

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

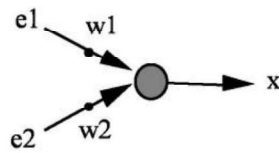


Figure 2. Réseau de 3 neurones (les 2 entrées sont considérées comme deux neurones) pour la résolution du problème exprimé table 2.

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) :

$$3/ \quad a = w_1 \cdot e_1 + w_2 \cdot e_2 - S = 0.0 \cdot 1 + 0.0 \cdot 1 - 0.0 = 0 \quad a \leq 0 \Rightarrow x = -1$$

4/ La sortie est fausse, il faut donc modifier les poids en appliquant :

$$w_1 = w_1 + e_1 \cdot x = 0.0 + 1 \cdot 1 = 1$$

$$w_2 = w_2 + e_2 \cdot x = 0.0 + 1 \cdot 1 = 1$$

2/ On passe à l'exemple suivant (2) :

$$3/ \quad a = 1 \cdot 1 + 1 \cdot -1 - 0.0 = 0 \quad a \leq 0 \Rightarrow x = -1$$

4/ La sortie est fausse, il faut donc modifier les poids en appliquant :

$$w_1 = 1 + 1 \cdot 1 = 2$$

$$w_2 = 1 + 1 \cdot -1 = 0$$

.../ 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 :

e ₁	e ₂	e ₃	e ₄	x
1	-1	1	-1	1
1	1	1	1	1
1	1	1	-1	-1
1	-1	-1	1	-1

Recherchez les valeurs de poids qui résolvent le problème.

Réponse : Cet algorithme d'apprentissage ne permet pas de trouver une solution à ce problème. Nous ne sommes capables d'exprimer une combinaison des activations en corrélation avec la sortie. Pourtant, il existe des solutions comme par exemple ($w_1 = -0.2$, $w_2 = -0.2$, $w_3 = 0.6$, $w_4 = 0.2$). Un algorithme de calcul efficace pour ce problème est l'apprentissage sur le modèle du Perceptron abordé au chapitre suivant.

Remarque : Il existe une possibilité de calculer les valeurs des connexions à partir des exemples, sans utiliser l'algorithme itératif. Si l'on initialise les poids à zéro et que l'on présente les exemples de la base d'apprentissage, la valeurs des poids à l'issue de l'apprentissage est :

$$w_{ij} = \sum_l x_{il} \cdot x_{jl} \quad \text{où } l \text{ est l'indice de l'exemple dans la base d'apprentissage}$$

2 La règle d'apprentissage du Perceptron, un exemple d'apprentissage supervisé

La règle de Hebb ne s'applique pas dans certain cas, bien qu'une solution existe (cf exercice du paragraphe précédent). Un autre algorithme d'apprentissage a donc été proposé, qui tient compte de l'erreur observée en sortie.

L'algorithme d'apprentissage du Perceptron est semblable à celui utilisé pour la loi de Hebb. Les différences se situent au niveau de la modification des poids.

- 1/ Initialisation des poids et du seuil S à des valeurs (petites) choisies au hasard.
- 2/ Présentation d'une entrée $E_l = (e_1, \dots, e_n)$ de la base d'apprentissage.
- 3/ Calcul de la sortie obtenue x pour cette entrée :

$$a = \sum (w_i \cdot e_i) - S$$

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

- 4/ Si la sortie x du Perceptron est différente de la sortie désirée d_l pour cet exemple d'entrée E_l alors modification des poids (μ le pas de modification) :

$$w_i(t+1) = w_i(t) + \mu \cdot ((d_l - x) \cdot e_i)$$

Rappel : $d_l = +1$ si E est de la classe 1, $d_l = -1$ si E est de la classe 2 et $(d_l - x)$ est une estimation de l'erreur.

- 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 de fonctionnement de l'algorithme d'apprentissage du Perceptron :

Base d'exemples d'apprentissage :

e_1	e_2	d	
1	1	1	(1)
-1	1	-1	(2)
-1	-1	-1	(3)
1	-1	-1	(4)

- 1/ Conditions initiales : $w_1 = -0.2$, $w_2 = +0.1$, $S = 0.2$ ($\mu = +0.1$)
- 2/ $a(1) = -0.2 + 0.1 - 0.2 = -0.3$
- 3/ $x(1) = -1$ (la sortie désirée $d(1) = +1$, d'où modification des poids)
- 4/ $w_1 = -0.2 + 0.1 \cdot (1 + 1) \cdot (+1) = 0$
 $w_2 = +0.1 + 0.1 \cdot (1 + 1) \cdot (+1) = +0.3$
- 2/ $a(2) = +0.3 - 0.2 = +0.1$
- 3/ $x(2) = +1$ Faux
- 4/ $w_1 = 0 + 0.1 \cdot (-1 - 1) \cdot (-1) = +0.2$
 $w_2 = +0.3 + 0.1 \cdot (-1 - 1) \cdot (+1) = +0.1$
- 2-3/ $a(3) = -0.2 - 0.1 - 0.2 = -0.5$ Ok
- 2-3/ $a(4) = +0.2 - 0.1 - 0.2 = -0.1$ Ok
- 2-3/ $a(1) = +0.2 + 0.1 - 0.2 = +0.1$ Ok
- 2-3/ $a(2) = -0.2 + 0.1 - 0.2 = -0.1$ Ok
- 5/ Tous les exemples de la base ont été correctement traités, l'apprentissage est terminé.

Le Perceptron réalise une partition de son espace d'entrée en 2 classes (1 et 2) selon la valeur de sa sortie (+1 ou -1). La séparation de ces deux zones est effectuée par un hyperplan (fig. 3).

L'équation de la droite séparatrice est :

$$w_1 \cdot e_1 + w_2 \cdot e_2 - S = 0$$