```
function [RMSE,yL,TRAINEDNET,minRMSE,yf,yfL,yf2]=annearly(y,maxlag,...
    nhiden,trset,HPF,lr);
%-------------------------------------------------------------------------
%This M-file forecasts y with minimum RMSE network.(early stopping)

% Outputs:

% - RMSE , root mean squares error.
% - yL, matrix of y's lags.
% - TRAINEDNET , a NET with minimum mean squares error.
% - minRMSE , minimum of root mean squares error.
% - yf , forecast of y.

% Inputs:

% - y , a time series in vertical vector form.(eg stock prices)
% - maxlag ,maximume lag that should be entered in model.(eg 5)
% - nhiden, number of hidden layer units.(eg 5)
% - trset,percent of observations for trainig set.(eg 80)
% - HPF ,number of priods that  should be forecasted.(eg 10)
% - lr , Learning rate.(eg  0.1)

% Written by Shapour Mohammadi and Hossein Abbasi-Nejad,2005-1
% FACULTY OF ECONOMICS
% UNIVERSITY OF TEHRAN
% shmohammadi@gmail.com
%-------------------------------------------------------------------------
% Building networks with different number of hiden units and input lags.
for lay=1:nhiden
    for lag=1:maxlag
    PR(1:maxlag,1)=[-1];
    PR(1:maxlag,2)=[1];
net(lay,lag)={newff(PR(1:lag,:),[lay 1],{'tansig','purelin'},'trainlm')};
net{lay,lag}.trainParam.lr = lr;
net{lay,lag}.trainParam.lr_inc = 1.05;
net{lay,lag}.trainParam.lr_dec = .7;
net{lay,lag}.trainparam.epochs=1000;
net{lay,lag}.trainparam.show=100;
net{lay,lag}.trainparam.goal=1e-5;

end
end

%-------------------------------------------------------------------------
%defining lags for y

y=y';

[ny,mny,mxy]=premnmx(y);
[nyr,nyc]=size(y);
trset=floor(nyc*trset/100)

for lag=1:maxlag
yL=zeros(lag,nyc);
end
for s1=1:nyc
```

```
    for s2=1:maxlag
yL(s2,s1)=NaN;
end
end
for lag=1:maxlag
yL(lag,1+lag:nyc)=ny(1:nyc-lag);
end
%-------------------------------------------------------------------------
%training nets

for lay=1:nhiden
    for lag=1:maxlag
        %early stopping
        val.P=yL(1:lag,trset+1:nyc);
        val.T=ny(1,trset+1:nyc);

[TRAINEDNET{lay,lag},tr]=train(net{lay,lag},yL(1:lag,lag+1:trset),ny(1,lag+1:trset)...
    ,[],[],val);
%RMSE calculation
yhat=sim(TRAINEDNET{lay,lag},yL(1:lag,trset+1:nyc));
rmse=((sum((yhat-ny(1,trset+1:nyc)).^2))/nyc)^.5;
RMSE(lay,lag)=rmse;

 RMSEE=RMSE;
 minRMSE=min(min(RMSE));
end
end
for lag=1:maxlag
for lay=1:nhiden

if RMSEE(lay,lag)==min(min(RMSE))
    optlay=lay
    optlag=lag
end
end
end
yfL=yL(1:optlag,:);
%-------------------------------------------------------------------------
%training again the optimal net with complete set of data.
TRAINEDNET{optlay,optlag}=train(TRAINEDNET{optlay,optlag},yL(1:optlag,:),ny(1,:))
%H period forecasting
for o=1:HPF
yf=sim(TRAINEDNET{optlay,optlag},yfL(1:optlag,nyc-1:o+nyc-1));
yf=[ny yf];
for flag=1:optlag
yfL(flag,1+flag:nyc+o)=yf(1:o+nyc-flag);
end
end
yf2=sim(TRAINEDNET{optlay,optlag},yL(1:optlag,optlag+1:nyc));
yf=yf';
yf2=yf2';
%converting the data back into unnormalized units.
yf=postmnmx(yf,mny,mxy);
yf2=postmnmx(yf2,mny,mxy);
yhatopt=sim(TRAINEDNET{optlay,optlag},yL(1:optlag,trset+1:nyc));
 yy(1:nyc-trset)=ny(trset+1:nyc);
```

```
 error=(yhatopt./yy-1);
 subplot(2,1,1);
 plot([1:nyc-trset],yhatopt,'g-',[1:nyc-trset],yy,'r-');
 legend('yhatopt','yy',0);
 ylabel('value');
 title(['Frecasted by Optimal Net']);

 subplot(2,1,2);
plot([1:nyc-trset],error,'b-')
xlabel('Time');
ylabel('Residulas');
title(['Forecast Error for Test Set']);
%END
```