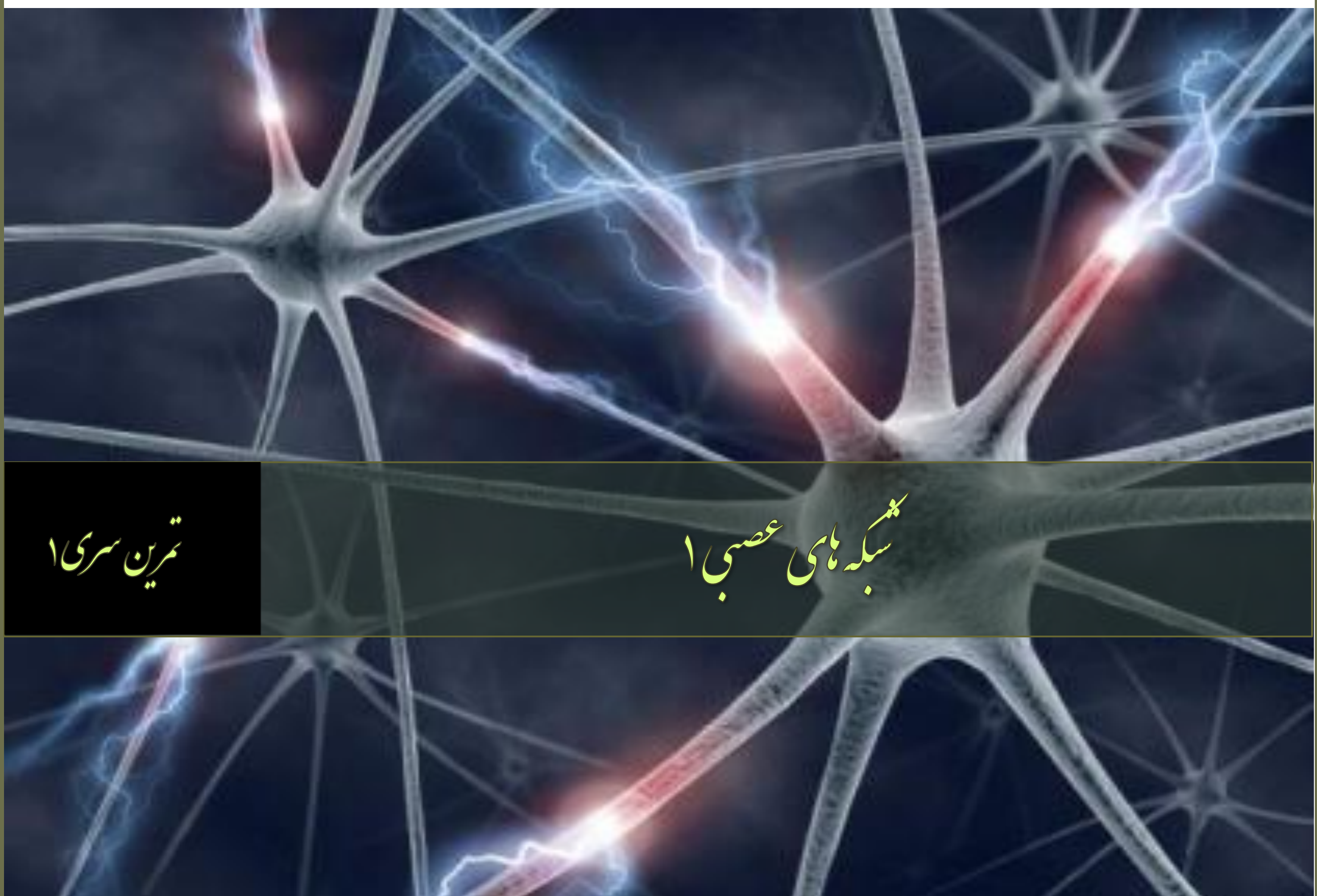
۱۳۹۱/۱۲/۶



شبکه های عصبی ۱

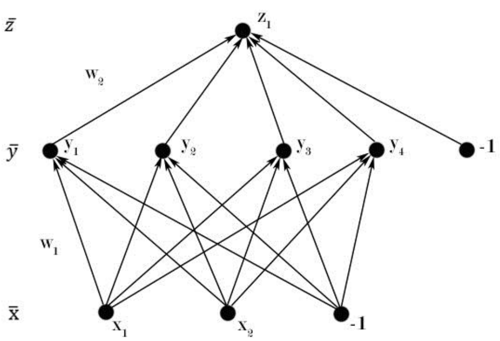
تمرین سری۱

استاد درس:

جناب آقای دکتر سید صالحی

دانشجو: بهتام عادلی | ۹۰۱۳۳۰۲۲

جواب ۱:



شبکه زیر با یک لایه پنهان را در نظر می‌گیریم:

$$w_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & -2 & 0 & -3 \end{bmatrix}$$ ، وزنهای لایه اول

$$w_2 = [1 \;\; 1 \;\; 1 \;\; 1 \;\; 3.5]^T$$ ، وزن های لایه دوم

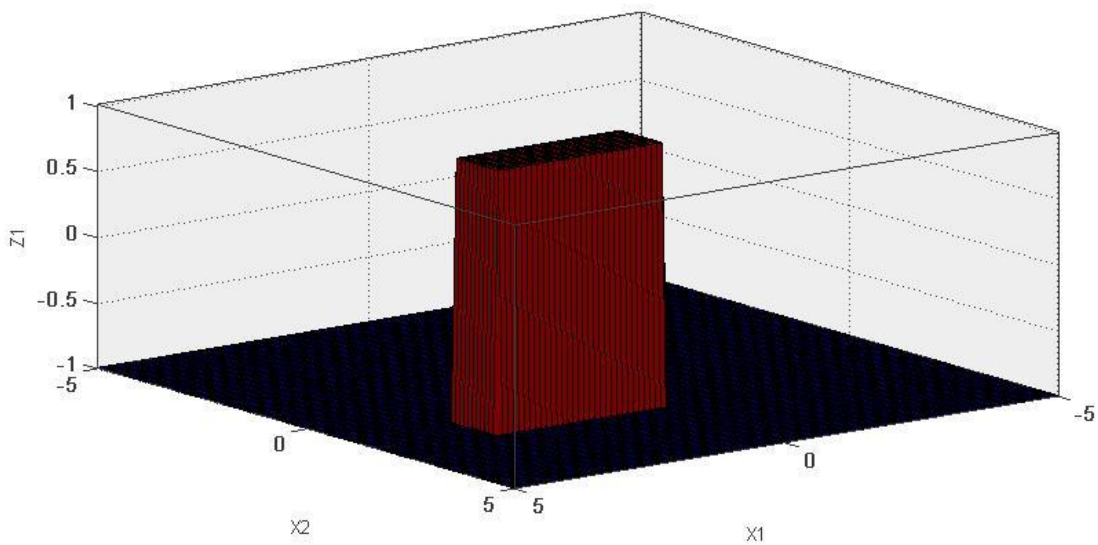
الف) با تابع غیر خطی پله‌ای

$$f(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

برای بدست آوردن مرز تصمیم و جواب خروجی لایه z از کدهای زیر در MATLAB استفاده می‌کنیم:

```
clear all;
close all;
clc;

m=-5:0.1:5;
w1=[1 -1 0 0;0 0 1 -1;1 -2 0 -3];
w2=[1 1 1 1 3.5];
y=zeros(5,length(m),length(m));
% t1=zeros(4,length(m),length(m));
z=zeros(length(m),length(m));
for i=1:4
    for j=1:length(m)
        x1=m(j);
        for k=1:length(m)
            x2=m(k);
            X=[x1 x2 -1];
            t1=dot(X,(w1(:,i)));
            if (t1<0)
              y(i,j,k)=-1;
            else
              y(i,j,k)=1;
            end
        end

    end

end

y(5,:,:)=-1;
for o=1:length(m)
    for p=1:length(m)
```

```
        t2=dot(y(:,o,p),w2);
        if (t2<0)
            z(o,p)=-1;
        else
            z(o,p)=1;
        end
    end
end
x1=-5:0.1:5;
x2=-5:0.1:5;
meshgrid(x1,x2);
surf(x1,x2,z);
grid on
hold on
```



ب) با تابع نرم :

