類神經網路HW1 感知機實驗報告

資工碩一113522053 蔡尚融

目錄

[一. 程式執行說明 3](#_Toc180003691)

[基礎題操作介面: 3](#_Toc180003692)

[進階題操作介面: 5](#_Toc180003693)

[二. 程式碼簡介 7](#_Toc180003694)

[基本題: 7](#_Toc180003695)

[Select(): 7](#_Toc180003696)

[Split\_data(): 8](#_Toc180003697)

[Data\_update(): 8](#_Toc180003698)

[W\_update(): 9](#_Toc180003699)

[training(): 9](#_Toc180003700)

[Testing(): 10](#_Toc180003701)

[進階題: 11](#_Toc180003702)

[Select\_multi(): 11](#_Toc180003703)

[ndGraphInitialize(): 12](#_Toc180003704)

[ndTrain(): 13](#_Toc180003705)

[nw\_update(w,ax,line,canvas): 15](#_Toc180003706)

[nd\_surface(w,ax): 16](#_Toc180003707)

[ndTest(): 16](#_Toc180003708)

[三. 實驗結果 18](#_Toc180003709)

[2Ccircle1 18](#_Toc180003710)

[2Circle1 19](#_Toc180003711)

[2CloseS 19](#_Toc180003712)

[2CloseS2 20](#_Toc180003713)

[2CloseS3 21](#_Toc180003714)

[2cring 22](#_Toc180003715)

[2CS 23](#_Toc180003716)

[2Hcircle1 23](#_Toc180003717)

[2ring 23](#_Toc180003718)

[perceptron1 24](#_Toc180003719)

[perceptron2 25](#_Toc180003720)

[四. 進階題實驗結果 26](#_Toc180003721)

[2Circle2 26](#_Toc180003722)

[4satellite-6 26](#_Toc180003723)

[5CloseS1 28](#_Toc180003724)

[8OX 28](#_Toc180003725)

[C3D 29](#_Toc180003726)

[C10D 30](#_Toc180003727)

[IRIS 31](#_Toc180003728)

[perceptron3 32](#_Toc180003729)

[perceptron4 33](#_Toc180003730)

[wine 34](#_Toc180003731)

[xor 35](#_Toc180003732)

[五. 實驗結果總結。 35](#_Toc180003733)

# 程式執行說明

報告內容

此程式作業實作包含:

基礎題

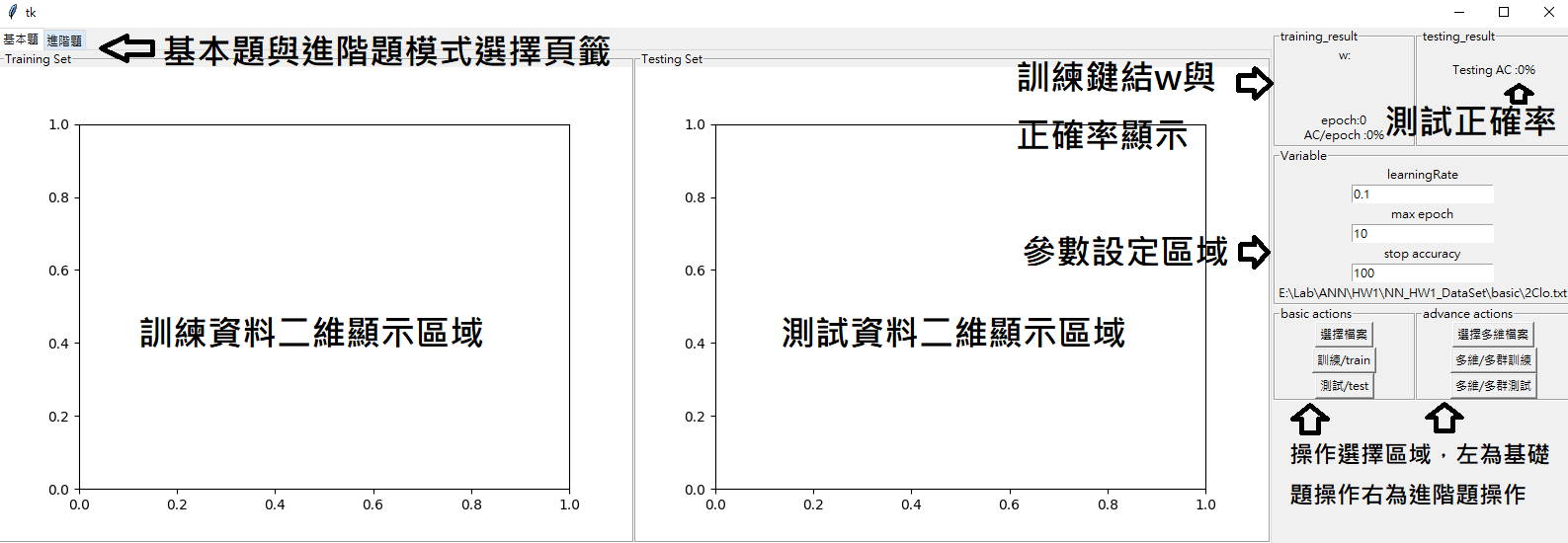
加分題

a. 三維資料圖形顯示介面。

b. 能夠處理多維資料(四維以上)。

d. 可以辨識兩群以上的資料。

## 基礎題操作介面:



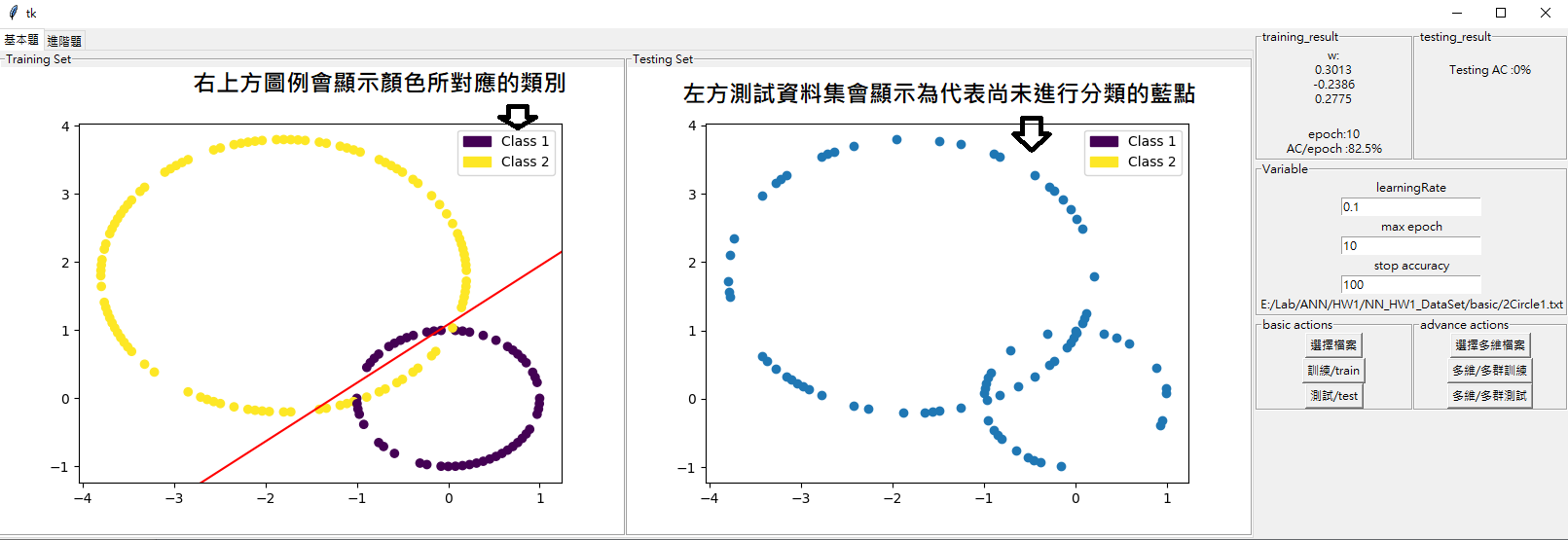
操作步驟:

Step1: 於左上方選擇基礎題頁籤。

Step2: 於右方參數設定區域分別設定learningRate(學習率)、max epoch(epoch 達到多少停止訓練)、stop accuracy(達到多少準確率停止訓練)。

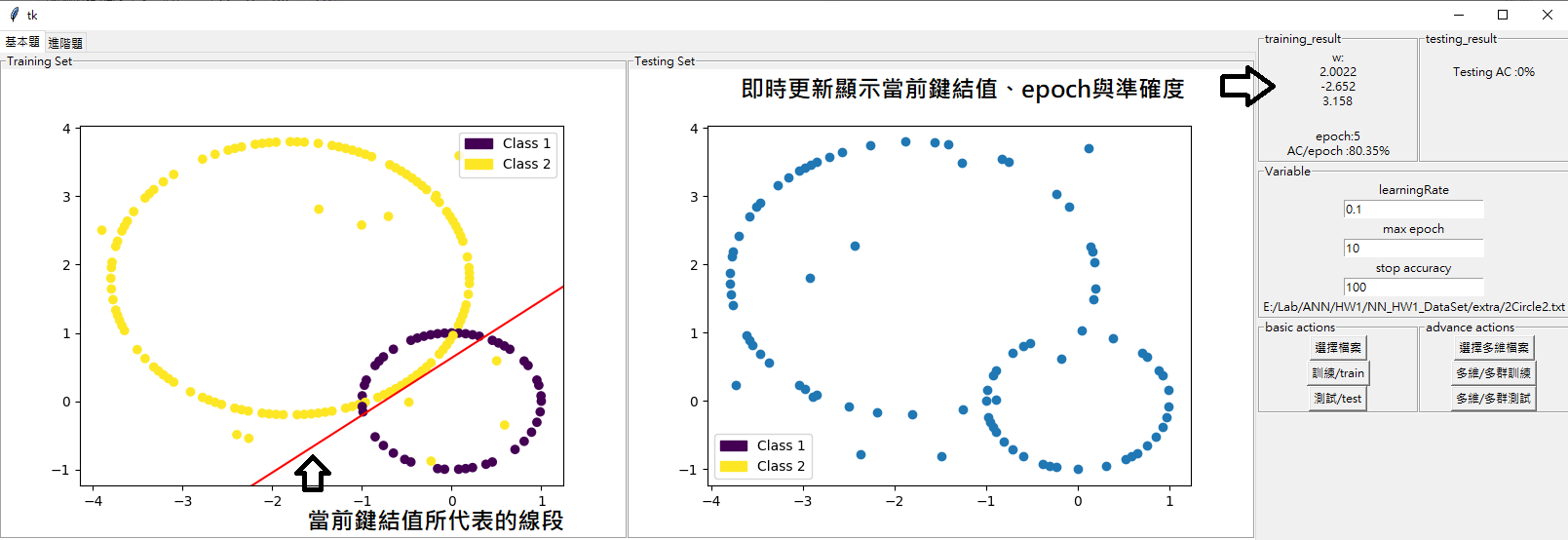
Step3: 點擊右下方basic actions中的”選擇檔案” Button。

