

HW2

B083040045 李亦晴

>>design ideas

>neural network

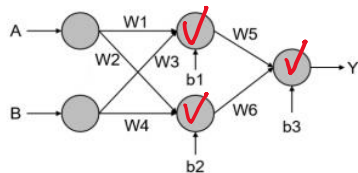


Fig.1 (b)

左圖為 XOR 在 neural network 的架構，由三個 Neuron 構成。此次作業大架構是先各別做出 sub-module Neuron，再在 module neural network 中為 input 作權重處理，並且重複呼叫 sub-module 計算結果。

	W1	W2	W3	W4	W5	W6	b1	b2	b3
value	10	-10	-10	10	10	10	-5	-5	-5

>Neuron(sub-module)

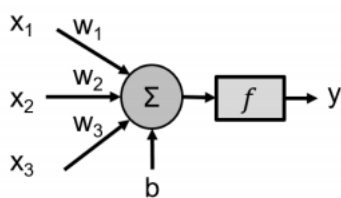


Fig.1 (a)

Fig.1(a)是 Neuron 的結構。先將已乘上權重的 input 和 b 做累加，並且將累加的結果作為輸入放入 Sigmoid 函數計算。

其中，Sigmoid 函數處理比較特殊，以下另外說明。

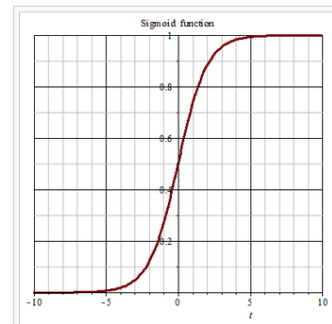
>Sigmoid 函數

一種常見的S函數是邏輯函數:

$$S(t) = \frac{1}{1 + e^{-t}}.$$

其級數展開為：

$$s := 1/2 + \frac{1}{4}t - \frac{1}{48}t^3 + \frac{1}{480}t^5 - \frac{17}{80640}t^7 + \frac{31}{1451520}t^9 - \frac{691}{319334400}t^{11} + O(t^{12})$$



以上是維基百科有關 Sigmoid 的說明，由圖可知其計算結果僅會在 0~1 之間但為方便計算，在 HW2 中 Sigmoid 函式簡化為:

$$\text{Sigmoid}(x) \approx \frac{1}{2} + \frac{1}{4}x - \frac{1}{48}x^3 + \frac{1}{480}x^5$$

但簡化後的計算結果並不僅落在 0~1 之間，而是會超出範圍，故在 Neuron 的 module 中，計算結果超過 1 的部分要手動歸為 1，而低於 0 的部分要歸為 0

另外有個想法，在寫功課初期有點疑惑 b 的功用，後來思考 b 設定為-5 應該和 Sigmoid 函數計算有關。由於 input 的權重分別可能是 10 及-10，所以累加的可能性為 10、0、-10，其中 0 代入 Sigmoid 函數中，結果為 0.5，並不

能符合 0 或 1 的結果，故而需要 b 將整體輸入向左挪 5，避開此種情況。

>>Code

依據上述想法，寫出下列程式：

>Neuron.h

```
SC_MODULE( Neuron ) {  
    public:  
        sc_in < float > input1, input2;  
        sc_out < float > output;  
        float t,i1,i2,count;  
        SC_HAS_PROCESS( Neuron );  
  
        // vvvvv put your code here vvvvv  
        Neuron( sc_module_name name ) :  
            sc_module( name )  
        {  
            SC_METHOD( neuron );  
            sensitive << input1 << input2;  
        }  
        // ^^^^^ put your code here ^^^^^  
  
    private :  
        void neuron();  
        float w1;  
        float w2;  
        float b;  
};  
#endif
```

t 為計算累加的暫存值
count 為 Sigmoid 計算結果，
其中 t 為 Sigmoid 的輸入值

當 input 改變時，即呼叫函式
void neuron()

> Neuron.cpp

```
void Neuron::neuron() {  
    // vvvvv put your code here vvvvv  
  
    i1=input1.read();  
    i2=input2.read();  
    t=i1+i2-5;  
    count=0.5+0.25*t-(1/48)*t*t*t+(1/480)*t*t*t*t*t;  
    if(count>=1)  
        output.write(1);  
    else if(count<=0)  
        output.write(0);  
    else  
        output.write(count);  
    // ^^^^^ put your code here ^^^^^  
}
```

