MNIST ou le *Hello Word* de la classification d'images

Mohammed Sedki

Projet à rendre le 28/02/2018

MSP/ES *Apprentissage et agrégation de modèles*

---

1. Le jeu de données MNIST

Le jeu de données MNIST est hébergé sur le page web de Yann LeCun. Ce jeu de données sert de référence pour les compétitions en apprentissage où la donnée explicative est sous forme d'image. Commençons par une visualisation du jeu de données et plus précisémment les 36 premières images. Pour cela nous allons installer le package keras qui installe à son tour tensorflow. Le block de code suivant permet cette visualisation

```
#install.packages("keras", dep=TRUE)
require(keras)
mnist   <- dataset_mnist()
x_train <- mnist$train$x
y_train <- mnist$train$y
x_test  <- mnist$test$x
y_test  <- mnist$test$y

# visualize the digits
par(mfcol=c(6,6))
par(mar=c(0, 0, 3, 0), xaxs='i', yaxs='i')
for (idx in 1:36) {
  im <- x_train[idx,,]
  im <- t(apply(im, 2, rev))
  image(1:28, 1:28, im, col=gray((0:255)/255),
        xaxt='n', main=paste(y_train[idx]))
}
```

Préparation des données

Afin de pouvoir mettre en place les méthodes d'apprentissage classiques, nous avons besoin d'applatir les images en vecteurs. Cela revient à transformer une image de dimension 28×28 pixels en un vecteur de dimension 784.

1. Préparer le jeu de données en appliquant un applatissement ainsi qu'une suppression des pixels nuls pour l'ensemble des images du jeu de données. La fonction nearZeroVar du package caret permet de repérer les variables de variance nulle.

2. Le nombre de variables du jeu de données après applatissement est très elevé. Proposer une procédure de réduction de dimension indépendante de la méthode d'apprentissage à utiliser. La fonction prePocess du package caret permet un ensemble de transformation de données.

```
y_train.freq <- table(y_train)
barplot(y_train.freq)
```