此時畫面會顯示選取檔案之資料點如下:



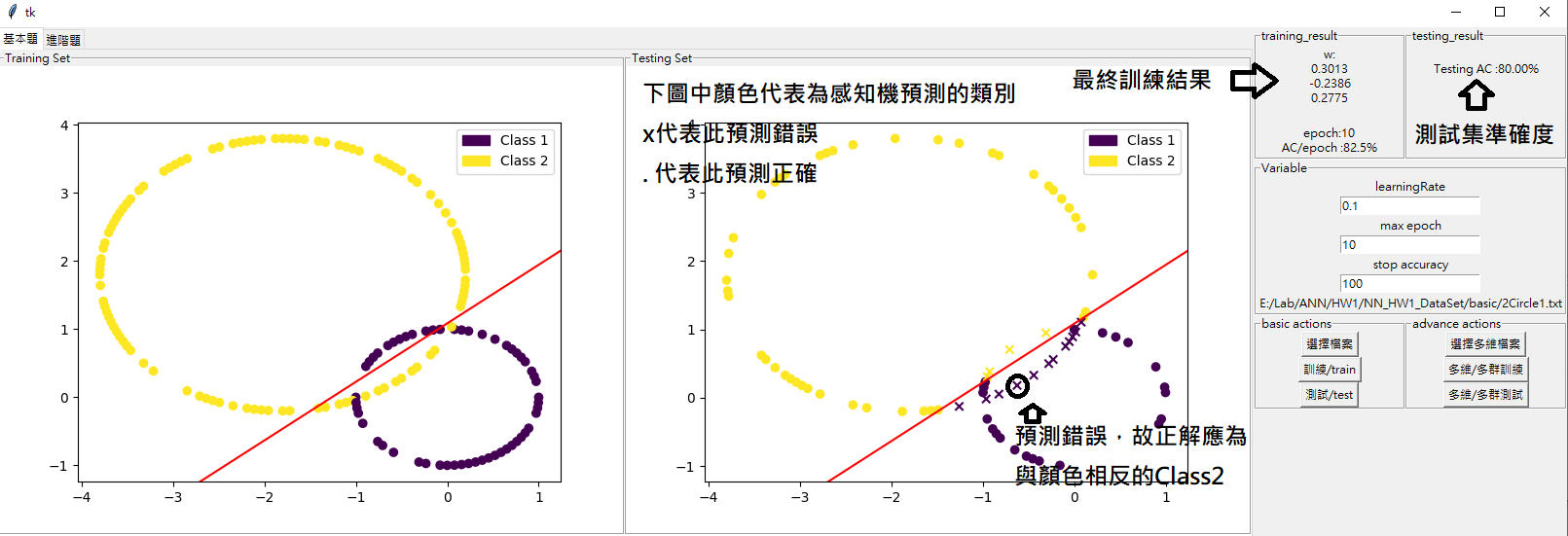
Step4: 點擊右下方basic actions中的 ”訓練/train” Button。

此時畫面會即時顯示訓練中狀態如下(示意圖不小心選到extra中的題目，故有雜點，但不影響說明)