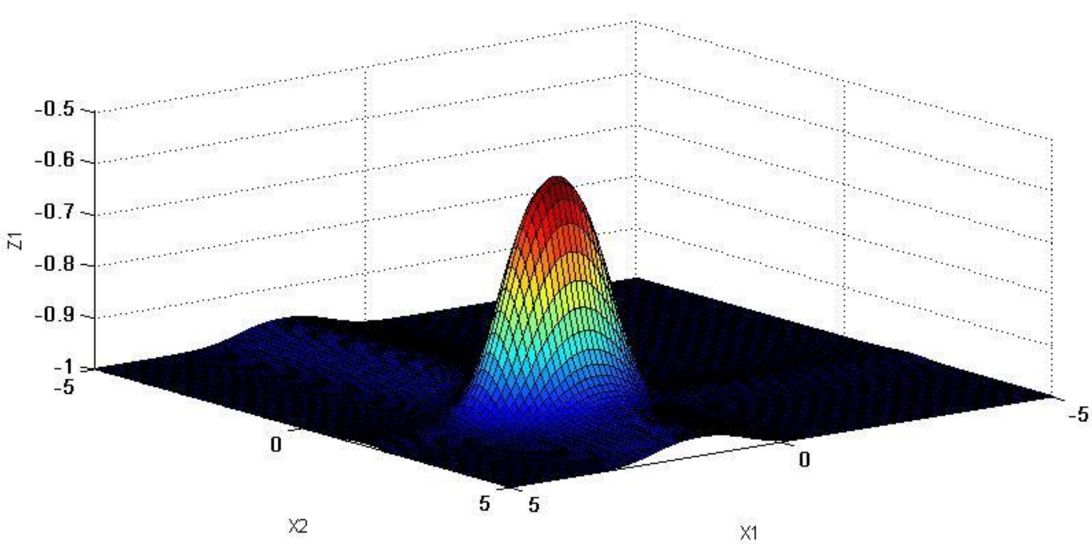
$$f(x) = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}} = \frac{2}{1 + e^{-\lambda x}} - 1 \qquad \lambda = 2.5$$

باید در برنامه تغییراتی به صورت زیر اعمال نماییم:

```
function f=sigmoid(landa,x)
f=(1-exp(-landa*x))/(1+exp(-landa*x));
end

clear all;
close all;
clc;

m=-5:0.1:5;
w1=[1 -1 0 0;0 0 1 -1;1 -2 0 -3];
w2=[1 1 1 1 3.5];
```

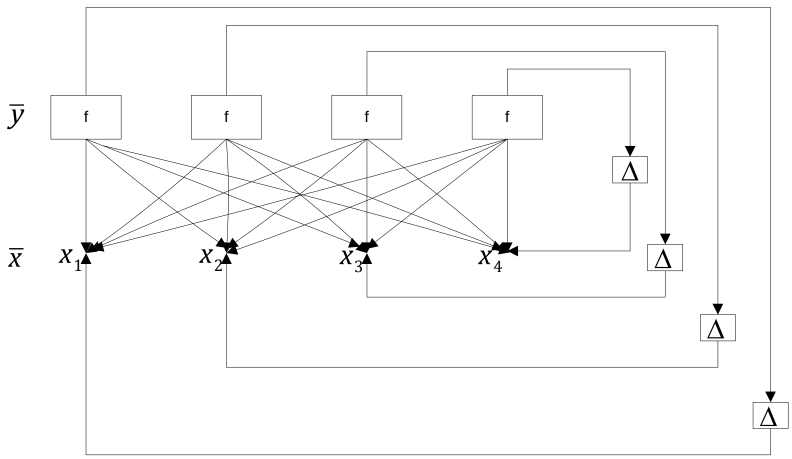
```
y=zeros(5,length(m),length(m));
% t1=zeros(4,length(m),length(m));
z=zeros(length(m),length(m));
for i=1:4
    for j=1:length(m)
        x1=m(j);
        for k=1:length(m)
            x2=m(k);
            X=[x1 x2 -1];
            t1=dot(X,(w1(:,i)));

              y(i,j,k)=sigmoid(2.5,t1);
        end

    end

end

y(5,:,:)=-1;
for o=1:length(m)
    for p=1:length(m)
        t2=dot(y(:,o,p),w2);
        z(o,p)=sigmoid(2.5,t2);
    end
end
x1=-5:0.1:5;
x2=-5:0.1:5;
meshgrid(x1,x2);
surf(x1,x2,z);
grid on
hold on
```

که نتیجه در تصویر زیر دیده می‌شود. نکته قابل توجه این است که با استفاده از این تابع خروجی ما هیچ‌گاه به یک نمی‌رسد. مرزها نرم تر شده ولی در کل همان شکل قبل می‌باشد.



