Classifieurs SVM et Réseaux de Neurones

Introduction

Les méthodes de classification ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains paramètres descriptifs. Elles s'appliquent à un grand nombre d'activités humaines et conviennent en particulier au problème de la prise de décision automatisée. La procédure de classification sera extraite automatiquement à partir d'un ensemble d'exemples. Un exemple consiste en la description d'un cas avec la classification correspondante. Un système d'apprentissage doit alors, à partir de cet ensemble d'exemples, extraire une procédure de classification, il s'agit en effet d'extraire une règle générale à partir des données observées. La procédure générée devra classifier correctement les exemples de l'échantillon et avoir un bon pouvoir prédictif pour classifier correctement de nouvelles descriptions.

Les méthodes utilisées pour la classification sont nombreuses, citons : la méthode des Séparateurs à Vastes Marges (SVM), les Réseaux de Neurones, etc. Nous présentons dans la suite de ce chapitre une étude détaillée des deux techniques SVM et réseaux de neurones. Ces méthodes ont montrés leurs efficacités dans de nombreux domaines d'applications tels que le traitement d'image, la catégorisation de textes et le diagnostique médicale.

II.1 Séparateurs à Vaste Marge (SVM)

II.1.1 Introduction

Les machines à vecteurs de support (Support Vector Machine, SVM) appelés aussi séparateurs à vaste marge sont des techniques d'apprentissage supervisées destinées à résoudre des problèmes de classification. Les machines à vecteurs supports exploitent les concepts relatifs à la théorie de l'apprentissage statistique et à la théorie des bornes de Vapnik et Chervonenkis [21]. La justification intuitive de cette méthode d'apprentissage est la suivante : si l'échantillon d'apprentissage est linéairement séparable, il semble naturel de séparer parfaitement les éléments des deux classes de telle sorte qu'ils soient le plus loin possibles de la frontière choisie. Ces fameuses

machines ont été inventées en 1992 par Boser et al. [22], mais leur dénomination par SVM n'est apparue qu'en 1995 avec Cortes et al. [23]. Depuis lors, de nombreux développements ont été réalisés pour proposer des variantes traitant le cas non-linéaire (voir [24] et [25]). Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent. Elles permettent d'aborder des problèmes très divers dont la classification. SVM est une méthode particulièrement bien adaptée pour traiter des données de très haute dimension.

II.1.2 Principe de la technique SVM

Cette technique est une méthode de classification à deux classes qui tente de séparer les exemples positifs des exemples négatifs dans l'ensemble des exemples. La méthode cherche alors l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Cela garantit une généralisation du principe car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être situés d'un coté ou l'autre de la frontière. L'intérêt de cette méthode est la sélection de vecteurs supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas, ce qui peut être considéré comme un avantage pour cette méthode.

II.1.3 Classifieur linéaire

Un classifieur est dit linéaire lorsqu'il est possible d'exprimer sa fonction de décision par une fonction linéaire en x. On peut exprimer une telle fonction par:

$$h(x) = \langle w, x \rangle + b = \sum_{i=1}^{n} w_i x_i + b \qquad (2.1)$$

où w ($\in \mathrm{R}^n$) est le vecteur de poids et b($\in \mathrm{R}^0$) le biais, alors que x est la variable du problème. X est l'espace d'entrée et qui correspond à R^n, où n est le nombre de composantes des vecteurs contenant les données. Notons que l'opérateur $\langle \quad \rangle$ désigne le produit scalaire usuel dans R^n. w et b sont les paramètres à estimer de la fonction de décision $h(x)$.

Pour décider à quelle catégorie un exemple estimé x' appartient, il suffit de prendre le signe de la fonction de décision : $y = sign(h(x'))$. La fonction $sign(\,)$ est appelée

classifieur. Géométriquement (voir figure II.1), cela revient à considérer un hyperplan qui est le lieu des points x satisfaisant $\langle w, x\rangle + b = 0$. En orientant l'hyperplan, la règle de décision correspond à observer de quel côté de l'hyperplan se trouve l'exemple x'. On voit que le vecteur w définit la pente de l'hyperplan (w est perpendiculaire à l'hyperplan). Le terme b quant à lui permet de translater l'hyperplan parallèlement à lui-même.

L'objectif de la discrimination linéaire est de trouver la bonne fonction de décision $h(x)$. La classe de tous les hyperplans qui en découle sera notée H.

