

電機3A/109501509/郭品萱 資料結構 作業一

1.編譯結果

```
root@LAPTOP-Q66SLPHM:/mnt/d/st/109501509_assignment_1/HW1# gcc -o main main.c -lm
```

2.執行結果

```
root@LAPTOP-Q66SLPHM:/mnt/d/st/109501509_assignment_1/HW1# gcc -o main main.c -lm
root@LAPTOP-Q66SLPHM:/mnt/d/st/109501509_assignment_1/HW1# ./main
Final Hidden Weights
[ [ 3.672099 3.670897 ] [ 5.874482 5.867946 ] ]
Final Hidden Biases
[ -5.610720 -2.422902 ]
Final Output Weights[ -8.068704 7.450911 ]
Final Output Biases
[ -3.355583 ]
Please enter the four bits in order, enter 99 at the first bit to exit
Input the first bit: 0
Input the second bit: 0
output : 0

Please enter the four bits in order, enter 99 at the first bit to exit
Input the first bit: 0
Input the second bit: 1
output : 1

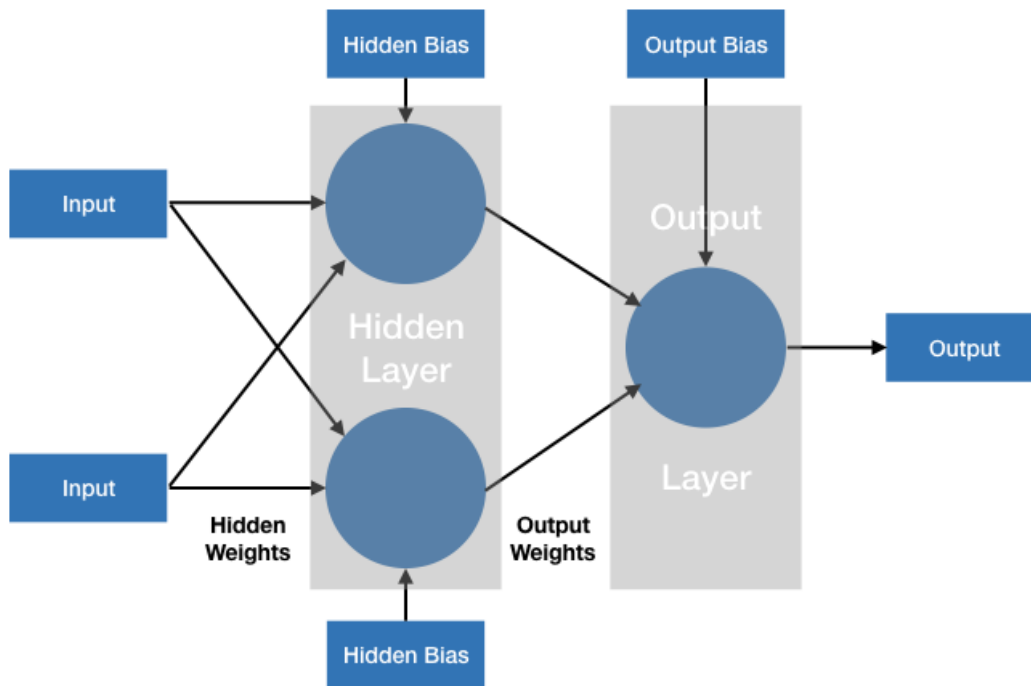
Please enter the four bits in order, enter 99 at the first bit to exit
Input the first bit: 1
Input the second bit: 0
output : 1

Please enter the four bits in order, enter 99 at the first bit to exit
Input the first bit: 1
Input the second bit: 1
output : 0
```

3.分析

在這次的作業中，我所參考的程式架構來自網站:

<https://towardsdatascience.com/simple-neural-network-implementation-in-c-663f51447547>



根據上圖，我採用了總共四層的神經網路，第一層是2bit的輸入，第二層是含兩個node的hidden layer，第三層是含一個node的output layer，第四層是一個1bit的輸出。從input到hidden layer會先經過一次權重的計算，程式碼如下：

```
neuron_temp1 = first_bit * (*(hiddenWeights+0)+0)) + second_bit * (*(hiddenWeights+1)+0)) + *(hiddenLayerBias+0);
neuron_temp2 = first_bit * (*(hiddenWeights+0)+1)) + second_bit * (*(hiddenWeights+1)+1)) + *(hiddenLayerBias+1);
```

由hidden layer 到 output layer也會再經歷一次相似的計算，程式碼如下：

```
outputLayer_temp = (sigmoid(neuron_temp1) * (*(outputWeights+0)+0))) + (sigmoid(neuron_temp2) *
(*(outputWeights+1)+0))) + *(outputLayerBias+0);
```

最後，將output layer得到的值歸一化就是我們要的output

由以上的程式碼可以發現，將第一個input和第一個權重的乘積加上第二個input和第三個權重的乘積，最後再和Bias相加是layer和layer之間計算權重的公式。

4.遇到的困難

在這次的作業中，我採用的hidden layer只有一層，也因此，在嘗試將輸入改成4bits、hidden nodes改成四個時遇到了難題。當輸入為1 1 1 1這樣的邊界值時，學習結果會停留在1，如果將訓練資料量調大到了100000，或是將學習率降低到0.01，便能使輸出變為1，然而，在增加訓練資料量以及調低的學習率的情況下，當輸入為0111時，學習結果會再次發生問題。我在嘗試了不同學習率以及資料筆數的搭配後，依然無法地到100%正確的結果。我認為，增加hidden layer的數量或許可以優化這個問題，日後我將找機會繼續嘗試。