

باسمه تعالی



دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس هوش محاسباتی

تمرین سری پنجم

علی محرابیان 96102331

استاد: دکتر رضایی

بهار 1399



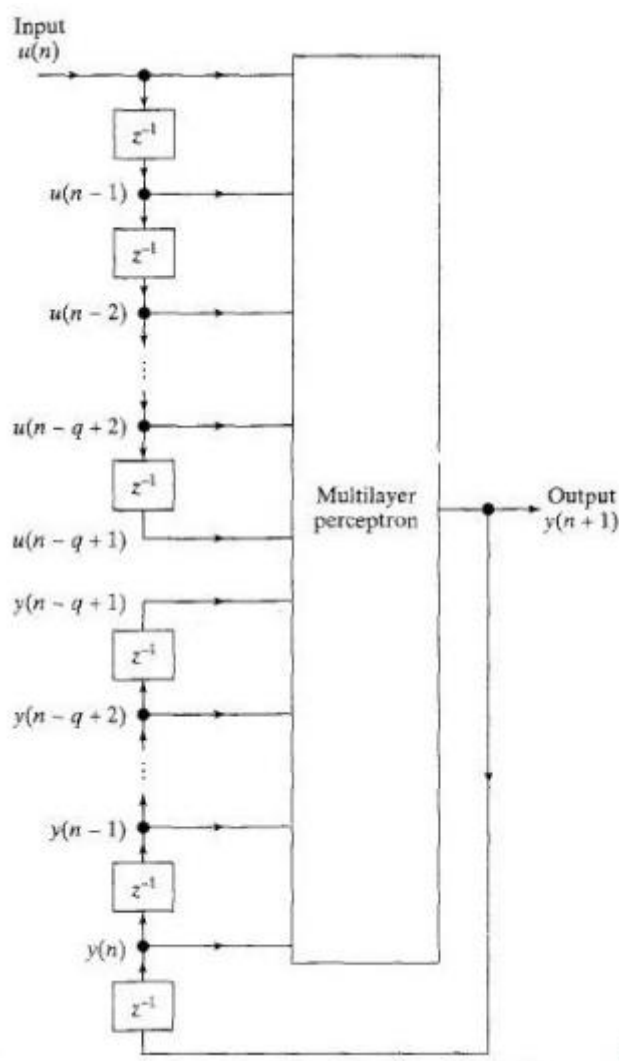
1.

در این قسمت با توجه به این که تابع ورودی در محدوده 2 تا -2 باید باشد، از تابع $\sqrt{2} \sin[k] + \cos[k]$ برای ساختن و آموزش داده ها استفاده کردیم.

همان طور که می دانیم با توجه به معادله NARMA، هدف تخمین تابع f می باشد.

$$\|f - \hat{f}\| < \epsilon \text{ for some } \epsilon \geq 0$$

ساختار پیاده سازی ما به صورت زیر خواهد بود.

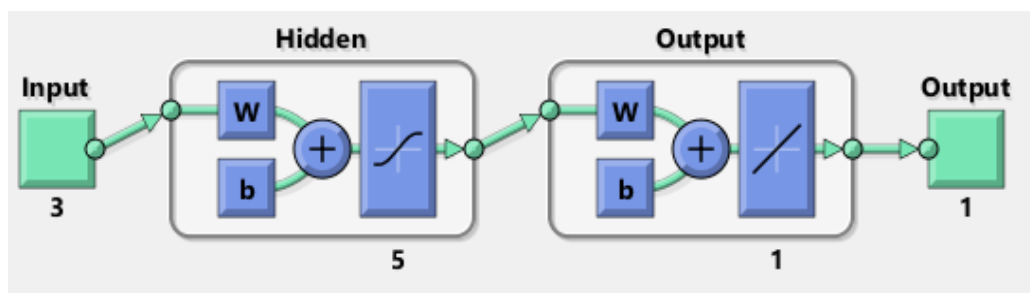




```
k=-pi:0.01:3*pi/2;  
n=length(k);  
u1=(sqrt(2)*sin(k)+cos(k))';  
u2=(0.1*u1.^3)';  
f=[0 0 u1(1) u2(1)];  
t=[0];  
for i=2:n  
    f1=zeros(1,4);  
    f1(3)=u1(i);  
    f1(4)=u2(i);  
    f1(2)=0.4*f(i-1,1);  
    f1(1)=0.1*t(i-1);  
    t1=sum(f1);  
    t=[t;t1];  
    f=[f;f1];  
end
```

در قسمت اول مقادیر $y[k+1]$ را برای نقاط بین 11- تا 401 محاسبه می کنیم. در هر بار محاسبه، باید از مقادیر به دست آمده در مرحله قبل استفاده کرده و همه موارد را جداگانه ذخیره می کنیم. لازم به ذکر است که ورودی های ما برای آموزش، مقادیر $u[k], y[k]$ هستند. در قسمت بعد این ورودی ها را جدا کرده و به همراه خروجی های به دست آمده برای آموزش استفاده می کنیم.

در قسمت اول از شبکه عصبی با یک لایه مخفی که دارای 5 نورون است، آموزش را انجام می دهیم.

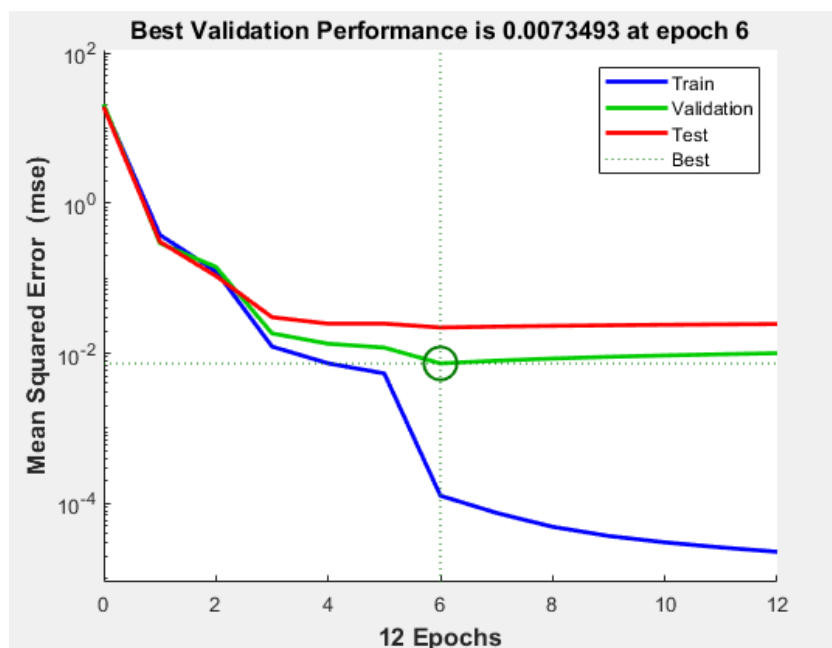


80% داده ها برای آموزش، 5% برای validation و 15% برای تست استفاده شده اند. خلاصه نتایج به صورت زیر است.