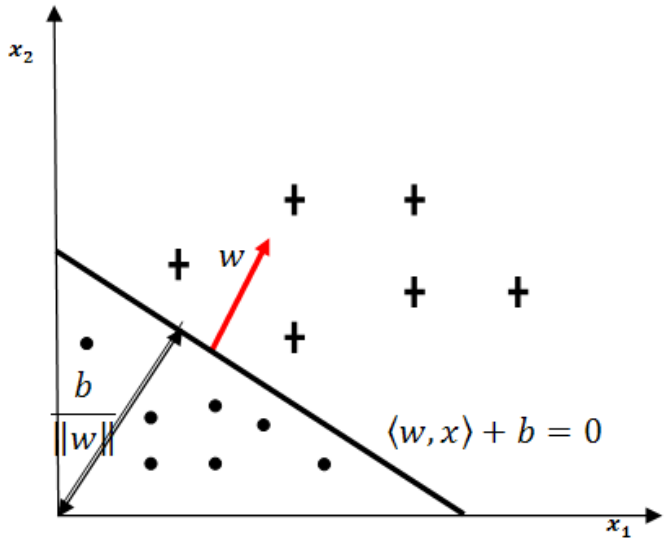

Figure II.1 : hyperplan séparateur $\langle w, x\rangle + b = 0$

II.1.4 Marge maximale de l'hyperplan

La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Dans les SVM, la frontière de séparation est choisie comme celle qui maximise la marge.

La marge géométrique représente la distance euclidienne prise perpendiculairement entre l'hyperplan et l'exemple x_i. En prenant un point quelconque x_p se trouvant sur l'hyperplan, la marge géométrique peut s'exprimer par :

$$\frac{w}{\|w\|}.(x_i - x_p) \qquad (2.2)$$

L'hyperplan à marge maximale est le modèle le plus utilisé dans les machines à vecteurs supports. L'estimation des paramètres $(w^*,\ b^*)$ de l'hyperplan qui maximise la marge se fait en résolvant le problème d'optimisation suivant :

$$(w^*, b^*) = \arg max_{(w,b)}\{\min_i(y_i(wx_i + b)), \|w\| = 1\} \quad (2.3)$$

Dire que les deux classes de l'échantillon d'apprentissage S sont linéairement séparables est équivalent à dire qu'il existe des paramètres (w^*, b^*) tels que l'on a pour tout $i\ (= 1,2, \dots, n)$:

$$w^*x_i + b^* > 0\ si\ y_i = 1 \quad (2.4)$$

$$w^*x_i + b^* < 0\ si\ y_i = -1 \quad (2.5)$$

ce qui est équivalent à :

$$y_i(w^*x_i + b^*) > 0\ ;\quad \forall\ i = 1,2, \dots, n \quad (2.6)$$

La définition consiste à dire qu'il doit exister un hyperplan laissant d'un coté toutes les données positives et de l'autre, toutes les données négatives. Dès lors, on peut définir deux plans se trouvant de part et d'autre de l'hyperplan et parallèles à celui-ci, sur lesquels reposent les exemples les plus proches. La figure II.2 illustre cette situation.

Dans notre définition de l'hyperplan, il est possible que différentes équations correspondent au même plan géométrique :

$$a(\langle w, x\rangle + b) = 0 \quad (2.7)$$

a est une constante quelconque.

Il est donc possible de redimensionner (w^*, b^*) de telle sorte que les deux plans parallèles aient respectivement pour équations :

$$(w^*x_i + b^*) = 1 \quad (2.8)$$

$$(w^*x_i + b^*) = -1 \quad (2.9)$$

Ces deux hyperplans sont appelés hyperplans canoniques. Ainsi la marge γ entre ces deux plans est égale à :

$$\gamma = \frac{2}{\|w^*\|} \quad (2.10)$$

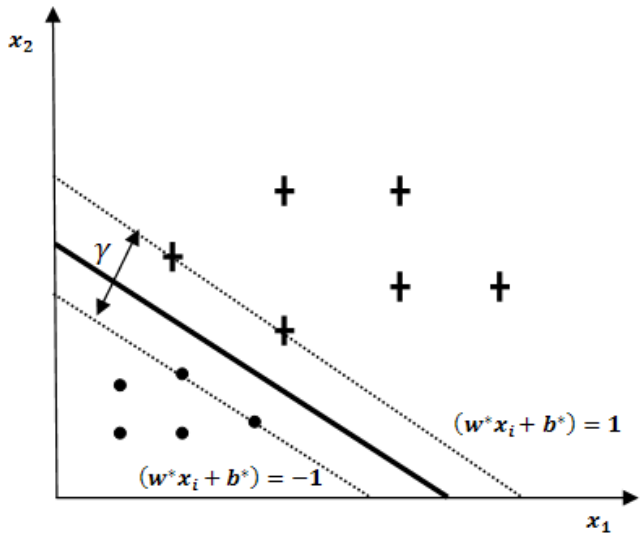

Figure II.2 : hyperplans canoniques et marge maximale

II.1.5 Minimisation quadratique sous contraintes

Maintenant que nous avons défini les notions de marges et d'hyperplans canoniques, nous pouvons formuler un problème d'optimisation mathématique tel que sa solution nous fournisse l'hyperplan optimal qui permet de maximiser la marge [23] :

