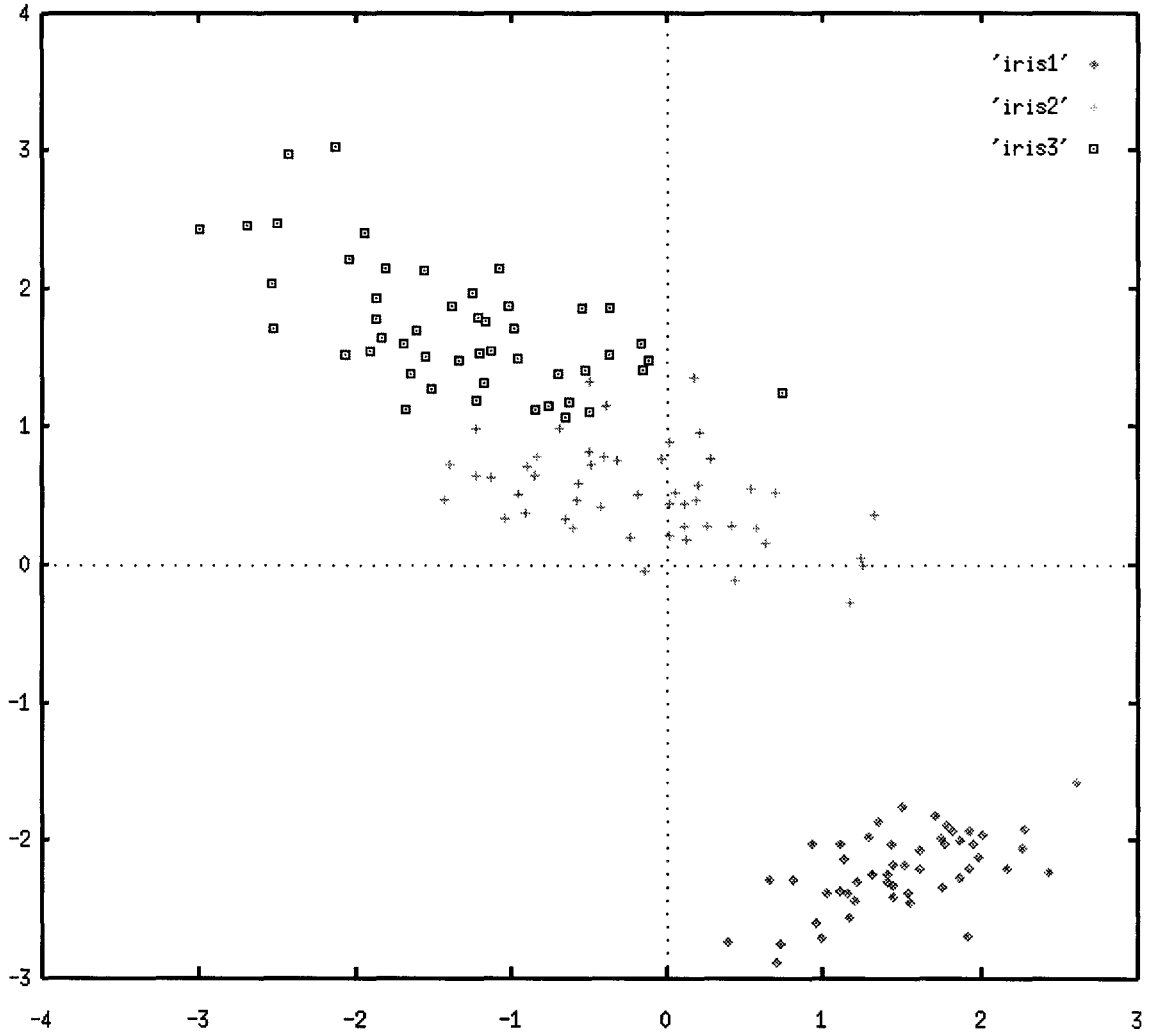


Figure 1.6. Plan de projection des Iris de Fisher obtenu par l'algorithme deSammon,

$E = 0{,}004$.

1.5. Conclusion

Dans ce chapitre nous avons exposé trois méthodes classiques de projection linéaires et non linéaires.

La méthode ACP est la méthode la plus connue pour la projection linéaire. Son principal attrait est qu'elle projette les données avec une perte minimale d'information. La méthode définit des axes de projection par des combinaisons linéaires des coordonnées initiales en privilégiant les axes de variances maximales. Dans ce sens la densité locale des observations n'est pas prise en compte.

La méthode des projections révélatrices est aussi une méthode de projection linéaire, mais contrairement à l'ACP, elle tient compte de la densité locale des observations dans le processus de projection. Son principal inconvénient est la difficulté de régler ses paramètres, rendant délicate sa mise en oeuvre pratique.

L'algorithme de Sammon figure au premier rang des méthodes de projection non linéaires regroupées sous le vocable de Multi-Dimensional Scaling. L'algorithme génère une configuration de points dans l'espace de projection sans fournir une fonction analytique liant l'espace d'observation à l'espace de projection. Comme la méthode de projections révélatrices, il tient compte de la densité locale des données. Par contre, son temps de calcul est assez long pour un ensemble d'observations de taille importante.

Les deux chapitres suivants présentent des méthodes de projection récentes par réseaux de neurones. Contrairement aux méthodes de projection classiques, elles apportent les avantages liés aux réseaux de neurones, à savoir l'apprentissage en ligne et l'implantation possible sur une architecture matérielle parallèle.

Chapitre 2

Réseaux de neurones à apprentissage par la règle de Hebb pour l'analyse en composantes principales

Chapitre 2 Réseaux de neurones à apprentissage par la règle de Hebb pour l'analyse en composantes principales

2.1 Introduction

Au cours du chapitre précédent, nous avons présenté des méthodes classiques linéaires et non-linéaires pour la projection de données multidimensionnelles. Parmi les méthodes de projection plane, l'ACP est la plus utilisée.

Dans ce chapitre, nous nous intéresserons aux réseaux de neurones multicouches à apprentissage non supervisé pour la projection plane. L'architecture de tels réseaux consiste en une couche d'entrée et une couche de sortie, avec ou sans couches cachées. Deux types de connexions existent: les connexions inter-couches transmettant les informations présentées à la couche d'entrée vers la couche de sortie et les connexions intra-couches ou latérales. En général, l'apprentissage consiste à modifier de façon répétitive les poids de toutes les connexions en réponse aux observations présentées aux entrées du réseau et en accord avec des règles prédéfinies jusqu'à obtenir une configuration de poids finale stable.

Une caractéristique importante des réseaux de neurones est leur capacité à élaborer une transformation analytique faisant appel à une fonction linéaire ou non-linéaire entre l'espace d'observation multidimensionnel et l'espace de projection [Dao 93], [Mao 94], [Bet 95]. La figure 2.1 présente des architectures neuronales pour la projection plane développées ces dernières années. Les règles d'apprentissage de ces réseaux sont élaborées soit à partir de constatations biologiques comme la règle de Hebb généralisée, soit tout simplement à partir de l'optimisation d'un critère mathématique.

Dans la suite nous présentons le neurone extracteur de la composante principale qui est l'élément de base des réseaux de neurones pour l'ACP.

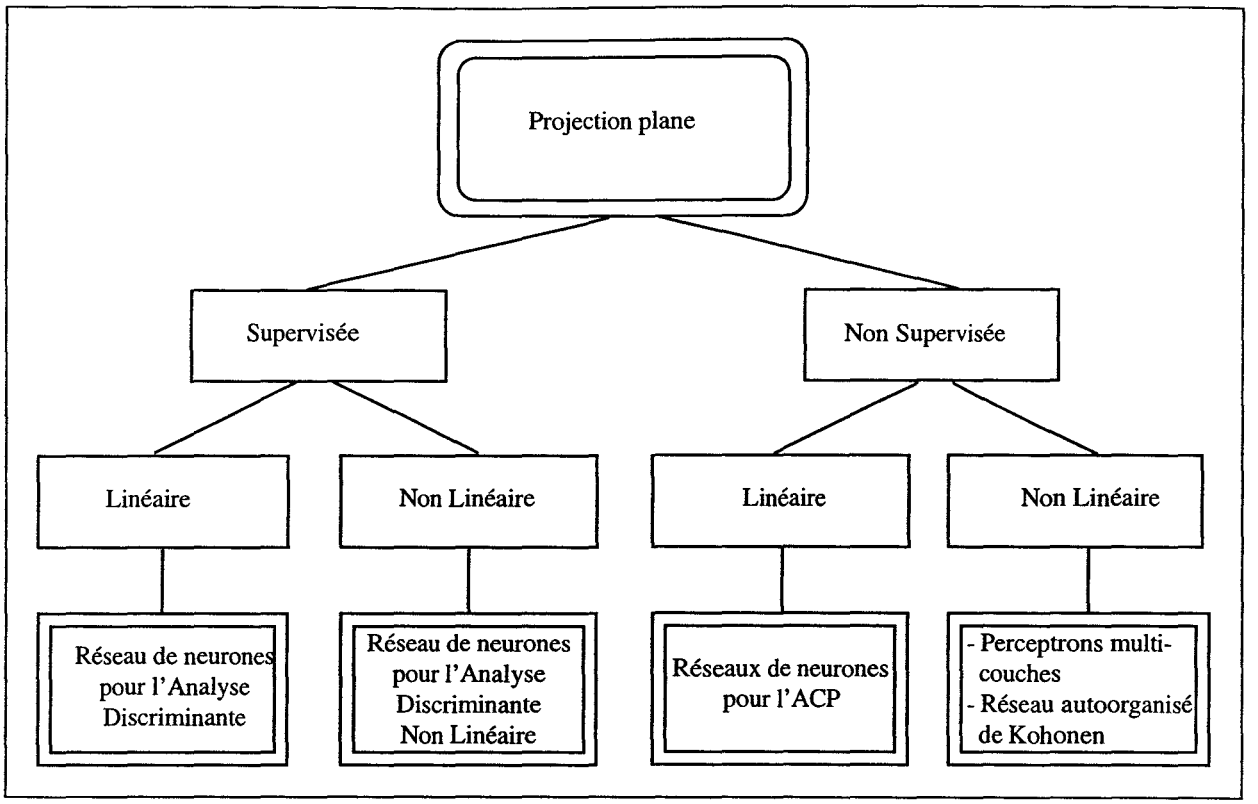


Figure 2.1: Familles des réseaux de neurones pour la projection plane.

2.2 Neurone extracteur de la composante principale

Le neurone extracteur de la composante principale, représenté sur la figure 2.2, est constitué d'un ensemble de D entrées dont le rôle est de recevoir les vecteurs observations et de les transmettre à la sortie du neurone grâce à des connexions pondérées [Hay 94].

L'apprentissage en ligne du neurone s'effectue en présentant à ses entrées, de façon répétitive, les composantes des vecteurs observations, X_1, X_2, ..., X_n, ..., X_N, qui définissent la base d'apprentissage du neurone. Plus précisément, il s'agit à chaque instant t d'effectuer un tirage au hasard d'un vecteur observation de cette base et de le présenter à l'entrée du neurone. Soit $X(t) = (x_1(t), x_2(t), ..., x_d(t), ..., x_D(t))^T$ le vecteur observation tiré à l'instant t et $W(t) = (w_1(t), w_2(t), ..., w_d(t), ..., w_D(t))^T$ le vecteur poids correspondant.

A la présentation d'un vecteur X(t) à l'instant t, la sortie du neurone est donnée par l'équation

$$y(t) = \sum_{d=1}^{D} w_d(t)\, x_d(t) \qquad (1)$$

qui s'écrit sous la forme vectorielle:

$$y(t)=W(t)^T X(t) \quad (2)$$

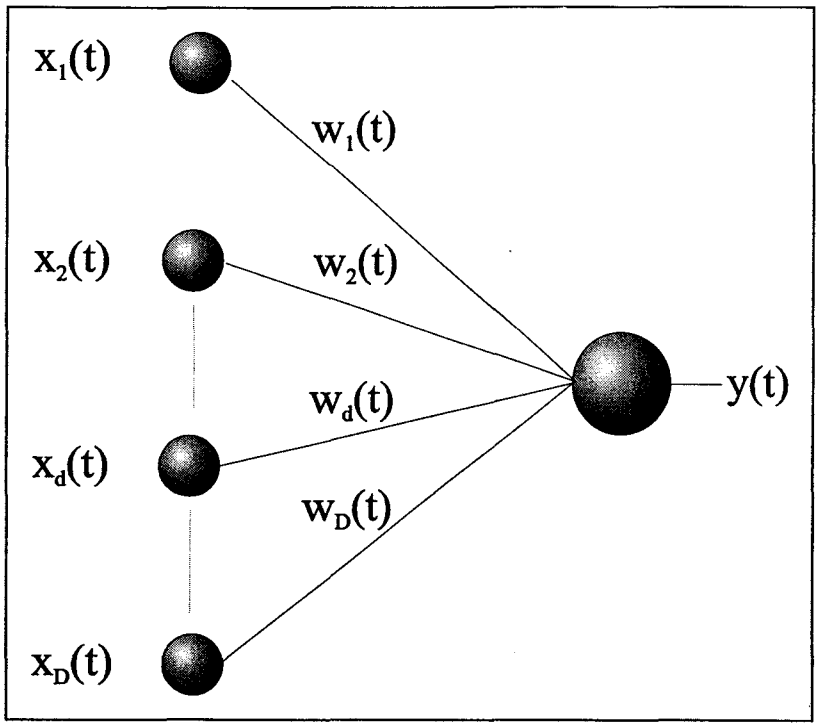


Figure 2.2: Neurone extracteur de la composante principale

Apprentissage par la règle de Hebb

Dans le contexte d'apprentissage d'un neurone biologique, la règle de Hebb stipule que *le poids d'une connexion reliant deux neurones augmente lorsque ces deux neurones sont activés en même temps.* L'expression mathématique de la règle de Hebb s'écrit:

$$w_d(t+1) = w_d(t) + \eta(t) y(t) x_d(t), \quad d = 1,\ldots,D \quad (3)$$

où $\eta(t)$ est un coefficient d'apprentissage qui peut être soit une constante, $\eta(t)=\eta(0)$, positive et inférieure à 1, soit une variable décroissante en fonction des itérations.

Bien que résultant de motivations biologiques, cette règle peut être reliée à une stratégie de montée du gradient:

$$w_d(t+1) = w_d(t) + \eta(t) \frac{\partial J}{\partial w_d} \quad (4)$$

où J est un critère d'optimisation défini par :

$$J(w) = \frac{1}{2} y^2 \tag{5}$$

En dérivant J par rapport à w et en employant l'équation (1), on obtient :

$$w_d(t+1) = w_d(t) + \eta(t) y(t) x_d(t), \quad d = 1, \ldots, D \tag{6}$$

qui est bien la règle de Hebb.

Apprentissage par la règle de Hebb généralisée linéaire

La règle d'apprentissage de Hebb, dans sa forme de base, conduit à une augmentation non bornée du module des poids w_d. Oja a introduit la règle de Hebb généralisée qui est une normalisation de la règle de Hebb standard. L'emploi d'un facteur de normalisation a pour effet d'introduire une compétition entre les poids du neurone, ce qui est essentiel pour la stabilité.

D'un point de vue mathématique, une normalisation convenable répondrait à celle introduite par Oja [Oja 89] :

$$w_d(t+1) = \frac{\tilde{w}_d(t+1)}{\left(\sum_{d=1}^{D} \tilde{w}_d^2(t+1) \right)^{\frac{1}{2}}} \tag{7}$$

où:

$$\tilde{w}_d(t+1) = w_d(t) + \eta(t) y(t) x_d(t), \quad d = 1, \ldots, D \tag{8}$$

En remplaçant l'équation (8) dans l'équation (7) et en développant en série de Taylor l'équation (7) pour un coefficient d'apprentissage $\eta(t)$ proche de 0, on obtient une nouvelle règle d'apprentissage définie par l'équation suivante [Oja 89] :

$$w_d(t+1) = w_d(t) + \eta(t) y(t) (x_d(t) - y(t) w_d(t)) \ . \tag{9}$$

Cette règle d'apprentissage constitue l'élément de base de l'apprentissage des réseaux de neurones pour l'ACP. Toutes les autres règles sont des variétés ou des généralisations de celle-ci. Compte tenu de son importance, nous montrons la démarche aboutissant à son élaboration.

Le remplacement de l'équation (8) dans l'équation (7) donne :

$$w_d(t+1) = \frac{w_d(t) + \eta(t).y(t).x_d(t)}{\left(\sum_{d=1}^{D}(w_d(t) + \eta(t).y(t).x_d(t))^2\right)^{\frac{1}{2}}} \quad (10)$$

Appelons U(η) l'inverse du dénominateur de l'équation (10) :

$$U(\eta(t)) = \left(\sum_{d=1}^{D}(w_d(t) + \eta(t).y(t).x(t))^2\right)^{-\frac{1}{2}} \quad (11)$$

Développons maintenant en série de Taylor l'équation (11), pour un coefficient d'apprentissage η proche de 0, selon l'équation suivante :

$$U(\eta) = U(0) + \eta\left(\frac{\partial U}{\partial \eta}\right)_{\eta=0} + O(\eta^2) \quad (12)$$

$$U(0) = \left(\sum_{d=1}^{D} w_d^2(t)\right)^{-\frac{1}{2}} \quad (13)$$

Comme le vecteur poids w(t) est normé, on déduit :

$$U(0)=1 \quad (14)$$

La dérivée de U par rapport à η est donnée par:

$$\frac{\partial U}{\partial \eta} = -y(t)x_d(t)\left(\sum_{d=1}^{D} w_d(t) + \eta(t)y(t)x_d(t)\right)U^3(\eta) \quad (15)$$

soit pour η=0:

$$\left(\frac{\partial U}{\partial \eta}\right)_{\eta=0} = -y(t)x_d(t)\left(\sum_{d=1}^{D} w_d(t)\right)U^3(0) \quad (16)$$

Comme U(0) =1, l'équation (16) devient:

$$\left(\frac{\partial U}{\partial \eta}\right)_{\eta=0} = -y(t)\left(\sum_{d=1}^{D} w_d(t)x_d(t)\right) \quad (17)$$

Or d'après l'équation (1), le terme entre parenthèses de l'équation (17) n'est autre que y(t), d'où:

$$\left(\frac{\partial U}{\partial \eta}\right)_{\eta=0} = -y^2(t) \tag{18}$$

Remplaçons les équations (14) et (18) dans l'équation (12). En négligeant le terme du 2ème ordre $O(\eta(t))$, on obtient:

$$U(\eta(t)) = 1 - \eta(t)y^2(t) \tag{19}$$

L'équation (10) devient alors:

$$w_d(t+1) = \left(w_d(t) + \eta(t)y(t)x_d(t)\right)\left(1 - \eta(t)y^2(t)\right) \tag{20}$$

En négligeant le terme η^2, on obtient finalement :

$$w_d(t+1) = w_d(t) + \eta(t)y(t)\left(x_d(t) - y(t)w_d(t)\right). \tag{21}$$

Le terme $+y(t)x_d(t)$ dans l'équation (9) représente la modification apportée par la règle de Hebb alors que le terme $-y^2(t)w_d(t)$ est introduit pour assurer la stabilité de la règle d'apprentissage.

Le coefficient d'apprentissage $\eta(t)$ se présente sous la forme d'une séquence de nombres réels positifs vérifiant les conditions [Hay 94]:

$$\begin{aligned} &\sum_{t=0}^{\infty} \eta(t) = \infty, \\ &\sum_{t=0}^{\infty} \eta^{\alpha}(t) < \infty \quad \text{pour} \quad \alpha > 1, \\ &\lim_{t \to \infty} \eta(t) = 0. \end{aligned} \tag{22}$$

La propriété intéressante de la règle de Oja est que, quand le rang d'itération t tend vers l'infini, le vecteur poids W converge vers le vecteur propre correspondant à la valeur propre maximale de la matrice de covariance de l'ensemble des observations [Hay 94].

2.3 Réseau de neurones linéaire à apprentissage par la règle de Hebb généralisée (GHA)

L'architecture des réseaux de neurones extracteurs des composantes principales est constituée d'une couche d'entrée et d'une couche de sortie à fonctions d'activation linéaires [Oja 89],

[San 89], [Hay 94]. L'apprentissage de ces réseaux est non supervisé et basé sur la règle de Hebb généralisée. Le principal intérêt de ce type de réseaux à fonctions d'activations linéaires est de faciliter la compréhension du fonctionnement des réseaux employant des fonctions d'activation non linéaires qui seront présentés dans le chapitre suivant.

Sanger a adapté la règle de Hebb afin qu'elle puisse s'appliquer à une couche composée de plusieurs neurones à fonctions d'activation linéaires, le but étant, évidemment, d'arriver à un réseau permettant d'extraire un nombre M de composantes principales, M étant inférieur ou égal à la dimension D de l'espace d'observation.

Sanger formalisa l'algorithme de Hebb généralisé, (GHA pour Generalized Hebbian Algorithm), en étendant la règle de Oja, définie par l'équation (9), à un réseau à propagation avant composé de D neurones d'entrée dont le rôle est de transmettre les valeurs des composantes du vecteur d'observation aux neurones de sortie et M neurones de sortie (cf. figure 2.3.) [Sang 89], [Hay 94].

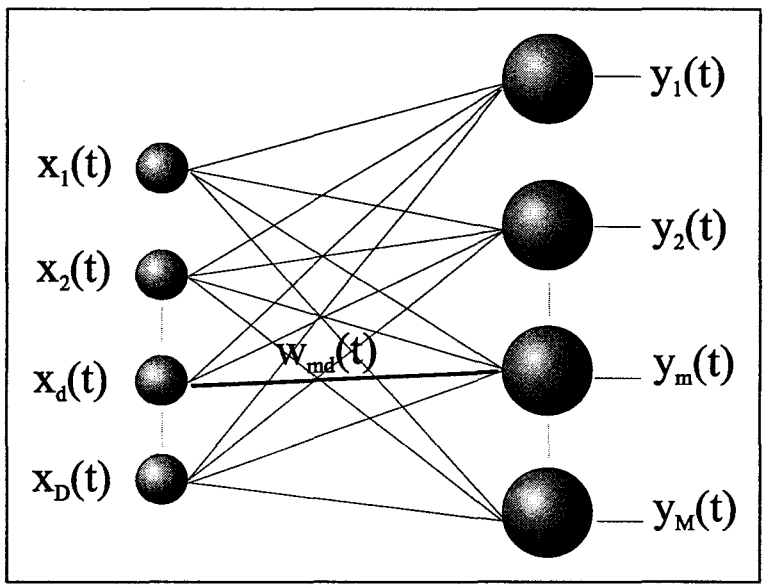


Figure 2.3: Réseau de neurones linéaire à apprentissage par la règle de Hebb généralisée

La sortie $y_m(t)$ du neurone d'indice m à l'itération de rang t, obtenue suite à la présentation du vecteur $(x_1(t), ..., x_d(t), ..., x_D(t))^T$ est donnée par:

$$y_m(t) = \sum_{d=1}^{D} w_{md}(t)\, x_d(t), \quad m = 1, \ldots, M, \tag{23}$$

w_{md} étant le poids de la connexion reliant l'entrée d'indice d au neurone d'indice m.

L'équation (23) s'écrit aussi sous la forme du produit scalaire:

$$y_m(t) = W_m^T(t)\,X(t), \quad m = 1, \ldots, M, \tag{24}$$

avec:

$$W_m(t) = (w_{m1}(t), w_{m2}(t), \ldots, w_{mD}(t))^T.$$

La règle d'actualisation de Sanger du poids synaptique w_{md} de la connexion reliant l'entrée d au $m^{ème}$ neurone est la suivante:

$$w_{md}(t+1) = w_{md}(t) + \eta(t) y_m(t)\left(x_d(t) - \sum_{\ell=1}^{m} y_\ell(t) w_{\ell d}(t) \right), \tag{25}$$

pour d = 1, ..., D et m = 1, ..., M.

L'équation (25) s'écrit aussi sous forme vectorielle:

$$W_m(t+1) = W_m(t) + \eta(t) y_m(t)\left(X(t) - \sum_{\ell=1}^{m} y_\ell(t) W_\ell(t) \right). \tag{26}$$

On peut remarquer que le terme $\sum_{\ell=1}^{m} y_\ell(t) W_\ell(t)$ n'est autre qu'une estimation du vecteur observation X(t):

$$\hat{X}(t) = \sum_{\ell=1}^{m} y_\ell(t) W_\ell(t) \tag{27}$$

à partir des neurones d'indices 1, 2, ..., m. Dans ce sens, l'algorithme d'apprentissage est appelé algorithme de re-estimation.

Avec cette règle, les vecteurs poids W_1, W_2, ..., W_M associés aux neurones d'indices 1, 2, ..., M, convergent, quand le nombre d'itérations tend vers l'infini, vers les M vecteurs propres de la matrice de covariance de l'ensemble des observations correspondant aux M plus grandes valeurs propres.

Notons que la règle de Oja est un cas particulier de cette règle correspondant au premier neurone du réseau GHA de Sanger (obtenue pour m=1).

Algorithme GHA

1. Initialiser les poids du réseau à des valeurs aléatoires voisines de zéro. Affecter une valeur positive faible au paramètre $\eta(t)$ et fixer le nombre maximal d'itérations T_f.
2. Sélectionner une observation X(t) de la base d'apprentissage et la présenter au réseau.
3. Pour m = 1, ..., M,

   - calculer les équations du réseau: $y_m(t) = \sum_{d=1}^{D} w_{md}(t)\, x_d(t)$

   - actualiser les poids: $W_m(t+1) = W_m(t) + \eta(t) y_m(t)\left(X(t) - \sum_{\ell=1}^{m} y_\ell(t) W_\ell(t) \right)$

4. Incrémenter t (t= t+1) et aller en 2 jusqu'à ce que $t = T_f$.

2.4 Réseau de neurones adaptatif (APEX)

Le réseau de neurones décrit dans le paragraphe précédent s'appuie uniquement sur l'emploi des connexions reliant les entrées vers les sorties du réseau pour extraire les composantes principales. Les neurones de sortie ne sont pas reliés entre-eux. Le réseau de neurones adaptatif, (APEX pour Adaptive Principal Extraction), contrairement au réseau GHA, utilise des connexions d'inhibition entre les neurones de sortie [Föl 89], [Hay 94]. Son architecture est représentée sur la figure 2.4 qui fait apparaître deux types de connexions:

- Les connexions excitatrices relient les neurones d'entrée du réseau aux neurones de sortie.
- Les connexions inhibitrices, latérales, relient chacun des neurones de la couche de sortie aux neurones d'indices supérieurs. Par l'intermédiaire de ces connexions, chaque neurone de sortie peut inhiber tous les neurones de la même couche et d'indices supérieurs.

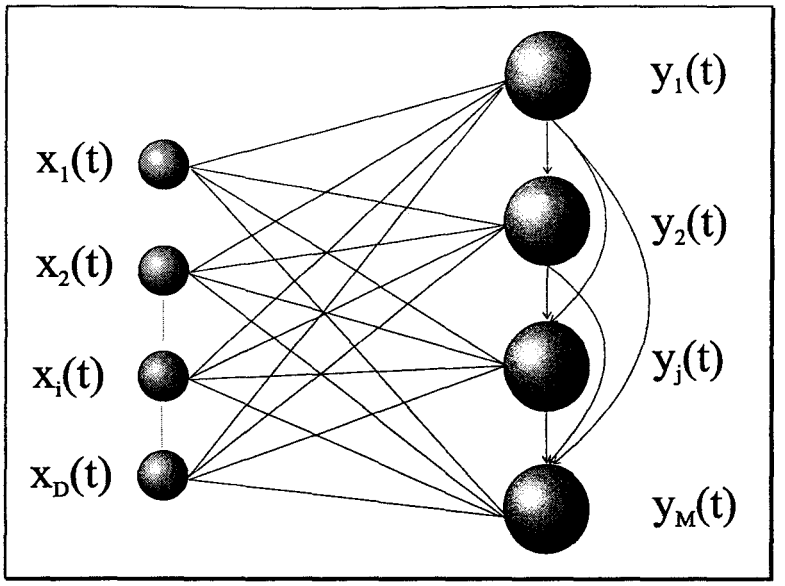


Figure 2.4: Réseau de neurones adaptatif

La sortie du neurone d'indice m de la couche de sortie est définie par:

$$y_m(t) = W_m^T(t)X(t) + \sum_{\ell=1}^{m-1} v_{m\ell}(t)\, y_\ell(t) \quad , \quad m = 1, \ldots, M. \tag{28}$$

L'équation (28) s'écrit aussi sous la forme plus condensée:

$$y_m(t) = W_m^T(t)X(t) + V_m^T(t)Y_{m-1}(t) \tag{29}$$

où le vecteur $Y_{m-1}(t) = \left(y_1(t), y_2(t), \ldots, y_{m-1}(t)\right)^T$, de dimension m-1, est le vecteur des sorties des neurones de la couche de sortie connectés au neurone m de cette même couche. $V_m(t) = \left(v_{m1}(t), v_{m2}(t), \ldots, v_{mm-1}(t)\right)^T$ est le vecteur poids de ces connexions.