Progress				
Epoch:	0	12 iterations		1000
Time:		0:00:00		
Performance:	20.1	2.28e-05		0.00
Gradient:	32.8	0.00119		1.00e-07
Mu:	0.00100	0.000100		1.00e+10
Validation Checks:	0	6		6



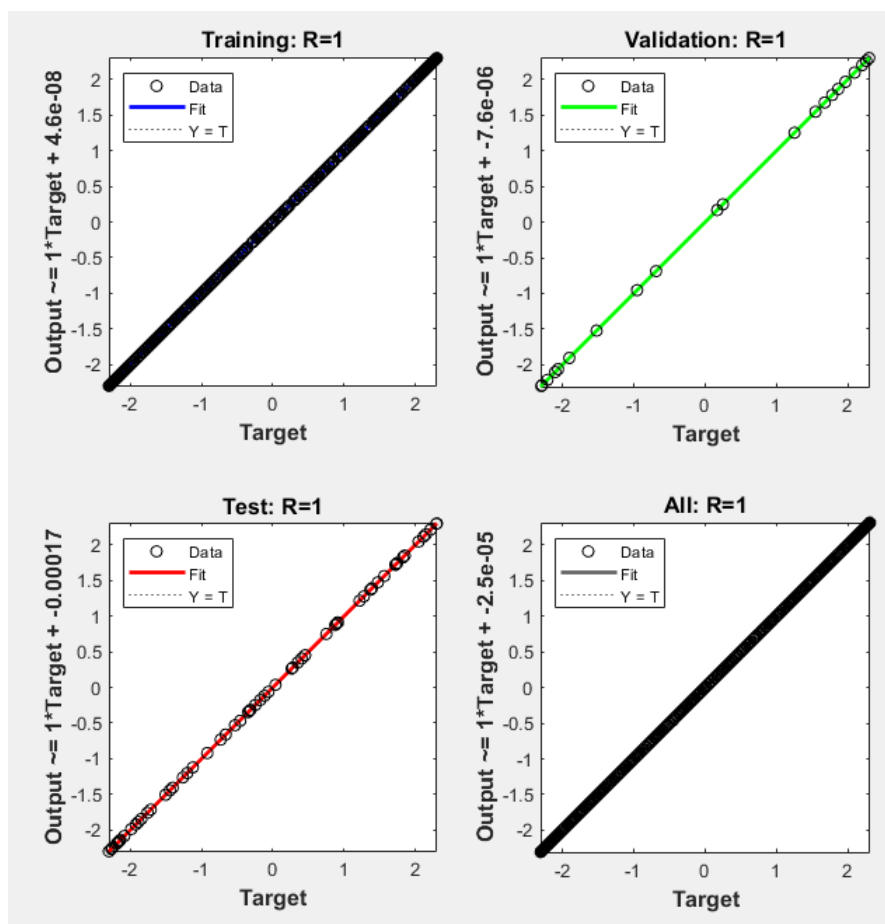
در ابتدا نمودار عملکرد و خطای کمینه مربعات به صورت زیر است.



همان طور که مشاهده می شود، خطای کمینه مربعات برای داده های تست در حدود 10^{-2} است که حکایت از مطلوب بودن آموزش دارد.



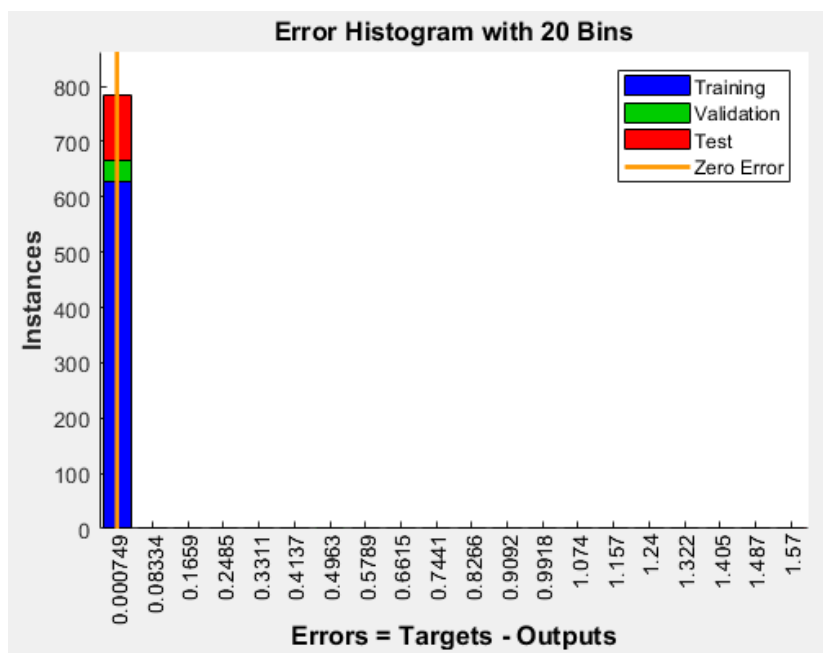
نمودار regression به برای قسمت های مختلف داده ها به صورت زیر است.



در منحنی های رگرسیون، هرچه شیب خطوط حاصله به 45 درجه نزدیک تر باشد، خروجی های ما به خروجی اصلی نزدیک تر بوده اند. همان طور که در شکل مشاهده می شود، برای تمامی دسته داده ها، خروجی مطلوب است.



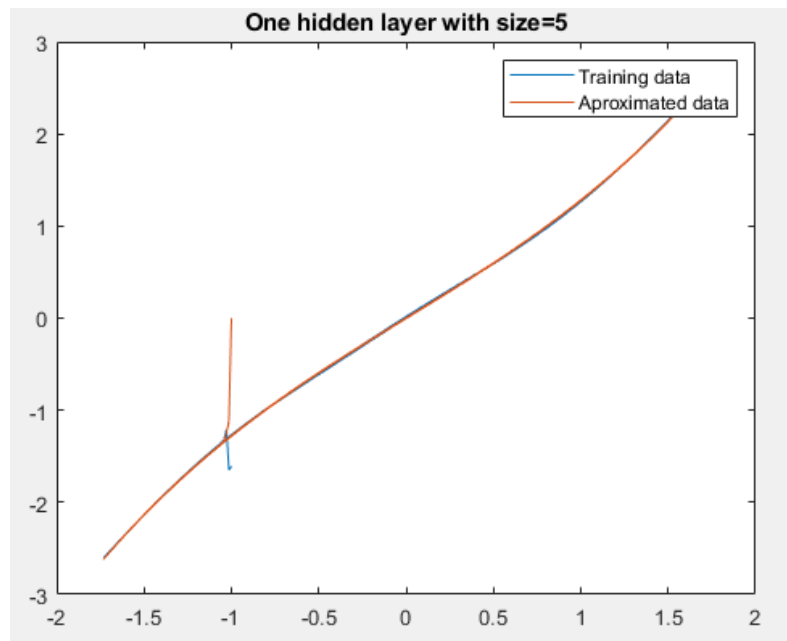
نمودار هیستوگرام برای خطا به صورت زیر است.



