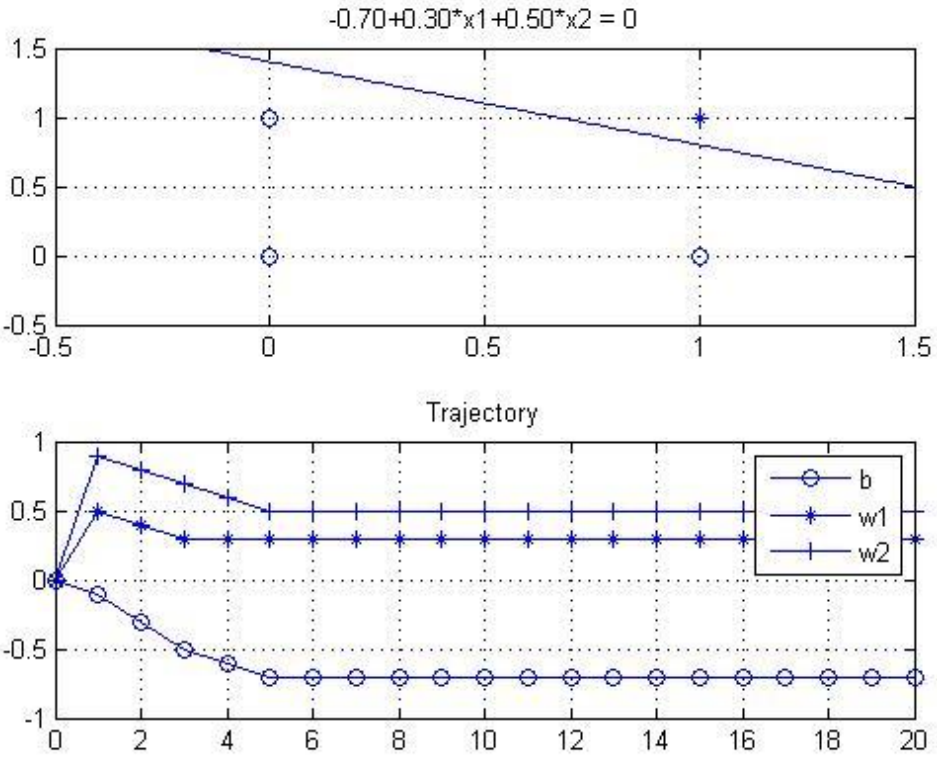
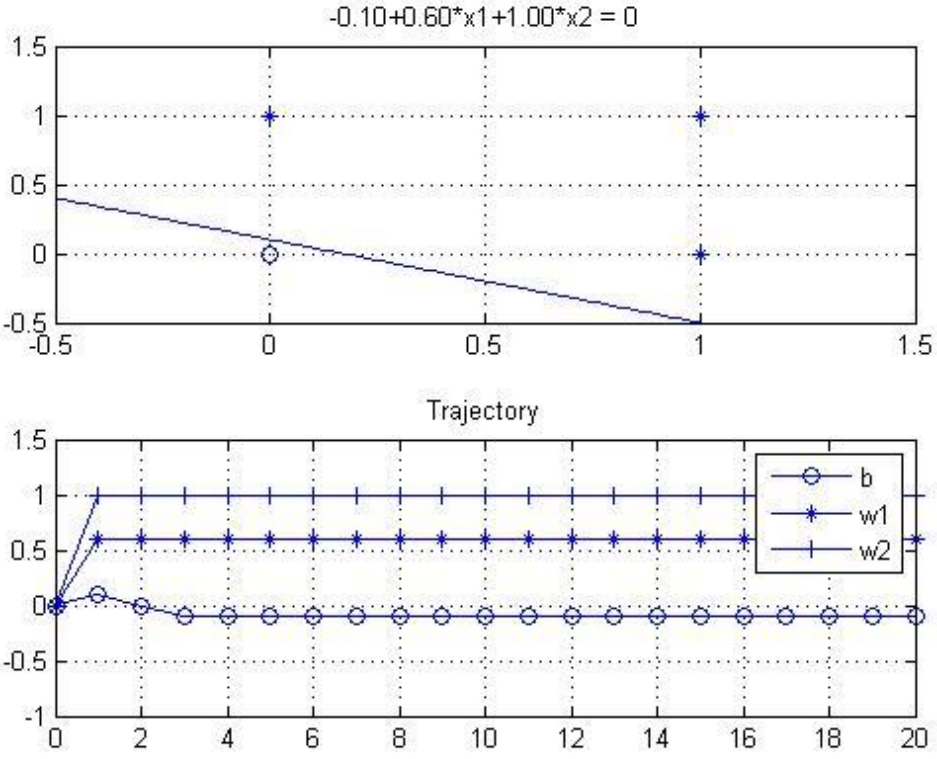
Q3.