Les poids des connexions excitatrices reliant les neurones d'entrée au neurone d'indice m sont actualisés selon la règle de Hebb:

$$W_m(t+1) = W_m(t) + \eta(t)\left[y_m(t)X(t) - y_{mj}^2(t)W_m(t)\right]. \tag{30}$$

Le terme $+y_m(t)X(t)$ de l'équation (30) représente la règle de Hebb classique alors que le terme $-y_m^2(t)W_m(t)$ est introduit pour assurer la stabilité de l'apprentissage.

Les poids des connexions latérales inhibitrices du neurone m sont actualisés selon la règle d'apprentissage dite anti-Hebb:

$$V_m(t+1) = V_m(t) - \eta(t)\left[y_m(t)Y_{m-1}(t) + y_m^2(t)V_m(t)\right], \qquad (31)$$

Le terme $-y_m(t)Y_{m-1}(t)$ de l'équation (31) représente la règle anti-Hebb alors que le terme $-y_m^2(t)V_m(t)$ est introduit pour assurer la stabilité de l'apprentissage.

Algorithme APEX

1. Initialiser les poids du réseau à des valeurs aléatoires voisines de 0, affecter une valeur positive faible au paramètre $\eta(t)$ et fixer le nombre maximal d'itérations T_f.

2. Sélectionner de façon aléatoire une observation X(t) de la base d'apprentissage et la présenter au réseau.

3. Pour m = 1,

- calculer les équations du réseau: $y_1(t) = W_1(t)X(t)$,

- actualiser les poids: $W_1(t+1) = W_1(t) + \eta(t)\left[y_1(t)X(t) - y_1^2(t)W_1(t)\right]$.

4. Pour m = 2, ..., M,

- calculer les équations du réseau: $y_m(t) = W_m^T(t)X(t) + V_m^T(t)Y_{m-1}(t)$,

où $Y_{m-1}(t) = \left(y_1(t), y_2(t), \ldots, y_{m-1}(t)\right)^T$,

- actualiser les poids: $W_m(t+1) = W_m(t) + \eta(t)\left[y_m(t)X(t) - y_m^2(t)W_m(t)\right]$,

$V_m(t+1) = V_m(t) - \eta(t)\left(y_m(t)Y_{m-1}(t) + y_m^2(t)v_m(t)\right)$.

4. Incrémenter t (t = t+1), et aller en 2 jusqu'à ce que $t = T_f$.

Lorsque le rang t de l'itération augmente le vecteur W_1 converge vers le vecteur propre associé à la plus grande valeur propre de la matrice de covariance Σ alors que les vecteurs W_m, m = 2,..., M, convergent vers les m vecteurs propres associés aux m valeurs propres de la matrice de covariance Σ ordonnés par valeurs décroissantes. Les vecteurs V_m m = 2, ..., M convergent vers 0. Les connexions latérales tendent à avoir des poids nuls et par conséquent le réseau tend vers l'architecture du réseau GHA.

2.5 Conclusion

Les deux types de réseaux de neurones à fonctions d'activations linéaires présentés dans les paragraphes précédents réalisent l'analyse en composantes principales. Les algorithmes d'apprentissage de ces réseaux reposent sur l'application de la règle de Hebb généralisée.

Le plan de projection obtenu par ce type de réseaux, comme celui de l'analyse en composantes principales, est celui des directions dans lesquelles les observations présentent les plus fortes dispersions. Par conséquent, il ne révèle pas toujours la structure locale des données ni la séparabilité des classes.

Lorsqu'on ajoute une fonction d'activation non linéaire, comme la fonction sigmoïde, sur les neurones de sortie, ces réseaux peuvent extraire plus d'informations que les réseaux précédents et peuvent ainsi mieux représenter la structure des données [Kar 94]. Dans le chapitre suivant, nous explorons d'autres types de réseaux de neurones réalisant la projection non linéaire.

Chapitre 3

Réseaux de neurones pour la projection non linéaire de données multidimensionnelles

Chapitre 3 Réseaux de neurones pour la projection non linéaire de données multidimensionnelles

3.1 Introduction

Nous avons vu, dans le chapitre précédent, que les réseaux de neurones à fonctions d'activation linéaires et à apprentissage par la règle de Hebb généralisée établissent une transformation linéaire entre l'espace des observations et l'espace des projections. Le plan de projection obtenu par ce type de réseaux est le même que celui résultant de l'analyse en composantes principales.

Récemment, de nouvelles architectures et règles d'apprentissage de réseaux de neurones ont été développées pour la projection non linéaire [Cot 88], [Bou 88], [Jai 92]. Certaines règles d'apprentissage font appel à des concepts d'apprentissage et d'autoorganisation d'inspiration biologique [Oja 95], [Koh 95]. D'autres résultent de l'optimisation d'un critère mathématique comme celui des moindres carrés [Rum 86] ou celui du stress de Sammon [Sam 69], [Jai 92]. L'intérêt de ces réseaux est qu'ils établissent une transformation non linéaire entre l'espace des observations et l'espace des projections tout en préservant en projection, autant que possible, la structure locale des observations multidimensionnelles.

Notre contribution dans ce chapitre se situe non seulement dans l'exploitation de ces réseaux pour la projection plane, mais aussi dans l'optimisation de leur architecture au moyen d'un critère informationnel.

3.2 Réseau de neurones à apprentissage par la règle de Hebb non linéaire

La figure 3.1 montre l'architecture du réseau qui est identique à celle présentée au paragraphe 2.3 du chapitre précédent. Elle est constituée de D entrées et de M neurones de sortie. A la présentation du vecteur observation X(t) aux entrées à l'instant t, les sorties du réseau sont données par l'équation:

$$y_m(t) = \sum_{d=1}^{D} w_{md}(t) x_d(t), \quad m = 1, \cdots, M, \tag{1}$$

qui s'écrit aussi sous la forme:

$$y_m(t) = W_m^T(t) X(t), \quad m = 1, \cdots, M. \tag{2}$$

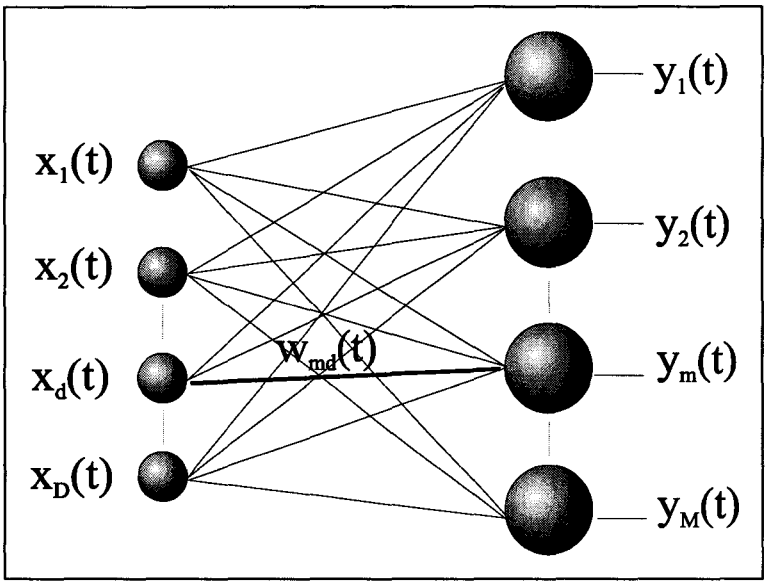


Figure 3.1: Réseau de neurones à apprentissage par la règle de Hebb non linéaire.

Oja a récemment proposé une version non linéaire de la règle d'apprentissage de Hebb généralisée exposée dans le chapitre précédent [Oja 95]. Pour un neurone de sortie $y_m(t)$, le critère utilisé est le logarithme du cosinus hyperbolique de cette sortie défini par:

$$J(W_m(t)) = \log\left[\cosh(y_m(t)\right], \tag{3}$$

qui s'écrit aussi en tenant compte de l'équation (2) sous la forme:

$$J(W_m(t)) = \log\left[\cosh\left(W_m^T(t) X(t)\right)\right]. \tag{4}$$

La règle d'apprentissage est déduite de la maximisation de $J(W_m(t))$ par la méthode de montée du gradient:

$$\tilde{W}_m(t+1) = W_m(t) + \eta(t) \frac{\partial J(W_m(t))}{\partial W_m(t)} \tag{5}$$

avec:

$$W_m(t+1) = \frac{\tilde{W}_m(t+1)}{\|W_m(t+1)\|} \tag{6}$$

En remplaçant l'équation (5) dans l'équation (6) et en développant en série de Taylor l'équation (6), on déduit la règle d'apprentissage suivante:

$$W_m(t+1) = W_m(t) + \eta(t)(I - W_m(t)W_m^T(t))\frac{\partial J(W_m(t))}{\partial W_m(t)}. \tag{7}$$

Le gradient de $J(W_m(t))$ par rapport à $W_m(t)$ s'écrit :

$$\frac{\partial J(W_m(t))}{\partial W_m(t)} = \tanh(y_m(t))X(t)). \tag{8}$$

Si on remplace l'équation (8) dans l'équation (7), on obtient :

$$W_m(t+1) = W_m(t) + \eta(t)(I - W_m(t)W_m^T(t))X(t)\tanh(y_m(t)) \tag{9}$$

Comme nous l'avons mentionné dans le chapitre précédent, le plan de projection obtenu par un réseau de neurones à apprentissage par la règle de Hebb linéaire est identique à celui obtenu par l'ACP. Il est défini par les axes de projection de variances maximales dans l'espace des observations. L'introduction de la non linéarité dans le critère à maximiser présente l'avantage de conserver en projection non seulement les statistiques d'ordres 1 et 2 (moyennes et variances), mais aussi les statistiques d'ordres supérieurs (kurtosis, etc.).

3.3 Réseau de neurones autoorganisé de Kohonen

L'architecture du réseau autoorganisé de Kohonen, appelé aussi carte de Kohonen, est inspirée de l'organisation biologique des neurones de certaines zones du cerveau [Koh 84]. Le réseau est constitué d'une couche d'entrée et d'une couche de sortie. Les neurones de la couche de sortie sont disposés sur une grille rectangulaire à deux dimensions (cf. figure 3.2).

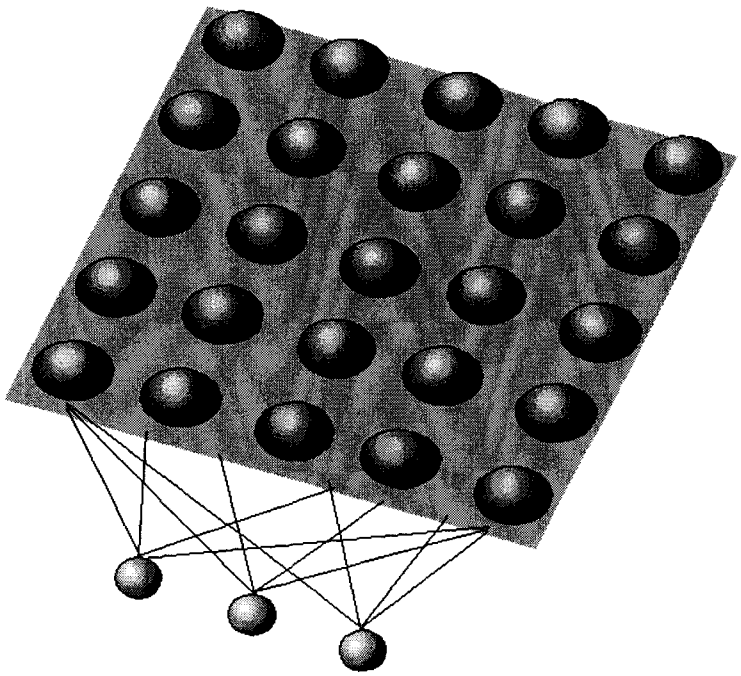


Figure 3.2: Réseau de neurones autoorganisé de Kohonen

Soient M le nombre total des neurones de la carte, $X(t) = (x_1(t), x_2(t)..., x_d(t), ..., x_D(t))^T$ le vecteur présenté à l'entrée du réseau et $W_m(t) = (w_{m1}(t), w_{m2}(t), ..., w_{md}(t), ..., w_{mD}(t))^T$ le vecteur poids du $m^{ème}$ neurone de la grille, m =1, ..., M.

A chaque présentation d'une observation X(t) de la base d'apprentissage à l'entrée du réseau, chacun des neurones calcule la distance entre son propre vecteur poids et cette observation. Le neurone dont le vecteur poids est le plus proche de l'observation est déclaré gagnant, sa sortie est mise à 1 et son vecteur poids est rapproché dans le sens de cette observation: c'est le principe de l'apprentissage compétitif. Soit k l'indice du neurone gagnant:

$$k = \text{Arg} \min_m \left\| X(t) - W_m(t) \right\| \tag{10}$$

Les sorties des neurones de la grille vérifient l'équation:

$$y_m(t) = \begin{cases} 1 & \text{si} \quad m = k \\ 0 & \text{si} \quad m \neq k \end{cases} \tag{11}$$

Au cours de l'apprentissage, Kohonen propose d'actualiser non seulement le vecteur poids du neurone gagnant mais aussi les vecteurs poids des neurones qui lui sont voisins sur la grille. Soit V(k,t) l'ensemble des neurones voisins du neurone k à l'instant t:

$$W_m(t+1) = \begin{cases} W_m(t) + \eta_{max}(t)(X(t) - W_m(t)) & \text{si } m = k \\ W_m(t) + \eta_m(t)(X(t) - W_m(t)) & \text{si } m \in V(k,t) \\ W_m(t) & \text{si } m \notin V(k,t) \text{ et } m \neq k. \end{cases} \quad (12)$$

$\eta_m(t)$ est un coefficient d'apprentissage qui est fonction du voisinage V(k, t). $\eta_m(t)$ prend sa valeur maximale $\eta_{max}(t)$ pour m = k.

Une forme simple de $\eta_m(t)$ est $\eta_m(t) = \begin{cases} \eta_0 & \text{si } m \in V(k,t) \\ 0 & \text{sinon} \end{cases}$

- η_0 est une valeur constante fixée par l'utilisateur.
- Au début de l'apprentissage, l'étendue du voisinage $V_m(k,t)$ du neurone gagnant est grande et décroît au cours de l'apprentissage.

Algorithme d'apprentissage

1. Initialiser les poids à des valeurs aléatoires faibles. Fixer le coefficient d'apprentissage et le nombre maximum d'itérations T_f,
2. Sélectionner une observation X(t) de la base d'apprentissage et la présenter au réseau,
3. Calculer pour chaque neurone m la distance: $\|X(t) - W_m(t)\|$,
4. Sélectionner le neurone gagnant k (cf. équation (10)),
5. Actualiser le vecteur poids du neurone gagnant et ceux de ses voisins (cf. équations (12)),
6. Répéter les étapes 2 à 5 jusqu'à ce que $t = T_f$.

L'apprentissage a pour effet que les neurones voisins sur la grille représentent des groupes d'observations voisins. De plus, les vecteurs poids de neurones voisins convergent vers les centres de groupes voisins. Le réseau crée un ordre topologique sur la grille qui reflète la structure des observations de la base d'apprentissage.

Pour exploiter le réseau en classification non supervisée, Ultsh représente les neurones par des pixels sur un écran graphique et affiche entre les pixels la distance séparant les vecteurs poids des neurones voisins [Ult 92]. Kraaijveld et al. proposent une solution plus simple en

affichant sur chacun des pixels un niveau de gris correspondant à la distance maximale entre le vecteur poids du neurone et ceux des neurones de son voisinage [Kra 92]. Grâce à cette représentation, on obtient une image en niveau de gris qui révéle la structure des observations de la base d'apprentissage. Ces extensions permettent d'exploiter le réseau de Kohonen dans le cadre de la classification interactive.

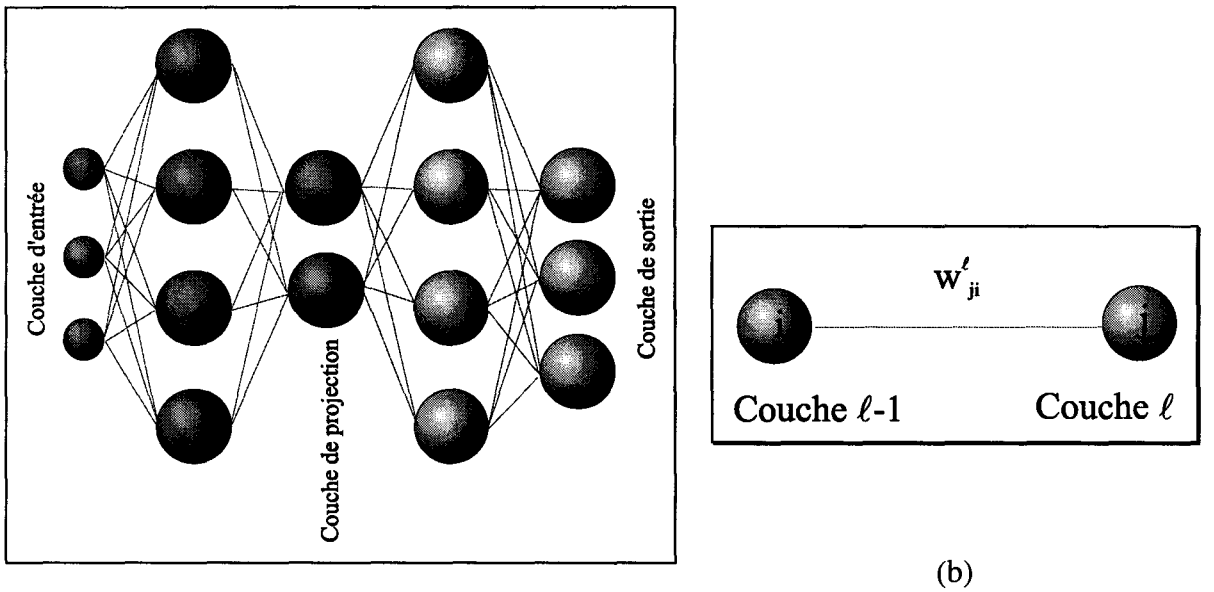
Dans les paragraphes suivants, nous présentons trois réseaux à architectures multicouches dont l'apprentissage s'effectue par la règle de retropropagation du gradient, mais avec des fonctions d'erreurs différentes.

3.4 Perceptron multicouche autoassociateur

L'architecture du Perceptron multicouche autoassociateur comprend cinq couches (cf. figure 3.3). Une couche d'entrée, trois couches cachées et une couche de sortie. Les couches d'entrée et de sortie ont la même dimension ; l'observation présentée à l'entrée du réseau est aussi présentée comme sortie désirée. De ce fait, le réseau doit fournir en sortie une estimation des observations présentées à son entrée.

La première et la troisième couche cachées ont souvent le même nombre de neurones. Le choix de ce nombre est délicat et peut être guidé par la théorie de l'information [Bet 95]. La dimension de la seconde couche cachée est volontairement réduite à deux pour permettre une visualisation plane. Le réseau présente une architecture symétrique par rapport à la couche cachée centrale. Les valeurs des sorties de cette couche fournissent les projections bidimensionnelles des observations présentées à l'entrée du réseau.

On peut remarquer que ce réseau est constitué de deux réseaux à trois couches "montés" en cascades. Le premier réseau est appelé "réseau projecteur", (partie sombre sur la figure 3.3 (a)). Le second est appelé "réseau reconstructeur" [Dao 93], [Bet 95], [Ham 95]. Ce type de réseau fournit à la fois la projection des observations et la reconstruction de ces observations à partir des projections.



(a) (b)

Figure 3.3: (a) Percetron multicouche autoassociateur à apprentissage non supervisé, (b) liaison du $i^{ème}$ neurone de la couche ℓ-1 vers le $j^{ème}$ neurone de la couche ℓ suivante.

A la présentation d'une observation X(t) à l'entrée du réseau, à l'instant t, les équations de sorties des neurones de la $\ell^{ème}$ couche en fonction des neurones de la $(\ell\text{-}1)^{ème}$ couche précédente sont données par (cf. figure 3.3 (b)):

$$y_j^{\ell}(t) = s\left(\sum_{i=1}^{n_{\ell-1}} w_{ji}^{\ell}(t) y_i^{\ell-1}(t)\right), \ \ell = 1, ..., 4, \tag{13}$$

où

- $y_d^0(t)$, $d = 1, \cdots D$, représentent les composantes de l'observation présentée à l'entrée du réseau,
- $y_d^4(t)$, $d = 1, \cdots D$, représentent les composantes de l'estimation $\hat{X}(t)$ obtenue à la sortie du réseau,
- $n_{\ell-1}$ désigne le nombre des neurones sur la couche ℓ-1.
- $s(\alpha)$ est une fonction d'activation définie par:

$$s(\alpha) = \frac{1}{1+e^{-\alpha}} \tag{14}$$

La fonction d'activation d'un neurone est une fonction continue, dérivable et bornée entre 0 et 1. La dérivée de cette fonction est donnée par:

$$s'(\alpha) = s(\alpha)\big(1 - s(\alpha)\big)$$

Les observations doivent être normalisées pour se situer à l'intérieur d'un hypercube de côté unité, avant d'être exploitées pour l'apprentissage. Cet apprentissage est effectué par la règle classique de rétropropagation du gradient de l'erreur [Rum 86].

Apprentissage par rétropropagation du gradient de l'erreur

Soit $E(t)$ la fonction erreur au carré entre la sortie désirée $X(t)$ qui est aussi l'entrée du réseau et la sortie $\hat{X}(t)$ estimée par ce réseau. Les composantes $\hat{x}_d(t)$ et $y_d^4(t)$, $d = 1, \cdots, D$ sont identiques:

$$E(t) = \frac{1}{2}\left\|X(t) - \hat{X}(t)\right\|^2$$

$$E(t) = \frac{1}{2}\sum_{d=1}^{D}(x_d(t) - \hat{x}_d(t))^2 \qquad (15)$$

La règle d'actualisation des poids w_{ji}^{ℓ} consiste à exploiter la sensibilité de la fonction erreur E(t) par rapport aux poids du réseau:

$$w_{ji}^{\ell}(t+1) = w_{ji}^{\ell}(t) - \eta(t)\frac{\partial E(t)}{\partial w_{ji}^{\ell}} \qquad (16)$$

Le terme $\dfrac{\partial E(t)}{\partial w_{ji}^{\ell}}$ est déduit des équations (13) (14) et (15) par la règle de dérivation en chaîne [Rum 86].

A partir des équations (13)-(16) on déduit:

$$w_{ji}^{\ell}(t+1) = w_{ji}^{\ell}(t) + \eta(t)\delta_j^{\ell}(t)y_i^{(\ell-1)}(t) \qquad (17)$$

Pour les poids aboutissant à la couche de sortie $\ell = 4$:

$$\delta_d^4(t) = e_d^4(t)\hat{x}_d(t)\big(1 - \hat{x}_d(t)\big), \quad d = 1, 2, \cdots, D, \qquad (18)$$

$$e_d^4(t) = x_d(t) - \hat{x}_d(t).$$

Pour les poids aboutissant à une couche cachée ℓ, ℓ <4:

$$\delta_j^{\ell}(t) = y_j^{\ell}(t)\left(1 - y_j^{\ell}(t)\right)\sum_{k=1}^{n_{\ell-1}} \delta_k^{\ell+1}(t) w_{kj}^{\ell+1}(t) \qquad (19)$$

A la fin de l'apprentissage, les poids du réseau sont fixés et seul le "réseau projecteur" est conservé pour effectuer la projection. Les observations de la base d'apprentissage sont à nouveau présentées séquentiellement à l'entrée du "réseau projecteur" et les valeurs obtenues en sortie sont affichées sur un plan de projection.

Bourlard et Kamp ont montré que le plan de projection obtenu par un Perceptron multicouche autoassociateur à fonctions d'activations linéaires est celui de l'ACP [Bou 88]. Cottrel et al. ont exploité la couche de projection du réseau dans le cadre de la compression d'images [Cot 88]. Dans [Dao 93], ce réseau a été utilisé pour la classification interactive. Dans ce travail nous avons employé ce réseau pour la projection plane et proposé des critères informationnels pour l'optimisation du nombre des neurones sur les couches cachées [Bet 95] [Ham 95].

Algorithme d'apprentissage

1. Initialiser les poids du réseau aléatoirement à des valeurs voisines de 0. Fixer le coefficient d'apprentissage $\eta(t)$ et le nombre maximal d'itérations T_f,

2. Choisir de façon aléatoire une observation X(t) de la base d'apprentissage et la présenter au réseau,

3. Calculer les équations (13) et (15) du réseau,

4. Actualiser les poids du réseau (équation (17)),

5. Incrémenter t (t = t +1) et aller en 2 jusqu'à ce que t = T_f.

Optimisation de l'architecture du réseau

Dans un Perceptron multicouche autoassociateur, le nombre des neurones de la couche d'entrée est égal au nombre des neurones de la couche de sortie qui est aussi égal à la dimension D de l'espace d'observation. Comme le nombre des neurones sur la couche de projection est fixé à deux, seuls les nombres des neurones de la première et la troisième

couches cachées sont inconnus. Pour simplifier le problème on impose que les deux couches ont le même nombre H de neurones et nous proposons d'optimiser ce nombre.

L'une des méthodes permettant de choisir le nombre H de neurones des couches cachées consiste à utiliser des fonctions exprimant un compromis entre le nombre total des poids du réseau et le logarithme de l'erreur moyenne de reconstitution des observations. Kramer a utilisé deux fonctions proposées par Akaike dans le cadre de l'identification des séries temporelles [Kra 91]: l'erreur de prédiction finale (FPE pour Final Prediction Error) :

$$FPE(H) = \overline{E}\frac{1+\frac{N_p}{ND}}{1-\frac{N_p}{ND}}, \tag{20}$$

et le critère informationnel d'Akaike (AIC pour Akaike Information Criterion) [Aka 74] :

$$AIC(H) = Ln(\overline{E}) + \frac{2N_p}{ND}, \tag{21}$$

où :

- Ln(.) est le logarithme népérien,
- $\overline{E}$ est la fonction erreur moyenne définie par :

$$\overline{E} = \frac{1}{2ND}\sum_{n=1}^{N}\sum_{d=1}^{D}(x_{nd} - \hat{x}_{nd})^2, \tag{22}$$

- $N_p = [H(D+1)]+[2(H+1)]+[H(2+1)]+[D(H+1)]$ représente le nombre total des poids du réseau. Les termes entre crochets indiquent le nombre de connexions entre deux couches successives en supposant la présence d'un neurone biais sur chaque couche, sauf sur la couche de sortie. Un neurone biais est un neurone dont l'entrée est toujours à 1.
- ND est le nombre d'éléments de la base d'apprentissage.

Nous avons utilisé ces fonctions pour optimiser l'architecture du réseau autoassociateur. Le principe consiste à faire varier le nombre de neurones sur la première et la troisième couche cachée et à calculer les valeurs de FPE et AIC pour chaque nombre de neurones. Le nombre de neurones minimisant ces deux fonctions est retenu [Bet 95], [Ham 95].

3.5 Perceptron multicouche de Mao pour la projection non linéaire

Récemment, un Perceptron multicouche à apprentissage par rétropropagation du gradient du stress de Sammon a été élaboré par Mao et appelé SAMANN (Sammon Artificial Neural Network [Mao 94]). Le réseau admet une couche d'entrée de D neurones, une couche cachée de H neurones et une couche de sortie de M neurones. Nous fixons M = 2 pour la commodité de la visualisation plane (cf. figure 3.4).

