1.編譯結果

```
opendds@opendds-VirtualBox:~$ cd ~/Desktop/Project1
opendds@opendds-VirtualBox:~/Desktop/Project1$ gcc -o main hw1.c genann.c genann
.h -lm
```

2.執行結果

```
opendds@opendds-VirtualBox:~/Desktop/Project1$ ./main
Train a small ANN to the XOR function using backpropagation.
Please input the number of bit:4
Please input how many times you want to train:2000
Please input how many problem you want to test(the problem will generate automa
tically):4
Please input how many hidden layers:1
Please input how many neurons:2
Output for [1, 1, 0, 0, ] is 0.
Output for [1, 0, 1, 1, ] is 0.
Output for [1, 0, 0, 1, ] is 0.
Output for [0, 0, 0, 0, ] is 0.
sh: 1: pause: not found
```

3.分析

首先，我將 bit 長度設為 4，hidden layer 設為 1，神經元數設為 2，問題數設為 4，根據不同的訓練次數作分析。

```
opendds@opendds-VirtualBox:~/Desktop/Project1$ ./main
Train a small ANN to the XOR function using backpropagation.
Please input the number of bit:4
Please input how many times you want to train:500
Please input how many problem you want to test(the problem will generate automa
tically):4
Please input how many hidden layers:1
Please input how many neurons:2
Output for [0, 0, 1, 1, ] is 0.
Output for [1, 1, 0, 1, ] is 0.
Output for [1, 0, 1, 0, ] is 0.
Output for [1, 1, 0, 0, ] is 0.
```

```
opendds@opendds-VirtualBox:~/Desktop/Project1$ ./main
Train a small ANN to the XOR function using backpropagation.
Please input the number of bit:4
Please input how many times you want to train:1000
Please input how many problem you want to test(the problem will generate automa
tically):4
Please input how many hidden layers:1
Please input how many neurons:2
Output for [0, 1, 1, 0, ] is 0.
Output for [1, 0, 0, 0, ] is 0.
Output for [0, 1, 0, 0, ] is 0.
Output for [0, 0, 1, 1, ] is 0.
sh: 1: pause: not found
opendds@opendds-VirtualBox:~/Desktop/Project1$
```

```
opendds@opendds-VirtualBox:~/Desktop/Project1$ ./main
Train a small ANN to the XOR function using backpropagation.
Please input the number of bit:4
Please input how many times you want to train:1500
Please input how many problem you want to test(the problem will generate automa
tically):4
Please input how many hidden layers:1
Please input how many neurons:2
Output for [1, 1, 0, 1, ] is 1.
Output for [0, 0, 0, 1, ] is 1.
Output for [0, 1, 1, 0, ] is 0.
Output for [0, 1, 0, 1, ] is 0.
sh: 1: pause: not found
opendds@opendds-VirtualBox:~/Desktop/Project1$
```

可以發現，在 1000 次的時候，有最高的準確率，再訓練更多次之後，準確率有所下降，這是多次實測之後得出的結果。猜測原因是在超過 1000 次之後，程式出現了過度擬合的情況。畢竟在 4 bits 的情況下，有 16 組數據，而程式又會先將訓練的資料中，將題目挑除(避免出現”背答案”的狀況)，在有 4 題的情況下，等於只有 12 種組合進行訓練，在大量重複訓練之後，出現過度擬合的情況算是滿合理的。