



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پروژه شبکه عصبی

شبکه عصبی پرسپترون چندلایه (MLP)

محمد رضا دولتی

۹۷۱۱۰۴۱۱

استاد:

دکتر آرش عبداله جراندوست

۱۴۰۰ بهمن

مقدمه:

برای انجام این پروژه از نرم افزار مطلب استفاده شده است. برای آموزش شبکه عصبی ابتدا واژه nftool در بخش command window وارد می شود تا محیط آموزش شبکه عصبی بازگردد. با اجرای آموزش شبکه عصبی و در انتهای، کدی که در آن می شود تنظیمات شبکه عصبی را تغییر داد را می توان استخراج نمود. از آنجاکه در برخی موارد نیاز است تا تنظیمات تغییر داده شود، از این کد استفاده می نماییم که با اجرای آن شبکه عصبی موردنظر با انتخاب ورودی ها و خروجی ها آموزش داده می شود.

بخش اول:

در این بخش ۶تابع برای آموزش در شبکه عصبی در نظر گرفته شده است. بازه داده های x برای هر تابع از $10 - 10$ و تعداد ورودی ها 100 در نظر گرفته شده است. در ادامه برای هر یک از این توابع، به بررسی پارامترهای خواسته شده، پرداخته شده است. به این صورت که پارامتر موردنظر تغییر می کند و بقیه پارامترها ثابت می مانند و نتایج باحال است اصلی مقایسه می گردد. در ادامه توابع از ساده به پیچیده آورده شده است. در هر مورد ابتدا تابع اصلی با تعداد ورودی 100 و در بازه $10 - 10$ آموزش داده می شود. این آموزش با چند آزمون و خطا، ساختاری از شبکه عصبی (لایه های پنهان) که نتایج مناسبی را دارد را شناسایی می کنیم. سپس اثر افزایش و کاهش نمونه ورودی را در آن می بینیم.

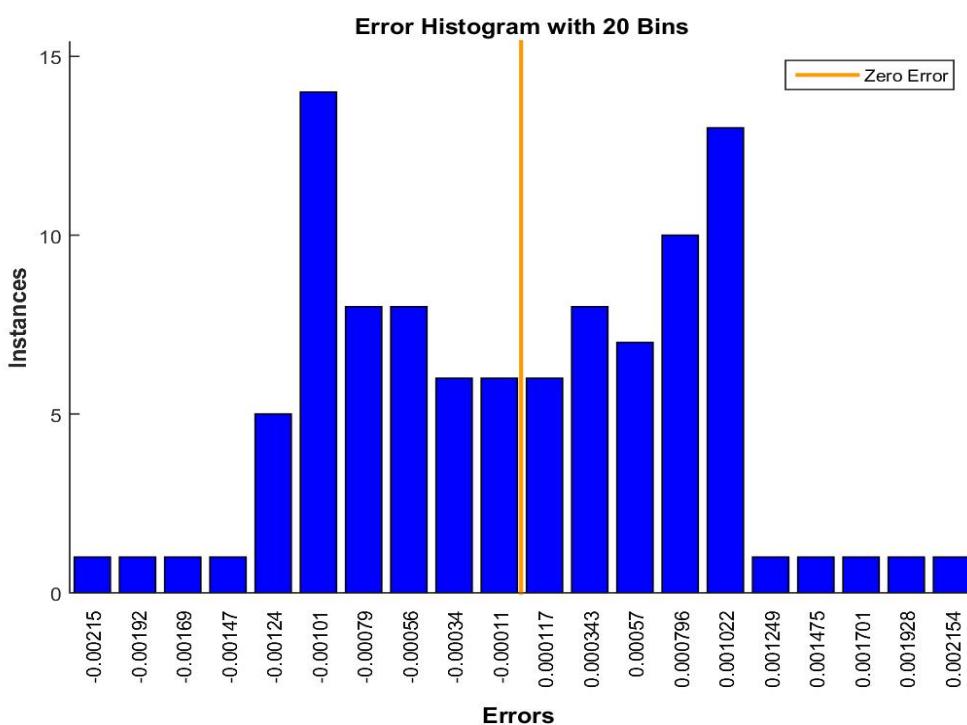
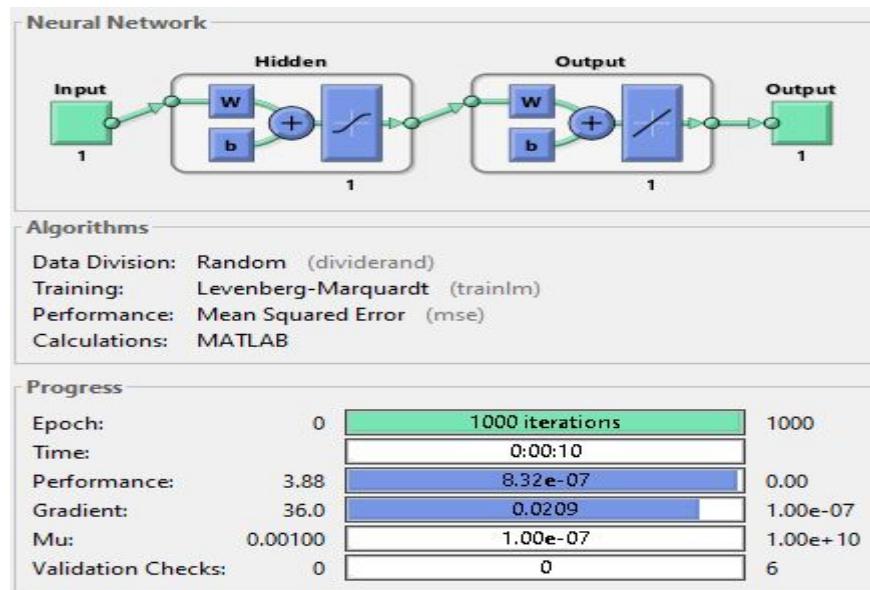
به ازای هر آموزش که 4 عکس خروجی گرفته می شود. عکس اول ساختار شبکه عصبی و عملکرد آن، عکس دوم نمودار رگرسیون (همبستگی)، عکس سوم نمودار خطای خروجی تابع و خروجی شبکه عصبی و عکس چهارم هم نمودار خروجی شبکه عصبی و نمونه های آموزش را نشان می دهد و فاصله هر نمونه از مقدار تخمینی توسط شبکه عصبی در آن مشخص است.

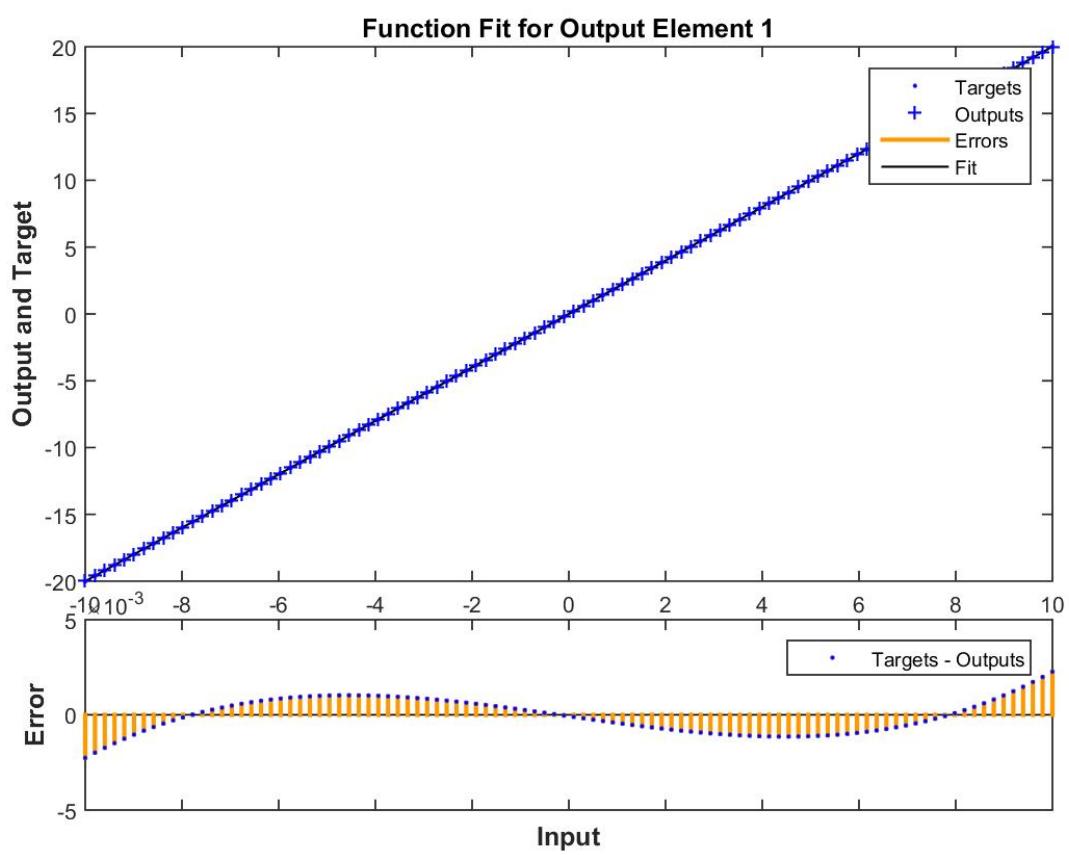
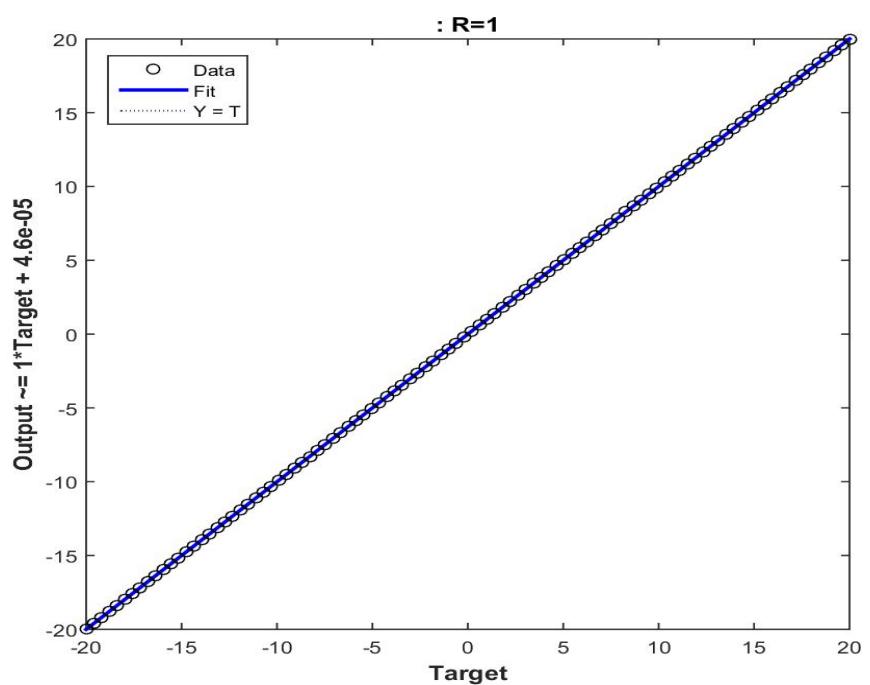
تابع اول:

$$y_1 = 2 * x;$$

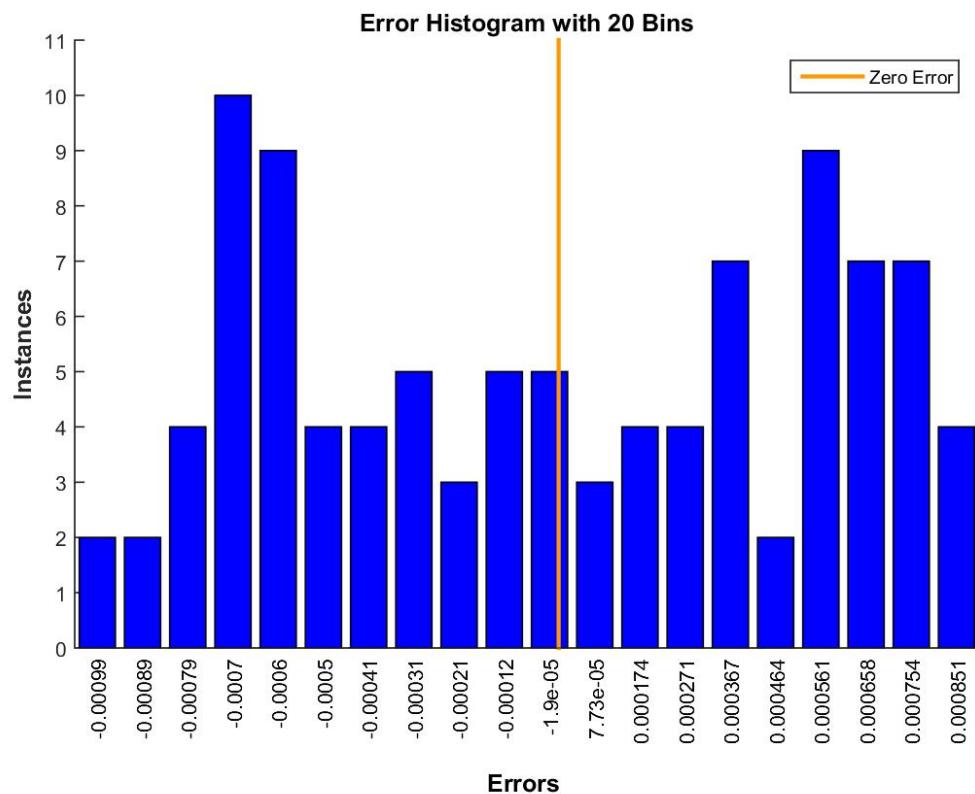
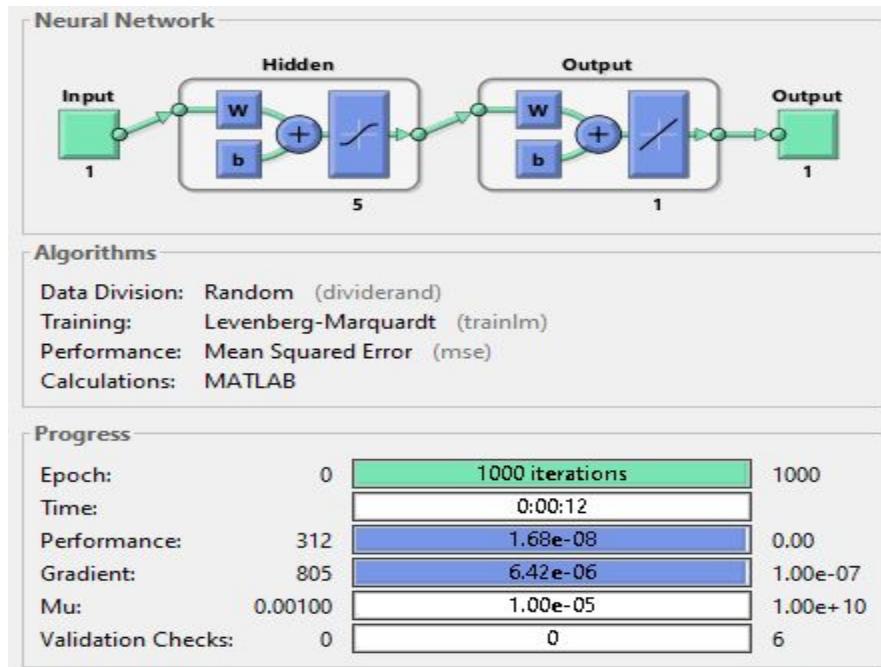
از این تابع به عنوان ساده ترین تابع مورد بررسی برای آموزش شبکه عصبی استفاده می کنیم. تعداد نمونه های حالت اول و دوم برابر با 100 است.

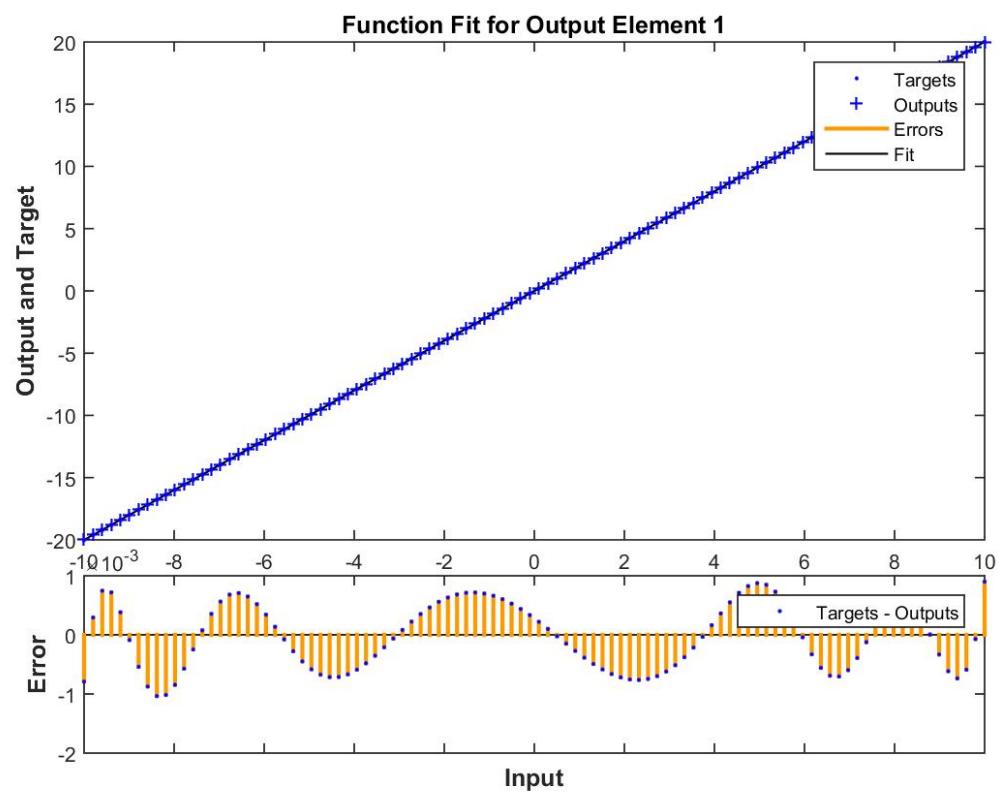
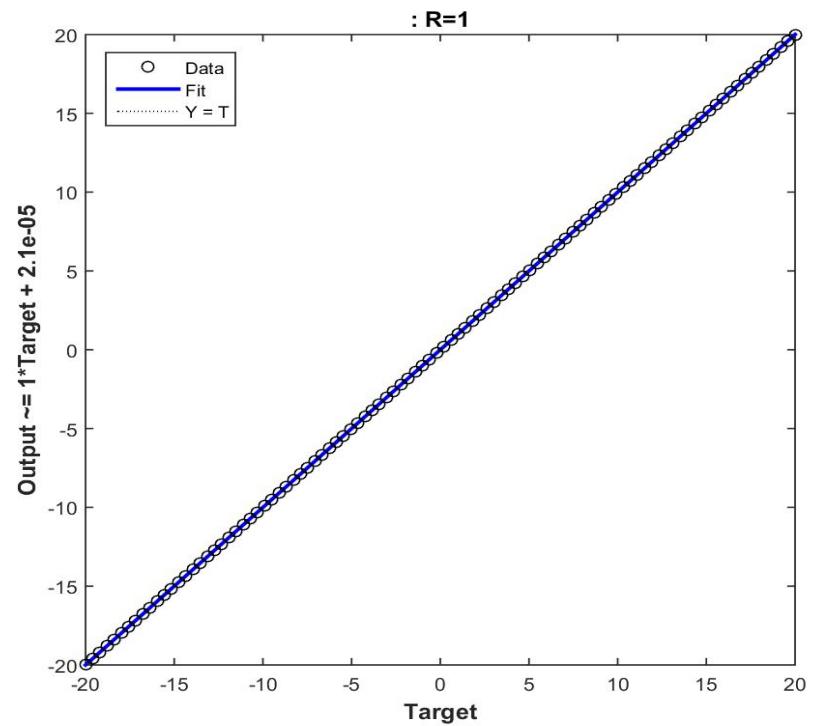
حالت اول: در این حالت تعداد نورون‌های لایه پنهان ۱ در نظر گرفته می‌شود. چون تابع ساده است و یک تابع خطی است، نتایج شبکه عصبی با توجه به نمودارهای رگرسیون و خطای همچنین ترسیم خروجی شبکه عصبی و نمونه‌ها، دقیق و قابل قبول است. در ادامه به ترتیب تصاویر ساختار و عملکرد شبکه عصبی، نمودار خطاهای، نمودار رگرسیون و خروجی شبکه عصبی و نمونه‌های آموزش آورده شده است.



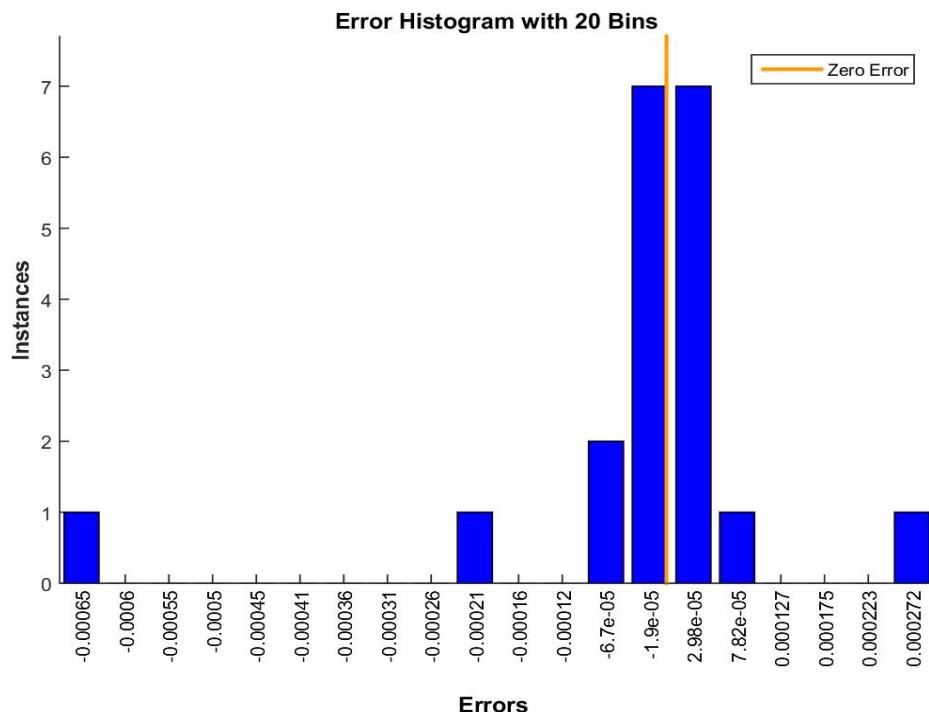
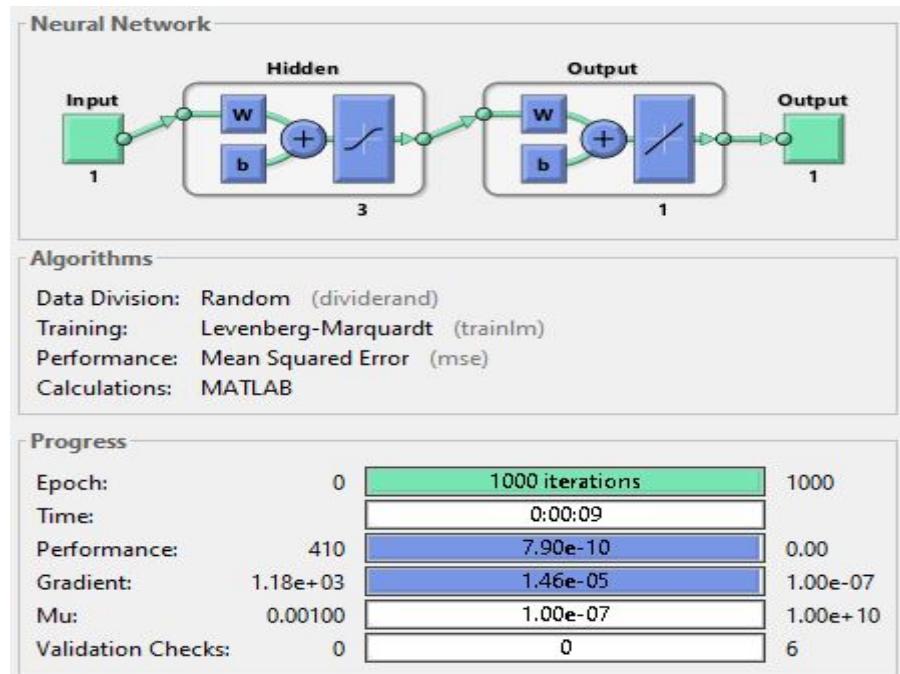


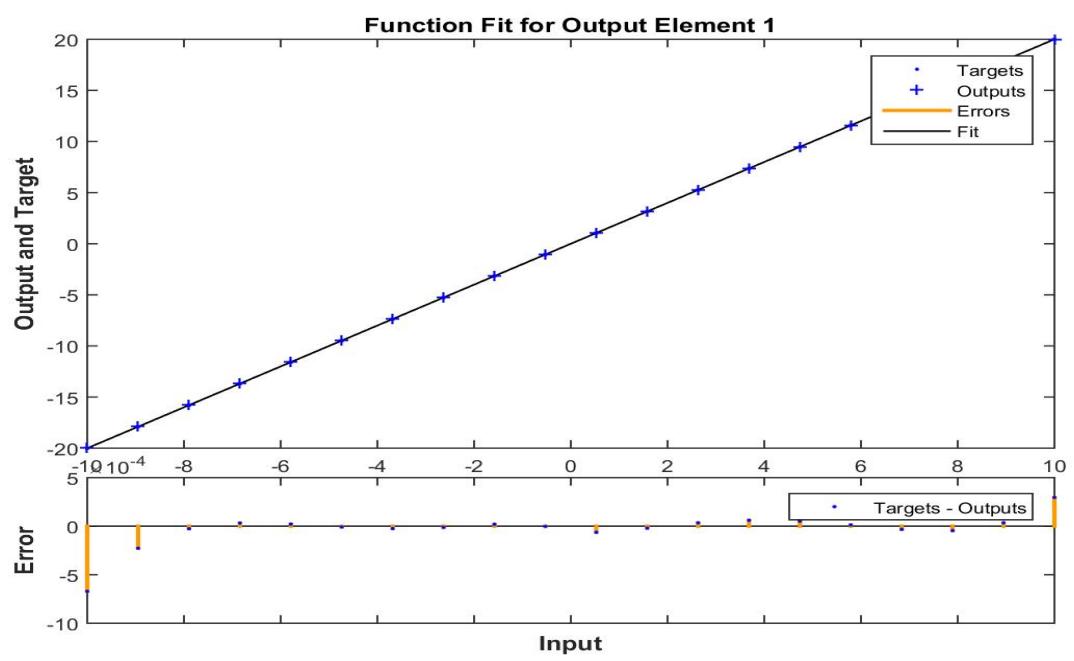
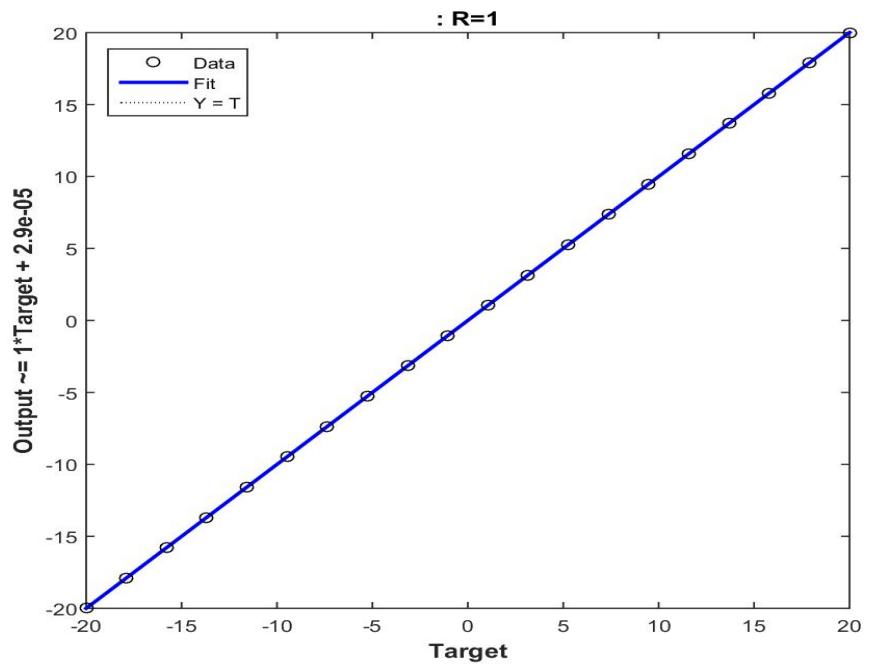
حالت دوم: در حالت دوم تعداد نورون‌های لایه پنهان ۵ در نظر گرفته شده است؛ اما همان‌طور که مشاهده می‌شود تأثیری در بهبود عملکرد شبکه عصبی ندارد.



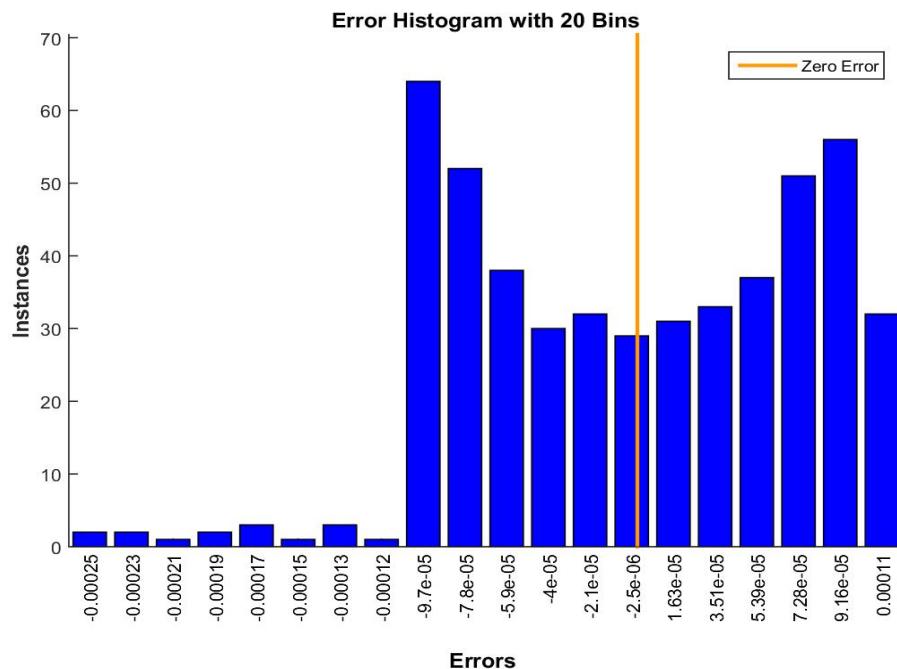
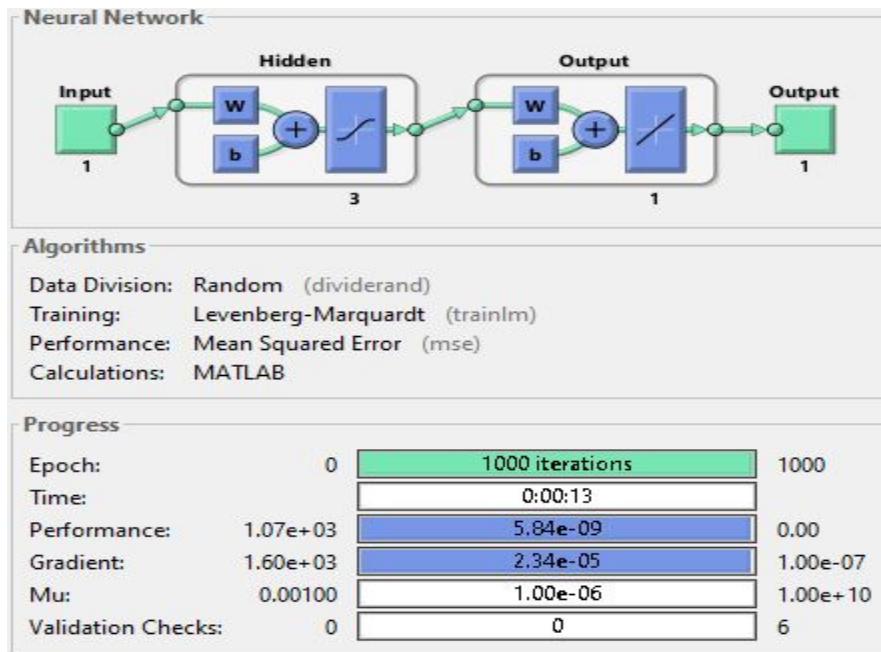


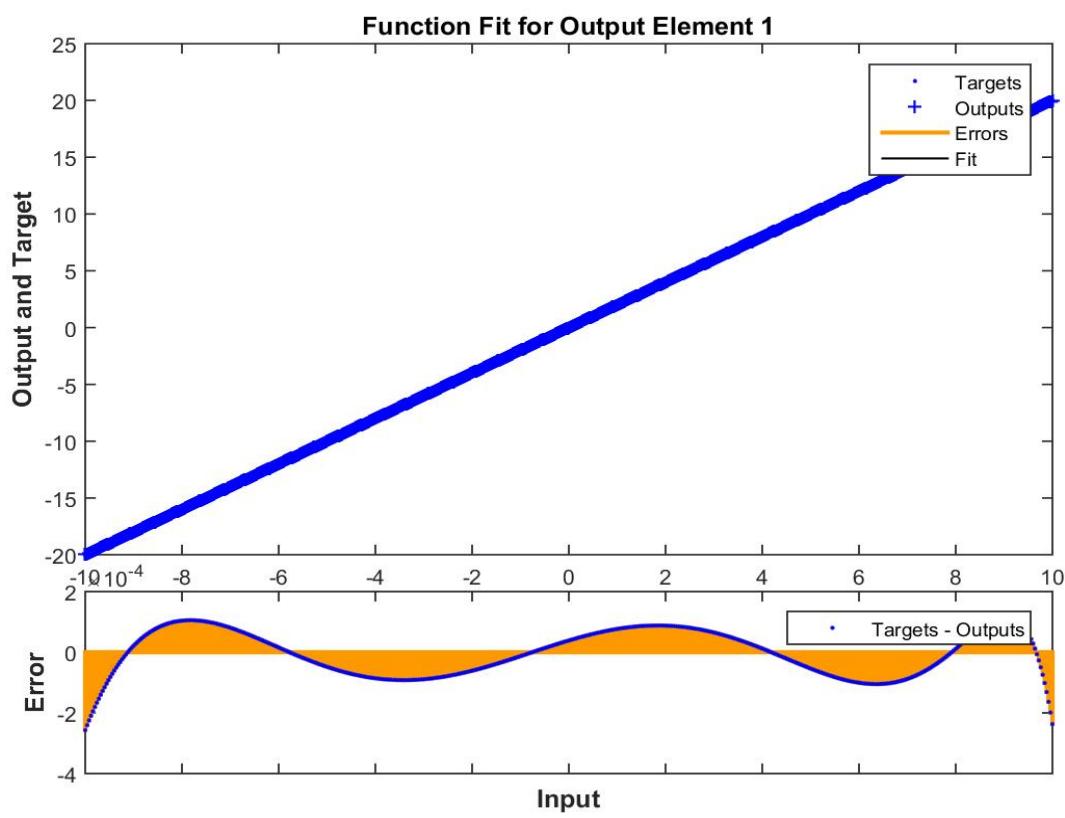
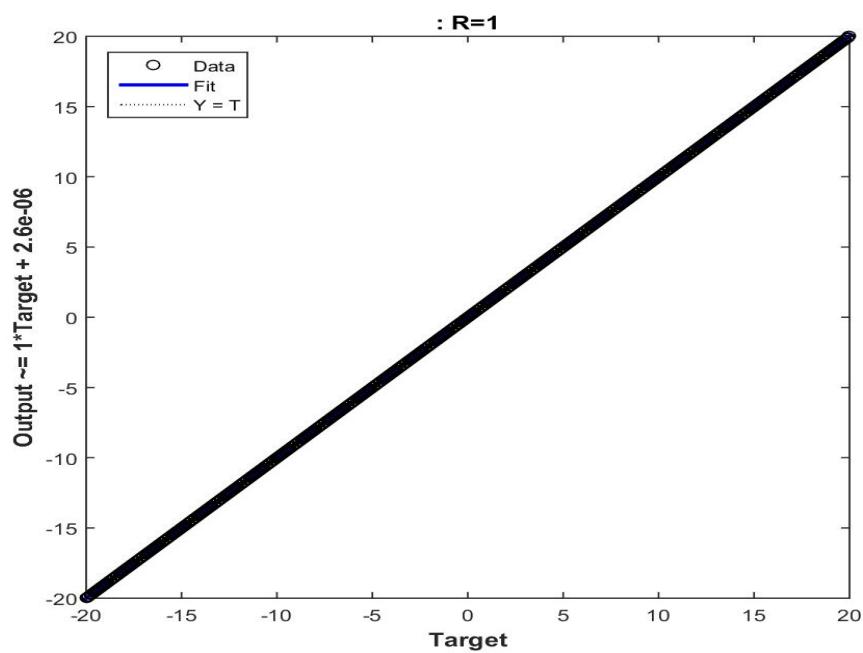
حالت سوم: در این حالت تعداد نمونه‌های آموزش برابر با ۲۰ در نظر گرفته می‌شود. همانطورکه مشاهده می‌شود به خاطر خطی بودنتابع عملکرد شبکه عصبی تضعیف نمی‌شود.





حالت چهارم: در این حالت تعداد نمونه‌های آموزش برابر با 500 در نظر گرفته شده است. همان‌طور که مشاهده می‌شود خطایها در رنج 10^{-4} و قابل قبول است.

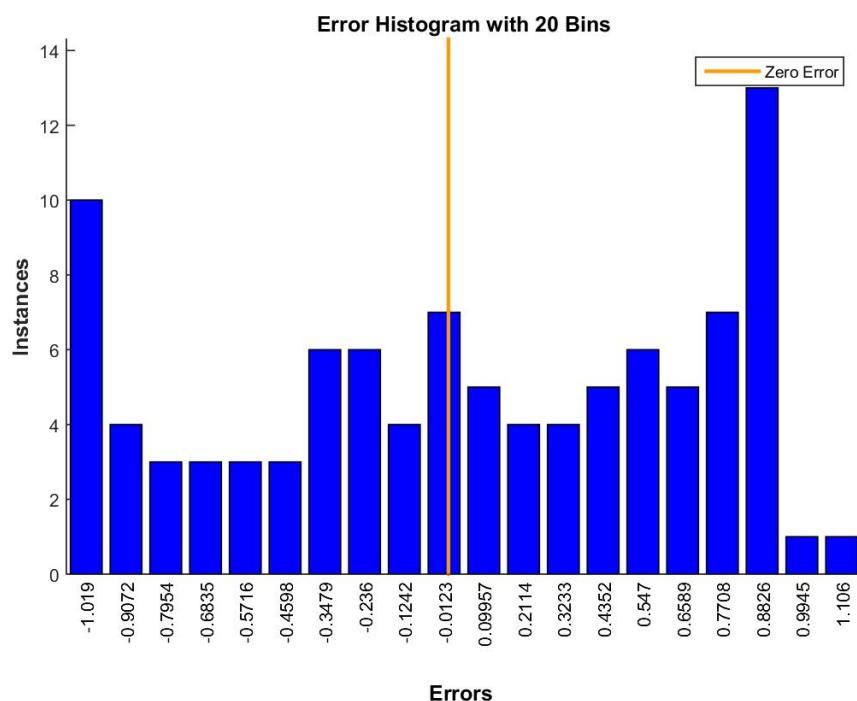
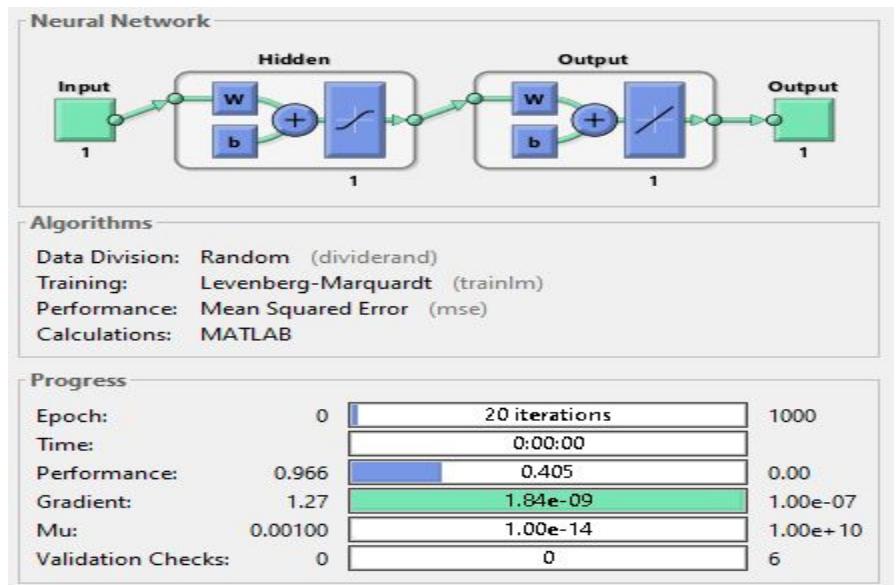


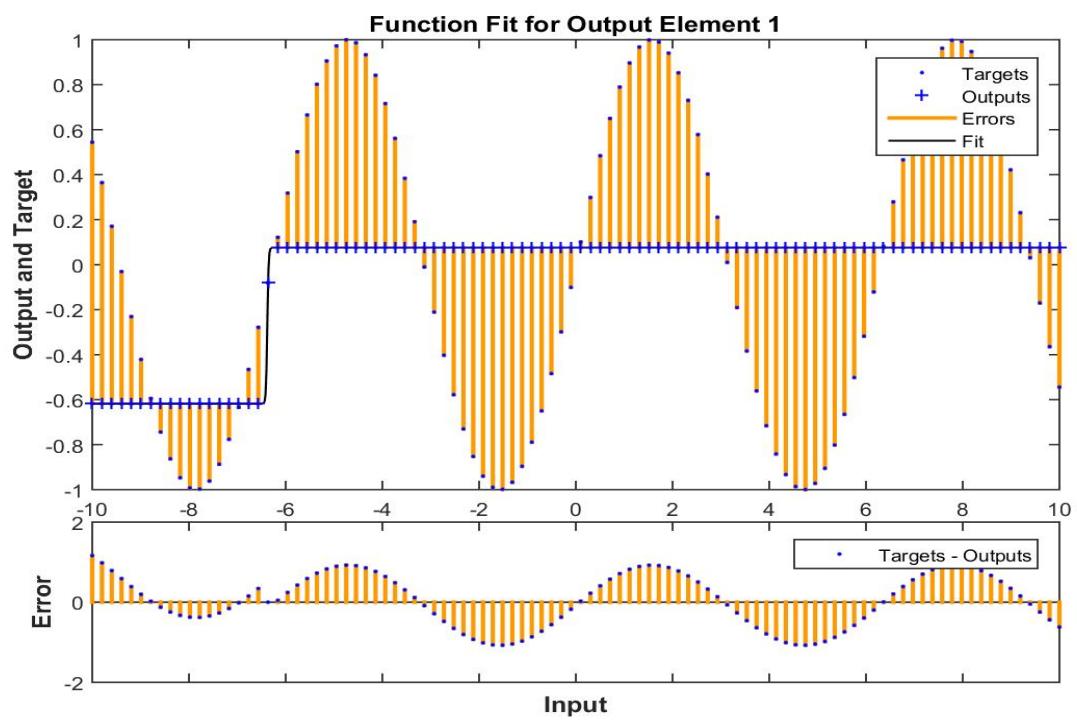
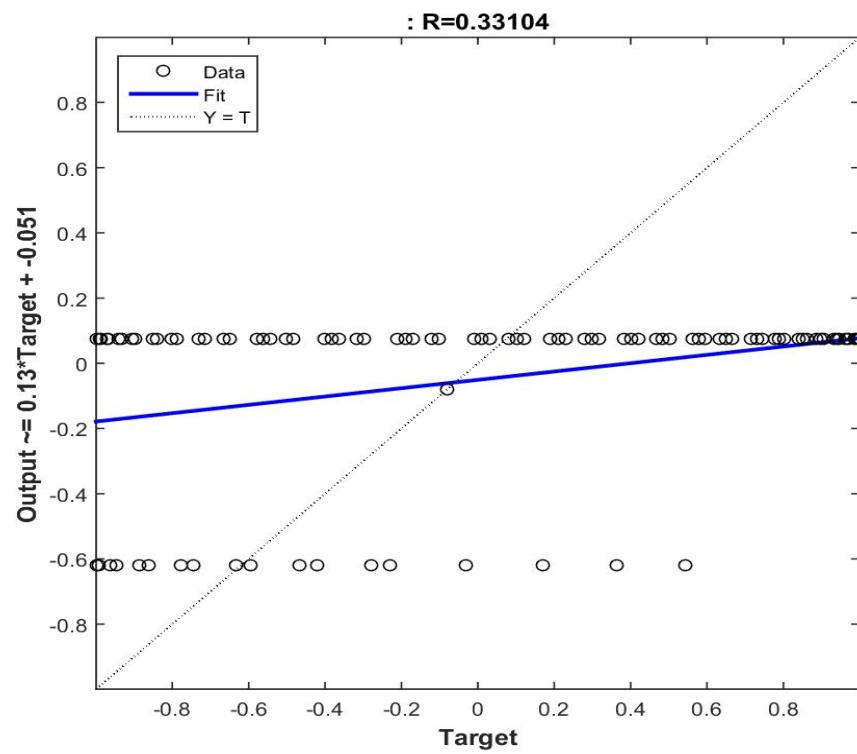


تابع دوم:

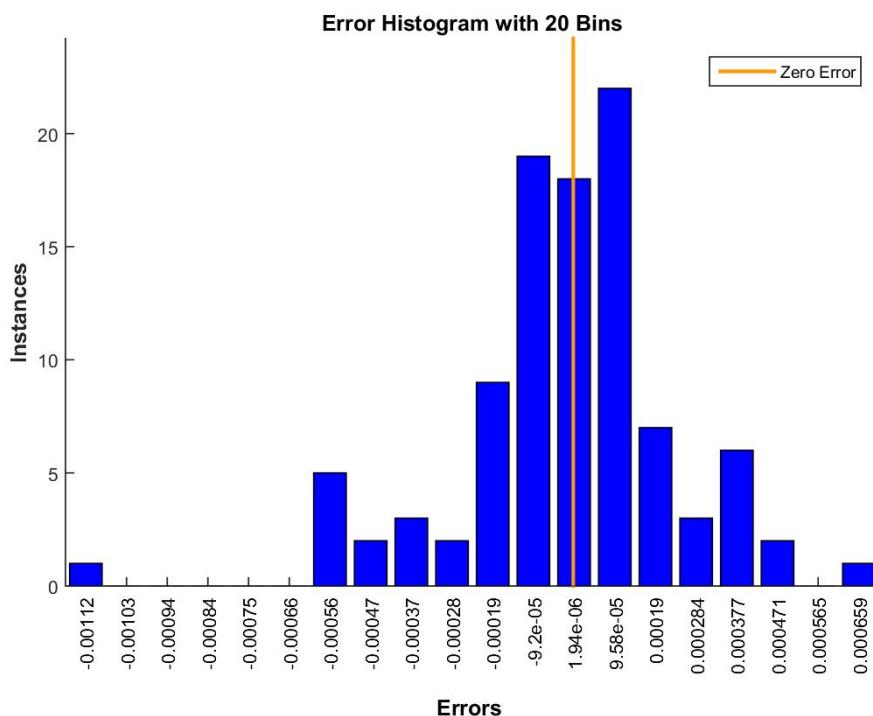
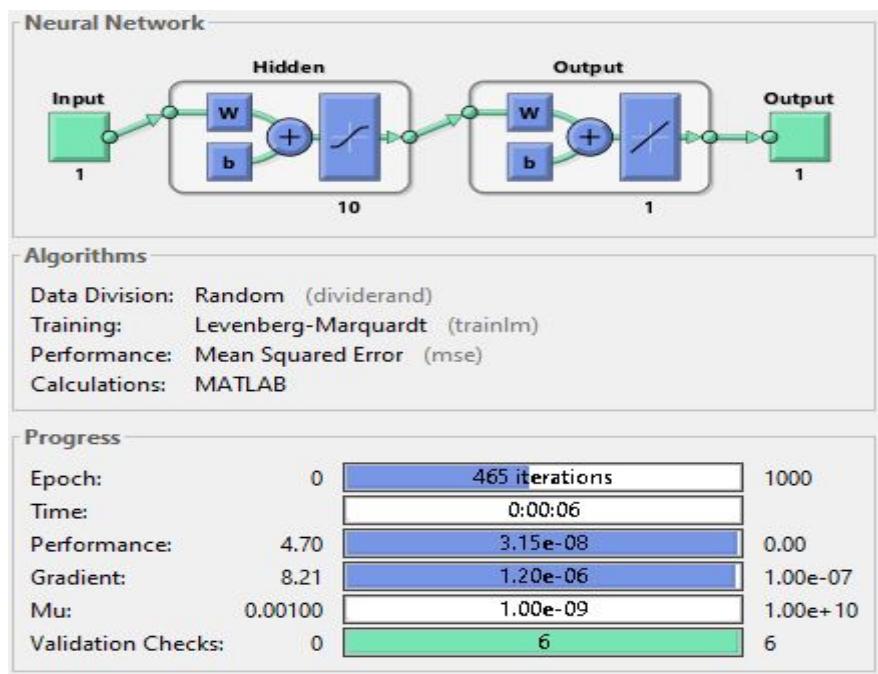
$$y_2 = \sin(x);$$

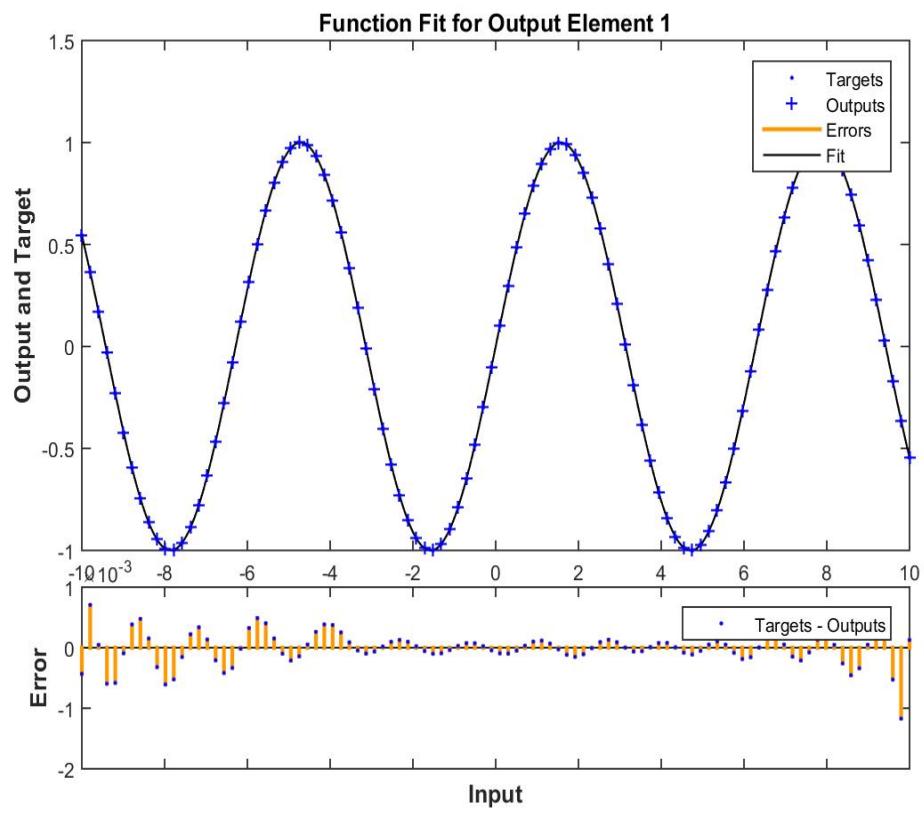
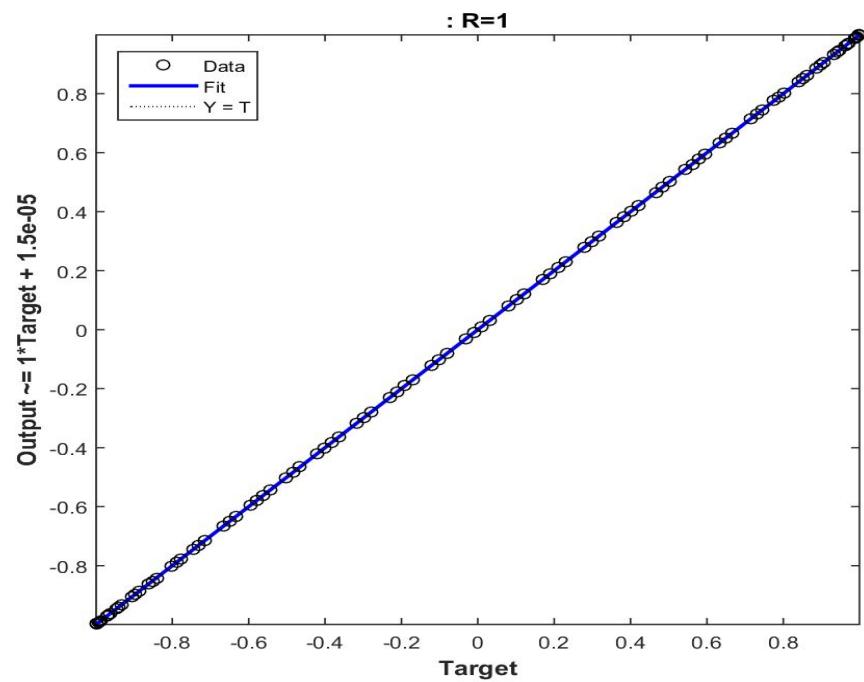
حالت اول: در این حالت ساختار شبکه عصبی را با یک لایه پنهان و یک نورون در نظر می‌گیریم. همان‌طور که مشخص است در این حالت نتایج به دست آمده غیرقابل قبول است، چون برای این تابع نیاز به ساختار پیچیده‌تر شبکه عصبی است.



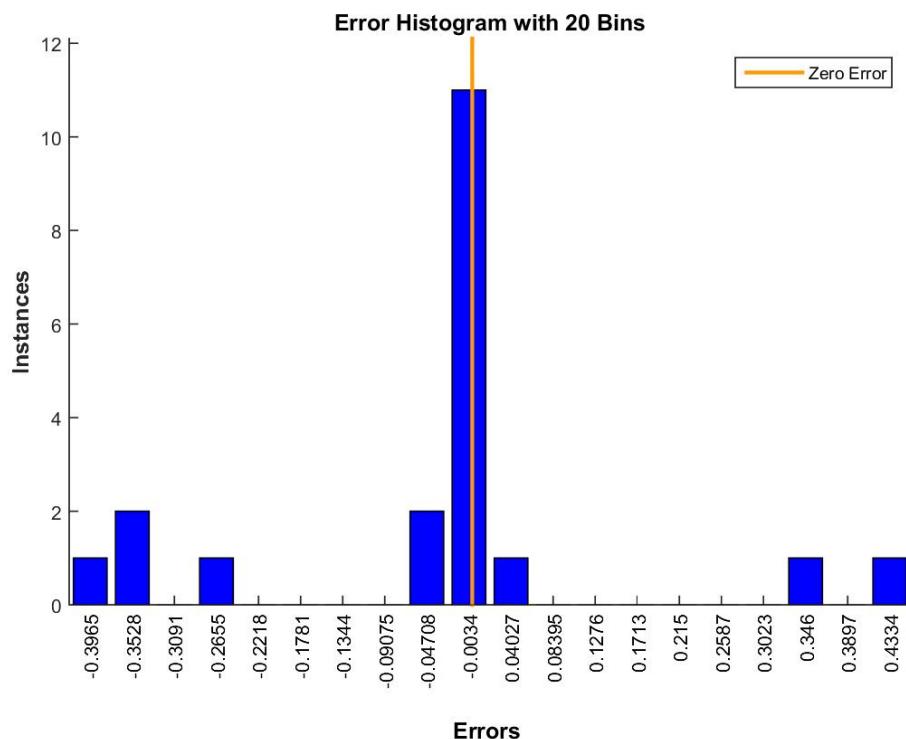
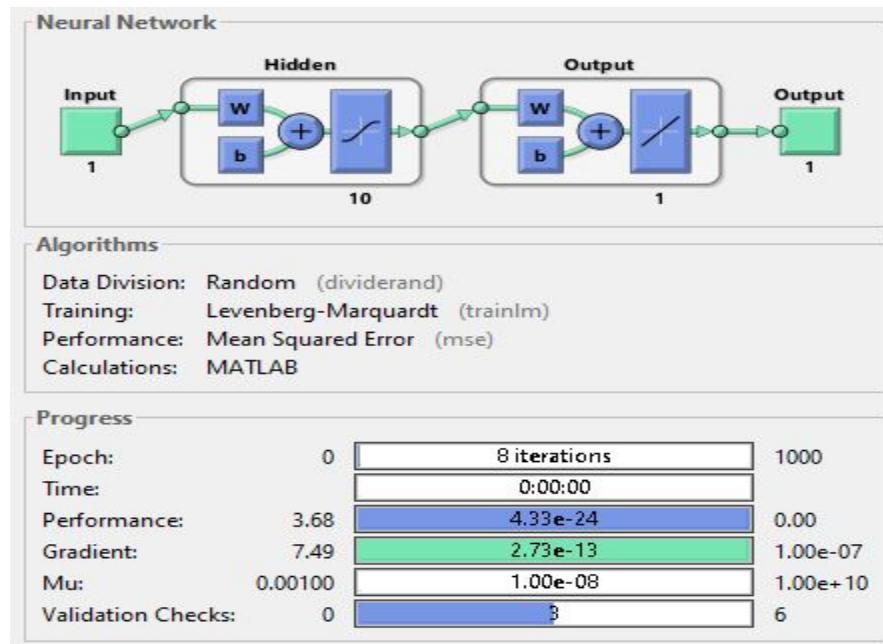


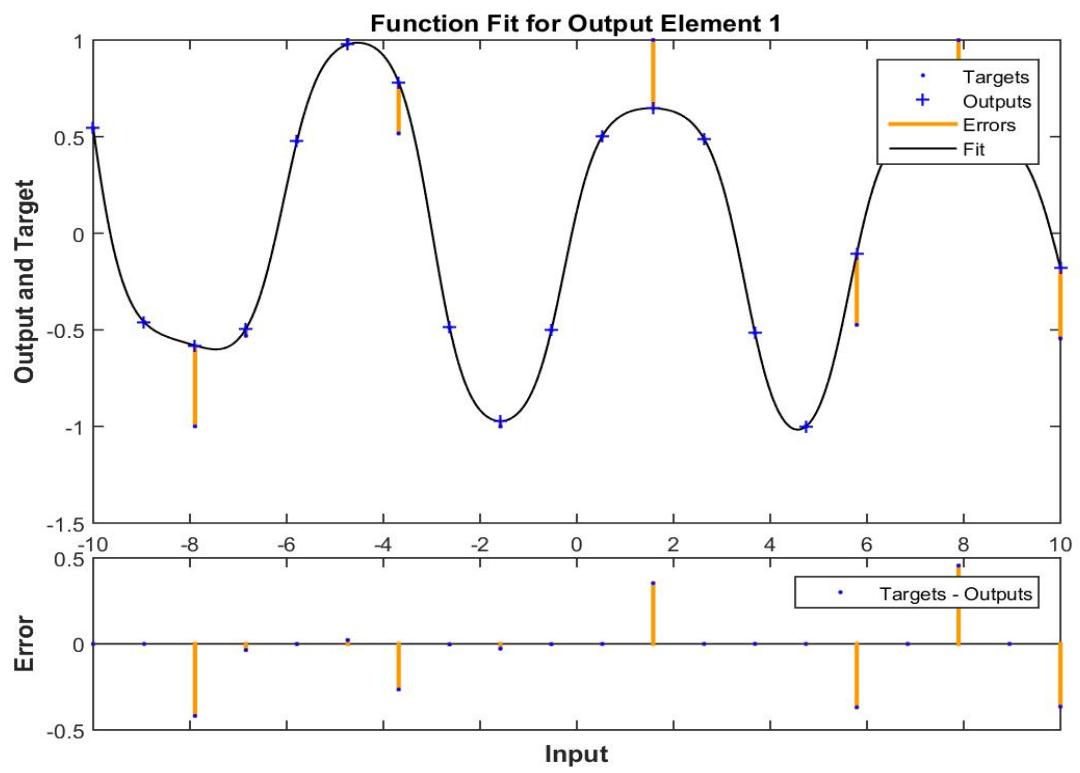
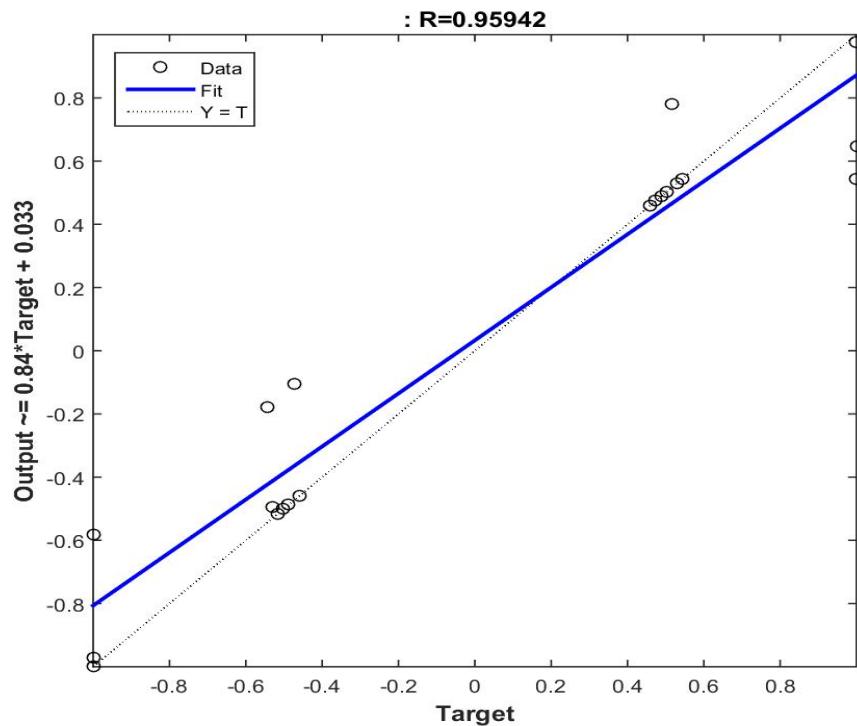
حالت دوم: در این حالت تعداد نورون‌های لایه مخفی را برابر با ۱۰ در نظر می‌گیریم. همان‌طور که مشخص است، نتایج به دست آمده از شبکه عصبی دقیق بالایی دارند و نمونه‌های آموزشی تقریباً روی نمودار به دست آمده از شبکه عصبی است.



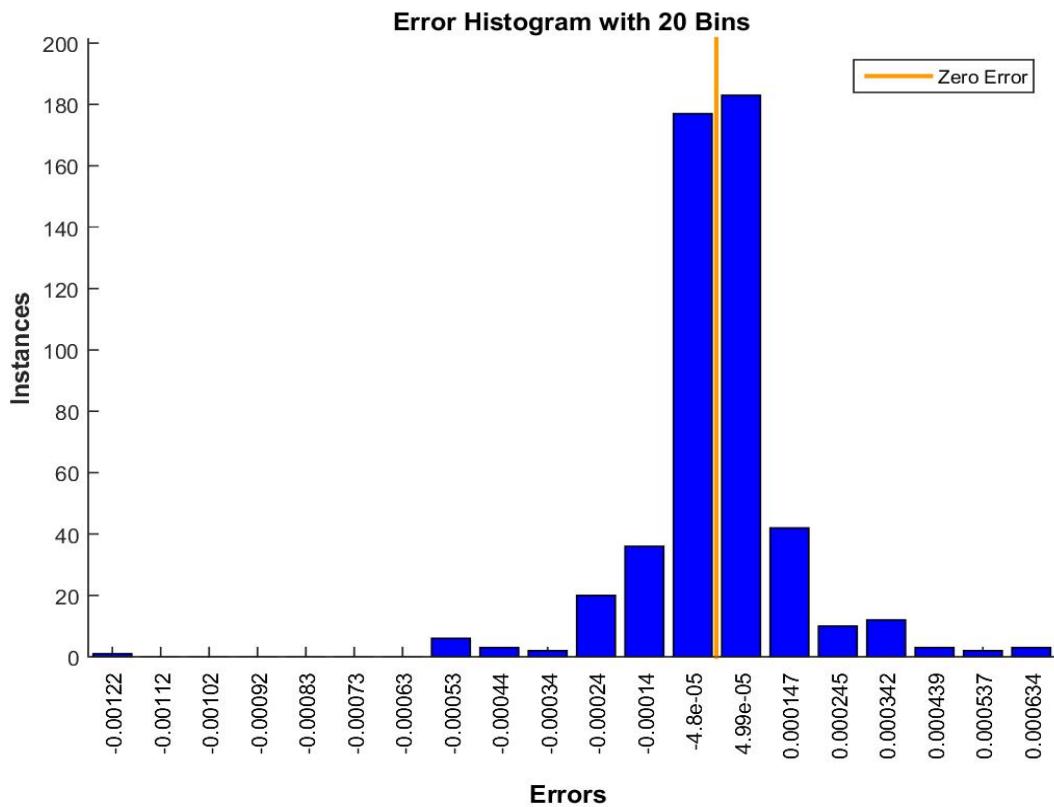
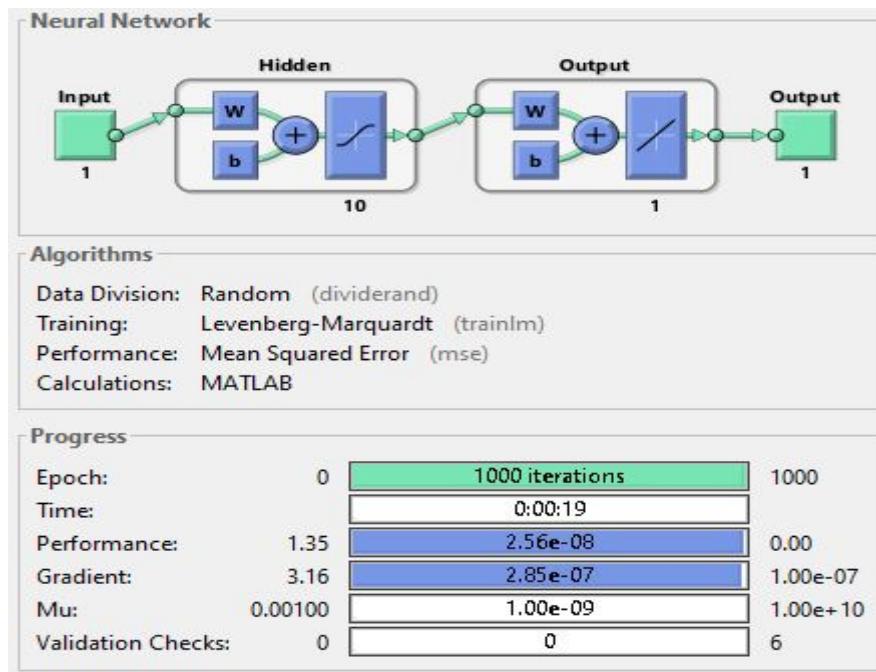


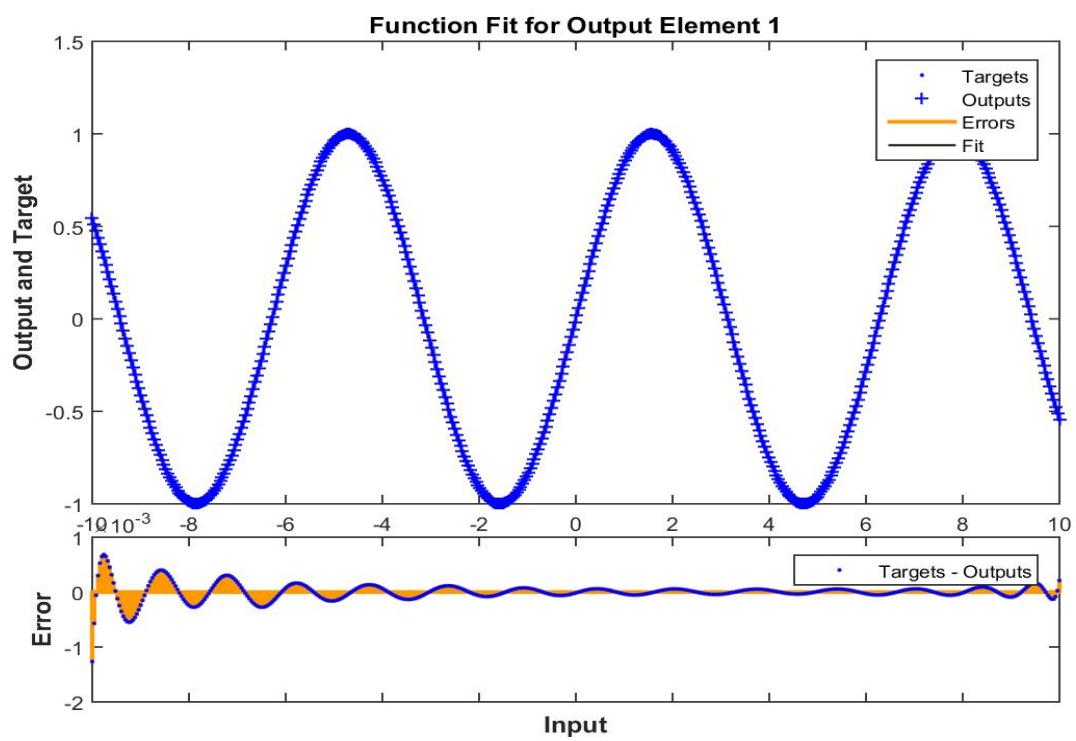
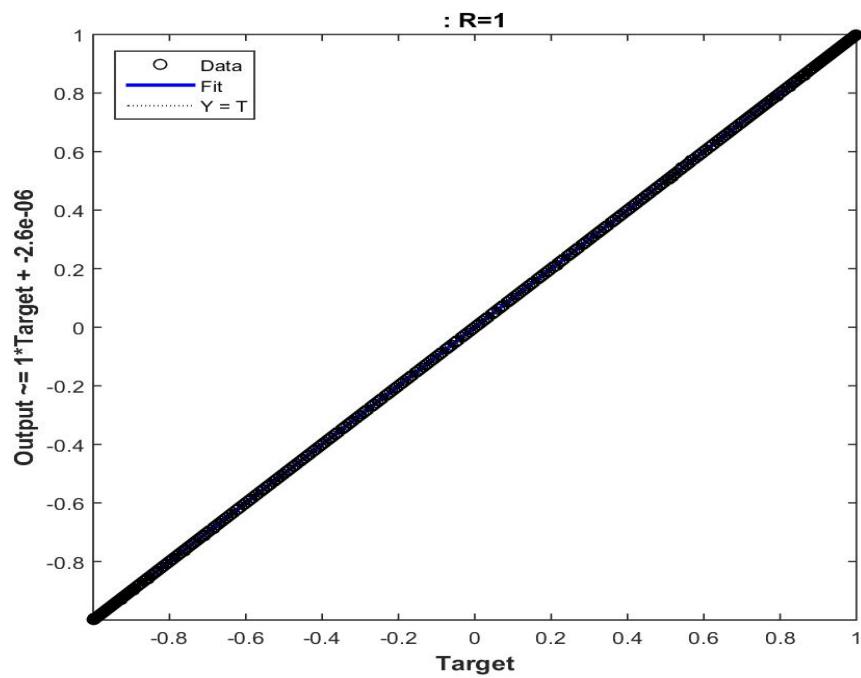
حالت سوم: در این حالت نمونه‌های آموزش را به ۲۰ کاهش می‌دهیم. همان‌طور که مشخص است دقت خروجی شبکه عصبی نسبت به حالت دوم که ۱۰۰ نمونه آموزش داشتیم خیلی کمتر است.





حالت چهارم: در این حالت تعداد ورودی‌های آموزش را به ۵۰۰ عدد افزایش می‌دهیم. در این حالت مشاهده می‌شود خروجی شبکه عصبی از ۲ حالت ۲۰ و ۱۰۰ نمونه دقیق‌تر است.

