

Chapitre 4

TECHNIQUES NEURONALES ET APPLICATIONS

Un cerveau séparé de l'être vivant est incapable de pensée et d'action. Le cerveau est un des éléments d'un circuit extrêmement complexe que nous appelons l'action...En réalité, l'homme pense avec tout son corps: il pense avec ses mains, ses pieds, ses oreilles, aussi bien qu'avec son cerveau. Il est absolument ridicule de dire que sa pensée dépend d'une partie de lui-même: c'est comme si l'on disait que notre habileté manuelle dépend de nos ongles. L'activité psychologique est une activité d'ensemble, et non pas une activité locale. Le cerveau est tout simplement un ensemble de commutateurs.

Pierre Janet

Les débuts de l'intelligence (Flammarion, 1935)

4.1 Le modèle neurophysiologique

Les réseaux de neurones, ou systèmes connexionnistes sont directement inspirés de la neurobiologie.

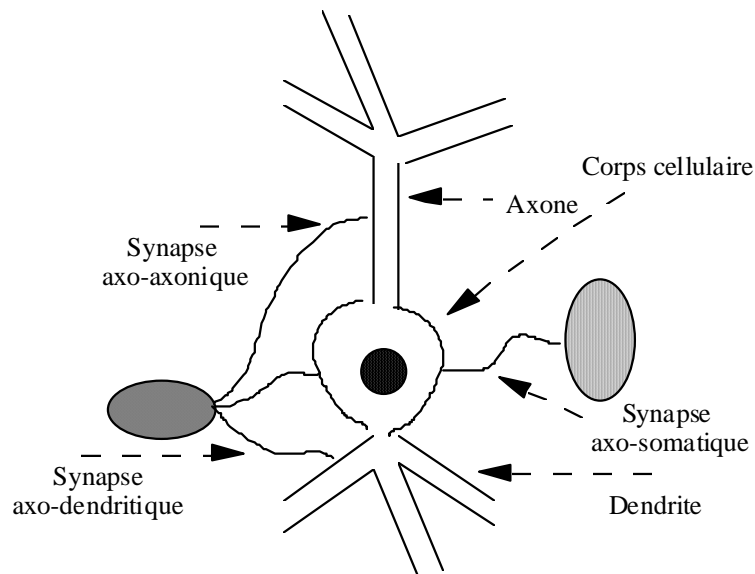


Fig. 4.1 Un neurone et son arborisation dendritique.

Le cerveau se compose d'environ 10^{12} neurones avec 1000 à 10000 synapses (connexions) par neurone. Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former des dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique. L'information est acheminée par les dendrites de l'extérieur vers le soma qui est le corps du

neurone. L'information traitée par le neurone chemine ensuite le long de l'*axone* (unique) pour être transmise aux autres neurones.

La jonction entre deux neurones est appelée *synapse*. Les neurones reçoivent des signaux d'autres neurones grâce aux dendrites et émettent des impulsions électriques dans l'axone qui se ramifie en plusieurs milliers de terminaisons où la synapse transforme le signal électrique propagé dans l'axone en signaux électriques qui inhibent ou activent le neurone post-synaptique.

Lorsqu'un neurone est plus activé qu'inhibé on dit qu'il «décharge» c'est-à-dire qu'il émet une impulsion électrique dans son propre axone. L'apprentissage consiste en une modification de l'efficacité des synapses, ce qui modifie l'influence des neurones entre eux.

4.2 Les modèles mathématiques

4.2.1 Le neurone formel

La première modélisation date des années 40. Elle a été présentée par Mac Culloch et W.Pitts [4.8] à partir de leurs travaux sur les neurones biologiques:

Un neurone formel est un automate qui fait la somme pondérée des potentiels d'actions qui lui parviennent. Chacun de ces potentiels est une valeur numérique représentant l'état du neurone qui l'a émis.

Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse (potentiel d'action) dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien. Le neurone formel est en fait un automate booléen (entrées et sortie booléennes).

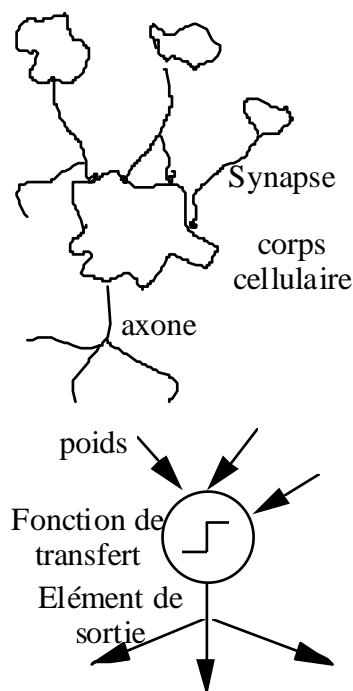


Fig. 4.2. Correspondance entre neurone biologique et neurone artificiel.

On introduit une fonction dite de seuillage ou d'activation $f(x)$ (Fig. 4.3):

$$f(x) = \begin{cases} 1 & \text{si } x > B \\ 0 & \text{si } x \leq B \end{cases}$$

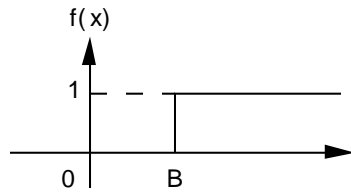


Fig. 4.3 Fonction de seuillage.

Le fonctionnement du neurone formel est exprimé par:

$$S = f\left(\sum_{i=1}^n w_i e_i\right) \quad (4.1)$$

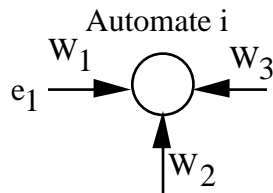


Fig. 4.4 Neurone vu comme automate.

(e_i) $i=1..n$: entrées du neurone formel,

S et B sortie et seuil,

w_i paramètres de pondération.

Les e_i représentent les valeurs de sortie des autres automates reliés à cet automate i .

Un neurone formel est caractérisé par cinq éléments (Fig. 4.5):

- Nature des entrées;
- Entrée totale: $E = h(e_1, e_2, \dots, e_n)$ définissant le prétraitement effectué sur les entrées;
- Activation (ou d'état): $A = f(E)$ définissant l'état interne du neurone en fonction de E ;
- Sortie: $S = g(A)$ qui calcule la sortie du neurone en fonction de son état d'activation;
- Nature de la sortie du neurone.

