

## TP4 : Réseaux de neurones

Nous allons manipuler dans ce TP des réseaux de neurones dans les cas d'une classification et d'une régression. Nous aurons besoin de librairie *neuralnet*

---

```
>library(neuralnet)
```

---

### Exercice 1 : Exemple introductif

On voudrait construire un réseau de neurone capable d'approximer la fonction racine carrée. Cela est un exemple simple pour faire par la suite de l'apprentissage dans des situations plus complexes.

1. Générer 50 nombres aléatoires uniformément entre 0 et 100.

---

```
train.input <- as.data.frame(runif(50, min=0, max=100))
train.output <- sqrt(train.input)
data.train <- cbind(train.input, train.output)
colnames(data.train) <- c("Input", "Output")
```

---

2. Utiliser la librairie neuralnet pour entraîner un réseau constitué de 10 neurones cachés. Consultez le help de neuralnet pour comprendre chaque paramètre utilisé

---

```
net.sqrt <- neuralnet(Output ~ Input, data= data.train,
hidden=10, threshold=0.01)
net.sqrt$result.matrix
plot(net.sqrt)
```

---

3. Quel est le taux d'erreur du réseau sur l'ensemble d'apprentissage ?
4. Analyser le réseau affiché avec plot.
5. Testez le réseau obtenu sur les données suivantes avec la fonction *compute*

---

```
data.test <- as.data.frame((1:10)^2)
prediction <- compute(net.sqrt, data.test)
```

---

6. Quelles sont les propriétés de l'objet *net.results*? Affichez les.

---

```
ls(prediction)
print(prediction$net.result)
```

---

7. Constituer un tableau similaire au suivant en utilisant la fonction *cbind* :

Entrée	Sortie attendue	Sortie du réseau neuronal
1	1	0.9623402772
4	2	2.0083461217
9	3	2.9958221776
16	4	4.0009548085
25	5	4.0009548085
36	6	5.9975810435
49	7	6.9968278722
64	8	8.0070028670
81	9	9.0019220736
...	...	...

## 1 Exercice 2 : réseau de neurones pour la classification

Nous allons utiliser un réseau de neurone pour faire de l'apprentissage sur la base de données des iris vu dans les TPs précédent.

1. Charger les données "*iris.csv*" disponibles sur le site de l'enseignant.
2. Normaliser les données.
3. Séparer les données en échantillon d'apprentissage *iris.app* (sur lequel le modèle sera entraîné) et un échantillon de test *iris.test* (sur lequel on mesurera les écarts entre prévisions et valeurs réellement observées). Les tailles respectives de ces deux jeux seront de 70% et 30%.
4. Binariser les catégories de l'ensemble d'apprentissage : au lieu d'avoir la colonne *Species* = *setosa*, on aura trois colonnes : *setosa* = *TRUE*, *virginica* = *FALSE* et *versicolor* = *FALSE*. Pour rajouter une nouvelle colonne *newColumn* au data frame *df* :

---

```
df$newColumn <- df$Column == "A"
```

---

Cela signifie que chaque ligne aura pour valeur *TRUE* si *data\$Column* a pour valeur *A*. Pour supprimer la colonne *Column* du data frame *df* :

---

```
df$Column <- NULL
```

---

5. Entraîner un réseau de neurones, avec une seule couche de 3 neurones cachés.
6. Afficher le réseau avec plot
7. Calculer la prédiction *prediction* de l'échantillon de test *iris.test* avec la fonction *compute*.
8. Utiliser le résultat de la prédiction *prediction\$net.result* et la fonction *which.max* pour déduire les labels des prédictions.
9. Comparer le résultat de la prédiction avec les valeurs réelles.
10. Afficher la matrice de confusion. Quel est le taux d'erreur du réseau sur *iris.test*?
11. Essayez d'ajouter des couches supplémentaires et analyser le taux d'erreurs à chaque fois.

## 2 Exercice 3 : réseau de neurones pour la régression

Nous allons utiliser un réseau de neurone pour résoudre un problème de régression.

1. Charger et analyser la base de données "gasoline.csv". Dans cet exercice, nous souhaitons prédire la consommation annuelle d'un véhicule en fonction de :
  - **capacity** : capacité du réservoir de carburant du véhicule (en litres).
  - **gasoline** : coût moyen de l'essence.
  - **hours** : heures de conduite par an du propriétaire.
2. Normaliser et séparer vos données en un ensemble d'apprentissage *gasoline.app* et un ensemble de test *gasoline.test*
3. Entraîner un réseau de neurones avec les paramètres suivants :

---

```
nn <- neuralnet(consumption ~ capacity + gasoline + hours,  
data=gasoline.app, hidden=c(2,1), linear.output=TRUE,  
threshold=0.01)  
nn$result.matrix  
plot(nn)
```

---