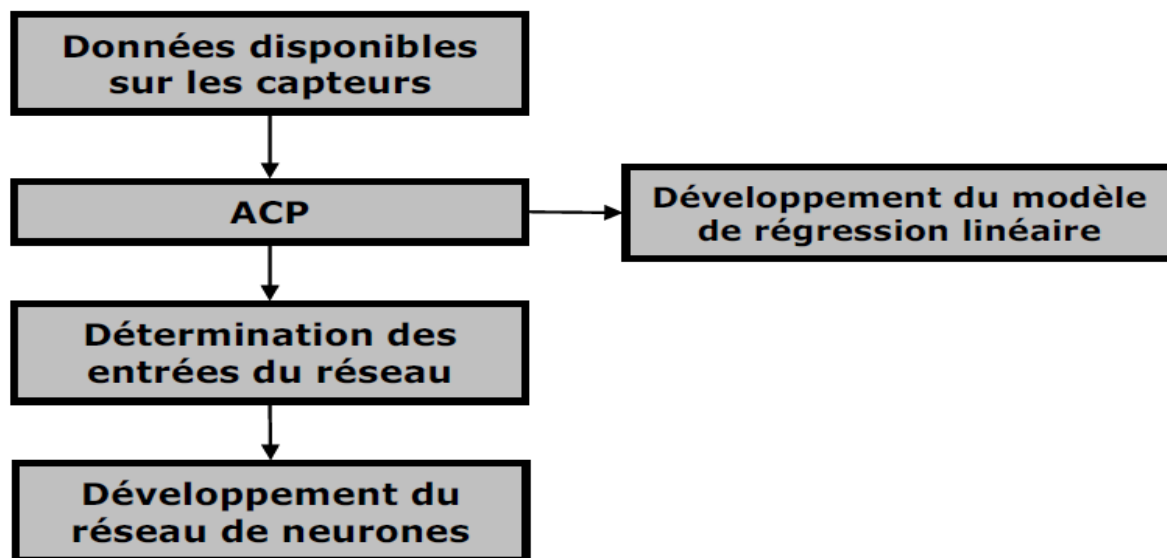


## 1. Introduction

Le développement d'un capteur logiciel a demandé le recours à l'élaboration d'un modèle de comportement du système à partir des données caractéristiques du béton tel que la résistance, etc. Ce chapitre donc consisté à développer un capteur logiciel basé sur la méthode d'analyse en composantes principales (ACP) et un réseau de neurones artificiel (RNA) permettant de prédire la force de résistance, sur la base des caractéristiques mesurées du béton. Nous présentons la méthodologie utilisée pour la construction du capteur logiciel à base de réseaux de neurones. Nous décrivons le modèle proprement dit ainsi que la méthode utilisée pour l'apprentissage et la sélection de l'architecture optimale du réseau et en particulier le recours à l'ACP (analyse en composantes principales) pour la détermination des entrées de ce capteur.

## 2. Description du capteur logiciel

La technique de mise en œuvre de capteur logiciel est basée sur deux opérations, la première faire une réduction de dimension de la base de données réelle qui déjà élaborée, donc on à acquérir une nouvelle base de donner de dimension réduire. Et la deuxième on à exploit le réseau de neurones artificiel pour faire l'apprentissage et les tests après la partage de la base de données nouvelle, et la **figure** présent les défèrent étapes :



**Figure III. 1** : Méthode pour le capteur logiciel basé sur l'ACP et les RNA.

### 3. Méthode d'analyse en composantes principales

La terminologie « analyse des données » désigne un sous-ensemble de ce qui est appelé plus généralement la statistique multivariée. L'analyse des données est un ensemble de techniques descriptives, dont l'outil mathématique majeur est l'algèbre matriciel, et qui s'exprime sans supposer a priori un modèle probabiliste. Certaines méthodes, pour la plupart géométriques, aident à faire ressortir les relations pouvant exister entre les différentes données et à en tirer une information statistique qui permette de décrire de façon plus succincte les principales informations contenues dans ces données. Pour cela, il est fondamental de mettre en place des outils de traitement de données permettant une meilleure compréhension de la valeur des connaissances disponibles dans ces données. La réduction des dimensions est l'une des plus vieilles approches permettant d'apporter des éléments de réponse à ce problème. Son objectif est de sélectionner ou d'extraire un sous-ensemble optimal de caractéristiques pertinentes pour un critère fixé auparavant. La sélection de ce sous-ensemble de caractéristiques permet d'éliminer les informations non-pertinentes et redondantes selon le critère utilisé. Cette sélection/extraction permet donc de réduire la dimension de l'espace des exemples et rendre l'ensemble des données plus représentatif du problème. En effet, les principaux objectifs de la réduction de dimension sont [22]:

- Faciliter la visualisation et la compréhension des données,
- Réduire l'espace de stockage nécessaire,
- Réduire le temps d'apprentissage et d'utilisation,
- Identifier les facteurs pertinents.

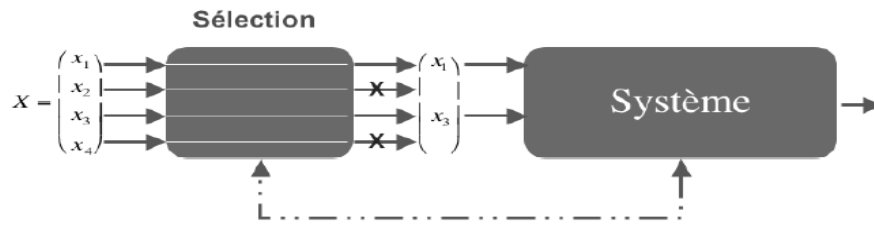
#### 3.1. Catégories de réduction des dimensions

Nous pouvons classer toutes les techniques mathématiques de réduction des dimensions en deux grandes catégories [23]:

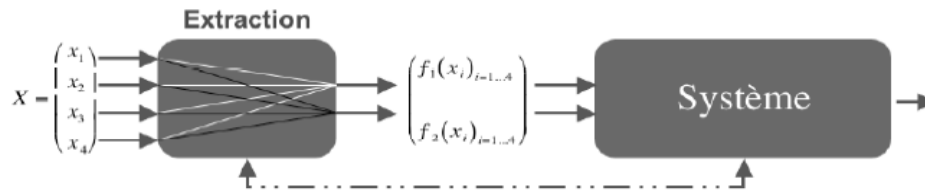
- **La sélection de variables:** qui consiste à choisir des caractéristiques dans l'espace de mesure,

(*Figure III.2*)

- **L'extraction de traits:** qui vise à sélectionner des caractéristiques dans un espace transformé dans un espace de projection (*Figure III.3*)