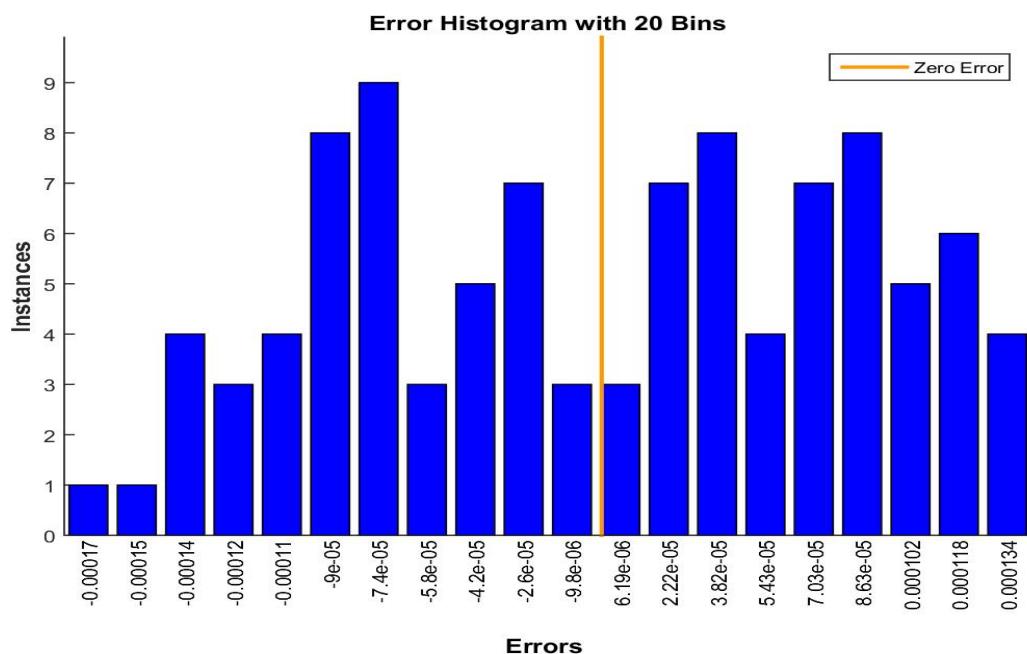
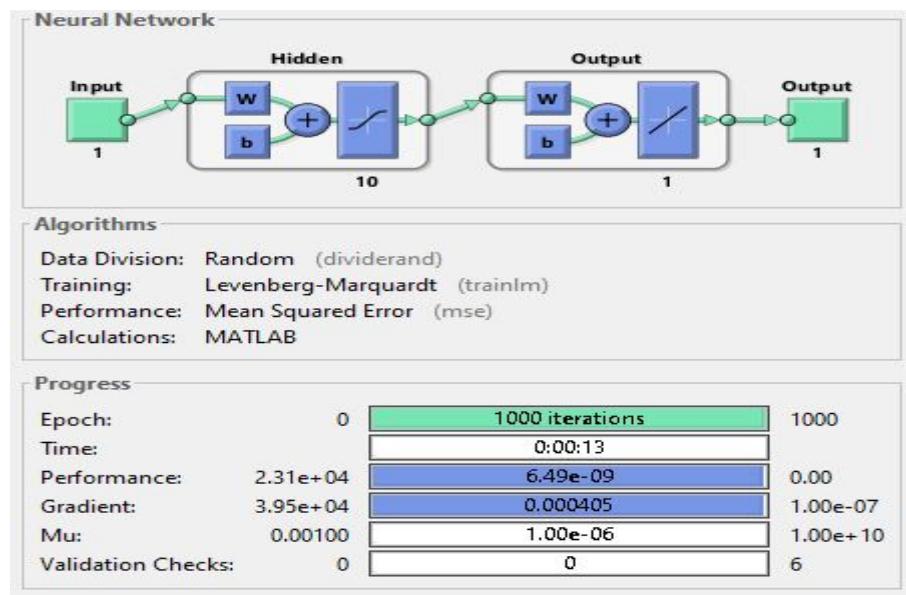


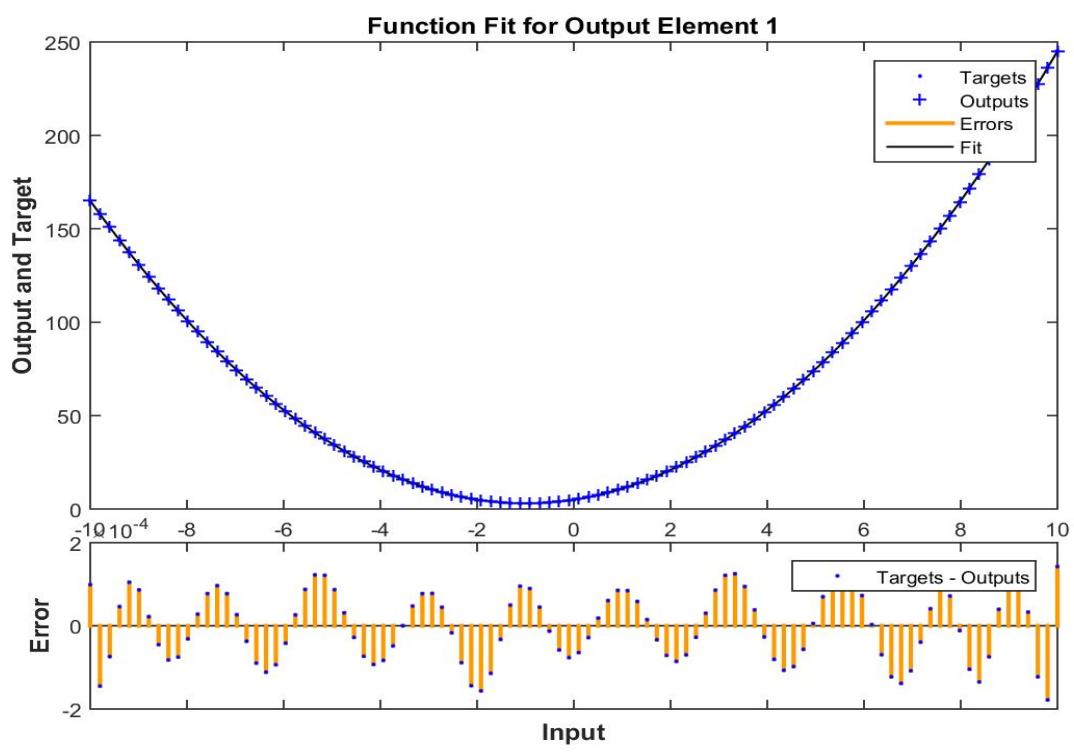
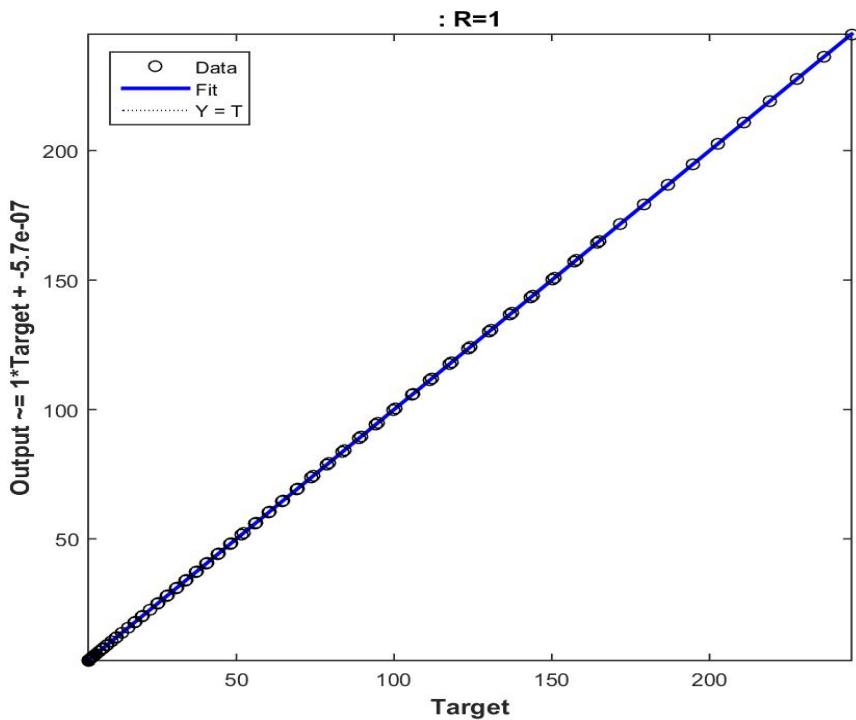


تابع سوم:

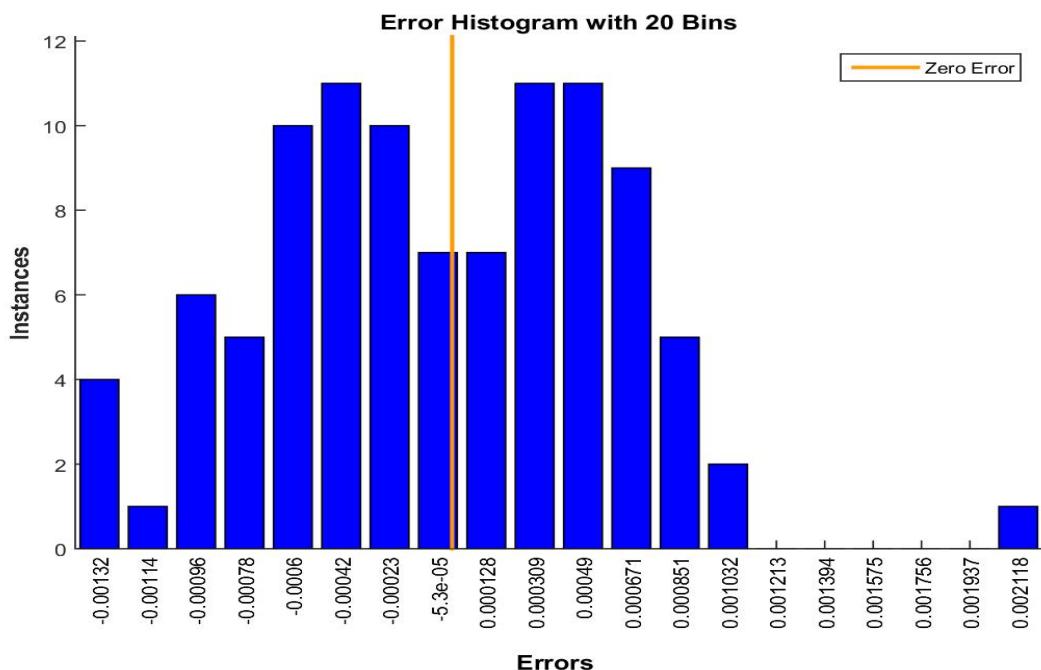
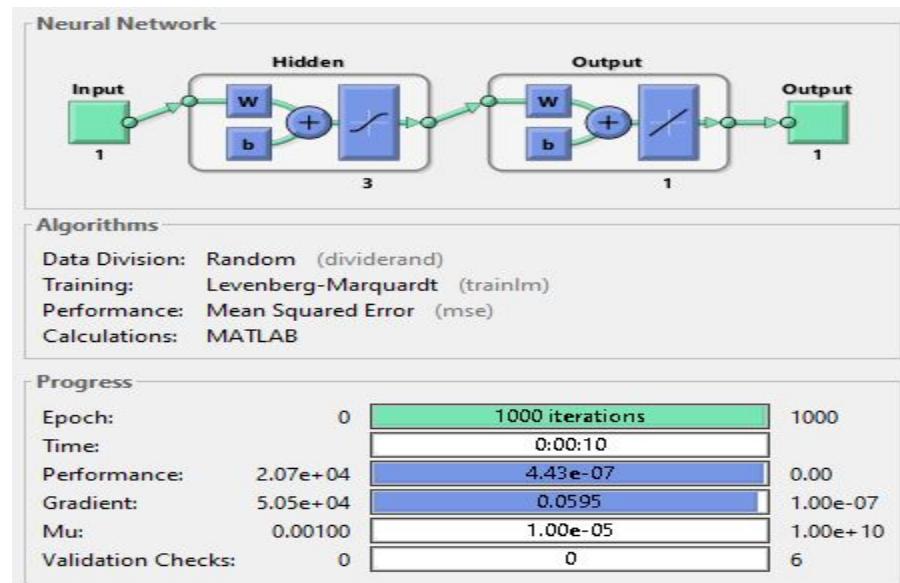
$$y_3 = 2 \cdot x^2 + 4 \cdot x + 5;$$

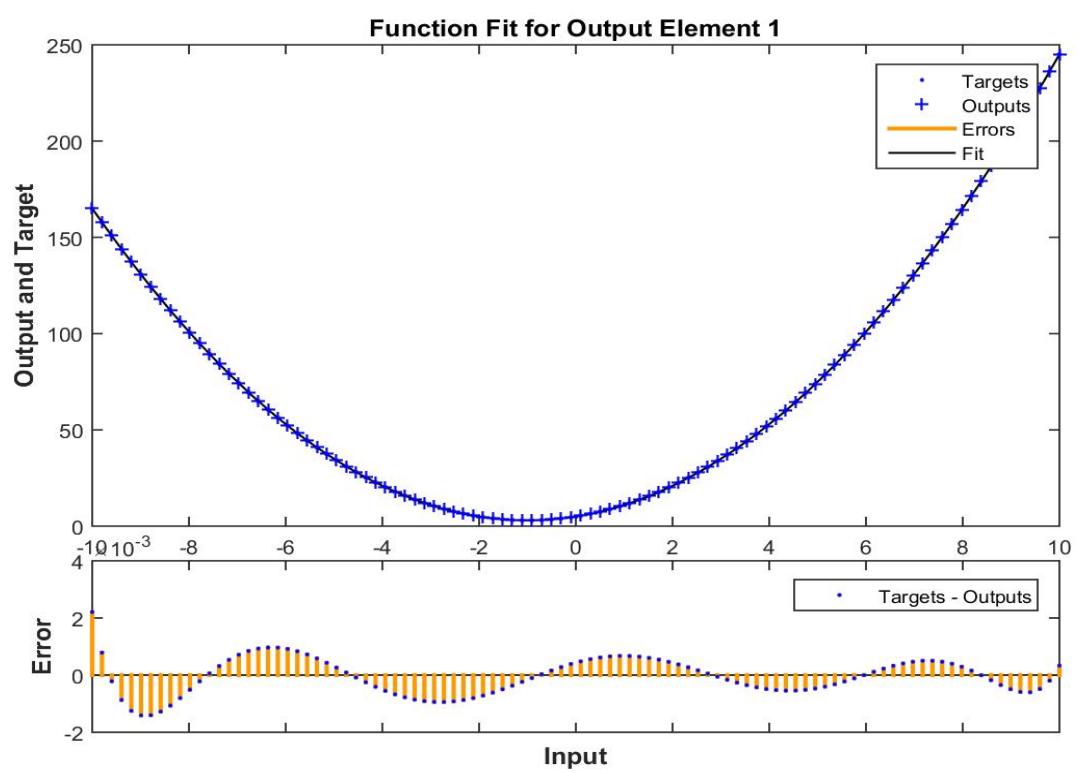
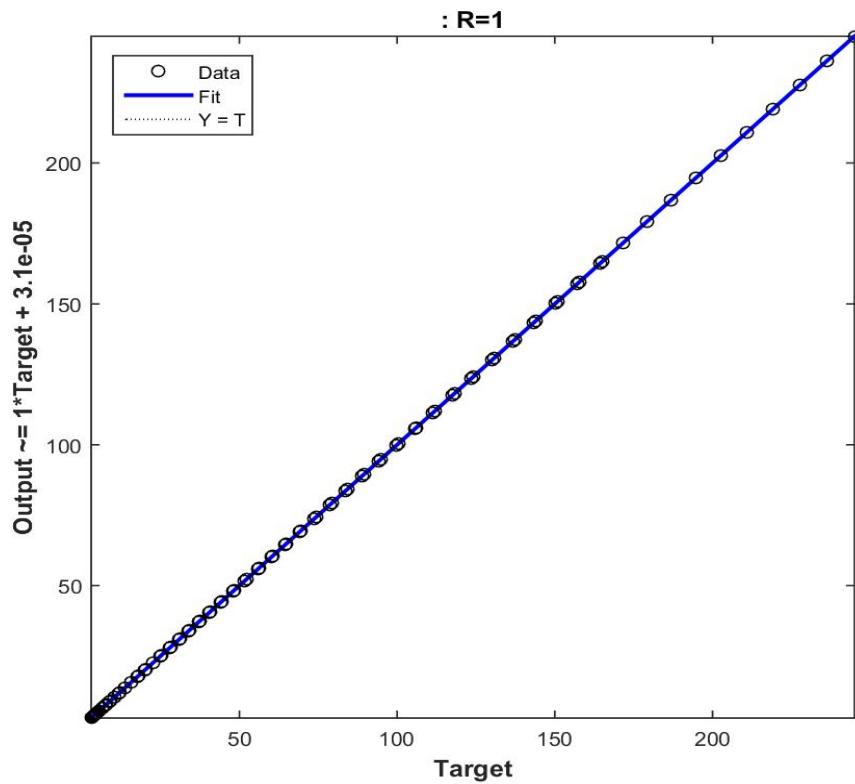
حالت اول: در این تابع ساختار با ۱۰ نورون در لایه مخفی در نظر گرفته می‌شود. مشخص است که نتایج دقت کافی را دارد هستند و نمونه‌های آموزش با خطای از رنج 10^{-4} روی نمودار به دست آمده از شبکه عصبی قرار می‌گیرند.



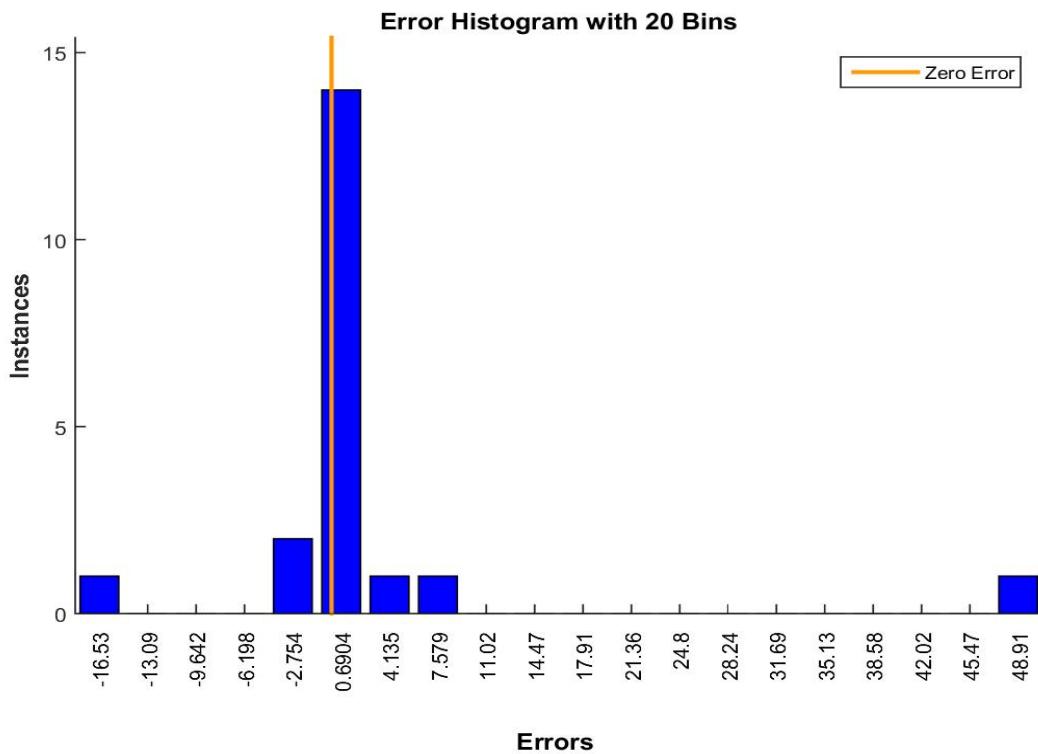
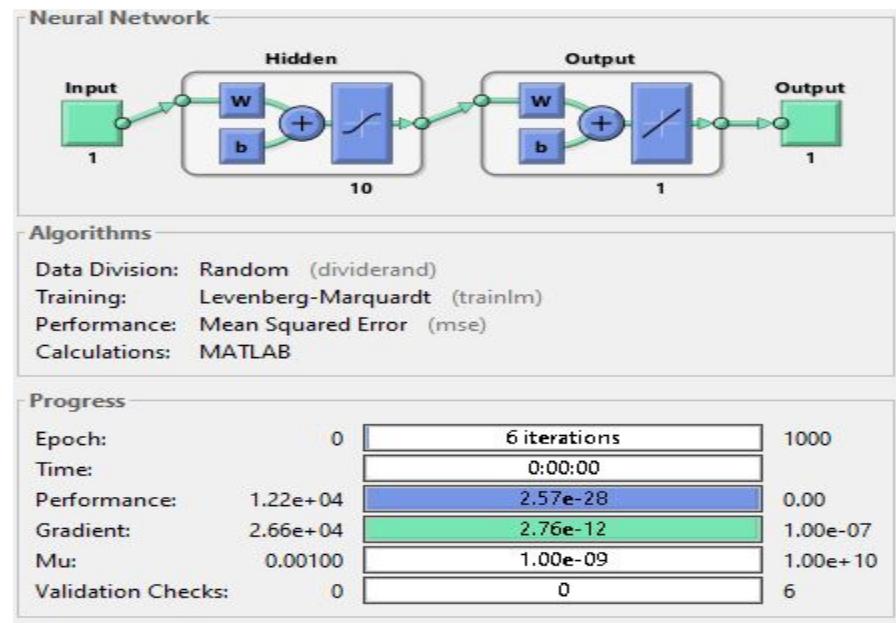


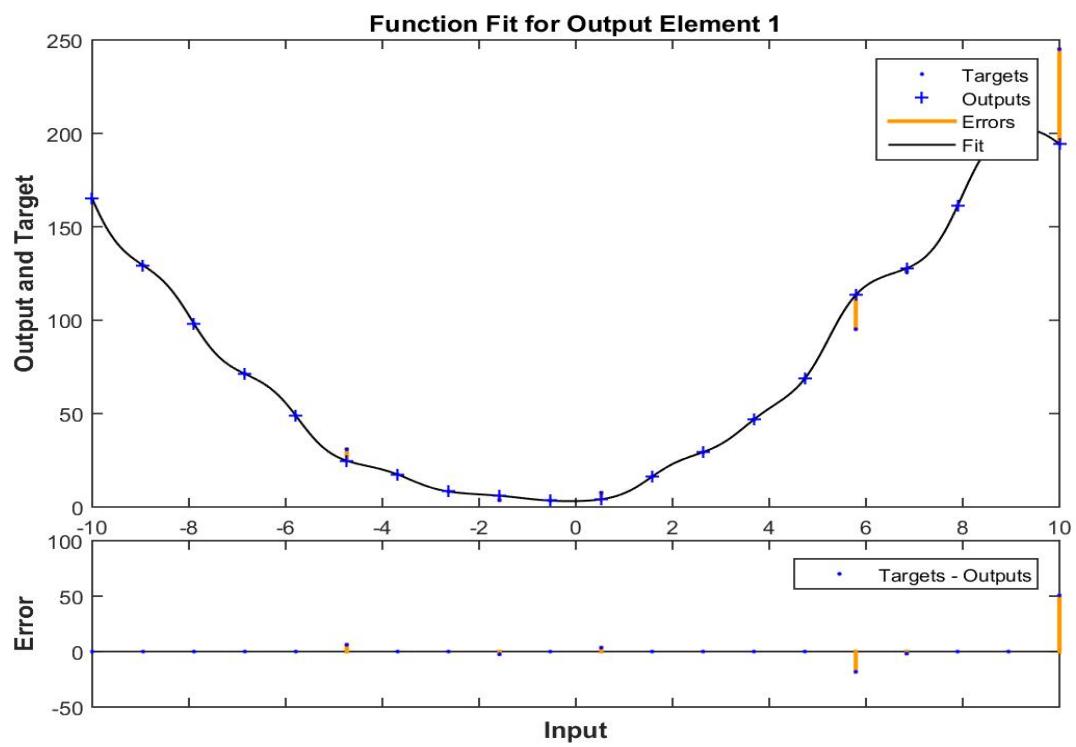
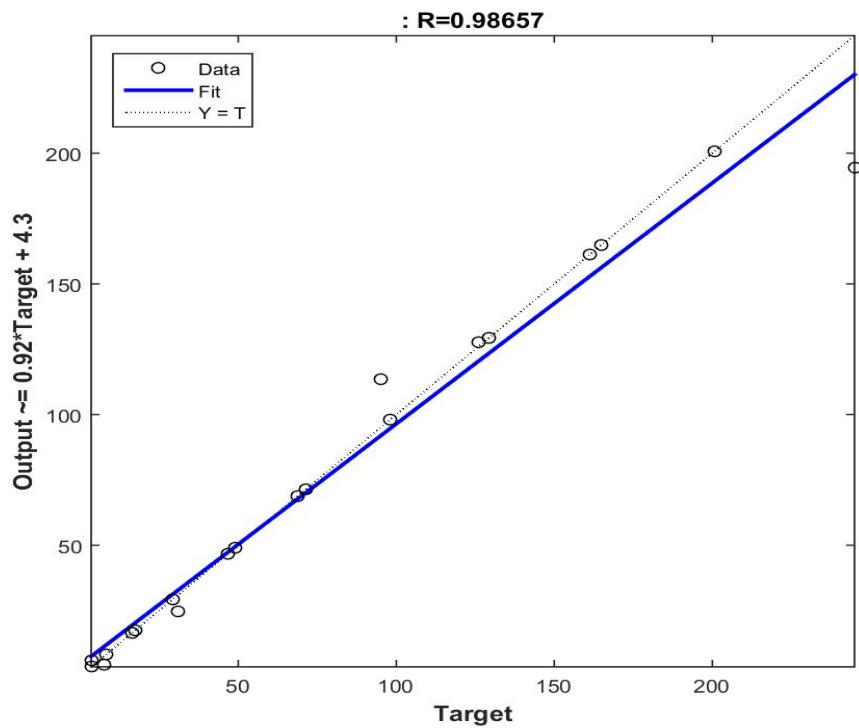
حالت دوم: در این تابع ساختار با ۳ نورون در لایه مخفی در نظر گرفته می‌شود. مشخص است که نتایج دقت حالت اول را ندارند و نمونه‌های آموزش با خطای از رنج 10^{-3} روی نمودار به دست آمده از شبکه عصبی قرار می‌گیرند؛ بنابراین کاهش تعداد نورون‌ها به ۳ باعث کاهش دقت شبکه عصبی می‌شود.



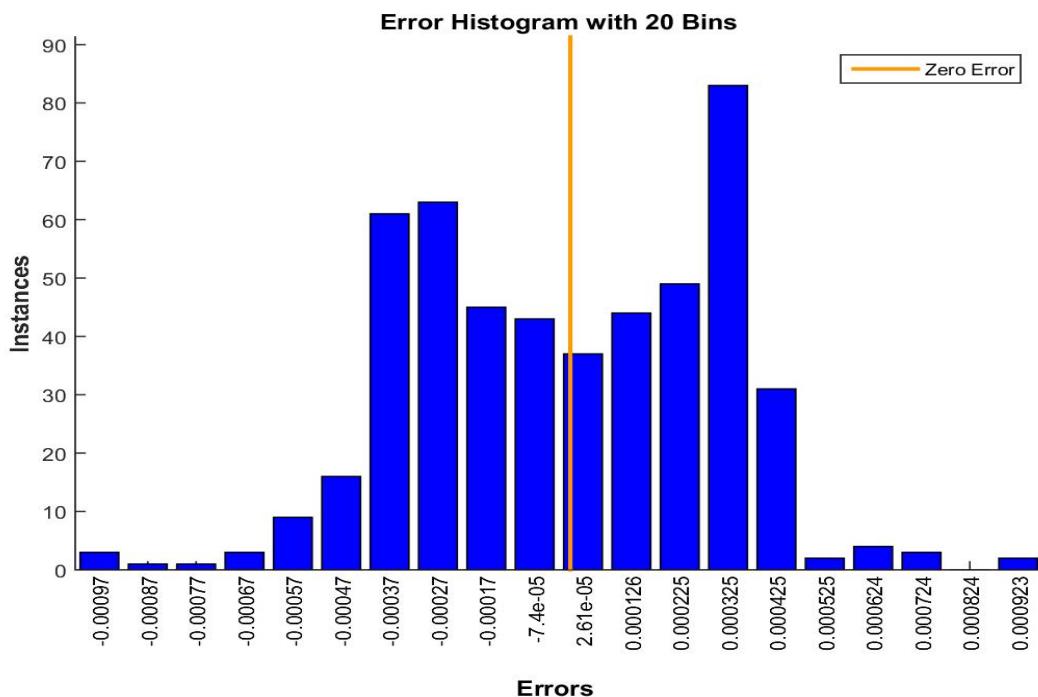
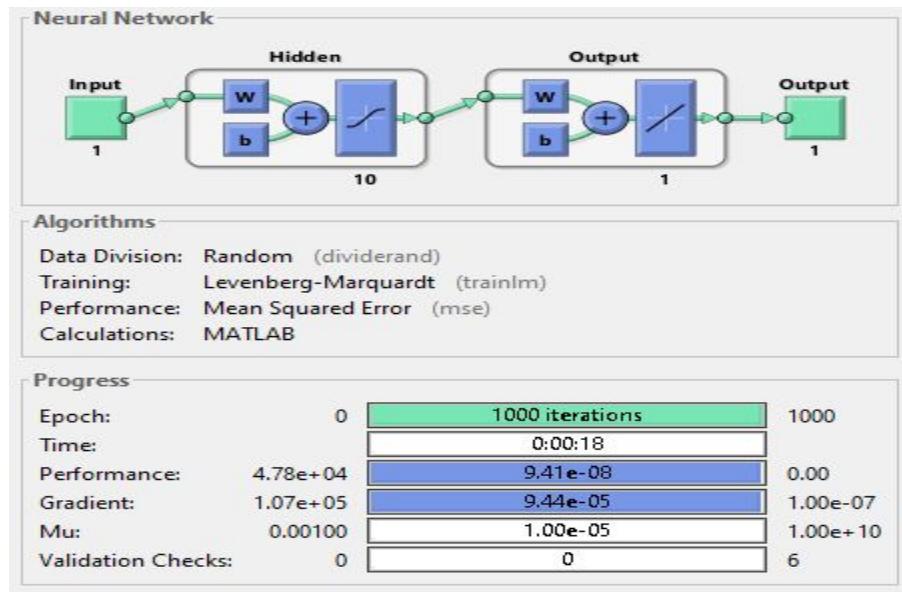


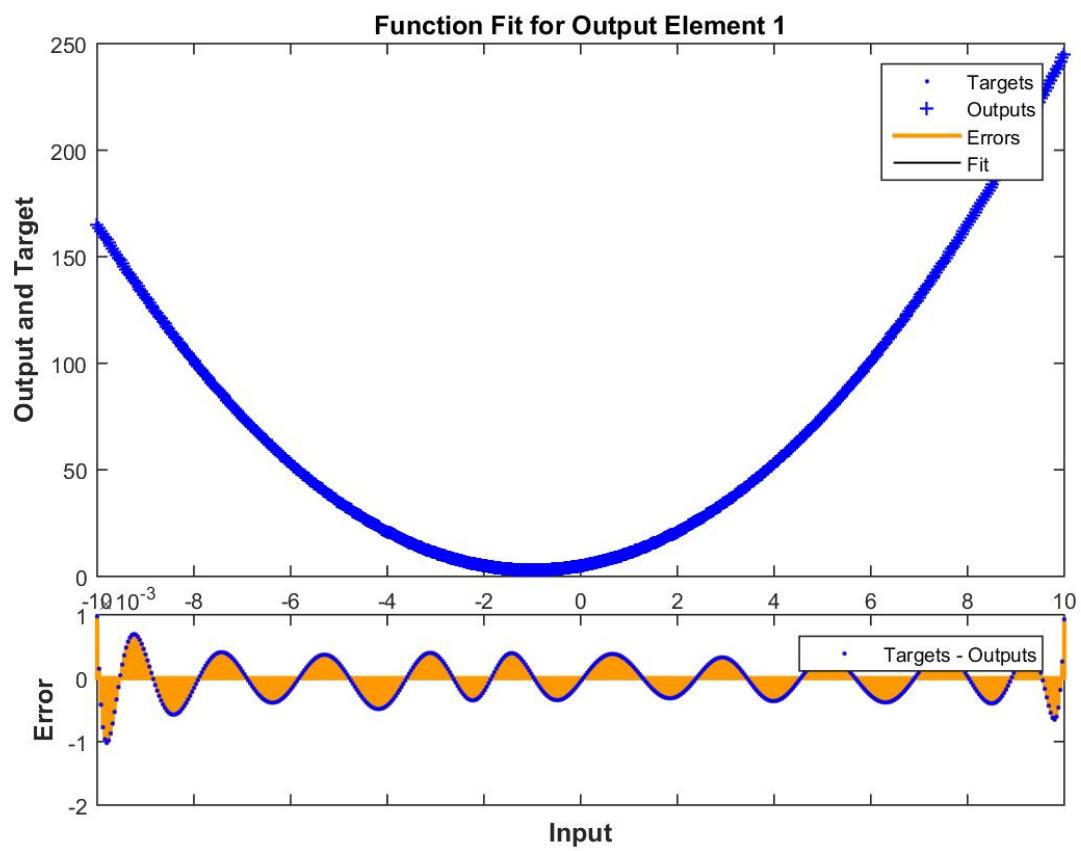
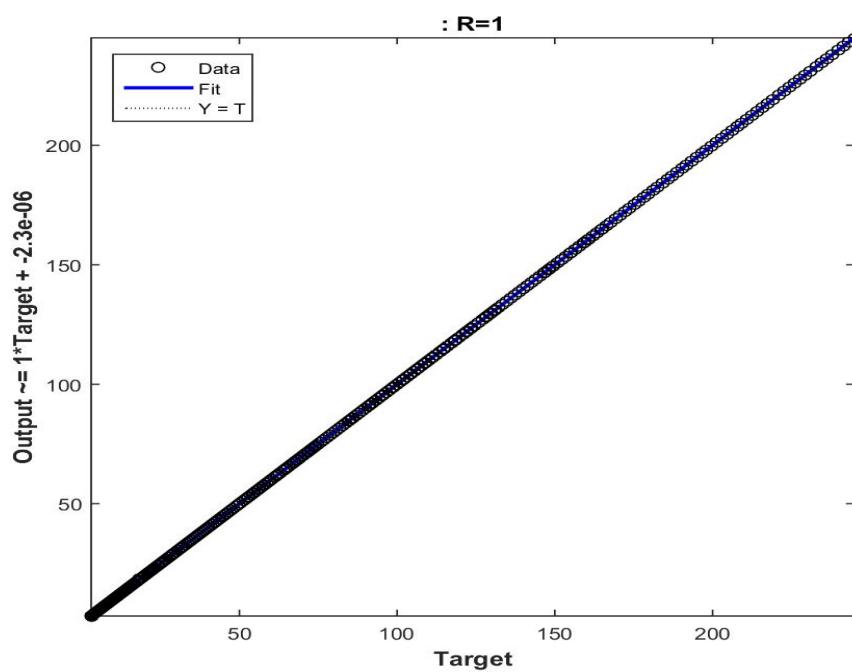
حالت سوم: در این حالت نمونه‌های آموزش را به ۲۰ کاهش می‌دهیم. همان‌طور که مشخص است دقت خروجی شبکه عصبی نسبت به حالت دوم که ۱۰۰ نمونه آموزش داشتیم کمتر است.





حالت چهارم: در این حالت نمونه‌های آموزش را به ۵۰۰ نمونه افزایش می‌دهیم. همان‌طور که مشخص است دقت خروجی شبکه عصبی نسبت به حالت اول با ساختار مشابه و تعداد ۱۰۰ نمونه آموزش داشتیم کمتر است؛ بنابراین می‌توان به این نتیجه رسید که با افزایش نمونه همیشه دقت افزایش نمی‌پابد.



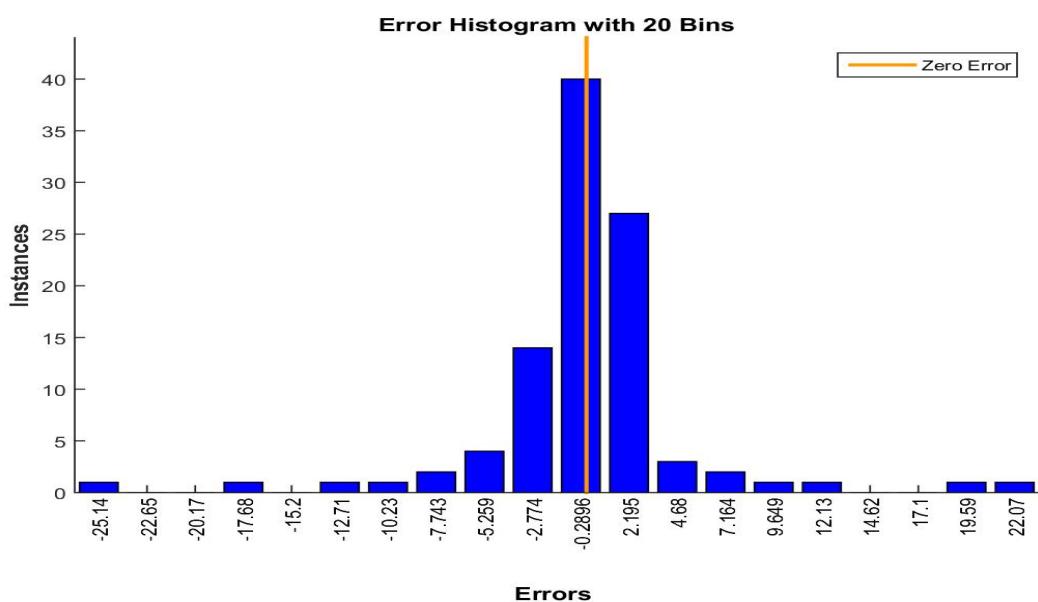
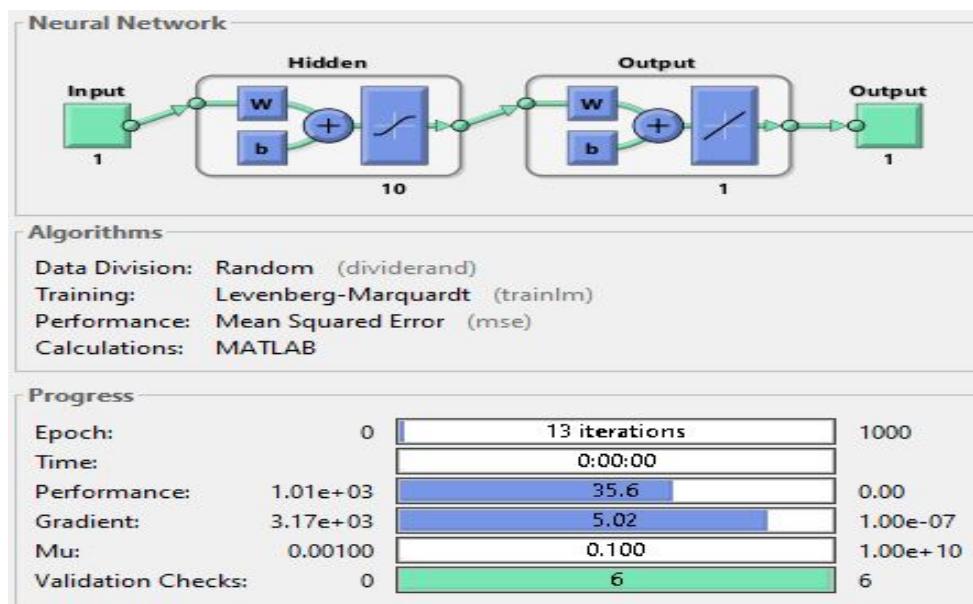


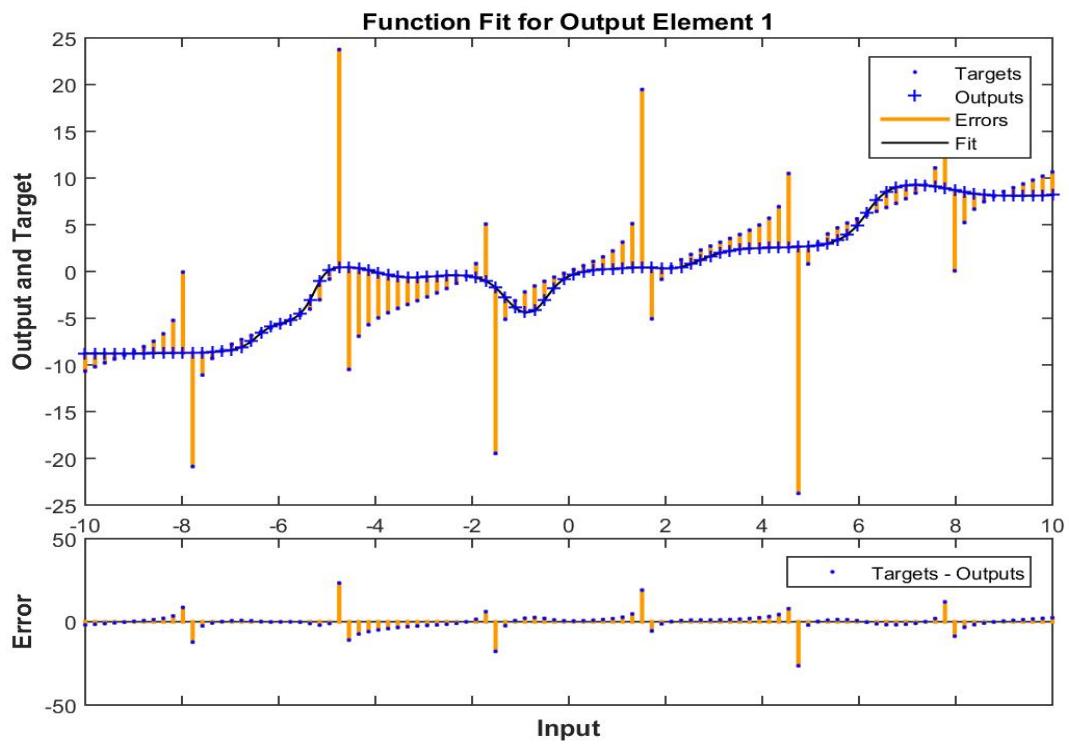
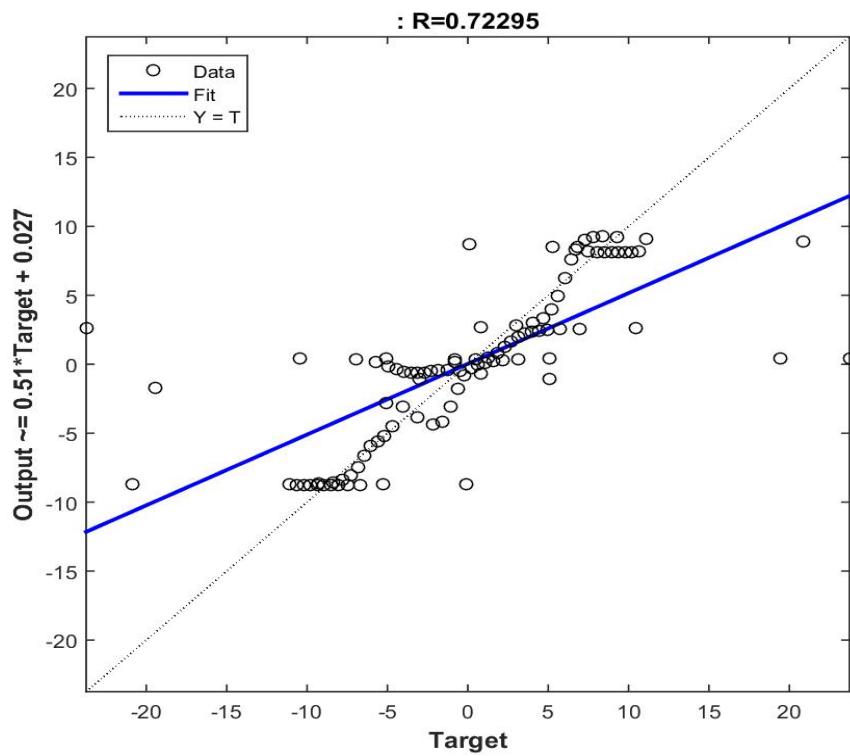
تابع چهارم:

$$y_4 = \tan(x) + x;$$

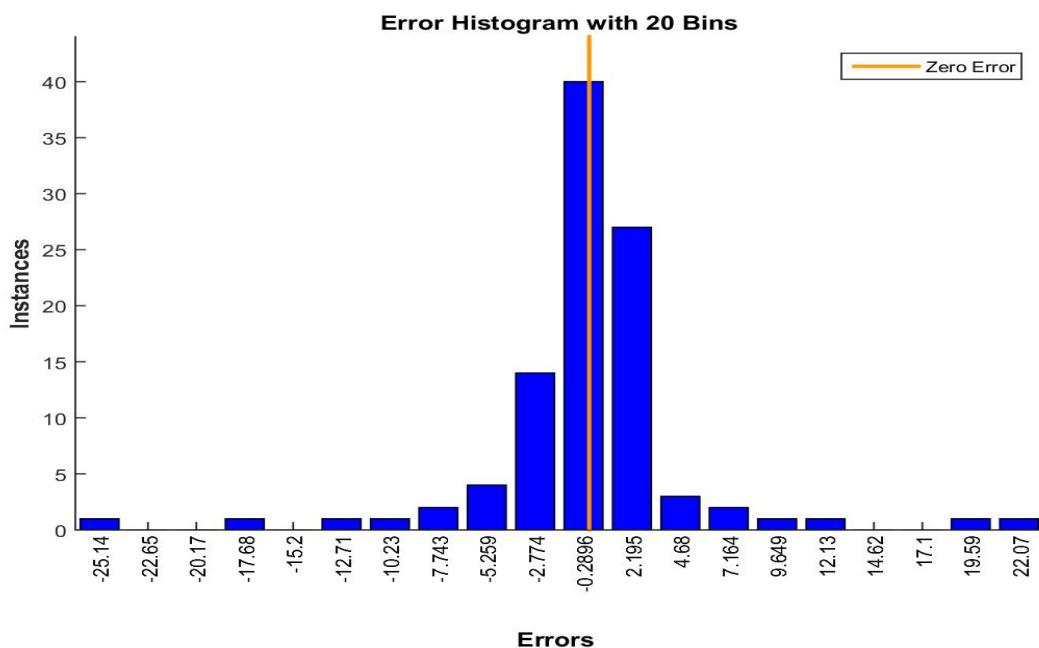
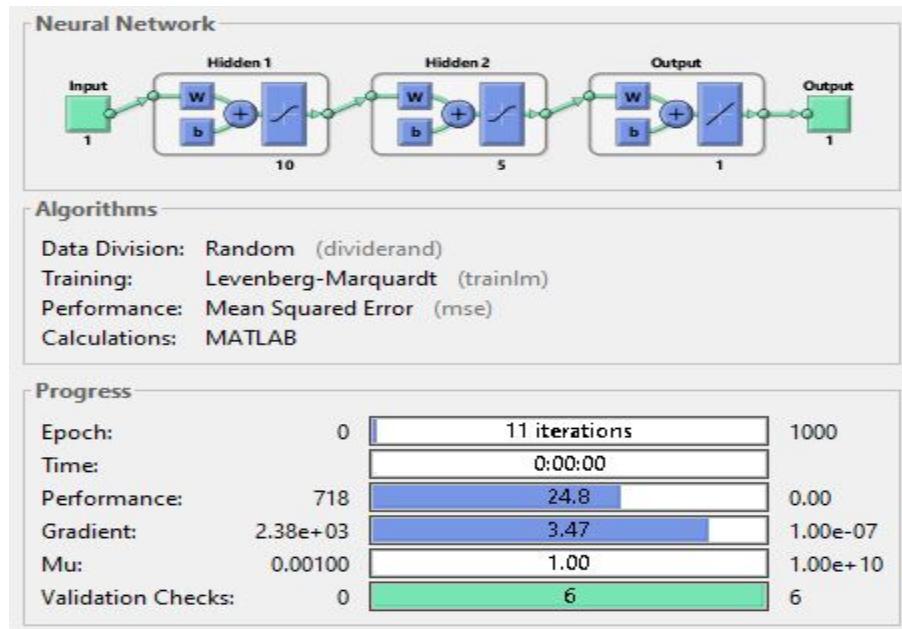
حالت اول: این تابع از آن جهت که در نقاط مضارب فرد $\frac{\pi}{2}$ در بازه مشخص شده به بینهایت میل می‌کند تابعی نیست که بتوان از

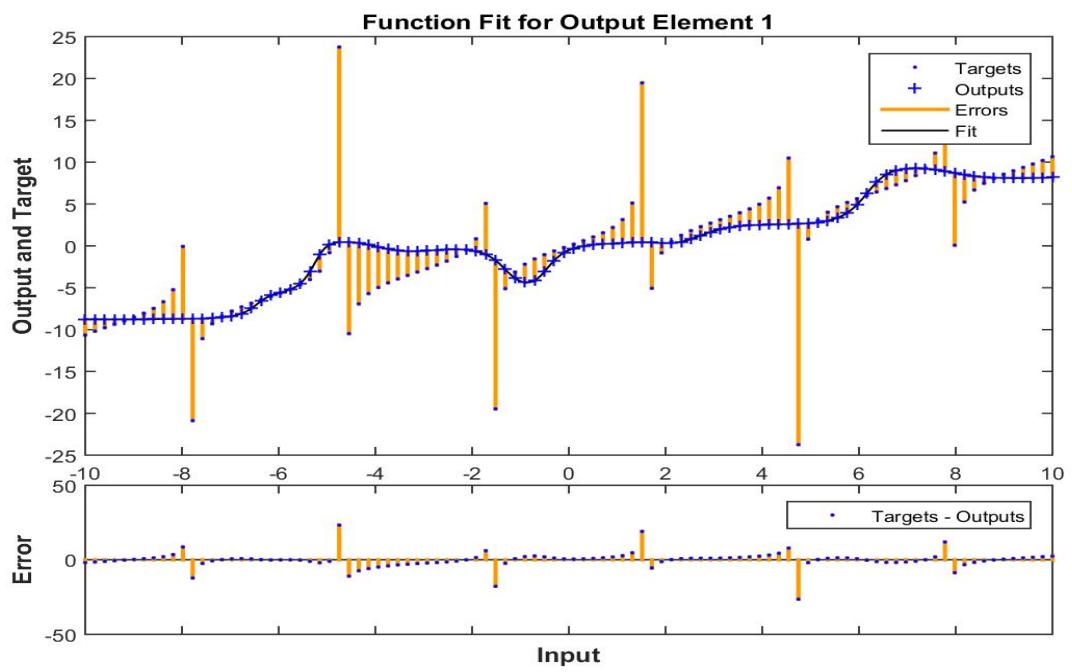
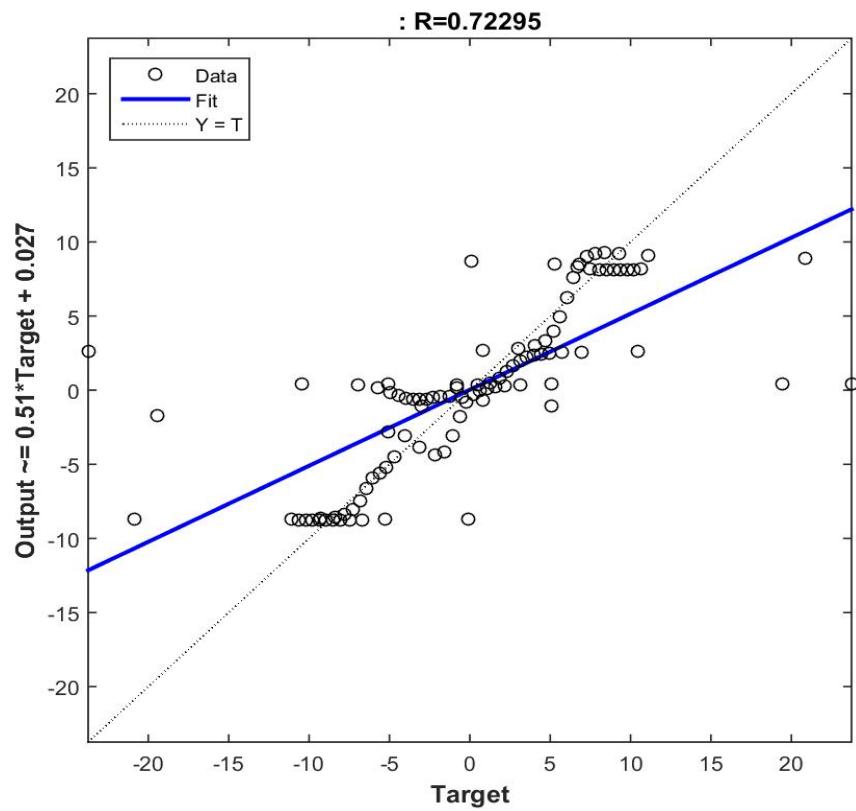
آموش آن در شبکه عصبی نتیجه مناسبی به دست آورد. در حالت اول ساختار ۱۰ نورون در لایه مخفی انتخاب می‌شود. همان‌طور که مشخص است نتیجه حاصله از شبکه عصبی خطای زیادی دارد.



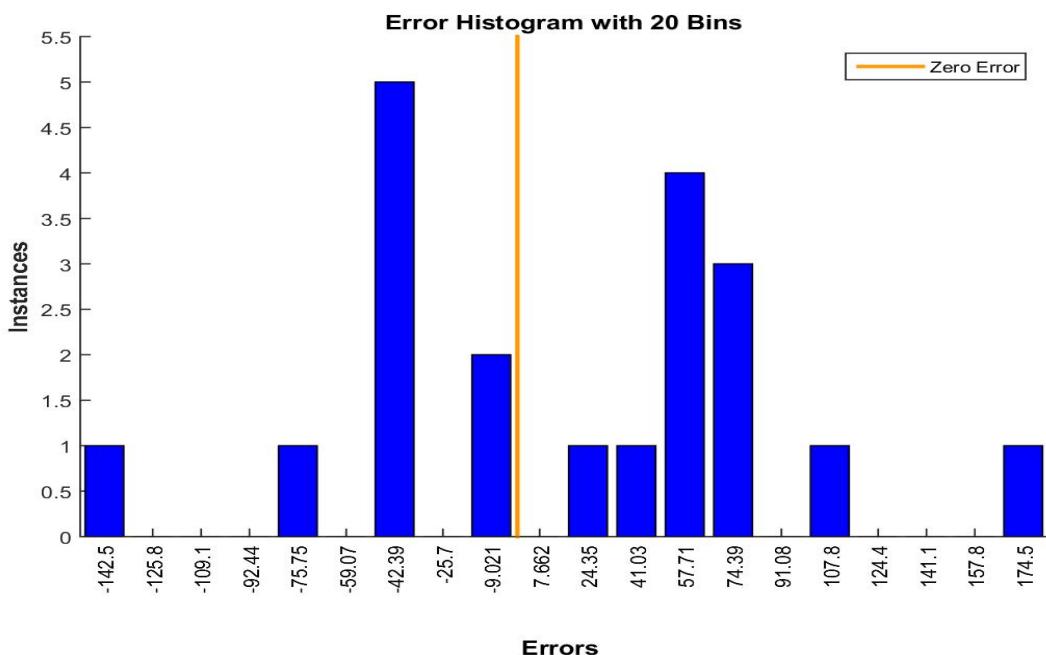
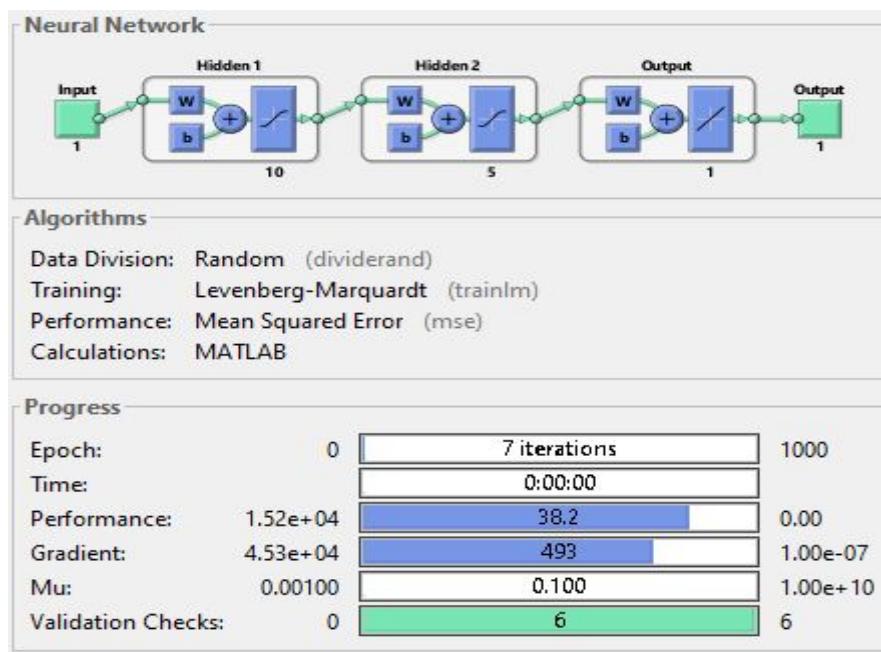


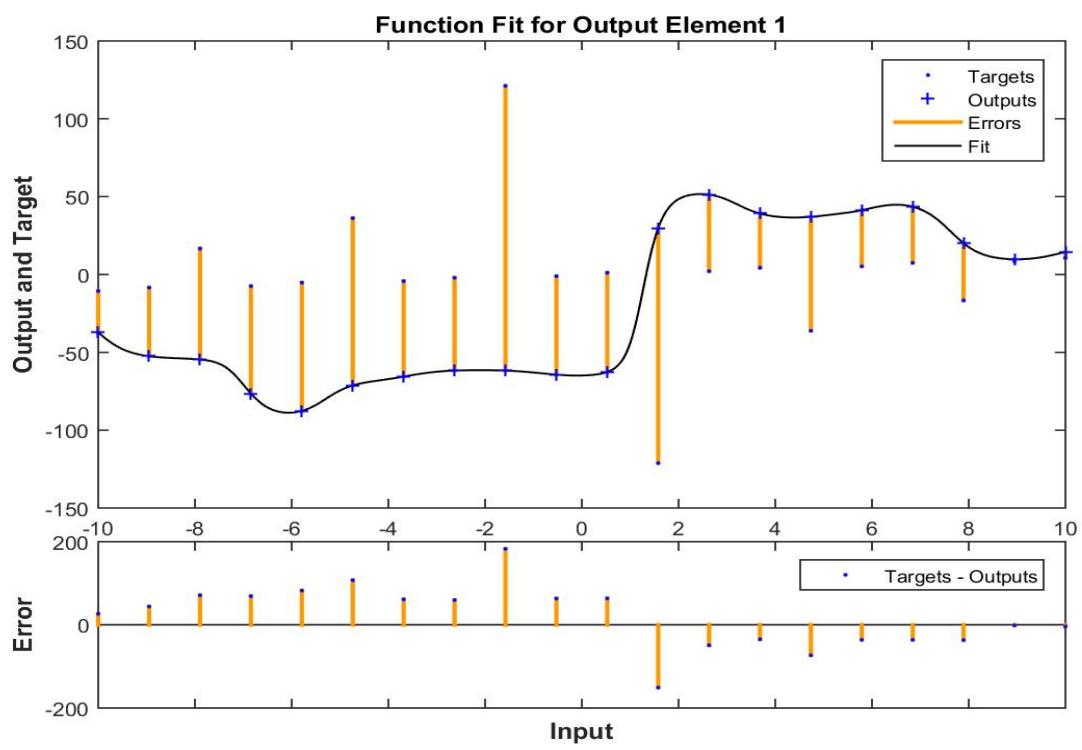
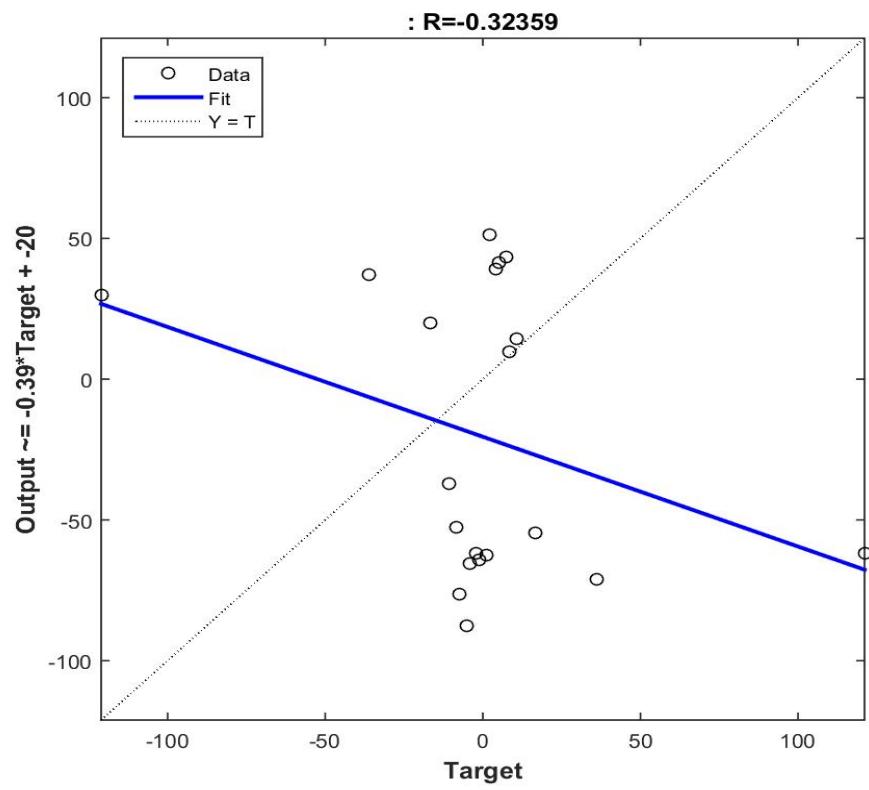
حالت دوم: در این حالت ساختار شبکه عصبی را با دو لایه پنهان ۱۰ و ۵ نورونی در نظر می‌گیریم. باز هم نتیجه مناسبی از آن به دست نمی‌آید و خطای شبکه عصبی نسبت به مقادیر آموزش زیاد است.



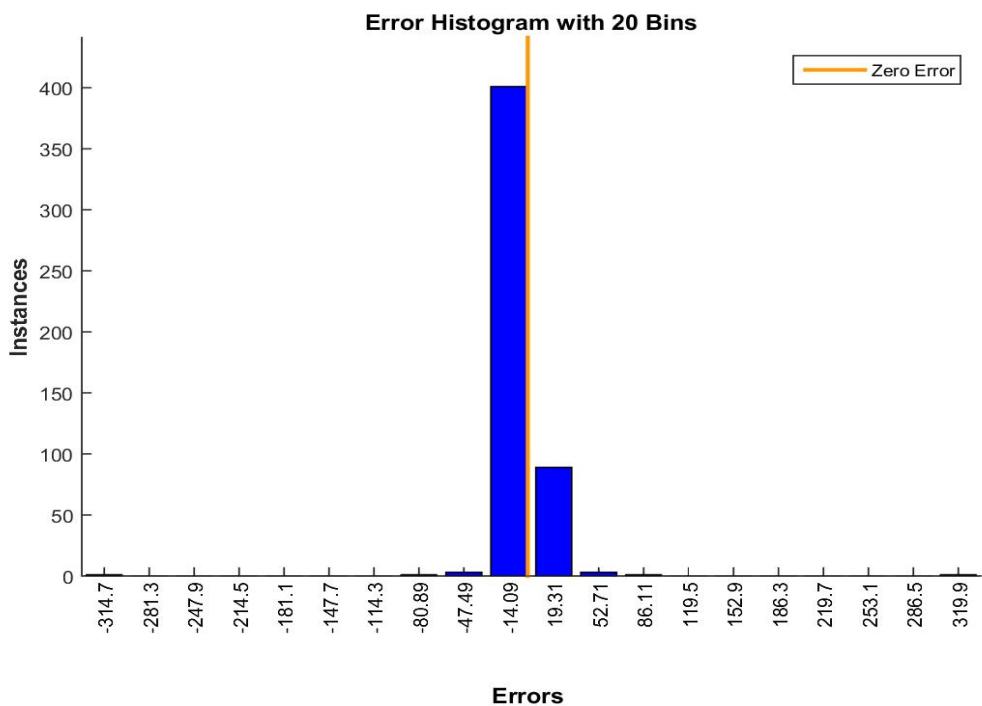
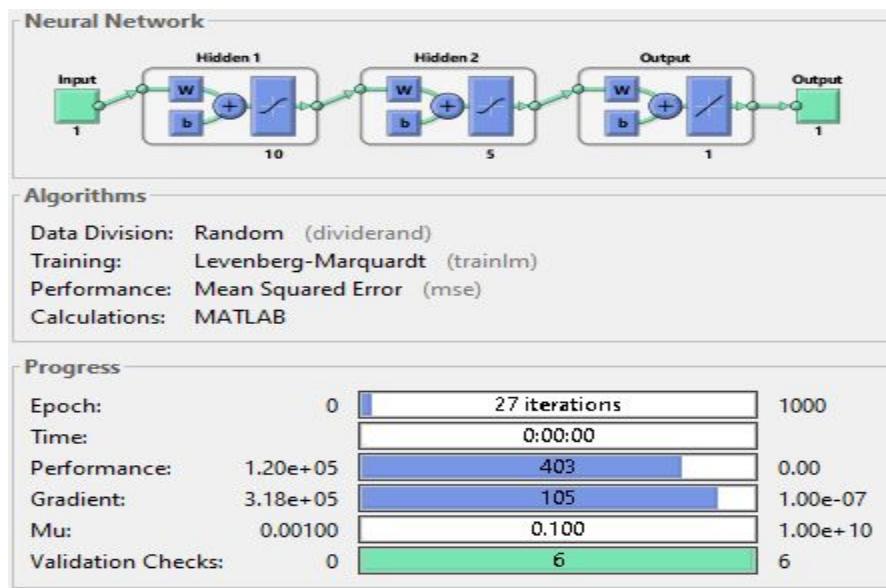


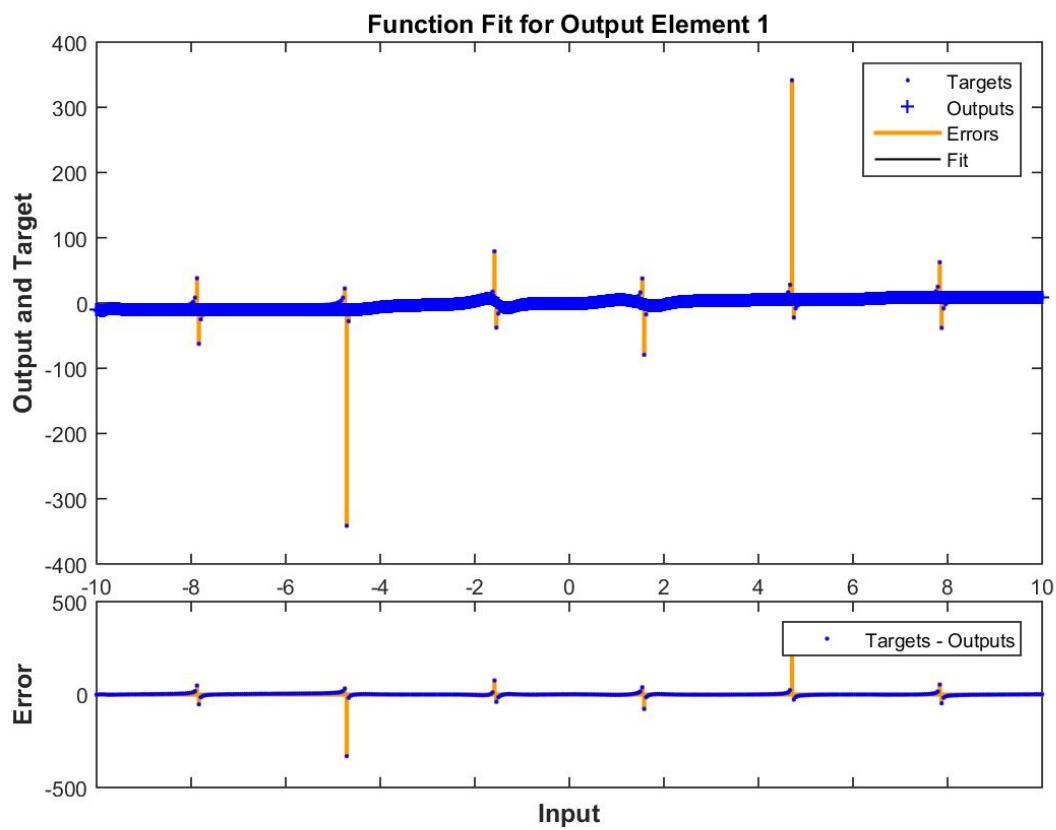
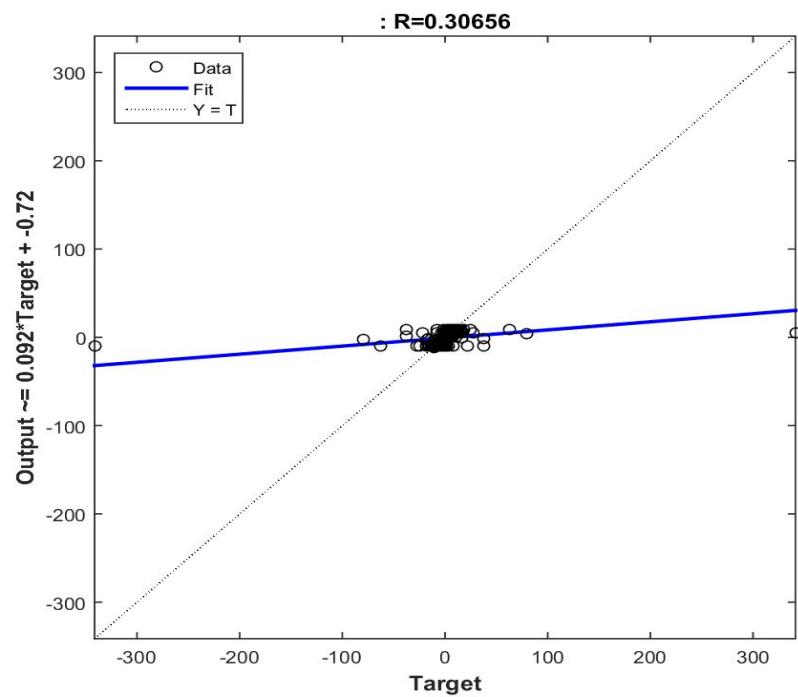
حالت سوم: با ساختار حالت دوم، تعداد ورودی‌ها را به ۲۰ کاهش می‌دهیم. نتیجه نسبت به حالت قبل بدتر خواهد شد؛ و دقت شبکه عصبی آموزش دیده شده از حالت قبلی کمتر می‌شود.





حالت چهارم: ساختار شبکه عصبی در این حالت نیز مشابه دو حالت قبل است اما با تعداد ۵۰۰ نمونه آموزش. نکته‌ای که مشخص است این است که خطای بعضی نقاط تا بیشتر از 350 نیز می‌رود. دلیل این اتفاق این است که با افزایش تعداد در همان بازه قبلی، اعدادی در مجموعه X ایجاد شود که به مضارب فرد $\frac{\pi}{2}$ نزدیک‌ترند و حاصل تابع \tan را زیاد می‌نمایند و این مقادیر به صورت ضربه‌ای در تابع پدیدار می‌شوند که آموزش را برای شبکه عصبی سخت می‌نماید.

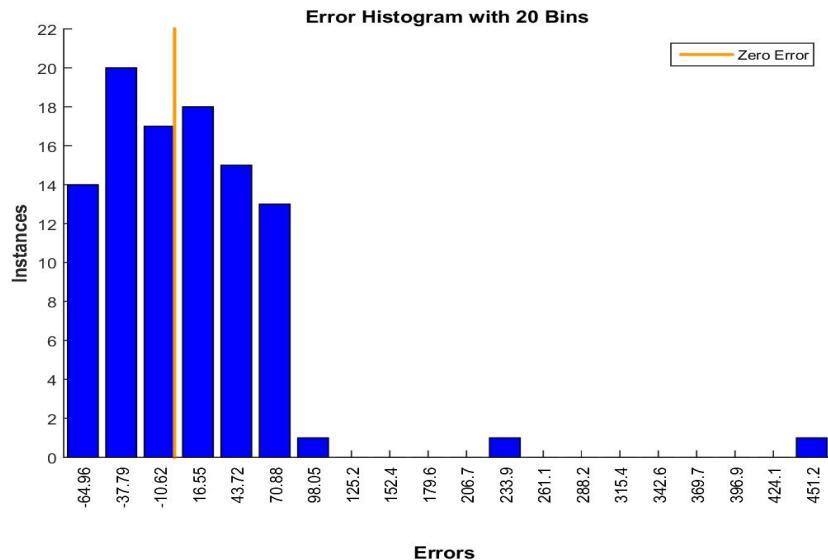
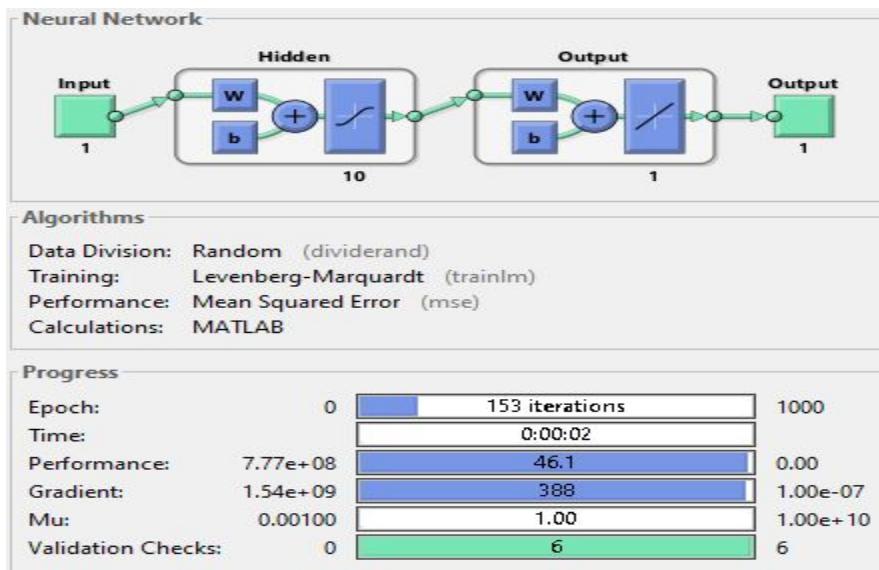


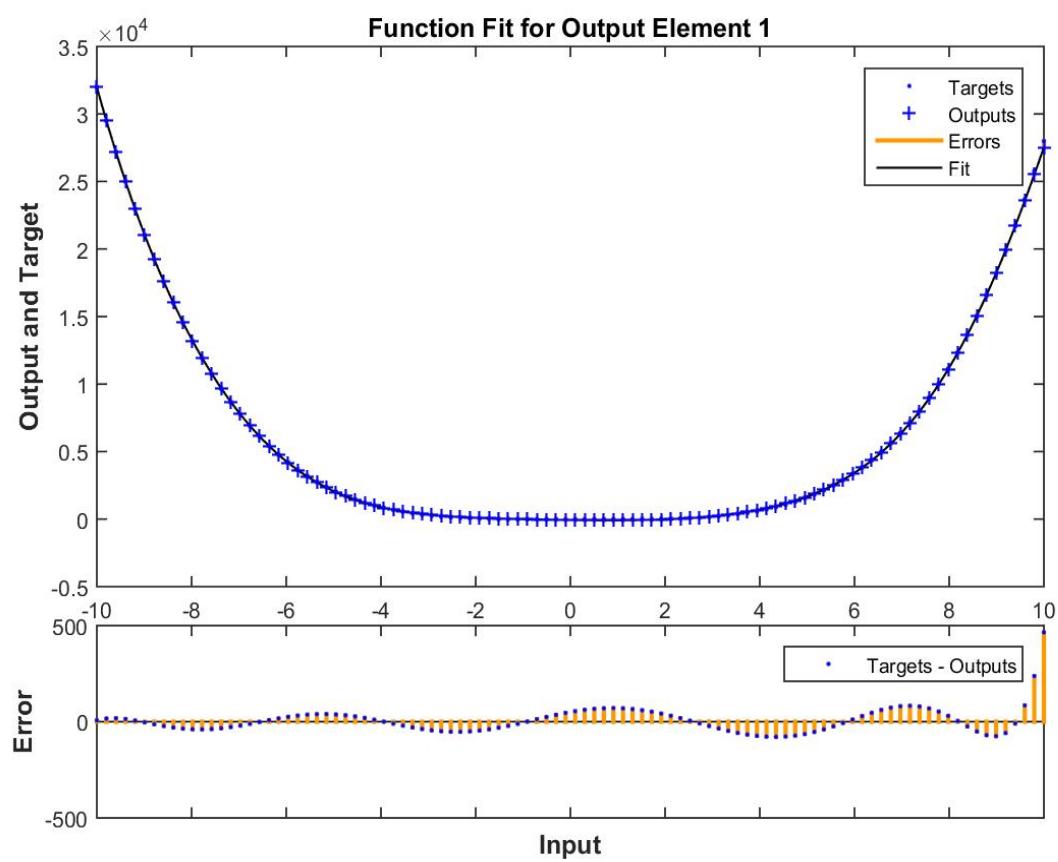
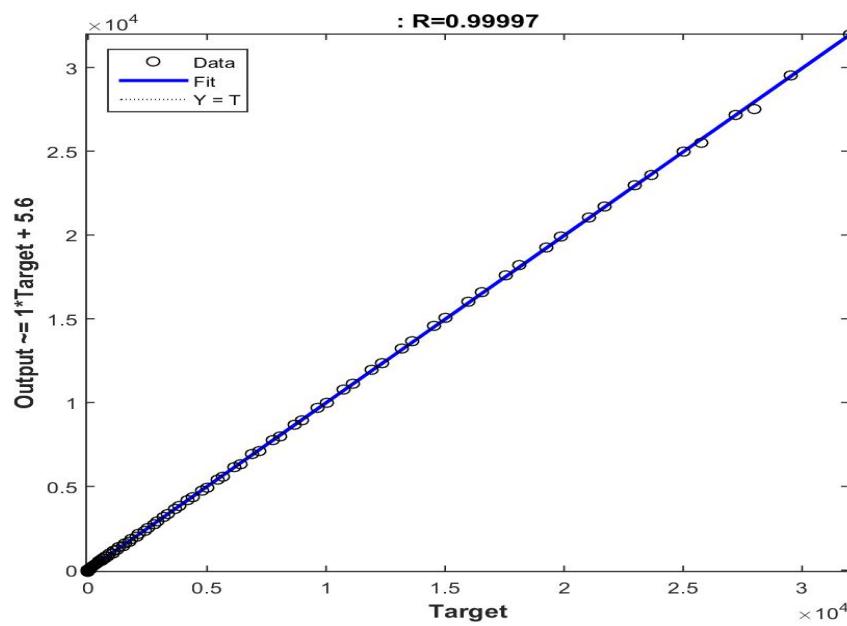


تابع پنجم:

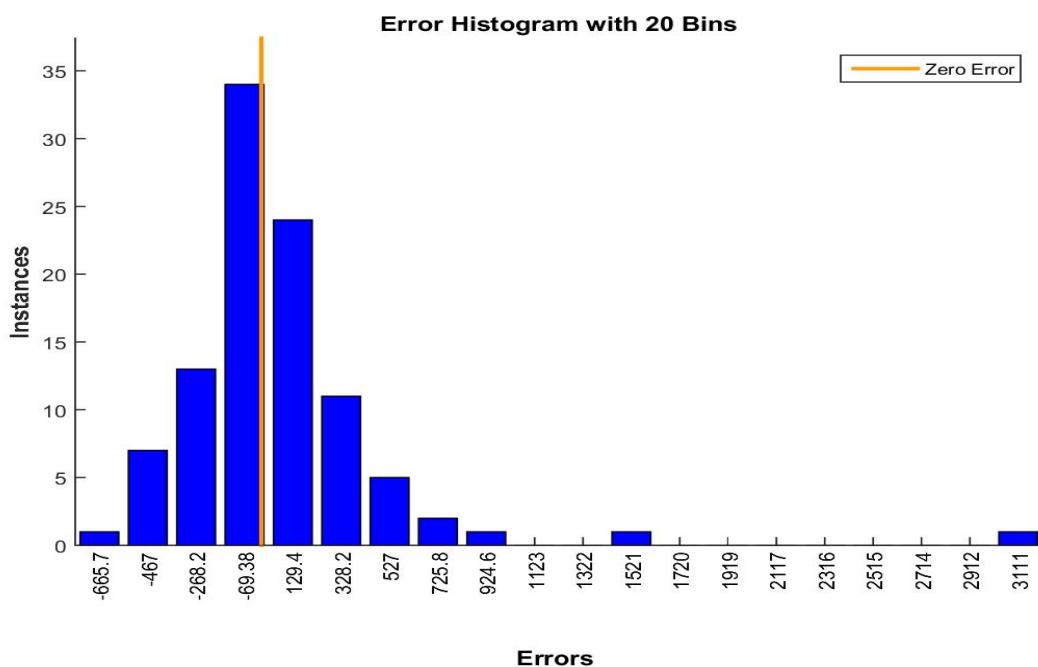
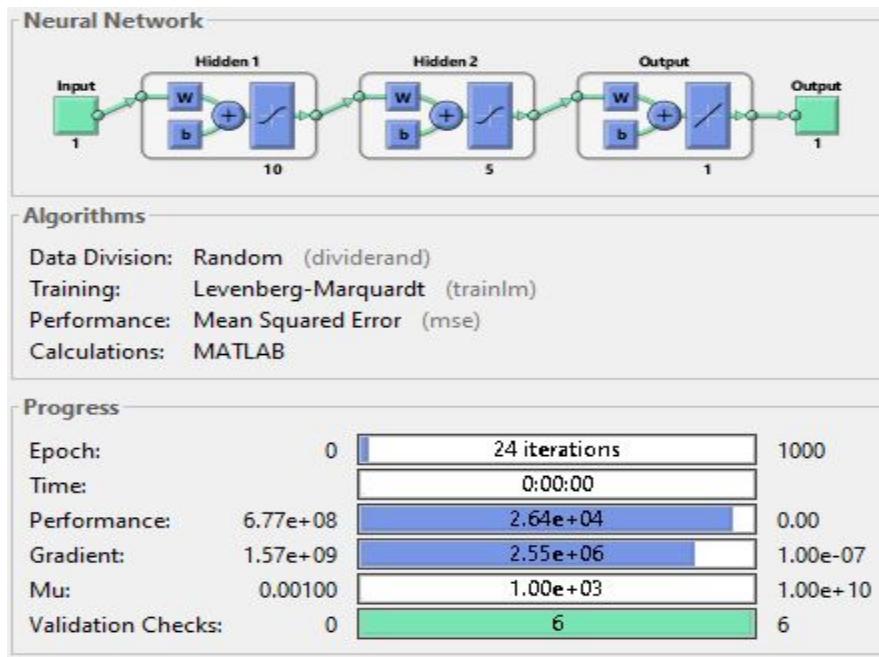
$$y_5 = 3x^4 - 2x^3 - 10;$$

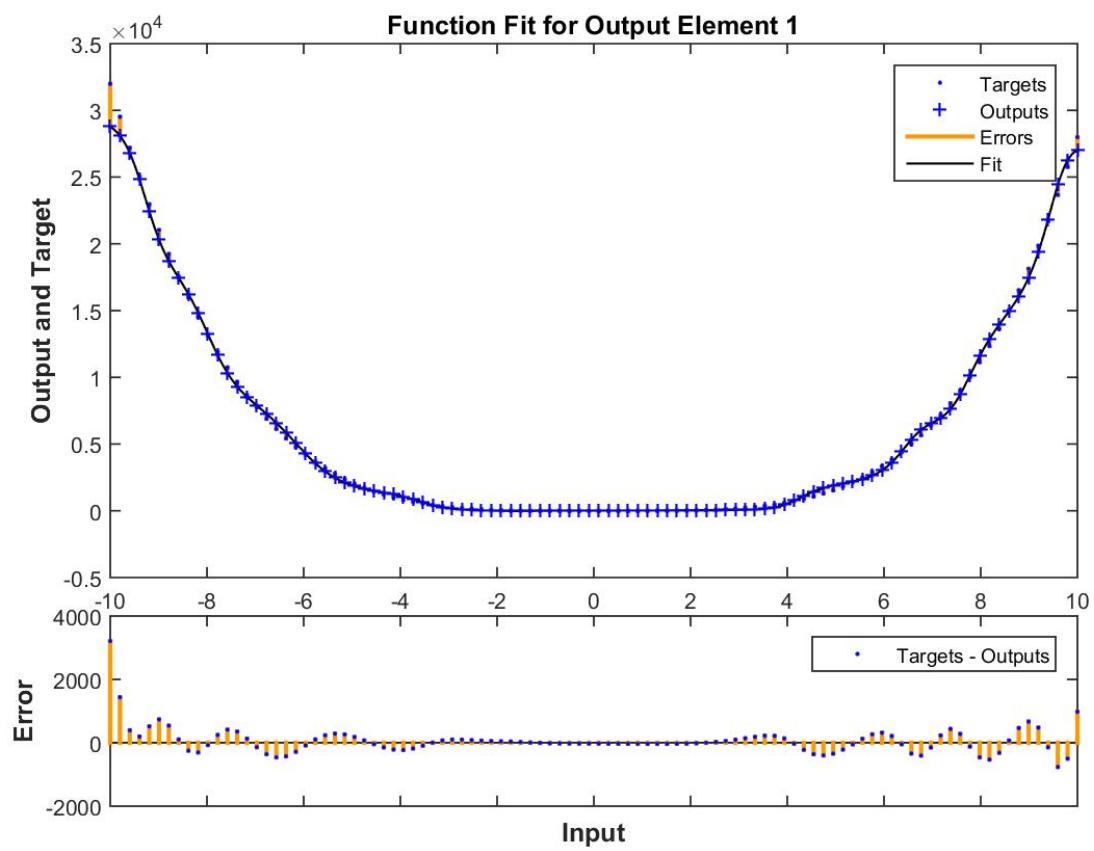
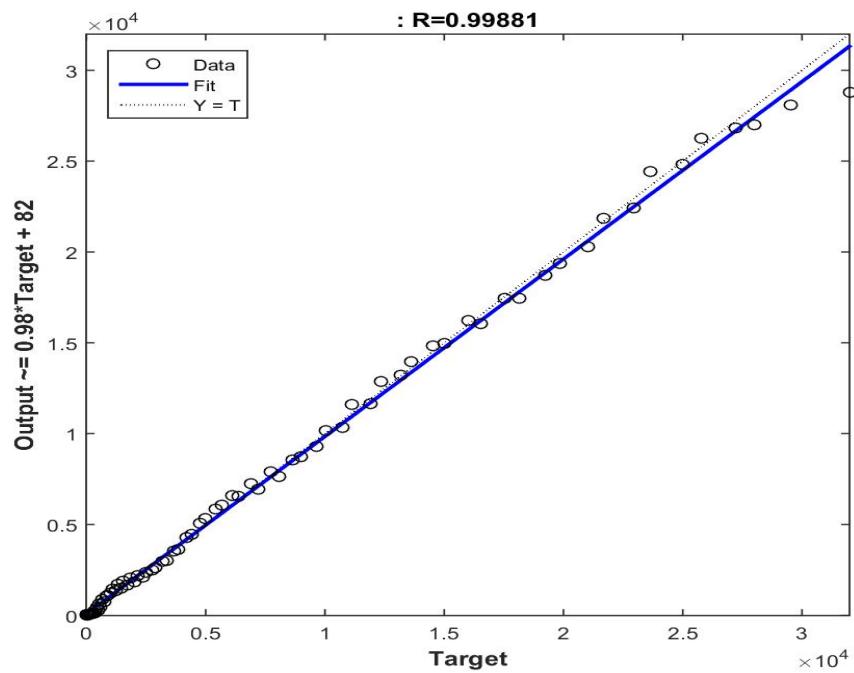
حالت اول: در این حالت شبکه با ساختار یک لایه پنهان با ۱۰ نورون در نظر رفته می‌شود. نتایج حاصله در شکل‌های زیر مشخص است. با توجه به شکل خطای شبکه عصبی، مشخص است که در مواردی خروجی تابع با خروجی شبکه عصبی مقدار ۴۵۰ است؛ اما با توجه به اینکه مقادیر خود تابع در رنج $[-10^4, 10^4]$ بنا براین می‌توان به طور نسبی آن را کم و خروجی شبکه عصبی را قابل قبول در نظر گرفت.



