類神經網路作業 1 108522072 林子恆

1. 程式執行說明

1.

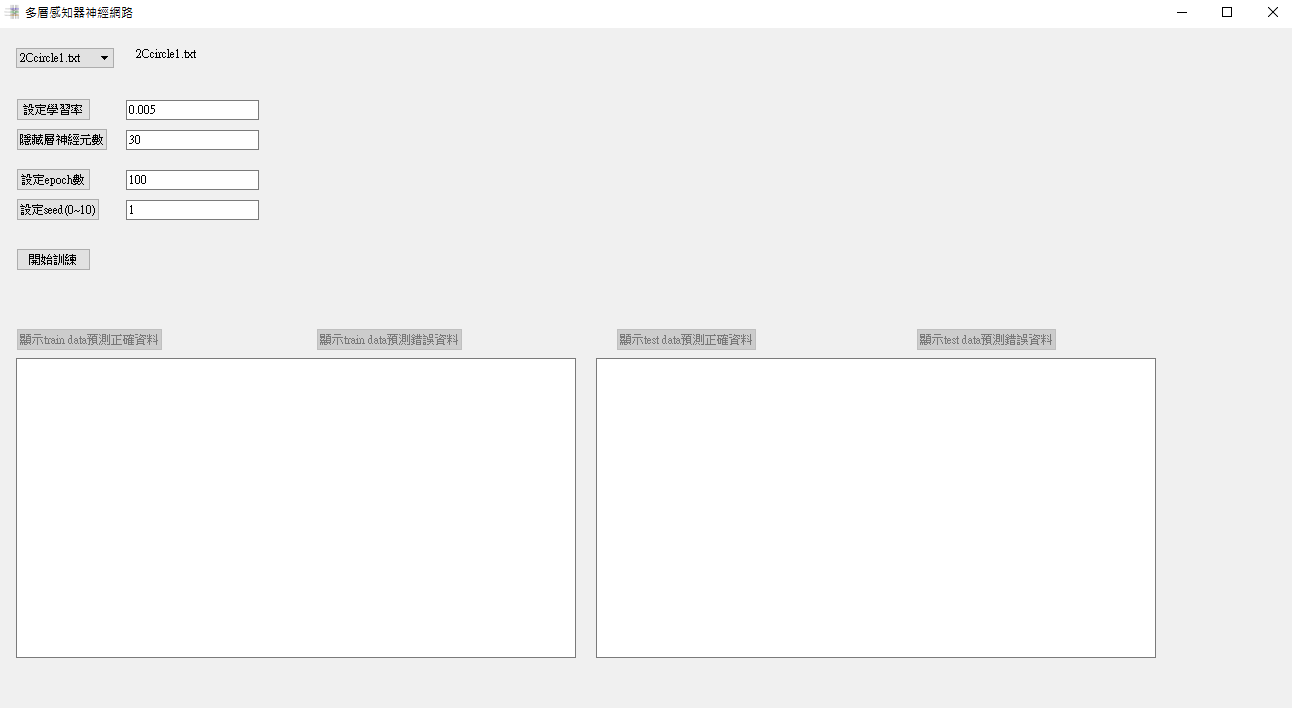
共有一個執行檔，名稱為MLP\_QT.exe、兩份python程式、兩張圖片名稱分別為sorry1.png和sorry2.png、一個名稱為datasets的資料夾，資料夾裡有所有助教給的測試資料集，執行執行檔時務必把所有檔案及資料夾放在同一個目錄下執行。

2.