**Figure III.2 :** Principe de la sélection de variables.



**Figure III.3 :** Principe de l'extraction de caractéristiques.

### 3.2. Les différentes méthodes de réduction

Les méthodes de réduction de dimension peuvent être supervisées ou non supervisées :

	ACP	ACI	LDA	MMD
Type	non-supervisé	non-supervisé	supervisé	supervisé
Principe : recherche les axes...	... qui ont la plus grande variance	... qui sont les plus indépendants	... qui sont les plus discriminants	... qui sont les plus discriminants
Avantages	permet de bien représenter les données, supprime les redondances	recherche les "sources", supprime les redondances	méthode simple	plus rapide que l'infomax, pas d'hypothèse de gaussianité
Inconvénients	ne préserve pas les classes	ne préserve pas les classes	hypothèse de gaussianité, garde les redondances	garde les redondances

**Table III.1:** Tableau comparatif des principales caractéristiques de 4 techniques de réduction des dimensions

et :

ACP : Analyse en composantes principales (PCA : Principal Components Analysis);

ACI : Analyse en composantes indépendantes (ICA : Independent Component Analysis);

ADL : Analyse discriminante linéaire (LDA: Linear Discriminant Analysis) ;

DMM : Diversité Marginale Maximale (MMD).

- La méthode d'analyse en composantes principale sera détaillée, car la méthode utilisée dans notre application.

### **3.3. Analyse en composantes principales (ACP)**

#### **3.3.1. Présentation**

L'Analyse en Composantes Principales (ou ACP) est une méthode de représentation des données que l'on utilise en général sans référence à des hypothèses de nature statistique ni à un modèle particulier. Avoir besoin de l'ACP, c'est tout d'abord disposer d'un tableau rectangulaire de mesures, dont les colonnes correspondent à des variables numériques continues et dont les lignes représentent les individus sur lesquels ses variables sont mesurées. D'une manière générale, la seule condition que doit remplir ce tableau numérique pour être l'objet d'une analyse en composante principale est la suivante : l'une au moins des dimensions du tableau (les lignes en général) doit être formée d'unités ayant un caractère répétitif, l'autre pouvant éventuellement être plus hétérogène. [24]

#### **3.3.2. Historique**

L'ACP prend sa source dans un article de Karl Pearson publié en 1901. Le père du Test du  $\chi^2$  y prolonge ses travaux dans le domaine de la régression et des corrélations entre plusieurs variables. Pearson utilise ces corrélations non plus pour expliquer une variable à partir des autres (comme en régression), mais pour décrire et résumer l'information contenue dans ces variables.

Encore connue sous le nom de transformée de Karhunen-Loève ou de transformée de Hotelling, l'ACP a été de nouveau développée et formalisée dans les années 1930 par Harold Hotelling. La puissance mathématique de l'économiste et statisticien américain le conduira aussi à développer l'analyse canonique, généralisation des analyses factorielles dont fait partie l'ACP. [22]

### 3.3.3. Principe

L'ACP est une méthode d'analyse descriptive des données, non supervisée, qui consiste à rechercher les corrélations existant entre  $N$  variables aléatoires parmi une population de  $K$  individus en recherchant les directions (ou composantes) de l'espace qui contiennent la plus grande quantité d'informations, la plus grande quantité de variance. Ces directions sont les composantes principales (CP). L'ACP est donc une méthode statistique descriptive multivariée qui permet de représenter un grand nombre de données de façon synthétique. Elle permet de réduire les  $N$  variables en quelques composantes les plus représentatives. Cette méthode permet à la fois de représenter de manière synthétique les individus et les variables selon les composantes principales. Elle est utilisée pour repérer des groupes d'individus dont l'ensemble des variables est homogène et de visualiser des différences de variables entre les individus. Elle a aussi pour objectif de mettre en avant des individus au comportement atypique par rapport au reste de la population. [25]

Pour faire une ACP, on part d'une matrice  $M$  de données à  $K$  lignes (individus ou échantillons) et  $N$  colonnes (variables).

$$M = \begin{bmatrix} X_{1,1} & X_{2,1} & \dots & X_{N,1} \\ X_{1,2} & \dots & \dots & X_{N,2} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ X_{1,K} & X_{2,K} & \dots & X_{N,K} \end{bmatrix} \dots\dots\dots (III.1)$$

Chaque variable  $X_N = (X_{N,1}, \dots, X_{N,K})$  a une moyenne  $\bar{X}_N$  et un écart-type  $\sigma_{X_N}$ .

En fonction de l'origine et de la diversité des données à traiter, l'ACP peut être normée ou non normée. La normalisation consiste à transformer des données lorsque celles-ci sont d'origine différente afin de les ramener à des espaces comparables. Il est donc nécessaire de normaliser les données lorsqu'elles sont de mesure, de moyenne ou de variabilité différente. La normalisation consiste à transformer la matrice de variables brutes  $M$  en une matrice de variables centrées réduites  $\bar{M}$  pour laquelle les variables sont représentées par:

$$Y_N = (Y_{N,1}, \dots, Y_{N,k}) \text{ avec } Y_{N,1} = \frac{X_{N,1} - \bar{X}_{N,1}}{\sqrt{V(X_{N,1})}} \dots\dots\dots (III.2)$$

Avec:

- $X$  : valeur de la variable
- $\bar{X}$  : moyenne
- $V(X)$  : variance

La matrice est ensuite diagonalisée afin de trouver les valeurs propres, permettant de déterminer le nombre d'axes que l'on va conserver. A l'aide des valeurs propres, les vecteurs propres vont être calculés par un procédé d'algèbre linéaire consistant à rechercher une base de vecteurs propres de l'espace vectoriel. On conserve généralement les composantes principales qui contiennent 90% de la variance. Souvent, 3 composantes principales suffisent à décrire 90% de l'information du système. Enfin, on calcule les coordonnées des variables dans l'espace conservé puis celles des individus. On obtient ainsi deux représentations graphiques: une représentation pour laquelle un point représente la projection d'une variable dans un plan défini par les composantes principales («loading plot») et une représentation pour laquelle un point représente la projection d'un individu dans un plan défini par les composantes principales («score plot»).

### 3.3.4. Calcul des covariances et des corrélations

Le calcul des matrices de covariances et de corrélations est classique, du moment que les données sont déjà structurées en matrices et éventuellement normalisées ou réduites, il faut juste multiplier celle-ci par sa transposée :

- La matrice de variance-covariance des  $X_1, \dots, X_N$  si «  $M$  » est juste normalisée :

$$\text{Covariance} = \frac{1}{k} * \bar{M}' * \bar{M} \dots\dots\dots (III.3)$$

- La matrice de corrélation des  $X_1, \dots, X_N$  si «  $M$  » est réduite :

$$\text{Corrélation} = \frac{1}{k} * \bar{M} * \bar{M}' \dots\dots\dots (III.4)$$

Ces deux matrices sont carrées (de taille  $N \times N$ ), symétriques et réelles. Elles sont donc diagonalisables dans une base orthonormée [26].