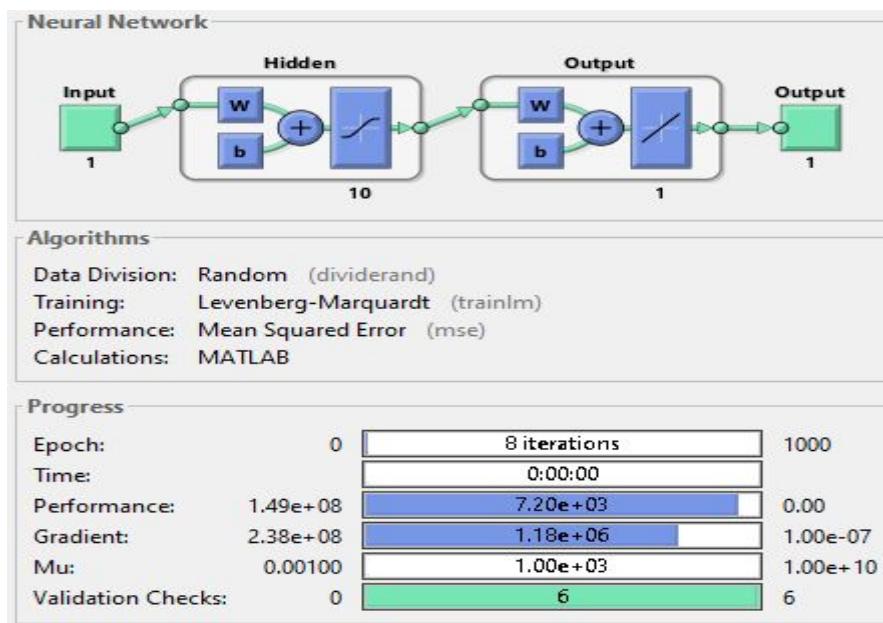


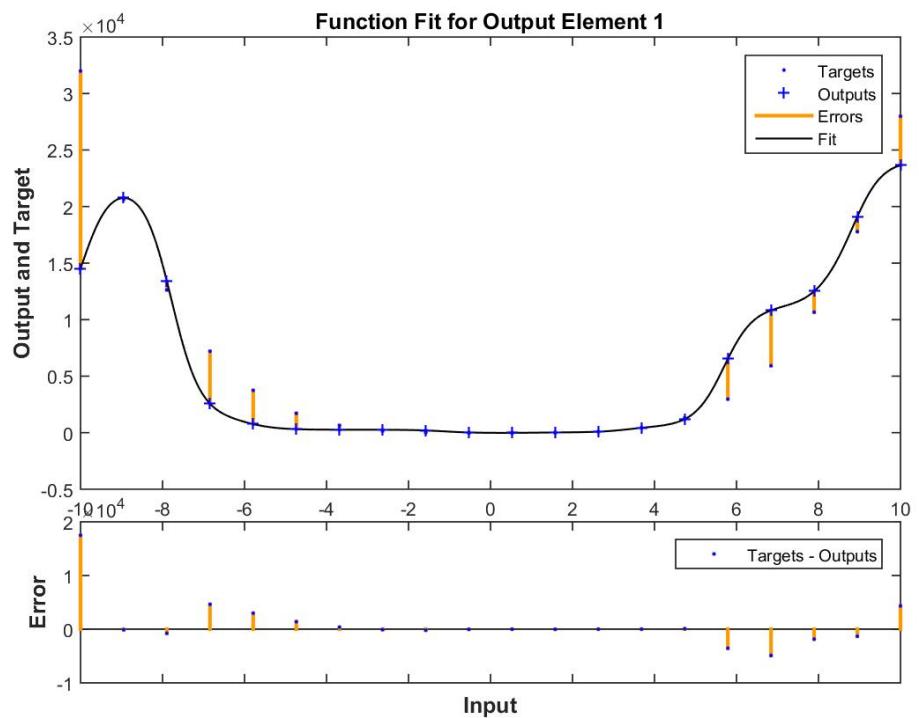
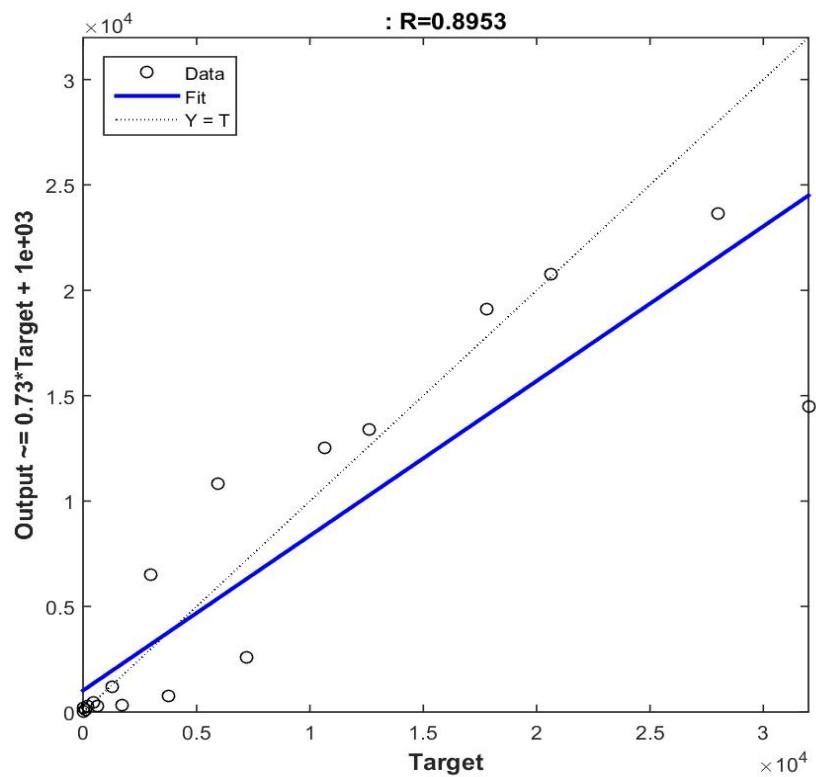
حالت دوم: در این حالت ساختار شبکه عصبی را با دولایه پنهان ۱۰ و ۵ نورونی در نظر می‌گیریم. نتایج به دست آمده در این حالت دقیق‌تری دارند.



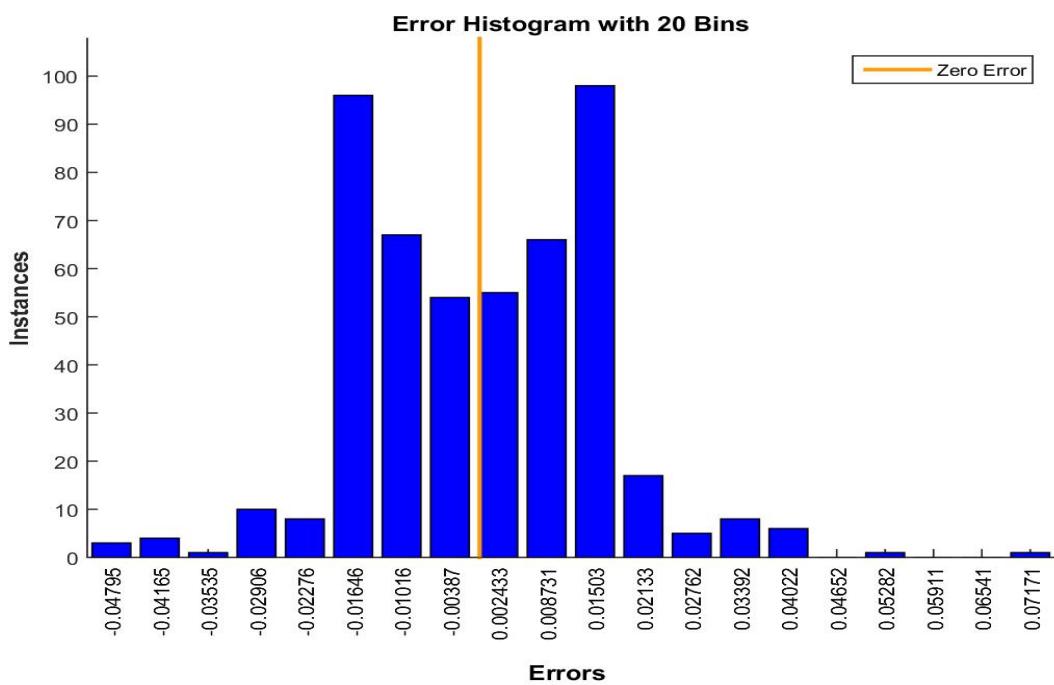
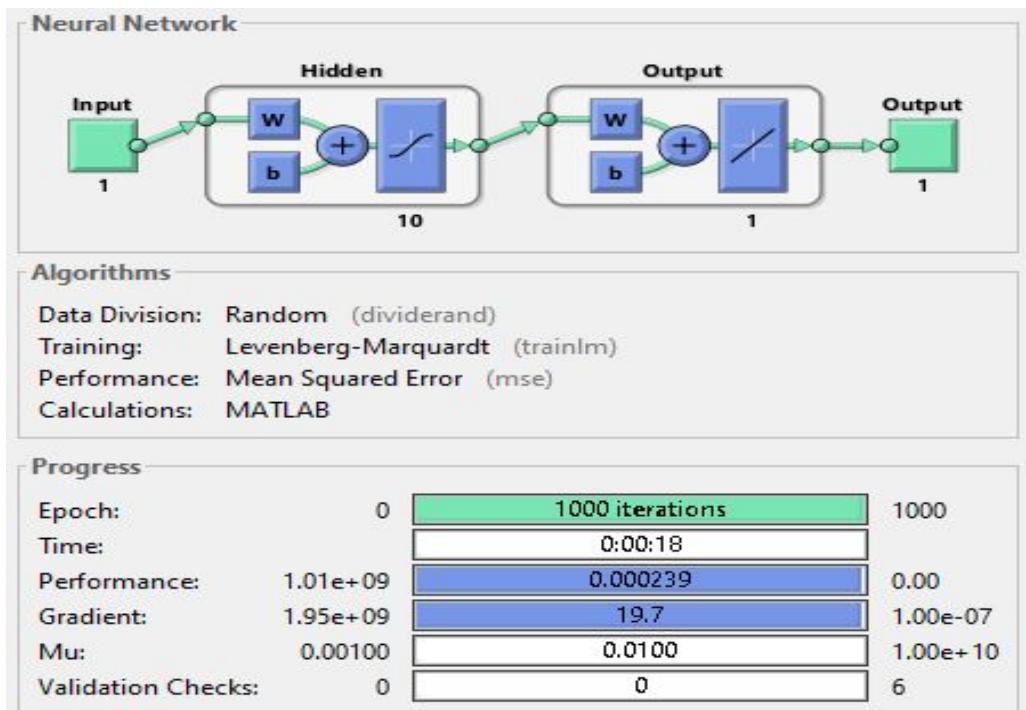


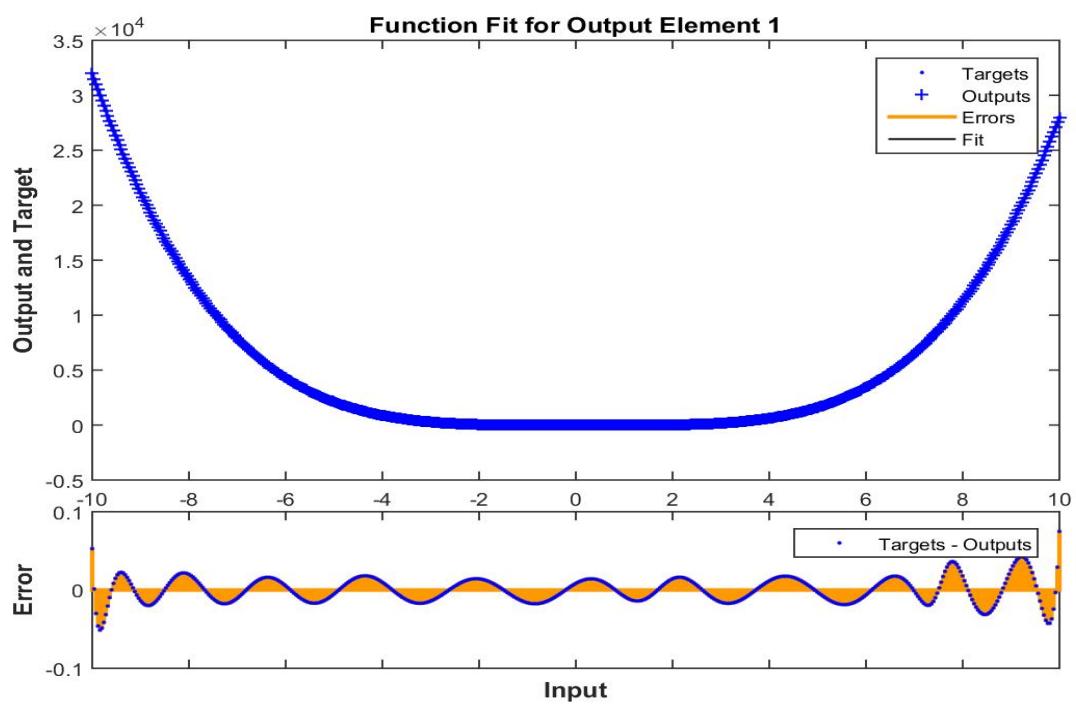
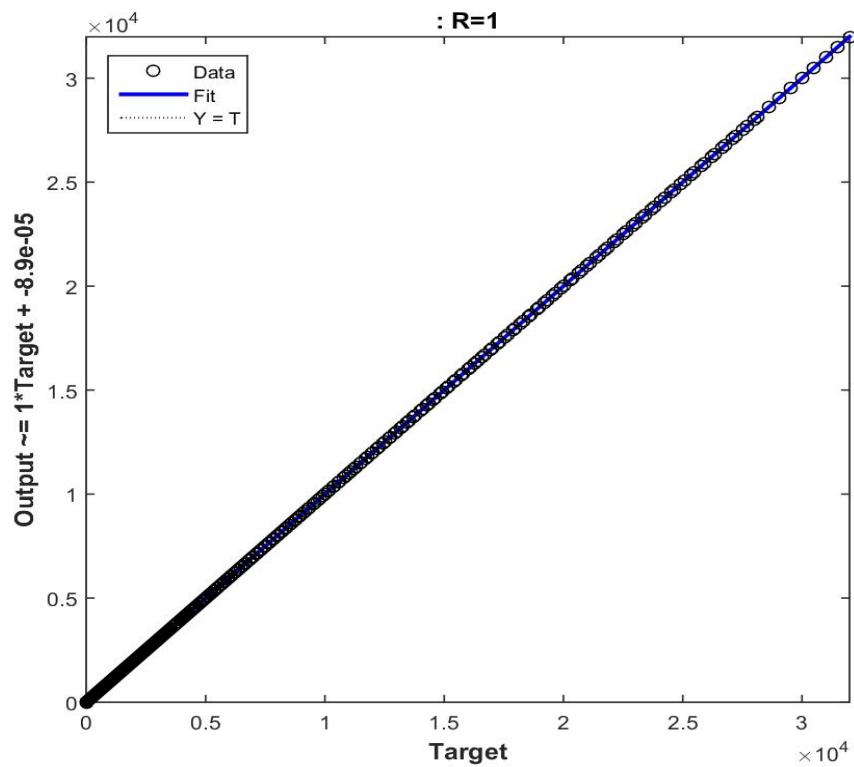
حالت سوم: در این حالت ساختار یک لایه پنهان و ۱۰ نورون در نظر گرفته شده است؛ اما تعداد نمونه های آموزش به ۲۰ کاهش پیدا کرده است. همان طور که مشاهده می شود، این حالت هم از نظر همبستگی و هم خطأ و هم نمودار خروجی شبکه عصبی شرایط نسبت به حالت اول با ۱۰۰ نمونه آزمایش شرایط بدتری دارد.





حالت چهارم: در این حالت ساختار یک لایه پنهان و ۱۰ نورون در نظر گرفته شده است؛ اما تعداد نمونه های آموزش به ۵۰۰ افزایش پیدا کرده است. مشاهده می شود که نتایج نسبت به حالت اول خیلی بهتر است و از تمامی جهات نتایج شبکه عصبی بهتر شده است.

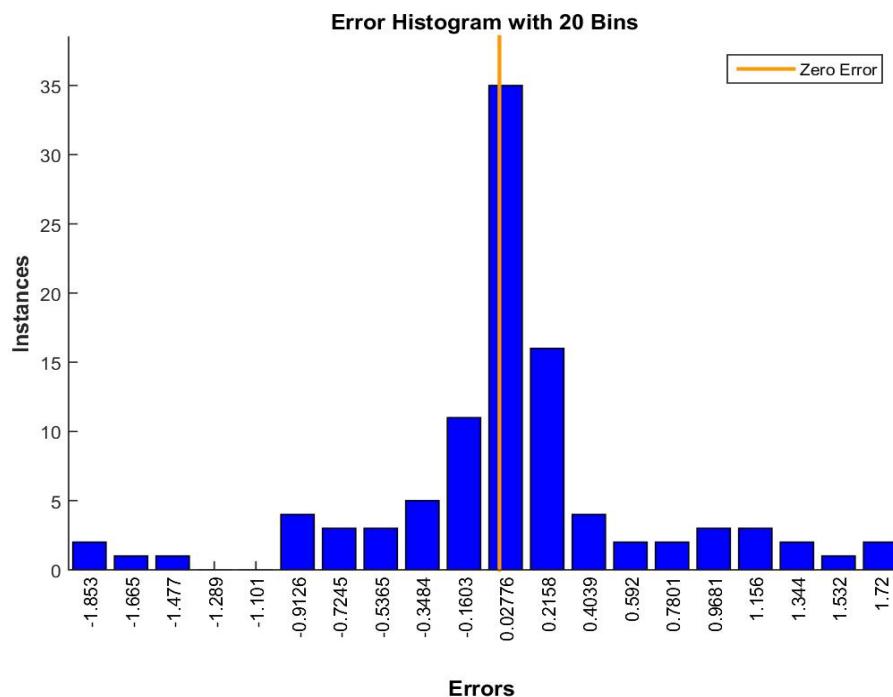
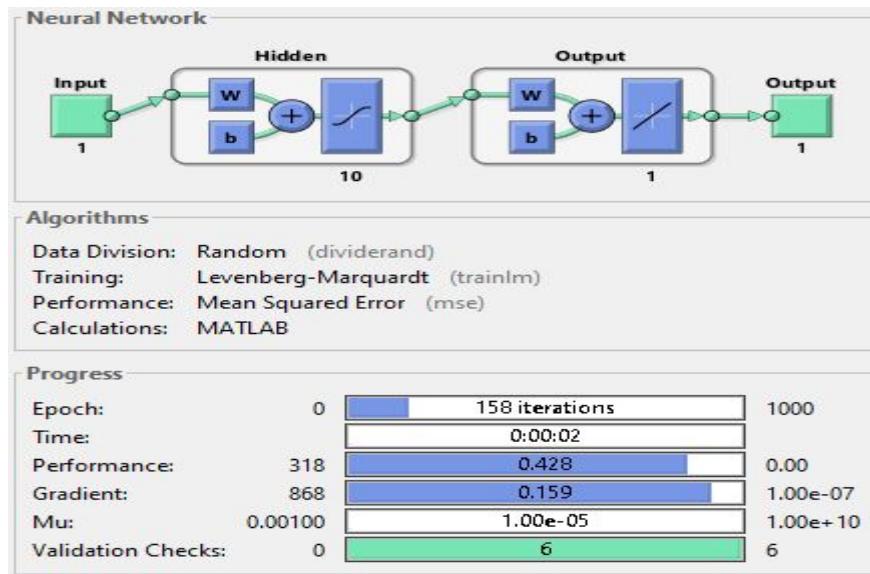


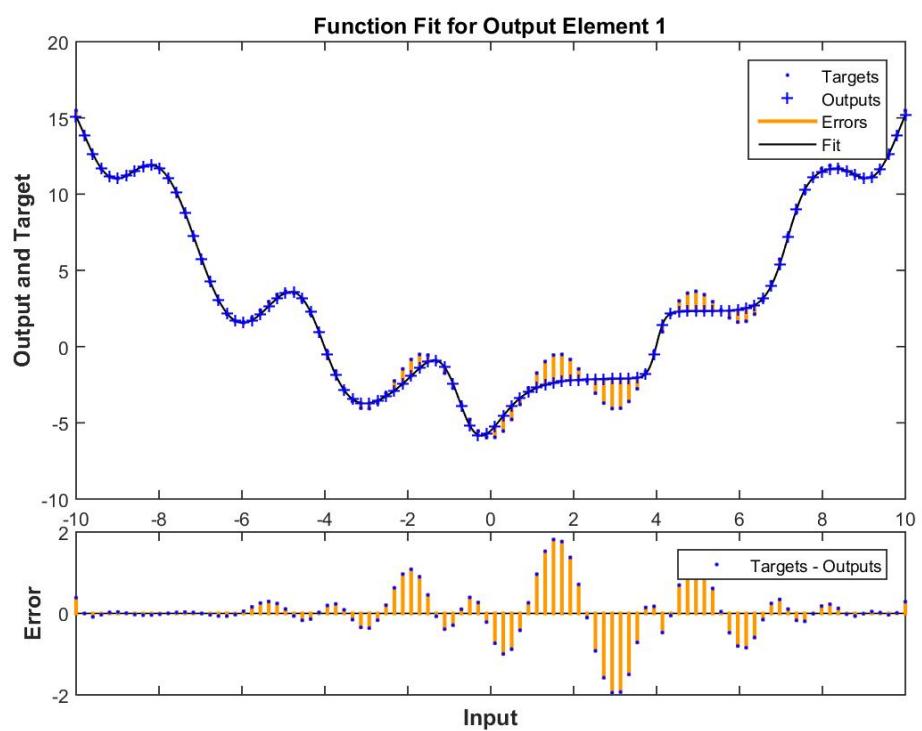
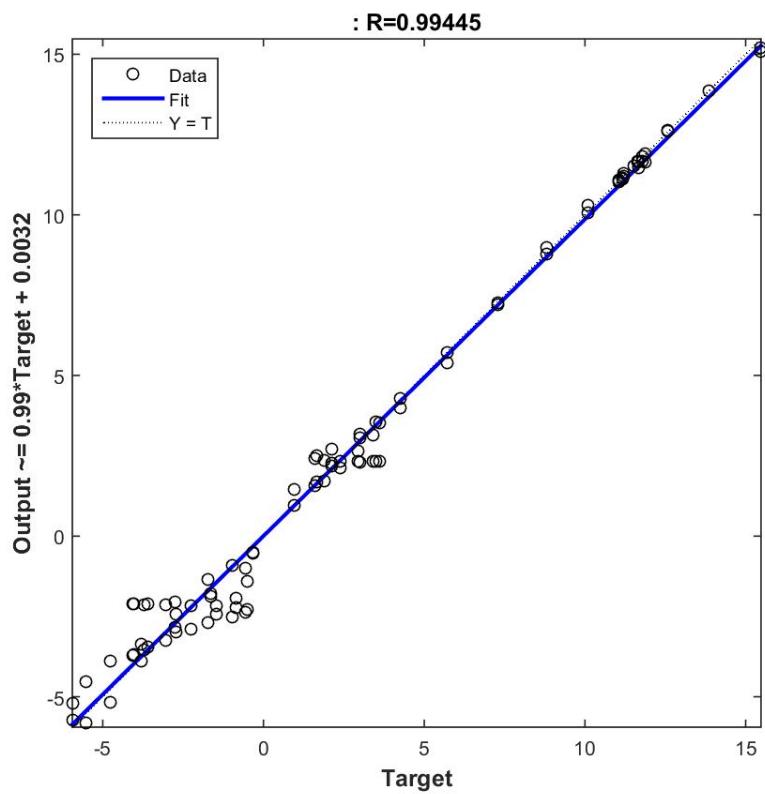


تابع ششم:

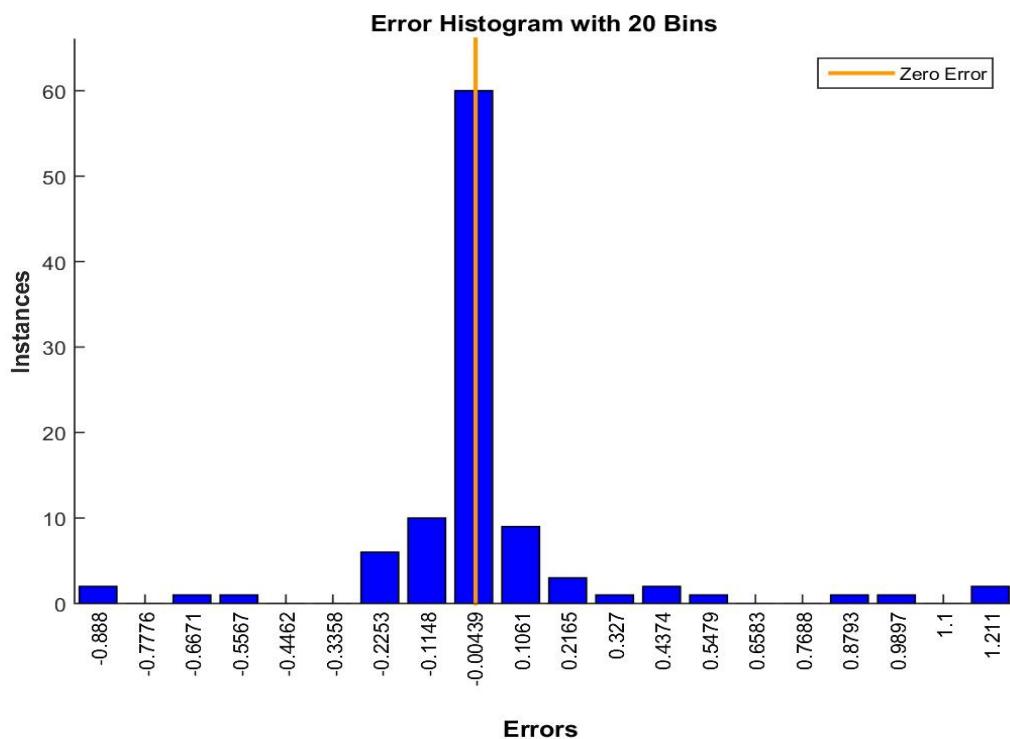
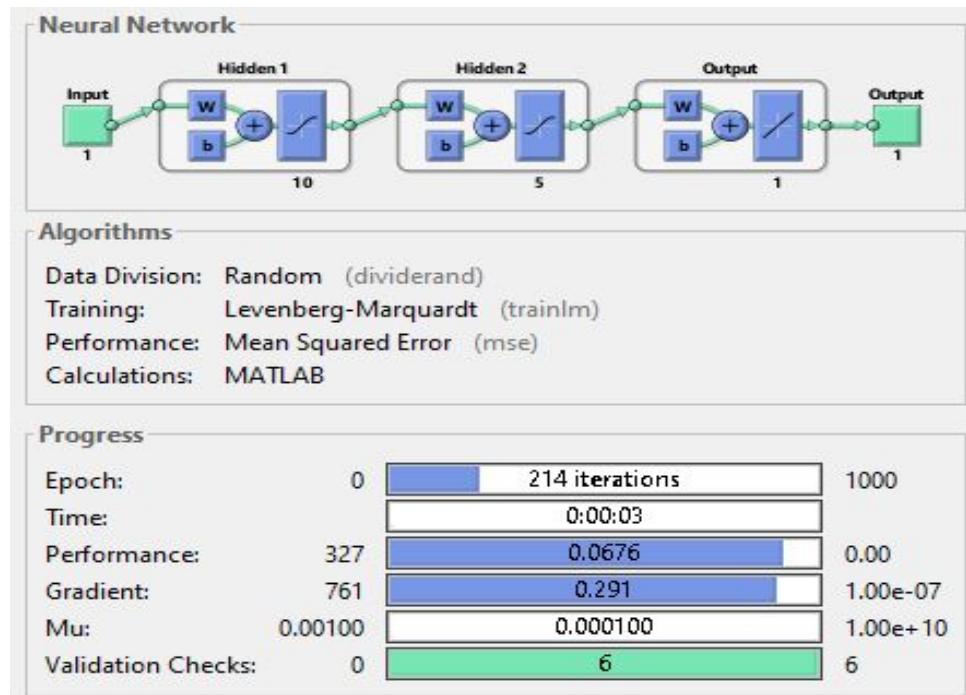
$$y_6 = 5 * \sin(x)^2 + 0.2 * x^2 - 6;$$

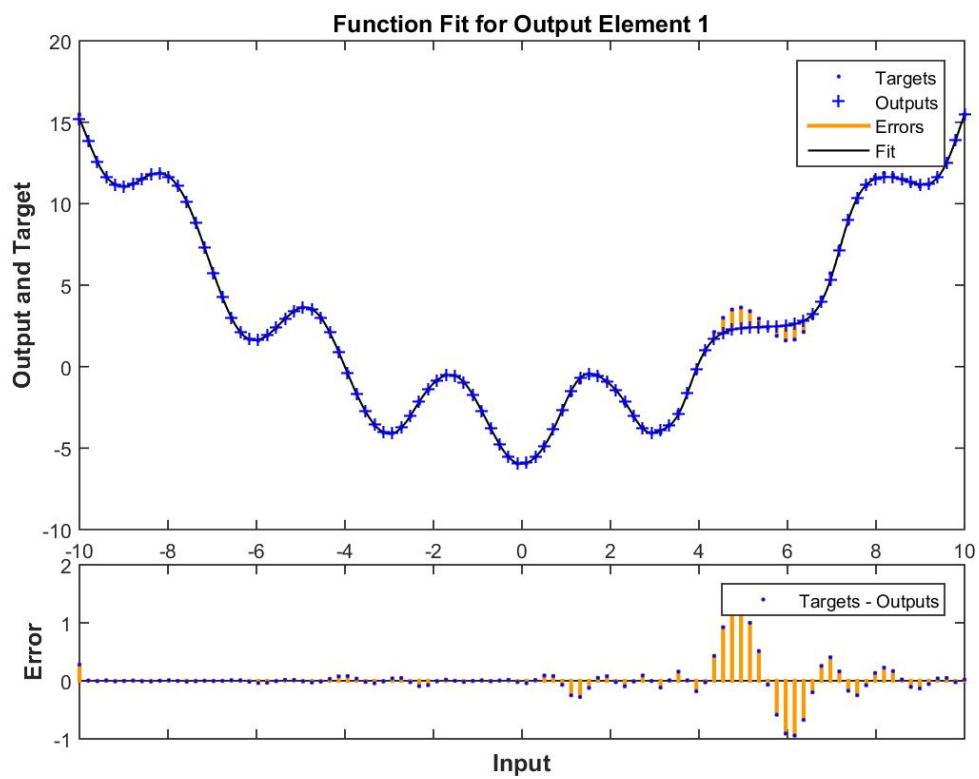
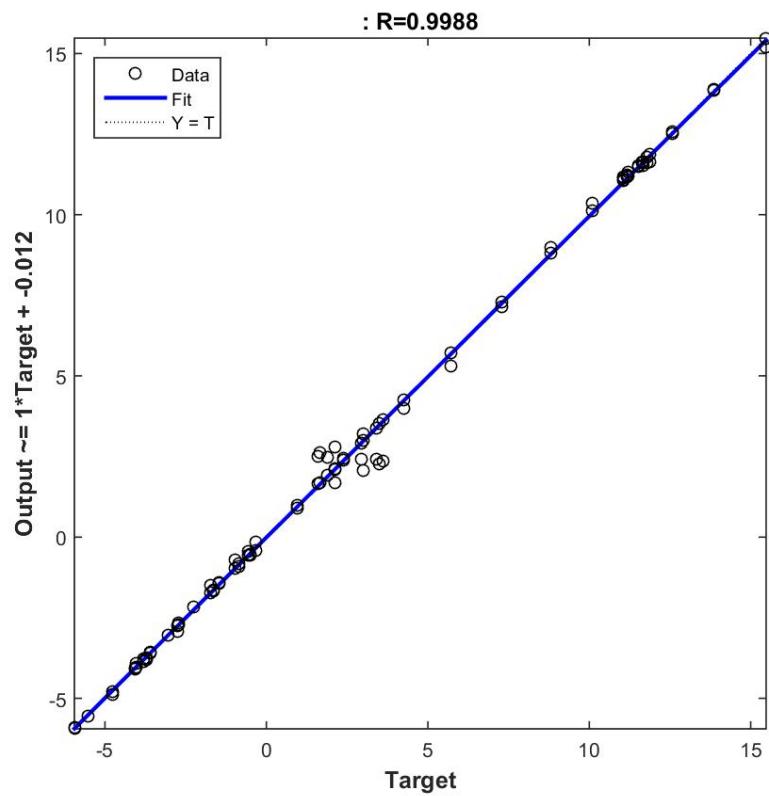
حالت اول: برای این تابع ابتدا از ساختار یک لایه پنهان و ۱۰ نورون استفاده می‌شود. همان‌طور که مشخص است نتایج مطلوب نیستند و در برخی نقاط خطای تا ۱.۸ مشاهده می‌شود. همچنین در شکل نمودار رگرسیون، نقاط کاملاً روی خط رگرسیون قرار نمی‌گیرند.



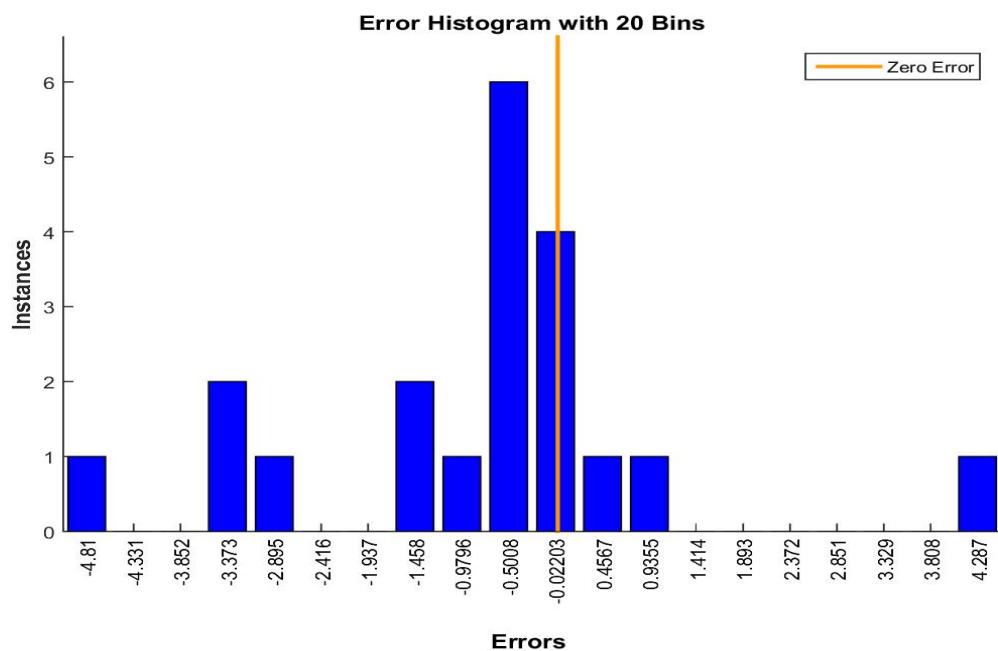
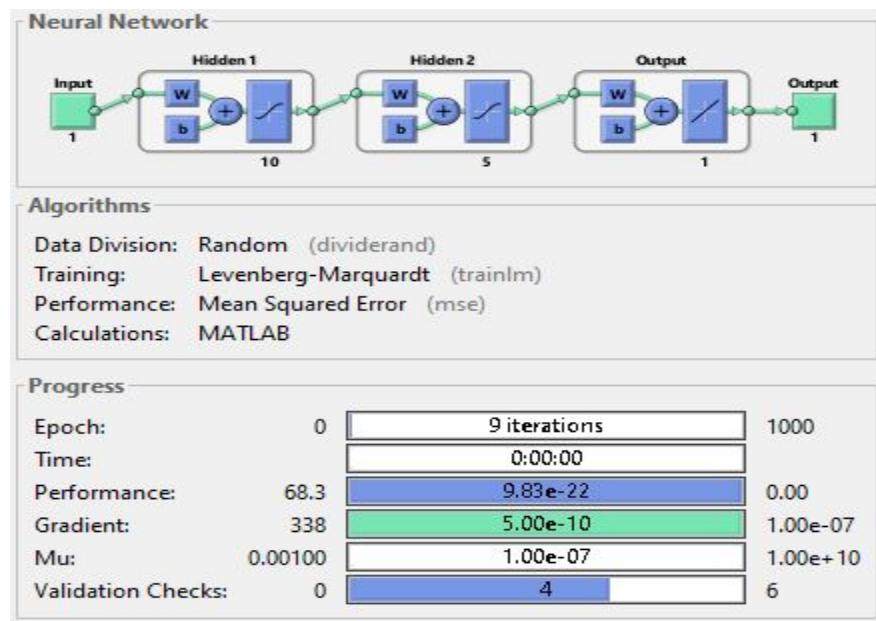


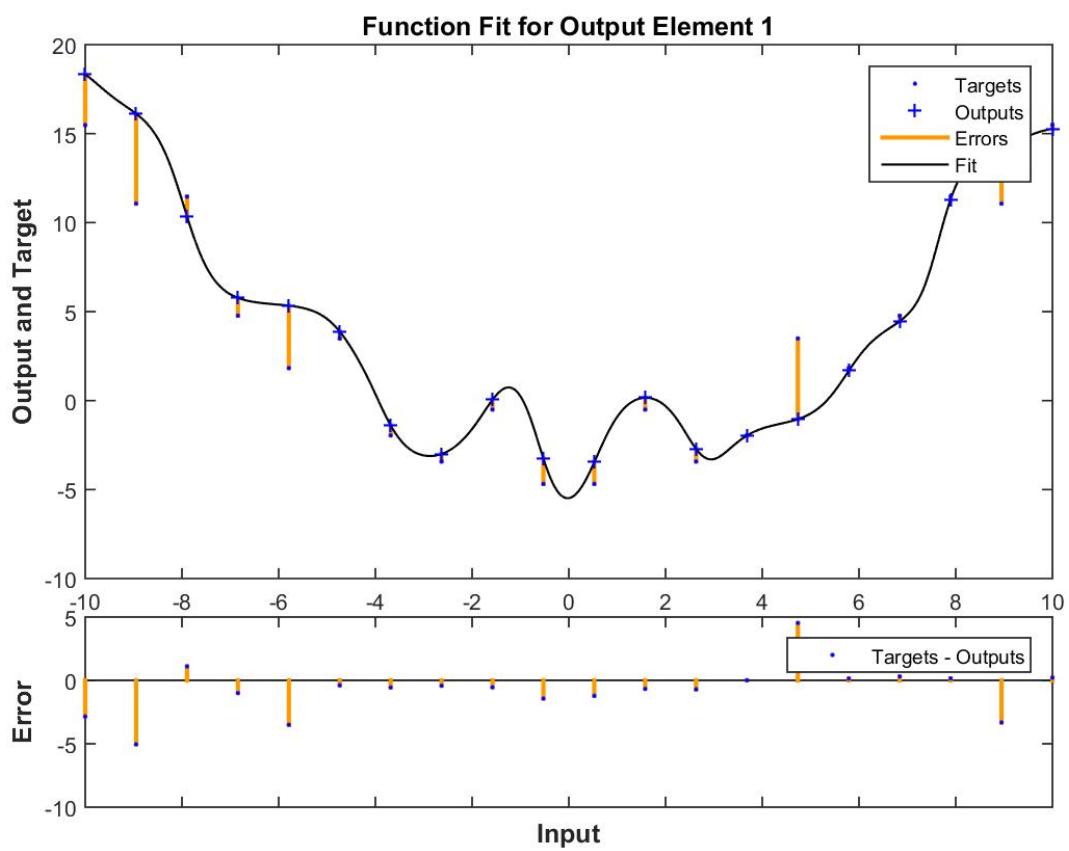
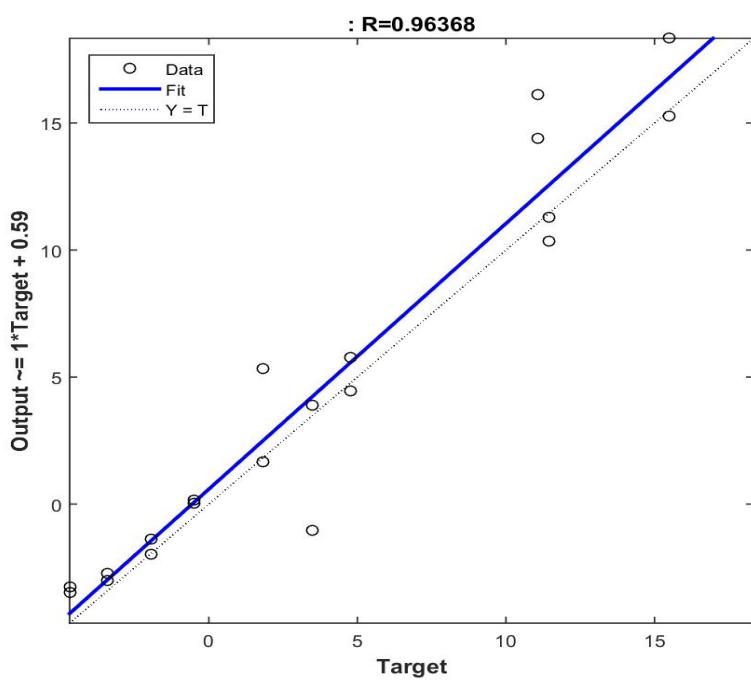
حالت دوم: با توجه به قسمت قبل ساختار ۲ لایه پنهان با ۱۰ و ۵ نورون برای شبکه عصبی در نظر گرفته می‌شود. در این حالت نتایج از حالت قبل بهتر است.



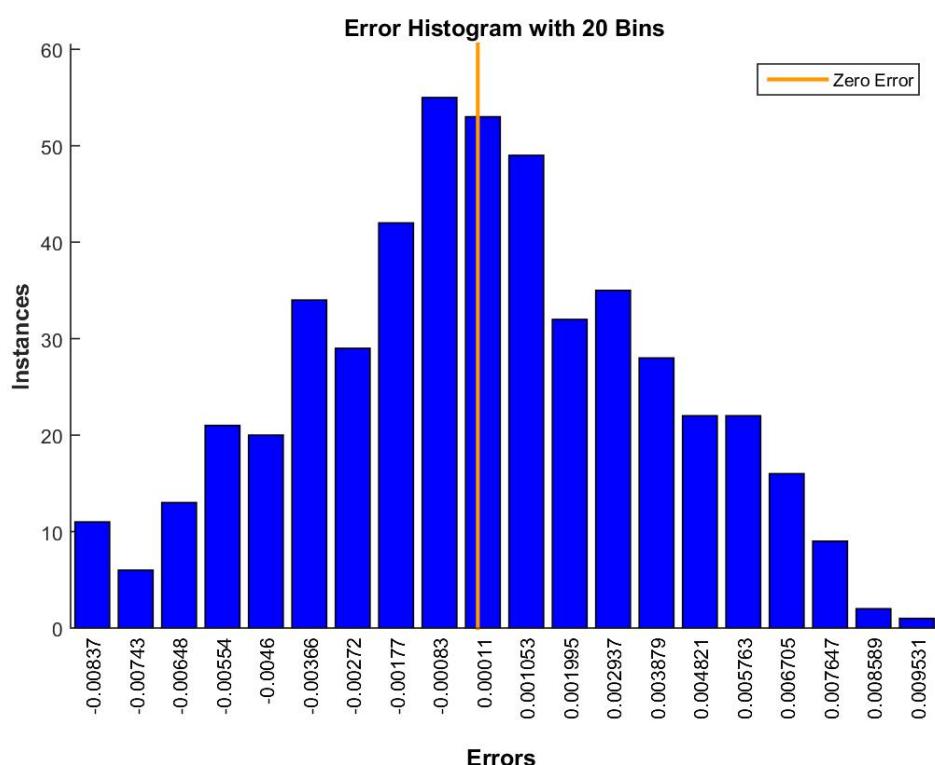
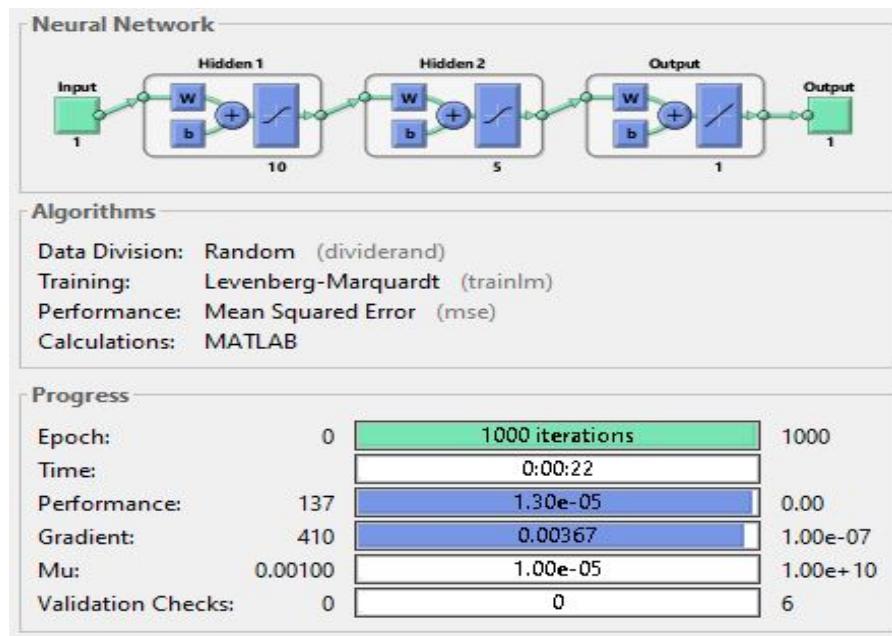


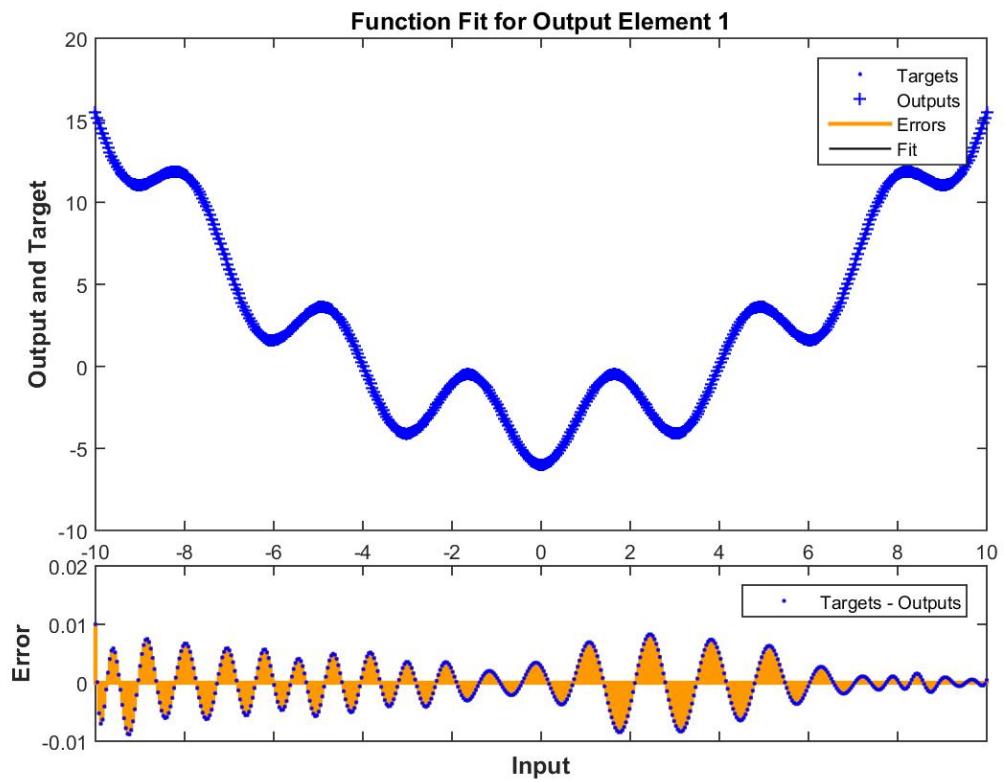
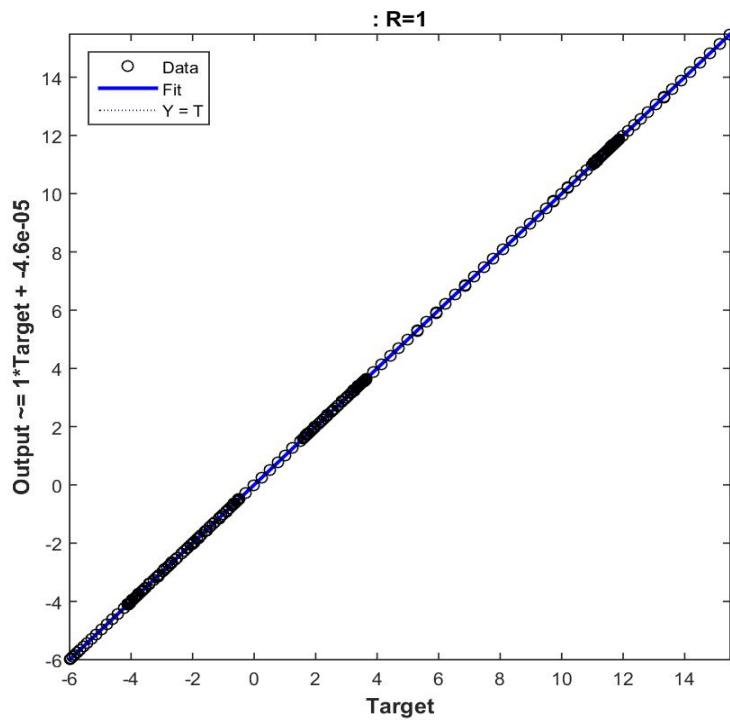
حالت سوم: در این حالت ساختار حالت قبل در نظر گرفته می‌شود؛ اما تعداد نمونه‌های آموزش به ۲۰ کاهش می‌یابد. همان‌طور که مشخص است نتایج حاصله قابل قبول نبوده و دقت شبکه عصبی نسبت به حالت قبل خیلی افت داشته است.





حالت چهارم: در این حالت ساختار حالت قبل در نظر گرفته می‌شود؛ اما تعداد نمونه‌های آموزش به ۵۰۰ افزایش می‌یابد. همان‌طور که مشخص است نتایج حاصله بسیار دقیق شده‌اند و از بیشینه خطای ۲ به بیشینه خطای حدود ۱۰۰ رسیده‌ایم که دقیق بالای شبکه عصبی را در این شرایط نشان می‌دهد.





نتیجه‌گیری: با توجه به بررسی حالت‌های مختلف شبکه عصبی برای توابع ذکر شده، می‌توان به نتایج زیر پی برد:

۱. معمولاً با پیچیدگی توابع بایستی برای دستیابی به دقت موردنظر تعداد نورون‌ها و تعداد لایه‌های پنهان را افزایش داد؛ اما این موضوع بدین معنی نیست که همیشه با افزایش تعداد لایه و تعداد نورون‌ها میزان دقت افزایش می‌یابد.
۲. به طور کلی افزایش تعداد نمونه‌های آزمایش می‌تواند موجب افزایش دقت گردد؛ اما این نتیجه هم کلی نبوده و همان‌طور که دیده شد، با افزایش تعداد نمونه‌های آزمایش به ۵۰۰ در برخی توابع دقت عملکرد شبکه عصبی کاهش پیدا می‌کرد.
۳. در برخی توابع که تغییرات ناگهانی در خروجی دارند، شبکه عصبی نمی‌تواند به خوبی آن تابع را تخمین بزند و دقت قابل قبولی ندارد.

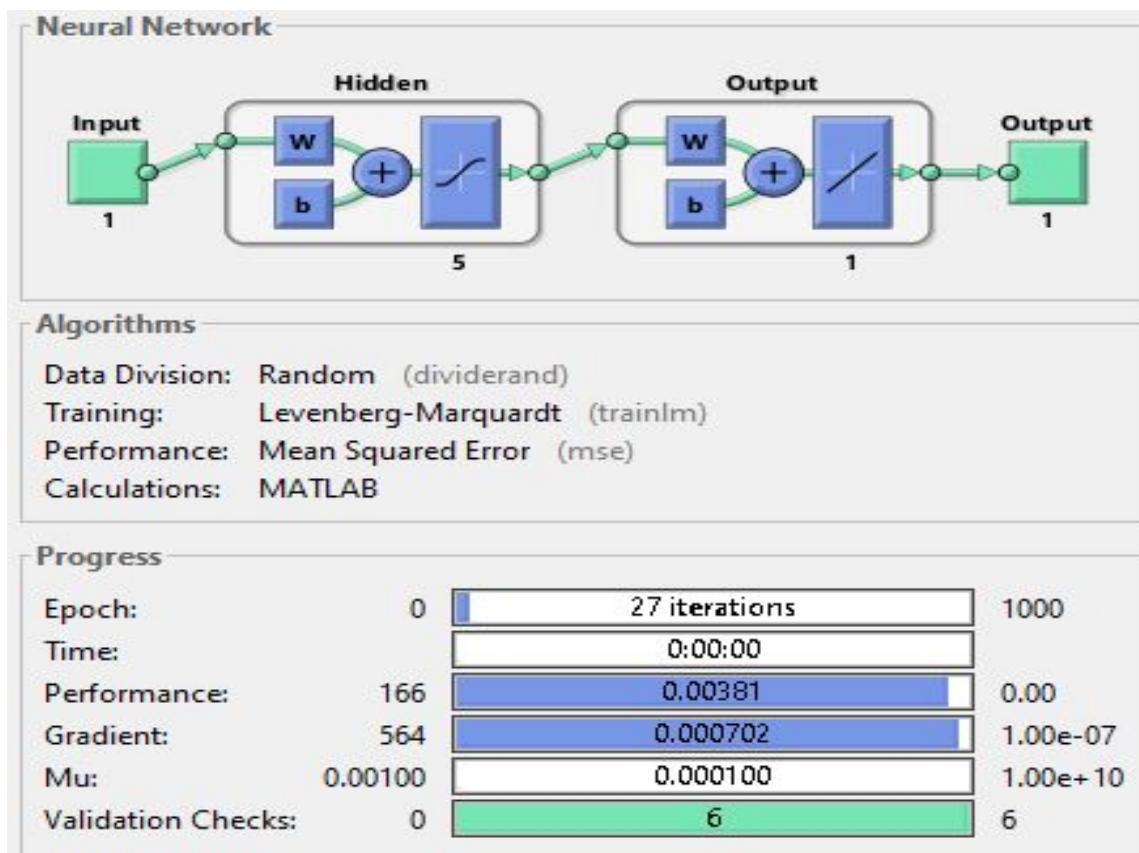
بخش دوم:

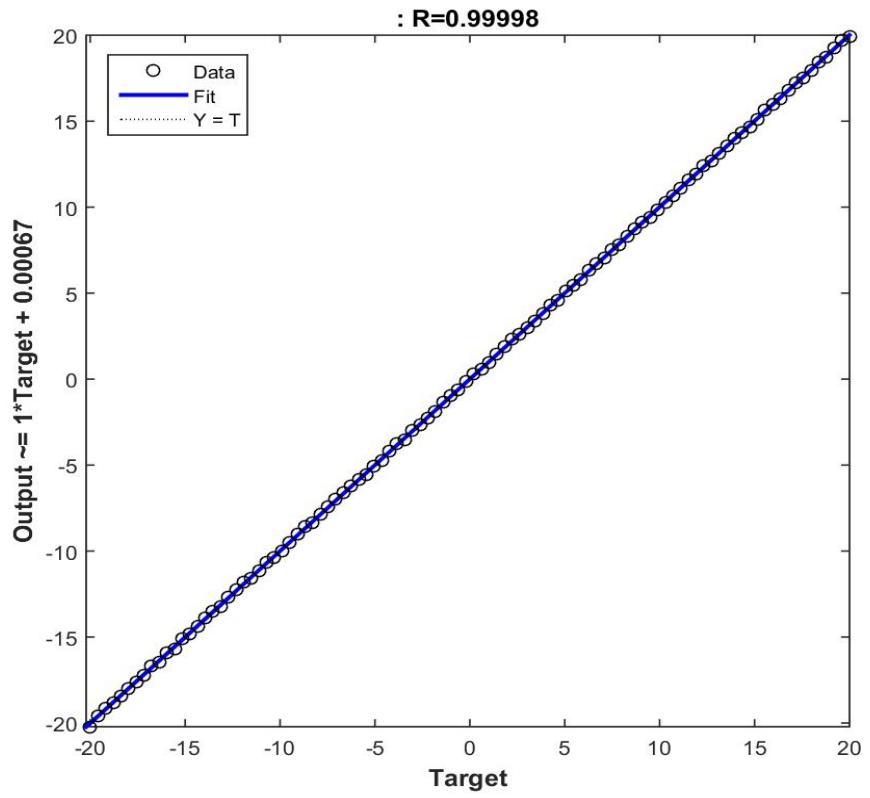
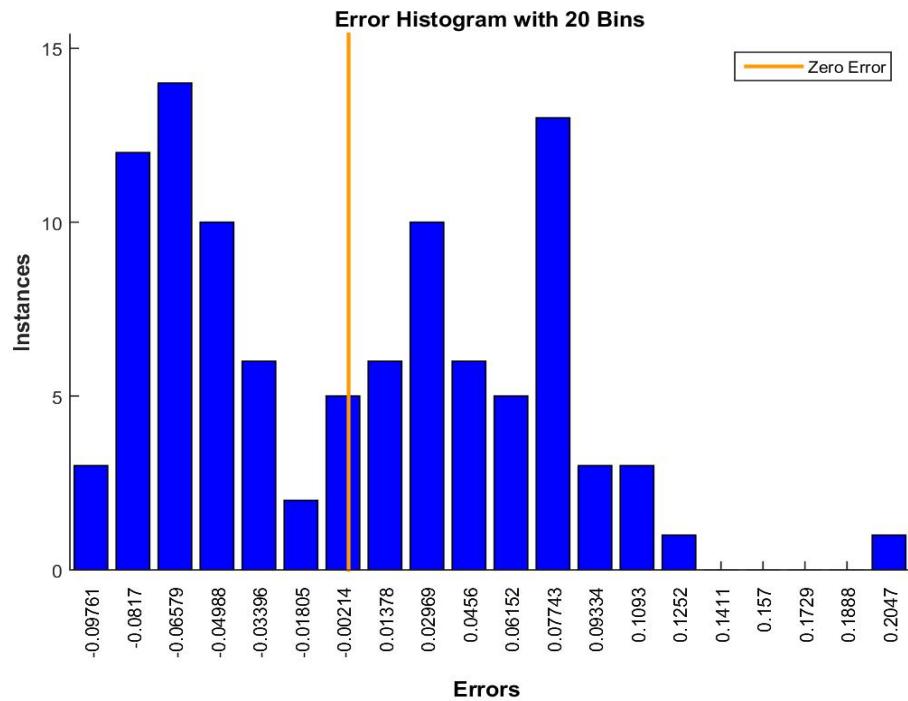
در این بخش ۶ تابع برای آموزش در شبکه عصبی در نظر گرفته شده است. بازه داده های x برای هر تابع از ۱۰- تا ۱۰ و تعداد ورودی ها ۱۰۰ در نظر گرفته شده است. برای اضافه نمودن نویز به ورودی ها، از تابع `rand` استفاده شده است، به این صورت که این تابع مقداری تصادفی بین ۱ و صفر بصورت یکنواخت تولید می کند. این مقادیر با ضرایبی که معرف دامنه نویز هستند به ورودی های اصلی اضافه می شوند. در این بخش نویز با دامنه ۰.۱ و ۵ در نظر گرفته شده است. در این بخش، ساختاری برای شبکه عصبی انتخاب می شود که بهترین نتایج را در بخش قبل داشته اند.

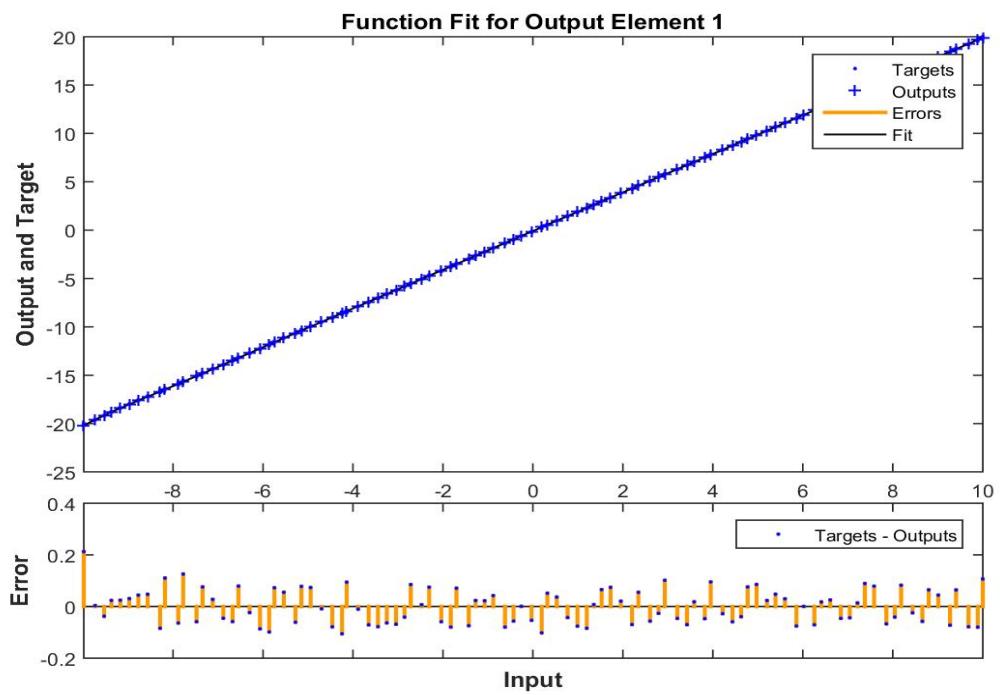
تابع اول:

$$y_1 = 2 * x;$$

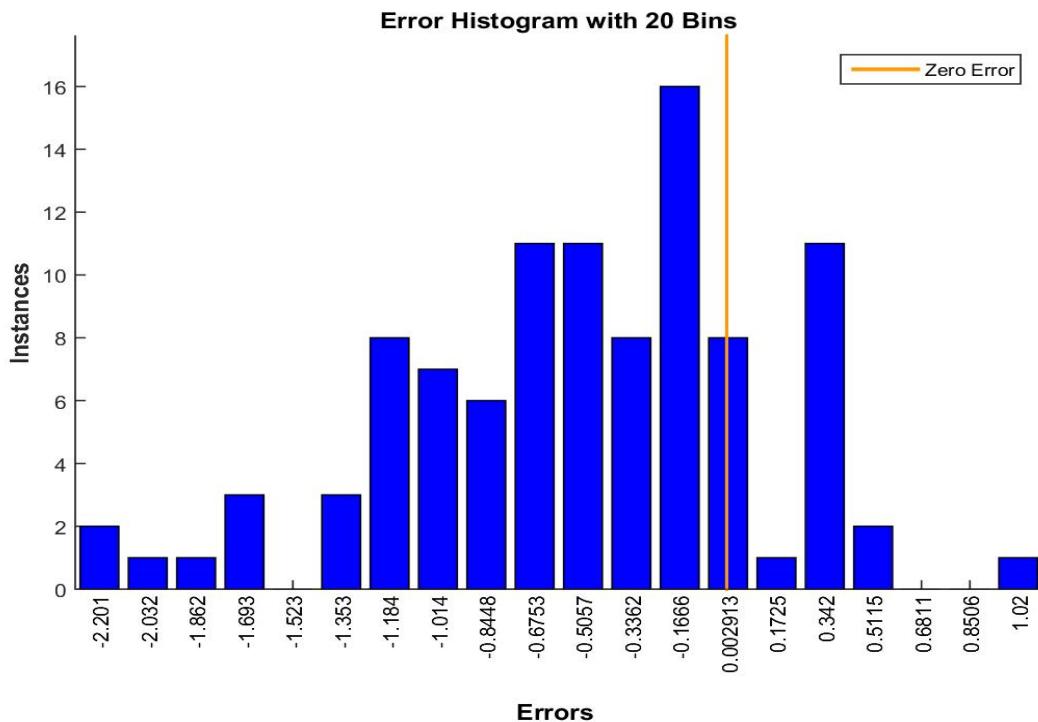
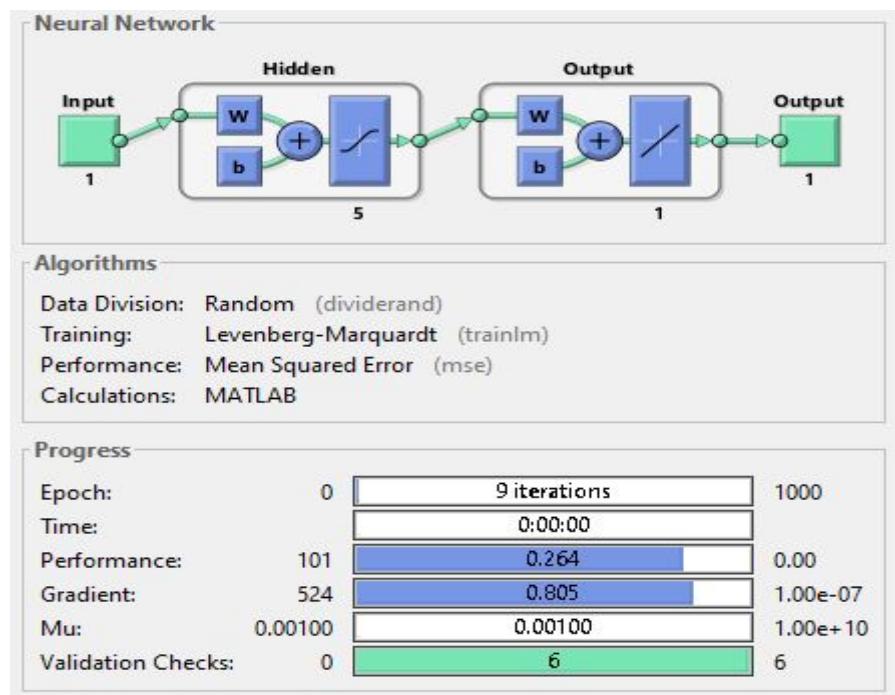
حالت اول: در حالت اول نویز با دامنه ۰.۱. یعنی کمترین مقدار در نظر گرفته شده به تابع اضافه می شود.

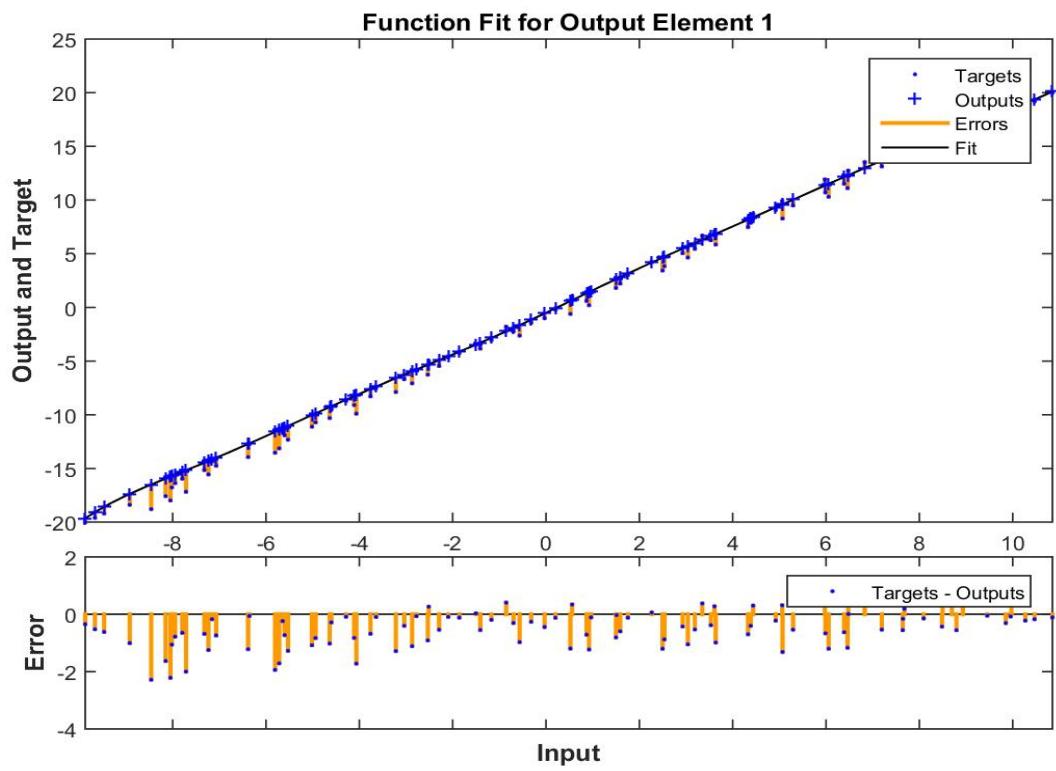
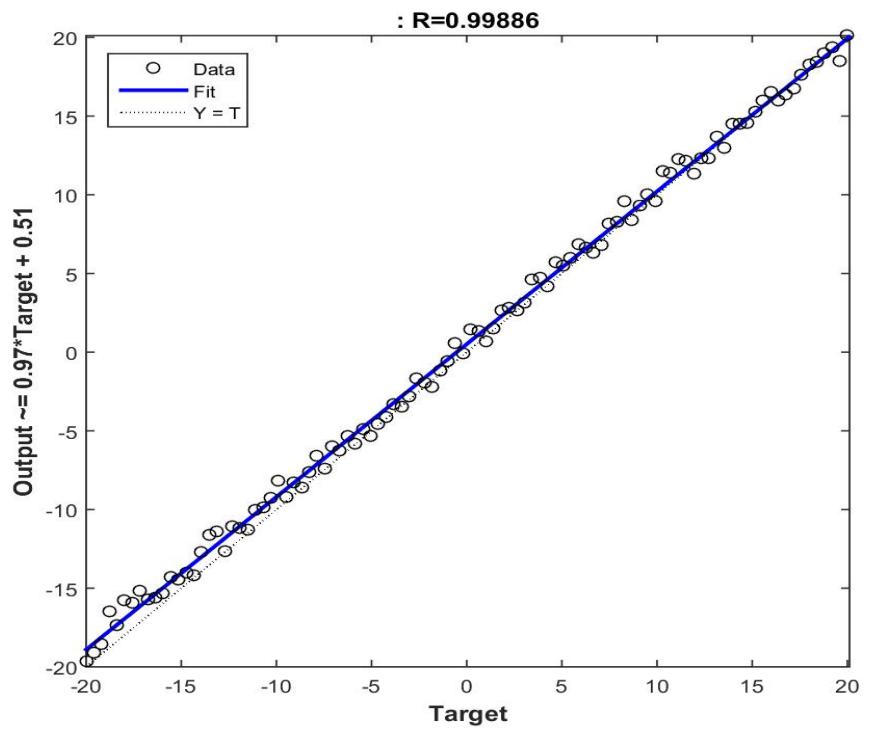




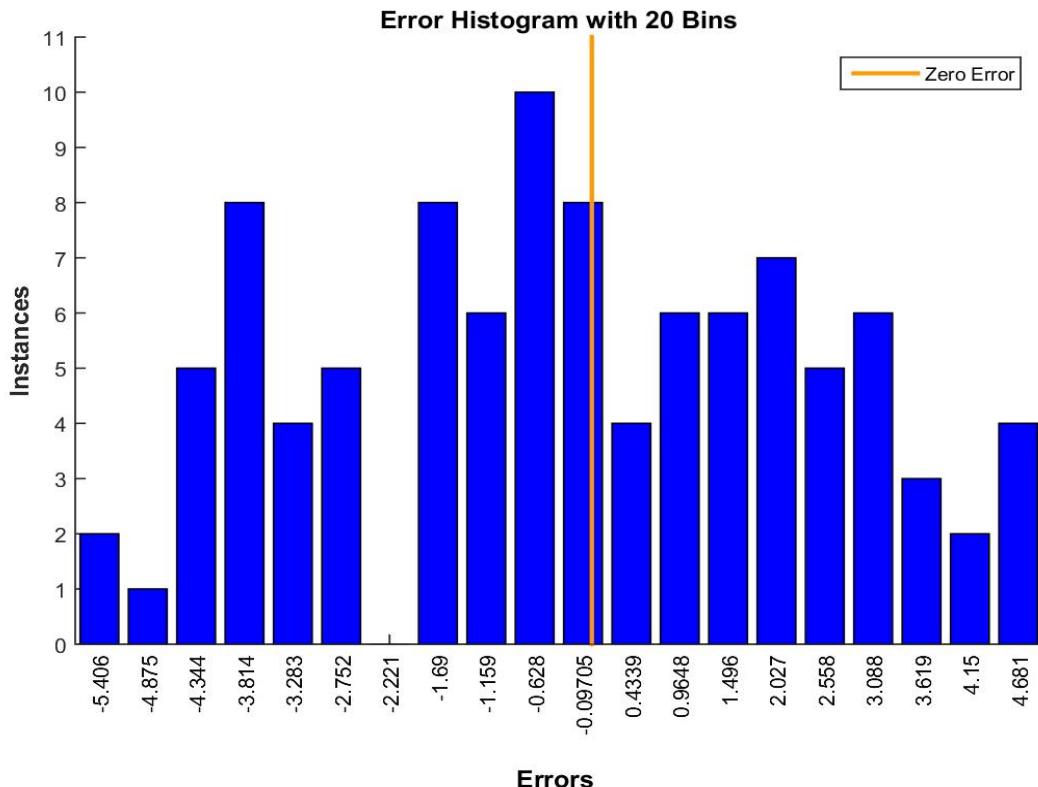
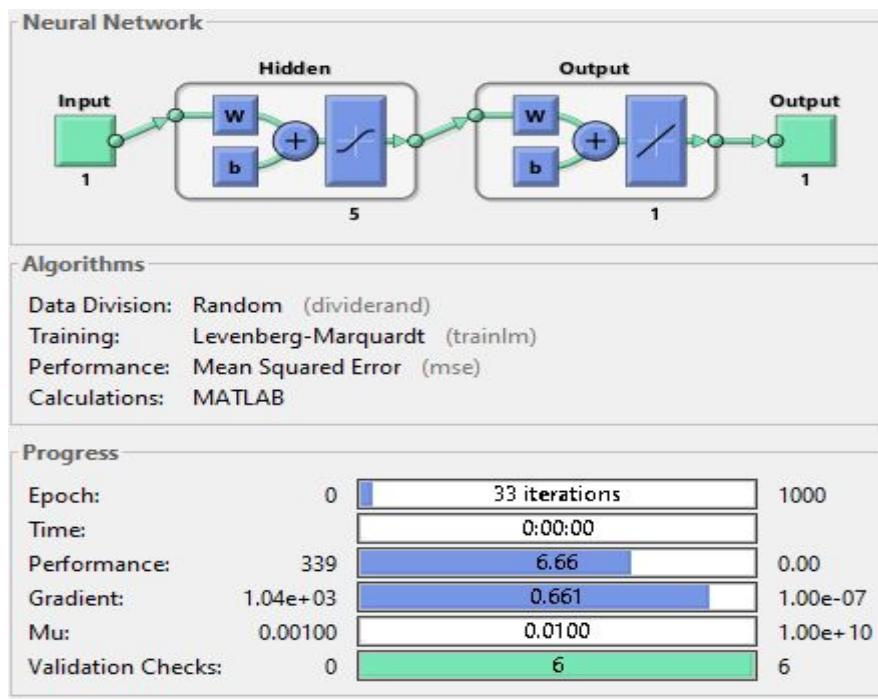


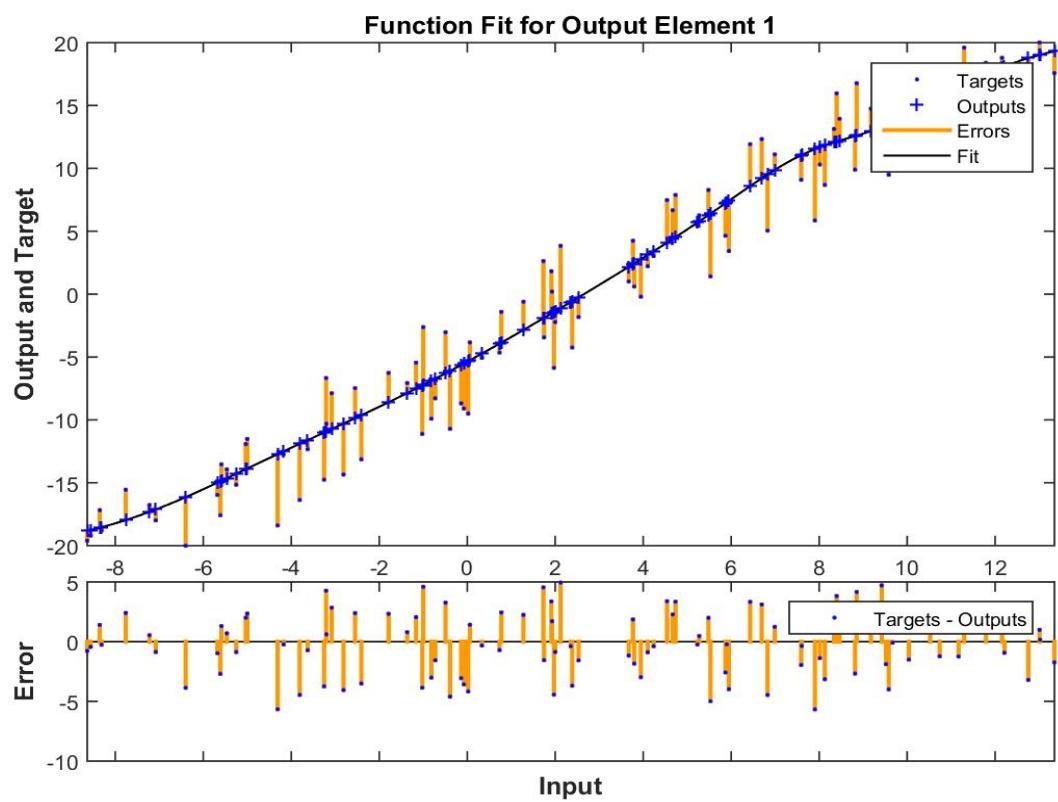
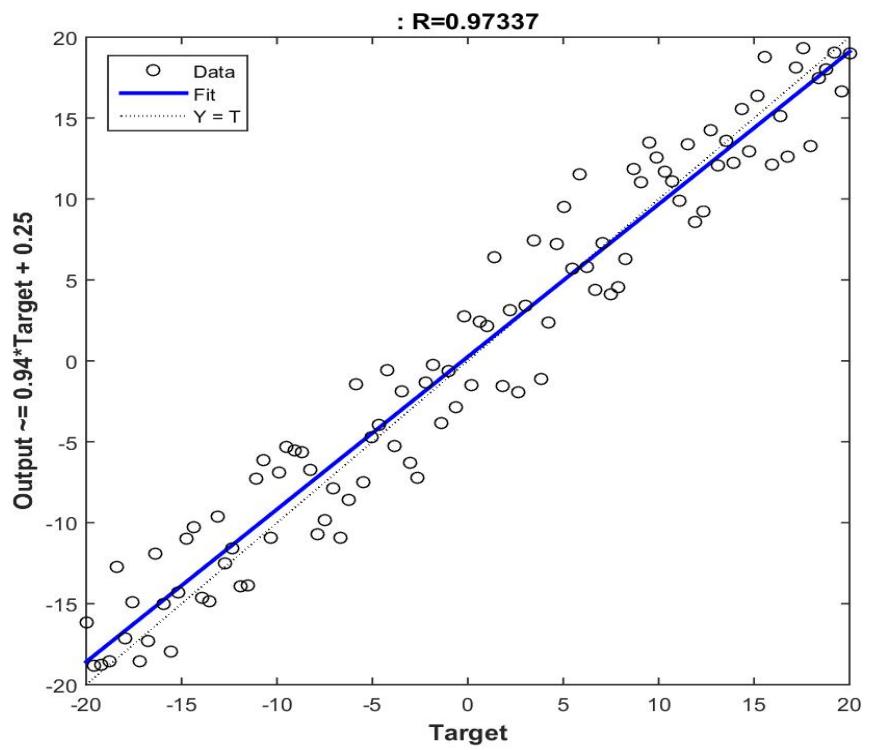
حالت دوم: در این حالت نویز با دامنه ۱ به سیگنال ورودی اضافه می‌شود.





