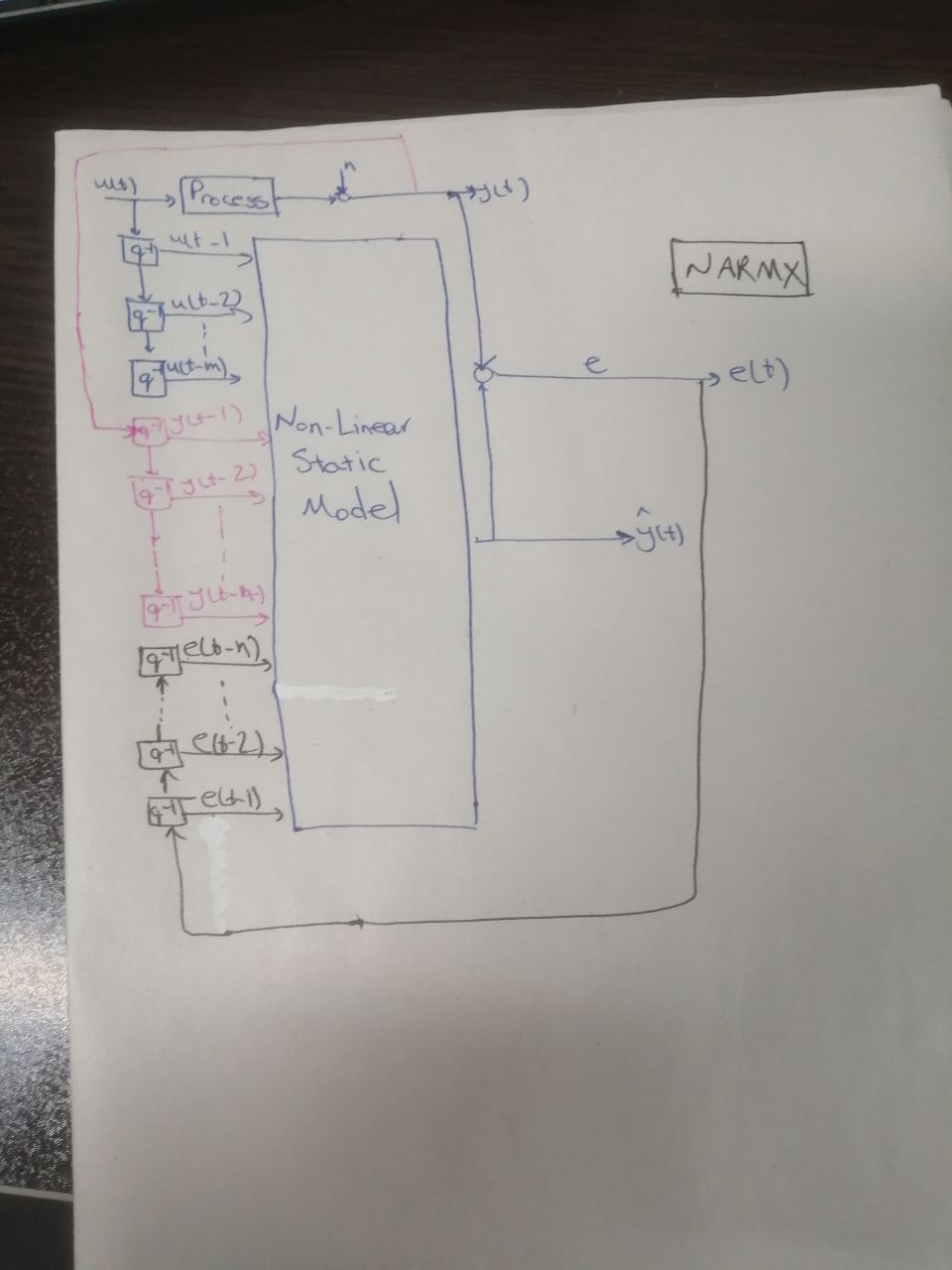
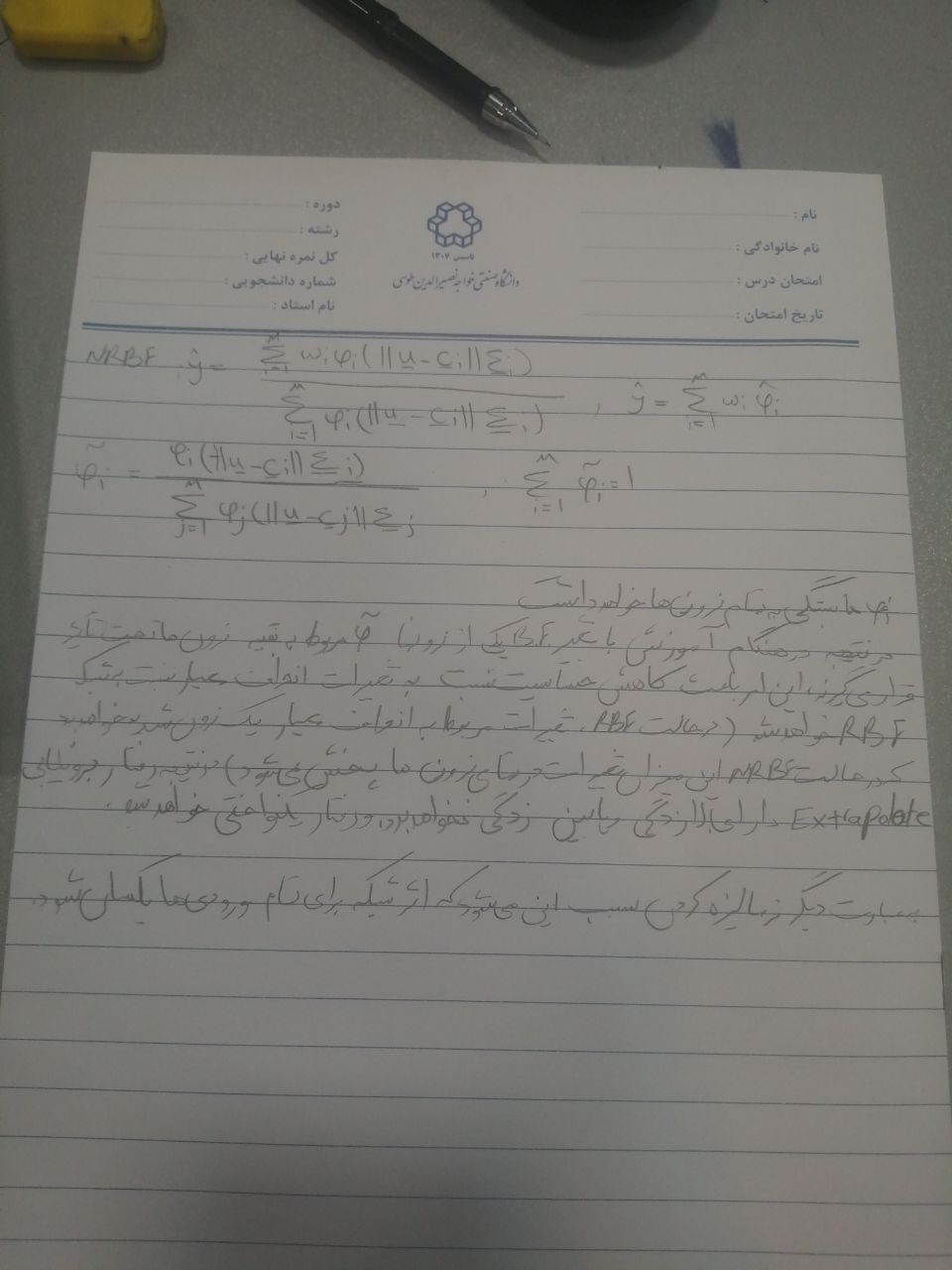
**Mohamad Alikhani**

**40109494**



**1 هماهنگ نشده**



نرمالیزه کردن داده‌ها به معنای تبدیل آن‌ها به یک مقیاس مشخص است که می‌تواند باعث بهبود عملکرد شبکه شود. زمانی که از RBF در لایه‌های میانی استفاده می‌شود، نرمالیزه کردن داده‌ها می‌تواند باعث شود که تفاوت‌های اندازه‌گیری بین ورودی‌ها کاهش یابد و شبکه بهتر بتواند الگوهای پیچیده‌تری را یاد بگیرد.

