TD 4 – Réseaux de neurones

Exercice 1 :

1- Rappeler le principe de l'apprentissage basé sur les réseaux de neurones.
2- L'apprentissage est-il incrémental ?

Exercice 2 : Calcul élémentaire

Un neurone i reçoit des entrées de quatre autres neurones dont les signaux de sortie respectifs sont : 10, -20, 4 et -2. Les poids synaptiques associés à ces sorties sont respectivement : 0.8, 0.2, -1.0 et -0.9. Calculer la sortie du neurone i dans les situations suivantes (on supposera que le biais est nul) :

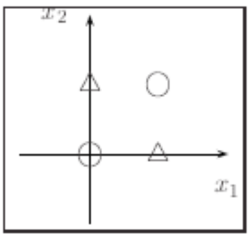
1. Le neurone i est linéaire.
2. Le neurone i a une fonction d'activation à seuil :

$$y_i = \begin{cases} 1 \text{ si } a_i \geq 0 \\ 0 \text{ sinon} \end{cases}$$

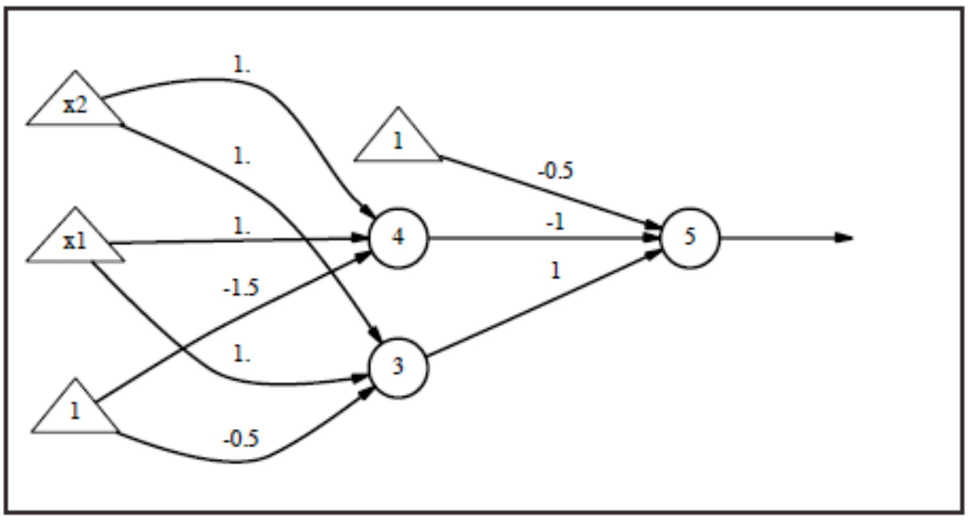
   où a_i dénote l'activation du neurone i (la somme pondéré de ses entrées).
3. Le neurone i est associé à une fonction d'activation logistique :

$$y_i = \frac{1}{1 + \exp(-a_i)}$$

Exercice 3 : XOR



Le problème XOR : les deux points △ sont des exemples de la même classe, les deux points ○ des exemples d'une autre classe. Les deux classes ne sont pas linéairement séparables.