$$Minimiser \ \frac{1}{2}\|w\|^2 \tag{2.11}$$

$$Tel \ que \ y_i(\langle w, x_i\rangle + b) \geq 1 \tag{2.12}$$

Il s'agit d'un problème quadratique convexe sous contraintes linéaires de forme primal dont la fonction objective est à minimiser. Cette fonction objective est le carré de l'inverse de la double marge. L'unique contrainte stipule que les exemples doivent être bien classés et qu'ils ne dépassent pas les hyperplans canoniques.

Dans cette formulation, les variables à fixer sont les composantes w_i et b. Le vecteur w possède un nombre de composantes égal à la dimension de l'espace d'entrée. Généralement dans ce type de cas on résout la forme dual du problème. Nous devons former ce que l'on appelle le Lagrangien. Il s'agit de faire rentrer les contraintes dans la fonction objective et de pondérer chacune d'entre elles par une variable dual [25]:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i(y_i(\langle w, x_i\rangle + b) - 1) \tag{2.13}$$

Les variables duales α_i intervenant dans le Lagrangien sont appelées multiplicateurs de Lagrange. Notons que L doit être minimisé par rapport aux variables primales w_i et b et maximisé par rapport aux variables duales α_i.

Le point selle (minimal par rapport à une variable, maximal par rapport à l'autre) doit donc satisfaire les conditions nécessaires de stationnarité qui correspondent aux conditions Karush Kuhn et Tucker (KKT), nous trouvons:

$$\frac{\partial L(w,b,\alpha)}{\partial w} = 0 \tag{2.14}$$

$$\frac{\partial L(w,b,\alpha)}{\partial b} = 0 \tag{2.15}$$

Ce qui nous permet d'obtenir :

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{2.16}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.17}$$

Remarquons qu'avec cette formulation, on peut calculer w en fixant seulement n paramètres. L'idée va donc être de formuler un problème dual dans lequel w est remplacé par sa nouvelle formulation. De cette façon, le nombre de paramètres à fixer est relatif au nombre d'exemples de l'échantillon d'apprentissage et non plus à la dimension de l'espace d'entrée. Pour se faire, nous substituons (2.16) et (2.17) dans le Lagrangien L, nous obtenons le problème dual équivalent suivant :

$$Maximiser_{\alpha} \quad W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \tag{2.18}$$

$$tel\ que \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{2.19}$$

$$\alpha_i \geq 0 \tag{2.20}$$

Ce dernier problème peut être résolu en utilisant des méthodes standards de programmation quadratique [25]. Une fois la solution optimale $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)$ du problème (2.18) obtenue, le vecteur de poids de l'hyperplan à marge maximale recherché s'écrit :

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i \tag{2.21}$$

Comme le paramètre b ne figure pas dans le problème dual, sa valeur optimale b^* peut être dérivée à partir des contraintes primales, soit donc :

$$b^* = -\frac{max_{y_i=-1}(\langle w^*, x_i \rangle) + min_{y_i=1}(\langle w^*, x_i \rangle)}{2} \tag{2.22}$$

Une fois les paramètres α^* et b^* calculés, la règle de classification d'une nouvelle observation x basée sur l'hyperplan à marge maximale est donnée par :

$$h(x) = sign\left(\sum_{i=1}^{n} y_i \alpha_i^* \langle x_i, x \rangle + b^*\right) \tag{2.23}$$

Notons qu'un grand nombre de termes de cette somme est nul. En effet, seuls les α_i^* correspondant aux exemples se trouvant sur les hyperplans canoniques (sur la contrainte) sont non nuls. Ces exemples sont appelés Supports Vecteurs (SV). On peut les voir comme les représentants de leurs catégories car si l'échantillon d'apprentissage n'était constitué que des SV, l'hyperplan optimal que l'on trouverait serait identique.

II.1.6 SVM non-linéaires

Le paragraphe précédent décrit le principe des SVM dans le cas ou les données sont linéairement séparables. Cependant, dans la plupart des problèmes réels, ce n'est pas toujours le cas et il est donc nécessaire de contourner ce problème (difficile de séparer n'importe quel jeu de données par un simple hyperplan). Si par exemple les données des deux classes se chevauchent sévèrement, aucun hyperplan séparateur ne sera satisfaisant.