Step5: 待訓練完成，點擊右下方basic actions中的 ”測試/Test” Button進行測試。

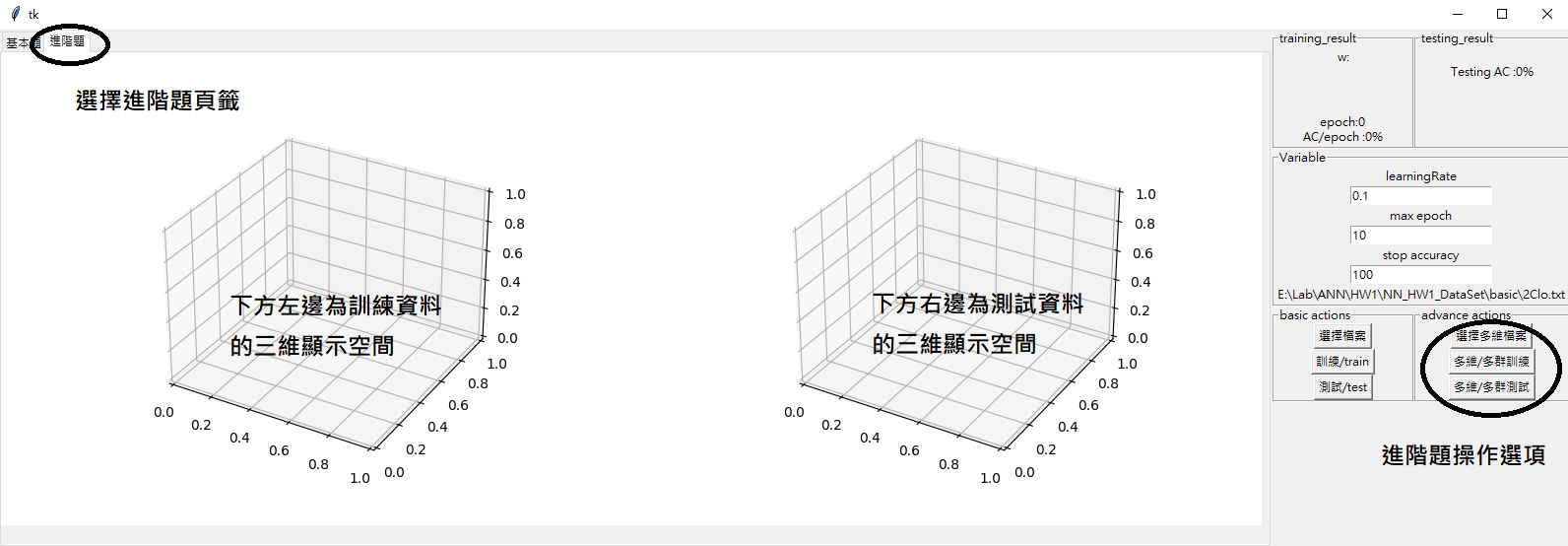
結果畫面將顯示預測判別結果如下:



Testing Set圖形顯示區域中顏色代表感知機的預測結果，符號”x”代表此筆預測為錯誤的(正確類別與顏色相反)，符號 ” . ” 代表此筆預測為正確的(正確類別與顏色相同)。

## 進階題操作介面:

Step1:於上方頁籤中選擇進階題頁籤。



Step2: 於右方參數設定區域分別設定learningRate(學習率)、max epoch(epoch 達到多少停止訓練)、stop accuracy(達到多少準確率停止訓練)。

Step3: 點擊右下方Advance actions中的”選擇多維檔案” Button。

左方顯示區域會展示3維以下(3維或2維的資料圖形)。

一張含有 文字, 圖表, 行, 繪圖軟體 的圖片

自動產生的描述

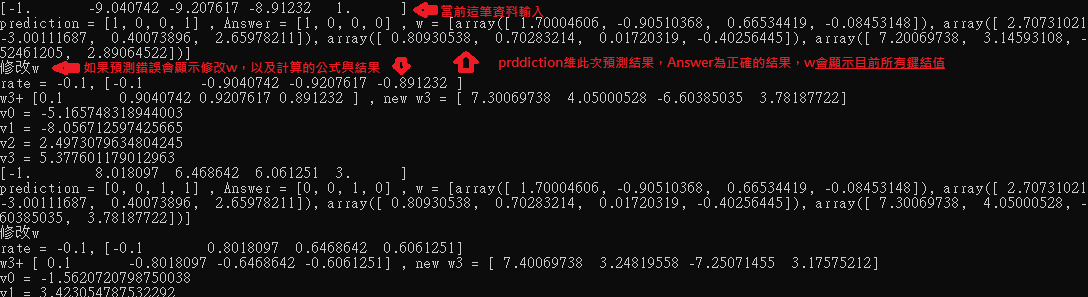
Step4: 點擊右下方Advance actions中的 ” 多維/多群訓練” Button。

會顯示使用多個面/線段對資料進行切割的結果，並於右上方顯示正確率。

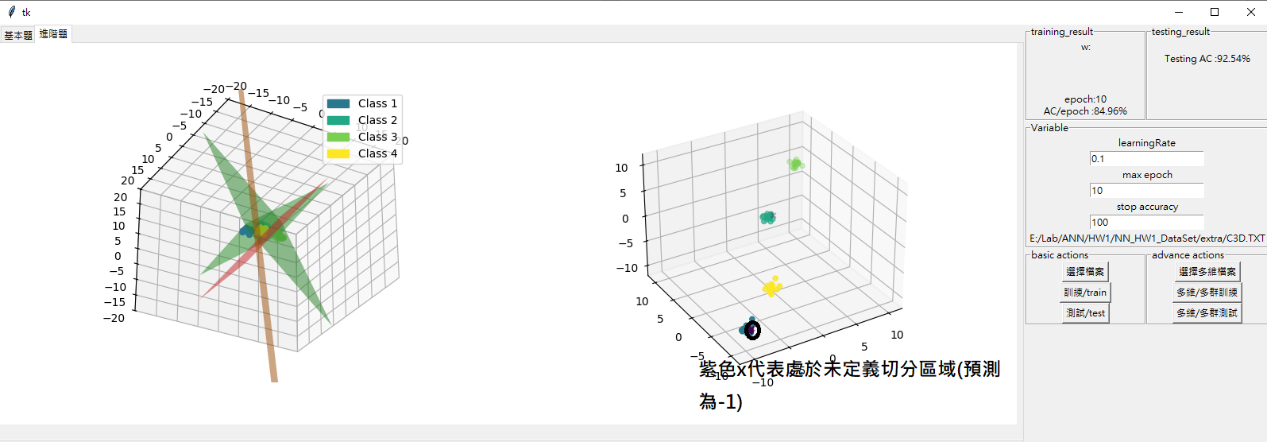
一張含有 文字, 圖表, 螢幕擷取畫面, 行 的圖片

自動產生的描述

(若超過4維資料不會顯示圖形，會於右上方Traning Result中顯示測試準確率，並於Consol視窗中顯示鍵結值)