جواب ۲:

شبکه مورد نظر با بردار وزن w در شکل زیر آمده است:



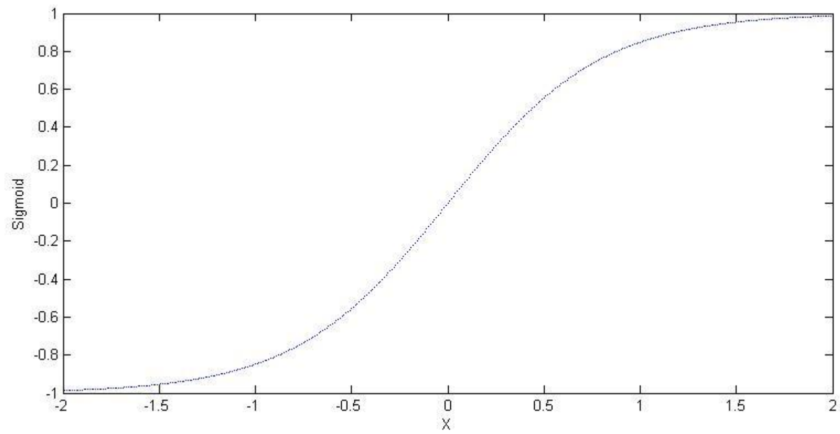
$$w = \begin{bmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 \\ 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{bmatrix}$$

ابتدا برای حالت تابع غیر خطی نرم سیگوئید (حالت سوم) پاسخ شبکه را برای تغییرات X_1 تا X_4 بین ۱- تا ۱ برای هر کدام، بررسی می‌کنیم. برای این کار، تغییرات هر یک از ورودی‌ها را بر روی شبکه که ۵۰ بار حلقه فیدبک کار کرده‌است با استفاده از دستور زیر در نرم‌افزار MATLAB بررسی می‌کنیم:

```
clear all;
close all;
clc;

a=-1:0.1:1;
W=[0 1 1 -1;1 0 1 -1;1 1 0 -1;-1 -1 -1 0];
y=zeros(4,length(a),length(a));
    for j=1:length(a)
        x1=a(j);
        for k=1:length(a)
            x2=a(k);
            for l=1:length(a)
                x3=a(l);
                for m=1:length(a)
                    x4=a(m);
                    X=[x1 x2 x3 x4];
                    for N=1:50
                        for i=1:4
                            t1=dot(X,(W(:,i)));
                            y(i,j,k,l,m)=sigmoid(2.5,t1);
                        end
                        X=y(:,j,k,l,m);
                    end
                end
```

```
            end
        end
    end
```

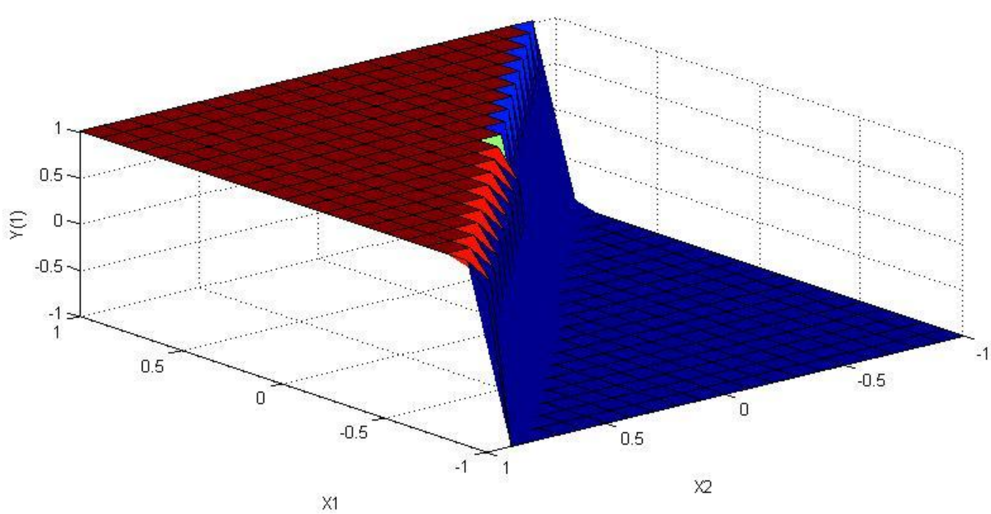
شکل تابع سیگموئید به صورت زیر خواهد بود:

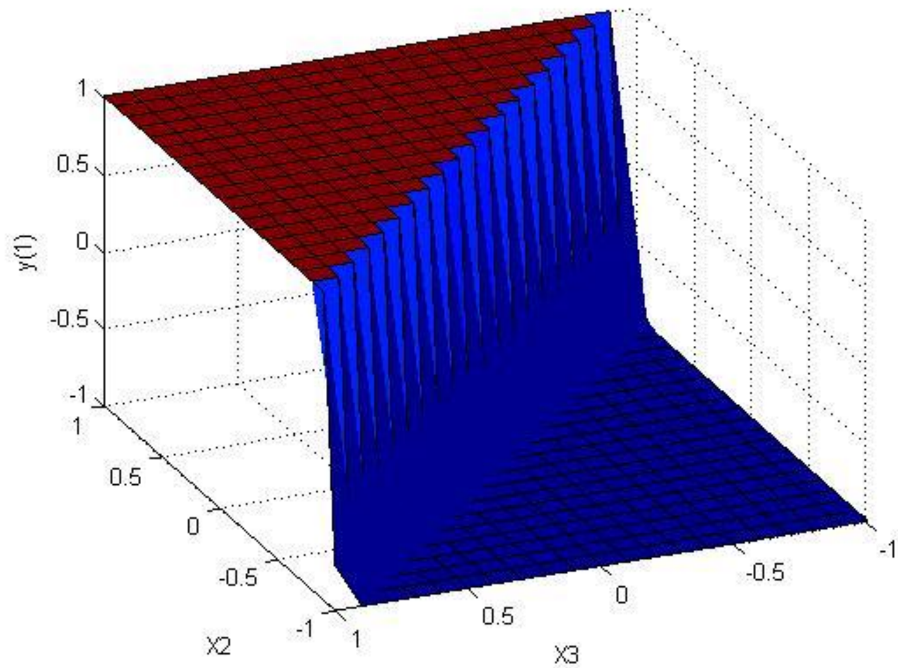


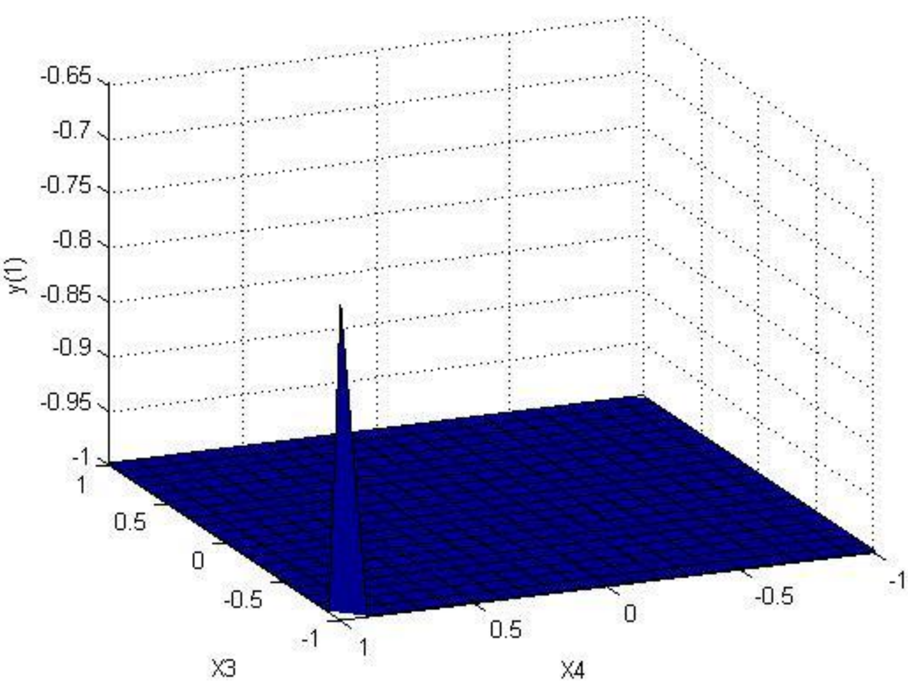
سپس دستور زیر را برای حالات متفاوت می توانیم نوشته و نمودار خروجی را بدست آوریم:

```
x1=-1:0.1:1;
x2=-1:0.1:1;
[X1,X2] = meshgrid(x1,x2);
surfc(X1,X2,squeeze(y(4,1,1,:,:)))
colormap hsv
grid on
hold on
xlabel('X3')
ylabel('X4')
zlabel('y(1)')
```

