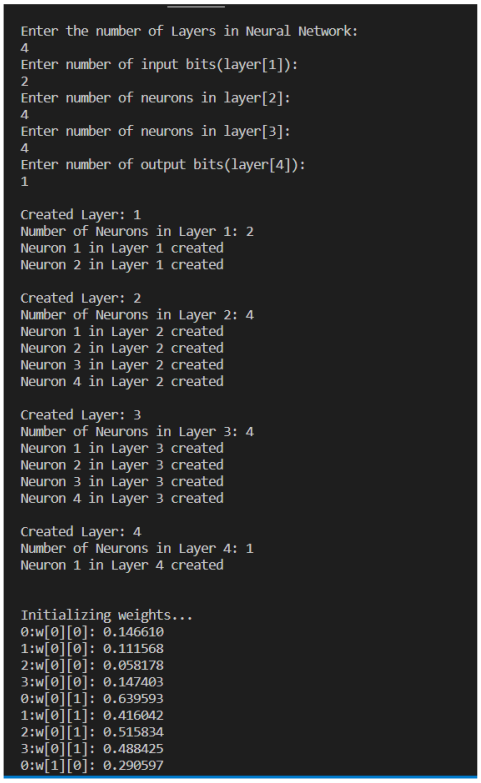
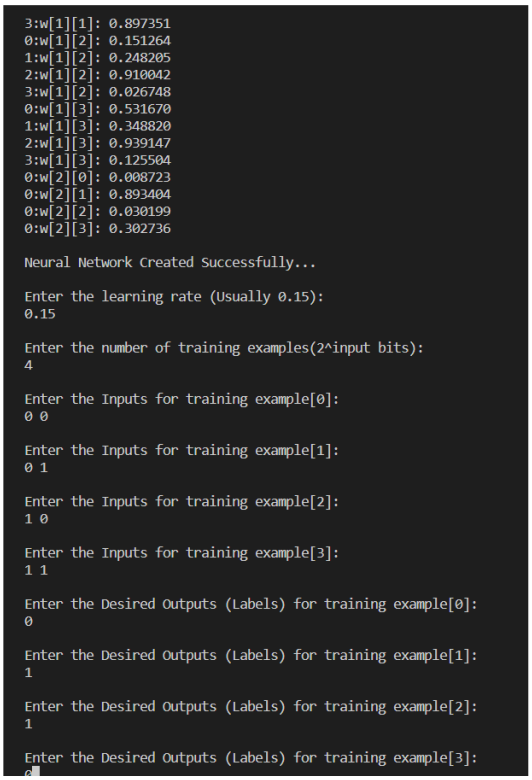
NN Final Report