### 3.3.5. Projection

Comme cité ci-dessus, l'analyse en composantes principales vise à projeter les données d'un espace initial vers un second espace, de dimension inférieure, tel que son premier axe «  $u$  » soit issu d'une combinaison linéaire des  $X_n$ , de manière à ce que la variance de la projection des points du nuage sur cet axe soit maximale. La projection de l'échantillon des  $X$  sur  $u$  «  $\pi$  » s'écrit :

$$\pi_u(M) = M * u \dots\dots\dots (III.5)$$

La variance de cette projection «  $\pi_u(M)$  » vaudra donc :

$$\pi_u(M)' * \frac{1}{k} * \pi_u(M) = u' * M' * \frac{1}{k} * M * u \dots\dots\dots (III.6)$$

$$C = M' * \frac{1}{k} * M$$

Et  $C$  la matrice de covariance.

Ayant vu précédemment que «  $C$  » était diagonalisable dans une base orthonormée, c'est cette diagonalisation qui va nous permettre en outre de voir que la variance exprimée par le  $K^{eme}$  vecteur propre valait  $\lambda_k$ . Et finalement, que la question de PCA nous ramène à un problème de diagonalisation de la matrice de corrélation. On notera «  $P$  » la matrice de changement de base associée et  $\Delta$  la matrice diagonale formée de son spectre :

$$\pi_u(M)' * \frac{1}{k} * \pi_u(M) = u' * P' * \Delta * P * u = (P * u)' \Delta (P * u) \dots\dots\dots (III.7)$$

$$\text{Et : } (P * u) = v$$

A partir de cette reformulation, on cherchera désormais le vecteur «  $v$  » maximisant  $v' * \Delta * v$  où  $\Delta = \text{Diag}(\lambda_1, \dots, \lambda_N)$  est une matrice diagonale dont les valeurs sont rangées dans un ordre décroissant. Là, il sera très facile de constater que le premier vecteur unitaire vérifiera  $v' * \Delta * v = \lambda$ . Formellement et mathématiquement, on va utiliser les multiplicateurs de Lagrange «  $\alpha$  » pour prouver ce résultat, et c'en maximisant la variance des données projetées sur  $u$  sous la contrainte que  $u$  soit de norme 1:

$$L(u, \alpha) = u' * C * u - \alpha(u' * u - 1) \dots\dots\dots (III.8)$$

En résolvant cette équation de Lagrange, deux résultats nous importent : en premier lieu «  $u$  » est un vecteur propre de  $C$  associé à la valeur propre  $\lambda_1$  ; et en suite, il est de norme « 1 ». La valeur propre de la matrice de covariance  $C$ : «  $\lambda_1$  » étant la variance sur le premier axe de PCA.

On poursuit la recherche du deuxième axe de projection  $w$  sur le même principe en imposant qu'il soit orthogonal à  $u$ , ainsi de suite, jusqu'à l'obtention des « 1 » vecteurs propres recherchés [25].

### - Algorithme de l'ACP

- Calcul de  $\bar{M}$ .
- $A = \text{cov } \bar{M}'$ .
- Calcul des valeurs et vecteurs propres.
- Calcul de « 1 »-le nombre de composantes à garder.
- Calcul de la matrice caractéristique =  $[\text{eig}_1, \text{eig}_2, \dots, \text{eig}_l]$ .
- Les données réduites = (la matrice caractéristique)' \*  $\bar{M}'$ .
- (Les données originales - la moyenne)' = (la matrice caractéristique)<sup>-1</sup> \* les données réduites
- (Les données originales)' = (la matrice caractéristique)<sup>T</sup> \* les données réduites + la moyenne.

Donc :

L'ACP est majoritairement utilisée pour:

- décrire et visualiser des données ;
- les décorrélérer ; la nouvelle base est constituée d'axes qui ne sont pas corrélés entre eux ;
- les débruiter, en considérant que les axes que l'on décide d'oublier sont des axes *bruités*.

## 4. Les réseaux de neurones artificiels

Les réseaux de neurones connaissent depuis quelques années un succès croissant dans divers domaines des sciences de l'ingénieur, celui du génie électronique n'échappe pas à cette règle. Dans ce cadre, nous nous intéressons à l'application des réseaux de neurones reconnaître des forme

Dans un premier temps, nous rappellerons les définitions et notations de base relatives aux réseaux de neurones. Nous poursuivrons en exposant les types et la méthodologie d'apprentissage, et le perceptron multicouche.

#### 4.1. Définition

Un réseau de neurones peut être considéré comme un modèle mathématique de traitement réparti, composé de plusieurs éléments de calcul non linéaire (neurones), opérant en parallèle et connectés entre eux par des poids.

Les neurones artificiels sont souvent utilisés sous forme de réseaux qui diffèrent selon le type de connections entre les neurones, une cinquantaine de types peut être dénombrée. En guise d'exemples nous citons : le perceptron de Rosembat, les réseaux de Hopfield etc.....

Ces derniers sont les plus utilisés dans le domaine de la modélisation et de la commande des procédés. Ils sont constitués d'un nombre fini de neurones qui sont arrangés sous forme de couches. Les neurones de deux couches adjacentes sont interconnectés par des poids. L'information dans le réseau se propage d'une couche à l'autre, on dit qu'ils sont de type « feed-forward ». Nous distinguons trois types de couches :

**Couche d'entrée** : les neurones de cette couche reçoivent les valeurs d'entrée du réseau et les transmettent aux neurones cachés. Chaque neurone reçoit une valeur, il ne fait pas donc de sommation.

**Couches cachées** : chaque neurone de cette couche reçoit l'information de plusieurs couches précédentes, effectue la sommation pondérée par les poids, puis la transforme selon sa fonction d'activation qui est en général une fonction sigmoïde. Par la suite, il envoie cette réponse aux neurones de la couche suivante.

**Couche de sortie** : elle joue le même rôle que les couches cachées, la seule différence entre ces deux types de couches est que la sortie des neurones de la couche de sortie n'est liée à aucun autre neurone. [27]

#### 4.2. Principe de fonctionnement

Le RNA peut donc se décomposer en trois ensembles distincts [28, 29] :

- l'ensemble E des neurones d'entrées qui reçoivent les données d'entrée sous la forme d'un vecteur de valeurs scalaires, le vecteur d'entrée représentant les paramètres du problème posé. Ces valeurs sont communiquées aux neurones par l'entremise de leurs valeurs d'entrée extérieure. Elles influencent donc l'activation de ces derniers, et par extension, le comportement du réseau entier.
- l'ensemble S des neurones de sortie dont les activations constituent le vecteur de sortie et sont collectivement interprétées comme le résultat du calcul du réseau. Un même neurone peut à la fois être à l'entrée et à la sortie du réseau. Donc E et S ne sont pas nécessairement disjoints.
- l'ensemble C des neurones cachés qui n'appartiennent pas aux ensembles E et S, mais les reliant, représente les corrélations encodées par le système. En général, la présence de neurones cachés dans un réseau démultiplie sa puissance de calcul, et lui permet d'aborder des problèmes plus complexes.