تغییرات $y(1)$ برای تغییرات ورودی‌ها به صورت زیر خواهد بود.







حال نتیجه نقاط کنج این ابرمکعب چهار بعدی را جهت سادگی بررسی می‌کنیم:

```
t(1,:)=y(:,1,1,1,1);
```

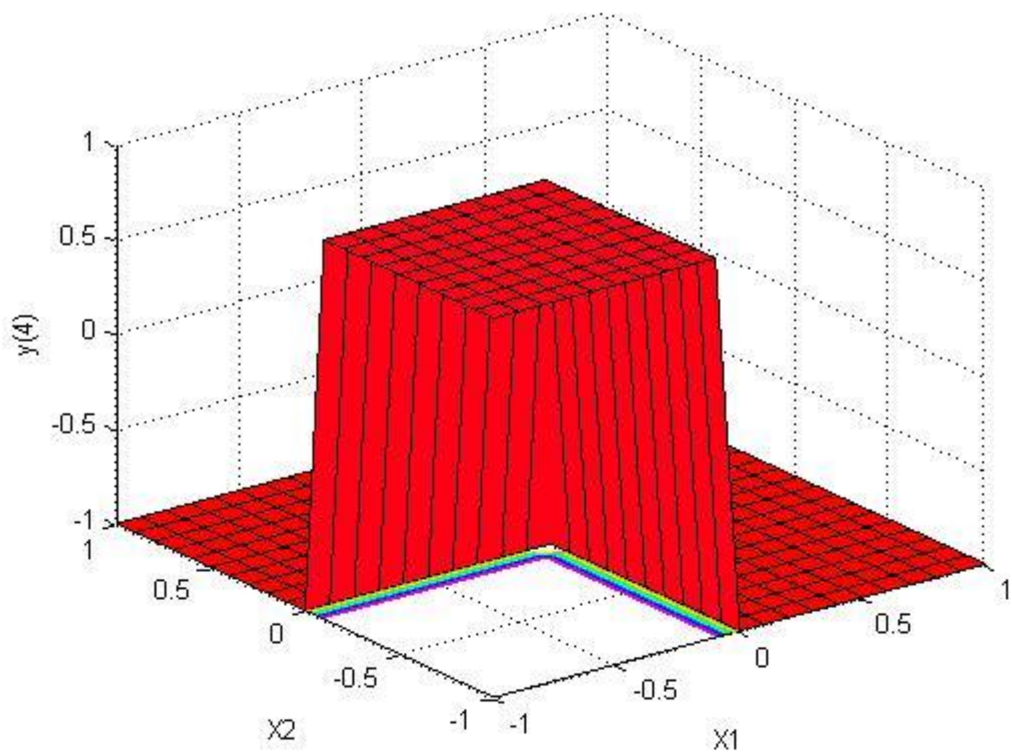
```
t(2,:)=y(:,1,1,1,21);
t(3,:)=y(:,1,1,21,1);
t(4,:)=y(:,1,21,1,1);
t(5,:)=y(:,21,1,1,1);
t(6,:)=y(:,1,1,21,21);
t(7,:)=y(:,1,21,21,1);
t(8,:)=y(:,21,21,1,1);
t(9,:)=y(:,1,21,1,21);
t(10,:)=y(:,21,1,1,21);
t(11,:)=y(:,21,1,21,1);
t(12,:)=y(:,1,21,21,21);
t(13,:)=y(:,21,21,21,1);
t(14,:)=y(:,21,1,21,21);
t(15,:)=y(:,21,21,1,21);
t(16,:)=y(:,21,21,21,21);
```