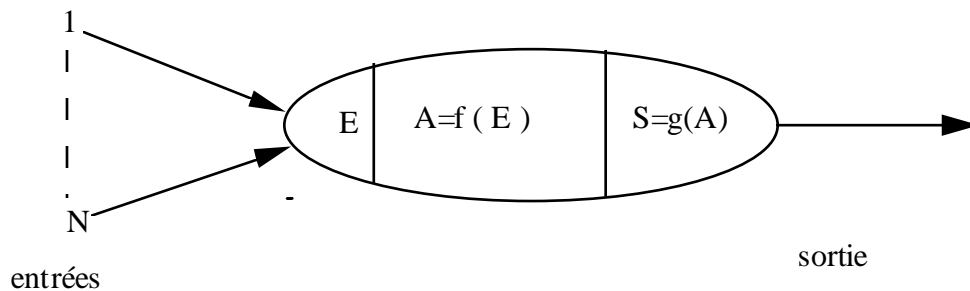


Fig. 4.5 Caractéristiques fonctionnelles d'un neurone formel.

Les entrées - sortie peuvent être: binaires (-1/+1) ou (0/1) ou réelles à valeurs bornées.

La fonction d'entrée totale h qui peut être booléenne, gaussienne (RBF [4.1]), linéaire ou affine, ces deux derniers cas étant les plus fréquents:

linéaire:

$$h(e_1, e_2, \dots, e_n) = \sum_{i=1}^n w_i e_i \quad (4.2)$$

affine:

$$h(e_1, e_2, \dots, e_n) = \sum_{i=1}^n w_i e_i - a \quad (4.3)$$

On peut interpréter la présence du terme (-a) comme l'action d'un neurone qui fournirait toujours l'entrée (-1) au neurone considéré, notation particulièrement utile dans les problèmes d'apprentissage avec des fonctions d'activations à seuil. La fonction d'activation peut être binaire à seuil (Heaviside ou Signe), linéaire à seuil ou multi-seuils, fonction sigmoïde:

$$f(x) = a(e^{kx} - 1) / (e^{kx} + 1) \quad (4.4)$$

ou toute autre fonction, généralement choisie croissante, impaire et bornée.

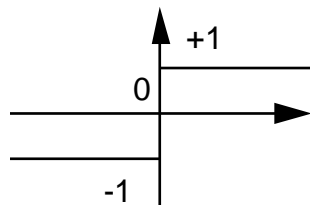


Fig. 4.6 Fonction Signe.

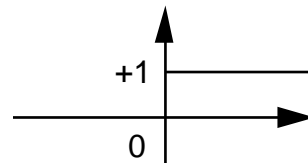


Fig. 4.7 Fonction de Heaviside.

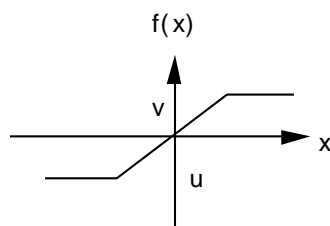


Fig. 4.8 Linéaire à seuil.

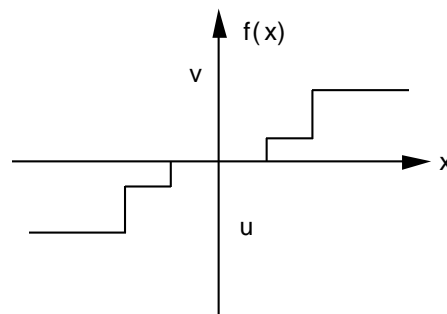


Fig. 4.9 Linéaire à multi-seuils.

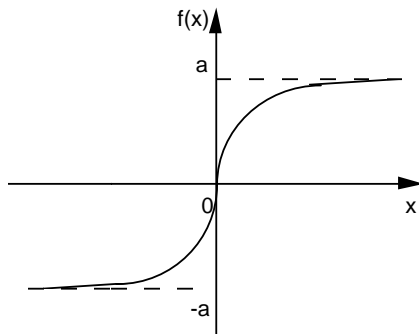


Fig. 4.10 Fonction sigmoïde.

Les neurones artificiels à activation sigmoïdale se rapprochent le plus des neurones réels.

La fonction de sortie g est en général de la forme:

$$S = f(E) = A = \text{constante} \quad (4.5)$$

Les cas les plus utilisés sont::

Les automates booléens

Entrées et sortie booléennes.
Sortie fonction booléenne des entrées

Les automates à seuil

Sortie binaire avec entrées binaires (ou réelles).
Fonction d'entrée totale affine.
Fonction d'activation f : signe (ou Heaviside)

Les automates linéaires

Entrées et sorties réelles.
Fonction d'entrée totale linéaire
Fonction de sortie identité.

Les automates à saturation

Entrées et sorties dans un intervalle $[u, v]$.
Fonction d'entrée totale linéaire.
Fonction d'activation linéaire à seuil.
Si les entrées et la sortie sont entières, on obtient un automate multi-seuils.

Les automates continus

Entrées et sortie réelles.
Fonction d'entrée totale linéaire (ou affine).
Fonction d'activation: sigmoïde.

4.2.2 Structure des connexions

Les architectures utilisables sont très variées. Pour créer un réseau de neurones qui effectuera une tâche particulière, il faut préciser le mode de connexion entre les neurones et déterminer le coefficient synaptique de chaque connexion: les connexions déterminent les influences possibles, et les coefficients synaptiques spécifient l'intensité de ces influences.

Les réseaux de neurones artificiels les plus courants sont constitués de trois couches d'éléments. La couche de neurones d'entrée est connectée à une couche de neurones "cachés" elle-même reliée à la couche de neurones de sortie. L'activité des neurones d'entrée consiste à coder l'information présentée au réseau.

L'activité de chaque neurone caché est déterminée par l'activité des neurones d'entrée et par les coefficients synaptiques des connexions entre les neurones d'entrée et les neurones cachés.

Le comportement des neurones de sortie dépend de l'activité des neurones cachés et des coefficients synaptiques reliant les neurones cachés aux neurones de sortie.

Le nombre de couches intermédiaires (cachées) est généralement variable. Les neurones cachés peuvent construire leurs propres représentations des entrées. Comme les coefficients synaptiques des connexions entre les neurones d'entrée et les neurones cachés déterminent l'activité des neurones cachés, ces derniers, en modifiant leurs coefficients synaptiques, peuvent choisir ce qu'ils représentent.