Un mécanisme de codage extérieur au réseau transforme les données en une séquence de vecteurs d'entrées. De même, un mécanisme de décodage récupère les vecteurs de sortie et les interprète pour produire des valeurs utilisables [28].

D'après la (Figure I.6), la sortie est donnée sous forme :

$$y(x, \theta) = \theta_0 + \sum_1^n \theta_i z_i \dots\dots\dots (III.9)$$

$$y(x, \theta) = \theta_0 + \sum_{i=1}^n \theta_i f \left( \theta_{i0} + \sum_{j=1}^n \theta_{ij} x_j \right) \dots\dots\dots (III.10)$$

Il est cependant à retenir que les neurones en amont ont une influence qui est fonction de leur propre activation et de la pondération du lien (poids synaptique  $\theta$ ) les reliant au neurone courant. L'influence totale de l'ensemble des neurones en amont se nomme activation pondérée [30].

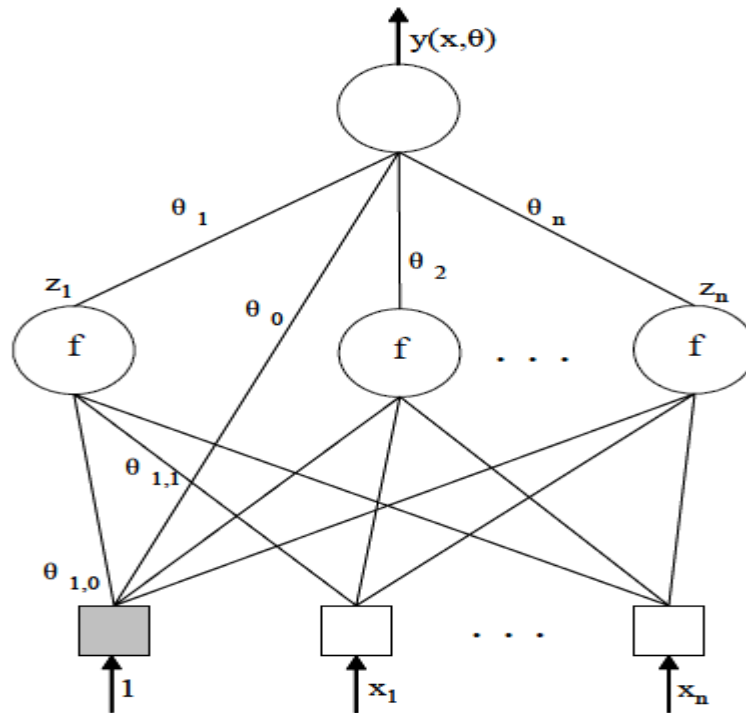


Figure III.4 : Structure générique d'un RNA.

### 4.3. Architecture des réseaux de neurones

On distingue deux structures de réseau, en fonction du graphe de leurs connexions, c'est-à-dire du graphe dont les nœuds sont les neurones et les arêtes les «connexions» entre ceux-ci :

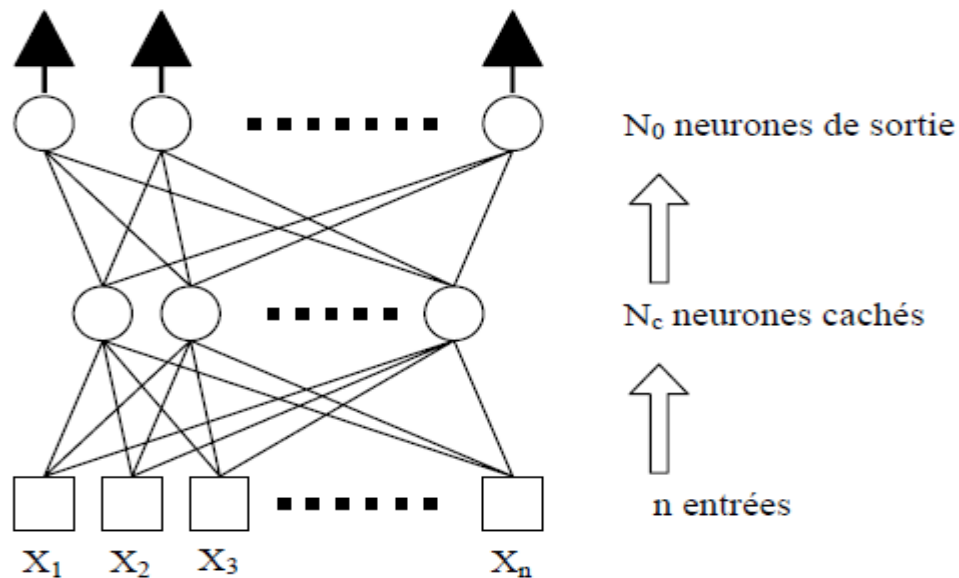
- Les réseaux de neurones statiques (ou acycliques, ou non bouclés).
- Les réseaux de neurones dynamiques (ou récurrents, ou bouclés).

#### 4.3.1. Les réseaux de neurones non bouclés

Un réseau de neurones non bouclé réalise une (ou plusieurs) fonction algébrique de ses entrées par composition des fonctions réalisées par chacun de ses neurones. Dans un tel réseau (Figure (I.8)), le flux d'information circule des entrées vers les sorties sans retour en arrière. Si l'on représente le réseau comme un graphe dont les nœuds sont les neurones et les arêtes les « connexions » entre ceux-ci, le graphe d'un réseau non bouclé est acyclique.

Tout neurone dont la sortie est une sortie du réseau est appelé « neurone de sortie ». Les autres, qui effectuent des calculs intermédiaires, sont des « neurones cachés ».

Il existe deux types de réseaux de neurones : les réseaux complètement connectés et les réseaux à couche. Le réseau de neurones à une couche cachée et une sortie linéaire est un cas particulier de ce dernier type.



**Figure III.5 :** Réseau de neurones à  $n$  entrées, une couche de  $N_c$  neurones cachés et  $N_0$  neurones de sortie.

