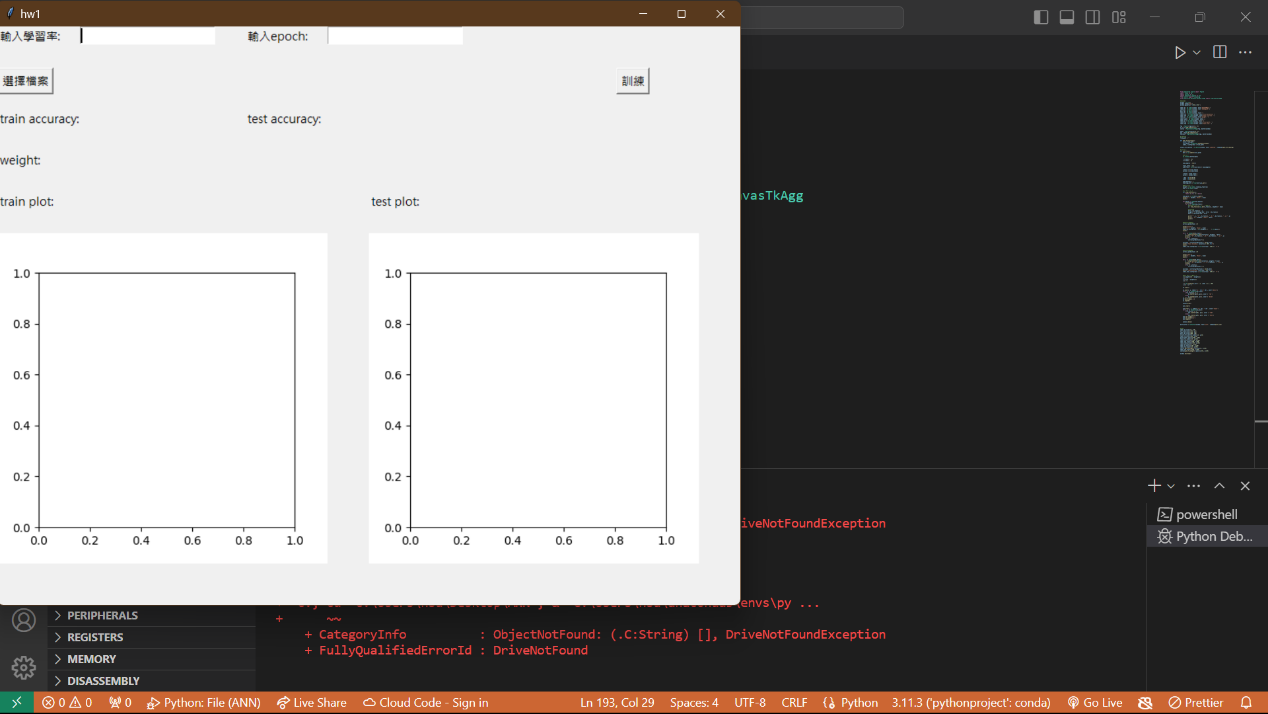
**類神經網路作業一**

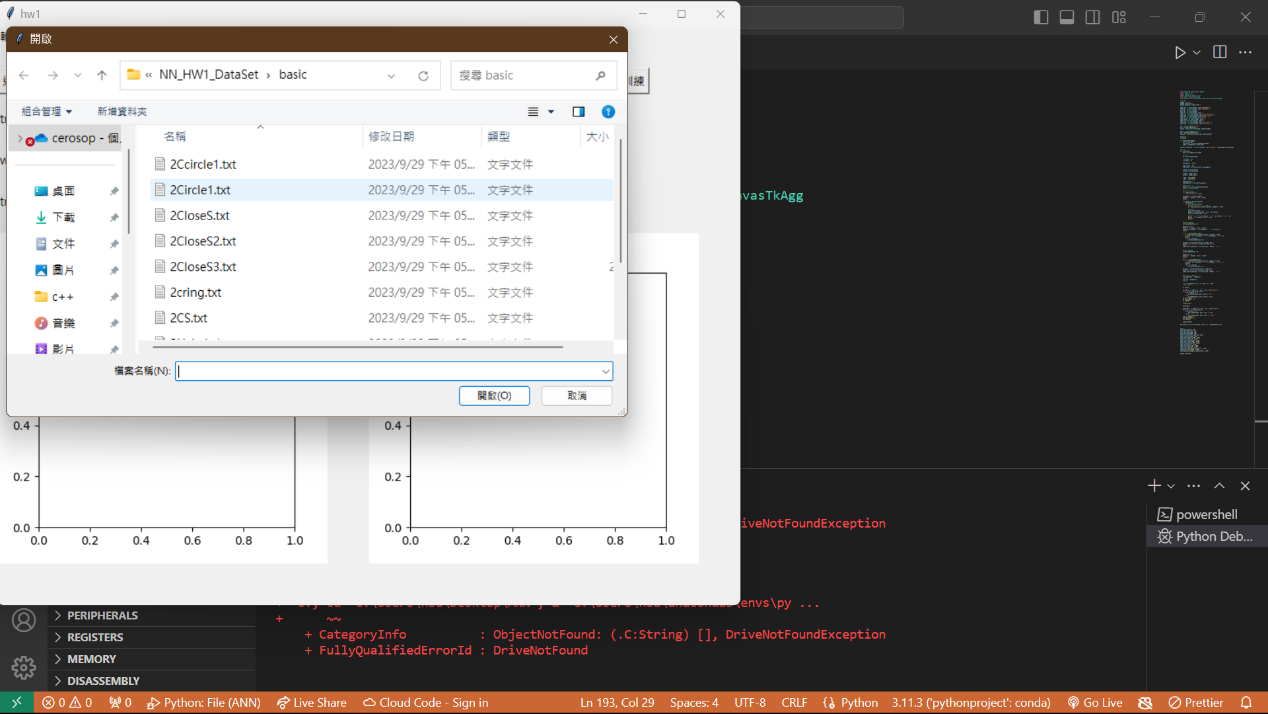
**110502516資工二A 許尚軒**

1. **GUI說明**

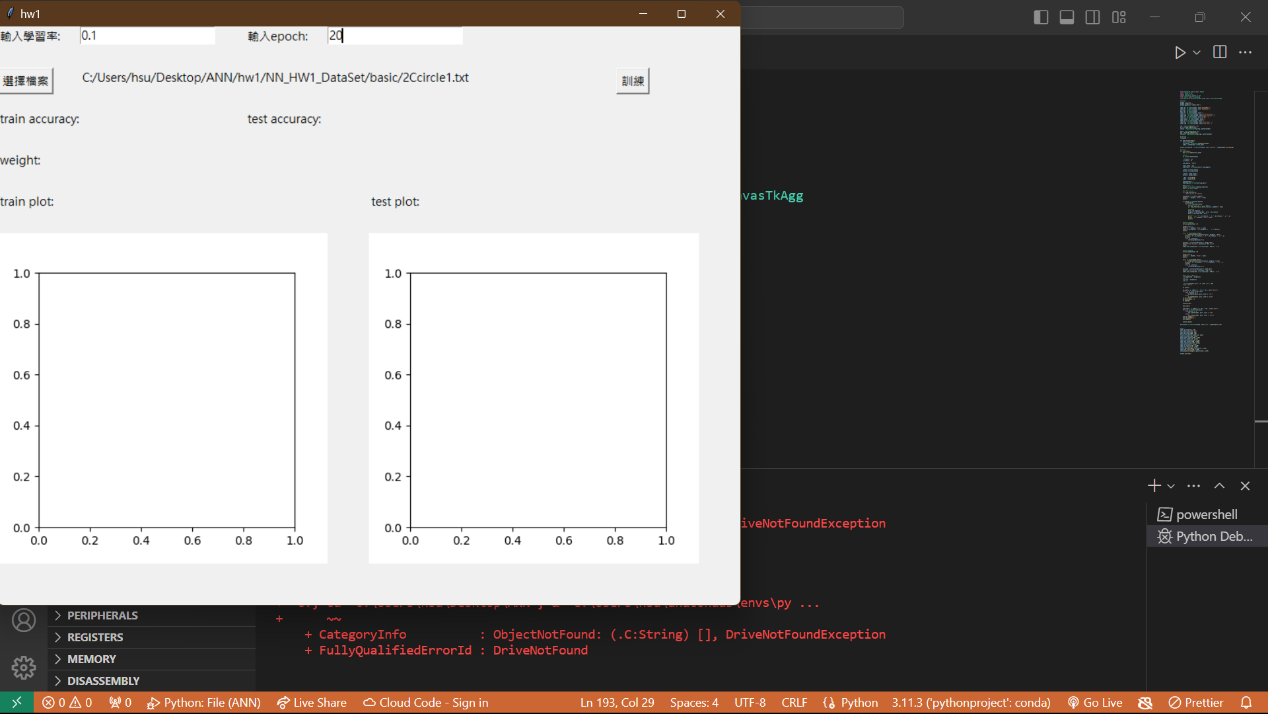
這是**起始畫面**，可以輸入學習率(浮點數)和epoch(正整數)，也可以按”選擇檔案”選擇要訓練的資料



