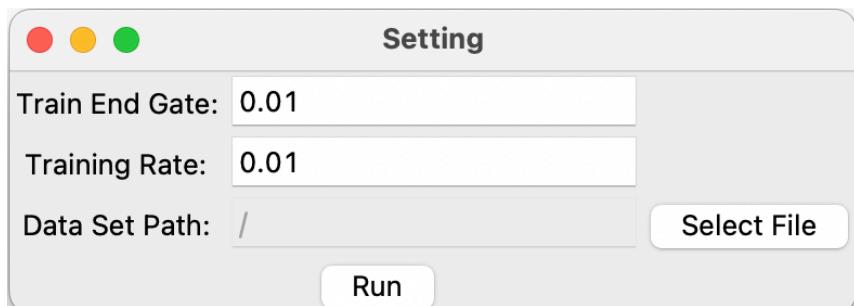


類神經網路作業一

學號：111526009 姓名：薛竣祐

一. 程式執行說明

程式提供兩執行檔，一為exe執行檔，一為Unix執行檔，兩執行檔結果相同，擇一執行。
執行後會呈現此畫面，可為Model設定兩初始值及選擇Data Set檔案。



Train End Gate : Loss達到此門檻後會停止訓練。

Training Rate : Model的學習率，太大或太小都會無法收斂。

Data Set Path : 使用「Select File」按鈕選擇Data Set檔案。* Note : 僅限選擇3*N的資料集。

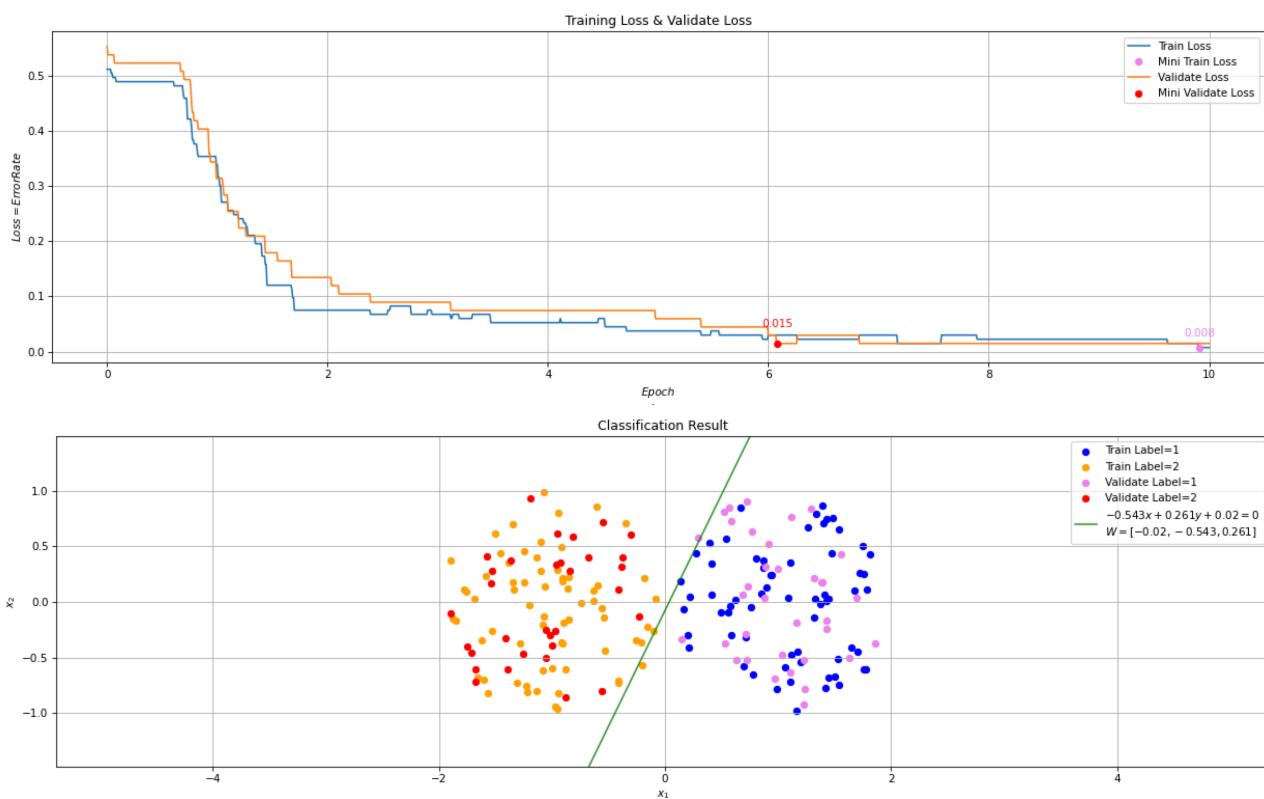
完成設定後，請點擊Run按鈕，此設定視窗會關閉，並在Terminal上顯示訓練過程，呈現Epoch、鏈結值、Loss，請靜待訓練結束。

* Note : 最大Epoch為512。

```
> /Users/alvin/Projects/Courses/NN/HW1/dist/LMS_merge_loss; exit
epoch: 01      w: [-0.61, 0.243, 0.527] 0.7486875
epoch: 02      w: [-0.2, 0.08, 0.392] 0.426625
epoch: 03      w: [0.05, -0.061, 0.198] 0.15531250000000002
epoch: 04      w: [0.1, -0.098, 0.09] 0.11854166666666667
epoch: 05      w: [0.08, -0.082, 0.072] 0.10854166666666668
epoch: 06      w: [0.05, -0.077, 0.069] 0.07408333333333333
epoch: 07      w: [0.05, -0.064, 0.06] 0.07229166666666667
epoch: 08      w: [0.04, -0.039, 0.044] 0.08672916666666668
epoch: 09      w: [0.03, -0.042, 0.028] 0.0909375
```

訓練完成後會顯示兩圖表，上圖為Loss

對Epoch的時序資料，下圖為資料集及鏈結值在平面座標上的視覺化圖表。



二. 程式碼簡介

Source Code為Python，單一檔案。

使用到的套件有：matplotlib、tkinter。

主要分為三個部分，Setting GUI、Model、Plot。

GUI使用tkinter製作畫面，設計了兩個input、一個檔案選擇器、一個執行按鈕。

使用mainloop function來等待畫面關閉。

GUI關閉後開始執行Model的資料切割、訓練，並隨時紀錄過程鏈結值的變化及Loss的變化。

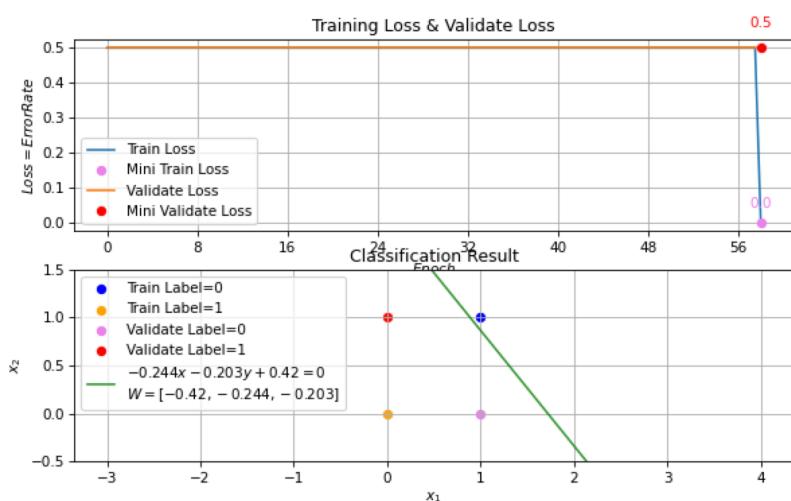
其中Loss主要使用最簡單的Error Rate，另也有實作MSE Loss，但因效果不佳，如需使用請在Source Code把MSE_LOSS的參數改為True。