Dans ce but et selon [23], l'idée est de projeter les points d'apprentissage x_i dans un espace T de dimension q, plus élevée que n grâce à une fonction non-linéaire ϕ qu'on appelle fonction noyau, choisie a priori et d'appliquer la même méthode d'optimisation de la marge dans l'espace T. L'espace T ainsi obtenu est appelé espace des caractéristiques ou aussi espace transformé.

Tout ce qu'il nous reste à faire c'est de résoudre le problème (2.18) dans l'espace T, en remplaçant $\langle x_i, x_j \rangle$ par $\langle \phi(x_i), \phi(x_j) \rangle$. L'hyperplan séparateur obtenu dans l'espace T

est appelé hyperplan optimal généralisé. Le produit scalaire $\langle \phi(x_i), \phi(x_j) \rangle$ peut se calculer facilement à l'aide d'une fonction symétrique K, dite noyau, définie par :

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{2.24}$$

Le noyau $K(x, y)$ est une fonction de deux variables, symétrique et positive. Dans ce cas, la frontière de décision devient :

$$h(x) = sign\left(\sum_{i=1}^{n} y_i \alpha_i^* K(x_i, x_j) + b^*\right) \tag{2.25}$$

Dans la pratique on choisit un noyau K qui satisfait les conditions de Mercer [26] afin de garantir la décomposition (2.24). Une famille de ces fonctions noyaux qui sont très appropriées aux besoins des SVM peut être définie, en voici les plus utilisés [26]:

- ✓ ***Noyau polynomial d'ordre p :***

$$K(x_i, x_j) = (\langle x_i, x_j \rangle + 1)^d \tag{2.26}$$

La dimension de l'espace transformé induit par un noyau polynomial est de l'ordre $\frac{(p+d)!}{p!d!}$, où p est la dimension de l'espace de départ

- ✓ ***Noyau linéaire :***

$$K(x_i, x_j) = x_i . x_j \tag{2.27}$$

- ✓ ***Noyau gaussien de largeur de bande σ :***

$$K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma}\right) \tag{2.28}$$

Le paramètre σ permet de régler la largeur de la gaussienne. En prenant un σ grand, la similarité d'un exemple par rapport à ceux qui l'entourent sera assez élevée, alors qu'on prenant un σ tendant vers 0, l'exemple ne sera similaire à aucun autre.

II.1.7 Relaxation des contraintes

Quand le domaine du problème d'optimisation est vide et il n'admet donc pas de solution, dans ce cas les données sont non linéairement séparables. Pour tenter de résoudre ce problème, on relâche les contraintes (2.12) dans le but d'autoriser quelques erreurs de classification. Cette généralisation de l'hyperplan à marge maximale à été

proposée par Cortes et al. [22] en introduisant les variables d'écart à la marge $(\xi_i)_{1\leq i\leq n}$. L'hyperplan optimal est celui qui satisfait les conditions suivantes :

- ✓ La distance entre les vecteurs bien classés et l'hyperplan optimal doit être maximale.
- ✓ La distance entre les vecteurs mal classés et l'hyperplan optimal doit être minimale.

Le problème (2.11) devient alors :

$$Minimiser\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{2.29}$$

$$Tel\ que\ y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \qquad i = 1,2,\dots,n \tag{2.30}$$

$$\xi_i \geq 0, \qquad i = 1,2,\dots,n \tag{2.31}$$

Autrement dit, on cherche à maximiser la marge en s'autorisant pour chaque contrainte une erreur positive ξ_i, la plus petite possible. Le paramètre supplémentaire C est appelé paramètre de pénalisation du relâchement. C est une constante positive fixée à l'avance qui permet de contrôler l'importance de l'erreur que l'on s'autorise par rapport à la taille de la marge. Plus C est important, moins d'erreurs sont autorisées.

II.1.8 SVM pour le cas multiclasses

La plupart des problèmes ne se contentent pas de deux classes de données. Il existe plusieurs méthodes pour faire la classification multiclasses. Citons les plus utilisées : La première méthode, celle que nous utilisons dans notre application, est une méthode dite Un-contre-Un. Au lieu d'apprendre N fonctions de décisions, ici chaque classe est discriminée d'une autre.

La deuxième méthode est appelé Un-Contre-Tous. C'est une approche étendant la notion de marge aux cas multiclasses. Cette formulation intéressante permet de poser un problème d'optimisation unique. Le problème fait intervenir N fonctions de décision

II.1.9 Avantages et inconvénients

Avantages

- ➢ Les SVM possèdent des fondements mathématiques solides.
- ➢ Les exemples de test sont comparés juste avec les supports vecteur et non pas avec tout les exemples d'apprentissage.

- Décision rapide. La classification d'un nouvel exemple consiste à voir le signe de la fonction de décision f(x).