بنابراین، در ساختارهایی مانند RBF، نرمالیزه کردن در لایه‌های میانی متداول است زیرا می‌تواند بهبود عملکرد شبکه و یادگیری الگوهای پیچیده‌تر کمک کند.

عمل normalize کردن تفسیرپذیری را کم می‌کند اما در مقابل، باعث می‌شود که داده‌ها در بازه یکسان قرار بگیرند. این عمل زمانی مهم می‌شود که داده‌ها در بازه‌های غیریکسان قرار داشته باشند و با عمل نرمالیزاسیون، داده‌ها در بازه و ارزش یکسان قرار بگیرند و شناسایی برای رگرسورهای مختلف به صورت یکسان صورت بگیرد. در غیر این صورت اگر داده‌ها در بازه‌های مختلف باش ند مشکلات زیادی ایجاد می‌شود.

**2 هماهنگ نشده**

برای ادغام دو مدل محلی در ساختار LLM و روش آموزشی LoLiMoT، می‌توان از روش‌های ترکیبی مانند ادغام مدل‌ها (model ensemble) و یا ادغام نتایج (result fusion) استفاده کرد. در ادغام مدل‌ها، دو مدل محلی به صورت موازی آموزش داده می‌شوند و سپس خروجی‌های آن‌ها توسط یک مدل مرکزی ترکیب می‌شوند. در ادغام نتایج، خروجی‌های دو مدل محلی به صورت موازی تولید می‌شوند و سپس توسط یک الگوریتم ترکیبی (مانند میانگین وزنی یا رای‌گیری) ترکیب می‌شوند.

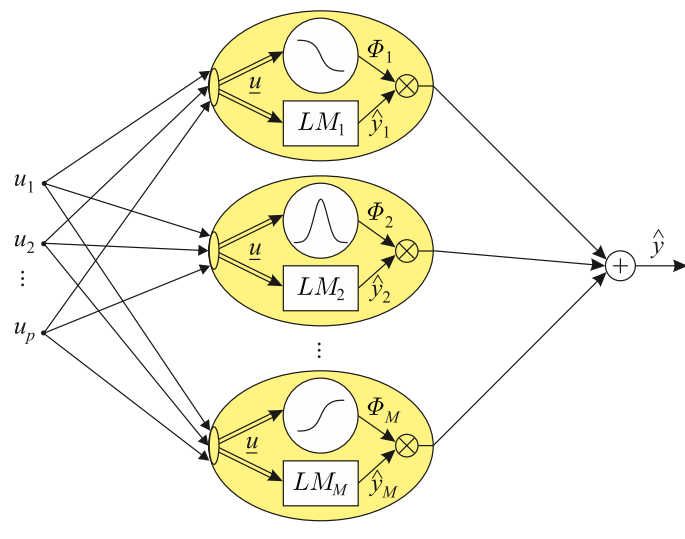
برای یک سیستم دو ورودی – یک خروجی، می‌توان از دو مدل محلی استفاده کرد که هر کدام از ورودی‌ها را به صورت جداگانه پردازش کنند و سپس خروجی‌های آن‌ها ترکیب شوند. به عنوان مثال، در یک سیستم تشخیص اشیاء، یک مدل محلی می‌تواند تصاویر را پردازش کرده و دیگری می‌تواند داده‌های سنسوری دیگر (مانند داده‌های صوتی یا لرزه) را پردازش کند. سپس خروجی‌های این دو مدل محلی توسط یک الگوریتم ترکیبی مانند میانگین وزنی یا رای‌گیری ترکیب می‌شوند تا خروجی نهایی سیستم تولید شود.

شرایط لازم برای ادغام دو مدل محلی شامل هماهنگی و توافق بر روی فرایند ترکیب خروجی‌ها، انطباق و همخوانی ورودی‌ها با مدل‌های محلی و تعیین وزن‌های مناسب برای ترکیب خروجی‌ها می‌باشد.

پیشنهاد ما برای یک سیستم دو ورودی – یک خروجی این است که از دو مدل محلی متفاوت با روش‌های آموزش متنوع استفاده شود تا اطمینان حاصل شود که تنها یک مدل محلی نیست که بهینه‌ترین خروجی را تولید می‌کند. سپس با استفاده از روش ادغام مناسب، خروجی‌های این دو مدل ترکیب شوند تا خروجی نهایی سیستم تولید شود.

در مدل local هر نتورک از دو بخش تشکیل شده است. اول LM و بخش دوم یک تابع برای validation است. پس یک خروجی y\_hat با دو ورودی u=[u1;u2] یک superposition از Mتا مدل وزن دار است.( M تعداد نورون هاست).

فانکشن validity قانون ها را ارائه داده و هر کدام از مدل های local این قانون ها را برای بدست آوردن Yhat اجرا میکنند. برای transition مدل های محلی فانکشن ها بین 0 و 1 هستند.



**3 هماهنگ نشده**

The cost function:

Second power coefficient:

The BPTT (Backpropagation through time) depends on the responses of the past samples, in another word, it is recursive. And the final equation for the algorithm is:

And we continue upon reaching:

As is shown, “Backpropagation Through Time” happens, and updating parameters requires the previous steps’ values. This happens in recurrent networks, and we do the propagation for a few steps back and ignore the rest of the steps. This method of countering this issue is called truncated BPTT.

**سوال 4 هماهنگ نشده**

Contributed by:

Jairo Espinosa

ESAT-SISTA KULEUVEN

Kardinaal Mercierlaan 94

B-3001 Heverlee Belgium

jairo.espinosa@esat.kuleuven.ac.be

Description:

The data comes from a model of a Steam Generator at Abbott Power Plant in Champaign IL.

The model is described in the paper of Pellegrineti [1].

Sampling:

3 sec

Number:

9600

Inputs:

u1: Fuel scaled 0-1

u2: Air scaled 0-1

u3: Reference level inches

u4: Disturbance definde by the load level

Outputs:

y1: Drum pressure PSI

y2: Excess Oxygen in exhaust gases %

y3: Level of water in the drum

y4: Steam Flow Kg./s

References:

[1] G. Pellegrinetti and J. Benstman, Nonlinear Control Oriented Boiler Modeling -A Benchamrk Problem for Controller Design, IEEE Tran. Control Systems Tech. Vol.4No.1 Jan.1996

[2] J. Espinosa and J. Vandewalle Predictive Control Using Fuzzy Models Applied to a Steam Generating Unit, Submitted to FLINS 98 3rd. International Workshop on Fuzzy Logic Systems and Intelligent Technologies for Nuclear Science and Industry

Properties:

To make possible the open loop identification the wter level was

stabilized by appliying to the water flow input a feedforward action proportional to the steam flow

with value 0.0403 and a PI action with values Kp=0.258 Ti=1.1026e-4 the reference of this controller

is the input u3.

Columns:

Column 1: time-steps

Column 2: input fuel

Column 3: input air

Column 4: input level ref.