حالت سوم: در این حالت نویز با دامنه ۵ به سیگنال ورودی اضافه می‌شود.

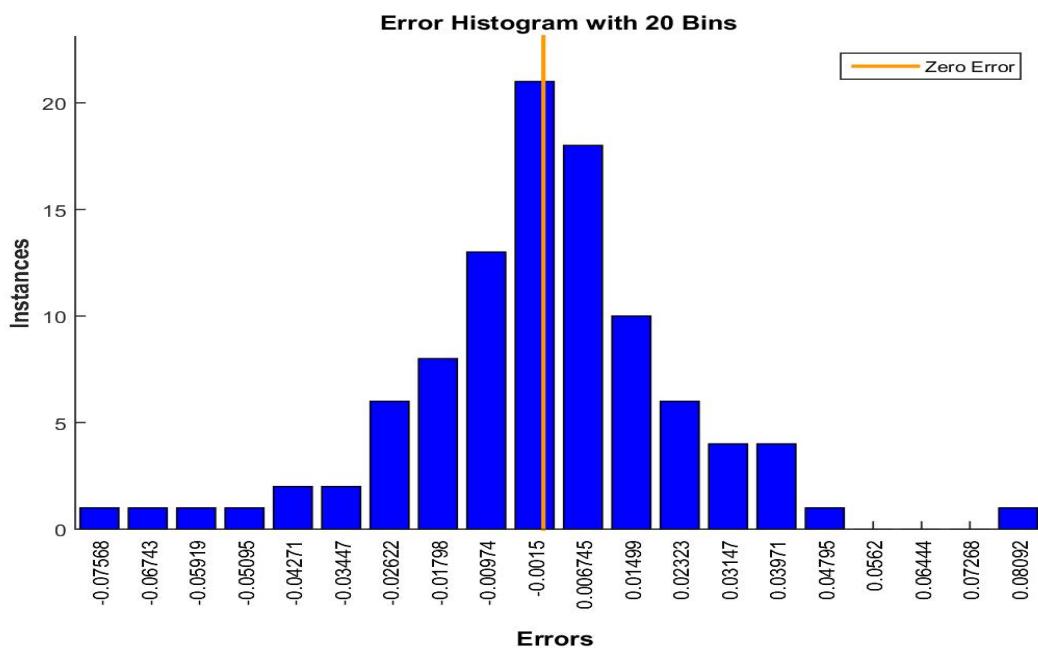
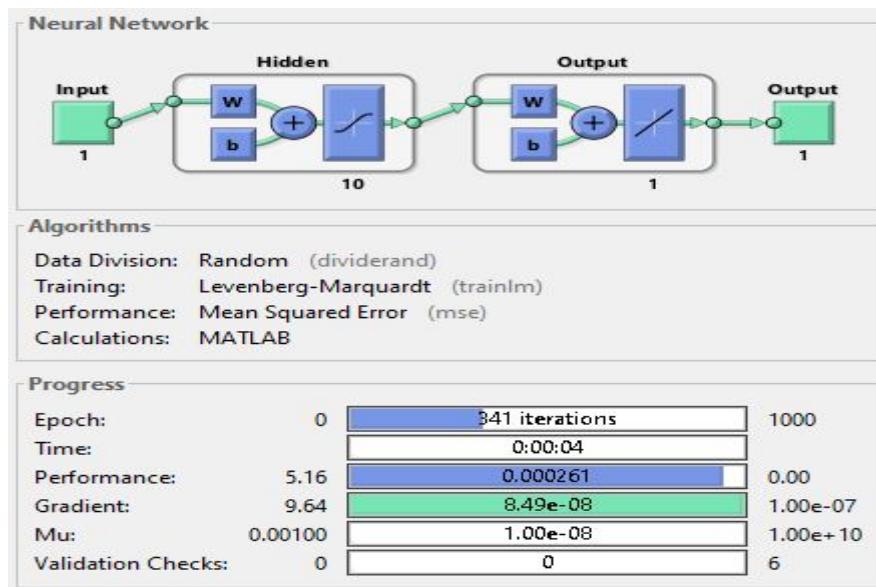


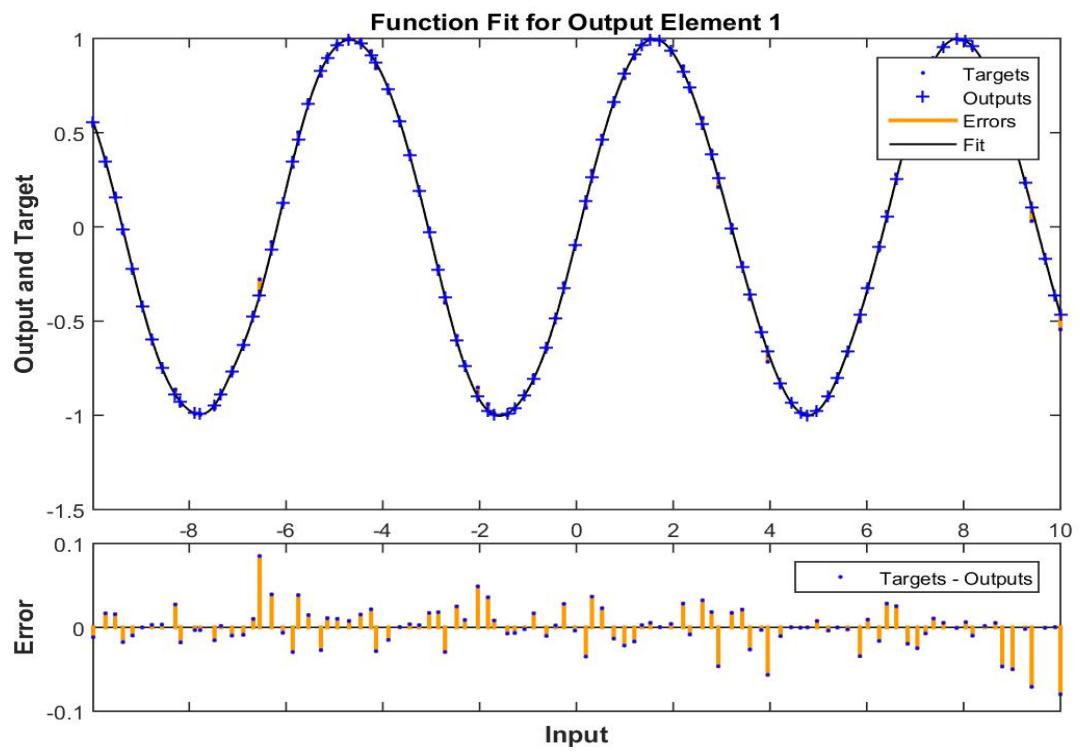
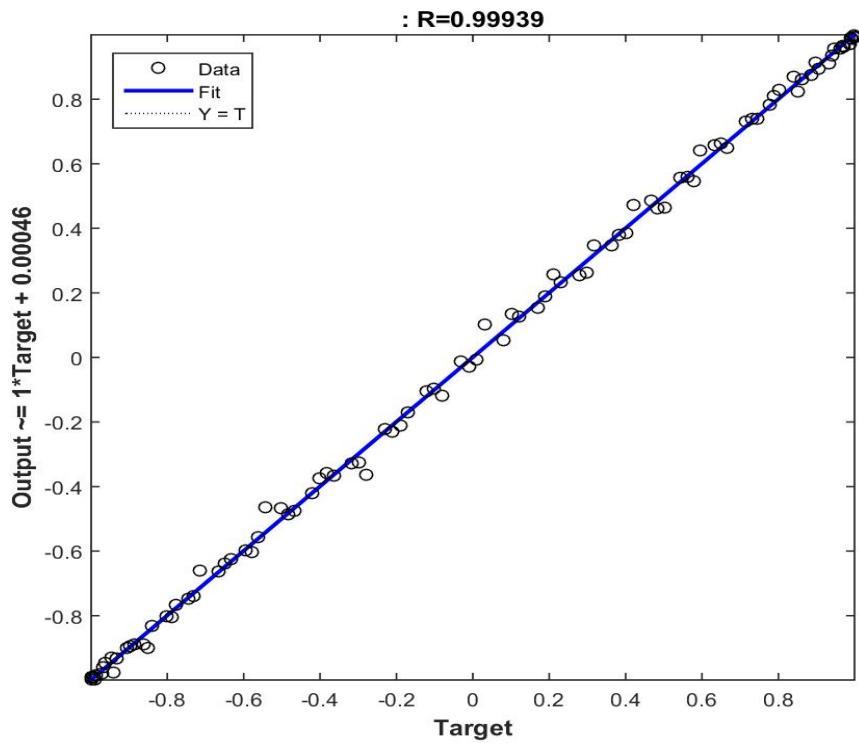


تابع دوم:

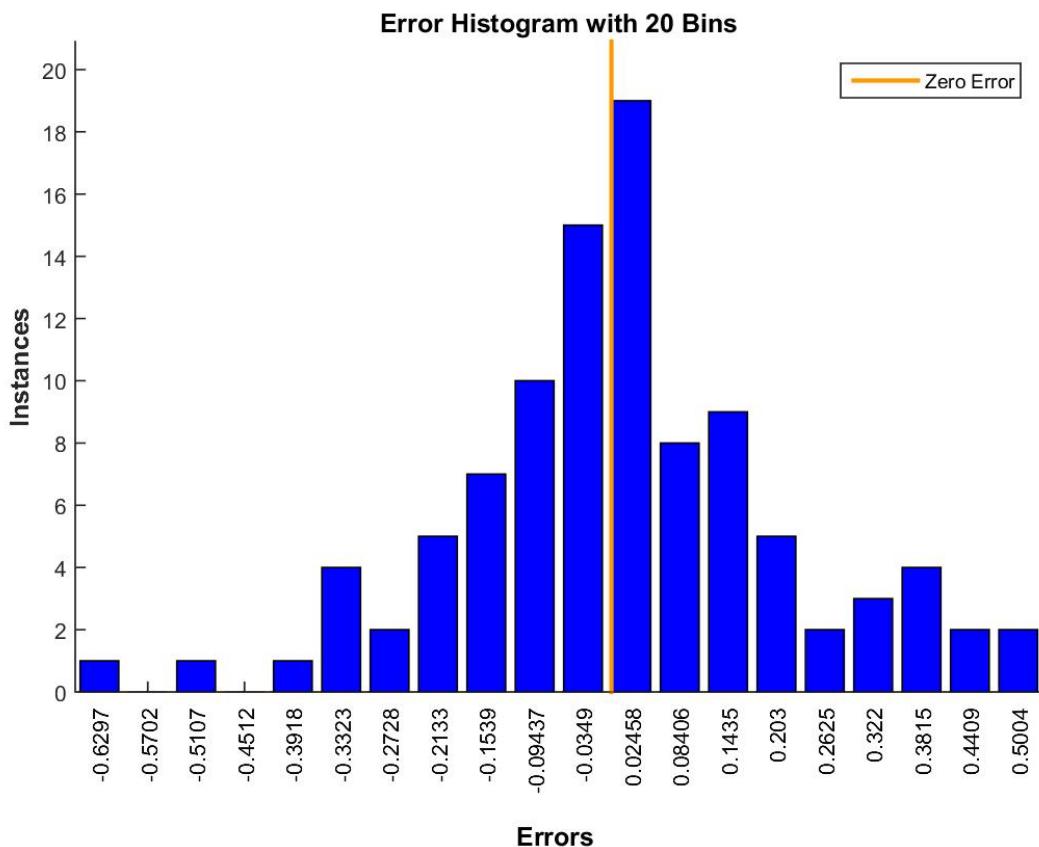
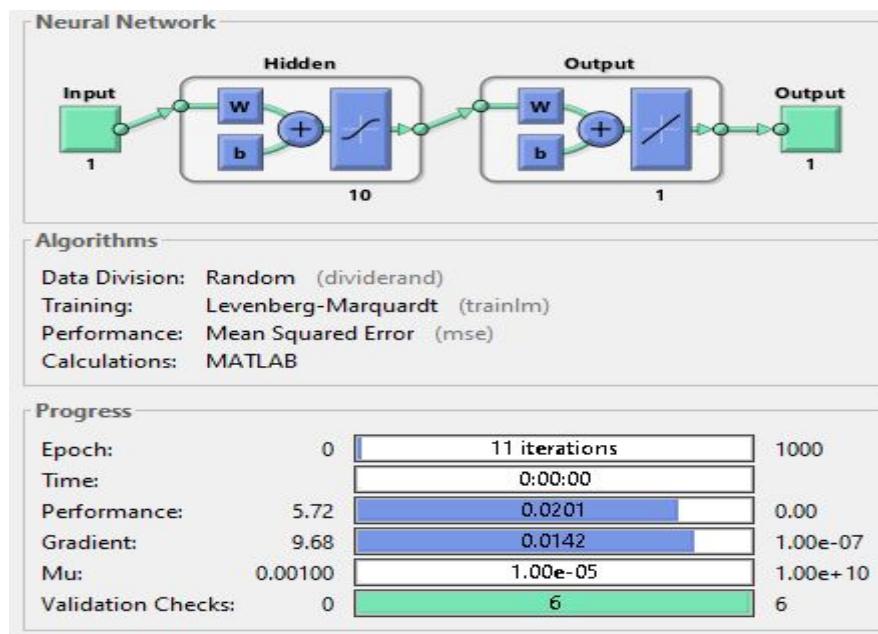
$$y_2 = \sin(x);$$

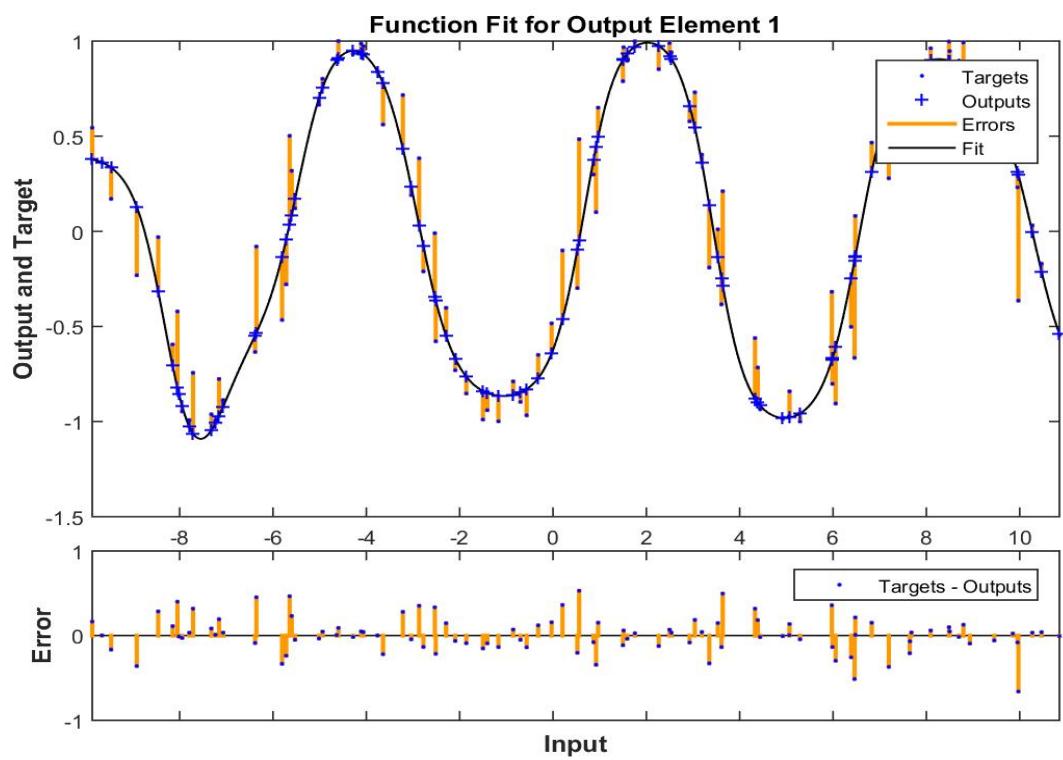
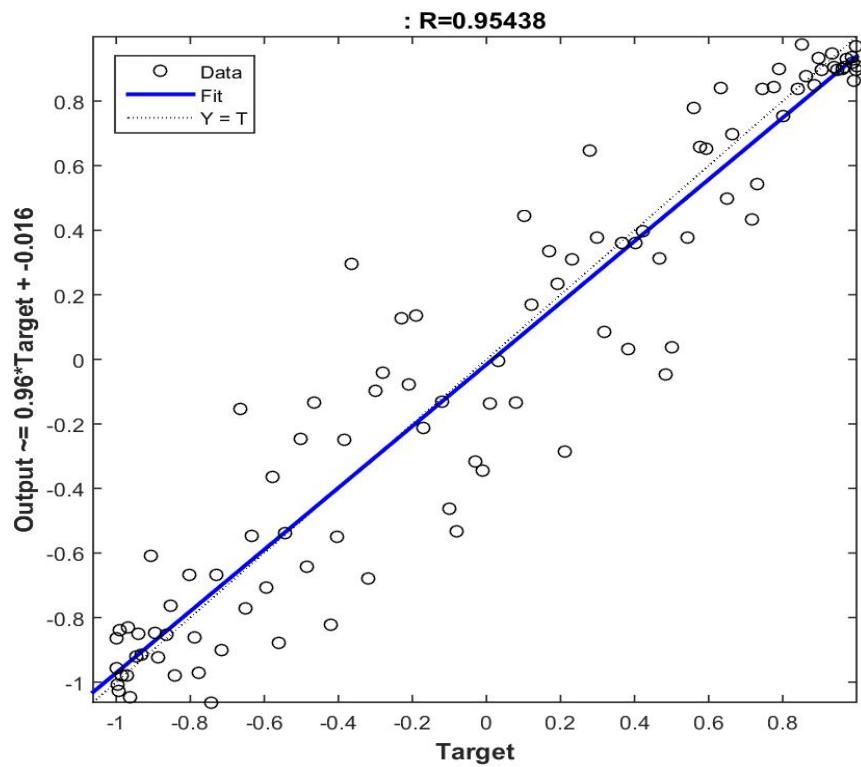
حالت اول: در حالت اول نویز با دامنه $[-0.1, 0.1]$ یعنی کمترین مقدار در نظر گرفته شده به تابع اضافه می‌شود.



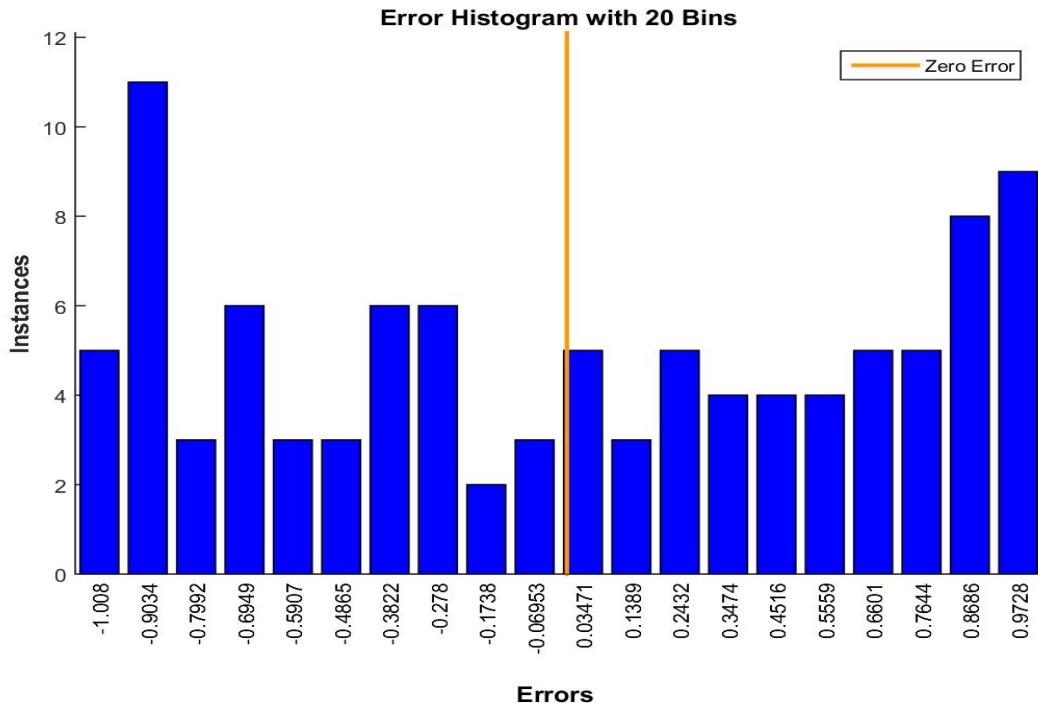
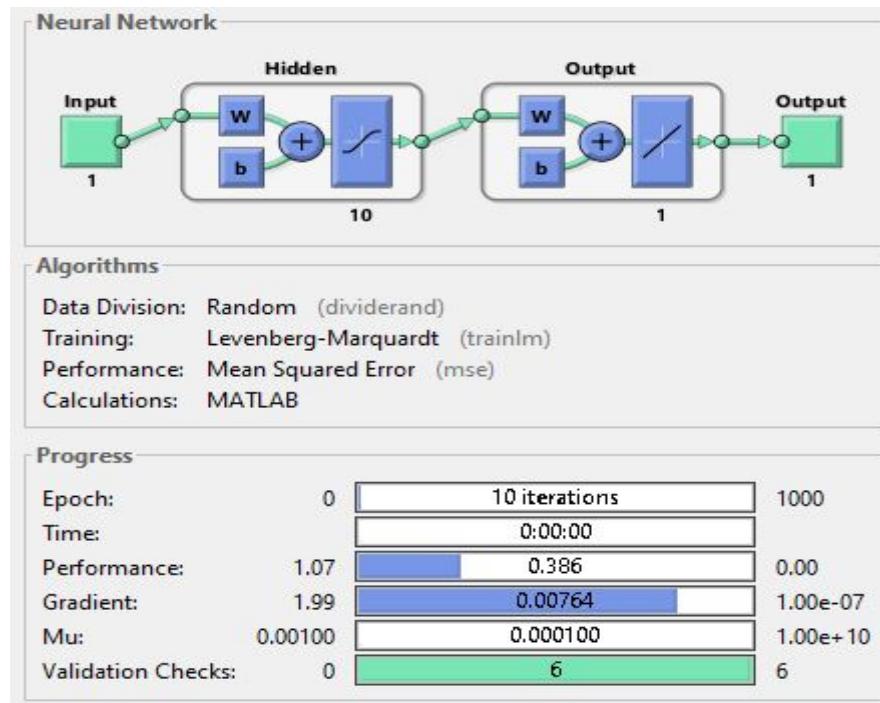


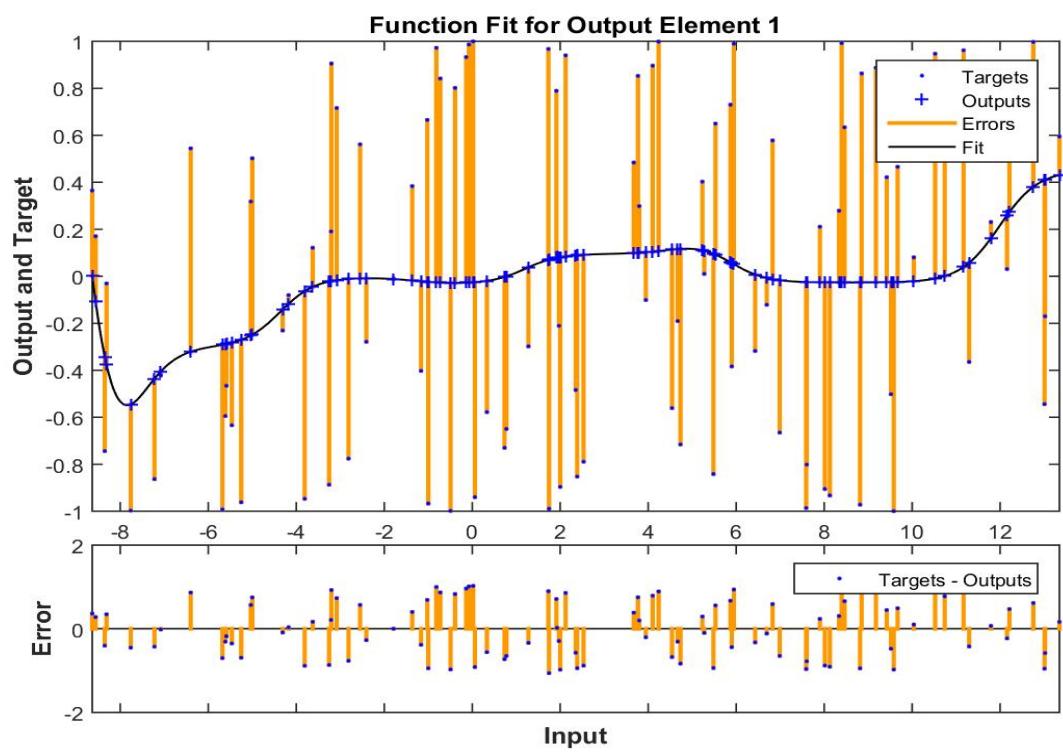
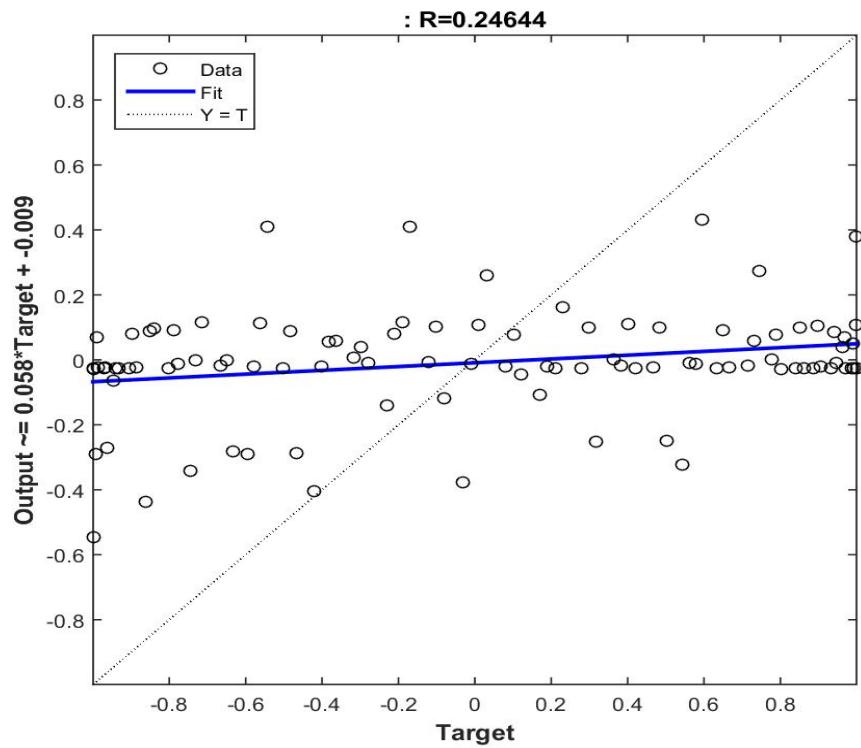
حالت دوم: در این حالت نویز با دامنه ۱ به سیگنال ورودی اضافه می‌شود.





حالت سوم: در این حالت نویز با دامنه ۵ به سیگنال ورودی اضافه می‌شود.

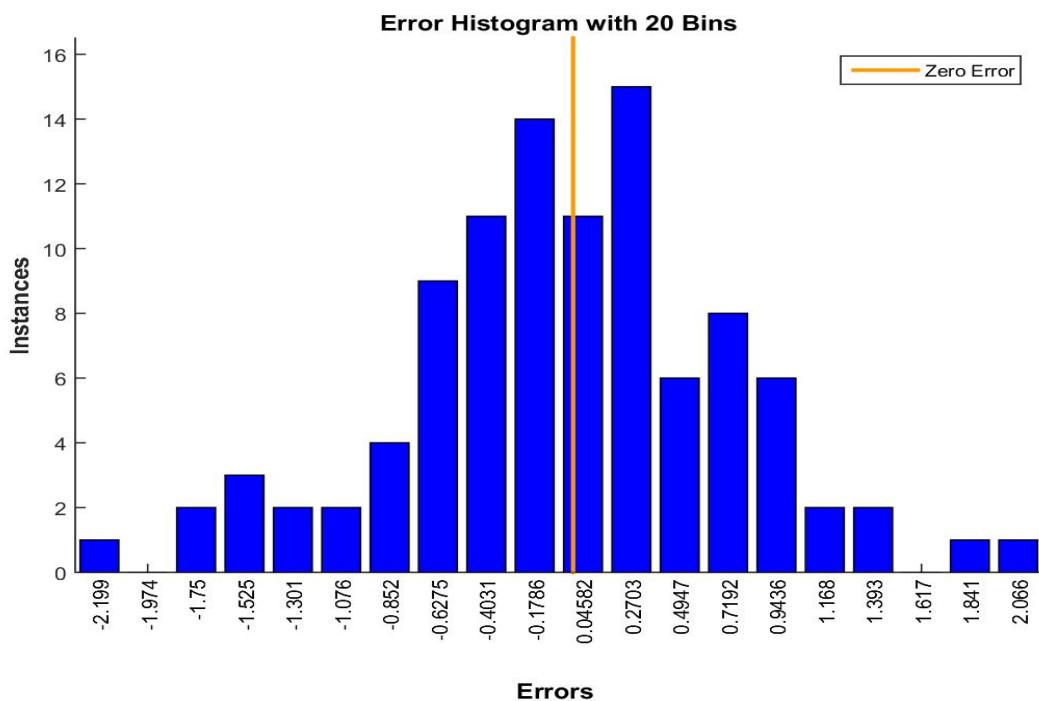
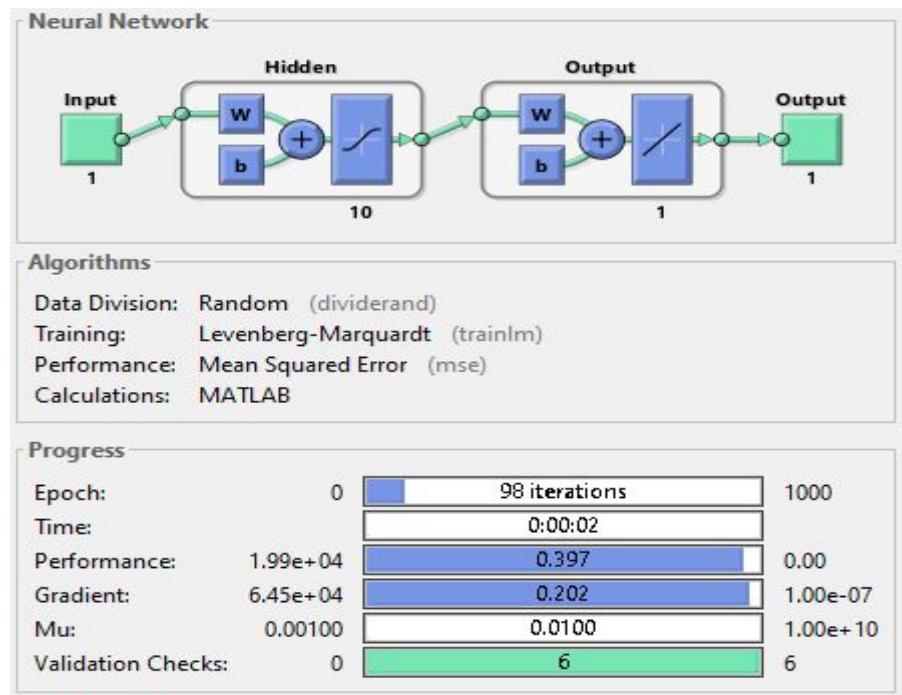


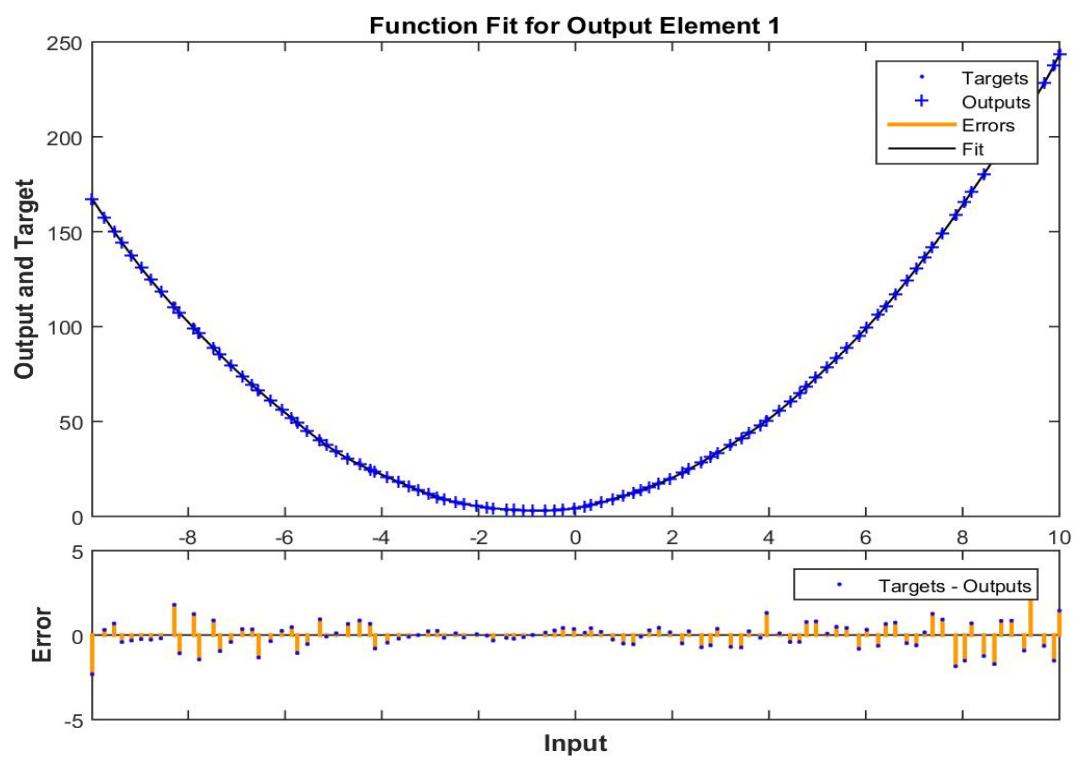
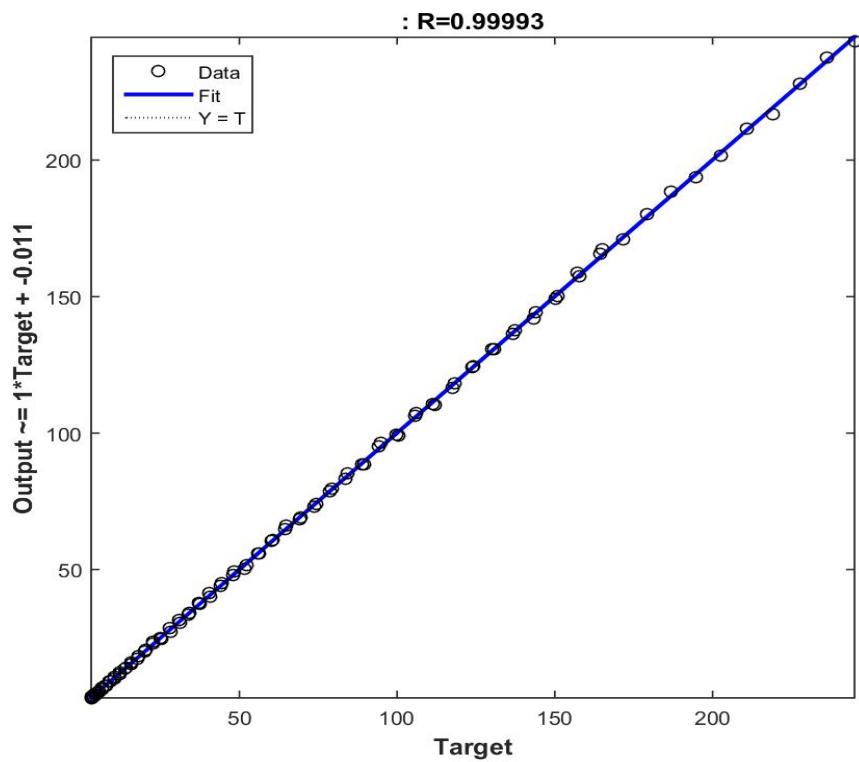


تابع سوم:

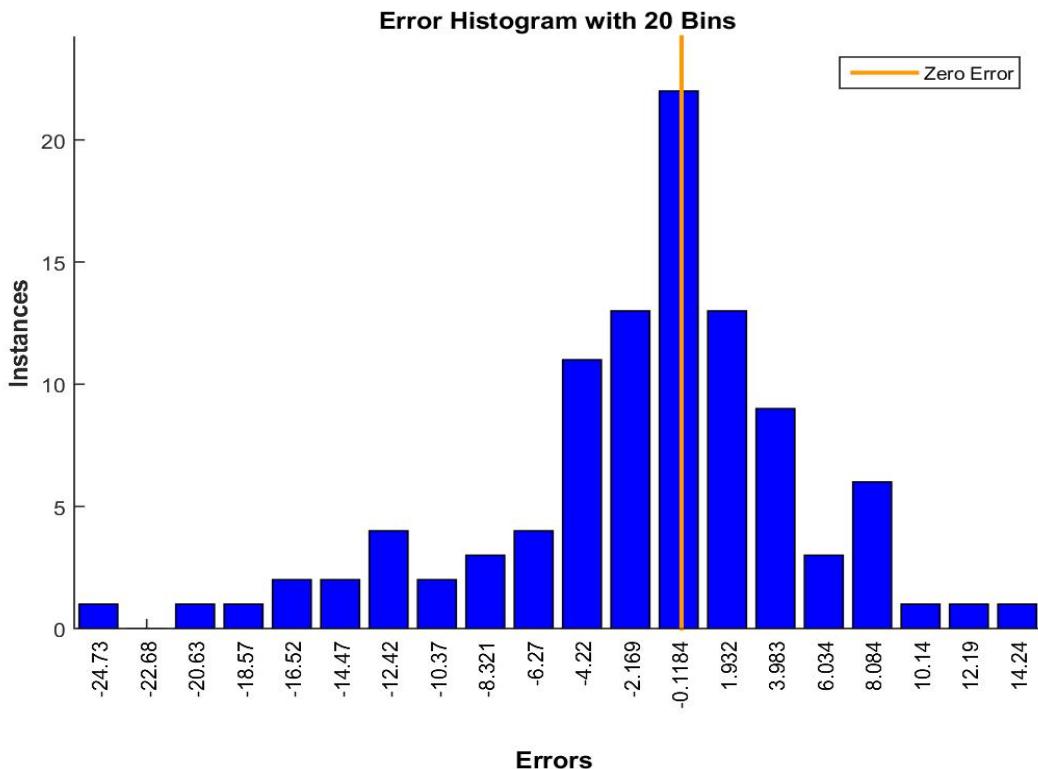
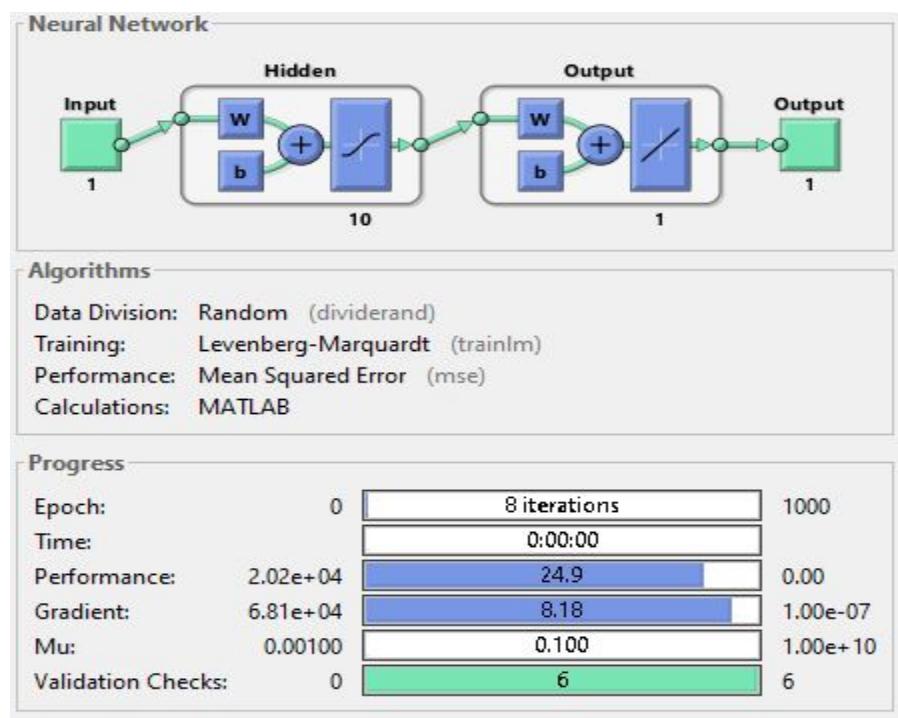
$$y_3 = 2 \cdot x^2 + 4 \cdot x + 5;$$

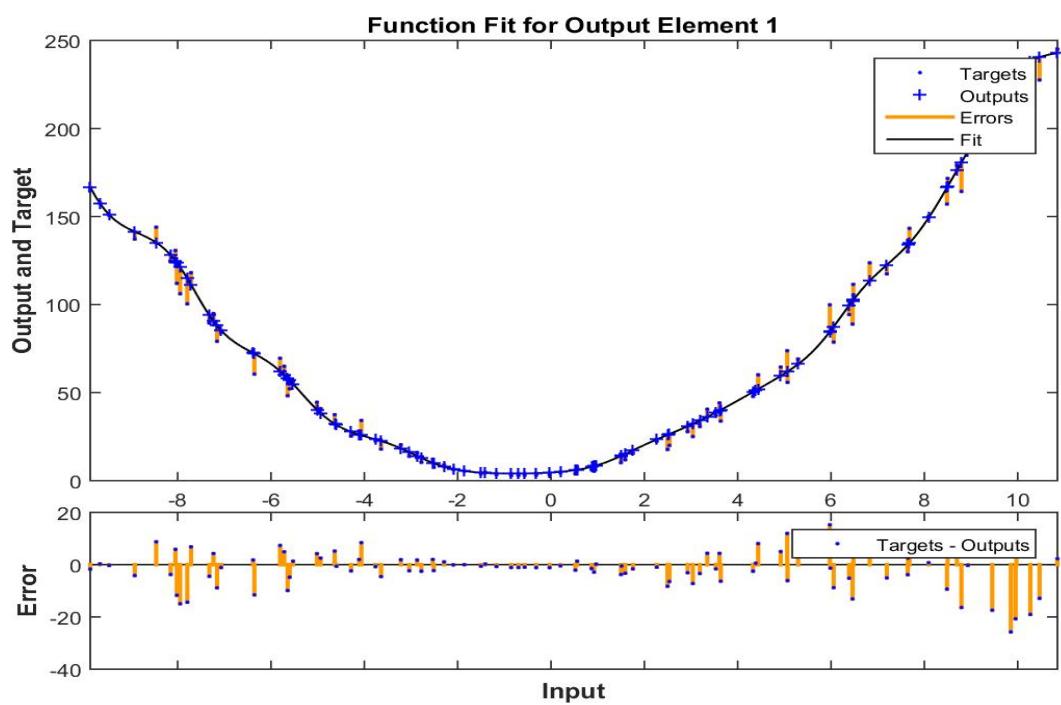
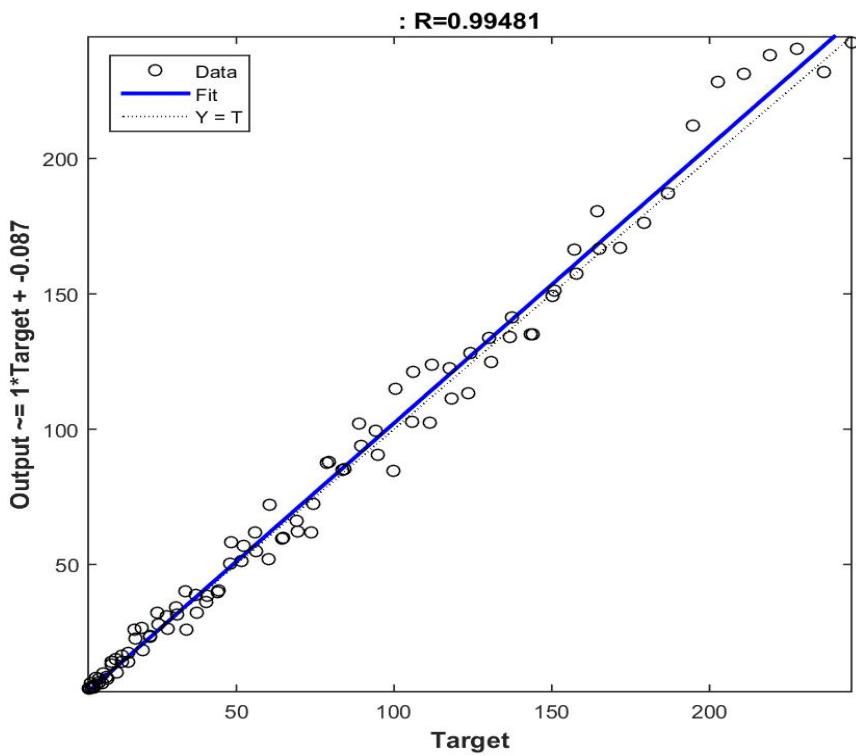
حالت اول: در حالت اول نویز با دامنه $[-1, 1]$ یعنی کمترین مقدار در نظر گرفته شده به تابع اضافه می‌شود.



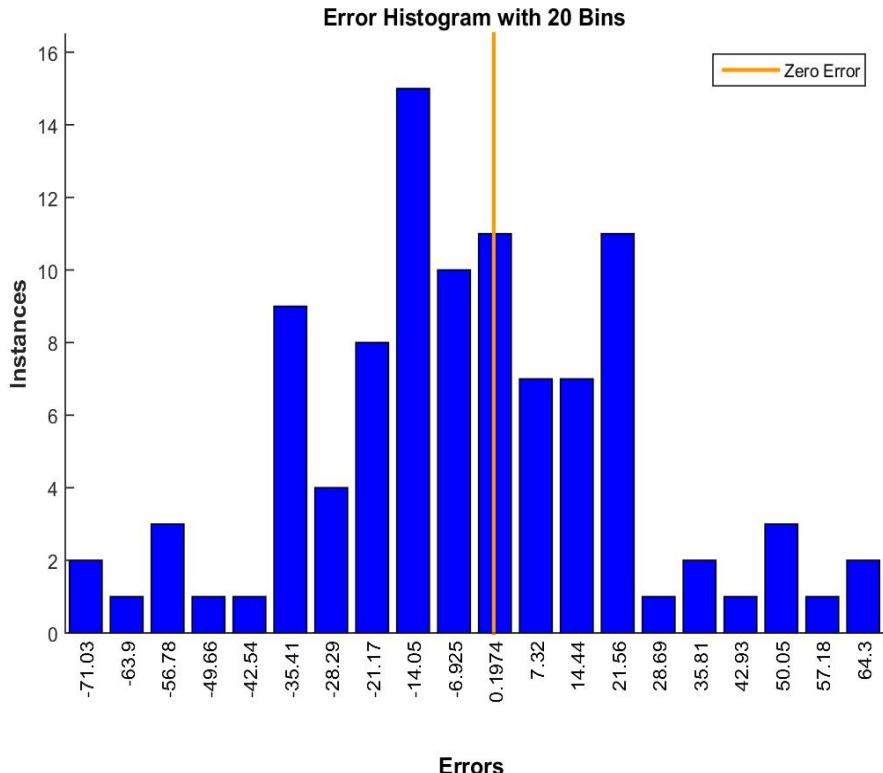
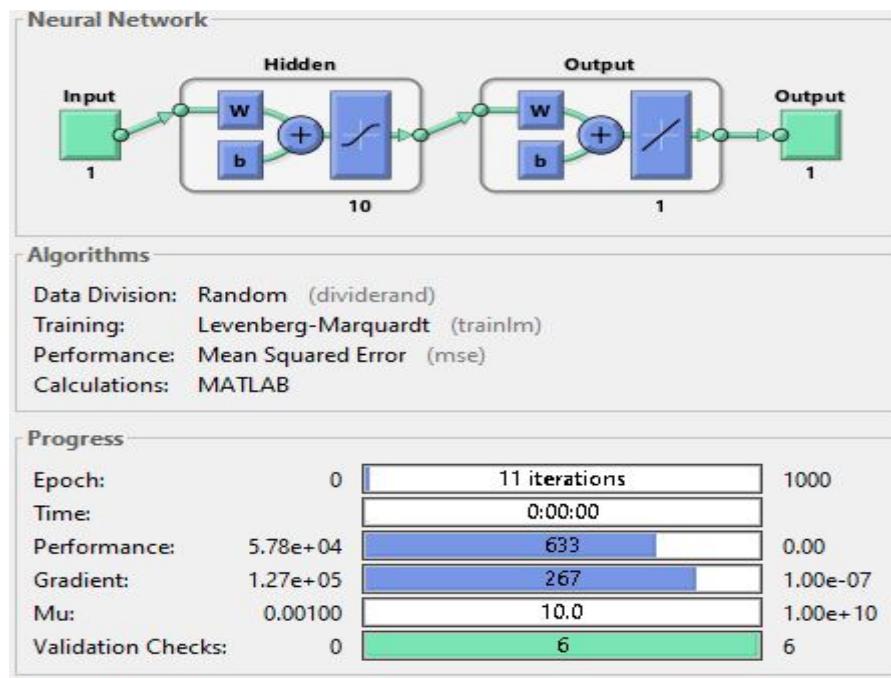


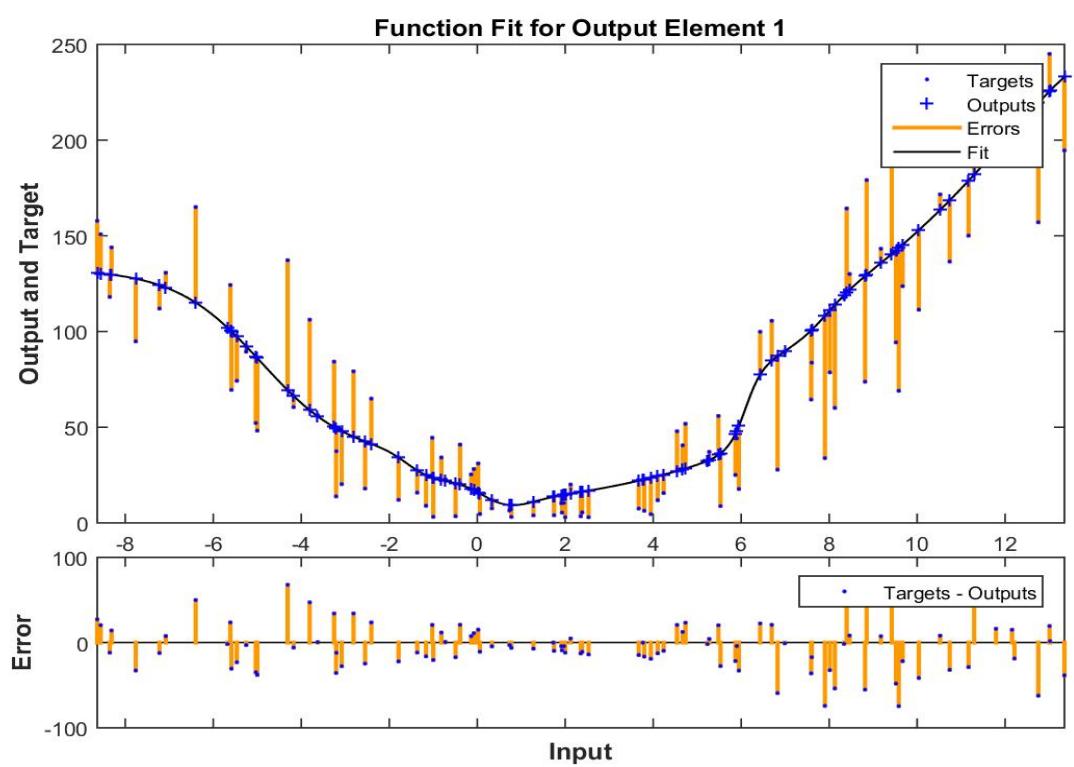
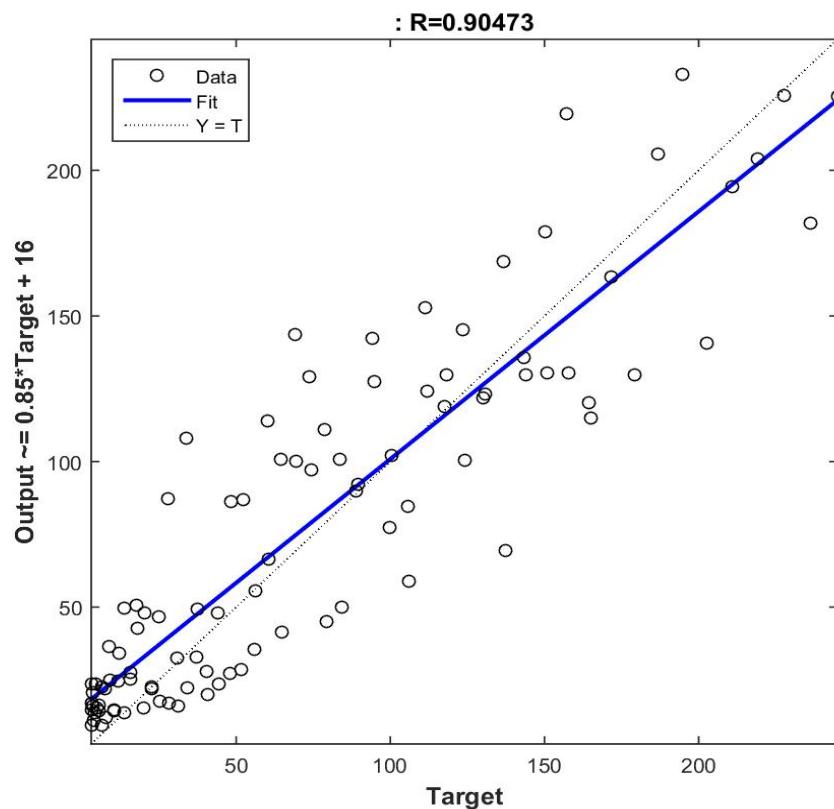
حالت دوم: در این حالت نویز با دامنه ۱ به سیگنال ورودی اضافه می‌شود.





حالت سوم: در این حالت نویز با دامنه ۵ به سیگنال ورودی اضافه می‌شود.

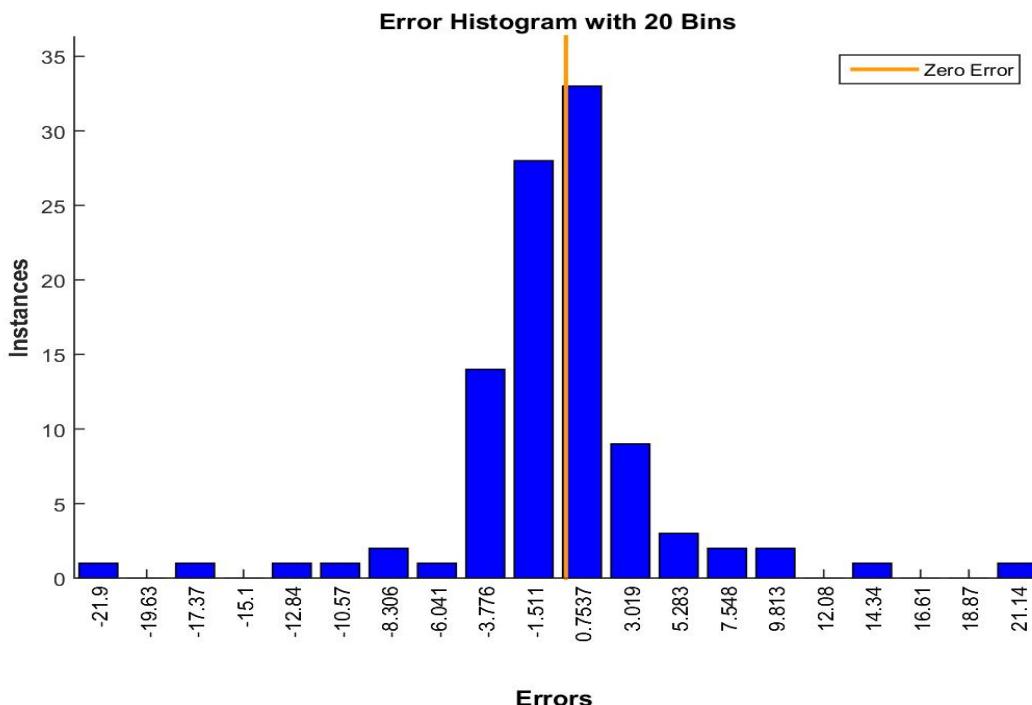
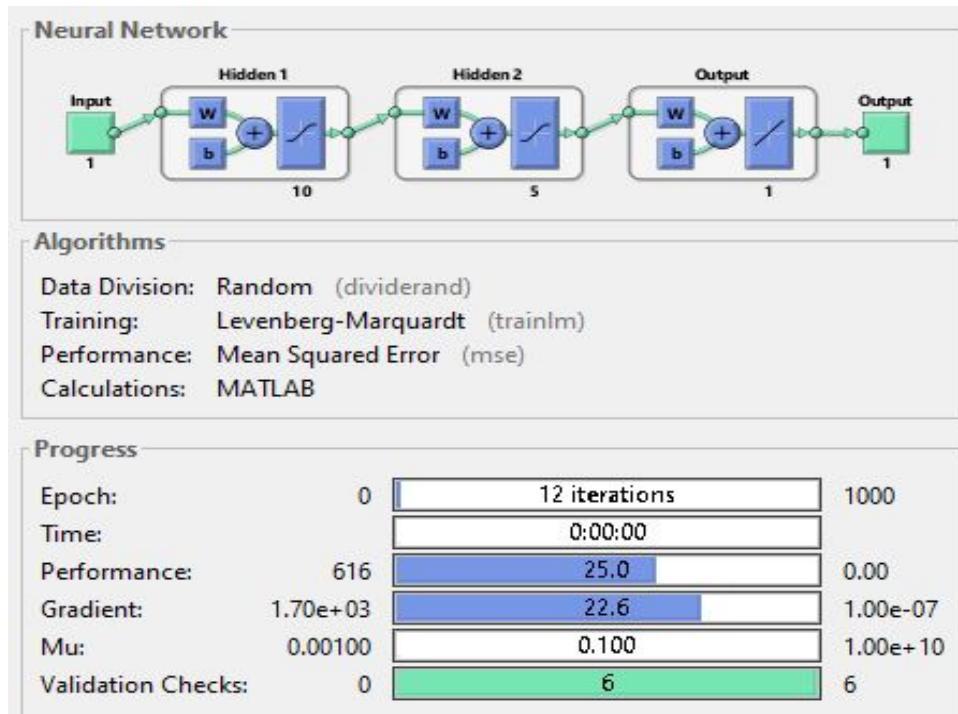


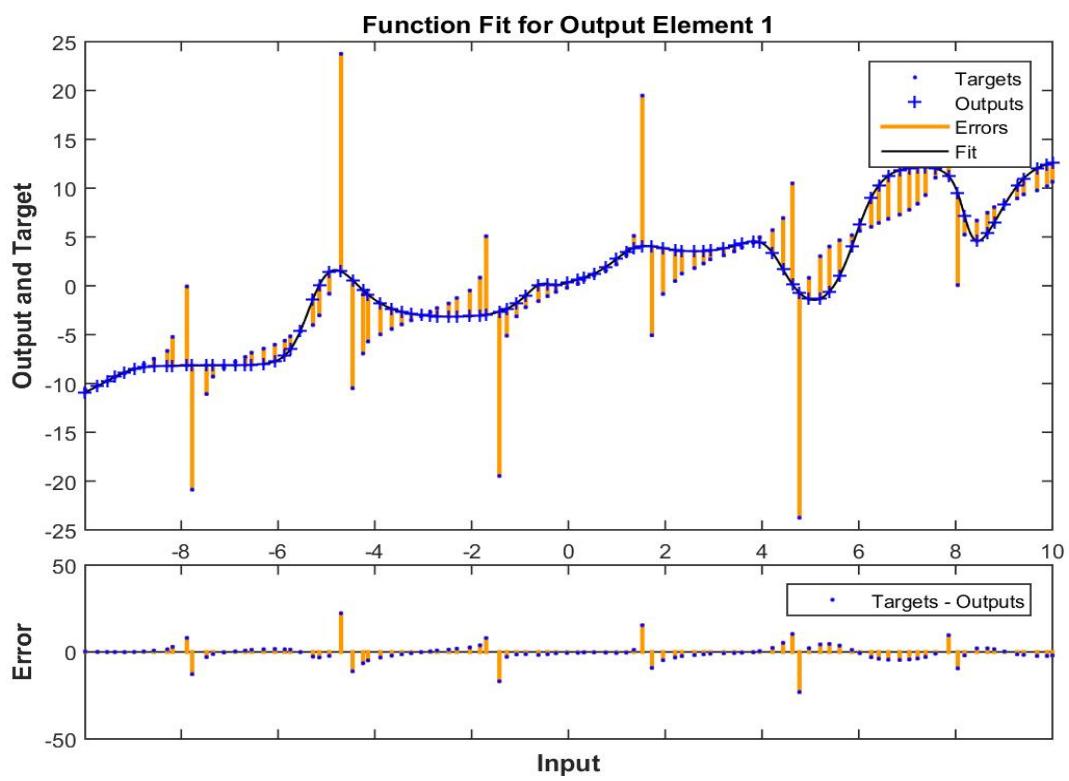
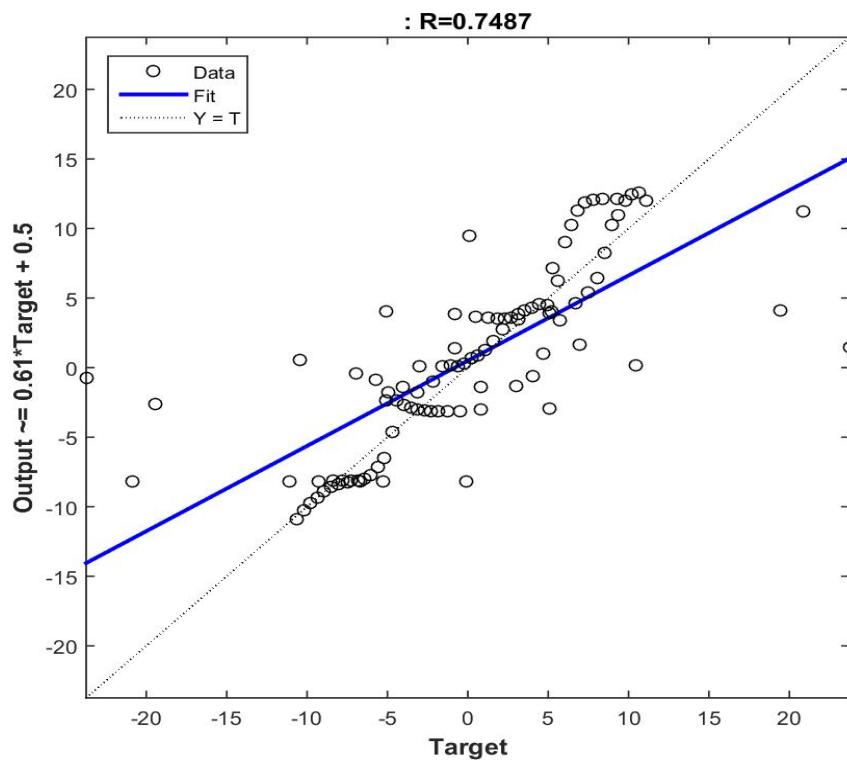


تابع چهارم:

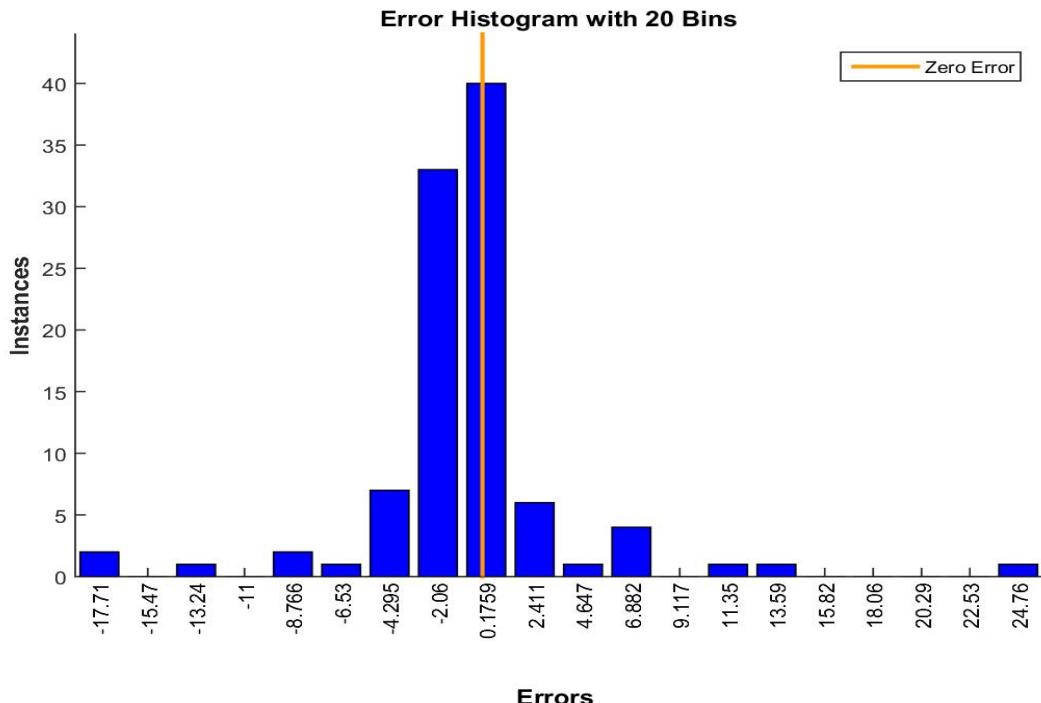
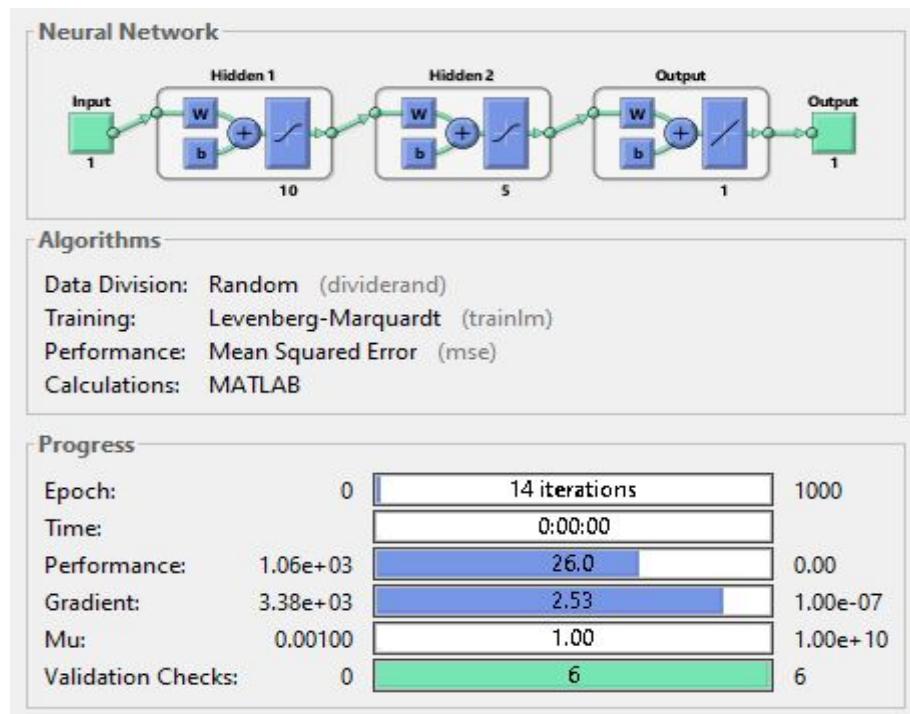
$$y_4 = \tan(x) + x;$$

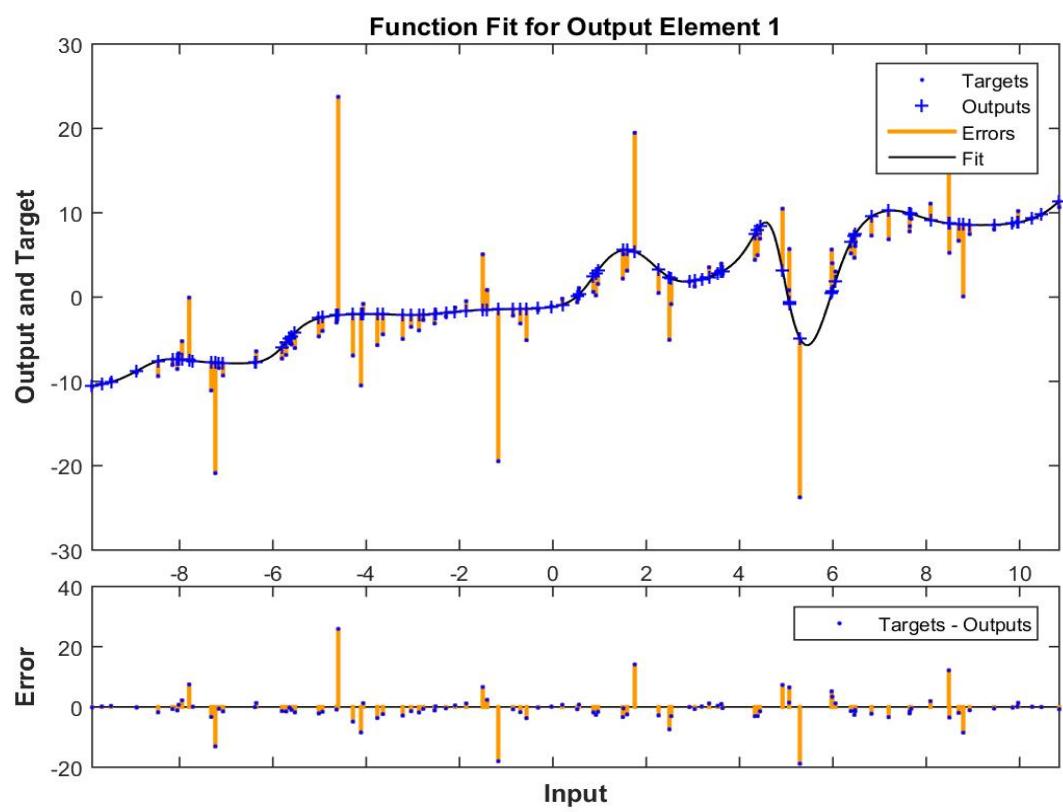
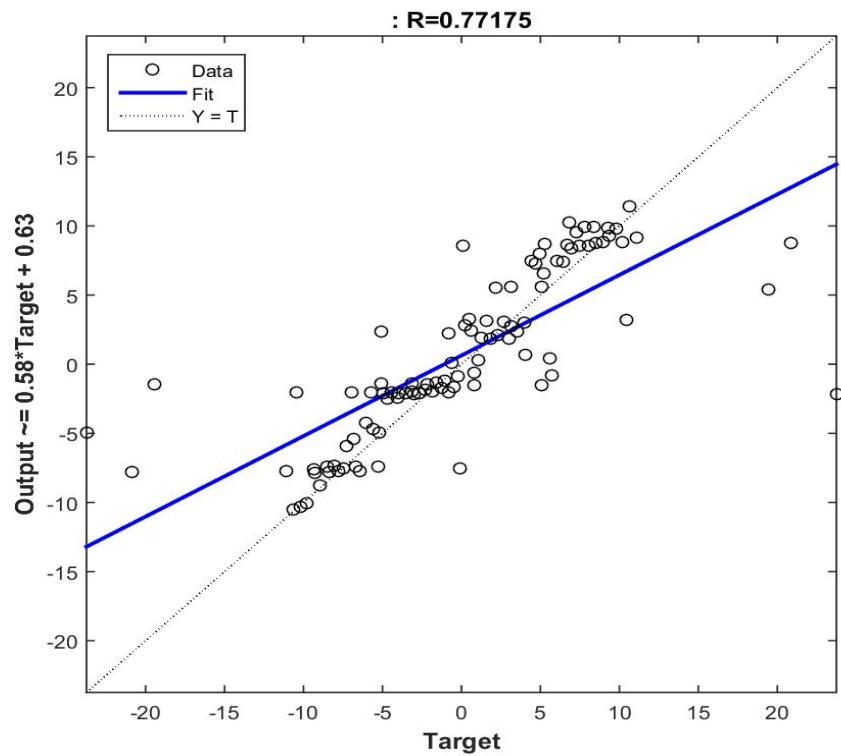
حالت اول: در حالت اول نویز با دامنه $[-10, 10]$ یعنی کمترین مقدار در نظر گرفته شده به تابع اضافه می‌شود.