在Model訓練完後Plot會將Loss、Data、鏈結值等資料繪製成兩張圖表，可使用拖曳及縮放等功能，並等待畫面關閉。

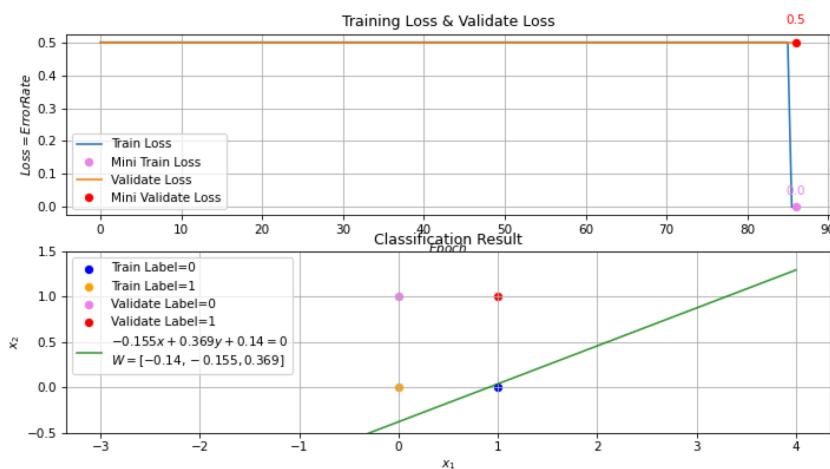
三. 實驗結果

基本題，除有寫明外，設定皆為預設值。

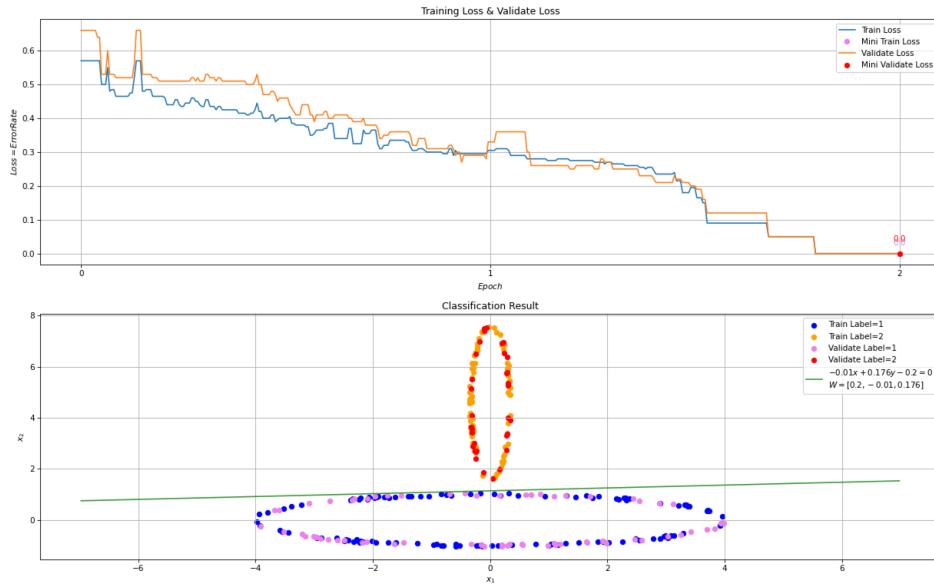
- perceptron1



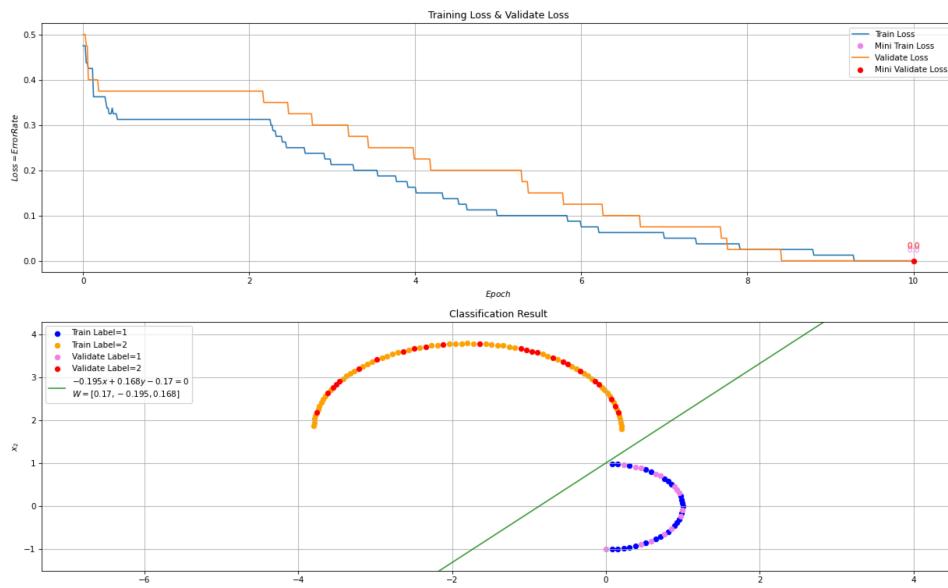
- perceptron2



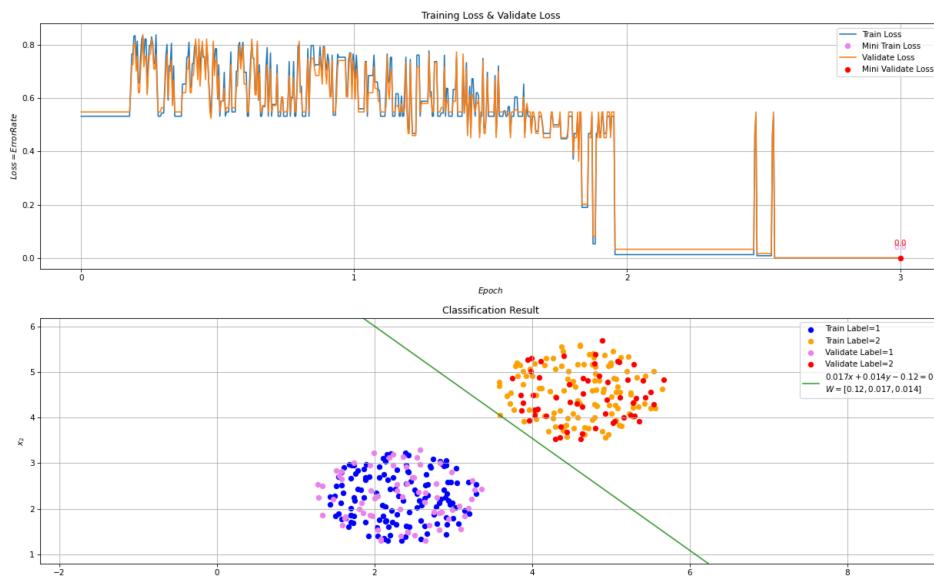