Apprentissage par minimisation de la fonction stress de Sammon

Le principe d'apprentissage du réseau consiste d'abord à prendre les observations par couples (X_u, X_v), u=1,...,N-1, v=u+1,...,N, à calculer leur distance dans l'espace des observations et à les présenter à l'entrée du réseau. On présente X_u, puis X_v. Ensuite la distance entre les projections Y_u et Y_v correspondant respectivement aux observations X_u, X_v est évaluée. Enfin les poids du réseau sont adaptés de telle sorte que la distance entre ces deux observations soit la plus proche possible de la distance entre leurs projections.

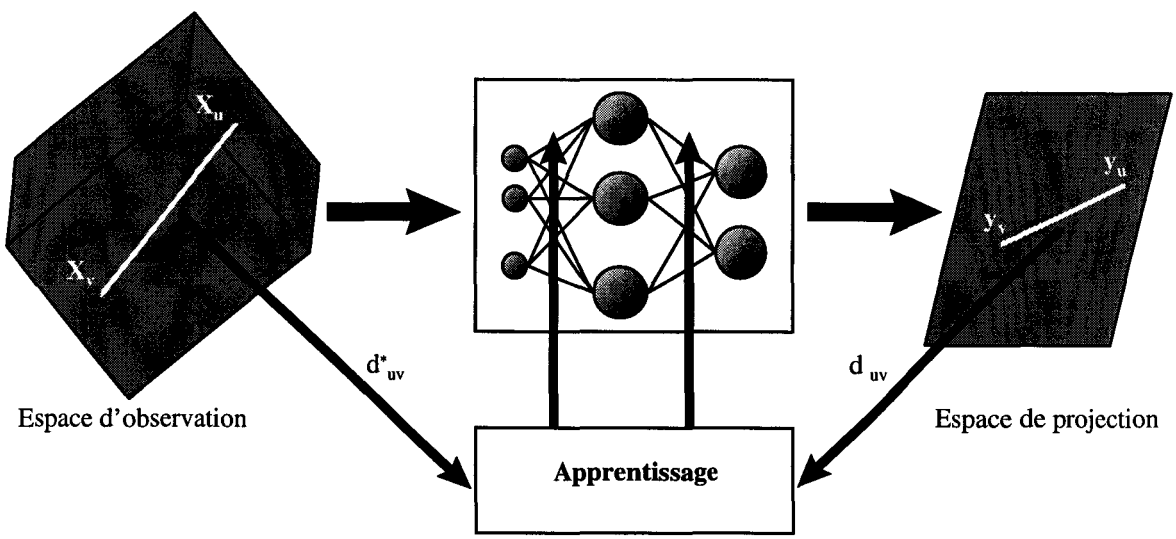


Figure 3.4: Perceptron multicouche minimisant la fonction stress de Sammon.

Soit $(X_u(t), X_v(t))$ le couple d'observations présentées à l'entrée du réseau à l'instant t.

Pour l'observation $X_u(t)$, les sorties des neurones de la couche de sortie et de la couche cachée sont données par les équations (23) et (24):

$$y_{um} = s(\sum_{h=1}^{H} w_{mh} z_{uh}) \ , m = 1, ..., M, \tag{23}$$

$$z_{uh} = s(\sum_{d=1}^{D} w_{hd} x_{ud}) \ , h = 1, ..., H. \tag{24}$$

Rappelons que M est le nombre de neurones en sortie du réseau, qui est ici égal à 2 et que H est le nombre de neurones sur la couche cachée.

On obtient, pour le vecteur X_v, des équations identiques à (23) et (24) en remplaçant l'indice u par l'indice v.

L'apprentissage du réseau consiste à minimiser la fonction stress de Sammon E, définie par:

$$E = \frac{1}{\sum_{u=1}^{N-1} \sum_{v=u+1}^{N} d^*_{uv}} \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \frac{(d^*_{uv} - d_{uv})^2}{d^*_{uv}} \tag{25}$$

où d^*_{uv} et d_{uv} sont les distances Euclidiennes entre les couples d'observations (X_u, X_v) et leurs projections respectives (Y_u, Y_v):

$$d^*_{uv} = \sqrt{\sum_{d=1}^{D} (x_{ud} - x_{vd})^2} \quad ; \quad d_{uv} = \sqrt{\sum_{m=1}^{M} (y_{um} - y_{vm})^2} \tag{26}$$

En posant:

$$E_{uv} = \frac{1}{\sum_{u=1}^{N-1} \sum_{v=u+1}^{N} d^*_{uv}} \frac{(d^*_{uv} - d_{uv})^2}{d^*_{uv}}, \tag{27}$$

la fonction stress de Sammon devient:

$$E = \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} E_{uv} \ . \tag{28}$$

L'actualisation des poids est effectuée par rétropropagation du gradient des fonctions E_{uv}:

$$w_{mh}(t+1) = w_{mh}(t) - \eta(t) \frac{\partial E_{uv}}{\partial w_{mh}} \tag{29}$$

$$w_{hd}(t+1) = w_{hd}(t) - \eta(t)\frac{\partial E_{uv}}{\partial w_{hd}} \qquad (30)$$

Le calcul des gradients dans les équations (29) et (30) est complexe et lourd à mettre en oeuvre. Nous proposons une forme plus simple pour le critère d'apprentissage définie par la minimisation de l'erreur entre les distances des couples d'observations et leurs projections.

3.6 Apprentissage par le critère du maximum de vraisemblance

Soit δ^* la variable aléatoire dont les réalisations sont les distances d^*_{uv}, $1 \le u < v \le N$ entre les couples d'observations (X_u, X_v) dans l'espace des observations et δ la variable aléatoire dont les réalisations sont les distances d_{uv}, $1 \le u < v \le N$, entre les couples de projections (Y_u, Y_v) sur le plan de projection.

Si on suppose que la variable δ suit une loi de probabilité conditionnelle Gaussienne de moyenne δ^* et de variance σ^2 de la forme:

$$p(\delta/\delta^*) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\frac{(\delta-\delta^*)^2}{\sigma^2}} \quad \text{où} \qquad (31)$$

σ^2 est estimé par $\dfrac{1}{\dfrac{N(N-1)}{2}} \sum_{1\le u<v\le N} (d_{uv} - d^*_{uv})^2$,

on définit la vraisemblance conditionnelle l'ensemble des réalisations de δ sachant δ^* par:

$$L = \prod_{1\le u<v\le N} p(\delta = d_{uv}/\delta^* = d^*_{uv}). \qquad (32)$$

Au lieu de maximiser la vraisemblance, il est généralement plus intéressant de minimiser l'opposé du logarithme de la vraisemblance. Ce sont deux procédures équivalentes puisque le logarithme est une fonction monotone croissante.

L'opposé du logarithme de la vraisemblance s'écrit:

$$-\log(L) = \sum_{1\le u<v\le N} p(\delta = d_{uv}/\delta^* = d^*_{uv}) \qquad (33)$$

En introduisant l'équation (31) dans (34) on obtient:

$$-\log(L) = \sum_{1 \le u < v \le N} \frac{1}{2} \frac{\left(d_{uv} - d^*_{uv}\right)^2}{\sigma^2} + \frac{N(N-1)}{2} \log\left(\sqrt{2\pi}\sigma\right) \qquad (34)$$

Remarquons que le second terme à droite de l'égalité (34) est proportionnel à la variance σ^2. Si on suppose que σ^2 est constante, la minimisation de l'équation (34), revient à minimiser le premier terme dont l'expression:

$$E = \frac{1}{2} \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \left(d^*_{uv} - d_{uv}\right)^2, \qquad (35)$$

n'est autre que l'erreur au carré entre les distances des couples d'observations et celles des projections correspondantes.

Dans la suite de ce paragraphe, nous développerons les calculs de propagation avant ainsi que ceux de la sensibilité de l'erreur par rapport aux poids du réseau que nous avons proposée.

Pour simplifier, nous considérons un réseau constitué de D entrées, une couche cachée composée de H neurones et une couche de sortie comprenant M neurones.

Soient X_u et X_v deux observations présentées à l'entrée du réseau à l'instant t.

Les sorties y_{uh} des neurones de la couche cachée sont calculées de la façon suivante:

$$y_{uh} = s\left(\sum_{d=1}^{D} w_{hd} y_{ud}\right), \; h = 1, 2, ..., H. \qquad (36)$$

De la même manière, les sorties des neurones de la couche de sortie sont données par:

$$y_{um} = s\left(\sum_{h=1}^{H} w_{mh} y_{uh}\right), \; m = 1, 2, ..., M, \qquad (37)$$

avec $s(\alpha) = \frac{1}{1 + e^{-\alpha}}$.

L'apprentissage du réseau reprend le même principe que celui que Mao a proposé mais le critère que nous minimisons est celui que nous avons défini dans l'équation (35). L'actualisation des poids du réseau est effectuée par rétropropagation du gradient de cette fonction erreur.

Posons $E_{uv} = \left(d^*_{uv} - d_{uv}\right)^2$, (38)

avec $d^*_{uv} = \sqrt{\sum_{d=1}^{D}(x_{ud} - x_{vd})^2}$ et $d_{uv} = \sqrt{\sum_{m=1}^{M}(y_{um} - y_{vm})^2}$.

La règle de mise à jour des poids reliant la couche cachée à la couche de sortie est la suivante:

$$w_{mh}(t+1) = w_{mh}(t) - \eta(t)\frac{\partial E_{uv}}{\partial w_{mh}}, \text{ m = 1, 2, ..., M et h = 1, 2, ..., H.} \quad (39)$$

Calculons la sensibilité de l'erreur E_{uv} par rapport aux poids de la couche de sortie.

$$\frac{\partial E_{uv}}{\partial w_{mh}} = \frac{\partial E_{uv}}{\partial d_{uv}} \frac{\partial d_{uv}}{\partial (y_{um} - y_{vm})} \frac{\partial (y_{um} - y_{vm})}{\partial w_{mh}}. \quad (40)$$

Or

$$\frac{\partial E_{uv}}{\partial d_{uv}} = -2(d^*_{uv} - d_{uv}),$$

$$\frac{\partial d_{uv}}{\partial (y_{um} - y_{vm})} = \frac{y_{um} - y_{vm}}{d_{uv}},$$

et

$$\frac{\partial (y_{um} - y_{vm})}{\partial w_{mh}} = y_{uh} y_{um}(1 - y_{um}) - y_{vh} y_{vm}(1 - y_{vm}).$$

L'équation (40) s'écrit alors:

$$\frac{\partial E_{uv}}{\partial w_{mh}} = -2(d^*_{uv} - d_{uv})\frac{y_{um} - y_{vm}}{d_{uv}}\left(y_{uh} y_{um}(1 - y_{um}) - y_{vh} y_{vm}(1 - y_{vm})\right). \quad (41)$$

Posons:

- $\delta_m(u,v) = -2(d^*_{uv} - d_{uv})\frac{y_{um} - y_{vm}}{d_{uv}}$,
- $\Delta_{mh}(u) = \delta_m(u,v) y_{um}(1 - y_{um})$,
- $\Delta_{mh}(v) = \delta_m(u,v) y_{vm}(1 - y_{vm})$,

Finalement, l'équation (41) s'écrit:

$$\frac{\partial E_{uv}}{\partial w_{mh}} = \Delta_{mh}(u) y_{uh} - \Delta_{mh}(v) y_{vh}. \quad (42)$$

La règle de mise à jour des poids reliant les neurones de la couche cachée à ceux de la couche de sortie devient:

$$w_{mh}(t+1) = w_{mh}(t) - \eta(t)\left(\Delta_{mh}(u)y_{uh} - \Delta_{mh}(v)y_{vh}\right). \tag{43}$$

De façon similaire, la sensibilité de l'erreur E_{uv}, par rapport aux poids reliant les entrées aux neurones de la couche cachée est donnée par:

$$\frac{\partial E_{uv}}{\partial w_{hd}} = \Delta_{hd}(u)y_d(u) - \Delta_{hd}(v)y_d(v), \tag{44}$$

avec:

- $\Delta_{hd}(u) = \delta_h(u)\left(1 - y_h(u)\right)$ et $\delta_h(u) = \sum_{m=1}^{M} \Delta_{mh}(u)w_{mh}$,
- $\Delta_{hd}(v) = \delta_h(v)\left(1 - y_h(v)\right)$ et $\delta_h(v) = \sum_{m=1}^{M} \Delta_{mh}(v)w_{mh}$.

Les poids reliant les entrées du réseau aux neurones de la couche cachée sont mis à jour selon la règle suivante:

$$w_{hd}(t+1) = w_{hd}(t) - \eta(t)\left(\Delta_{hd}(u)y_{hd} - \Delta_{hd}(v)y_{hd}\right). \tag{45}$$

Résumons l'algorithme d'apprentissage du Perceptron multicouche minimisant l'erreur entre la distance du couple d'observations (X_u, X_v) et la distance de leurs projections respectives (Y_u, Y_v).

Algorithme d'apprentissage

1. Initialiser les poids du réseau aléatoirement à des valeurs voisines de 0. Fixer le coefficient d'apprentissage $\eta(t)$ et le nombre total d'itérations T_f,
2. Sélectionner de façon aléatoire une paire d'observations (X_u, X_v) de la base d'apprentissage et la présenter au réseau,
3. Calculer les équations (36) et (37) du réseau,
4. Actualiser les poids du réseau en commençant par la couche de sortie (équations 43 et 45),
5. Incrémenter t (t = t+1) et aller en 2 jusqu'à ce que $t = T_f$.

Ajustement du nombre de neurones sur la couche cachée

Dans le paragraphe précédent, nous avons présenté un Perceptron multicouche réalisant la projection non linéaire de données multidimensionnelles d'un espace d'observations de dimension D vers un espace de projection de dimension M (M<D). Le réseau proposé comprend D entrées, une couche cachée composée de H neurones et une couche de sortie comprenant M neurones. Si le nombre d'entrées et celui des neurones de sortie sont parfaitement connus et correspondent aux dimensions des espaces d'observation et de projection, le nombre de neurones H sur la couche cachée est quant à lui inconnu et délicat à choisir. En effet, un nombre important de neurones sur la couche cachée risque de détériorer les performances du réseau au niveau de la généralisation (c'est le problème "over-fitting") alors qu'un nombre réduit de neurones sur la couche cachée risque de fournir un modèle de représentation trop simple. Nous proposons d'optimiser le nombre des neurones sur la couche cachée grâce à la théorie de l'information. Pour ce faire, nous utilisons les deux critères de Akaike: FPE (Final Prediction Error) et AIC (Akaike Information Criterion) [Aka 72] [Aka 74]. Ces critères sont définis par:

$$FPE(H) = \bar{E}\frac{1+\frac{[H(D+1) + M(H+1)]}{ND}}{1-\frac{[H(D+1) + M(H+1)]}{ND}} \qquad (46)$$

et

$$AIC(H) = Ln(\overline{E}) + \frac{[H(D+1) + M(H+1)]}{ND}, \qquad (47)$$

où:

- $\overline{E}$ est l'erreur moyenne entre les distances des couples d'observations et celles des projections correspondantes obtenue à partir de l'équation (35), qu'on peut aussi écrire sous la forme:

$$\overline{E} = \frac{1}{2ND} \sum_{u=1}^{N-1} \sum_{v=u+1}^{N} \left(d^*_{uv} - d_{uv}\right)^2 \qquad (48)$$

- les termes entre crochets au numérateur de l'équation (47) comptabilisent le nombre des connexions du réseau alors que le dénominateur indique le nombre total des éléments de la base d'apprentissage NxD.

L'optimisation de l'architecture du réseau est effectuée en faisant varier le nombre H des neurones cachés et en calculant les critères FPE et AIC, (équations (46) (47)), pour chaque nombre. Le nombre optimum est celui qui correspond aux extremums des deux critères [Bet 97a], [Bet 97b].

Dans le chapitre suivant nous présentons, sur deux exemples, les résultats obtenus par ces deux critères.

3.7 Conclusion

Nous avons présenté différents réseaux de neurones offrants de nouvelles méthodes de visualisation plane pour la classification interactive. Les règles d'apprentissage de ces réseaux sont soit d'inspiration biologique comme le concept d'apprentissage compétitif et d'autoorganisation de Kohonen, soit le résultat de l'optimisation d'un critère mathématique comme le Perceptron multicouche.

Les performances des réseaux pour la projection non linéaire dépendent fortement du nombre des neurones sur les couches cachées. Ce nombre a été optimisé grâce aux critères informationnels de Akaike.

Dans le chapitre suivant nous montrons sur deux exemples l'influence du choix de l'architecture des réseaux de neurones et des coefficients d'apprentissage sur leurs performances en projection plane.

Chapitre 4

Résultats expérimentaux

Chapitre 4 Résultats expérimentaux

4.1. Introduction

La classification interactive consiste, une fois terminé l'apprentissage du réseau de neurones, à afficher les éléments de l'ensemble de projection sur un système de coordonnées. La visualisation et l'analyse de l'ensemble des projections donnent une idée sur la structure de l'ensemble des observations dans l'espace d'origine.

Cependant, la visualisation plane n'a de sens que si les projections sont une représentation suffisamment fidèle des observations. La fidélité est mesurée par un critère d'erreur qui est associé à la méthode de projection. Or le choix du nombre de neurones dans l'architecture des réseaux et les paramètres de réglage intervenant dans leurs algorithmes d'apprentissage ont une influence directe sur la qualité des résultats obtenus.

Dans ce chapitre nous nous intéressons plus particulièrement au réseau autoassociateur et au réseau à apprentissage par le maximum de vraisemblance (cf. paragraphe 3.4 et 3.6, chapitre 3). Nous montrons, sur les deux exemples présentés dans le chapitre 1, l'influence du choix du nombre des neurones des couches cachées sur la qualité de la projection plane. Les deux critères informationnels utilisés pour l'optimisation de l'architecture de ces réseaux sont:

$$FPE(H) = \bar{E}\frac{1+\frac{N_p}{ND}}{1-\frac{N_p}{ND}},$$

$$AIC(H) = Ln(\bar{E}) + \frac{2N_p}{ND}.$$

où N_p indique le nombre des paramètres libres du réseau autrement dit le nombre des poids à estimer (cf. paragraphe 3.4 et 3.6, chapitre 3).

Notons que pour le premier réseau constitué de 3 couches cachées, le nombre N_p est donné par:

$$N_p = [H(D+1)] + [2(H+1)] + [H(2+1)] + [D(H+1)].$$

alors que pour le second, qui ne comporte qu'une seule couche cachée, ce nombre est égal à:

$$N_p = [H(D+1)] + [M(H+1)].$$

Nous verrons finalement l'influence du coefficient d'apprentissage η et du momentum α sur la vitesse de convergence des algorithmes d'apprentissages.

4.2 Exemple de quatre classes Gaussiennes

Rappelons pour cet exemple que l'on dispose d'un mélange de quatre classes dont chacune est composée de 100 observations dans un espace à trois dimensions. Il s'agit de classes Gaussiennes dont les vecteurs moyennes sont positionnés sur 4 des 8 sommets d'un cube de côtés 1 (cf. figure 4.1). La classe 1 est centrée sur l'origine alors que les trois autres sont centrées sur les sommets du cube dont les côtés sont situés sur les axes x_1, x_2 et x_3 respectivement. Les caractéristiques statistiques des classes, sont résumées dans le tableau 4.1.

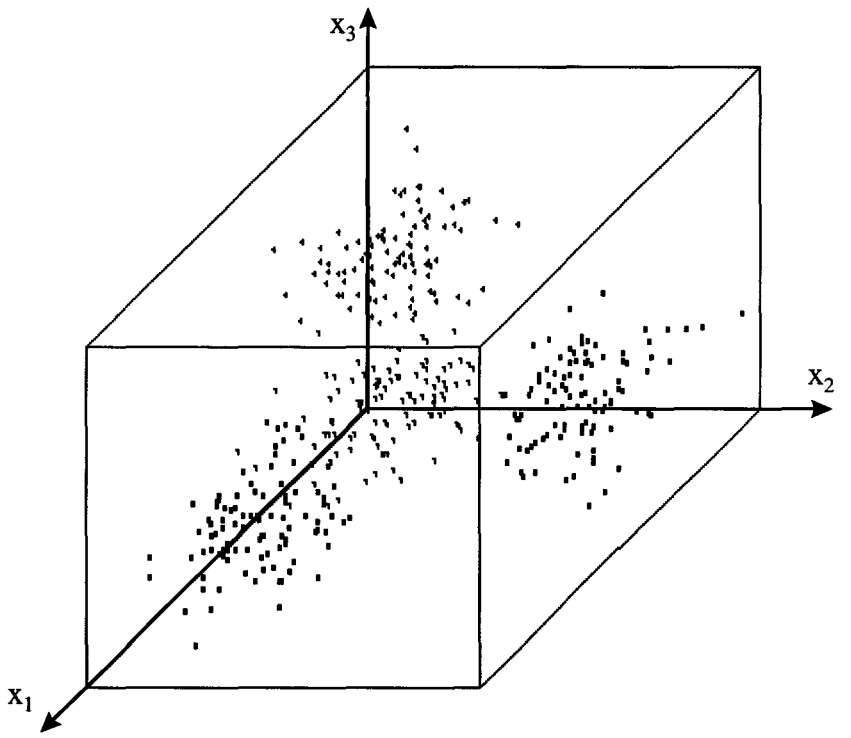


Figure 4.1 Représentation des 4 classes Gaussiennes dans l'espace

	Classe 1	Classe 2	Classe 3	Classe 4
Taille	100	100	100	100
Vecteurs moyennes	0 0 0	1 0 0	0 1 0	0 0 1
Matrices de variance-covariance	0,05 0 0 0 0,05 0 0 0 0,05	0,05 0 0 0 0,05 0 0 0 0,05	0,05 0 0 0 0,05 0 0 0 0,05	0,05 0 0 0 0,05 0 0 0 0,05

Tableau 4.1. Caractéristiques statistiques de 4 classes Gaussiennes.

4.2.1 Projection par le réseau autoassociateur

Pour le réseau autoassociateur (cf. paragraphe 3.4, chapitre 3), nous avons fait varier le nombre de neurones H de la première et troisième couches cachées de 2 à 15 en calculant à chaque fois l'erreur E ainsi que les critères informationnels FPE et AIC. Les paramètres d'apprentissage η et α ont été fixés respectivement à 0.1 et 0.8.

Les résultats obtenus sont consignés dans le tableau 4.2. Notons que les valeurs obtenues par le critère FPE sont positives alors que les valeurs du critère AIC sont négatives. Sur la dernière colonne nous avons représenté l'erreur quadratique moyenne normalisée EN, définie par:

$$EN = \frac{\sum_{n=1}^{N}\sum_{d=1}^{D}(x_{nd} - \hat{x}_{nd})^2}{\sum_{n=1}^{N}\sum_{d=1}^{D}(x_{nd} - \overline{x}_{nd})^2}$$

Le principale avantage de cette formulation de l'erreur est qu'elle est sans unité.

Les valeurs extremums des critères informationnels FPE et AIC ont été obtenues pour H=5.

Nombre de neurones sur la première et troisième couche cachée (H)	FPE	AIC	EN
2	0.166	-1.8	0.158
3	0.0835	-2.48	0.078
4	**0.078**	-2.54	0.072
5	**0.078**	**-2.56**	**0.070**
6	0.0785	**-2.56**	0.069
10	0.0895	-2.41	0.073
15	0.0996	-2.31	0.073

Tableau 4.2. Valeurs des critères informationnels obtenus avec le réseau autoassociateur sur les 4 classes Gaussiennes.

La figure 4.2 montre la projection obtenue par le réseau autoassociateur (paragraphe 3.4 du chapitre 3) à la fin de son apprentissage avec la valeur optimum de H égale à 5.

On peut remarquer que globalement le réseau préserve, comme l'algorithme de Sammon (cf. figure 1.5, page 20), la structure des données. En effet, la classe centrée sur l'origine se retrouve bien entourée par les trois classes se situant sur les coins du cube de côté un. La structure locale de deux classes n'a pas été totalement préservée, en effet les nuages représentant les classes 2 et 3 sont plus étendus que les deux autres.

Notons que pour un nombre de neurones égal à deux (H=2) sur la première et la dernière couche cachées (cf. figure 4.3), nous avons obtenu une bonne visualisation des données proche de celle fournie par l'algorithme Sammon alors que l'erreur de reconstruction est relativement importante (EN=0.158). Pour H=10 la structure des classes en projection n'est pas conservée (cf. figure 4.4) alors que l'erreur de reconstruction est faible (EN=0.073). Ces constatations trouvent leur justification dans le fait que l'apprentissage est basé sur un critère qui minimise l'erreur de reconstruction des données présentées à l'entrée du réseau et non pas sur la préservation en projection de la structure des classes. La projection est alors obtenue indirectement en exploitant les sorties des neurones de la couche cachée centrale.

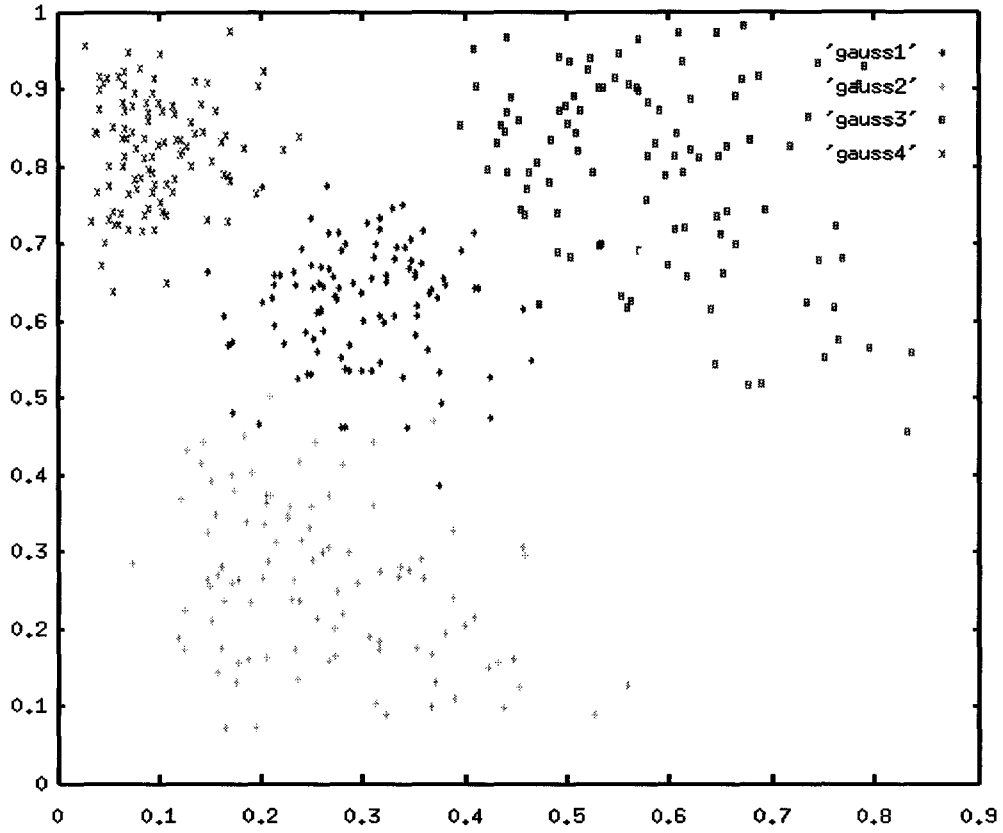


Figure 4.2 Projection plane des classes Gaussiennes par un autoassociateur (H=5, EN=0.07).

