

# assignment 08

November 22, 2018

- 1 This is assignment08
- 2 Name:PENG CIYUAN
- 3 Student ID:2018220161
- 4 Link:<https://github.com/pcyyyy/assignment08.git>
- 5 Getting train data and test data

```
In [3]: import matplotlib.pyplot as plt
import numpy as np
from scipy import signal

file_data_train = "mnist_train.csv"
file_data_test  = "mnist_test.csv"

h_data_train    = open(file_data_train, "r")
h_data_test     = open(file_data_test, "r")

data_train      = h_data_train.readlines()
data_test       = h_data_test.readlines()

h_data_train.close()
h_data_test.close()

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

num_train       = len(data_train)    # number of training images
num_test        = len(data_test)     # number of testing images

#
# 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)

#
# example of distance function between two vectors x and y
#
def distance(x, y):

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

    return(s)

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

list_image_test     = np.empty((size_row * size_col, num_test), dtype=float)
list_label_test     = np.empty(num_test, dtype=int)

count = 0

for line in data_train:

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

    list_label_train[count]    = label
    list_image_train[:, count] = im_vector

    count += 1

count = 0

for line in data_test:

    line_data    = line.split(',')
    label        = line_data[0]

```

```

im_vector    = np.asfarray(line_data[1:])
im_vector    = normalize(im_vector)

list_label_test[count]      = label
list_image_test[:, count]   = im_vector

count += 1

```

## 6 Making Convolution Feature6\*6

```

In [5]: def filt(matrix):
        kernel = np.array([[ -1, -1, -1, -1, -1,-1],[-1, -1, -1, -1, -1,-1],[-1, 0, 9
        matrix = signal.convolve2d(matrix, kernel, boundary='symm', mode='same')
        return matrix

```

## 7 Computing the optimal set A of model parameters

```

In [6]: xa=list_image_train.T
        x=[]
        for i in range(len(xa)):
            x_ = xa[i].reshape((size_row, size_col))
            x_ = filt(x_)
            x_ = np.ravel(x_, order='C')
            x.append(x_)
        x=np.array(x)

        def vecY(Y):
            for i in range(0, len(Y)):
                if(int(Y[i]) == 0):
                    Y[i] = 1
                else:
                    Y[i] = -1
            return Y

        y = vecY( list_label_train)

        A1=np.linalg.inv((x.T).dot(x))
        A2=(x.T).dot(y)
        A=(A1).dot(A2)
        print(A)

[ 9.38775052e+14  2.59374095e+14 -5.83312265e+15 -1.35743344e+13
  2.95550148e+15  5.01836918e+14  4.98277969e+14  5.86391101e+14
 -3.15390129e+15  1.22630628e+14  1.71879695e+15  7.18812299e+13
 -9.38110160e+14  8.47233688e+14 -6.08290205e+14  2.88909343e+14
  7.65565477e+14 -3.07032457e+14 -2.55031691e+15  1.00175344e+15
  1.78927975e+15  5.30756026e+14  1.88487794e+14 -6.65434906e+14