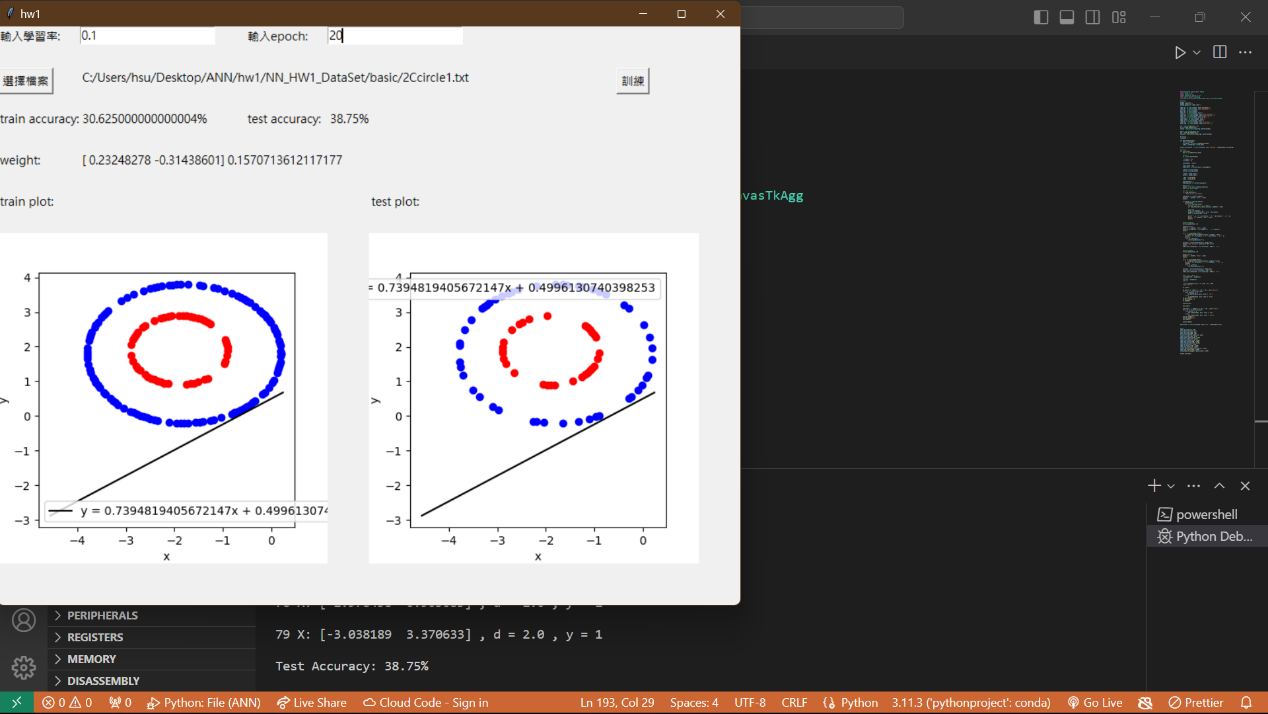
按下”選擇檔案”後會跳出此視窗選擇要訓練的資料



選擇好後會將檔案顯示在畫面上

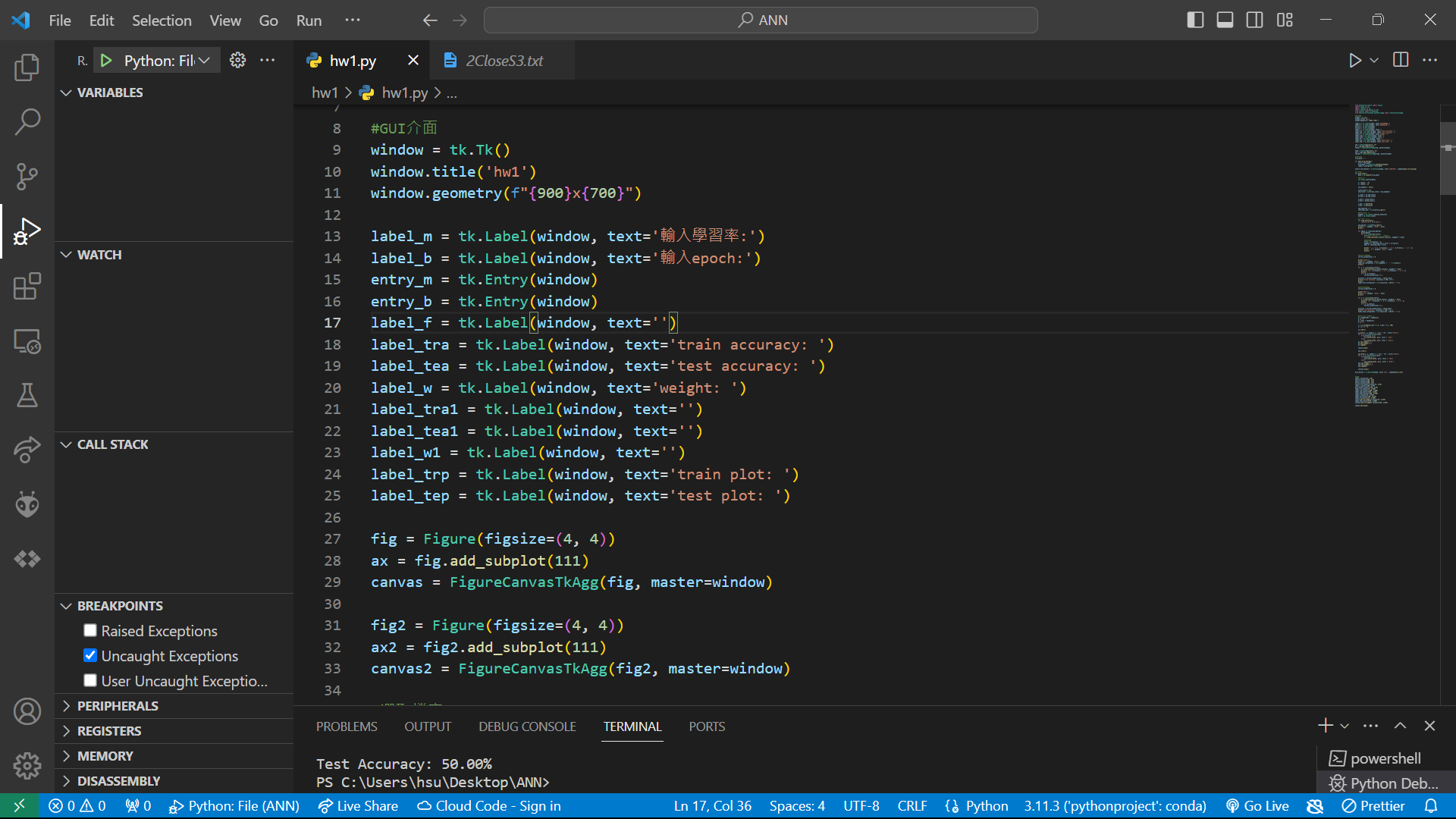


按下訓練後就會開始訓練，訓練過程中會在，訓練完成後會將結果畫在下圖，**左邊是train data，右邊是test data，紅點代表期望輸出為1的點，藍點為另一個，黑線為鍵節值和閥值在二維圖上代表的直線**