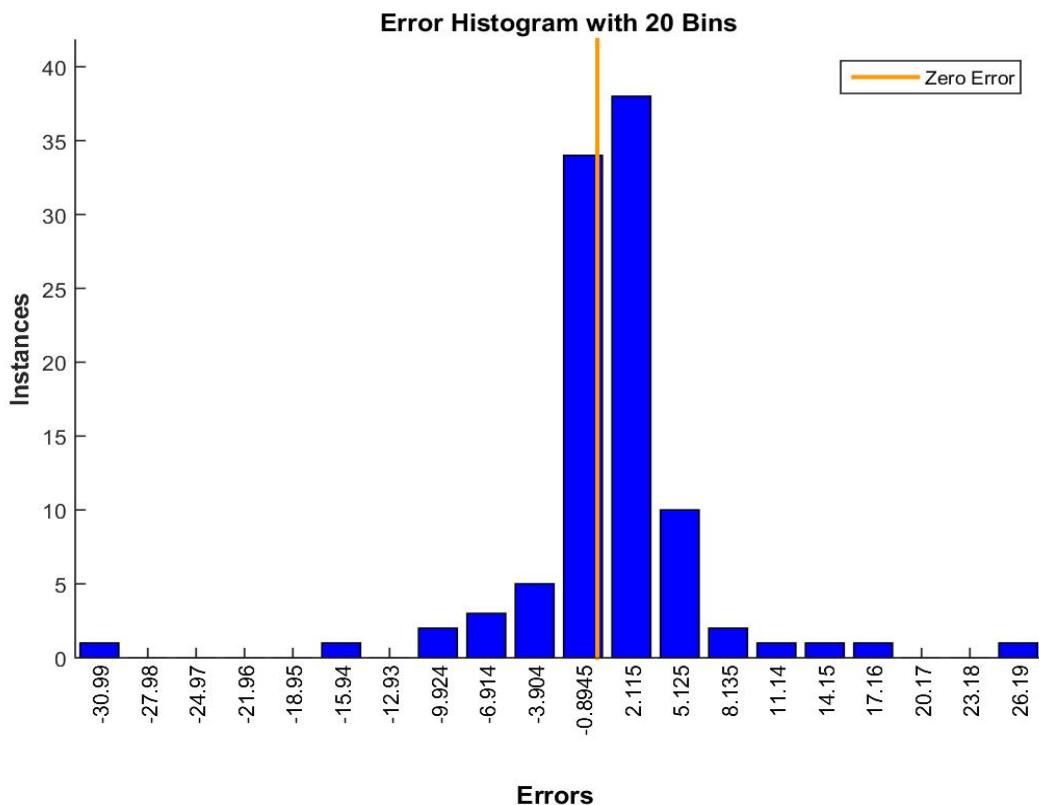
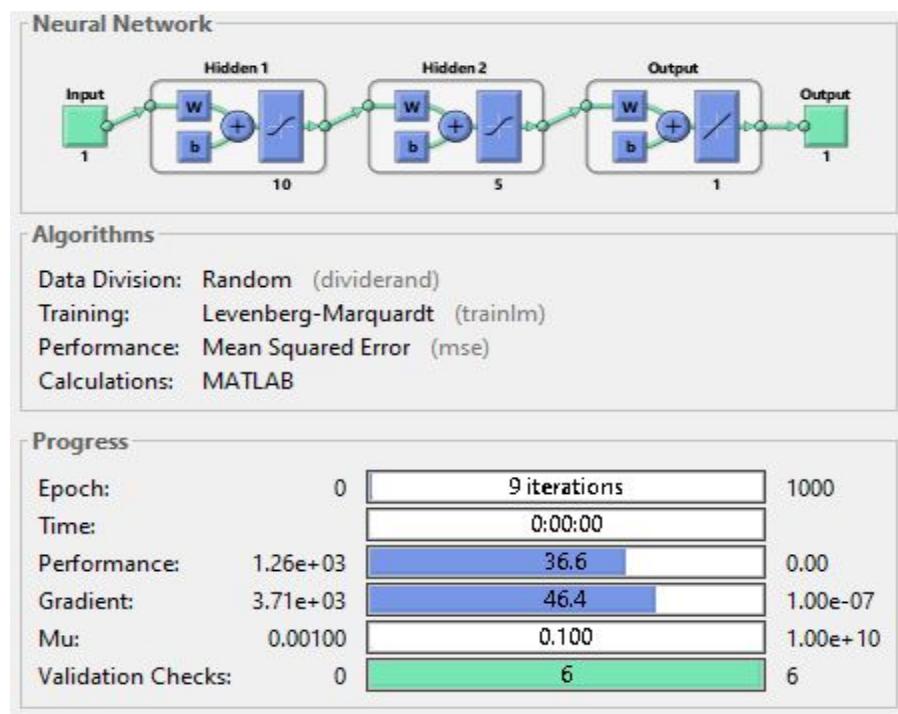


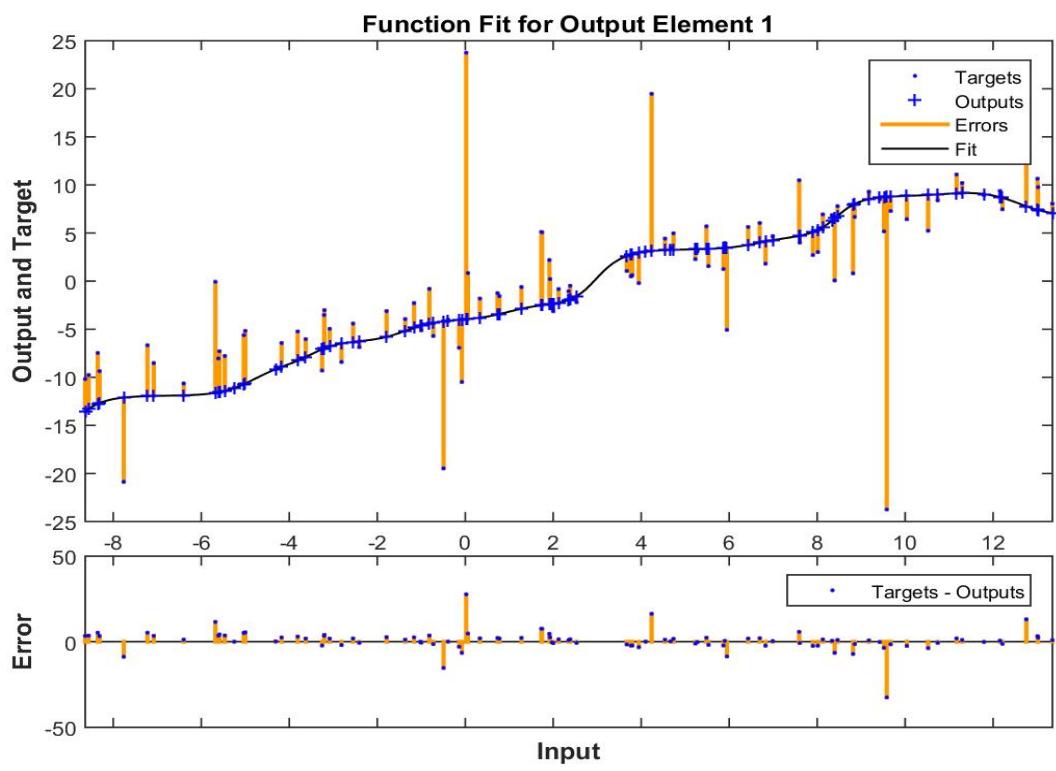
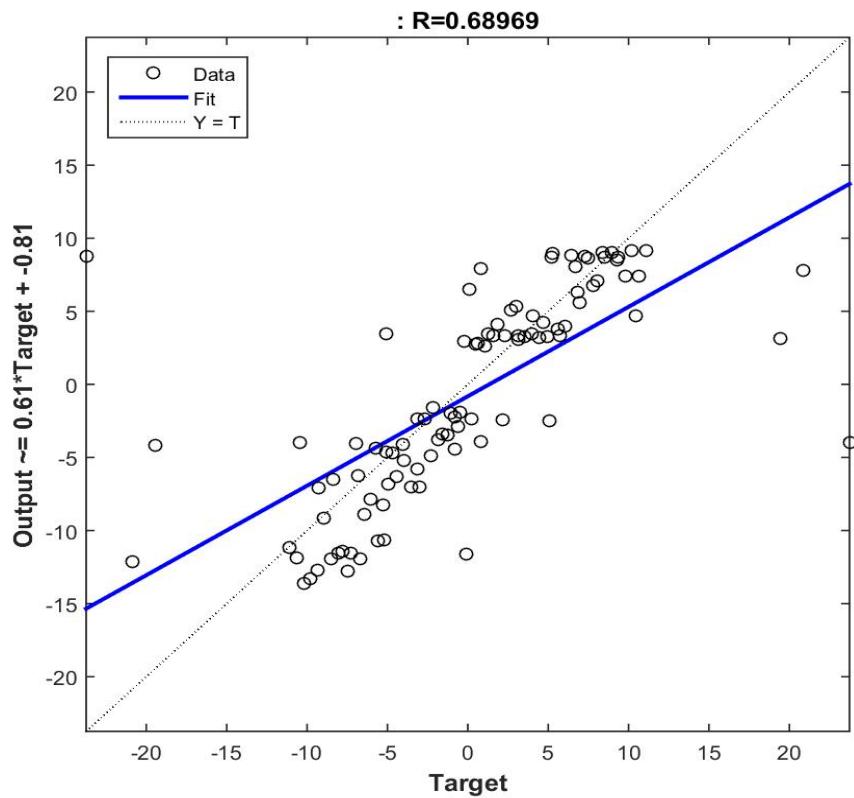
حالت دوم: در این حالت نویز با دامنه ۱ به سیگنال ورودی اضافه می‌شود.





حالت سوم: در این حالت نویز با دامنه ۵ به سیگنال ورودی اضافه می‌شود.

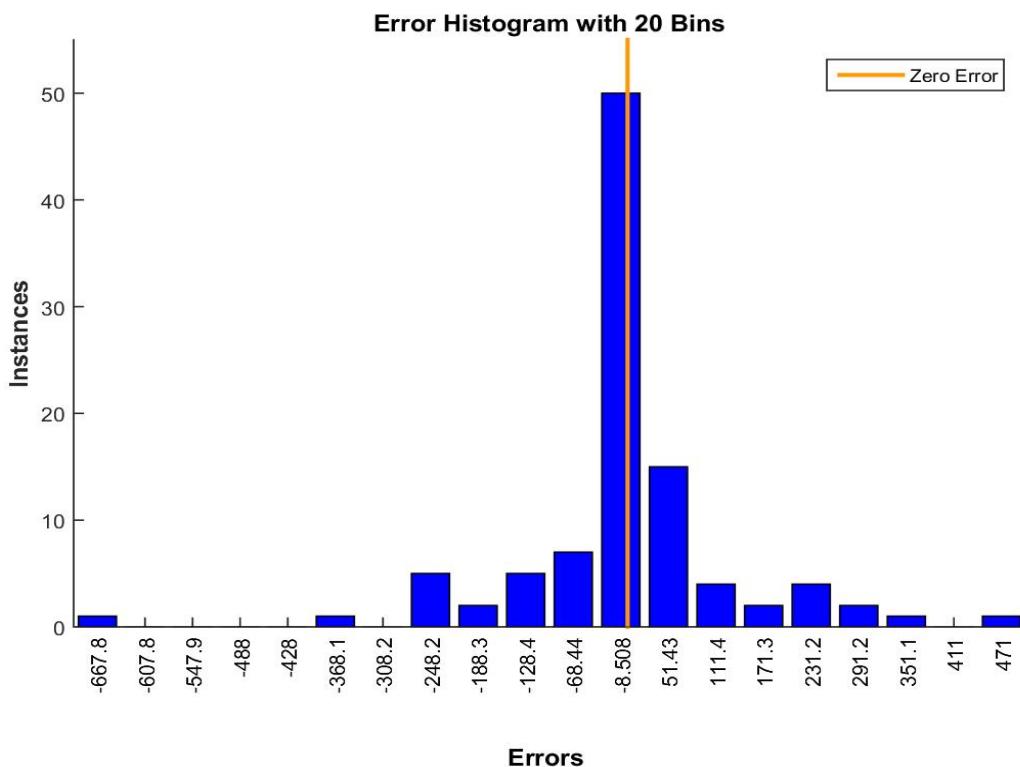
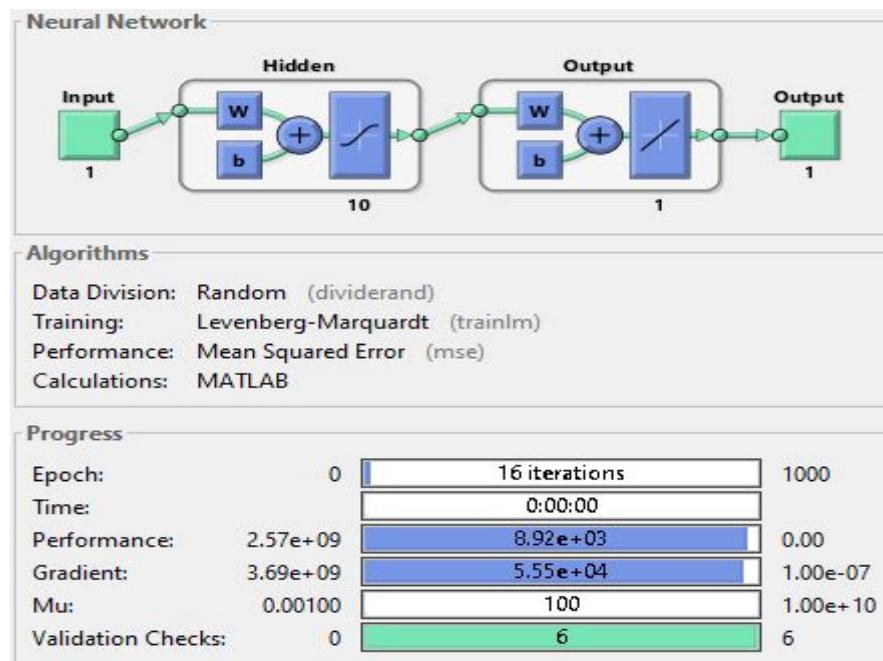


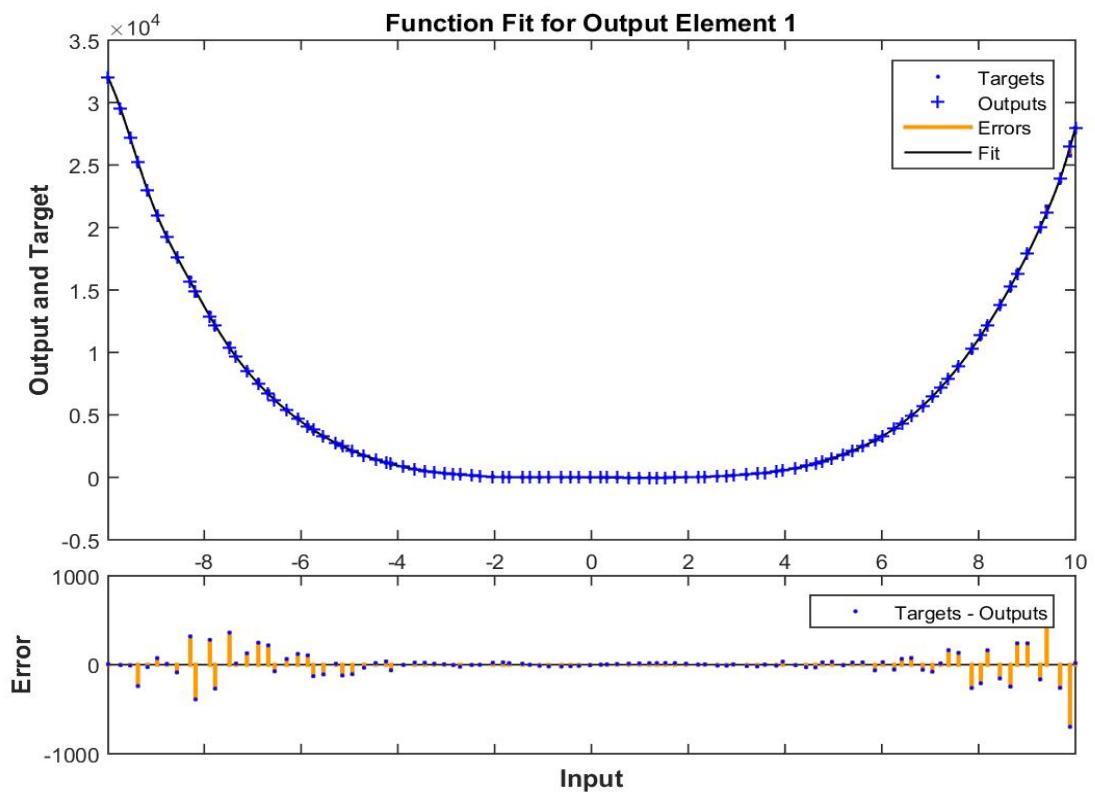
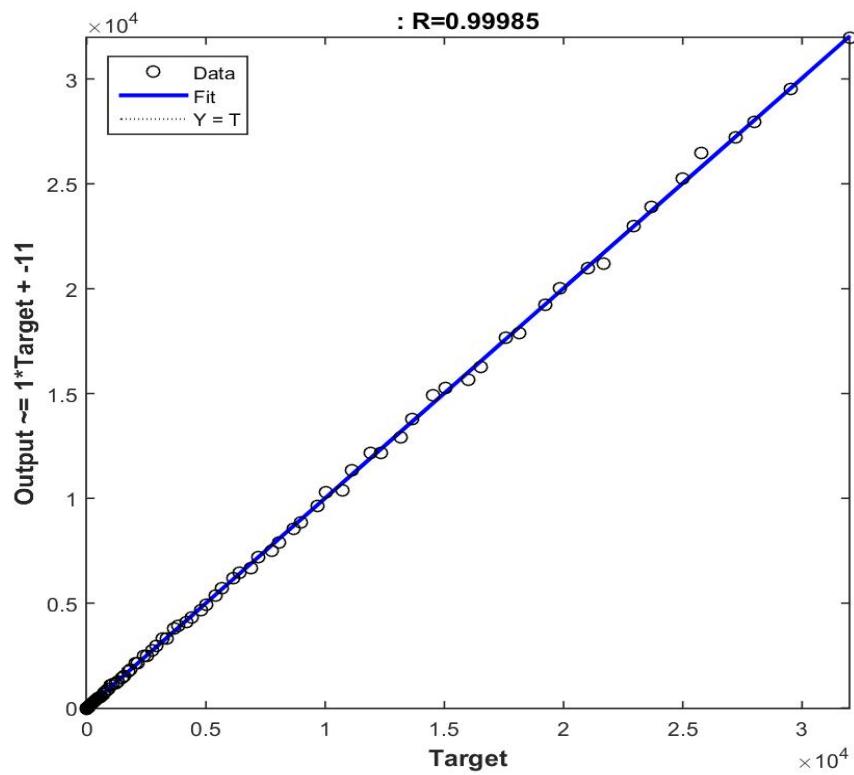


تابع پنجم:

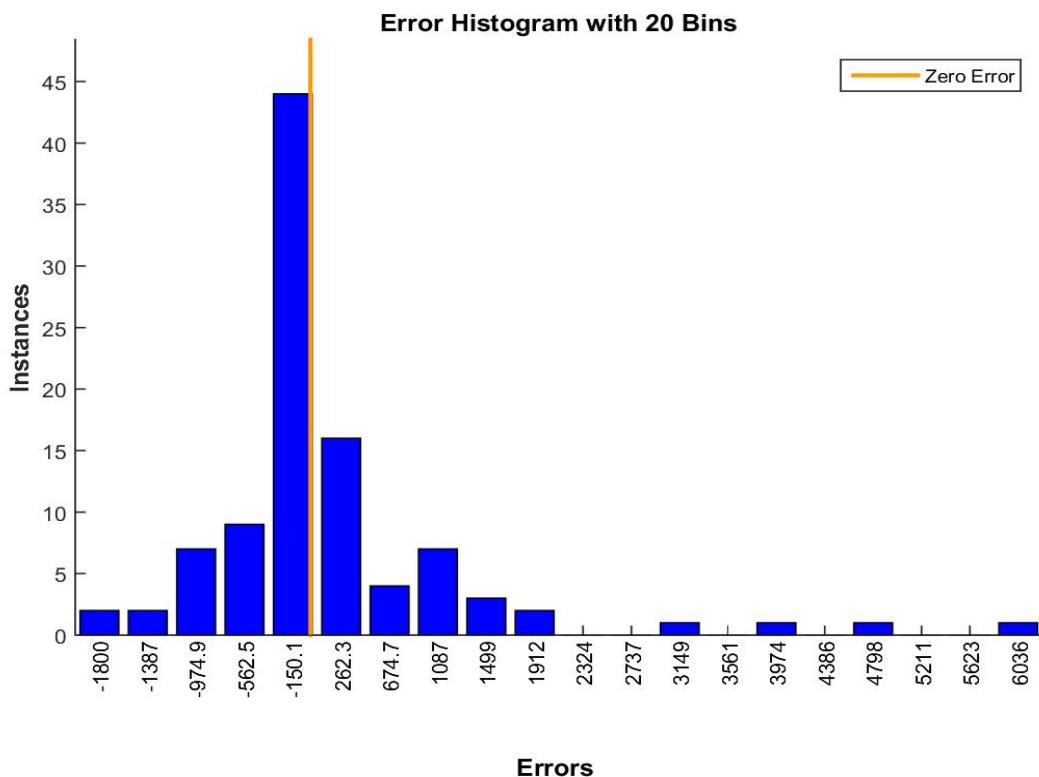
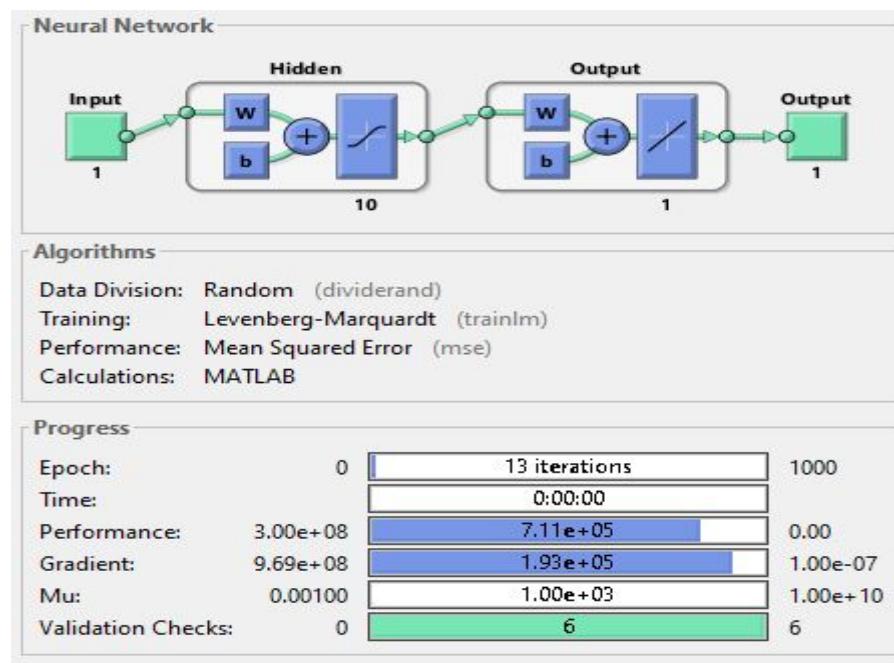
$$y_5 = 3x^4 - 2x^3 - 10;$$

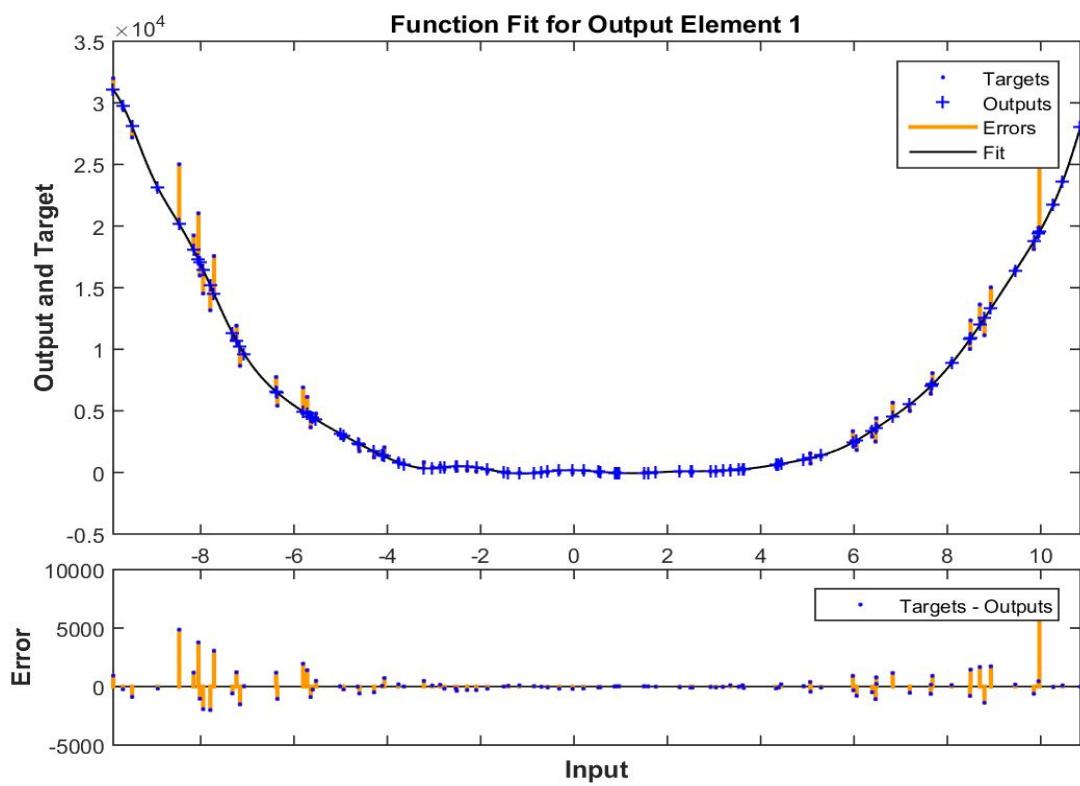
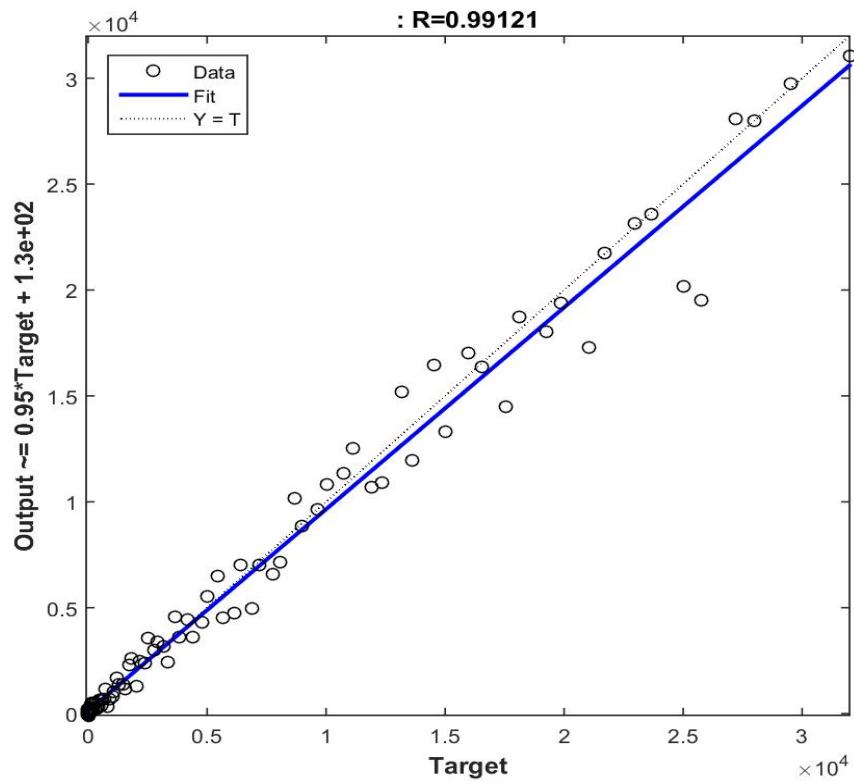
حالت اول: در حالت اول نویز با دامنه $[-1, 1]$ یعنی کمترین مقدار در نظر گرفته شده به تابع اضافه می‌شود.



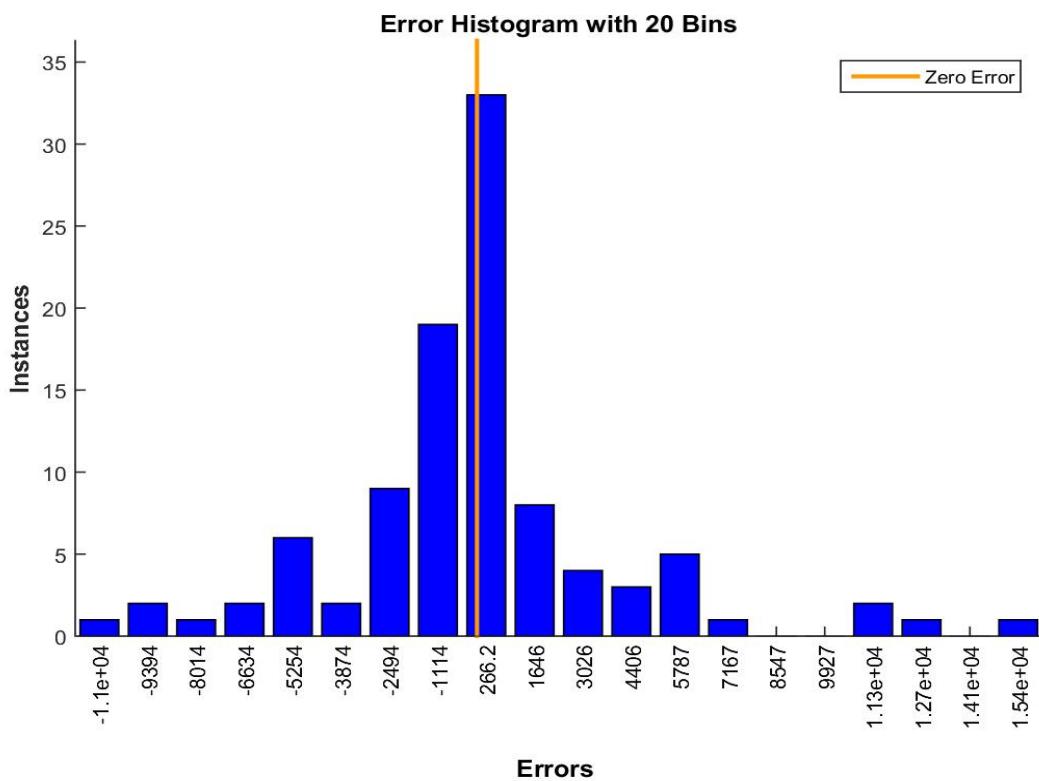
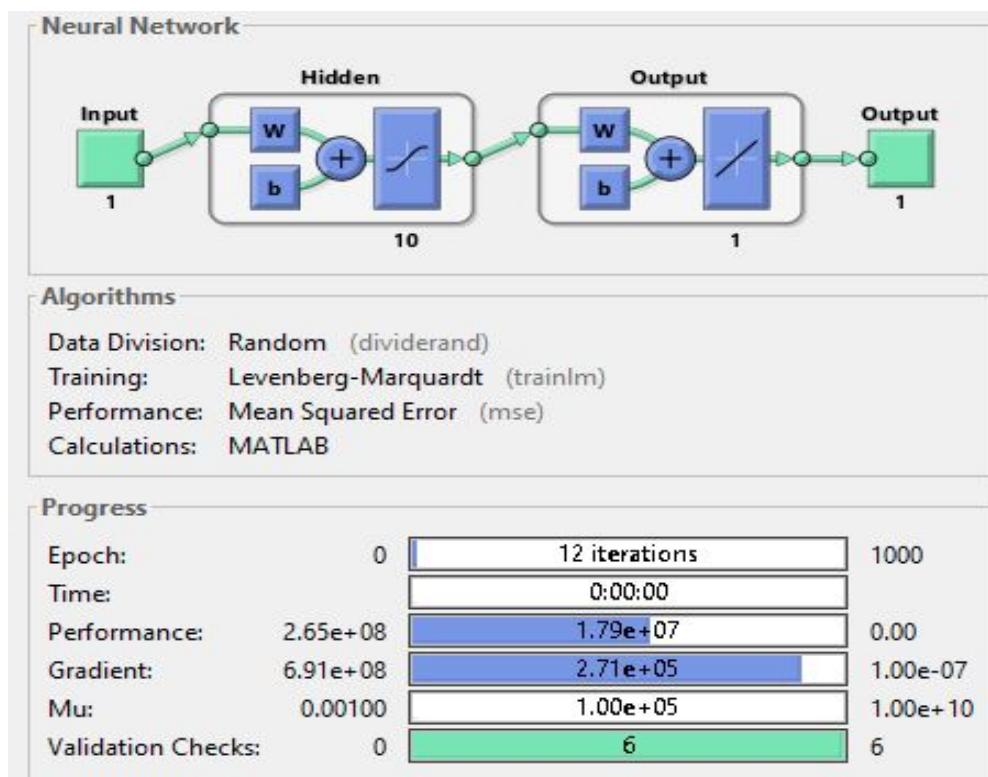


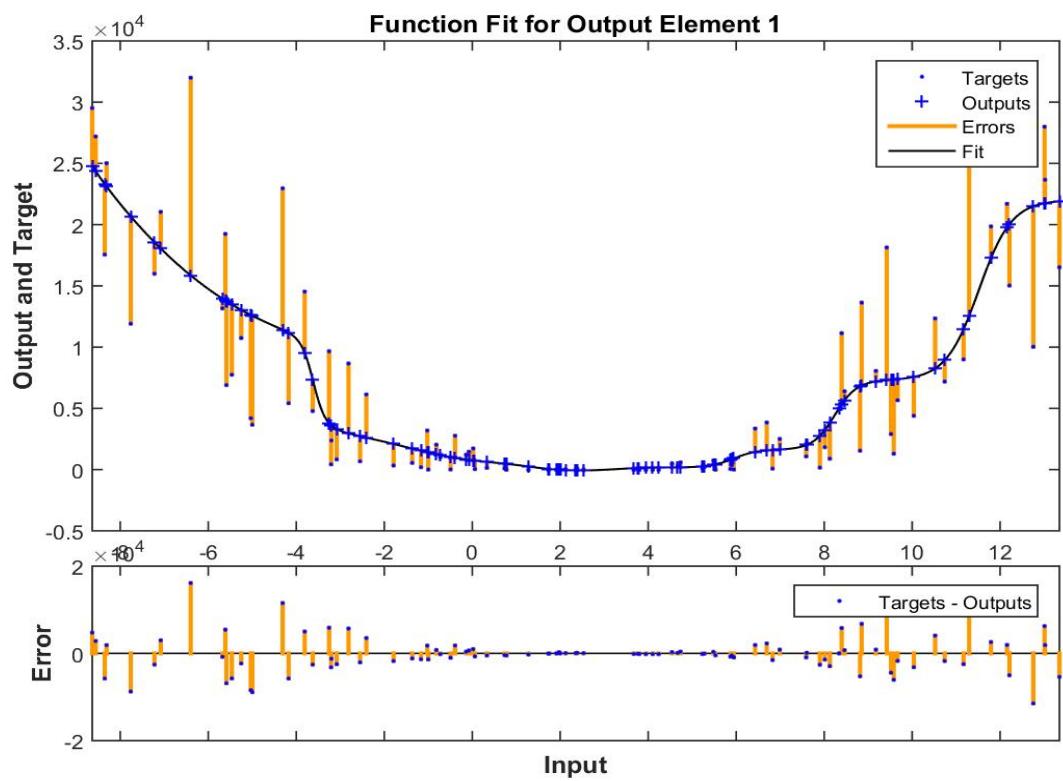
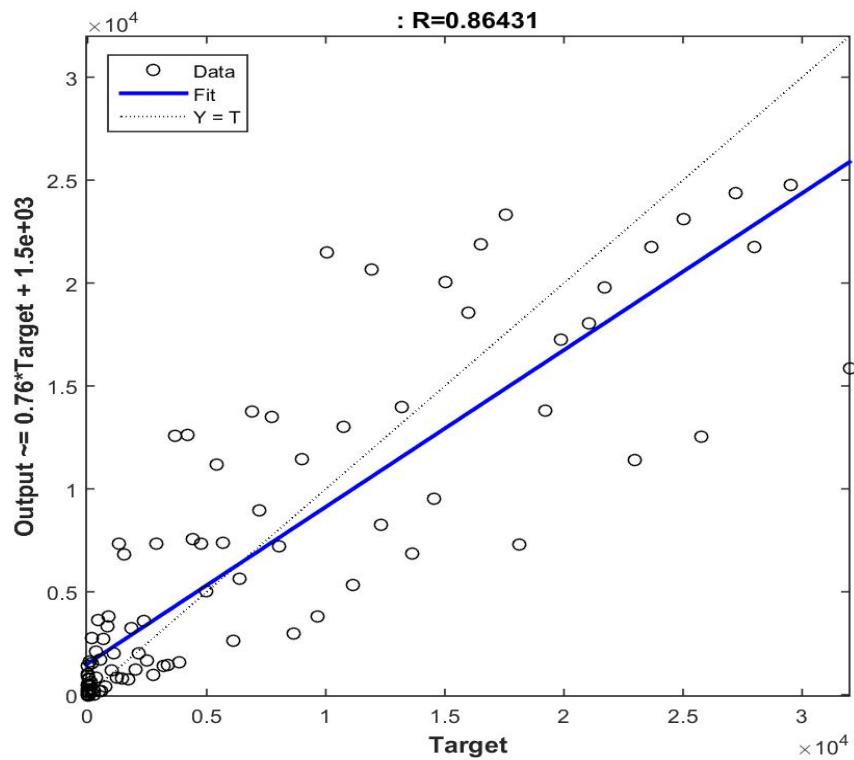
حالت دوم: در این حالت نویز با دامنه ۱ به سیگنال ورودی اضافه می‌شود.





حالت سوم: در این حالت نویز با دامنه ۵ به سیگنال ورودی اضافه می‌شود.

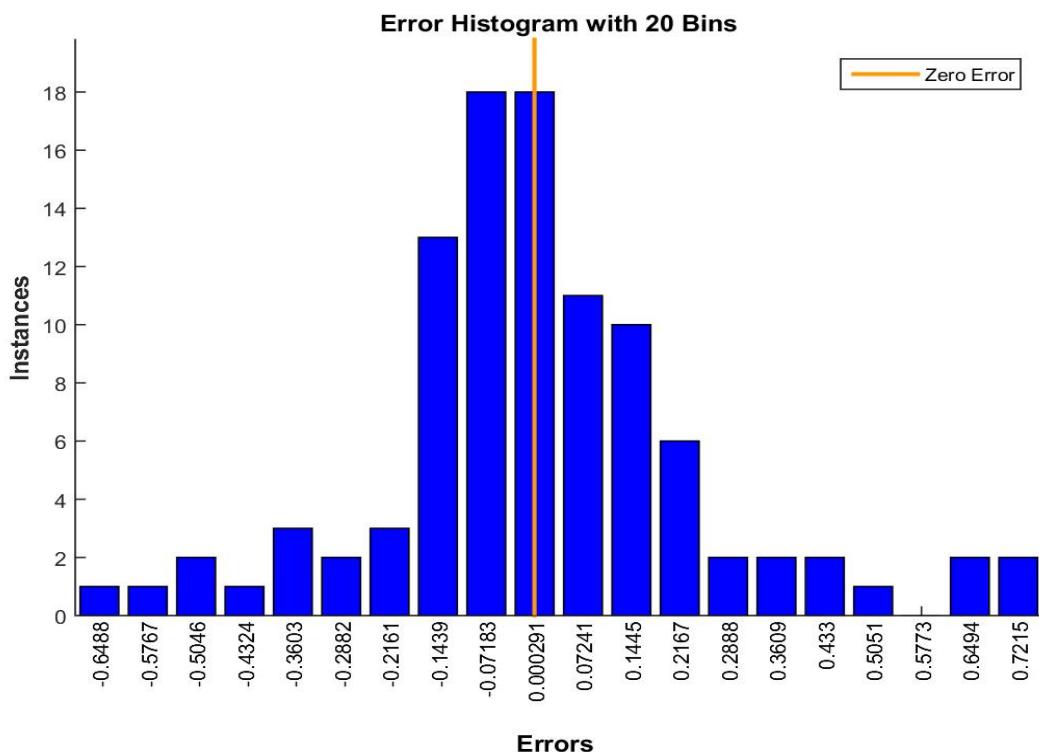
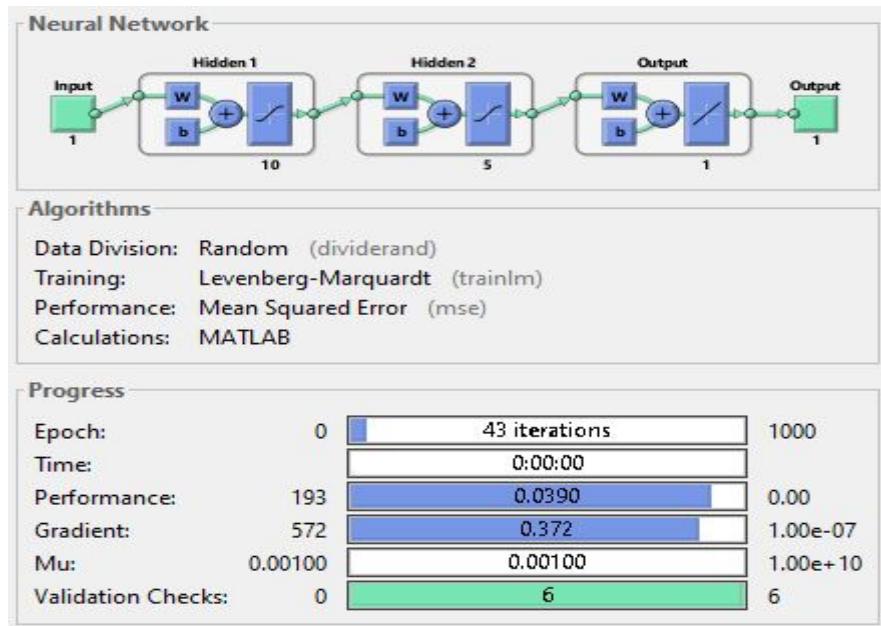


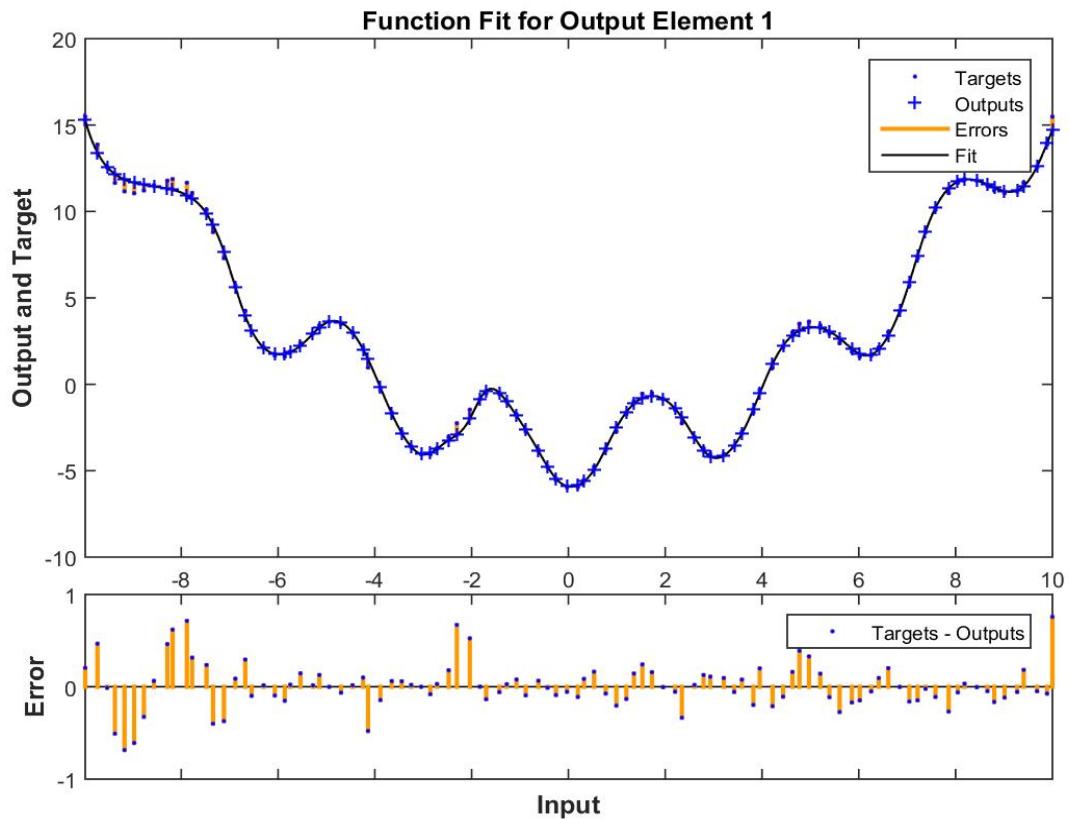
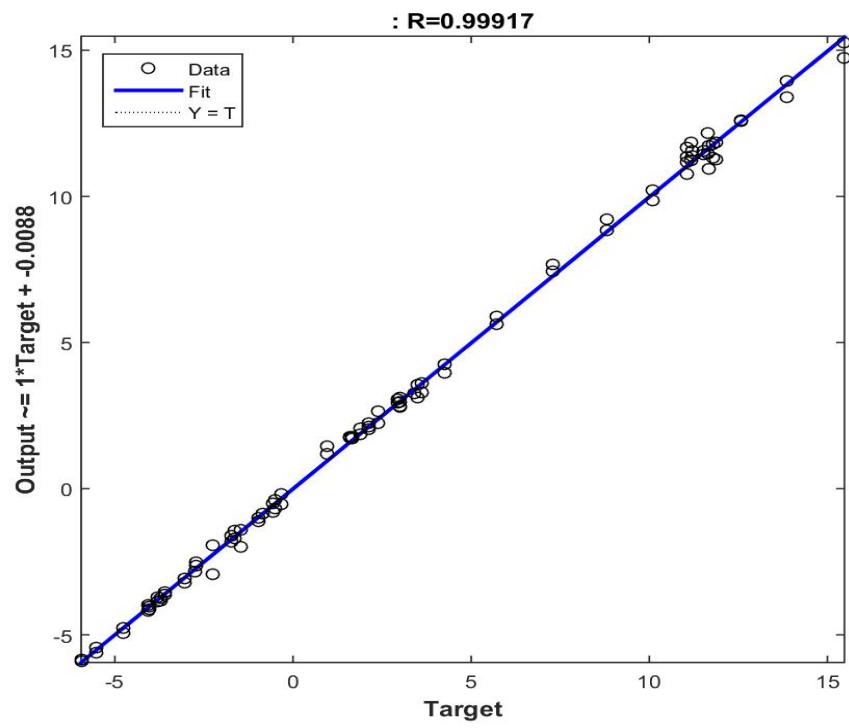


تابع ششم:

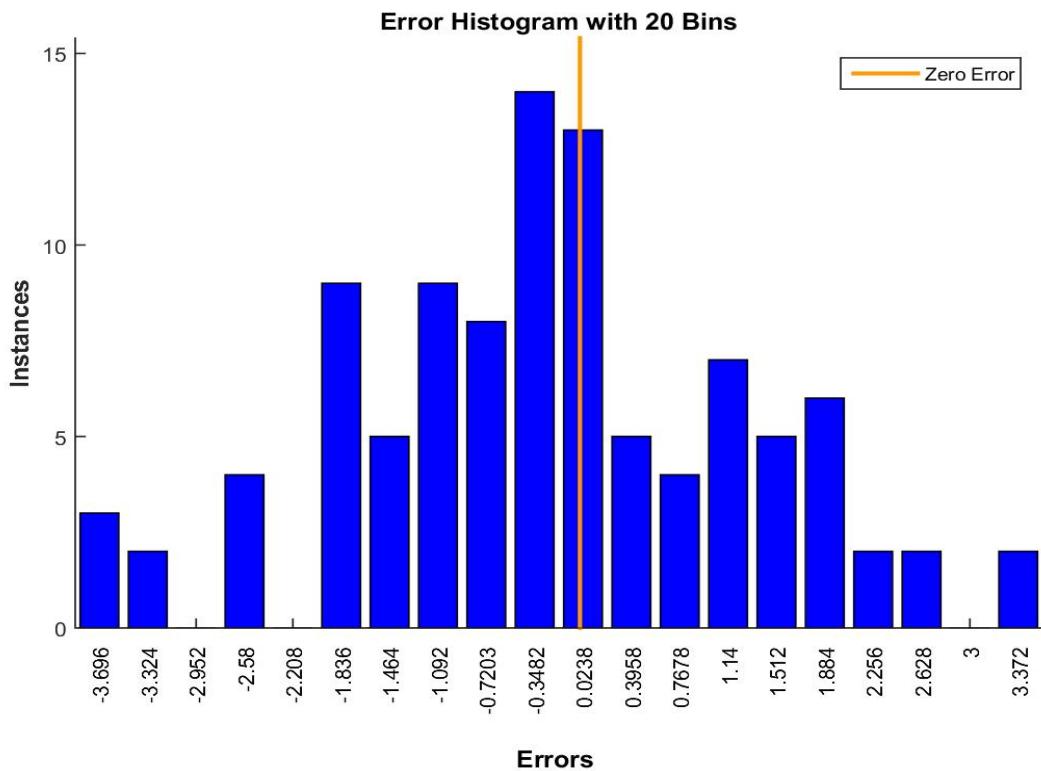
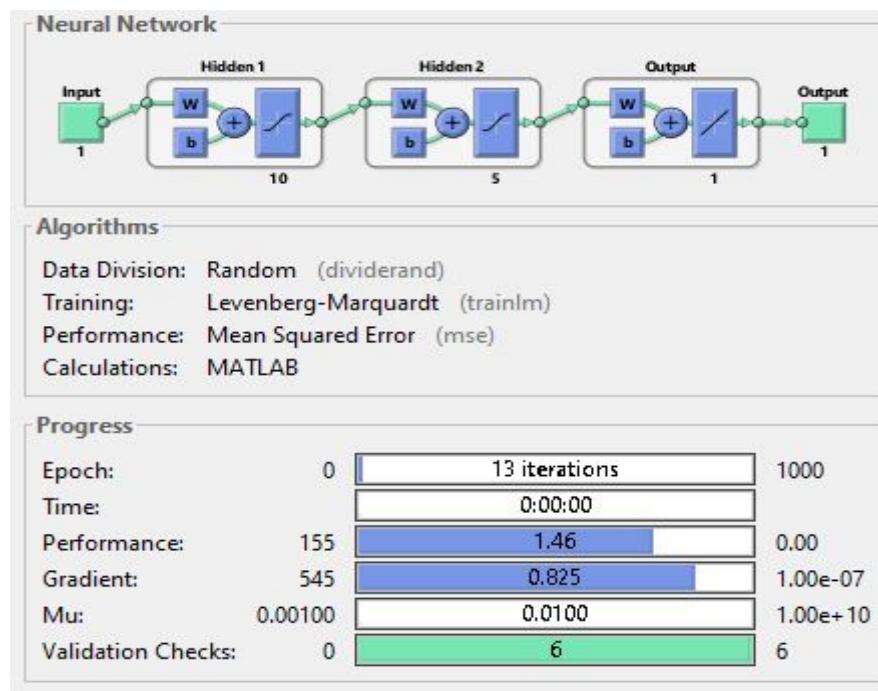
$$y_6 = 5 * \sin(x)^2 + 0.2 * x^2 - 6;$$

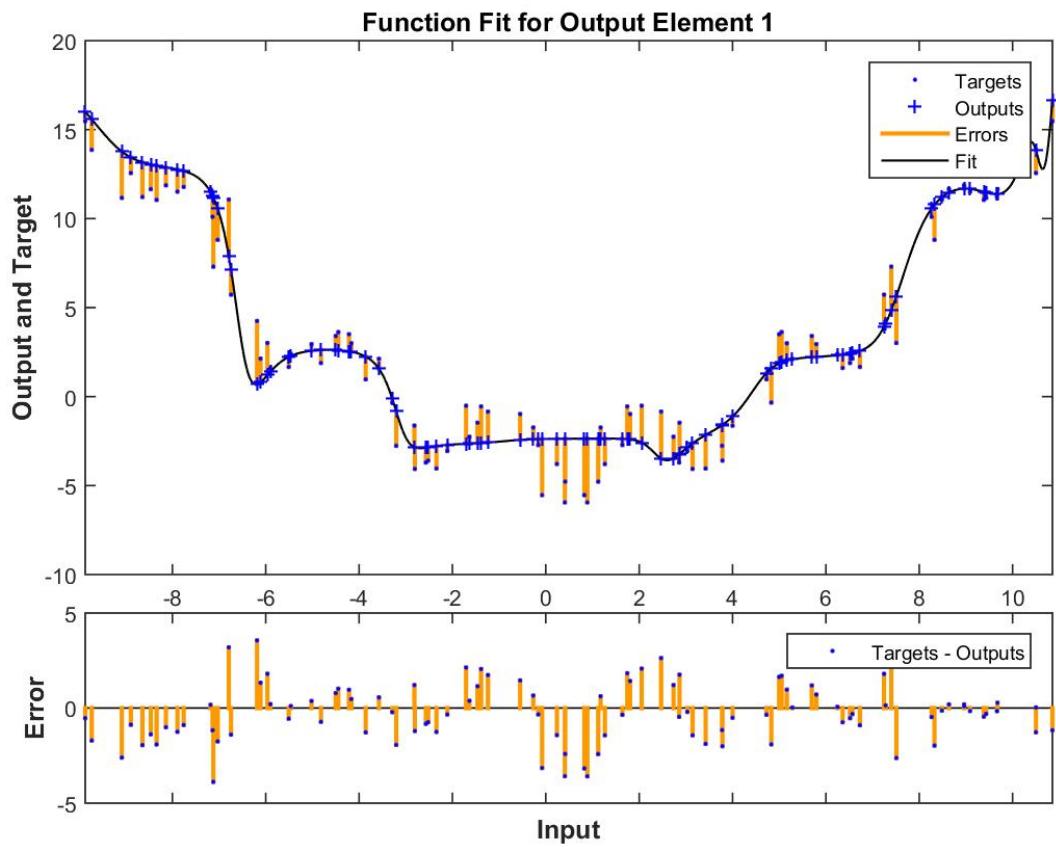
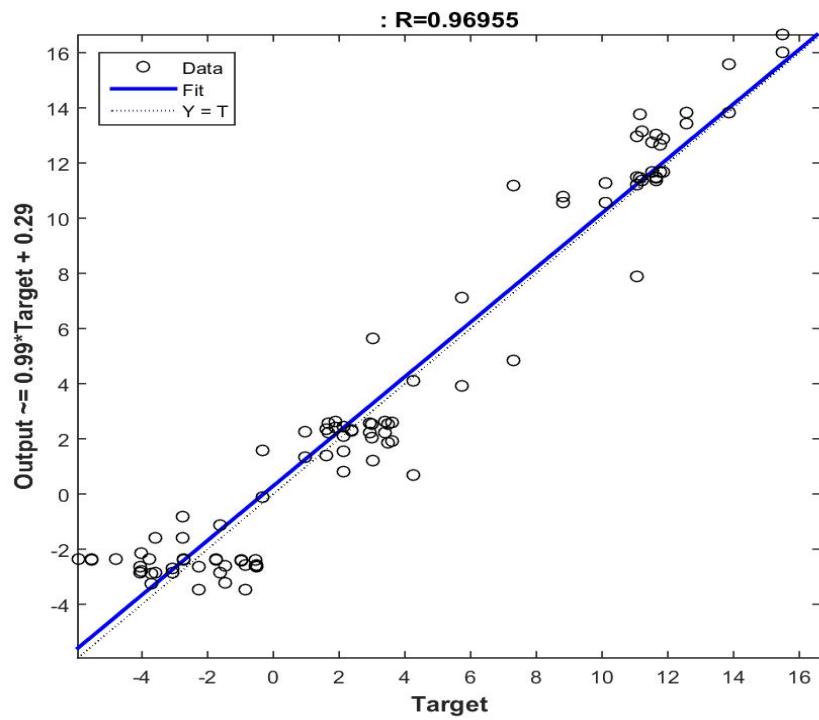
حالت اول: در حالت اول نویز با دامنه $[0, 1]$ یعنی کمترین مقدار در نظر گرفته شده به تابع اضافه می‌شود.



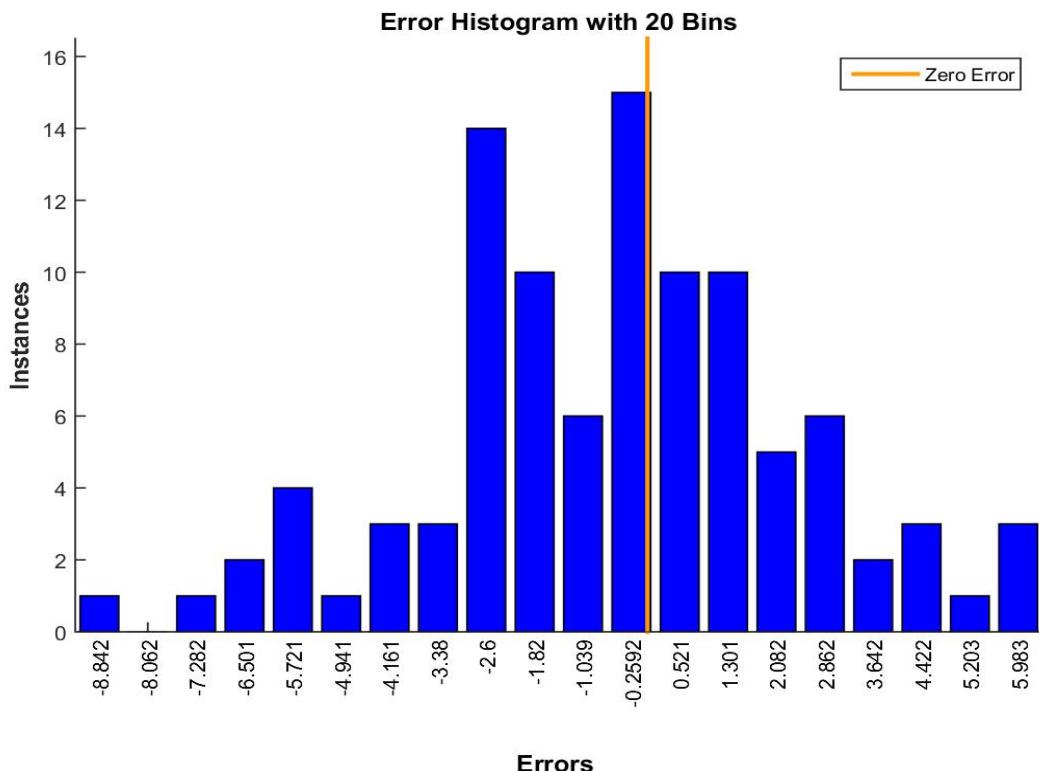
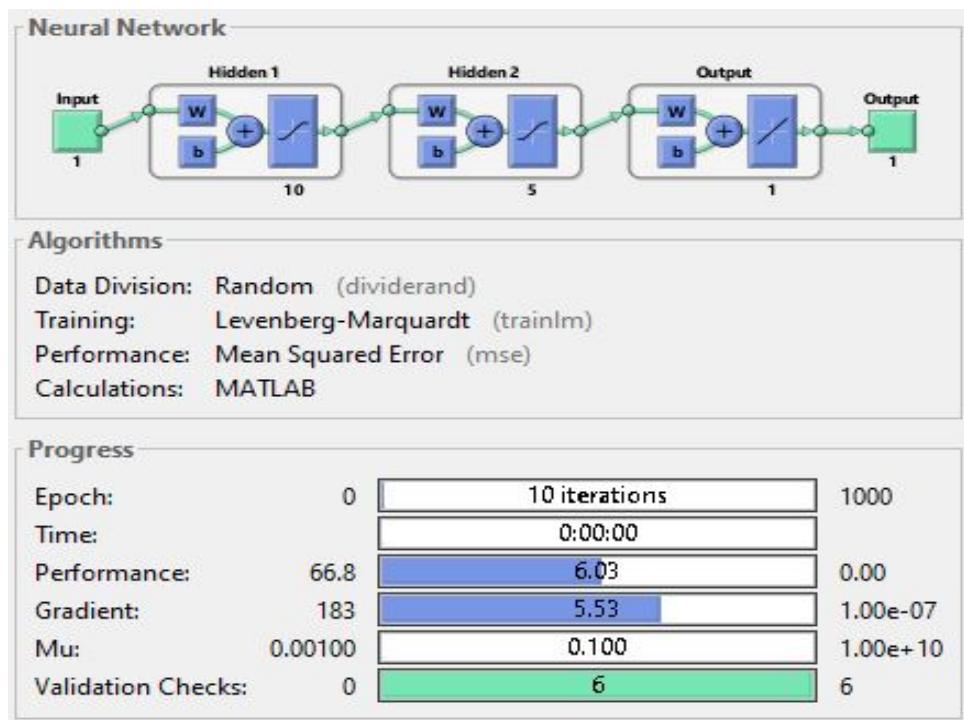


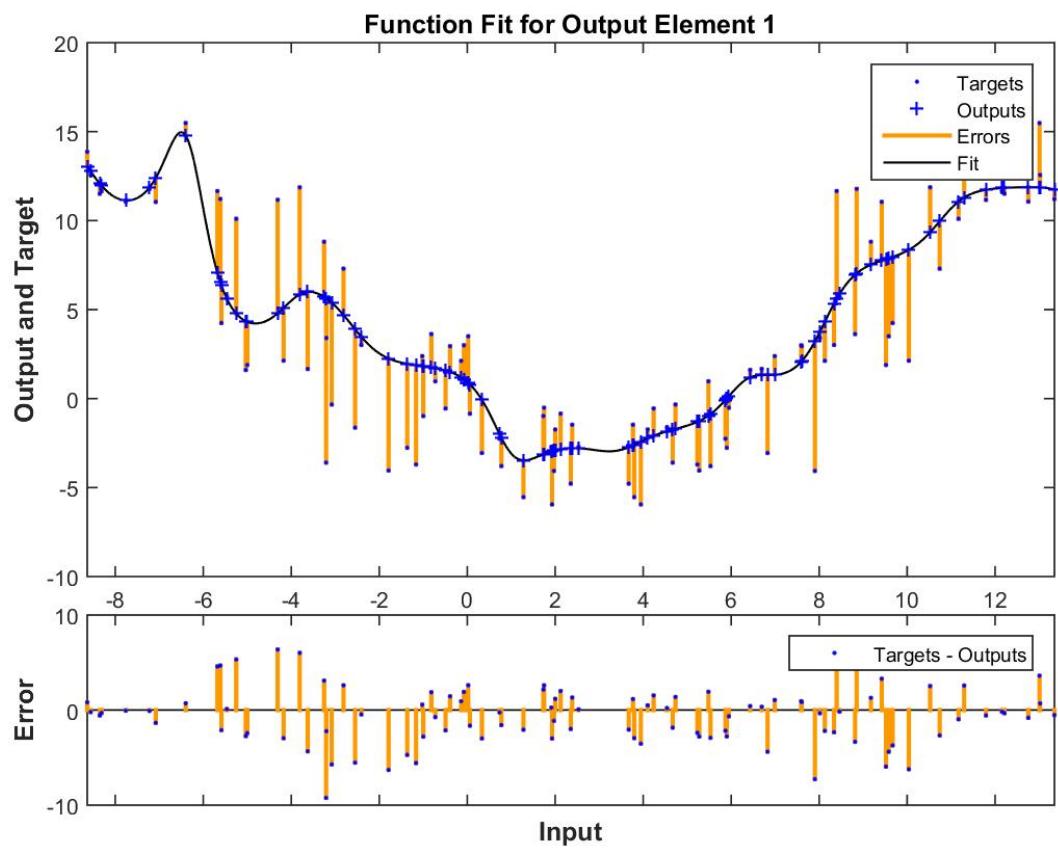
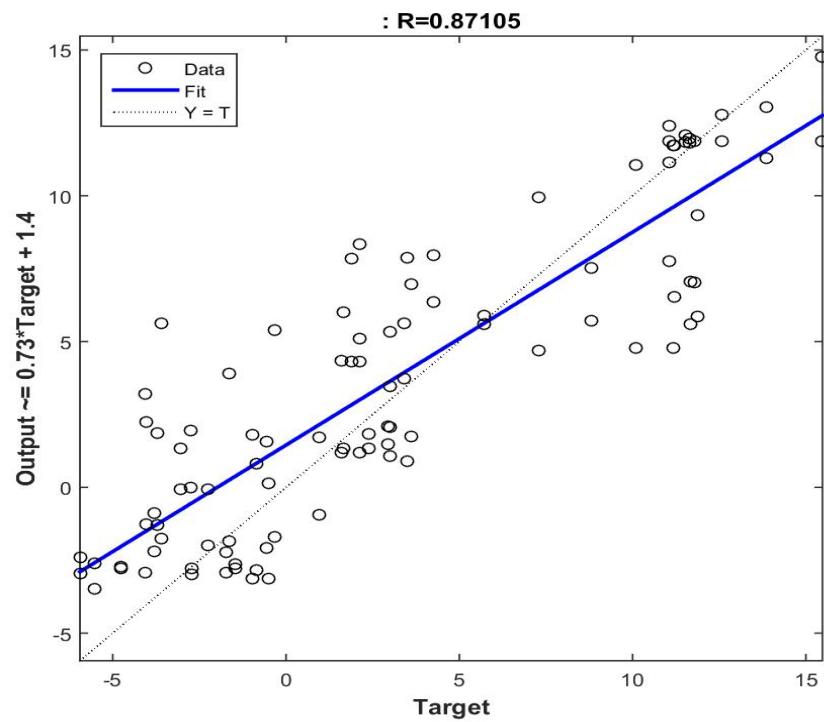
حالت دوم: در این حالت نویز با دامنه ۱ به سیگنال ورودی اضافه می‌شود.





حالت سوم: در این حالت نویز با دامنه ۱ به سیگنال ورودی اضافه می‌شود.





نتیجه‌گیری:

با توجه به نتایج بدست آمده از این بخش و مقایسه آن با تابع اصلی و نتایج بخش اول مشاهده می‌شود در حالت اول که نویز با دامنه ۰.۱ اضافه می‌شود، تاثیر چندانی در نتایج نخواهد داشت و خروجی شبکه عصبی همانند نتایج بخش ۱ و تابع اصلی می‌شود. یعنی نمودار خروجی شبکه عصبی همانند نتیجه اصلی بخش اول است که نویز‌ها تنها موجب می‌شود فاصله‌هایی اطراف نمودار خروجی و نمونه‌های آموزش ایجاد شود که این فاصله‌ها همان نویز‌های ما هستند. بنابراین می‌توان گفت شبکه عصبی در نویز‌های با دامنه کم قابلیت حذف آن را دارد. اما وقتی نویز افزایش می‌یابد، نمودار خروجی شبکه عصبی با نمودار اصلی تابع خیلی متفاوت خواهد بود و همچنین خطای شبکه عصبی نسبت به ورودی‌های با نویز نیز زیاد خواهد بود. چون نویز‌ها مقادیر تصادفی هستند و رابطه منطقی بین آنها نیست که شبکه عصبی بتواند آن‌ها را تشخیص بدهد.

بخش سوم:

در این بخش ۶ تابع برای آموزش در شبکه عصبی در نظر گرفته شده است. توابع دارای سه ورودی x_1 , x_2 و x_3 هستند که در هر یک در بازه ۱۰-تا ۱۰ بوده و تعداد ورودی‌ها ۱۰۰ در نظر گرفته شده است.

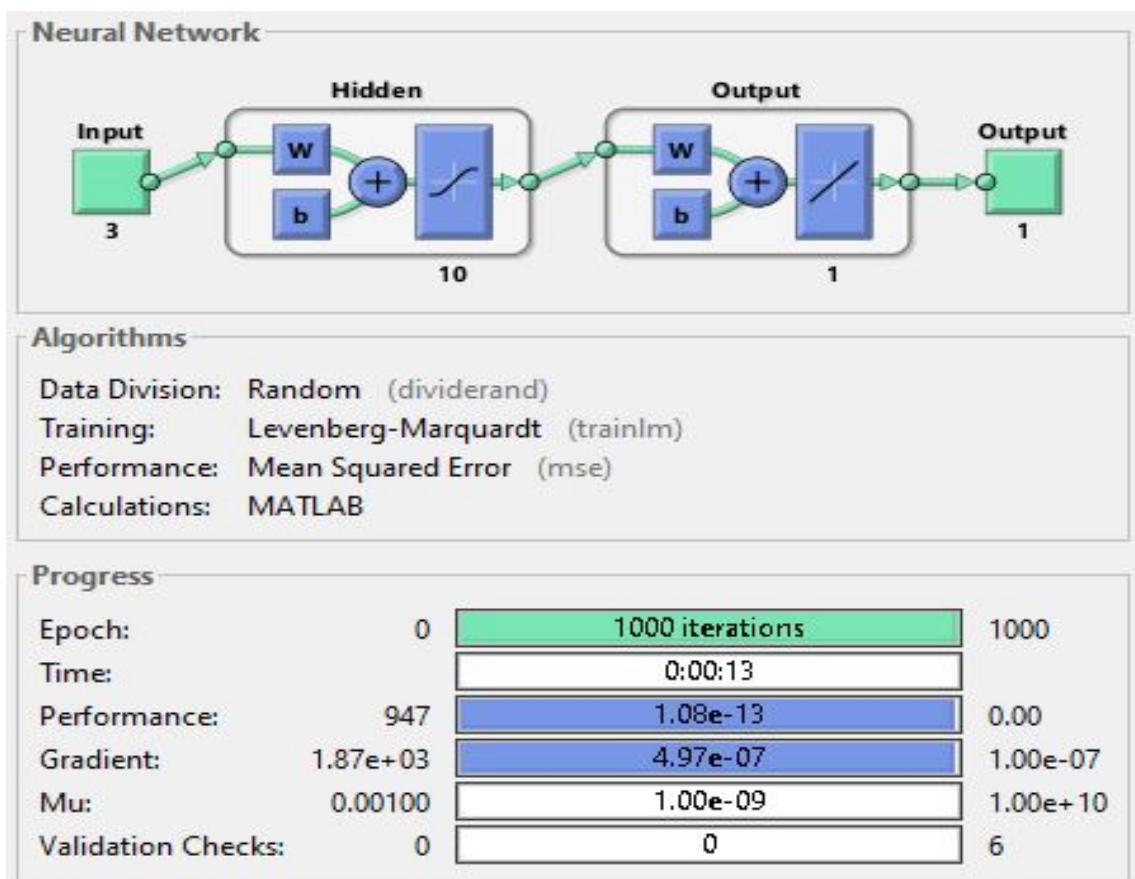
در دو حالت ابتدایی سعی شده است ساختار شبکه مناسب انتخاب شود و در حالت سوم و چهارم به ترتیب آموزش با ۲۰ و ۵۰۰ نمونه انجام شده است.

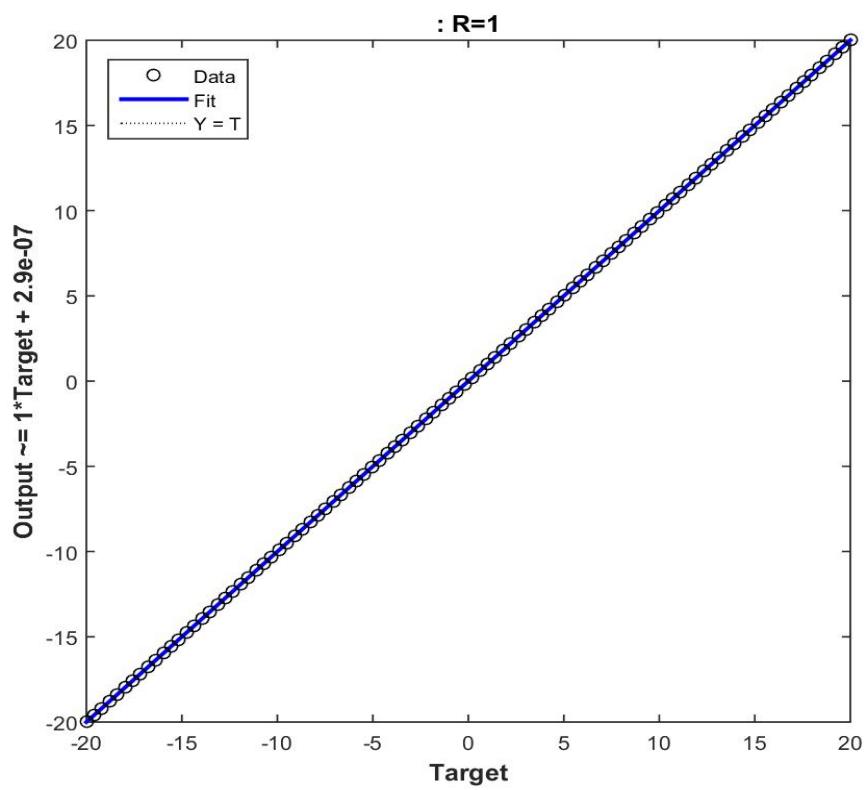
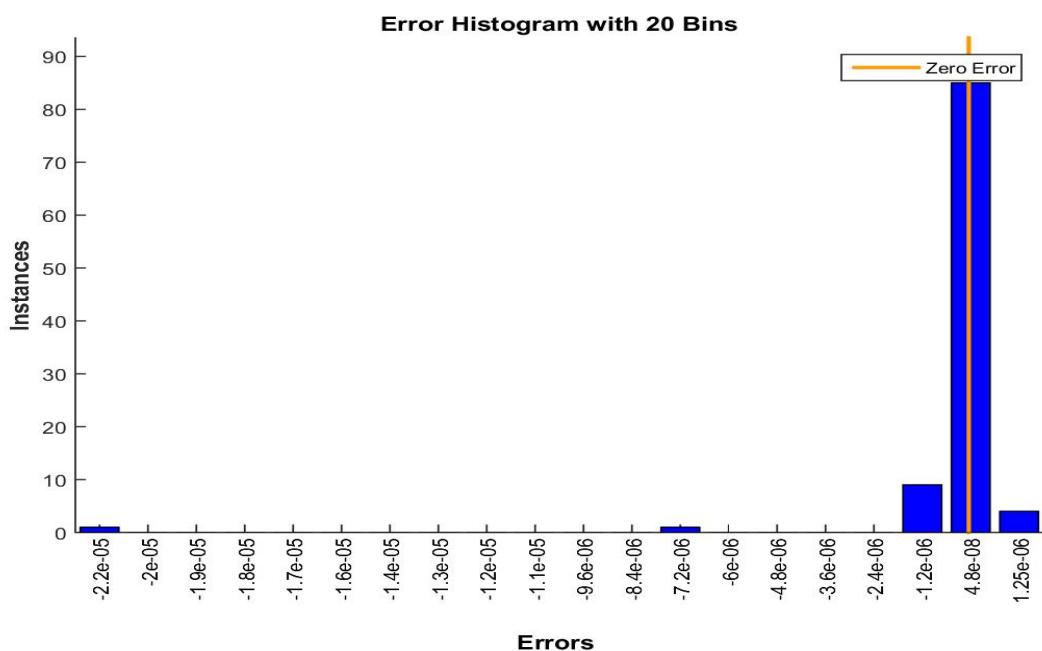
در این بخش هم از تابع اول تا ششم، پیچیدگی توابع بیشتر می‌شوند. همانطور که دیده می‌شود در ساختار شبکه عصبی ۳ ورودی و یک خروجی وجود خواهد داشت.