```

-4.40902515e+15	1.06268353e+15	4.04727969e+15	8.49418508e+14
2.85682950e+11	1.54722262e+11	-4.29130378e+10	-3.69415404e+09
-8.54057107e+09	4.52109349e+09	-6.64878825e+09	5.44559461e+09
-5.99969540e+09	5.61921035e+09	-5.76649632e+09	5.59687548e+09
-5.60927171e+09	5.51249420e+09	-5.49052850e+09	5.43916987e+09
-5.42972503e+09	5.42294946e+09	-5.44505231e+09	5.47848395e+09
-5.52994149e+09	5.58910075e+09	-5.65702713e+09	5.72475866e+09
-5.81367888e+09	5.94409097e+09	-6.30142587e+09	5.08986626e+09
3.09136946e+14	-3.07961737e+14	3.25534880e+14	-3.47070221e+14
1.81425215e+13	-3.06507283e+14	6.11284234e+14	-3.11426262e+14
3.40188958e+14	-3.41526762e+14	1.80151881e+13	-3.10823462e+14
5.84342714e+14	-3.12691625e+14	3.74464297e+14	-3.34280480e+14
1.29148697e+13	-3.18953417e+14	5.48096755e+14	-3.09972049e+14
4.20110687e+14	-3.29967522e+14	4.45788541e+12	-3.25901304e+14
5.08826020e+14	-3.06287893e+14	4.66750756e+14	-3.28672184e+14
-9.43560954e+10	-3.09737136e+10	4.80632992e+10	7.16076927e+09
-8.95398269e+08	-9.01291983e+08	4.24068110e+08	-7.23472969e+08
3.94349335e+08	-6.02360605e+08	4.96293036e+08	-6.08059422e+08
5.95186044e+08	-6.67926183e+08	6.84337020e+08	-7.25740455e+08
7.33722094e+08	-7.40569963e+08	7.24438026e+08	-7.00422392e+08
6.64532050e+08	-6.26010154e+08	5.90495781e+08	-5.60641588e+08
5.64209427e+08	-6.94460647e+08	1.15265328e+09	-7.25433723e+08
1.84887666e+12	-3.36449140e+12	3.13934478e+12	-3.07419048e+12
2.43123943e+12	-3.54555842e+12	4.15491145e+12	-3.11817451e+12
3.96190973e+12	-3.25195136e+12	1.09065343e+12	-3.49155086e+12
4.91156540e+12	-3.05452393e+12	4.24727801e+12	-3.42002954e+12
2.91150599e+11	-3.26175861e+12	5.36438542e+12	-3.23756956e+12
4.40638051e+12	-3.39821984e+12	1.47841339e+10	-3.10710386e+12
5.24712555e+12	-3.47061638e+12	4.75927981e+12	-3.20495248e+12
1.08844427e+10	1.19693869e+09	-1.48148223e+10	-1.27620690e+09
4.01320077e+09	2.86520280e+08	8.67229248e+07	2.43928156e+07
-6.80817243e+07	6.35220899e+07	-1.21060506e+08	7.96784798e+07
-9.09704156e+07	5.14683348e+07	-4.27517098e+07	1.61759222e+07
-1.05783270e+07	4.32046003e+06	-1.26183775e+07	2.46082615e+07
-4.17705650e+07	5.86512213e+07	-6.46425668e+07	5.83502658e+07
-1.03101676e+07	4.10374311e+07	-6.34407656e+07	1.51041756e+07
6.79100461e+10	-3.64269581e+10	-2.45393272e+10	-4.37257972e+10
2.36784272e+10	-3.26702776e+10	1.14308875e+11	-4.49623682e+10
-3.21178074e+10	-3.22985989e+10	4.81894163e+10	-3.87488693e+10
8.44918353e+10	-4.99894412e+10	-7.19046602e+09	-2.06354062e+10
4.09454580e+10	-4.85507965e+10	6.52496232e+10	-4.49748498e+10
2.88902121e+10	-2.24719786e+10	1.57907676e+10	-5.07728478e+10
5.93425297e+10	-3.52950049e+10	5.45982517e+10	-3.81945458e+10
-3.47010312e+08	8.55836238e+07	1.16183552e+09	1.88940905e+07
-1.14262557e+09	-3.95219734e+07	3.88343953e+08	-2.69890252e+07
4.65273560e+07	1.69721593e+06	-1.62557934e+07	1.84061874e+07
-2.65699840e+07	1.28794831e+07	-1.00165782e+07	-3.17206237e+06
6.30729545e+06	-1.09710610e+07	8.24247100e+06	-4.33244610e+06

