

A, 實作多層感知機 可自由設定層數及隱藏層神經元個數

B, 程式執行說明 我用的程式語言是 Python, 環境是 colab 開發自google, 點進

每一個並按下 Shift+Enter 即可執行, 程式必須從上執行到下, 其中, 以下程式

碼決定使用的資料:

```
data = np.loadtxt("2ring.txt",dtype=np.float,delimiter=' ')
```

```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.001)
4 nn.train(1)
5
```

參數分別是

feature_num, class_num, hiddenLayer_num (設定層數), hiddenNeural_num (設定隱藏層神經元個數), data_x, data_y, learning_rate = 0.1 (學習率)

nn.train(1) 表示開始訓練一次

如上圖, 第一行的最後 0.66 即將資料分成 2/3 訓練 1/3 測試

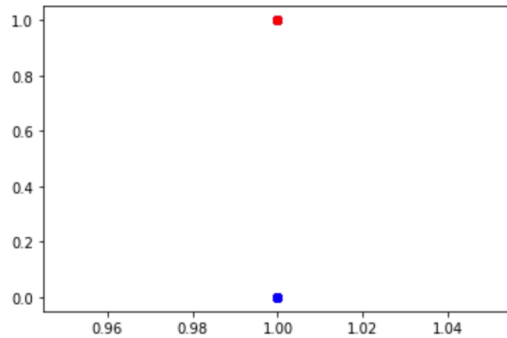
C, 以下依序顯示實驗結果並上訓練結果和測試結果

perceptron1.txt、perceptron2.txt、2Ccircle1、2Circle1、2Circle2、2CloseS、
2CloseS2、2CloseS3、2cring、2CS、 2Hcircle1、2ring

perceptron1.txt

```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.1)
4 nn.train(100)
5
```

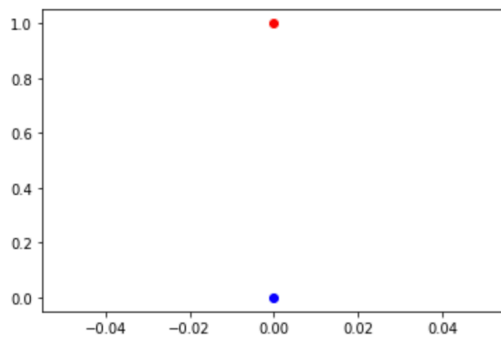
RMSE=
0.47948598544076193
辨識率
0.625



'train done'

```
1 nn.data_x = X_test
2 nn.data_y = y_test
3 nn.train(1)
```

RMSE=
0.47986313373026773
辨識率
0.6237623762376238



'train done'

hidden layer 鍵結值

```
[[ 1.15339496  0.32211336]
 [-0.20071091 -0.03261026]
 [ 0.4980058   0.21024325]]
```

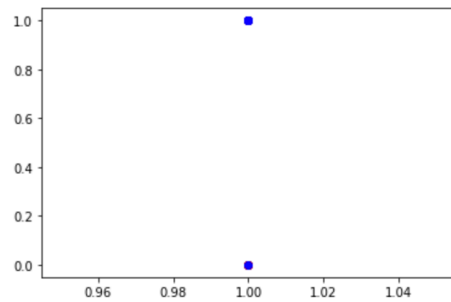
output layer 鍵結值

```
array([[ 0.34272677,  0.36178769,  0.36363878],
       [ 0.00923291,  0.02829383,  0.03014492],
       [-0.00465802,  0.0144029 ,  0.01625399]])
```

perceptron2.txt

```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 1)
4 nn.train(100)
5
```

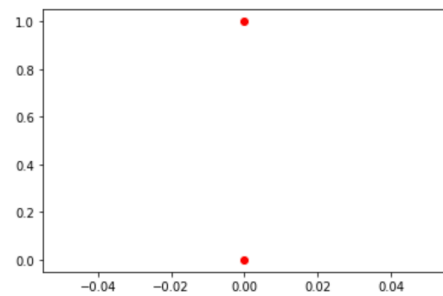
RMSE=
0.486955430352238
辨識率
0.59



'train done'

```
1 nn.data_x = X_test
2 nn.data_y = y_test
3 nn.train(1)
```

RMSE=
0.4874547132009893
辨識率
0.5891089108910891



'train done'

hidden layer 鍵結值

```
[[ 0.99509217  0.56041317]
 [ 0.06307201 -0.12134155]
 [ 1.47766012  1.90540887]]
```