Inconvénients

- Classification binaire d'où la nécessité d'utiliser l'approche un-contre-un.
- Grande quantité d'exemples en entrées implique un calcul matriciel important.
- Temps de calcul élevé lors d'une régularisation des paramètres de la fonction noyau.

II.2 Réseaux de Neurones

II.2.1 Introduction

Le rêve de créer une machine dotée d'une forme d'intelligence est présent depuis longtemps dans l'imagination humaine. Alors comment l'homme fait il pour raisonner, calculer, parler, apprendre, … ? C'est ces questions là qui mènent les chercheurs à essayer de comprendre le fonctionnement du cerveau humain et essayer de s'y inspirer pour pouvoir trouver de nouvelles techniques de résolutions de problèmes dans le monde informatique.

L'intelligence artificielle a apparue et ne cesse de progresser, il existe de nombreux programmes capables de diriger un robot, résoudre des problèmes etc. Néanmoins ils ne sont pas capables de rivaliser avec un cerveau humain. Outre la capacité de calcul incroyable des ordinateurs, mais ces derniers n'ont pas la faculté d'apprendre. Ils ne progressent pas si personne ne les modifie. Voilà ce à quoi les chercheurs ont essayé de remédier.

Avec l'avancée dans le domaine de la neurobiologie concernant le fonctionnement du cerveau et des neurones, des mathématiciens ont essayé de modéliser le fonctionnement du cerveau en intégrant ces connaissances en biologie dans des programmes informatiques pour leur donner la possibilité d'apprendre : c'est la naissance des réseaux de neurones [27].

II.2.2 Historique

- 1943 : Les premiers à proposer un modèle sont deux bio-physiciens de Chicago, McCulloch et Pitts, qui inventent le premier neurone formel qui portera leurs noms (neurone de McCulloch-Pitts) [28].

- 1949 : Les travaux de McCulloch et Pitts n'ont pas donné d'indication sur une méthode pour adapter les coefficients synaptiques. Cette question au cœur des réflexions sur l'apprentissage a connu un début de réponse grâce aux travaux du physiologiste américain Donald Hebb [29] qui décrit l'apprentissage dans son ouvrage (The Organization of Behaviour). Hebb a proposé une règle simple qui permet de modifier la valeur des coefficients synaptiques en fonction de l'activité des unités qu'ils relient. Cette règle aujourd'hui connue sous le nom de « règle de Hebb ».
- 1958 : Le perceptron de *Frank Rosenblatt* [30]: Il constitue donc le premier système artificiel présentant une faculté jusque là réservée aux êtres vivants : la capacité d'apprendre par l'expérience.
- 1969 : critique violente du Perceptron par *Minsky* et *Papert* [31]. Ils montrent dans un livre « *Perceptrons* » toutes les limites de ce modèle, et soulèvent particulièrement l'incapacité du Perceptron à résoudre les problèmes non linéaire tels que le célèbre problème du XOR (OU exclusif). Paraissant alors une impasse, la recherche sur les réseaux de neurones perdit une grande partie de ses financements publics, et le secteur industriel s'en détourna aussi. Il s'en suivra alors, face à la déception, une période noire d'une quinzaine d'années dans le domaine des réseaux de neurones artificiels.
- 1982 : John Joseph Hopfield [32], physicien reconnu, donna un nouveau souffle au neuronal en publiant un article introduisant un nouveau modèle de réseau de neurones (complètement récurrent). Le modèle de Hopfield souffrait néanmoins des principales limitations des modèles des années 1960, notamment l'impossibilité de traiter les problèmes non-linéaires.
- 1986 : Rumelhart [33] introduit le Perceptron Multicouche : une nouvelle génération de réseaux de neurones, capables de traiter avec succès des phénomènes non-linéaires. Le perceptron multicouche ne possède pas les défauts mis en évidence par Minsky.

II.2.3 Neurone biologique

Le système nerveux compte plus de 1000 milliards de neurones interconnectés. Les neurones ne sont pas tous identiques, ni dans leurs formes ni dans leurs caractéristiques. En effet les neurones n'ont pas tous un comportement similaire en

fonction de leur position dans le cerveau [27]. La figure II.3 montre le schéma d'un neurone biologique.

