

```
%% Classification with a 2-input Perceptron
```

```
% 檔名：perceptron21.m 原檔案：C:\MATLAB7\toolbox\nnet\ndemos\demop1.m
```

```
% 讀者可在 Matlab 軟體 >>demo => Neural Network Perceptrons => Classification with a 2-input perceptron 中找到，Matlab 也將此寫成一個 GUI 程式，有投影片教學喔。
```

```
% 對於類神經網路 初學者，亦可在 Matlab 軟體中，執行 >>nntool ，可以進入一 GUI 畫面，直接進行學習，這部分可參考：
```

```
% Matlab 入門與進階，蒙以正 編著，儒林
```

```
% 【請在 Matlab 中執行 >> nnd2n1 ，其實這才是最簡單的類神經網路(1_input 1_layer 1_neuron 1_output)°】
```

```
% 【本例相當於 >> nnd2n2 】
```

```
% 南台科大 電機系 Chun-Tang Chao 趙春棠 解說
```

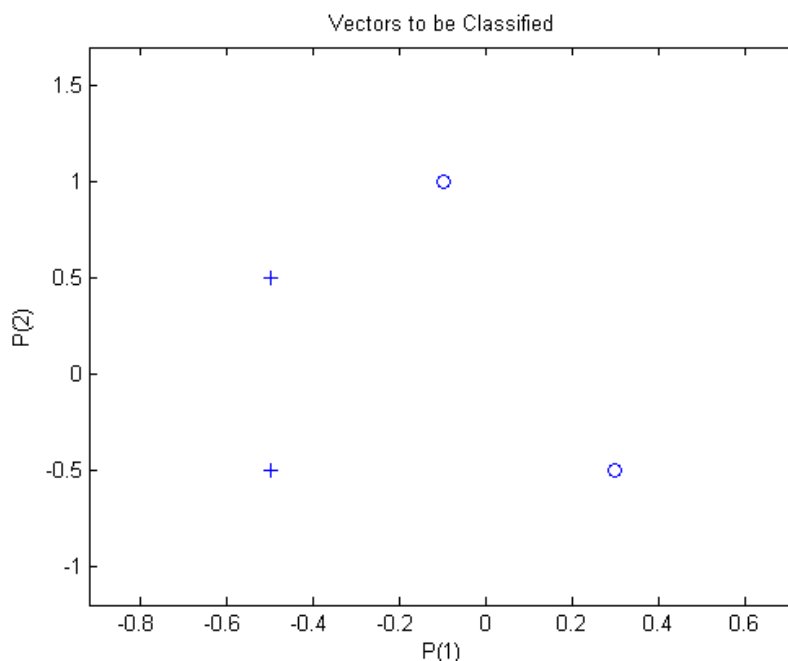
```
P = [-0.5 -0.5 +0.3 -0.1; ...  
      -0.5 +0.5 -0.5 +1.0];
```

```
T = [1 1 0 0];
```

```
% 意即(-0.5,-0.5)、(-0.5,+0.5) 這兩點，希望輸出為 1 (分類在同一區)
```

```
% 而 (+0.3,-0.5)、(-0.1,+1.0) 這兩點，希望輸出為 0 (分類在同一區)
```

```
plotpv(P,T); % PLOTPV Plot perceptron input/target vectors.
```



```
% 圖形輸出：
```

```
%%
```

```
% The perceptron must properly classify the 5 input vectors in P into the two
```

```
% categories defined by T. Perceptrons have HARDLIM neurons. These neurons are
```

```
% capable of separating an input space with a straight line into two categories
```

```
% (0 and 1).
```

```
%
```

```
% NEWP creates a network object and configures it as a perceptron. The first
```

```
% argument specifies the expected ranges of two inputs. The second determines
```

```
% that there is only one neuron in the layer.
```

```
net = newp([-1 1;-1 1],1); % NEWP Create a perceptron. 注意這個 "p"
```

```
% input1,2 輸入範圍 皆為：[-1 1]，後面的 1 表示 一個神經元
```

```
% 此時 觀察 IW(Input Layer) 初始值 >> net.IW{1,1} 得 ans = 0 0
```

```
% 此時 觀察 b(bias) 初始值 >> net.b{1} 得 ans = 0
```

```
% 此時也可以任意給定初始值，若給的好，可以加速學習
```

```
%%
```

```
% The input vectors are replotted with the neuron's initial attempt at
```

```
% classification.
```

```

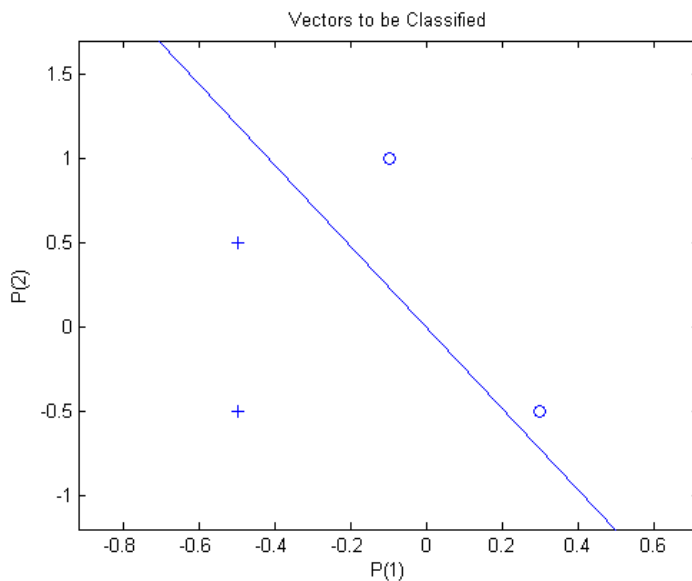
%
% The initial weights are set to zero, so any input gives the same output and
% the classification line does not even appear on the plot.  Fear not... we are
% going to train it!

plotpv(P,T);
plotpc(net.IW{1},net.b{1});
% PLOTPC Plot a classification line on a perceptron vector plot. 寫法：PLOTPC(W,B)

%%
% ADAPT returns a new network object that performs as a better classifier, the
% network output, and the error.

net.adaptParam.passes = 3; % 這是學 3 次的意思
net = adapt(net,P,T);      % 這指令結束後，就表示學完三次了
% 以上也可改為 net = train(net,P,T); 結果比較好喔！
plotpc(net.IW{1},net.b{1}); % 觀察輸出圖，就會發現已經成功的 分類 了

