# PLA DESIGN REPORT

The project can use two part to finish.

Firstly, we need to create some train data for the PLA to use.

## The step can follow this:

According to that w0+w1\*x1+w2\*x2 > 0 or w0+w1\*x1+w2\*x2 > 0

Then we can append x1,x2,’+’ or x1 ,x2,’-’into list1 and list2

t list

Use the ‘sys’ to input the [w0,w1,w2] m n

Make another new list to make the list1 and list 2 together

Generate file called train.txt

## The codes :

import random

import numpy as np

import sys

list = [sys.argv[1].split(',')]

w0 = int(list[0][0][1])

w1 = int(list[0][1])

w2 = int(list[0][2][0])

m = int(sys.argv[2])

n = int(sys.argv[3])

positives = []

neigitivs = []

fo1 = open('train.txt','w')

for i in range(m):

while True:

x1 = np.random.uniform(-10,10,500)

x2 = np.random.uniform(-10,10,500)

if x1[i] \* w1 + x2[i] \* w2 + w0 > 0:

positives.append('{},{},{}'.format(x1[i],x2[i],"+"))

print(x1[i],x2[i],'+')

break

for j in range(n):

while True:

x1 = np.random.uniform(-10,10,500)

x2 = np.random.uniform(-10,10,500)

if x1[j] \* w1 + x2[j] \* w2 + w0 < 0:

neigitivs.append('{},{},{}'.format(x1[j],x2[j],"-"))

print(x1[j],x2[j],'-')

break

total = neigitivs + positives

random.shuffle(total)

for i in range(len(total)):

fo1.write(total[i]+'\n')

fo1.close()

Then we can get the data from the “train.txt”, and we can use the PLA to get the result.

## The step can follow this:

Get the data from “train.txt”

Deal with the data from train.txt

Chang the “+” into “1”,and change the “-”into “-1”

Then we need to make the x1,x2 into a array

Label into the other array

And use the Gradient descent algorithm to get the final answer

Oupt the picture we need

## The codes :

import sys

import numpy as np

import matplotlib.pyplot as plt

file = open(sys.argv[1],'r')

data\_set = []

data\_label = []

data\_back = []

data\_reback1 = []

data\_reback2 = []

for line in file:

line=line.rstrip("\n")

line1 = line.split(',')

data\_back.append(line1)

file.close()

for i in range(len(data\_back)):

if data\_back[i][-1] == '+':

data\_reback1.append('{},{},{}'.format(data\_back[i][0],data\_back[i][1],'1'))

else:

data\_reback2.append('{},{},{}'.format(data\_back[i][0],data\_back[i][1],'-1'))

data\_reback = data\_reback1+data\_reback2

fo1 = open('3.txt','w')

for i in range(len(data\_reback)):

fo1.write(data\_reback[i]+'\n')

fo1.close()

file1 = open('3.txt','r')

for line in file1:

line=line.rstrip("\n")

line1 = line.split(',')

for i in range(len(line1)):

line1[i] = float(line1[i])

data\_set.append(line1[0:2])

data\_label.append(int(line1[-1]))

file1.close()

data = np.array(data\_set)

label = np.array(data\_label)

w = np.array([0, 0])

w0 = 0

alpha = 1

f = (np.dot(data, w.T) + w0) \* label

idx = np.where(f <= 0)

w\_countlist = []

while f[idx].size != 0:

point = np.random.randint((f[idx].shape[0]))

x = data[idx[0][point], :]

y = label[idx[0][point]]

w = w + alpha \* y \* x

w0 = w0 + alpha \* y

w\_countlist.append('{},{},{}'.format(w0,w[0],w[1]))

f = (np.dot(data, w.T) + w0) \* label

idx = np.where(f <= 0)

file2 = open('PLA.txt','w')

for i in range(len(w\_countlist)):

file2.write(w\_countlist[i]+'\n')

print([w\_countlist[i]])

file2.close()

x1 = np.arange(-10, 10, 0.01)

x2 = (w[0] \* x1 + w0) / (-w[1])

idx\_p = np.where(label == 1)

idx\_n = np.where(label != 1)

data\_p = data[idx\_p]

data\_n = data[idx\_n]

plt.plot(x1, x2,color='black')

plt.scatter(data\_p[:, 0], data\_p[:, 1], color='red',marker="+",label='+')

plt.scatter(data\_n[:, 0], data\_n[:, 1], color='blue',marker="o",label='--')

plt.legend(loc='best')

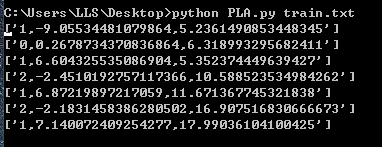
plt.show()

