

Assignment#1

1. 編譯結果

```
ivy@ivy-VirtualBox:~/Downloads/109208001_assignment_1/Linux$ gcc -o nncc nncc.c func.c -lm
ivy@ivy-VirtualBox:~/Downloads/109208001_assignment_1/Linux$
```

2. 執行結果

```
ivy@ivy-VirtualBox:~/Downloads/109208001_assignment_1/Linux$ gcc -o nncc nncc.c func.c -lm
ivy@ivy-VirtualBox:~/Downloads/109208001_assignment_1/Linux$ ./nncc
training data:
0,0,0,
0,0,0,
0,0,0,
0,1,0,
0,1,0,
target data:
0,0,
0,0,
0,0,
0,1,
0,1,
Epoch 0 : Error = 0.525897
Epoch 100 : Error = 0.002885
Epoch 200 : Error = 0.001125
Epoch 300 : Error = 0.000690
Epoch 400 : Error = 0.000494

NETWORK DATA - EPOCH 483

Pat   Input1 Input2 Target1 Output1
1     0     0     0     0.011936
2     0     0     0     0.011928
3     1     0     1     0.983966
4     1     0     1     0.983957

It's your turn!
Please enter 8 numbers composed of 0 or 1(tab needed)
Sorry! I can't do the testing data now!

nncc: malloc.c:2617: sysmalloc: Assertion `(old_top == initial_top (av) && old_size == 0) || ((unsigned long) (old_size) >= MIN_SIZE && prev_inuse (old_top) && ((unsigned long) old_end & (pagesize - 1)) == 0)' failed.
Aborted (core dumped)
ivy@ivy-VirtualBox:~/Downloads/109208001_assignment_1/Linux$
```

3. 分析

＊ ＊ 如果要看 epoch 遞減的速率分析圖以及 weighting 如何計算的話，我在第一次交作業時有分析過了，這邊不再做分析。

＊ ＊ 主要可以看出，用指標定義的矩陣 epoch 重複 483 次就把 weighting 誤差降到最小了，而上次用矩陣做的 epoch 則需要幾千次。

＊ ＊ 最後，可以看到 Output1（用最終產生的 weighting 求出來的 Output）與 Target1 的值很相近。分別相差（1.194%、1.193%、1.603%、1.604%）

4. 遇到的問題

(a) 這次要交的是動態記憶體版本的 nn，我不能再像上次一樣，直接定義 target data 矩陣，和 training data 矩陣，必須要一格一格給數值。因此，我選用 random 產生隨機的 0,1 training 矩陣，若兩筆 training data 相加為 1，則 target data 為 1（出現奇數個 1 輸出 1）；若 training data 相加為 0 或 2（出現偶數個 1 輸出 0，xor 定義），則 target data 為 0。

(b) 然後就是我的測試資料 testing data 的部分，相比第一次的 code，這次的 code 沒有辦法讓我輸入 testing data，我上網查了可能的原因，我猜應該是我重複定義某一部分函數，但是此函數的初始值還沒有 free 掉，所以兩筆數據會重合，進而 error。因為我實在找不到是哪裡有這種 bug 所以我就先繳交了。我有問莊于鋐助教，他說因為 training data set 和 testing data set 會一樣，所以可以不用加入 testing data。