-1.11986625e+06	6.02020424e+06	-4.44181964e+06	9.43422783e+06
-5.94418412e+06	2.02963304e+06	3.80559619e+06	-5.62400534e+06
-1.09185794e+09	-9.22126460e+08	2.96058583e+09	-9.01991871e+08
8.53990076e+08	-1.44618989e+09	-1.08820802e+08	-3.54763362e+08
2.41139444e+09	-9.93192966e+08	-5.39500631e+08	-1.52063236e+09
2.03274507e+09	-4.21035398e+08	1.14667971e+09	-1.12323020e+09
-8.19969461e+08	-8.71009109e+08	3.11688784e+09	-1.18842255e+09
3.61549550e+08	-9.68082974e+08	-4.54695973e+08	-2.36725516e+08
2.59085685e+09	-2.07694799e+09	8.97042461e+08	-4.04802807e+08
9.45227742e+07	1.21276333e+07	-5.84062256e+07	1.45212386e+07
3.16842064e+07	2.92682042e+06	-8.17610535e+07	-2.95390560e+06
4.31842533e+07	-6.22576032e+06	4.48159688e+06	5.35844589e+06
-8.77433727e+06	7.35734635e+06	-7.22960508e+06	2.47485858e+06
-1.15209050e+06	-1.49956346e+06	1.12962834e+06	-5.04990382e+05
-8.50279752e+05	3.12740451e+06	-3.60893168e+06	4.55559103e+06
-3.12366807e+06	2.60695328e+06	-1.61920920e+06	-7.42241406e+04
9.59788230e+07	-1.00499404e+08	-1.22915036e+07	-1.63502558e+08
2.17406815e+08	-1.18284587e+08	1.50203744e+08	-9.86784603e+07
-2.86788052e+06	-1.19093925e+08	2.34807794e+08	-1.41739075e+08
1.01720593e+08	-1.30909401e+08	5.68566887e+07	-6.96826992e+07
1.82039605e+08	-1.61896250e+08	1.09969656e+08	-1.37913352e+08
1.27831172e+08	-6.34591547e+07	7.63383526e+07	-1.62656709e+08
1.48582174e+08	-1.17709553e+08	1.85410614e+08	-1.14436984e+08
6.26812850e+06	9.16831727e+05	2.10185297e+07	2.45965541e+06
-1.11968154e+07	3.48884922e+06	-2.75726695e+06	-9.61599609e+04
-3.56862782e+06	-1.18449361e+06	5.02479418e+06	-1.31843115e+05
-8.09972701e+05	1.99571361e+06	-2.31370039e+06	1.24953071e+06
-8.76619553e+05	-2.23207104e+05	3.18323357e+05	-3.42901747e+05
-3.30954116e+04	6.57246662e+05	-8.62991695e+05	1.26978386e+06
-1.00969753e+06	7.64536796e+05	-3.44228828e+05	-4.40963906e+04
-1.05643740e+07	1.40293938e+06	2.39103319e+06	-2.88556012e+06
-4.46754766e+04	-2.41197713e+06	5.78147750e+05	2.63828646e+06
1.97855556e+06	2.47731701e+05	-2.44156125e+06	-3.59384166e+06
3.48454369e+06	1.95087846e+06	-2.64888468e+06	4.03076105e+05
-9.32069692e+05	-1.69433521e+06	6.74828167e+06	-4.64760870e+05
-5.35447093e+06	6.70497648e+05	-1.31361268e+06	1.23870095e+06
5.98864789e+06	-4.17119840e+06	-2.48150033e+06	1.96441145e+06
2.26426750e+06	-9.04779688e+04	3.10116719e+06	1.68584625e+05
1.20099461e+06	7.75878531e+05	-2.08538318e+06	5.28625680e+05
-3.26013524e+05	-2.36658916e+05	2.38381622e+05	-1.03924819e+05
3.16831638e+05	3.30694431e+05	-4.82632697e+05	4.25033748e+05
-3.70192496e+05	5.72130801e+04	2.90380811e+03	-9.30007187e+04
2.04530835e+04	1.30515711e+05	-2.09155123e+05	3.29230252e+05
-2.72609525e+05	2.24754968e+05	-1.13555129e+05	4.83907812e+03
6.00640750e+05	-9.03161000e+05	-1.32052625e+06	-7.68751656e+05
6.89808323e+05	-8.99039372e+05	8.58207535e+05	-5.47705890e+05
6.22346457e+05	-6.76418319e+05	9.17710056e+05	-7.41741574e+05
3.90847592e+05	-6.70372655e+05	5.46352211e+05	-5.34879229e+05

9.64259402e+05	-8.25297373e+05	6.80965773e+05	-7.28557079e+05
5.32021420e+05	-3.81707202e+05	4.87044761e+05	-8.42563602e+05
8.99635439e+05	-7.69744341e+05	9.59040812e+05	-5.54317641e+05
-2.03050500e+05	4.21485875e+05	7.84831000e+05	-1.25803488e+05
4.57800766e+05	6.96441172e+04	-7.71659502e+04	1.31485105e+05
-2.43193086e+05	1.29054336e+04	5.17761143e+04	-4.92687617e+04
5.12354167e+04	4.79323237e+04	-5.13230068e+04	9.50726074e+04
-9.65919551e+04	3.77465010e+04	-2.18160986e+04	-1.60560190e+04
5.33367480e+03	2.04291699e+04	-4.00712676e+04	7.25347285e+04
-6.29503672e+04	5.48696855e+04	-2.70707461e+04	2.44162500e+03
-1.04081025e+06	-3.99799812e+05	1.95886875e+04	-3.64077930e+04
-2.35456857e+05	2.45428125e+03	-4.66048828e+03	-2.13582383e+04
5.10959941e+04	-9.24354688e+03	2.35004102e+03	-1.62758066e+04
-1.03983438e+04	-3.65313477e+03	2.14747852e+03	-2.26826572e+04
4.50906934e+04	-3.43622334e+04	3.93552476e+04	-7.70018555e+03
-2.16435781e+04	-4.79234961e+03	7.06368359e+03	-2.26871328e+04
6.20531562e+04	-4.09852764e+04	4.38461719e+03	1.21464062e+03
1.20118988e+06	4.04693625e+05	-8.66101875e+04	3.67343516e+04
1.49534656e+05	-1.80851992e+04	4.56731279e+04	9.27006287e+03
-1.39122725e+04	4.48628906e+02	-1.69335132e+03	-1.33850029e+04
2.01024243e+04	-1.50337939e+03	-2.33650195e+03	1.59029482e+04
-1.59462827e+04	9.99524829e+03	-7.96193799e+03	-1.48306641e+03
1.31158203e+02	2.75257812e+03	-6.63196094e+03	1.36314258e+04
-1.19501348e+04	1.13133701e+04	-5.61769922e+03	5.21421875e+02
-8.68123000e+05	-5.62074812e+05	5.79209375e+04	-1.13204277e+05
5.05112695e+03	-3.03089648e+04	-1.58608105e+04	-6.30178125e+03
4.38841797e+02	-1.82386328e+03	-1.83513477e+03	3.12557617e+03
-7.67833789e+03	1.13357422e+03	-4.94506836e+02	-4.13143262e+03
7.94531958e+03	-6.15560840e+03	5.67162207e+03	-3.29317969e+03
6.41275391e+02	-2.00076953e+03	3.97909375e+03	-7.05752734e+03
8.25479102e+03	-6.89158301e+03	6.91616016e+03	-2.86348438e+03
8.04240250e+05	6.59602812e+05	-1.71990750e+05	1.81713641e+05
-6.85634512e+04	6.38469453e+04	-9.17943750e+03	1.88269746e+04
-2.87615039e+03	2.95798926e+03	2.56083691e+03	-4.95161230e+03
6.82574463e+03	-4.75576074e+03	4.70579297e+03	-8.12162598e+02
-1.47797852e+01	3.96464966e+02	-4.57863281e+02	-8.84943359e+02
4.36945312e+02	-1.98050781e+02	-5.28750000e+02	1.85305859e+03
-1.66100977e+03	1.79201074e+03	-1.07273438e+03	6.86250000e+01
-7.82819250e+05	-7.39668781e+05	2.45436250e+05	-2.42413832e+05
8.93588984e+04	-9.61268418e+04	4.20042832e+04	-3.61724766e+04
1.36150703e+04	-1.00405215e+04	1.72375391e+03	3.28125000e-01
-3.18834961e+03	2.63087402e+03	-2.99169043e+03	1.61059472e+03
-9.13783691e+02	4.97390625e+02	-2.37849609e+02	3.96154297e+02
-3.95308594e+02	2.08632812e+01	2.63281250e+02	-7.90039062e+02
1.31726953e+03	-1.11944629e+03	1.16683203e+03	-5.80773438e+02
4.86638125e+05	8.81060812e+05	-3.32794023e+05	3.02778789e+05
-1.15856555e+05	1.27548352e+05	-6.63765273e+04	5.34918809e+04
-2.61165166e+04	1.85445293e+04	-7.79107031e+03	3.87649658e+03