Dans les réseaux à connexions locales une topologie particulière de connexion est introduite entre les neurones de deux couches consécutives.

Chaque neurone est en relation avec un nombre réduit et localisé de neurones de la couche aval. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

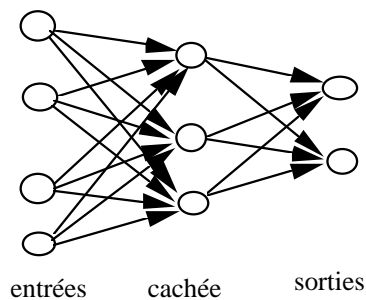


Fig. 4.11 Réseau à multicouches.

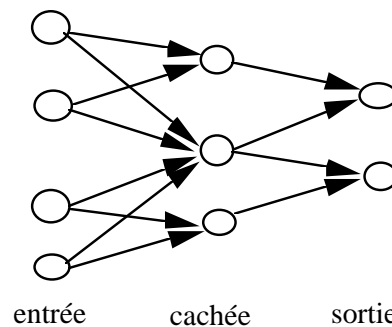


Fig. 4.12 Réseau à connexions locales.

Dans un réseau à connexions récurrentes (Fig. 4.13), les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche.

Le réseau à connexion complète (Fig 4.14) est la structure d'interconnexion la plus générale. Chaque neurone est connecté à tous les neurones du réseau (et à lui-même).

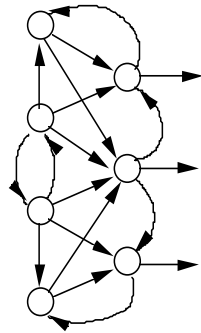


Fig. 4.13 Réseau à connexions récurrentes.

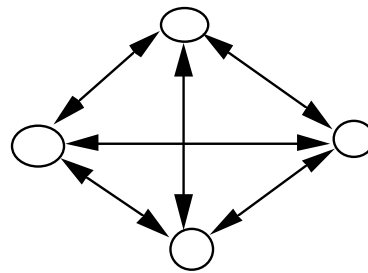


Fig. 4.14 Réseau à connexion complète.

4.3 Réseaux multicouches

4.3.1 La méthode de rétropropagation

Le vecteur de sortie calculé par le réseau est comparé au vecteur de sortie désiré. On calcule ensuite une fonction de coût ou fonction d'erreur correspondant généralement l'écart quadratique entre la sortie réelle et la sortie désirée. Puis on modifie le coefficient synaptique de chaque connexion de façon à réduire ce coût.

Il s'agit d'une méthode de gradient: chaque coefficient est modifié d'une quantité proportionnelle au taux de changement de l'erreur totale des neurones de sortie en fonction du changement du coefficient synaptiques.

Ces dérivées du coût par rapport aux coefficients synaptiques sont difficiles à calculer. L'algorithme de rétropropagation fournit une méthode itérative qui permet de les calculer de la sortie vers l'entrée du réseau.

4.3.2 Modification des poids dans la rétropropagation

On considère un réseau de neurones à n entrée et m sorties, avec un nombre quelconque de couches cachées. On suppose qu'un neurone n'est connecté qu'à des neurones en aval, dans le sens entrée-sortie et pas uniquement à des neurones situés sur la couche qui suit immédiatement.

On pose:

$$X = (x_1, x_2, \dots, x_n) \quad \text{vecteur des entrées}$$

$$Y = (y_1, y_2, \dots, y_m) \quad \text{vecteur des sorties désirées}$$

$$S = (s_1, s_2, \dots, s_m) \quad \text{vecteur des sorties effectivement obtenues}$$

f fonction sigmoïde d'un neurone (de dérivée f');
 o_j : sortie du neurone j ;
 $e(k)$: pas du gradient à l'étape k .
 L'erreur commise sur un passage complet de l'ensemble des exemples est:

$$E(w) = \sum_k E^k(w) \quad (4.6)$$

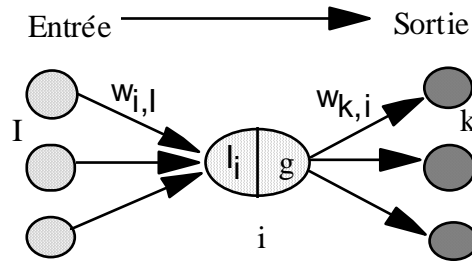


Fig. 4.15 Modèle de neurone.

$$E^k(w) = (S^k - Y^k)^2 = \sum_{i=1}^m (S_i^k - Y_i^k)^2 \quad (4.7)$$

Après chaque passage k , chaque poids est modifié comme suit:

$$w_{ij}(k) = w_{ij}(k-1) - e(k) \frac{\partial E^k}{\partial w_{ij}} \quad (4.8)$$

$$w_{ij}(k) = w_{ij}(k-1) - e(k) d_j o_j \quad (4.9)$$

avec:

$$d_i = 2f'(l_i)(S_i - y_i) \quad (4.10)$$

pour la couche de sortie,

$$d_i = \sum_h d_h w_{hi} f'(l_i) \quad (4.11)$$

pour les couches cachées.

L'indice h porte sur les neurones en amont de i .

La procédure se déroule comme suit:

- 1) Initiation des poids (valeurs aléatoires de faible grandeur);
- 2) Sélection d'un exemple d'apprentissage dans la base d'apprentissage;
- 3) Présentation de la forme d'entrée;
- 4) Calcul par propagation directe de la sortie obtenue;
- 5) Si erreur en sortie alors pour tous les neurones i (de la sortie à l'entrée).

$$d_i = 2f'(l_i)(S_i - y_i) \quad \text{si } i \text{ est un neurone de sortie}$$

$$d_i = \sum_h d_h w_{hi} f'(l_i) \quad \text{si } i \text{ est un neurone caché ou d'entrée.}$$

6) Application de la formule du gradient:

$$w_{ij}(k) = w_{ij}(k-1) - e(k)d_j o_j$$

7) Si erreur trop importante, retour à l'étape 2, autrement sortie.