Step5: 待訓練完成，點擊右下方basic actions中的 ”多維/多群測試” Button進行測試。

(若超過4維資料不會顯示圖形，會於介面右上方Result Area中顯示測試準確率，並於consol中顯示Code:每個感知機預測的結果、prediction:該code結果傳換為預測類別後的數值，以及該筆資料正確的類別)。

一張含有 螢幕擷取畫面, 樣式, 黑色, 藝術 的圖片

自動產生的描述

# 程式碼簡介

## 基本題:

### Select():

介面中Basic action下的選擇檔案Button會觸發select() function，此function的主要工作為透過np.loadtxt()去讀取所選的檔案，並讀取data[: , 2]中的所有類別資料來判斷總共有哪些類別cata(用在確認類別總數、以及進階題決定需要幾個感知機)。

這個Function會接著呼叫split\_data()、data\_update()

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

### Split\_data():

透過random來隨機選取資料中的2/3的index作為training data，剩下的1/3資料作為test data。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

### Data\_update():

此function的主要功能是在選擇新的資料時或重新開始訓練時將圖表進行初始化，包括將資料點繪製在Traning Set以及Testing Set兩張二維圖表上，並未兩張圖表都加上圖例legend。

一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述

### W\_update():

此function的主要目的是讓程式能夠在感知機訓練時能夠將鍵結值所代表的線段動態更新至圖表上，主要透過對於data\_update()中宣告的line2D物件line進行line.set\_data()操作來進行線段更新。

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

### training():

Training function中包含了主要的感知實作(主要步驟如下):

1. 首先透過w = np.random.rand(3) \* 0.01將鍵結值以及bias進行隨機初始化
2. 呼叫w\_update將初始化的鍵結直線段繪製至圖表上。

一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

1. 接著進入epoch迴圈，在epoch迴圈下用correct變數計算正確率(初始化為train data 的資料筆數，每當錯誤時-1)，接下來會開始迭代每一筆trainging data的迴圈，在這個迴圈中，會於輸入資料前方新增-1用於與bias相乘，接著計算v = np.dot(w,t\_data[0:3])也就是w與輸入資料的內積，並用y = cata[1] if v>0 else cata[0] if v<0 else 0來計算出該筆資料的預測結果是哪個種類。接下來程式碼透過if y !=t\_data[3]:來檢查此筆預測是否正確，若預測結果錯誤則對w的值進行修改，先透過rate = learningRate if y < t\_data[3] else -learningRate來判斷修正的方向，並w = w + rate\*t\_data[0:3]讓w等於learning rate乘上該筆輸入的資料。

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

### Testing():

此function於使用者點擊介面中basic aciton下，”訓練/Train” Button時觸發，會透過test\_result矩陣紀錄所有test\_data迭代進行 v = np.dot(w,t\_data[0:3])

矩陣內積的結果(同樣會於每筆輸入前插入-1用於與bias項相乘)。

一張含有 文字, 螢幕擷取畫面, 字型, 陳列 的圖片

自動產生的描述

接著將test\_result中的所有預測結果與正確答案進行比對得到correct\_index矩陣，最後將test\_data[correct\_index]用scattor ”o”的符號以及預計的顏色繪製至圖表上並讓不正確的點透過 dots3 = ax2.scatter(np.delete(x,correct\_index,axis=0),np.delete(y,correct\_index,axis=0),c=np.delete(test\_result,correct\_index,axis=0),marker="x",cmap='viridis',)以符號” x ”繪製至圖表上。

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

## 進階題:

### Select\_multi():

介面中advance action下的 ”選擇多維檔案” Button會觸發select\_multi() function，此function的實作與基礎題主要差在新增了透過cata(種類總數)來設定n(總共要使用幾個感知機)的判別式，並透過data.shape判斷如果維度小於4再呼叫ndGraphInitialize()進行三維圖表的初始化。

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

### ndGraphInitialize():

根據資料是二維或三維來初始化訓練與測試兩張圖表，並新增圖例。

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

### ndTrain():

由介面中advance action下的 ”多維/多群訓練” Button觸發，為主要進行訓練的function。感知機部分的實作與基礎題的training function大致相同，主要的差別在w初始化時會使用dim+1(dimention)、將多個感知機針對一筆資料的預測結果存成code矩陣，用每一位代表一個種類來進行分類(ex:1000代表class1、0100代表class2、0010代表class3、0001代表class4)藉此達成多群的分類。

w初始化時會使用dim+1(dimention)藉此可以實現加分題b. 能夠處理多維資料(四維以上):

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

將多個感知機針對一筆資料的預測結果存成code矩陣藉此可實現加分題d. 可以辨識兩群以上的資料:

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

判斷正確與否並修正w部分(後方為圖表更新):