Column 5: input disturbance

Column 6: output drum pressure

Column 7: output excess oxygen

Column 8: output water level

Column 9: output steam flow

Category:

Process industry systems

Where:

ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/espinosa/datasets/powplant.dat

ابتدا دیتا را در متلب بارگذاری میکنیم:

%% data

load steamgen

c=steamgen;

p=[c((87-7):(9600-7),2)';c(87:9600,3)';c(87:9600,4)';c((87-1):(9600-1),5)';c((1):(9600-86),5)';c((87-9):(9600-9),6)';c((87-6):(9600-6),7)';c((87-49):(9600-49),8)'];

T=c(87:9600,9)';

num\_traindata=3500;

num\_testdata=4000;

با ساتفاده از دستور های زیر شبکه را نرمالیزه میکنیم:

%% normalizing

[p2,ps] = mapminmax(p);

[t2,ts] = mapminmax(T);

**شبکه MLPتک لابه**

شبکه عصبی MLP تک لایه با 25 نرون لایه پنهان

m=25;

epochs =150;

net=newff(minmax(p2),[20 1],{'tansig' 'purelin'},'trainlm')

net.trainParam.epochs = epochs;

net.trainParam.lr=0.00001;

[net,tr]=train(net,p2(:,1:num\_traindata),t2(1:num\_traindata));

% train

a2 = sim(net,p2(:,1:num\_traindata));

e=a2-t2(1:num\_traindata);

msetrain=mse(e)

% interpolate

a3 = sim(net,p2(:,(num\_traindata+1):num\_testdata));

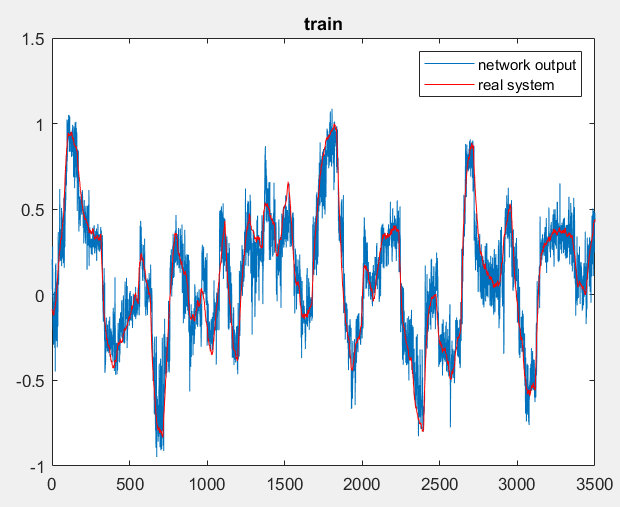
e1=a3-t2((num\_traindata+1):num\_testdata);

msetest=mse(e1)

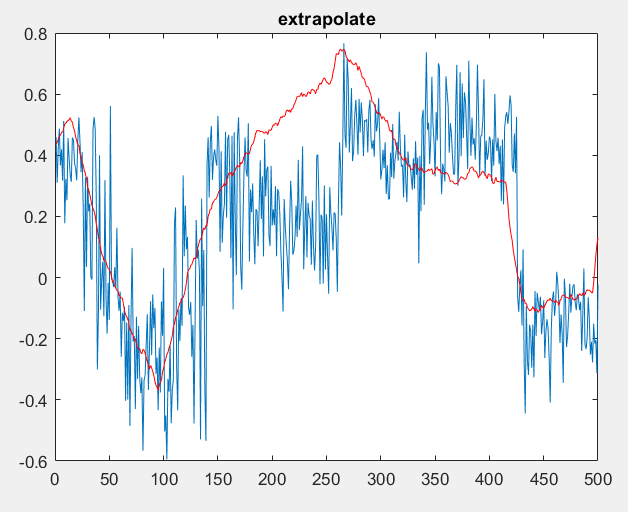
پلات نهایی

در کل دو عدد پلات برای این مساله برای شناسایی هر سیستم MISO داریم:

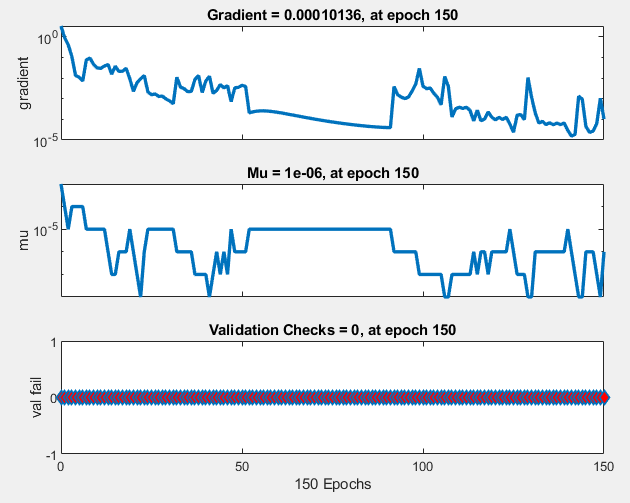
**4ورودی خروجی اول**

****

در شکل بالا دیده میشود که عمل آموزش نسبتا خوب انجام شده است.

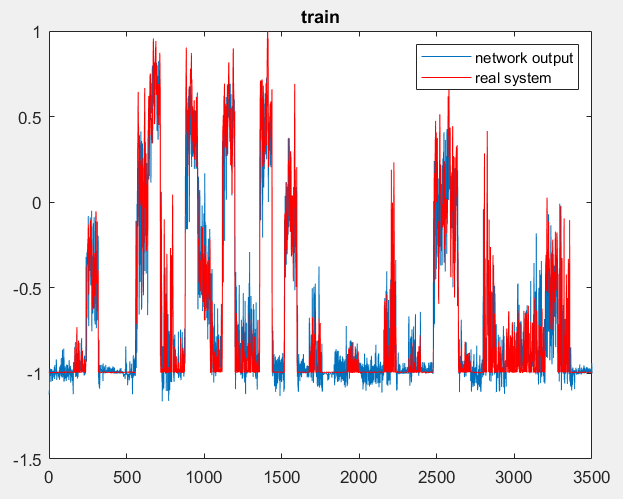


به دلیل سریع بودن دینامیک های خروجی دنبال کنندگی خروجی به خوبی انجام نشده است.

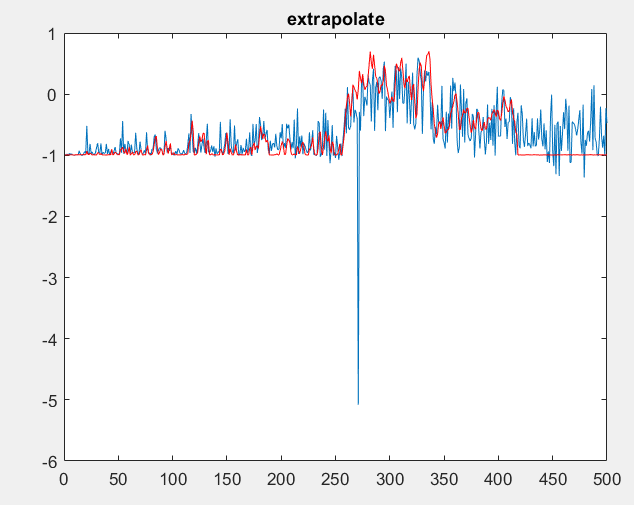


نمودار های خطا نوسانات زیادی داشته است. بخشی از آن میتواند به خاطر نویز گرادیان باشد و یا میتواند به علت بزرگ بودن نرخ آموزش باشد. ویا اینکه شبکه مناسب نیست.