#### ➤ Les réseaux de neurones complètement connectés

Dans un réseau complètement connecté, les entrées puis les neurones (cachés et de sortie) sont numérotés, et, pour chaque neurone :

- Ses entrées sont toutes les entrées du réseau ainsi que les sorties des neurones de numéro inférieur.
- Sa sortie est connectée aux entrées de tous les neurones de numéro supérieur.

#### ➤ Les réseaux de neurones à couches

Dans une architecture de réseaux à couches, les neurones cachés sont organisés en couches, les neurones d'une même couche n'étant pas connectés entre eux. De plus les connexions entre deux couches de neurones non consécutives sont éliminées.

Une telle architecture est historiquement très utilisée, surtout en raison de sa pertinence en classification.

➤ **Remarque**

Dans un réseau de neurones non bouclé, le temps ne joue aucun rôle fonctionnel : si les entrées sont constantes, les sorties le sont également. Le temps nécessaire pour le calcul de la fonction réalisée par chaque neurone est négligeable et on peut considérer ce calcul comme instantané.

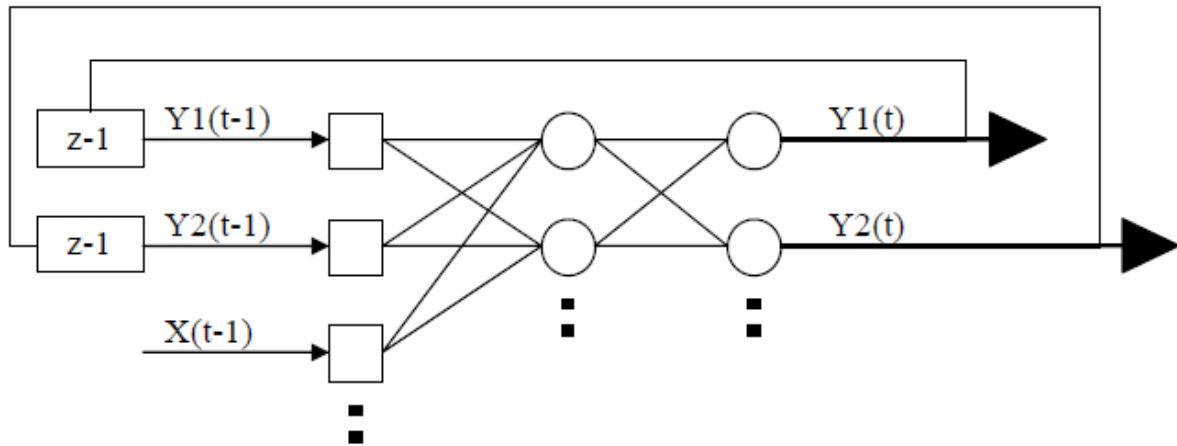
Pour cette raison, les réseaux non bouclés sont souvent appelés « réseaux statiques », par opposition aux réseaux bouclés ou « dynamiques ». Ils sont utilisés en classification, reconnaissance des formes (caractères, parole, ...), en prédiction [28].

### **4.3.2. Les réseaux de neurones bouclés**

L'architecture la plus générale pour un réseau de neurones est le « réseau bouclé », dont le graphe des connexions est cyclique : lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de « cycle »). La sortie d'un neurone du réseau peut donc être fonction d'elle même; cela n'est évidemment concevable que si la notion de temps est explicitement prise en considération [28].

Ainsi, à chaque connexion d'un réseau de neurones bouclé (ou à chaque arête de son graphe) est attaché, outre un poids comme pour les réseaux non bouclés, un retard, multiple entier (éventuellement nul) de l'unité de temps choisie. Une grandeur, à un instant donné, ne pouvant pas être fonction de sa propre valeur au même instant, tout cycle du graphe du réseau doit avoir un retard non nul.

Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locales. Pour éliminer le problème de la détermination de l'état du réseau par bouclage, on introduit sur chaque connexion « en retour » un retard qui permet de conserver le mode de fonctionnement séquentiel du réseau *Figure (III.6)*.



**Figure III.6 :** Réseau de neurone bouclé.

Le graphe des connexions de réseaux récurrents est cyclique. Ces réseaux sont décrits par un système d'équations aux différences.

➤ **Forme canonique des réseaux récurrents**

Il a été montré que tout réseau bouclé peut être mis sous une forme particulière, appelée forme canonique, qui est la représentation d'état minimale de la fonction réalisée par ce réseau. Cette forme canonique est constituée d'un graphe acyclique, et de connexions à retard unité reliant certaines sorties de ce graphe à ses entrées. La fonction réalisée par un réseau de neurones ayant cette structure particulière est décrite par les équations aux différences suivantes [32] :

$$x(k+1) = \Phi(x(k), u(k+1)) \quad \text{..... (III.11)}$$

$$y(k+1) = \Psi(x(k+1), u(k+1)) \quad \text{..... (III.12)}$$

Où  $x(k)$  est le vecteur d'état à l'instant  $k$ ,  $u(k)$  est le vecteur des variables de commande exogènes,  $y(k)$  le vecteur des sorties,  $\Psi$  et  $\Phi$  sont deux fonctions qui dépendent de la structure de la partie acyclique du réseau [31].

#### 4.4. Conception d'un réseau de neurones

Les réseaux de neurones réalisent des fonctions non linéaires paramétrées. Leurs mises en œuvre nécessitent [27] :

- La détermination des entrées et des sorties pertinentes, c'est à dire les grandeurs qui ont une influence significative sur le phénomène que l'on cherche à modéliser.
- La collecte des données nécessaires à l'apprentissage et à l'évaluation des performances du réseau de neurones.
- La détermination du nombre de neurones cachés nécessaires pour obtenir une approximation satisfaisante.
- La réalisation de l'apprentissage.
- L'évaluation des performances du réseau de neurones à l'issue de l'apprentissage.

#### **4.4.1. Détermination des entrées/sorties du réseau de neurones**

Pour toute conception de modèle, la sélection des entrées doit prendre en compte deux points essentiels [27] :

- Premièrement, la dimension intrinsèque du vecteur des entrées doit être aussi petite que possible, en d'autre terme, la représentation des entrées doit être la plus compacte possible, tout en conservant pour l'essentiel la même quantité d'information, et en gardant à l'esprit que les différentes entrées doivent être indépendantes.
- En second lieu, toutes les informations présentées dans les entrées doivent être pertinentes pour la grandeur que l'on cherche à modéliser : elles doivent donc avoir une influence réelle sur la valeur de la sortie.

#### **4.4.2. Choix et préparation des échantillons**

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix et la préparation des échantillons de données. La façon dont se présente l'échantillon conditionne le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation. Il faut donc déterminer les grandeurs qui ont une influence significative sur le phénomène que l'on cherche à modéliser [27].