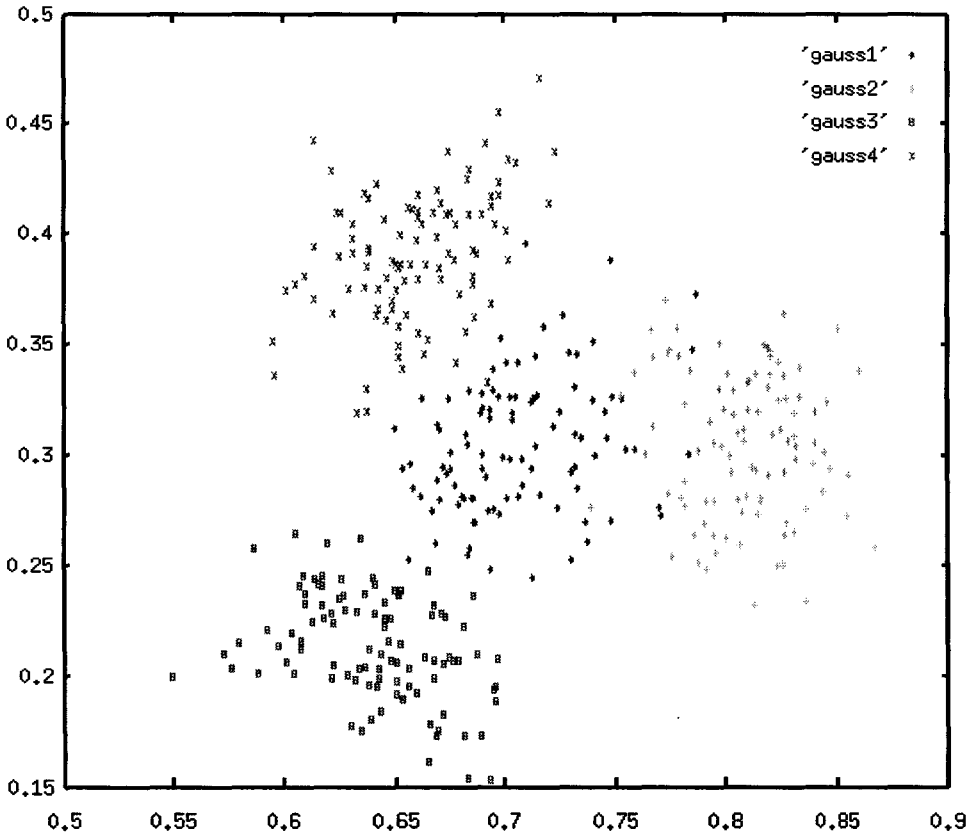


Figure 4.3 Visualisation plane des 4 classes par un réseau autoassociateur (H=2, EN=0.158).

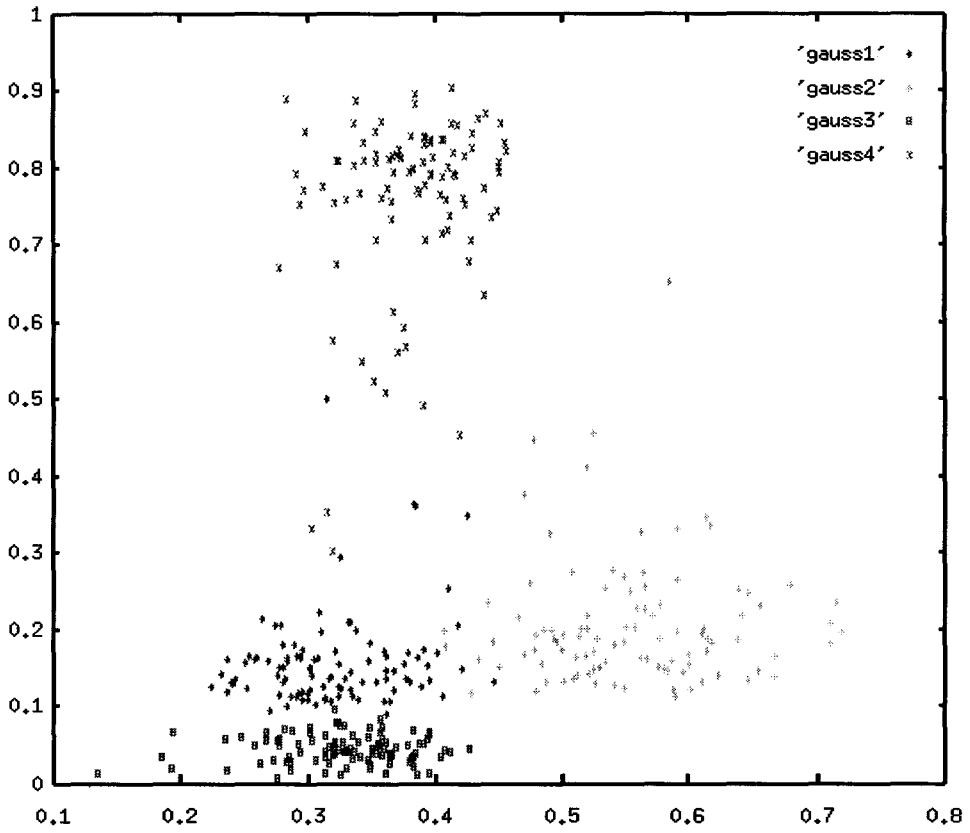


Figure 4.4 Visualisation plane des 4 classes par un autoassociateur (H=10, EN=0.073).

4.2.2 Projection par le réseau à apprentissage par le maximum de vraisemblance

Pour ce réseau (cf. paragraphe 3.6, chapitre 3), nous avons fait varier le nombre de neurones de l'unique couche cachée de 2 à 15 et nous avons calculé les valeurs des critères FPE et AIC, ainsi que celle de l'erreur normalisée EN. Le tableau 4.3 présente les différentes valeurs des critères en fonction du nombre H des neurones cachés. Les valeurs extremums des critères informationnels FPE et AIC ont été obtenues pour H=10.

La figure 4.5 montre le plan de projection obtenu avec le réseau à apprentissage par le maximum de vraisemblance pour H= 10. Nous remarquons que la structure locale des classes est bien préservée en projection. La visualisation plane obtenue en sortie du réseau est comparable au plan de projection de l'algorithme de Sammon: les dispersions des 4 classes sont assez voisines.

Nombre de neurones sur la couche cachée (H)	FPE	AIC	$\overline{E}$
2	0.206	-1.59	0.187
3	0.0972	-2.34	0.085
4	0.10	-2.31	0.084
5	0.113	-2.19	0.091
6	0.120	-2.13	0.093
10	**0.0776**	**-2.57**	**0.051**
15	0.102	-2.31	0.054

Tableau 4.3. Résultats obtenus avec le réseau à apprentissage par le maximum de vraisemblance sur les 4 classes Gaussiennes.

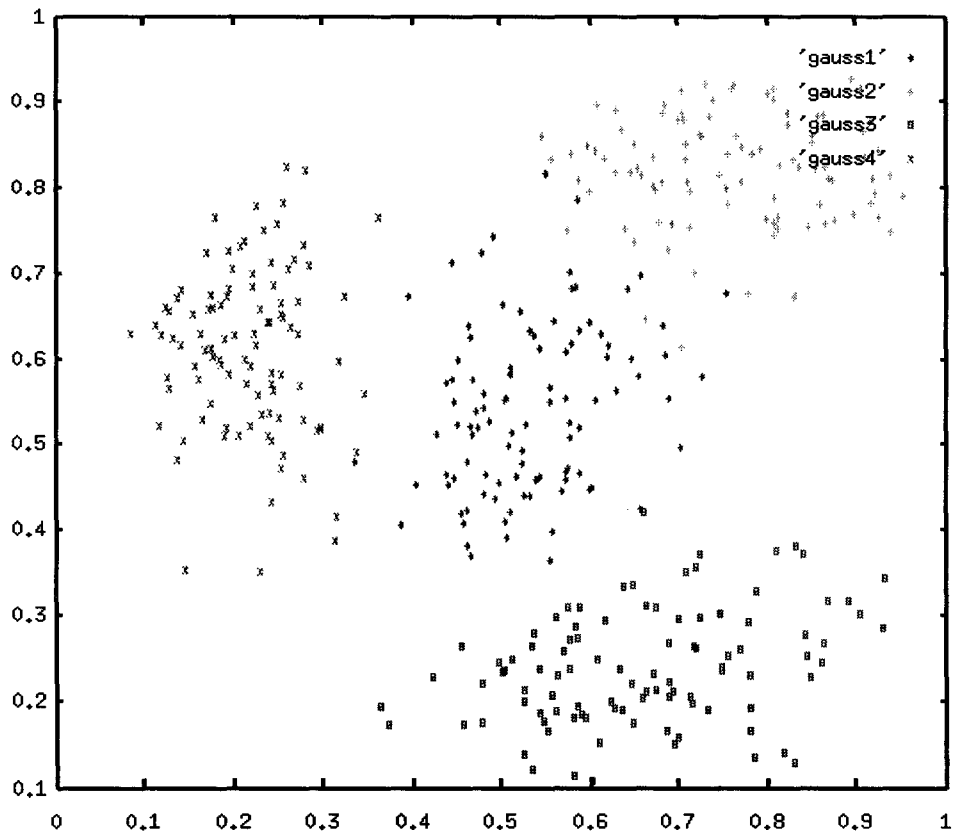


Figure 4.5 Projection plane des 4 classes par un Perceptron multicouches à apprentissage par le maximum de vraisemblance (H= 10, EN= 0.051).

4.3 Exemple des Iris de Fisher

Rappelons qu'il s'agit de 3 variétés de fleurs d'Iris (Setosa, Versicolor et Verginica) caractérisées par les longueur et largeurs des pétales et des sépales (en cm). Pour chaque variété on dispose d'un échantillon de taille 50. La population totale de 150 observations constituée de 3 classes de 50 observations chacune est représentée dans un espace de dimension 4. Le tableau 4.4 montre les caractéristiques statistiques des 3 classes.

	Classe 1 (Setosa)	**Classe 2 (Versicolor)**	**Classe 3 (Verginica)**
Vecteurs moyennes	2,46 14,62 34,28 50,1	13,26 43,22 27,64 59,36	20,06 55,52 29,74 65,88
Matrices de variance-covariance	0,124 0,099 0,016 0,010 0,099 0,144 0,012 0,009 0,016 0,012 0,030 0,006 0,010 0,009 0,006 0,011	0,266 0,085 0,183 0,056 0,085 0,098 0,083 0,041 0,183 0,083 0,221 0,073 0,056 0,041 0,073 0,039	0,404 0,094 0,303 0,049 0,094 0,104 0,071 0,048 0,303 0,071 0,304 0,049 0,049 0,048 0,049 0,075

Tableau 4.4. Caractéristiques statistiques des Iris de Fisher.

4.3.1 Projection par le réseau autoassociateur

De même que pour les données Gaussiennes, nous avons étudié l'influence du nombre de neurones des couches cachées du réseau autoassociateur sur la qualité de la visualisation plane. Nous avons fait varier le nombre H de 2 à 15 en calculant à chaque fois l'erreur E de reconstruction des données présentées à l'entrée du réseau ainsi que les critères informationnels FPE et AIC. Le tableau 4.5 résume les résultats obtenus.

Les valeurs extrêmums des critères FPE et AIC sont obtenues pour H=5. La figure 4.6 montre les projections planes obtenues pour cette valeur de H. Comme pour l'algorithme de Sammon, nous distinguons une classe bien isolée des deux autres qui se chevauchent fortement.

Nombre de neurones sur la première et troisième couche cachée (H)	FPE	AIC	EN
2	0.0605	-2.81	0.053782
3	0.0572	-2.86	0.048706
4	0.0570	-2.88	0.045398
5	**0.0510**	**-3.00**	**0.038698**
6	0.0555	-2.89	0.041021
10	0.0620	-2.78	0.037853
15			

Tableau 4.5 Valeurs des critères informationnels obtenus avec le réseau autoassociateur sur les Iris.

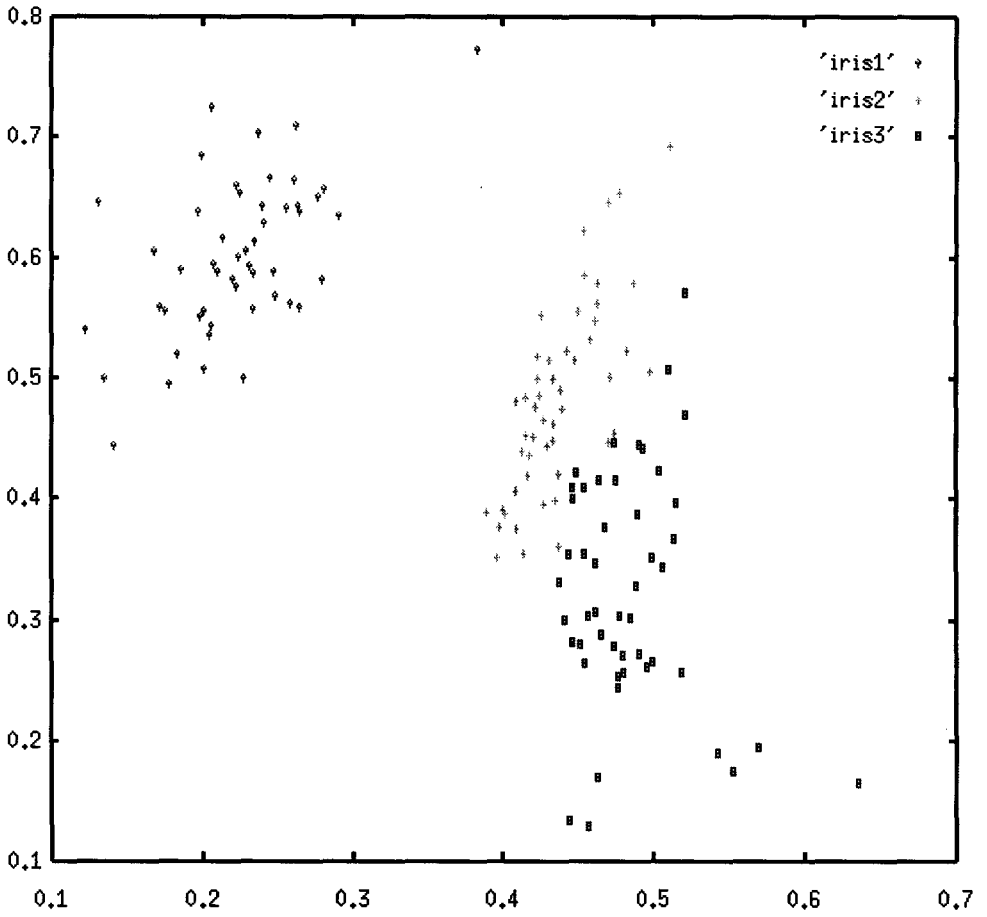


Figure 4.6 Projection plane des Iris par un réseau autoassociateur (H=5, EN=0.038).

4.3.2 Projection par le réseau à apprentissage par le maximum de vraisemblance

Nous avons fait varier le nombre de neurones de l'unique couche cachée de 2 à 15 et nous avons calculé les valeurs des critères FPE et AIC, ainsi que celle de l'erreur normalisée EN. Le tableau 4.5 présente les différentes valeurs des critères en fonction du nombre H des neurones cachés. Les valeurs extremums des critères informationnels FPE et AIC ont été obtenues pour H=6.

Nombre de neurones sur la couche cachée (H)	FPE	AIC	EN
2	0.164	-1.81	0.155
3	0.065	-2.73	0.06
4	0.044	-3.12	0.04
5	0.033	-3.42	0.029
6	**0.032**	**-3.43**	**0.028**
10	0.038	-3.27	0.030
15	0.053	-2.94	0.037

Tableau 4.5. Valeurs des critères informationnels obtenus avec le réseau à apprentissage par le maximum de vraisemblance sur les Iris.

La figure 4.7 montre les projections planes obtenues avec le réseau à apprentissage par le maximum de vraisemblance pour H= 6. Nous distinguons une classe bien séparée des deux autres (classe2 et classe3) qui se chevauchent de façon significative. Toutefois les projections planes obtenue restent meilleures que celles de la figure 4.6.

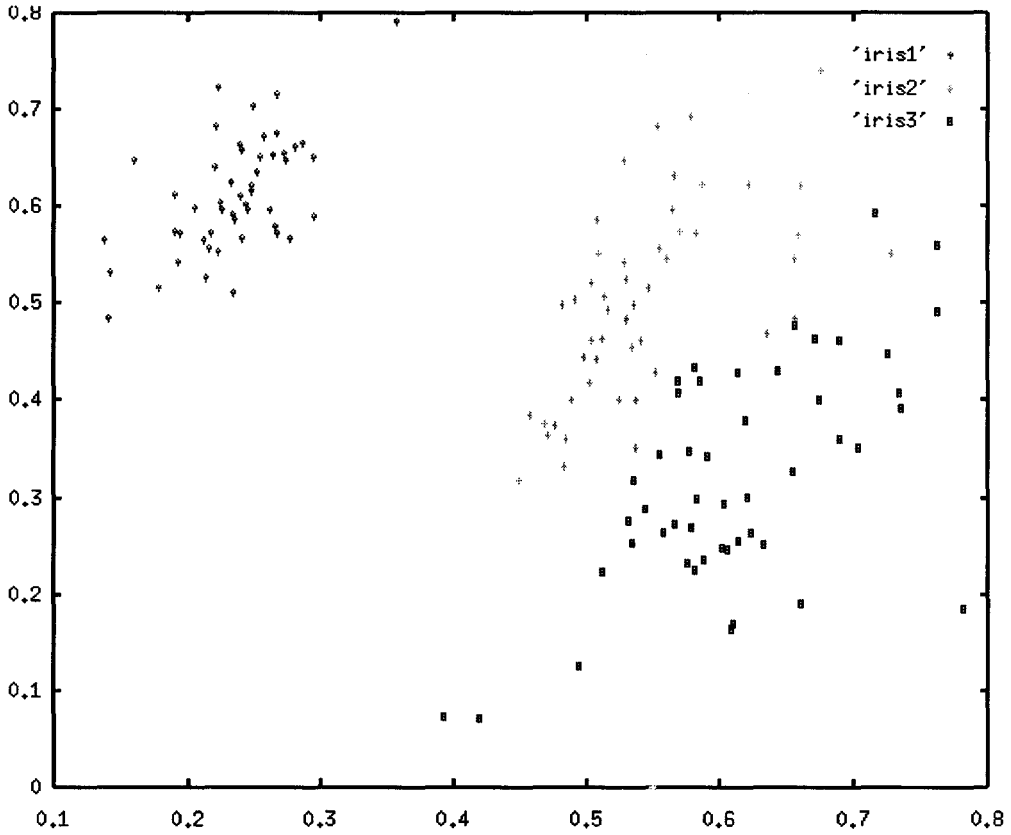


Figure 4.7 Projection plane des Iris par le réseau à apprentissage par le maximum de vraisemblance (H=5, EN=0.038).

4.4 Influence du coefficient d'apprentissage sur le critère d'erreur

Dans ce paragraphe nous étudions l'influence du coefficient d'apprentissage η et du momentum α pour une architecture donnée du réseau à apprentissage par le maximum de vraisemblance appliqué sur les Iris de Fisher. Nous avons fixé le coefficient d'apprentissage η à 0.1 et nous avons fait varier le momentum α de 0.1 à 0.8.

Nous avons remarqué que lorsque α est voisin de zéro (cf. figure 4.8), la courbe de l'erreur E en fonction des itérations présente des oscillations au début de l'apprentissage et sa convergence est assez lente.

Lorsque le momentum est voisin de un, il produit un effet de stabilisation de la courbe de l'erreur qui oscille moins au début de l'apprentissage (cf. figure 4.9) et décroît vite vers un minimum plus bas que celui obtenu sans ce terme (cf. figure 4.8).

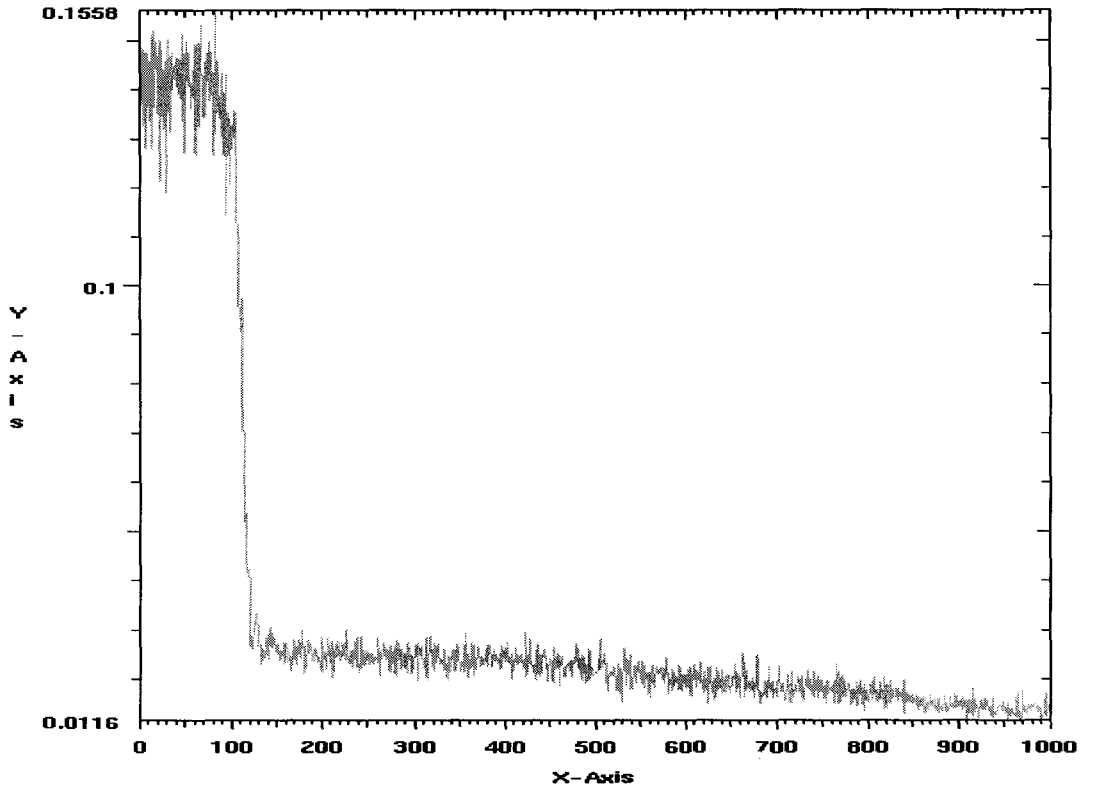


Figure 4.8 Evolution du critère d'erreur au cours des itérations pour η= 0.1 et α=0.2.

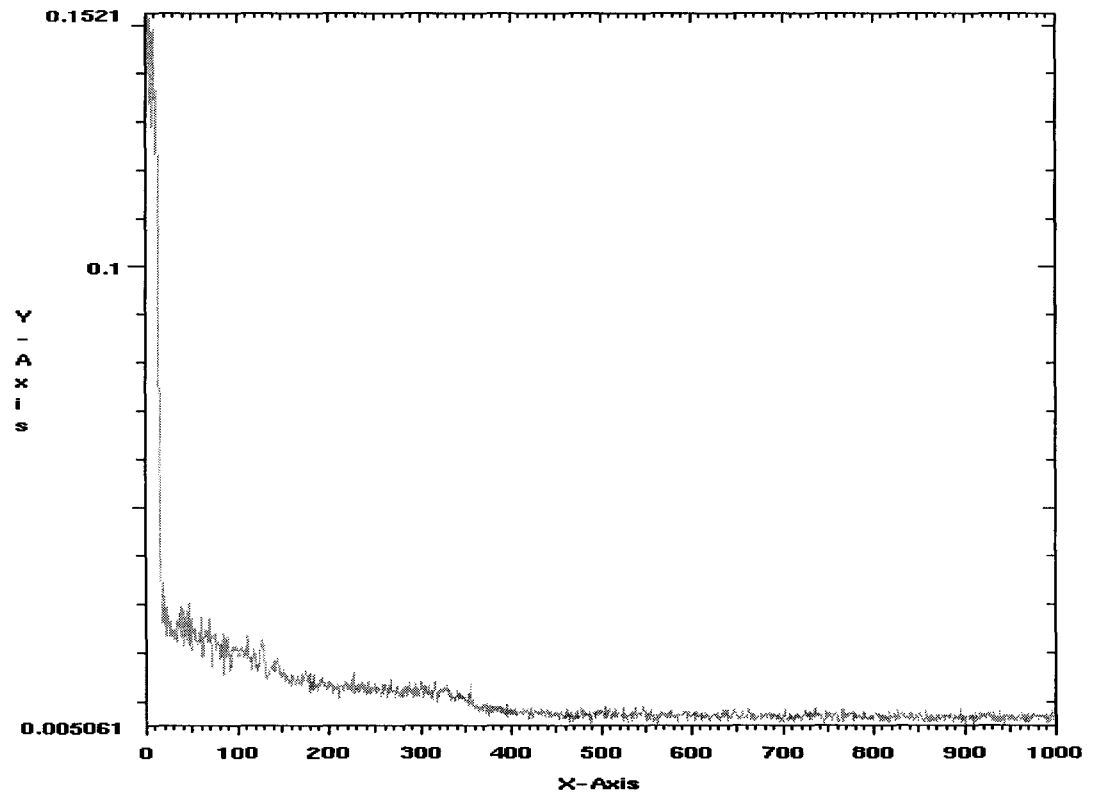


Figure 4.9 Evolution du critère d'erreur au cours des itérations pour η= 0.1 et α= 0.8.

4.5 Discussions et conclusion

L'étude expérimentale présentée a mis en évidence l'avantage du réseau à apprentissage par le maximum de vraisemblance pour la visualisation plane de données multidimensionnelles. En effet, le critère d'erreur employé dans son processus d'apprentissage est basé sur la minimisation des distances entre les points dans l'espace des observations et leurs projections respectives sur le plan. Le réseau autoassociateur quant à lui utilise un critère basé l'erreur entre les observations et leurs estimations. La qualité de la projection résulte indirectement de la qualité de la reconstruction en utilisant l'argument suivant: si les estimations sont assez proches des observations c'est que le réseau a effectué une "bonne projection plane". Nous avons montré que cette affirmation n'est pas toujours vérifiée. Finalement, nous avons constaté qu'un choix judicieux des paramètres d'apprentissage accélère la convergence de l'algorithme d'apprentissage et peut éviter de "tomber" dans un minimum local.

Dans les deux chapitres suivants, nous présentons deux applications industrielles utilisant les réseaux de neurones pour le diagnostic. La première concerne la détection de défauts sur des bouteilles en verre par analyse d'images et réseaux de neurones. La seconde application est relative à la surveillance d'une éolienne par les méthodes de traitement du signal et les réseaux de neurones artificiels.

Chapitre 5

Détection de glaçures sur les bagues des bouteilles en verre

Chapitre 5 Détection de glaçures sur les bagues des bouteilles en verre

5.1 Introduction

Le succès de l'emploi du verre comme matériau de conditionnement est principalement lié à ses caractéristiques techniques et commerciales. Le verre constitue une barrière absolue aux micro-organismes: il n'existe pas de danger d'altération des produits alimentaires, il est étanche aux gaz et aux liquides et il possède une bonne résistance mécanique et thermique. Le verre permet aussi de réaliser des récipients de formes et de couleurs esthétiques qui permettent la mise en valeur du produit. Enfin, c'est une matière première bon marché et pouvant être recyclée à 100%.

Dans l'industrie verrière, les fabricants doivent faire face à deux objectifs antagonistes: baisser le coût global de fabrication tout en améliorant la qualité de la production. Pour tendre vers une production de qualité parfaite, il a été nécessaire pour tous les grands groupes verrier de surveiller et de contrôler le plus efficacement possible la qualité de leur production, notamment celle qui concerne les bouteilles et les flacons.

En effet, tout le long de la chaîne de production, les bouteilles en verre sont affectées par différentes agressions thermiques et mécaniques conduisant à l'apparition de différents types de défauts. Parmi les défauts critiques importants à détecter est celui des glaçures sur la bague de la bouteille. Une glaçure est une fissure pouvant traverser partiellement ou totalement le verre. Ce type de défaut est critique dans la mesure où il peut endommager la bouteille avant où après l'encapsulation conduisant à l'endommagement des installations et à la contamination des produits emballés [Fir 97].

Le principe du dispositif multicapteur de détection de défaut actuellement utilisé consiste à faire tourner chaque bouteille sur elle même tout en éclairant son goulot à l'aide d'un faisceau lumineux et à capter grâce à un capteur optoélectronique les rayons émis. Si le flux lumineux est supérieur à un seuil, la bouteille est rejetée. Bien que ce dispositif donne de bons résultats,

il est très sensible aux réflexions parasites dues au filetage du goulot et nécessite un temps de réglage important par des techniciens spécialisés pour ajuster les positions des sources d'éclairages et des capteurs à chaque changement de type de bouteilles.

L'objectif de notre étude est d'évaluer la faisabilité du remplacement des capteurs optoélectroniques du dispositif par un système de vision en ligne associé à un réseau de neurones. Ce système doit être très rapide pour suivre la haute cadence de production, robuste par rapport aux variations de l'environnement de production et assez fiable pour assurer une grande qualité de la production. La conception d'un tel système nécessite trois étapes: l'acquisition de l'image à l'aide d'une caméra matricielle CCD, le traitement et l'extraction d'attributs à l'aide des techniques d'analyse de l'image et enfin la décision à l'aide de réseaux de neurones artificiels.

