Modèles Linéaires

Correction Séance 7
Licence 3 MIASHS (2022-2023)

Guillaume Metzler, Francesco Amato
Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr ; francesco.amato@univ-lyon2.fr

Résumé

Dans cette séance, nous allons traiter les exercices de la fiche de TD 5. On verra les détailles pour codifier le réseau des neurones du point 2.5 du TD en R.

Les détails théoriques exploités pour ces deux exercices sont présentés dans l'énoncé de la fiche de TD, directement disponible sur le site de Stéphane Chrétien.

Rappel : Objectif : classification !

Notre but est développer un classifieur pour pour classifier les points de données de la catégorie cercles rouges et de la catégorie croix bleues. Par exemple, on peut imaginer que les données peuvent montrer des sites de forage pétrolier sur une carte, où la catégorie A (cercles rouges) indique un résultat négatif et la catégorie B (croix bleues) inique un résultat positif, c'est-à-dire on a trouvé du pétrole là-bas. On veu créer un classifieur pour prédire la catégorie d'un point future et obtenir une frontière de décision pour "délimiter" la zone pétrolière.

Donc, en utilisant un réseau comme lequel en figure 1, la fonction descriptive du réseau sera :

$$F(x) = \sigma\left(W^{[4]}\sigma\left(W^{[3]}\sigma\left(W^{[2]}x + b^{[2]}\right) + b^{[3]}\right) + b^{[4]}\right) \in \mathbb{R}^2.$$

0.1 Données

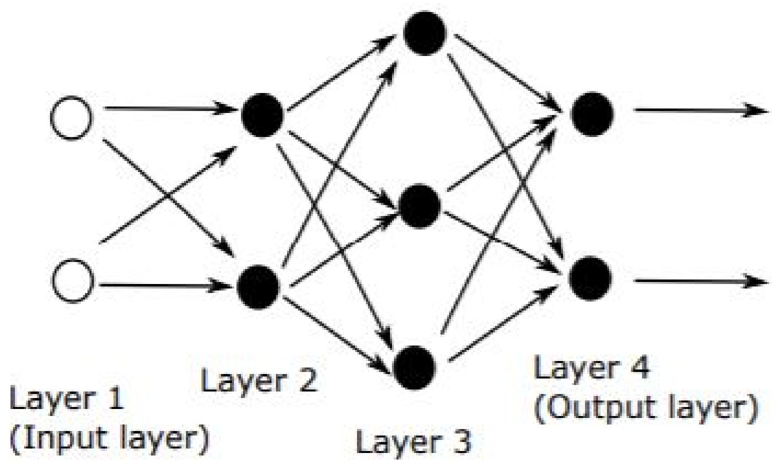
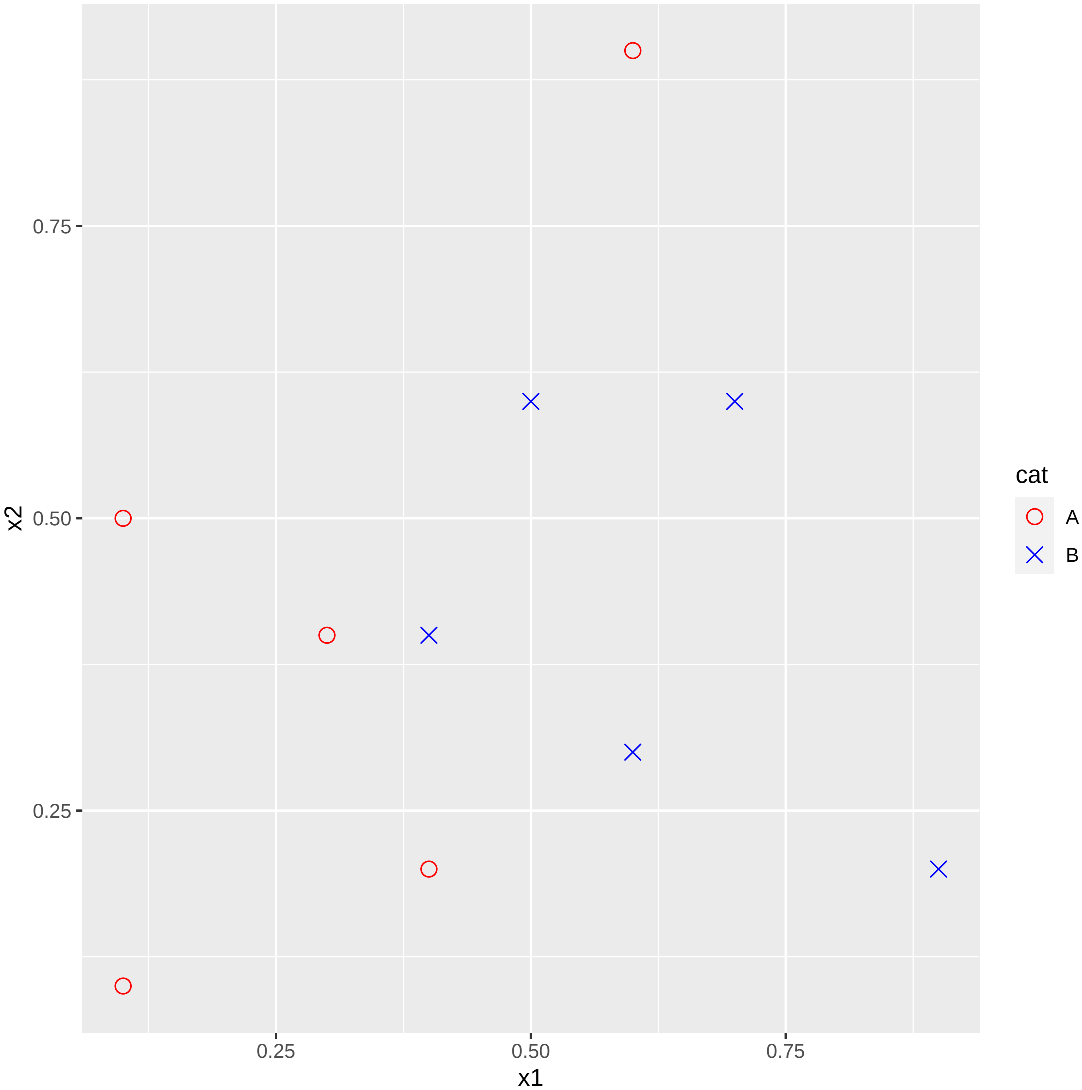


