assignment09

May 29, 2019

```
In [1]: import matplotlib.pyplot as plt
        import numpy as np
        import copy

        #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()

        #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 [2]: #
        # make a matrix each column of which represents an images in a vector form
        #
        list_image  = np.zeros((num_image, size_row * size_col), dtype=float)
        list_label  = np.zeros(num_image, dtype=int)

        count = 0
        for line in data:
            #the number of lables is at the front. so split and put it into lable value.
            line_data   = line.split(',')
            list_label[count]  = line_data[0]
            list_image[count]  = np.asfarray(line_data[1:])
            count += 1
```

```
In [3]: #my file data path
        file_data = "C:\\Users\\recognize_data\\mnist_test.csv"
        handle_file = open(file_data, "r")
        #read data with line
        data = handle_file.readlines()
        handle_file.close()

        t_list_image  = np.zeros((num_image, size_row * size_col), dtype=float)
        t_list_label  = np.zeros(num_image, dtype=int)
        count = 0
        for line in data:
            #the number of lables is at the front. so split and put it into lable value.
            line_data   = line.split(',')
            t_list_label[count] = line_data[0]
            t_list_image[count] = np.asfarray(line_data[1:])
            count += 1
```

```
In [4]: #make matrix
        matrix = np.zeros((num_image, size_row * size_col+1), dtype=float)
        for i in range(num_image):
            for j in range(size_row * size_col+1):
                if(j == 0):
                    matrix[i,j] = 1
                else:
                    matrix[i,j] = list_image[i,j-1]
```

```
In [5]: # assign y value
        y = np.zeros((num_image,1), dtype=float)
        count = 0
        for i in list_label:
            if(i == 0):
                y[count] = 1
            else:
                y[count] = -1
            count += 1
```

```
In [6]: seta = np.zeros((size_row * size_col+1, 1), dtype=float)
        values = copy.deepcopy(np.linalg.pinv((np.mat(matrix.T)*np.mat(matrix))*np.mat(matrix
```

```
In [7]: seta = np.ravel(values)
        seta = np.reshape(np.array(seta),(size_row*size_col +1,1))

        matrix = np.reshape(np.array(matrix),(num_image,size_row*size_col +1))
```

1 Compute an optimal model parameter using the training dataset

```
In [8]: print(seta)
```

```
[[-6.84400682e-01]
 [ 3.49426949e-09]
```

```
[-2.39950944e-10]
[-2.74822237e-11]
[-2.05729495e-10]
[ 9.56589032e-12]
[-4.20560571e-10]
[ 5.31913243e-10]
[-1.11322731e-09]
[-5.34923981e-10]
[ 2.88744993e-10]
[ 5.32582631e-10]
[ 4.14908747e-10]
[ 6.04326103e-04]
[ 7.77727972e-04]
[-6.07077762e-04]
[-2.52943447e-05]