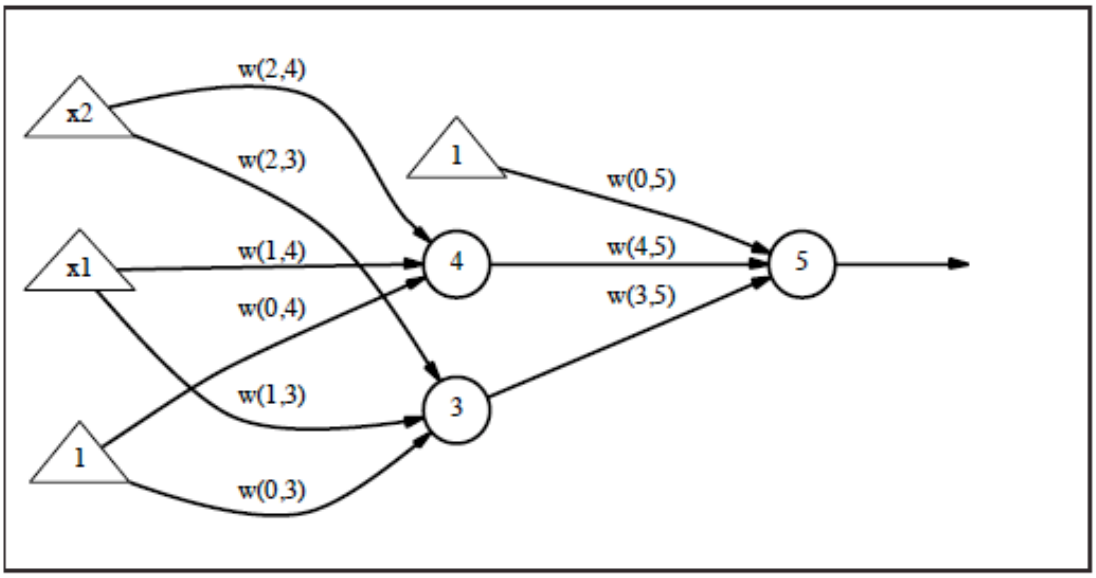


En utilisant le réseau de neurones ci-dessus, et la fonction seuil, compléter le tableau suivant :

x_1	x_2	σ_3	y_3	σ_4	y_4	σ_5	y_5
0	0						
0	1						
1	0						
1	1						

Exercice 4 : Rétro-propagation du gradient

Soit le réseau de neurones suivant.



On initialise les poids comme suit :

$$w(0,3) = 0.2 \quad w(1,3) = 0.1 \quad w(2,3) = 0.3$$
$$w(0,4) = -0.3 \quad w(1,4) = -0.2 \quad w(2,4) = 0.4$$
$$w(0,5) = 0.4 \quad w(3,5) = 0.5 \quad w(4,5) = -0.4$$

Pour xT = (1, 1), on obtient les valeurs suivantes, avec la fonction sigmoïde :

Neurone formel j	σ_j	y_j
3	$0.2 + 0.1 \times 1 + 0.3 \times 1 = 0.6$	$\frac{1}{1+e^{-0.6}} \simeq \mathbf{0.65}$
4	$-0.3 + -0.2 \times 1 + 0.4 \times 1 = -0.1$	$\frac{1}{1+e^{0.1}} \simeq \mathbf{0.48}$
5	$0.4 + 0.5 \times \mathbf{0.65} - 0.4 \times \mathbf{0.48} = 0.53$	$\frac{1}{1+e^{-0.53}} \simeq 0.63$

1- La sortie désirée (réelle) pour $x^T = (1,\ 1)$ est $u = 0$. On souhaite appliquer l'algorithme de rétropropagation du gradient (voir en annexe) pour modifier les poids du RNA.
   a. Calculer δ_5
   b. On fixe $\alpha = 1$. Calculer les nouveaux poids w(i, j).
2- En utilisant les nouveaux poids :
   a. Calculer la nouvelle valeur de y_5. Est-elle inférieure ou supérieure à la précédente ?
   b. L'algorithme va-t-il convergé ? (sortie obtenue y très proche de la sortie désirée u)

Exercices supplémentaires

Exercice S1 :

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	low	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

Etant donné l'échantillon d'apprentissage ci-dessus, Utiliser Weka pour apprendre un réseau de neurones multicouches, et indiquer la classe de l'objet X suivant ?

$$\mathbf{X} = (age = youth, income = medium, student = yes, credit = fair)$$

Exercice S2 :

sex	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

Déterminer la classe de l'objet suivant :

sex	height (feet)	weight (lbs)	foot size(inches)
?	6	130	8

Annexes :

Algorithme 10.2 Apprentissage du perceptron multicouche.

tant que l'apprentissage n'a pas convergé **faire**
  tirer au hasard un point d'apprentissage
  pour chaque couche, en partant de celle du haut **faire**
    pour chaque neurone formel de cette couche **faire**
      calculer δ_j
      pour chaque connexion $w(i,j)$ menant au neurone formel j **faire**
        calculer $\Delta w(i,j) = \alpha \, \delta_j \, y_i$
      fin pour
    fin pour
  fin pour
  pour chaque connexion $w(i,j)$ **faire**
    $w(i,j) \leftarrow w(i,j) + \Delta w(i,j)$
  fin pour
fin tant que

Calcul de δ_j :

$$\delta_j = (u_j - y_j) \; y_j \; (1 - y_j)$$