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

在修正w值時則根據code中哪一位的結果不正確就對該w進行修正(ex: code = [0 ,1, 1, 0],Ans =[0, 1, 0, 0]時就對code[2]的w值也就是w2進行修改)透過surface矩陣紀錄與更新當前多個鍵結值所代表的多個平面或線段(透過對ax.plot\_surface物件進行.remove()並重新繪製來達成更新面的效果)。

### nw\_update(w,ax,line,canvas):

這個funciton用於進階題中，二維資料的線段更新(於三維空間中繪製或更新二維線段)。

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

### nd\_surface(w,ax):

用於繪製三維資料訓練時，w鍵結值所代表的平面。

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

### ndTest():

由介面中advance action下的 ”多維/多群測試” Button觸發，原理與基礎題的testing function大致相同，會先對所有測試資料透過內積算出預測結果code儲存至predictions矩陣中。一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

再判別預測正確的index有哪些(correct\_index)，計算正確率並將正確的預測與錯誤的預測用預測出來的類別作為顏色，分別用不同的符號”o”與”x”繪製至圖表上。

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

Code轉換回prediction class部分的程式碼。(if n==1的部分判別如果總共只有一個感知機則用不同的類別定義方法，非0100等而是單純0或1)一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

# 實驗結果

### 2Ccircle1

由於圖形非線性可分割，訓練10個epoch與2個epoch的結果相似，訓練準確度分別達到53.75%與60%，測試準確度分別為46.25%與52.5%。

一張含有 螢幕擷取畫面, 文字, 圖表, 繪圖 的圖片

自動產生的描述

一張含有 螢幕擷取畫面, 圖表, 繪圖, 行 的圖片

自動產生的描述

### 2Circle1

這題也非線性可分割問題，在訓練2 epoch後模型找到兩個圓交接處附近的一條線，訓練準確度為83.12%測試準確度為86.25%。

一張含有 文字, 圖表, 行, 繪圖 的圖片

自動產生的描述

訓練10 epoch的訓練準確度為82.25%測試準確度為87.5%

一張含有 圖表, 文字, 螢幕擷取畫面, 繪圖 的圖片

自動產生的描述

### 2CloseS

這題為線性可分割問題，感知機在epoch 2就獲得了訓練集與測試集都為100%準確率的成果。

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

此時就算將epoch設為10，程式也會在epoch 2時因為已經達到設定的stop accuracy 100%停止。

一張含有 圖表, 行, 繪圖, 文字 的圖片

自動產生的描述

### 2CloseS2

此題雖為線性可分割問題，但兩類點較為接近，第一次測試epoch設為10，感知機在epoch 2時因達到設定的accuracy 100%停止，但於testing set中得到了一個錯誤點，準確度來到98.51%。

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

觀察訓練過程會發現線段跳躍的幅度較大(下圖)，故可將learning rate下降

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

下圖將learning rate下降後線段跳躍的幅度顯著減少，並於epoch 4時找到訓練集與測試集階為100%準確度的結果。

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

我們也可以嘗試反向將learning rate加大到0.8

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

可以看到感知機無超過epoch 10都無法找到訓練及準確度100%的小縫隙，這是因為每次跳動修正的幅度過大導致。

### 2CloseS3

這題也非線性分割問題，感知機在epoch 10得到訓練資料準確度99.4%，測試準確度99.1%的結果。

一張含有 文字, 圖表, 繪圖軟體, 軟體 的圖片

自動產生的描述

第二次測試也得到相似的結果，感知機在epoch 10得到訓練資料準確度99.4%，測試準確度98.8%

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述

### 2cring

這題為線性可分割問題，感知機在epoch 2就獲得了訓練集與測試集都為100%準確率的成果。

一張含有 文字, 螢幕擷取畫面, 圖表, 繪圖 的圖片

自動產生的描述

### 2CS

這題也為線性可分割問題，感知機與上題一樣在epoch 2就獲得了訓練集與測試集都為100%準確率的成果。

一張含有 文字, 螢幕擷取畫面, 行, 圖表 的圖片

自動產生的描述

### 2Hcircle1

這題也為線性可分割問題，感知機與上題一樣在epoch 3就獲得了訓練集與測試集都為100%準確率的成果。

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

### 2ring

這題也為線性可分割問題，感知機與上題一樣在epoch 2就獲得了訓練集與測試集都為100%準確率的成果。

一張含有 文字, 行, 繪圖, 圖表 的圖片

自動產生的描述

### perceptron1

這題為線性可分割問題，但第一次測試中訓練及抽到的兩點皆為class 1，導致感知機沒有進行到訓練，結果為訓練集100%，測試集0%準確度

一張含有 軟體, 螢幕擷取畫面 的圖片

自動產生的描述

第二次測試雖抽到了兩點不同class可以進行訓練，但由於訓練資料的代表性不足(沒有很好的表現出母體分布特徵)導致感知機沒有找到真正的母體分布規律，訓練集100%，測試集0%準確度。

