Filière ingénieure - ING2-MI

Fouille de Données T.P. N^o 4

Réseaux de neurones

23 octobre 2023

1 Apprendre la fonction AND

Nous supposons dans cette section que la fonction d'activation est la fonction Heaviside.

On construit un perceptron pour la porte logique AND avec les caractéristiques suivantes :

- les entrées sont binaires et les sorties bipolaires ;
- le taux d'apprentissage est $\mu = 1$;
- le vecteur de poids initaux $W = [3; 2; 3]$ où la première composante correspond au biais.

Exercice 1 Appliquer l'algorithme du perceptron.

Exercice 2 Donner le plan séparateur de la solution.

Exercice 3 Calculer pour chaque exemple sa distance au plan séparateur,

2 Réseau de neurones pour le XOR

Exercice 4 Démontrer qu'un seul neurone (perceptron) est suffisant pour calculer la fonction logique $x_1 \wedge \neg x_2$

Exercice 5 Trouver un réseau de neurones qui calcule la fonction logique XOR en utilisant le perceptron précédent ainsi que les deux perceptrons qui calculent respectivement les fonction AND et OR vus en cours, justifier votre réponse.

3 Etude pratique avec R

Le but de cette section est de se familiariser avec le package Neuralnet :

1. **construction du dataset XOR**

   ```
   library(neuralnet)
   XOR <- c(0,1,1,0)
   xor.data <- data.frame(expand.grid(c(0,1), c(0,1)), XOR)
   ```

2. **Construction d'un réseau de neurone adapté** Construction d'un réseau de neurone avec une couchée contenant 2 neurones. Regarder la

   ```
   net.xor <- neuralnet( XOR~Var1+Var2, xor.data, hidden=2, rep=5)
   print(net.xor)
   plot(net.xor, rep="best")
   ```

3. **Caractéristique du réseau construit :** Observer la fonction d'activation, la fonction erreur ; les données et les résultats

   ```
   net.xor$act.fct
   net.xor$err.fct
   net.xor$data
   net.xor$net.result
   ```

Exercice 6 Effectuer la commande suivante pour examiner les paramètre du réseau construit.

```
net.xor$result.matrix
```

Relever les valeurs des erreurs, les nombres d'époque ainsi que les taux d'apprentissage μ à partir des trois premières ligne de la matrice affichée.

Exercice 7 Calculer la sortie du réseau pour l'exemple (1,0).

3.1 Etude du dataset IRIS

1. **Exploration de l'ensemble des données IRIS** :
2. **Caractéristiques** :

```
dim(iris)
names(iris)
str(iris)
attributes(iris)
```

3. **Ajout de trois variables booléennes** :

```
iris$setosa <- iris$Species=="setosa"
iris$virginica <- iris$Species == "virginica"
iris$versicolor <- iris$Species == "versicolor"
```

4. **Centrer et réduire les données**

```
mean=apply(iris[,1:4],2,mean)
sd=sapply(iris[,1:4], sd)
data.scaled = scale(iris[,1:4], center = mean, scale = sd) # Scale the data
iris[,1:4]=data.scaled
```

5. **Devision en train et test**

```
iris.train.idx <- sample(x = nrow(iris), size = nrow(iris)*0.75)
iris.train <- iris[iris.train.idx,]
iris.valid <- iris[-iris.train.idx,]
```

6. **Construction d'un réseau de neuone**

```
iris.net <- neuralnet(setosa+versicolor+virginica ~
Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
data=iris.train, hidden=5, rep = 5,
linear.output = F)
plot(iris.net, rep="best")
```

7. **Prédictions et matrices de confusion**

```
iris.prediction <- compute(iris.net, iris.valid[1:4])
idx <- apply(iris.prediction$net.result, 1, which.max)
predicted <- c('setosa', 'versicolor', 'virginica')[idx]
table(predicted, iris.valid$Species)
```

Exercice 8 Relever l'erreur, le nombre d'époque ainsi que le taux d'apprentissage pour le meilleur réseau parmi les 5.

Exercice 9 Construire un nouveau réseau avec deux couches cachées la première contenant 5 neurones et la deuxième 2 neurones. Comparer les résultats sur l'ensemble d'apprentissage et de validation avec ceux du réseau précédent.

Exercice 10 Refaire le même travail décrit dans cette partie pratique sur le dataset *PimaIndiansDiabetes* de la libraire *mlbench*