Eléments de réponses: Série 2

Q1) - L'apprentissage supervisé dispose d'un observateur (ou professeur) la mise a jours des poids se fait en dans le but de minimiser l'erreur entre les sorties du RNA (sorties estimées) et les données réelles.

Exemple: Prédiction de valeurs boursières, charge électrique, etc.

L'apprentissage non supervisé se base sur une mise a jours des poids en fonction des entrées et des sorties du RNA seulement. Les données réelles équivalentes dans ce cas n'existent pas.

Exemple: classification ou clustering

- L'apprentissage amélioré, affecte l'algorithme d'apprentissage notamment influe sur le pas d'apprentissage pour éviter les minimum locaux. Les principales méthodes d'apprentissage amélioré sont:
-
    - La méthode du moment
    - La méthode du pas d'apprentissage adaptatif
    - Mise a jour des nouvelles entrées
    - Techniques d'ordre supérieur

Q2) les principales différences entre la règle delta et le perceptron sont:

- La règle du perceptron est applicable a une neurone de type signe, la règle delta a besoin de neurones a fonctions d'activations continue différentiable (sigmoid).
- La règle delta utilise une méthode basée sur la décente du gradient (minimisation de l'erreur quadratique approximation de l'erreur moyenne quadratique) pas le perceptron.
- La règle delta est généralisable au Perceptron Multi Couche.

Q3) Les entrées du réseau peuvent êtres les 64 états des pixels, 1 pour blanc et -1 pour noirs. Les pixels noirs peuvent avoir comme poids correspondant des w=-1, et les pixels blanc des poids mis a 1. Un 3 parfait donnera un produit X*W=64, il suffira de mettre le biais a b=-63, le résultat donnée par le perceptron avec une fonction seuil sera de 1. Il sera négatif ou égale à 0 pour tout autre représentation du 3.

Q4)

1. Le problème fig 1, est non linéairement séparable, il faut au moins deux droites pour séparer les deux types de données. Voir Fig 2.
2. Non il n'est pas possible de résoudre le problème avec un neurone de type perceptron. Ce dernier n'implémentera qu'une ligne séparatrice. Il faudra pour solutionner le problème, au moins, deux neurones de type perceptrons (neurones 1 et 2) plus un neurone linéaire (neurone 3) pour la combinaison Figure 3.
3. Le réseau est donné par la Figure 3.

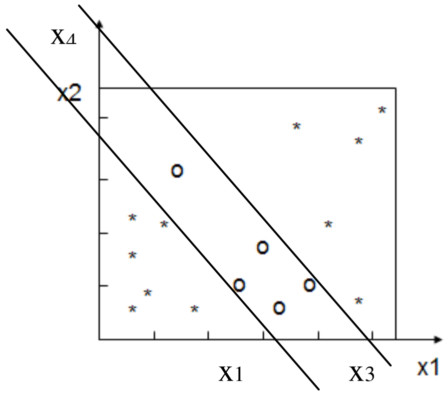

Fig 2

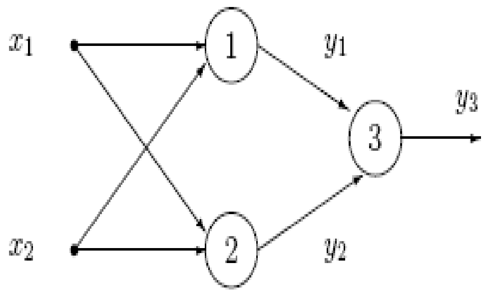

Fig 3

La décision du neurones 1 est donnée par la droite: $w_{11}\, x_1 > w_{12}\, x_2 + b_1$
La décision du neurones 2 est donnée par la droite: $w_{21}\, x_4 > w_{22}\, x_3 + b_2$

Les poids et les biais w et b peuvent être calculés en prenant plusieurs points sur chaque droite de décisions, créant un systèmes équations a 2 ou 3 inconnues.

La combinaison des deux droites peut être donnée par: $w_{31}\, y_1 + w_{32}\, y_2 > b_3$ ou y_1 et y_2 sont les résultats du neurones 1 et 2 respectivement.

Q5) La table de vérité de la fonction `A et ¬B` est donné par:

A	B	¬B	A et ¬B
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

1. La fonction `A et ¬B` est linéairement séparable, un perceptron associé à la règle d'apprentissage du perceptron est capable d'implémenter la fonction.
2. L'apprentissage delta est donné dans les notes de support du cours théorique, page 33 et 34. Si on choisit une fonction d'activation de type logarithme sigmoidal le regle delta de mise à jour des poids est donnée par l'équation (2.68).

Soit les entrée: $X_0=[0\ 0]$; $X_1=[0\ 1]$; $X_2=[1\ 0]$; $X_3=[1\ 1]$; $Y=[0\ 0\ 1\ 0]$.

$W_1= W_0+ (Y_0- \log([X_0*W_0)(1- (\log(X_0*W_0)^2)*X_0+b_0$

Cette mise à jour sera répéter pour les 4 vecteurs entrés X_0 à X_3 pour faire une époque. Il faudra renouveler l'opération pour faire l'apprentissage de deux époques.

<u>Un exemple pour l'entrée $X_0=[0\ 0]$, Y=0</u>

3. W=[1 -2], b=-1. pour une fonction seuil Y=1 pour tout entrée Z>=0.

Q6)

1. Le problème du OU logique est un problème linéairement séparable. Un seul neurone implémentant une ligne de décision avec une règle d'apprentissage du perceptron et une fonction signe, est suffisant pour classifier les sorties.

2. *Les poids et les biais égalent a 1, μ=1*

   Vecteur d'entrée *X=[0 0], Y=0*
   La sortie du réseau est égale a *: y=sgn(X*W-1)=sgn(-1)=0*

   Pas de mise à jour

   Vecteur d'entrée *X=[0 1], Y=0*
   La sortie du réseau est égale a : *y=sgn(X*W-1)=sgn(0)=1*

$$\begin{bmatrix} w_1^1 \\ w_2^1 \\ b^1 \end{bmatrix} = \begin{bmatrix} w_1^0 \\ w_2^0 \\ b^0 \end{bmatrix} + \mu\, e \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - 1 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

   Vecteur d'entrée *X=[1 0] Y=0*
   La sortie du réseau est égale a : *y=sgn(X*W-0)=sgn(1)=1*

$$\begin{bmatrix} w_1^2 \\ w_2^2 \\ b^2 \end{bmatrix} = \begin{bmatrix} w_1^1 \\ w_2^1 \\ b^1 \end{bmatrix} + \mu\, e \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - 1 \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

   Vecteur d'entrée *X=[1 1], Y=1*
   La sortie du réseau est égale a : *y=sgn(X*W+1)=sgn(1)=1*

   e=0 pas de mise a jour

Une époque est suffisante pour s'apercevoir que les poids et biais initiaux suffisent à garantir une solution au problème du OU.