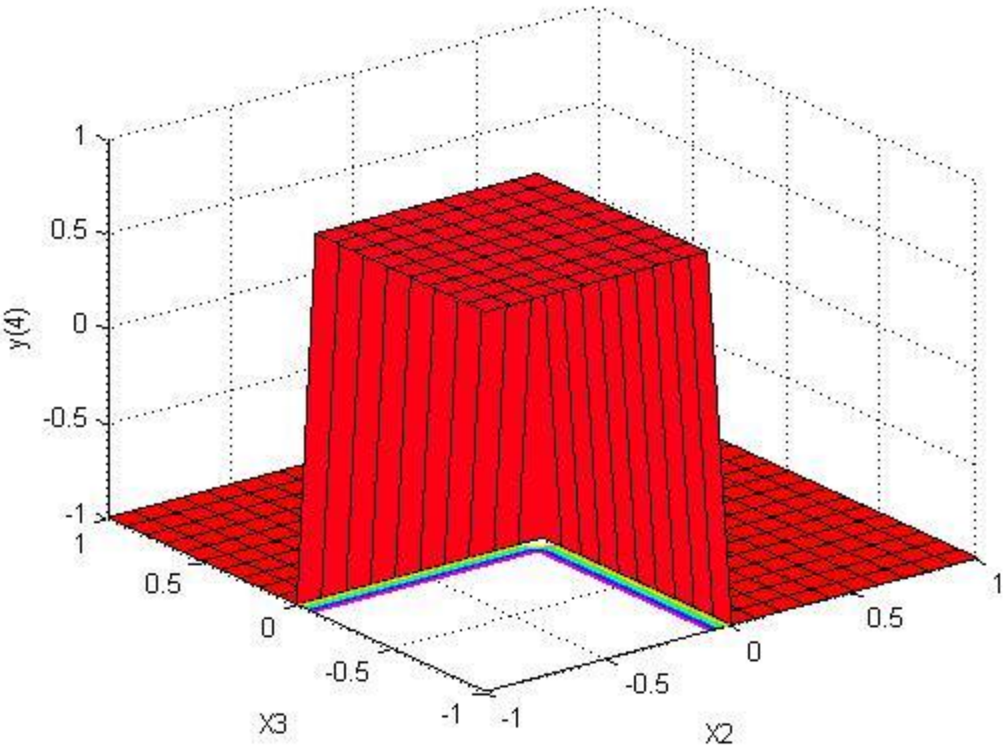
```
t =

   -0.9989   -0.9989   -0.9989    0.9989

   -0.9989   -0.9989   -0.9989    0.9989

   -0.7101   -0.7101    0.7107   -0.7107

   -0.7101    0.7107   -0.7101   -0.7107

    0.7107   -0.7101   -0.7101   -0.7107

   -0.9989   -0.9989   -0.9989    0.9989

    0.9989    0.9989    0.9989   -0.9989

    0.9989    0.9989    0.9989   -0.9989

   -0.9989   -0.9989   -0.9989    0.9989

   -0.9989   -0.9989   -0.9989    0.9989

    0.9989    0.9989    0.9989   -0.9989

   -0.7102    0.7107    0.7107    0.7102

    0.9989    0.9989    0.9989   -0.9989

    0.7107   -0.7102    0.7107    0.7102

    0.7107    0.7107   -0.7102    0.7102

    0.9989    0.9989    0.9989   -0.9989
```

حال به بررسی همان سیستم با همان ورودی (X1=X2=X3=X4=-1:0.1:1) ولی این بار با تابع تصمیم غیرخطی سخت (پله واحد دوقطبی) می‌پردازیم. برای این کار از همان کدهای قبلی تنها با یک تفاوت کوچک استفاده می‌نماییم (استفاده از دستور زیر به جای تابع سیگموئید):