Remarques

a) Le gradient est une estimation locale de la pente de la surface de coût. La minimisation du gradient consiste à parcourir cette surface orthogonalement aux courbes de niveau d'un pas fixé. L'existence de zones très plates et des minima locaux peuvent compliquer le processus.

b) La convergence de l'algorithme dépend du gain de la procédure, de la sélection des exemples pour l'apprentissage, de la configuration initiale des poids, de la pente de la fonction sigmoïde, du choix du codage des informations (en entrée et en sortie), du nombre d'itérations. Le temps d'apprentissage est proportionnel à la taille du réseau.

4.3.3 Minima locaux et accélération de l'apprentissage

Comment éviter ces minima locaux?

a) En ajoutant un terme de bruit l'équation (4.8) devient:

$$w_{ij}(k) = w_{ij}(k-1) - e(k) \frac{\partial E^k}{\partial w_{ij}} + n_t \quad (4.12)$$

n_t bruit blanc, gaussien, de variance 1.

On peut aussi ajouter un bruit n_t à tous les vecteurs d'entrée.

b) L'apprentissage conduit parfois à des oscillations de valeurs des poids. On ajoute alors au terme donné par la formule (4.8) un terme proportionnel (inertie ou moment) à la dernière variation des poids.

$$w_{ij}(k) = w_{ij}(k-1) - e(k) \frac{\partial E^k}{\partial w_{ij}} + b \Delta w_{ij}(k-1) \quad (4.13)$$

le deuxième terme étant égal à:

$$-\beta e(k-1) \frac{\partial E^{k-1}}{\partial w_{ij}} \quad (4.14)$$

$0 < \beta < 1$ (de façon usuelle, $\beta = 0,9$)

La fonction sigmoïdale peut être remplacée de façon avantageuse par une fonction linéaire ou par une gaussienne. D'autres techniques permettent d'éviter les zones très plates et les minima locaux et donc d'accélérer ainsi l'apprentissage [4.1].

4.4 Réseaux récurrents

Une autre approche naturelle consiste à utiliser des réseaux récurrents non bouclés dans lesquels certaines connexions effectuent un bouclage arrière. Le problème s'apparente à celui des techniques de filtrage adaptif. Les réseaux récurrents peuvent modéliser des phénomènes plus complexes que les modèles AR comme les modèles NARMA (Nonlinear Autoregressive Moving Average Models) Dans les réseaux "récurrents simples", les connexions en bouclage arrière ne sont pas adaptives. L'algorithme de rétropropagation du gradient peut être conservé bien que l'architecture du réseau ait été modifiée par l'introduction de connexions rétrogrades.

Une couche d'unités appelées unités de contexte, sert à mémoriser le passé de la série. Les entrées sont introduites dans le réseau une à une, à chaque itération d'apprentissage. La taille de la mémoire du réseau peut varier au cours du temps, parce que contenue dans les activités et les connexions des unités de contexte dont la valeur évolue au cours de l'apprentissage. Nous décrivons à cet effet le réseau d'Elman (Fig. 4.16):

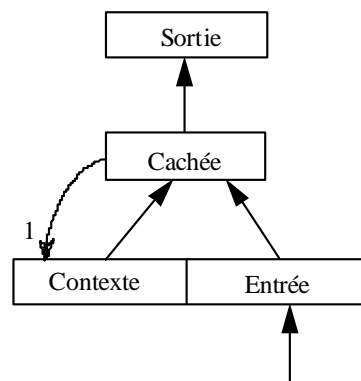


Fig. 4.16 Réseau d'Elman.

1) Initiation:

Les valeurs des neurones de la couche de contexte et les poids des connexions directes sont initiés aléatoirement, les connexions en bouclage arrière ayant un poids unité.

2) Présentation d'un exemple à l'instant t .

3) Propagation directe vers la sortie.

4) Calcul de l'erreur en sortie.

5) Calcul du gradient et modification des poids (rétropropagation classique).

6) Recopie des valeurs de la couche cachée dans la couche de contexte précédente.

7) Présentation d'un nouvel exemple à l'instant ($t+1$).

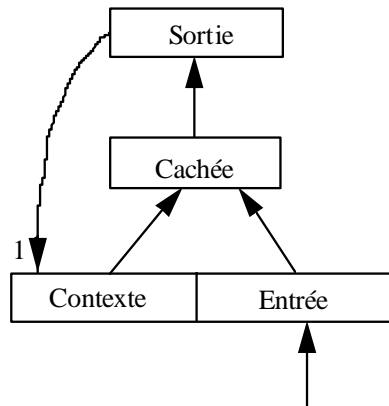


Fig. 4.17 Réseau de Jordan.

Les connexions en bouclage arrière ont donc pour seul effet de donner à certains neurones (couche de contexte) une sorte de mémoire (du fait de la copie).

Tout se passe comme si la mémoire à long terme était codée grâce aux poids, la mémoire à court terme étant codée dans l'activation des neurones des couches cachées qui n'ont aucun contact avec l'environnement.

Le modèle de Jordan (Fig. 4.17) est une variante du modèle d'Elman.

Il s'agit d'une structure multicouche dans laquelle les boucles de rétropropagation relient la dernière couche avec la première couche. Les neurones de sortie sont rebouclés sur les unités de contexte par des connexions fixées à 1. L'algorithme d'apprentissage est une généralisation de la rétropropagation de gradient.

4.5 Auto-organisation et compétition

Il existe deux types d'apprentissage:

L'apprentissage supervisé: en phase d'apprentissage on présente simultanément au réseau une entrée et la sortie désirée (cf. réseaux à couches).

L'apprentissage non supervisé: consiste à détecter automatiquement les régularités qui figurent dans les exemples présentés et modifier les poids des connexions pour que les exemples ayant les mêmes caractéristiques de régularité donnent la même sortie. Cet apprentissage est également utilisé dans une classe de règles d'apprentissage dites règles d'apprentissage compétitif:

- une seule unité est active à un instant donné;
- les unités de sortie sont en compétition pour savoir laquelle est active ("winner-take-all").

