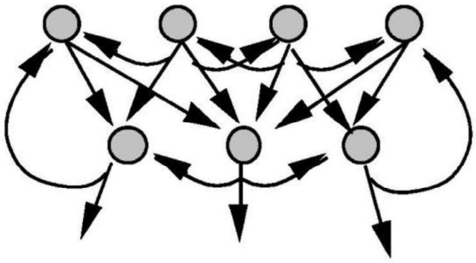

Figure 6. Réseau à connexions récurrentes

Réseau à connexion complète : c'est la structure d'interconnexion la plus générale (fig. 7). Chaque neurone est connecté à tous les neurones du réseau (et à lui-même).

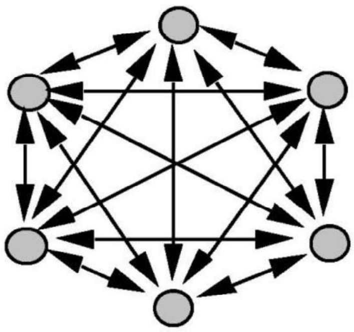

Figure 7. Réseau à connexions complète

Il existe de nombreuse autres topologies possibles, mais elles n'ont pas eu à ce jour la notoriété des quelques unes que nous avons décrites ici.

## 4 Fonctionnement

### 4.1 Perceptron

Avant d'aborder le comportement collectif d'un ensemble de neurones, nous allons présenter le Perceptron (un seul neurone) en phase d'utilisation. L'apprentissage ayant été réalisé, les poids sont fixes. Le neurone de la figure 8 réalise une simple somme pondérée de ses entrées, compare une valeur de seuil, et fourni une réponse binaire en sortie. Par exemple, on peut interpréter sa décision comme classe 1 si la valeur de x est +1 et classe 2 si la valeur de x est -1.

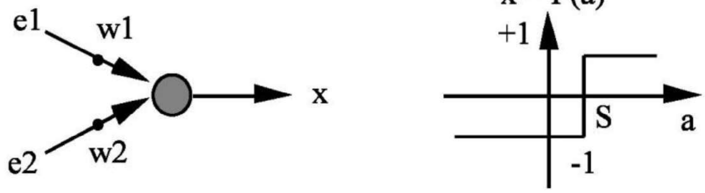

Figure 8. Le Perceptron : structure et comportement. Les connexions des deux entrées $e_1$ et $e_2$ au neurone sont pondérées par les poids $w_1$ et $w_2$. La valeur de sortie du neurone est notée x. Ele est obtenue après somme pondérée des entrées (a) et comparaison à une valeur de seuil S.

Question : Sachant que les poids du Perceptron à deux entrées sont les suivants : $w_1 = 0.5$, $w_2 = 0.2$ et que la valeur de seuil est $S = 0.0$, déterminez son comportement, sachant que les comportements du ET logique, OU logique et OU exclusif sont rappelés Table 1 :

| ET | | | OU | | | OU Exclusif | | |
|---|---|---|---|---|---|---|---|---|
| e1 | e2 | x | e1 | e2 | x | e1 | e2 | |
| 11 | 1 | | 1 | 1 | 1 | 1 | 1 | 1 |
| 1-1 | -1 | | 1 | -1 | 1 | 1 | -1 | -1 |
| -11 | -1 | | -1 | 1 | 1 | -1 | 1 | -1 |
| -1-1 | -1 | | -1 | -1 | -1 | -1 | -1 | 1 |

?

# Apprentissage

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux. Elle ne concerne cependant pas tous les modèles, mais les plus utilisés.

Définition :

*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. L'apprentissage neuronal fait appel à des exemples de comportement.*

Dans le cas des réseaux de neurones artificiels, on ajoute souvent à la description du modèle l'algorithme d'apprentissage. Le modèle sans apprentissage présente en effet peu d'intérêt. Dans la majorité des algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions. L'apprentissage est la modification des poids du réseau dans l'optique d'accorder la réponse du réseau aux exemples et à l'expérience. Il est souvent impossible de décider à priori des valeurs des poids des connexions d'un réseau pour une application donnée. A l'issu de l'apprentissage, les poids sont fixés : c'est alors la phase d'utilisation. Certains modèles de réseaux sont improprement dénommés à apprentissage permanent. Dans ce cas il est vrai que l'apprentissage ne s'arrête jamais, cependant on peut toujours distinguer une phase d'apprentissage (en fait de remise à jour du comportement) et une phase d'utilisation. Cette technique permet de conserver au réseau un comportement adapté malgré les fluctuations dans les données d'entrées.

Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé. Cette distinction repose sur la forme des exemples d'apprentissage. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie associée) alors que l'on ne dispose que des valeurs (Entrée) pour l'apprentissage non supervisé. Remarquons cependant que les modèles à apprentissage non supervisé nécessite avant la phase d'utilisation une étape de labélisation effectuée l'opérateur, qui n'est pas autre chose qu'une part de supervision.

