

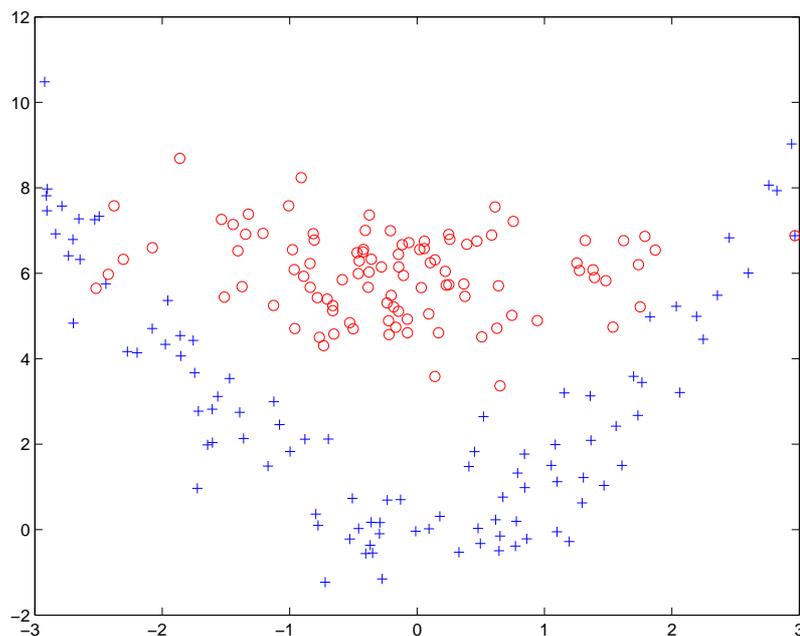
# Rétropropagation

Soizic GESLIN  
Maxime CHAMBREUIL

1<sup>er</sup> avril 2003

## 1 Explication du réseau utilisé

Nous avons utilisé un réseau de neurones comportant une couche cachée, et ayant 2 entrée et 1 sortie. Les 2 entrées sont les variables  $x_1$  et  $x_2$  qui permettent de placer les points en 2 dimensions, et la sortie est la classe d'appartenance de nos points ( $y=1$  ou  $-1$  selon points bleus ou rouges).



Notre but est donc d'apprendre le modèle pour séparer les points rouges des points bleus, ce qui ne peut pas se faire avec une fonction simple. Nous utilisons donc un perceptron multicouche.

## 2 Le code

### 2.1 Fonctions

```
function [w1 w2] = mlpinit(m)
```

Initialise aléatoirement les poids d'un perceptron multicouche à  $m$  neurones dans la couche cachée,  $w1$  et  $w2$  sont les poids des arcs, leurs dimensions permettent d'avoir 2 entrées plus le biais et 1 sortie, et de correspondre aux  $m$  neurones (matrices  $m \times 2 + 1$  et  $1 \times m + 1$ ).

```
function [y] = mlpvale(x, w1, w2)
```

Calcule la sortie d'un perceptron multicouche défini par ces matrices de paramètres  $w1$  et  $w2$

```
function [w1 w2] = mlpfit(xi, yi, w1, w2)
```

Estime les paramètres d'un perceptron multicouche à partir de l'échantillon ( $x_i$ ,  $y_i$ )

### 2.2 Réglage des pas

Nous avons choisi la méthode Rprop pour régler les pas au fur et à mesure des itérations.

Ces pas étaient réglés à la main avant, et un bon réglage dépendait du type de points à classifier.

Le réglage dynamique des pas fait appel à 2 paramètres :  $u$ , pour l'accélération, et  $d$  pour le ralentissement. Nous avons donc du régler ces paramètres, en différenciant le réglage du pas1 de  $w1$  et du pas2 de  $w2$  !

Nous avons ainsi apporté 4 paramètres qui sont alors indépendants de la distribution à classifier, ils s'appliquent aussi bien pour le problème de 'la pomme et la banane' que pour celui du 'carré coupé en quatre'.