یکی از کاربردهای منحنی هیستوگرام، تشخیص داده های outlier نسبت به بقیه داده ها است. البته همان طور که در شکل می بینیم، داده outlier نداریم. با توجه به خطای حاصله، می توان دریافت که آموزش مطلوبی بر روی داده صورت گرفته است.



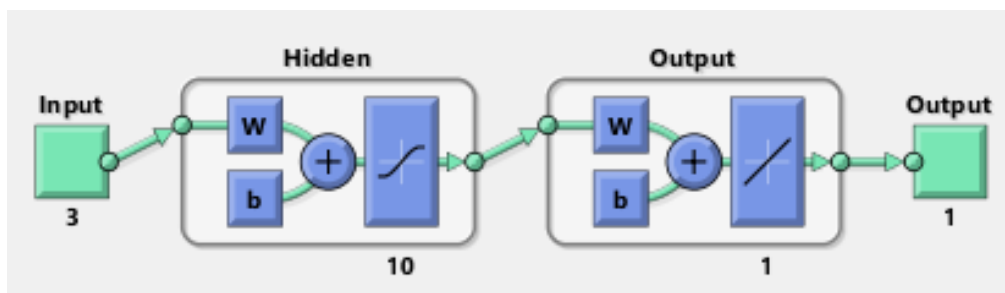
یکی از منحنی هایی که شهود خوبی با ما درباره فرآیند آموزش می تواند بدهد، رسم خروجی $y[k + 1]$ بر حسب $u[k]$ می باشد. در شکل زیر، برای داده های اولیه و خروجی شبکه عصبی، این تابع رسم شده است.



با توجه به شکل می توان دید که انطباق قابل قبولی حاصل شده است.



در قسمت بعد از شبکه عصبی با یک لایه مخفی که دارای تعداد 10 نورون است، استفاده می کنیم. تقسیم بندی داده ها مشابه قسمت قبل است.



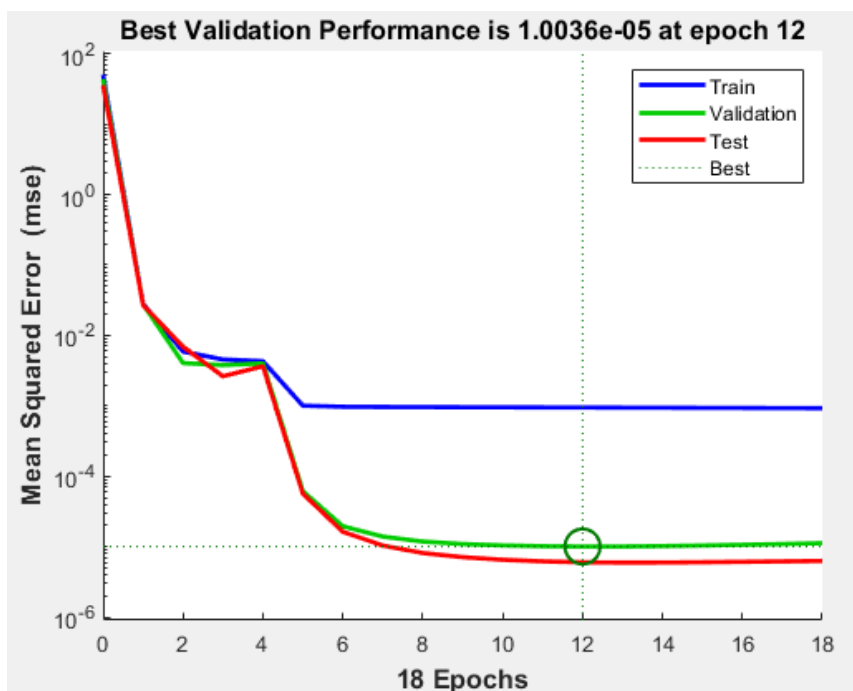
شرح مختصری از نتایج به صورت زیر است.

Progress			
Epoch:	0	18 iterations	1000
Time:		0:00:05	
Performance:	50.5	0.000925	0.00
Gradient:	75.4	0.00144	1.00e-07
Mu:	0.00100	1.00e-05	1.00e+10
Validation Checks:	0	6	6

منحنی ها در صفحات بعد آورده شده اند.



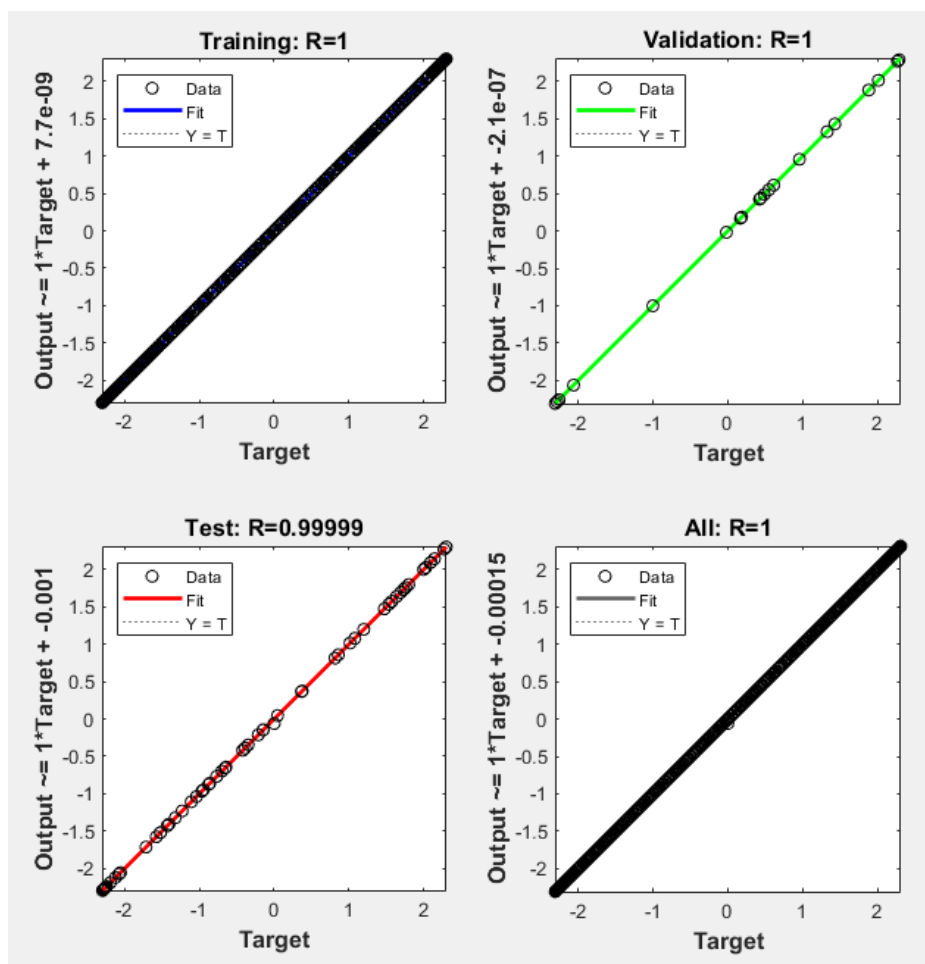
نمودار عملکرد و خطای کمینه مربعات به صورت زیر است.