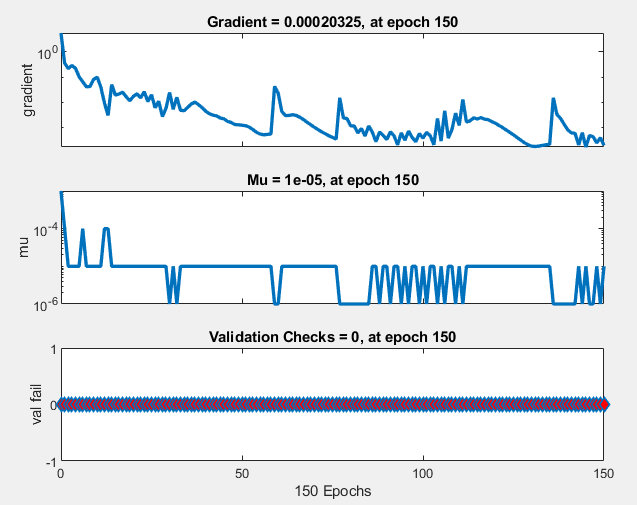
**4ورودی خروجی دوم**



همینطور که مشاهده میشود به علت دینامیک های سنگین خروجی دوم، موفقیت شناسایی با MLP تک لایه کمی دور از انتظار به نظر میرسد.

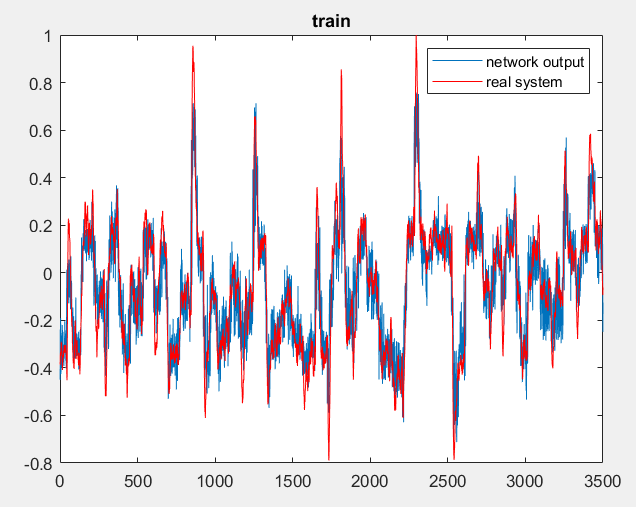


با توجه به نمودار میتوان اینگون هنتیجه گرفت آموزش به درستی انجام نشده است. و میتوانستیم با افزایش پیچیدگی شبکه جواب بهتری بگیریم.

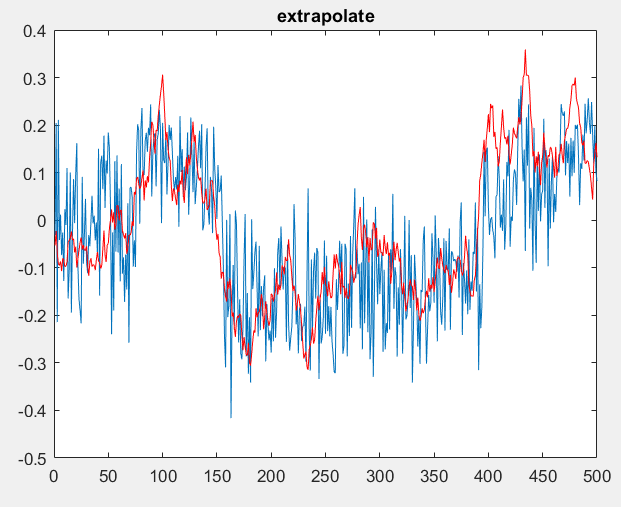


در ایپاک های 60 ، 75 و 130 دارای افزایش خطا هستیم. که این عمر زیبنده نیست☺

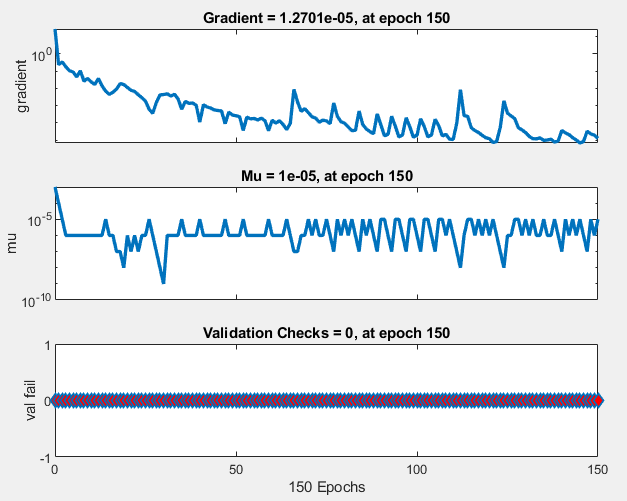
**4ورودی خروجی سوم**



آموزش به صورت قابل قبولی صورت گرفته است

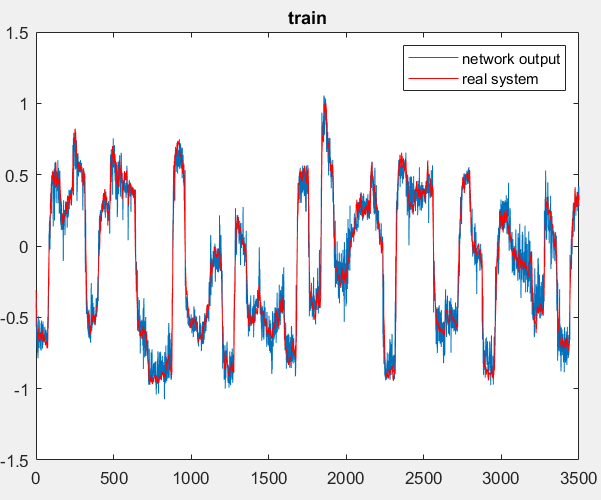


نتیجه حاصل از آموزش خروجی 3.

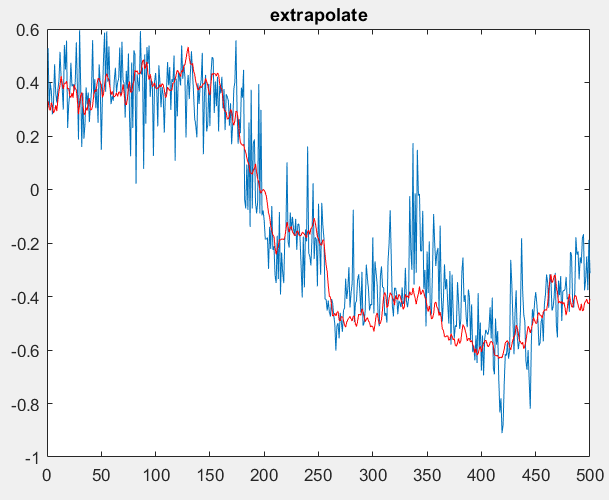


از ایپاک 100 به بعد مشاهده می‌شود که در دو مرحله میزان خطا افزایش یافنه است.