執行方式為經由cmd到執行檔所在目錄，並輸入 .\MLP\_QT.exe，執行執行檔時會花一些時間。

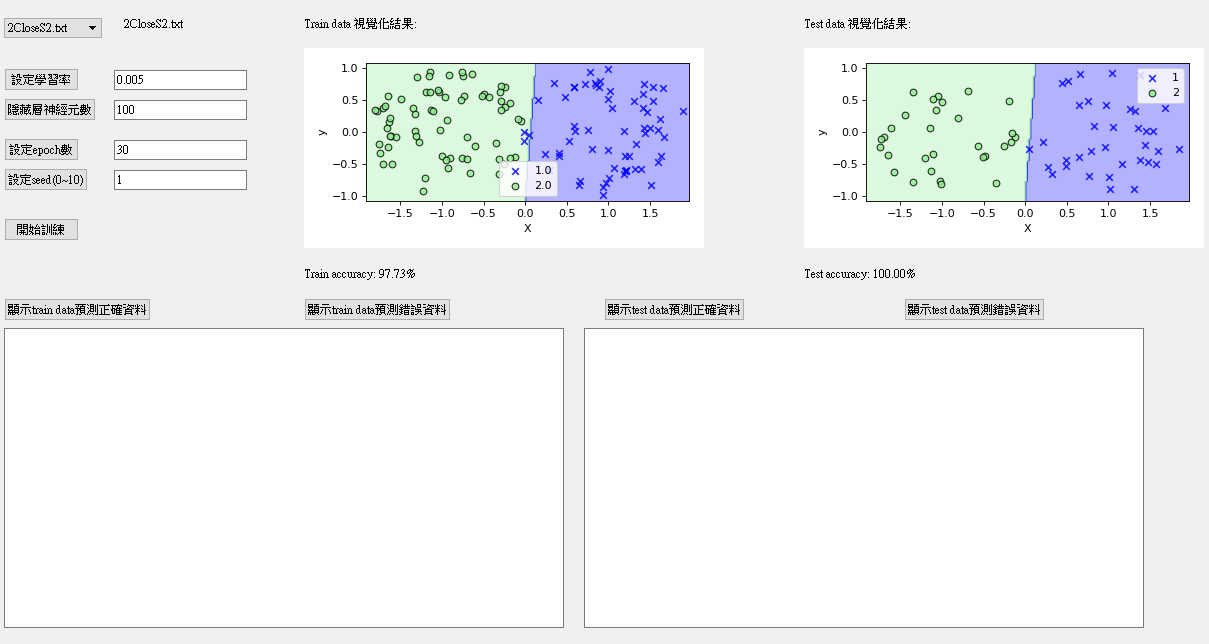
3.

執行程式時會出現如圖一的視窗圖，左上角藍色框部分是選擇要訓練的資料集、及設定神經網路學習率、隱藏層的神經元數目、epoch數及隨機取訓練資料和測試資料的seed。

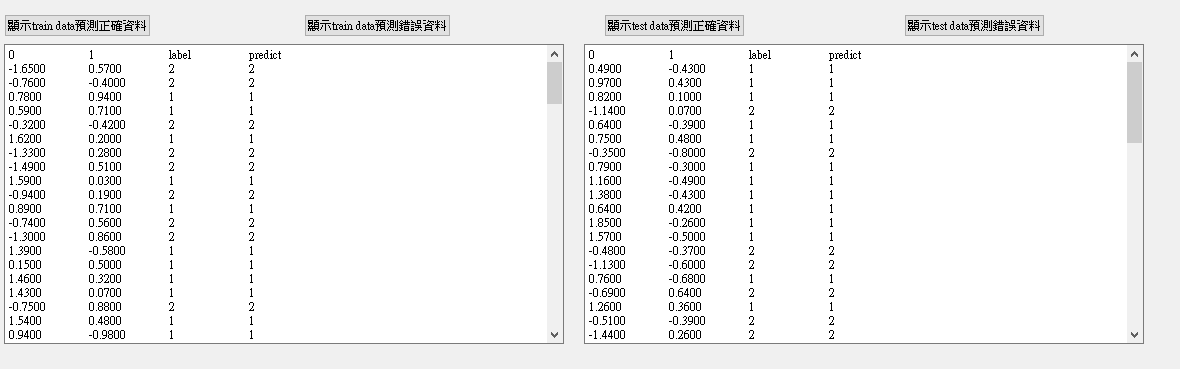


(圖一)