- 2ring



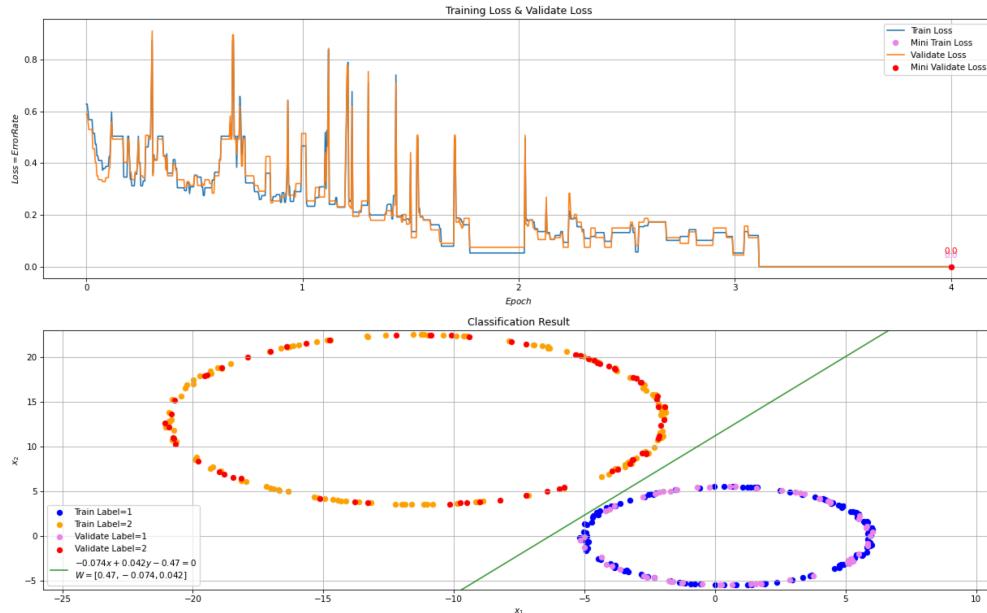
- 2Hcircle1



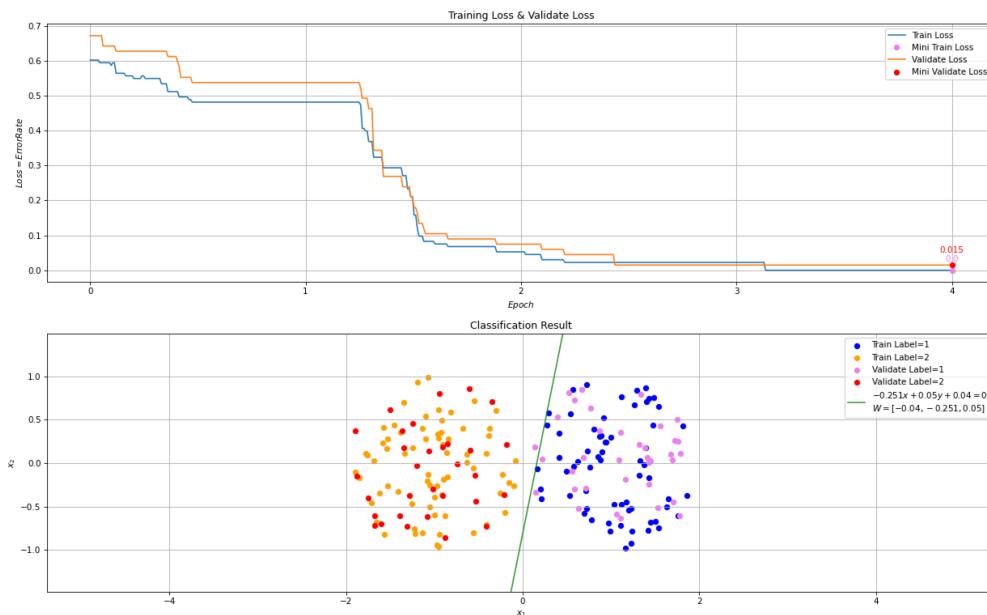
- 2CS



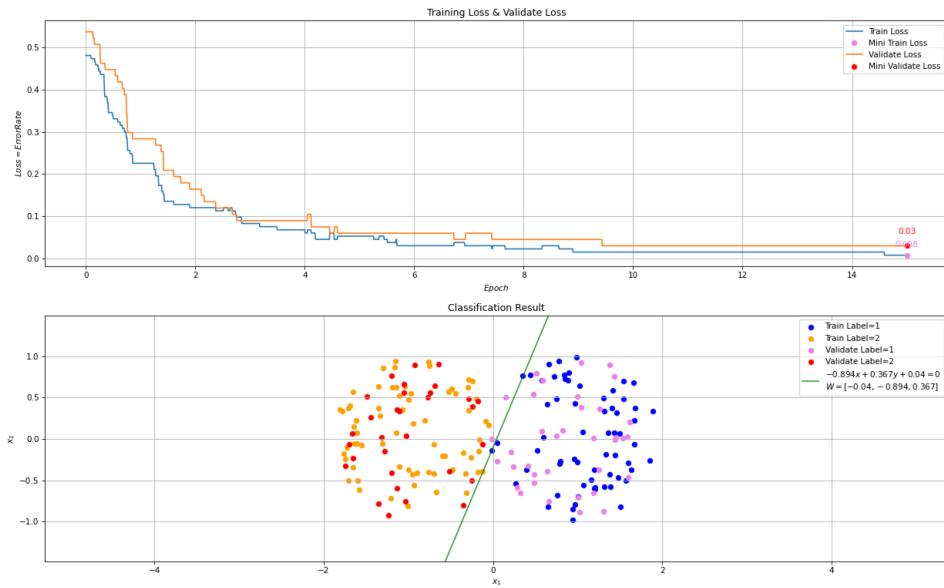
- 2cring



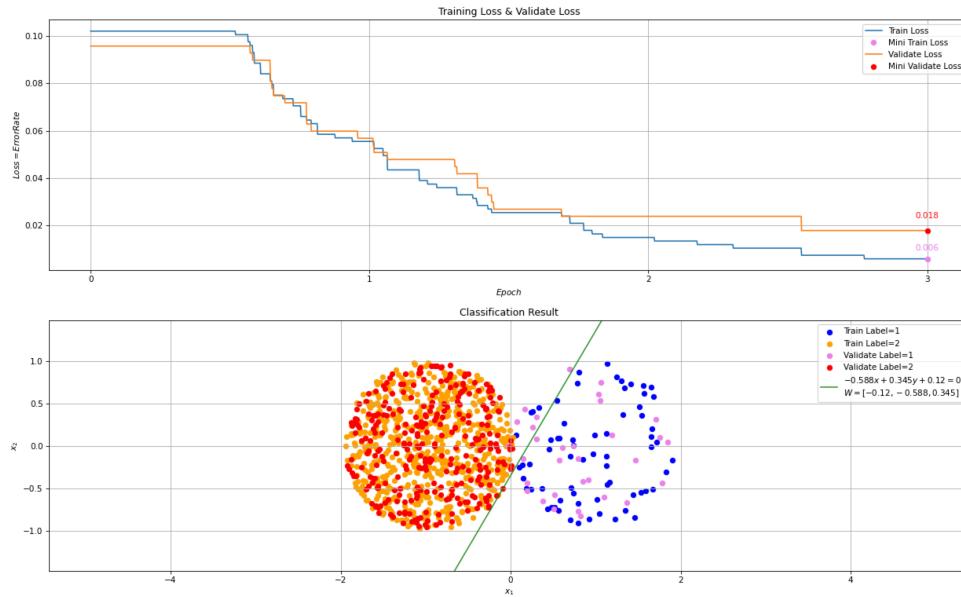
- 2CloseS



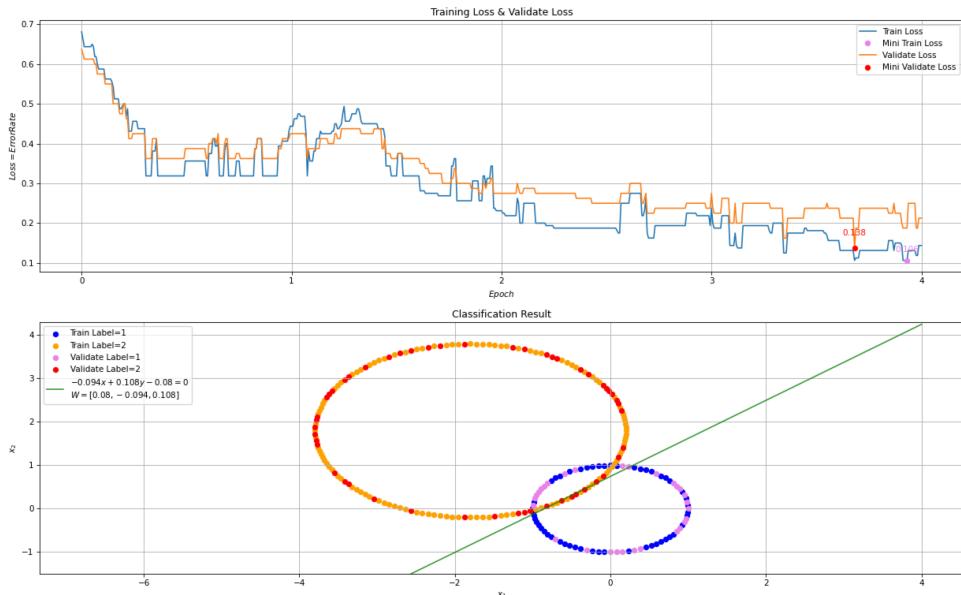