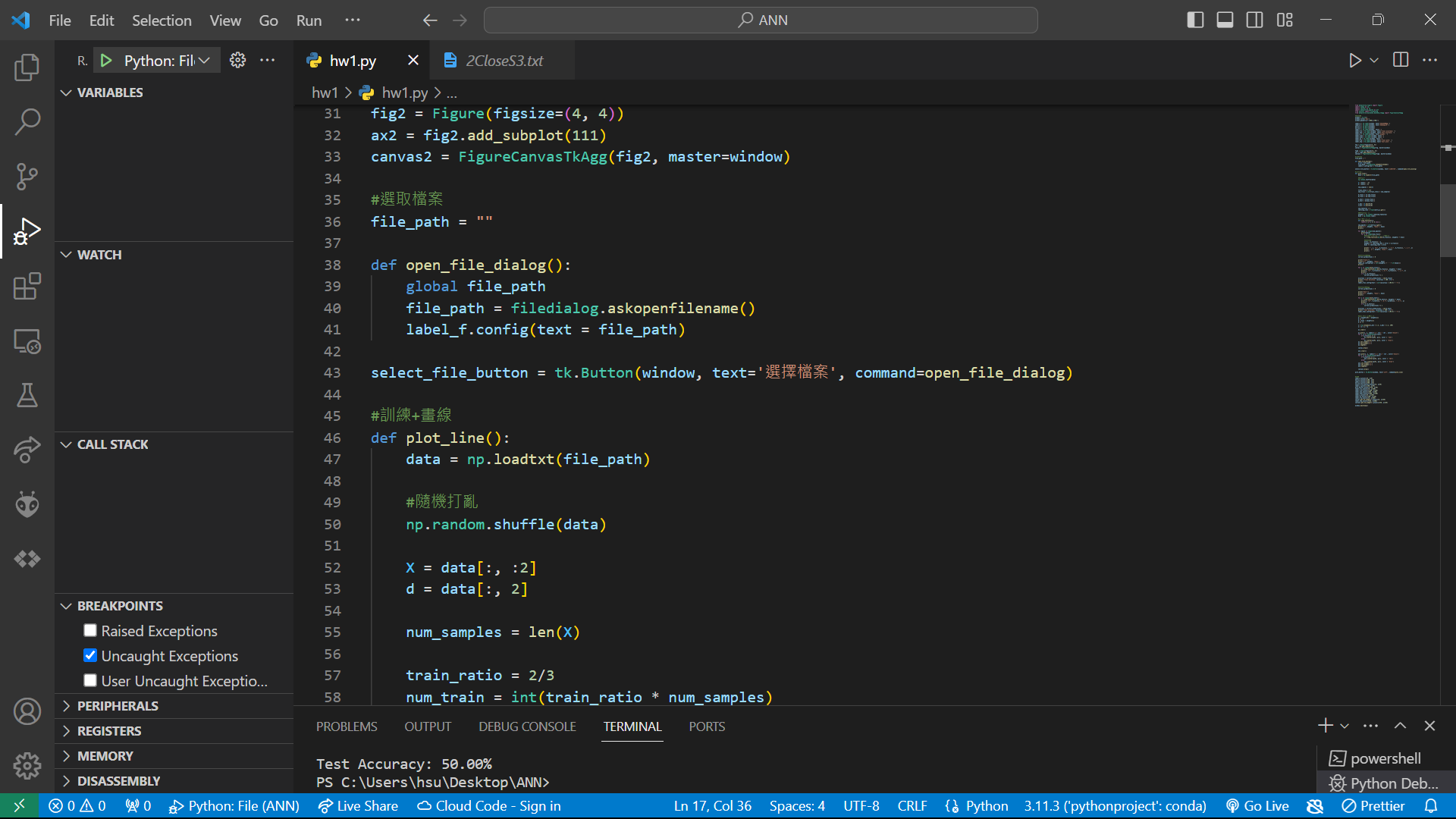


1. **程式碼說明**

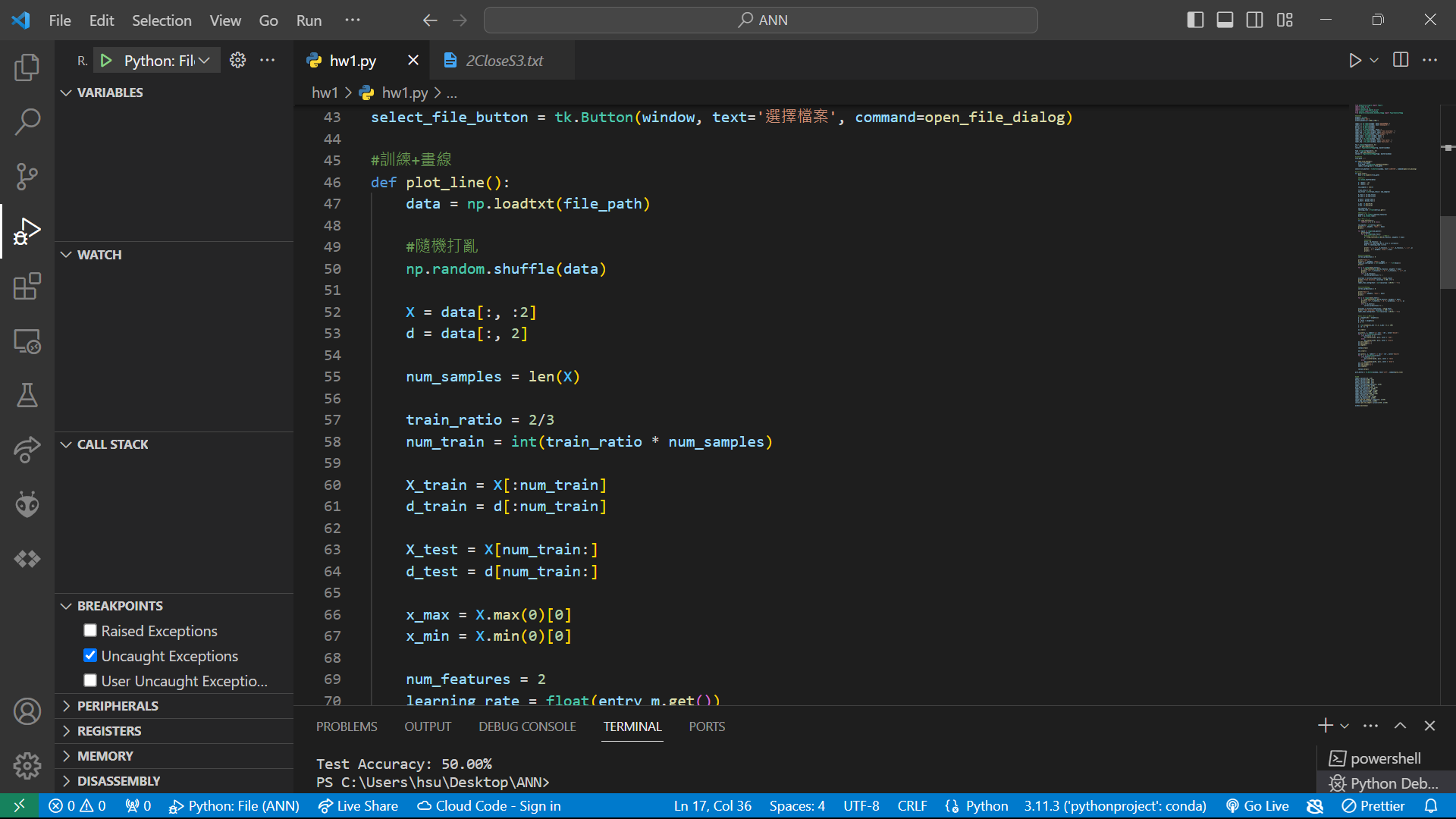
建立視窗和要顯示在視窗上的物件



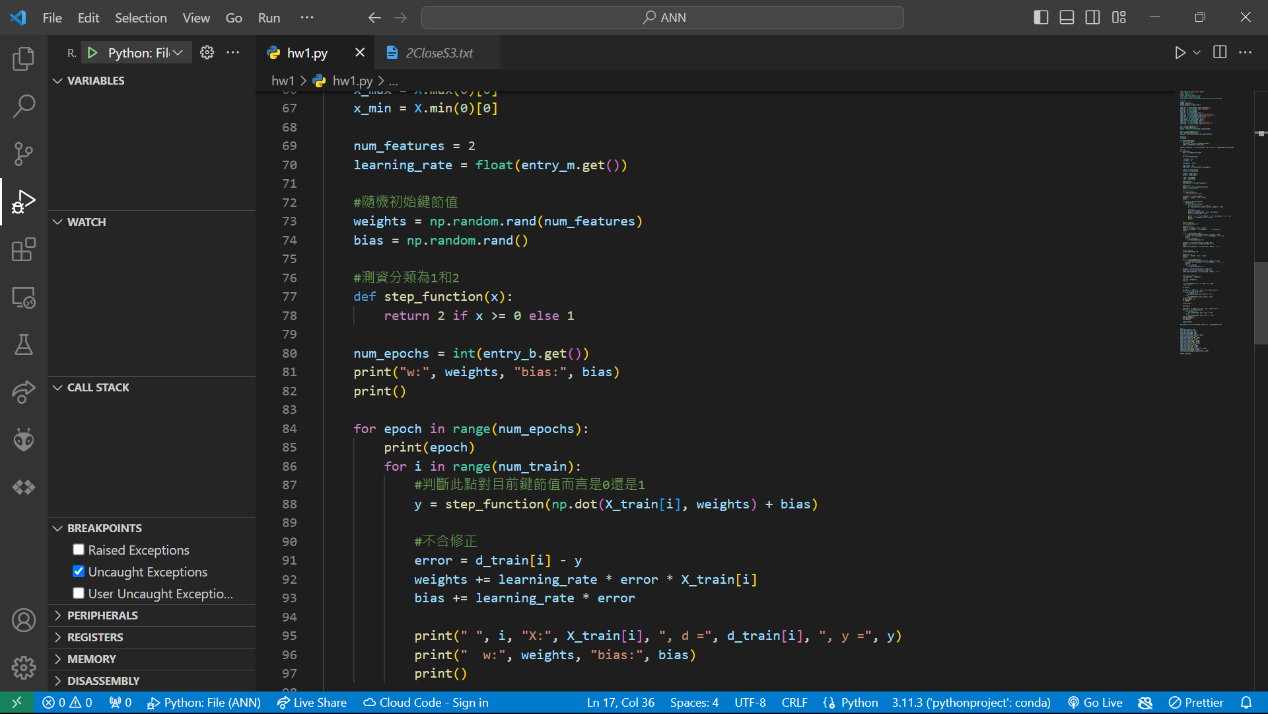
建立選取檔案按鈕並存取選取檔名



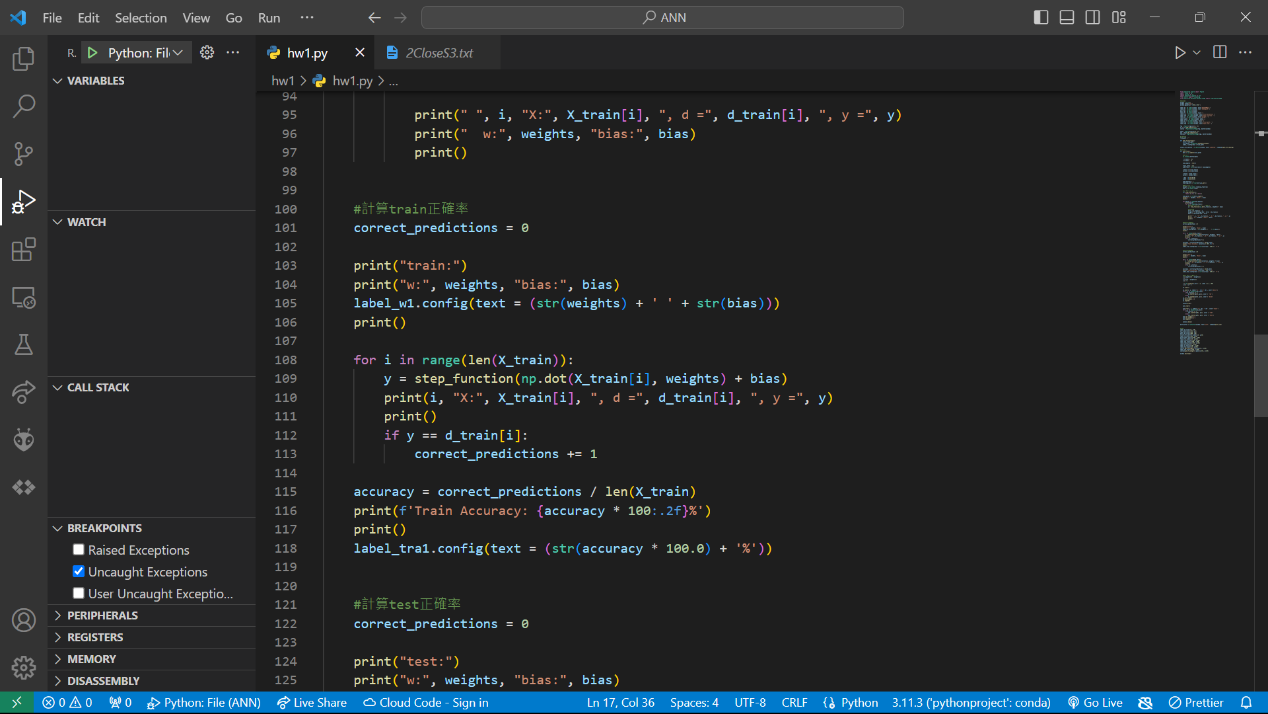
建立訓練按鈕，先獲取資料並隨機分train data和test data