[-1.08758569e-10]
[-1.01824522e-10]
[ 1.46833939e-10]
[ 1.89107654e-10]
[ 5.04019308e-11]
[-2.15247331e-11]
[ 1.71569671e-10]
[-1.70336512e-13]
[-1.46927567e-11]
[ 6.53607371e-11]
[ 3.02922839e-11]
[-1.39328410e-10]
[ 2.75964590e-11]
[ 2.02360983e-11]
[-1.09155052e-11]
[-5.86346062e-11]
[-1.59945741e-02]
[ 4.14204254e-03]
[ 4.74947342e-04]
[ 8.35473231e-04]
[ 2.50260680e-04]
[ 1.04342121e-05]
[ 2.15409587e-04]
[ 4.56837736e-04]
[-2.85422292e-04]
[ 1.07011459e-03]
[-6.39450389e-04]
[ 2.42222719e-04]
[ 7.40242646e-07]
[ 3.02358751e-04]
[-2.43025533e-04]
[ 6.01151185e-05]
[ 3.35425856e-04]
```

```
[-5.56004360e-04]
[ 9.33034848e-04]
[ 1.42219374e-04]
[-7.01995902e-11]
[ 1.63811829e-10]
[ 1.55981498e-11]
[-3.92099336e-11]
[-5.19866084e-11]
[ 3.06711003e-11]
[-3.27715151e-02]
[ 3.83110400e-02]
[ 2.09591860e-03]
[-3.13972128e-03]
[-7.42339597e-04]
[ 3.95846954e-04]
[-2.06978185e-04]
[-3.20799890e-04]
[-2.93442009e-04]
[ 1.96977724e-04]
[-1.93346496e-04]
[-5.58880946e-05]
[-1.33110634e-04]
[ 1.43459270e-05]
[-4.34767160e-04]
[-1.01564417e-04]
[-1.71785297e-04]
[-1.84766705e-04]
[-3.87117330e-04]
[ 3.97242953e-04]
[-6.62298823e-04]
[ 4.06127377e-04]
[-9.64104813e-04]
[ 1.60552547e-03]
[-3.12611860e-11]
[-5.51288493e-11]
[ 1.64011230e-11]
[-8.08766317e-11]
[ 1.55016952e-02]
[-7.03348647e-03]
[-3.34105396e-04]
[-3.20272883e-04]
[-2.49271561e-04]
[-7.99253589e-06]
[ 9.22399653e-05]
[-2.43955973e-04]
[-1.32616239e-04]
[ 2.14760565e-04]
[-2.33087330e-04]
```

```
 [ 1.80955599e-04]
 [-1.79704339e-04]
 [-1.20518041e-04]
 [-6.89204958e-05]
 [-1.60595817e-04]
 [-4.05889481e-04]
 [-1.80041333e-04]
 [-4.23941893e-04]
 [-4.11984617e-04]
 [-2.71637015e-04]
 [-2.63139133e-04]
 [ 4.24219153e-04]
 [-4.38783278e-04]
 [ 8.96841833e-04]
 [ 3.35774370e-12]
 [ 7.38338498e-12]
 [ 2.42773064e-03]
 [-1.12478240e-03]
 [-2.24680735e-04]
 [ 2.86936795e-04]
 [-4.63961940e-05]
 [-8.04723390e-05]
 [-1.95447766e-04]
 [ 1.46168000e-04]
 [-1.70752690e-04]
 [-8.78153047e-05]
 [-1.11027651e-04]
 [ 6.53300378e-06]
 [ 1.20304370e-04]
 [-1.36653349e-04]
 [ 1.53296660e-04]
 [-6.51623858e-05]
 [ 1.48795670e-05]
 [-9.51852082e-06]
 [ 1.00919167e-04]
 [-2.70794065e-05]
 [ 1.08952357e-04]
 [-9.64767537e-05]
 [-2.59731232e-04]
 [-2.52634177e-04]
 [-1.40570159e-04]
 [-9.52712827e-04]
 [ 5.20511681e-03]
 [ 2.75653297e-12]
 [ 1.06757888e-11]
 [-3.60183910e-03]
 [-2.95491939e-04]
 [-5.46745741e-05]
```

```
[-1.58397643e-04]
[ 1.51064279e-04]
[-5.95756191e-05]
[-1.15187205e-04]
[-9.62948406e-05]
[-7.71324642e-05]
[ 3.95501850e-06]
[-8.81384029e-05]
[ 1.40384602e-04]
[ 4.36485558e-07]
[ 7.61845073e-05]
[ 3.82886079e-05]
[ 5.55291465e-05]
[-5.79430096e-05]
[ 7.45529027e-05]
[-9.98412895e-05]
[-7.14724242e-05]
[-2.00395131e-04]
[-2.46707459e-04]
[-9.26043903e-05]
[-1.12734091e-04]
[-4.38045481e-04]
[-3.51582822e-03]