Le réseau à apprentissage compétitif de base est un réseau à deux couches sans rétroaction: chaque nœud d'entrée est connecté à chaque nœud de sortie. La suite finie des poids des connexions reliant les nœuds d'entrée à un nœud de sortie constitue le vecteur poids de ce nœud; un réseau classe le vecteur motif d'entrée par comparaison avec les vecteurs poids des nœuds de sortie par produits scalaires de ces vecteurs:

$$h_i = \sum_j w_{ij} e_j = W_i \cdot E \quad (\text{produit scalaire}) \quad (4.15)$$

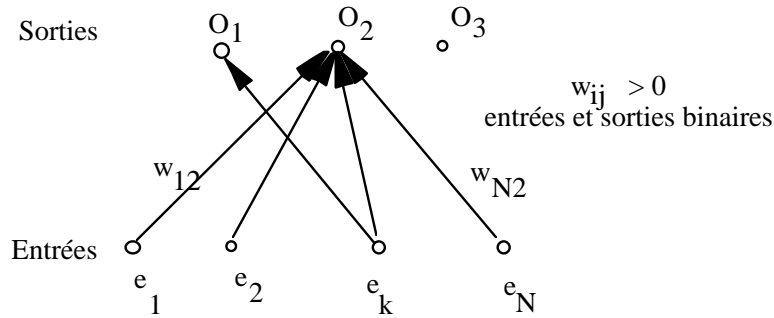


Fig. 4.18 Réseau d'apprentissage concurrentiel

Le neurone gagnant est celui pour lequel h_i a la valeur la plus grande + pour l'entrée donnée. Par exemple, i^* est gagnant si:

$$W_{i^*} \cdot E \geq W_i \cdot E \quad \text{pour tout } i \quad (4.16)$$

$$O_{i^*} = 1 \quad (4.17)$$

Si les poids sont normalisés ($|W_i| = 1$) cette formule est équivalente à :

$$|W_{i^*} \cdot E| \leq |W_i \cdot E| \quad \text{pour tout } i \quad (4.18)$$

On peut mettre en œuvre la règle de “winner-take-all” en utilisant l'inhibition mutuelle. Les activités de certains neurones des aires corticales sont en effet fortement corrélées.

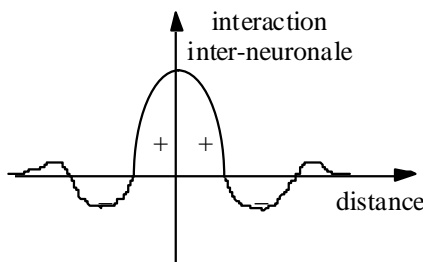


Fig. 4.19 La forme de chapeau mexicain (interaction latérale).

L'image du chapeau mexicain est une bonne représentation de ces corrélations: excitation latérale à courte portée (50 à 100 μm), inhibition à moyenne portée (200 à 500 μm), interaction excitatrice plus faible tendant rapidement vers 0 à plus longue portée.

Cartes topologiques de Kohonen

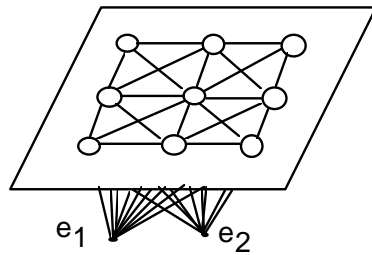


Fig. 4.20 Architecture d'une carte auto-organisatrice (Réseau 2D).

Ce modèle propose un mécanisme pour réaliser ces corrélations. Ce type de réseau est appelé carte parce qu'il conserve, au niveau de la couche de connexions qui réalise cette transformation, les relations topologiques qui existaient dans le signal d'entrée: si deux formes sont voisines en entrée, elles seront représentées par deux neurones ou deux amas de neurones voisins.

L'algorithme réalise un classement des données selon des critères générés par le réseau lui-même. L'algorithme d'apprentissage est le suivant:

- 1) Initiation des poids pour N entrées et M sorties (aléatoires de faible grandeur, et de la dimension initiale du voisinage);
- 2) Présentation d'une entrée à l'instant t ;
- 3) Calcul de la distance des vecteurs poids de chacun des neurones de la carte à l'entrée:

$$d_j = \sum_{i=0}^{N-1} (e_i(t) - w_{ij}(t))^2 \quad (4.19)$$

- 4) Sélection du neurone le plus proche:

$$d_{j^*} = \min d_j \quad 0 \leq j \leq M-1 \quad (4.20)$$

- 5) Modification des poids pour le neurone choisi (j^*) et son voisinage V_{j^*} :

$$w_{ij}(t+1) = w_{ij}(t) + h(t) \cdot (e_i(t) - w_{ij}(t)) \quad (4.21)$$

$$j \in V_{j^*}(t) \quad \text{et} \quad 0 \leq i \leq N-1$$

- $\eta(t)$ tel que ($0 < \eta(t) < 1$): gain décroissant lentement en fonction du temps
- 6) Tant que les performances sont insuffisantes, retour à l'étape 2.

Remarques

- 1) La précision finale de la transformation dépend du nombre de pas d'itérations qui doit être assez grand.
- 2) Il y a 2 phases dans le déroulement de l'algorithme:

- *ordonnement des poids* conduisant à la formation d'un classement grossièrement correct;
 - convergence, plus longue que la phase précédente, qui conduit au *raffinement du classement* et à la mise en place d'une carte finale.
- 3) Ces phases correspondent à des valeurs différentes de la taille du voisinage et du coefficient d'apprentissage η :
- voisinage assez grand par rapport à la taille du réseau η décroît de 1 vers zéro (de l'ordre de 0,1);
 - taille du voisinage de plus en plus petite, η petit, constant ou presque.

Le modèle de Kohonen présente deux caractéristiques: auto-adaptation et projection. L'auto-adaptation entraîne la conservation d'activation neuronale au niveau de la carte, des relations topologiques présentes dans les signaux d'entrées. Elle permet de faire un module de pré-traitement ou d'orthogonalisation en reconnaissance des formes. La projection effectuée par les filtres de Kohonen assure à la fois un rôle de compression des données et une meilleure résistance au bruit pour les problèmes de mémoires auto ou hétéro-associatives.

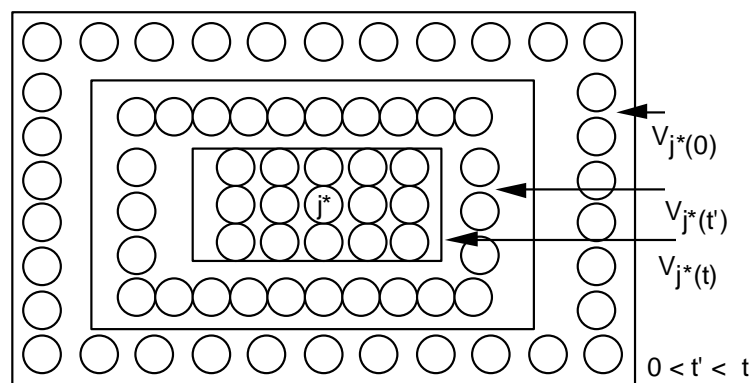


Fig. 4.21 Evolution temporelle du voisinage 2D.