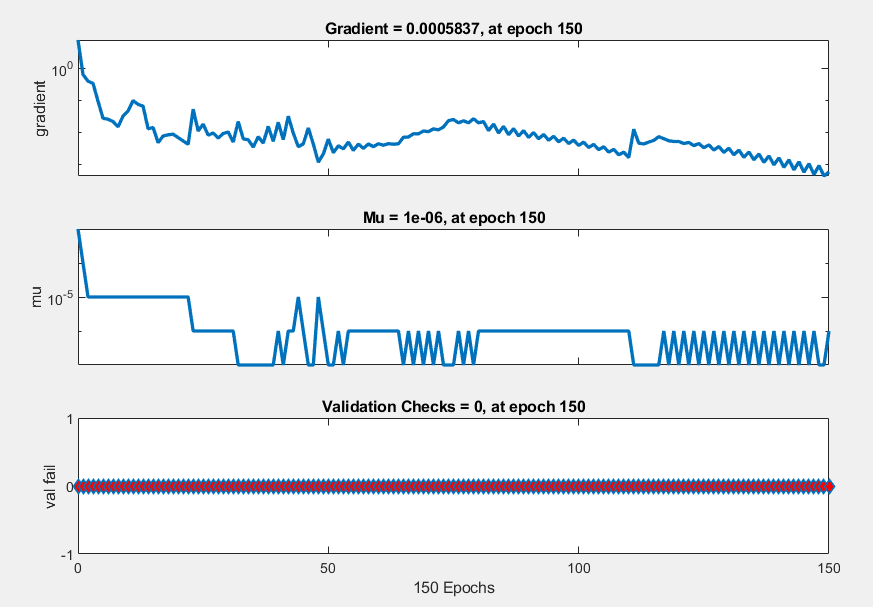
**4ورودی خروجی چهارم**

****

نتیجه آموزش شبکه با داده های حروجی 4

****

خروجی حاصل از مرحله تست.



خطای نهایی برای خروجی 4

چون سیستم یک سیستم صنعتی واقعی بود نمیتوانستیم با یک لایه MLP آن را شناسایی کنیم، نیاز به شبکه پیچیده تری داریم. در همه موارد شبکه underfit شده است به این دلیل که شبکه برای داده های موجود زیادی ضعیف است.

**شبکه MLP دو لایه**

با توجه به underfit شدن شبکه تک لایه، انتظار داریم نتایج حاصل از شبکه دولایه بهتر شوند.

%% newff

epochs =150;

net=newff(minmax(p2),[8 25 1],{'tansig' 'tansig' 'purelin'},'trainlm')

net.trainParam.epochs = epochs;

net.trainParam.lr=0.00001;

[net,tr]=train(net,p2(:,1:num\_traindata),t2(1:num\_traindata));

a2 = sim(net,p2(:,1:num\_traindata));

e=a2-t2(1:num\_traindata);

msetrain=mse(e)

a3 = sim(net,p2(:,(num\_traindata+1):num\_testdata));

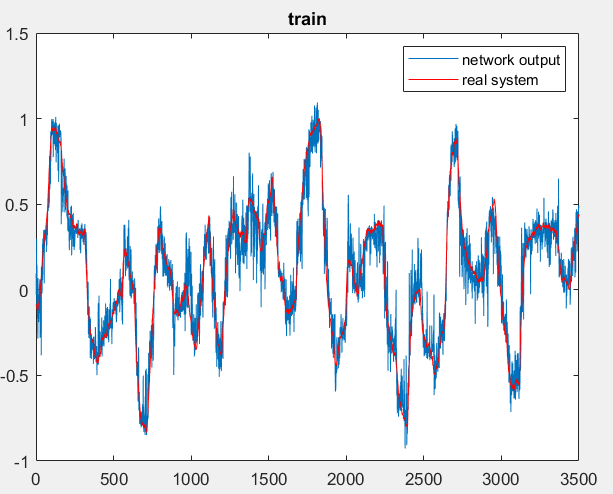
e1=a3-t2((num\_traindata+1):num\_testdata);

msetest=mse(e1)

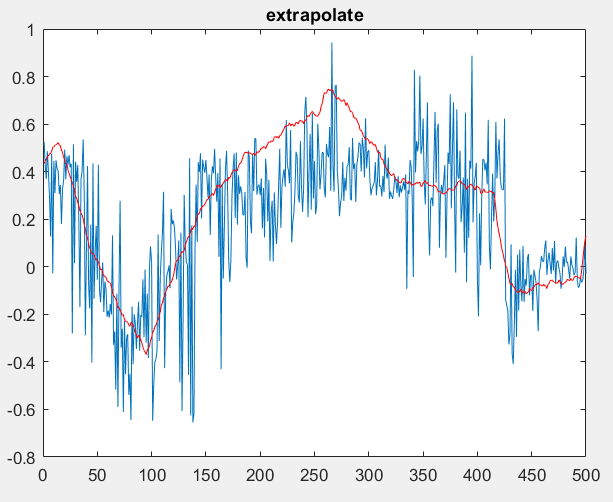
8 نرون لایه اول

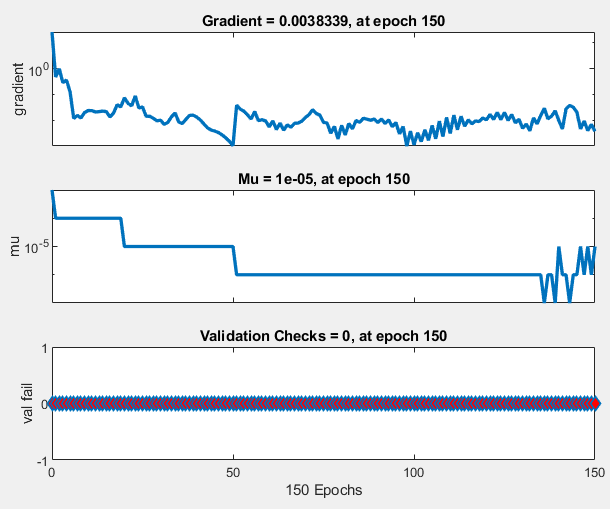
25 نرون لایه دوم

4**ورودی خروجی اول**



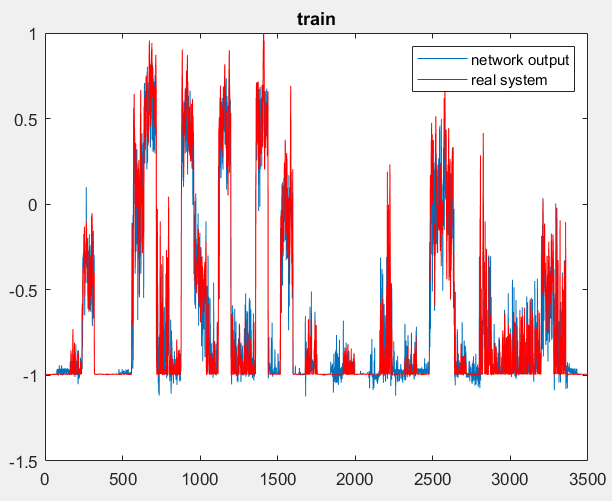
نتایج حاصل از اموزش شبکه دولایه به روی خروجی اول



نتیجه شبکه بر روی داده های تست. بهتر از حالت قبل انجام شده. 