設定完參數之後，按下開始訓練按鈕，會開始訓練並把結果視覺化(圖二) ，紅色框會顯示訓練資料的準確率和測試資料準確率。下方紅色框的四個按鈕，是分別顯示訓練資料預測正確筆數的詳細資料、訓練資料預測錯誤筆數的詳細資料、測試資料預測正確筆數的詳細資料、測試資料預測錯誤筆數的詳細資料(圖三)。



(圖二)

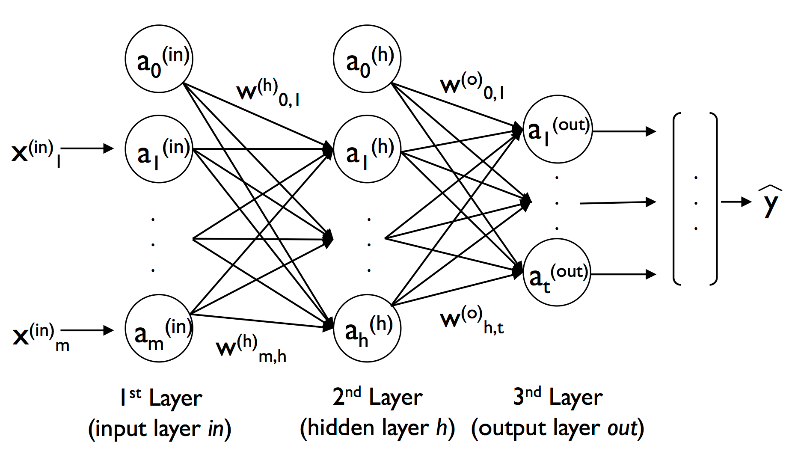


(圖三)

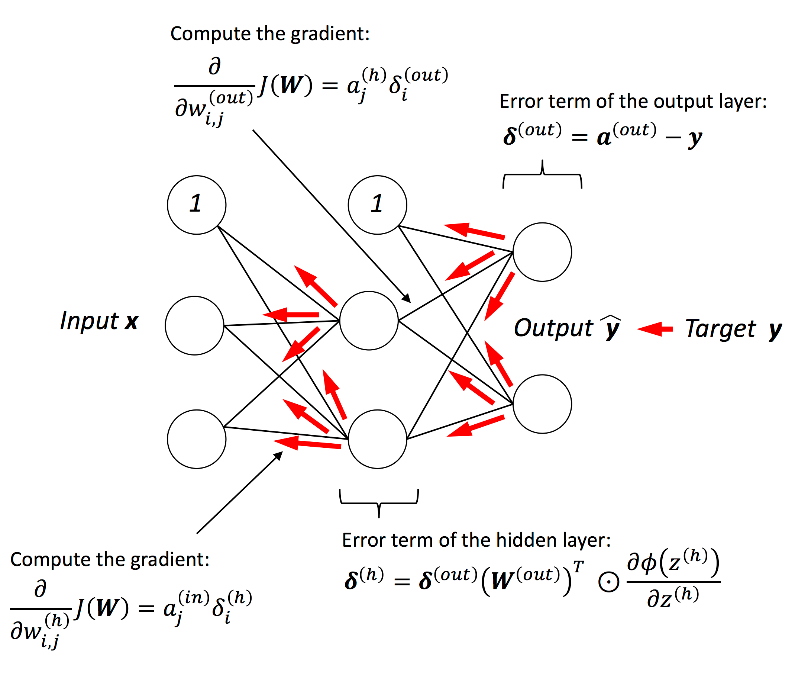
1. 程式簡介

由MLP.py和MLP\_QT.py兩份python程式組成，MLP\_QT.py是編寫GUI圖形介面程式，MLP.py則是多層感知機神經網路，包含一個輸入層、一個隱藏層、一個輸出層，使用者可以藉由圖形介面設定隱藏層的神經元數目。

