به نام خدا

پیاده سازی توابع منطقی توسط شبکه های عصبی

نویسنده

محمد نحوی

کلمات کلیدی

شبکه های عصبی MLP.متلب.

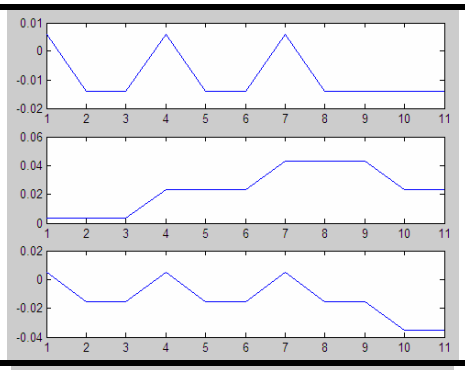
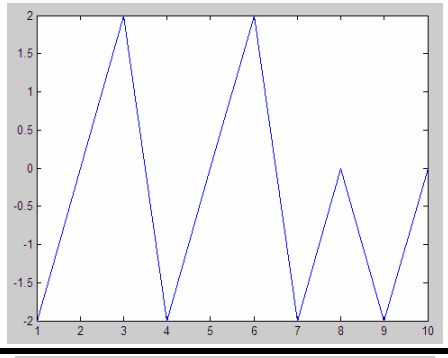
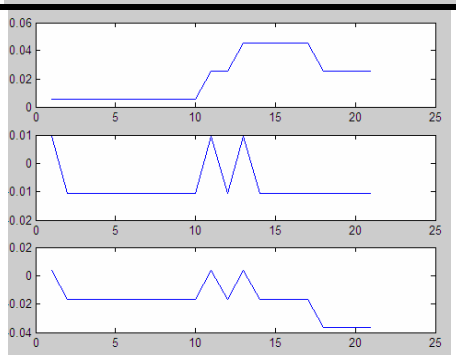
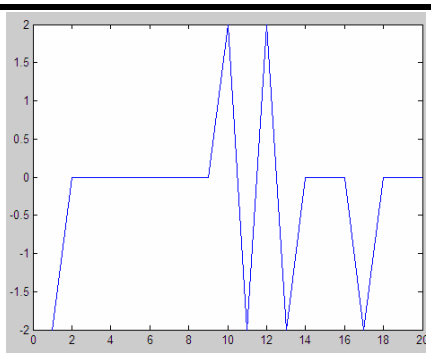
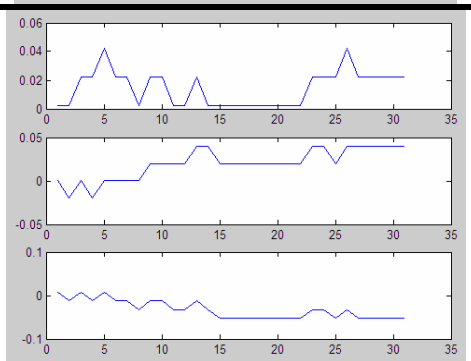
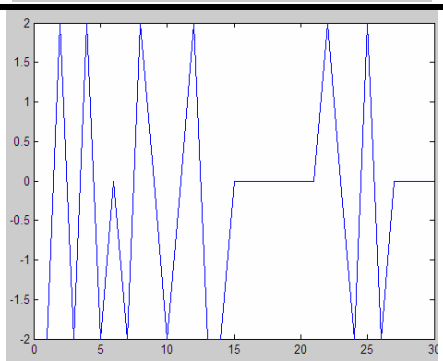
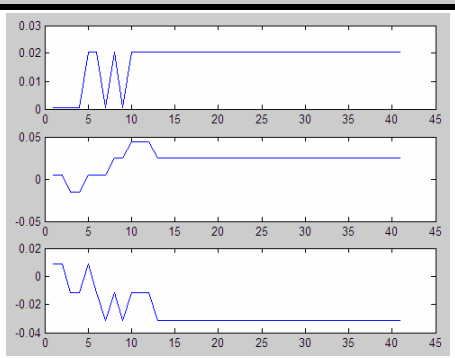
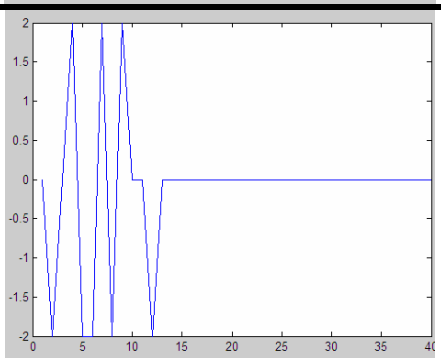
چکیده

در این مقاله چندین دروازه منطقی توسط شبکه های عصبی MLP پیاده سازی شده است.

۱– شبیه سازی AND با یادگیری آنلاین به همراه بررسی اثرات η ,W , K بر همگرایی شبکه :

