## 類神經網路-Hw1-報告-108602532-王鼎元

A、GUI介紹:



4

3

2

1

5

8

7

6

[1] 選擇要用來訓練的資料集

[2] 輸入學習率以及收斂條件，學習率建議低一點，收斂條件是指在訓練組上測試後正確率達到多少%就停止訓練開始測試，輸入0-100之間的數字，一開始也是低一點必免一直無法收斂，可以慢慢增加來測試最好的準確率是多少

[3] 開始執行程式

[4] 會顯示訓練完的權重，以畫成線條的公式形式表示

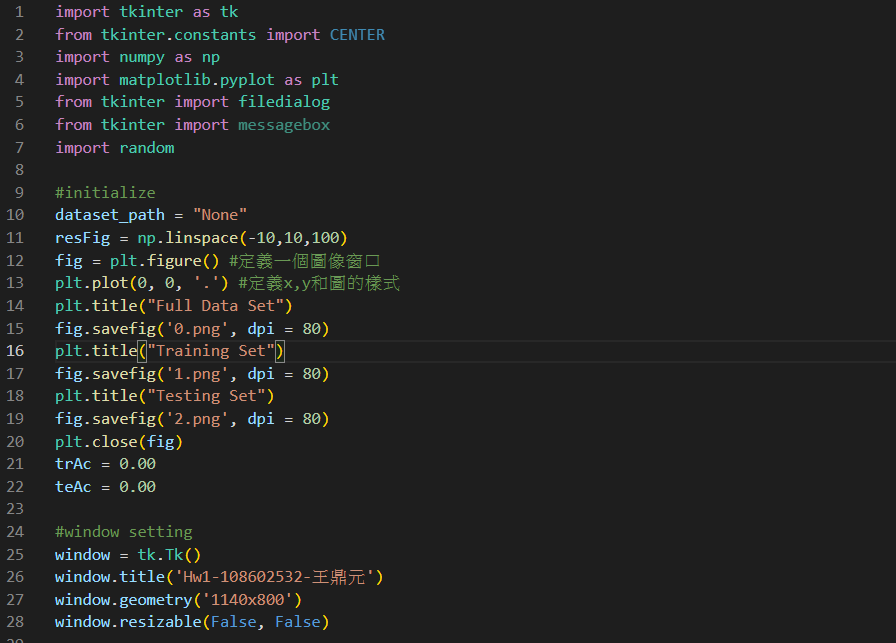
[5] 顯示訓練完的權重在訓練組以及測試組上的準確率有多少

[6] 視覺化的原始資料集，兩種顏色的點代表不同類別

[7] 視覺化的訓練資料集，兩種顏色的點代表不同類別，黑色的線是權重

[8] 視覺化的測試資料集，兩種顏色的點代表不同類別，黑色的線是權重

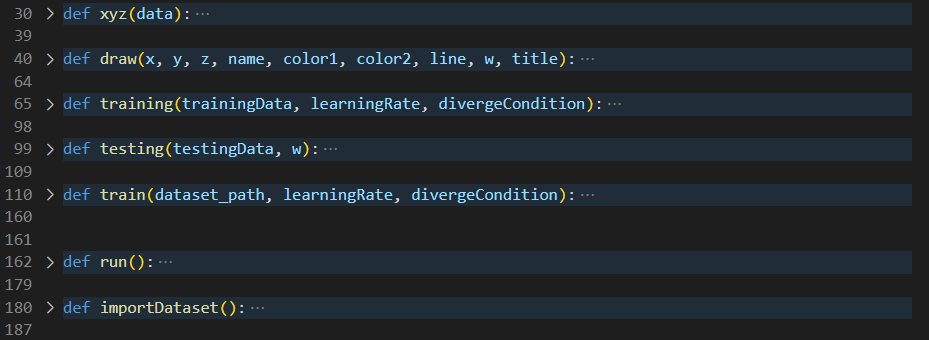
B、程式碼介紹:



設定GUI視窗

初始化資料路徑，準確率、視覺化圖表

Import 函式庫

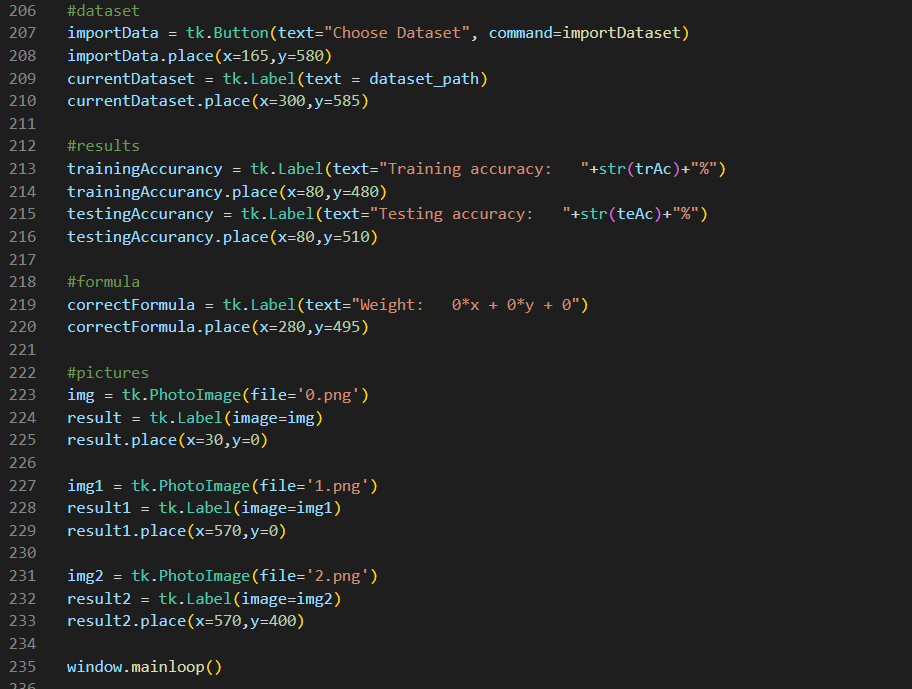


30~187為function，稍後介紹



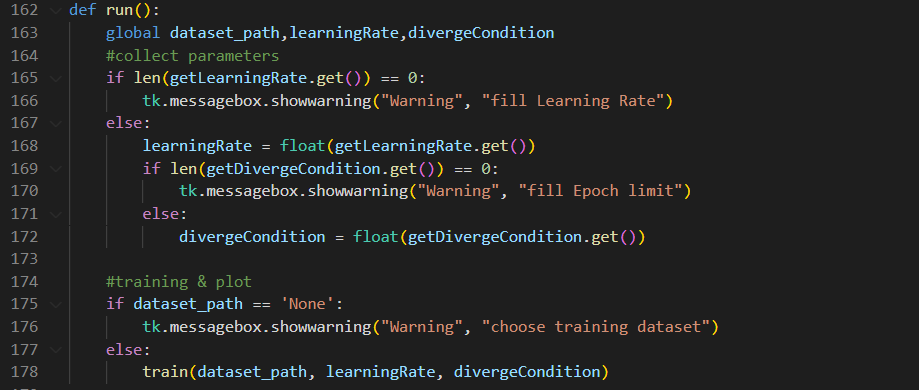
按下按鈕執行run function

設定各項Label、按鈕



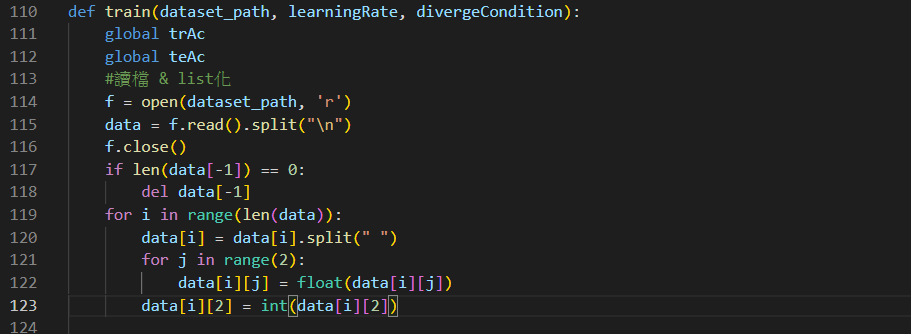
主程式結束

設定各項Label、按鈕

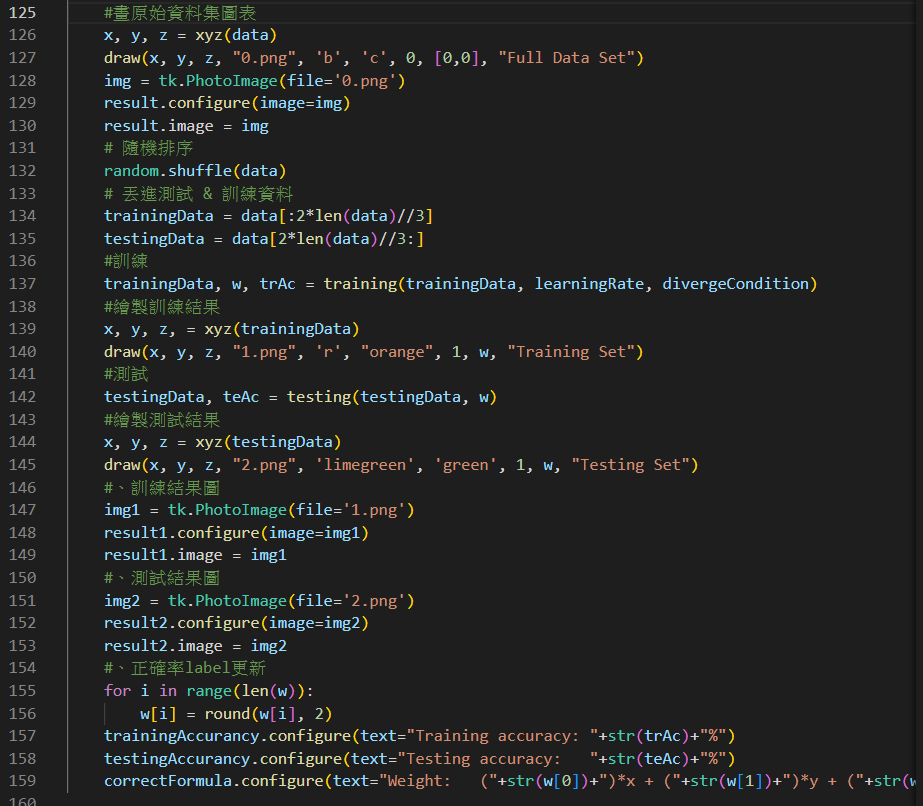


輸入完畢，執行train function