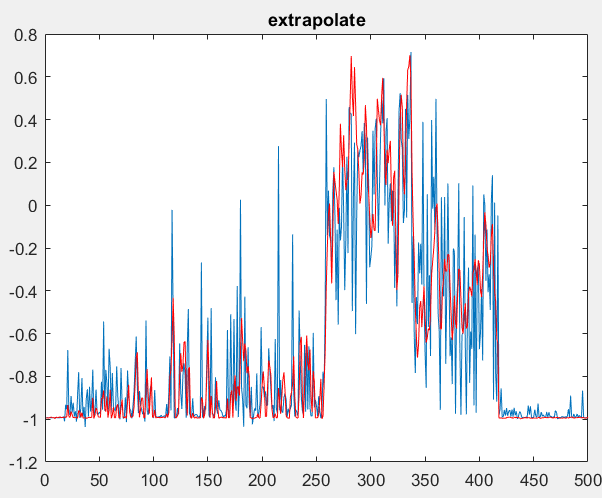
حتی منحنی خطا هم بهتر و نرم تر شده است

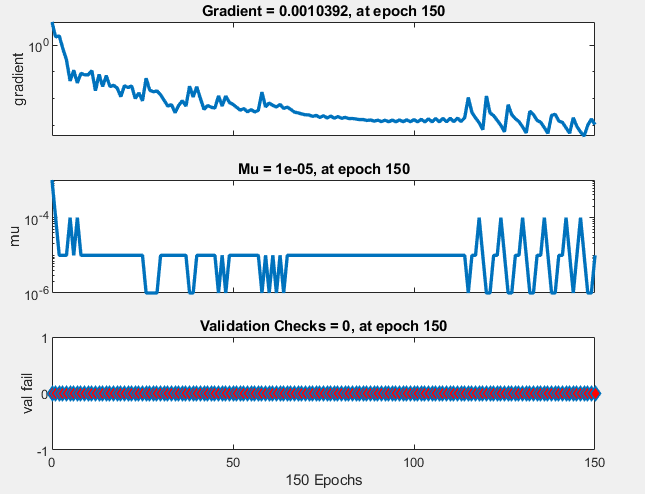
**4ورودی خروجی دوم**



نتایج آموزش روی خروجی 2، با توجه به دینامیک سنگین داده میتوان نتیجه گرفت که حتی شبکه دولایه هم برای آموزش این خروجی کافی نست.

البته با مقایسه این نمودار و تمودار ورودی اول مشخص میشود که تا حدی توانسته دینامیک های خروجی را دنبال کند.

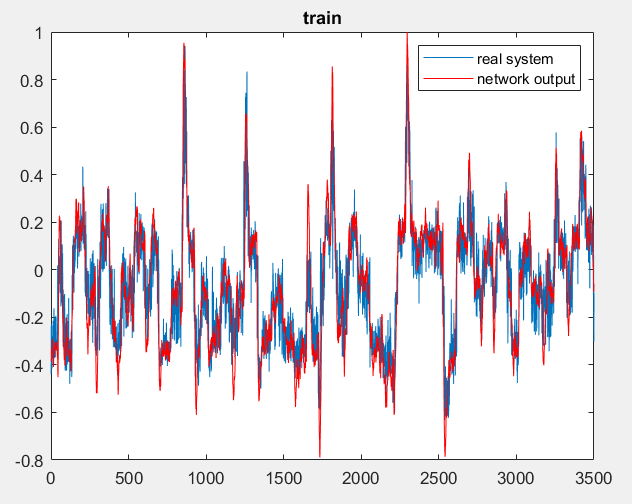




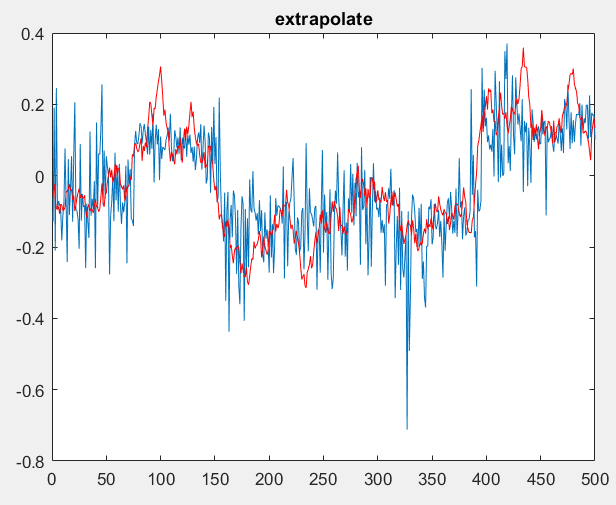
با این وجود، خطا تا حد مناسبی کاهش پیدا کرده است

توجه شود که در راهنمای شکل های آموزش اشتباه شده بود. از الان به بعد تصحیح شده است.

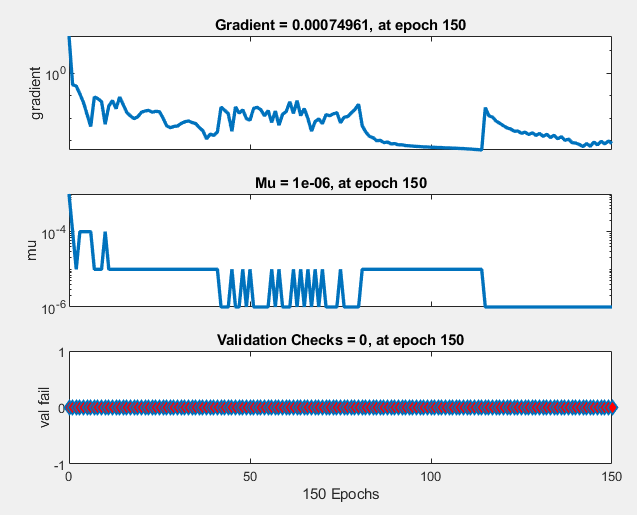
**4ورودی خروجی سوم**



نمودار آموزش



نمودار تست



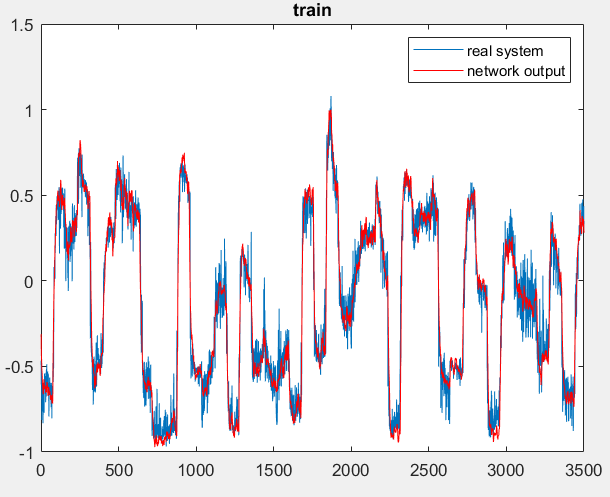
خطا به حد قابل ملاحظه ای کاهش یافته است(در حدود 0.001) و این چیز خوبی است. این موضوع نشاندهنده ان است که:

**We are on the right track**

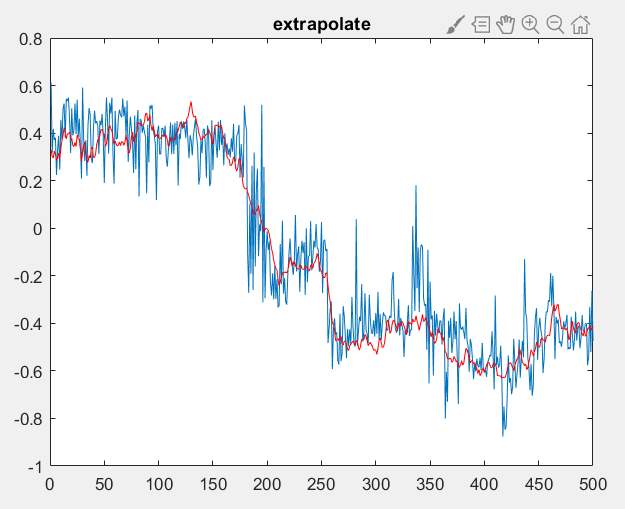
**4ورودی خروجی چهارم**

حال به سراغ یک ابتکار میرویم. ایا افزایش تعداد ایپاک سبب بهتر شدن نتیجه میشود؟

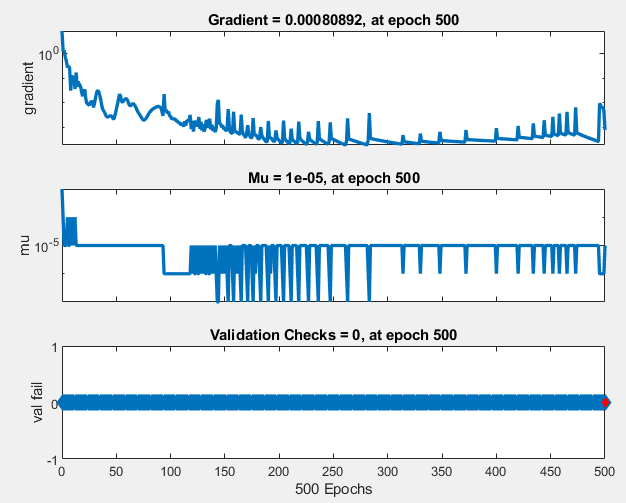
ایپاک 500



نتیجه آموزش به نظر بهتر از حالت های قبل است.



