

# Forward Propagation in the Neural Networks

```
In [0]: from google.colab import drive
drive.mount('/content/drive')
```

1. Plot the average image

```
In [56]: import matplotlib.pyplot as plt
import numpy as np
np.set_printoptions(linewidth=2000)

file_data = "/content/drive/My Drive/Colab Notebooks/assignment8/mnist_test.csv"
handle_file = open(file_data, "r")
data = handle_file.readlines()
handle_file.close()

size_row = 28 # height of the image
size_col = 28 # width of the image

num_image = len(data)
count = 0 # count for the number of images

#
# make a matrix each column of which represents an images in a vector form
#
list_image = np.empty((size_row * size_col, num_image), dtype=float)
list_label = np.empty(num_image, dtype=int)

for line in data:

    line_data = line.split(',')
    label = line_data[0]
    im_vector = np.asfarray(line_data[1:])

    list_label[count] = label
    list_image[:, count] = im_vector

    count += 1
    #print("count = ",count)

#
# plot first 100 images out of 10,000 with their labels
#
f1 = plt.figure(10, figsize=(6,9))
for i in range(100):

    label = list_label[i]
    im_vector = list_image[:, i]
    im_matrix = im_vector.reshape((size_row, size_col))

    plt.subplot(10, 10, i+1)
    plt.title(label)
    plt.imshow(im_matrix, cmap='Greys', interpolation='None')

    frame = plt.gca()
    frame.axes.get_xaxis().set_visible(False)
    frame.axes.get_yaxis().set_visible(False)

plt.show()
```



```
In [58]: num_label = 10;
label_count = np.full((1,10),0)
image_sum = np.empty((size_row * size_col, num_label), dtype=float)

for i in range(10):
    image_sum[:,i] = np.full((1,784),0.0)
#print(image_sum[:,0].reshape((28,28)))

for i in range(10):
    for j in range(count):
        if list_label[j]==i:
            label_count[:,i] += 1
            image_sum[:,i] += list_image[:,j]
        image_sum[:,i] /= label_count[:,i]
    plt.subplot(2,5,i+1)
    plt.title(i)
    plt.imshow(image_sum[:,i].reshape((size_row, size_col)), cmap='Greys', interpolation='None')

    frame = plt.gca()
    frame.axes.get_xaxis().set_visible(False)
    frame.axes.get_yaxis().set_visible(False)

plt.show()
#print(label_count)
#print(image_sum[:, 8].reshape((28,28)))
#print(list_image[:, 99].reshape((28,28)))
```