Ce chapitre est organisé de la façon suivante: le paragraphe 2 décrit le principe du système de vision, le troisième est consacré au traitement de l'image et l'extraction des attributs, enfin le paragraphe 4 présente les résultats expérimentaux sur des séquences d'images réelles de bouteilles de bières qui ont été réalisées à l'aide de différents types de réseaux de neurones.

5.2 Description de la machine du système de vision

Le nouveau système d'inspection de défauts remplace le dispositif optoélectronique par une caméra matricielle CCD associée à un miroir cylindrique. Les bouteilles arrivent l'une derrière l'autre dans la machine d'inspection visuelle où elles subissent une rotation complète en face du système d'éclairage composé d'un flux lumineux dirigé vers le goulot. Puisque les glaçures peuvent apparaître dans n'importe quelle position du goulot et peuvent avoir n'importe quelle orientation, un miroir de forme cylindrique est utilisé afin de diriger tous les faisceaux lumineux vers la caméra qui réalise l'acquisition d'une séquence de 16 images, 128x128 à 256 niveaux de gris, correspondant à un tour complet de la bouteille. La vitesse de rotation de la bouteille est réglée de telle sorte que la tâche, due à une glaçure même petite, soit présente dans au moins trois images successives. La figure 5.1 montre le principe du système d'acquisition d'image utilisé et les figures 5.2 (a) et (b) montrent respectivement deux images de la séquence sans et avec présence d'une glaçure. Notons que les réflexions dues aux filetages du goulot de la bouteille aussi bien qu'à la présence de glaçures sont les zones de l'image les plus claires sur un fond sombre.

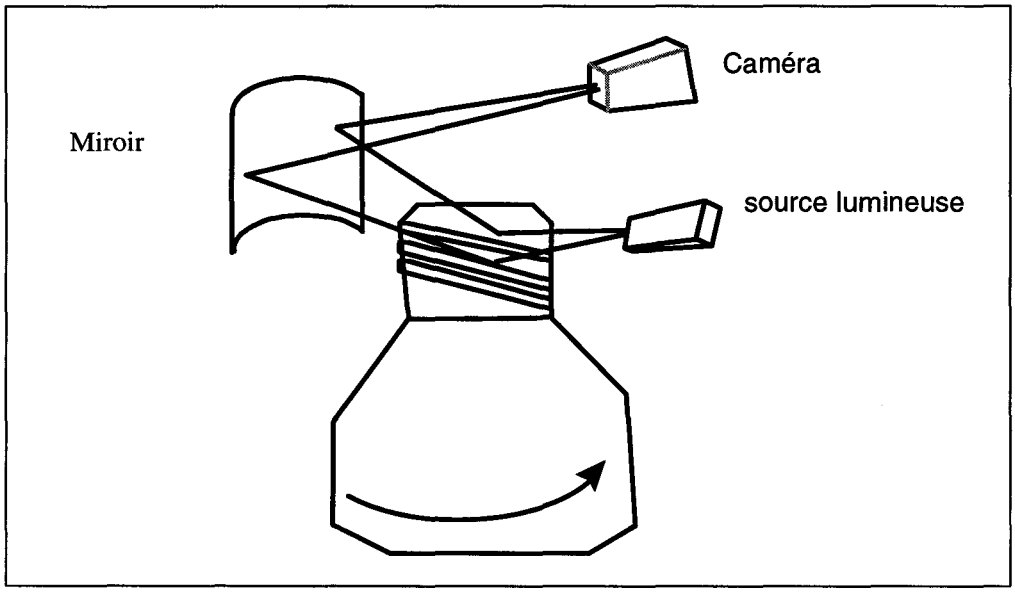


Figure 5.1 Principe du système d'acquisition des images.

(a) (b)

Figure 5.2. Réflexions dues aux réflexions du filetage (a) et celles dues à une glaçure (b) sur le goulot d'une bouteille.

5.3 Traitement et extraction des attributs

L'image captée par la caméra doit être traitée en vue de localiser les régions claires sur un fond sombre qui sont dues à des réflexions du filetage pour les bouteilles sans défaut et à des réflexions du filetage et des glaçures pour des bouteilles présentant un défaut. Il est important

de rappeler que le système de vision doit être robuste quant aux variations de la couleur du verre, aux variations de l'intensité de la source d'éclairage et aux variations possibles des conditions de l'éclairage ambiant. Pour ce faire, une binarisation de l'image avec un seuil adaptatif est nécessaire à l'extraction des tâches, dues aux réflexions et éventuellement aux glaçures, à partir des différentes images. L'analyse des histogrammes des niveaux de gris des différentes images montre que leur allure générale reste similaire (figure 5.3). Ils sont constitués d'un seul mode correspondant au fond sombre de l'image. En effet, la contribution des pixels clairs due aux réflexions est difficilement perceptible sur l'histogramme car la taille des impacts lumineux est très petite [Fir 96].

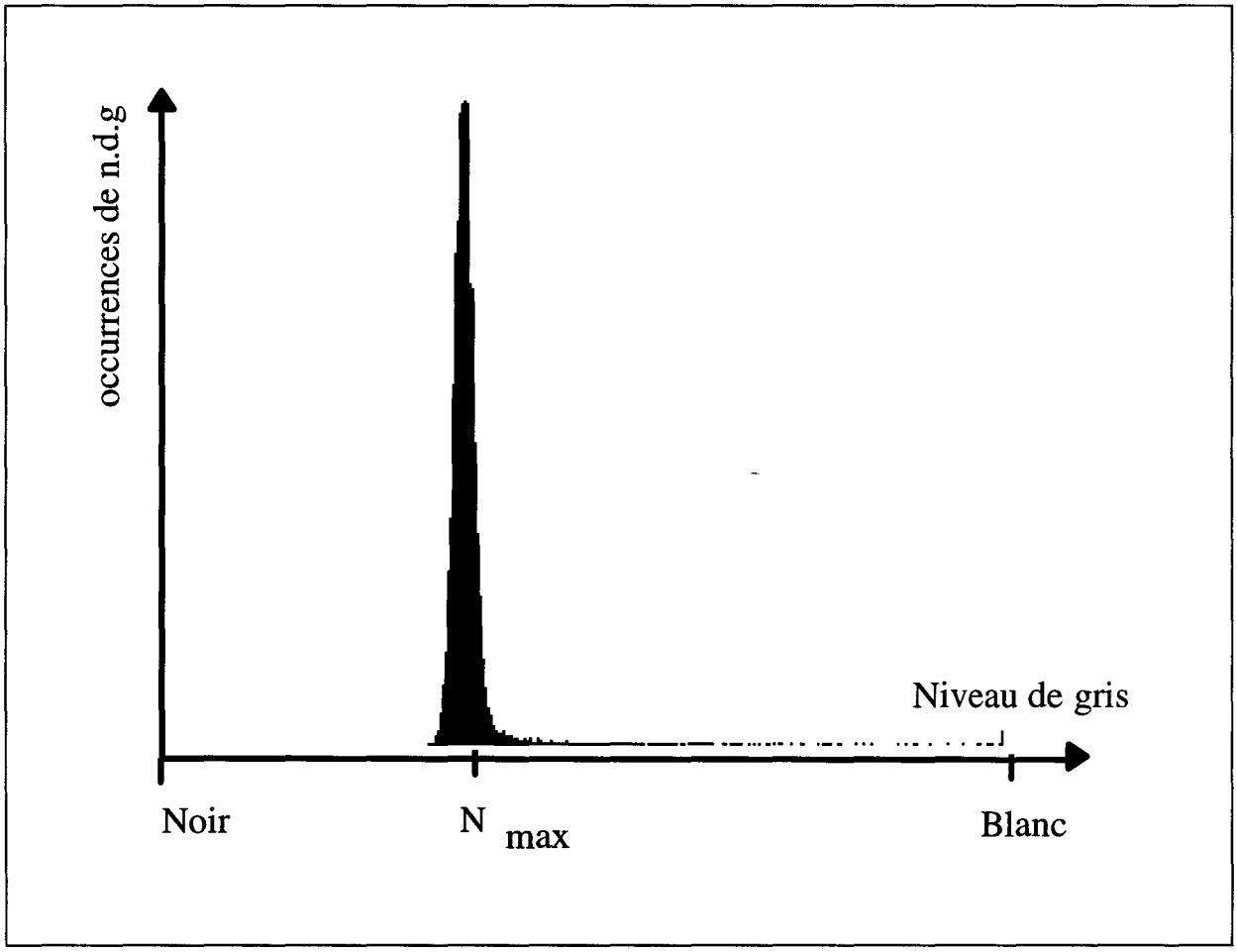


Figure 5.3 Histogramme typique d'une image de la séquence

Si on suppose que l'histogramme est Gaussien et que le nombre des pixels clairs est très petit par rapport au nombre de pixels représentant le fond, l'histogramme peut alors être considéré

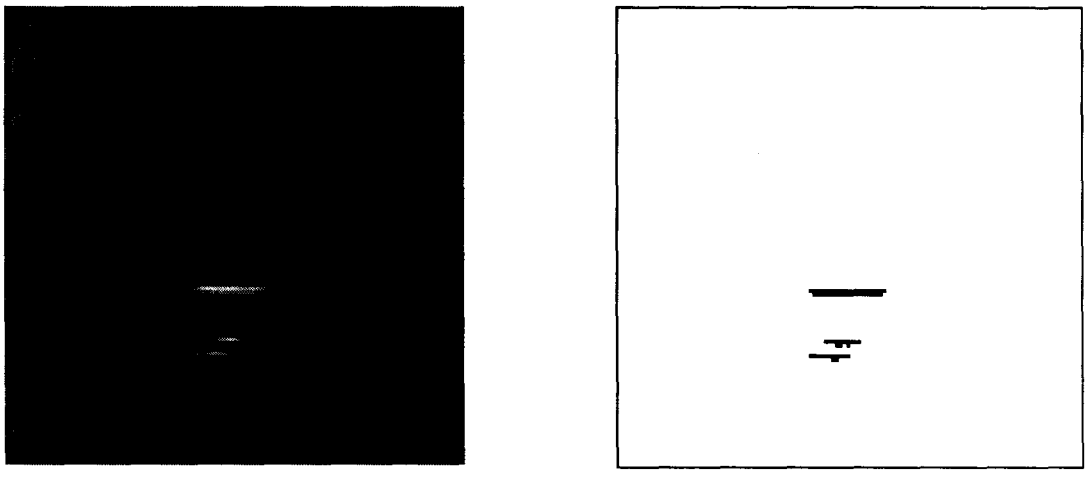
comme ayant une seule composante Gaussienne. Dans ces conditions, nous avons choisi d'ajuster le seuil de binarisation S_0 à la valeur:

$$S_0 = N_{max} + \alpha\sigma \qquad (1)$$

- N_{max} est le niveau de gris du mode de l'histogramme,
- σ est la déviation standard estimée à partir de la distribution du niveau de gris qui représente grossièrement le fond,
- α est un paramètre ajusté par l'utilisateur.

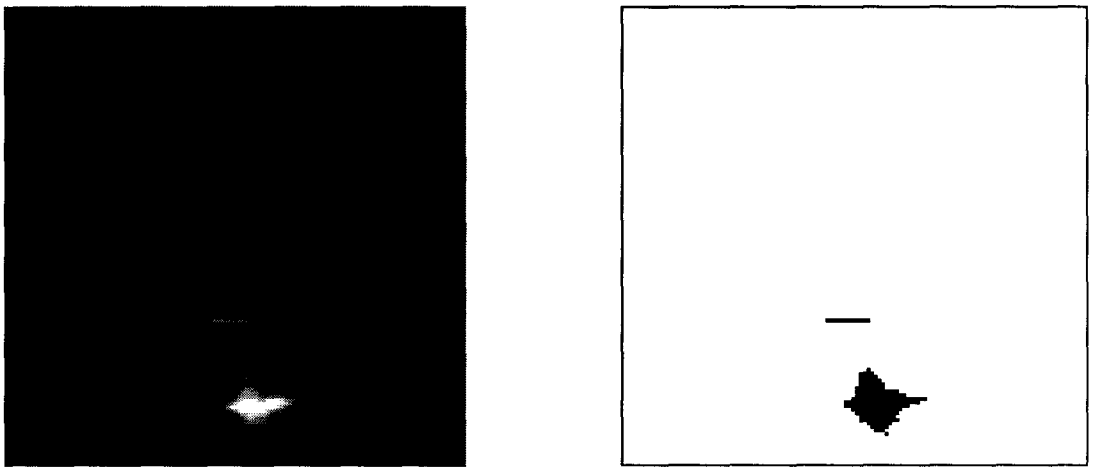
Sous les hypothèses normales, 68 % approximativement des niveaux de gris sont situés dans l'intervalle $[N_{max}-\sigma, N_{max}+\sigma]$. Dans le cas d'une image 128x128 pixels, ceci veut dire que les niveaux de gris de 11140 pixels appartiennent à cet intervalle. En supposant que le plus grand pic de l'histogramme correspond à la moyenne de la distribution, il est facile d'estimer σ en comptant les pixels. Cette approximation est justifiée par le petit nombre de pixels dus aux réflexions.

La figure 5.4 représente une image, avant et après une binarisation adaptative, avec uniquement des réflexions dues au filetage du goulot de la bouteille correspondante. La figure 5.5 correspond à une image représentant les réflexions d'une glaçure et quelques réflexions dues à la surface du goulot.



(a) (b)

Figure 5.4 Une image avec des réflexions dues à la surface du goulot avant (a) et après binarisation (en vidéo inverse) (b).



(a) (b)

Figure 5.5 Une image avec des réflexions dues à la surface du goulot et à une glaçure avant (a) et après binarisation (en vidéo inverse) (b).

Il est important de noter qu'à ce stade du traitement de l'image, il est impossible de distinguer les réflexions produites par la surface du goulot de la bouteille de celles provenant des glaçures. Comme les réflexions dues à la surface du goulot de la bouteille et celles résultant de la présence d'une glaçure contribuent à la création de petites tâches claires appelées régions d'intérêt, nous définissons la zone d'intérêt comme étant le plus petit rectangle contenant toutes les zones d'intérêt.

La figure 5.6 présente une image avec une région d'intérêt contenant des zones d'intérêt. Il est à noter que la taille de la zone d'intérêt est pratiquement constante pour une bouteille sans glaçures et est généralement plus grande lorsqu'une bouteille présente des glaçures.

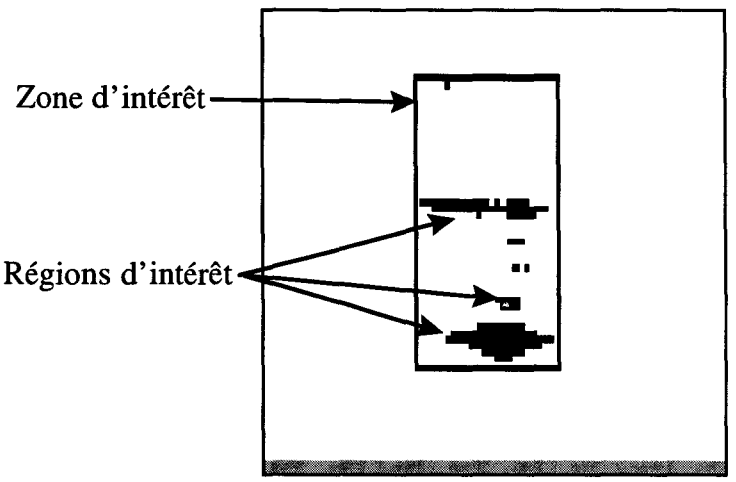


Figure 5.6 Une image présentant une zone d'intérêt contenant des régions d'intérêt.

Nous allons alors extraire quelques attributs des images en niveaux de gris ainsi que de celles binarisées, afin de distinguer les images provenant des bouteilles sans glaçures de celles présentant une glaçure. Les attributs qui ont été extraits d'une image sont de nature morphométriques (8 attributs) et photométriques (6 attributs):

Attributs morphométriques

Notation	Explication	Formule
• x_1	Nombre de pixels $P_i \in P$: P_i est de coordonnées (x_i, y_i) et P est la zone d'intérêt	N
• x_2	Position moyenne en x des pixels $P_i \in P$	$\overline{x} = \frac{1}{N}\sum_{P_i \in P}(x_i - x_{min})$
• x_3	Position moyenne en y des $P_i \in P$	$\overline{y} = \frac{1}{N}\sum_{P_i \in P}(y_i - y_{min})$
• x_4	Longueur de la zone d'intérêt	$amp_x = x_{max} - x_{min} + 1,$
• x_5	Largeur de la zone d'intérêt	$amp_y = y_{max} - y_{min} + 1,$
• x_6	Variance des coordonnées en x des régions d'intérêt	$v_x = \sum_{p_i \in P}\left[(x_i - x_{min}) - \overline{x}\right]$
• x_7	Variance des coordonnées en y des régions d'intérêt	$v_y = \sum_{p_i \in P}\left[(y_i - y_{min}) - \overline{y}\right]$
• x_8	Rapport entre le nombre N de pixels $P_i \in P$ et la surface de la zone d'intérêt	$R = \frac{N}{amp_x amp_y}$

Ces seuls attributs morphologiques ne suffisent pas à caractériser les images avec et sans glaçure. En effet, on peut remarquer, sur la majorité des images que le niveau de gris est très important: les rayons lumineux réfléchis par une glaçure laissent une trace très intense alors que la trace laissée par la paroi extérieure de la bague est peu lumineuse. Nous allons donc utiliser cette information. Tous les attributs dits photométriques que nous allons extraire sont

calculés uniquement sur les pixels dont le niveau de gris est supérieur au seuil S_0 fixé précédemment.

En retenant tous ces attributs, qui semblent a priori aussi pertinents les uns que les autres, chaque image se retrouve caractérisée par un vecteur $X=(x_1, x_2,...,x_d, ..., x_{14})^T$ de dimension 14 où les x_d, d=1,...,14, prennent leurs valeurs dans R^+. Cependant, tous les attributs ne sont pas discriminants. Dans le paragraphe suivant nous cherchons les attributs discriminants par la méthode d'analyse discriminante pas à pas.

Attributs photométriques

Notation	Explication	Formule
• x_9	Niveau de gris moyen des $P_i \in P$	$\overline{g} = \frac{1}{N} \sum_{p_i \in p} g_i$
• x_{10}	Variance en niveau de gris des $P_i \in P$	$v_g = \sum_{P_i \in P} (g_i - \overline{g})^2$
• x_{11}	Amplitude des variations du niveau de gris des $P_i \in P$	$amp_g = g_{max} - g_{min}$
• x_{12}	Valeur moyenne en x, pondérée par le niveau de gris des $P_i \in P$	$\overline{x}_p = \frac{\sum_{p_i \in p} g_i x_i}{\sum_{p_i \in p} g_i} - x_{min}$
• x_{13}	Valeur moyenne en y, pondérée par le niveau de gris, des $P_i \in P$	$\overline{y}_p = \frac{\sum_{p_i \in p} g_i y_i}{\sum_{p_i \in p} g_i} - y_{min}$
• x_{14}	Distance entre la position moyenne et la position moyenne pondérée par le niveau de gris des pixels $P_i \in P$	$D = \sqrt{(\overline{x} - \overline{x}_p)^2 + (\overline{y} - \overline{y}_p)^2}$

5.4 Analyse discriminante pas à pas

Pour sélectionner les attributs les plus discriminants, nous employons une solution itérative ordonnant les attributs selon leur pouvoir discriminant [Dud 73].

Soient C_1 et C_2 les classes d'observations obtenues à partir des images des bouteilles sans et avec défauts.

La qualité d'un sous-ensemble d'attributs est fonction de sa capacité à discriminer les classes dans le sous-espace défini par ces attributs. Cette discrimination est d'autant plus aisée que les classes sont plus éloignées les unes des autres et que chacune est plus compacte. Le critère de sélection sera donc basé sur la compacité et la séparabilité des classes.

- Critère de compacité de la classe C_1 et C_2:

$$\Sigma_i = \frac{1}{N_i} \sum_{X_n \in C_i} (X_n - \mu_i)(X_n - \mu_i)^T;\ i = 1,\ 2 \tag{2}$$

où μ_i est le vecteur moyenne de la classe C_i:

$$\mu_i = \frac{1}{N_i} \sum_{X_n \in C_i} X_n;\ i = 1,\ 2\,. \tag{3}$$

La matrice de dispersion intra-classe (within) totale est :

$$\Sigma_W = \frac{1}{N_1 + N_2}\left(\Sigma_1 + \Sigma_2\right). \tag{4}$$

- Critère de séparabilité des classes : matrice de dispersion interclasse (between) :

$$\Sigma_B = \frac{1}{N_1 + N_2}\left(N_1(\mu_1 - \mu)(\mu_1 - \mu)^T + N_2(\mu_2 - \mu)(\mu_2 - \mu)^T\right) \tag{5}$$

où μ est le vecteur moyenne de l'ensemble des observations.

Les classes sont d'autant plus séparables que leur dispersion ou inertie intra-classe est faible et que la dispersion interclasse est grande. D'où, le critère de sélection des attributs:

$$J = \mathrm{Trace}(\Sigma_W^{-1}.\Sigma_B)\,. \tag{6}$$

L'analyse discriminante linéaire affecte une observation à la classe la plus proche au sens de la distance de Mahalanobis. L'algorithme d'analyse discriminante pas à pas introduit les attributs dans l'ordre décroissant de leur pouvoir discriminant. L'algorithme cherche séquentiellement

le sous-ensemble de attributs assurant la meilleure discrimination. A chaque pas, un attribut supplémentaire est ajouté au sous-ensemble de attributs retenus au pas précédent.

Il existe habituellement quatre critères d'introduction de attributs: la maximisation du critère de la trace J, le critère de Wilks, le pourcentage d'observations bien classées et la maximisation des différences entre les moyennes conditionnelles pour les différentes classes. Le critère de maximisation de la trace permet d'introduire l'attribut qui maximise le rapport entre la dispersion interclasse et la dispersion totale, alors que le critère de Wilks sélectionne l'attribut qui minimise le rapport entre la dispersion intra-classe et la dispersion totale. Les deux critères sont équivalents. L'algorithme de sélection pas à pas est le suivant.

Algorithme de sélection des attributs

- Etape 1. Calculer le critère J pour chacun des 14 attributs et sélectionner l'attribut qui maximise ce critère.
- Etape 2. Ajouter à cet attribut chacun des D-1 attributs restants et sélectionner le couple qui maximise le critère J.
- Etape 3. Ajouter au couple retenu à l'étape 2 chacun des D-2 attributs restants et sélectionner le triplet qui maximise le critère J, etc..
- Etape D-1. Ajouter au sous-ensemble constitué par les D-2 attributs retenus chacun des 2 attributs restant et sélectionner l'un des deux sous-ensembles de D-1 attributs qui maximise le critère.

Les attributs x_6, x_5, x_4 et x_{11} sont les plus discriminants, dans la suite on les appellera x'_1, x'_2, x'_3 et x'_4 respectivement.

5.5 Constitution de la base de données

Afin de construire une base de données complète, nous avons collecté de façon manuelle des bouteilles présentant des formes différentes du goulot et des opacités de degrés différents. Les images choisies proviennent de bouteilles de bière de deux types: le premier type correspond à

des bouteilles de bière ayant une bague avec filet et l'autre correspond à des bouteilles ayant une bague sans filet. Ces bouteilles ont été présentées une à une en face de la machine d'inspection et des séquences d'images ont été réalisées et stockées. Puisqu'une glaçure peut apparaître trois à quatre fois au cours d'une séquence, nous avons étiqueté ces images de façon manuelle aussi.

L'ensemble des observations ainsi obtenu a été divisé en deux ensembles : un ensemble d'apprentissage et un ensemble de test. Parmi les 208 images constituant les observations d'apprentissage, 104 images proviennent de bouteilles bonnes et 94 images proviennent de bouteilles avec glaçures. L'ensemble des observations de test contient 162 images parmi lesquelles 130 images représentes des bouteilles bonnes et 59 sont des images de bouteilles contenant une glaçure (c.f. Tableau 5.1).

	Images sans glaçure	Images avec glaçure	Nombre total des images
Ensemble d'apprentissage	114	94	208
Ensemble de test	103	59	162

Tableau 5.1 Ensemble des observations: ensemble d'apprentissage et ensemble de test

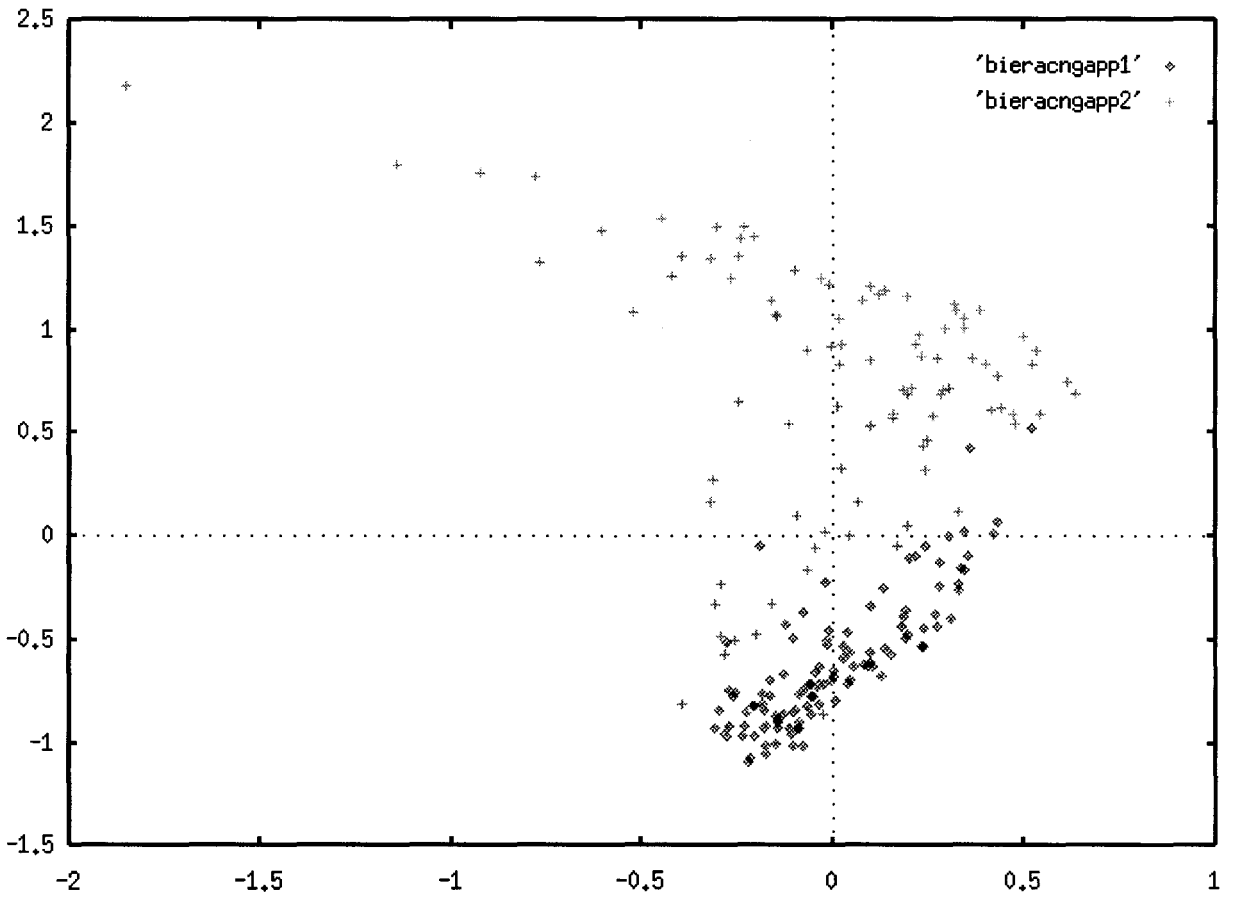
Dans le paragraphe qui suit nous allons projeter les 4 attributs sur le plan afin d'avoir une idée sur la complexité des frontières de décision des points correspondants à des bouteilles avec et sans glaçures.