Figure 1 – Un réseau de neurones très simple

```
# Define data
x1 <- c(0.1,0.3,0.1,0.6,0.4,0.6,0.5,0.9,0.4,0.7)
x2 <- c(0.1,0.4,0.5,0.9,0.2,0.3,0.6,0.2,0.4,0.6)
y <- matrix(c(rep(1,5), rep(0,5), rep(0,5), rep(1,5)), nrow=2,byrow = T)
```

On va plotter les données

```
## Plot
x_plot = data.frame(x1,x2,cat = c(rep('A',5),rep('B',5)))
library(ggplot2)
ggplot(x_plot, aes(x = x1, y = x2, color = cat, shape = cat)) +
  scale_shape_manual(values = c(1, 4)) +
  scale_color_manual(values=c('red','blue')) +
  geom_point(size=3)
```



On doit ensuite definir la fonctione de activation

```r
# Define function for sigmoid activation
activate <- function(x, W, b) {
  y <- 1/(1 + exp(-(W %*% x + b)))
  return(y)
}
```

Il nous reste à calculer le numérateur et le dénominateur qui servent à définir notre statistique de test r pour la déterminer complètement

```r
# a2 <- activate(x, W2, b2)
# a3 <- activate(a2, W3, b3)
# a4 <- activate(a3, W4, b4)

 # Define function for cost
cost <- function(W2, W3, W4, b2, b3, b4) {
  costvec <- numeric(10)
  for (i in 1:10) {
    x <- c(x1[i], x2[i])
    a2 <- activate(x, W2, b2)
    a3 <- activate(a2, W3, b3)
    a4 <- activate(a3, W4, b4)
    costvec[i] <- norm(y[,i] - a4, "2")
  }
  costval <- norm(costvec, "2")^2
  return(costval)
}
```

```r
# Borne inférieure de l'intervalle de confiance
borne_inf = qf(0.025,76,3)
# Borne supérieure de l'intervalle de confiance
borne_sup = qf(0.975,76,3)

# Il reste à faire le test
ifelse(((r<borne_inf)|(r>borne_sup)),"On rejette H0",
"On ne rejette pas H0")

## Error in ifelse(((r < borne_inf) | (r > borne_sup)), "On rejette H0", :
object 'r' not found
```

Le test nous conduit donc à rejeter l'hypothèse H_0. On peut donc en conclure qu'au moins deux ateliers sont exposés à des degrés différents.