در این قسمت نیز خطا برای داده های تست در حدود 10^{-5} است که البته با توجه به نمودارها، نسبت به قسمت قبل، بهبود داشته ایم.



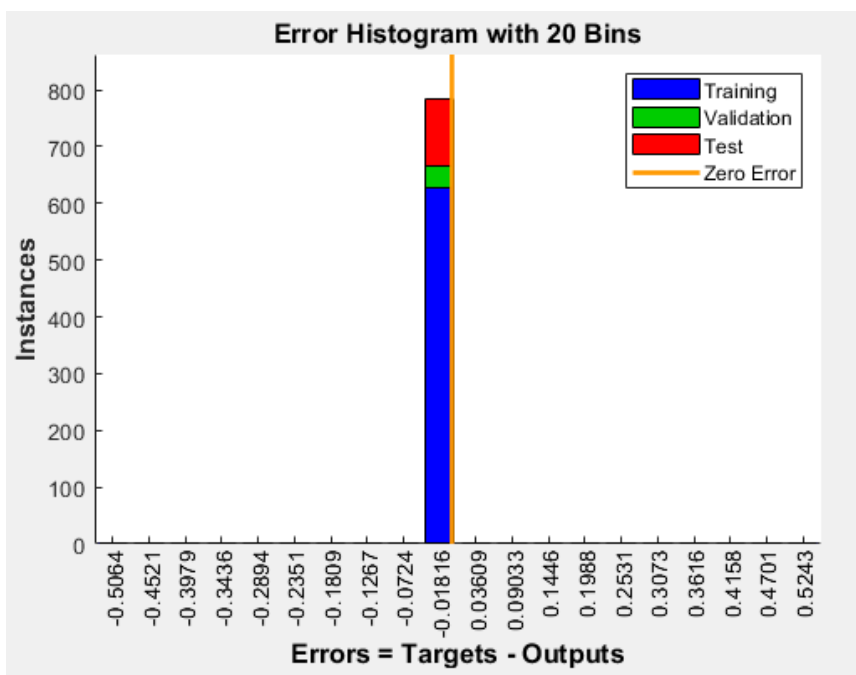
نمودار رگرسیون به صورت زیر است.



در این قسمت نیز نتایج نشان می دهد که آموزش به خوبی صورت گرفته است.



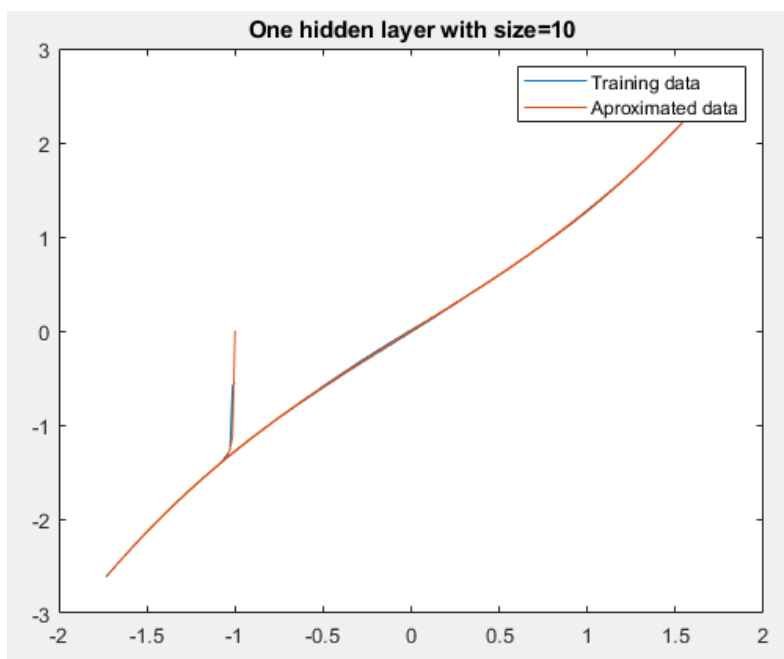
نمودار هیستوگرام خطا به صورت زیر است.



در این قسمت نیز با توجه به این که مقدار خطا کم است، آموزش مطلوب می باشد.



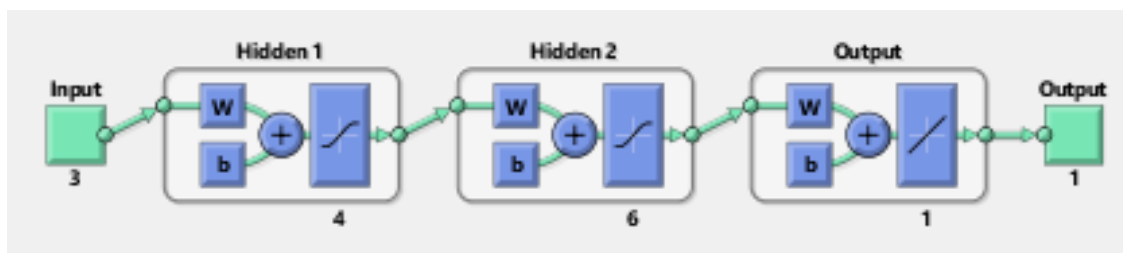
منحنی خروجی بر حسب تابع ورودی به صورت زیر است.



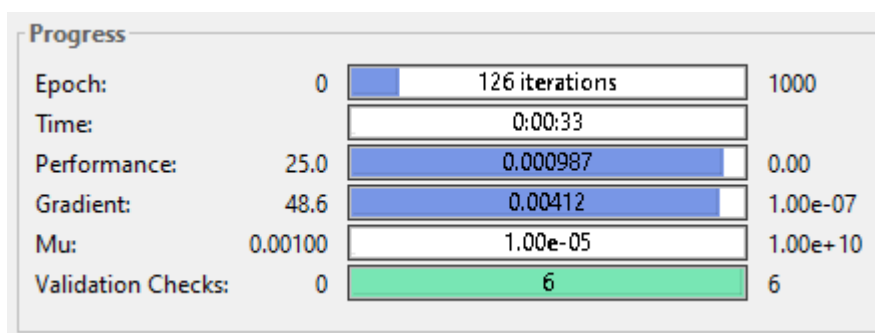
در این حالت نسبت به قسمت قبل حتی در قسمت غیرخطی نیز انطباق داریم که به این معنی است که آموزش خوبی صورت گرفته است.



در قسمت بعدی از شبکه عصبی با دو لایه مخفی که در لایه اول 4 و در لایه دوم 6 نورون داشته باشد، استفاده می کنیم.