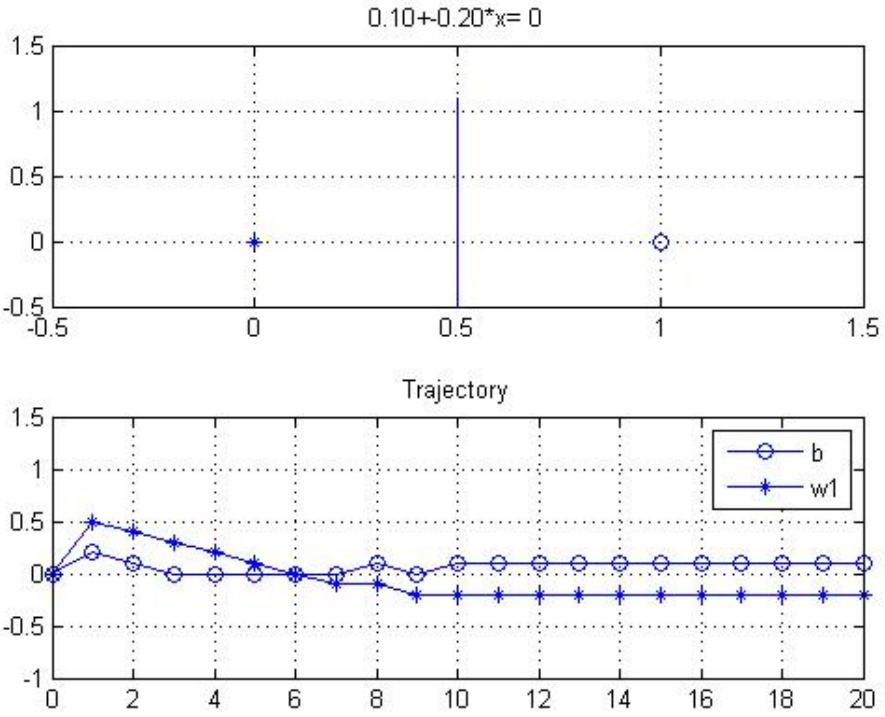
2) AND: W= [0.2 0.6 1]; speed=0.1;



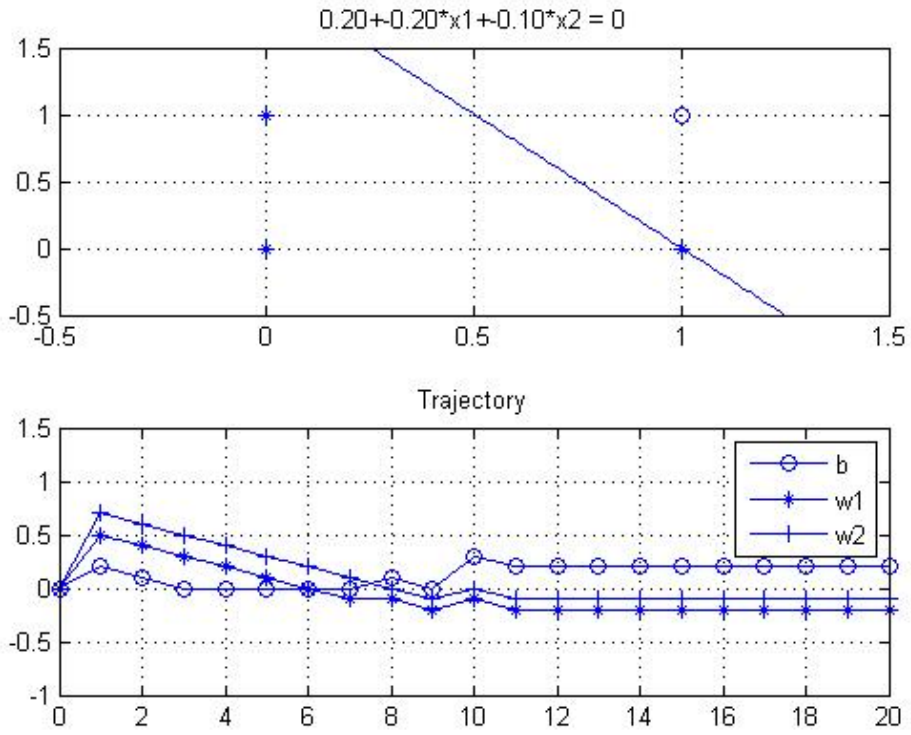
OR: W= [0.2 0.6 1]; speed=0.1;