نتیجه تست، تقریبا به همان منوال سابق است.



خطا در بازه 0.01 و 0.001 است

نتیجه با افزایش تعداد ایپاک بهتر نشده است.

**RBF**

با خط های پایین داده را نرمالیزه میکنیم

%% normalizing

for i=1: num\_testdata

u(i,1)=(U(i,1)-min(U(:,1)))/(max(U(:,1))-min(U(:,1)));

u(i,2)=(U(i,2)-min(U(:,2)))/(max(U(:,2))-min(U(:,2)));

u(i,3)=(U(i,3)-min(U(:,3)))/(max(U(:,3))-min(U(:,3)));

u(i,4)=(U(i,4)-min(U(:,4)))/(max(U(:,4))-min(U(:,4)));

u(i,5)=(U(i,5)-min(U(:,5)))/(max(U(:,5))-min(U(:,5)));

u(i,6)=(U(i,6)-min(U(:,6)))/(max(U(:,6))-min(U(:,6)));

u(i,7)=(U(i,7)-min(U(:,7)))/(max(U(:,7))-min(U(:,7)));

z(i)=(T(i)-min(T))/(max(T)-min(T));

end

کد های مزبوط به شبکه RBF

%% rbf

m=30

epoch=10;

etha1=.01;

etha=0.8;

sigma1=.5;

sigma2=.5;

sigma3=.5;

sigma4=.5;

sigma5=.5;

sigma6=.5;

sigma7=.5;

c=1+5\*rand(7,m);

w=-1+2\*rand(m,1);

p=1000\*eye(m);

for k=1:epoch

% data=randperm(121);

for i=1:num\_traindata

for j=1:m

t1(1,j)= ((u(i,1)-c(1,j))/sigma1)^2;

t2(1,j)=((u(i,2)-c(2,j))/sigma2)^2;

t3(1,j)=((u(i,3)-c(3,j))/sigma3)^2;

t4(1,j)=((u(i,4)-c(4,j))/sigma4)^2;

t5(1,j)=((u(i,5)-c(5,j))/sigma5)^2;

t6(1,j)=((u(i,6)-c(6,j))/sigma6)^2;

t7(1,j)=((u(i,7)-c(7,j))/sigma7)^2;

phi(1,j)=exp(-.5\*(sqrt(t1(j)+t2(j)+t3(j)+t4(j)+t5(j)+t6(j)+t7(j))));

end

xt=phi';

yhat(i)=xt'\*w;

e1(i)=z(i)-yhat(i);

for j=1:m

c(:,j)=c(:,j)+etha\*e1(i)\*w(j,1)\*[(u(i,1)-c(1,j))/(sigma1^2);(u(i,2)-c(2,j))/(sigma2^2);((u(i,3)-c(3,j))/sigma3)^2;(u(i,4)-c(4,j))/(sigma4^2);(u(i,5)-c(5,j))/(sigma2^5);((u(i,6)-c(6,j))/sigma6)^2;((u(i,7)-c(7,j))/sigma7)^2]\*phi(j);

sigma1=sigma1+etha1\*e1(i)\*w(j,1)\*phi(j)\*(1/(sigma1)^3)\*((u(i,1)-c(1,j))^2);

sigma2=sigma2+etha1\*e1(i)\*w(j,1)\*phi(j)\*(1/(sigma2)^3)\*((u(i,2)-c(2,j))^2);

sigma3=sigma3+etha1\*e1(i)\*w(j,1)\*phi(j)\*(1/(sigma3)^3)\*((u(i,3)-c(3,j))^2);

sigma4=sigma4+etha1\*e1(i)\*w(j,1)\*phi(j)\*(1/(sigma4)^3)\*((u(i,4)-c(4,j))^2);

sigma5=sigma5+etha1\*e1(i)\*w(j,1)\*phi(j)\*(1/(sigma5)^3)\*((u(i,5)-c(5,j))^2);

sigma6=sigma6+etha1\*e1(i)\*w(j,1)\*phi(j)\*(1/(sigma6)^3)\*((u(i,6)-c(6,j))^2);

sigma7=sigma7+etha1\*e1(i)\*w(j,1)\*phi(j)\*(1/(sigma7)^3)\*((u(i,7)-c(7,j))^2);

end

gama=p\*xt/(1+xt'\*p\*xt);

p=(eye(m)-gama\*xt')\*p;

t(i)= trace(p);

if t(i)>1000

p=1000\*eye(m);

end

w=w+gama\*e1(i);

end

mse\_train(k)=mse(e1)

for h=(num\_traindata+1):num\_testdata

for g=1:m

phi1(1,g)=exp(-.5\*(sqrt(((u(h,1)-c(1,g))/sigma1)^2+((u(h,2)-c(2,g))/sigma2)^2+((u(h,3)-c(3,g))/sigma3)^2+((u(h,4)-c(4,g))/sigma4)^2+((u(h,5)-c(5,g))/sigma5)^2+((u(h,6)-c(6,g))/sigma6)^2+((u(h,7)-c(7,g))/sigma7)^2)));

end

xt1=phi1';

yhat(h)=xt1'\*w;

e2(h)=z(h)-yhat(h);

gama=p\*xt1/(1+xt1'\*p\*xt1);

p=(eye(m)-gama\*xt1')\*p;

t(h)= trace(p);

if t(h)>1000

p=1000\*eye(m);

end

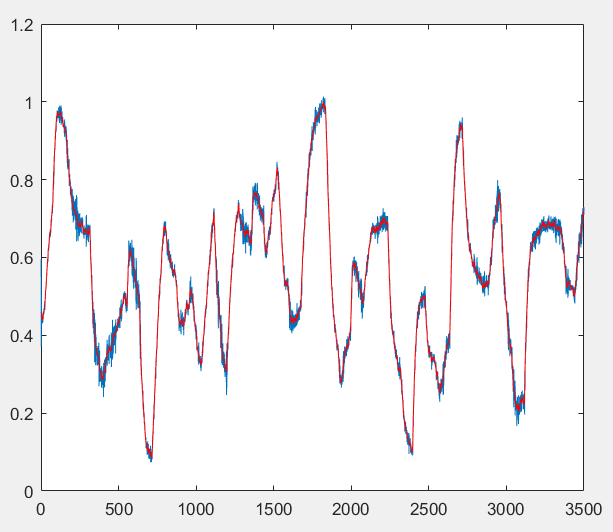
w=w+gama\*e2(h);