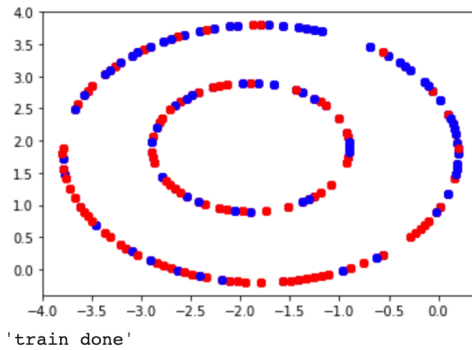
output layer 鍵結值

```
array([[0.99303188, 0.79311108, 0.77294739],
       [1.05469514, 0.85477435, 0.83461066],
       [0.43007793, 0.23015714, 0.20999345]])
```

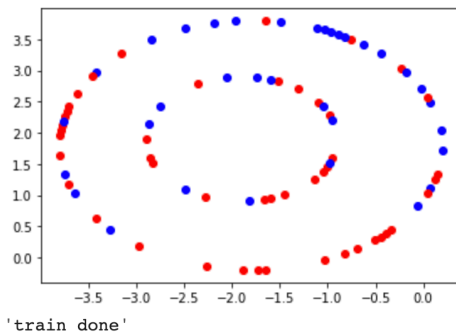
2Ccircle1

```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.01)
4 nn.train(5)
5
```

RMSE=
1.2952952647646756
辨識率
0.45031446540880504



RMSE=
1.2934008699396637
辨識率
0.454337899543379



hidden layer 鍵結值

```
[[0.1091492  0.7465109 ]
 [0.60538742 0.2107549 ]
 [0.51064649 0.87281501]]
```

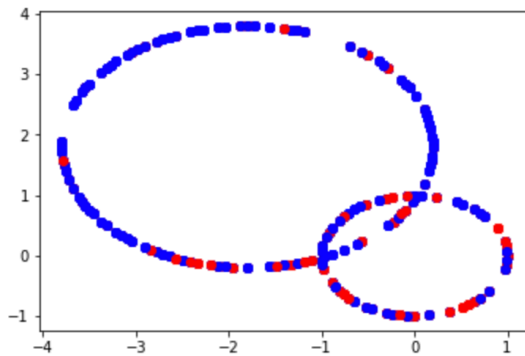
output layer 鍵結值

```
array([[0.38081204, 0.38252619, 0.38290386],
       [0.19129469, 0.19300883, 0.1933865 ],
       [0.32985697, 0.33157112, 0.33194879]])
```

2Circle1

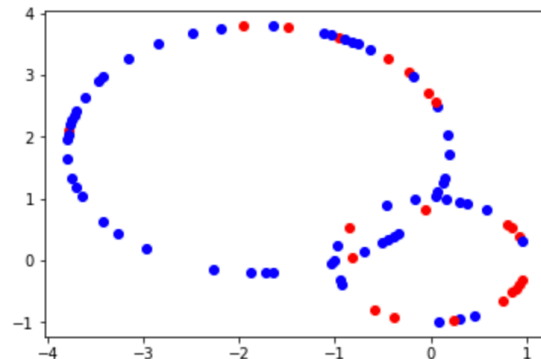
```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.1)
4 nn.train(5)
5
```

RMSE=
1.1938203357368742
辨識率
0.6352201257861635



'train done'

RMSE=
1.1938626347671475
辨識率
0.6415525114155252



'train done'

2Circle2

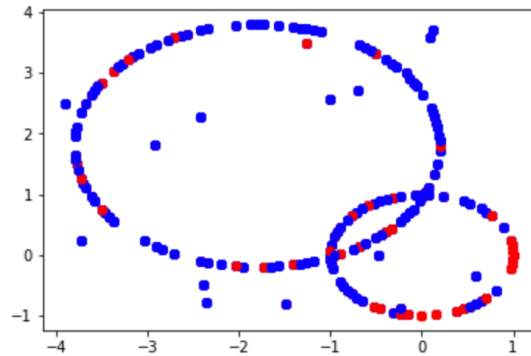
```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.1)
4 nn.train(5)
5
```

