

```

clear all; clc; cla; clf;
pause_flag = 0;
max_epoch = 200;
P = [
    2 3 2 3 3 4 5 1 1 1 -1 -4 -2 -2;
    0 0 1 4 5 6 6 3 4 5 3 4 3 4;
    1 1 1 1 1 1 1 1 1 1 1 1 1 1
];

T = [
    0 0 0 0 0 0 0 1 1 1 1 1 1 1;
    0 0 0 1 1 1 1 1 1 1 0 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);

%sum squared error performance function
epoch = 1;
while (sse(E) && (epoch <= max_epoch))
    Ai = hardlim(W*P);
    Ei = T-Ai;
    dWq = learnnp(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);

```

```

E = T-A;
epoch = epoch +1;
end
watchoff;
disp('Target is ')
T
disp('Solution reached ')
A
disp('With weights')
W

p = [0.5; 0.5; 1];
a = hardlim(W*P)
plotpv(p(1:2), a);
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 =

Columns 1 through 13

0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	0	0	0

Column 14

1
0

Solution reached

A =

Columns 1 through 13

0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	0	0	0

Column 14

1
0

With weights

W =

-31.2345	12.1869	3.4456
5.7952	8.4898	-25.3537

a =

Columns 1 through 13

0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	0	0	0

Column 14

1
0

