**附錄**

**程式說明：**

本程式利用Python編寫，Vs code編寫器，並從其中拷貝

共包含五個Python文檔：

myPLA.py && problem13.py && problem14.py && problem15.py && problem16.py

我覺得助教可能看著有顏色的代碼可能好一些XD

文檔myPLA.py

import numpy as np

#本程式的目的是實現PLA算法，並計算題目給定情形下wPLA的平均模數

#定義一個判別函式，用於判別分類是否完成

def Judge(X, y, w):

    #獲取資料的行數

    n = X.shape[0]

    num = np.sum(X.dot(w) \* y > 0)

    return num == n

#定義一個函式進行數據預處理

def preprocess(data):

    #獲取資料的行和列

    n, d = data.shape

    #從資料中分离X矩陣

    X = data[:, :-1]

    #左側增廣1

    X = np.c\_[np.ones(n), X]

    #分离y

    y = data[:, -1]

    return X, y

#我的PLA算法，X，y來自資料，eta是作業題目2中的係數，迭代次數max\_step置為無窮大

def PLA(X, y, eta=1, max\_step=np.inf):

    #獲取維度

    n, d = X.shape

    #按照d初始化向量

    w = np.zeros(d)

    #記錄迭代的次數

    t = 0

    #用於記錄元素的下標

    i = 0

    #記錄最後一個錯誤的下標

    last = 0

    #初始化模數

    ret=0

    while not(Judge(X, y, w)) and t < max\_step:

        if np.sign(X[i, :].dot(w) \* y[i]) <= 0:

            #迭代次數遞增

            t += 1

            w += eta \* y[i] \* X[i, :]

            #完成最後一個錯誤點的更新

            last = i

        #移動至下一個元素

        i += 1

        #當i達到n，重新置為0

        if i == n:

            i = 0

    ret = np.linalg.norm(w,ord=2,axis=0) # 對象是逐行獲取，求每行元素的平方和再開方

    return ret\*ret, t, last, w

#運行PLA算法，func為佔位，統計平均的wPLA模數，係數默認是1，迭代次數無窮

def f(func, X, y, n, eta=1, max\_step=np.inf):

    #創建一個新的矩陣，用來存放每次運行得到的wPLA

    result = []

    #資料是增廣矩陣，重新拼接起來

    data = np.c\_[X, y]

    for i in range(n):

        #對所給的資料進行洗牌

        np.random.shuffle(data)

        X = data[:, :-1]

        y = data[:, -1]

        #把PLA算法的第一個返回值，模數，給到result存儲

        result.append(func(X, y, eta=eta, max\_step=max\_step)[0])

    print(np.mean(result))

文檔problem13.py

import numpy as np

import myPLA as my

#將網頁上的資料下載到桌面，存儲到同一個目錄下，打開命名

data = np.genfromtxt("hw1\_train.dat.txt")

#獲取幾行幾列

n, d = data.shape

#分离出X

X = data[:, :-1]

#偏置增廣列向量1

X = np.c\_[np.ones(n), X]

#從資料剝離y

y = data[:, -1]

#problem 13，迭代1000次

print(my.PLA(X, y))

my.f(my.PLA, X, y, 1000, 1)

文檔problem14.py

import numpy as np

import myPLA as my

#將網頁上的資料下載到桌面，存儲到同一個目錄下，打開命名

data = np.genfromtxt("hw1\_train.dat.txt")

#獲取幾行幾列

n, d = data.shape

#分离出X

X = data[:, :-1]

#偏置增廣列向量2

X = np.c\_[np.ones(n)\*2, X]

#從資料剝離y

y = data[:, -1]

#problem 14，迭代1000次

print(my.PLA(2\*X, y))

my.f(my.PLA, 2\*X, y, 1000, 1)

文檔problem15.py

import numpy as np

import myPLA as my

from numpy.core.numeric import ones

#將網頁上的資料下載到桌面，存儲到同一個目錄下，打開命名

data = np.genfromtxt("hw1\_train.dat.txt")

#獲取幾行幾列

n, d = data.shape

#分离出X

X = data[:, :-1]

#偏置增廣列向量1

X = np.c\_[np.ones(n), X]

#從資料剝離y

y = data[:, -1]

ret = np.linalg.norm(X,ord=None,axis=1) # 對象是逐列，求每行元素平方和開方

r,c=data.shape

#定義一個X1存儲標準化後的資料

X1=ones([r,c])

for i in range(r):

    X1[i]=np.array(X[i]/ret[i])

#problem 15，迭代1000次

print(my.PLA(X1, y))

my.f(my.PLA, X1, y, 1000, 1)

文檔problem16.py

import numpy as np

import myPLA as my

#將網頁上的資料下載到桌面，存儲到同一個目錄下，打開命名

data = np.genfromtxt("hw1\_train.dat.txt")

#獲取幾行幾列

n, d = data.shape

#分离出X

X = data[:, :-1]

#偏置增廣列向量0

X = np.c\_[np.ones(n)\*0, X]

#從資料剝離y

y = data[:, -1]

#problem 16，迭代1000次

print(my.PLA(X, y))

my.f(my.PLA, X, y, 1000, 1)