

Assignment #1

1. 編譯結果

```
~/visual studio/c++/datastructurehw/HW1$ g++ main.c layer.c neuron.c -o main
~/visual studio/c++/datastructurehw/HW1$
```

2. 執行結果

```
wch@wch-virtual-machine:~/visual studio/c++/datastructurehw/HW1$ ./main
Enter the number of Layers in Neural Network:
4
Enter number of neurons in layer[1]:
2
Enter number of neurons in layer[2]:
4
Enter number of neurons in layer[3]:
4
Enter number of neurons in layer[4]:
1
```

```
Created Layer: 1
Number of Neurons in Layer 1: 2
Neuron 1 in Layer 1 created
Neuron 2 in Layer 1 created
```

```
Created Layer: 2
Number of Neurons in Layer 2: 4
Neuron 1 in Layer 2 created
Neuron 2 in Layer 2 created
Neuron 3 in Layer 2 created
Neuron 4 in Layer 2 created
```

```
Created Layer: 3
Number of Neurons in Layer 3: 4
Neuron 1 in Layer 3 created
Neuron 2 in Layer 3 created
Neuron 3 in Layer 3 created
Neuron 4 in Layer 3 created
```

```
Created Layer: 4
Number of Neurons in Layer 4: 1
Neuron 1 in Layer 4 created
```

Initializing weights...

```
0:w[0][0]: 0.075140
1:w[0][0]: 0.486141
2:w[0][0]: 0.884873
3:w[0][0]: 0.146346
0:w[0][1]: 0.655257
1:w[0][1]: 0.124192
2:w[0][1]: 0.471281
3:w[0][1]: 0.847435
0:w[1][0]: 0.654386
1:w[1][0]: 0.280163
2:w[1][0]: 0.568565
3:w[1][0]: 0.680024
0:w[1][1]: 0.611201
1:w[1][1]: 0.563687
2:w[1][1]: 0.440676
3:w[1][1]: 0.615470
0:w[1][2]: 0.969583
1:w[1][2]: 0.541368
2:w[1][2]: 0.152239
3:w[1][2]: 0.812727
0:w[1][3]: 0.057567
1:w[1][3]: 0.761173
2:w[1][3]: 0.175777
3:w[1][3]: 0.141472
0:w[2][0]: 0.411955
0:w[2][1]: 0.987029
0:w[2][2]: 0.677382
0:w[2][3]: 0.633440
```

Neural Network Created Successfully...

```

Enter the learning rate (Usually 0.15):
0.15

Enter the number of training examples:
4

Enter the Inputs for training example[0]:
0 0

Enter the Inputs for training example[1]:
0 1

Enter the Inputs for training example[2]:
1 0

Enter the Inputs for training example[3]:
1 1

Enter the Desired Outputs (Labels) for training example[0]:
0

Enter the Desired Outputs (Labels) for training example[1]:
1

Enter the Desired Outputs (Labels) for training example[2]:
1

Enter the Desired Outputs (Labels) for training example[3]:
0

```

```

Enter input to test:

```

3. 分析

在測驗模型的部分，我分成了兩個部份來測試。首先是在 00~11 的範圍內，測試輸入不同數量及組合的 training data，在訓練之後這個系統的準確度及輸出結果。可以從圖表中看到，以將所有數值都當成 training data 輸入為對質找組，輸入不完全的 training data 則會讓準確率大幅下降。而剩下為當成 training data 的數值則當作 testing data，系統的輸出值則會根據 training data 的內容變化為全部都輸出 1 或是 0。推測是因為 training data 數量太少而早成如此極端的結果。

	2bits				
layer	4				
number of neurons in layer1	2				
number of neurons in layer2	4				
number of neurons in layer3	4				
number of neurons in layer4	1				
learning rate	0.15				
Gradient Descent each training data	20000				
all data number	4	4	4	4	4
training data number	4	2	2	2	3
accuracy	100%	75%	50%	50%	75%
training data		0 0	0 0	0 1	0 0
		0 1	1 1	1 0	0 1
					1 0
teating data ouput		all one	all zero	all one	all one

而第二個部分則是在 0000 ~ 1111 的範圍內做測試。因為在第一次輸入時發現準確率並非是 100%，因此透過改變迭代次數來確定準確性。在調整迭代次數後發現，隨著迭代次數的增加，整體準確率會呈現震盪但往 100% 的地方靠近。預期未來持續增加迭代次數將可以增加系統準確率。

	4bits									
layer	4									
number of neurons in layer1	4									
number of neurons in layer2	4									
number of neurons in layer3	4									
number of neurons in layer4	1									
learning rate	0.15									
Gradient Descent each training data	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
all data number	16									
training data number	16									
accuracy	94%	50%	94%	81.25%	94%	88%	100%	81%	100%	88%