COMPLEMENT: W= [0.3 0.6]; speed=0.1;



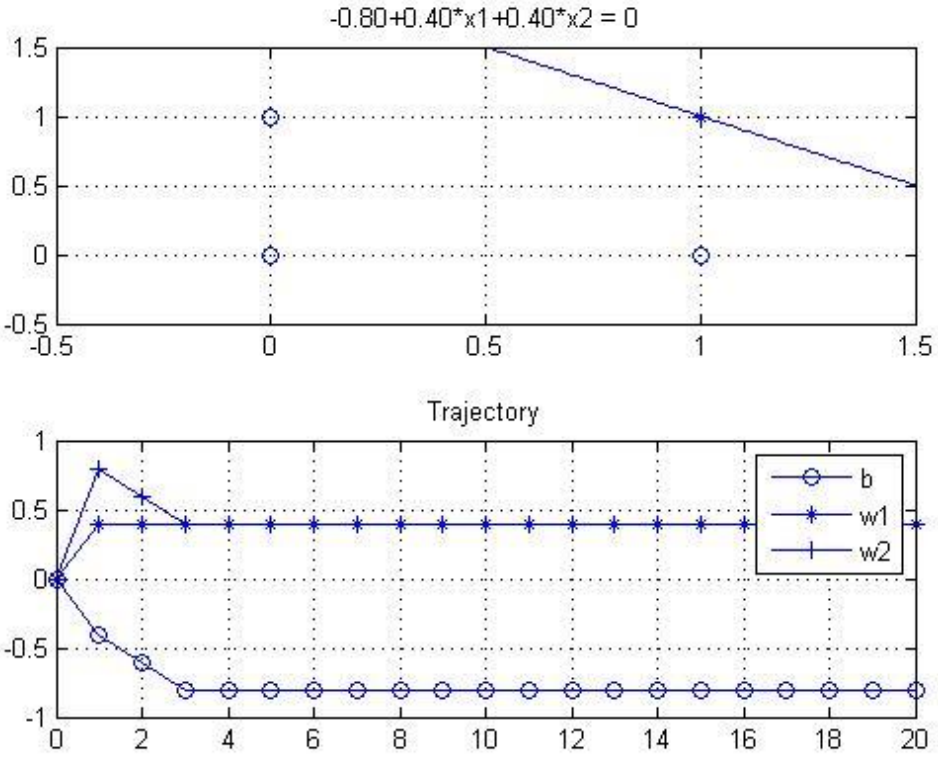
OR: W= [0.3 0.6 0.8]; speed=0.1;



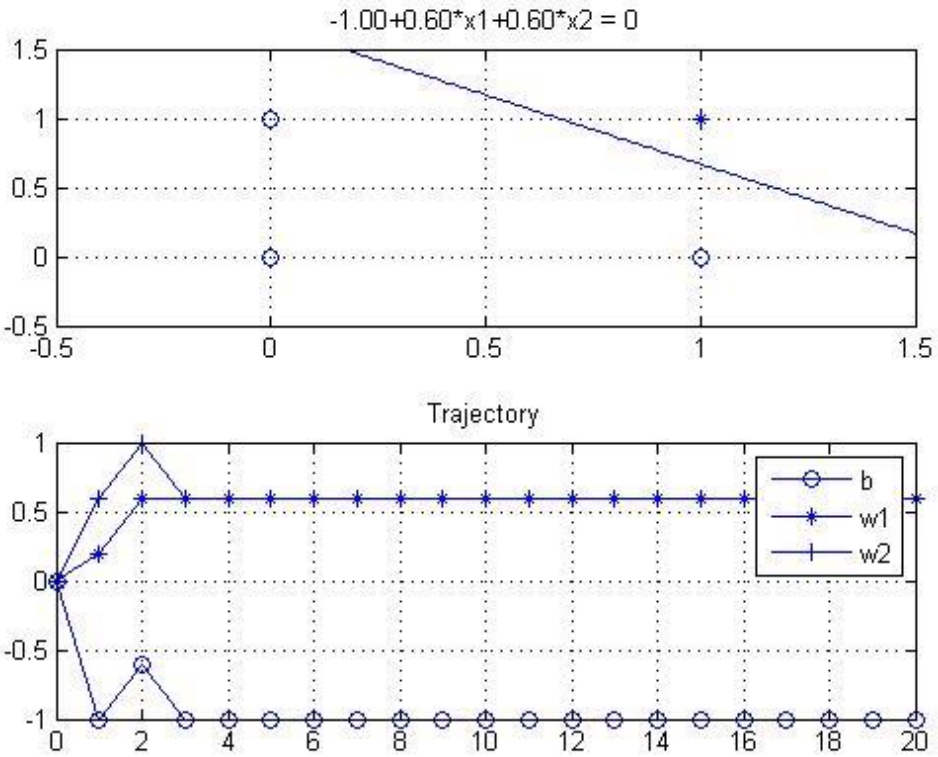
Comparing with the results with those obtained in (a), both of these fulfill the requirement of question. But the value of weight is quite different because of the difference of learning speed and origin value of weight.