5.6 Visualisation des observations sur un plan

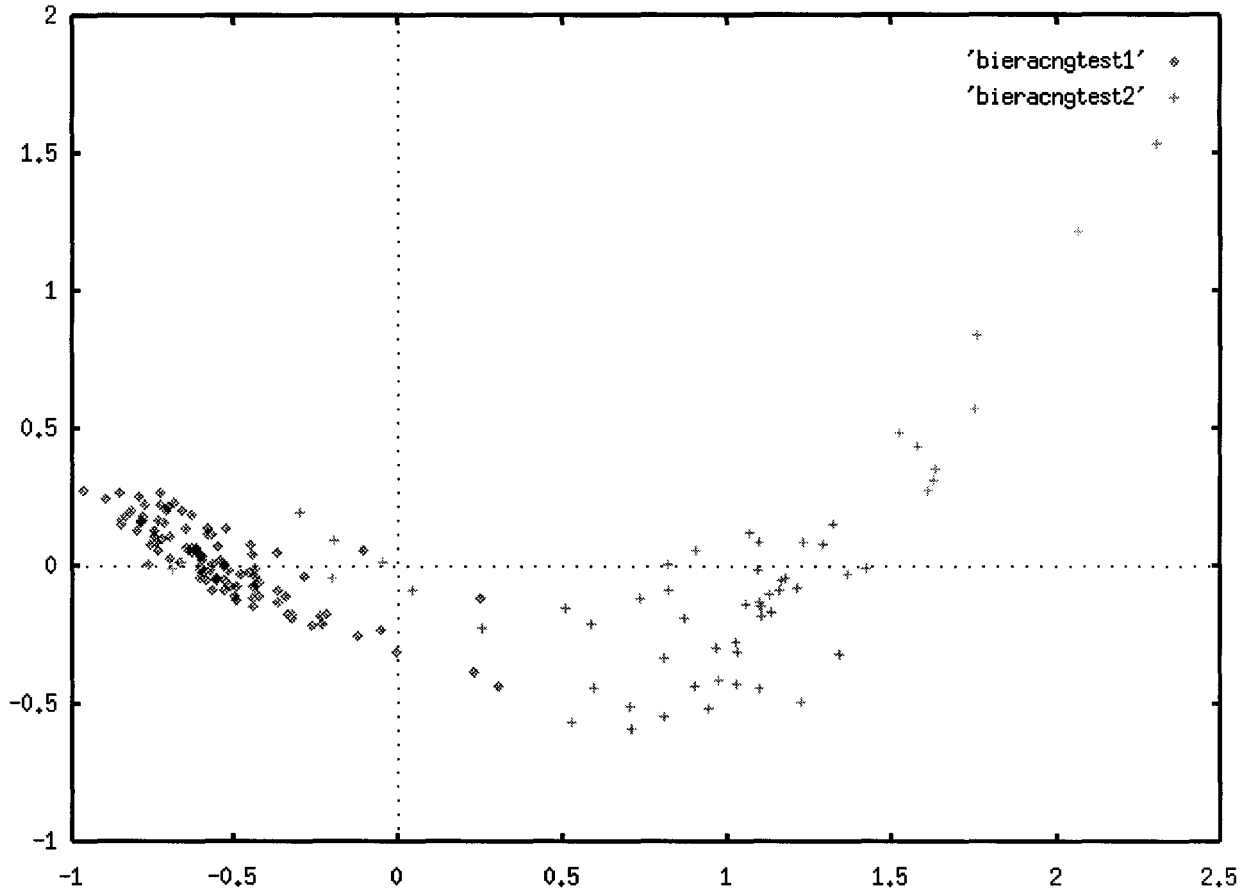
Afin d'avoir une idée sur la structure des observations constituant la base de données des images des différentes bouteilles, nous proposons de comparer deux méthodes de visualisation des observations ; la visualisation plane par analyse en composantes principales et celle obtenue par le Perceptron à apprentissage par le maximum de vraisemblance.

5.6.1. Projection linéaire par analyse en composantes principales

L'ensemble des observations de la base d'apprentissage aussi bien que l'ensemble des observations de la base de test, constituent deux nuages de points dans un espace à 4 dimensions. Pour bien visualiser la structure des données, nous avons procédé à une projection linéaire de la base d'apprentissage et de la base de test en utilisant la méthode d'analyse en composantes principales (ACP) (cf. figures 5.5 (a) et (b)). On peut remarquer que pour les deux bases, l'ensemble des points de la classe des bouteilles sans glaçure est partiellement entouré par les points représentant la classe des bouteilles avec glaçures.



(a)



(b)

Figure 5.5 Visualisation des observations sur le plan principal

(a) ensemble d'apprentissage

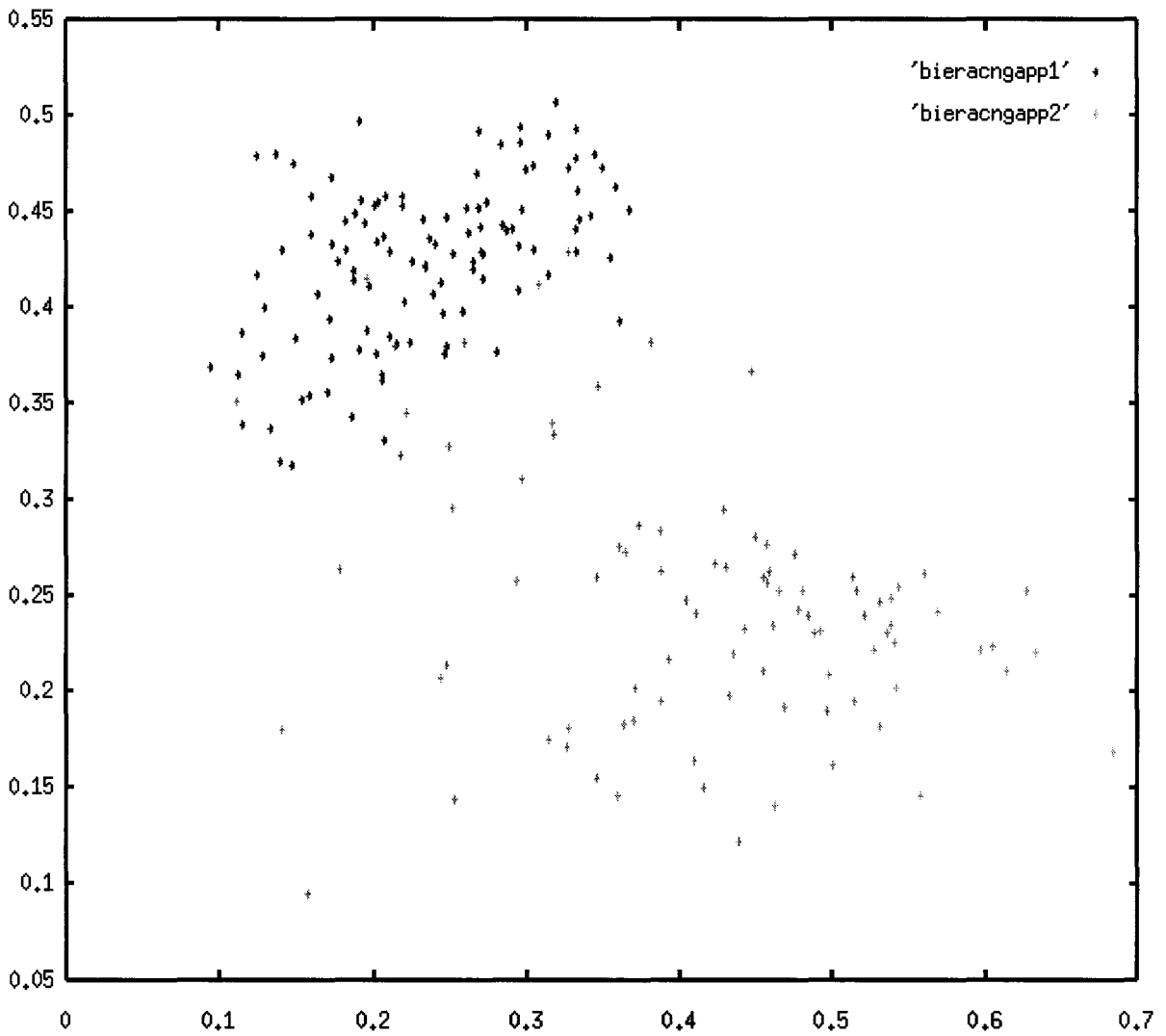
(b) ensemble de test

Le nuage de points représentés par la couleur rouge centré sur (-0.5, 0) est assez compact et correspond à la classe des images sans défaut. Le nuage de points représentés par la couleur verte correspond à la classe des images avec défaut est plus dispersé.

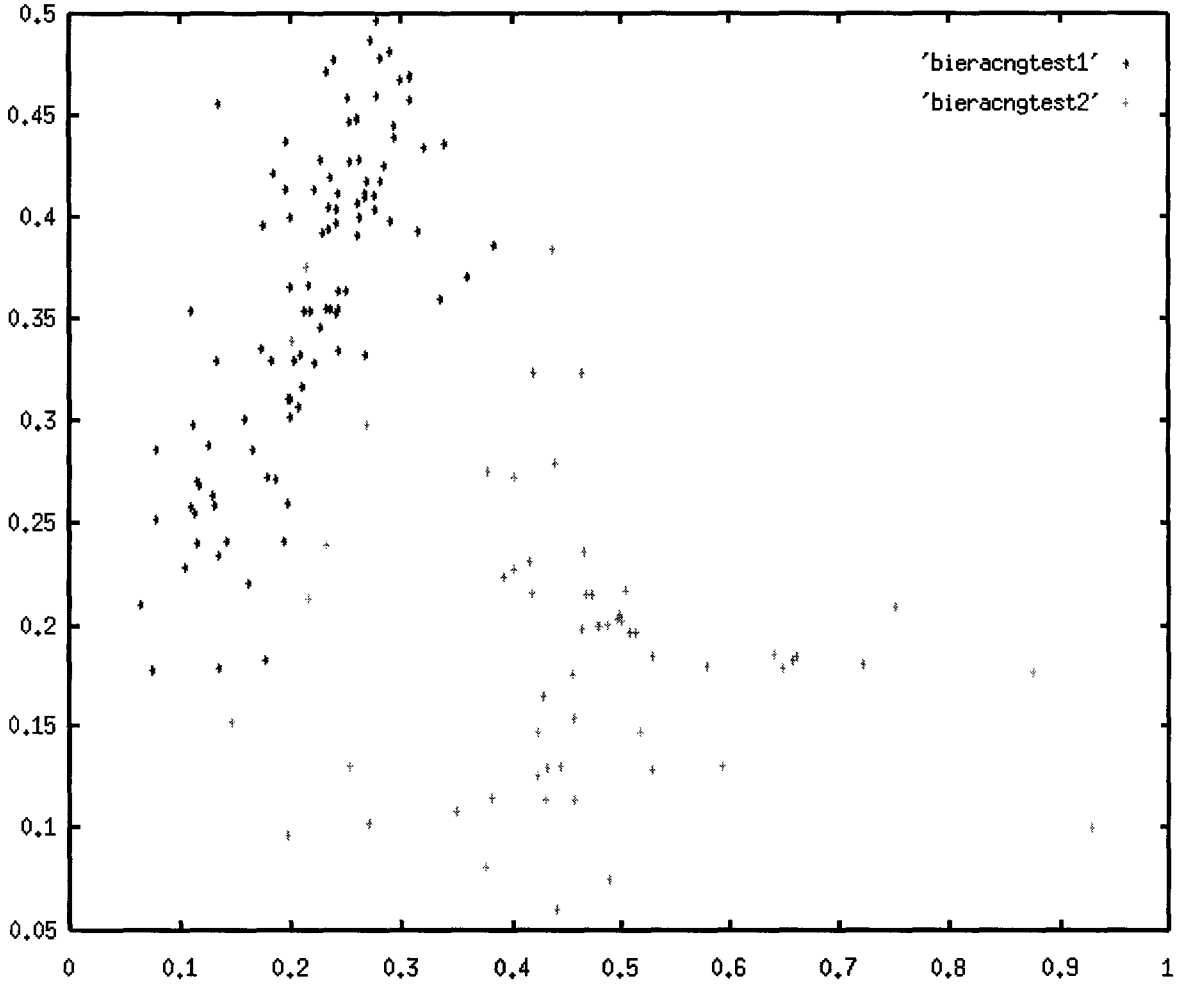
Cependant, le plan principal ne nous renseigne pas sur la structure réelle des données dans l'espace de départ, en particulier sur les distances entre les différents points. Pour cela nous complétons cette visualisation par une projection non linéaire.

5.6.2 Projection non linéaire

L'algorithme de projection non linéaire utilisé est celui du Perceptron à apprentissage par le maximum de vraisemblance qui consiste à minimiser le critère défini dans le paragraphe 3.6 du chapitre 3. Son principe de base est que deux points proches dans l'espace de départ, ici R^4, doivent rester, autant que possible, proches dans le plan de projection.



(a)



(b)

Figure 5.6 Visualisation plane par le réseau à apprentissage par le maximum de vraisemblance

(a) ensemble d'apprentissage

(b) ensemble de test.

Une analyse des figures 5.6 (a) et (b) nous permet de constater que les points représentant des images de bonnes bouteilles sont bien regroupés, ce qui a aussi été constaté dans le plan principal. Certains points représentant des images avec glaçures se trouvent dans le nuage des points représentatifs des images des bonnes bouteilles. Ceci est dû d'une part à la difficulté de l'opération d'étiquetage que nous avons effectuée manuellement et qui nécessite dans certains cas un seuillage pour décider si l'image est bonne ou non, d'autre part, sur certaines images, celles correspondant au début et à la fin d'apparition de la glaçure, les attributs extraits ne sont pas suffisamment discriminants pour distinguer l'image présentant une glaçure d'une image

sans glaçure. Notons enfin qu'il est difficile de séparer les deux nuages de points par un simple hyperplan.

Dans le paragraphe suivant nous effectuons une étude comparative de différents réseaux de neurones pour la détection de défauts: le Perceptron multicouche (MLP), le réseau à fonctions radiales de base (RBF), le réseau de neurones probabiliste (PNN) et le réseau à apprentissage par quantification vectorielle (LVQ). Nous étudions de plus leurs performance en fonction de leurs paramètres libres tels le nombre de neurones sur la couche cachée [Ham 98].

5.7 Détection de défauts par réseaux de neurones

Les réseaux de neurones ont été appliqués avec succès dans différents domaines des sciences de l'ingénieur [Gug 96] [Leo 91]. Basés sur le principe de l'apprentissage par les exemples, ils sont capables de résoudre des problèmes complexes où les méthodes classiques ont échoué. La formulation standard de l'ensemble d'apprentissage d'un réseau de neurones pour le problème de la détection nécessite un ensemble de couples d'entrées-sorties (X_n,T_n), n=1,...,N, où X_n est un vecteur d'entrée et T_n est la sortie correspondante appelée aussi la sortie désirée. T_n est égale à 1 si le vecteur X_n est extrait à partir d'une image représentant une bouteille bonne et vaut 0 si X_n provient d'une image représentant une bouteille avec glaçure.

Dans la suite, nous présenterons brièvement différentes architectures de réseaux de neurones afin de comparer leurs performances dans la détection des glaçures.

5.7.1 Perceptron multicouche (MLP)

Dans ce travail nous avons utilisé un MLP à trois couches. Les observations sont présentées séquentiellement à la couche d'entrée. Chaque entrée reçoit la valeur d'un attribut correspondant à son indice et le propage à travers les connexions pondérées vers tous les neurones de la couche cachée. Le nombre H de neurones de cette couche est choisi en appliquant les critère informationnels d'Akaike (FPE et AIC). Chaque neurone de la couche cachée propage à son tour le résultat calculé vers l'unique neurone de sortie qui décide de la présence ou de l'absence d'une glaçure dans l'image représentée par le vecteur d'attributs.

La sortie du réseau MLP s'écrit:

$$y(X) = f\left(b_1 + \sum_{h=1}^{H} v_h f\left(b_0 + \sum_{d=1}^{14} w_{hd} x_d \right) \right) \quad (8)$$

où :

- $f(u) = \frac{1}{1+e^{-u}}$ (7)

est la fonction sigmoïde.

- b_0 et b_1 les biais utilisés respectivement dans la couche d'entrée et la couche cachée,
- w_{hd} est le poids de la connexion reliant le $d^{ème}$ neurone de la couche d'entrée au $h^{ème}$ neurone de la couche cachée,
- v_h est le poids de la connexion du $h^{ème}$ neurone de la couche cachée vers le neurone de sortie.

L'apprentissage du réseau est réalisé en adaptant les poids de façon à minimiser la somme des erreurs au carré entre les sorties y_n du réseau résultant des entrées X_n et les sorties désirées T_n:

$$J(W) = \frac{1}{2} \sum_{n=1}^{N} \left\| y_n - T_n \right\|^2 . \quad (9)$$

L'actualisation des poids est réalisée par l'algorithme de rétropropagation du gradient utilisant la méthode de descente du gradient [Rum 86].

5.7.2 Réseau à fonctions radiales de base (RBF)

Les réseaux de neurones de type RBF trouvent la justification de leur appellation dans le fait qu'ils sont construits à partir de fonctions Gaussiennes symétriques des entrées [Moo 88] [Leo 91]. Dans notre étude, nous avons utilisé des fonctions Gaussiennes plus générales que les fonctions radiales symétriques: leurs surfaces potentielles constantes sont ellipsoïdes alors que celles des fonctions symétriques Gaussiennes sont sphériques.

Les quatre entrées du réseau réalisent une simple propagation à travers des connexions non pondérés vers les neurones cachés. Ainsi, chaque neurone caché h reçoit des valeurs d'entrés non altérées, calcule la distance entre le vecteur d'entrée X et son centre appelé vecteur moyenne. Sa sortie est une fonction non linéaire de cette distance qui est de la forme:

$$\phi_h(X,\mu_h,\Sigma_h)=\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_h|^{\frac{1}{2}}}e^{-\frac{1}{2}(X-\mu_h)^T\Sigma_h^{-1}(X-\mu_h)} \quad (10)$$

où :

- D représente la dimension de l'espace des attributs,
- H est le nombre de neurones sur la couche cachée,
- μ_h, Σ_h représentent respectivement le vecteur moyenne et la matrice de covariance correspondant au neurone caché h. Les matrices Σ_h sont souvent proportionnelles à la matrice unité: $\Sigma_h = \sigma_h I_D$, où σ_h est le paramètre de "scaling" et I_D est la matrice unité de dimension D.

La sortie du réseau RBF est la fonction suivante:

$$y(X)=\sum_{h=1}^{H} v_h \phi_h(X,\mu_h,\Sigma_h). \quad (11)$$

L'apprentissage du réseau RBF consiste à adapter les paramètres μ_h, Σ_h de chaque fonction radiale de base aussi bien que les poids des connexions reliant les neurones cachés au neurone de sortie. La phase d'apprentissage peut être décomposée en trois étapes. D'abord, les vecteurs moyennes μ_h sont déterminés par l'algorithme des K-means. Ensuite, les paramètres σ_h sont déterminés par une heuristique des plus proches voisins. Finalement, les poids des connexions reliant les entrées aux neurones cachés sont déterminés par une régression linéaire.

Dans notre étude, nous avons utilisé l'approche du maximum de vraisemblance et l'algorithme d'estimation de maximisation pour la détermination des paramètres des fonctions radiales de base. Les coefficients v_h sont alors estimés par l'algorithme des moindres moyennes au carré.

5.7.3 Réseau de neurones propabiliste (PNN)

Le classifieur proposé par Specht est une implémentation neuronale de l'estimateur du noyau statistique de Parzen [Spe 90]. Dans cet estimateur, une fonction de base identité qui est souvent une fonction Gaussienne symétrique, est centrée sur chaque observation de l'ensemble d'apprentissage. Ainsi, le nombre des fonctions noyaux est égal au cardinal de l'ensemble d'apprentissage.

Les quatre entrées du réseau PNN sont reliées aux neurones des fonctions noyaux à travers des connexions non pondérées. Les sommes des sorties des fonctions noyaux centrées sur les observations d'une classe fournissent la fonction de densité de probabilité de cette classe:

$$y_k(X) = \frac{1}{N_k (2\pi\sigma^2)^{\frac{D}{2}}} \sum_{m=1}^{N_k} e^{-\frac{1}{2\sigma^2}(X-X_m)^T(X-X_m)} \qquad (12)$$

où:

- N_k est le nombre total des observations de la classe C_k, k=1,...,K,
- X_m est la $m^{ème}$ observation de l'ensemble d'apprentissage m =1, ..., N,
- D est la dimension de l'espace des observations,
- σ est le paramètre de lissage.

Il est important de noter que le réseau PNN ne nécessite pas une procédure d'apprentissage.

5.7.4 Réseau à apprentissage par quantification vectorielle (LVQ)

Le réseau LVQ suppose que plusieurs prototypes sont affectés à chaque classe [Koh 95]. Le nombre total, noté H, des prototypes est alors le paramètre libre du réseau. La phase d'apprentissage consiste à placer les prototypes dans l'espace d'entrée de façon que les erreurs dues aux observations mal classées soient minimisées.

Un vecteur X est affecté à la classe à laquelle appartient le prototype. Soit W_c le prototype le plus proche de X, de façon que:

$$c = \arg \quad \min\left(\left\|X - W_h\right\|^2\right), h = 1,\ldots,H. \qquad (13)$$

Les équations suivantes définissent la procédure standard LVQ:

- $W_c(t+1) = W_c(t) + \alpha(t)\left(X - W_c(t)\right)$ si X est classé correctement, (14-a)
- $W_c(t+1) = W_c(t) - \alpha(t)\left(X - W_c(t)\right)$ si X est mal classé, (14-b)
- $W_h(t+1) = W_h(t)$ pour $h \neq c$. (14-c)

$\alpha(t)$ est un coefficient d'apprentissage qui prend ses valeurs entre 0 et 1 et décroît de façon monotone. Dans nos expériences, nous avons utilisé la procédure optimisée de LVQ appelée OLVQ décrite dans [Koh 88].

5.8 Etude comparative

L'ensemble des données d'apprentissage a été utilisé dans la phase d'apprentissage des différents types de réseaux de neurones présentés dans le paragraphe 5.7. Pour chaque type de réseau, nous avons fait varier les paramètres libres qui représentent le nombre de neurones cachés (MLP et RBF) ou la paramètre de lissage (PNN) ou encore le nombre de neurones prototypes (LVQ).

Ensuite, les performances de chacun de ces réseaux ont été évaluées en utilisant les observations de l'ensemble de test. Les résultats obtenus, exprimés en terme de pourcentage de reconnaissance correcte des images avec ou sans glaçure sont présentés dans le tableau 5.2. Les nombres en gras indiquent l'architecture conduisant à l'erreur minimale pour chaque type de réseau de neurones.

Toutes les images provenant de bouteilles sans défaut ont été reconnues par les réseaux MLP, RBF et PNN. Le réseau LVQ n'a pas reconnu 1% des images de bouteilles sans défaut utilisés pour l'apprentissage.

Les résultats obtenus par ces réseaux sont pratiquement similaires. Cependant, le taux minimum d'erreur a été atteint par les réseaux RBF et PNN pour lesquelles toutes les images sans glaçures ont été reconnues et 86.4 % des images présentant une glaçure ont été reconnues. Afin d'expliquer les 13.6 % d'images non détectées, nous avons examiné chacune de celles-ci. Nous avons alors constaté que toutes ces images présentent de petites zones de réflexions de glaçures qu'on peut difficilement distinguer des réflexions dues au filetage du goulot des bouteilles.

Heureusement, quant une bouteille avec défaut fait une rotation autour d'elle même en face de la machine de vision, une séquence de 16 images peut contenir au moins trois à quatre images avec des réflexions dues aux glaçures. Si les petites zones claires correspondant au début ou à la fin d'apparition d'une glaçure ne sont pas détectées, le système de diagnostic reconnaît dans la séquence au moins une image avec défaut et par conséquent une bouteille avec défaut peut être détectée au moins une fois. Nous avons présenté toutes les séquences des images des bouteilles avec défaut aux quatre types de réseaux. Les réseaux RBF, MLP et PNN ont détecté les bouteilles défectueuses avec 100% de réussite.

Type de réseau de neurones	H	Performances en % sur des images de bonnes bouteilles	Performances en % sur des images de bouteilles avec glaçure
MLP	**2**	**100**	**83.1**
(H = nombre de neurones cachés)	3	100	76.3
	4	100	74.6
	5	100	79.7
	6	100	74.6
	7	100	78.0
	8	100	74.6
	9	100	76.3
RBF	2	100	83.1
(H = nombre de neurones cachés)	3	100	76.3
	4	**100**	**86.4**
	5	100	78.0
	6	100	78.0
	7	100	79.7
	8	98.1	81.4
	9	97.1	81.4
	30	99.0	88.1
PNN	38	99.0	88.1
(H = Paramètre de lissage)	44	99.0	86.4
	52	**100**	**86.4**
	60	100	86.4
	68	100	86.4
LVQ	2	98.1	84.8
(H = Nombre de neurones prototypes)	3	96.1	84.8
	4	99.0	83.1
	6	97.1	84.8
	8	**99.0**	**84.8**
	10	98.1	84.8
	12	98.1	84.8

Tableau 5.2 Performance des réseaux MLP, RBF, PNN et LVQ pour les résultats des tests réalisés sur des bouteilles avec et sans défauts.

5.9 Conclusion

Nous avons présenté un système d'inspection de défauts utilisant une caméra matricielle CCD pour la détection de glaçures sur la bague des bouteilles en verre. Après extraction et sélection des attributs les plus discriminants par l'algorithme de l'analyse discriminante pas à pas, nous avons appliqué deux méthodes de projection planes pour l'analyse de la structure des données. Nous avons ensuite appliqué les réseaux les plus connus pour décider de la présence ou de l'absence de glaçures dans chaque image d'une bouteille observée par une séquence de 16 images. Les résultats obtenus avec ces réseaux sont assez similaires. Le réseau à fonctions radiales de base (RBF) et le réseau probabiliste (PNN) semblent les mieux adaptés pour la détection de défauts. En effet, ces réseaux créent des domaines de décisions fermés ce qui a pour effet de permettre de refuser ou rejeter une bouteille se situant hors des domaines définis par ces réseaux.

Chapitre 6

Suivi de fonctionnement d'une éolienne par réseaux de neurones

Chapitre 6 Suivi de fonctionnement d'une éolienne par réseaux de neurones

6.1 Introduction

L'énergie éolienne est une énergie non polluante qui rencontre de plus en plus de succès dans le contexte écologique actuel. Cependant, pour assurer une meilleure utilisation et une plus large diffusion de cette énergie, il est nécessaire de bien maîtriser son coût d'exploitation et d'optimiser son rendement qui reste largement lié aux conditions climatiques. Une amélioration substantielle de notre compréhension des générateurs éoliens ainsi que notre capacité à prévoir leur comportement sont essentielles pour fournir de manière précise et fiable une prédiction de cette énergie. De plus, la détection précoce des dysfonctionnements ou des pannes est une caractéristique très importante des systèmes de suivi de fonctionnement des structures éoliennes soumises à des conditions climatiques variables et exigeant peu de maintenance.

Ce chapitre présente un système de suivi de fonctionnement d'une éolienne dont le but est de réduire les coûts de maintenance et d'améliorer le rapport coût/bénéfice énergétique. Les systèmes de suivi de fonctionnement en temps réel ont été implantés avec succès dans différents domaines industriels tels que les centrales nucléaires [PAR 94], l'industrie chimique [WAT 89] [SOR 91], l'aérospatiale, ou l'industrie automobile afin d'améliorer la sûreté et la fiabilité de leur fonctionnement [MAK 97]. Ces systèmes de suivi et de diagnostic nécessitent généralement la connaissance d'un modèle physique d'entrées-sorties du processus à surveiller. Or, la diversité de conception des éoliennes disponibles dans le commerce rend le développement d'un système de suivi générique très délicat à réaliser. De plus, les grandeurs d'entrée d'une éolienne, c'est à dire la vitesse du vent et la direction, ne peuvent pas être mesurées de façon exacte. De ce fait, les méthodes basées sur la connaissance d'un modèle entrées-sorties ne peuvent pas s'appliquer [CAS 94].

D'une manière générale, un système de suivi de fonctionnement est structuré en trois étapes:

- Acquisition des signaux issus des capteurs installés sur le processus à surveiller,
- Traitement des signaux et extraction des attributs caractéristiques du fonctionnement du processus,
- Détection d'un dysfonctionnement du processus.