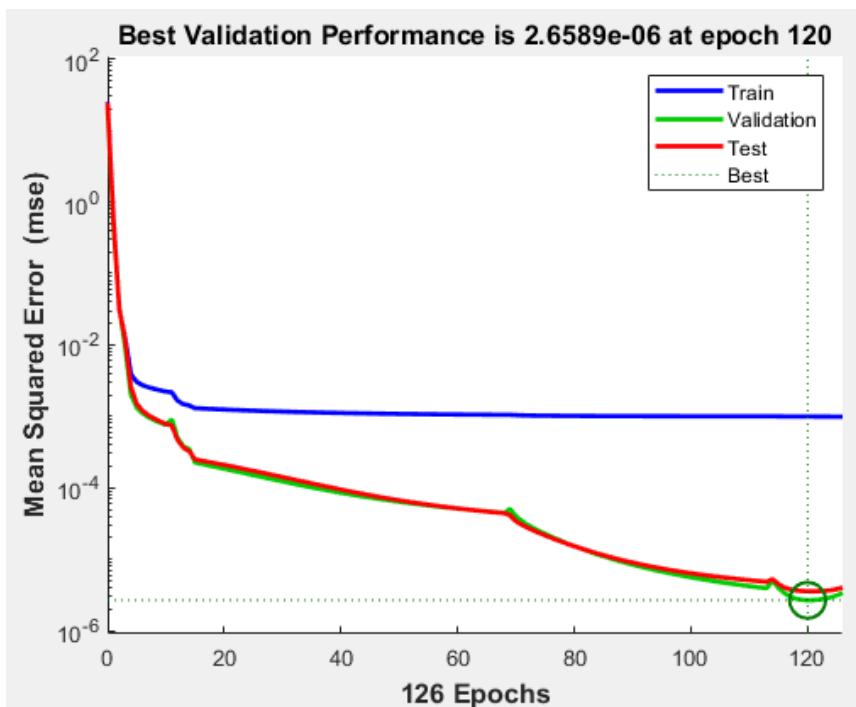
تقسیم بندی داده ها مشابه قبل است. شرح نتایج به صورت زیر است.



منحنی های مربوطه در صفحه های بعد هستند.



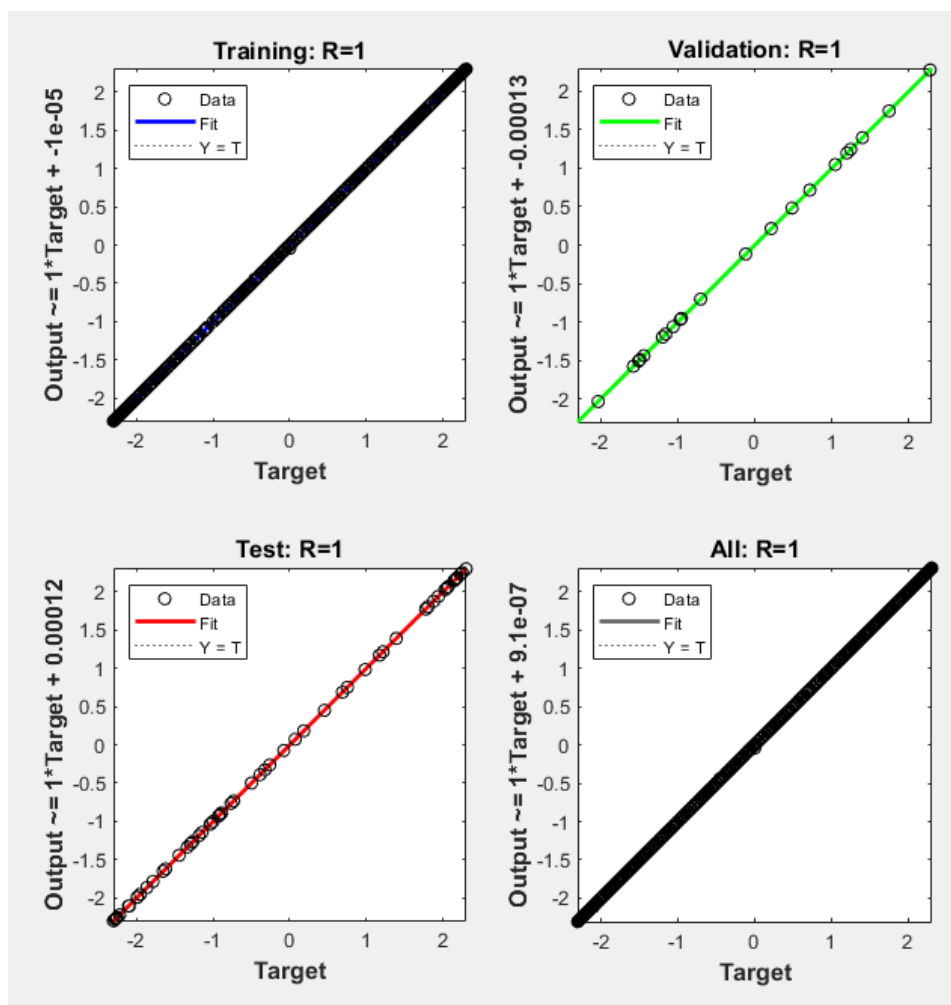
منحنی ارزیابی و خطای کمینه مربعات به صورت زیر است.



در این قسمت هم خطا به مقدار قابل توجهی کوچک است که به این معنی است که آموزش خوب بوده است.



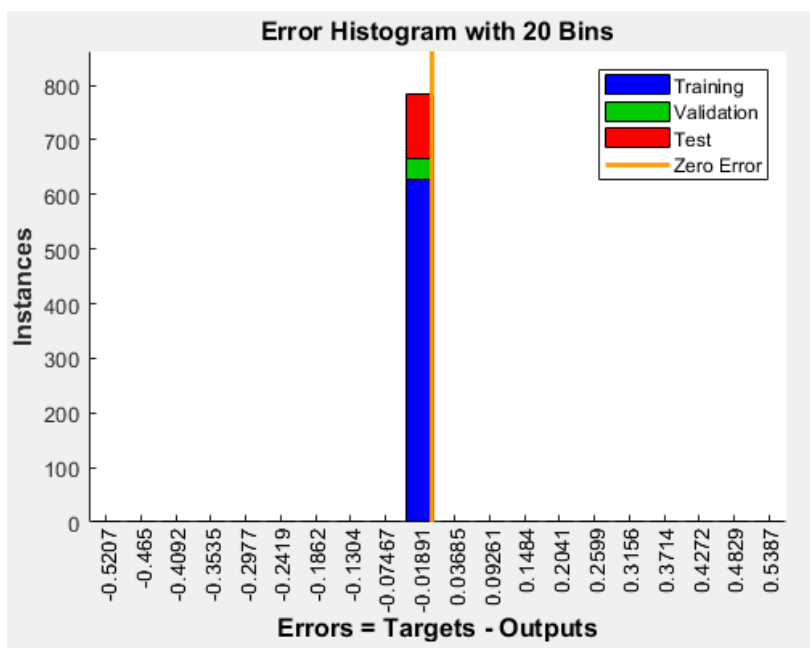
نمودار رگرسیون به صورت زیر است.