RMSE=

1.528266519254069

辨識率

0.5734104046242775



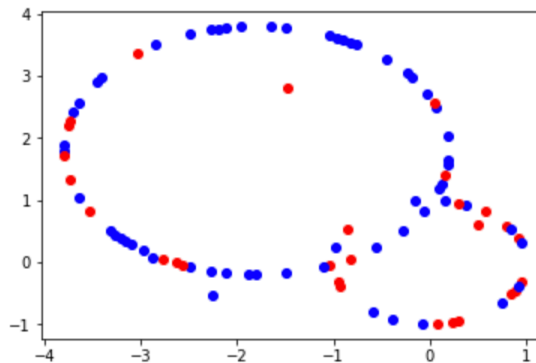
'train done'

RMSE=

1.5161084120791963

辨識率

0.582983193277311



'train done'

hidden layer 鍵結值

`[[-1.06085486 -1.16718687]`

`[-0.88955627 -0.82555929]`

`[1.04195278 0.97582249]]`

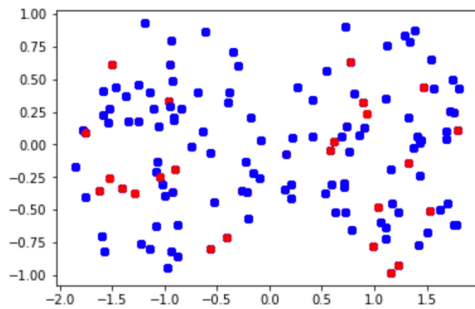
output layer 鍵結值

`array([[0.25601676, 0.27682897, 0.27714872],`
 `[0.02434561, 0.04515782, 0.04547757],`
 `[0.35452983, 0.37534204, 0.37566179]])`

2CloseS

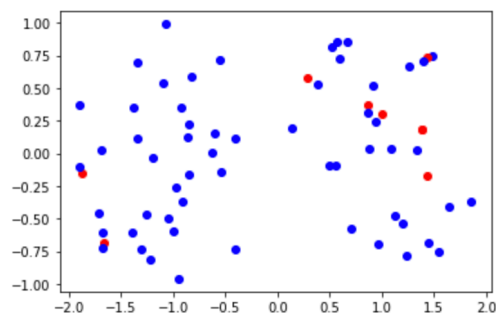
```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.5)
4 nn.train(10)
5
```

RMSE=
1.0358154778064377
辨識率
0.5090225563909775



'train done'

RMSE=
1.0348795989759603
辨識率
0.5125268432355047



'train done'

hidden layer 鍵結值

```
[[-3.93343738 -3.91630262]
 [-0.48585256 -0.5015177 ]
 [-0.15395835 -0.02640925]]
```

output layer 鍵結值

```
array([[0.82060658, 0.9433373 , 0.9432988 ],
       [0.33044357, 0.45317429, 0.45313579],
       [0.461042 , 0.58377272, 0.58373422]])
```

2CloseS2

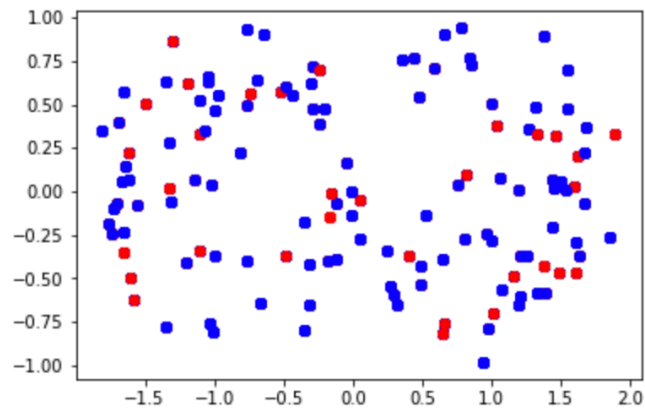
```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.1)
4 nn.train(10)
5
```