4.6 Applications

4.6.1 Transmission en radiocommunication numérique

Il s'agit de l'identification automatique de modes de transmission [4.2]. Dans le domaine civil il faut être en mesure de contrôler les radiocommunications; dans le domaine militaire les méthodes de transmission protégées rendent de plus en plus difficiles les techniques de contre-mesure. Il s'agit d'un problème de reconnaissance des formes qui se divise en deux grandes phases:

- a) Une phase de traitement du signal: on extrait du signal reçu les signatures des modulations étudiées. C'est la phase d'acquisition des données.

b) Une phase de classification: cette classification est faite sur les données fournies par la phase précédente en considérant un ensemble de modulations: amplitude sans porteuse (AMP), fréquence (FREQ), bande latérale unique (BLU), double déplacement de phase (BPSK), quadruple déplacement de phase (QPSK), double déplacement de fréquence (BFSK)

Le prétraitement effectué sur le signal inconnu reçu permet d'en récupérer l'enveloppe complexe échantillonnée, suite de valeurs complexes dont on peut extraire l'amplitude, la fréquence et la phase instantanées (en supposant connue la bande de fréquence dans laquelle le signal est émis). On peut alors construire les histogrammes qui seront les signatures de chacune des modulations et du bruit. On crée pour chacun de ces modulations un fichier de données. Chacun des signaux modulés simulés est bruité avec différents niveaux de rapport signal sur bruit dans l'ordre décroissant de puissance de bruitage. On utilise trois réseaux à trois couches qui utilisent l'algorithme de rétropropagation pour chacune des trois variables considérées.

Les étapes de construction et de validation des réseaux sont réparties en quatre phases: choix des entrées et des sorties des réseaux, choix de l'architecture totale des trois réseaux étudiés, tests des réseaux sélectionnés sur des nouveaux exemples. Les trois premières phases ne se sont pas indépendantes et se déroulent simultanément.

Entrées

- normalisation des valeurs dans un intervalle $[0,1]$;
- construction d'histogrammes par quantification des valeurs normées;
- normalisation des ces histogrammes dans l'intervalle $[0,1]$.

Sorties

En fonction de la variable choisie, fréquence ou phase instantanée ou amplitude, on choisit des ensembles de classes de modulation. Par exemple:

- Fréquence instantanée: AMP, BLU, FREQ ou BRUIT, QPSK, BPSK ou BFSK.
- Amplitude: AMP ou BLU, BRUIT, BPSK ou BFSK ou QPSK ou FREQ.

- Phase instantanée: AMP, QPSK, BPSK, BRUIT ou FREQ ou BLU ou BFSK.

Une classe est attribuée à chaque neurone de sortie et réciproquement.

Architecture totale

On utilise trois réseaux. A titre d'exemple, pour la variable fréquence instantanée: 32 neurones en entrée, 18 neurones en couche cachée, 5 neurones en sortie. Pour tous ces réseaux, l'apprentissage est achevé lorsque l'erreur maximale sur tous les neurones de sortie à 0.1 pour une entrée quelconque de la base d'apprentissage.

Trois problèmes doivent être signalés:

- La sensibilité de l'algorithme de rétropropagation (convergence) à l'ordre des présentations de la base d'exemples et au prétraitement effectué;
- L'importance du temps de calcul;
- L'indétermination liée au tirage aléatoire des poids des connexions lors de la création d'un réseau.

4.6.2 Reconnaissance de chiffres manuscrits

Ces chiffres manuscrits sont par exemple issus des en-têtes de fax de format fixe. Les cadres dans lesquels les scripteurs peuvent écrire librement permettent de délimiter des zones facilement extractibles. Quelques opérations préliminaires sont nécessaires à la reconnaissance des chiffres: extraction des champs, extraction des chiffres, normalisation.

Extraction des champs

L'extraction des champs se fait par recherche des angles gauches hauts et bas dans une zone limitée - pour accélérer le processus.

Des coordonnées de cet angle va dépendre la position de la zone suivante, et ainsi de suite jusqu'au dernier champ. Toutes ces zones de recherche sont indispensables - malgré la connaissance a priori du format du fax - pour prévenir les variations en rotation.

Une fois les deux angles trouvés, le champ est extrait en suivant le bord supérieur; ceci permet de redresser le contenu des champs qui ont subi des transformations.

Extraction des chiffres

Plusieurs problèmes se posent en raison de la variabilité de l'écriture manuscrite. Il existe en effet plusieurs formats pour écrire un numéro de téléphone: couples de chiffres seuls, couples de chiffres séparés par un point, couples de chiffres séparés par des traits, mélange des trois premières propositions. D'autre part, les couples de chiffres sont eux-mêmes sujet à de fortes variations: caractères liés, caractères imbriqués, caractères penchés à droite ou à gauche, caractères parasites.

L'algorithme est basé sur l'étude des histogrammes verticaux et horizontaux des champs extraits, ce qui permet facilement d'éliminer les pixels parasites ainsi que les points ou les traits séparant deux couples de chiffres. On calcule la largeur moyenne des caractères de façon à déceler d'éventuels caractères liés. Cette moyenne est remise à jour à chaque fax pour s'affranchir des variations de l'écriture. La séparation de caractères liés est obtenue par "suivi de contour".

Normalisation

La normalisation conserve le rapport hauteur/largeur constant. On normalise suivant la plus grande valeur (hauteur ou largeur) et on garde le même rapport de réduction dans les deux dimensions. Ceci évite d'avoir des caractères difformes (les "1" sont souvent plus hauts que larges et les "0" parfois plus larges que hauts). Afin de garder une information sur les bords du caractère, on normalise sur cinq niveaux de gris (valeur entre 0 et 1) dans une fenêtre d'un nombre donné de pixels. A titre d'exemple, on a pu réaliser une recherche avec les éléments suivants:

Base de données

test: 2000 chiffres;

apprentissage: 6010 chiffres;

Réseau

réseau de Le Cun à poids partagés [4.7] avec:

une fenêtre d'entrée 15 x 20 pixels

couches cachées;

une couche de sortie de 10 neurones;

1161 neurones;

22430 connexions;

3231 poids libres;

temps d'apprentissage:

2 heures (algorithme de rétropropagation du gradient) sur SPARC10, la station utilisée.