با توجه انطباق داده ها بر خطوط با شیب 45 درجه، نتایج و آموزش مطلوب هستند.



نمودار هیستوگرام خطا به صورت زیر است.

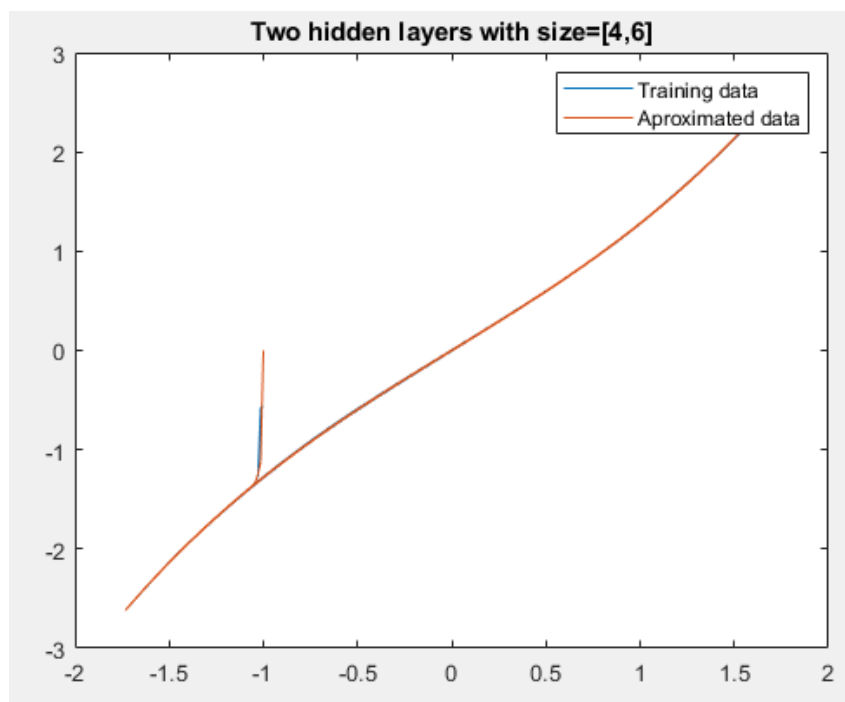


در این قسمت نیز خطا اندک بوده و خروجی مطلوب است.

برای قسمت های بیان شده، تفاوت ها خیلی محسوس نیستند و تقریباً آموزش خوبی برای همه داده ها داشتیم. ولی به طور کلی، آموزش به شبکه دو لایه تا حدودی بهینه تر انجام می شود.



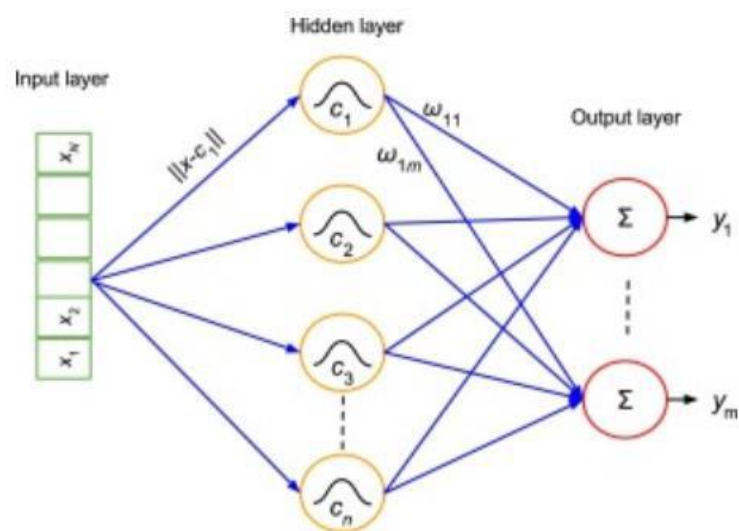
منحنی خروجی بر حسب تابع ورودی به صورت زیر است.



در این حالت نیز خروجی مطلوب است.



در این قسمت از شبکه RBF برای آموزش داده ها به کمک دستور newrbe استفاده می کنیم. ساختار کلی آن به صورت زیر است. با توجه به این که متلب نمودار قابل توجه ای برای RBF به ما نمی دهد، در این قسمت فقط خطای MSE را برای داده های تست گزارش می کنیم. تقسیم بندی داده ها مشابه قبل است.



برای شبکه RBF، خطای کمینه مربعات برای داده های تست به صورت زیر است.

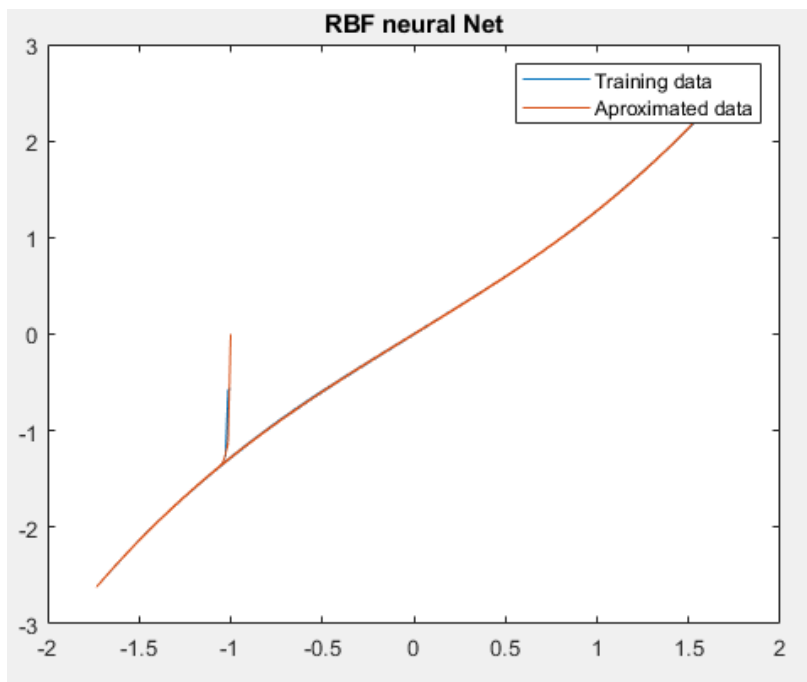
ee =

0.0104

این خطای نسبت به قسمت های قبل بیشتر است ولی همچنان قابل قبول می باشد.



منحنی خروجی بر حسب تابع ورودی به شکل زیر است.



