

## Eléments de réponses: Série 3

Q1) La table de vérité de la fonction implémenter par le ou inclusif est donné par:

A	B	A ou inclusif B
0	0	1
0	1	0
1	0	0
1	1	1

1. le RNA qui résout le problème est donné par un réseau composé de deux neurones de type perceptrons (neurones 1 et 2) plus un neurone linéaire (neurone 3) pour la combinaison. Ceci implémentera deux droite de séparation, étend donnée que le problème du ou inclusif n'est pas linéairement séparable.
2. L'apprentissage par rétro propagation de l'erreur, se passe en deux phases alternées.
  - Dans la première, on présente un exemple à l'entrée, et on propage les signaux de couche en couche jusqu' à la sortie, à travers les poids connectant les sorties des neurones d'une couche à ceux de la couche suivante. On peut alors mettre en évidence les erreurs apparaissant sur les neurones de sortie.
  - Dans la seconde phase, on applique l'algorithme de mise à jour, dans la dernière couche de poids, puis on propage les erreurs de sortie à travers cette couche de poids sur l'avant dernière couche, ce qui permet alors de réitérer l'algorithme en mettant à jour l'avant-dernières couche de poids, et ainsi de suite.
3. Pour le neurone de sortie la règle de mise à jour est donnée par l'équation suivante:

$$\Delta w_{ji} = -\mu \frac{\partial J}{\partial w_{ji}} = 2\mu e_j f'_j(z_j) y_i = \mu \delta_j y_i$$

4. Pour les neurones de la couche cachée la règle de mise à jour est donnée par l'équation suivante:

$$\Delta w_{ji} = 2\mu f'_j(z_j) \frac{\partial J}{\partial w_{ji}} = \mu f'_j(z_j) \sum_{k=1}^M \delta_k w_{kj} y_i$$

Avant d'appliquer chaque règles de mise à jour, on doit d'abord calculer la sortie du réseau pour une fonction d'activation tangente sigmoïdal pour les neurones de la couche cachée donnée par:

$$f_t(z) = \frac{2}{1 + e^{-\beta z}} - 1$$

Soit les vecteur poids suivants,  $W_1=[W_{11} \ W_{12}]$ ,  $W_2=[W_{21} \ W_{22}]$ ,  $W_3=[W_{31} \ W_{32}]$ .

### Vecteur d'entrée $X=[0 \ 0]$

La sortie du réseau est égale a :

$$y_1 = f_1(X*W_1-0.1) = -0.5250; \quad y_2 = f_2(X*W_2-0.1) = -0.5250$$

$$y = W_{31}*f_1(X*W_1-0.1) + W_{32}*f_2(X*W_2-0.1) - 0.1 = -0.5511$$

### Mise a jour des poids $W_3$ :

$$W_3(1) = W_3(0) + 2*e*(1+y^2)* [y_1 \ y_2] = [-0.1 \ -0.1]^T + 2*(1+0.5511)*(1+0.5511^2)* [-0.5250 \ -0.5250]^T = [-2.2233 \ -2.2233]^T$$

### Mise a jour des poids $W_1$ :

$W_{11}(1) = W_{11}(0) + (1+y_1^2)* y_1 * (1+y^2) * e * W_{31} * X_1 = W_{11}$ , pas de mise a jours car  $X_1=0$  est bien classifié comme sortie

Le résultat sera similaire pour  $W_{12}$ .

$$W_{12}(1) = W_{12}(0) + (1+y_1^2)* y_1 * (1+y^2) * e * W_{31} * X_2 = W_{12}$$

### Mise a jour des poids $W_2$ :

$$W_{21}(1) = W_{21}(0) + (1+y_2^2)* y_2 * (1+y^2) * e * W_{32} * X_1 = W_{21}(0)$$

$$W_{22}(1) = W_{22}(0) + (1+y_2^2)* y_2 * (1+y^2) * e * W_{32} * X_2 = W_{22}(0)$$

### Vecteur d'entrée $X=[0 \ 1]$

Toutes les étapes précédentes sont a réitérer (refaire) avec les nouvelles valeurs de  $W_1$ ,  $W_2$ ,  $W_3$ . Ainsi que pour les deux autres entrées  $X=[1 \ 0]$  et  $X=[1 \ 1]$ .

Le problème est solutionné en 5 époques.

**Q2)** Le phénomène est appelé sur apprentissage, le réseau tend à apprendre les données d'apprentissage par cœur, il en résulte un faible pouvoir de généralisation.

Pour le détecter il faut appliquer un apprentissage supervisé sur l'erreur de validation et arrête l'apprentissage quand cette dernière commence a croître. Arrêt prématuré.

**Q3)** Le phénomène est appelé sur apprentissage, le réseaux tend a apprendre les données d'apprentissage par cœur, il es résulte un faible pouvoir de généralisation.

Pour le détecter il faut appliquer un apprentissage supervisé sur l'erreur de validation et arrête l'apprentissage quand cette dernière commence a croître. Arrêt prématuré.

1. A l'époque 1000 on remarque que l'erreur de validation est élevé par rapport a son minimum qui se trouve autour de l'époque 500. Ceci traduit le sur apprentissage du réseaux, il en découlera des performances amoindries voir médiocres.
2. Il est peut probable que l'erreur décrois, a moins que le minimum atteint a l'époque 500 ne soit que un minimum local, et que l'apprentissage tombe sur le minimum global. Mais il y a peu de chance que cela se passe, car le réseau est déjà en sur apprentissage.
3. Si malgré l'apprentissage *split sample*, les performances demeure médiocres, il faut soit modifier la topologie du réseau ou la méthode d'apprentissage.

**Q4)**

1. Voir support de cours pages 41-42-43
2. L'équation de mise à jour comptera une sommation de plus, car la sortie du neurone en questions affecte toute les sorties de la deuxième couche cachée. Deux couches impliqueront deux sommes, trois couche trois sommes, etc.