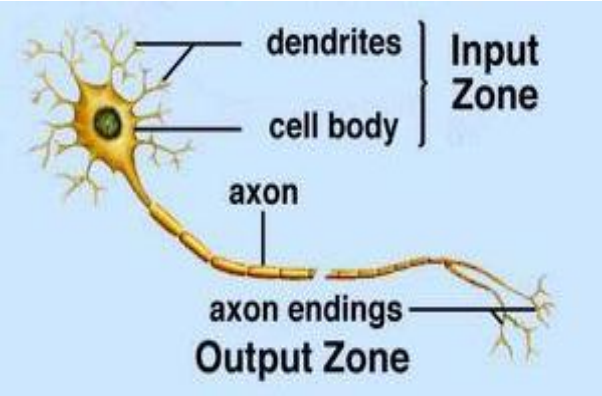

Figure II.3 : neurone biologique

II.2.4 Principe de fonctionnement

L'information ne se transmet que dans un seul sens : des dendrites vers l'axone. Le neurone va donc recevoir des informations, venant d'autres neurones, grâce à ses dendrites. Il va ensuite y avoir une sommation, au niveau du corps cellulaire, de toutes ces informations et via un signal électrique. Le résultat de l'analyse va transiter le long de l'axone jusqu'aux terminaisons synaptiques. Il va y avoir libération des neurotransmetteurs (médiateurs chimiques) dans la fente synaptique. Le signal électrique ne pouvant pas passer la synapse, les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre. Les neurotransmetteurs excitent ou inhibent le neurone suivant et peuvent ainsi générer ou interdire la propagation d'un nouvel influx nerveux. Les synapses possèdent une sorte de «mémoire» qui leur permet d'ajuster leur fonctionnement. En fonction de leur «histoire», c'est-à-dire de leur activation répétée ou non entre deux neurones, les connexions synaptiques vont se modifier : mécanisme d'apprentissage [27].

II.2.4.1 Qu'est-ce qu'un réseau de neurones ? [34]

Tout d'abord, ce que l'on désigne habituellement par réseau de neurones. Est en fait un réseau de neurones artificiels basé sur un modèle simplifié de neurone. Ce modèle permet certaines fonctions du cerveau, comme la mémorisation associative, l'apprentissage par l'exemple, le travail en parallèle, mais le neurone artificiel est loin de posséder toutes les capacités du neurone biologique. Les réseaux de neurones biologiques sont ainsi beaucoup plus compliqués que les modèles mathématiques et informatiques.

Il n'y a pas de définition universellement acceptée de « réseau de neurones ». On considère généralement qu'un réseau de neurones est constitué d'un grand ensemble d'unités (ou neurones), ayant chacune une petite mémoire locale. Ces unités sont reliées par des canaux de communication (les connexions, aussi appelées synapses d'après le terme biologique correspondant), qui transportent des données numériques. Les unités peuvent uniquement agir sur leurs données locales et sur les entrées qu'elles reçoivent par leurs connexions.

Certains réseaux de neurones sont des modèles de réseaux biologiques, mais d'autres ne le sont pas. Historiquement l'inspiration pour les réseaux de neurones provient cependant de la volonté de créer des systèmes artificiels sophistiqués, voire intelligents, capables d'effectuer des opérations semblables à celles que le cerveau humain effectue de manière routinière, et d'essayer par là d'améliorer la compréhension du cerveau.

La plupart des réseaux de neurones ont une certaine capacité d'apprentissage, cela signifie qu'ils apprennent à partir d'exemples. Le réseau peut ensuite dans une certaine mesure être capable de généraliser, c'est-à-dire de produire des résultats corrects sur des nouveaux cas qui ne lui avaient pas été présentés au cours de l'apprentissage.

II.2.4.2 Neurone formel

Les réseaux de neurone formels ou artificiels sont des réseaux dont l'architecture est inspirée de celle des réseaux de neurones biologiques (naturels), ils sont généralement optimisés par des méthodes d'apprentissage de type statistique. Leur modélisation revient à décrire le modèle du neurone (unité de base) et le modèle des connexions entre les neurones. Le premier neurone formel est apparu en 1943, introduits par MacCulloch et Pitts (unité à seuil). La figure (II.4) ci-dessous montre un schéma d'un neurone formel.

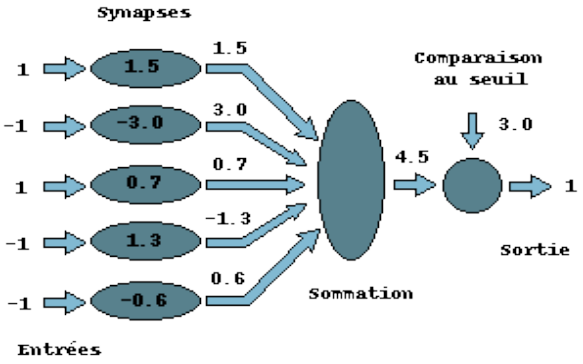

Figure II.4 : neurone formel (artificiel)

Le neurone formel est donc une modélisation mathématique qui reprend les principes du fonctionnement du neurone biologique, en particulier la sommation des entrées. Sachant qu'au niveau biologique, les synapses n'ont pas toutes la même valeur (les connexions entre les neurones étant plus ou moins fortes), les auteurs ont donc créé un algorithme qui pondère la somme de ses entrées par des poids synaptiques (sommation des multiplications des valeurs d'entrées par les coefficients de pondération). De plus sur le schéma, les 1 et les -1 en entrée sont pour différencier entre une synapse excitatrice ou inhibitrice, la sortie est obtenue après la comparaison de la sommation pondérée des entrées avec un seuil (si le résultat est supérieur au seuil, alors la valeur renvoyée est 1, sinon 0), d'où une fonction d'activation se présente (ici c'est une fonction à seuil).