設計感知機神經網路架構如圖四，其中包含幾項重要功能

 (圖四)

1. 對資料的label做one-hot encoding
2. 實作前饋式神經網路
3. 使用邏輯斯成本函數(logistic coss function)計算神經網路的loss function
4. 實作反傳遞法更新神經網路的weight值(圖五)



(圖五)

1. 實驗結果

以下是對感知器神經網路超參數設定:

學習率: 0.005

隱藏層神經元數:50

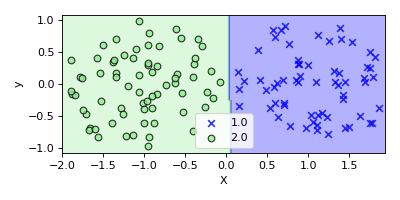
Epoch數:100

Random seed:8

設定好參數之後，把感知器神經網路對每份資料開始訓練結果如下:

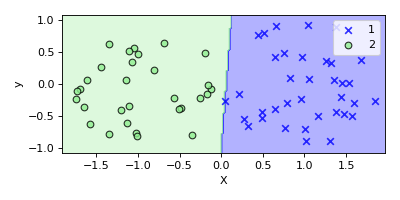
1. 2CloseS.txt檔案:

Train data結果:



準確率: 100%

Test data 結果:



準確率: 100%

結論:視覺化的結果中，平面的顏色為神經網路對所有資料集的形成二

維資料平面，對每個座標點進行預測，並對預測label畫上對應顏色，而

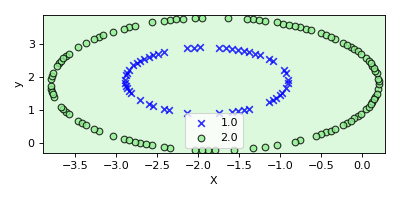
平面上的標記點和顏色則是實際測試資料座標點的label，由圖可以看

出，神經網路預測是否正確(以下資料以此類推)。

而2CloseS.txt檔案中，神經網路可以對所有的label做正確分類

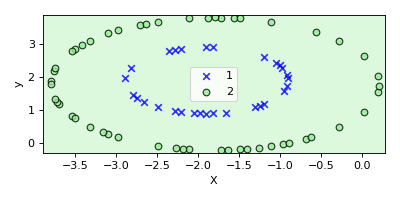
1. 2Ccircle1.txt

Train data結果:



準確率: 68.12%

Test data 結果:



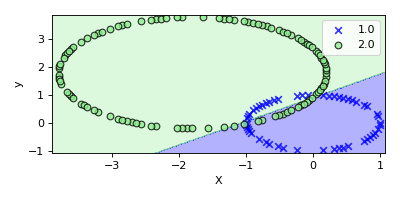
準確率:63.75%

結論: 神經網路把所有label都預測為2



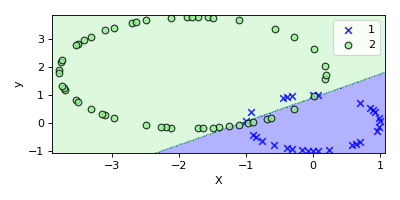
2Circle1.txt

Train data結果:



準確率:82.50%

Test data結果:

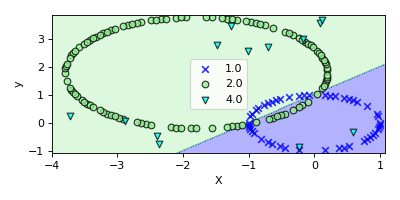


準確率: 83.75%

結論: 可以正確分類80%以上資料

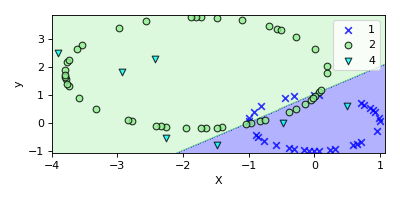
1. 2Circle2.txt

Train data結果:



準確率: 78.49%

Test data 結果:

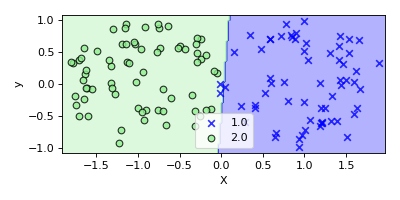


準確率: 73.26%

結論: label 1和label 2大部分資料都有分類正確，但label4資料都分類錯誤。

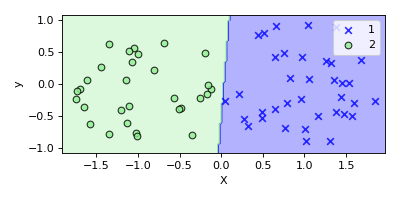
1. 2CloseS2.txt

Train data:



準確率: 98.48%

Test data:

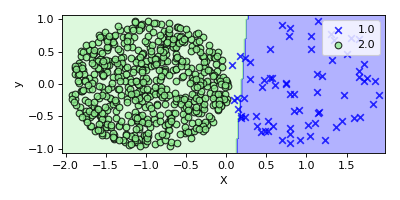


準確率: 100%

結論:可以接近100%分類資料

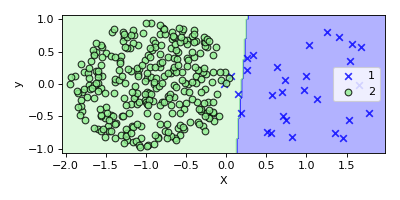
1. 2CloseS3.txt

Train data:



準確率: 99.25%

Test data:

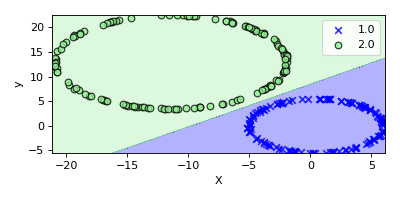


準確率: 99.10%

結論:可以接近100%分類資料

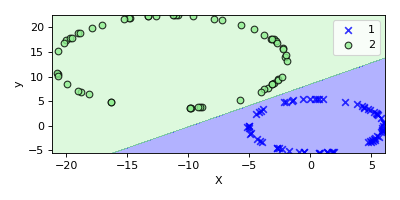
1. 2cring.txt

Train data:



準確率: 100.00%

Test data:

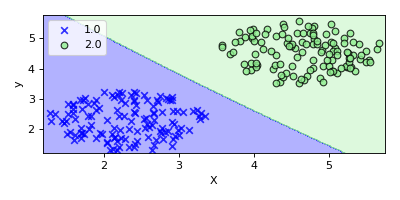


準確率: 100.00%

結論:可以接近100%分類資料

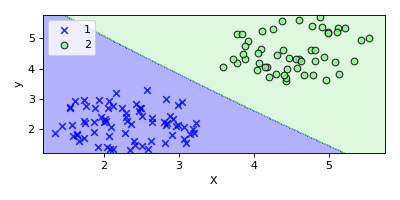
1. 2CS.txt

Train data:



準確率: 100.00%

Test data:

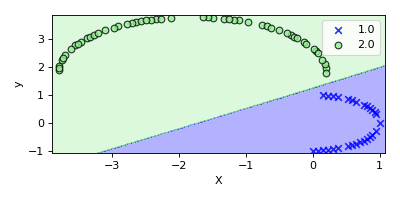


準確率: 100.00%

結論:可以接近100%分類資料

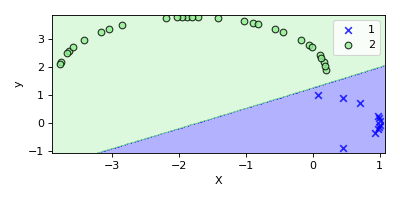
1. 2Hcircle1.txt

Train data:



準確率: 100.00%

Test data:

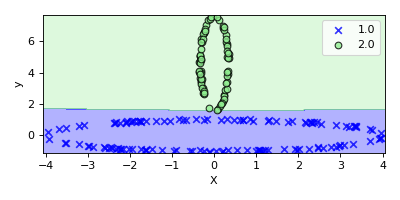


準確率: 100.00%

結論:可以接近100%分類資料

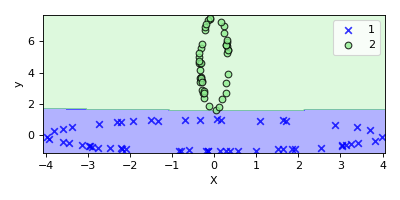
1. 2ring.txt

Train data:



準確率: 100.00%

Test data:



準確率: 100.00%

結論:可以接近100%分類資料

1. 4-satellite-6.txt

Train data: 因為維度資料超過二維,很抱歉無法視覺化呈現train data!

準確率: 27.80%

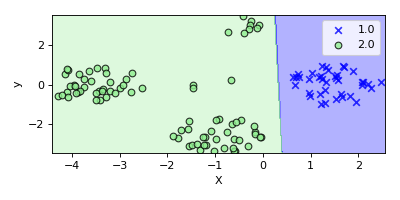
Test data: 因為維度資料超過二維,很抱歉無法視覺化呈現test data!

準確率: 27.37%

結論:分類結果不好

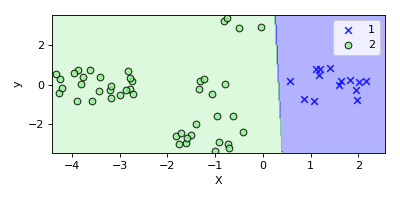
1. 5CloseS1.txt

Train data:



準確率: 100.00%

Test data:



準確率: 100.00%

結論:可以接近100%分類資料

1. 8XOR.txt

Train data: 因為維度資料超過二維,很抱歉無法視覺化呈現train data!

準確率: 100.00%

Test data: 因為維度資料超過二維,很抱歉無法視覺化呈現test data!

準確率: 66.67%

結論:可以分類66%筆測試資料

1. C10D.txt

Train data: 因為維度資料超過二維,很抱歉無法視覺化呈現train data!

準確率: 100.00%

Test data: 因為維度資料超過二維,很抱歉無法視覺化呈現test data!

準確率: 100.00%

結論:可以分類100%筆測試資料

1. C3D.txt

Train data: 因為維度資料超過二維,很抱歉無法視覺化呈現train data!

準確率: 100.00%

Test data: 因為維度資料超過二維,很抱歉無法視覺化呈現test data!

準確率: 100.00%

結論:可以分類100%筆測試資料

1. IRIS.TXT

Train data: 因為維度資料超過二維,很抱歉無法視覺化呈現train data!

準確率: 69.00%

Test data: 因為維度資料超過二維,很抱歉無法視覺化呈現test data!

準確率: 62.00%

結論:可以分類60%筆測試資料

1. Number.txt

Train data: 因為維度資料超過二維,很抱歉無法視覺化呈現train data!

準確率: 33.33%

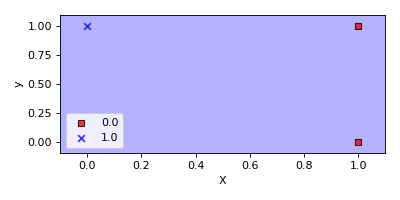
Test data: 因為維度資料超過二維,很抱歉無法視覺化呈現test data!