- 2CloseS2



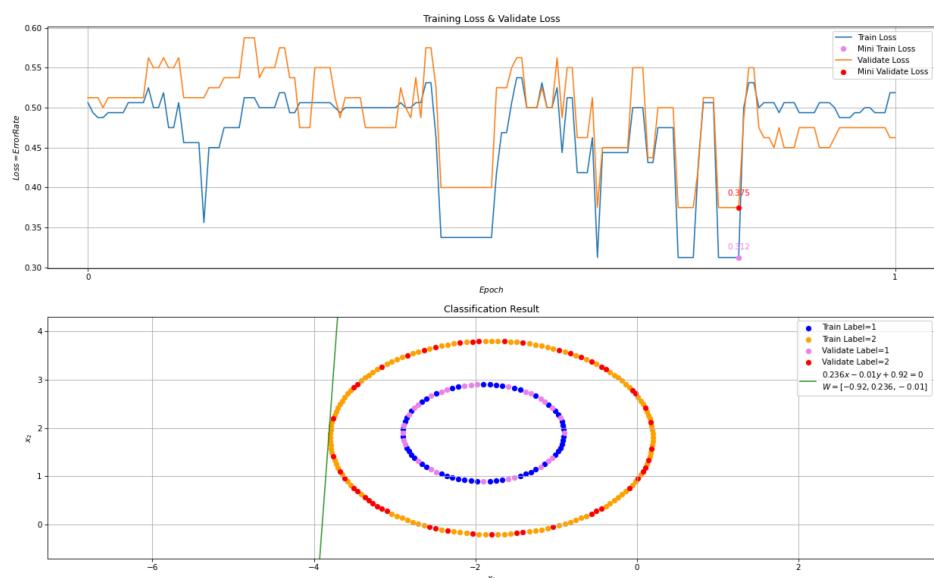
- 2CloseS3



- 2Circle1 , Train End Gate = 0.12

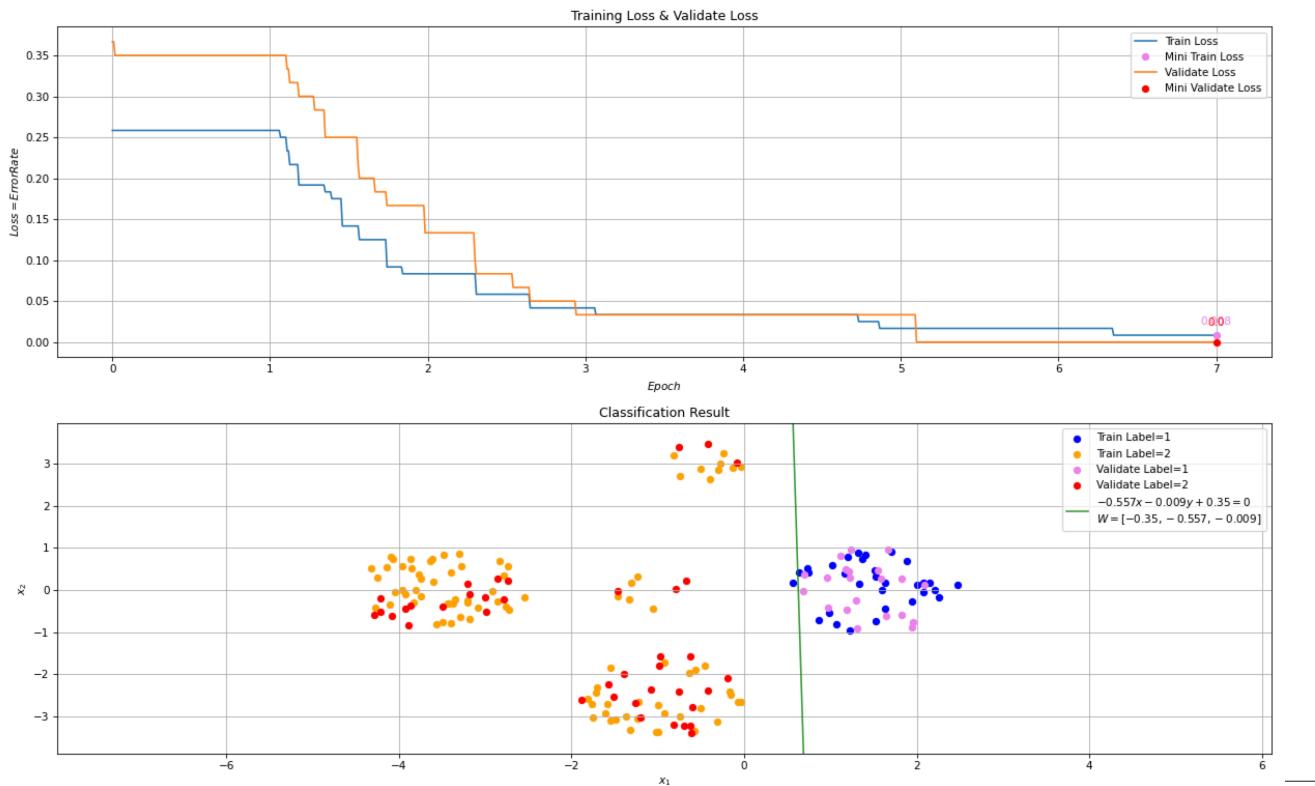


- 2Ccircle1 , Train End Gate = 0.35



加分題

- 5CloseS1



四. 實驗結果分析與討論

對於線性可分割的資料集來說，成果大部分都有正確的分割出兩群，甚至是加分題的5CloseS1，但perceptron