[-2.08523971e-12]
[-1.00064657e-02]
[ 6.95084265e-04]
[ 2.61924852e-05]
[ 7.41631076e-05]
[-1.13400763e-04]
[ 5.78863723e-05]
[-6.56148213e-05]
[ 7.29504781e-05]
[-1.05548500e-04]
[-8.68870342e-05]
[-4.17208954e-05]
[-6.81586886e-06]
[ 8.96426032e-05]
[ 1.15736337e-05]
[-1.96651866e-05]
[ 4.71509359e-05]
[-4.42554837e-06]
[ 5.90350150e-05]
[-5.43032075e-05]
[ 1.09030186e-05]
[ 7.54106828e-05]
[ 2.29011022e-05]
[-1.49769345e-04]
[-4.07710306e-04]
```

```
[-1.23805410e-04]
[-2.99172264e-04]
[-3.38574985e-04]
[-1.18914518e-01]
[-2.16229514e-04]
[-1.22846980e-03]
[ 1.61922259e-04]
[-5.07236809e-05]
[ 5.89246996e-05]
[-3.07236553e-05]
[ 1.72831724e-05]
[-1.01529290e-04]
[ 1.40738498e-04]
[-6.21810812e-05]
[ 5.58545918e-05]
[ 5.65902455e-05]
[ 5.39387051e-05]
[ 1.44750127e-04]
[ 1.25981616e-04]
[ 7.21789152e-06]
[ 1.50039876e-04]
[ 8.42352616e-05]
[ 7.41245611e-05]
[-8.22725844e-05]
[-2.75724353e-05]
[ 1.43651840e-05]
[-6.63334439e-05]
[-4.28690527e-04]
[-1.84026750e-04]
[-1.32266903e-04]
[ 4.67336290e-04]
[ 3.03429501e-02]
[ 3.15313494e-04]
[ 1.25859131e-04]
[-1.49142168e-04]
[-1.79606419e-05]
[-4.24308427e-05]
[ 1.48267504e-04]
[-4.01433966e-05]
[ 1.93321191e-05]
[-1.80935598e-05]
[ 4.71918913e-06]
[-5.06238395e-05]
[ 5.91495744e-05]
[ 1.62091366e-05]
[ 7.84787697e-05]
[ 7.11430138e-05]
[ 2.04562806e-04]
[ 5.52392392e-05]
[ 6.26020792e-05]
[ 3.89989784e-06]
[ 1.22574283e-04]
[-1.06072969e-04]
[ 4.84782039e-05]
[-3.65035548e-06]
[-4.81381655e-04]
[-4.97114420e-04]
[-1.04338885e-04]
[-2.94030930e-04]
[-2.47671326e-03]
[ 4.97375464e-04]
[-2.08512626e-04]
[ 1.75105181e-05]
[-6.74949369e-05]
[-6.86975829e-05]
[-9.10553418e-06]
[ 8.87243255e-05]
[ 4.60263544e-05]
[ 1.96283747e-05]
[ 4.30155510e-05]
[-7.05426544e-05]
[-2.48968233e-05]
[ 1.14604441e-04]
[ 1.01368804e-04]
[ 1.28875990e-04]
[ 1.67649773e-05]
[ 4.59890322e-05]
[ 3.43810247e-05]
[ 1.62772186e-04]
[ 4.14124853e-05]
[ 8.23575190e-05]
[-1.37738766e-04]
[ 1.04903144e-04]
[-1.30509009e-04]
[-8.62662899e-04]
[ 1.41821135e-05]
[-2.91083339e-04]
[ 1.65344167e-03]
[-8.57795749e-04]
[-2.36820940e-05]
[-8.30907340e-05]
[ 1.72491913e-05]
[ 8.82461884e-05]
[-4.85639151e-06]
[-5.35873617e-05]
[-6.39806972e-05]
```

```
[ 1.12157717e-04]
[-1.55205656e-05]
[ 4.31141215e-05]
[ 6.78885620e-05]
[ 3.56959258e-05]
[ 3.48753023e-05]
[ 1.19444725e-04]
[ 7.49127407e-05]
[ 1.41533658e-04]
[ 1.34517960e-04]
[ 3.10147400e-05]
[ 2.61822903e-04]
[ 1.29468581e-04]
[ 1.70288722e-04]
[ 2.94952852e-04]
[-2.99205002e-04]
[-8.75932026e-04]
[-6.61373133e-04]
[ 1.52583005e-03]
[-1.49439681e-03]
[ 1.16891839e-03]
[-2.20337520e-04]
[-1.34418479e-04]
[ 1.09798882e-05]
[-2.73460074e-04]
[ 3.74424942e-06]
[ 7.58844702e-05]
[ 4.02571380e-05]
[-5.07624839e-05]
[ 9.23364134e-05]
[ 5.71294500e-05]
[-8.07044620e-05]
[-5.71445547e-06]
[-1.65932686e-04]
[-1.00237528e-04]