```



```

% 圖形輸出：
% 以上 若令 net.adaptParam.passes = 1，個別學習 3 次，可紀錄三次學習後的係數如下：
% first >> net.IW{1,1} ans = -0.2000 -0.5000 >> net.b{1} ans = -2
% second >> net.IW{1,1} ans = -1.2000 -0.5000 >> net.b{1} ans = 0
% third => 學習結果同 第二次，可見已達到穩定。
%%
% Now SIM is used to classify any other input vector, like [0.7; 1.2]. A plot of
% this new point with the original training set shows how the network performs.
% To distinguish it from the training set, color it red.

```

```

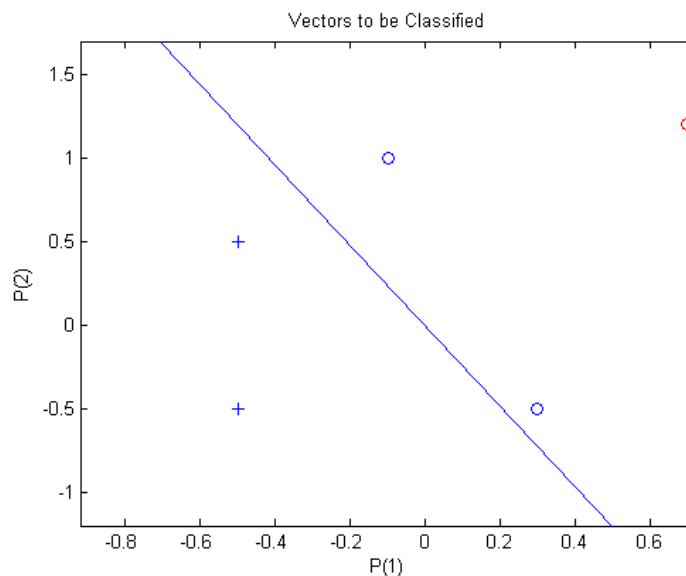
p = [0.7; 1.2]; % 加入一個測試點，看看結果如何？
a = sim(net,p);
plotpv(p,a);
point = findobj(gca,'type','line');
set(point,'Color','red');

%%
% Turn on "hold" so the previous plot is not erased and plot the training set
% and the classification line.
%
% The perceptron correctly classified our new point (in red) as category "zero"

```

% (represented by a circle) and not a "one" (represented by a plus).

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



### ※ 補充說明：

以上例子，類神經網路學習，主要是利用指令：

```
net = adapt(net,P,T);
```

這 `adapt` 指令，有 “Incremental Learning” 及 “Batch Learning” 兩種：（`train` 只能用在 `Batch Learning`）

#### ■ Batch Learning

如上例。以上例來說，一批次有 4 筆資料，每累積四次後，再作一次修正。使用資料型別如下：

- `adapt` 指令：I/O 用普通陣列
- `train` 指令：I/O 最好用 Cell Array

#### ■ Incremental Learning

一步一步學，對每一筆資料作學習。此時須將 I/O data `P,T` 以 Cell Array 表示，請讀者如下修改，重作上例比較之。

```
>> P={[-0.5;-0.5],[-0.5;0.5],[0.3; -0.5],[-0.1; 1.0]}
```

```
P = [2x1 double] [2x1 double] [2x1 double] [2x1 double]
```

```
>> T = {1 1 0 0}
```

輸出可用 `[net, a, e, pf] = adapt(net,P,T);` 輸出 `a`, 及誤差 `e` 亦為 Cell Array.

每呼叫一次 `adapt`，就步進學習 4 次，這叫一個 `pass`，`net.adaptParam.passes` 為 `pass` 數

Reference: 類神經網路入門 活用 Matlab，周鵬程 編著，全華