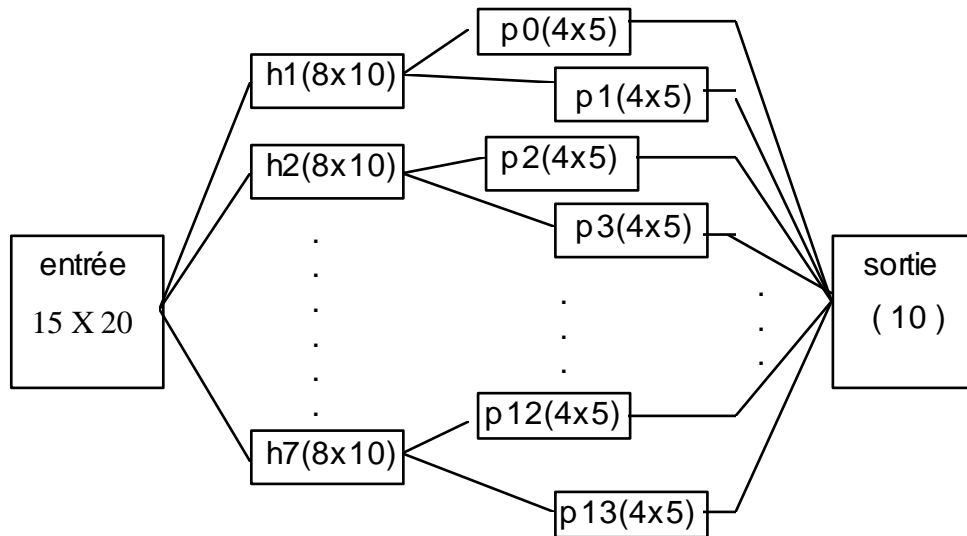


Fig. 4.22 Réseau LeCun à poids partagés.

4.6.3 Identification d'un signal radioélectrique (GSM)

La réutilisation des canaux radio est limitée notamment par un phénomène de brouillage entre les émissions sur une même porteuse. On y remédie de trois façons:

- réglage de puissance pour réduire au minimum le brouillage entre les cellules utilisant un même canal radio;
- émission discontinue (mécanisme activé par un détecteur d'activité vocale);
- saut de fréquence à chaque intervalle de temps.

Nous nous plaçons dans le cadre d'une généralisation de la technique de synchronisation en fréquence utilisée dans le cadre du système de radiotéléphonie cellulaire GSM aux futurs systèmes de radiotéléphonie par satellite. L'application consiste à traiter, par réseau de neurones, deux fonctions, si possible en parallèle [4.9]:

- reconnaître s'il s'agit d'un paquet FCCH donc d'une sinusoïde,
- mesurer une valeur approchée de sa fréquence dans un temps minimum

Le signal considéré est une sinusoïde de fréquence 67.708 kHz affectée d'un Doppler dans l'intervalle [-10kHz;+10kHz] portée par le paquet FCCH. Le problème a été traité en deux temps :

- détermination de la fréquence d'une sinusoïde bruitée;
- mise en place d'une alarme lorsqu'on obtient une sinusoïde bruitée et non du bruit blanc gaussien.

Architecture du réseau et algorithme d'apprentissage

Il s'agit d'un réseau multicouche avec une seule couche cachée. Le réseau est entraîné par l'algorithme de rétropropagation du gradient. Les entrées du réseau représentent l'échantillonnage à un point par bit du signal porté par le FCCH. Les neurones de la couche cachée ont une fonction

d'activation sigmoïdale. Le neurone en sortie du réseau qui code une valeur non bornée, telle que la fréquence, a une fonction d'activation linéaire.

Pour résoudre le problème de synchronisation temporelle et fréquentielle dans les systèmes de radiocommunications par satellite utilisant des techniques connexionnistes, une solution est la suivante (Fig. 4.23):

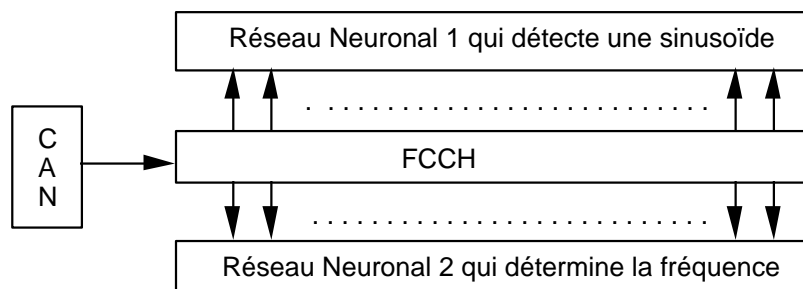


Fig. 4.23 Synchronisation temporelle et fréquentielle.

Le réseau 1 détecte (ou non) le paquet FCCH, et le réseau 2 détermine une fréquence; cette fréquence est la fréquence recherchée dans le cas où le réseau 1 a bien détecté une sinusoïde.

4.6.4 Traitement de la parole

Le signal de parole est très redondant: 300000b/s pour coder intégralement le signal alors quelques centaines de bits seraient suffisantes pour coder le sens du message parlé. Un pré-traitement permet de comprimer le signal soit par des méthodes générales de traitement du signal, soit par des techniques spécifiques au signal de parole (vocodeurs à canaux ou à prédiction linéaire).

Un vocodeur à canaux est un banc de filtres de fréquences différentes; chacun analyse le signal dans la bande de fréquences correspondante. Dans le vocodeur à prédiction linéaire, on établit un modèle théorique du conduit vocal, dont on estime les paramètres en échantillonnant le signal de parole: ce sont les coefficients LPC (Linear Predictive Coding).

Une autre méthode de compression du signal de parole fait appel à la notion de *phonème*: le nombre de sons effectivement utilisés -les phonèmes- dans une langue donnée est limité. Le problème se ramène donc à l'extraction des phonèmes (ou diphonèmes) généralement à partir du signal de parole prétraité.

Reconnaissance automatique de la parole

a) Le problème

La reconnaissance est effectuée en deux temps: le signal de parole est d'abord mis en " formes " comparées ensuite à des " modèles " stockés en mémoire. La prononciation d'un mot peut donner lieu à une infinité potentielle de " formes " et le bon système doit être capable de détecter dans ces formes

des régularités et des corrélations suffisantes pour conduire à une représentation phonèmes ou mots.

