

## Assignment 1

太空三 109607504 賴以芸

### 1. 編譯結果

```
yeeeyun0301@LAPTOP-V9DEG1U3:/mnt/d/assignment_1-yeeeyun0301/src$ gcc -o main main.c layer.c neuron.c -lm
```

### 2. 執行結果

```
Enter input to test:
0
0
Output: 0

Enter input to test:
0
1
Output: 1

Enter input to test:
1
0
Output: 1

Enter input to test:
1
1
Output: 0
```

-----執行過程-----

```
yeeeyun0301@LAPTOP-V9DEG1U3:/mnt/d/assignment_1-yeeeyun0301/src$ ./main
Enter the number of Layers in Neural Network:
3
Enter number of neurons in layer[1]:
2
Enter number of neurons in layer[2]:
4
Enter number of neurons in layer[3]:
1

Created Layer: 1
Number of Neurons in Layer 1: 2
Neuron 1 in Layer 1 created
Neuron 2 in Layer 1 created

Created Layer: 2
Number of Neurons in Layer 2: 4
Neuron 1 in Layer 2 created
Neuron 2 in Layer 2 created
Neuron 3 in Layer 2 created
Neuron 4 in Layer 2 created

Created Layer: 3
Number of Neurons in Layer 3: 1
Neuron 1 in Layer 3 created

Initializing weights...
0:w[0][0]: 0.840934
1:w[0][0]: 0.962804
2:w[0][0]: 0.325800
3:w[0][0]: 0.949878
0:w[0][1]: 0.272300
1:w[0][1]: 0.006064
2:w[0][1]: 0.539701
3:w[0][1]: 0.672473
0:w[1][0]: 0.133101
0:w[1][1]: 0.272046
0:w[1][2]: 0.124568
0:w[1][3]: 0.298000

Neural Network Created Successfully...
```

```
Enter the learning rate (Usually 0.15):
0.15

Enter the number of training examples:
4

Enter the Inputs for training example[0]:
0
0

Enter the Inputs for training example[1]:
0
1

Enter the Inputs for training example[2]:
1
0

Enter the Inputs for training example[3]:
1
1

Enter the Desired Outputs (Labels) for training example[0]:
0

Enter the Desired Outputs (Labels) for training example[1]:
1

Enter the Desired Outputs (Labels) for training example[2]:
1

Enter the Desired Outputs (Labels) for training example[3]:
0
```

### 3. 分析

隱藏層層數	隱藏層 neuron 數		執行成果	結果
1	2		<pre> Enter input to test: 0 0 Output: 1  Enter input to test: 0 1 Output: 1  Enter input to test: 1 1 Output: 0  Enter input to test: 1 0 Output: 1 </pre>	(0, 0)的輸出結果錯誤
1	3		<pre> Enter input to test: 0 0 Output: 0  Enter input to test: 0 1 Output: 1  Enter input to test: 1 0 Output: 1  Enter input to test: 1 1 Output: 0 </pre>	皆正確
2	2	2	<pre> Enter input to test: 0 0 Output: 0  Enter input to test: 0 1 Output: 1  Enter input to test: 1 0 Output: 1  Enter input to test: 1 1 Output: 0 </pre>	皆正確

可以看到利用類神經學習在計算 XOR 運算中，只要中間隱藏層有多一層，或是在隱藏層多一個神經元，都可以達到我們預想的計算成果。

#### 4. 心得

在學期初的時候我一直無法理解類神經網路學習的架構，為什麼要訓練模型時要 **backpropagation**，權重更新，但從老師提供的網站連結以及助教課，再加上我剛好這個學期系上的必修課內容為機器學習，讓我搞清楚類神經網路學習的步驟，以及模型訓練的相關知識。雖然今天要運算的是 **XOR** 而已，但此架構可大幅運用在較複雜的計算中，也難怪類神經網路成為現今熱門的話題，雖然一開始要花時間理解，但會使用之後很多複雜的問題都能迎刃而解，不需要人類反覆計算，丟給電腦運算就好，人類就能去做更多有意義的思考或精進。