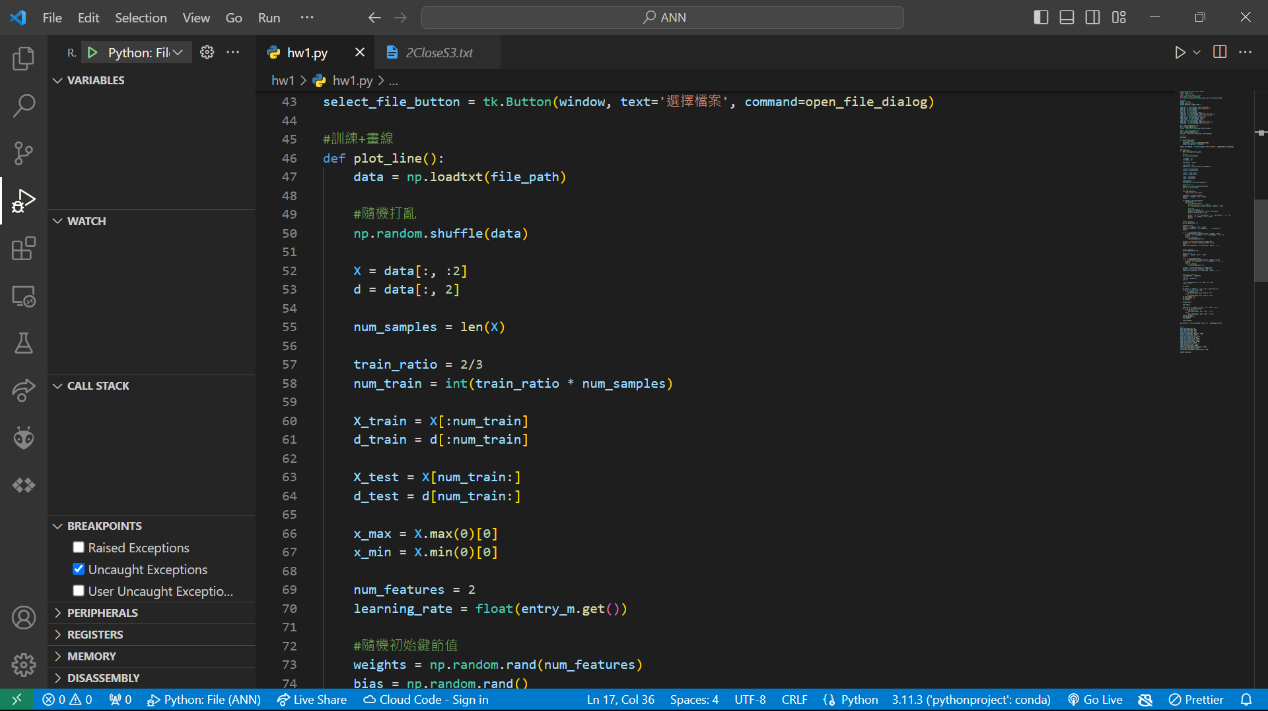
訓練過程，先隨機設初始鍵節值和閥值，**step\_function即為活化函數**，因為除了perceptron外的資料都分類為1和2，因此寫當>=0時return 2，其他return 1，之後跑epoch次每筆資料的訓練並以error修改鍵節值和閥值，**error為期望輸出和實際輸出的差**，每筆資料訓練後會輸出該筆資料和改變後的鍵節值和閥值



