

assignment03

March 28, 2019

1 Load MNIST training dataset

```
In [1]: #20160040 Ko Hoyun
        #Visualize average images(L2-norm)
        import matplotlib.pyplot as plt
        import numpy as np

        #my file data path
        file_data = "C:\\Users\\recognize_data\\mnist_train.csv"
        handle_file = open(file_data, "r")

        #read data with line
        data = handle_file.readlines()
        handle_file.close()

In [2]: #image size
        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

In [3]: #
        # normalize the values of the input data to be [0, 1]
        #
        def normalize(data):

            data_normalized = (data - min(data)) / (max(data) - min(data))

            return(data_normalized)

        #function for squaring and
        def square(x):
            d = x ** 2
            s = x
            return(s)
```

```

In [4]: #
        # 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)

        #2-D array for putting square values of im_vector with the same label
        collect = [[0 for col in range(1)] for row in range(10)]

        #for average. divisor
        n = np.array([0,0,0,0,0,0,0,0,0,0], dtype = int)
        for line in data:

            #the number of lables is at the front. so split and put it into lable value.
            line_data = line.split(',')
            label      = line_data[0]

            #remain values put in the im_vector
            im_vector  = np.asfarray(line_data[1:])
            im_vector  = normalize(im_vector)

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

            n[int(label)] += 1
            #putting square values of im_vector with the same label
            collect[int(label)][0] += square(im_vector)

            count += 1

In [5]: #average image
        for i in range(0,10):
            collect[i][0] = collect[i][0]/n[i]

In [6]: # insert average image value
        for i in range(0,10):
            list_image[:, i] = collect[i][0]

In [7]: #for i in range(0,10):
        # print(i, collect[i][0])

        #l2 norm(squre was done)
        def l2_norm(x):

            s = np.sum(x)
            r = np.sqrt(s)

            return(r)

```

2 Compute the average images for each label based on L2-norm.

```
In [8]: l2 = np.array([0,0,0,0,0,0,0,0,0,0], dtype = float)
```

```
    for i in range(0,10):

        l2[i] = l2_norm(collect[i][0])
        print(i, l2[i])
```

```
0 11.662078587357021
1 7.720221295572655
2 10.809595343502995
3 10.535599185520221
4 9.75653437658465
5 10.048953545326638
6 10.37743308278527
7 9.477759305724035
8 10.851858801667873
9 9.805094168101338
```

3 Visualize the average images.

```
In [9]: #
        # plot images
        #
        f1 = plt.figure(1)

        for i in range(0,10):

            label      = i
            collect[i][0] = list_image[:, i]
            im_matrix  = collect[i][0].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()
```