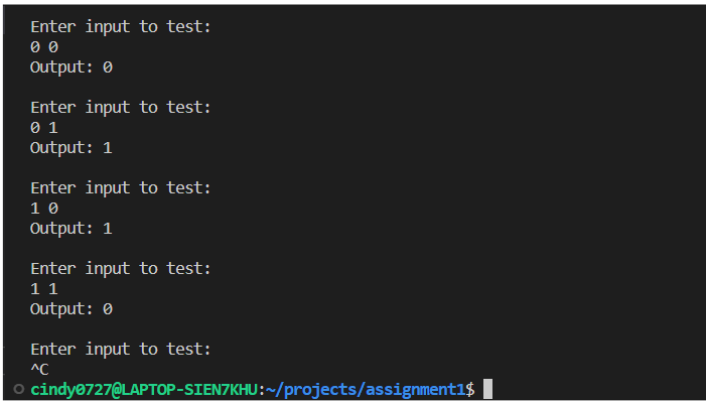
110503513 通訊二王家欣

此部分有執行結果、遇到的問題、待改善事項和心得。

一、執行結果

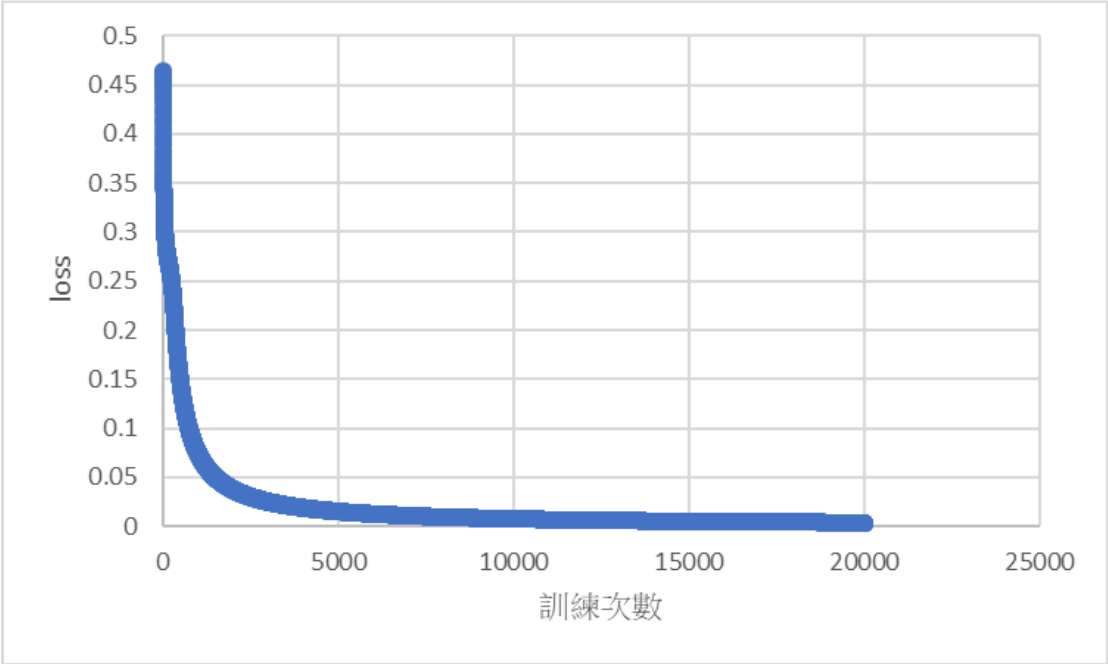






二、分析

1. 使用excel以訓練次數為橫軸，loss為縱軸作圖。



由此圖可知，訓練的次數越多，loss 的值會越來越小，代表更接近 desired output。這裡的 loss 計算方式是使用 MSE(均方誤差)。

2. 訓練次數多寡造成output的差別

訓練 50 次

```
#define trainingtimes 50

layer *lay = NULL;
int num_layers;
int *num_neurons;
float alpha;
float *cost;
float full_cost;
float **input;
float **desired_outputs;
int num_training_ex;
```

```
Input: 1.000000
Output: 1

Enter input to test:
0 0
Output: 1

Enter input to test:
1 0
Output: 0

Enter input to test:
0 1
Output: 0

Enter input to test:
1 1
Output: 0

Enter input to test:
```

訓練 500 次

```
#define trainingtimes 500

layer *lay = NULL;
int num_layers;
int *num_neurons;
float alpha;
float *cost;
float full_cost;
float **input;
float **desired_outputs;
int num_training_ex;
```

```
Input: 1.000000
Output: 1

Enter input to test:
0 0
Output: 1

Enter input to test:
0 1
Output: 1

Enter input to test:
1 0
Output: 1

Enter input to test:
1 1
Output: 1

Enter input to test:
```

訓練 5000 次

```
#define trainingtimes 5000

layer *lay = NULL;
int num_layers;
int *num_neurons;
float alpha;
float *cost;
float full_cost;
float **input;
float **desired_outputs;
int num_training_ex;
```

```
Input: 1.000000
Output: 0

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

Enter input to test:
```

由這三張圖可知，output 次數被修改得越少，也就是訓練的次數越少，得到的 output 值離 desired output 值越遠，造成和正確數值的誤差。

三、遇到的問題

- 問題：第一次遇到malloc、memset等使用動態記憶體的函式，不清楚動態記憶體的運作規則。

解決方法：上網查他人寫的範例(參考資料2)，用malloc和陣列對比的簡單程式理解malloc、memset的規則，再去看參考資料1中提供xor的code，透過別人的範例，和將code中的pointer和malloc改成陣列形式思考，更加理解參考資料1使用malloc配置多大的記憶體和memset的用意。

- 問題：將資料轉移至csv檔時會缺少資料。

解決方法：原本是將關檔寫在 test_nn(測試數值是否符合 desire output)後，然而，平常測試完成後直接按 ctrl+c(強制終止)導致一直無法執行到關檔的程式，所以要將關檔放在測試前，這件事告訴我，**原來沒關檔會出錯！**

心得：

這次的作業是我第一次用別人的程式碼來完成作業，也是第一次使用 linux。以前使用 visual studio，程式碼編譯和執行只要一個按鍵就能解決，這次才真正了解到原來編譯跟執行是兩個不同的步驟。看別人的程式碼時，因為還不太會用工具輔助，只能一直滑滑鼠對照哪些東西在哪裡出現過，經過助教課，下次會更懂得用 gdb 來輔助。最後，雖然這次沒有動手寫完整個程式，卻也意外看懂，理解這次功課的運作內容，因為參考資料 1 的 code 都是使用 pointer 和 malloc，經過這次的作業讓我對指位器跟需要多少記憶體來儲存有更多的了解。