準確率: 0.00%

結論: 資料筆數太少，導致訓練overfitting

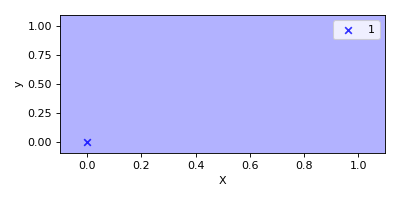
1. Perceptron1.txt

Train data:



準確率: 33.33%

Test data:

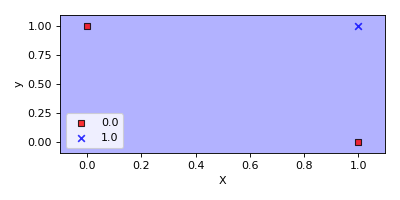


準確率: 100.00%

結論: 資料筆數太少，導致神經網路訓練overfitting

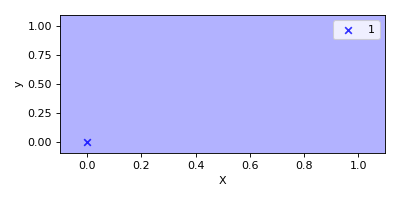
1. Perceptron2.txt

Train data:



準確率: 33.33%

Test data:



準確率: 100.00%

結論: 資料筆數太少，導致神經網路訓練overfitting

1. Perceptron3.txt

Train data: 因為維度資料超過二維,很抱歉無法視覺化呈現train data!

準確率: 33.33%

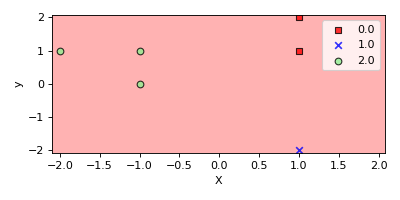
Test data: 因為維度資料超過二維,很抱歉無法視覺化呈現test data!

準確率: 100.00%

結論: 資料筆數太少，導致訓練overfitting

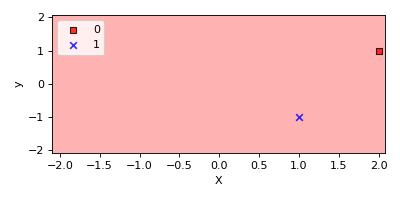
1. Perceptron4.txt

Train data:



準確率: 33.33%

Test data:



準確率: 50.00%

結論: 資料筆數太少，導致神經網路訓練overfitting

1. wine.txt

Train data: 因為維度資料超過二維,很抱歉無法視覺化呈現train data!

準確率: 35.59%

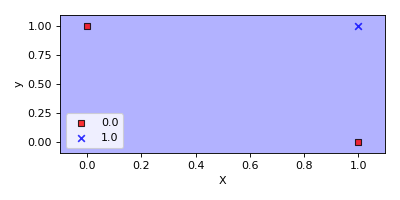
Test data: 因為維度資料超過二維,很抱歉無法視覺化呈現test data!

準確率: 27.12%

結論: 訓練結果不好

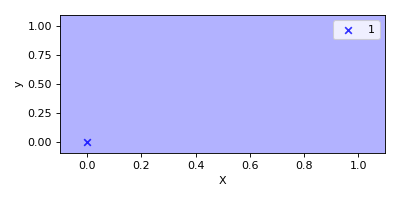
1. xor.txt

Train data:



準確率: 33.33%

Test data:



準確率: 100.00%

結論: 資料筆數太少，導致神經網路訓練overfitting

1. 實驗結果分析與討論

由上述實驗可以發現，當訓練的資料是不是線性可分割平面時，自己設計的類神經網路都無法完美分割，原因是在於我只有實作一層hidden layer，因此如果要增進非線性可分割平面資料準確率時，應該要時做多層hidden layer，這是第一個可以改進的地方。再來如果資料筆數太少，訓練時容易overfitting，除此之外大部分線性可分割資料及多label資料都可以準確分辨80%資料。

1. 預計可以完成的加分項目

1.能夠處理多維資料(四維以上)

2.可辨識兩群以上的資料 (神經網路可以成功辨C3D.txt、C10D.txt、IRIS.txt等資料)