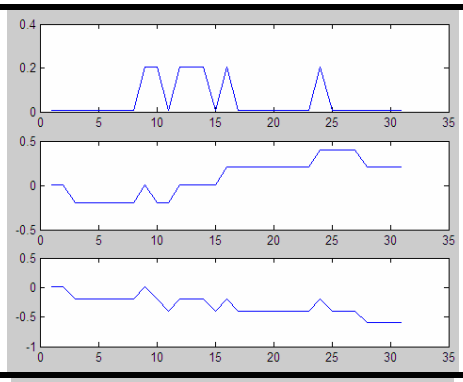
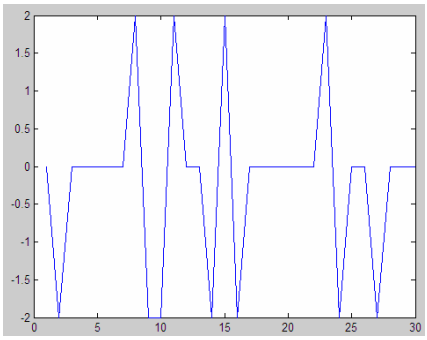
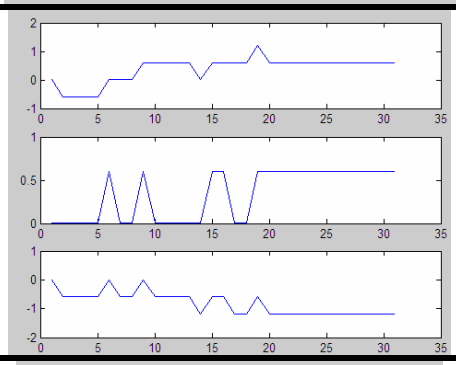
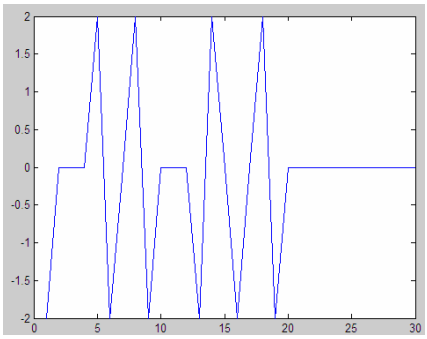
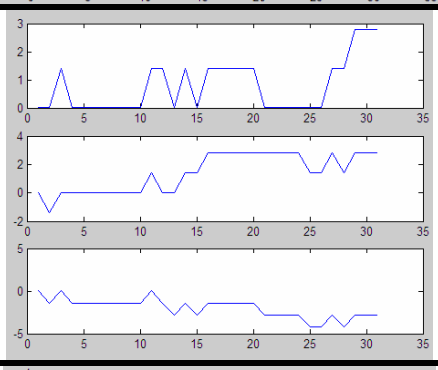
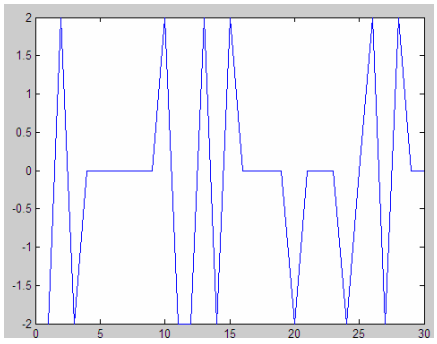
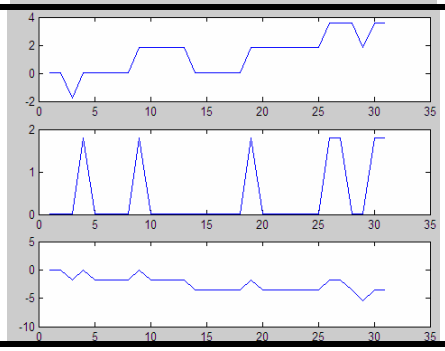
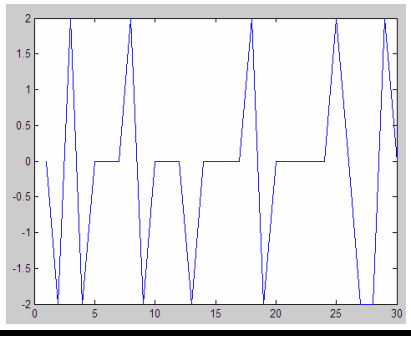
اثر تغییرات K بر روی خروجی :

با افزایش مقدار K شبکه قطعا همگرا خواهد شد

η	K	$0.001 \le Rand \ldots W \le 0.009$	E	*Result*
0.01	*10*			*Not train*
0.01	*20*			*Not train*
0.01	*30*			*Train*
0.01	*40*			*Train*

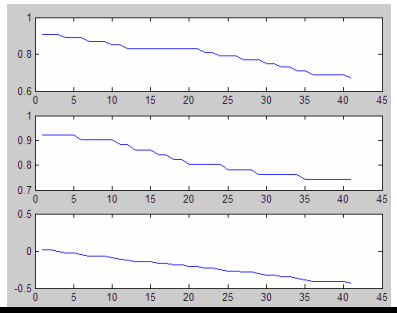
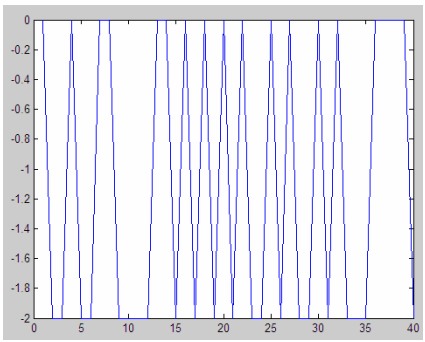
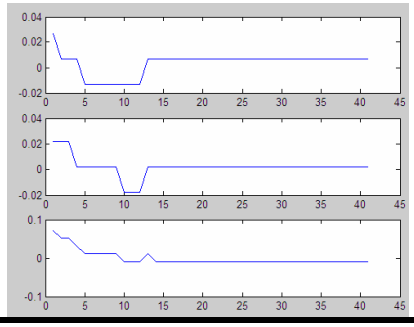
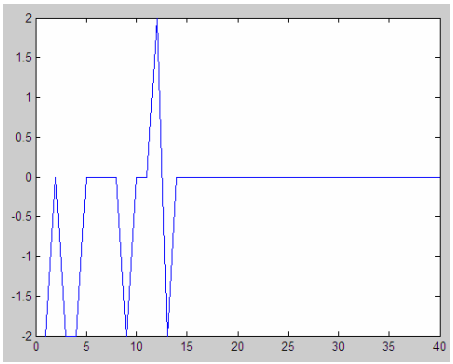
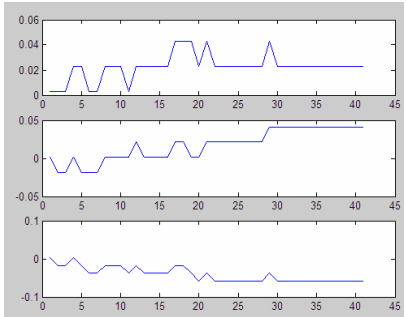
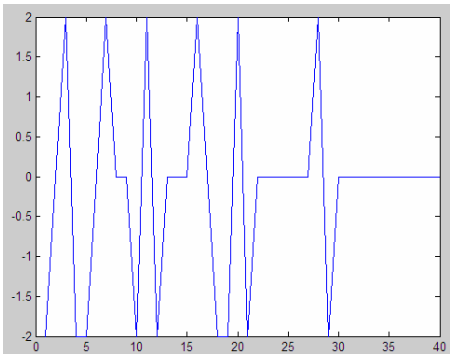
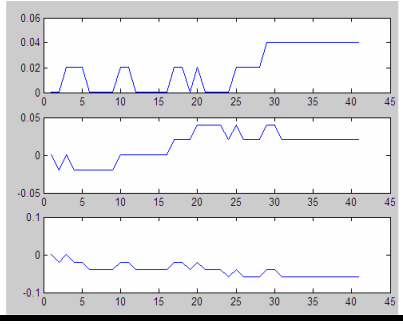
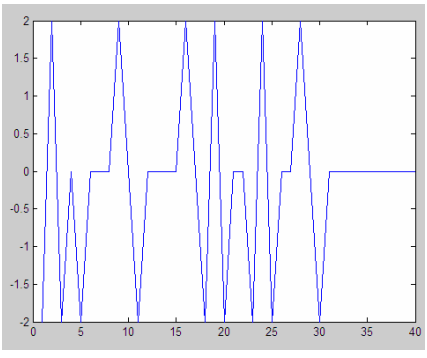
اثر تغییرات η بر روی خروجی :

با افزایش مقدار η شبکه واگرا می شود .

