

## 1. Perceptron\_AND

- (1)讀取訓練資料"AND\_training.txt"和測試資料"AND\_testing.txt"。
- (2)初始化 train 函數需要的變量，使用數據開始訓練。
- (3)訓練後，得到權重 W。
- (4)將測試數據和訓練好的權重放入 train 函數，得到測試資料的預測值。
- (5)透過測試函數獲取測試數據的準確率。

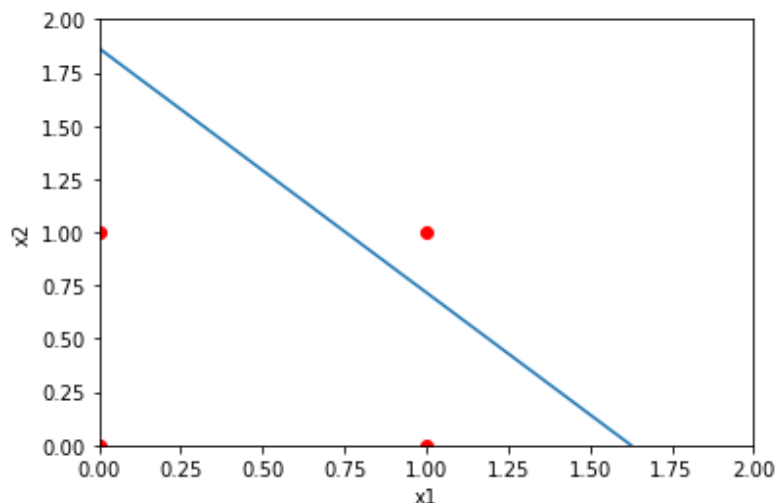
神經網路架構：

Epochs 設為 9000

Learning rate 設為 0.15

執行畫面：

```
In [27]: runfile('G:/NCKU Graduate
AND')
W0: 0.7775817484711453
W1: 0.47834004532740315
W2: 0.4171117995855723
Total accuracy is: 1.0
```



## 2. Backpropagation\_XOR

神經網路架構：

input:  $X = [x_1, x_2]$

Layer1(隱藏層)有兩個 Neuron: Neuron1、Neuron2

Neuron1:

weights:  $W11 = [W11-1, W11-2]$

bias:  $b11$

$z11 = W11 * X + b11$

$a11 = \text{sigmoid}(z11)$

Neuron2:

weights:  $W12 = [W12-1, W12-2]$

bias: b12

$z_{12} = W_{12} * X + b_{12}$

$a_{12} = \text{sigmoid}(z_{12})$

Layer2 有一個 Neuron: Neuron3

Neuron3:

weights:  $W_{21} = [W_{21-1}, W_{21-2}]$

bias: b21

$z_{21} = W_{21} * X + b_{21}$

$a_{21} = \text{sigmoid}(z_{21})$

Loss function 為 cross-entropy loss

Epochs 設為 9000

Learning rate 設為 0.05

Momentum 設為 0.85

執行畫面：

```
In [24]: runfile('G:/NCKU Graduate/神經網路/hw1/XOR/xor_bpn.py')
訓練後權重:
Neuron1(隱藏層): 6.952665903411992    6.954245088689913
Neuron2(隱藏層): 5.0494942988441665    5.048597442090256
Neuron3: 10.97026986227679    -11.75168253973326

測試結果:
0.0  0.0  [0.00982845109094075]
0.0  1.0  [0.9927044937137072]
1.0  0.0  [0.9926972839269411]
1.0  1.0  [0.0077958529526895234]
```

Training Loss:

