Data Structure Assignment 1
利用"類神經網路"學習XOR的運算Guide

通訊二 110503518 李秉宸

December 29, 2022

1 System architecture

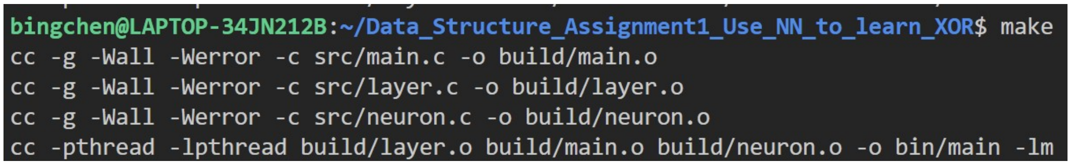
Compiler：Visual Studio Code
Version: 1.74.0
OS: Windows_NT x64 10.0.22000

Building environment：Windows 10 Linux Subsystem(WSL2)
Release: Ubuntu 20.04 LTS
Kernel: Linus 5.10.16.3-microsoft-standard-WSL2

Extensions
gcc version：9.3.0
gdb version：9.2
git version：2.25.1

2 Build Guide

In this assignment, please enters **”make”** to compile the code and generate the files.

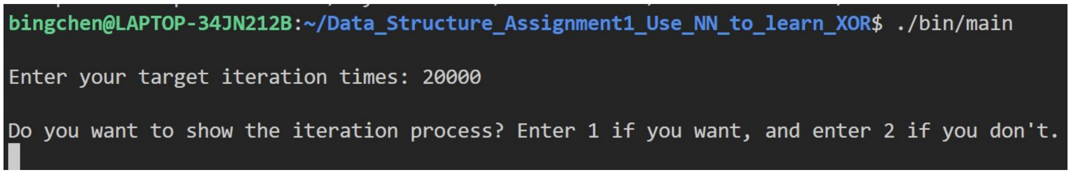


```
bingchen@LAPTOP-34JN212B:~/Data_Structure_Assignment1_Use_NN_to_learn_XOR$ make
cc -g -Wall -Werror -c src/main.c -o build/main.o
cc -g -Wall -Werror -c src/layer.c -o build/layer.o
cc -g -Wall -Werror -c src/neuron.c -o build/neuron.o
cc -pthread -lpthread build/layer.o build/main.o build/neuron.o -o bin/main -lm
```

Figure 1: Makefile

3 Execute Guide

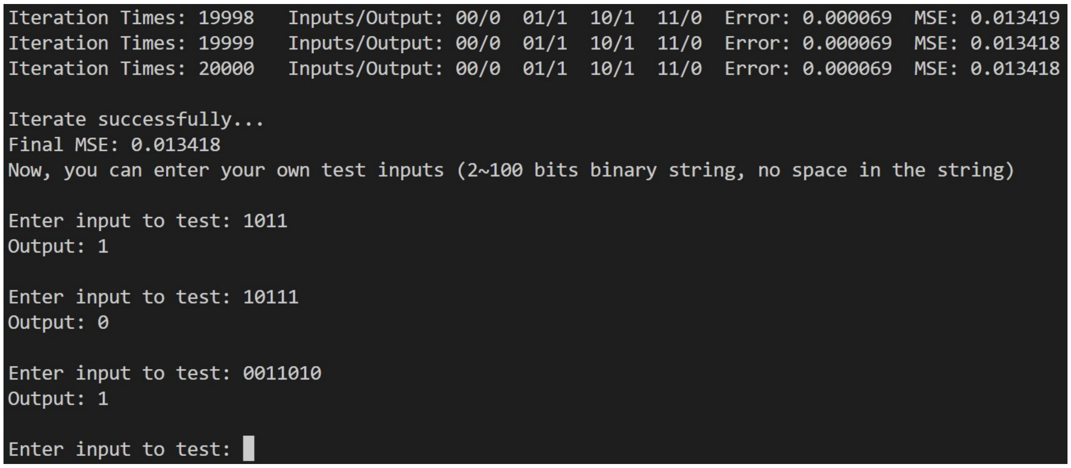
Please enter **”./bin/main”** to run the file. Figure 2 to Figure 4 are the results of run file. When the users enter the ”iteration times”, the program will ask you to enter 1 or 2 to decide if you want to show the iteration process or not.



```
bingchen@LAPTOP-34JN212B:~/Data_Structure_Assignment1_Use_NN_to_learn_XOR$ ./bin/main

Enter your target iteration times: 20000

Do you want to show the iteration process? Enter 1 if you want, and enter 2 if you don't.
```

Figure 2: Run file 1-1: Show the iteration process

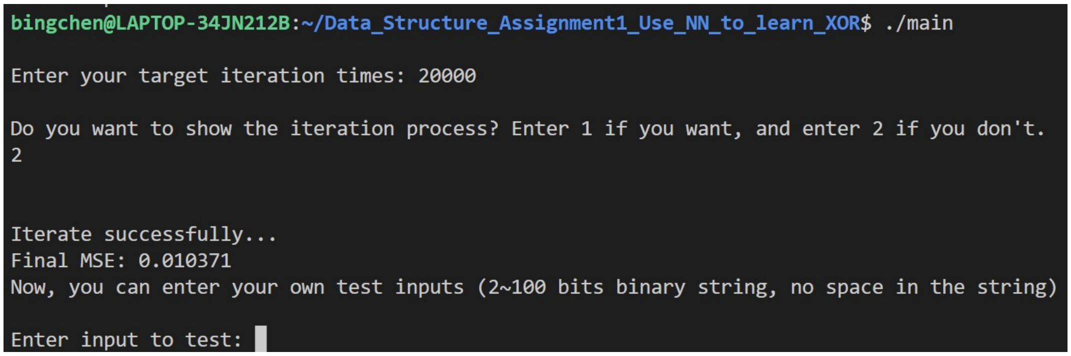
If the user enter 1, the program will show the data just like Figure 3 (including iteration times, the training inputs and output results, sum of the square error and mean square error).



```
Iteration Times: 19998   Inputs/Output: 00/0  01/1  10/1  11/0  Error: 0.000069  MSE: 0.013419
Iteration Times: 19999   Inputs/Output: 00/0  01/1  10/1  11/0  Error: 0.000069  MSE: 0.013418
Iteration Times: 20000   Inputs/Output: 00/0  01/1  10/1  11/0  Error: 0.000069  MSE: 0.013418

Iterate successfully...
Final MSE: 0.013418
Now, you can enter your own test inputs (2~100 bits binary string, no space in the string)

Enter input to test: 1011
Output: 1

Enter input to test: 10111
Output: 0

Enter input to test: 0011010
Output: 1

Enter input to test:
```

Figure 3: Run file 1-2: Show the iteration process

If the user enter 2, it will show the data like Figure 4.

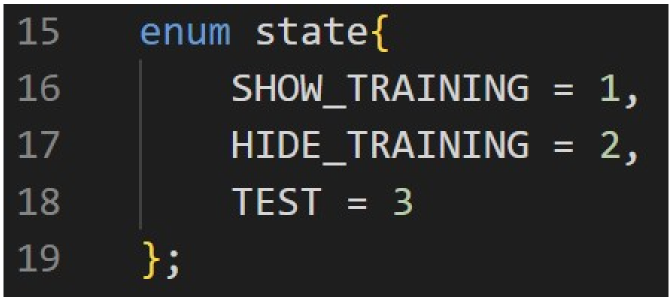


```
bingchen@LAPTOP-34JN212B:~/Data_Structure_Assignment1_Use_NN_to_learn_XOR$ ./main

Enter your target iteration times: 20000

Do you want to show the iteration process? Enter 1 if you want, and enter 2 if you don't.
2


Iterate successfully...
Final MSE: 0.010371
Now, you can enter your own test inputs (2~100 bits binary string, no space in the string)

Enter input to test:
```

Figure 4: Run file 2: Hide the iteration process

Otherwise, I set a enum called **”state”** to let the users decide the display mode by themselves. When the user **enter 1**, the program will set the state into **”SHOW_TRAINING”**, and into **”HIDE_TRAINING”** when the user **enter 2**.

After training, the program will set the state into **”TEST”** automatically, and let the user enter the inputs they want in a type of binary string between 2 to 100 bits.



```
enum state{
    SHOW_TRAINING = 1,
    HIDE_TRAINING = 2,
    TEST = 3
};
```

Figure 5: The awy to show or hide the iteration process

4 Reference 參考資料

[1] GitHub：Neural-Network-framework-using-Backpropogation-in-C
(https://github.com/mayurbhole/Neural-Network-framework-using-Backpropogation-in-C.git)