#### **4.4.3. Elaboration de la structure du réseau**

La structure du réseau dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : un perceptron standard, un réseau de Hopfield, un réseau à décalage temporel (TDNN), un réseau de Kohonen, un ARTMAP etc...

Par exemple, dans le cas du perceptron multicouches, il faudra aussi bien choisir le nombre de couches cachées que le nombre de neurones dans cette couche [27].

#### **4.4.4. Apprentissage**

Une caractéristique des réseaux de neurones est leur capacité à apprendre (par exemple à reconnaître une lettre, un son...). Mais cette connaissance n'est pas acquise dès le départ. La plupart des réseaux de neurones apprennent par l'exemple.

On appelle « phase d'apprentissage » des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau. Afin que celui-ci remplisse au mieux la tâche qui lui est affectée. La phase de l'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.

##### **a. Types d'apprentissage**

Selon le critère d'apprentissage les réseaux de neurones se subdivisent en deux classes, les réseaux à apprentissage supervisé (Supervised learning) et les réseaux à apprentissage non supervisé (Unsupervised learning).

- Pour les réseaux à apprentissage supervisé, on présente au réseau des entrées et au même temps les sorties que l'on désirerait pour cette entrée. Le réseau doit alors se reconfigurer. C'est-à-dire calculer ses poids afin que la sortie qu'il donne corresponde bien à la sortie désirée.

- Pour les réseaux à apprentissage non supervisé, on présente une entrée au réseau et on le laisse évoluer librement jusqu'à ce qu'il se stabilise.

Il existe aussi des réseaux à apprentissage dit semi supervisé (reinforcement learning) qui ne tiennent compte que d'une évaluation partielle ou qualitative des sorties.

##### **b. Application des réseaux de neurones selon le mode d'apprentissage**

Les réseaux à apprentissage supervisé servent à :

- La modélisation statique des processus non linéaires décrits par des équations algébriques.

- La modélisation dynamique des processus décrits par des équations récurrentes.
- La commande de processus. Ils servent aussi à imposer une dynamique de réponse à une sollicitation.

Les réseaux à apprentissage non supervisé permettent l'analyse, la classification et la visualisation des données. [32]

#### **4.4.5. Validation et Tests**

Alors que les tests concernent la vérification des performances d'un réseau de neurones hors échantillon et sa capacité de généralisation, la validation est parfois utilisée lors de l'apprentissage. Une fois le réseau de neurones développé, des tests s'imposent afin de vérifier la qualité des prévisions du modèle neuronal.

Cette dernière étape doit permettre d'estimer la qualité du réseau obtenu en lui présentant des exemples qui ne font pas partie de l'ensemble d'apprentissage. Une validation rigoureuse du modèle développé se traduit par une proportion importante de prédictions exactes sur l'ensemble de la validation.

Si les performances du réseau ne sont pas satisfaisantes, il faudra, soit modifier l'architecture du réseau, soit modifier la base d'apprentissage [27].

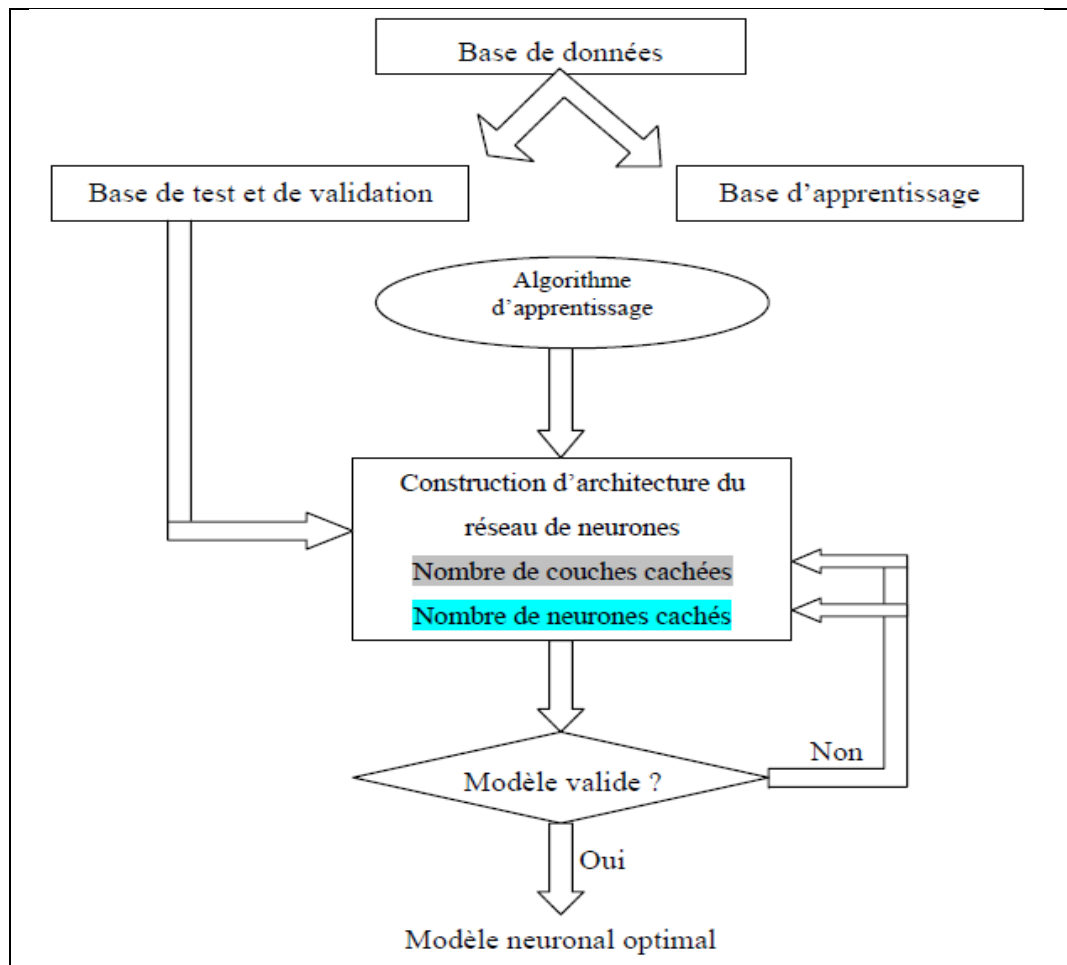


Figure III.7: Organigramme de conception d'un réseau de neurones.

## 4.5. Les réseaux de neurones les plus utilisés

### 4.5.1. Les Perceptrons

#### a. Le perceptron monocouche (Perceptron simple)

Le perceptron est un réseau de neurones très simple, puisqu'il ne se compose que d'une couche d'entrée et d'une couche de sortie. Il peut aussi être utilisé pour faire de la classification et pour résoudre des opérations logiques simples, il suit généralement un apprentissage supervisé selon la règle de correction de l'erreur [33].