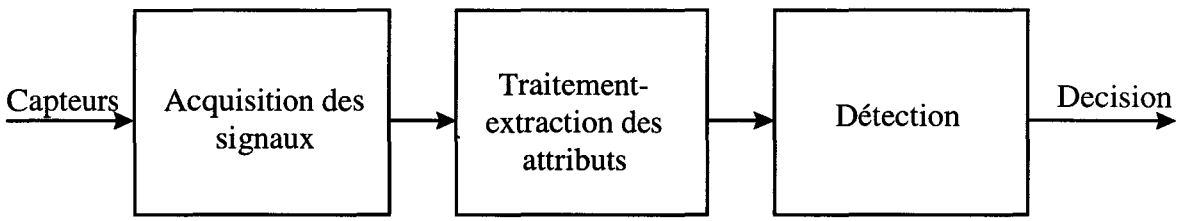


Figure 6.1 Description du système de suivi de fonctionnement

Comme il est difficile, voire dangereux, de créer une panne sur une machine réelle, nous proposons une solution neuronale basée uniquement sur la connaissance du fonctionnement normal de l'éolienne. Le réseau de neurones est conçu pour déterminer la frontière entre les états de fonctionnement normal et ceux correspondant à un fonctionnement anormal en utilisant uniquement les signaux enregistrés sous des conditions de fonctionnement normal de l'éolienne. L'approche que nous proposons emploie un réseau autoassociateur et nécessite uniquement la disponibilité d'une base de données suffisante traduisant le fonctionnement normal.

Le paragraphe 2 décrit les composants d'une éolienne et leurs défauts. Le paragraphe 3 montre l'installation des capteurs sur les différentes parties de l'éolienne pilote que nous avons instrumentée. Dans les paragraphes 4 et 5 nous présentons le système d'acquisition, les procédures de prétraitement du signal et d'extraction des attributs. Le domaine du fonctionnement normal de la machine est défini à partir des erreurs de reconstruction des entrées du réseau.

6.2 Les principaux composants d'une éolienne et leurs défauts typiques

L'éolienne, sujet de notre étude, est installée à Malo les Bains dans le Nord de la France. C'est une machine à trois pâles de diamètre 30 mètres et délivre une puissance de 300 KW. Pour

bien comprendre le système de suivi de fonctionnement de l'éolienne, nous décrivons brièvement les composants de cette machine.

L'éolienne est constituée d'une nacelle supportée par un mât de 21,7 mètres de hauteur. La nacelle comporte un arbre lent qui supporte les trois pâles, un multiplicateur de vitesse, et une génératrice asynchrone. Une vue globale des principaux éléments de l'éolienne est présentée dans la figure 6.2.

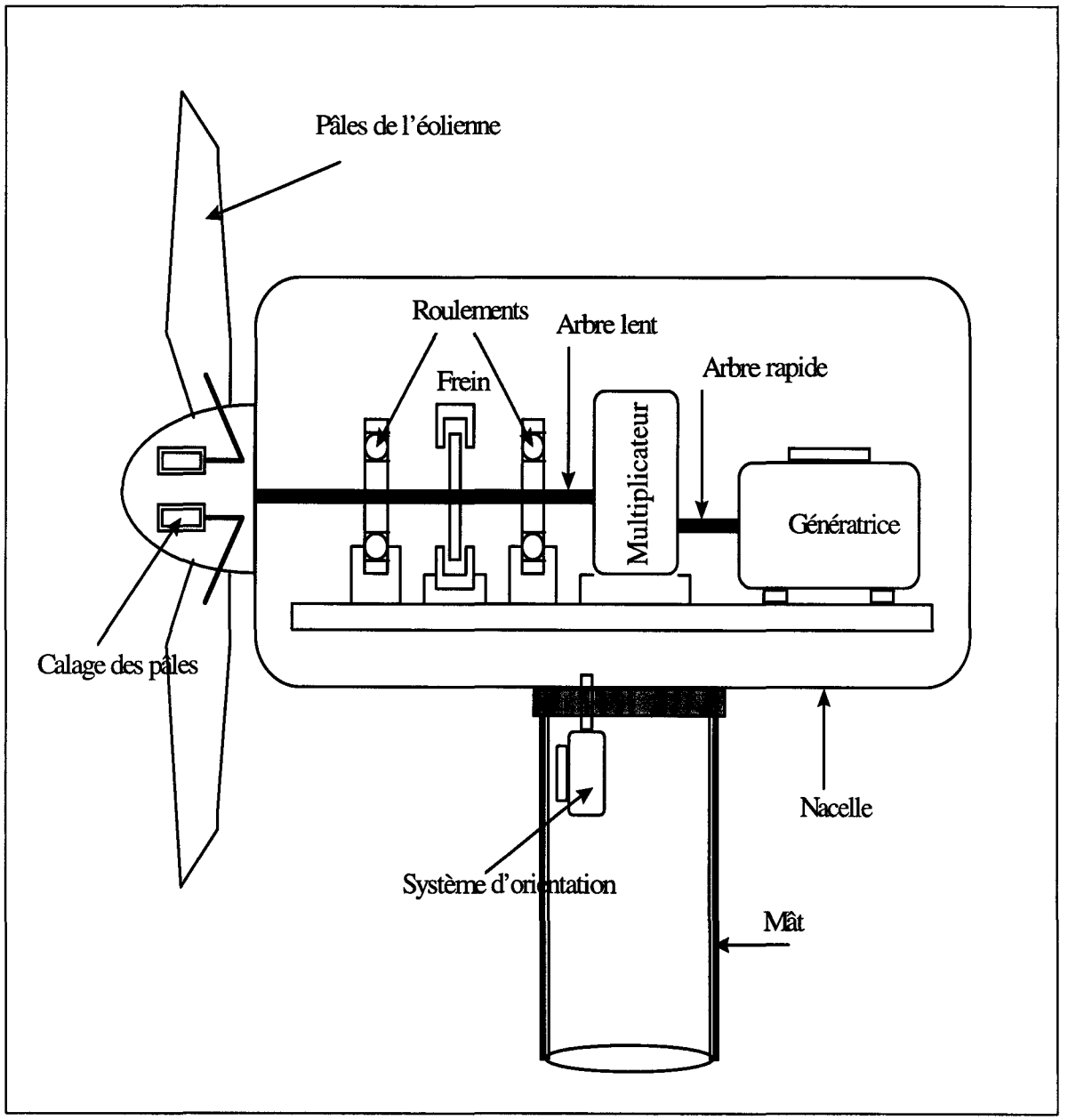


Figure 6.2 Vue générale des éléments composant une éolienne

L'hélice à trois pâles assure un couple de démarrage élevé. Le calage des pâles est fixe et la régulation se fait par l'intermédiaire d'une palette disposée dans le même plan que l'hélice, donc normale à la direction du vent.

Pour éviter que la machine ne soit endommagée lorsque le vent atteint des vitesses trop importantes, cette palette place l'éolienne dans le lit du vent (mise en drapeau de l'éolienne), provoquant ainsi son arrêt. On remarquera que ce système de production d'énergie mécanique présente l'inconvénient de ne pas fonctionner pour les vents faibles (inférieures à 5 m/s), ni pour les vents très forts (supérieurs à 18 m/s). Son exploitation n'est donc valable que dans les zones à vents moyens réguliers et à durées significatives.

6.2.1 Les pales et le rotor

Les constructeurs d'éoliennes ont identifié des pannes typiques liées aux pâles. Le profil des contours des pâles, les variations des angles, ou les déviations aérodynamiques de fabrication peuvent provoquer des problèmes sérieux sur la production du courant. De plus, les variations des contours des pâles tendent à paraître comme la conséquence des effets de fatigue dus au vieillissement des matériaux. Ces changements des profils aérodynamiques des pâles résultant d'une imperfection dans la fabrication ou causés par les effets de fatigue peuvent contribuer à une augmentation significative des émissions de bruit et provoquer ainsi une dégradation de la qualité de l'énergie produite. D'autres pannes typiques du rotor peuvent trouver leur origine dans la rugosité de la surface, le gel, les fissures, etc...

6.2.2 Le multiplicateur de vitesse

L'inconvénient de la conception classique d'une éolienne asynchrone tournant à une vitesse constante est la haute charge mécanique que supportent ses composants, en particulier le multiplicateur de vitesse. Selon les rapports de maintenance, le multiplicateur de vitesse doit être remplacé tous les cinq à dix ans, ce qui influe bien sur le prix du KWh produit. Le multiplicateur est souvent considéré comme l'élément le plus fragile et le plus coûteux, en particulier quand la technologie permet la fabrication de machines dont la puissance est de l'ordre des Mega Watt.

Les pannes au niveau du multiplicateur sont généralement dues à la fatigue des roulements et des arbres de transmission et à la détérioration des dents des engrenages. Ces pannes peuvent être détectées par une analyse des vibrations ou une mesure de la température de l'huile.

6.2.3 La génératrice

Le déséquilibre des phases est le type de pannes le plus fréquemment rencontré sur les génératrices asynchrones. Ce type de panne peut être détecté par mesure des phases du courant aussi bien que des signaux vibratoires de la génératrice.

Dans la suite on s'intéresse au suivi de fonctionnement du multiplicateur et de la génératrice par analyse de leurs signaux vibratoires.

6.3 Installation des capteurs

Selon le guide de certification Germanischer Lloyd [GER 93], le système de suivi de fonctionnement et de diagnostic d'une éolienne nécessite la surveillance de la vitesse de rotation du rotor, des déplacements de la nacelle et de la température du multiplicateur etc. Pour cela différents types de capteurs doivent être installés sur les principaux composants de l'éolienne.

Le choix des capteurs et de leurs emplacements est d'une importance primordiale car les signaux issus de ces capteurs doivent traduire les changements inhérents de l'état de fonctionnement de la machine. Pour ce faire, nous nous sommes guidés par la méthode AMDEC (Analyse des Modes de Défaillances de leurs Effets et de leur Criticité [RAP 97].

Pour surveiller les différents éléments de l'éolienne, nous avons installé les capteurs suivants :

- deux accéléromètres positionnés en quadrature de phase mesurent les vibrations sur le multiplicateur de vitesse : le premier est parallèle à l'axe du rotor et le second est perpendiculaire à cet axe,
- un accéléromètre mesure les vibrations de la génératrice,
- un codeur incrémental, couplé à l'arbre lent, mesure la vitesse de rotation du rotor,
- trois codeurs inductifs mesurent l'angle d'orientation de la nacelle,

- un capteur de déplacement mesure le pas des pâles,
- une pince ampèremétrique mesure le courant délivré par la génératrice,
- deux pinces mesurent les tensions sur deux des trois phases du courant triphasé.

La figure 6.3 montre uniquement l'emplacement des accéléromètres sur le multiplicateur et la génératrice ainsi que la pince ampèremétrique.

Les signaux issus des capteurs fournissent à chaque instant l'état de fonctionnement des composants de l'éolienne. Généralement, ces grandeurs mesurées sont comparées à des seuils prédéterminés. Cependant, le développement d'un système de suivi ne peut pas reposer uniquement sur cette simple technique de comparaison pour révéler les effets lents et évolutifs de fatigue des composants de la machine qui dégradent ses performances jusqu'à l'apparition d'une défaillance.

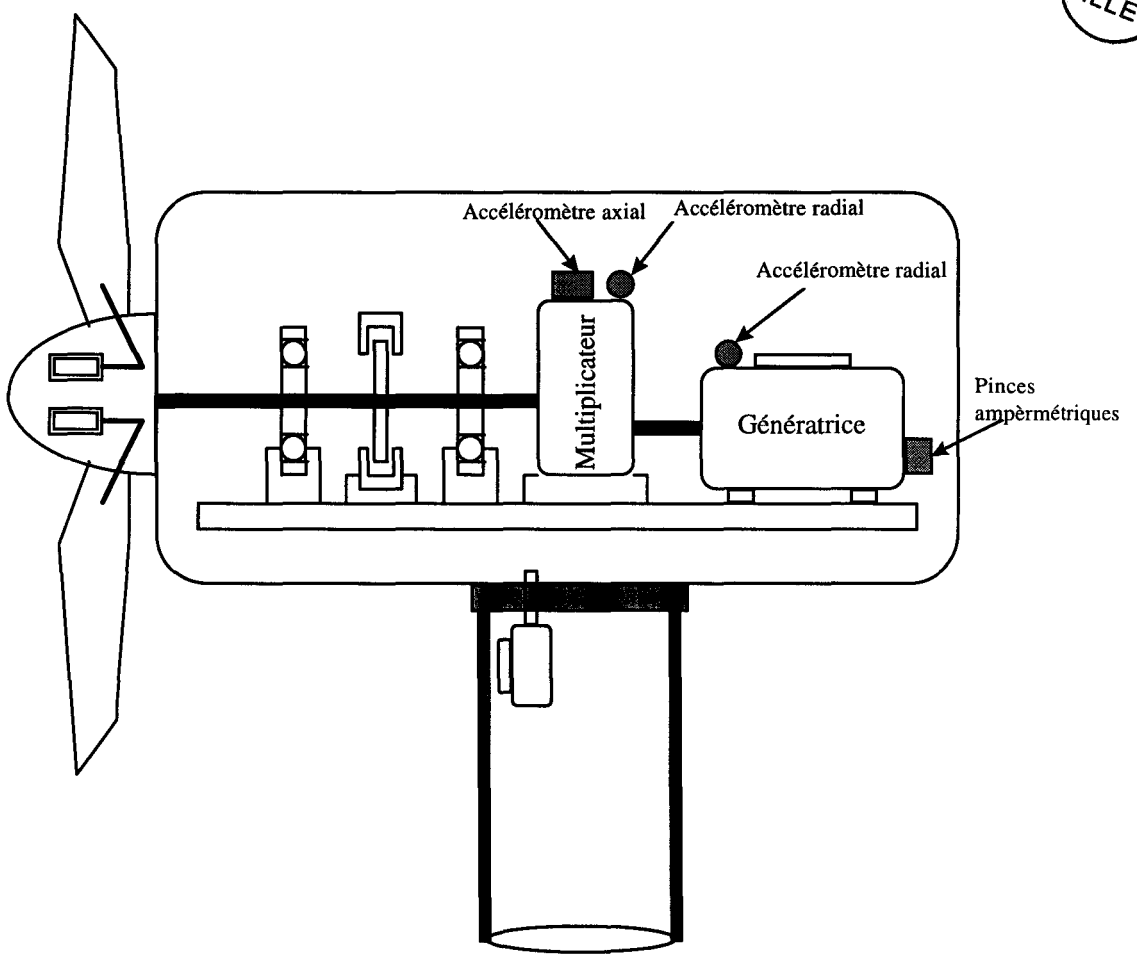


Figure 6.3 Capteurs installés sur l'éolienne.

6.4 Système d'acquisition des signaux

Pour constituer une base de données permettant de suivre l'état de fonctionnement de l'éolienne, les signaux des capteurs installés sur les différents composants sont enregistrés grâce à une carte d'acquisition à 16 entrées pouvant atteindre un taux d'échantillonnage de 250 KHz. Le taux d'échantillonnage des signaux a été fixé à 31 KHz sur chaque entrée. Cette carte est installée dans un ordinateur placé au pied de l'éolienne (cf. figure 6.3). Nous avons développé un programme d'acquisition événementiel qui effectue l'enregistrement des signaux selon des conditions définies par l'utilisateur : puissance délivrée, vitesse du vent etc. Ce programme permet un enregistrement automatique si certaines de ces conditions sont remplies. Cette base de données issue des capteurs est enregistrée sur un disque amovible à haute capacité de stockage.

6.5 Traitement des signaux et extraction des attributs

La figure 6.4 (a) représente un signal temporel issu de l'accéléromètre axial du multiplicateur. Ce signal a été échantillonné à 31 KHz et la longueur de la fenêtre de Hanning a été choisie à 8192 échantillons afin de respecter un compromis entre la résolution du spectre et le temps de calcul. La figure 6.4 (b) représente le spectre du signal obtenu par la transformée de Fourier rapide appliquée sur la fenêtre temporelle de Hanning [Ber 98a].

Les fréquences 555, 679 et 843 Hz correspondent aux plus grandes amplitudes et représentent les vibrations physiques des éléments mécaniques du multiplicateur. Surveiller le multiplicateur revient à estimer périodiquement le spectre du signal de son accéléromètre et à suivre son évolution au cours du temps.

Comme les spectres ont des profils qui varient au cours du temps, il est intéressant de les visualiser tous en même temps pour avoir une vision dynamique de leur évolution. Le principe consiste à élaborer un spectrogramme du signal constitué d'une succession de spectres calculés sur un enregistrement qui dans notre cas est d'une durée de 10 secondes comme le montre la figure 6.4 (c). L'axe horizontal représente les fréquences du spectre du signal dont l'amplitude est représentée en couleur (le noir correspond à des amplitudes faibles alors que le vert

correspond à l'amplitude maximale). L'axe vertical est celui du temps en secondes. Il s'agit donc d'une représentation de l'amplitude du spectre en fonction de la fréquence et du temps.

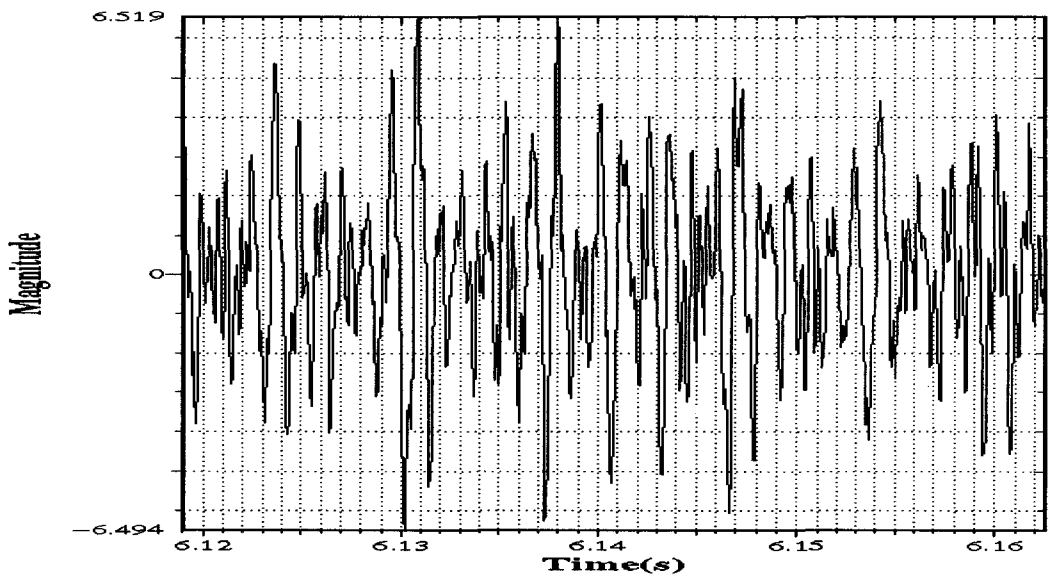


Figure 6.4 (a) Signal temporel des vibrations du multiplicateur.

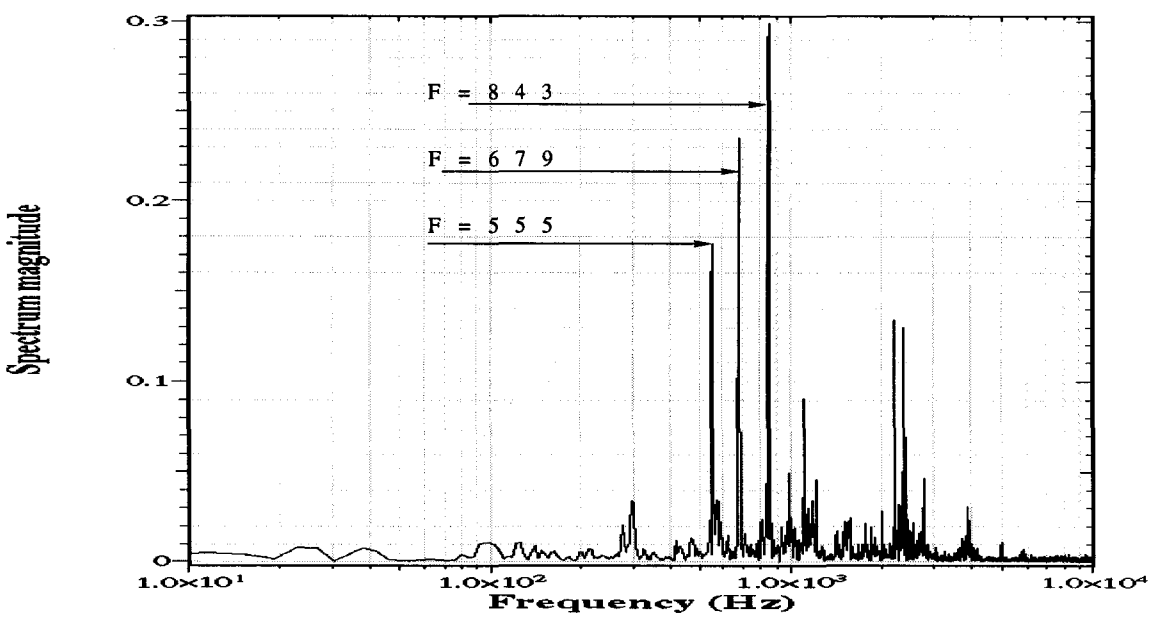


Figure 6.4 (b) Spectre du signal de vibration du multiplicateur.

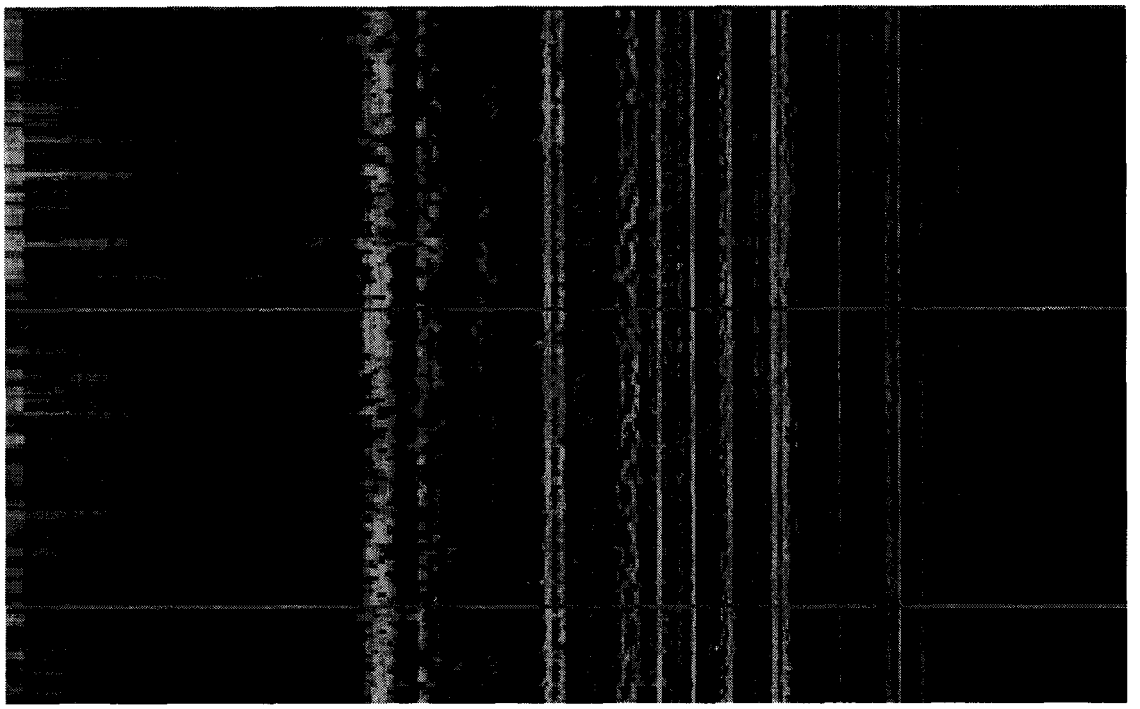


Figure 6.4 (c) Spectrogramme du signal

Pour exploiter efficacement le spectre du signal, nous avons employé une procédure de seuillage manuel en observant les différents spectres extraits des fenêtres de Hanning. Les bandes étroites ont été centrées autour des principaux pics et séparées par des bandes plus larges traduisant une absence de pics dans le spectre (cf. figure 6.4 (d)). La sensibilité du système de supervision conditionne le choix de la largeur des bandes. En effet, toute détérioration d'un élément mécanique peut se traduire soit par le déplacement des fréquences correspondantes, soit par l'apparition de nouvelles fréquences.

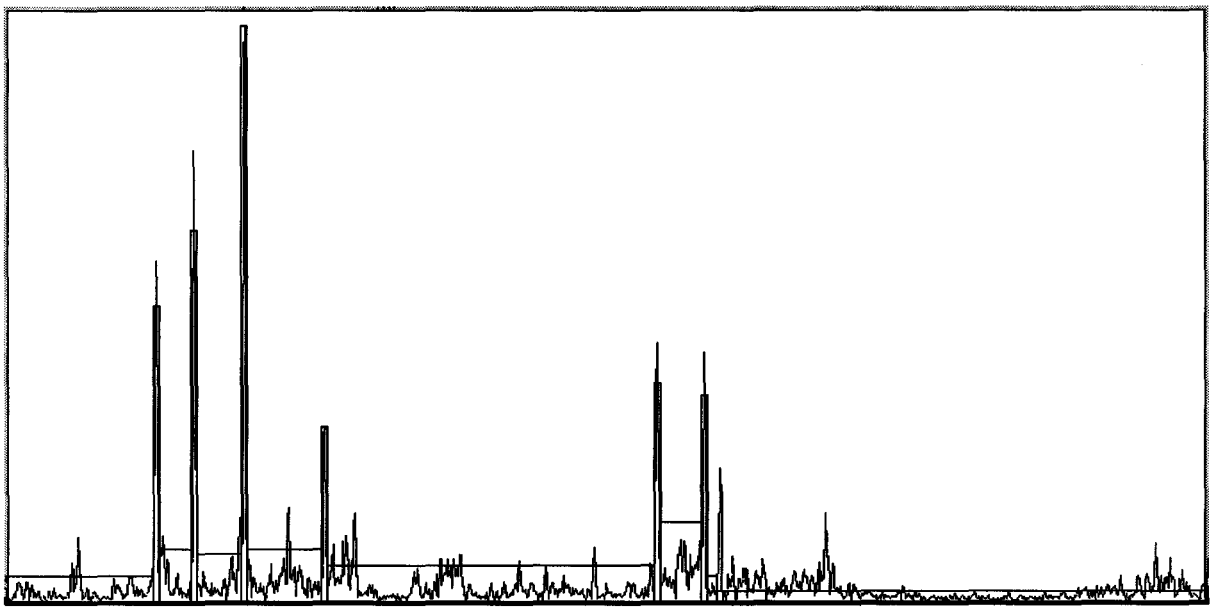


Figure 6.4 (d) Décomposition manuelle en bandes du spectre

La figure 6.4 (e) montre le spectre réduit où les amplitudes sont les moyennes des amplitudes des pics à l'intérieur de chaque bande. Les amplitudes des bandes au nombre de 14 constituent les entrées ainsi que les sorties du réseau utilisé pour suivre le fonctionnement de l'éolienne [Ber 98a].