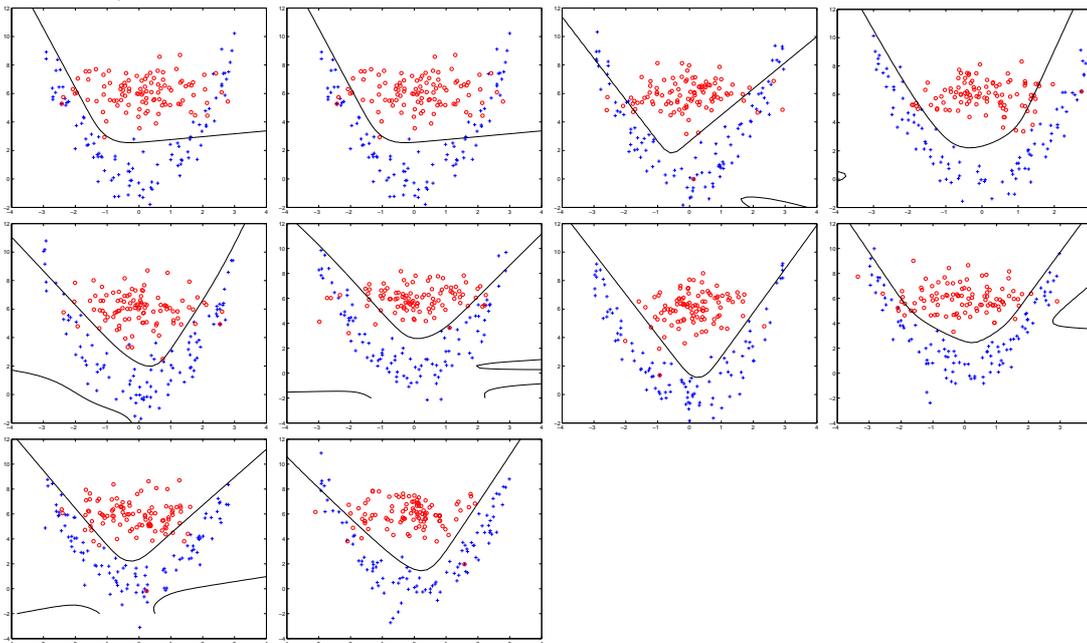
## 3 Résultats

### 3.1 La pomme et la banane

Critère d'arrêt : nombre d'erreurs.

Nombre de neurones	Nombre d'erreurs	Nombre d'itérations
1	30	pas de résultat
2	30	121
3	30	18
10	30	9
4	15	15
5	15	26
10	15	29
2	10	194
4	10	1530
6	10	16
10	10	53

Voici les tracés, dans le même ordre que dans le tableau (sauf pour 1 neurone bien sur) :

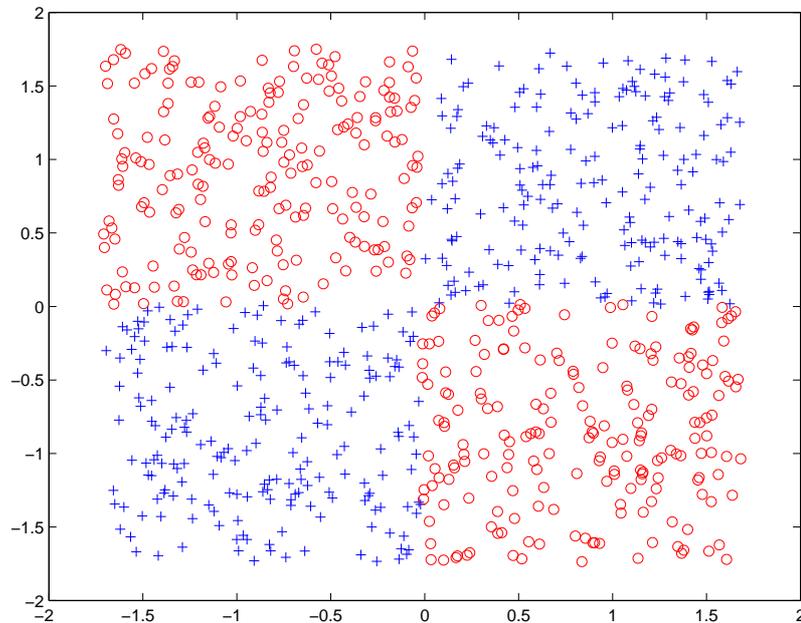


Nous déduisons de ces différents essais que :

- Suivant la distribution des points, on n'est pas forcé d'avoir une solution, ou bien cette solution n'est pas comparable aux autres, si le seul critère d'arrêt est le nombre d'erreurs.
- Nous ne pouvons pas trouver de relation de proportionnalité selon les nombres de neurones ou d'itérations car les résultats sont trop dépendants de la distribution de départ.
- Plus il y a de neurones, plus il y a de coefficients à régler, et donc plus le nombre d'itérations est important.

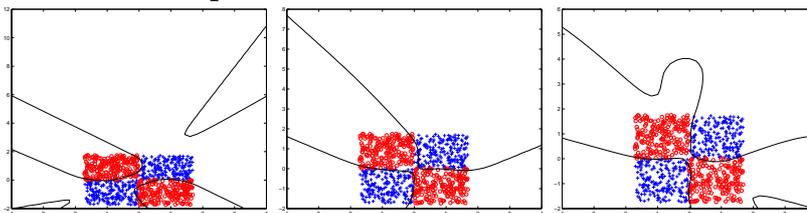
(affichez à chaque fois : le nombre d'itérations, l'erreur quadratique moyenne et le taux de bonne classification)

### 3.2 Le carré coupé en quatre



Nombre de neurones	Nombre d'erreurs	Nombre d'itérations
5	60	...
6	40	342
7	37	57
10	20	40

Tracés (sauf pour 5 neurones et 60 erreurs) :



**Conclusion :**

Pour ce problème, le perceptron multicouche n'est pas efficace en dessous de 6 neurones et selon le nombre d'erreurs. Nous pouvons atteindre 2.5% d'erreurs avec 10 neurones en un nombre raisonnable d'itérations.