1.57474609e+02 -1.15858740e+03 2.19424512e+03 -1.67949634e+03  
 1.60777325e+03 -1.10269336e+03 7.99752930e+02 -6.96195312e+02  
 4.01042969e+02 -2.23554688e+02 -4.03164062e+01 3.56046875e+02  
 -4.63496094e+02 6.07197754e+02 -7.49734375e+02 4.80906250e+02  
 -5.96184750e+05 -8.75254281e+05 1.37966062e+05 -3.01428867e+05  
 1.17674311e+05 -1.28468904e+05 6.78391992e+04 -5.55662598e+04  
 3.02082080e+04 -2.14153311e+04 1.05927930e+04 -6.36171582e+03  
 2.14850098e+03 -7.57977539e+02 -4.97327637e+02 5.63142578e+02  
 -6.41142578e+02 4.00748047e+02 -2.09126953e+02 1.27871094e+01  
 1.69003906e+02 -3.68386719e+02 5.72960938e+02 -7.39583984e+02  
 9.10634766e+02 -9.36195312e+02 9.84808594e+02 -9.05078125e+02  
 7.11910000e+05 8.65710219e+05 6.88513906e+04 2.98055125e+05  
 -1.20050141e+05 1.29084285e+05 -6.91089629e+04 5.74737686e+04  
 -3.36003330e+04 2.41738813e+04 -1.35906367e+04 8.73830676e+03  
 -4.39641455e+03 2.44713159e+03 -9.36895508e+02 4.74003296e+02  
 -1.62994751e+02 2.35019531e+02 -3.28191406e+02 4.85701172e+02  
 -6.61207031e+02 8.11218750e+02 -9.63375000e+02 1.09633984e+03  
 -1.19300781e+03 1.23343750e+03 -1.14414453e+03 1.14393750e+03  
 -1.49576838e+06 -9.74513625e+05 6.79518125e+04 -2.36844000e+05  
 8.18303086e+04 -9.76207920e+04 5.39383486e+04 -4.53328955e+04  
 2.75742139e+04 -2.04749702e+04 1.26239570e+04 -8.87788159e+03  
 5.56519556e+03 -4.02893311e+03 2.84942310e+03 -2.42176318e+03  
 2.15075391e+03 -2.15109570e+03 2.19208008e+03 -2.30009375e+03  
 2.41060156e+03 -2.52148242e+03 2.62821875e+03 -2.70838086e+03  
 2.77701660e+03 -2.79424609e+03 2.68545312e+03 -2.54625000e+03  
 2.60766800e+06 1.02103950e+06 -2.11685125e+05 1.76002266e+05  
 -4.32864453e+04 6.52448984e+04 -3.79828945e+04 3.23210195e+04  
 -2.06700039e+04 1.59168984e+04 -1.07877832e+04 8.16239062e+03  
 -5.91672656e+03 4.76320312e+03 -3.90732617e+03 3.52429688e+03  
 -3.28318164e+03 3.22885547e+03 -3.22298438e+03 3.27053125e+03  
 -3.32969922e+03 3.39166797e+03 -3.45203125e+03 3.49839844e+03  
 -3.52842969e+03 3.53672852e+03 -3.43590625e+03 3.16990625e+03  
 -1.01341500e+06 -4.71714906e+05 9.87560000e+04 -8.77675391e+04  
 2.13039297e+04 -3.22749219e+04 1.86723516e+04 -1.58760234e+04  
 1.00455391e+04 -7.67075000e+03 5.10481250e+03 -3.79221875e+03  
 2.66811719e+03 -2.09164062e+03 1.66273438e+03 -1.47173438e+03  
 1.35187500e+03 -1.32440625e+03 1.32137500e+03 -1.34552344e+03  
 1.37472656e+03 -1.40642969e+03 1.43680469e+03 -1.45918750e+03  
 1.47189844e+03 -1.47337500e+03 1.41364844e+03 -1.30351758e+03  
 -9.78450500e+05 -8.07650000e+03 7.14668750e+03 -9.84375000e+01  
 8.35000000e+01 -3.15703125e+01 -3.46093750e+00 1.19248047e+01  
 -1.54843750e+01 6.83203125e+00 -8.48046875e+00 5.09375000e+00  
 -5.83984375e+00 4.34765625e+00 -4.64843750e+00 3.89843750e+00  
 -3.86328125e+00 3.18750000e+00 -3.10546875e+00 2.51171875e+00  
 -2.73242188e+00 2.07226562e+00 -3.20898438e+00 1.85546875e+00  
 -5.06542969e+00 2.42187500e-01 -1.51582031e+01 -7.38281250e-01]

