Assignment #1

1. 編譯結果

```
tsai@LAPTOP-HJB975Q4:~/projects/helloworld$ gcc -c assignment_108501537_1.c
tsai@LAPTOP-HJB975Q4:~/projects/helloworld$
```

2. 執行結果

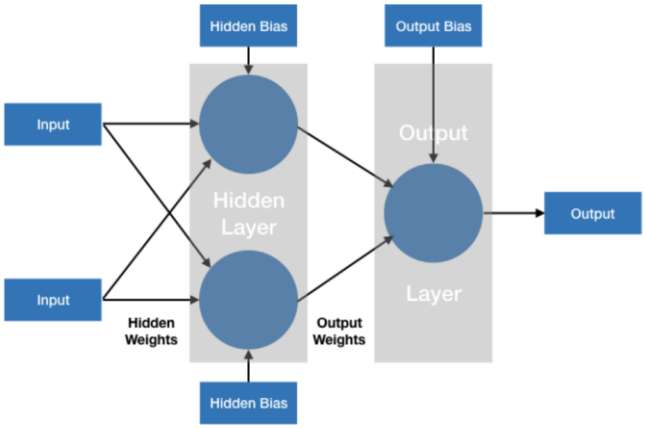
```
Train Time : 9991
Input:0 0    Output:0.058561    Expected Output: 0    Loss Function MSE: 0.001497
Input:1 0    Output:0.945323    Expected Output: 1    Loss Function MSE: 0.001715
Input:0 1    Output:0.945343    Expected Output: 1    Loss Function MSE: 0.001495
Input:1 1    Output:0.059606    Expected Output: 0    Loss Function MSE: 0.001494
Train Time : 9992
Input:1 0    Output:0.945349    Expected Output: 1    Loss Function MSE: 0.001776
Input:0 1    Output:0.945369    Expected Output: 1    Loss Function MSE: 0.001493
Input:0 0    Output:0.058593    Expected Output: 0    Loss Function MSE: 0.001492
Input:1 1    Output:0.059599    Expected Output: 0    Loss Function MSE: 0.001717
Train Time : 9993
Input:1 0    Output:0.945355    Expected Output: 1    Loss Function MSE: 0.001776
Input:0 0    Output:0.058568    Expected Output: 0    Loss Function MSE: 0.001493
Input:0 1    Output:0.945354    Expected Output: 1    Loss Function MSE: 0.001715
Input:1 1    Output:0.059592    Expected Output: 0    Loss Function MSE: 0.001493
Train Time : 9994
Input:1 0    Output:0.945361    Expected Output: 1    Loss Function MSE: 0.001776
Input:0 0    Output:0.058563    Expected Output: 0    Loss Function MSE: 0.001493
Input:0 1    Output:0.945360    Expected Output: 1    Loss Function MSE: 0.001715
Input:1 1    Output:0.059584    Expected Output: 0    Loss Function MSE: 0.001493
Train Time : 9995
Input:1 0    Output:0.945367    Expected Output: 1    Loss Function MSE: 0.001775
Input:1 1    Output:0.059535    Expected Output: 0    Loss Function MSE: 0.001492
Input:0 1    Output:0.945319    Expected Output: 1    Loss Function MSE: 0.001772
Input:0 0    Output:0.058555    Expected Output: 0    Loss Function MSE: 0.001495
Train Time : 9996
Input:1 1    Output:0.059467    Expected Output: 0    Loss Function MSE: 0.001714
Input:1 0    Output:0.945305    Expected Output: 1    Loss Function MSE: 0.001768
Input:0 0    Output:0.058531    Expected Output: 0    Loss Function MSE: 0.001496
Input:0 1    Output:0.945305    Expected Output: 1    Loss Function MSE: 0.001713
Train Time : 9997
Input:1 1    Output:0.059460    Expected Output: 0    Loss Function MSE: 0.001496
Input:1 0    Output:0.945312    Expected Output: 1    Loss Function MSE: 0.001768
Input:0 1    Output:0.945331    Expected Output: 1    Loss Function MSE: 0.001495
Input:0 0    Output:0.058544    Expected Output: 0    Loss Function MSE: 0.001494
Train Time : 9998
Input:0 0    Output:0.058521    Expected Output: 0    Loss Function MSE: 0.001714
Input:1 1    Output:0.059434    Expected Output: 0    Loss Function MSE: 0.001712
Input:1 0    Output:0.945297    Expected Output: 1    Loss Function MSE: 0.001766
Input:0 1    Output:0.945317    Expected Output: 1    Loss Function MSE: 0.001496
Train Time : 9999
Input:1 1    Output:0.059447    Expected Output: 0    Loss Function MSE: 0.001495
Input:0 1    Output:0.945303    Expected Output: 1    Loss Function MSE: 0.001767
Input:1 0    Output:0.945365    Expected Output: 1    Loss Function MSE: 0.001496
Input:0 0    Output:0.058533    Expected Output: 0    Loss Function MSE: 0.001492
Final Hidden Weights
[ [ 3.672178 3.670992  ][ 5.874518 5.867958  ] ]
Final Hidden Biases
[ -5.610872 -2.422974  ]
Final Output Weights[ -8.068997 7.451116 ]
Final Output Biases
[ -3.355662 ]

 **** Please input the two bits number Separately for predict XOR Logic (ex:00,01,10,11) : ****
 First bit : 1
 Second bit : 1
 output result : 0 !!!


tsai@LAPTOP-HJB975Q4:~/projects/helloworld$
```