Change the learning rates of AND perceptron: 0.2 0.4 0.6 0.8 1.0

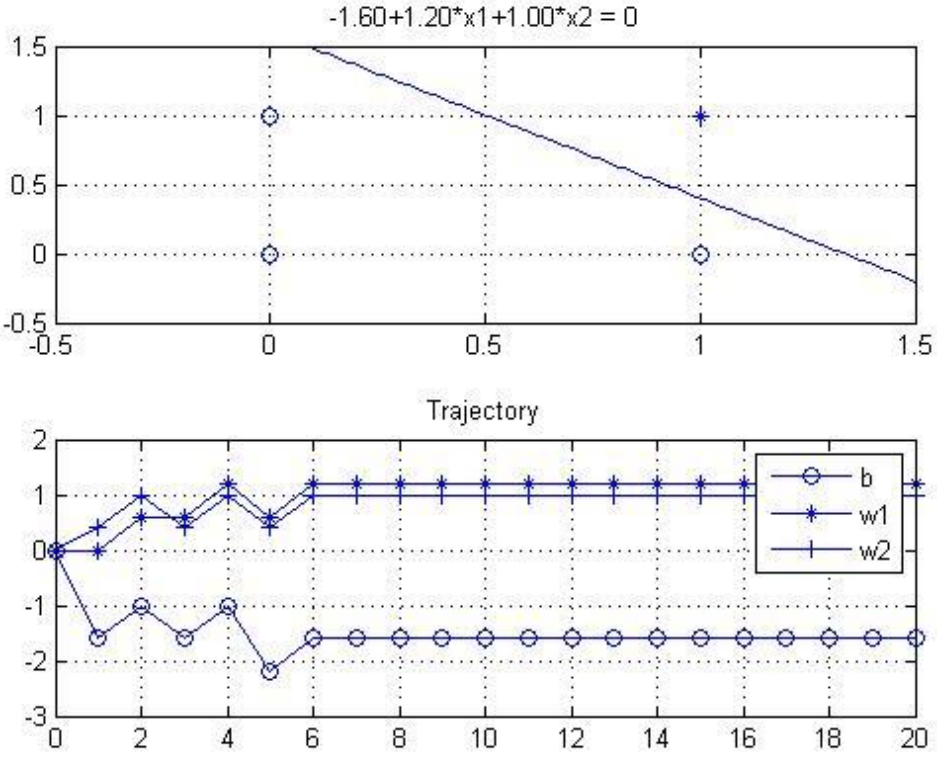
AND: W= [0.2 0.6 1]; speed=0.2;



