

Assignment #1

A. 編譯結果

```
PS C:\Users\samul\OneDrive\桌面\資料結構\作業\作業一\108501569_assignment_1> gcc -o 108501569_assignment_1_code 108501569_assignment_1_code.cpp
PS C:\Users\samul\OneDrive\桌面\資料結構\作業\作業一\108501569_assignment_1>
```

圖一 編譯結果(使用 gcc 編譯)

B. 執行結果

```
PS C:\Users\samul\OneDrive\桌面\資料結構\作業\作業一\108501569_assignment_1> .\108501569_assignment_1_code
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	JUPYTER
	Input:1.000000 0.000000	Output: 0.947864	Expected Output: 1.000000	loss: 0.052136
	Input:0.000000 1.000000	Output: 0.948046	Expected Output: 1.000000	loss: 0.051954
	Input:0.000000 0.000000	Output: 0.056742	Expected Output: 0.000000	loss: 0.056742
	Input:1.000000 0.000000	Output: 0.947926	Expected Output: 1.000000	loss: 0.052074
	Input:1.000000 1.000000	Output: 0.056166	Expected Output: 0.000000	loss: 0.056166
	Final Hidden Weights			
	[[3.701660 3.710399] [5.834992 5.879662]]			
	Final Hidden Biases			
	[-5.675879 -2.452836]			
	Final Output Weights[-8.199578 7.528274]			
	Final Output Biases			
	[-3.379701]			
	input1:	1010101		
	input2:	0010001		
	input1 xor input2:1000100			

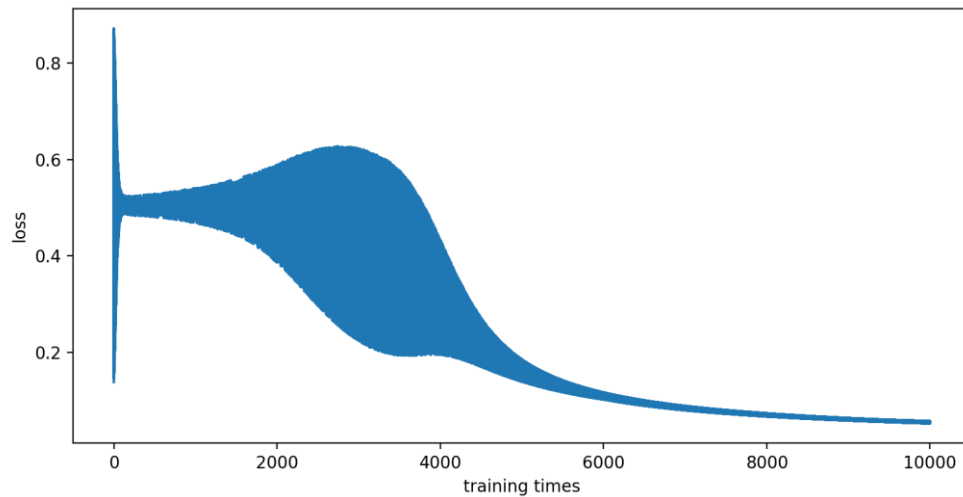
圖二 執行結果(使用 vscode 的 terminal)

由執行結果可見，將 input 值輸入訓練好的模型後，輸出值確實是這兩個 input 經過 xor 運算後的結果。

C. 分析

本次作業使用的 loss function 為「用 weight 和 bias 計算出的 output」以及「output 預期值」相減後取絕對值，如下式：

$$\text{loss} = |\text{output}_{\text{actual}} - \text{output}_{\text{expect}}|$$



圖三 loss 趨勢圖

上圖為由訓練中取得的 loss 繪製而成的趨勢圖(縱軸為 loss，橫軸為訓練次數)。由上圖可知，隨著訓練次數增加，loss 值漸漸地收斂並逐漸趨近於 0，可見此神經網路的訓練效果頗佳。

註：因為程式中每一次訓練都使用多組 input、進而產生多組 output(同時產生多個 loss 值)，所以導致 loss 趨勢圖中每一個橫軸值會對應到多個縱軸值，但是此趨勢圖表現出的趨勢仍是我們所要求的。