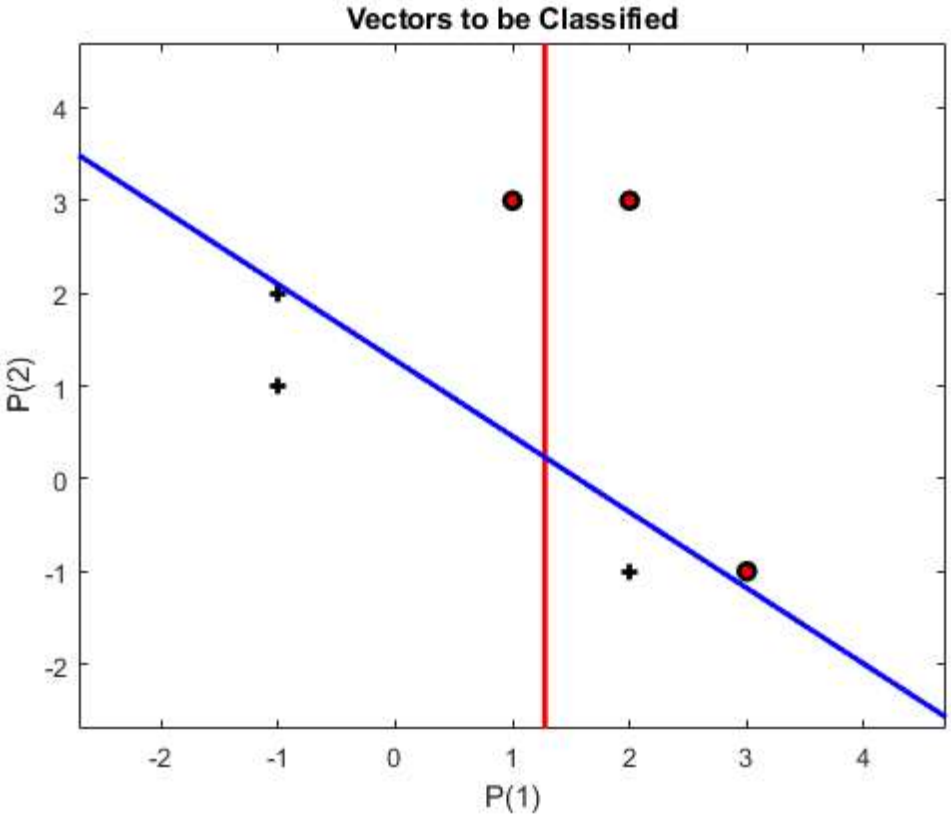
```
clear all; clc; cla; clf;
pause_flag = 0;
max_epoch = 200;

%Input vectors
P = [
    -1.0 -1.0 2.0 1.0 2.0 3.0;
    1.0 2.0 -1.0 3.0 3.0 -1.0;
    1.0 1.0 1.0 1.0 1.0 1.0
    ];

%Target vector
T = [1 1 1 0 0 0];

%Input layer
[R, Q] = size(P); [S, Q] = size(T);

%Initialize network parameters
figure(1);
plotpv(P(1:R-1,:), T);
Change_Marker

%Initialize weights randomly
W = rand(S,R);
Wp = W(:, 1:R-1);
Bp = W(:,R);

%display initial values
%The input vectors are replotted
plotpv(P(1:R-1,:), T);

plotpc(Wp, Bp);

watchon;
cla;
plotpv(P(1:R-1,:), T);

pause(3);
figure(1);
E=1;
linehandle = plotpc(Wp, Bp);
%disp('Hit something to continue');

%sum squared error performance function
epoch = 1;
while (sse(E) && (epoch <= max_epoch))
    Ai = hardlim(W*P);
    Ei = T-Ai;
    dWq = learnp(W, P, [], [], [], [], Ei, [], [], [], [], []);
    W = W+dWq;
    Wp = W(:, R-1);
    Bp = W(:, R);
    linehandle = plotpc(Wp, Bp, linehandle);
    lines = findobj(gcf, 'Type', 'Line');
    Change_LineWidth
    Change_Marker
    drawnow;
    if(pause_flag == 1)
        pause(1);
```

```
    end
    A = hardlim(W*P);
    %error - Target minus calculated this epoch
    E = T-A;
    %disp(E)
    epoch = epoch +1;
end
watchoff;
disp('Target is ')
T
disp('Solution reached of ')
A
disp('With weights ')
W

testPoint = findobj(gca, 'Type', 'Line');
set(testPoint, 'Color', 'red');
hold on;
plotpv(P(1:R-1, :), T)
Wp = W(:, 1:R-1);
Bp = W(:, R);
plotpc(Wp, Bp);
Change_LineWidth
Change_Marker
hold off;
```

```
Target is

T =

     1     1     1     0     0     0

Solution reached of

A =

     1     1     1     0     0     0

With weights

W =

   -2.2487   -2.7449    3.5060
```



Published with MATLAB® R2022a