[-8.18370983e-05]
[ 2.81358773e-06]
[ 3.86890265e-05]
[ 1.63909511e-04]
[ 7.86074386e-05]
[ 2.52059956e-04]
[ 5.80506576e-05]
[ 2.77168821e-04]
[ 4.20348119e-04]
[-1.33800685e-03]
[ 5.72796301e-05]
[-2.63194031e-03]
[ 5.22902209e-04]
```

```
[-9.31909063e-04]
[-2.94130020e-04]
[ 8.96352669e-05]
[-2.94337644e-04]
[-3.92139533e-05]
[ 7.30401139e-05]
[-6.05969837e-05]
[ 7.75251429e-05]
[ 3.76828337e-05]
[ 6.86647284e-05]
[-9.06970774e-05]
[ 5.79208370e-05]
[-4.32975329e-05]
[-3.00024286e-04]
[-1.13701524e-04]
[-1.67711633e-04]
[-1.31620073e-04]
[-1.46072921e-04]
[-1.12794507e-04]
[ 1.04969563e-04]
[ 6.40609883e-05]
[ 1.10839687e-04]
[ 5.04200557e-04]
[ 3.27219116e-04]
[-1.47709216e-04]
[-1.28877468e-03]
[ 2.76703753e-03]
[ 1.42773262e-03]
[ 2.79454785e-03]
[ 9.58220912e-05]
[-8.17388053e-06]
[-5.02660465e-05]
[-4.22790457e-05]
[-8.17645695e-05]
[ 7.65486483e-05]
[-8.73660317e-05]
[ 6.03380495e-05]
[ 3.95094853e-05]
[ 3.06904309e-05]
[-1.10393349e-04]
[-1.62850958e-04]
[-2.03959426e-04]
[-1.03073378e-04]
[-9.03318324e-05]
[-1.90542363e-04]
[-1.95913438e-04]
[-9.08122650e-05]
[-1.78198172e-04]
```

```
[ 7.18573853e-05]
[ 3.46158407e-04]
[ 2.82039701e-04]
[-3.77427381e-04]
[ 4.02388153e-07]
[ 4.67348216e-04]
[-1.61625762e-03]
[-1.31516825e-03]
[-1.71275300e-03]
[-2.43431355e-04]
[-2.36184406e-04]
[-5.44568538e-04]
[ 8.19733785e-05]
[ 1.54103643e-04]
[ 9.46563733e-05]
[ 5.68725002e-05]
[ 8.06111687e-05]
[ 1.96616318e-04]
[ 3.86822174e-07]
[-1.68802394e-04]
[ 2.02758790e-05]
[-2.73506401e-04]
[ 2.20200868e-05]
[-1.56837201e-04]
[-1.68223118e-04]
[-1.15831481e-04]
[-1.50176855e-04]
[ 7.28190108e-05]
[ 3.15518563e-04]
[ 1.23487492e-06]
[ 9.32734982e-05]
[ 1.15604188e-04]
[-2.04990880e-04]
[-2.33492540e-04]
[ 2.60005469e-03]
[-1.68334851e-03]
[ 9.87336654e-04]
[ 6.85826477e-05]
[-1.22507050e-04]
[ 1.90838650e-04]
[ 2.23753517e-04]
[-1.43977096e-04]
[ 1.37962773e-04]
[ 1.30003154e-04]
[ 1.46589415e-04]
[ 1.24360728e-04]
[-5.64353775e-05]
[-1.26878955e-04]
```

```
[-8.67116004e-05]
[-2.26726979e-04]
[-3.17842837e-05]
[-1.94511086e-04]
[-1.57387033e-04]
[-1.16331129e-04]
[-5.57711255e-05]
[ 1.24330758e-04]
[-1.96869598e-05]
[-5.35517453e-06]
[ 1.00811142e-04]
[-9.96864008e-05]
[-2.01613204e-04]
[-1.23010710e-04]
[ 5.81420199e-04]
[-7.15838971e-03]
[ 2.05147667e-04]
[ 1.17847077e-03]
[-9.27670420e-04]
[ 2.94898726e-05]
[ 3.93309897e-05]
[ 1.15396502e-04]
[ 2.71380243e-05]
[ 1.06848306e-04]
[ 3.74999149e-05]
[ 8.27839544e-05]
[-3.99799381e-05]
[-2.20365186e-04]
[-1.44911100e-04]
[-1.38817269e-04]
[-1.17517345e-04]
[-1.60475715e-04]
[-1.75930866e-04]
[-2.18215355e-05]
[ 6.03992520e-05]
[ 5.45599055e-05]
[ 4.21722031e-05]
[-2.14738267e-04]
[ 9.66146407e-05]
[ 9.22848707e-05]
[-4.94792715e-05]
[-2.50357486e-04]
[-1.66804819e-03]
[ 6.15451683e-13]
[-5.04300570e-04]
[-7.33969255e-04]