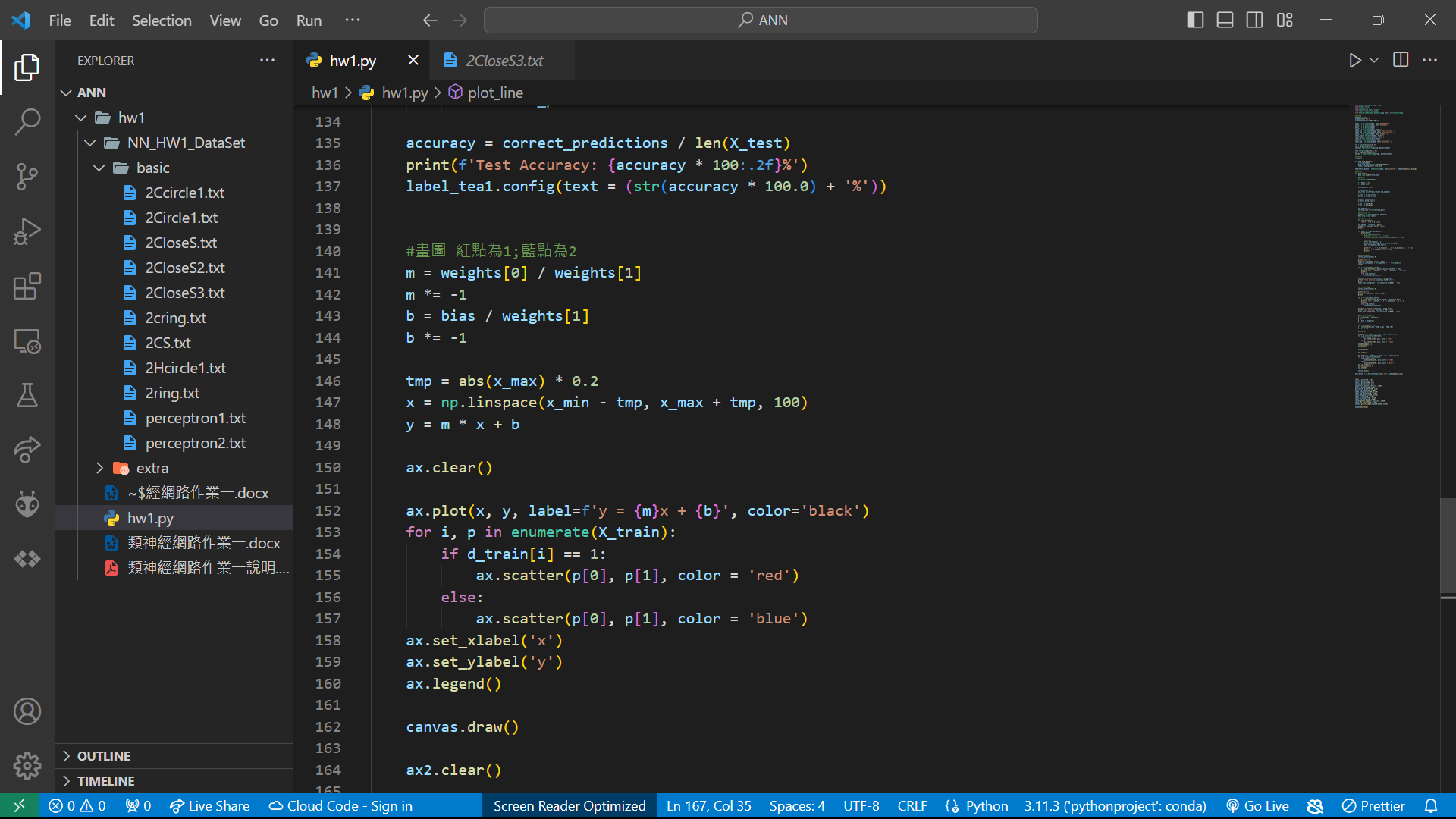
訓練完畢後，會計算正確率，下圖為計算train的正確率的程式，計算test的程式與之相似