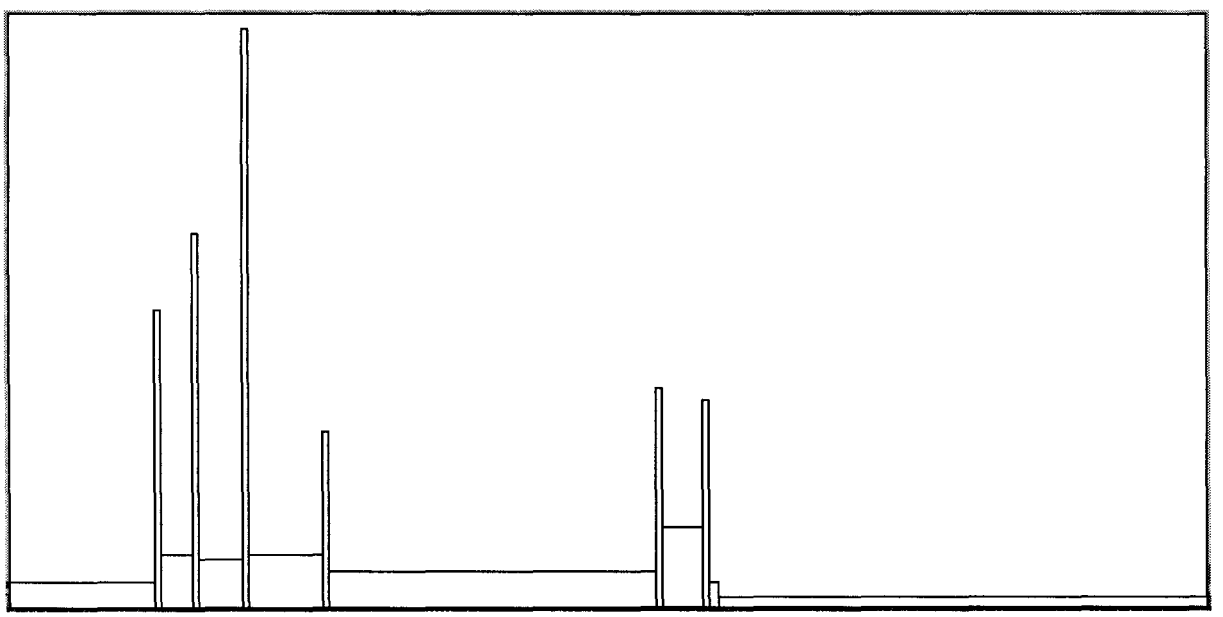


Figure 6.4 (e) Spectre réduit à 14 bandes où les amplitudes représentent les moyennes des pics dans chaque bande

Les spectres réduits constituent la base de données qui sera utilisée par un réseau de neurones autoassociateur pour le suivi de fonctionnement du multiplicateur.

6.6 Suivi de fonctionnement par réseau de neurones autoassociateur

Puisqu'il est très difficile de créer une dégradation sur un élément quelconque de l'éolienne, nous proposons un système de surveillance basé uniquement sur la connaissance contenue dans la base de donnés des signaux acquis pendant le fonctionnement normal de l'éolienne. Ainsi, nous utiliserons un réseau de neurones à apprentissage non supervisé pour mémoriser les modes de fonctionnement normal. L'architecture du réseau autoassociateur du type 14x10x8x10x14 a été expliquée dans le chapitre 3. Les entrées du réseau sont reproduites à sa sortie à travers les couches cachées. Ce réseau admet donc 14 entrées et 14 sorties correspondant aux amplitudes des 14 bandes du spectre réduit (cf. figure 6.5 (a)). Il admet deux couches cachées de taille 10 de part et d'autre de la couche cachée centrale qui est de taille 8. [Ber 98a], [Ber 98b]. L'architecture du réseau a été optimisée grâce aux critères informationnels FPE et AIC expliqués au chapitre 3.

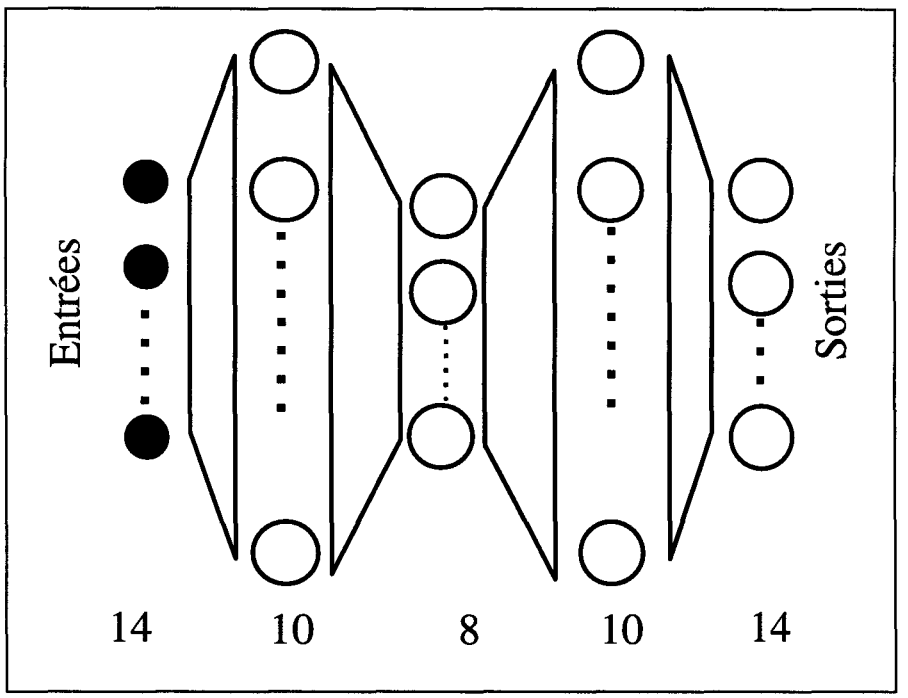


Figure 6.5. (a) Réseau autoassociateur à 3 couches cachées.

Les sorties de la couche cachée centrale, ou couche de projection, constituée de huit neurones sont utilisées pour réaliser la projection non linéaire des données présentées à l'entrée du réseau. En effet, le réseau autoassociateur peut être divisé en deux réseaux : le premier implémente la fonction de projection alors que le second réalise la fonction de reconstruction.

L'apprentissage consiste à appliquer aux entrées aussi bien qu'aux sorties les amplitudes des spectres réduits et à adapter les poids pour minimiser le critère d'erreur du réseau. A la fin de l'apprentissage le réseau mémorise les différentes situations de fonctionnement normal enregistrées de l'éolienne.

Signalons que le réseau autoassociateur est ici employé non pas pour projeter et visualiser les valeurs des entrées sur un plan mais plutôt dans le but de fournir un modèle du fonctionnement de l'éolienne. Le réseau apprend dans sa structure le fonctionnement normal qu'il reproduit en sortie.

Lors de la phase de test on présente un spectre réduit représentant une fenêtre d'enregistrement du fonctionnement de l'éolienne. Le réseau fournit en sortie les amplitudes estimées des bandes de fréquences correspondant au fonctionnement normal. La comparaison entre le spectre réduit et le spectre estimé permettent de conclure à l'état de fonctionnement de l'éolienne.

Nous avons représenté sur le tableau 6.1 et sur la figure 6.6 les pourcentages des erreurs obtenues à la sortie du réseau pour les phases d'apprentissage et de test qui sont assez voisines. Il suffit alors de fixer un seuil de détection dépendant de l'erreur de reconstruction de l'entrée du réseau. A chaque présentation d'un spectre aux entrées du réseau celui-ci fournit en sortie une estimation. Les écarts entre les entrées et les sorties estimées sont calculés et comparés aux seuils de détections. Tout écart dépassant le seuil correspondra à un changement de fréquence et correspondra à un fonctionnement anormal.

Notons que les entrées correspondant à des bandes larges de fréquence comme les entrées E_1 et E_{14} sont reconstruites avec un taux d'erreur relativement important. Par contre les entrées représentant des pics de fréquence comme les entrée E_2, E_4 et E_6 sont assez bien reconstruites. Or ce sont surtout certaines entrées correspondant à des pics de fréquences spécifiques qui nous intéressent le plus car elles sont caractéristiques du multiplicateur.

Erreurs partielles	% d'erreur en mode apprentissage	% d'erreur en mode test
E_1	8.02	8.83
E_2	2.33	2.62
E_3	5.63	6.37
E_4	1.98	2.44
E_5	6.23	7.41
E_6	4.79	5.39
E_7	4.82	5.54
E_8	6.90	7.14
E_9	5.65	5.76
E_{10}	4.79	5.02
E_{11}	6.47	7.20
E_{12}	2.28	2.78
E_{13}	5.57	6.37
E_{14}	6.44	7.50

Tableau 6.1. Pourcentages d'erreurs en apprentissage et en test

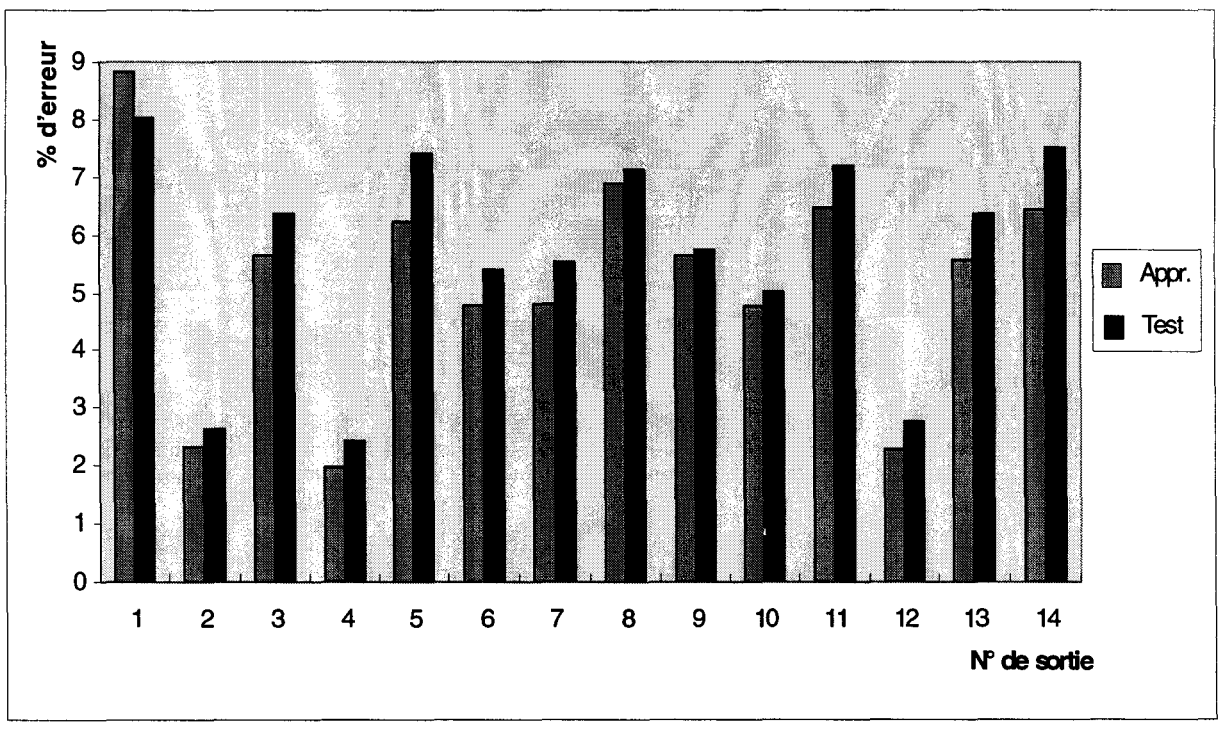


Figure 6.6. Pourcentage d'erreur en apprentissage et test du réseau autoassociateur

6.7 Conclusion

Dans ce chapitre nous avons proposé une méthode de suivi du fonctionnement du multiplicateur d'une éolienne pilote. Comme il est difficile de créer des pannes sur l'éolienne, nous avons procédé à la surveillance du fonctionnement normal de la machine. Pour ce faire nous avons installé sur l'éolienne différents types de capteurs et en particuliers des d'accéléromètres.

Après la phase de prétraitement du signal de vibration recueilli par l'accéléromètre du multiplicateur, nous avons extrait les attributs caractéristiques à partir du spectrogramme du signal. Nous avons ainsi constitué une base de données pour l'étude du comportement du multiplicateur. Un réseau de neurones autoassociateur a été employé pour modéliser le fonctionnement normal du multiplicateur à la fin de son apprentissage. Le réseau autoassociateur admet deux types d'utilisations: en projection et en reconstruction. Dans la phase de reconstruction, le réseau modélise le spectre du signal du multiplicateur en mode de fonctionnement normal.

Lors de son exploitation, le spectre acquis sur une fenêtre donnée est présenté à l'entrée du réseau. Celui-ci fournit une estimation du spectre représentant le fonctionnement normal. La comparaison entre le spectre présenté et le spectre obtenu par le réseau permet de juger de la situation du multiplicateur. En effet tout écart significatif entre les deux spectres sera considéré comme correspondant à un fonctionnement anormal.

Conclusion générale

Conclusion générale

Dans ce travail nous avons présenté une synthèse des méthodes de projection plane linéaires et non linéaires basées sur des architectures neuronales à apprentissage non supervisé. Ces méthodes non seulement complètent les méthodes de projection classiques, telles que l'analyse en composante principales et l'algorithme de Sammon, mais apportent une contribution indéniable à la réduction de la dimension. En effet, la projection est élaborée par l'architecture du réseau dont les coefficients sont réglés par une stratégie d'apprentissage. Une fois l'apprentissage terminé, les équations de fonctionnement du réseau établissent une transformation analytique entre l'espace des observations à l'espace des projections.

Nous avons exposé une famille de réseaux de neurones à apprentissage par la règle de Hebb qui réalisent une analyse en composantes principales linéaire. La famille des réseaux multicouche constituent des méthodes d'analyse en composantes principales non-linéaires. Pour ces réseaux, nous avons proposé deux critères informationnels de Akaike pour optimiser leurs architectures. Nous avons proposé une nouvelle technique d'apprentissage par le maximum de vraisemblance basée sur la minimisation des distances entre les observations et leurs projections. Différentes méthodes de projection ont été programmées et leurs résultats comparés.

Au cours de notre recherche nous avons travaillé sur deux contrats industriels pour le contrôle qualité et le suivi de fonctionnement des machines. Le premier a porté sur la détection de glaçures sur les goulots des bouteilles en verre. Le travail réalisé a été divisé en deux grandes parties: la première a concerné l'acquisition des images, leurs analyse en vu de l'extraction des attributs pertinents. Dans la deuxième partie, nous avons exploité les attributs extraits pour constituer une base d'apprentissage et de test pour comparer les résultats des réseaux de neurones les plus connus en classification. Ces réseaux ont donné des résultats intéressants dans la distinction entre bonnes bouteilles et celles présentant des glaçures.

La seconde application industrielle a consisté à étudier les signaux provenant des vibrations de certains éléments d'une éolienne pilote installée à Dunkerque en vue de la prédiction de dysfonctionnements éventuels. Nous avons instrumenté l'éolienne à l'aide de capteurs

appropriés. Nous avons ensuite constitué une base de données de signaux issus des capteurs représentant le fonctionnement normal de l'éolienne. Cette base a servi à l'apprentissage du réseau autoassociateur pour modéliser le fonctionnement normal de l'éolienne.

Ce travail est poursuivi pour approfondir la caractérisation du fonctionnement de l'éolienne par des signatures fréquentielles et temporelles. Des réseaux de neurones multicouches et modulaires à apprentissages supervisés seront employés pour modéliser et exploiter les relations pouvant exister entre les signaux des différents capteurs.

Publications personnelles

Publications personnelles

Publication dans une revue

1. D. Hamad, M. Betrouni, P. Biela and J.-G. Postaire, **"Neural Networks for Glass Bottles Production: a Comparative Study".** International Journal of Pattern Recognition and Artificial Intelligence, World Scientific Publishing Company, Vol. 12, No. 4, 505-516, 1998.

Communications dans des conférences internationales avec comité de lecture et publication des actes

2. D. Hamad and M. Betrouni, **"Artificial Neural Networks for Non Linear Projection and Exploratory Data Analysis"**. International Conference on Artificial Neural Networks and Genetic Algorithms, ICANNGA'95, pp. 164-167, Alès, FRANCE, Avril 18-21, 1995.

3. M. Betrouni, S. Delsert S. and D. Hamad, **"Interactive Pattern Classification by Means of Artificial Neural Networks".** IEEE International Conference on Systems, Man, and Cybernetics (SMC), Vol. IV, pp. 3275-3279, Vancouver, British Columbia, CANADA, October 22-25, 1995.

4. M. Betrouni, D. Hamad and J.-G. Postaire, **"Feature selection and fault detection in glass bottles production".** 1st International Conference on Engineering Design and Automation, Bangkok, THAILAND, March 18-19, 1997.

5. M. Betrouni, D. Hamad and J.-G. Postaire, **"Feature Selection and Fault Detection in Glass Bottles Production".** Neural Networks in Engineering Systems, Proceeding of the 1997 International Conference on Engineering Applications of Neural Networks, EANN'97, pp. 49-52, Stockholm, SWEDEN, 16-18 June, 1997.

6. R. Bertrand, N. El Hor, M. Betrouni, and D. Hamad, **"On-line supervision system for wind energy converters"**. Engineering Benefits from Neural Networks, pp. 269-272. Editors, A. B. Bulsari, J. Fernandez de Caneta and S. Kallio. Proceedings of the fourth International Conference on Engineering Applications of Neural Networks, EANN'98, Gibraltar, June 10-12, 1998.

Résumé dans un congrès national

7. M. Betrouni et D. Hamad, ***"Réseau de Neurones Multicouches Appliqué à la Classification Interactive de Données Multidimensionnelles".*** Secondes Rencontres de la Société Francophone de Classification, Tours, FRANCE, 12-13 Septembre, 1994.

Références bibliographiques

Références bibliographiques

[Aka 72] H. Akaike, **"Information theory and an extension of the maximum likelihood principle"**. 2nd international Symposium on Information Theory, pp. 267-281, 1972.

[Aka 74] H. Akaike, "**A New look at the statistical model identification**". IEEE Trans. on Automatic Control, Vol. AC-19, No. 6, pp. 716-722, 1974.

[Bet 95] M. Betrouni, S. Delsert and D. Hamad, "**Interactive pattern classification by means of artificial neural networks**". IEEE International Conference on System Man and Cybernetics (SMC), Vol. IV, pp. 3275-3279, Vancouver, British Columbia, CANADA, October 22-25, 1995.

[Bet 97a] M. Betrouni, D. Hamad and J.-G. Postaire, **"Feature selection and fault detection in glass bottles production".** 1st International Conference on Engineering Design and Automation, Bangkok, THAILAND, March 18-19, 1997.

[Bet 97b] M. Betrouni, D. Hamad and J.-G. Postaire, **"Feature Selection and Fault Detection in Glass Bottles Production".** International Conference on Engineering Applications of Neural Networks, EANN'97, Stockholm, Sweden, 16-18 June, 1997.

[Ber 98a] R. Bertrand R., N. El Hor, M. Betrouni, and D. Hamad, "**On-line supervision system for wind energy converters**". Engineering Benefits from Neural Networks, pp. 269-272. Editors, A. B. Bulsari, J. Fernandez de Caneta and S. Kallio. Proceedings of the fourth International Conference on Engineering Applications of Neural Networks, EANN'98, pp. 269-272, GIBRALTAR, June 10-12, 1998.

[Ber 98b] Bertrand R. N. El Hor, Hamad D., and Postaire J.-G. "***Supervision of wind energy converters by non linear mapping neural network".*** International ICSC/IFAC Symposium on Neural Computation NC'98, Vienna, AUSTRIA, September 23-25, 1998.

[Bou 88] H. Bourlard and Y. Kamp, **"Auto-association by multilayer perceptrons and singular value decomposition"**. Biological Cybernetics, Springer-verlag, Vol. 59, pp. 291-294, 1988.

[Cas 94] P. Caselitz, J. Glebhardt and M. Mevenkamp, "**On-line fault detection and prediction in wind energy converters**", EWEC'94, pp. 623-627, 1994.

[Cel 89] G. Celeux, E. Diday, G. Govaert, Y. Lechevalier, H. Ralambondrainy, **"Classification automatique des données"**. Dunod Informatique, Bordas, Paris, 1989.

[Cel 92] G. Celeux, **"Modèles probabilistes en classification"**. Chapitre 6 du livre Modéles pour l'analyse des données multidimensionnelles, édition Economica, J.-J. Droesbeke, B. Fichet, P. Tassi, éditeurs, pp. 165-214, 1992.

[Cot 88] G.W. Cottrell, P.W. Monro and D. Zipser, **"Image compression by back propagation: a demonstration of extension programming"**. In: Sharkey NE. Advances in cognitive science, Vol 2., Abbex, Norwood, (NJ), 1988.

[Dao 93] M. Daoudi, **"Classification interactive multidimensionnelle par les réseaux de neurones et la morphologie méthématique".** Thèse de doctorat, Université des Sciences et Technologies de Lille, 23 Novembre 1993.

[Dao 94] M. Daoudi, D. Hamad and J.-G. Postaire, **"A New Interactive Pattern Classification Approach Using Five-Layer Neural Networks and Mathematical Morphology".** EUFIT'94, Aachen, GERMANY, September 20-23, 1994.

[Dud 73] R.O. Duda and P. E. Hart, "**Pattern classification and scene analysis**". J. Wiley, New York, 1973.

[Fir 96] C. Firmin, D. Hamad, J.-G. Postaire and R. D. Zhang, "**Feature Extraction and Selection for Fault Detection in Production of Glass Bottles**" Machine Graphics & Vision Inter. Journal, Vol. 5, pp. 77-86, 1996.

[Fir 97] C. Firmin, **"Optimisation des réseaux de neurones à fonctions radiales de base. Applications à la détection de défauts en production de bouteilles"**. Thèse de doctorat, Université des Sciences et Technologie de Lille, 7 Mars 1997.

[Fir 97] C. Firmin, D. Hamad, J.-G. Postaire and R. D. Zhang, "**Gaussian Neural Networks for Glass Bottles Production: a Learning Procedure**" International Journal of Neural Systems, Special issue on Neural Networks for Computer Vision Applications, Vol. 8, pp. 41-46, 1997.

[Fis 36] R.A. Fisher, **"The use of multiple measurements in taxonomic problems"**. Annual Eugenics, 7, Part II, pp.179-188, 1936.

[Föl 89] P. Földiak, **"Adaptive network for optimal linear feature extraction"**. International Joint Conference on Neural Networks, Vol. 1, pp. 401-405, Washington, 1989.

[Fri 74] J. H. Friedman and J.W. Tuckey, **"A projection pursuit algorithm for exploratory data analysis".** IEEE Transaction Computer, Vol. C-23, pp. 881-890, 1974.

[Fuk 82] K. Fukunaga and J.M. Mantock, **"A non parametric two-dimensional display for classification".** IEEE Transaction on Pattern Analysis and Machine Intelligence, PAMI-4, pp; 427-436, 1982.

[Fuk 90] K. Fukunaga, **"Introduction to statistical pattern recognition".** Second edition, San Diego: Academic press, 1990.

[Ham 95] D. Hamad and M. Betrouni, "**Artificial neural networks for non linear projection and exploratory data analysis**". International Conference on

Artificial Neural Networks and Genetic Algorithms, ICANNGA'95, pp. 164-167, Alès, FRANCE, April 18-21, 1995.

[Ham 96] D. Hamad, C. Firmin and J.-G. Postaire, **"Unsupervised pattern classification by neural networks".** Journal of Mathematics and Computers in Simulation, Elsevier Science, pp. 1-8, 1996.

[Hay 94] S. Haykin, "**Neural networks, a comprehensive foundation**". IEEE Computer Society Press, Editor: J. Griffin, 1994.

[Hay 96] S. Haykin, "**Neural networks expand SP's horizons. Advanced algorithms for signal processing simultaneously account for nonlinearity, nonstationarity and non-Gaussianity**". IEEE Signal Processing Magazine, pp. 24-49, March 1996.

[Jai 92] A.K. Jain and J. Mao, **"Artificial neural network for nonlinear projection of multivariate data".** In Proc. IEEE Int. Joint. Conf. On Neural Networks, Vol. 3, pp. 335-340, Baltimore, MARYLAND, June 1992.

[Kar 94] J. Karhunen, **"Optimisation criteria and nonlinear PCA neural networks".** IEEE Inter. conf. on Neural Networks (NN), Orlando, FLORIDA, USA, pp. 1241-1246, June 28-July 2, 1994.

[Kit 96] J. Kittler, M. Hatef and R.P.W. Duin, **"Combining classifier".** 13th International Conference on Pattern Recognition, Part B, pp. 897-901, Vienne, AUSTRIA, 1996.

[Koh 84] T. Kohonen, "**Self organisation and associative memory**". Berlin: Springer-Verlag, 1984.

[Koh 88] T. Kohonen T. **"Learning vector quantization".** Neural Networks, No. 1, 1988.

[Koh 95] T. Kohonen, J. Hynninen, J. Kangas , J. Laaksonen and K. Torkkola, "**LVQ_PAK: The Vector Quantization Program Package**" University of Technology, FINLAND, April 7 1995.

[Kra 91] M.A. Kramer, **"Nonlinear principal component analysis using autoassociative neural networks".** AIChE Journal, Vol. 37, pp. 233-243, February 1991.

[Kra 92] M.A. Kraaijveld, J. Mao, and A.K. Jain, **"A non-linear projection method based on Kohonen's topology preserving maps".** In Proc. of Int. Conf. on Pattern Recognition, Vol. 2, pp. 41-45, The Hagues, NETHERLANDS 1992.

[Kru 64] J. B. Kruskal **"Nonmetric multidimensional scaling: a numerical method".** Psychometrika, Vol. 29-2, pp. 115-129,1964.

[Lee 79] R.C.T. Lee, J.R. Slagle and H. Blum, **"A triangulation method for sequential mapping of points from N-space to two-space".** IEEE Transaction on Computer, Vol. 26, pp. 288-292, 1979.

[Leo 91] J. Leonard and M. A. Kramer, "**Radial Basis Function Networks for Classifying Fault**," IEEE Control Systems, Vol. 11, pp. 31-38, 1991.

[Mak 97] Y. Maki and A. K. Loparo, "**A neural network approach to fault detection and diagnosis in Industrial processes**", IEEE Transaction on Controls Systems technology, Vol. 5, No.6, pp. 529-541, 1997.

[Mao 94] J. Mao, "**Design and analysis networks for pattern recognition**". PhD, Michigan State University, 1994.

[Moo 88] T.J. Moody and C.J. Darken, "**Learning with localized receptive fields**". Proceedings of the 1988 Connectionist Models Summer School, eds. Touretzky, Hinton and Sejnowski, Morgan-Kaufmann, Publishers, 1988.

[Moo 89] J. Moody and C. J. Darken, "**Fast Learning in Networks of Locally-Tuned Processing Units**," Neural Computation Vol. 1, pp. 281-294, 1989.

[Oja 89] E. Oja, "**Neural networks, principal components and subspaces**". International Journal of Neural Systems, Vol. 1, pp. 61-68, 1989.

[Par 94] A. G. Parlos, J. Mathusami and A. F. Atiya, "**Incipient fault detection and identification in process systems using accelerated neural network learning**". Nuclear technology, Vol. 105, pp. 145-159, February 1994.

[Rum 86] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "**Learning Representations by Back-Propagating Errors**". Nature, Vol. 332, pp. 533-536, 1986.

[Rum 86] D. Rumelhart, J.L. McClelland and the PDP group, "**Parallel distributed processing**". Cambridge, MA: MIT press, Vol. 1, pp. 323-352, 1986.

[Sam 69] J.W. Sammon, "**A non linear mapping for data structure analysis**". IEEE Transactions on computers, Vol. C-18, No. 5, pp. 401-409, 1969.

[San 89] T.D. Sanger, "**Optimal unsupervised learning in a single-layer feedforward neural network**". Neural Networks, Vol. 12, pp. 459-473, 1989.

[Seb 71] G.F.A. Seber, "**Multivariate observations**". J. Wiley New York, 1971.

[Sie 88a] W. Siedlecki, K. Siedlecka and J. Slansky, "**An overview of mapping techniques for exploratory analysis**". Pattern Recognition, Vol. 21, No. 5, pp. 411-429, 1988.

[Sie 88b] W. Siedlecki, K. Siedlecka and J. Slansky, "**Experiments on mapping techniques for exploratory pettern recognition**". Pattern Recognition, Vol. 21, pp. 431-438, 1988.

[Sor 91] T. Sorsa, H. N. Koivo and H. Koivisto, "**Neural networks in process fault diagnosis**". IEEE Transaction on Systems Man and Cybetnetics, Vol. 21, No 4, pp. 815- 825, 1991.

[Spe 90] D. Specht, "**Probabilistic neural networks**". Neural Networks, Vol. 3, pp. 109-118, 1990.

[Tip 96] M.E. Tipping, "**Topographic mapping and feed-forward neural networks**". PhD, The University of Aston in Birmingham, February, 1996.

[Ult 92] A. Ultsch, "**Self organizing neural networks for knowledge acquisition**". Proc ECAI, pp. 208-210, Wien, AUSTRIA, 1992.

[Wat 89] K. Watanabe, I. Matsuura, M. Abe, M. Kubota and D. M. Himmelblau, "**Incipient fault diagnosis of chemical processes via artificial neural networks**". AIChE Journal, Vol. 105, No.11, pp. 1803-1812, 1989.