[-1.07834268e-04]
[ 6.57189190e-05]
[ 9.51659259e-05]
[ 1.11097627e-04]
[ 9.76544422e-05]
[-1.64565005e-05]
[ 5.98150500e-05]
[ 6.62385406e-05]
[-5.90066028e-05]
[-5.66823300e-05]
[-1.73904398e-04]
[-2.19406214e-04]
[-1.88192865e-04]
[-7.14022343e-05]
[ 4.28411676e-05]
[ 3.61944081e-06]
[ 8.53065420e-05]
[-5.65919773e-05]
[-6.32895481e-05]
[ 2.12429971e-04]
[-9.48322492e-05]
[ 1.98197595e-05]
[-7.10062506e-05]
[ 6.46945764e-04]
[ 6.46917056e-03]
[-8.58986409e-04]
[-5.24852594e-04]
[ 9.97643762e-04]
[-9.27293110e-04]
[-1.87886718e-05]
[ 1.34125084e-04]
[-3.30200205e-05]
[ 1.59355901e-04]
[ 1.30802856e-04]
[ 1.56612290e-05]
[-3.06972910e-05]
[ 3.34295876e-05]
[-1.04114807e-04]
[-7.17951942e-05]
[-1.20125666e-04]
[ 1.12911511e-05]
[-5.12057649e-05]
[-4.44298991e-05]
[-5.62617819e-05]
[-4.75154305e-06]
[-1.07974994e-04]
[-2.08559895e-05]
[-2.62757747e-05]
[-2.78204528e-04]
[ 2.54226794e-04]
```

```
[-1.36028125e-04]
[-7.52834096e-04]
[-4.84610929e-03]
[-4.40007646e-03]
[ 9.51090865e-04]
[-3.58836395e-04]
[ 4.93593170e-05]
[-2.53005590e-05]
[-1.06517731e-04]
[ 6.08218036e-05]
[ 9.78296824e-05]
[ 4.82179028e-05]
[ 1.35539753e-04]
[-2.06457048e-05]
[ 7.08821389e-05]
[ 5.18468205e-05]
[ 4.53854496e-05]
[ 9.30531428e-06]
[-4.69071858e-05]
[-1.44644599e-05]
[ 8.88933933e-06]
[ 2.02406538e-07]
[-5.23170579e-05]
[ 2.82039574e-05]
[-3.40608950e-05]
[ 3.13038617e-06]
[ 5.03339655e-05]
[-7.12299309e-05]
[-2.07457465e-04]
[ 1.60676131e-04]
[ 3.36935759e-03]
[ 4.42008983e-13]
[ 3.88389271e-03]
[ 3.96813969e-04]
[-4.03362832e-04]
[-1.07775097e-04]
[ 6.72715145e-05]
[-6.27049038e-05]
[-1.54175217e-06]
[ 4.37476324e-05]
[ 6.43942602e-06]
[ 8.22159254e-05]
[ 2.22434396e-06]
[ 8.57708020e-05]
[ 4.96408122e-05]
[ 5.72838208e-07]
[ 2.61618910e-05]
[-1.01085886e-04]
```

```
[ 3.25230279e-05]
[-7.69466597e-05]
[ 6.95929831e-06]
[-7.32257276e-05]
[ 1.26304924e-04]
[-7.80485055e-05]
[-2.26794520e-04]
[ 3.23896660e-05]
[ 3.83538492e-04]
[-4.27906780e-04]
[-1.20154682e-03]
[ 1.40982589e-03]
[-4.07094265e-03]
[-1.20558917e-04]
[-2.51843807e-05]
[ 2.20535287e-04]
[-1.00834829e-04]
[-3.07048365e-05]
[-3.83672127e-05]
[-3.63912453e-05]
[ 1.07823750e-04]
[ 6.66495723e-05]
[ 8.33012349e-05]
[-3.06360396e-06]
[-4.91578827e-05]
[ 1.25531001e-05]
[ 1.05247537e-04]
[ 2.26350318e-05]
[-9.24089505e-05]
[ 2.46135292e-05]
[-3.36869226e-05]
[-1.49892107e-05]
[-6.35036359e-05]
[ 4.65493452e-05]
[-1.71145698e-04]
[ 8.62899140e-05]
[-4.25121695e-05]
[ 1.02726210e-03]
[ 2.24948211e-03]
[ 1.36576884e-03]
[-4.02954403e-03]
[-6.91403290e-05]
[ 3.93783920e-05]
[-4.55891252e-04]
[ 1.00512305e-04]
[ 1.27903263e-04]
[ 3.94697057e-05]
[ 2.16811898e-05]
[ 6.67717552e-05]
[ 1.10116801e-05]
[ 7.58941178e-05]
[ 5.79426456e-05]
[ 1.65383018e-04]
[ 4.27858109e-05]
[-6.00567248e-05]
[-2.36783386e-05]
[-2.26763700e-05]
[-1.37481865e-04]