## 1 La loi de Hebb, un exemple d'apprentissage non supervisé

La loi de Hebb (1949) s'applique aux connexions entre neurones, comme le représente la figure 1.

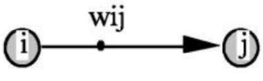


Figure 1. i le neurone amont, j le neurone aval et $w_{ij}$ le poids de la connexion.

Elle s'exprime de la façon suivante

"*Si 2 cellules sont activées en même temps alors la force de la connexion augmente*".

La modification de poids dépend de la coactivation des neurones présynaptique et post synaptique, ainsi que le montre la table 1. $x_i$ et $x_j$ sont respectivement les valeurs d'activation des neurones i et j, $\partial wij$ (dérivée partielle du poids) correspond à la modification de poids réalisée.

| $x_i$ | $x_j$ | $\partial w_{ij}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | + |

Table 1. La loi de Hebb.

La loi de Hebb peut être modélisée par les équations suivantes (w(t+1) est le nouveau poids, wij(t) l'ancien) :

$w_{ij}(t+1) = w_{ij}(t) + \partial w_{ij}(t)$

$\partial w_{ij}(t) = x_i \, . \, x_j$ (la coactivité est modélisée comme le produit des deux valeurs d'activation) ***= d . e***

L'algorithme d'apprentissage modifie de façon itérative (petit à petit) les poids pour adapter la réponse obtenue à la réponse désirée. Il s'agit en fait de modifier les poids lorsqu'il y a erreur seulement.

1/ Initialisation des poids et du seuil S à des valeurs (petites) choisies au hasard.

2/ Présentation d'une entrée $E_l = (e_1, \ldots e_n)$ de la base d'apprentissage.

3/ Calcul de la sortie obtenue x pour cette entrée :

$a = \sum (w_i \, . \, e_i) - S$ (la valeur de seuil est introduite ici dans le calcul de la somme pondérée)

$x = \text{signe}(a)$ ( si $a > 0$ alors $x = +1$ sinon $a \leq 0$ alors $x = -1$ )

4/ Si la sortie x est différente de la sortie désirée $d_l$ pour cet exemple d'entrée $E_l$ alors modification des poids (μ est une constante positive, qui spécifie le pas de modification des poids) :

$w_{ij}(t+1) = w_{ij}(t) + \mu.(x_i \, . \, x_j)$ $\Rightarrow$ ***Wij(t+1)=wij(t)+u( d . e)***

5/ Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement (i.e. modification des poids), retour à l'étape 2.

Exemple d'application de l'algorithme d'apprentissage de Hebb :

Choisissons pour les neurones un comportement binaire. Les entrées $e_1$ et $e_2$ sont considérées comme des neurones (fig. 2).

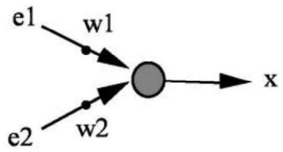


Figure 2. Réseau de 3 neurones (les 2 entrées sont considérées comme deux neurones) pour la résolution du problème exprimé table 2.

Nous allons réaliser l'apprentissage sur un problème très simple. La base d'apprentissage est décrite par la table 2 :

| $e_1$ | $e_2$ | | |
|---|---|---|---|
| 1 | 1 | 1 | (1) |
| 1 | -1 | 1 | (2) |
| -1 | 1 | -1 | (3) |
| -1 | -1 | -1 | (4) |

Table 2. Base d'exemples d'apprentissage pour la loi de Hebb.

1/ Conditions initiales : $\mu = +1$, les poids et le seuil sont nuls.
2/ Calculons la valeur de x pour l'exemple (1) :
3/ $a = w_1.e_1 + w_2.e_2 - S = 0.0 . 1 + 0.0 . 1 - 0.0 = 0$ $a \leq 0 \Rightarrow x = -1$
4/ La sortie est fausse, il faut donc modifier les poids en applicant :
$w_1 = w_1 + e_1.x = 0.0 + 1.1 = 1$
$w_2 = w_2 + e_2.x = 0.0 + 1.1 = 1$
2/ On passe à l'exemple suivant (2) :
3/ $a = 1.1 + 1.-1 -0.0 = 0$ $a \leq 0 \Rightarrow x = -1$
4/ La sortie est fausse, il faut donc modifier les poids en applicant :
$w_1 = 1 + 1.1 = 2$
$w_2 = 1 + 1.-1 = 0$
.../ L'exemple suivant (3) est correctement traité : a = -2 et x = -1 (la sortie est bonne).
On passe directement, sans modification des poids à l'exemple (4). Celui-ci aussi est correctement traité. On revient alors au début de la base d'apprentissage : l'exemple (1). Il est correctement traité, ainsi que le second (2). L'algorithme d'apprentissage est alors terminé : toute la base d'apprentissage a été passée en revue sans modification des poids.

Question : Soit le réseau composé de 4 neurones d'entrée et d'un neurone de sortie ($w_1 = w_2 = w_3 = w_4 = S = 0$) et la base d'apprentissage :