Le choix d'une fonction d'activation se révèle être un élément constitutif important dans des réseaux de neurones. A titre illustratif voici quelques fonctions couramment utilisées comme fonctions d'activation : le sigmoïde standard (encore appelé fonction logistique), la tangente hyperbolique, la fonction gaussienne, une fonction à seuil.

II.2.4.3 Calcul des poids synaptiques

La rétropropagation est une méthode de calcul des poids (aussi appelés poids synaptiques du nom des synapses, terme désignant la connexion biologique entre deux neurones) pour un réseau à apprentissage supervisé qui consiste à minimiser l'erreur quadratique de la sortie (somme des carrés de l'erreur de chaque composante entre la sortie réelle et la sortie désirée).

D'autres méthodes de modification des poids sont plus locales, chaque neurone modifie ses poids en fonction de l'activation ou non des neurones proches.

II.2.5 Quelques réseaux célèbres

Il y a de très nombreuses sortes de réseaux de neurones actuellement. Personne ne sait exactement combien. De nouveaux réseaux (ou du moins des variations de réseaux plus anciens) sont inventés chaque semaine. On en présente ici de très classiques [35].

II.2.5.1 Le perceptron

C'est l'un des premiers réseaux de neurones, conçu en 1958 par *Rosenblatt.* Il est linéaire et monocouche. Il est inspiré du système visuel. La première couche (d'entrée) représente la rétine. Les neurones de la couche suivante (unique, d'où le qualificatif de monocouche) sont les cellules d'association, et la couche finale les

cellules de décision. Les sorties des neurones ne peuvent prendre que deux états (-1 et 1 ou 0 et 1). Seuls les poids des liaisons entre la couche d'association et la couche finale peuvent être modifiés. La règle de modification des poids utilisée est la règle de *Widrow-Hoff* : si la sortie du réseau (donc celle d'une cellule de décision) est égale à la sortie désirée, le poids de la connexion entre ce neurone et le neurone d'association qui lui est connecté n'est pas modifié.

Dans le cas contraire le poids est modifié proportionnellement. À la différence entre la sortie obtenue et la sortie désirée : **w <= w + k (d - s),** Où **s** est la sortie obtenue, **d** la sortie désirée et **k** une constante positive.

II.2.5.2 Les perceptrons multicouches (PMC)

Ils sont une amélioration du perceptron comprenant une ou plusieurs couches intermédiaires dites couches cachées, dans le sens où elles n'ont qu'une utilité intrinsèque pour le réseau de neurones et pas de contact direct avec l'extérieur. Chaque neurone n'est relié qu'aux neurones des couches directement précédente et suivante, mais à tous les neurones de ces couches (voir figure II.5).

Les PMC utilisent, pour modifier leurs poids, un algorithme d'apprentissage, il existe une centaine mais le plus populaire est la rétropropagation du gradient, qui est une généralisation de la règle de *Widrow-Hoff.* Il s'agit toujours de minimiser l'erreur quadratique, on propage la modification des poids de la couche de sortie jusqu'à la couche d'entrée, donc cet algorithme passe par deux phases :

- Les entrées sont propagées de couche en couche jusqu'à la couche de sortie.
- Si la sortie du PMC est différente de la sortie désirée alors l'erreur est propagée de la couche de sortie vers la couche d'entrée en modifiant les poids durant cette propagation.

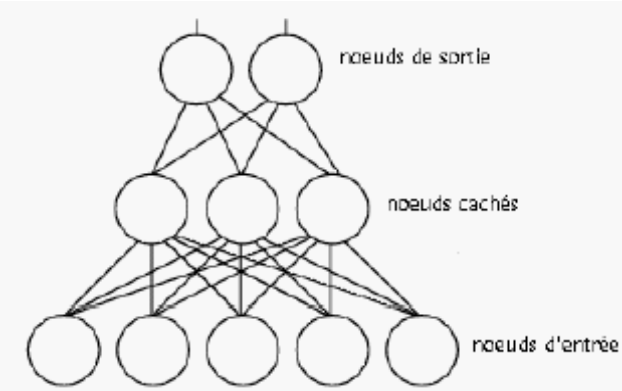

Figure II.5 : schéma d'un perceptron multicouche

II.2.5.3 Les réseaux de RBF (**R**adial **B**asis **F**onction)

Le réseau RBF est un réseau de neurones supervisé. Il s'agit d'une spécialisation d'un PMC. Un RBF est constitué uniquement de 3 couches (voir figure II.6)

- La couche d'entrée : elle retransmet les entrées sans distorsion.
- La couche RBF : couche cachée qui contient les neurones RBF.
- La couche de sortie : simple couche qui contient une fonction linéaire.