檢查各項參數是否輸入完畢



從選定的路徑中讀取資料並整理，有些資料集最後會有換行，將其移除



如同註解說明

當中xyz()是用來整理資料以便繪圖的function

draw()是用來繪圖的function

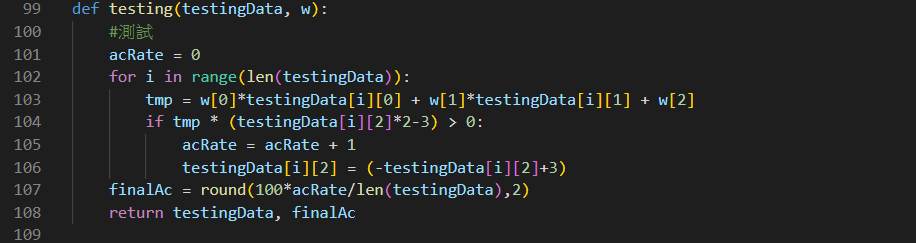


更新權重

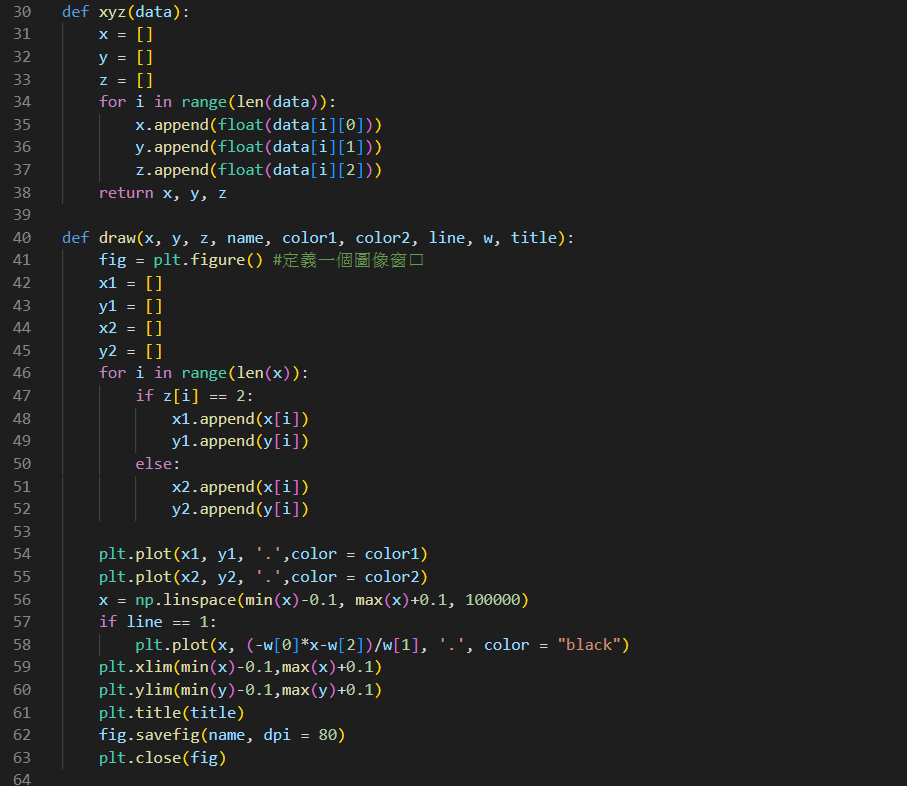
計算正確率

達到收斂條件後返回結果

判斷單筆資料是否正確

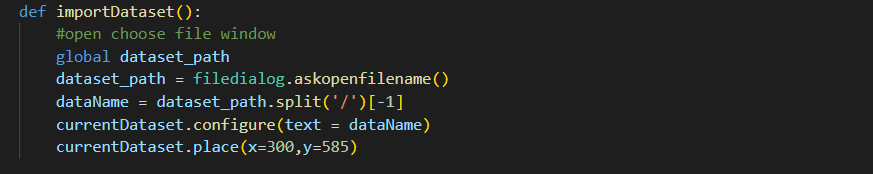


計算正確率



繪圖