手動處理 Sigmoid 計
算結果，使能符合輸
出 1/0

> neural_network.h

```
SC_MODULE( Neural_Network ) {

    sc_in < float > input1, input2;
    sc_out < float > output;
    sc_signal < float > i1,i2,w1,w2,w3,w4,w5,w6,o1,o2,o3;

    Neuron *N1,*N2,*N3;
    void appand()
    {
        i1=input1.read();
        i2=input2.read();
    }
    void run()
    {
        w1=i1*10;
        w2=i1*(-10);
        w3=i2*(-10);
        w4=i2*10;
        w5=o1*10;
        w6=o2*10;
    }
    void assign()
    {
        output.write(o3);
    }

    SC_CTOR( Neural_Network ) {

        // vvvvv put your code here vvvvv
        SC_METHOD(appand);
        sensitive<<input1<<input2;

        SC_METHOD(run);
        sensitive<<i1<<i2<<o1<<o2;

        SC_METHOD(assign);
        sensitive<<o3;

        N1=new Neuron("N1");
        (*N1)(w1,w3,o1);
        N2=new Neuron("N2");
        (*N2)(w2,w4,o2);
        N3=new Neuron("N3");
        (*N3)(w5,w6,o3);
        // ^^^^^ put your code here ^^^^^
    }
}
```

run 的功能
將輸入值依據題目
設定，乘上權重

將三個 Neuron(神
經元)連接起來

>>result in Moba

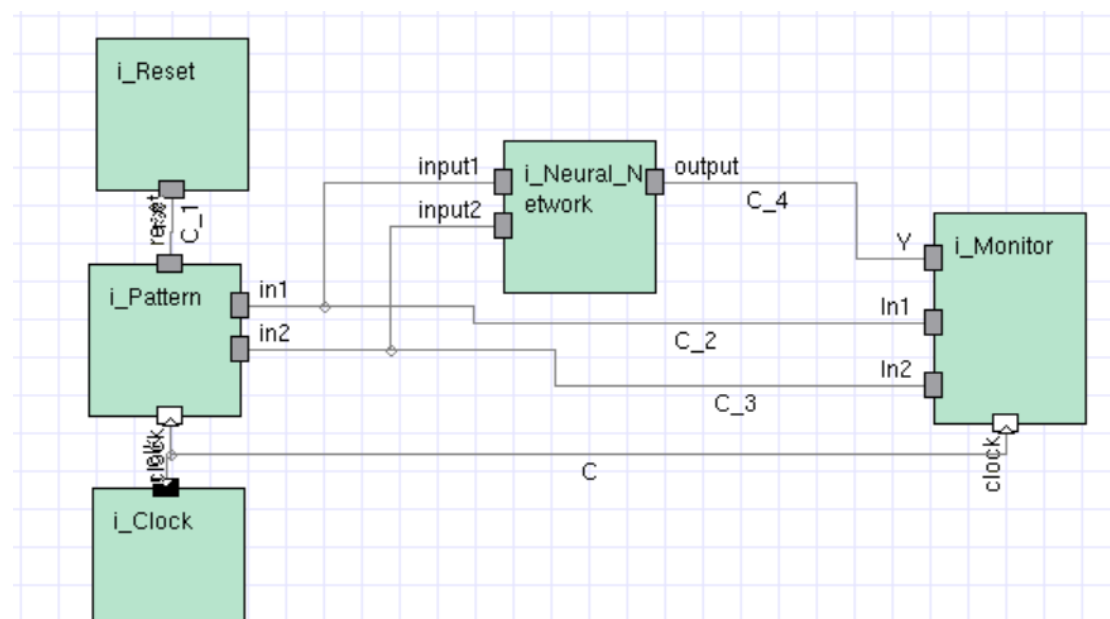
```
b083040045@ZEUS /home/b083040045/Desktop/HW2% make
g++ -I . -I /usr/systemc/include -L . -L /usr/systemc/lib-linux64 -o LAB *.cpp -lsystemc -lm -DSC_INCLUDE_FX
b083040045@ZEUS /home/b083040045/Desktop/HW2% LAB

SystemC 2.3.1-Accellera --- Jun 22 2017 17:44:29
Copyright (c) 1996-2014 by all Contributors,
ALL RIGHTS RESERVED

In1      In2      Y
1         1         0
0         0         0
1         1         0
0         1         1
0         1         1
0         0         1
1         0         1
0         0         0

Info: /OSCI/SystemC: Simulation stopped by user.
```

>>block diagram



>>simulation result of PA

In1	In2	Y
1	0	1
0	1	1
1	0	1
1	0	1
1	1	0
0	0	0
0	0	0

SystemC: simulation stopped by user.