

# Visualize Average Images

March 26, 2019

## 1 Mnist dataset

The Mnist database is a large database of handwritten digits that is commonly used for training various image processing systems. The MNIST database contains 60,000 training images and 10,000 testing images.

**Professor:** Hong Hyung-Woo

**Deadline:** 2019-03-28

**Student NB:** 20166450

**Major:** Software Engineering

**Name:** Kim Young Min

The goal is a visualization of average Images for each Labels.

### What we are going to do

- Load MNIST training dataset.
- Compute the average images for each label (digit) based on L2-norm.
- Visualize the average images.

```
In [397]: import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

## 1.1 Load MNIST training dataset.

```
In [386]: file_data      = "mnist_train.csv"
          handle_file    = open(file_data, "r")
          data           = handle_file.readlines()
          handle_file.close()

In [398]: 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
```

**normalize the values of the input data to be [0,1]**

```
In [388]: def normalize(data):

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

          return(data_normalized)
```

**Example of distance function between two vectors x and y**

```
In [389]: def distance(x, y):

          d = (x - y) ** 2
          s = np.sum(d)
          # r = np.sqrt(s)

          return(s)
```

**A matrix each column of which represents an images in a vector form**

```
In [391]: list_image = np.empty((size_row * size_col, num_image), dtype=float)
          list_label  = np.empty(num_image, dtype=int)

In [392]: for line in data:

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

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

count += 1
```

**Plot first 100 images out of 10,000 with their labels**

In [424]: f1 = plt.figure(1)

```
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()
```



## 1.2 Compute the average images for each label (digit) based on L2-norm.

- The basic principle used for average images is  $avg(x) = 1^T x / n$
- `avg_list_image` : List for vectors of the average images which is computed for each labels.
- `cnt1` : This is 'n' for each labels

```
In [405]: avg_list_image = [[0]*784]*10
```

```
cnt1=[0,0,0,0,0,0,0,0,0,0]
```

```
for i in range(60000):
    if(list_label[i]== 0):                                #label = 0
        avg_list_image[0] += list_image[:,i]
        cnt1[0]+=1
    elif (list_label[i]== 1):                             #label = 1
        avg_list_image[1] += list_image[:, i]
        cnt1[1]+=1
    elif (list_label[i]== 2):                             #label = 2
        avg_list_image[2] += list_image[:, i]
        cnt1[2]+=1
    elif (list_label[i]== 3):                             #label = 3
        avg_list_image[3] += list_image[:, i]
        cnt1[3]+=1
    elif (list_label[i]== 4):                             #label = 4
        avg_list_image[4] += list_image[:, i]
        cnt1[4]+=1
    elif (list_label[i]== 5):                             #label = 5
        avg_list_image[5] += list_image[:, i]
        cnt1[5]+=1
    elif (list_label[i]== 6):                             #label = 6
        avg_list_image[6] += list_image[:, i]
        cnt1[6]+=1
    elif (list_label[i]== 7):                             #label = 7
        avg_list_image[7] += list_image[:, i]
        cnt1[7]+=1
    elif (list_label[i]== 8):                             #label = 8
        avg_list_image[8] += list_image[:, i]
        cnt1[8]+=1
    else:
        avg_list_image[9] += list_image[:, i]            #label = 9
        cnt1[9]+=1

for i in range(10):
    avg_list_image[i]= avg_list_image[i]/cnt1[i]
```

### 1.3 Visualize the average images.

- Visualization of 'im\_matrix\_avg'

```
In [428]: for i in range(10):
           im_matrix_avg = avg_list_image[i].reshape((size_row, size_col))

           plt.figure(1, figsize=(12,3.2))
           plt.text(22,25.5,"%d" % (i-1), fontsize=12)
           plt.subplot(1, 10, i+1)
           plt.imshow(im_matrix_avg, cmap='Greys', interpolation='None')

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

           plt.show()
```