η	K	$0.001 \le Rand \ldots W \le 0.009$	E	*Result*
0.1	*30*			*Train*
0.3	*30*			*Train*
0.7	*30*			*Not train*
0.9	*30*			*Not train*

اثر تغییرات W های اولیه بر روی نتیجه :

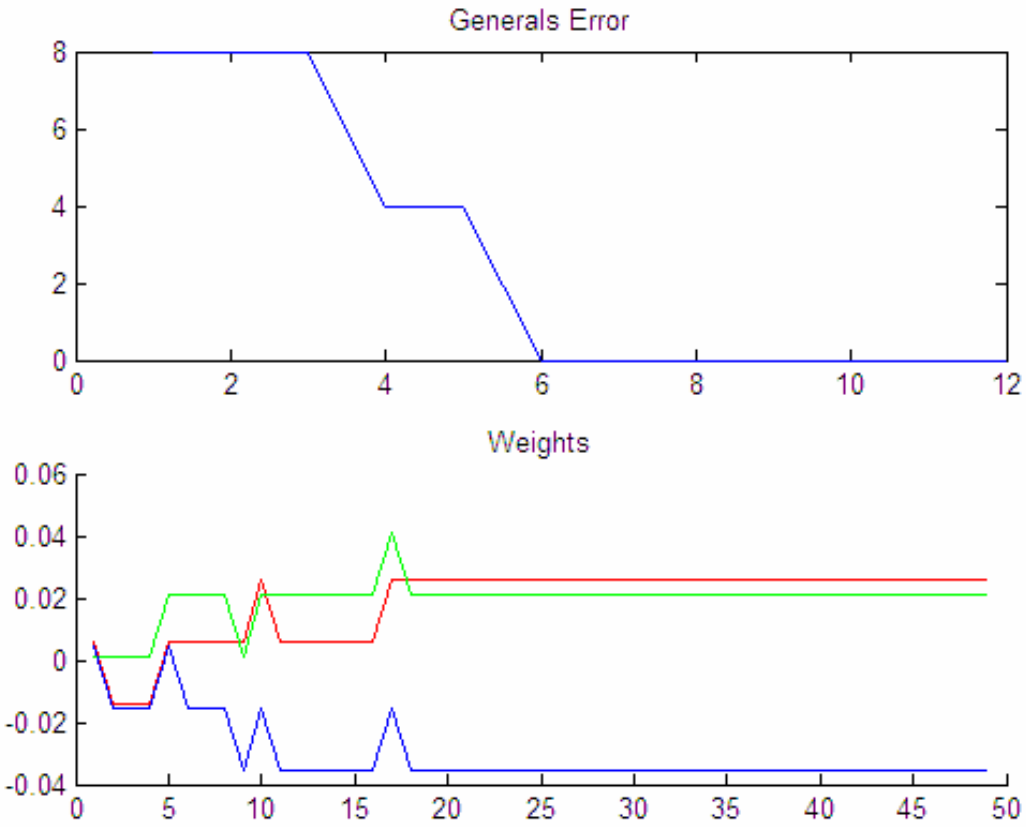
انتخاب مقادیر بزرگ برای وزن های اول باعث می شود که شبکه دیرتر همگرا شود .

η	K	W	E	Result
0.01	40	$0.1 \le Rand \ldots W \le 0.9$ 		Not train
0.01	40	$0.01 \le Rand \ldots W \le 0.09$ 		Train
0.01	40	$0.001 \le Rand \ldots W \le 0.009$ 		Train
0.01	40	$0.0001 \le Rand \ldots W \le 0.0009$ 		Train

کد:

```
close all;
clear all;
%---------------------------------------------- input variable
X=input('enter x ');
T=input('enter t');
K=40;%input('number of replay');
%---------------------------------------------- constant
ETA=0.01;
sizex=size(X);
X(:,sizex(2)+1)=1;
sizex=size(X);
W=zeros(sizex(2),1);
%----------------------------------------------
for i=1:sizex(2)
   W(i,1)=rand(1)/1000;
end
prndnum=0;
rndnum=0;
for step=1:K
   while (rndnum==prndnum)
      rndnum=fix(4* rand(1)+1);
   end
   prndnum=rndnum;
   tmp1(:,1)=W(:,step);
   if (sign(X(rndnum,:)*tmp1))==0
      E(step)=T(1,rndnum)-1;
   else
      tmp=sign(X(rndnum,:)*tmp1);
      E(step)= T(1,rndnum)-tmp;
   end
   DeltaW=(ETA*E(step)).*X(rndnum,:);
   W(:,step+1)=W(:,step)+DeltaW';
end
%---------------------------------------------- for test
%tmp1(:,1)=W(:,step+1);
%for i=1:4
%   sign(X(i,:)*tmp1)
%end
```

۲– شبیه سازی AND با تابع BBF با یادگیری آنلاین به همراه جنرال ارور:

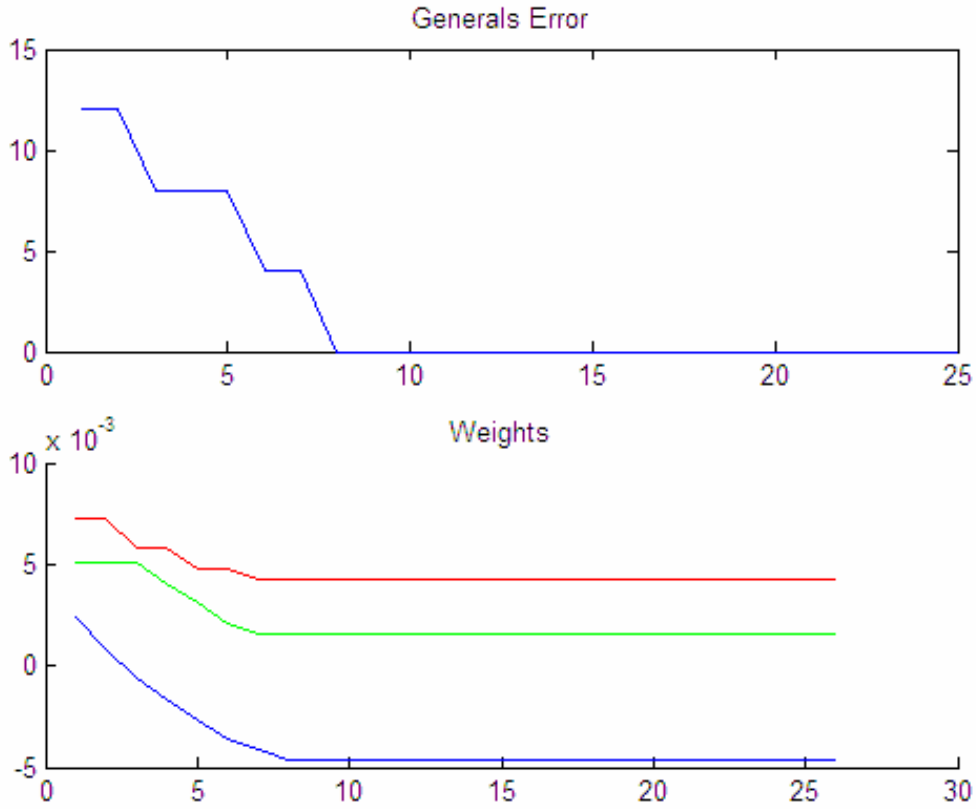


کد:

```
close all;
clear all;
%---------------------------------------------- input variable
X=[0 0;0 1;1 0; 1 1];%input('enter x ');
T=[-1 -1 -1 1];%input('enter t');
K=50;%input('number of replay');
%---------------------------------------------- constant
ETA=0.01;
sizex=size(X);
X(:,sizex(2)+1)=1;
sizex=size(X);
%----------------------------------------------
W=rand(sizex(2),1)/100;
prndnum=0;
rndnum=0;
Xbax(1:sizex(1))=0;
for step1=1:K/sizex(1)
   for step=1:sizex(1)
```

```
    rndnum=fix   (sizex(1)*rand(1)+1);
    while (Xbax(rndnum)~=0)
       rndnum=fix(sizex(1)*rand(1)+1);
    end
    Xbax(rndnum)=1;


    E((step1-1)*sizex(1)+step)=T(rndnum)-sign(X(rndnum,:)*W(:,(step1-
1)*sizex(1)+step));
    DeltaW=(ETA*E((step1-1)*sizex(1)+step)).*X(rndnum,:);
    W(:,(step1-1)*sizex(1)+step+1)=W(:,(step1-1)*sizex(1)+step)+DeltaW';
  end
  Xbax(1:sizex(1))=0;


  E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))=E((step1-1)*sizex(1)+1:(step1-
1)*sizex(1)+sizex(1)).^2;
  Error(step1)=sum((E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))),2);


end
subplot(2,1,1)
plot(Error)
TITLE('Generals Error')
subplot(2,1,2)
TITLE('Weights')
hold on;
plot(W(1,:),'r')
plot(W(2,:),'g')
plot(W(1,:),'b')
```

۳- شبیه سازی AND با تابع BBF با یادگیری $\sum e$:



توضیحات :

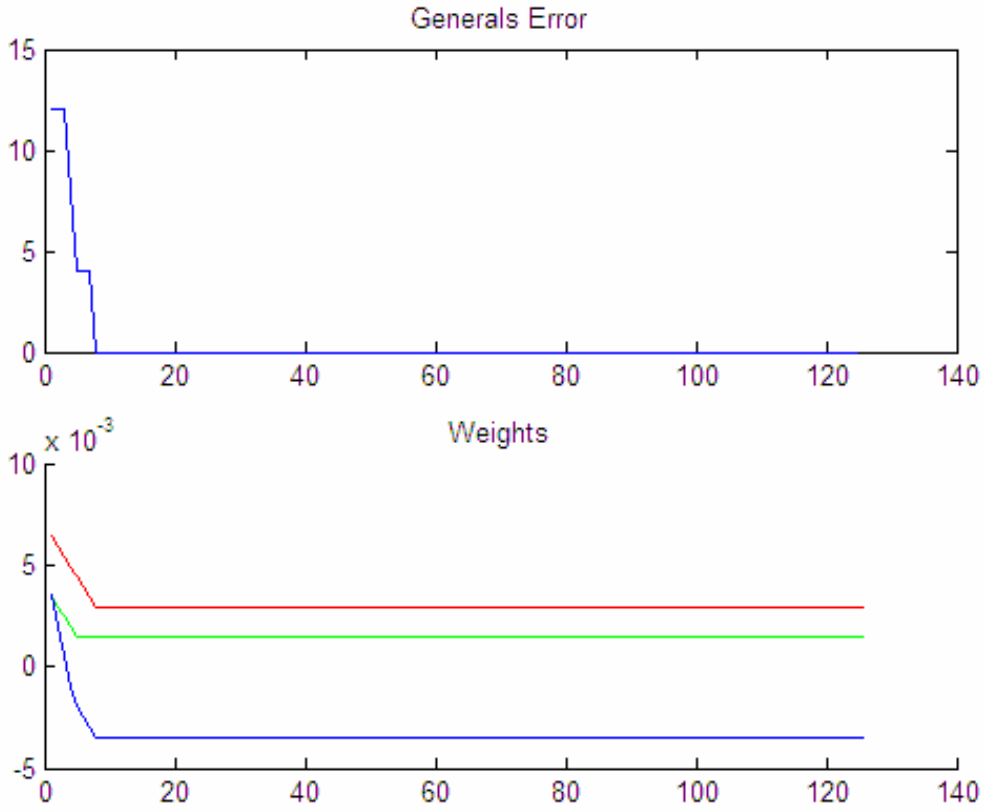
در این روش با توجه به اینکه مقدار خطا برای بروز شدن وزن ها از جمع خطای تمام ورودی ها می باشد ، ممکن است حالاتی پیش آید که برای ۲ خروجی خطای متقارن ایجاد شود . مثلا ۲ و ۲- . در این حالت جمع خطا ها صفر خواهد شد در صورتی که شبکه همگرا نشده است . بنابراین این روش همواره همگرا نخواهد شد.

کد :

```
close all;
clear all;
%---------------------------------------------- input variable
X=[0 0;0 1;1 0; 1 1];%input('enter x ');
T=[-1 -1 -1 1];%input('enter t');
K=100;%input('number of replay');
%---------------------------------------------- constant
ETA=0.001;
sizex=size(X);
X(:,sizex(2)+1)=1;
```

```
sizex=size(X);
%-----------------------------------------
W=rand(sizex(2),1)/100;
rndnum=0;
Xbax(1:sizex(1))=0;
for step1=1:K/sizex(1)
   for step=1:sizex(1)
      rndnum=fix   (sizex(1)*rand(1)+1);
      while (Xbax(rndnum)~=0)
         rndnum=fix(sizex(1)*rand(1)+1);
      end
      Xbax(rndnum)=1;
      E((step1-1)*sizex(1)+step)=T(rndnum)-sign(X(rndnum,:)*W(:,step1));
   end
   Xbax(1:sizex(1))=0;


   Error(step1)=(sum(((E((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1)))),2))/sizex(1);


   DeltaW=ETA*Error(step1).*X(rndnum,:);
   W(:,step1+1)=W(:,step1)+DeltaW';


   E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))=E((step1-1)*sizex(1)+1:(step1-
1)*sizex(1)+sizex(1)).^2;
   Error1(step1)=sum((E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))),2);


end
subplot(2,1,1)
plot(Error1)
TITLE('Generals Error')
subplot(2,1,2)
TITLE('Weights')
hold on;
plot(W(1,:),'r')
plot(W(2,:),'g')
plot(W(3,:),'b')
```