1因資料過少，且作業要求1/3作為驗證資料，成果相當的差，但如果只取1/4作為驗證資料，此資料集結果會好很多。而2CloseS3因其中一群資料量較多，Loss容易受影響，若將END_LOSS_RANGE降低效果也會更好。

對於非線性可分割的資料集來說，結果差異相當大。如2Circle1因重疊處資料量較少，模型也可大致分出兩群。但2Ccircle1來說，最低的Loss唯一的可能就是將外圈完全分類正確，內圈完全分類錯誤，因此畫出來的鏈結值結果近乎相當於外圈的切線。

因資料大多都是有明顯的分群，驗證資料的Loss大致上也與上述狀況相同，且Perceptron會在訓練資料分類都正確時停止更新鏈結值，因此在驗證資料集上Loss通常高於訓練資料集。

另外也有發現，雖然模型設定最高Epoch為512，但若為線性可分割資料且資料量夠多，通常在4~6個Epoch就能將Training Data的Loss降至近乎於0。但如果未調整非線性可分割資料集的END_LOSS_RANGE，很容易造成Epoch跑至512仍無法收斂。這現象應該是因為Loss Function的設計過於簡單，若使用MSE、BCE Loss Function，可能有機會較接近預期的結果。Source Code中有嘗試實作MSE，但效果沒有明顯提升，且會反而造成可線性分割資料的結果較差。因此我認為最佳的改良方法是改為多層感知機，且使用Cross Entropy Loss Function。

而Learning Rate我認為0.01已是較佳的設定，我一開始有嘗試照教授簡報中的0.8做設定，但就算達3000Epoch仍無法收斂，後來發現太高的Learning Rate會造成鏈結值改變太大，不斷在最佳解的左右彈跳，且因Perceptron的特性，模型會自然的根據每筆資料進行更新，跟Epoch或Loss其實無關，因此如果Learning Rate太高會造成把每個資料點的權重看得太高。