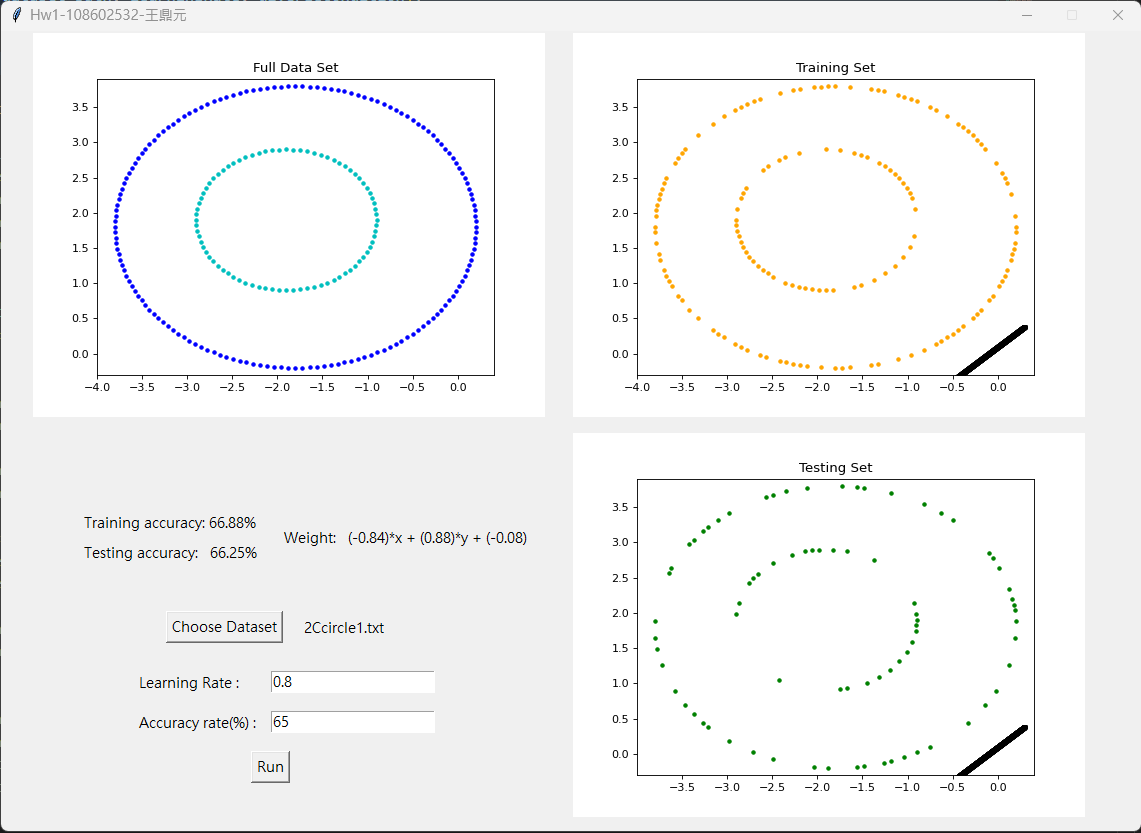
分類顏色



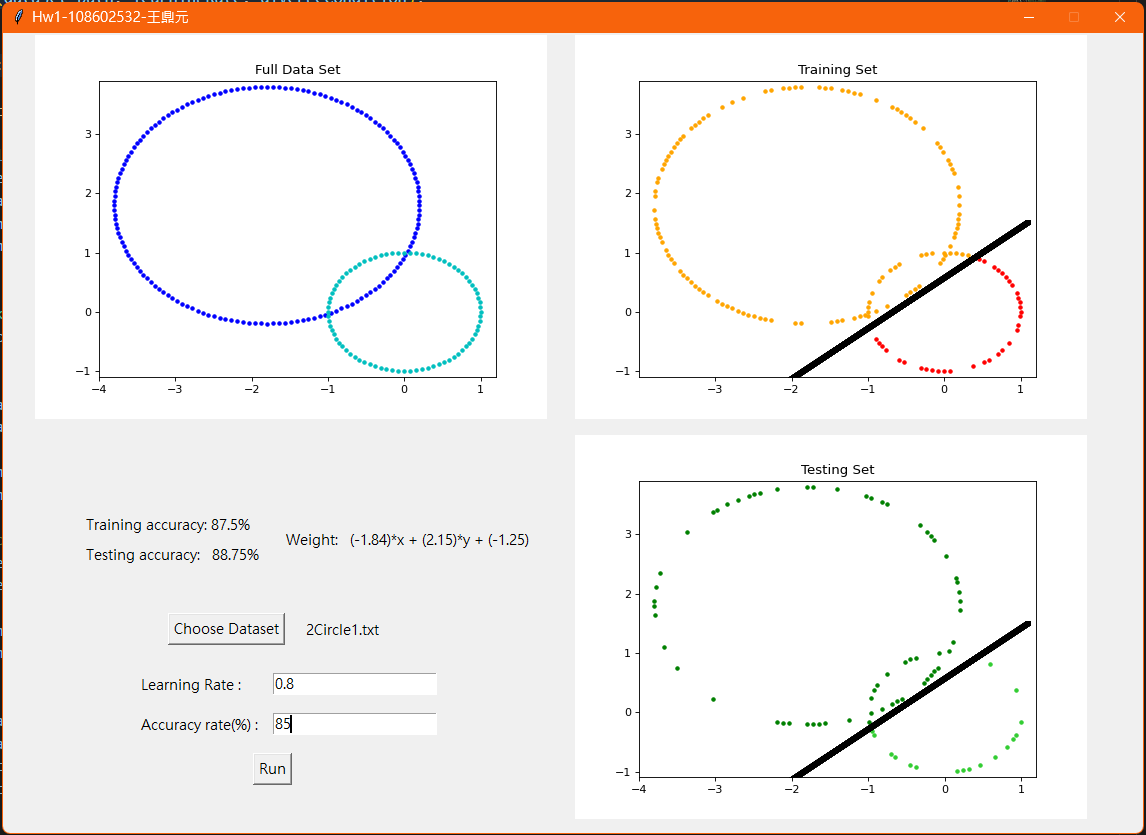
讀檔

C、實驗結果

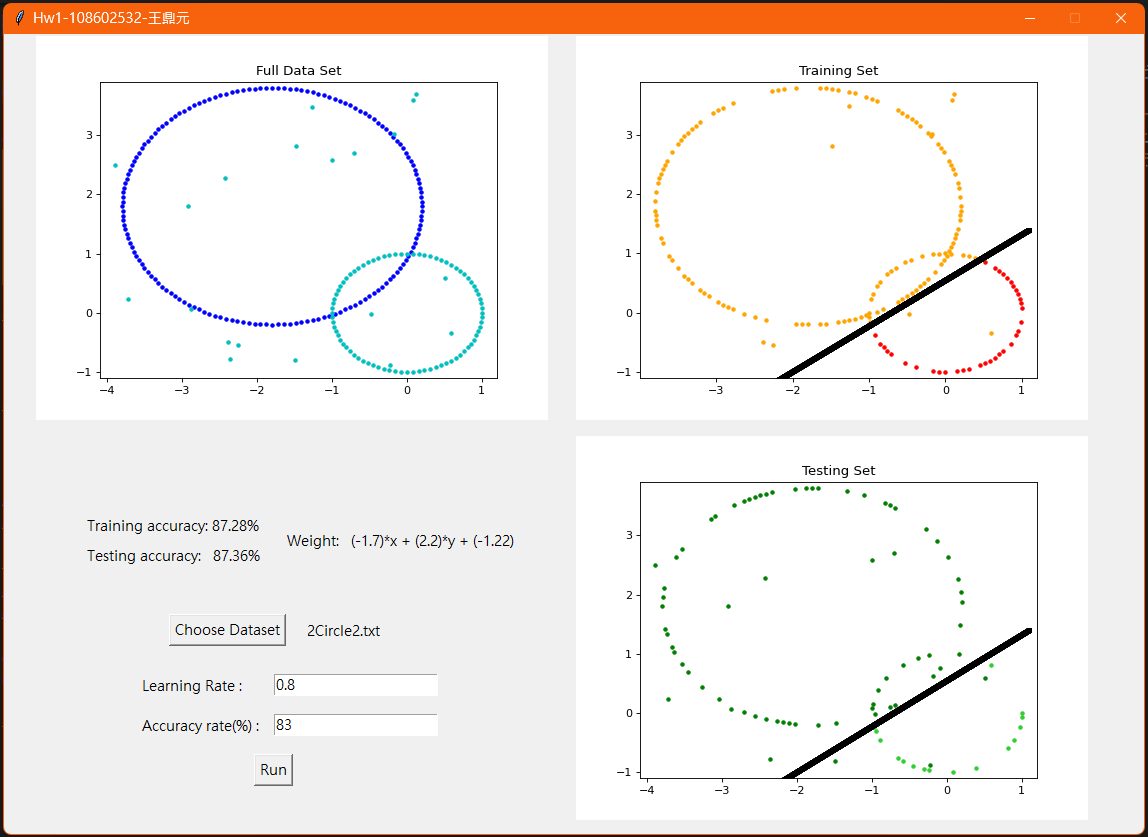
1. 2Ccircle1.txt



2. 2Circle1.txt



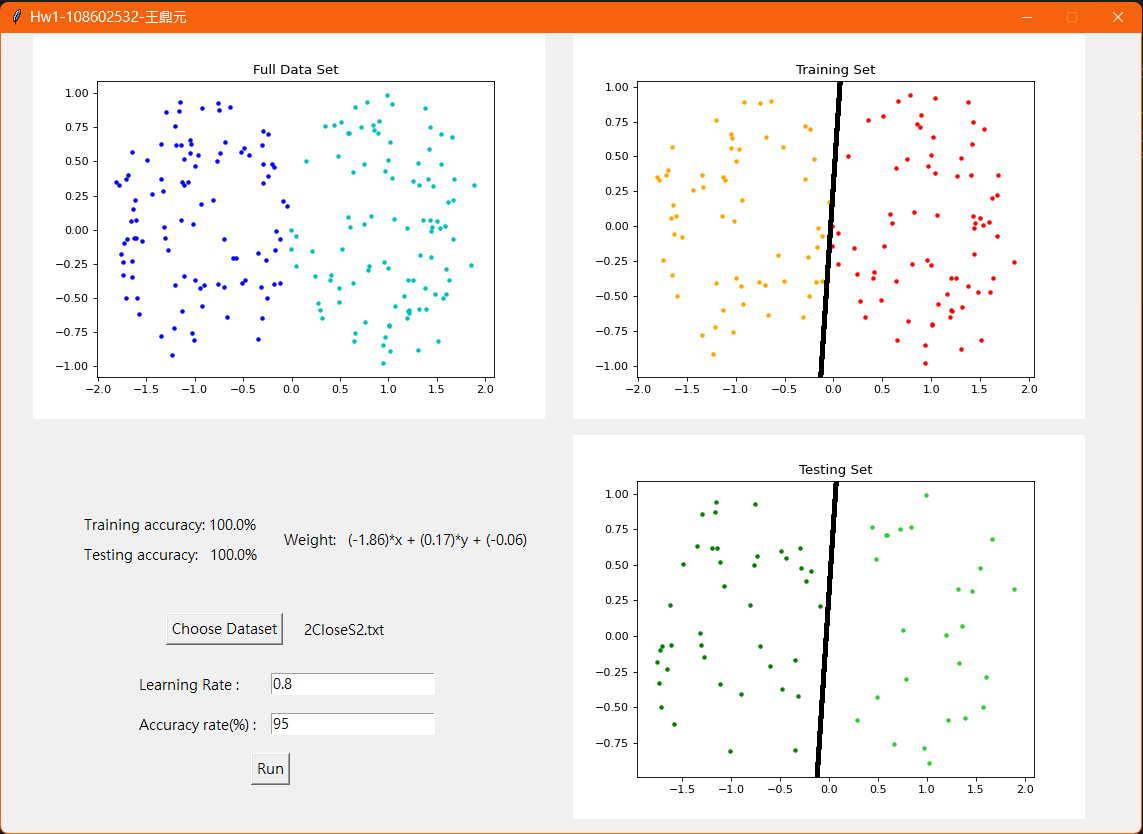
3. 2Circle2.txt



4. 2CloseS.txt



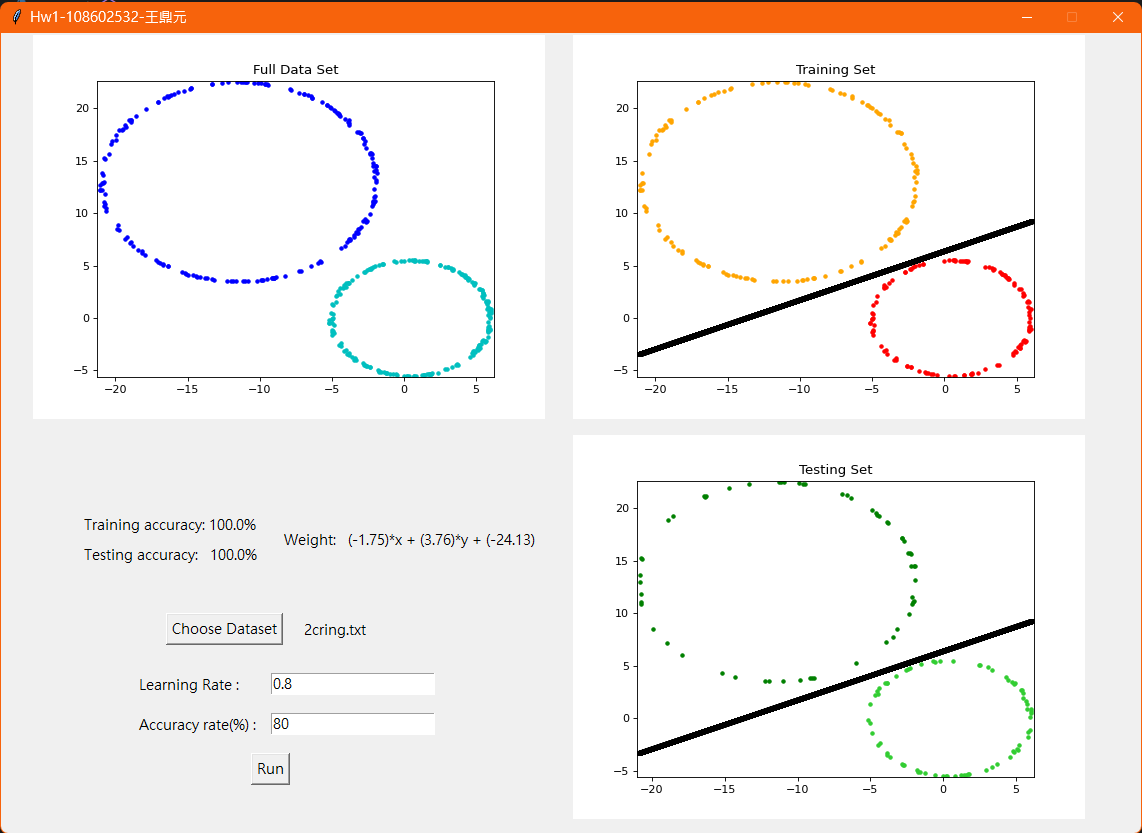
5. 2CloseS2.txt