Actuellement, aucun système n'est capable de résoudre le problème dans le cadre de grands vocabulaires avec indépendance du locuteur, continuité de parole, bruit (enregistrement en milieu normal ou qualité téléphonique). Les différents systèmes imposent des contraintes sur l'une ou l'autre des dimensions du problème: systèmes multilocuteurs, petits vocabulaires de mots isolés en reconnaissance sur ligne téléphonique, par exemple. Deux approches se sont succédées en reconnaissance automatique de la parole: l'approche par modèle stochastique (chaînes de Markov cachées) et approche neuronale.

Les réseaux de neurones sont d'excellents outils de discrimination; ils sont capables de modéliser les relations entre entrée-sortie sans autre information explicite que la donnée des exemples, et de représenter ces relations. Le problème essentiel est celui de la prise en considération du temps car le signal de parole est un signal temporel. De plus, la reconnaissance d'une trame (fenêtre temporelle, de courte durée 26 ms par exemple) exige généralement la connaissance de trames voisines: l'effet de contexte est souvent déterminant. On a d'abord utilisé des réseaux statiques, dans lesquels la dimension temporelle n'est prise en considération que par l'intermédiaire d'une fenêtre de contexte, de dimension fixe, qui est passée sur le signal de parole. Deux exemples illustrent bien cette approche connexionniste.

b) Les TDNN: reconnaissance de mots

Les Time Delay Neural Networks ou TDNN ont été introduits [4.6] pour tenir compte la dimension temporelle du signal de parole. Pour reconnaître un mot dans le signal de parole, on peut déterminer des événements acoustiques élémentaires, puis détecter ces événements dans le signal; un mot est alors constitué par une suite spécifique d'événements.

Un événement acoustique caractéristique doit être reconnu quelle que soit sa position dans le mot d'où la mise au point des filtres d'événements ou détecteurs de traits, invariante par translation temporelle.

Pour réaliser ce filtre, on contraint un ensemble de cellules à partager les mêmes poids (coefficients du filtre); les différentes cellules sont connectées à différentes parties du signal, et détectent donc l'événement caractérisé par le filtre dans différentes portions du signal.

La technique TDNN permet donc de tenir compte efficacement de la dimension temporelle du signal de parole. Les coefficients sont " appris " en utilisant une modification de l'algorithme classique de rétropropagation du gradient; les filtres résultant de l'apprentissage seront donc optimisés sur une base d'exemples.

c) Réseaux non bouclés: reconnaissance de phonèmes

Les phonèmes constituent une méthode de codage compact du signal de parole. Certains, comme les voyelles, sont très faciles à reconnaître; d'autres, comme les plosives sont beaucoup plus difficiles. Un travail exploratoire dans le domaine de la reconnaissance de la parole par réseaux connexionnistes a été consacré à la reconnaissance des 3 plosives /b/, /d/, /g/ dans le contexte des trois voyelles /a/, /i/, /u/.

La base de données a été constituée à partir d'enregistrements d'un seul locuteur ayant prononcé 56 exemples de chacune des syllabes: /ba/, /bi/, /bu/, /da/, /di/, /du/, /ga/, /gi/, /gu/. La moitié des exemples a été utilisée pour l'apprentissage, l'autre pour le test. Au cours de l'apprentissage, le signal a été bruité additivement (ce qui améliore en général les performances en généralisation).

Le réseau est multicouche, avec une couche cachée de 3 cellules. La couche d'entrée a 16x20 cellules. Trois versions sont proposées pour la couche de sortie:

- dans les deux premières, il y a 3 cellules pour classer le signal d'entrée dans l'une des 3 classes /b/, /d/, /g/ ou /a/, /i/, /u/
- dans la troisième, 9 pour les classes /ba/, /bi/, /bu/, /da/, /di/, /du/, /ga/, /gi/, /gu/.

Les couches sont complètement connectées.

La très grande majorité des travaux connexionnistes en matière de traitement de la parole utilisent le modèle de base du réseau multicouches avec l'algorithme d'apprentissage de la rétropropagation de gradient, modifié de diverses façons (modèle récurrent, TDNN à poids partagés...).

Il existe également quelques applications basées sur d'autres algorithmes, mais les résultats obtenus par ces techniques semblent, pour l'instant, moins convaincants.

Les tout derniers travaux en reconnaissance automatique de la parole semblent indiquer une voie qui utiliserait des systèmes hybrides, mêlant plusieurs algorithmes, connexionnistes ou pas. En particulier, des architectures mixtes réseaux multicouches et HMM semblent très prometteuses [4.10].

Sources bibliographiques

- [4.1] **HERTZ, J.; KROGH, A.; PALMER, R.:** *Introduction to the theory of neural computation*. Addison-Wesley, 1991
- [4.2] **DAVALO, E.; NAÏM, P.:** *Des réseaux de neurones*, Paris: Eyrolles, 1992
- [4.3] **FREEMAN, J A.; DAVID, S M.:** *Neural networks: algorithms, applications, and programming techniques*, Addison-Wesley, 1991
- [4.4] **BOURRET, P.; REGGIA, J.; SAMUELIDES, M.:** *Réseaux neuronaux. Une approche connexionniste de l'intelligence artificielle*, Teknea, 1991
- [4.5] **RABINER, L.R.; SCHAFER R.W.:** *Digital Processing of Speech Signals*, Englewood Cliffs, N. J. , Prentice Hall, 1978

-
- [4.6] **WAIBEL, A.:** *Modular construction of Time-Delay Neural Networks for Speech Recognition*, Neural Computation 1, 39-46, 1989
- [4.7] **LE CUN, Y.; BOSER, B.; AL.:** *Handwritten Digit Recognition with a Back-Propagation Network*, Advances in Neural Information Processing Systems II, pp. 396-04, Morgan Kaufmann, 1990
- [4.8] **MAC CULLOCH; PITTS, W.:** "A logical calculus of the ideas imminent in nervous activity" Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943
- [4.9] **SULPYCE, V:** *Techniques et réseaux de neurones* Rapport de stage INT, 1995
- [4.10] **BOULARD, H.; WELLEKENS, C. J.:** "Links between Markov models and Multi-Layer Perceptrons " , Research Report M 263, Philips Research Laboratory, Brussels, 1988