# assignment11

June 13, 2019

### 0.0.1 Digits classifier based on k random features

### 0.0.2 Name : Park Jinhyuk

### 0.0.3 Student-ID : 20141775

In [16]:
```
import matplotlib.pyplot as plt
import numpy as np
import random

file_data                = "mnist_train.csv"
file_test_data        = "mnist_test.csv"
handle_file        = open(file_data, "r")
handle_test_file        = open(file_test_data, "r")
data                              = handle_file.readlines()
test_data                       = handle_test_file.readlines()
handle_file.close()
handle_test_file.close()

size_row        = 28    # height of the image
size_col          = 28    # width of the image
num_image        = len(data)
test_num_image= len(test_data)
count               = 0     # count for the number of images
numSize = 10
vectorSize = 1000

list_image  = np.empty((num_image, vectorSize + 1), dtype=float)
list_label  = np.empty((numSize, num_image), dtype=int)
list_dataLabel  = np.empty(num_image, dtype=int)
classification = np.zeros(num_image, dtype=float)

test_list_image  = np.empty((test_num_image, vectorSize + 1), dtype=float)
test_list_dataLabel  = np.empty(test_num_image, dtype=int)
test_classification = np.zeros(test_num_image, dtype=float)

list_random_vector = np.empty((vectorSize, size_row * size_col), dtype=float)
```

```
for i in range(vectorSize):
    list_random_vector[i] = np.random.randint(-100, 100, size=size_row * size_col)

for num in range(numSize):
    count = 0
    for line in data:
        line_data   = line.split(',')
        list_dataLabel[count] = line_data[0]
        if line_data[0] == str(num):
            label = 1
        else:
            label = -1

        if num == 0:
                im_vector    = np.asfarray(line_data[1:])
                list_image[count, 0] = 1

                for i in range(vectorSize):
                    value = np.dot(im_vector, list_random_vector[i])
                    if value < 0:
                        value = 0
                    list_image[count, i+1] = value

        list_label[num, count] = label
        count = count + 1


    count = 0
    for line in test_data:
        test_line_data   = line.split(',')
        test_list_dataLabel[count] = test_line_data[0]

        if num == 0:
            im_vector    = np.asfarray(test_line_data[1:])
            test_list_image[count, 0] = 1
            for i in range(vectorSize):
                value = np.dot(im_vector, list_random_vector[i])
                if value < 0:
                    value = 0
                test_list_image[count, i+1] = value


        count = count + 1
```

### 0.0.4 Compute optimal model parameter

```
In [17]: d = vectorSize
         list_X = np.zeros((numSize, d+1))
         for num in range(numSize):
             X = np.zeros(d+1)
             X = np.dot(np.linalg.pinv(list_image),list_label[num])
             list_X[num] = X
         print(list_X)
```

```
[[-1.01358084e+00 -2.69395169e-08  6.04266731e-08 ... -4.22820679e-08
  -1.36234296e-07  4.57927374e-08]
 [-6.60942166e-01 -8.51336276e-08  1.59642044e-07 ... -1.13161183e-07
  -1.05003399e-09 -8.13340367e-08]
 [-1.08958339e+00  1.75196143e-07 -1.86862144e-07 ... -5.05009235e-08
  -1.26388548e-07  1.82580477e-07]
 ...
 [-8.10972947e-01  2.54490871e-07 -4.16785901e-07 ...  9.71385773e-08
   2.06861198e-07 -2.83661806e-07]
 [-5.48142918e-01 -1.33673435e-07 -3.36934979e-08 ... -2.29268933e-07
   2.35750131e-07 -1.74272875e-07]
 [-6.98264455e-01 -1.74349437e-07  4.34046133e-07 ...  3.36593253e-07
  -2.99815580e-07  6.13778880e-07]]
```

### 0.0.5 Compute tp rate, error rate on training set

```
In [18]: for i in range(num_image):
             max = np.dot(list_image[i], list_X[0])
             for j in range(1, numSize):
                 check = np.dot(list_image[i], list_X[j])
                 if  check > max:
                     max = check
                     classification[i] = j

         tp = 0
         er = 0

         for i in range(num_image):
             if list_dataLabel[i] ==  classification[i]:
                 tp = tp + 1
             else:
                 er = er + 1


         print("true positive rate : ", tp / num_image)
         print("error rate : ", er / num_image)
```

```
true positive rate :  0.94615
```

```
error rate :  0.05385
```

### 0.0.6 Compute tp rate, error rate on testing set

```
In [19]: for i in range(test_num_image):
             max = np.dot(test_list_image[i], list_X[0])
             for j in range(1, numSize):
                 check = np.dot(test_list_image[i], list_X[j])
                 if  check > max:
                     max = check
                     test_classification[i] = j

         tp = 0
         er = 0

         for i in range(test_num_image):
             if test_list_dataLabel[i] ==  test_classification[i]:
                 tp = tp + 1
             else:
                 er = er + 1

         print("true positive rate : ", tp / test_num_image)
         print("error rate : ", er / test_num_image)
```

```
true positive rate :  0.9421
error rate :  0.0579
```

```
In [ ]:
```