

以下是解「[“Neural Network Design,” Hagan, Demuth, Beale, Thomson Learning, 1996](#)」書中 Chapter 4 的 P4_3，對照書中的空間作圖，可以對 Neural Network 有更進一步的了解。

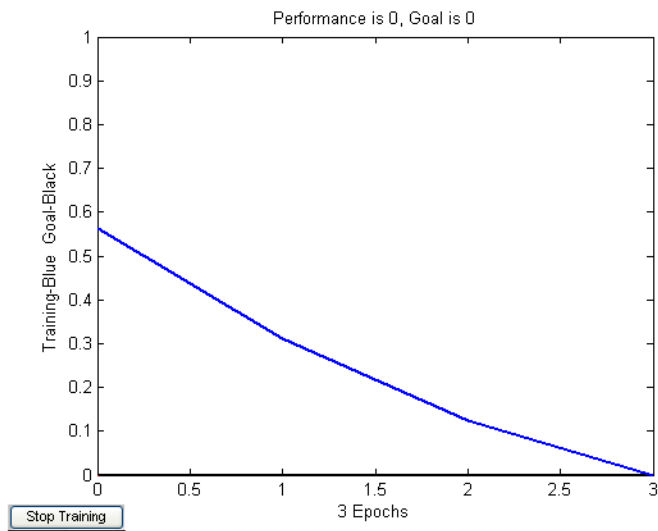
P4_3 We have a classification problem with four classes of input vector. The four classes are

$$\text{Class 1: } \left\{ p_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, p_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\}, \text{Class 2: } \left\{ p_3 = \begin{bmatrix} 2 \\ -1 \end{bmatrix}, p_4 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\}$$

$$\text{Class 3: } \left\{ p_5 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, p_6 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \right\}, \text{Class 4: } \left\{ p_7 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, p_8 = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right\}$$

Sol. Matlab 程式解如下：【2_input 1_layer 2_neuron 2_output Perceptrons）】

```
>> P=[1 1 2 2 -1 -2 -1 -2; 1 2 -1 0 2 1 -1 -2]
P = 1 1 2 2 -1 -2 -1 -2
    1 2 -1 0 2 1 -1 -2
>> T=[0 0 0 0 1 1 1 1; 0 0 1 1 0 0 1 1]
T = 0 0 0 0 1 1 1 1
    0 0 1 1 0 0 1 1
>> net=newp(minmax(P),2,'hardlim','learnp'); % minmax(P)=[ -2 2; -2 2]
>> net.IW{1,1}=randn(2); % 本例若不給初值(系統 default 為 0)，讀者可試試，一樣 OK! 不過分類時，
區域切割的不是挺好。人看起來，切的地方不在兩類區域的中間。因為當輸出值恰完全等於 desired value 時，
學習自然「見好就收」，不再更動學習結果的呀！
>> net.IW{1,1} % 我們給定的亂數初值
ans = -0.4326 0.1253
      -1.6656 0.2877
>> net.b{1}=randn(2,1);
>> net.b{1} % 我們給定的 bias 亂數初值
ans = -1.1465
      1.1909
>> a=sim(net,P) % 還沒開始學，顯示結果不正確
a = 0 0 0 0 0 0 0 0
    0 1 0 0 1 1 1 1
>> net=train(net,P,T);
TRAINC, Epoch 0/100
TRAINC, Epoch 3/100
TRAINC, Performance goal met.
```



```
>> net.IW{1,1}
```

```
ans = -3.4326 -0.8747
      0.3344 -3.7123
```

```
>> net.b{1}
```

```
ans = -0.1465
      1.1909
```

```
>> a=sim(net,P)
```

```
%學習結果正確
```

```
a = 0 0 0 0 1 1 1 1
     0 0 1 1 0 0 1 1
```

```
>> figure(2); plotpv(P,T); hold on; plotpc(net.IW{1},net.b{1});
```