۴- شبیه سازی AND با تابع BBF با یادگیری آفلاین :



توضیحات :

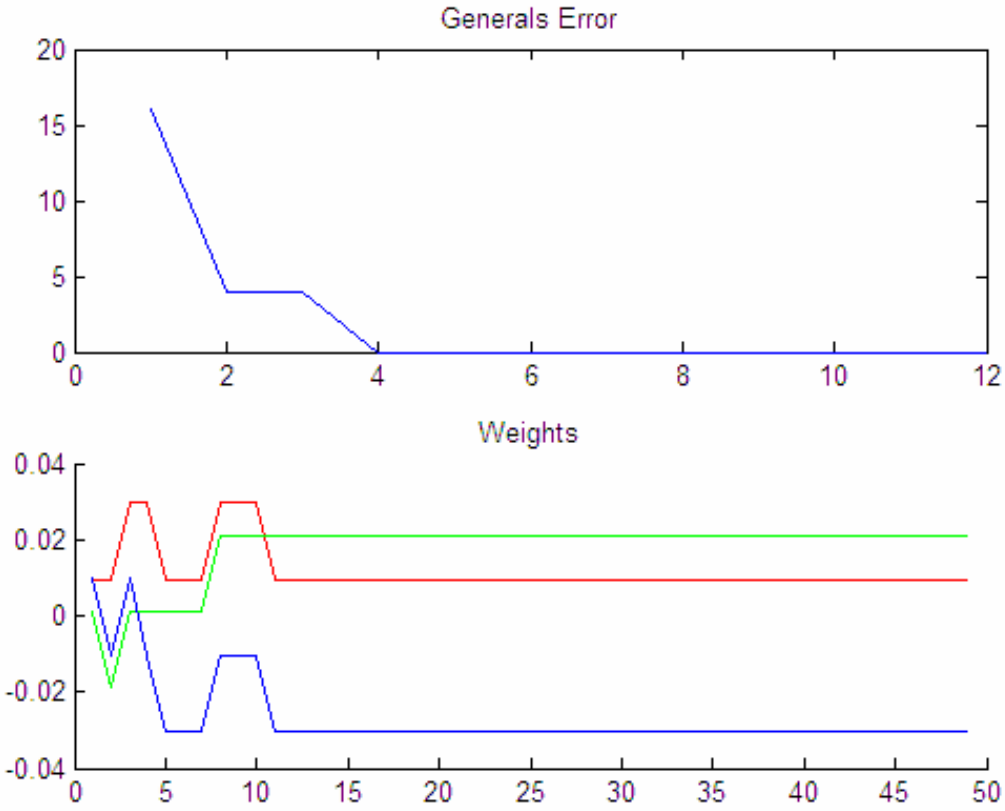
در این روش بعد از اعمال داده به شبکه مقدار خطا محاسبه می شود ولی به وزن های قبلی اعمال نمی شود . و زمانی که تمامی داده های ورودی داده شد ، از مقادیر خطا میانگین گرفته شده و یکباره به وزن ها اعمال می شود . همانطور که مشاهده می شود ، این روش برای داده های ورودی کم مانند روش آنلاین است ولی احتمالا برای داده های ورودی زیاد ، زمان همگرا شدن شبکه مدت زمان بیشتری طول خواهد کشید .

کد :

```
close all;
clear all;
%---------------------------------------------- input variable
X=[0 0;0 1;1 0; 1 1];%input('enter x ');
T=[-1 -1 -1 1];%input('enter t');
K=500;%input('number of replay');
%---------------------------------------------- constant
ETA=0.001;
sizex=size(X);
X(:,sizex(2)+1)=1;
```

```
sizex=size(X);
%---------------------------------------------
W=rand(sizex(2),1)/100;
prndnum=0;
rndnum=0;
Xbax(1:sizex(1))=0;
for step1=1:K/sizex(1)
   for step=1:sizex(1)
      rndnum=fix(sizex(1)*rand(1)+1);
      while (Xbax(rndnum)~=0)
         rndnum=fix(sizex(1)*rand(1)+1);
      end
      Xbax(rndnum)=1;
      E((step1-1)*sizex(1)+step)=T(rndnum)-sign(X(rndnum,:)*W(:,step1));
      DeltaW(step,:)=(ETA*E((step1-1)*sizex(1)+step)).*X(rndnum,:);
   end
   Xbax(1:sizex(1))=0;
   Delta=mean(DeltaW);
   W(:,step1+1)=W(:,step1)+Delta';


   E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))=E((step1-1)*sizex(1)+1:(step1-
1)*sizex(1)+sizex(1)).^2;
   Error1(step1)=sum((E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))),2);


end


subplot(2,1,1)
plot(Error1)
TITLE('Generals Error')
subplot(2,1,2)
TITLE('Weights')
hold on;
plot(W(1,:),'r')
plot(W(2,:),'g')
plot(W(3,:),'b')
```

۵– شبیه سازی AND با تابع BSF با یادگیری آنلاین :



توضیحات :

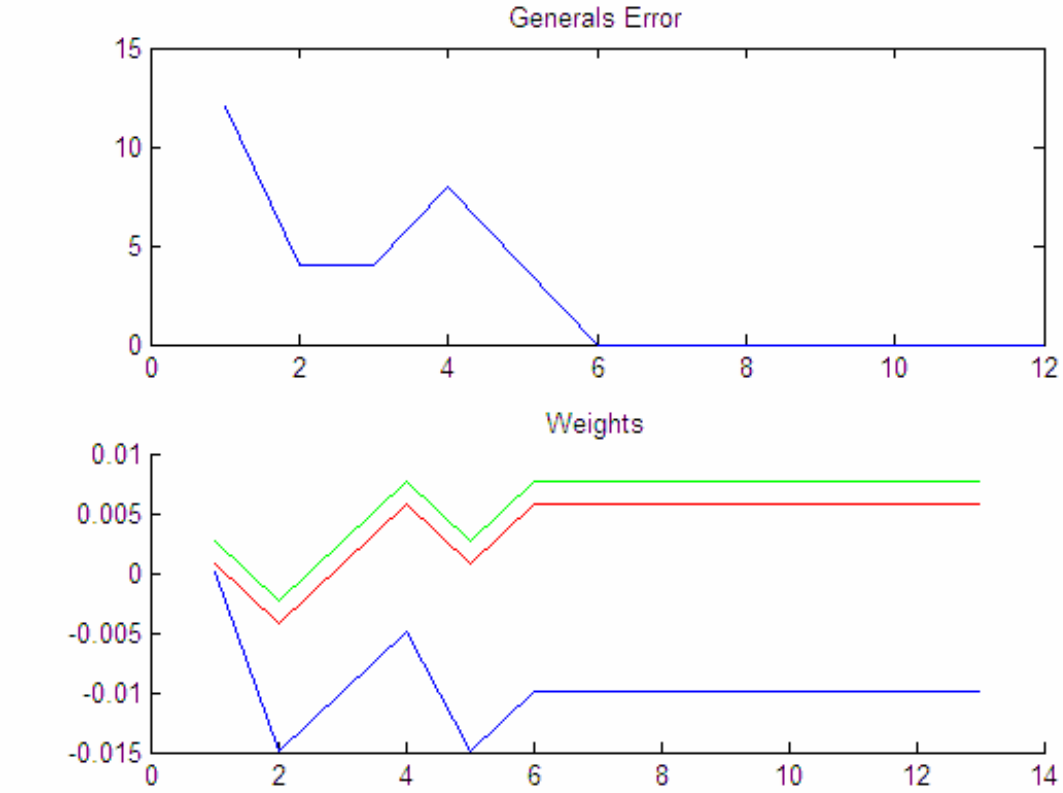
در این روش به جای استفاده از تابع BBF از تابع BSF استفاده شده است . مهمترین تفاوت این ۲ تابع این است که اولی مشتق ناپذیر است ولی دومی مشتق پذیر . این خاصیت سبب می شود که فرمول بدست آوردن خطای وزن ها بتوان از مشتق تابع نیز بهره جست که این کار دقت را بالا می برد .

کد :

```
close all;
clear all;
%---------------------------------------------- input variable
X=[0 0;0 1;1 0; 1 1];%input('enter x ');
T=[-1 -1 -1 1];%input('enter t');
K=80;%input('number of replay');
%---------------------------------------------- constant
ETA=0.01;
sizex=size(X);
X(:,sizex(2)+1)=1;
sizex=size(X);
%----------------------------------------------
```

```
W=rand(sizex(2),1)/100;
prndnum=0;
rndnum=0;
Xbax(1:sizex(1))=0;
for step1=1:K/sizex(1)
   for step=1:sizex(1)
      rndnum=fix(sizex(1)*rand(1)+1);
      while (Xbax(rndnum)~=0)
         rndnum=fix(sizex(1)*rand(1)+1);
      end
      Xbax(rndnum)=1;


      E((step1-1)*sizex(1)+step)=T(rndnum)-sign(BSF(X(rndnum,:)*W(:,(step1-1)*sizex(1)+step)));
      DeltaW=(ETA*E((step1-1)*sizex(1)+step)*dBSF(X(rndnum,:)*W(:,(step1-
1)*sizex(1)+step))).*X(rndnum,:);
      W(:,(step1-1)*sizex(1)+step+1)=W(:,(step1-1)*sizex(1)+step)+DeltaW';
   end
   Xbax(1:sizex(1))=0;


   E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))=E((step1-1)*sizex(1)+1:(step1-
1)*sizex(1)+sizex(1)).^2;
   Error1(step1)=sum((E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))),2);


end


subplot(2,1,1)
plot(Error1)
TITLE('Generals Error')
subplot(2,1,2)
TITLE('Weights')
hold on;
plot(W(1,:),'r')
plot(W(2,:),'g')
plot(W(3,:),'b')

function [F]=BSF(U)
F=(1-exp(-2*U))./(1+exp(-2*U));

function [dF]=dBSF(U)
dF=(4*exp(-2*U))./((1+exp(-2*U)).^2);
```

۶- شبیه سازی AND با تابع BSF با یادگیری آفلاین :



توضیحات :

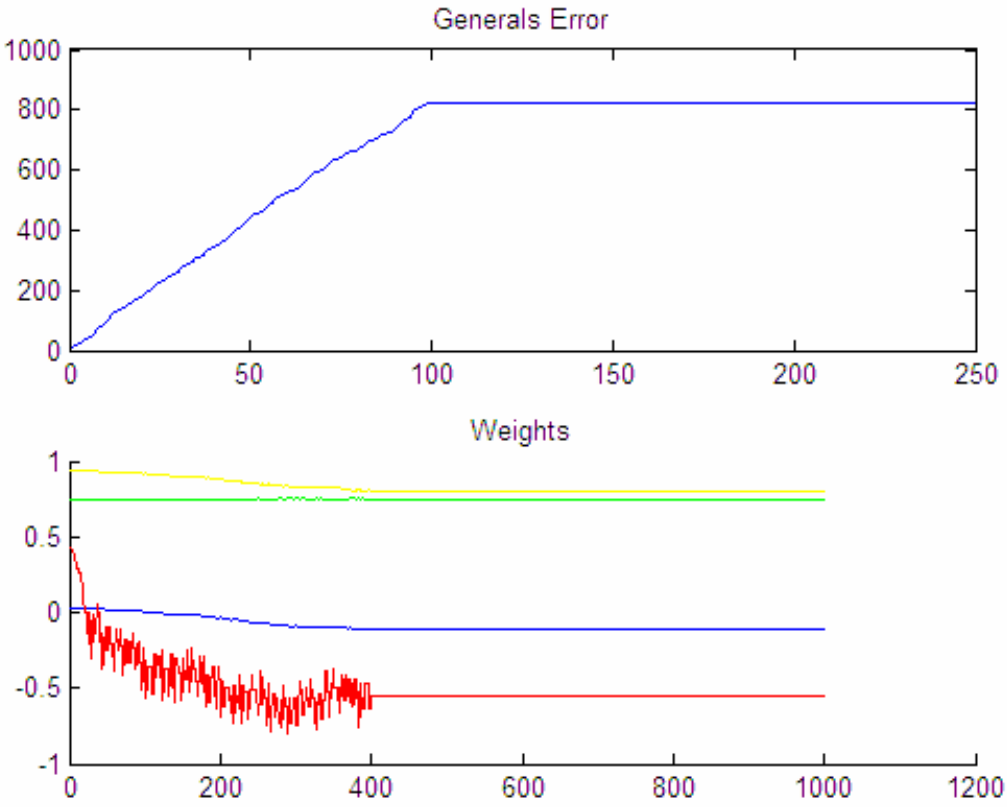
این تمامی مشخصات این شبکه مانند شبکه بالا می باشد ولی این شبکه به دلیل اینکه وزن ها به صورت آفلاین آپدیت می شوند ، دیر تر همگرا شده است .

کد :

```
close all;
clear all;
%---------------------------------------------- input variable
X=[0 0;0 1;1 0; 1 1];%input('enter x ');
T=[-1 -1 -1 1];%input('enter t');
K=50;%input('number of replay');
%---------------------------------------------- constant
ETA=0.01;
sizex=size(X);
X(:,sizex(2)+1)=1;
sizex=size(X);
%----------------------------------------------
```

```
W=rand(sizex(2),1)/100;
rndnum=0;
Xbax(1:sizex(1))=0;
for step1=1:K/sizex(1)
   for step=1:sizex(1)
      rndnum=fix(sizex(1)*rand(1)+1);
      while (Xbax(rndnum)~=0)
         rndnum=fix(sizex(1)*rand(1)+1);
      end
      Xbax(rndnum)=1;
      E((step1-1)*sizex(1)+step)=T(rndnum)-sign(BSF(X(rndnum,:)*W(:,step1)));
      DeltaW(step,:)=(ETA*E((step1-
1)*sizex(1)+step)*dBSF(X(rndnum,:)*W(:,step1))).*X(rndnum,:);
   end
   Xbax(1:sizex(1))=0;


   Delta=mean(DeltaW);
   W(:,step1+1)=W(:,step1)+Delta';


   E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))=E((step1-1)*sizex(1)+1:(step1-
1)*sizex(1)+sizex(1)).^2;
   Error1(step1)=sum((E1((step1-1)*sizex(1)+1:(step1-1)*sizex(1)+sizex(1))),2);
end
subplot(2,1,1)
plot(Error1)
TITLE('Generals Error')
subplot(2,1,2)
TITLE('Weights')
hold on;
plot(W(1,:),'r')
plot(W(2,:),'g')
plot(W(3,:),'b')

function [F]=BSF(U)
F=(1-exp(-2*U))./(1+exp(-2*U));

function [dF]=dBSF(U)
dF=(4*exp(-2*U))./((1+exp(-2*U)).^2);
```

۷- شبیه سازی XOR با تابع BSF با یادگیری آنلاین :



توضیحات :

در این روش برای اینکه مشتق تابع BSF به اشباع نرود ، باید وزن ها رو بزرگ انتخاب کرد . همانطور که در شکل بالا ملاحظه می کنید . مقادیری که وزن ها به آن همگرا شده اند بعضا بزرگتر از ۱ هم می باشند . نکته قابل توجه این است که در این برنامه ضریب فراموشی لحاظ نشده است . بنابراین همیشه شبکه همگرا نمی شود نکته دیگری که باید به آن توجه شود مقادیر اتا ورودی و خروجی می باشد . مقدار اتا باید در حد صدم باشد و همچنین اتای خروجی باید ۱۰ برابر بزرگتر از اتای ورودی باشد . اگه هر ۲ اتا با هم برابر باشند تغییرات وزنهای شبکه بسیار زیاد خواهد بود و شبکه همگرا نمی شود .

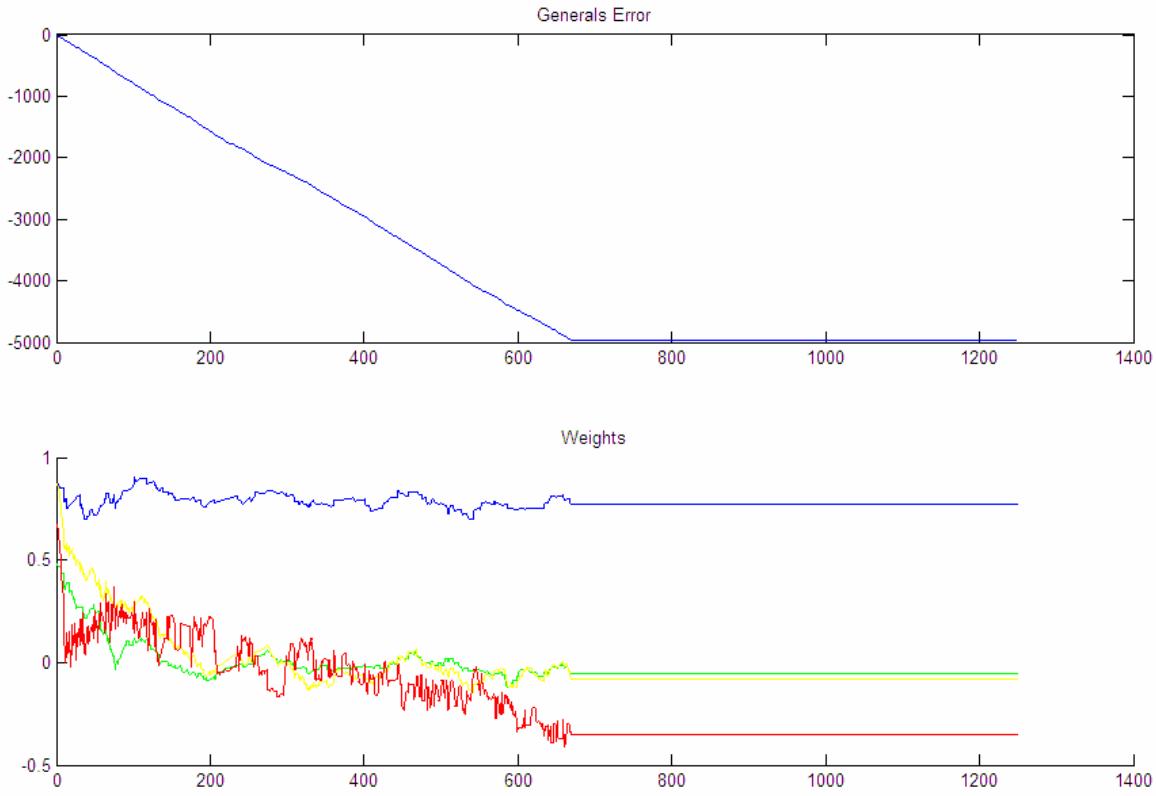
کد :

```
clear all;
close all;
%function [WI]=newNN(D,P,T)
%---------------------------------------------- input variable
D=[6 1];
P=[0 0;0 1;1 0; 1 1];%input('enter x ');
```

```
T=[-1 1 1 -1];%input('enter t');
K=1000;%input('number of replay');
%---------------------------------------------- constant
ETAO=0.1;
ETAI=0.01;
sizep=size(P);
P(:,sizep(2)+1)=1;
sizep=size(P);
%----------------------------------------------
WI=rand(sizep(2),D(1),1)/1;
WO=rand(D(1),D(2),1)/1;
Error(1)=0
rndnum=0;
Pbax(1:sizep(1))=0;
for step1=1:K/sizep(1)
   for step=1:sizep(1)
      rndnum=fix(sizep(1)*rand(1)+1);
      while (Pbax(rndnum)~=0)
         rndnum=fix(sizep(1)*rand(1)+1);
      end
      Pbax(rndnum)=1;
      Y1(:,(step1-1)*sizep(1)+step)=(BSF(P(rndnum,:)*WI(:,:,(step1-1)*sizep(1)+step)))';
      try
         Yout(:,(step1-1)*sizep(1)+step)=(sign(BSF(Y1(:,(step1-1)*sizep(1)+step)'*WO(:,:,(step1-
1)*sizep(1)+step))))';
      catch
         Yout(:,(step1-1)*sizep(1)+step)=(sign(BSF(Y1(:,(step1-1)*sizep(1)+step)'*WO(:,(step1-
1)*sizep(1)+step))))';
      end
      E(:,(step1-1)*sizep(1)+step)=T(:,rndnum)-Yout(:,(step1-1)*sizep(1)+step);


      GAMA=E(:,(step1-1)*sizep(1)+step)*(dbsf(Y1(:,(step1-1)*sizep(1)+step)'*WO(:,(step1-
1)*sizep(1)+step)));
      GAMA1=GAMA;


      GAMA=GAMA*(dbsf(P(rndnum,:)*WI(:,:,(step1-1)*sizep(1)+step)))'.*WO(:,(step1-
1)*sizep(1)+step);


      WI(:,:,(step1-1)*sizep(1)+step+1)=WI(:,:,(step1-
1)*sizep(1)+step)+(ETAI*GAMA*P(rndnum,:))';
      WO(:,(step1-1)*sizep(1)+step+1)=WO(:,(step1-
1)*sizep(1)+step)+ETAO*GAMA1*Y1(:,(step1-1)*sizep(1)+step);
   end
   Pbax(1:sizep(1))=0;
```

```
   E1((step1-1)*sizep(1)+1:(step1-1)*sizep(1)+sizep(1))=E((step1-1)*sizep(1)+1:(step1-
1)*sizep(1)+sizep(1)).^2;
   Error1(step1)=sum((E1((step1-1)*sizep(1)+1:(step1-1)*sizep(1)+sizep(1))),2);
   try
      Error(step1)=Error(step1-1)+Error1(step1);
   catch
      Error(step1)=Error1(step1);
   end


end
%Error=-Error;
hold on;


subplot(2,1,1)
plot(Error)
TITLE('Generals Error')
subplot(2,1,2)
TITLE('Weights')
hold on;
tmp(:,:)=WI(1,1,:);
plot(tmp,'g')
tmp(:,:)=WI(2,1,:);
plot(tmp)
tmp(:,:)=WI(3,1,:);
plot(tmp,'y')


plot(WO(1,:),'r')

function [F]=BSF(U)
F=(1-exp(-2*U))./(1+exp(-2*U));

function [dF]=dBSF(U)
dF=(4*exp(-2*U))./((1+exp(-2*U)).^2);
```