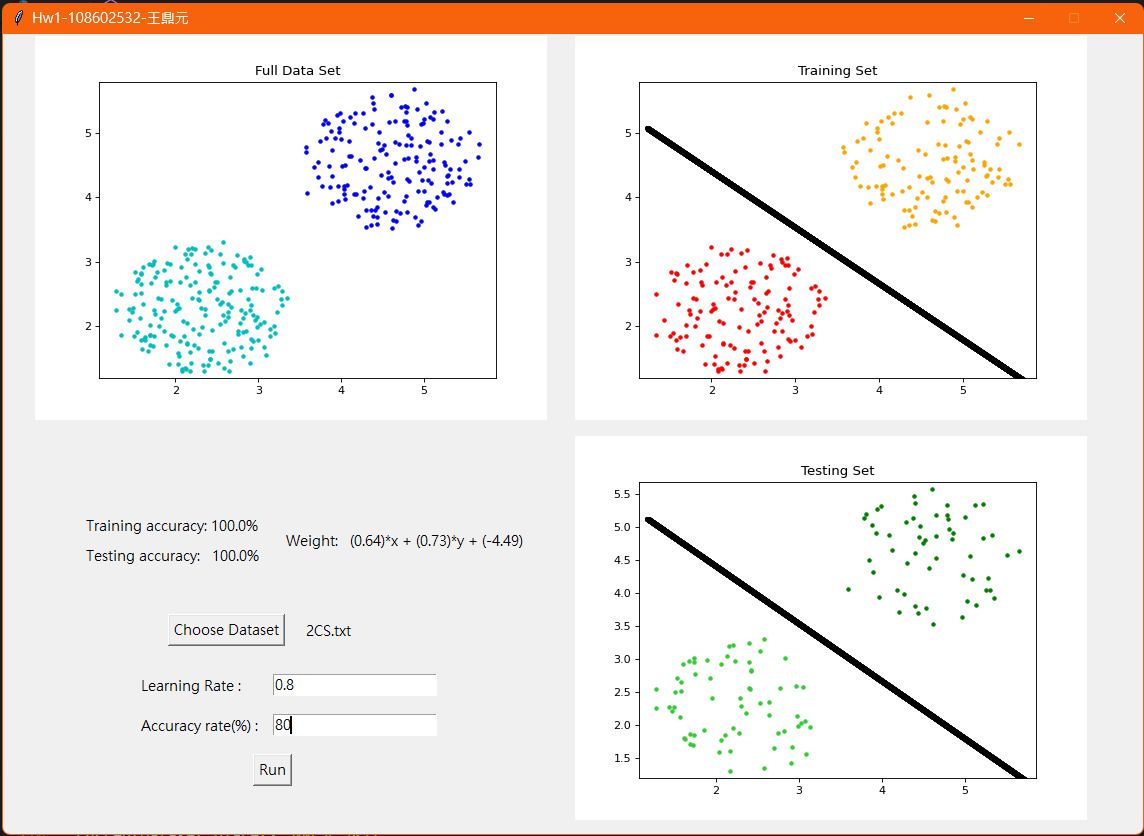
6. 2CloseS3.txt



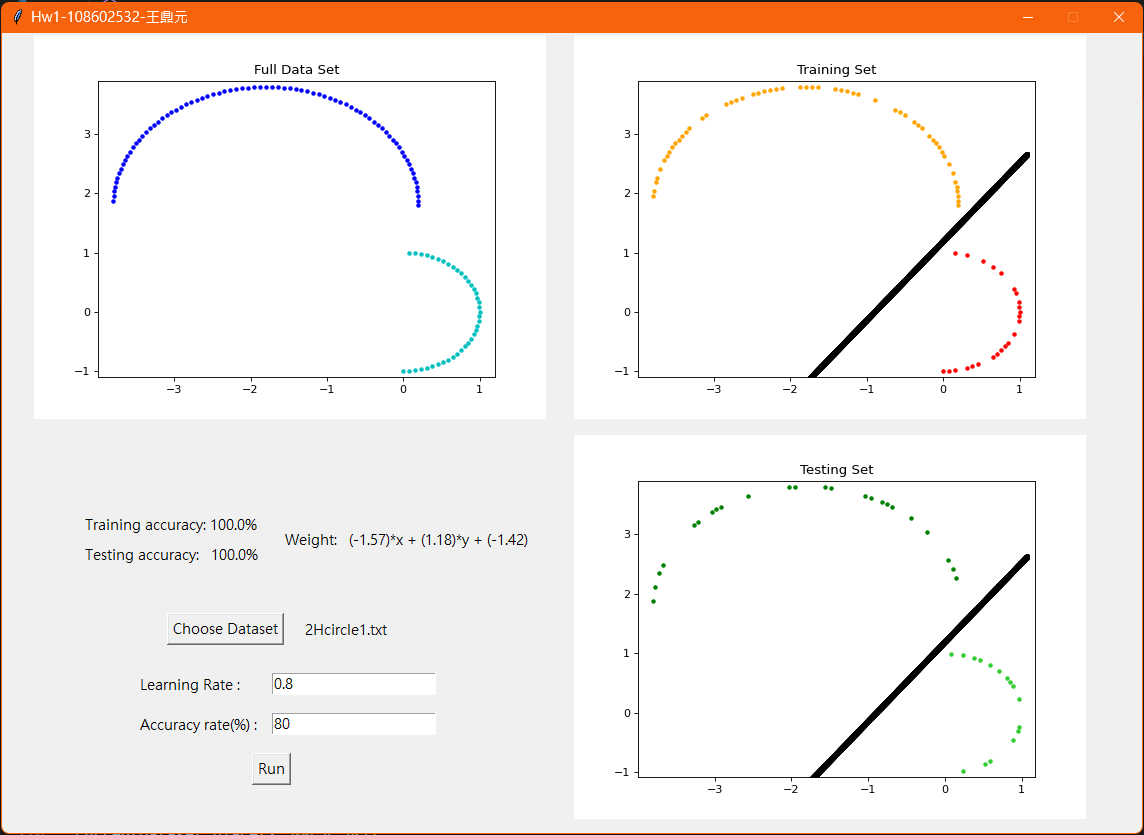
7. 2cring.txt



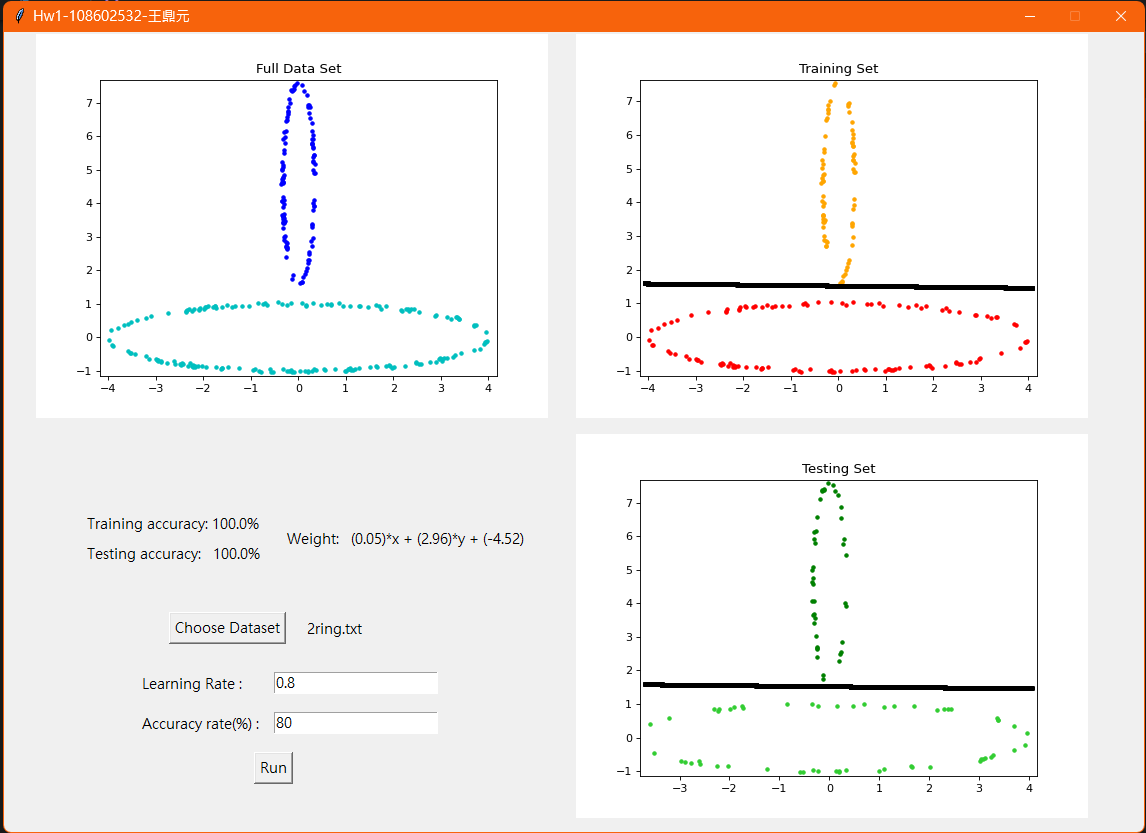
8. 2CS.txt



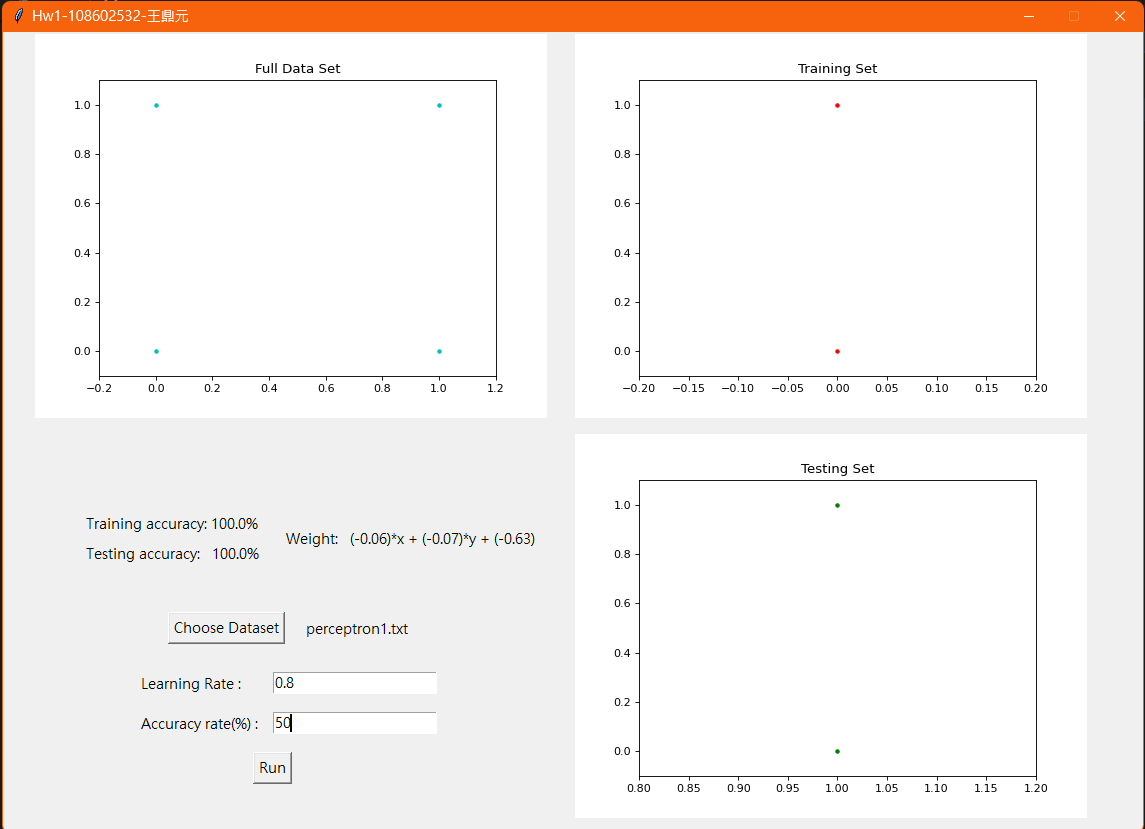
9. 2Hcircle1.txt



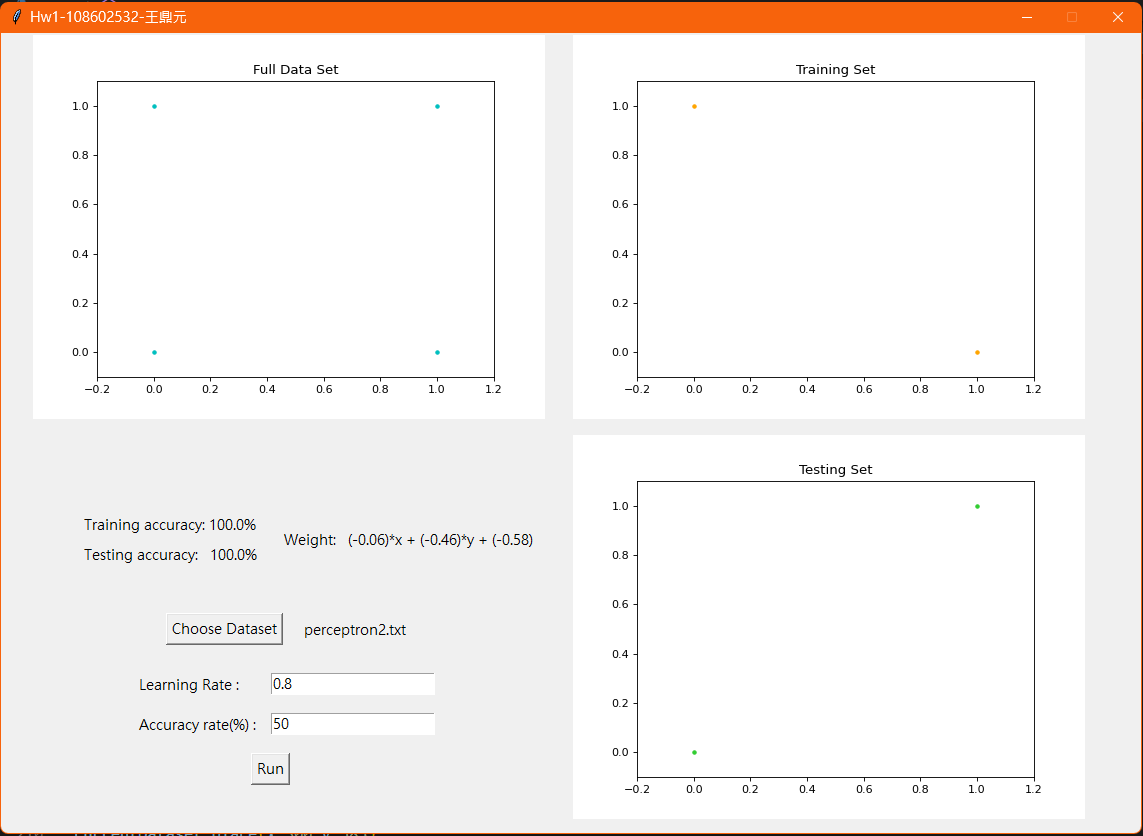
10. 2ring.txt