一張含有 文字, 行, 圖表, 繪圖 的圖片

自動產生的描述

第三次測試，雖然訓練資料集展現了母體的主要規律，但感知機仍以一點點的誤差於測試集中只拿到50%的準確度。

一張含有 文字, 行, 圖表, 繪圖 的圖片

自動產生的描述

第四次測試，雖然不是最完美的分割線，但感知機成功得到了兩個100%的結果。

一張含有 螢幕擷取畫面, 文字, 行, 繪圖 的圖片

自動產生的描述

### perceptron2

這題的母體資料分布為XOR，故單個感知機無法達成完全正確的分類，最終得到結果訓練集100%測試集0%的準確度。

一張含有 文字, 圖表, 行, 繪圖 的圖片

自動產生的描述

# 進階題實驗結果

### 2Circle2

此題包含多群分類總共4種類別，但class3中沒有資料，程式使用了4條(對應4個class)嘗試進行分類，其中綠色線段對應沒資料的class3故訓練過程中沒更新，而由於class4的黃點分布較為隨機，紅色線段沒能成功判別初class4的資料點，測試資料中的紫色點代表項是code 0110、1010等沒有被定義的類別，是直接使用等同類別數量的感知機所會碰到的問題，最後得到訓練60.12%測試78.16%的結果。

一張含有 圖表, 文字, 螢幕擷取畫面, 行 的圖片

自動產生的描述

第二次測試3 epoch也得到相似的結果。

一張含有 文字, 圖表, 螢幕擷取畫面, 行 的圖片

自動產生的描述

### 4satellite-6

這題為4維資料，總共有6種類別。

從consol中可以觀察訓練與修正過程:

一張含有 文字, 螢幕擷取畫面, 字型, 黑色 的圖片

自動產生的描述

上圖中為一筆資料的訓練與修正過程，圖中間顯示這筆資料的預測結果prediction code為010001但正確結果為000010，故於最下方對w1、w4、w5進行了修正。可看到epoch10的最終訓練準確度為42.68%，最下方list為每個訓練epoch的準確度，可以看到最高在42%就難以繼續上升了。

同樣可於consol中觀察測試結果: 一張含有 文字, 螢幕擷取畫面, 黑與白, 圖書 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

最終epoch 10達到訓練集42.68%，測試46.82%的結果。

### 5CloseS1

這題較為簡單，二維且只有兩類，程式使用了兩個感知機來預測並於epoch2就達成了兩個100%的結果。

一張含有 文字, 圖表, 行, 螢幕擷取畫面 的圖片

自動產生的描述

### 8OX

這題為8維資料、共3個class的分類，程式使用三個感知機。

第一次測試10個epoch， 訓練準確率達到63.33%，測試準確度為33%

一張含有 文字, 螢幕擷取畫面, 字型, 黑色 的圖片

自動產生的描述

第二次測試提升epoch到20，得到訓練集70%，測試集53.33%的結果有些微提升。

一張含有 文字, 字型, 螢幕擷取畫面, 黑色 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

第三次測試提升epoch到30，雖得到訓練集73.33%，但測試集53.33%的結果沒有提升。

一張含有 螢幕擷取畫面, 黑色, 文字, 黑與白 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

第四次測試提升epoch到40，雖得到訓練集80%的準確度看似有顯著的提升，但測試集的結果不僅沒有提升反而是大幅的下降到了剩下40%，可見明顯出現對於訓練資料過擬合的問題了。

一張含有 文字, 螢幕擷取畫面, 黑色, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

### C3D

第一次測試，5個epoch，訓練正確率達到79.7%測試正確率達到67.16%

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

第二次測試，10個epoch，訓練正確率達到89.47%測試正確率還是67.16%

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

第三次測試,共30個epoch訓練正確率達到91.73%測試正確率達到82.09%有顯著的提升。

一張含有 文字, 圖表, 繪圖, 繪圖軟體 的圖片

自動產生的描述

### C10D

這題為10維共4 class的分類，程式使用了4個感知機。

第一次測試10個epoch，訓練正確率達到57.89%測試正確率達到46.27%

一張含有 文字, 螢幕擷取畫面, 黑色, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

第二次測試提升epoch至20，結果沒有提升。

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

第三測試提升epoch至30，測試集的結果顯著提升至67.16%。

一張含有 文字, 螢幕擷取畫面, 黑色, 黑與白 的圖片

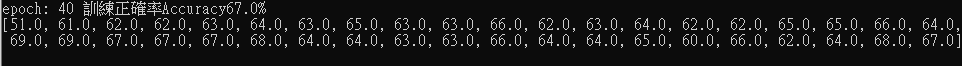
自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