[ 6.41938151e-05]
[-1.90769457e-04]
[ 7.98650914e-05]
[ 5.15310521e-05]
[-5.82966992e-05]
[ 7.14421184e-05]
[ 4.02301238e-05]
[ 1.38863052e-03]
[ 4.15289005e-03]
[-4.14261297e-14]
[ 4.29190170e-16]
[ 4.92180494e-04]
[-9.15302068e-05]
[-2.30530348e-04]
[-7.19434847e-06]
[-2.36930460e-04]
[ 9.97219980e-05]
[-7.47962445e-05]
[ 5.15891893e-05]
[ 1.23549644e-05]
[ 8.40027876e-05]
[ 1.00722066e-04]
[ 1.33904592e-05]
[ 1.18677641e-04]
[ 1.48353618e-04]
[ 8.52029599e-05]
[ 8.33944881e-05]
[-7.03078540e-05]
[-8.66887568e-05]
[-4.36937685e-05]
[-2.07712704e-04]
[ 3.59742760e-05]
[ 7.56242549e-06]
[ 6.40804865e-05]
[-5.75036742e-04]
[-2.39526537e-03]
[ 5.17399575e-15]
[-7.68032726e-15]
```

```
[ 4.45242754e-15]
[-2.64121693e-03]
[ 6.56297110e-04]
[ 9.03023662e-05]
[-3.47583481e-04]
[ 2.84702284e-04]
[-1.41178289e-04]
[ 1.99170645e-05]
[-1.07194795e-04]
[-1.91456192e-05]
[-1.79249105e-04]
[-5.46680782e-05]
[-7.07987978e-05]
[-1.24413675e-04]
[-5.40313144e-05]
[-1.58708528e-05]
[ 1.62087464e-05]
[ 1.52101196e-04]
[-3.50574654e-05]
[ 1.74822051e-04]
[ 1.25070770e-05]
[-3.97415441e-04]
[ 1.06006709e-04]
[-5.52829940e-04]
[ 9.74250493e-04]
[ 1.74319699e-02]
[-2.27381972e-16]
[-3.79186154e-16]
[ 7.53627981e-17]
[ 2.25341542e-03]
[ 2.75221482e-04]
[-5.77482627e-06]
[-5.34487568e-04]
[-2.16488118e-04]
[-4.12862617e-04]
[-2.16807793e-04]
[-1.88468149e-04]
[-2.61263532e-04]
[-1.05909376e-04]
[-2.54171812e-04]
[-1.20879956e-04]
[-1.72935523e-04]
[-4.30410087e-05]
[-1.18880037e-04]
[-1.34502771e-04]
[-8.61876851e-05]
[ 4.39729917e-05]
[-2.82200685e-04]
```

```
[ 3.21052294e-05]
[ 5.59128755e-04]
[-6.65668087e-04]
[-7.14167773e-04]
[-1.94927675e-03]
[-1.66503421e-02]
[ 0.00000000e+00]
[ 0.00000000e+00]
[ 0.00000000e+00]
[ 0.00000000e+00]
[-3.63158966e-03]
[ 1.02635695e-03]
[-8.10016186e-04]
[ 2.99840982e-05]
[-3.59731874e-04]
[-1.85729781e-04]
[-4.57211057e-04]
[-1.69018963e-04]
[-3.97451693e-04]
[-1.32292881e-04]
[-3.53779214e-04]
[-1.83733190e-04]
[-3.83393264e-04]
[-2.12321917e-04]
[-1.28394434e-04]
[-2.92137001e-04]
[-2.97131272e-05]
[ 3.34916931e-05]
[-6.11080331e-04]
[-2.49530830e-04]
[-1.03429228e-03]
[ 8.08379010e-03]
[-1.07073367e-03]
[ 0.00000000e+00]
[ 0.00000000e+00]
[ 0.00000000e+00]
[ 0.00000000e+00]
[ 0.00000000e+00]
[ 0.00000000e+00]
[ 3.13353810e-03]
[ 9.94812755e-05]
[-1.48160953e-03]
[ 2.36676229e-04]
[-5.05664454e-04]
[-4.44827477e-04]
[-1.36452384e-04]
[-7.04442263e-05]
[-5.33115291e-04]
```

```
 [-3.00935871e-05]
 [-5.32395717e-04]
 [-9.63116627e-05]
 [-1.69477102e-04]
 [-4.51131693e-04]
 [-1.75895508e-04]
 [ 4.27275341e-05]
 [-4.03498300e-04]
 [ 7.23424500e-04]
 [-1.37293248e-03]
 [-8.18069255e-03]
 [ 0.00000000e+00]
 [ 0.00000000e+00]
 [ 0.00000000e+00]
 [ 0.00000000e+00]]
```

```
In [9]: estimation = np.zeros((num_image,1), dtype=float)
        for i in range(num_image):