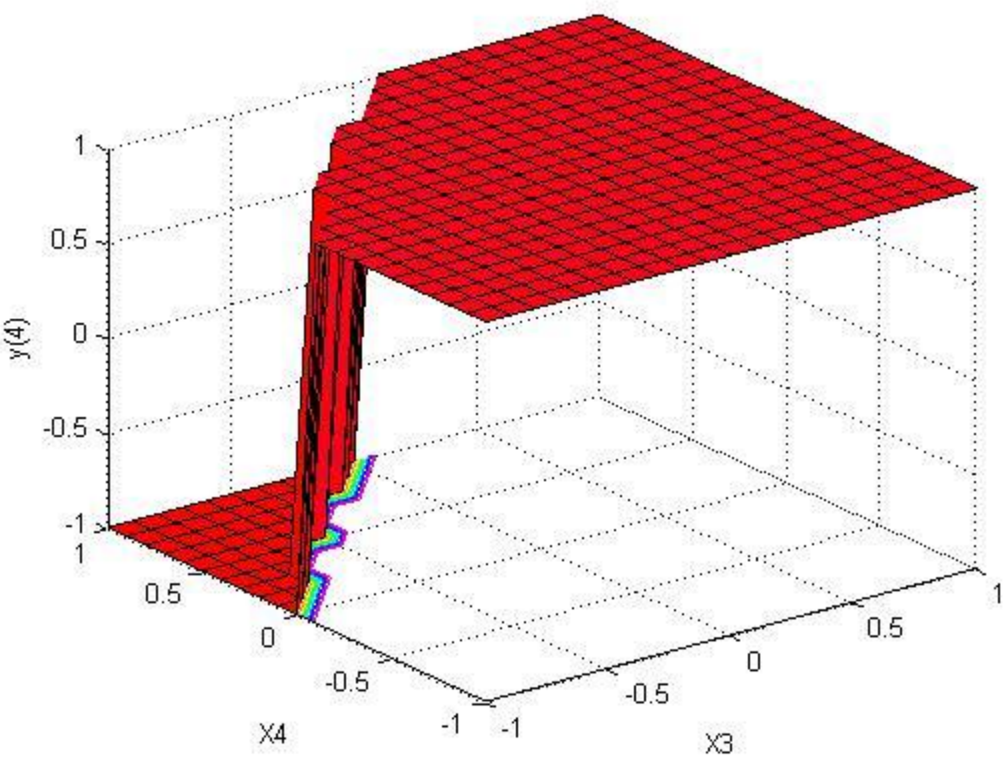
```
for N=1:50
      for i=1:4
          t1=dot(X,(W(:,i)));
          if t1<0
          y(i,j,k,l,m)=-1;
          t(i,j,k,l,m)=-1;
          else
              y(i,j,k,l,m)=1;
          t(i,j,k,l,m)=1;
          end
      end
      X=y(:,j,k,l,m);
  End
```

نتایج برای خروجی y(4) در اشکال زیر رسم شده‌است:





y(4)
1
0.5
0
-0.5
-1
X3
X2



y(4)
1
0.5
0
-0.5
-1
X4
X3

بررسی نقاط مرزی ابر معکب چهار بعدی:

```
T(1,:)=y(:,1,1,1,1);
T(2,:)=y(:,1,1,1,21);
T(3,:)=y(:,1,1,21,1);
T(4,:)=y(:,1,21,1,1);
T(5,:)=y(:,21,1,1,1);
T(6,:)=y(:,1,1,21,21);
T(7,:)=y(:,1,21,21,1);
T(8,:)=y(:,21,21,1,1);
T(9,:)=y(:,1,21,1,21);
T(10,:)=y(:,21,1,1,21);
T(11,:)=y(:,21,1,21,1);
T(12,:)=y(:,1,21,21,21);
T(13,:)=y(:,21,21,21,1);
T(14,:)=y(:,21,1,21,21);
T(15,:)=y(:,21,21,1,21);
T(16,:)=y(:,21,21,21,21);
```

T =

-1	-1	-1	1
-1	-1	-1	1
-1	-1	1	-1
-1	1	-1	-1
1	-1	-1	-1
-1	-1	-1	1
1	1	1	-1
1	1	1	-1
-1	-1	-1	1
-1	-1	-1	1
1	1	1	-1
-1	1	1	1
1	1	1	-1
1	-1	1	1
1	1	-1	1
1	1	1	-1