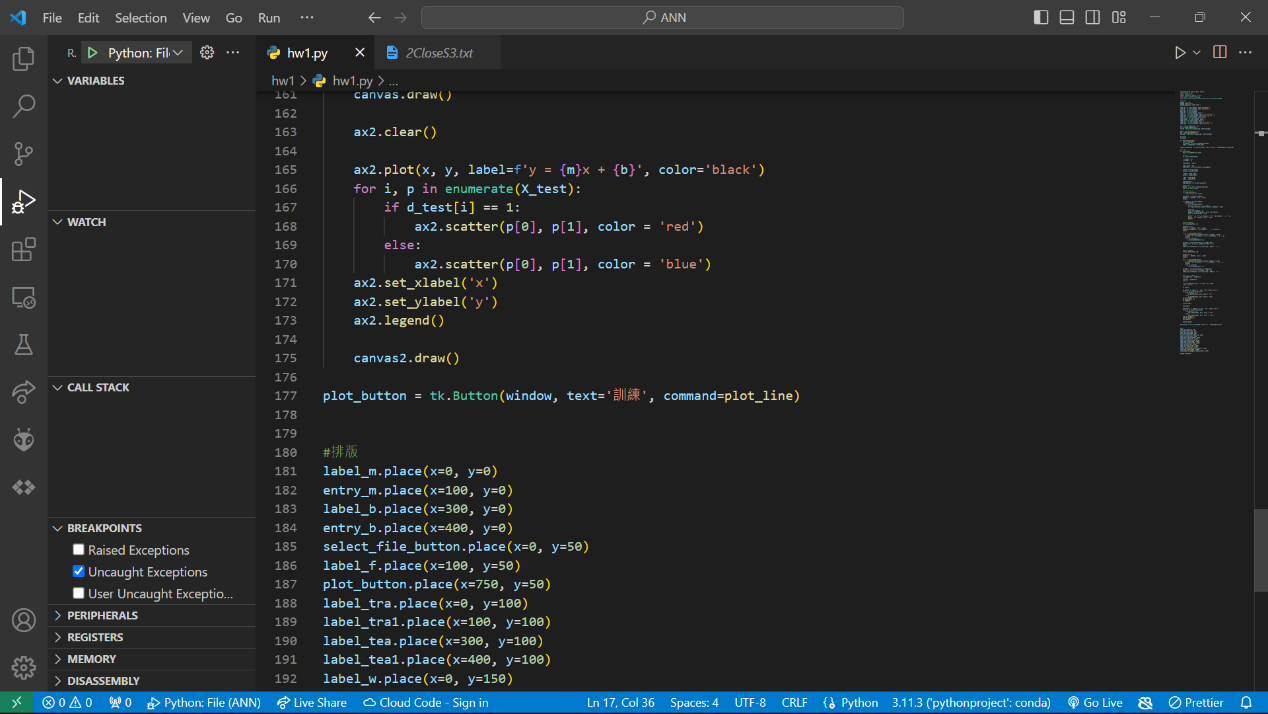
畫圖前先去找到該資料中x1的最大最小值



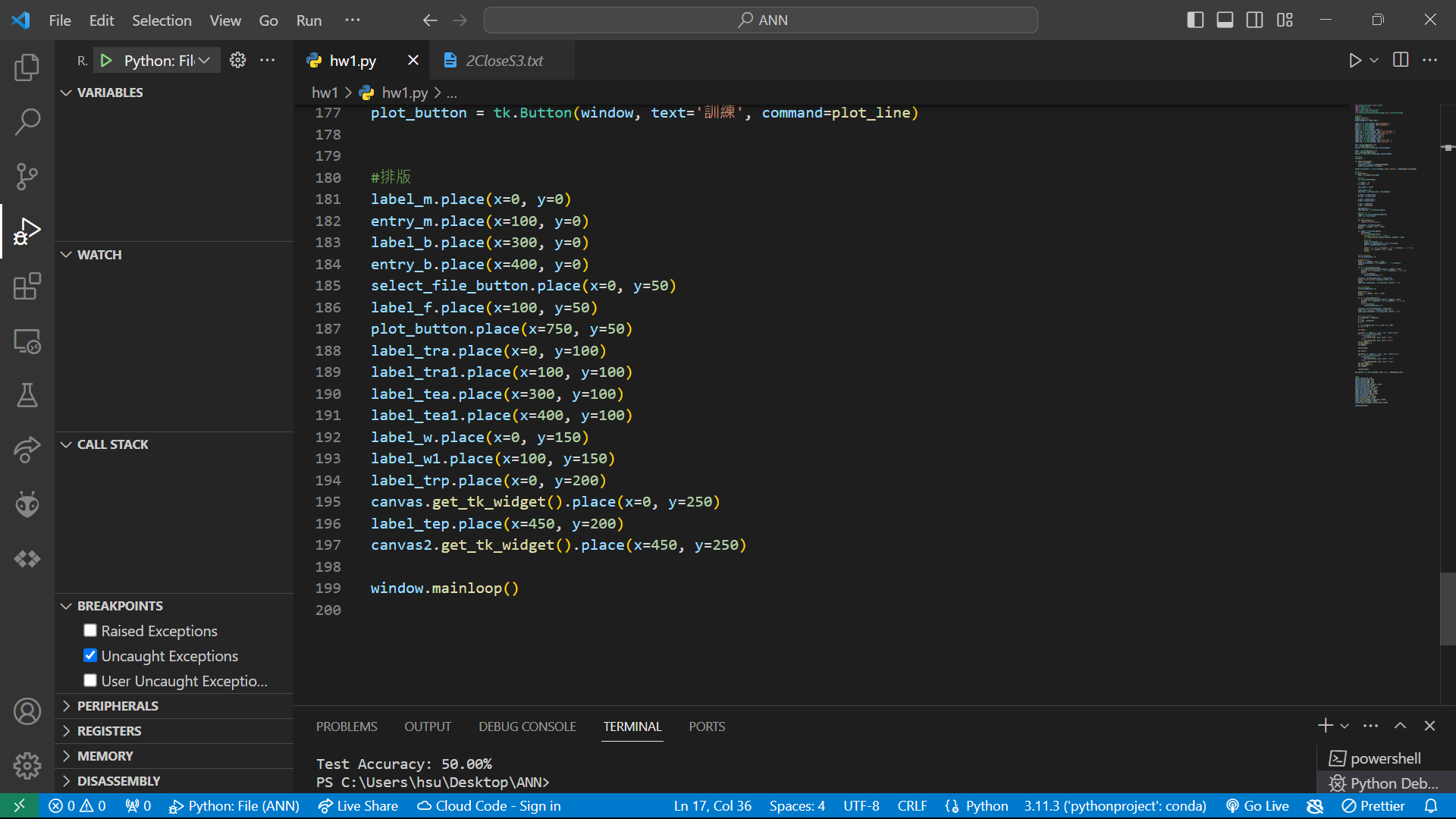
並以該最大最小值再往外突出0.2倍來畫圖，線用黑色畫，點則為紅色和藍色



再將上述訓練按鈕的function放入訓練按鈕的指令中



最後就是排版

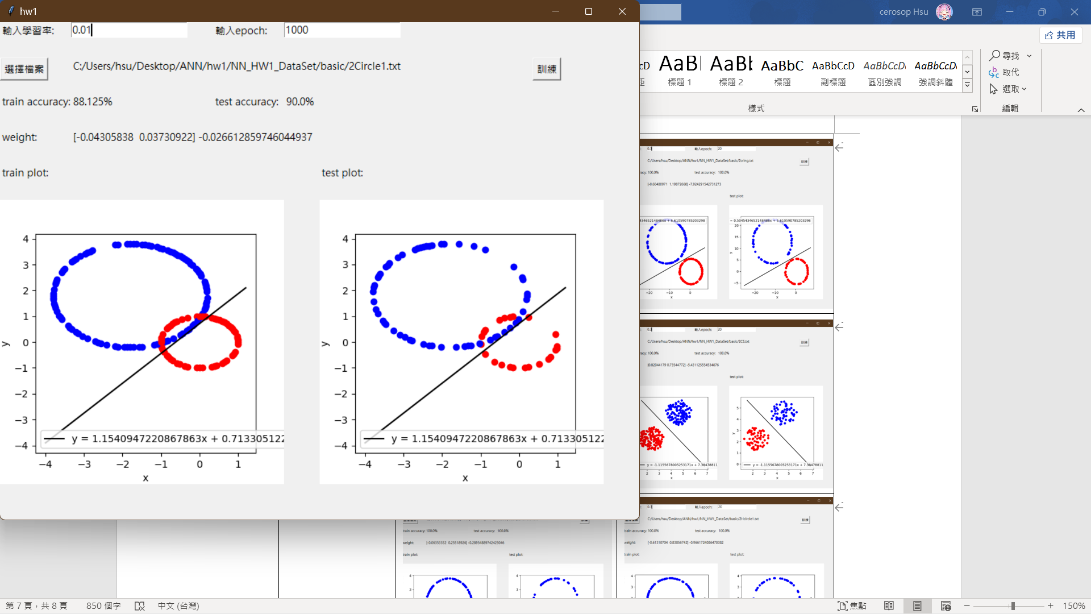


1. **實驗結果**

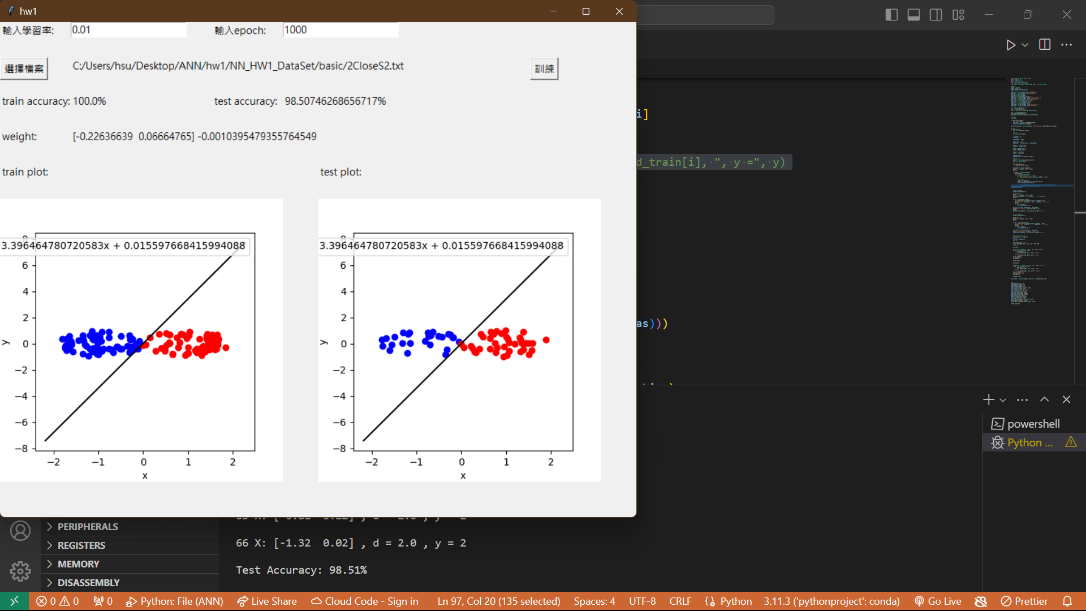
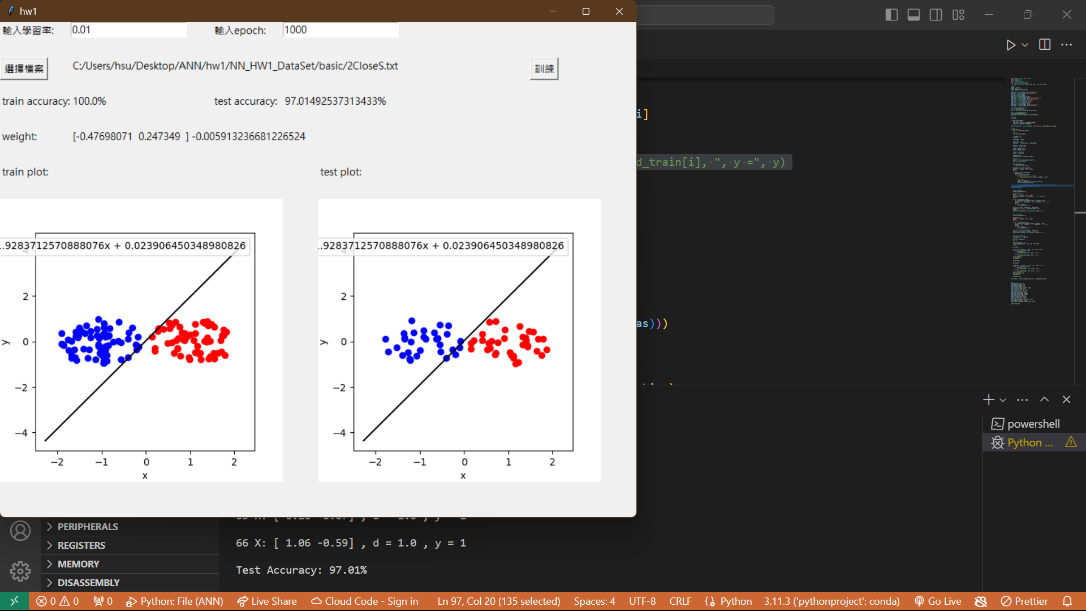
|  |  |  |
| --- | --- | --- |
| **檔案\學習率** | 0.1 | 0.3 |
| 2Ccircle1 |  |  |