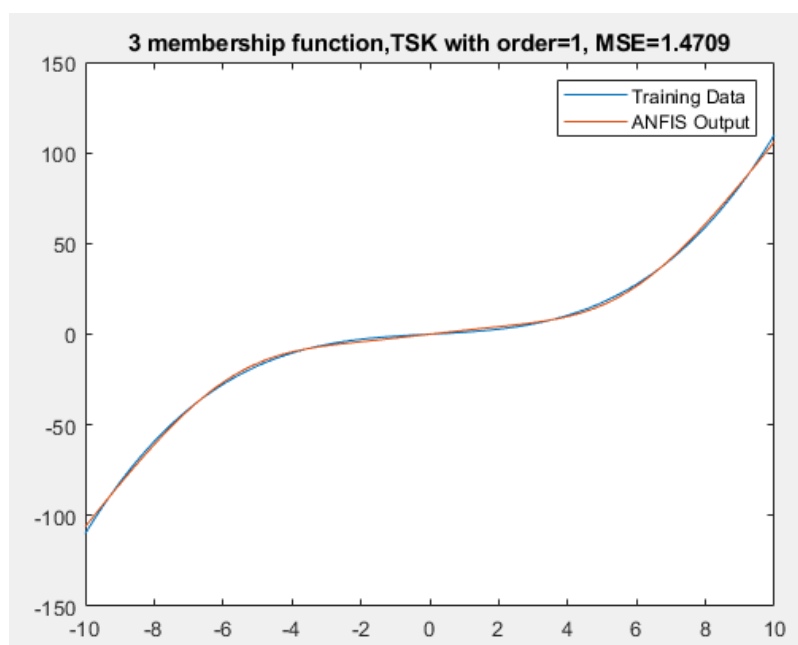
در این حالت نیز انطباق قابل قبولی داشته و آموزش مطلوبی صورت گرفته است.



2.

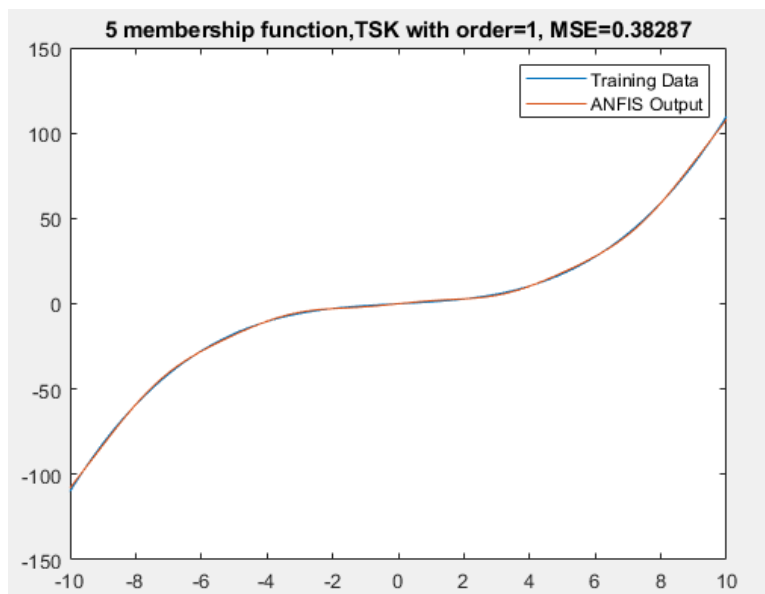
در این قسمت هدف تخمین زدن تابع $f(x) = x + 0.1x^3$ به کمک سیستم neuro-fuzzy می باشد. از تابع Anfis برای پیاده سازی قسمت های مختلف استفاده می کنیم. لازم به ذکر است که در هر مورد، خطای کمینه مربعات را گزارش کرده ایم.

در قسمت اول از TSK مرتبه اول در حالی که 3 تابع عضویت ورودی داریم استفاده می کنیم. خروجی به صورت زیر است.





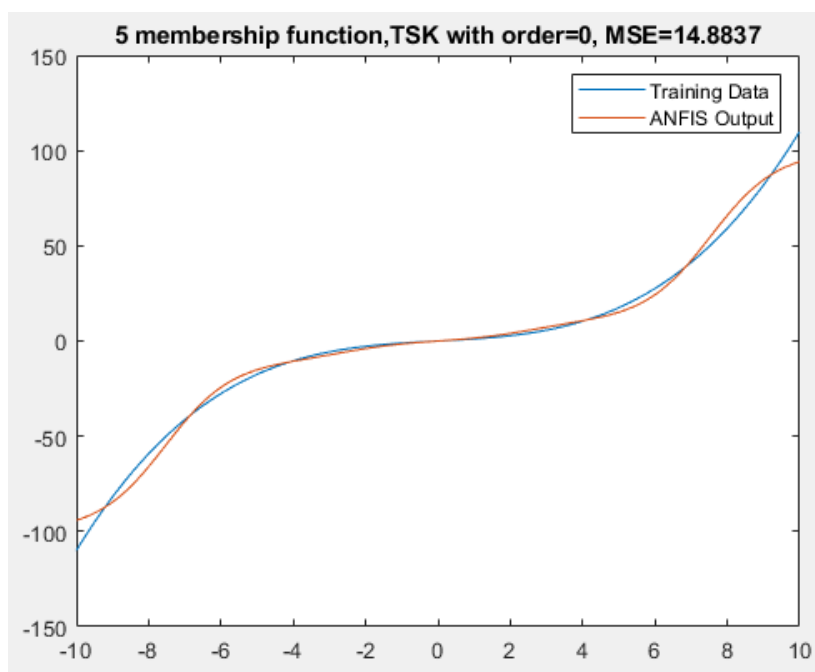
در این قسمت از TSK مرتبه اول با 5 تابع عضویت ورودی استفاده کرده ایم.



این حالت بهترین خروجی را برای ما به ارمغان می آورد، چرا که اولاً استفاده از 5 تابع خطی برای محاسبه خروجی یک آزادی عمل بسیار عالی برای ما ایجاد می کند که از خاصیت های سیستم TSK مرتبه 1 است. ثانیاً استفاده از 5 تابع عضویت ورودی نیز جزئیات انجام کار را بیشتر و بهینه تر می کند.