end

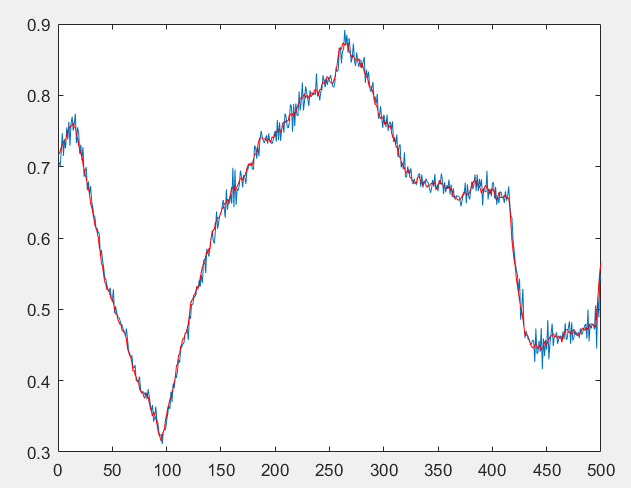
mse\_test(k)=mse(e2((num\_traindata+1):num\_testdata))

end

**خروجی اول**

****

به نظر شبکه خیلی بهتر از حالت MLP جواب میدهد

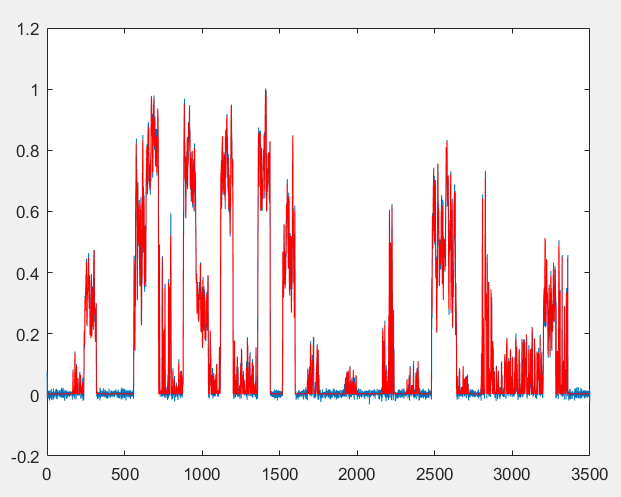


نتایج تست هم به نظر بهتر شده اند

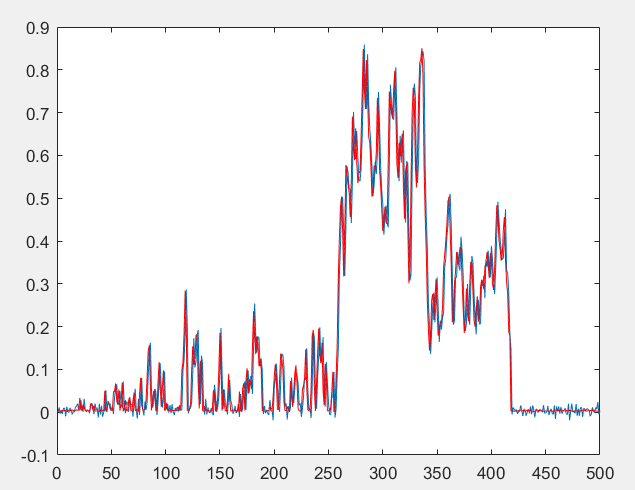
آبی نمودار واقعی است

و قرمز نمودار تخمینی است

**خروجی دوم**

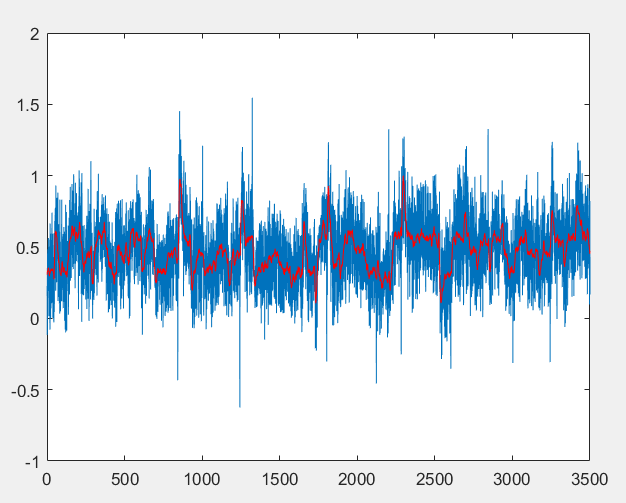


داده های اموزش تا حد زیادی نمودار تخمین و نمودرا واقعی روی هم افتاده اند.



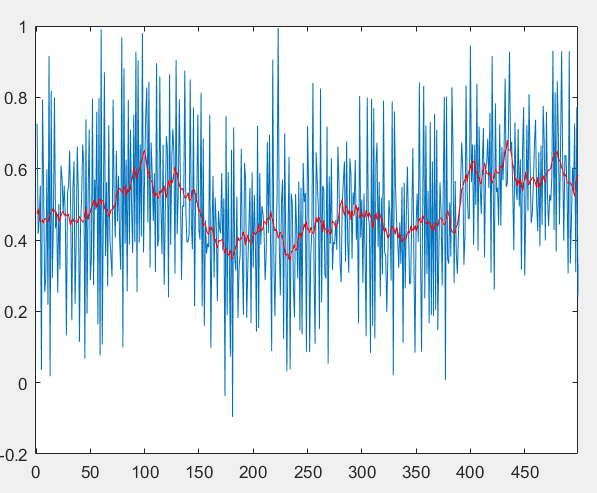
از داده تست مشاهده میشود که عملکرد شبکه rbf از MLP خیلی بهتر است.

**خروجی سوم**



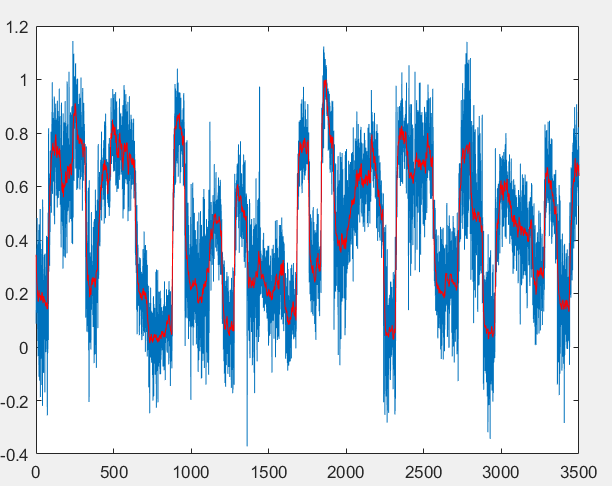
داده اموزش

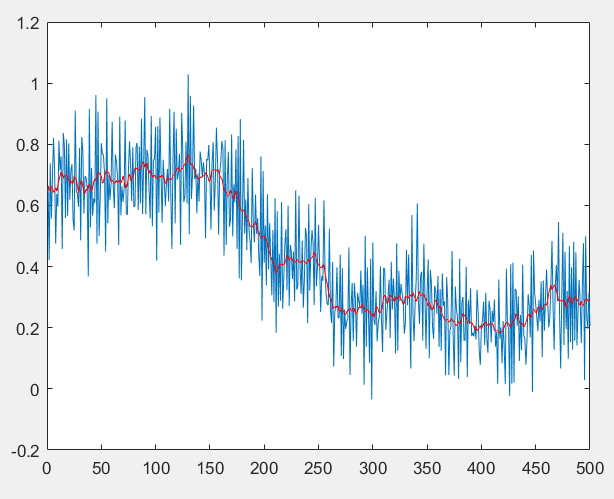
به مانند حالت قبل این خروجی سخت ترین خروجی برای مدل سازی است



از نتابج تست هم پیدا ست که مدلستزی به خوبی صورت نگرفته است

**خروجی چهارم**





نتیجه تست به روی داده های خروجی 4. همچنان داده ها به خوبی مدل نشده اند. چون دینامیک هاتی خروجی از حالت های قبل سخت تر هستند