

TP07-MNIST-DNN

September 18, 2019

1 Deep Neural Networks on MNIST

Let's build a 5 layer MLP (with resp. 256, 128, 64, 28 hidden units) and fit it on MNIST.

- 1) How good is it in from a shallow 2 layer network ?
- 2) Where can we track vanishing gradient on Tensorboard ?

```
[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
from sklearn import preprocessing

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")

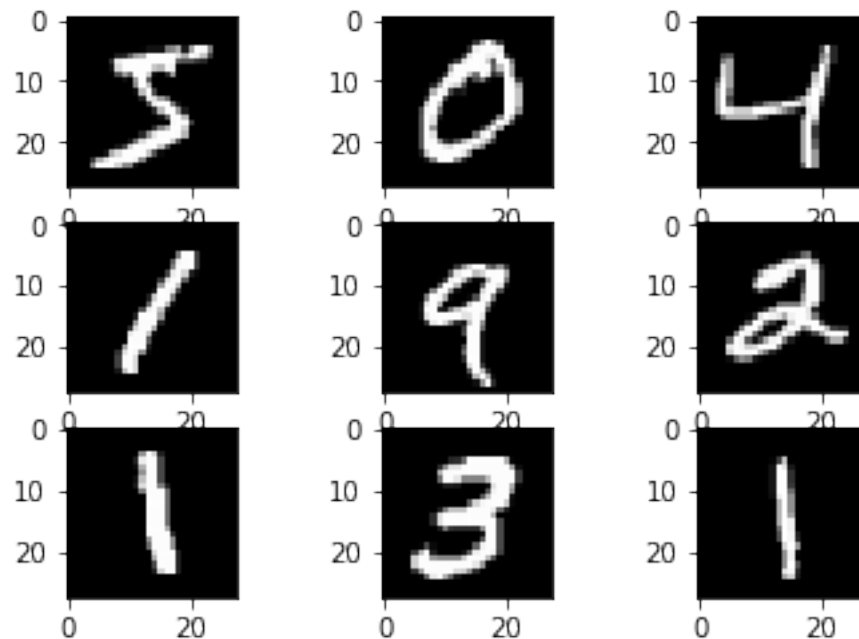
Xtrain, Ytrain = Xmnist[:50000], ymnist[:50000]
Xvalid, Yvalid = Xmnist[50000:60000], ymnist[50000:60000]
Xtest, Ytest = Xmnist[-10000:], ymnist[-10000:]

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()

scaler = preprocessing.StandardScaler().fit(Xtrain)
Xtrain = scaler.transform(Xtrain)
Xvalid = scaler.transform(Xvalid)
Xtest = scaler.transform(Xtest)

```

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



1.1 Preventing vanishing gradient

Vanishing gradient can be prevented by: - using weight regularization [doc](#). - using non saturating activation functions [simple](#) [advanced](#) - adding dropout [doc](#) - adding batch normalization [doc](#)

You can also play on the batch size and the learning rate.

- 1) Try to modify your code to include the aforementioned features.
- 2) Observe the effect of the modification on the vanishing gradient on?