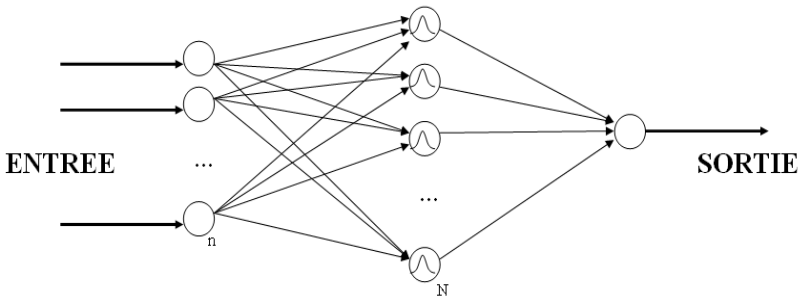

Figure II.6 : schéma d'un RBF

Chaque neurone RBF contient une gaussienne qui est centrée sur un point de l'espace d'entrée. Pour une entrée donnée, la sortie du neurone RBF est la hauteur de la gaussienne en ce point. La fonction gaussienne permet aux neurones de ne répondre qu'à une petite région de l'espace d'entrée, région sur laquelle la gaussienne est centrée. Donc il y a quatre paramètres principaux à régler dans un réseau RBF.

- Le nombre de neurones RBF (nombre de neurones dans l'unique couche cachée).
- La position des centres des gaussiennes de chacun des neurones.
- La largeur de ces gaussiennes.
- Le poids des connexions entre les neurones RBF et le(s) neurone(s) de sortie.

Toute modification d'un de ces paramètres entraîne directement un changement du comportement du réseau.

II.2.5.4 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).

II.1.5.5 Réseaux de Kohonen

Contrairement aux réseaux de *Hopfield* où les neurones sont modélisés de la façon la plus simple possible, on recherche ici un modèle de neurone plus proche de la réalité. Ces réseaux sont inspirés des observations biologiques du fonctionnement des systèmes nerveux de perception des mammifères.

Une loi de *Hebb* modifiée (tenant compte de l'oubli) est utilisée pour l'apprentissage. La connexion est renforcée dans le cas où les neurones reliés ont une activité simultanée et diminuée dans le cas contraire (alors qu'il ne se passait précédemment rien dans ce cas).

Tous ces réseaux ont des applications dans la classification, le traitement d'image, l'aide à la décision et l'optimisation.

II.2.6 Apprentissage supervisé et non supervisé

Une caractéristique des réseaux de neurones est leur capacité à apprendre (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 en suivant un algorithme d'apprentissage. Il y a deux algorithmes principaux : l'apprentissage supervisé et l'apprentissage non supervisé.

Lors d'un apprentissage supervisé, les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classement, c-à-d : on dispose d'un comportement de référence précis que l'on désire faire apprendre au réseau. L'apprentissage doit mesurer l'écart entre le comportement du réseau et celui de référence et ajuster les poids synaptiques du réseau de façon à réduire cet écart.

Lors d'un apprentissage non supervisé, on ne fournit pas au réseau les sorties que l'on désire obtenir. On le laisse évoluer librement jusqu'à ce qu'il se stabilise.

II.2.7 Utilisation des réseaux de neurones

Se trouvant à l'intersection de différents domaines (informatique, électronique robotique, science cognitive, neurobiologie et même philosophie), l'étude des réseaux de neurones est une voie prometteuse de l'Intelligence Artificielle (IA) en tant que système capable d'apprendre, mettent en œuvre le principe de l'induction, c-à-d

l'apprentissage par expérience et grâce à leur capacité de classification et de généralisation, ils servent aujourd'hui à toutes sortes d'applications et dans de nombreux domaines.

II.2.8 Avantages et inconvénients

Avantages

- Classifieur très précis (si bien paramétré).
- Apprentissage automatique des poids.
- Possibilité de faire le parallélisme (les éléments de chaque couche peuvent fonctionner en parallèle).
- Résistance aux pannes (si un neurone ne fonctionne plus, le réseau ne se perturbe pas).

Inconvénients

- Détermination de l'architecture du réseau est complexe.
- Paramètres difficiles à interpréter (boite noire).
- Difficulté de paramétrage surtout pour le nombre de neurone dans la couche cachée.