TP06-MNIST

September 18, 2019

1 MNIST

According to wikipedia, the MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.

It consists in greyscale pictures of 28x28 size belonging to 1 out of 10 categories.

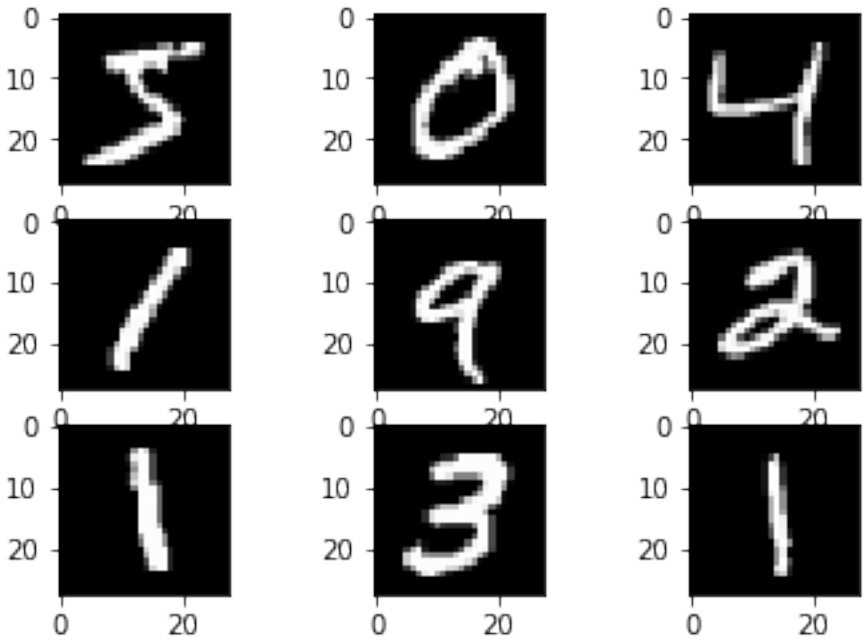
Let's play a little bit with it.

The following script downloads and prepares the MNIST database. Take care it is nearly 500MB, it will be saved in your `/tmp` folder.

[1]:
```
import numpy as np
import tensorflow as tf
from tensorflow import keras

import matplotlib.pyplot as plt
%matplotlib inline
import datetime as dt

from pathlib import Path

from sklearn.datasets import fetch_openml

data_home = '/tmp/scikit_learn_data/'
datafile = '/tmp/mnist.npz'

datapath = Path(datafile)
if not(datapath.exists()):
    print("Data File not found... downloading it")
    Xmnist, ymnist = fetch_openml('mnist_784',
                                  version=1,
                                  return_X_y=True,
                                  data_home=data_home)
    np.savez(datapath.as_posix(),
             X=np.array(Xmnist,dtype='u8'),
             y=np.array(ymnist,dtype='u8'))
    print("Data File downloaded and saved")
    del Xmnist, ymnist
```

```
print("Data File found... loading it into memory")
data = np.load(datapath.as_posix())
Xmnist = data['X']
ymnist = keras.utils.to_categorical(data['y'])
print("Data File loaded")

Xtrain, Ytrain = Xmnist[:50000], ymnist[:50000]
Xvalid, Yvalid = Xmnist[50000:60000], ymnist[50000:60000]
Xtest, Ytest = Xmnist[-10000:], ymnist[-10000:]

plt.figure()
for i in range(9):
    plt.subplot(3,3,i+1)
    plt.imshow(Xtrain[i].reshape((28,28)),cmap='gray', vmin=0, vmax=255)
plt.show()
```

```
Data File found... loading it into memory
Data File loaded
```



2 Exercises

Build a 2 layer MLP (28 hidden units) and fit it on MNIST

You can use `tanh` activation function on the hidden layer, and `softmax` activation function on the last layer. You should get near or above 90% of accuracy.