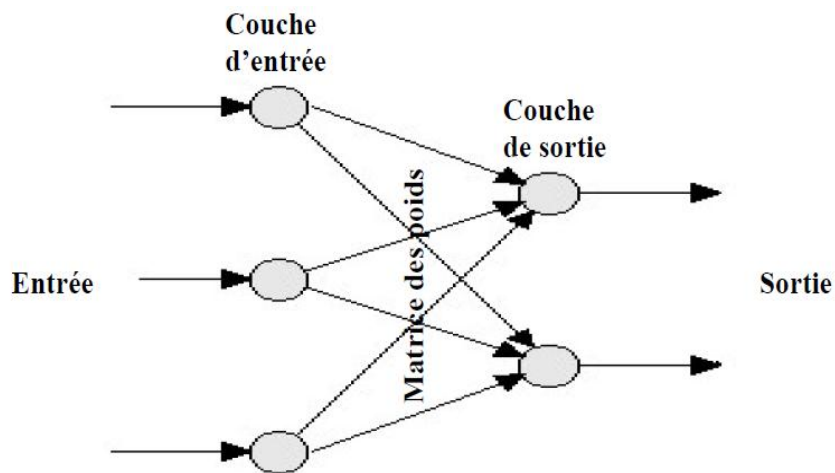


Figure III.8 : Architecture d'un perceptron

### b. Le perceptron multicouche (en anglais Multi Layer Perceptron)

C'est une extension du précédent, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante. Il peut résoudre des problèmes non linéairement séparables et des problèmes logiques plus compliqués. Il suit un apprentissage supervisé selon la règle de correction de l'erreur, ce réseau sera détaillé par la suite [34].

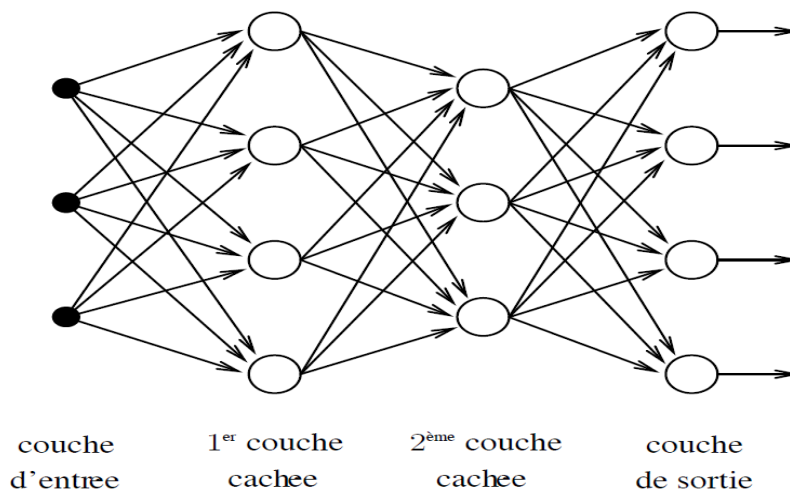


Figure III.9 : Exemple d'un réseau de type perceptron multicouche.

### 4.5.2. Les réseaux à fonction radiale

Les réseaux à fonctions de bases radiales (RBF) ou plus simplement réseaux à bases radiales ont été proposés par J. Moody et C. Darken. Les RBF seront donc employés dans les

mêmes types de problèmes que les PMC à savoir, en classification et en approximation de fonctions, particulièrement. L'apprentissage le plus utilisé pour les RBF est le mode hybride et les règles sont soit la règle de correction de l'erreur soit la règle d'apprentissage par compétition [35].

#### **4.5.3. Les cartes auto-organisatrices de Kohonen (SOM)**

Les cartes topologiques ou cartes auto organisatrices ont été introduites pour la première fois par T. Kohonen en 1982. Ce sont des réseaux à apprentissage non-supervisé qui établissent une carte discrète, ordonnée topologiquement, en fonction de patterns d'entrée. Le réseau forme ainsi une sorte de treillis dont chaque nœud est un neurone associé à un vecteur de poids. La correspondance entre chaque vecteur de poids est calculée pour chaque entrée. Par la suite, le vecteur de poids ayant la meilleure corrélation, ainsi que certains de ses voisins, vont être modifiés afin d'augmenter encore cette corrélation, ce réseau sera détaillé par la suite [38].

#### **4.5.4. Les réseaux de Hopfield**

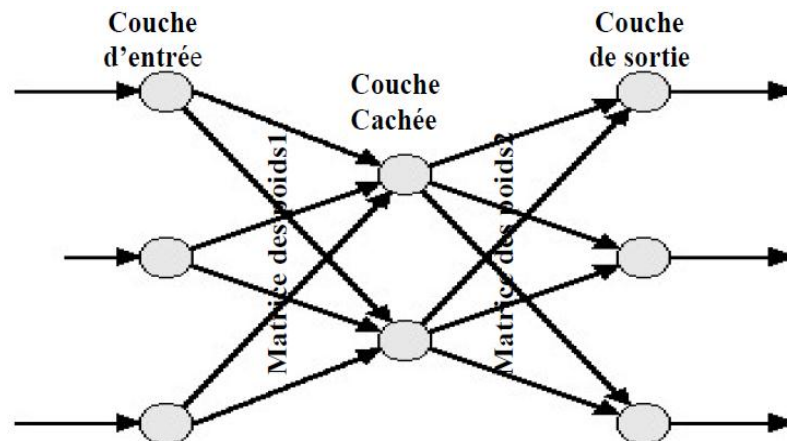
Il s'agit d'un réseau constitué de neurones à deux états (-1 et 1, ou 0 et 1), dont la loi d'apprentissage est la règle de Hebb (1949), qui veut qu'une synapse améliore son activité si et seulement si l'activité de ses deux neurones est corrélée (c'est-à-dire que le poids d'une connexion entre deux neurones augmente quand les deux neurones sont activés au même temps) [36].

### **4.6. Le perceptron multicouche**

Les réseaux multicouches sont actuellement les plus employés. Plusieurs couches de traitement leur permettent de réaliser des associations non linéaires entre l'entrée et la sortie.

Le perceptron à niveaux multiples (perceptron multi couche 'PMC' ou 'MLP' en anglais) est une extension du perceptron qui a une ou plusieurs sous-couches cachées entre ses couches d'entrée et de sortie. A cause de sa structure étendue, un perceptron à niveaux multiples est capable de résoudre toute opération logique incluant le problème XOR. La méthode d'apprentissage est supervisée. L'algorithme d'apprentissage est la règle d'apprentissage delta ( $\Delta$ ) qui affecte le changement des poids en multipliant l'entrée d'un

neurone par la différence entre la sortie réelle et la sortie désirée et le taux d'apprentissage du réseau [32].