| 2Circle1 |  |  |
| 2CloseS |  |  |
| 2CloseS2 |  |  |
| 2CloseS3 |  |  |
| 2cring |  |  |
| 2CS |  |  |
| 2Hcircle1 |  |  |
| 2ring |  |  |
| perceptron1 |  |  |
| perceptron2 |  |  |

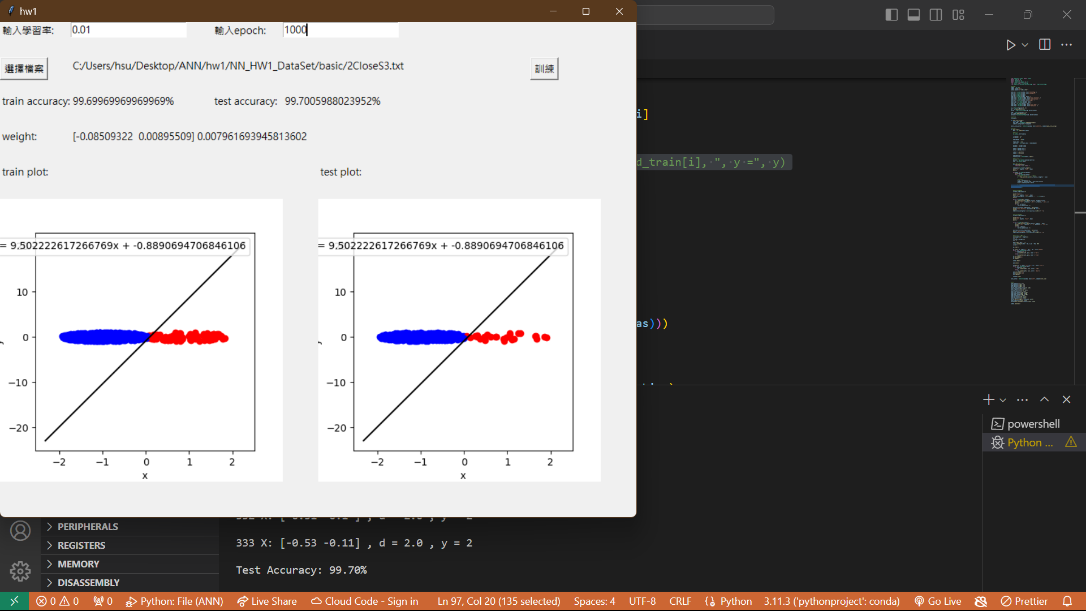
1. **實驗結果分析和討論**

2Ccircle1、2Circle1都是無法被一刀正確分兩區的，因此可以看到正確率皆無法達到100%，而2Circle1點分布長得像兩圓相交，但為何訓練產生的鍵節值沒準確切在相交點上我認為是因為(1)**epoch不夠**，為了驗證，我調整epoch到1000並下調學習率至0.01，如下圖，可以發現即使訓練這麼多，還是無法準確地剛好切在交點上，我認為是因為(2)**兩圓相交區域與原點的距離不同**的關係導致兩圓錯誤區域對鍵節值的影響不同。



2CloseS、2CloseS2、2CloseS3皆為兩區分布相近，但是可以被一刀正確區分的圖，所以可以看到訓練後的鍵節值可以達到幾乎100%的正確率，但為何train accuracy沒達到100%我認為是因為**epoch不夠**，因此我將epoch調至1000且學習率調至0.01後，的確正確率都十分接近100%了，如下圖。





2cring、2CS、2Hcircle1、2ring皆為兩區分比較開，且可以被一刀正確區分的圖，所以可以看到訓練後的結果可以正確將資料**成功分為兩區**。

Perceptron1、perceptron2皆為只有4個點的圖，因此無論如何調整epoch和學習率都**無法精確**地在每次訓練都由train data去訓練出正確的感知機來區分test data。