            for j in range(size_row * size_col+1):
                estimation[i] += matrix[i][j]*seta[j]
```

```
In [10]: label_zero = []
         label_non_zero = []
         for i in range(num_image):
             if(list_label[i] == 0):
                 label_zero = np.append(label_zero, i)
             else:
                 label_non_zero = np.append(label_non_zero, i)


         label_zero_size = len(label_zero)
         label_non_zero_size = len(label_non_zero)
```

```
In [11]: def TP_FP(label, esti, size):
             count = 0
             for i in label:
                 if(esti[int(i)] > 0):
                     count += 1

             return (count/size)

         def TN_FN(label, esti, size):
             count = 0
             for i in label:
                 if(esti[int(i)] <= 0):
                     count += 1

             return (count/size)
```

```
In [12]: tp = TP_FP(label_zero,estimation, label_zero_size)
         fp = TP_FP(label_non_zero,estimation, label_non_zero_size)

         tn = TN_FN(label_non_zero,estimation, label_non_zero_size)
         fn = TN_FN(label_zero,estimation, label_zero_size)
```

2 Compute TP, FP, TN, FN - Traning dataset

```
In [13]: print("train_TP: ",tp)
         print("train_FP: ",fp)
         print("train_TN: ",tn)
         print("train_FN: ",fn)
```

```
train_TP:  0.8725308120884687
train_FP:  0.003310094864729922
train_TN:  0.9966899051352701
train_FN:  0.1274691879115313
```

3 test

```
In [14]: t_label_zero = []
         t_label_non_zero = []
         for i in range(num_image):
             if(t_list_label[i] == 0):
                 t_label_zero = np.append(t_label_zero, i)
             else:
                 t_label_non_zero = np.append(t_label_non_zero, i)


         t_label_zero_size = len(t_label_zero)
         t_label_non_zero_size = len(t_label_non_zero)
```

```
In [15]: t_estimation = np.zeros((num_image,1), dtype=float)
         for i in range(num_image):
             for j in range(size_row * size_col+1):
                 if(j == 0):
                     t_estimation[i] += seta[j]
                 else:
                     t_estimation[i] += t_list_image[i][j-1]*seta[j]
```

```
In [16]: t_tp = TP_FP(t_label_zero,t_estimation, t_label_zero_size)
         t_fp = TP_FP(t_label_non_zero,t_estimation, t_label_non_zero_size)

         t_tn = TN_FN(t_label_non_zero,t_estimation, t_label_non_zero_size)
         t_fn = TN_FN(t_label_zero,t_estimation, t_label_zero_size)
```

4 Compute TP, FP, TN, FN - Testing dataset

```
In [17]: print("test_TP: ",t_tp)
         print("test_FP: ",t_fp)
         print("test_TN: ",t_tn)
         print("test_FN: ",t_fn)
```

```
test_TP:  0.016987053746567282
test_FP:  0.004767184035476719
test_TN:  0.9952328159645233
test_FN:  0.9830129462534327
```

```
In [ ]:
```