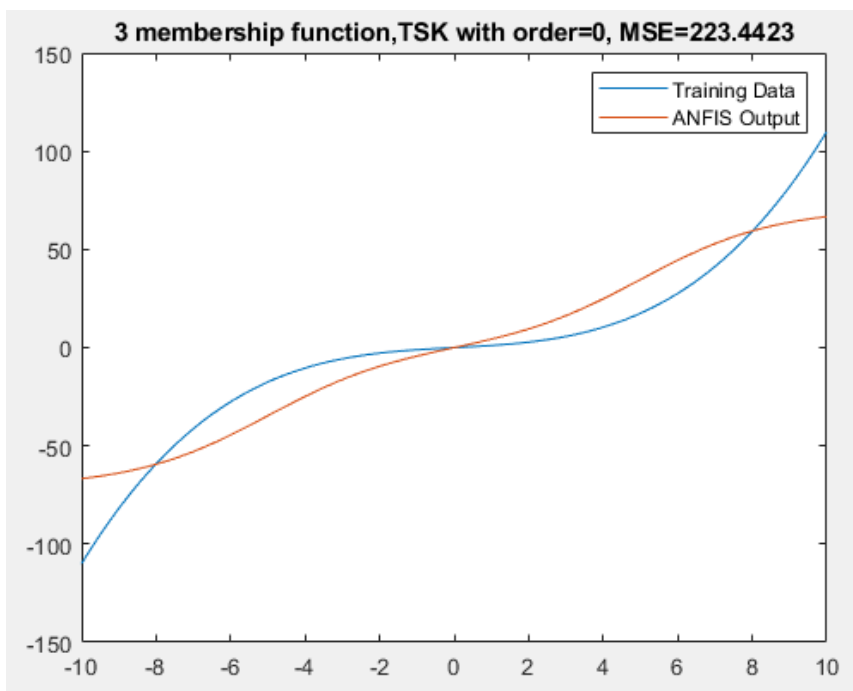
در این قسمت از TSK مرتبه 0 با 5 تابع عضویت ورودی استفاده می کنیم. برای تغییر مرتبه، type خروجی سیستم را به constant تغییر می دهیم. خروجی به صورت زیر است.



از این قسمت به بعد، خروجی ها نسبت به دو حالت قبل، افت محسوسی خواهند داشت. دلیل اصلی آن هم همان طور که ذکر شد، تغییر مرتبه سیستم از 1 به 0 است.



برای TSK مرتبه صفر با 3 تابع عضویت ورودی، خروجی به صورت زیر است.

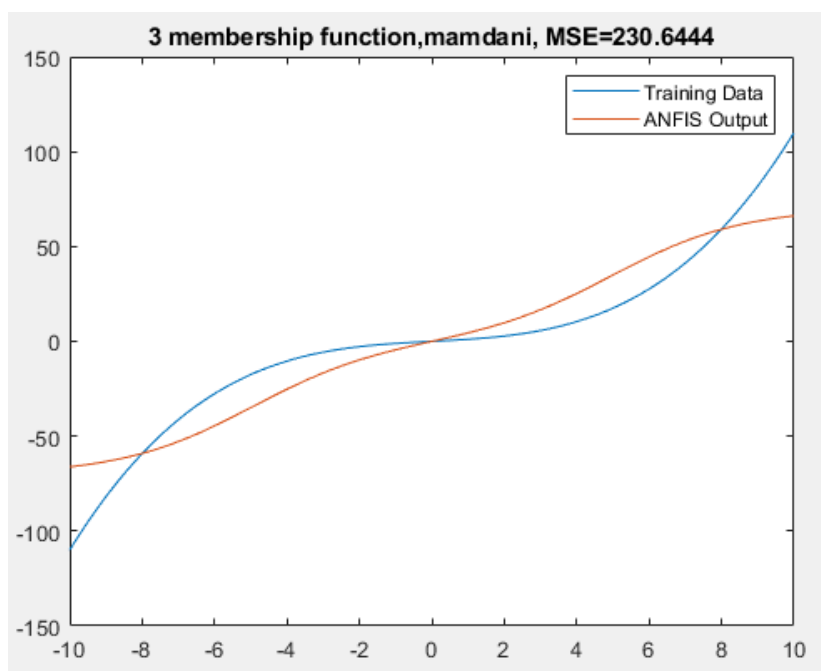


همان طور که می بینیم، خروجی اصلاً مطلوب نبوده و تخمین نامناسبی در این حالت داریم.



برای پیاده سازی سیستم ممدانی، ابتدا باید توجه کنیم که ورودی Anfis فقط می تواند سیستم TSK باشد. بنابراین با تغییر method های سیستم TSK، سیستم مطلوب را می سازیم. شیوه And کردن سیستم را به min و شیوه Or کردن را به max تغییر می دهیم. در ادامه نیز type خروجی سیستم را به constant تغییر می دهیم.

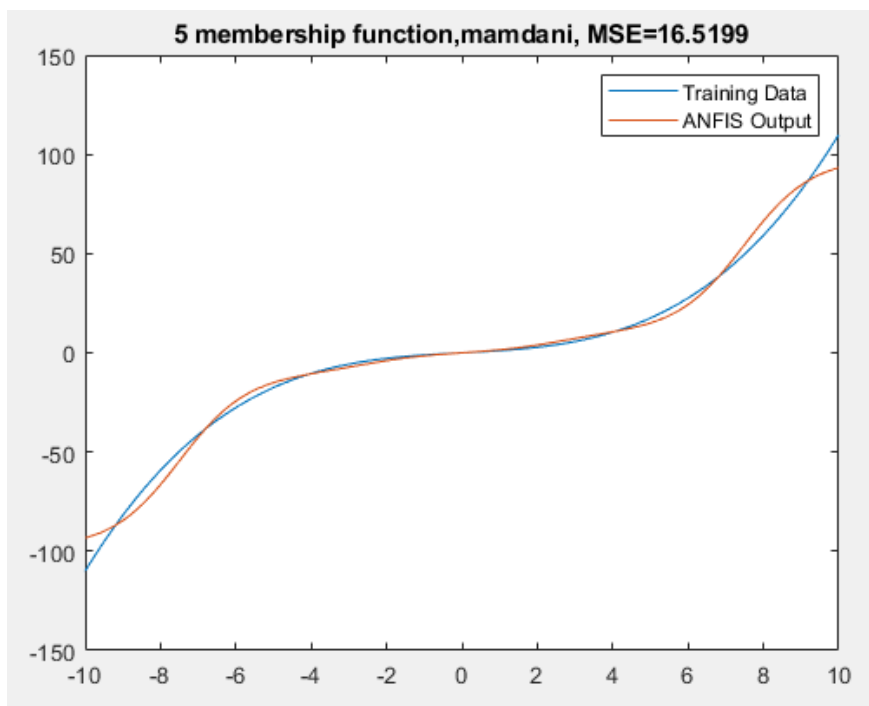
خروجی برای سیستم ممدانی با 3 تابع عضویت ورودی به صورت زیر است.



همان طور که می توان دید، خروجی چندان مطلوبی برای این قسمت نداریم.



برای ممدانی با 5 تابع عضویت ورودی، خروجی به صورت زیر است.



خروجی نسبت به حالت قبل بهتر است.

پس به طور کلی، TSK های مرتبه 1 خروجی های بسیار مطلوبی برای تخمین توابع داشتند. بعد از آن TSK های مرتبه صفر خروجی بهتری داشتند. در آخر هم سیستم ممدانی نسبت به دو سیستم دیگر قرار داشت. البته لازم به ذکر است که در ابعاد بالا، سیستم های ممدانی سادگی خیلی بیشتری نسبت به سیستم های TSK دارند که ممکن است آن ها را قابل استفاده کند. ولی به طور کلی سیستم های TSK، آزادی عمل خیلی بیشتری دارند.