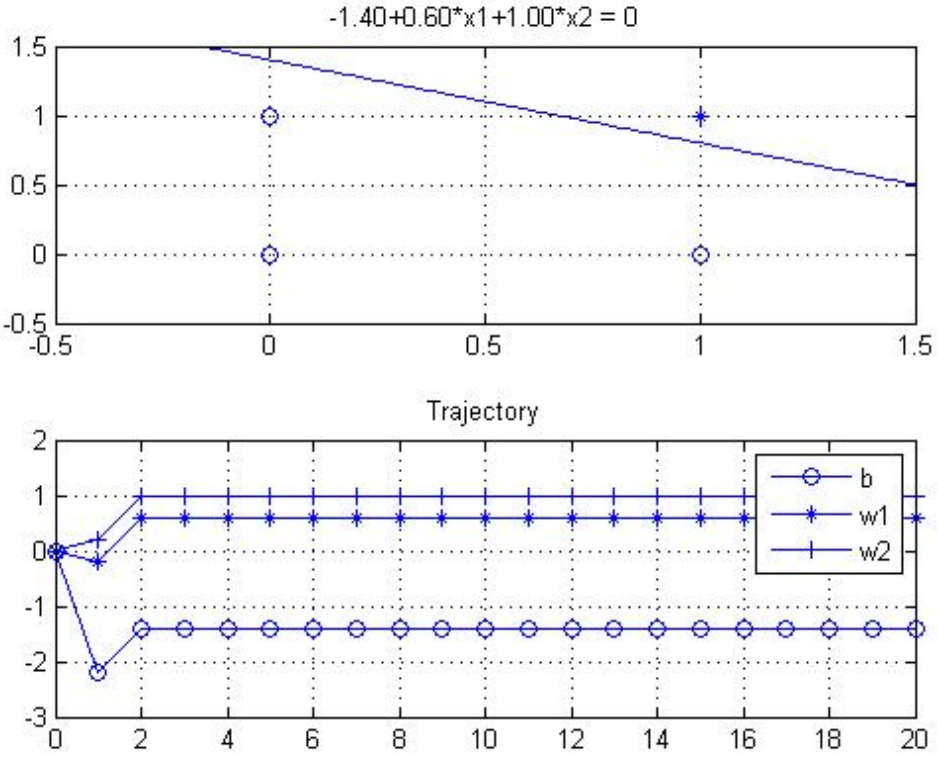
AND: W= [0.2 0.6 1]; speed=0.4;