*Figure III.10 : Architecture du perceptron Multicouche*

Le perceptron a les caractéristiques suivantes :

- Le perceptron multicouche est une suite de couches contenant chacune des neurones formels.
- Les entrées de la couche  $n$  sont les sorties de la couche  $n-1$ .
- Les neurones de la première couche sont reliés à l'extérieur et reçoivent tous la même entrée (vectorielle).
- La sortie du perceptron multicouches est un vecteur constitué des sorties des neurones de la dernière couche.

#### 4.6.1. Apprentissage du perceptron multicouche

De la même manière que le perceptron monocouche, le perceptron multicouche est lui aussi capable d'apprentissage. En effet, il existe également un algorithme permettant de corriger les poids vis à vis d'un ensemble d'exemples donnés. Cet algorithme est appelé

##### a. Algorithme de rétropropagation du gradient

Cet algorithme utilise la même règle de modification des poids ("delta rule") que l'algorithme de Widrow-Hoff. L'algorithme va être donné dans sa version la plus générale, c'est à dire avec plusieurs couches cachées. On notera  $g$  la fonction d'activation. Une démonstration de l'efficacité de cet algorithme exige de la fonction d'activation qu'elle soit indéfiniment dérivable. On notera  $s_i$  la sortie du neurone  $i$  de la couche de sortie et  $y_i$  la sortie attendue pour ce même neurone. Enfin, pour des neurones d'une couche cachée, on notera  $o_i$  la sortie calculée du neurone.

```

Learn(Rétine X[],Sortie Attendue A[])
{
    //Calcul de la sortie
    S=Out(X);
    //Calcul des erreurs Dab (a étant la couche et b le neurone) par rétropropagation
    Pour tout Neurone i de la dernière couche j faire
        Dji=Si*(1-Si)*(A[i]-Si);
    Fin Pour
    Pour chaque couche i de l'avant dernière couche à la première faire
        //Si = Sortie de la couche i
        Pour chaque neurone j de la couche i faire
            Dij=0;
            //Sij=Sortie du neurone j de la couche i
            Pour chaque neurone k de la couche l=i+1 faire
                //Wlkj = Poids du neurone k associé à l'entrée en provenance du neurone j
                Dij+=Dlk*Wlkj;
            Fin Pour
            Dij*=Sij*(1-Sij);
            Fin Pour
        Fin Pour
        //Mise à jour des poids
        Pour tout poids Wijk (poids du neurone j de la couche i associé à l'entrée provenant du
            neurone k de la
            couche i-1) faire
            //Sk = Sortie du neurone k
            Wijk+=Dij*Sk
        Fin Pour
    }
}

```

La variable "di" apparaît deux fois dans le code. Il s'agit de deux variables différentes, car en fait on suppose que les neurones sont numérotés de sorte que l'on puisse associer à un identifiant un neurone et réciproquement. Par conséquent, le 'i' de "di" identifie un neurone et ainsi on peut effectuer la dernière boucle de manière uniforme sans différencier pour la couche de sortie et les couches cachées [32], [37].

### **Voici quelques remarques sur cet algorithme**

- L'algorithme de rétropropagation du gradient est une extension de l'algorithme de Widrow-Hoff. En effet, dans les deux cas, les poids sont mis à jour à chaque présentation d'exemple et donc on tend à minimiser l'erreur calculée pour chaque exemple et pas l'erreur globale.
- La méthode donne de bons résultats pratiques. Dans la plupart des cas, on rencontre peu de problèmes dus aux minima locaux, mais il y en a. Toutefois, il est moins performant que d'autres algorithmes de propagation d'erreur : il tend moins rapidement vers des poids plus ou moins optimaux.

- Il n'y a pas de condition d'arrêt pour le REPETER. C'est à vous de fixer le critère. On peut par exemple répéter cela jusqu'à ce que l'erreur sur chaque exemple descende en dessous d'un certain nombre.
- Le choix de l'architecture initiale du réseau reste un problème difficile. Ce choix peut être fait par l'expérience. Des méthodes dites "auto-constructives" existent : il s'agit d'ajouter des neurones au cours de l'apprentissage pour que l'apprentissage se fasse bien.

#### 4.6.2. Considérations pratiques pour le choix du RNA

Avant de procéder à l'apprentissage d'un réseau de neurones artificiels, un certain nombre de paramètres doivent être fixés (dimensionnement du réseau, pas d'apprentissage..., etc.), au risque d'engendrer et de compromettre le processus d'entraînement dans le cas d'un mauvais choix [38].

##### - Choix et dimensionnement d'un réseau

D'une manière générale, il n'existe pas de méthodes systématiques pour la détermination du nombre de couches dans le réseau ou le nombre de neurones par couche. Néanmoins, il existe dans ce sens des règles pratiques tirées à partir de l'expérience, parmi lesquelles nous pouvons citer:

- **Règle 1** : le nombre de neurones par couche doit être d'autant plus élevé que la relation entre les données en entrée et les sorties désirées est complexe.
- **Règle 2** : pour les réseaux à une seule couche cachée, le nombre de neurones dans cette couche doit être égal de 3 à 13
- **Règle 3** : le nombre de couches cachées doit être égal au nombre de relations qui existent entre les neurones de la couche d'entrée et ceux de la couche de sortie.

#### 4.6.3. Evaluation

Il y a par rapport au perceptron monocouche une étape supplémentaire lors de l'évaluation de la sortie du réseau de neurone. On obtient ainsi une matrice des poids pour passer des informations à la couche cachée, et une autre matrice de poids pour passer de la couche cachée à la couche de sortie. Ensuite, cela revient au même que pour le réseau monocouche.

## 5. Conclusion

Dans ce chapitre, nous avons parlé sur la réduction de dimension des données et leur objectif ainsi que les méthodes exploitées dans ce contexte ; on a insisté sur la méthode d'analyse en composantes principales qui est choisie dans notre approche. L'ACP est une méthode de traitement statistique de données dont le but est de représenter et d'expliquer les liaisons statistiques entre les phénomènes. Elle permet d'identifier des variables sous-jacentes, ou facteurs qui expliquent les corrélations à l'intérieur d'un ensemble de variables observées.

Nous avons rappelé les concepts les plus importants pour comprendre l'intérêt des réseaux de neurones comme outil de régression. Ces outils permettent de calculer des fonctions, adaptables à un ensemble d'exemples par le biais d'algorithmes d'optimisation utilisant la technique de la rétro-propagation.

Parmi les différents types de modèle de comportement possibles, le modèle à base de réseaux de neurones possède l'avantage de pouvoir intrinsèquement décrire des relations non-linéaires entre les variables d'entrées qui à obtenu par l'ACP et celles de sorties d'un système.