۸- شبیه سازی XOR با تابع BSF با یادگیری آفلاین :



توضیحات :

با توجه به شکل در این حالت شبکه دیر تر از حالت آنلاین همگرا شده است که نتیجه ای دور از انتظار نیست.

کد :

```
clear all;
close all;
%function [WI]=newNN(D,P,T)
%---------------------------------------------- input variable
D=[6 1];
P=[0 0;0 1;1 0; 1 1];%input('enter x ');
T=[-1 1 1 -1];%input('enter t');
K=10000;%input('number of replay');
%---------------------------------------------- constant
ETAO=0.5;
ETAI=0.05;
sizep=size(P);
P(:,sizep(2)+1)=1;
sizep=size(P);
%----------------------------------------------
WI=rand(sizep(2),D(1),1)/1;
```

```
WO=rand(D(1),D(2),1)/1;
Error(1)=0;
rndnum=0;
Pbax(1:sizep(1))=0;
for step1=1:K/sizep(1)
   for step=1:sizep(1)
      rndnum=fix(sizep(1)*rand(1)+1);
      while (Pbax(rndnum)~=0)
         rndnum=fix(sizep(1)*rand(1)+1);
      end
      Pbax(rndnum)=1;
      Y1(:,(step1-1)*sizep(1)+step)=(BSF(P(rndnum,:)*WI(:,:,step1)))';
      try
         Yout(:,(step1-1)*sizep(1)+step)=(sign(BSF(Y1(:,(step1-1)*sizep(1)+step)'*WO(:,:,step1))))';
      catch
         Yout(:,(step1-1)*sizep(1)+step)=(sign(BSF(Y1(:,(step1-1)*sizep(1)+step)'*WO(:,step1))))';
      end
      E(:,(step1-1)*sizep(1)+step)=T(:,rndnum)-Yout(:,(step1-1)*sizep(1)+step);


      GAMA(:,step)=E(:,(step1-1)*sizep(1)+step)*(dbsf(Y1(:,(step1-
1)*sizep(1)+step)'*WO(:,step1)));
      GAMA1(:,step)=GAMA(:,step)*(dbsf(P(rndnum,:)*WI(:,:,step1)))'.*WO(:,step1);


   end
%-----------------------------------------------
   GAMA=mean(GAMA,2);
   GAMA1=mean(GAMA1,2)
   WI(:,:,step1+1)=WI(:,:,step1)+(ETAI*GAMA1*P(rndnum,:))';
   WO(:,step1+1)=WO(:,step1)+ETAO*GAMA*Y1(:,(step1-1)*sizep(1)+step);
%-----------------------------------------------
   Pbax(1:sizep(1))=0;
%-----------------------------------------------
   E1((step1-1)*sizep(1)+1:(step1-1)*sizep(1)+sizep(1))=E((step1-1)*sizep(1)+1:(step1-
1)*sizep(1)+sizep(1)).^2;
   Error1(step1)=sum((E1((step1-1)*sizep(1)+1:(step1-1)*sizep(1)+sizep(1))),2);
   try
      Error(step1)=Error(step1-1)+Error1(step1);
   catch
      Error(step1)=Error1(step1);
   end


end


hold on;
```

```
subplot(2,1,1)
plot(Error)
TITLE('Generals Error')
subplot(2,1,2)
TITLE('Weights')
hold on;
tmp(:,:)=WI(1,1,:)
plot(tmp,'g')
tmp(:,:)=WI(2,1,:)
plot(tmp)
tmp(:,:)=WI(3,1,:)
plot(tmp,'y')


plot(WO(1,:),'r')
function [F]=BSF(U)
F=(1-exp(-2*U))./(1+exp(-2*U));

function [dF]=dBSF(U)
dF=(4*exp(-2*U))./((1+exp(-2*U)).^2);
```