3. 分析

- 神經元架構分析

採用前饋神經網絡解決 XOR 問題，通過引入其他邏輯運算與 XOR 進行對比分析，設計適合解決 XOR 運算的網絡結構模型。在邏輯運算中我們可以找到一條直線對它們進行準確的分類，屬於線性可分。然而在 XOR 問題中，我們無法找到一條直線將其進行準確的分類，XOR 屬於一種線性不可分。

由於單層 NN 只能解決線性問題，無法解決非線性問題。要解決 XOR 運算問題，需要非線性的邊界。因此，使用多層網絡進行求解，在單層前饋 NN 的基礎上，加入一層隱含層，即二層的前饋神經網絡進行 XOR 的運算。我設計了一個二層的前饋神經網絡對 XOR 運算進行求解。網絡結構由一個輸入層、一個隱含層和一個輸出層構成，其中輸入層有兩個神經元，隱含層有兩個神經元，輸出層有一個神經元構成。



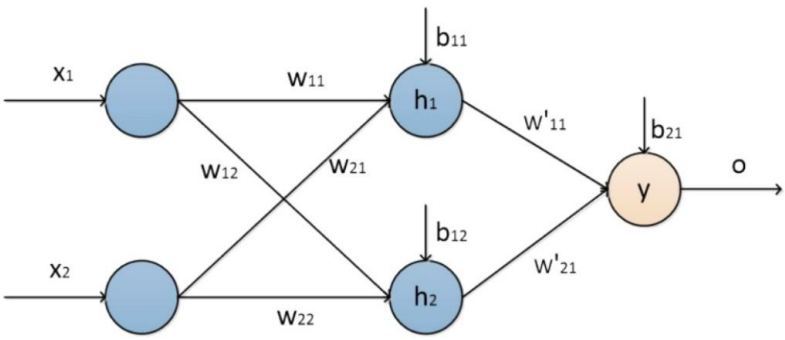

- HiddenLayer 表示式由(1)(2)表示

$h_1 = W_{11}*X_1+W_{21}*X_2+b_{11}$ ---- (1)

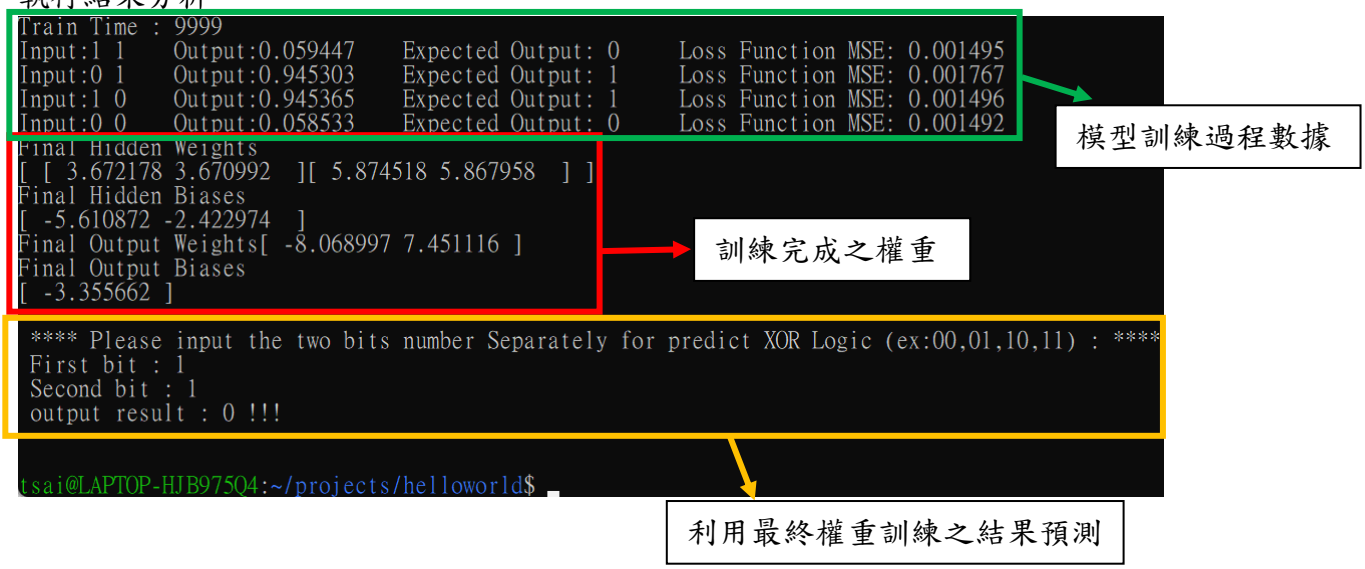
$h_2 = W_{12}*X_1+W_{22}*X_2+b_{12}$ ---- (2)

- OutputLayer 表示式由(3)表示

$y = W'_{11}*X_1+W'_{21}*X_2+b_{11}$ ---- (3)



➢ 執行結果分析




➢ 總結

由模擬結果可知，HiddenLayer 的作用，HiddenLayer 不直接接受外界的信號，也不直接向外界發送信號，通過對輸入的值進行加權的處理，將其轉化為更能被 OutputLayer 接受的型式，加入 HiddenLayer 可以提高神經網絡的非線性處理能力。

➢ Loss Function 分析

因為執行結果之視窗結果眾多，無法一一截圖於上，但可由 Loss Function 中發現，當訓練次數較少時，其 Loss Function 的數值較小，代表其預測的結果和實際之結果的誤差較大，當訓練次數較多時，可發現 Loss Function 漸漸的向 0 收斂，最後會在大約 0.0014-0.0017 之間抖動，其結果符合一般 NN 的 Loss Function 型態。