RMSE=

1.08428753669029

辨識率

0.47443609022556393



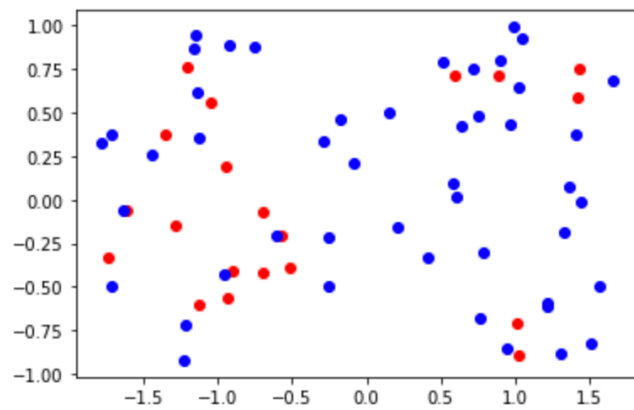
'train done'

RMSE=

1.0612505646432804

辨識率

0.5089477451682176



'train done'

hidden layer 鍵結值

[[-2.16635622 -2.18829693]

[-0.48460741 -0.42381584]

[0.77628363 0.24970953]]

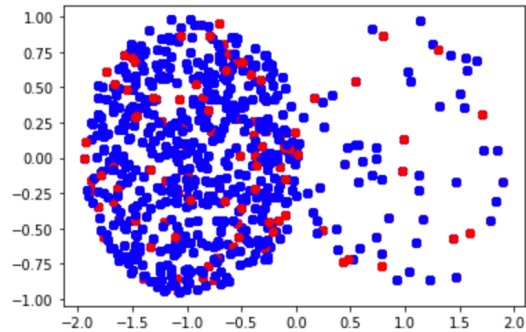
output layer 鍵結值

array([[0.13122554, 0.14319937, 0.14301347],
[0.90353361, 0.91550743, 0.91532154],
[0.20699007, 0.21896389, 0.21877799]])

2CloseS3

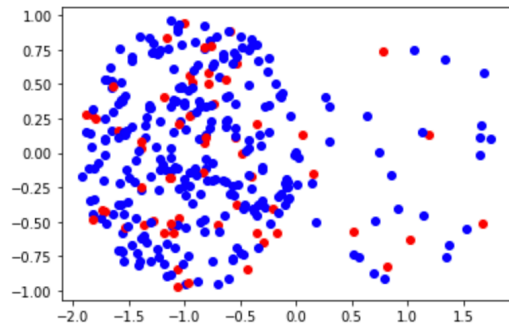
```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.1)
4 nn.train(10)
5
```

RMSE=
1.3336131962677666
辨識率
0.7600600600600601



'train done'

RMSE=
1.3334314815388004
辨識率
0.7603660280240205



'train done'

hidden layer 鍵結值

```
[[-3.61974706 -3.74819806]
 [-1.6926329  -1.45766505]
 [ 0.43418523  0.40353004]]
```

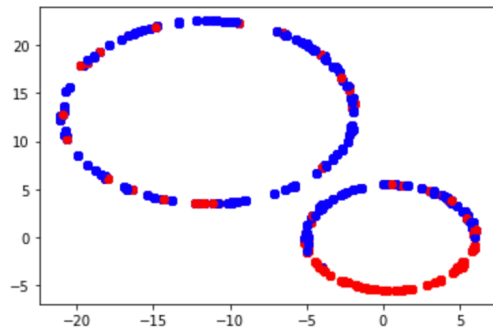
output layer 鍵結值

```
array([[0.93864677, 1.01424211, 1.0142182 ],
       [0.15137819, 0.22697353, 0.22694962],
       [0.19696602, 0.27256136, 0.27253745]])
```

2cring

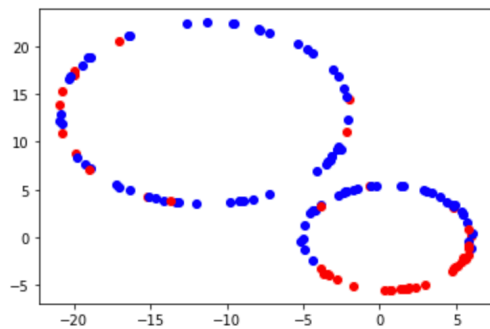
```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.1)
4 nn.train(5)
5
```

RMSE=
1.0525149703448016
辨識率
0.7022556390977444



'train done'

RMSE=
1.051013870604339
辨識率
0.6974043715846995



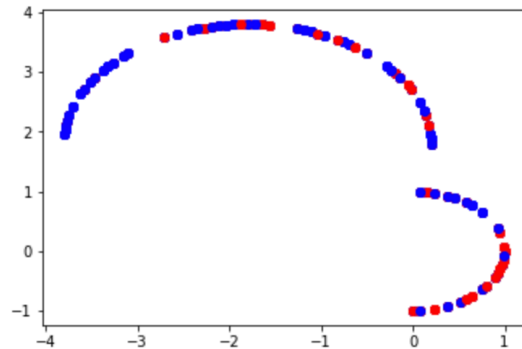
'train done'

hidden layer 鍵結值
[[-0.22600176 -0.25207515]
[0.07634592 -0.03431864]
[1.27552356 0.89306898]]
output layer 鍵結值
array([[0.73928523, 0.81522135, 0.8152641],
[0.15713513, 0.23307124, 0.233114],
[0.37130861, 0.44724472, 0.44728747]])

2Hircle1

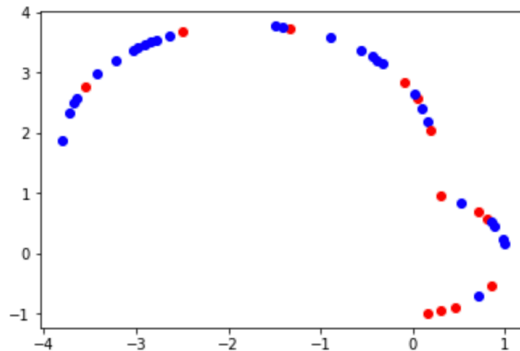
```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.01)
4 nn.train(5)
5
```

RMSE=
1.2103480904661863
辨識率
0.6708860759493671



'train done'

RMSE=
1.2097492896786068
辨識率
0.6743119266055045



'train done'

hidden layer 鍵結值

```
[[ -0.22600176 -0.25207515]
 [  0.07634592 -0.03431864]
 [  1.27552356  0.89306898]]
```

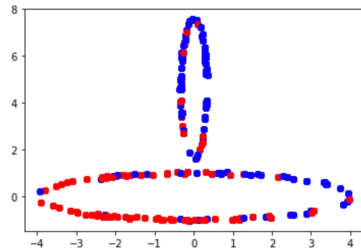
output layer 鍵結值

```
array([[0.73928523, 0.81522135, 0.8152641 ],
       [0.15713513, 0.23307124, 0.233114  ],
       [0.37130861, 0.44724472, 0.44728747]])
```

2ring

```
1 X_train, X_test, y_train, y_test = load_train_test_data(train_ratio=0.666)
2
3 nn = MultiNeuralNetwork(X_train.shape[1], 1, 1, 2, X_train, y_train, 0.01)
4 nn.train(5)
5
```

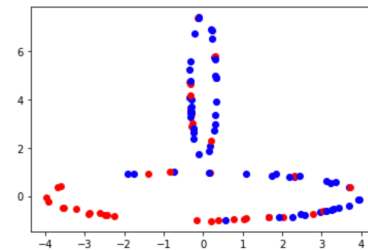
RMSE=
0.9736175394632313
辨識率
0.7185929648241206



'train done'

```
1 nn.data_x = X_test
2 nn.data_y = y_test
3 nn.train(1)
```

RMSE=
0.9782965738695297
辨識率
0.708029197080292



'train done'

hidden layer 鍵結值

```
[[0.31573665 0.82912899]
 [0.69849699 0.63220223]
 [0.65842176 0.90473376]]
```

output layer 鍵結值

```
array([[0.08764134, 0.09382421, 0.09388306],
       [0.15957286, 0.16575573, 0.16581459],
       [0.41791918, 0.42410205, 0.42416091]])
```

D, 訓練結果不太理想，有試過換不同的激發函數但結果依舊不佳，有想過是不是計算或是步驟有問題，後來有試過把根據老師的範例去設定值，出來的值是一樣的，但就分類分的不甚理想。另外有發現就是 learning rate 若設太大會不容易有好的結果。

E, 加分項目：隱藏層層數設定、隱藏層神經元個數設定