## 8 Getting feature function

```
In [7]: testx=[]
        xb=list_image_test.T
        for i in range(len(xb)):
            x = xb[i].reshape((size_row, size_col))
            x = filt(x)
            x = np.ravel(x, order='C')
            testx.append(x)
        testx=np.array(testx)
        testy=vecY( list_label_test)
        Y = (testx).dot(A)
```

## 9 Binary Classification:FN,FP,TN,TP

```
In [10]: FN = []; FP = []; TN = []; TP = []

        for i in range(0, len(Y)):
            if(float(Y[i]) > 0):
                if(int(testy[i]) == 1):
                    TP.append(testx[i])
                else:
                    FP.append(testx[i])
            else:
                if(int(testy[i]) == 1):
                    FN.append(testx[i])
                else:
                    TN.append(testx[i])

        accuracy = ((len(TN)+len(TP))/10000)*100
        print("Accuracy: ",accuracy)
```

Accuracy: 50.81

## 10 Compute the Accuracy

```
In [11]: def computeMin(vec):
        vec = np.mat(vec)
        mean_ = np.mean(vec, axis=0)
        return mean_
```

## 11 Plot the average image of TP,FP,TN,FN

```
In [13]: f2 = plt.figure(2)
```



```

im_average = np.zeros((size_row * size_col, 10), dtype=float)
im_count    = np.zeros(10, dtype=int)

P1 = plt.subplot(2, 2, 1)
P1.set_title('False Negative')
im_average = computeMin(FN)
plt.imshow(im_average.reshape((size_row, size_col)), cmap='Greys', interpolation='None')
plt.axis('off')

P2 = plt.subplot(2, 2, 2)
P2.set_title('True Negative')
im_average = computeMin(TN)
plt.imshow(im_average.reshape((size_row, size_col)), cmap='Greys', interpolation='None')
plt.axis('off')

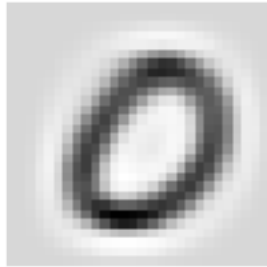
P3 = plt.subplot(2, 2, 3)
P3.set_title('False Positive')
im_average = computeMin(FP)
plt.imshow(im_average.reshape((size_row, size_col)), cmap='Greys', interpolation='None')
plt.axis('off')

P4 = plt.subplot(2, 2, 4)
P4.set_title('True Positive')
im_average = computeMin(TP)
plt.imshow(im_average.reshape((size_row, size_col)), cmap='Greys', interpolation='None')
plt.axis('off')

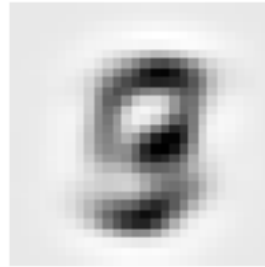
```

```
Out[13]: (-0.5, 27.5, 27.5, -0.5)
```

False Negative



True Negative



False Positive



True Positive