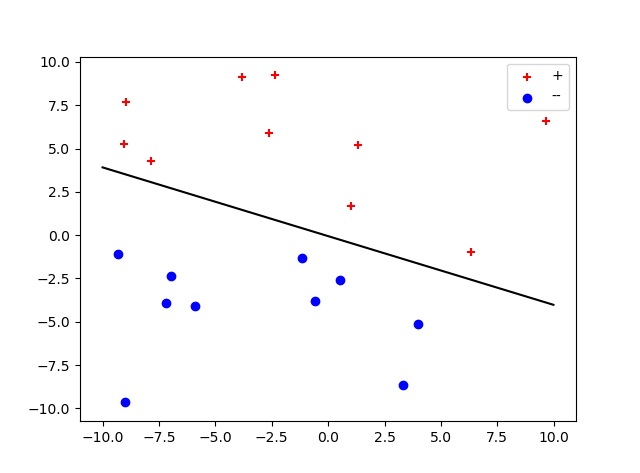
## The test

1. DataEmit.py [5,2,3] 10 10



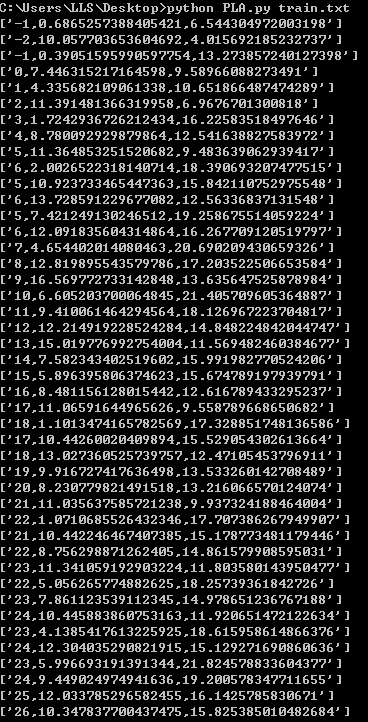
W0 ,w1,w2 we can see from the next picture:

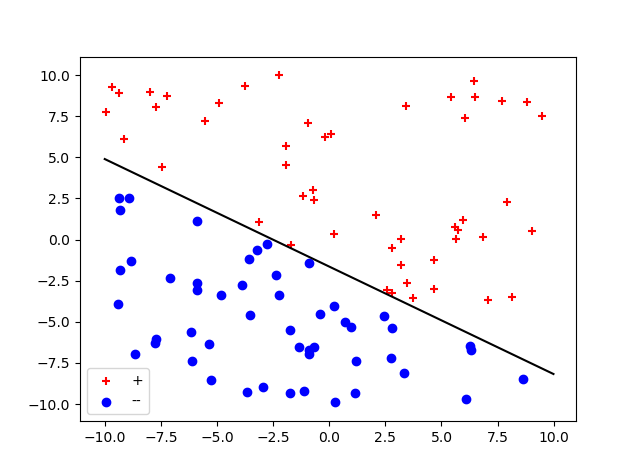




2. DataEmit.py [5,2,3] 50 50

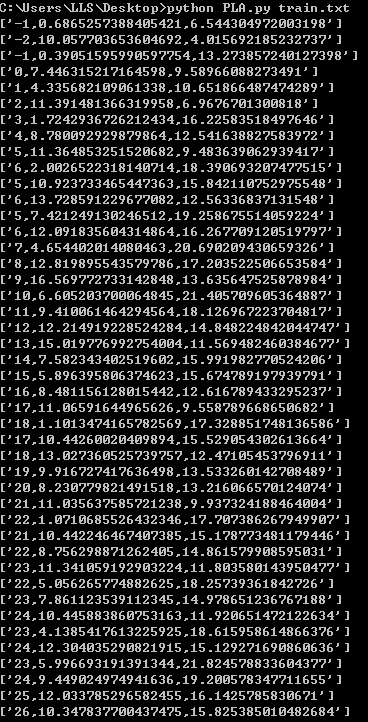


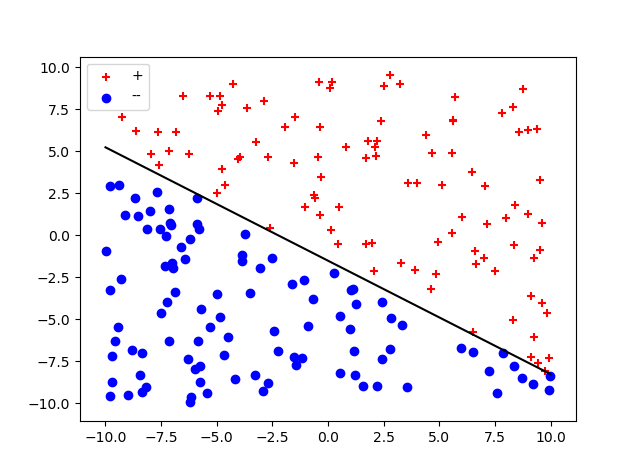




3. DataEmit.py [5,2,3] 100 100

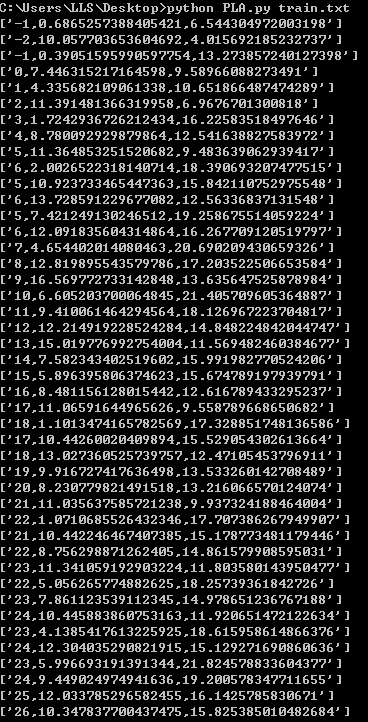


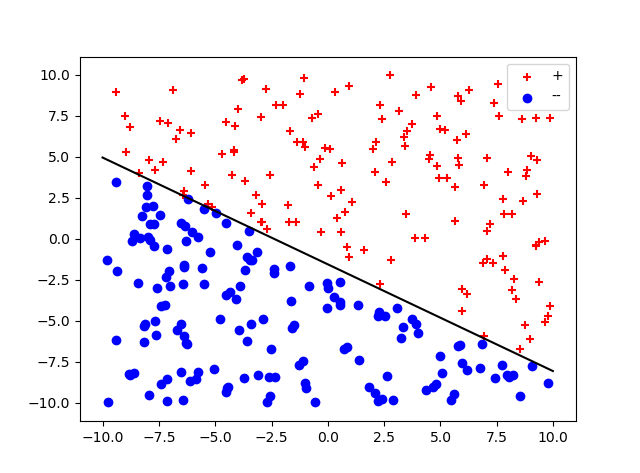




4. DataEmit.py [5,2,3] 150 150

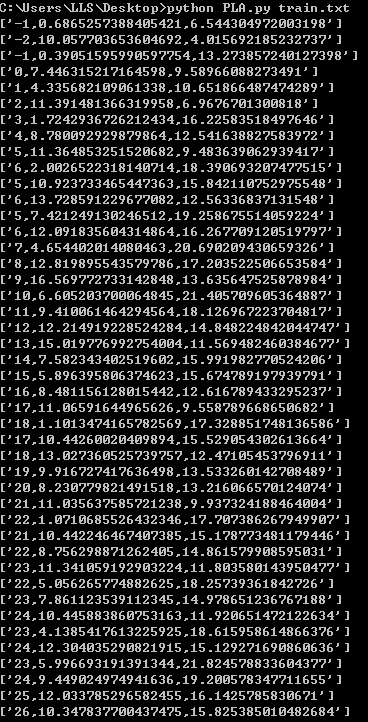


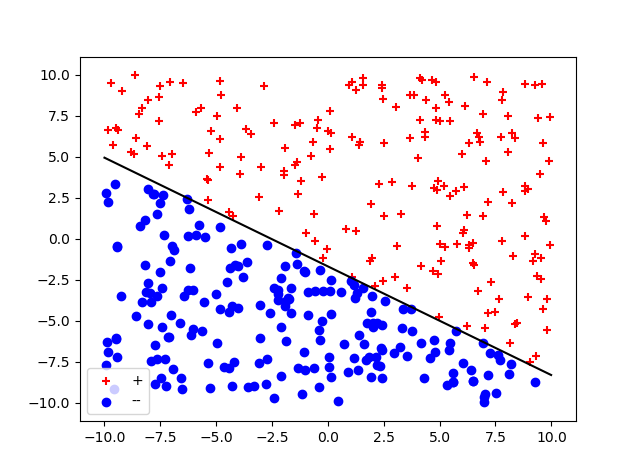




5. DataEmit.py [5,2,3] 200 200







## Summarize:

What I can learn from the test is that when the data getting bigger and bigger, the final result need more calculation to get a right number of w0,w1 ,w2 and when the data getting bigger and bigger we can see the line don’t change obviously.

## My own test

## Balanced:

|  |  |  |
| --- | --- | --- |
| DataEmit.py [5,3,8] 10 10 |  |  |
| DataEmit.py [5,3,8] 150 150 |  |  |
| DataEmit.py [5,3,8] 500 500 |  |  |

## Unbalanced:

|  |  |  |
| --- | --- | --- |
| DataEmit.py [5,3,8] 100 50 |  |  |
| DataEmit.py [5,3,8] 300 150 |  |  |
| DataEmit.py [5,3,8] 500 300 |  |  |