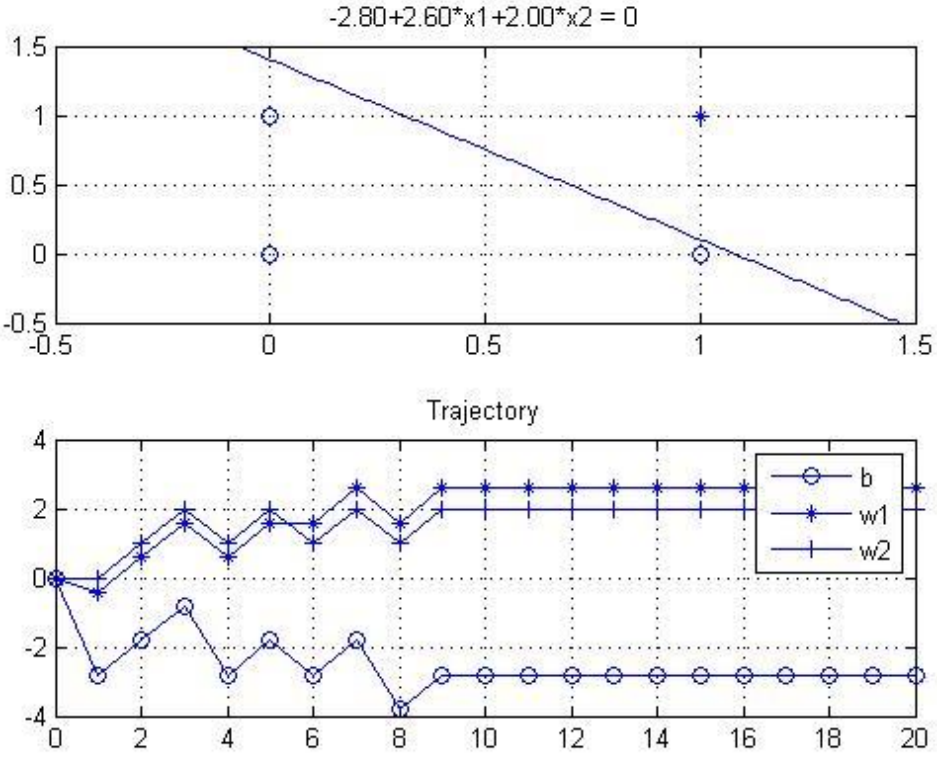
AND: W= [0.2 0.6 1]; speed=0.6；



AND: W= [0.2 0.6 1]; speed=0.8；



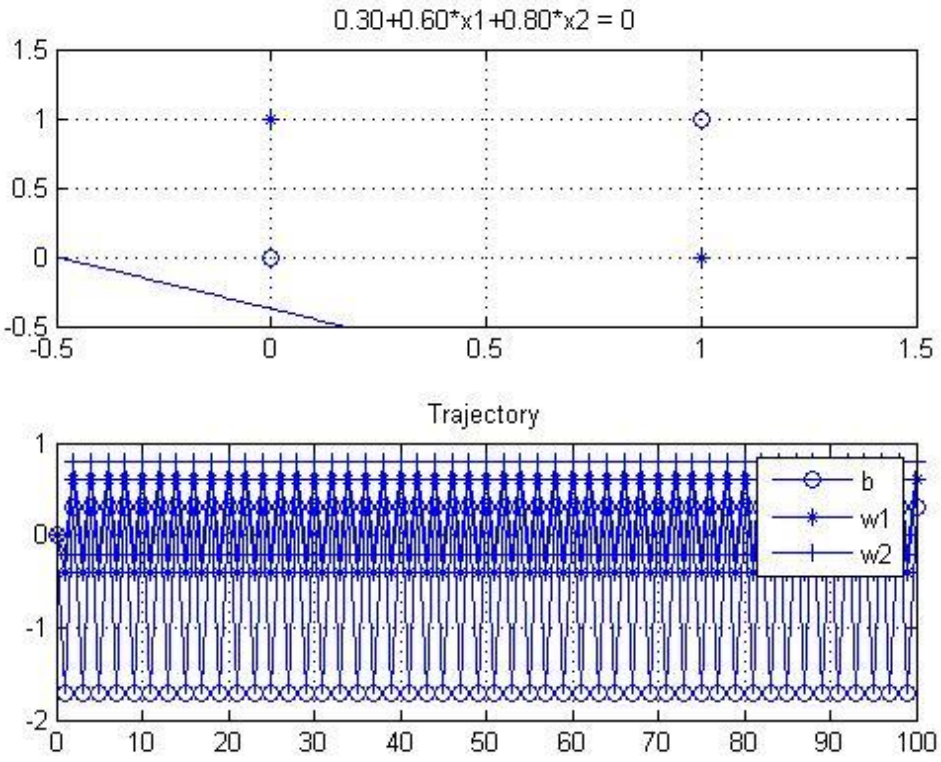
AND: W= [0.2 0.6 1]; speed=1.0；



Changing the learning with value: 0.2, 0.4, 0.6, 0.8, and 1.0.

All the result could fulfill the requirement that perform numerous logic functions. For these value, the speed 0.8 is the fastest way to get the final value. With the increasing of speed, the final value may have the higher magnitude.

C) If the perceptron is applied to implement the EXCLUSIVE OR function with selection of weights by learning procedure. The computer experiment result is shown below.



From the figure, the trajectory of weights shows that the algorithm could not converges after the first epoch (100 times) for this problem. And the result straight line could not fulfill the requirement that classify the nodes into two parts. Since the XOR is not linearly separable, it's impossible to classify the nodes with one straight line.

The following is the MATLAB code of AND:

```
%%INPUT
X=[1 1 1 1
    0 0 1 1
    0 1 0 1];
d=[0 0 0 1];
W=[0.2 0.6 1];
speed=0.1;
times=20;

%%MAIN CODE
i=0;
while 1
      y=hardlim(W*X);
      error=d-y;
      W = W+speed*error*X';
      i=i+1;
      w1(i+1) = W(1);
      w2(i+1) = W(2);
      w3(i+1) = W(3);
      if (i>=times)
            break;
      end
end

%%OUTPUT
figure
subplot(2,1,1);
plot([0,0,1],[0,1,0],'o');
grid on;
hold on;
plot(1,1,'*');
axis([-0.5,1.5,-0.5,1.5]);
s = sprintf('%.2f+%.2f*x1+%.2f*x2 = 0',W(1),W(2),W(3));
title(s);
x = -0.5:.01:1.5;
y = x*(-W(2)/W(3))-W(1)/W(3);
plot(x,y);

subplot(2,1,2);
x = 0:i;
plot(x,w1,'o-');
grid on;
```

```
hold on;
plot(x,w2,'*-');
hold on;
plot(x,w3,'+-');
axis([0,20,-1,1]);
legend('b','w1','w2');
title('Trajectory');
hold off;
```