### IRIS

第一次測試40 epoch，訓練集準確度67%測試集準確度62%



一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

第二測試10 epoch，訓練集準確度63%測試集準確度66%

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

另外分別測試20 epoch與30 epoch結果沒有顯著差別

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

### perceptron3

第一次測試，模型在epoch3時得到訓練集100%測試集50%的結果

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

第二次測試，模型在epoch3時得到訓練集100%測試集100%的結果

一張含有 文字, 圖表, 行, 繪圖 的圖片

自動產生的描述

個人認為因為訓練資料太少不足以有效表現出所有母體特徵，以至於有點碰運氣切割的感覺在。

### perceptron4

第一次測試:訓練集中沒有選到class二的資料點以至於第三條線沒有正常訓練到，整體效過並不好雖訓練集準確度100%但測試集準確率只有33%。

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

第二次測試於epoch 3得到訓練集準確度100%測試集準確率66%效果還不錯。

一張含有 文字, 圖表, 行, 螢幕擷取畫面 的圖片

自動產生的描述

第三次測試: 於epoch 3得到訓練集準確度100%測試集準確率100%

一張含有 文字, 圖表, 螢幕擷取畫面, 行 的圖片

自動產生的描述

### wine

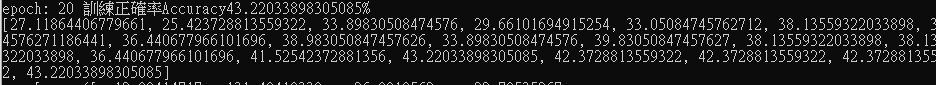
共13維3種class

第一次測試: 10 epoch訓練集準確度33.9%測試集準確度40%

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

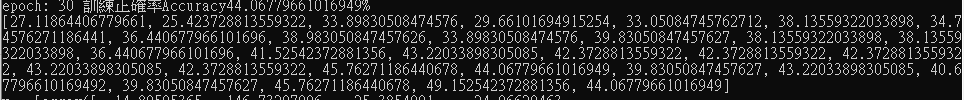
第二次測試: 20 epoch訓練集準確度43.2%測試集準確度38.33%



一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

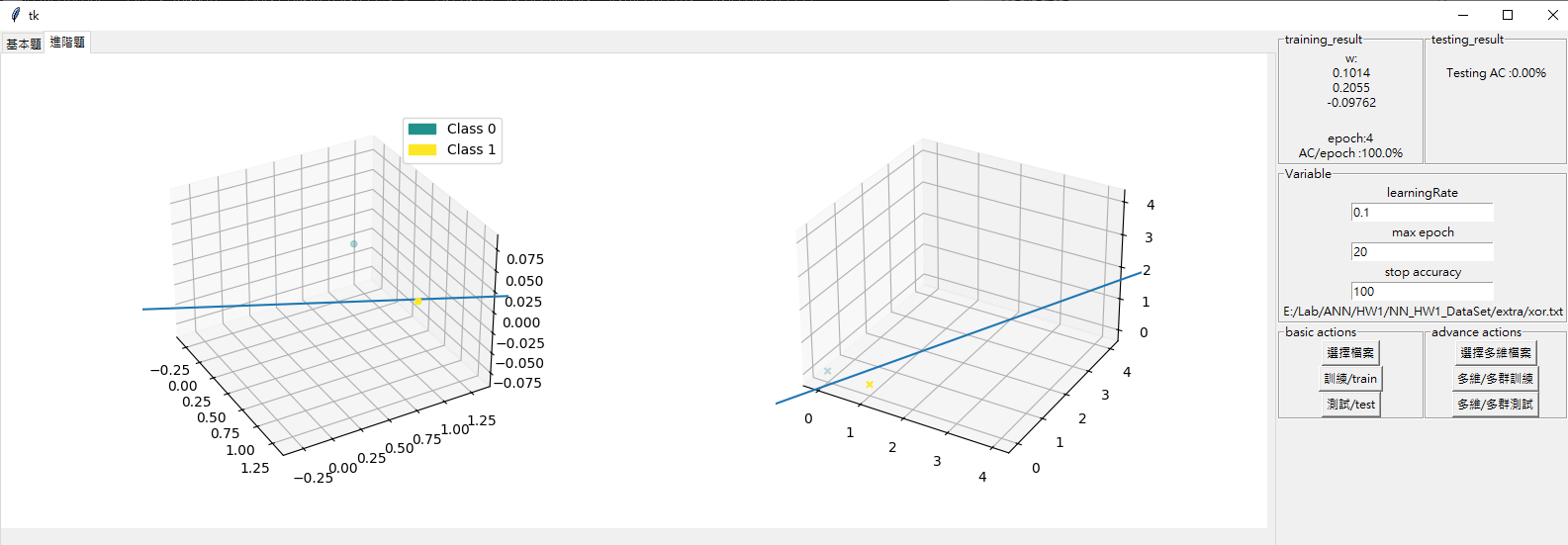
第三次測試:30 epoch訓練集準確度44.07%測試集準確度40%。沒有提升



一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

### xor

這題XOR需要多個感知機才能正確分類出所有點，程式使用1個感知機故無法正確分類，得到了訓練集100%測試集0%的結果。

# 實驗結果總結。

這次實驗中觀察到了以下現象

* 透過圖形化介面觀察了線性可分割與非線性可分割兩種類型對於感知機切割時的效果。
* 透過基本題中線性可分割的2CloseS2，觀察測試了過大learning rate可能造成每次修正過大而無法找到真正最優解的問題。
* 於基礎題perceptron1、加分題的perceptron3、perceptron4等題目觀察到，當資料太少或品質不佳時可能會由於訓練資料的代表性不足(訓練資料沒有很好的表現出母體分佈特徵)導致感知機無法找到真正的母體分布規律的問題ex:(在上述這幾題中多次碰到訓練集的所有點皆維同類，或抽到的兩點沒有表現出特殊排列規則)。
* 於進階題的8OX題目觀察了過擬合(提升epoch訓練集準確度提升但訓練集準確度顯著下降的狀況)。