11. perceptron1.txt \*黑線在顯示範圍外，將所有的點分成一類



12. perceptron2.txt \*黑線在顯示範圍外，將所有的點分成一類



D、實驗結果分析

[1] 2Ccircle1.txt

此題為同心圓，正確的分割方法應該是要在大小兩個同心圓之間分割，但是我這次設定的神經元只有一次式，因此這題訓練不出來

**訓練一次準確率: 50% 訓練多次準確率: 65%**

[2] 2Circle1.txt、2Circle2.txt

兩題圖案長得很像，差別在於第二題的圖案多了一些雜訊點，一樣因為有交錯，因此一次方程式的直線無法切開兩者，但是比起第一題，兩個圓已經有很大一部分是分開的了，因此正確率方面有顯著進步

**訓練一次準確率: 80% 訓練多次準確率: 86%**

[3] 2CloseS.txt、2CloseS2、2CloseS3、2CS.txt

四題是同類型的題目，差別只在點的分布，數據量多寡以及稀疏程度而已，這三題兩個類別的點分布都是分散的，一次方程式幾乎可以完全切開，因此準確率極度接近100%

**訓練一次準確率: 99% 訓練多次準確率: 99%**

[4] 2cring.txt、2Hcircle1.txt、2ring.txt

三題也是同類型的題目，與上一個類別的差別是這幾題的分布不是均勻的，是有一個圖形在，但是由於兩個類別也可以完全被一直線分開，因此準確率野是極度接近100%，與上一個類別差別不大

**訓練一次準確率: 100% 訓練多次準確率: 100%**

[5] perceptron1.txt、perceptron2.txt

這兩題是同類型的題目，資料量極少，因此雖然我的截圖上準確率是100%，但是根據隨機拆分訓練組和測試組的情形，以及一開是設定初始權重的random number，結果也十分有機會是50%或是0%，以四筆資料來看，training set只有兩組資料，如果剛好得到同一類型的資料那就那就基本上不用訓練了，怎麼測都對，那就算是得到不同類型的也只會有一次的修改機會，因此我認為這兩題對感知機來說基本上等於用猜的

**訓練一次準確率: ?% 訓練多次準確率: ?%**