تابع اول:

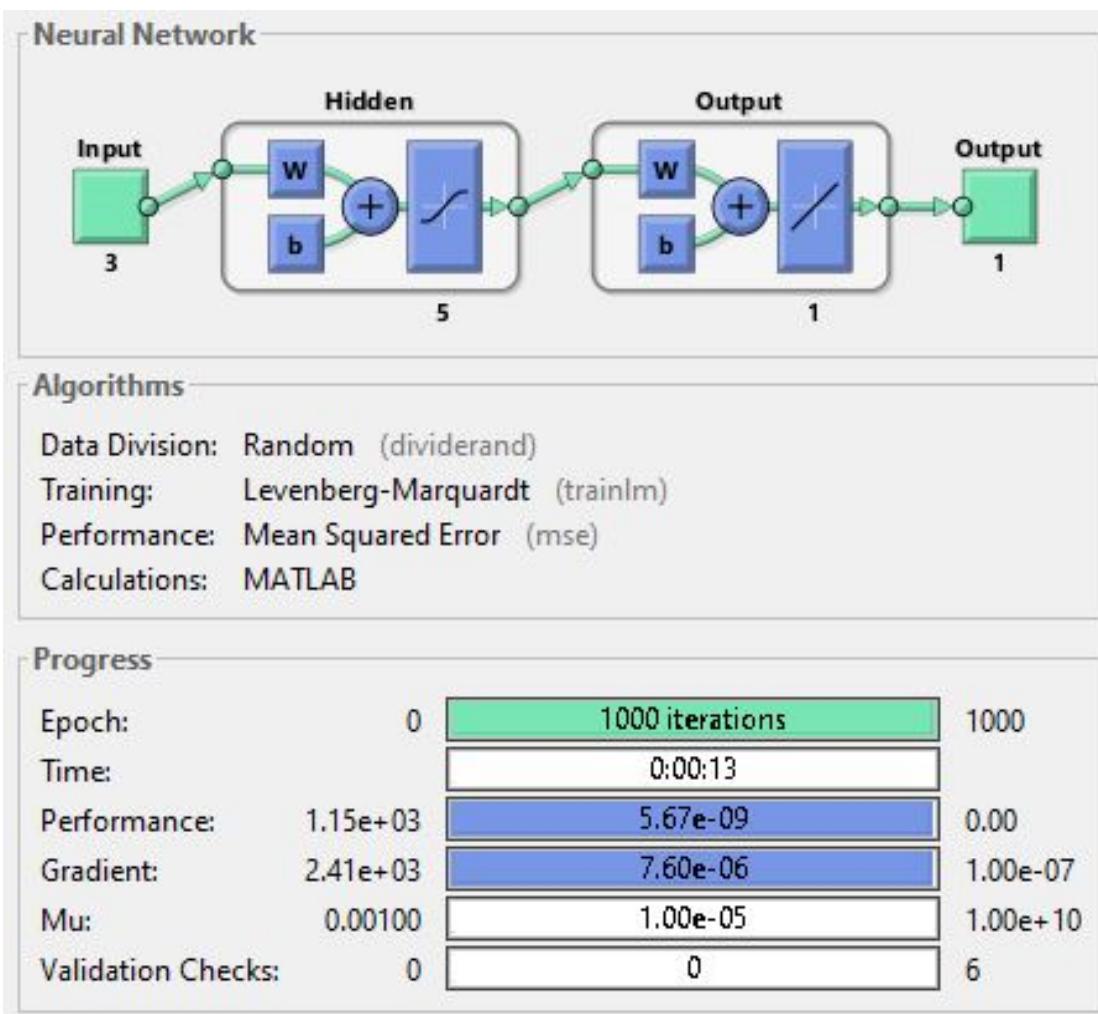
$$y_1 = 2 * x_1 + x_2 - x_3;$$

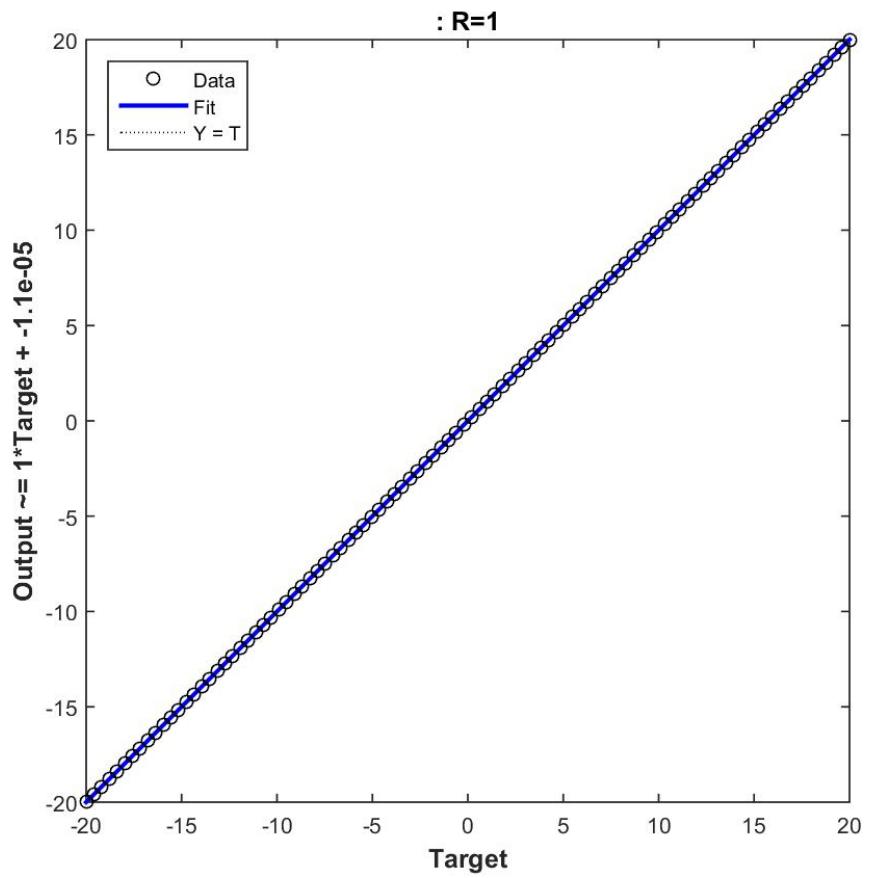
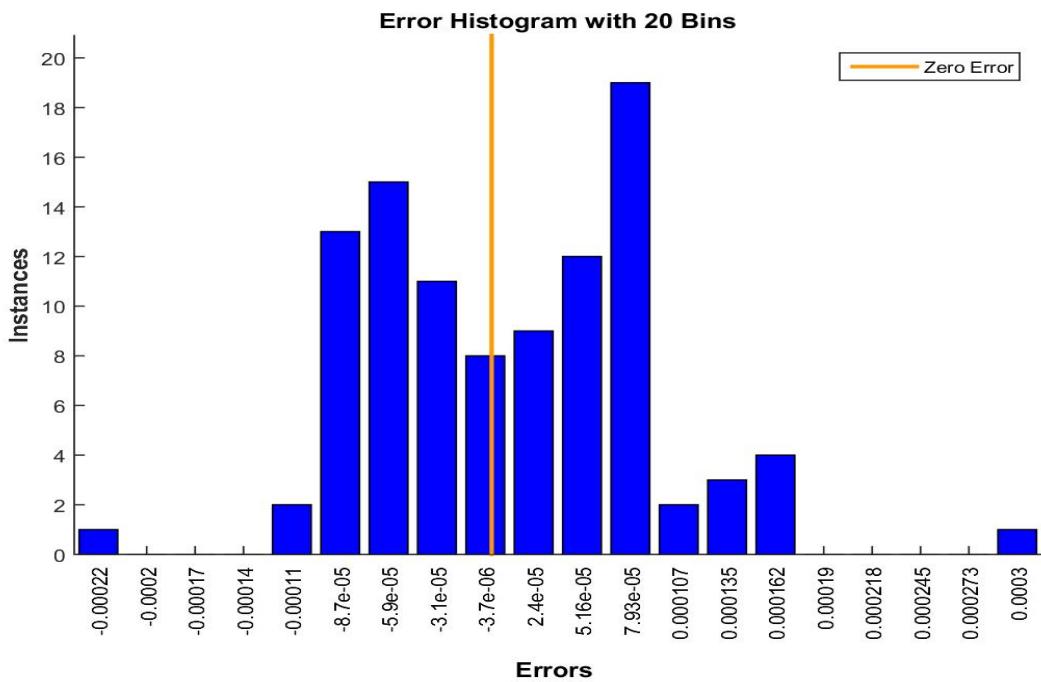
حالت اول:



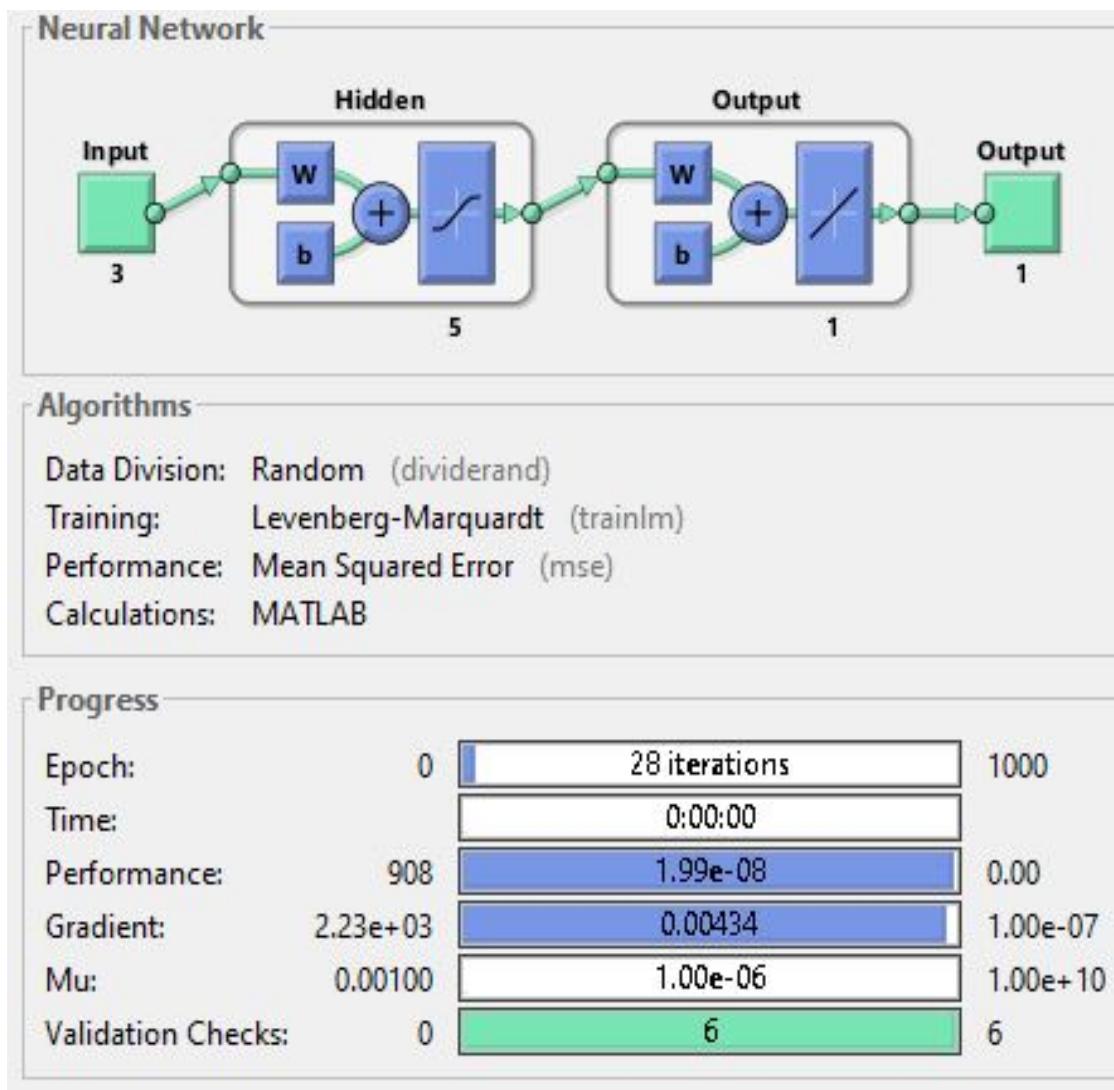


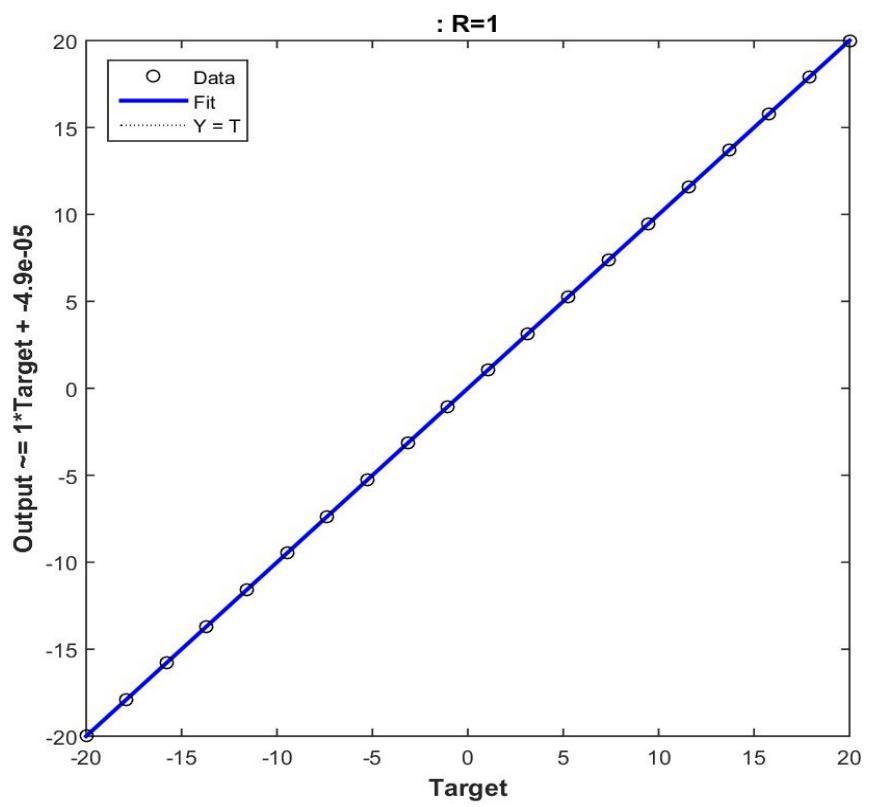
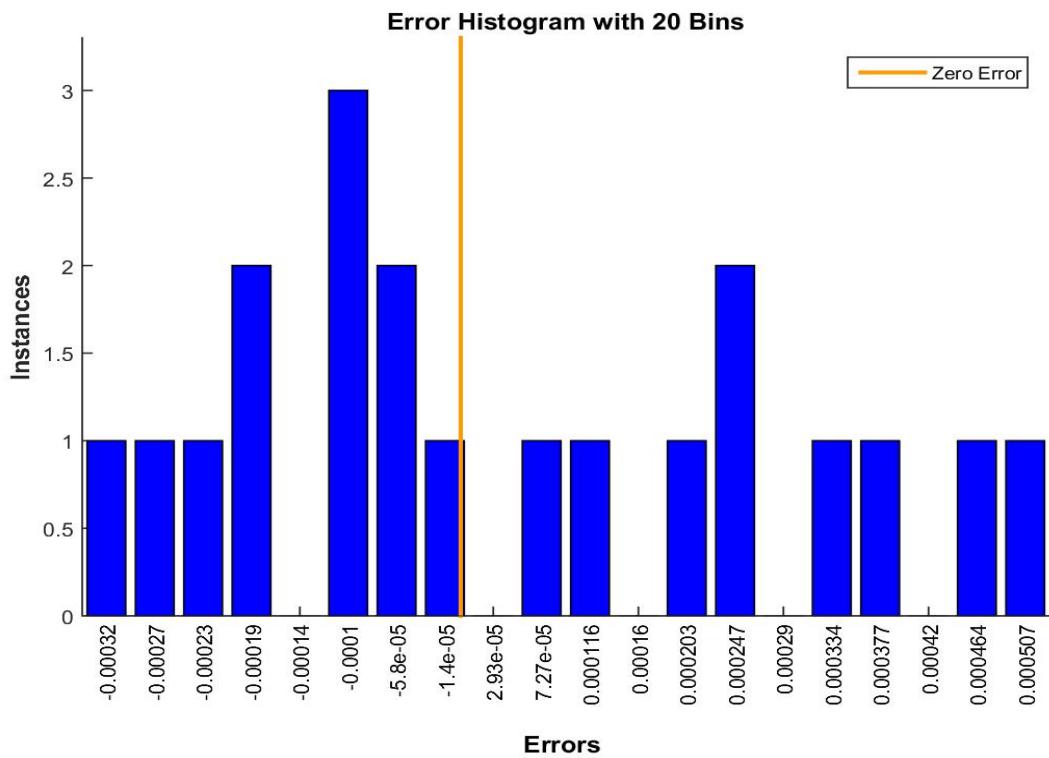
حالت دوم:



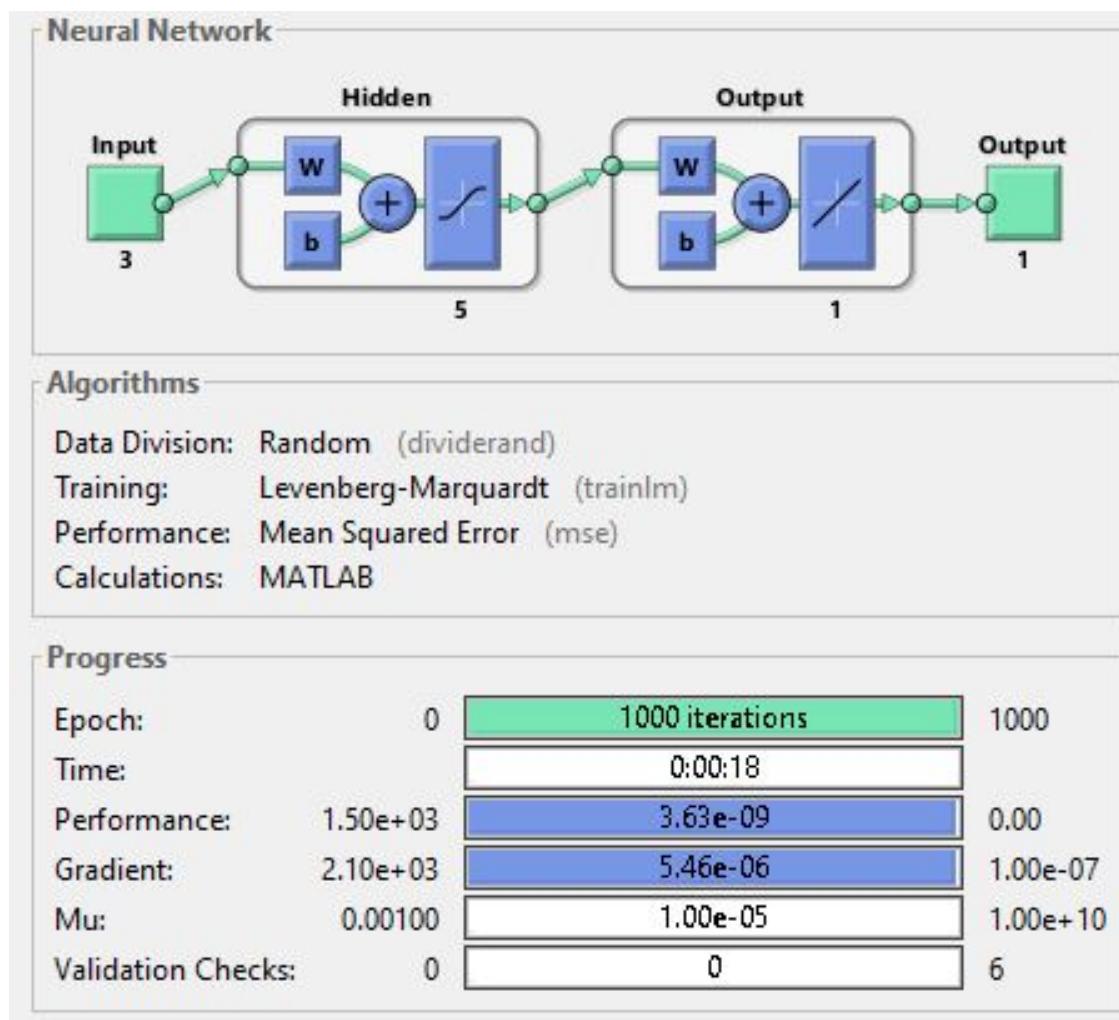


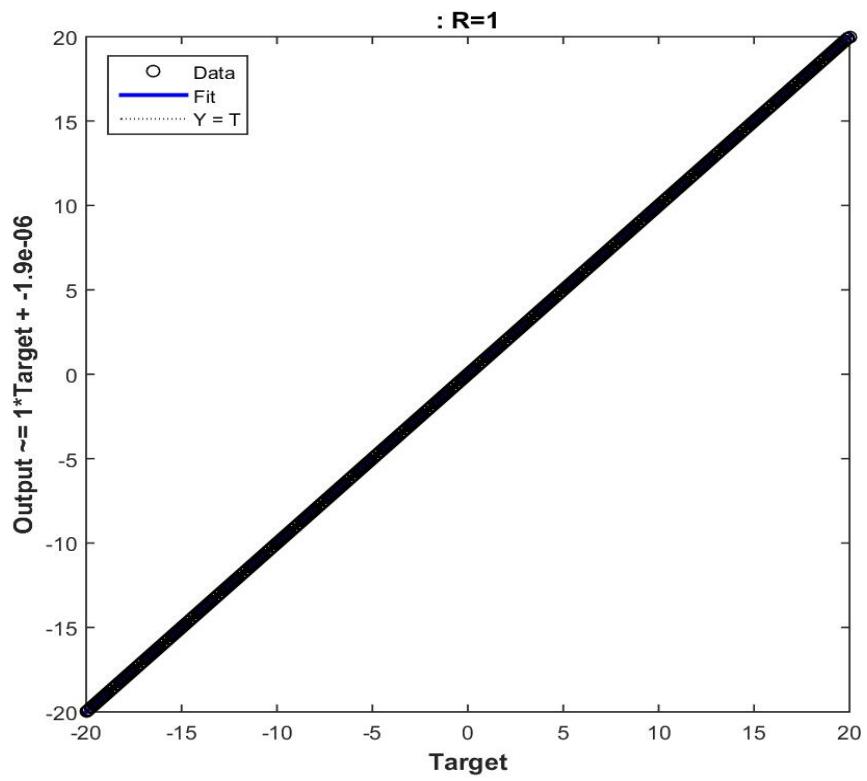
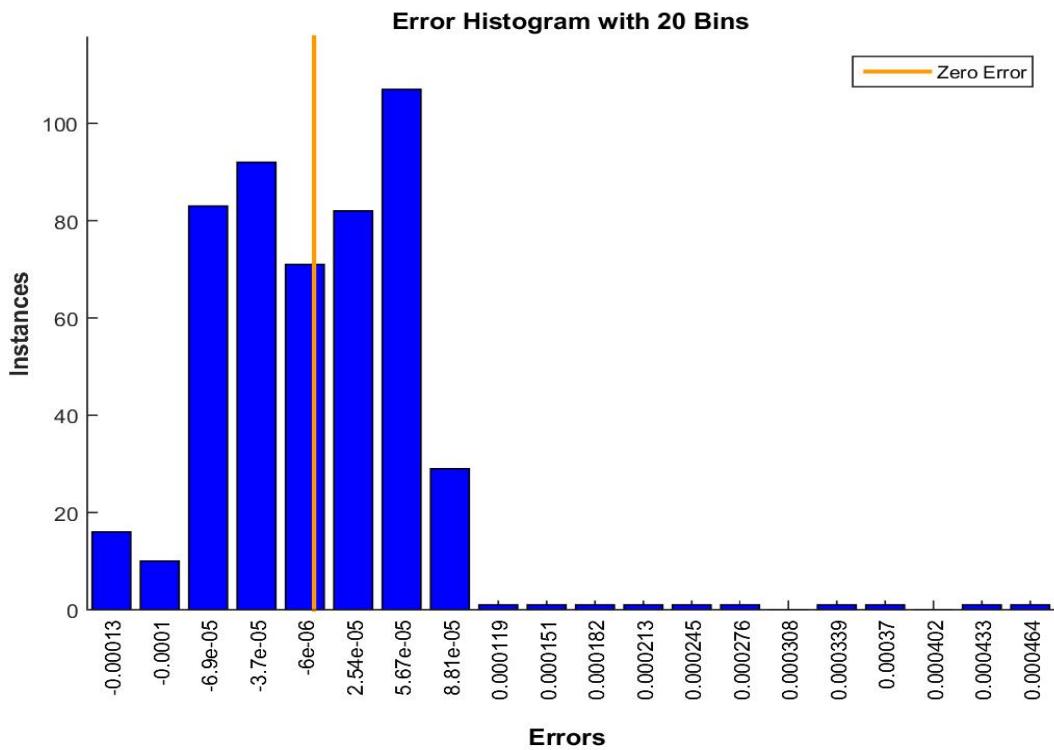
حالت سوم:





حالت چهارم:

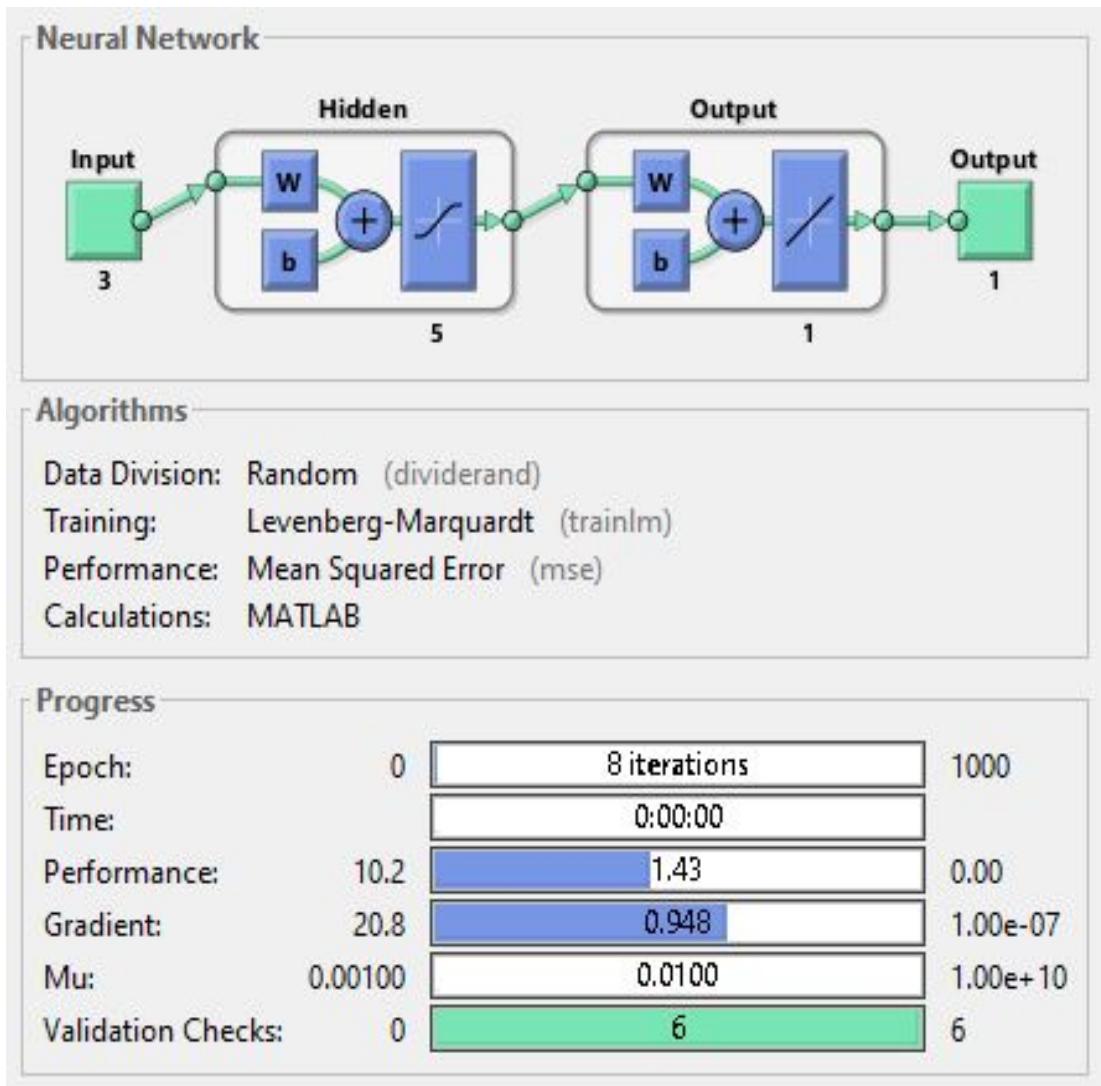


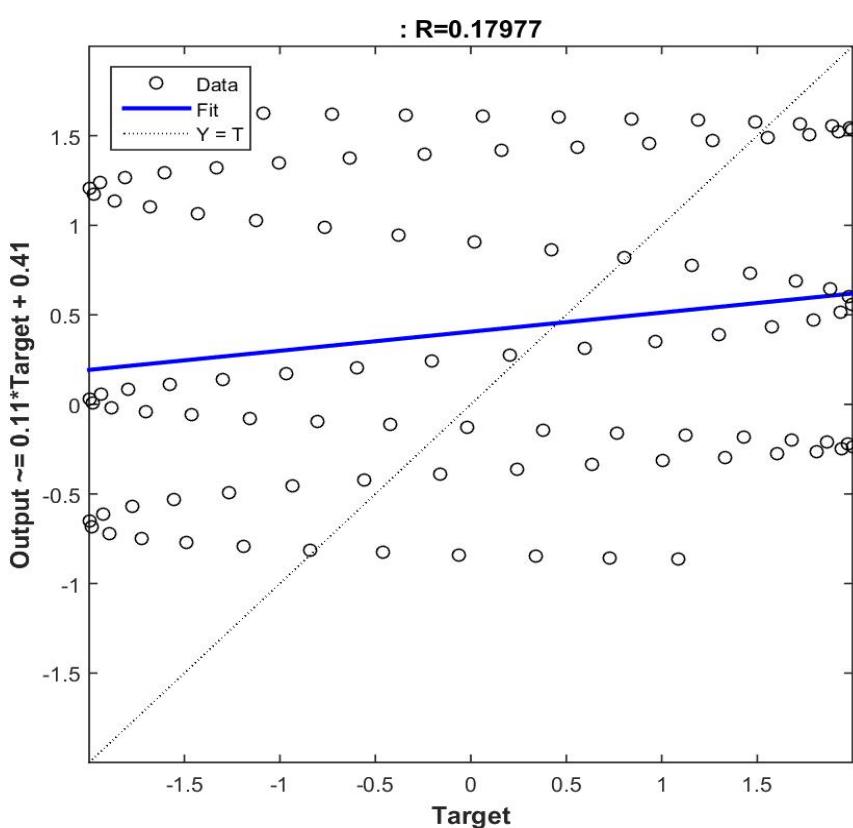
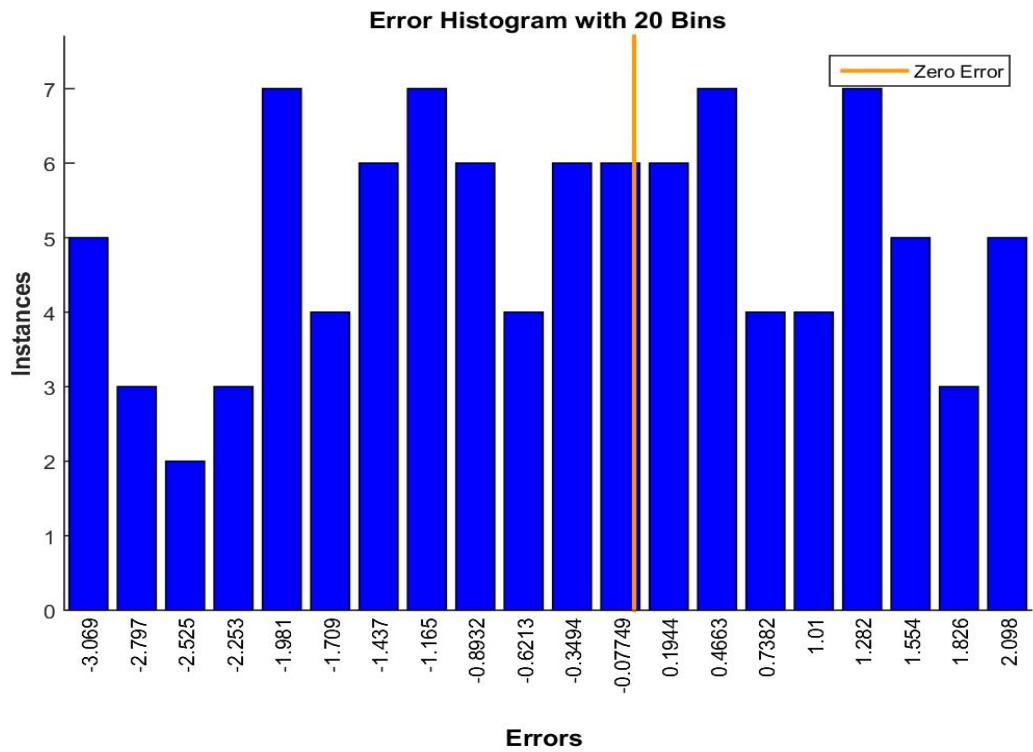


تابع دوم:

$$y_2 = \sin(x_1) + \sin(x_2)$$

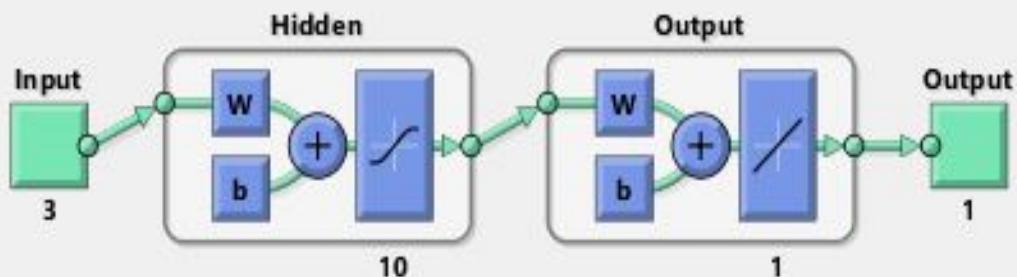
حالت اول:





حالت دوم:

Neural Network

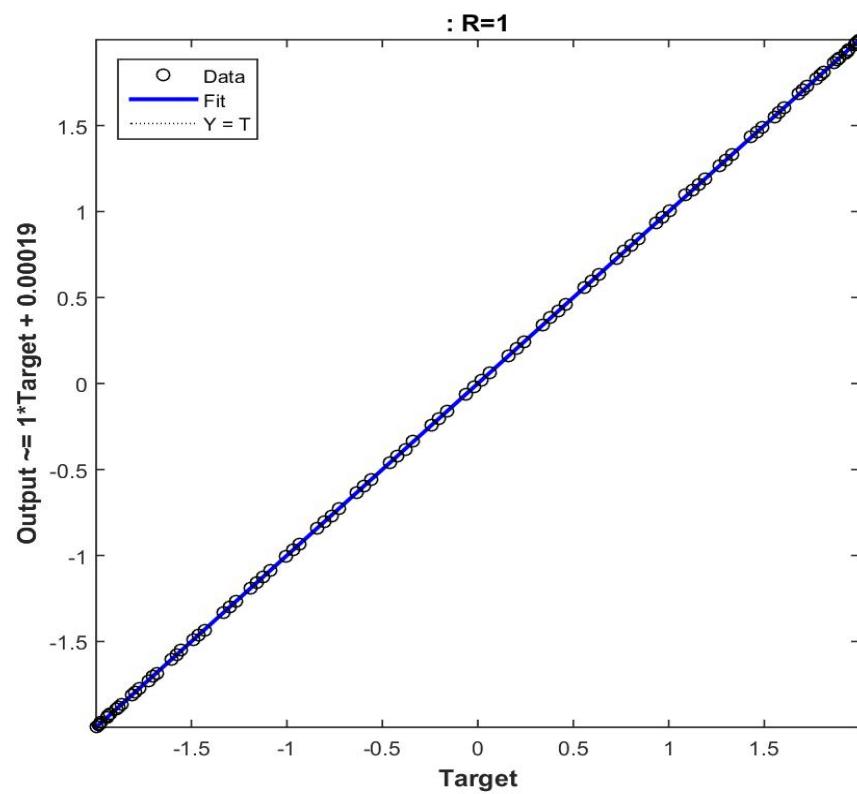
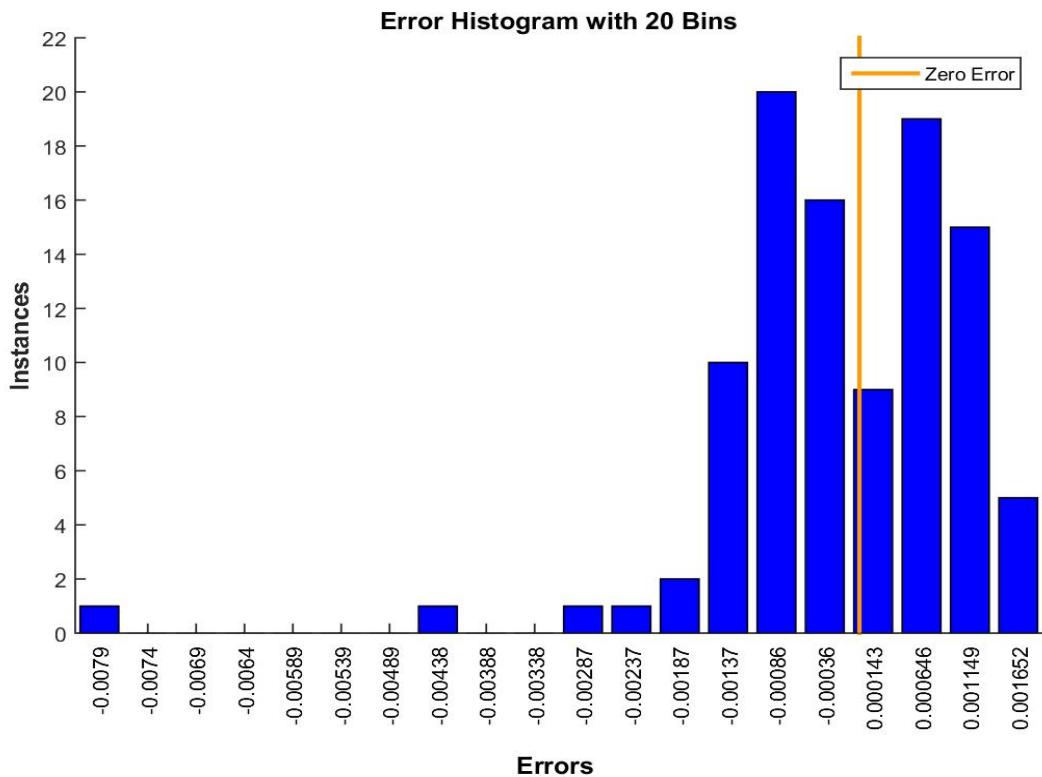


Algorithms

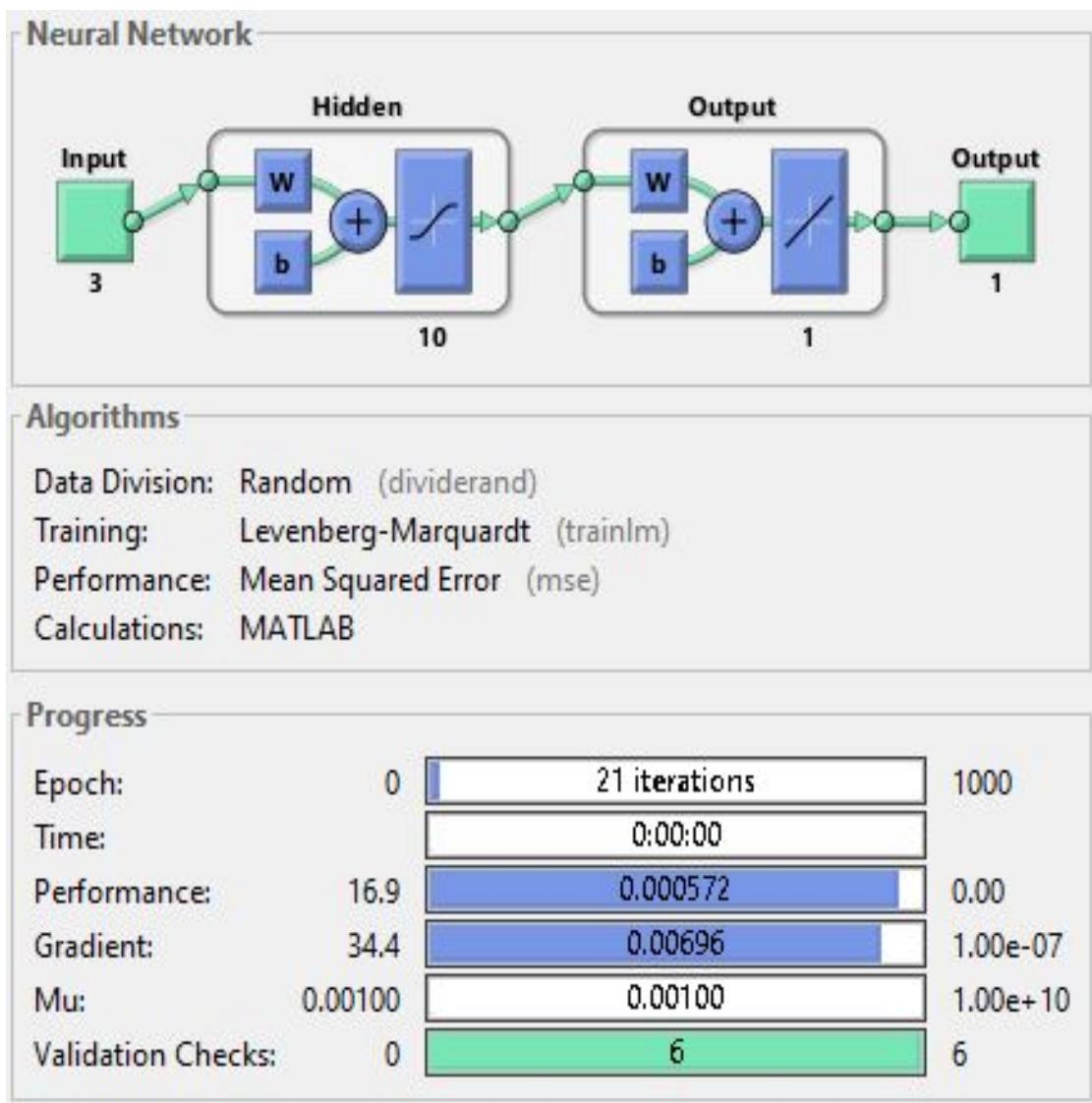
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MATLAB

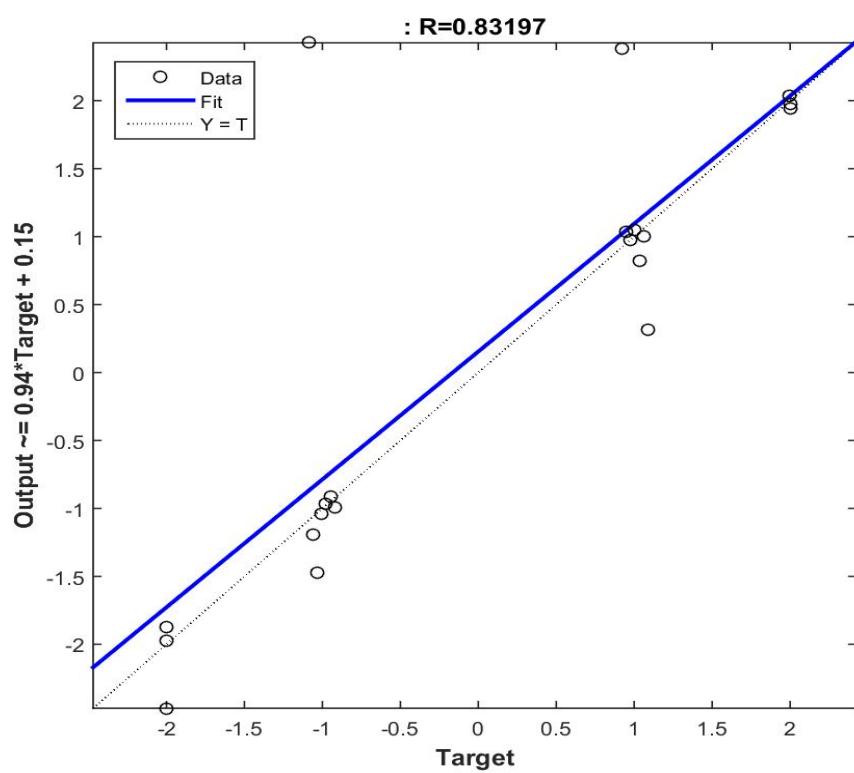
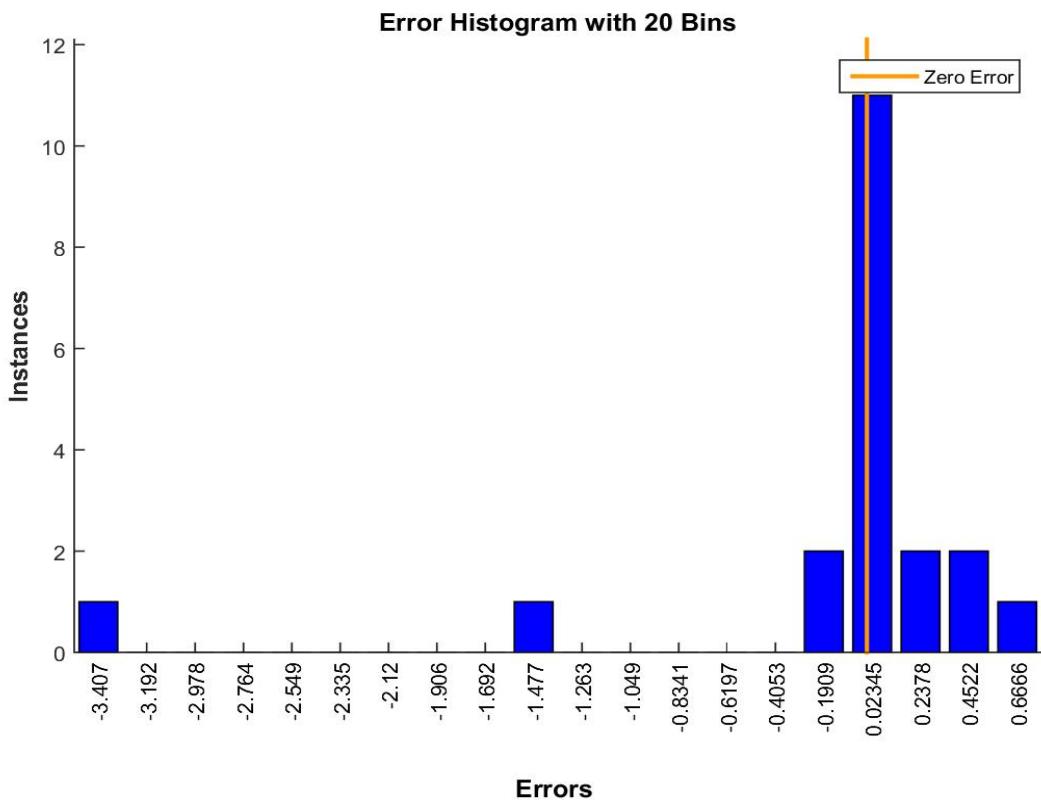
Progress

Epoch:	0	184 iterations	1000
Time:		0:00:02	
Performance:	16.3	8.94e-07	0.00
Gradient:	33.0	2.34e-05	1.00e-07
Mu:	0.00100	1.00e-06	1.00e+10
Validation Checks:	0	6	6

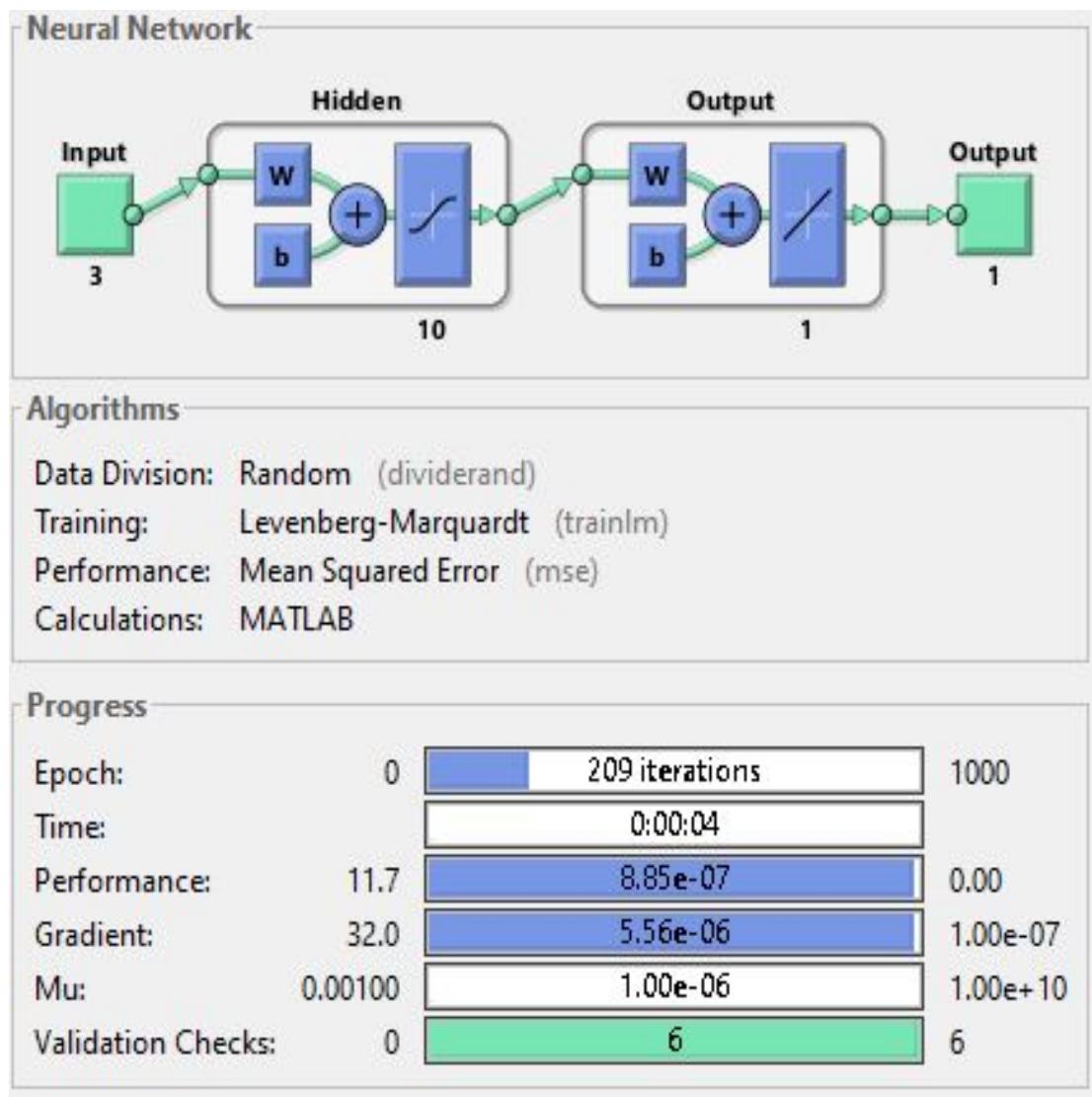


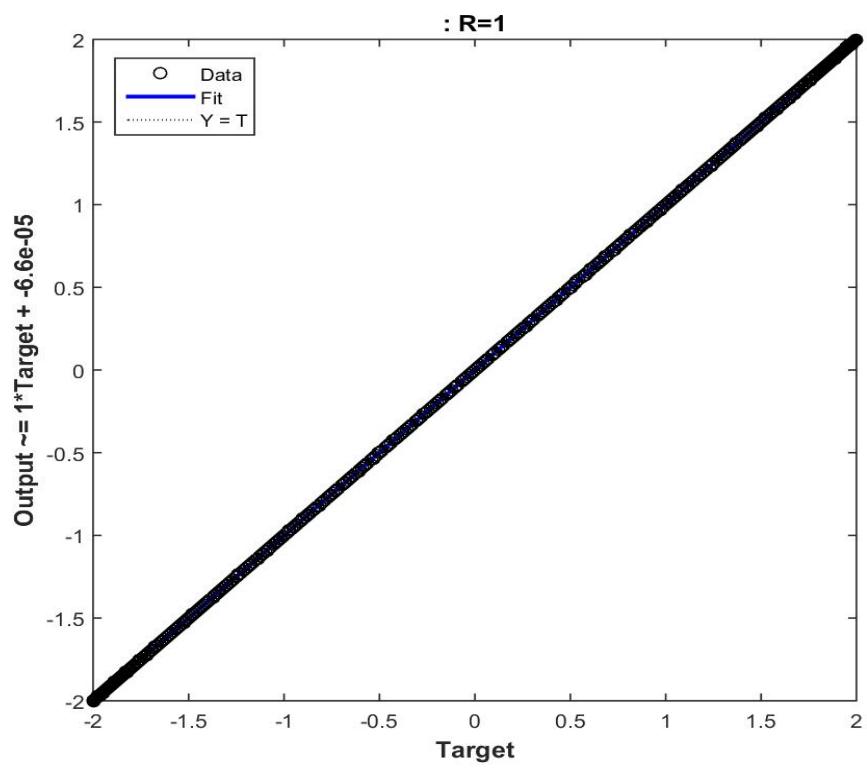
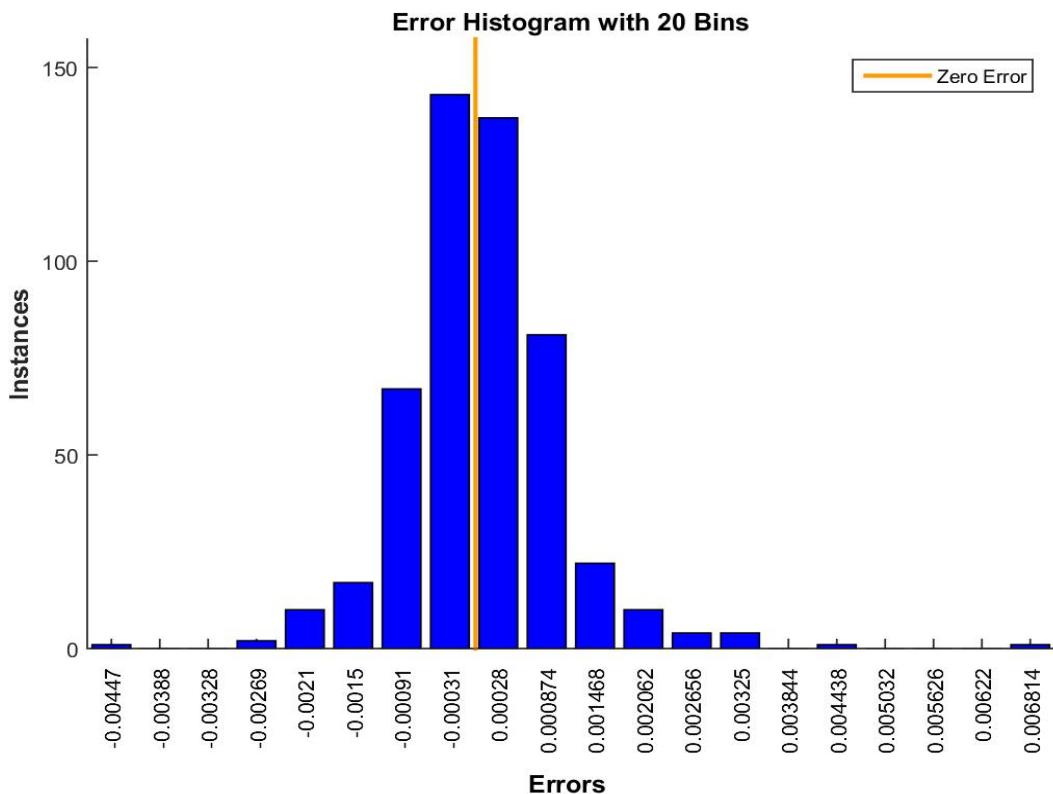
حالت سوم:





حالت چهارم:

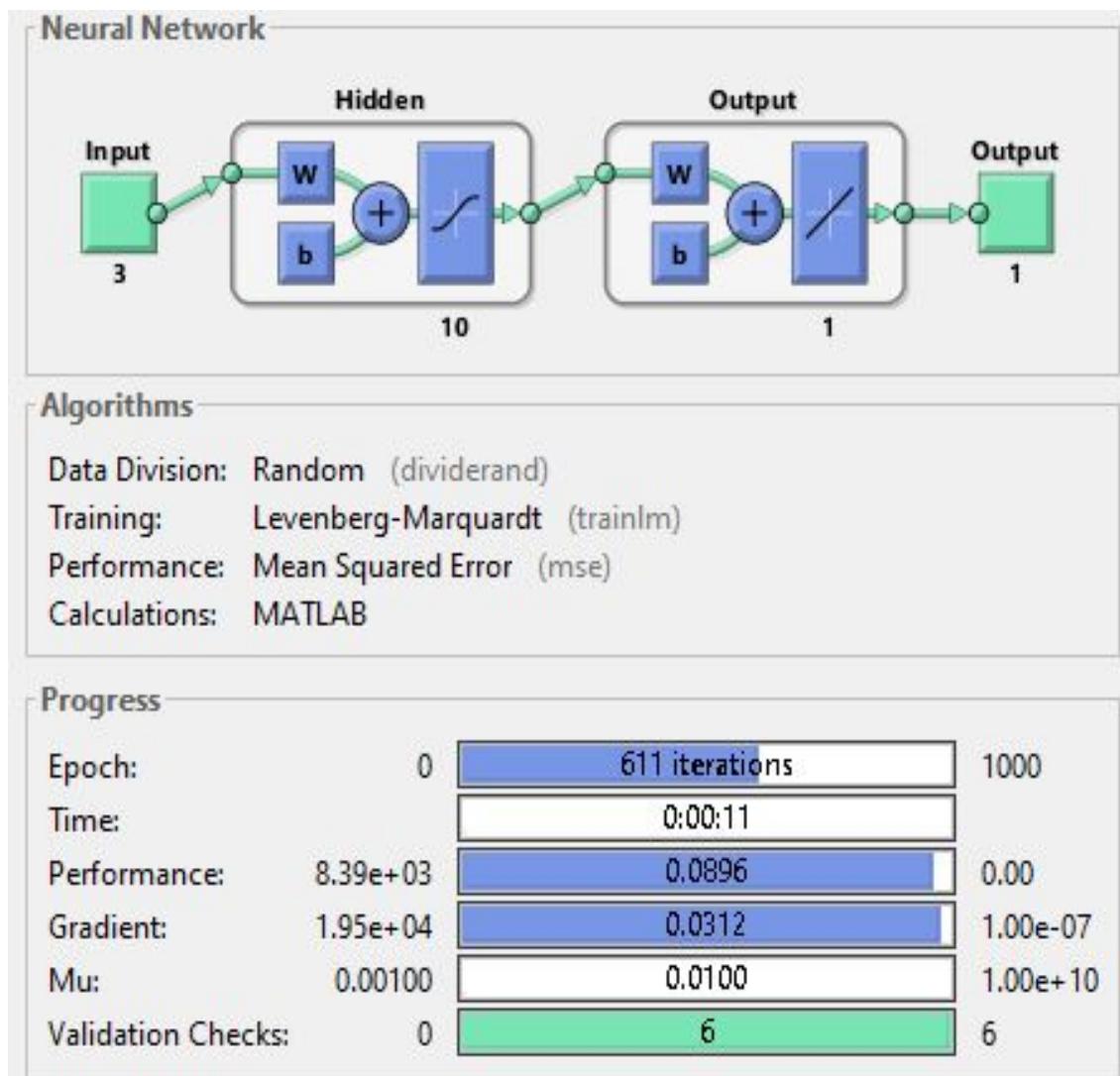


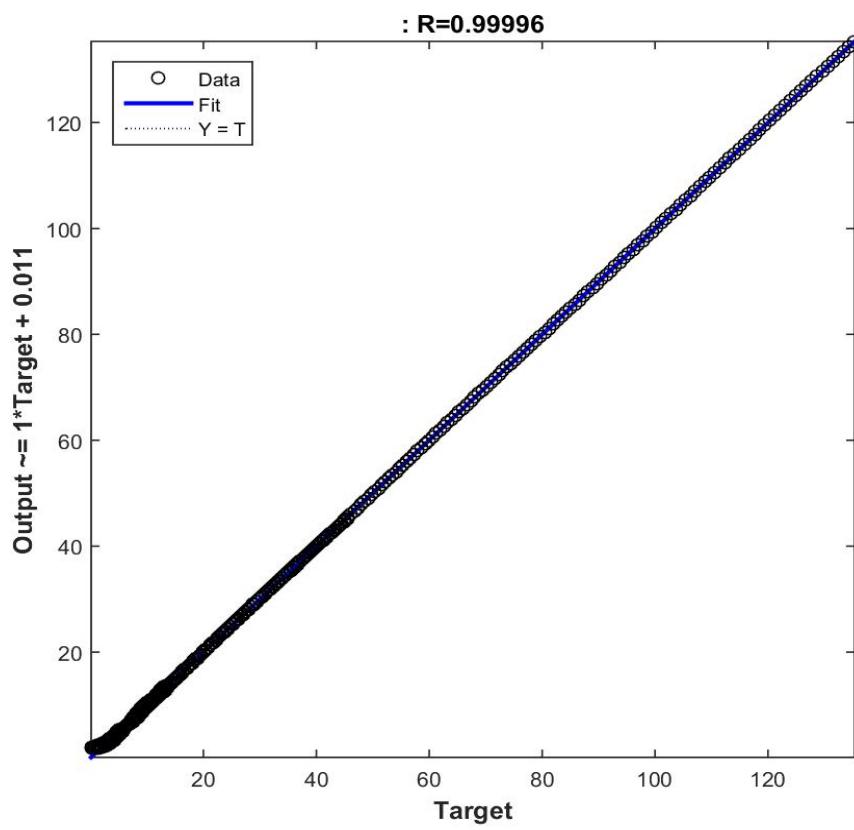
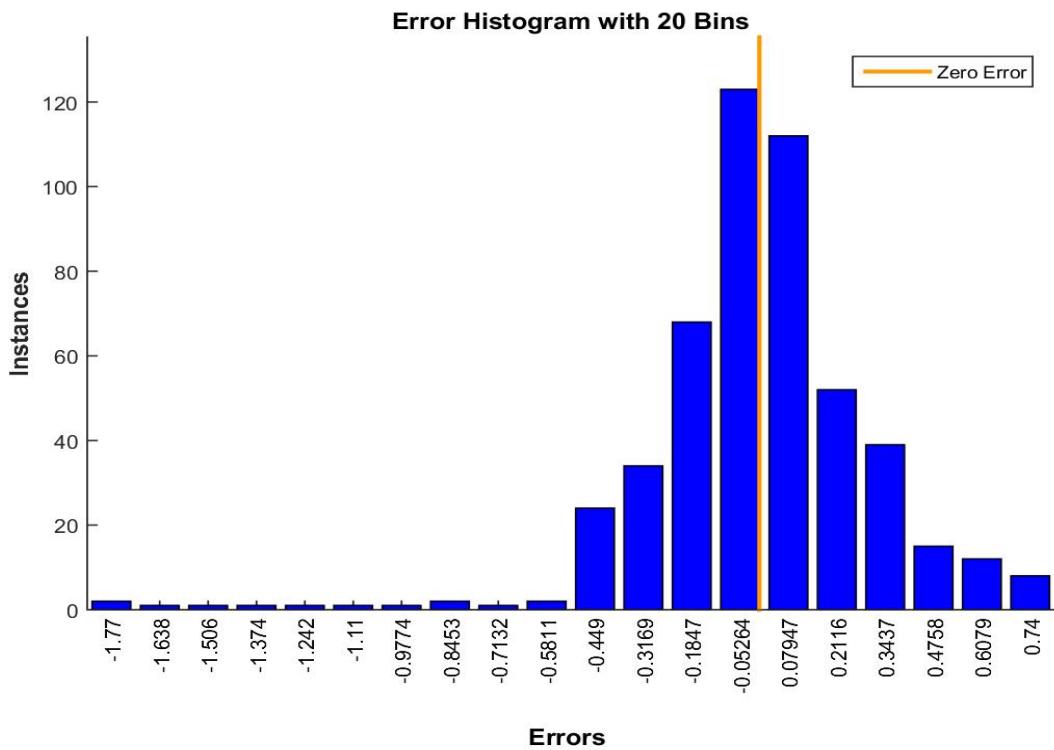


تابع سوم:

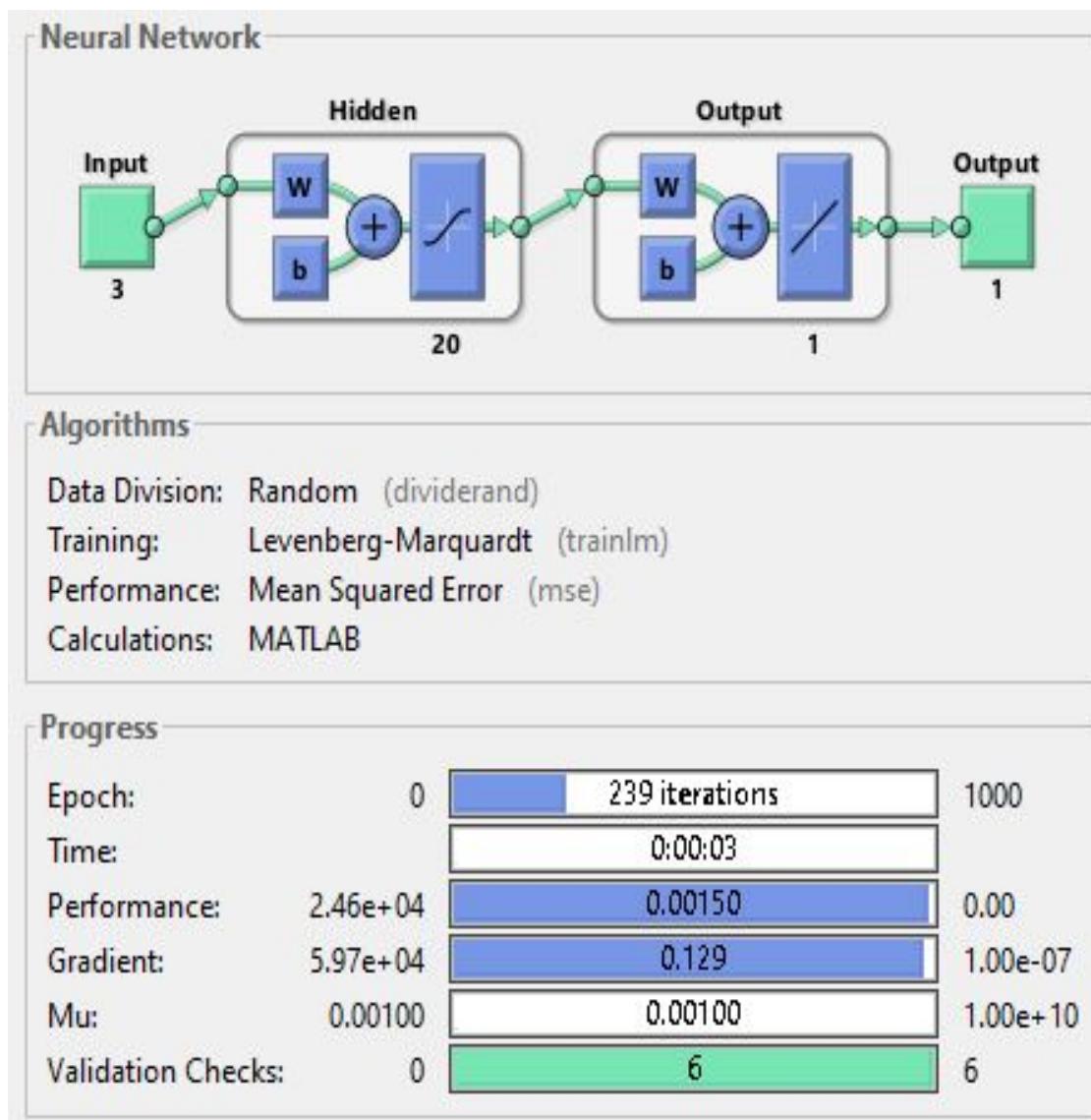
$$y_3 = \text{abs}(2*x1.^2 + 4*x1 + 5 - 3*x2.^2 + 3*\sqrt{x3})$$

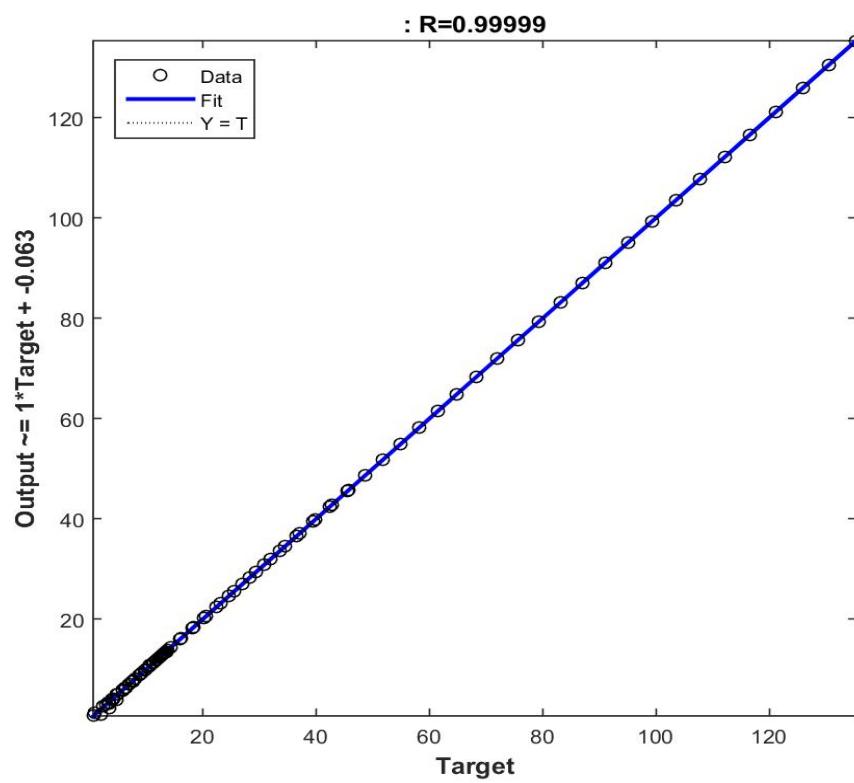
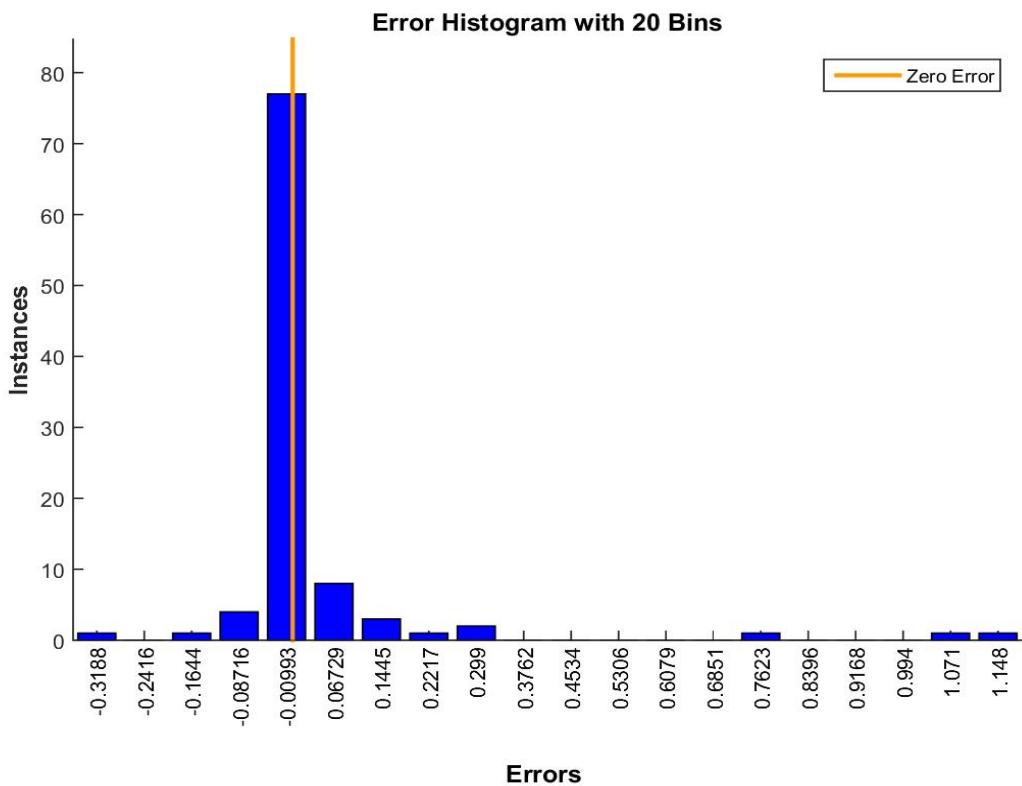
حالت اول:



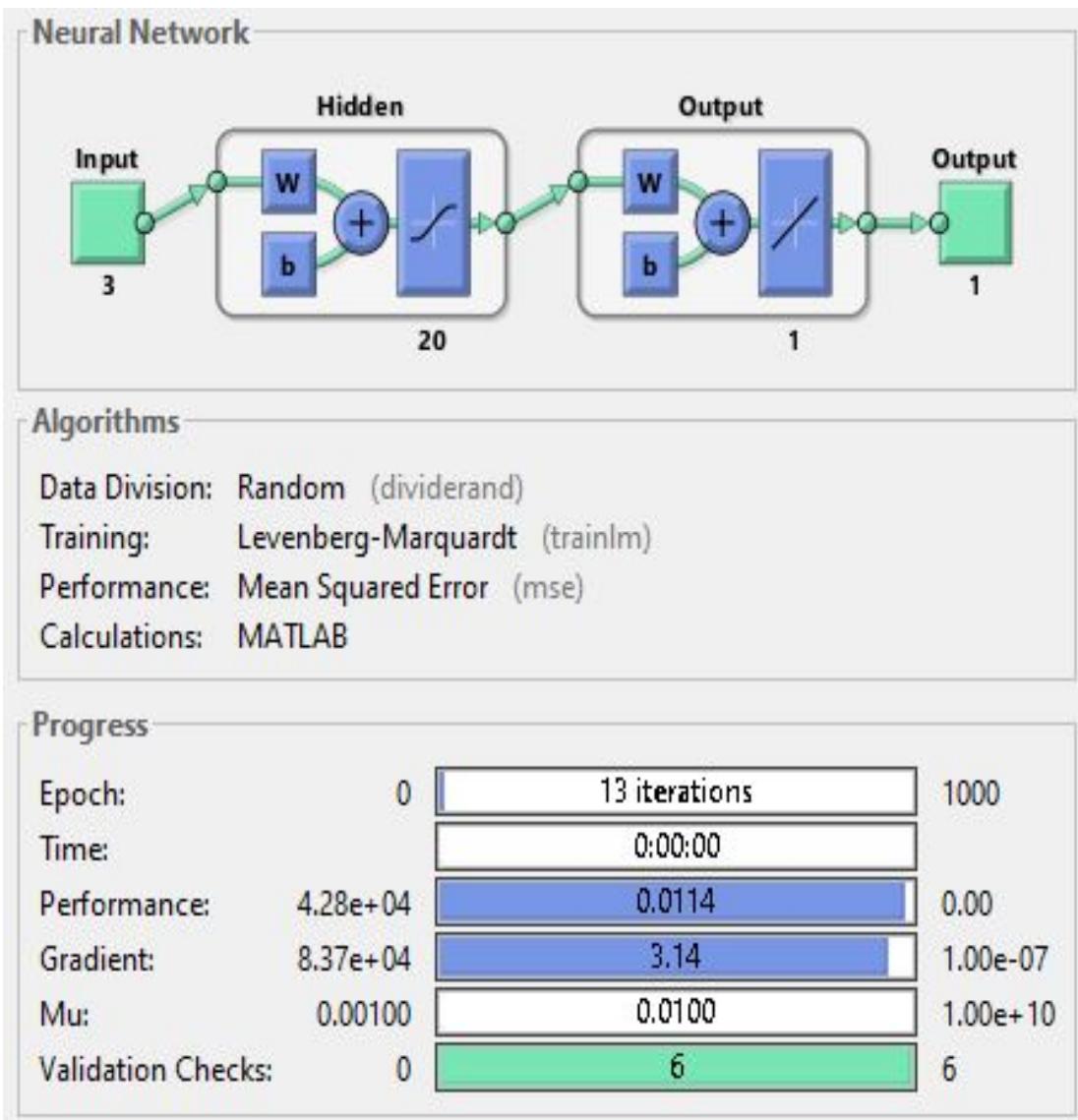


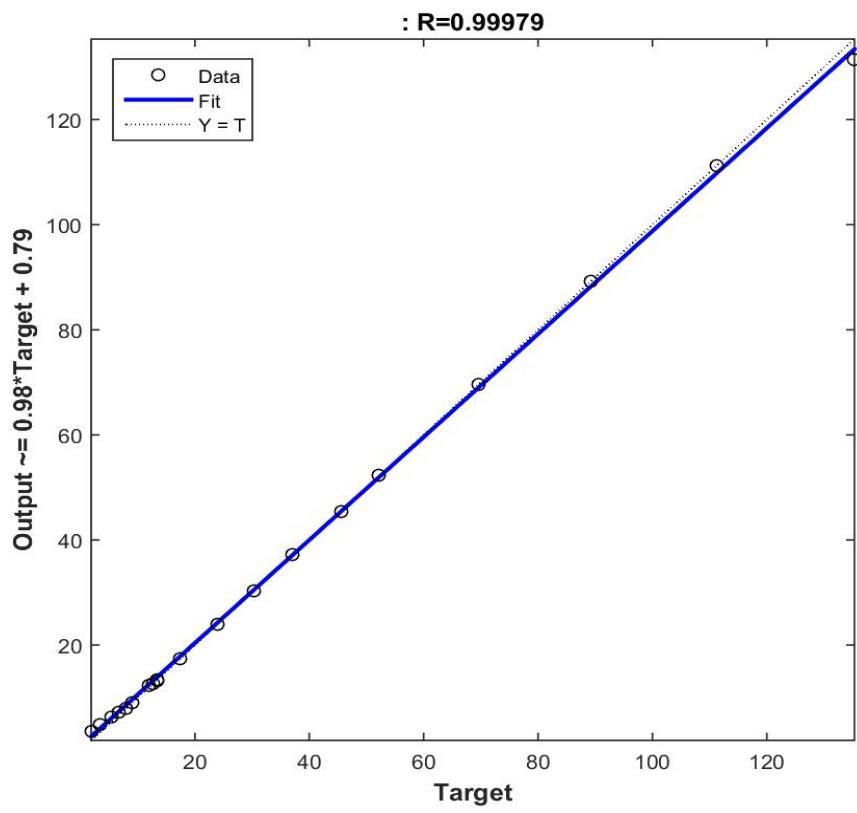
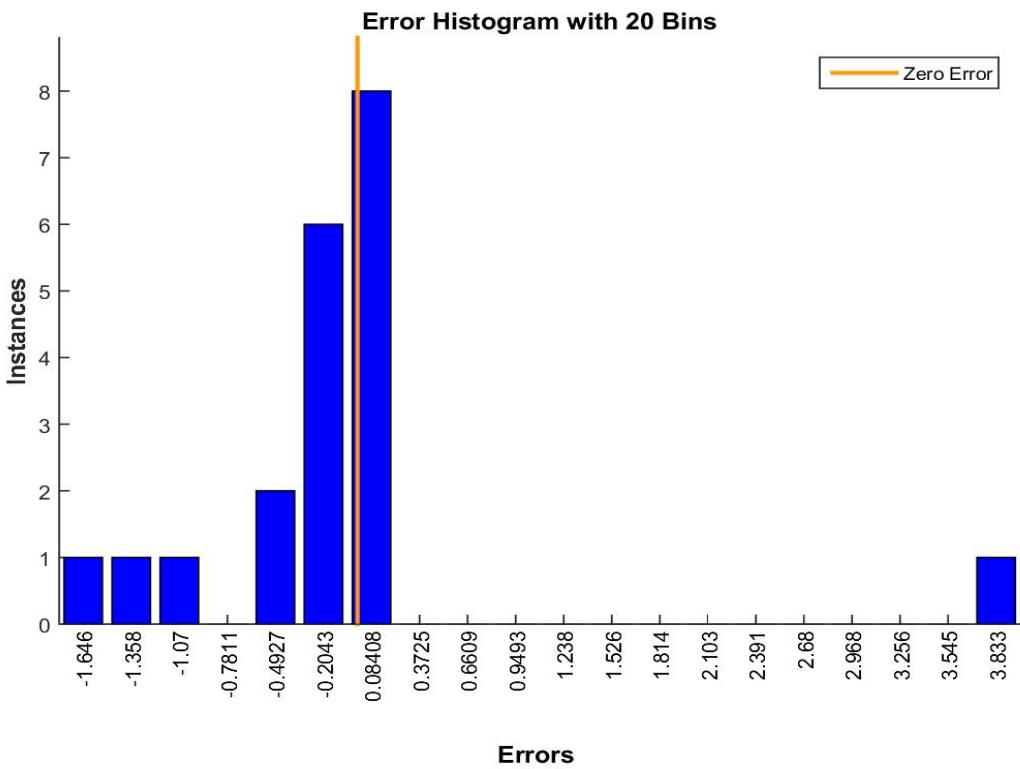
حالت دوم:





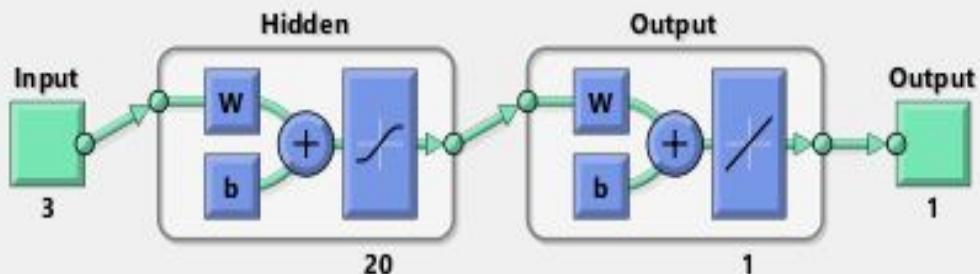
حالت سوم:





حالت چهارم:

Neural Network

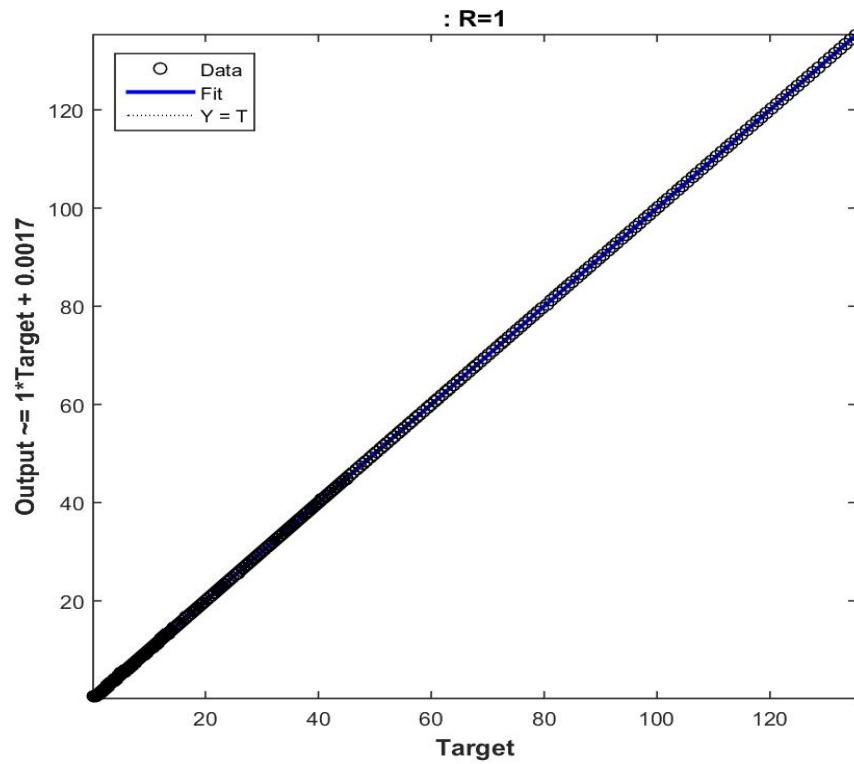
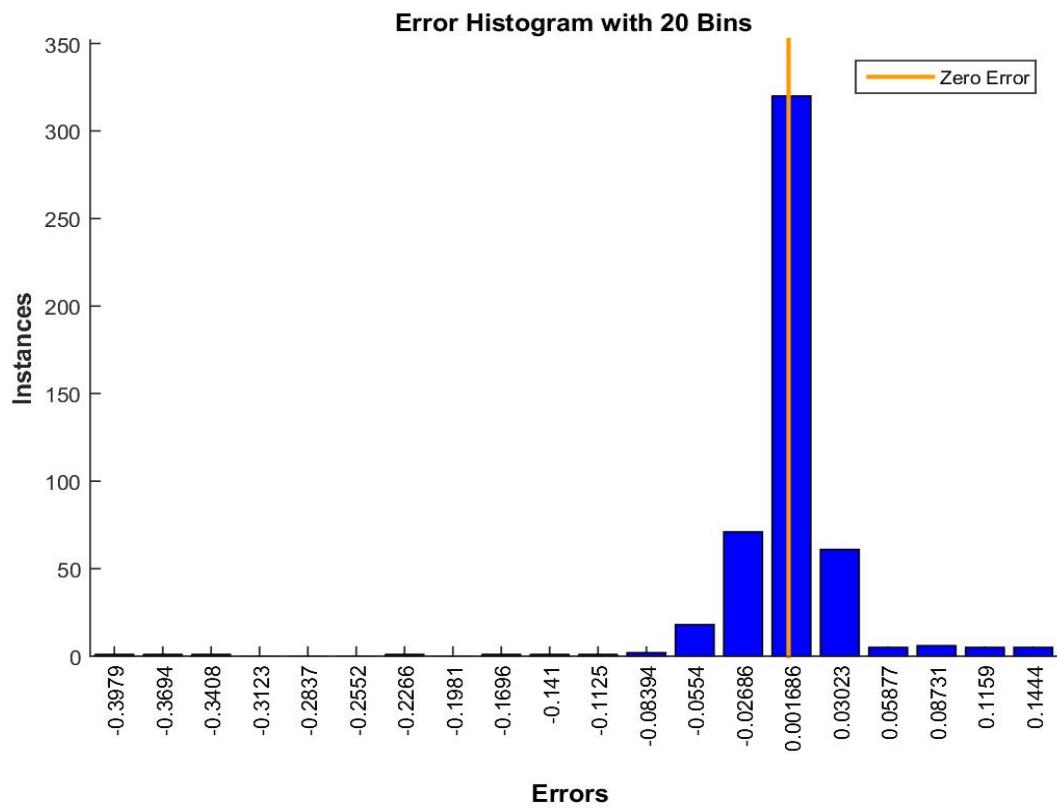


Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MATLAB

Progress

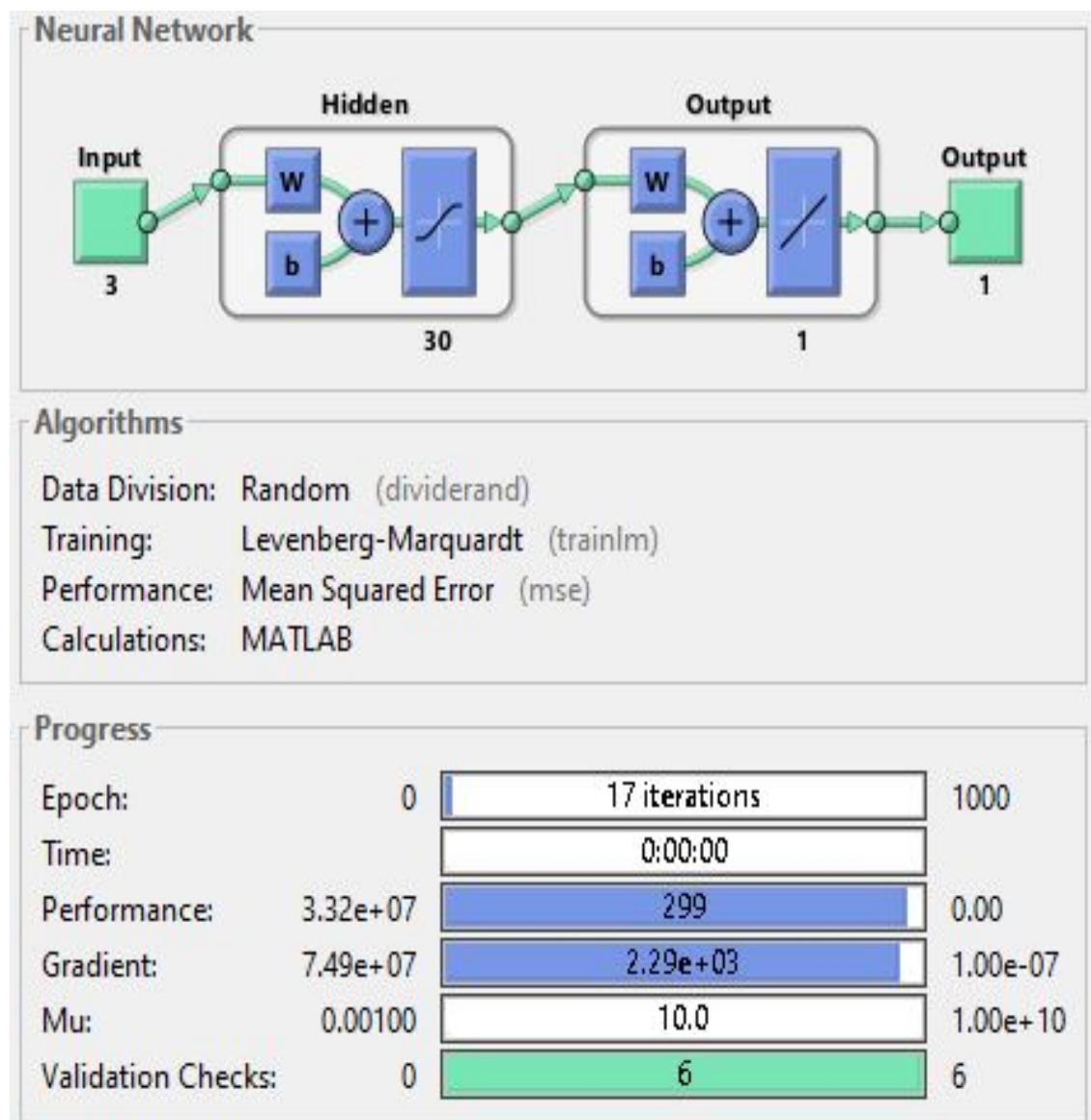
Epoch:	0	1000 iterations	1000
Time:		0:00:22	
Performance:	1.95e+04	0.00155	0.00
Gradient:	5.03e+04	0.0650	1.00e-07
Mu:	0.00100	0.000100	1.00e+10
Validation Checks:	0	0	6

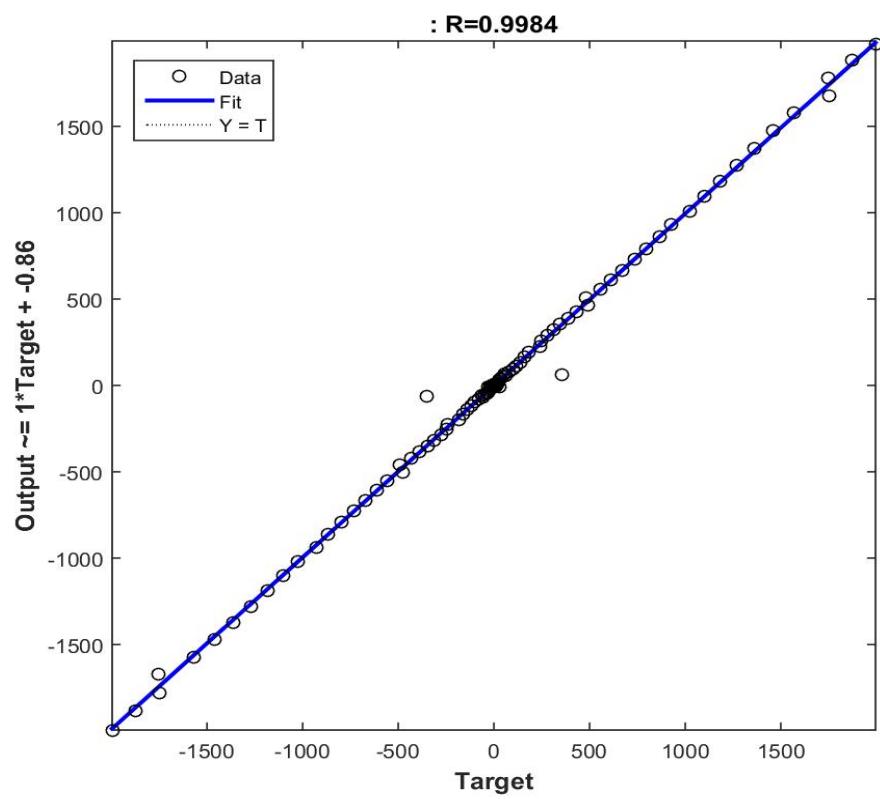
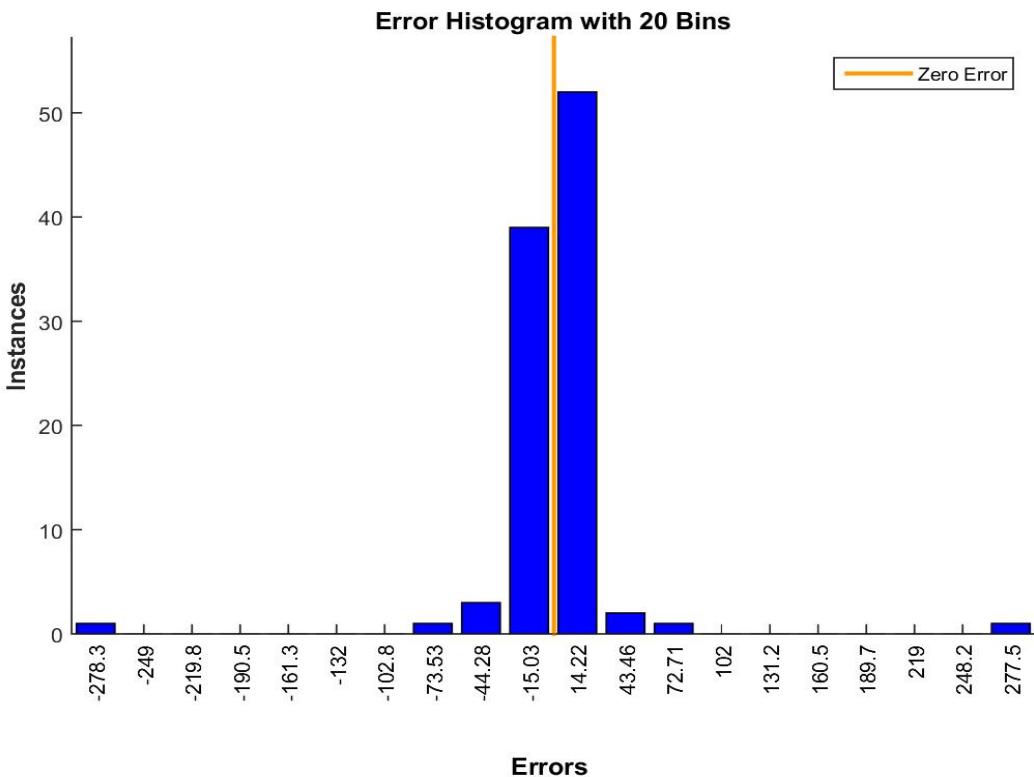


تابع چهارم:

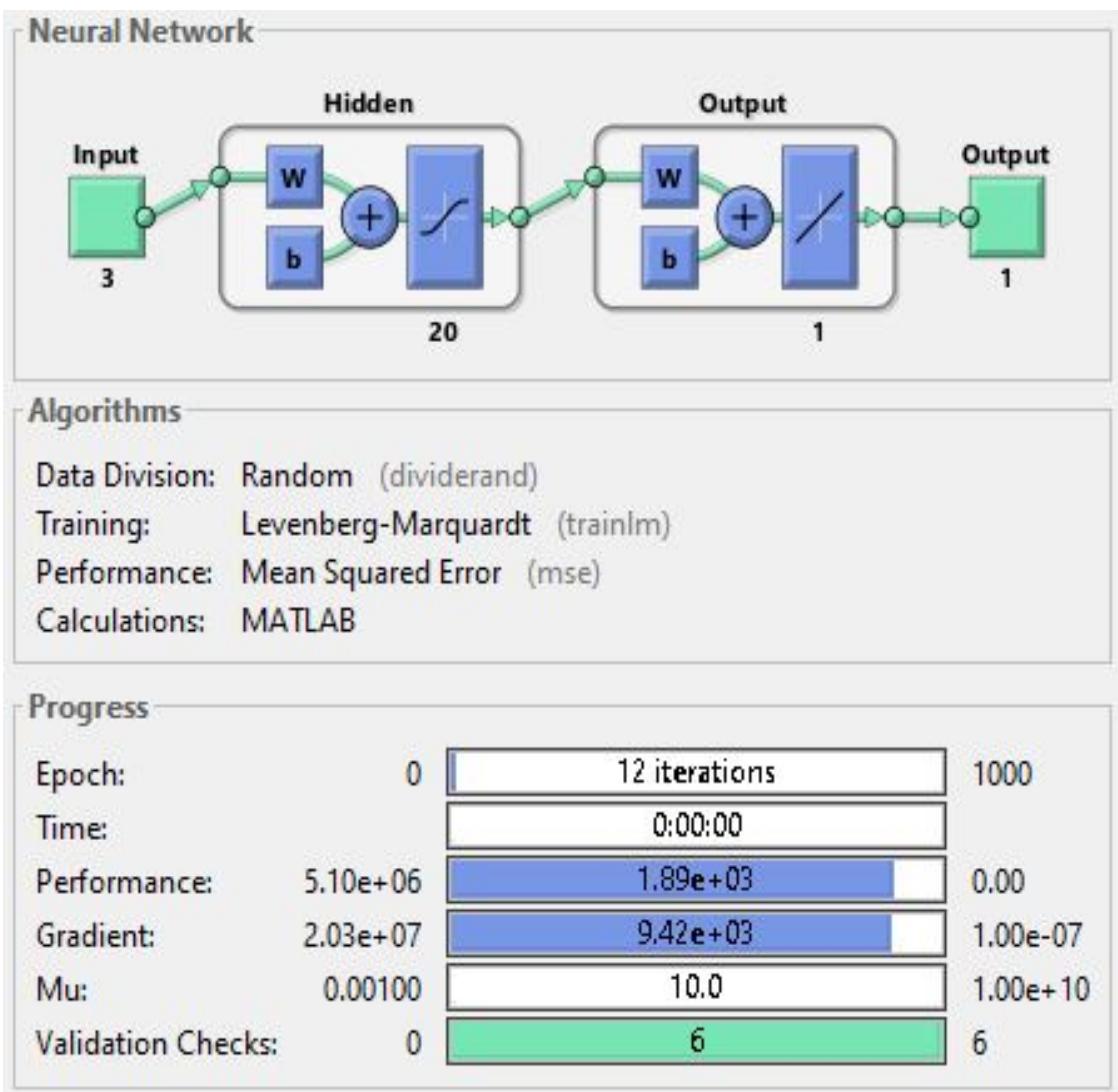
$$y_4 = \tan(x_1) + 3 * \cot(x_2) - 2 * x_3.^3;$$

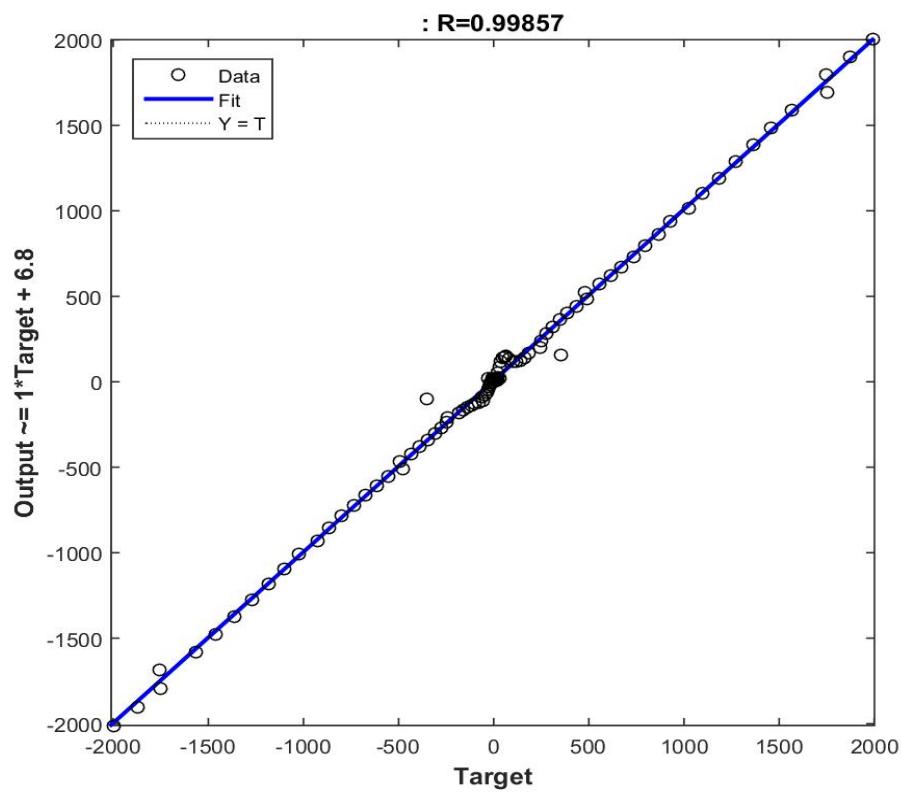
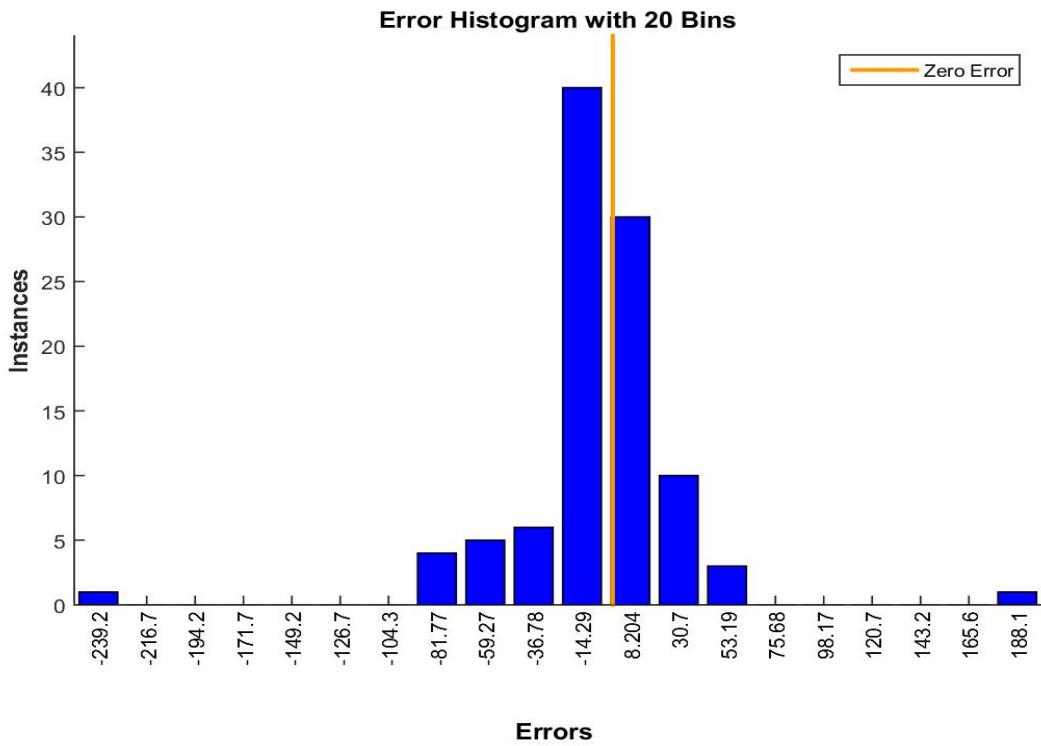
حالت اول:



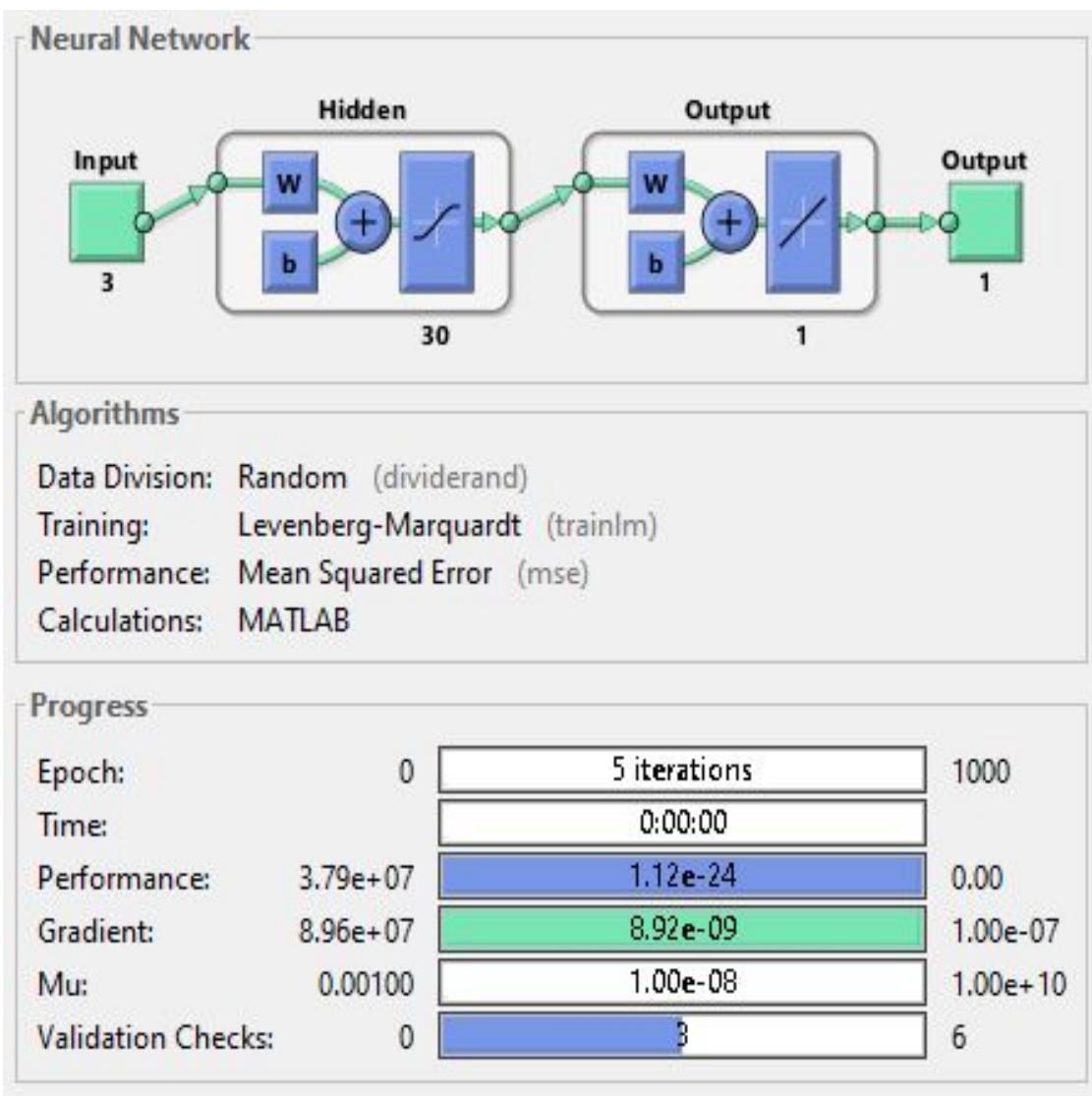


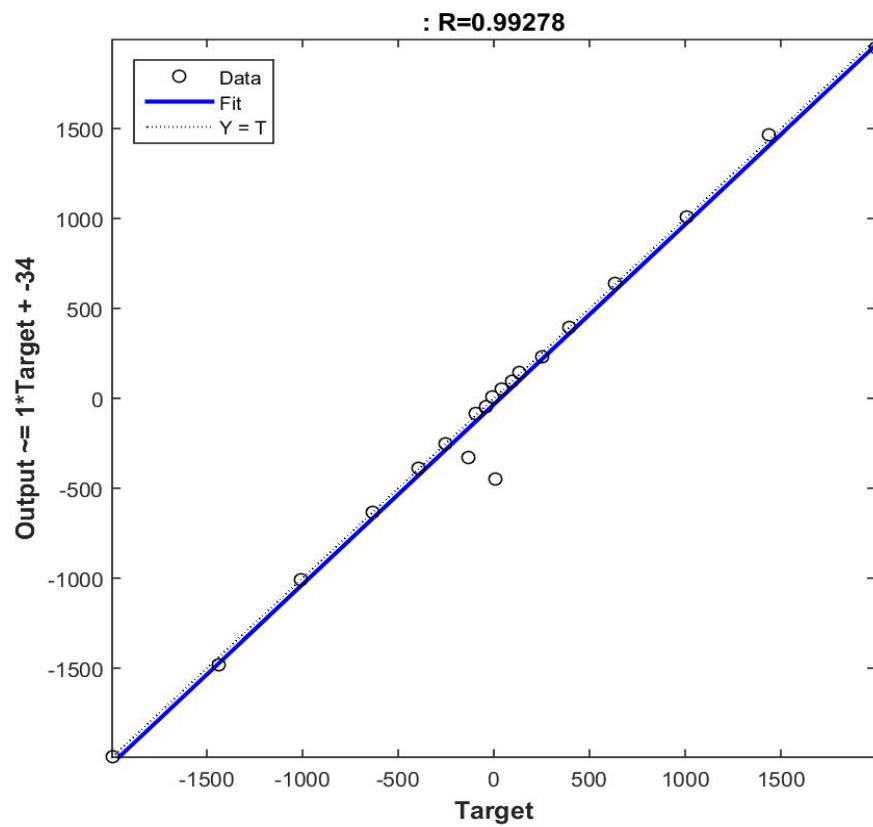
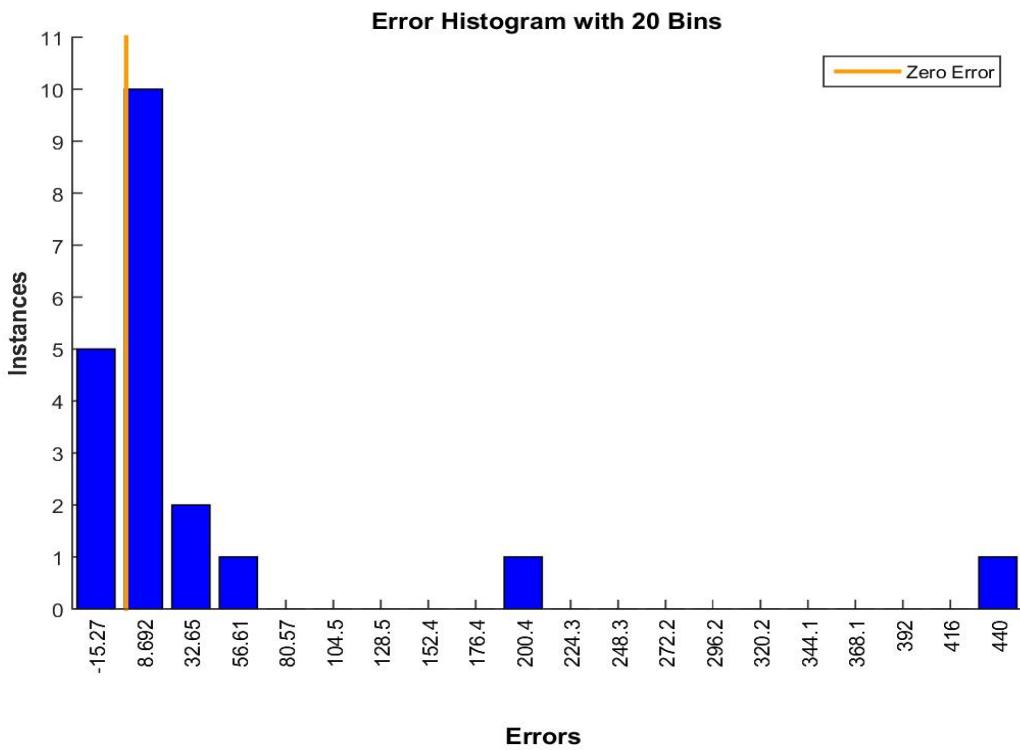
حالت دوم:



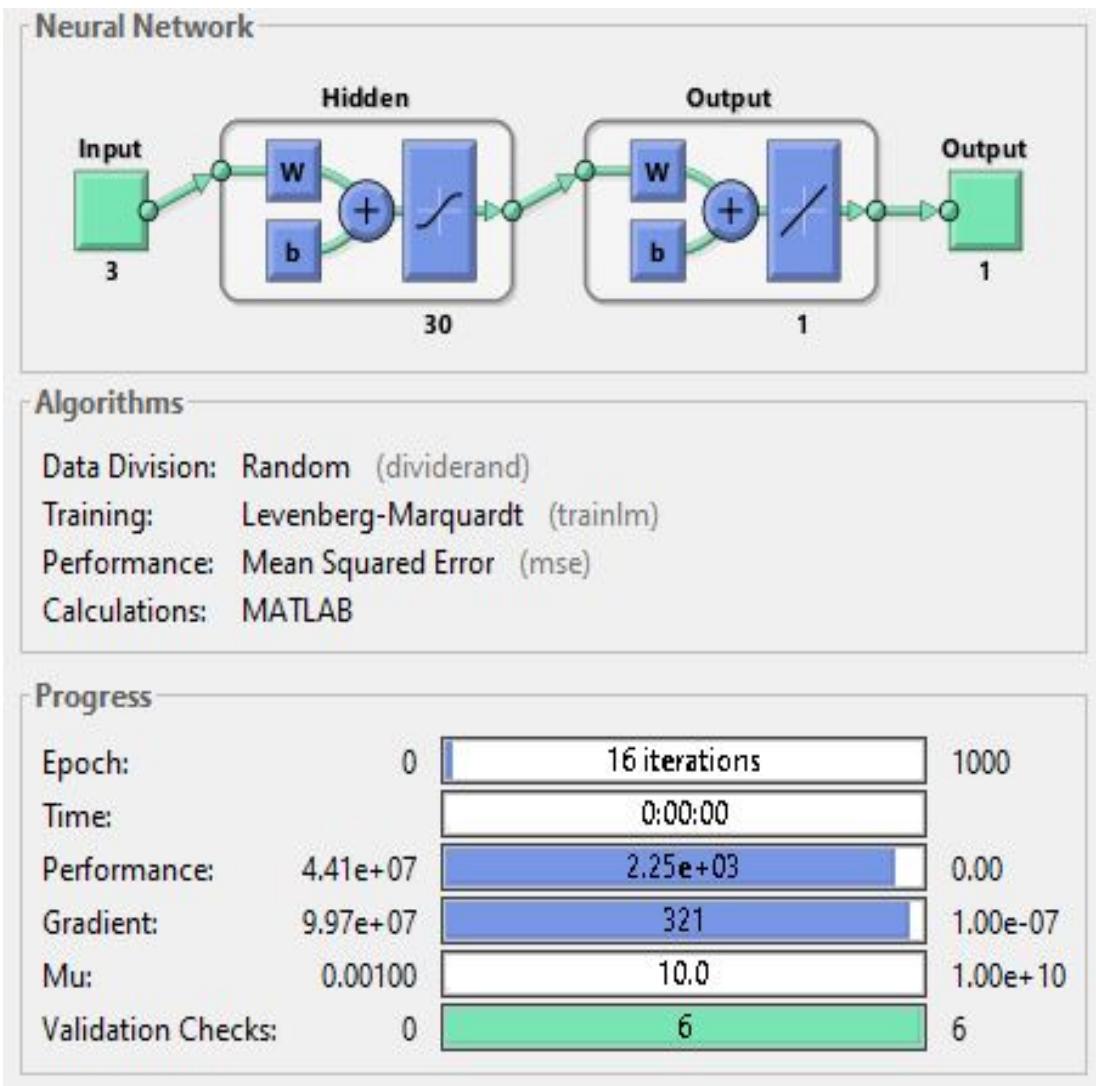


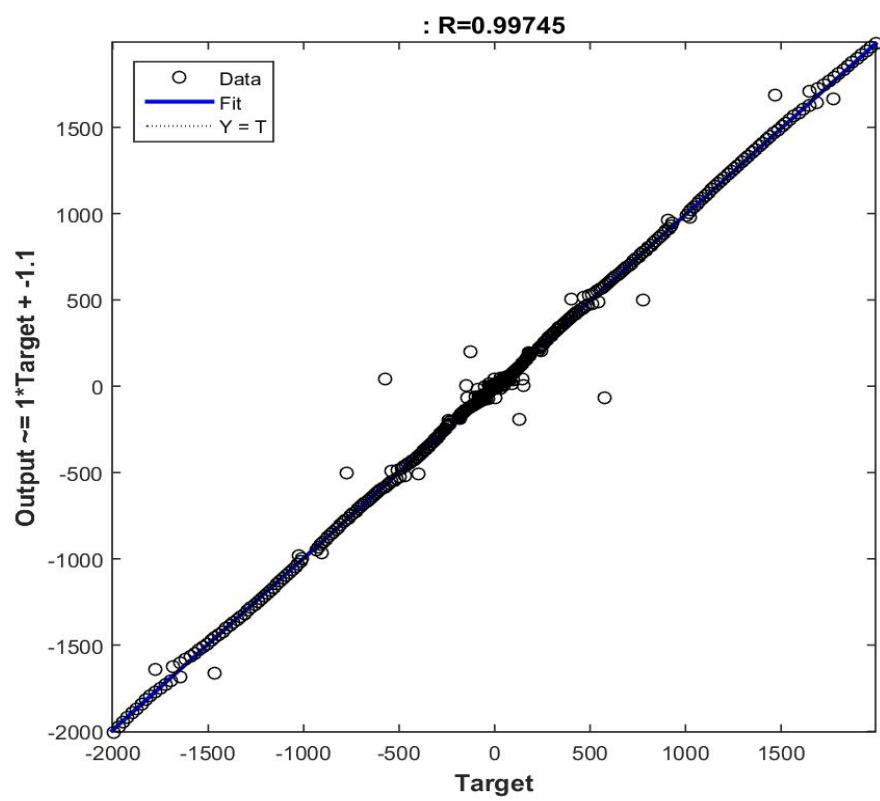
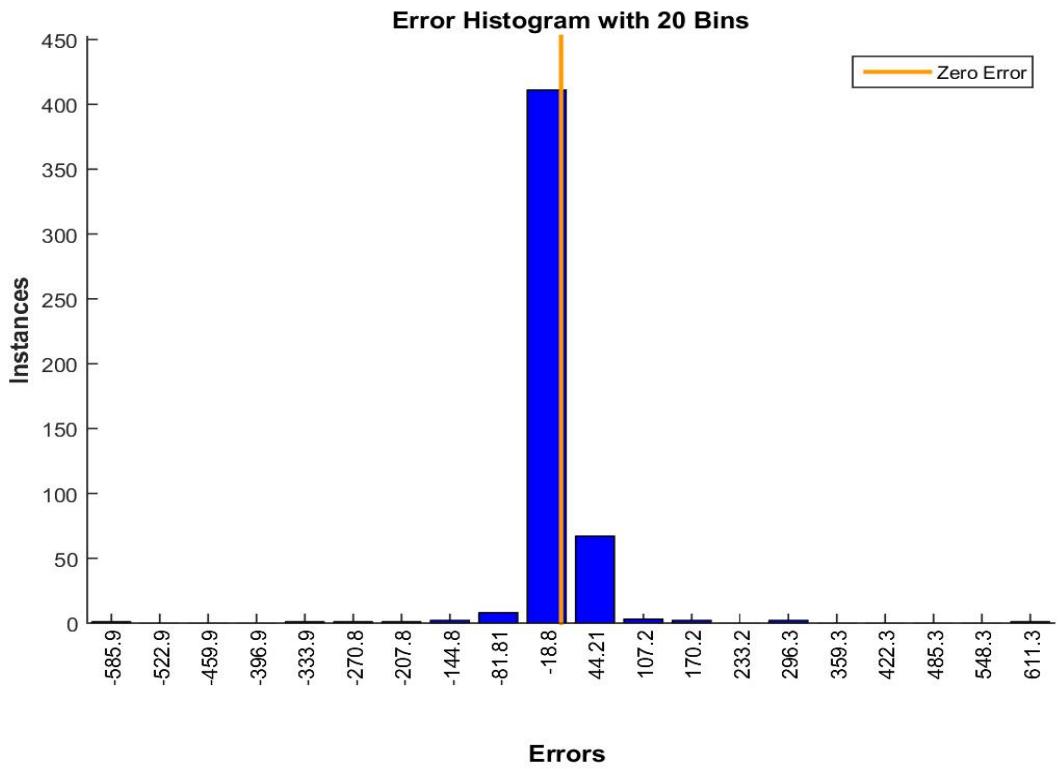
حالت سوم:





حالت چهارم:

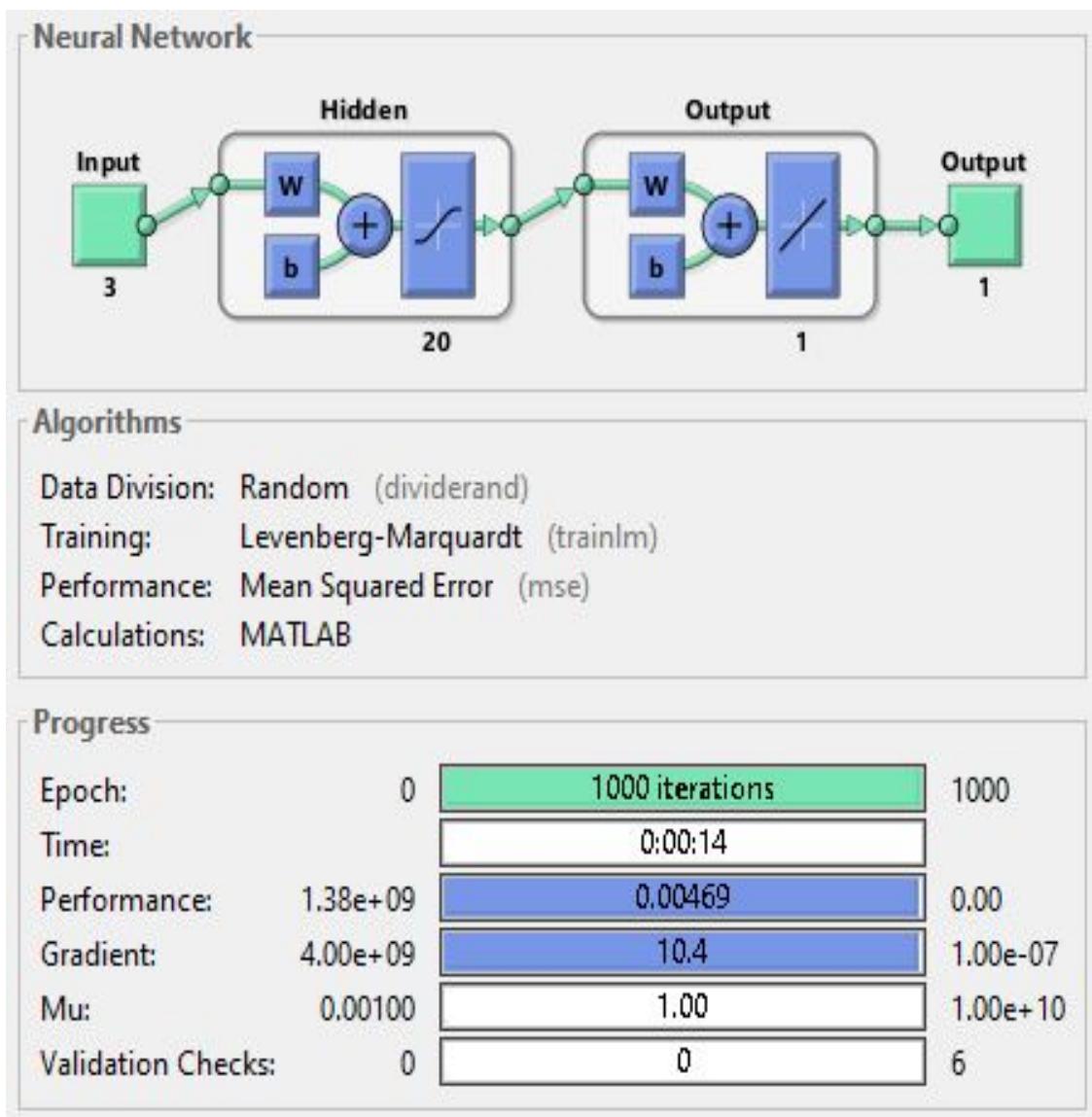


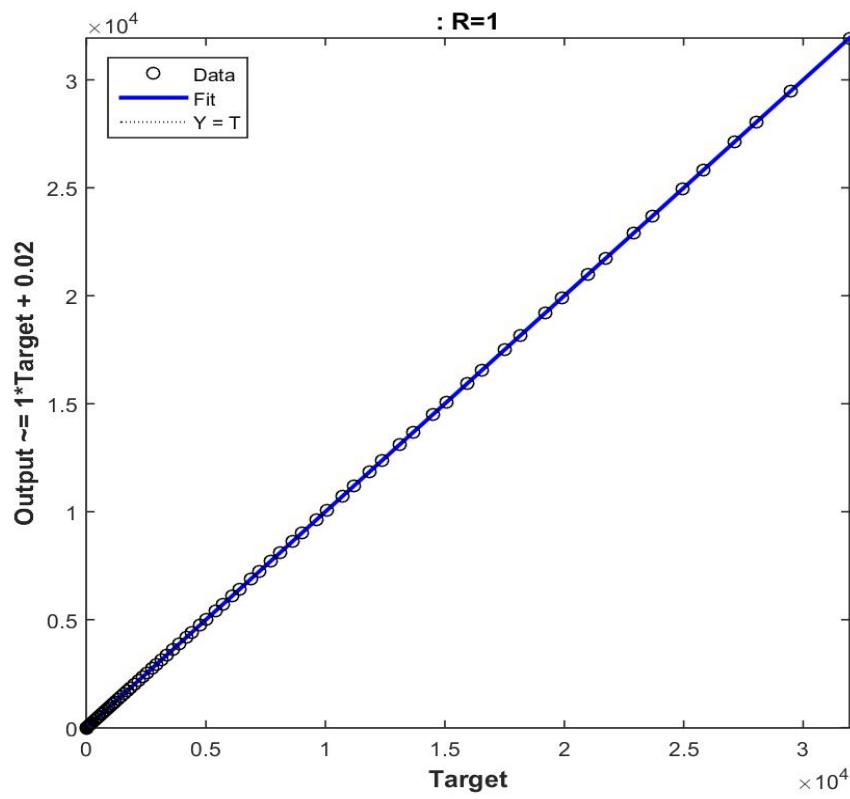
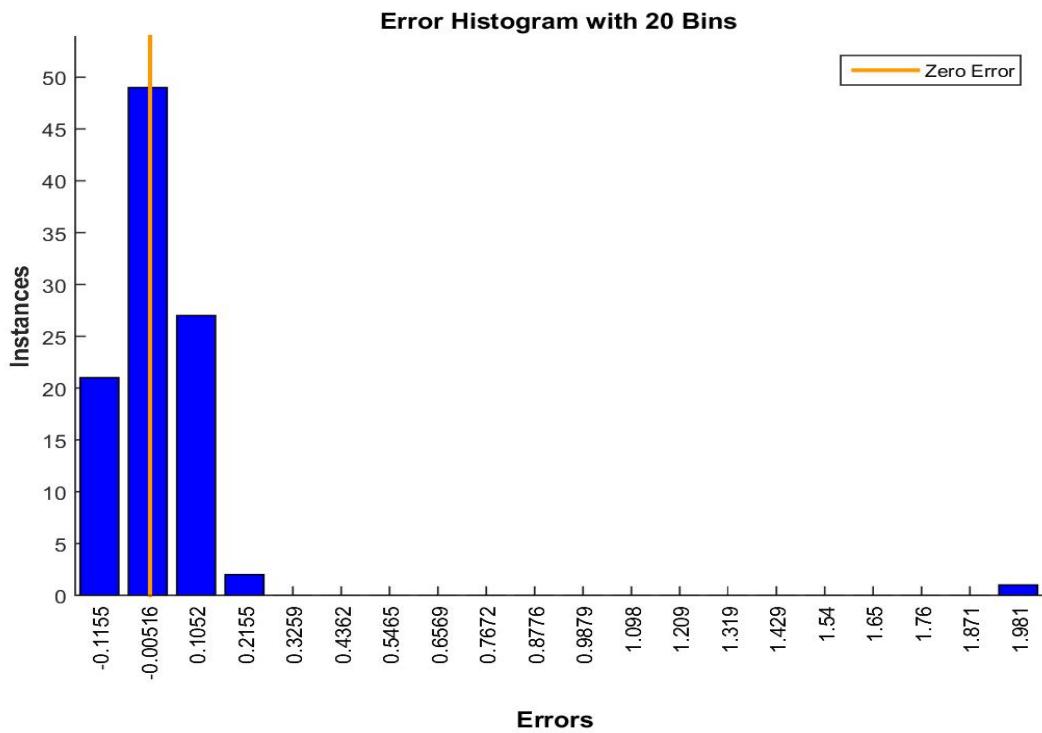


تابع پنجم:

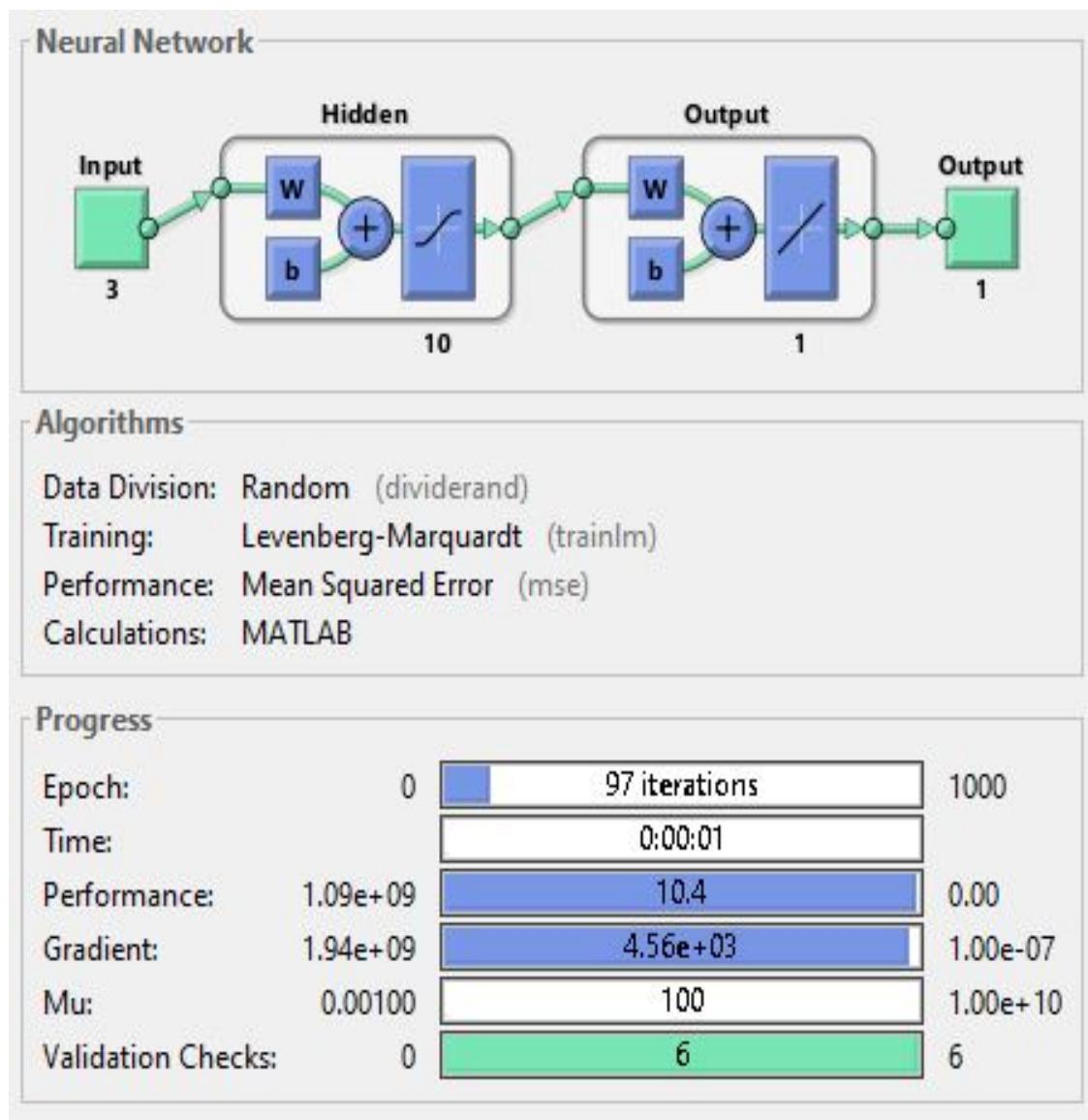
$$y_5 = 3 \cdot x_1^4 - 2 \cdot x_2^3 - 10 + 5 \cdot x_3 - 2;$$

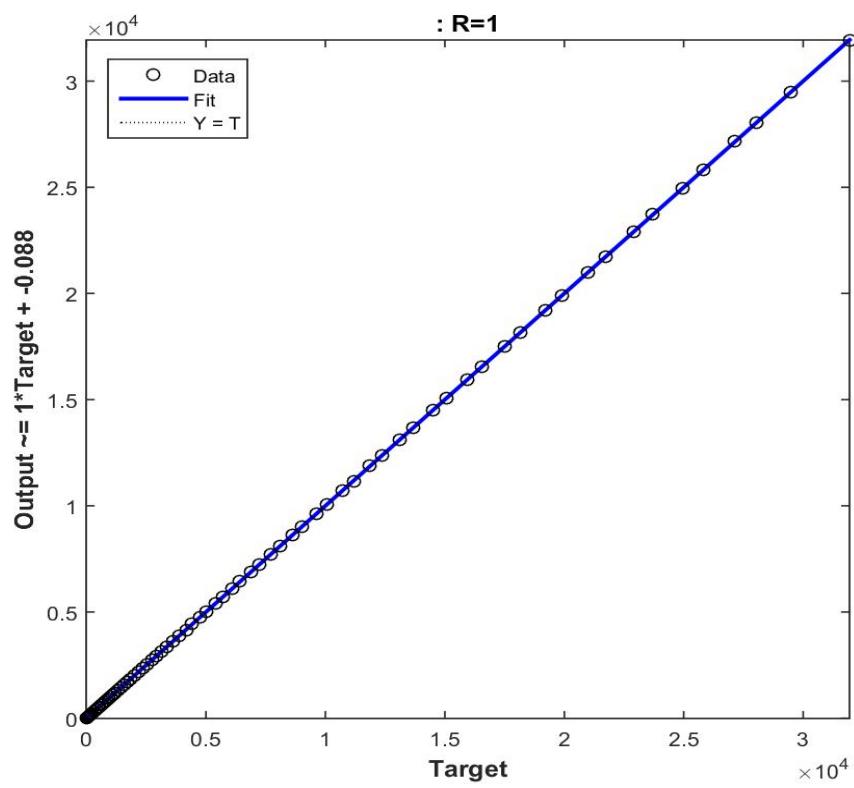
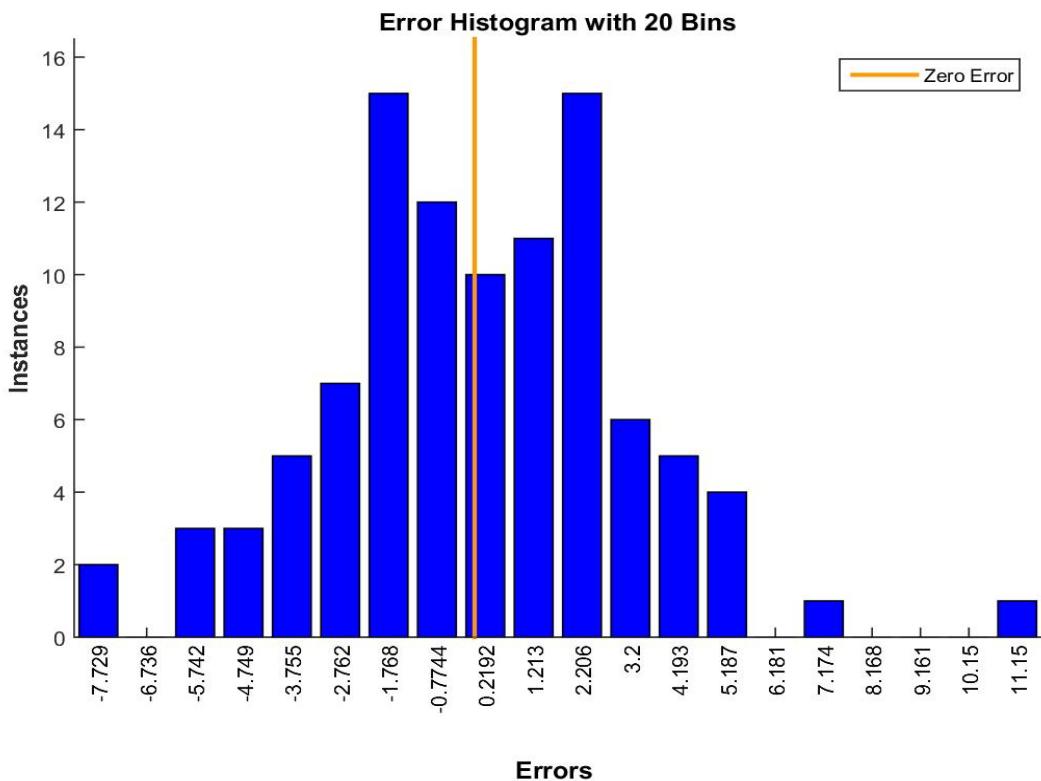
حالت اول:



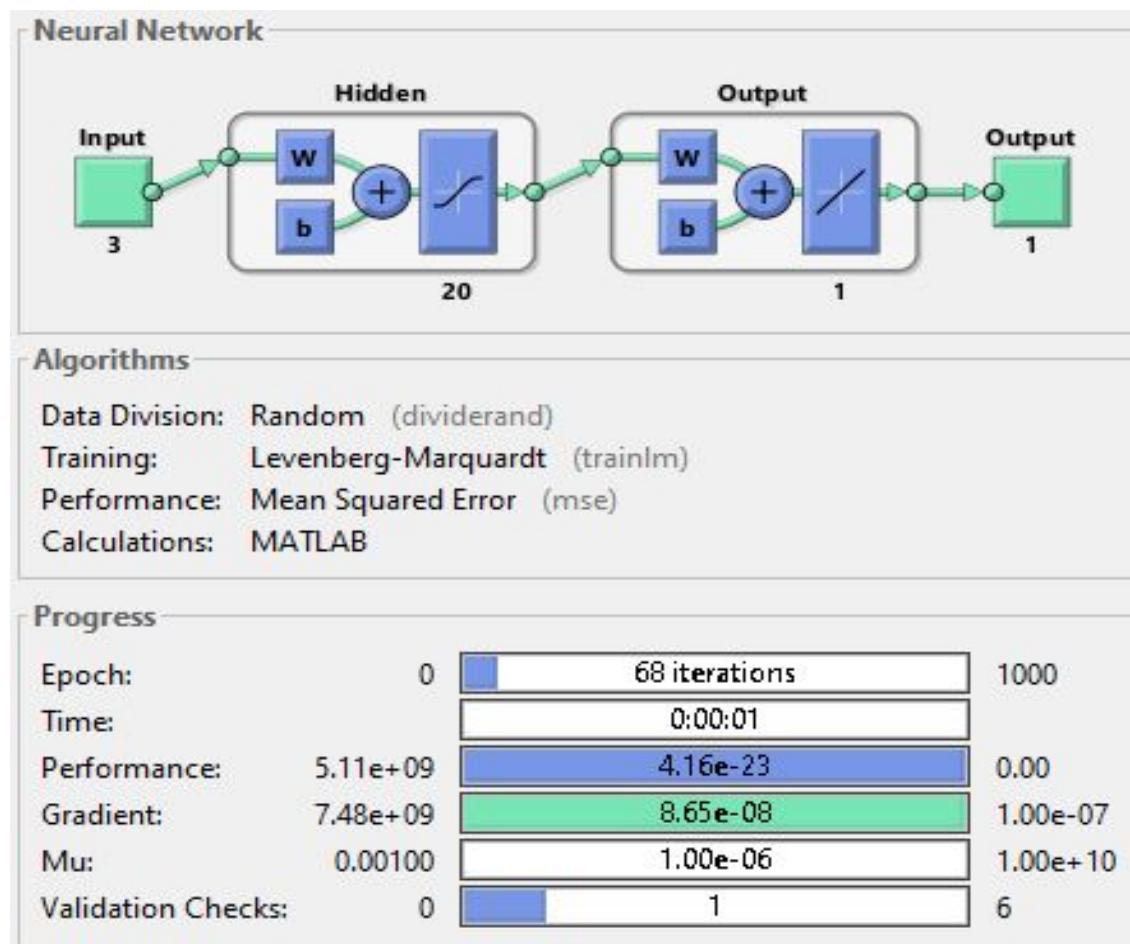


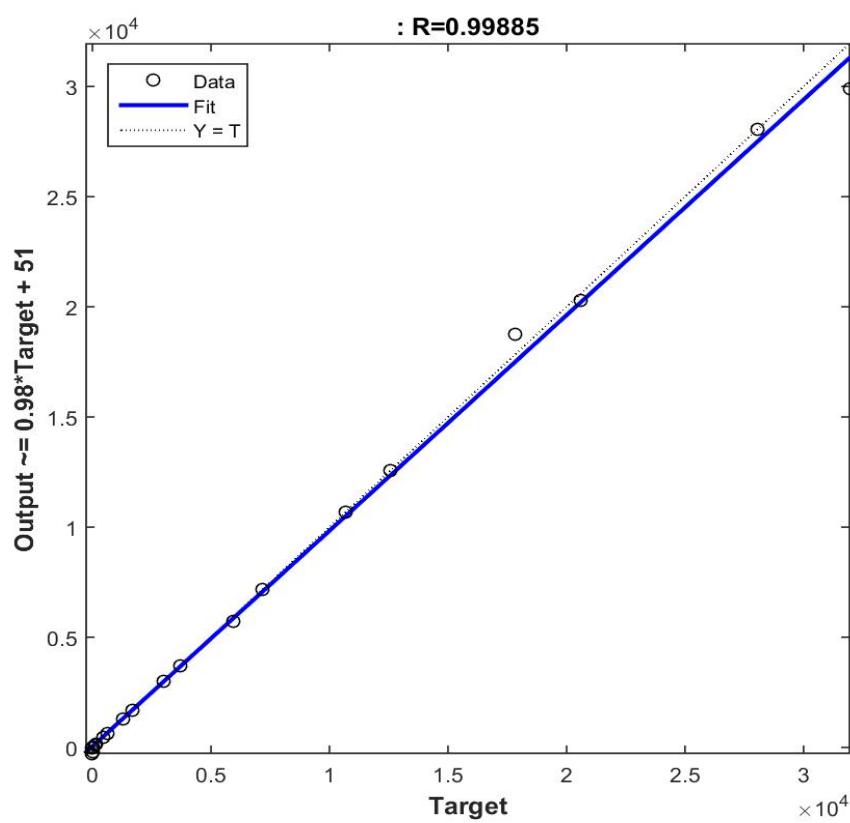
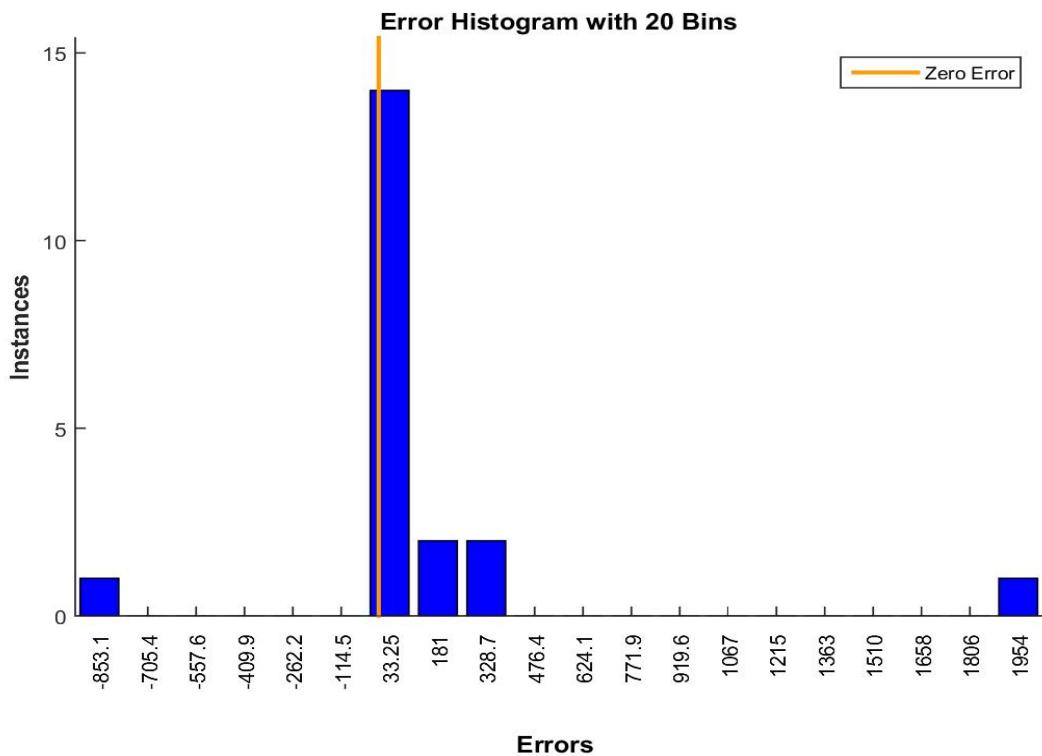
حالت دوم:



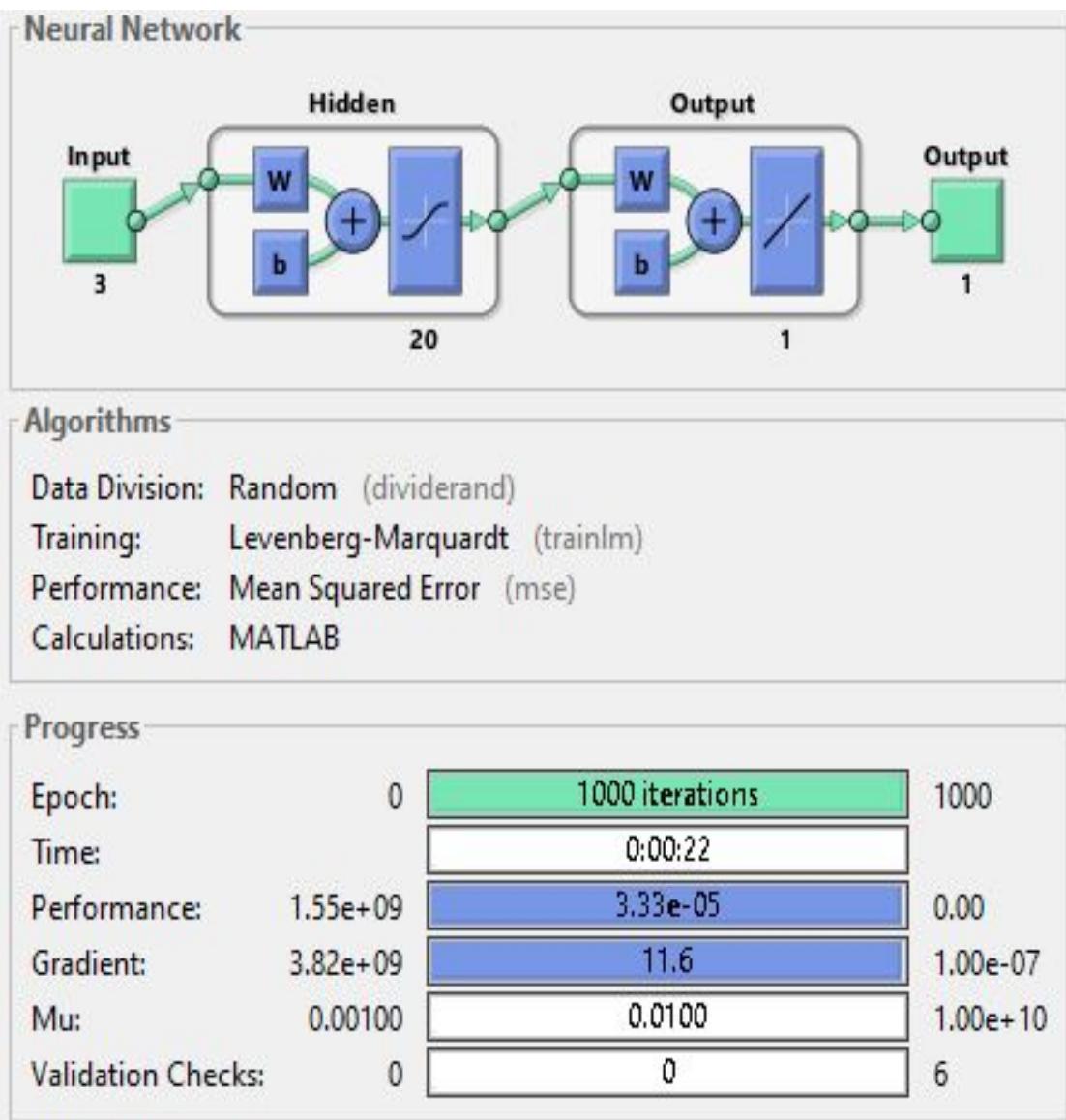


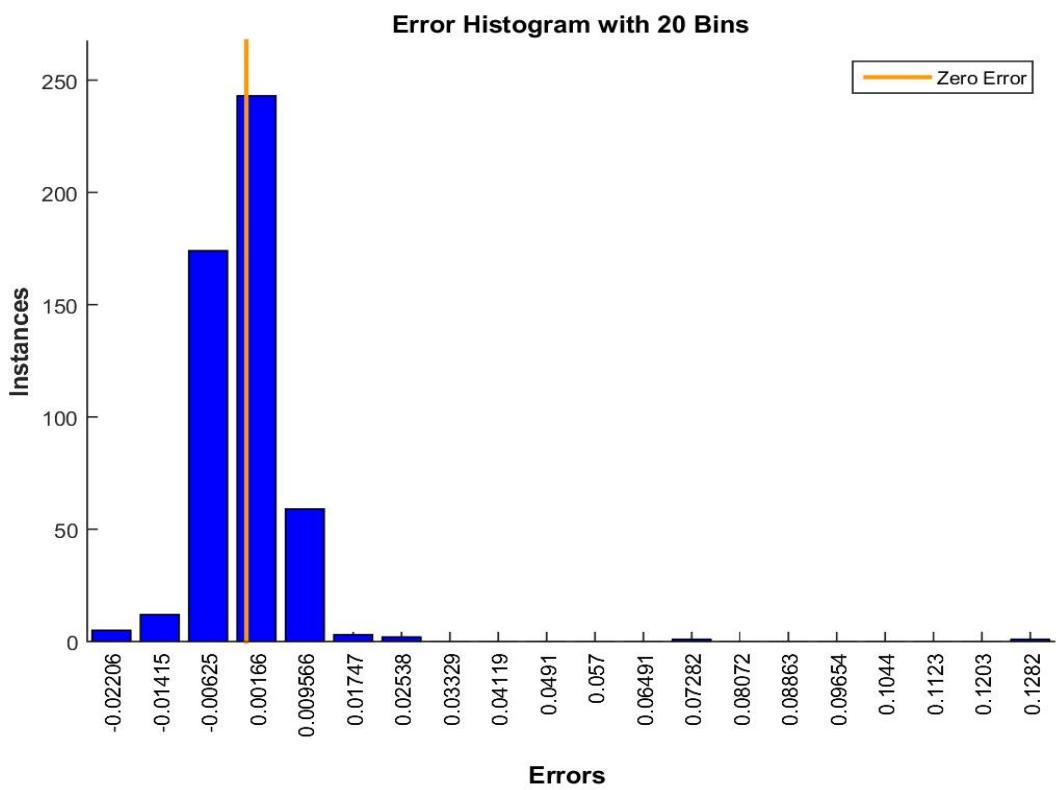
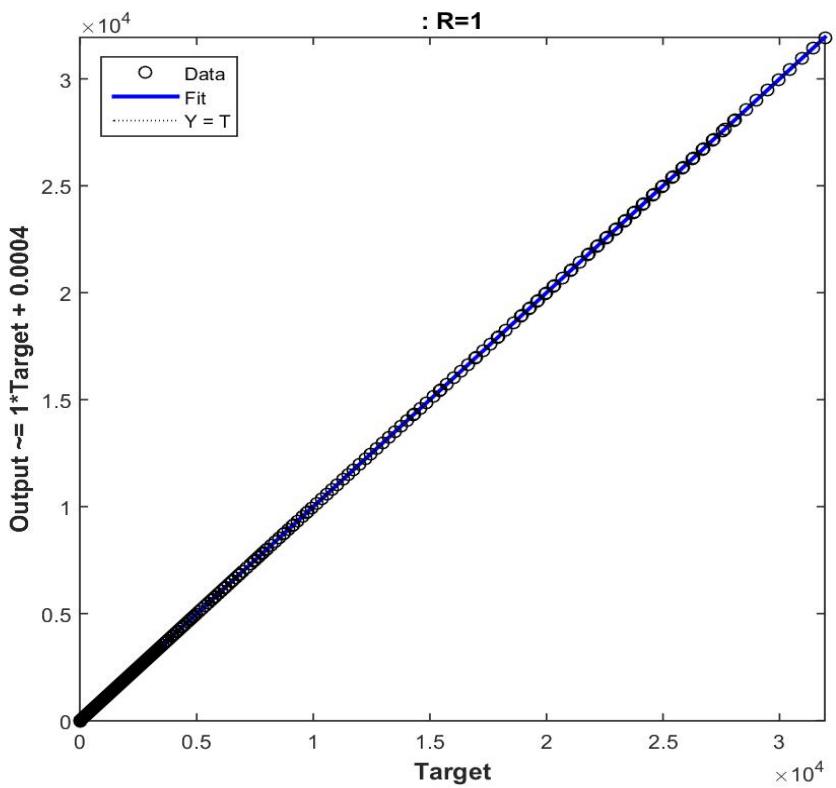
حالت سوم:





حالت چهارم:

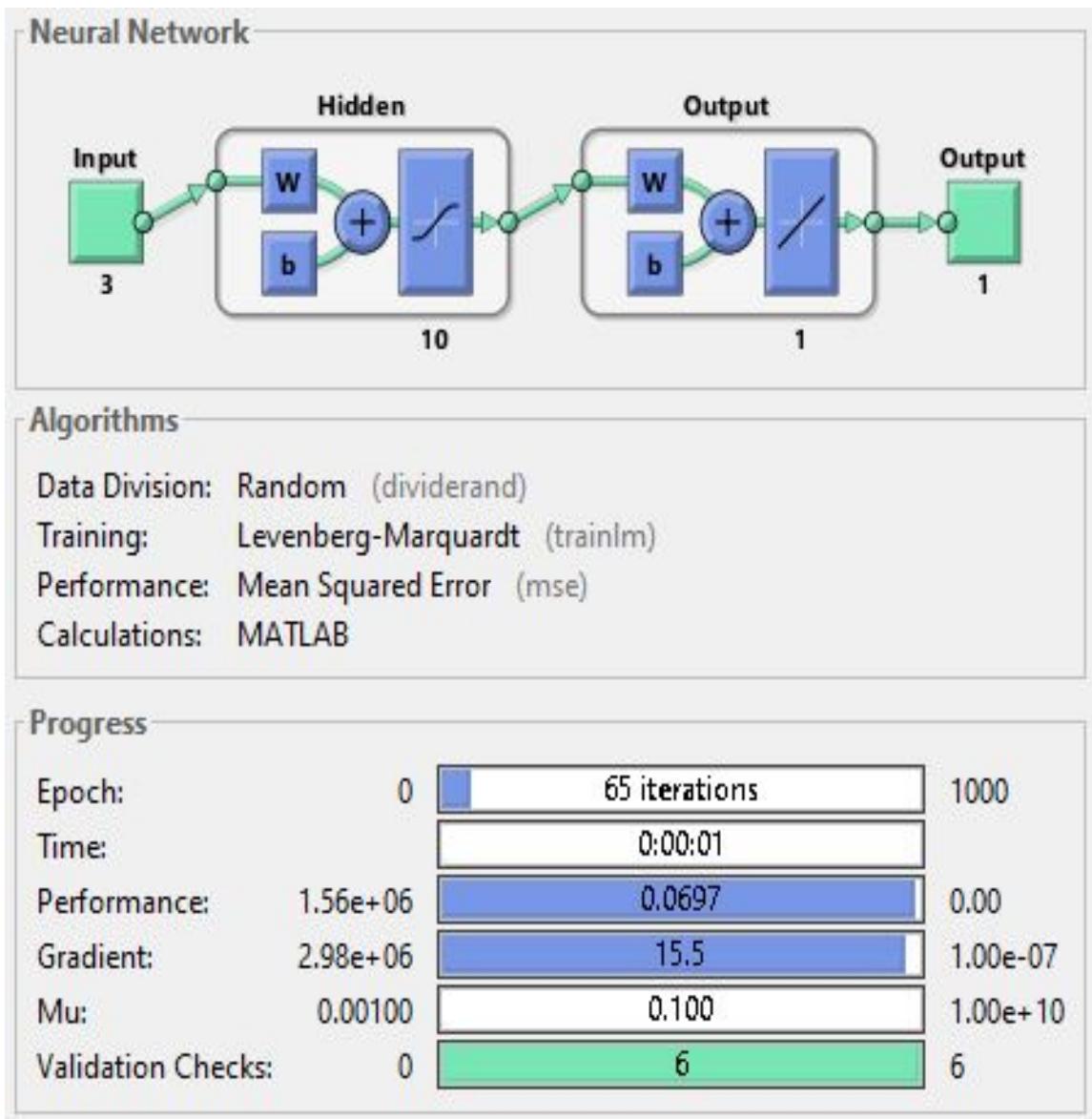


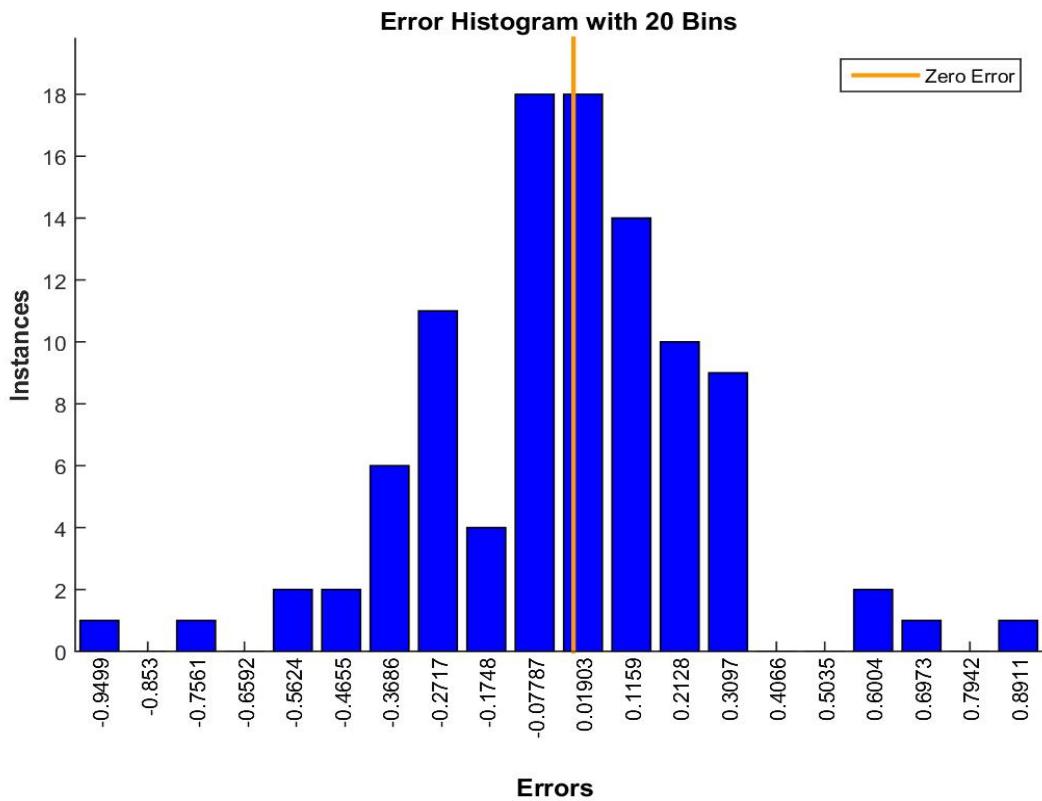
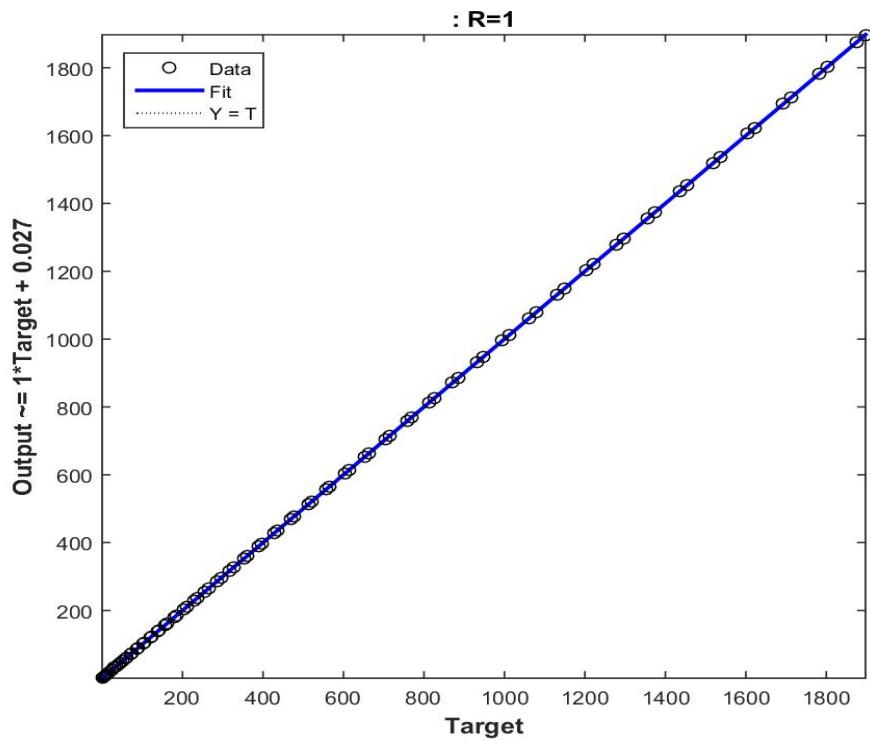


تابع ششم:

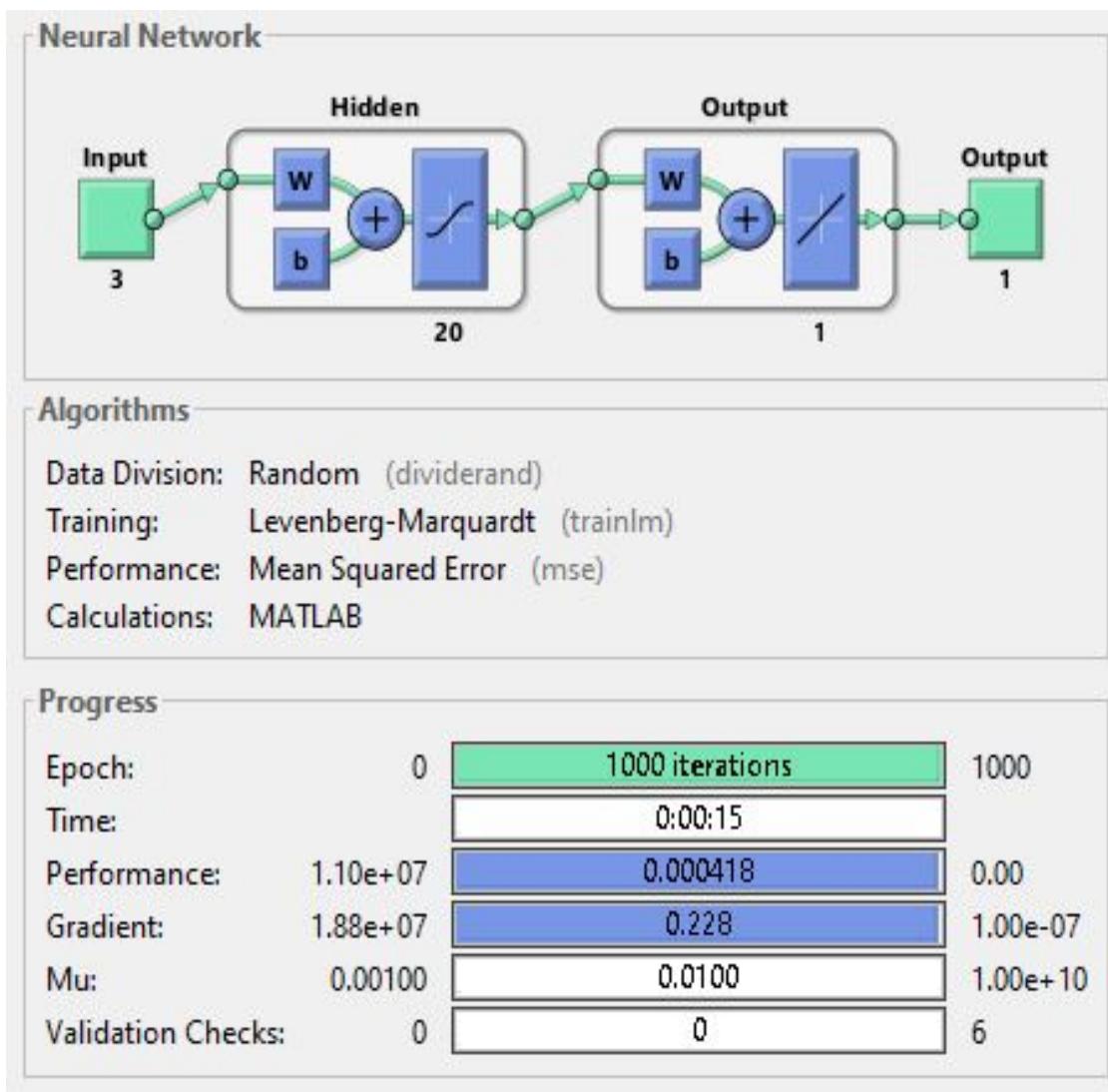
$$y_6 = \text{abs}(5 * \sin(x_1)^2 + 0.2 * x_3^2 - 6 * \sqrt{x_2^5});$$

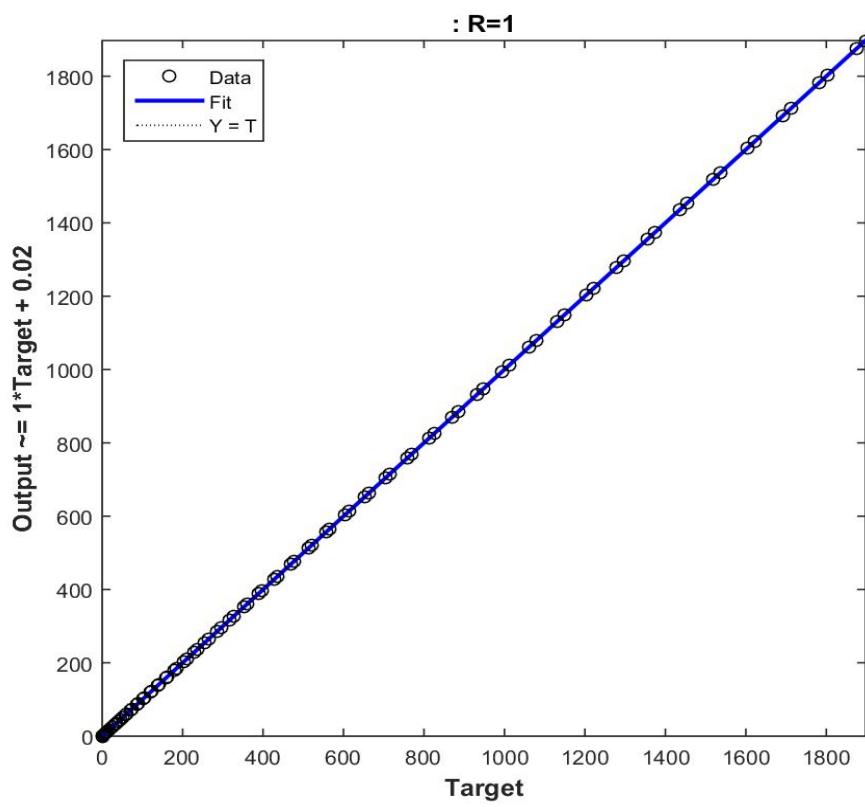
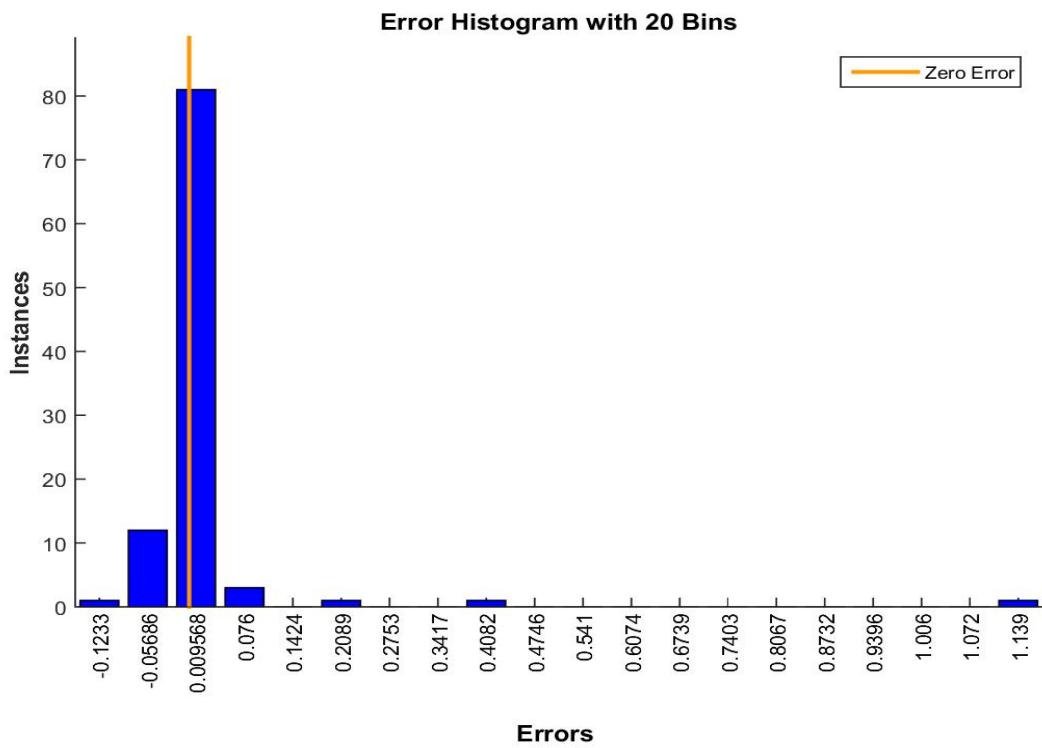
حالت اول:



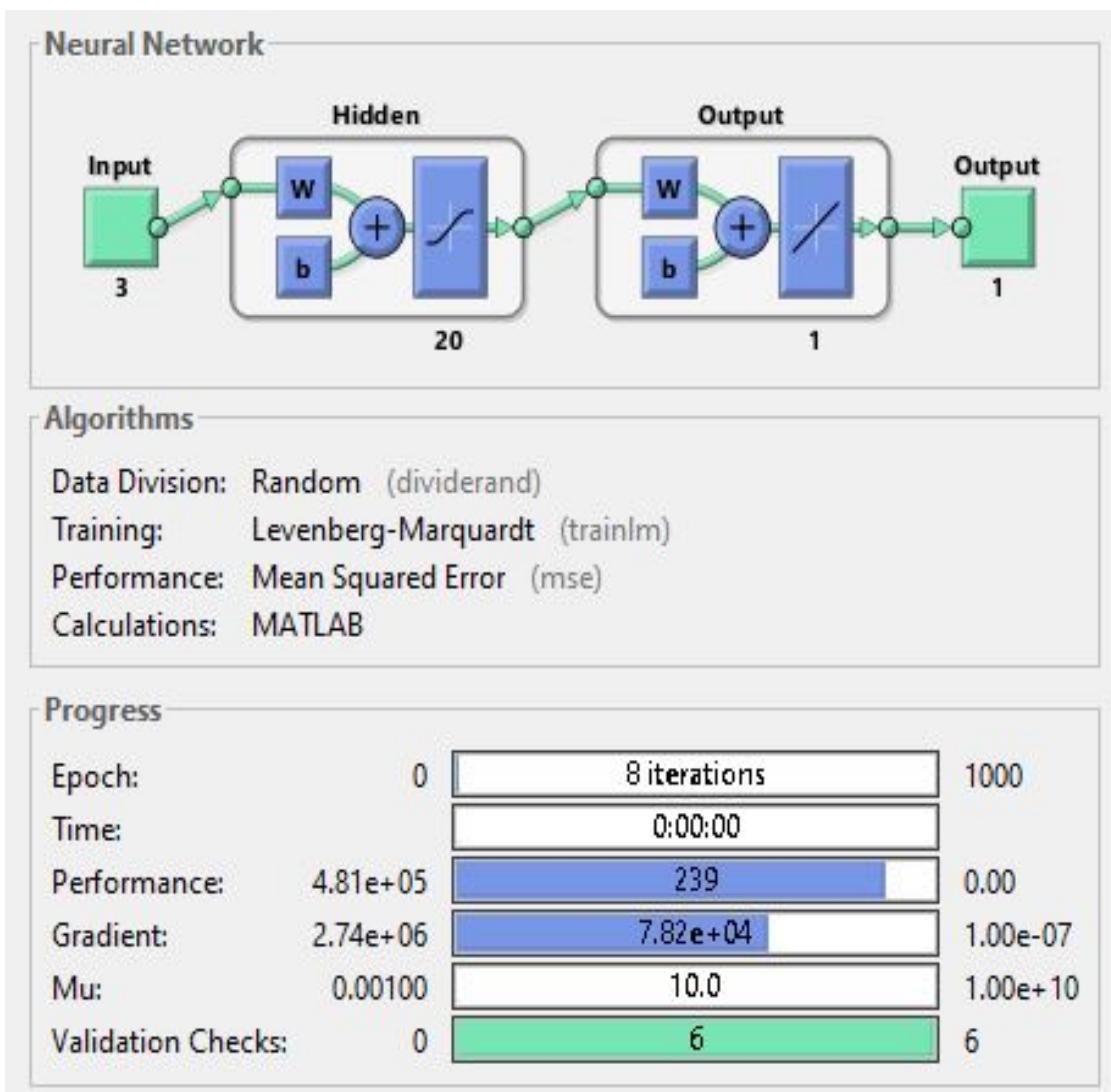


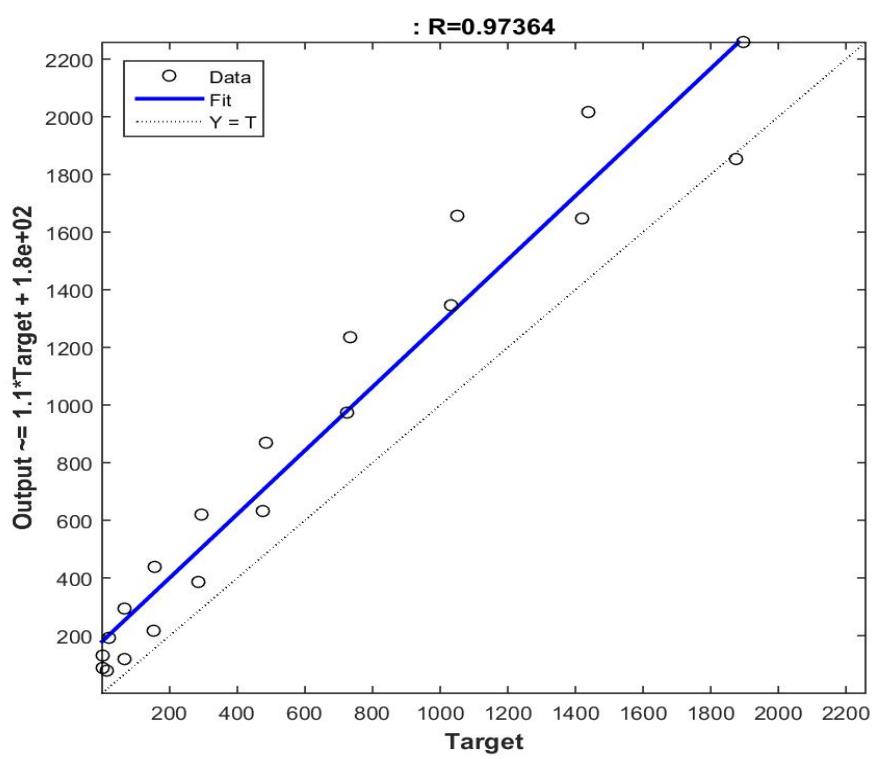
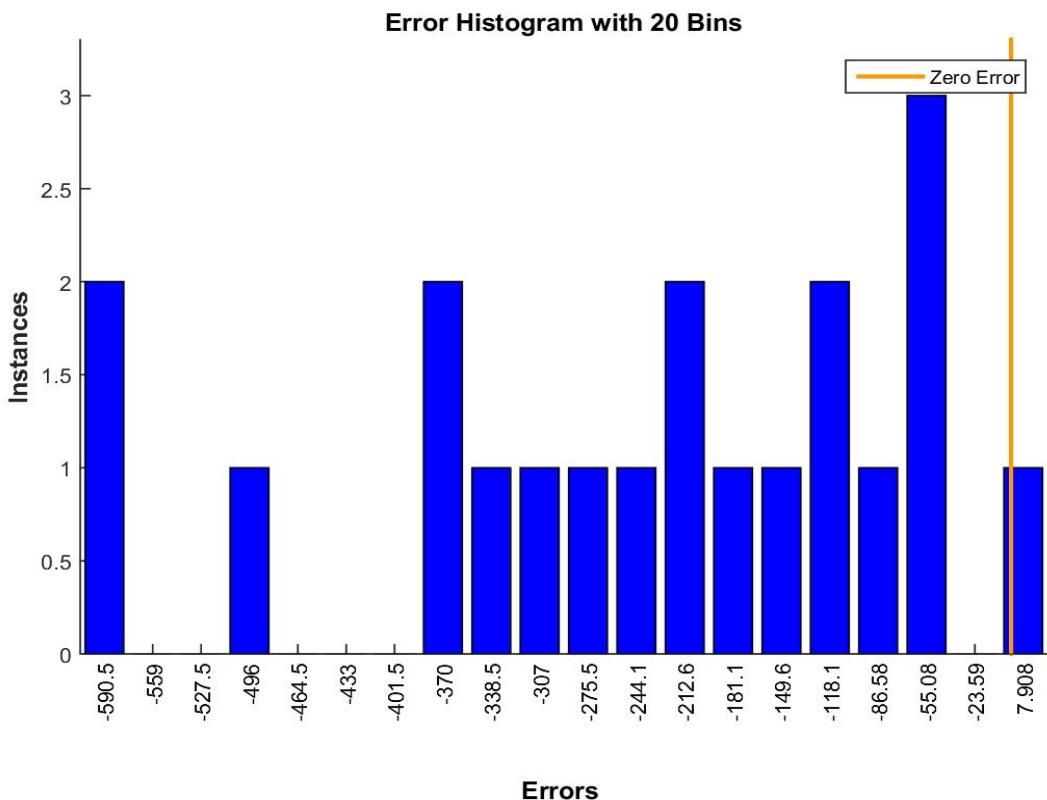
حالت دوم:



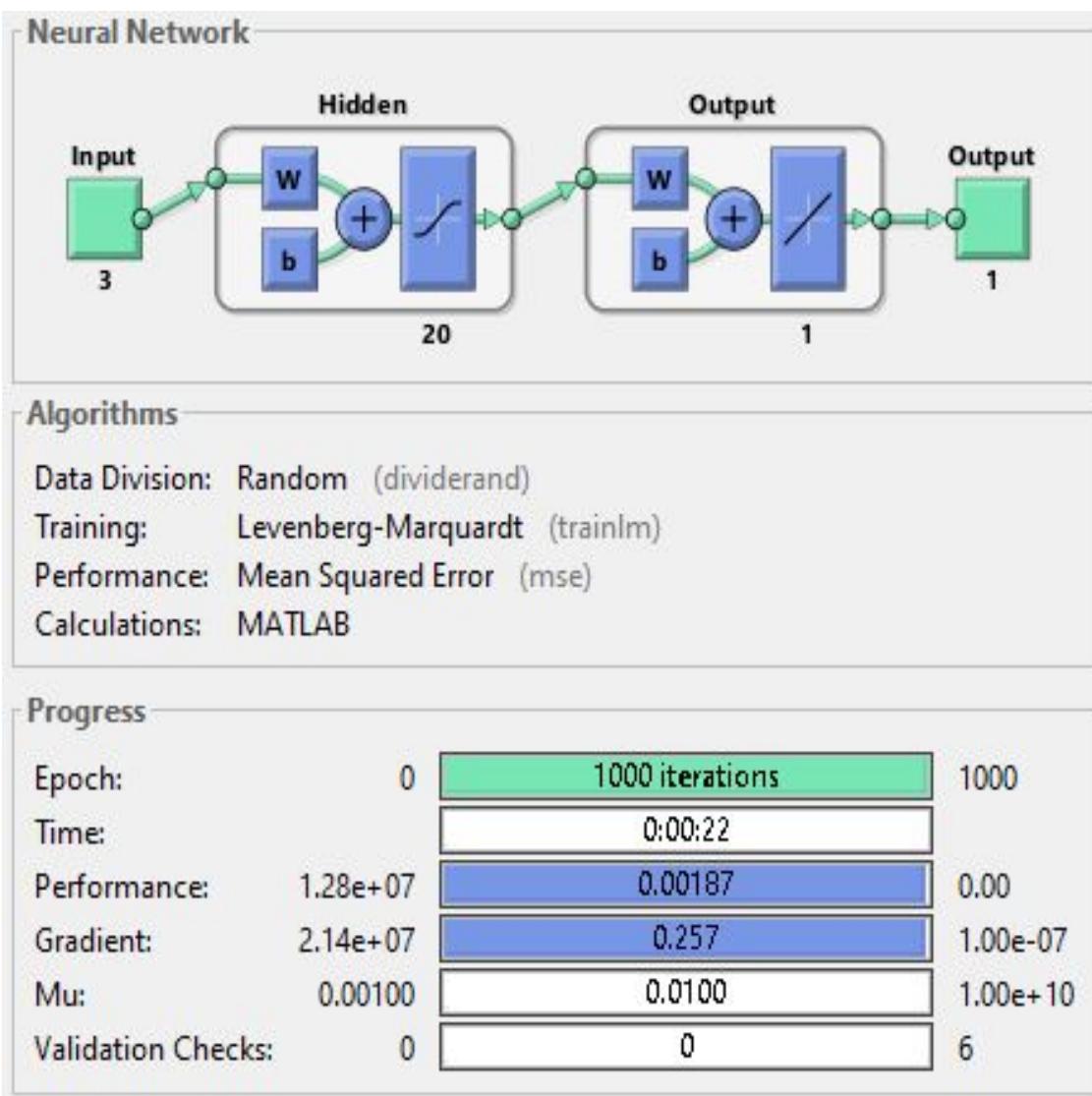


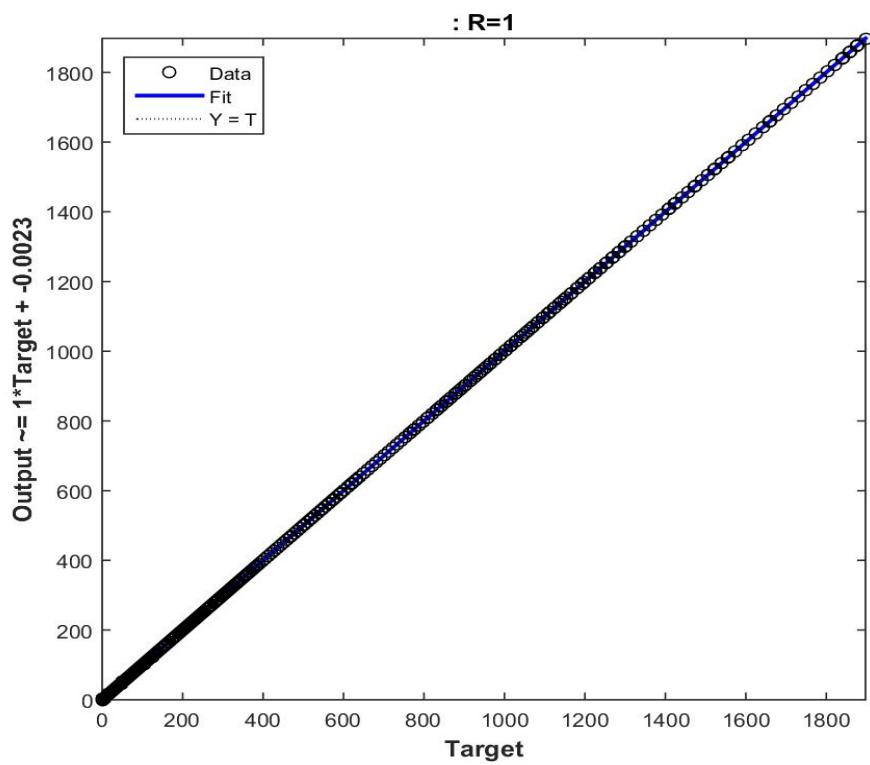
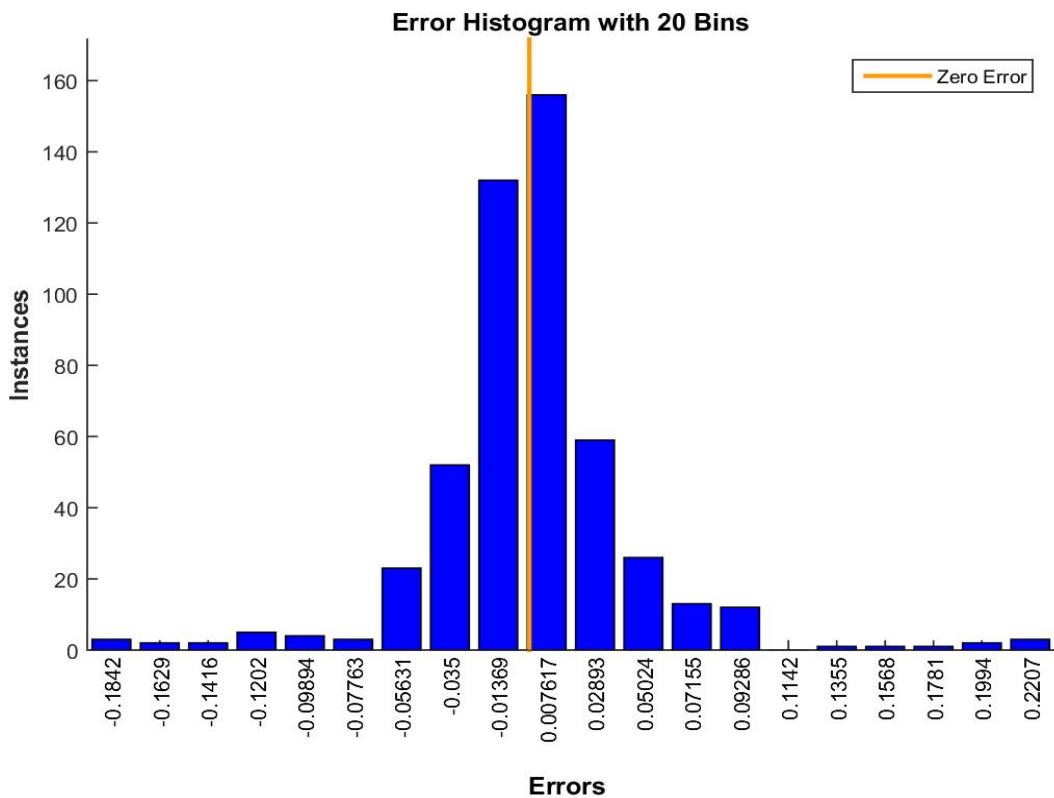
حالت سوم:





حالت چهارم:





نتیجه‌گیری: با توجه به بررسی حالت‌های مختلف شبکه عصبی برای توابع ذکر شده، می‌توان به نتایج زیر بی‌برد:

1. معمولاً با پیچیدگی توابع بایستی برای دستیابی به دقیقیت موردنظر تعداد نورون‌ها و تعداد لایه‌های پنهان را افزایش داد؛ اما لزوماً این که همیشه با افزایش تعداد لایه و تعداد نورون‌ها، دقیق نیز افزایش یابد درست نیست.
2. به طور کلی افزایش تعداد نمونه‌های آزمایش می‌تواند موجب افزایش دقیق گردد.

بخش چهارم:

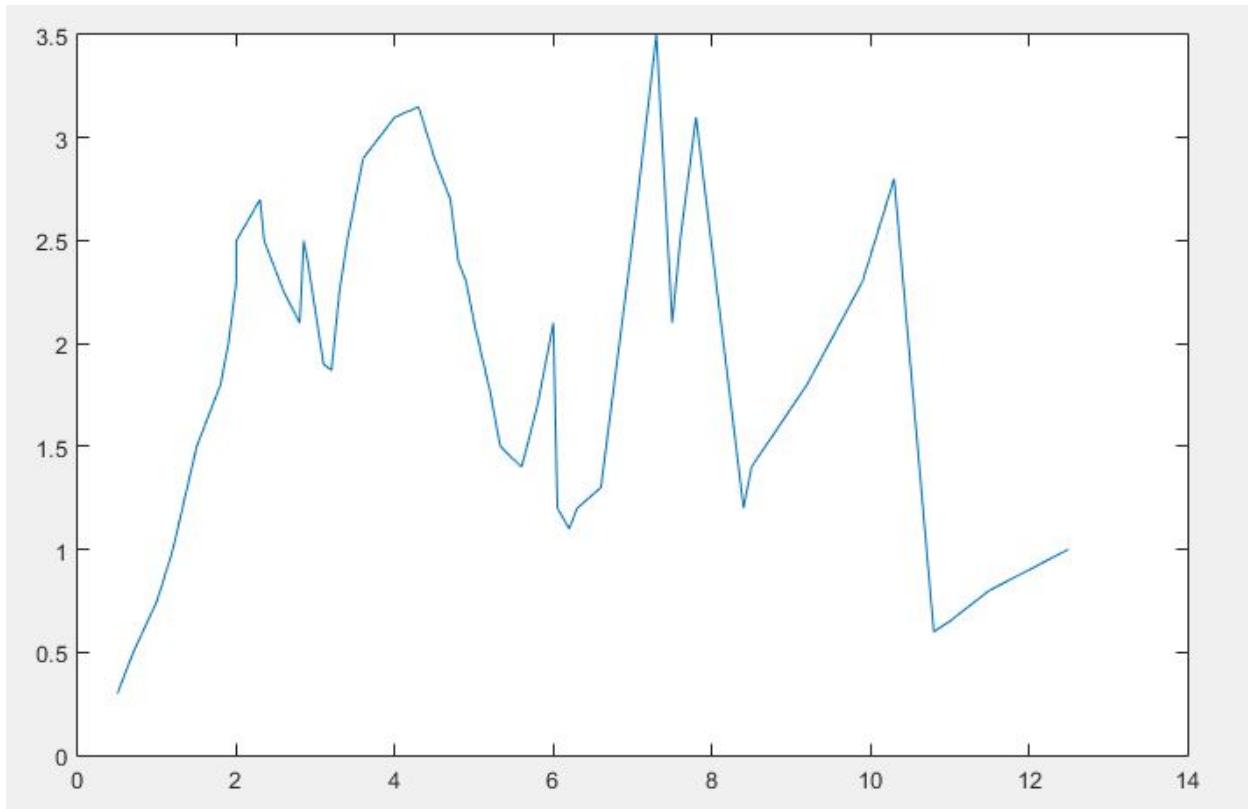
در این بخش نمودار خشم خود را توسط شبکه عصبی آموزش می‌دهیم. به این منظور ابتدا همانطور که خواسته شد، طرحی از خشم خود را در `paint` می‌کشیم. قسمت قرمز رنگ طبق تعریف پروژه نمودار خشم ماست. با شطرنجی نمودن نمودار می‌توان نقاط x و y منحنی را بدست آورد.



مختصات 55 نقطه استخراج شده است که x و y نقاط استخراج شده عبارتند از:

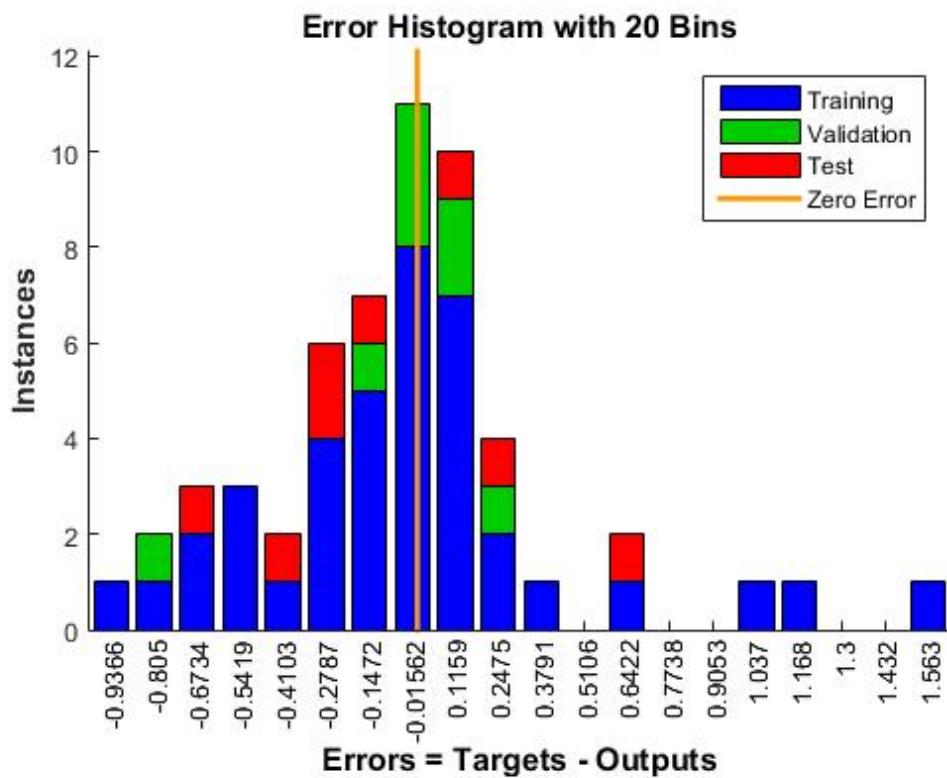
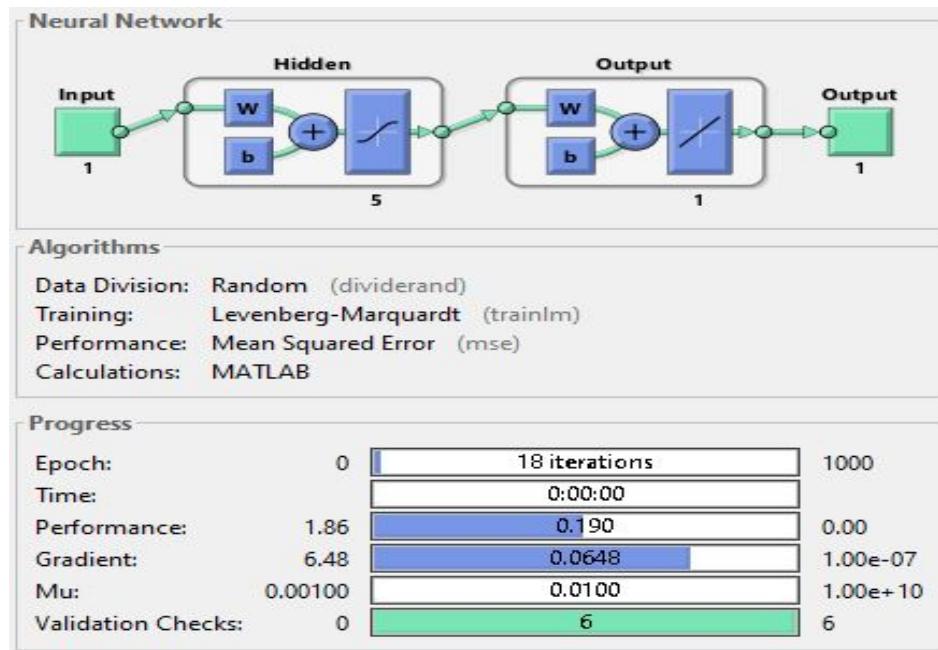
```
x=[0.5,0.7,1,1.2,1.5,1.8,1.9,2,2.,2.3,2.35,2.6,2.8,2.85,2.9,3,3.1,3.2,3.3,3.4  
 ,3.5,3.6,3.8,4,4.3,4.5,4.7,4.8,4.9,5,5.2,5.33,5.6,5.8,6,6.05,6.2,6.3,6.6,6.8,  
 7,7.3,7.5,7.6,7.8,8.4,8.5,9.2,9.9,10.3,10.8,11,11.5,12,12.5]  
y=[3.7,3.5,3.25,3,2.5,2.2,2,1.7,1.5,1.3,1.5,1.75,1.9,1.5,1.6,1.85,2.1,2.13,1.  
 75,1.5,1.3,1.1,1,0.9,0.85,1.1,1.3,1.6,1.7,1.9,2.23,2.5,2.6,2.3,1.9,2.8,2.9,2.  
 8,2.7,2.1,1.5,0.5,1.9,1.5,0.9,2.8,2.6,2.2,1.7,1.2,3.4,3.35,3.2,3.1,3]
```

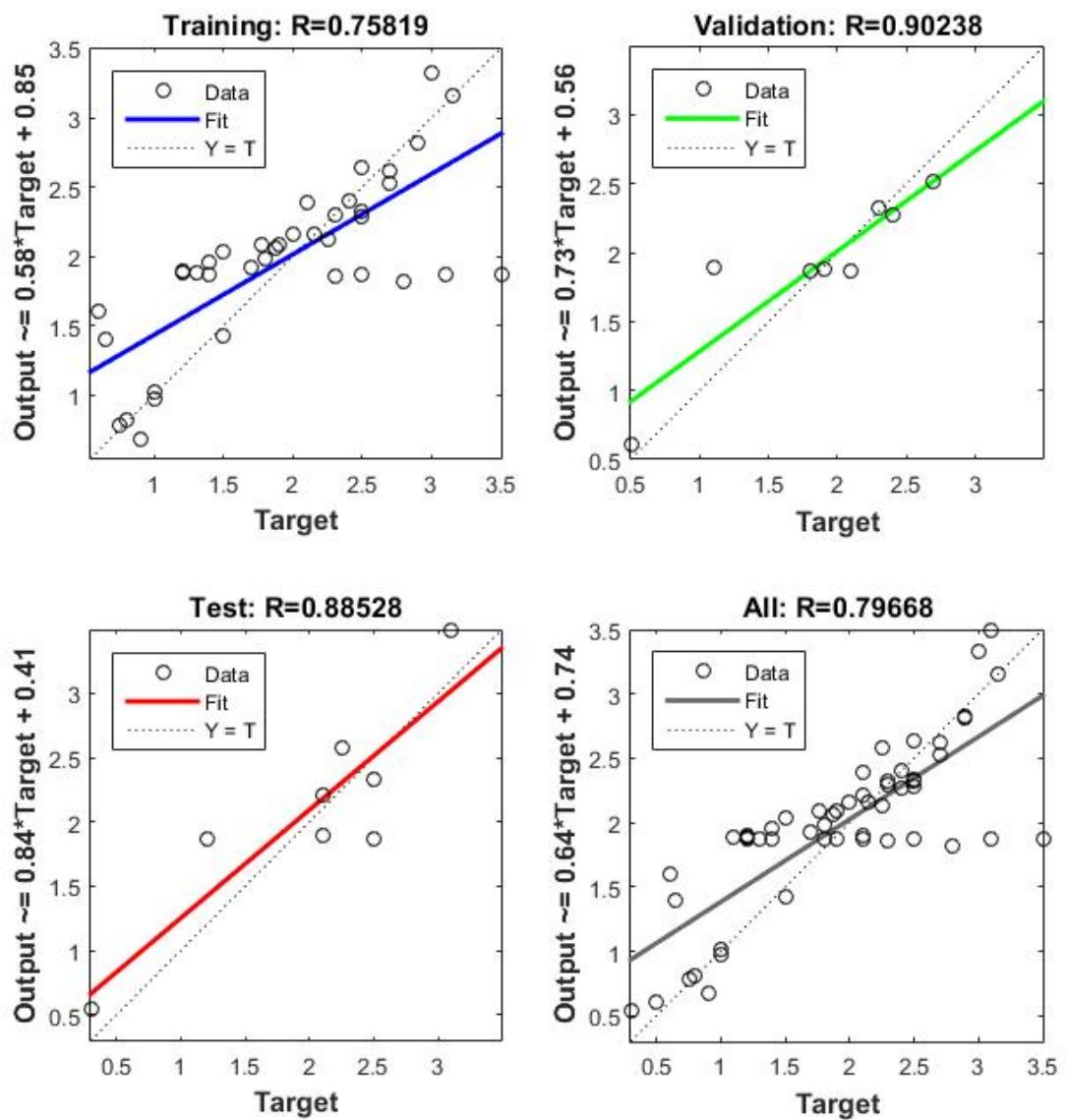
همانطور که مشاهده می‌شود مختصات نمودار y ها از بالا به پایین است. بنابراین مقادیر y نمودار خشم به ازای تمامی x ها برابر با 4- در نظر گرفته می‌شود. نتیجه نمودار استخراج شده از متلب در صفحه بعد نمایش داده شده است:

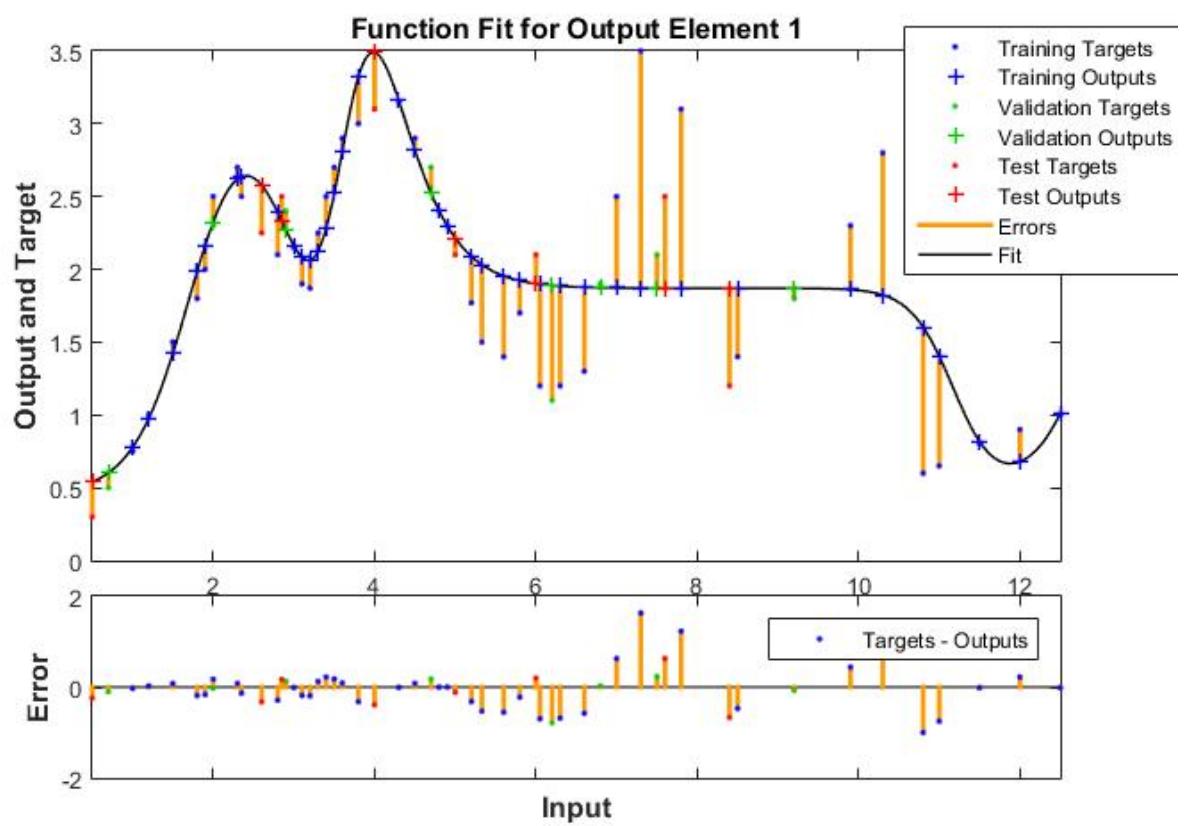


بنابراین تابعی که باید آموزش داده شود به شکل بالا خواهد بود.
مقادیر x و y عنوان ورودی و مقادیر هدف تابع به شبکه عصبی اعمال می‌شود. در ادامه چند حالت از ساختارهای شبکه عصبی و نتایج ارائه می‌شوند.

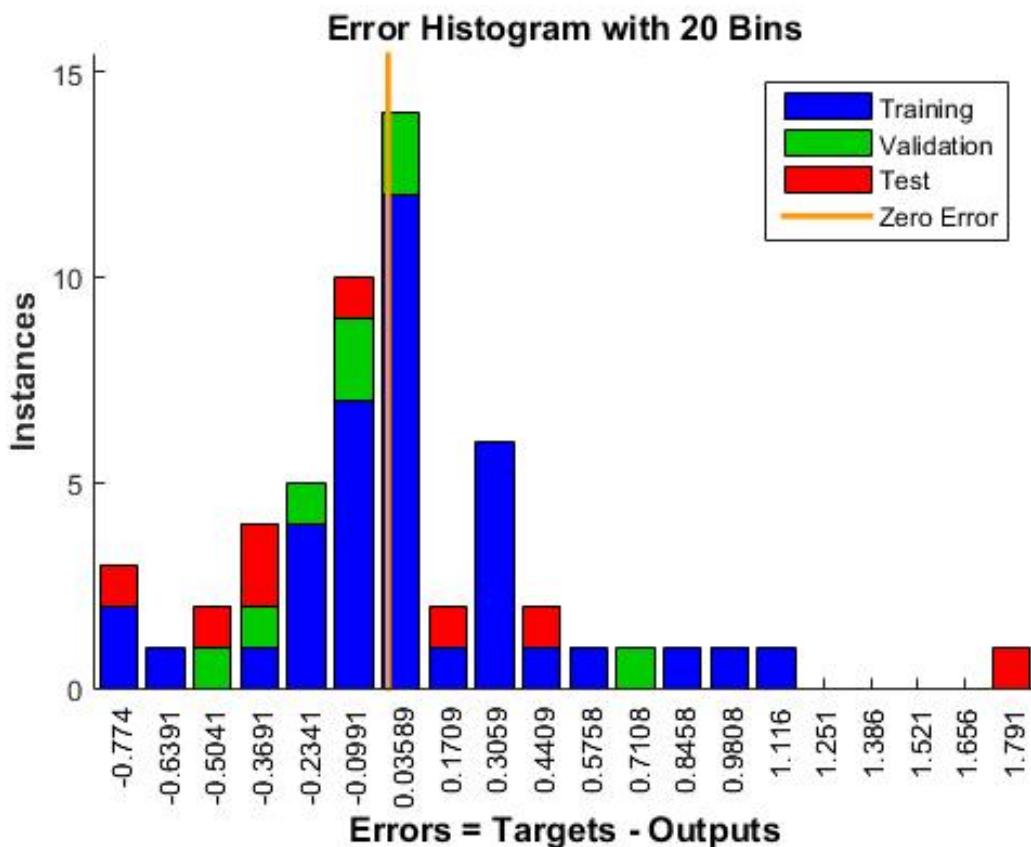
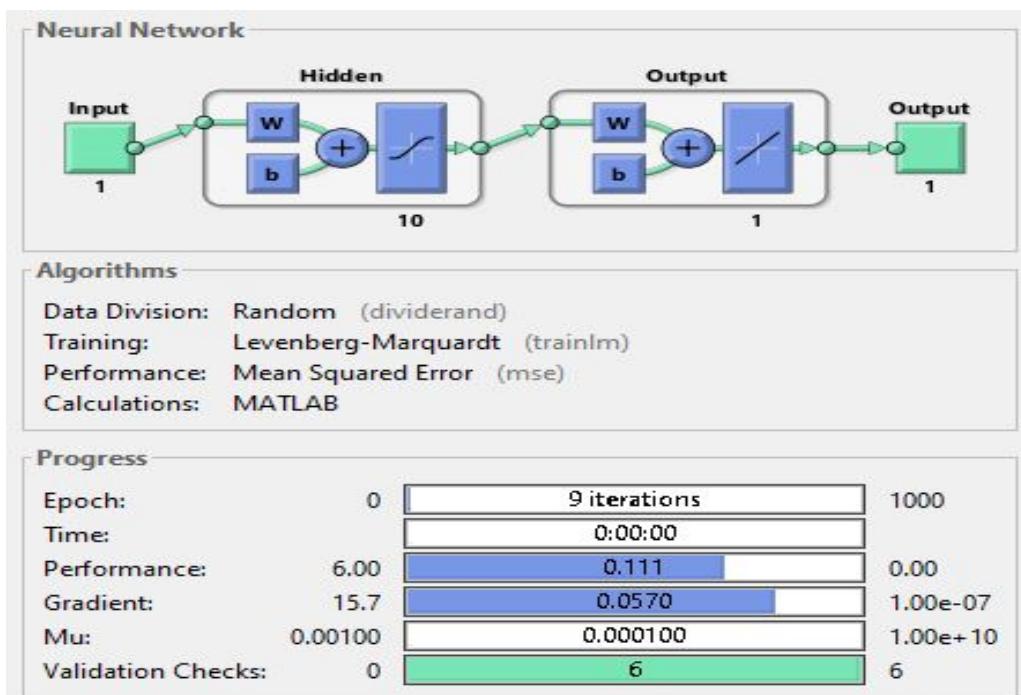
حالت اول: در این حالت یک لایه مخفی با ۵ نورون در نظر گرفته شده است.

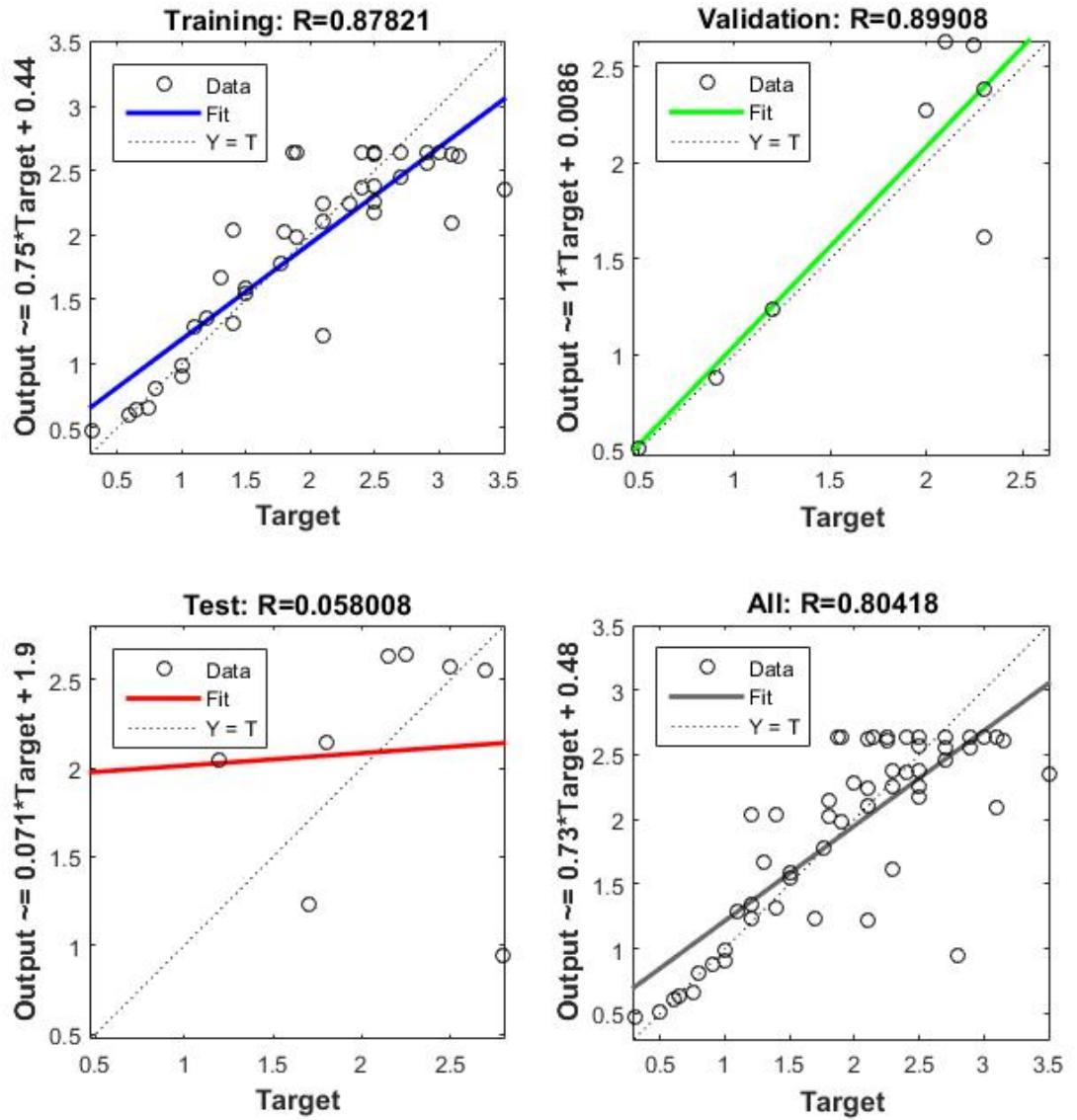


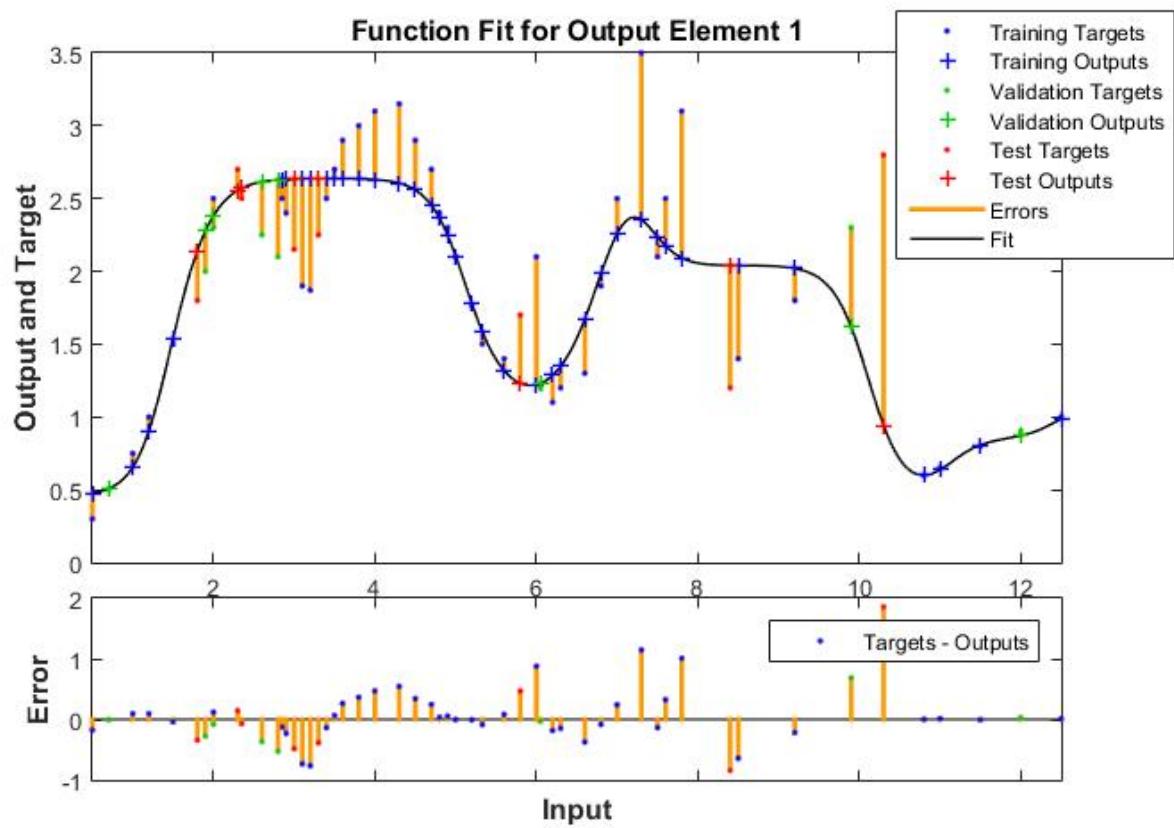




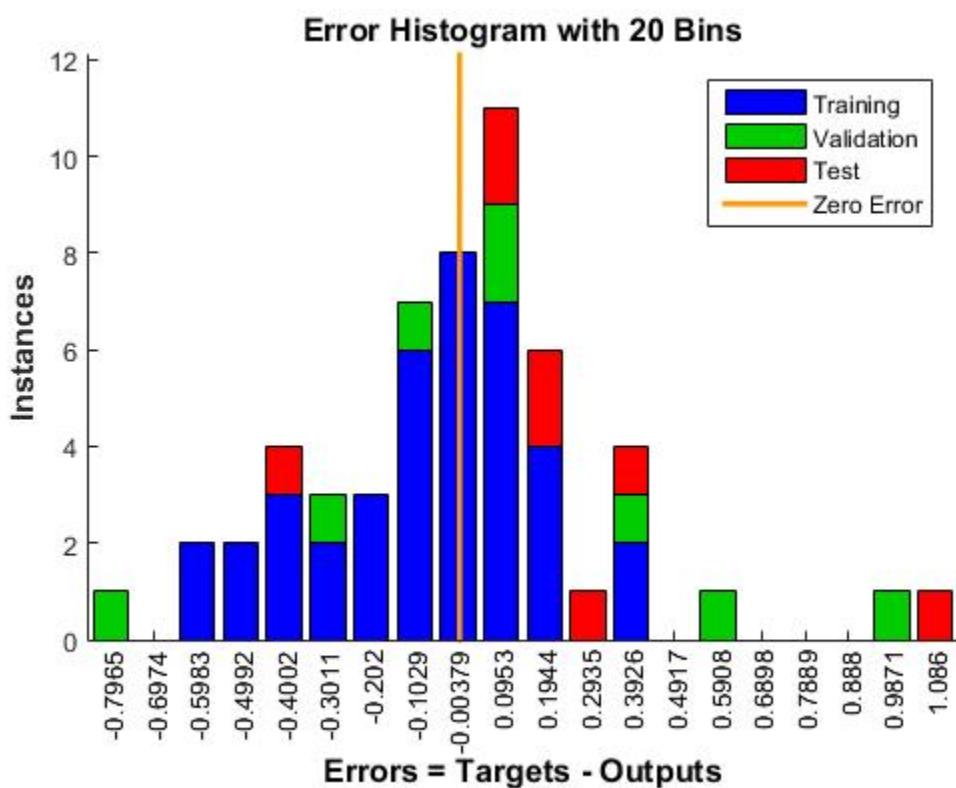
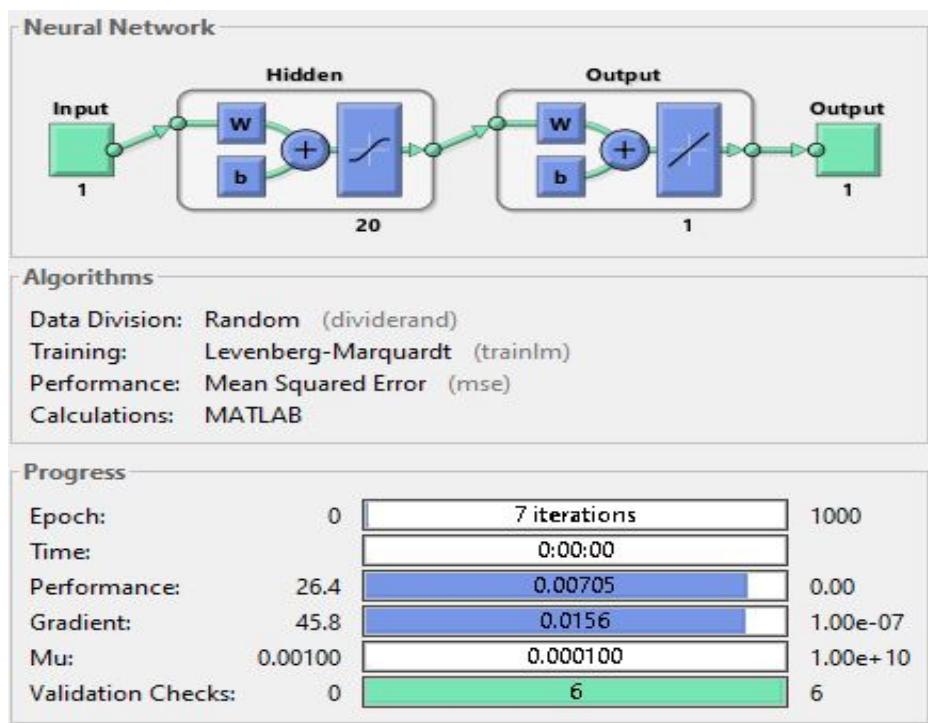
حالت دوم: در حالت دوم تعداد نورون‌های لایه پنهان ۱۰ در نظر گرفته شده است؛

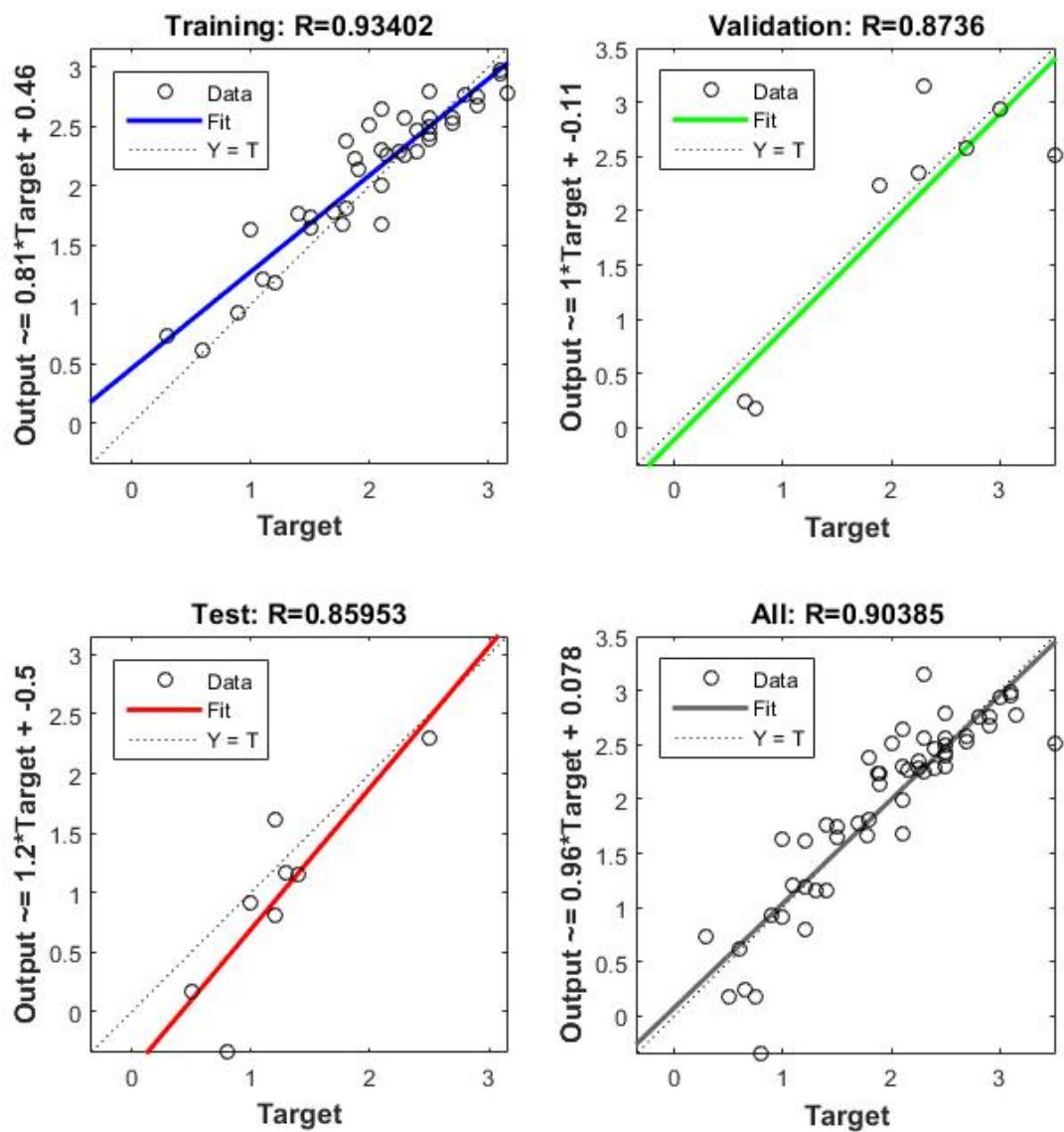


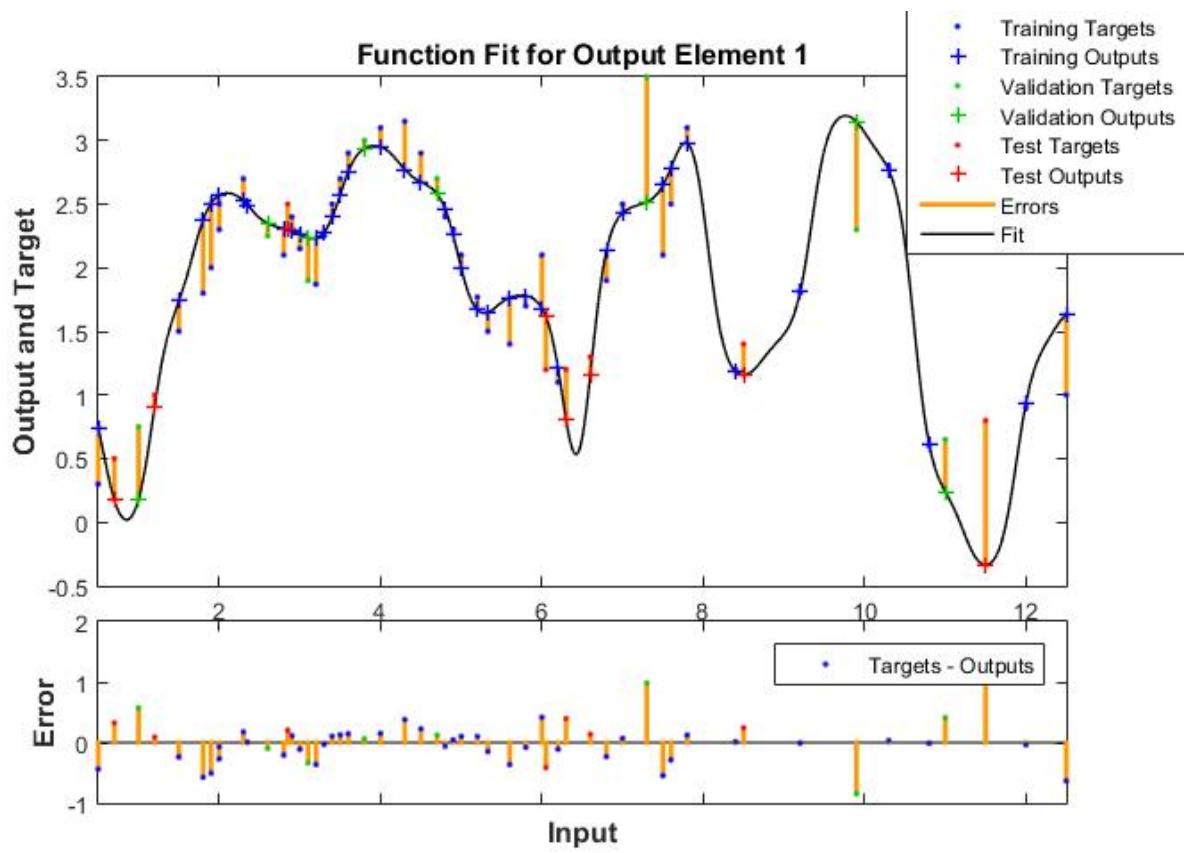




حالت سوم: در این حالت تعداد نمونه‌های آموزش برابر با ۲۰ در نظر گرفته می‌شود.







نتیجه‌گیری: با توجه به بررسی حالت‌های مختلف شبکه عصبی برای تابع خشم معرفی شده، می‌توان به نتایج زیر پی‌برد:

۱. در موارد اول و دوم نتایج مناسب نیست و شکل خروجی از شبکه عصبی خیلی متفاوت است با نمودار اصلی. اما در حالت سوم با ۲۰ نورون در لایه مخفی، مقادیر خطأ کمتر است اما همچنان با نمودار اصلی فاصله دارد.
۲. مهمترین عاملی که باعث می‌شود شبکه عصبی نتواند بخوبی نمودار اصلی را تخمین بزند، وجود پرسش‌های ناگهانی نمودار خشم است. همان طور که در بخش اول نیز با آن مواجه شده بودیم، در تابع چهارم بخارط وجود پرسش‌ها، تقریب مناسبی از تابع بدست نمی‌آمد.

بخش پنجم:

در این بخش کاربرد شبکه عصبی در پردازش تصویر بررسی می شود. به این منظور از دیتاست MINIST استفاده شده است که یکی از شناخته شده ترین دیتاست هایی است که برای کاربردهای ابتدایی پردازش تصویر استفاده می شود. مجموعه ای که در نرم افزار بارگذاری می شود شامل ۶۰۰۰۰ نمونه آموزش و ۱۰۰۰ نمونه تست است. عکس هایی که در آن اعداد ۰ تا ۹ کشیده شده است. در این بخش سعی می شود شبکه عصبی آموزش داده شود تا بتواند این اعداد را تشخیص بدهد.

بعد از فراخوانی دیتاست تصاویر بصورت خاکستری، تصاویر بصورت ماتریس ۲۸ در ۲۸ ذخیره می شود که برای آموزش به شبکه عصبی نیاز هست که تبدیل به بردار شوند. بنابراین هریک از تصاویر در محیط متلب به بردارهایی با ۷۸۴ درایه تبدیل می شوند. برای نرمال سازی این بردارها، تمام درایه های آنها را بر ۲۵۵ تقسیم می کنیم تا بین ۰ و ۱ نرمال شوند. خروجی مساله نیز عدد مربوط به عکس است. بنابراین قرار است تصاویر در ۱۰ گروه دسته بندی شوند.

با توجه خروجی های مساله که اعداد ۰ تا ۹ هستند، آموزش شبکه عصبی ۳ روش وجود خواهد داشت:

۱. اینکه همین اعداد ۰ تا ۹ بعنوان خروجی هر تصویر در نظر گرفته شود.
۲. اینکه برای هر تصویر یک بردار ۱۰ عددي تهیه شود که تصویر هر عددی را نشان می داد درایه عدد آن تصویر برابر ۱ و بقیه درایه ها برابر صفر شوند

برای مثال بردار خروجی متناسب با تصویر زیر برابر است با:

[۰ ۱ ۰ ۰ ۰ ۰ ۰ ۰ ۰ ۰]'



که دومین درایه آن ۱ است. برای تصاویر عدد ۰، درایه ۱۰ آنها ۱ و بقیه صفر خواهد بود.

در این حالت شبکه عصبی ۷۸۴ ورودی و ۱۰ خروجی خواهد داشت.

۳. در این روش برای هر گروه یک شبکه عصبی آموزش داده می شود که در مجموع ۱۰ شبکه عصبی خواهیم داشت. به این صورت که اگر تصویر مثلا عدد ۱ را نشان میداد، همان خروجی تصویر برای آموزش گروه ۱ برابر ۱ است و خروجی آن برای بقیه گروه ها برابر در نظر گرفته می شود. بعد از تعیین خروجی مربوط به هر گروه، شبکه های عصبی هر گروه با ورودی های یکسان و خروجی های متعلق به آن گروه بطور جداگانه آموزش داده می شوند.

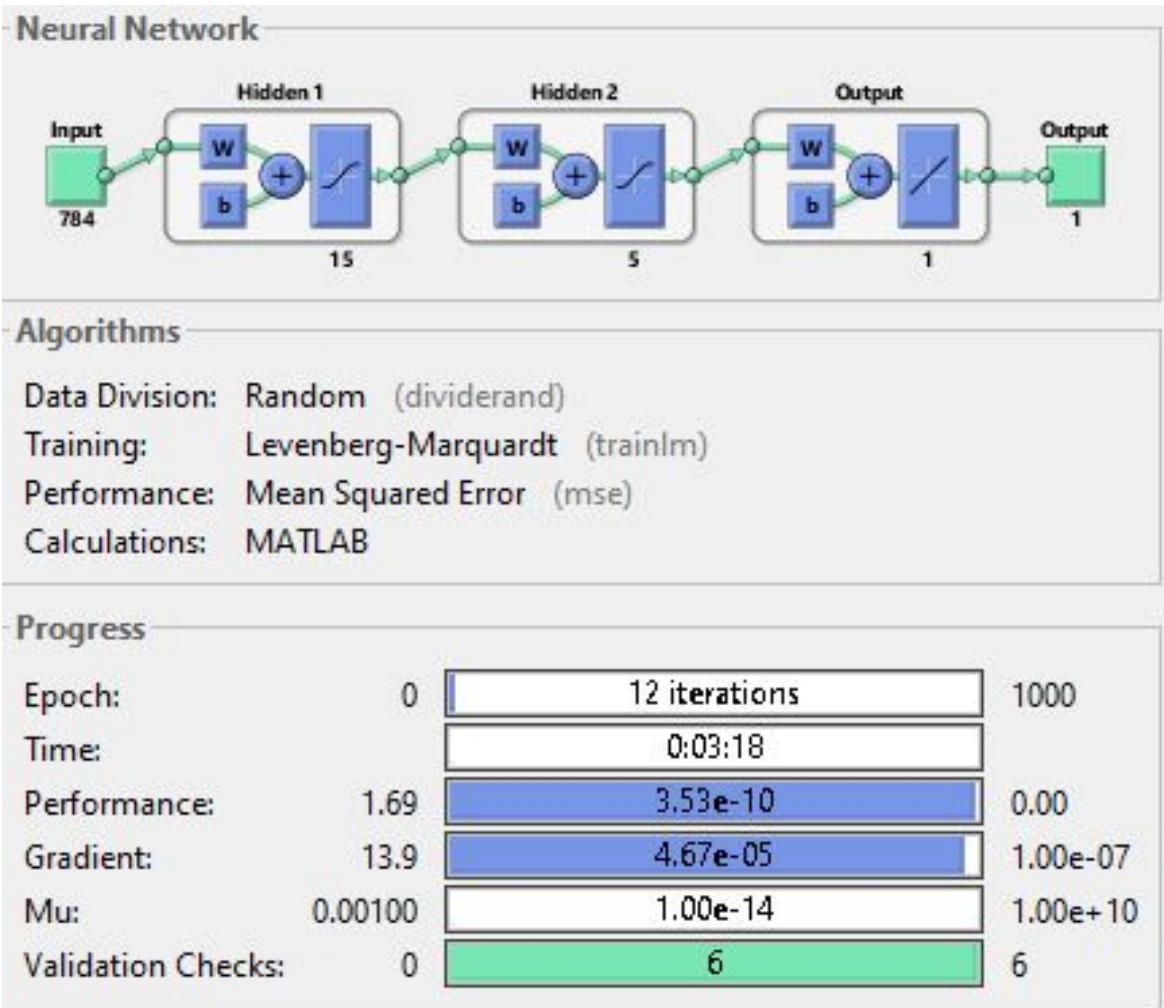
روش های بالا برای آموزش شبکه عصبی بررسی شد. روش ۳ از نظر پیچیدگی، ساده تر و از نظر سرعت، سریع تر است و دقت قابل قبول تری نسبت به دو روش قبل را دارد. نتایج بر اساس همین روش ارائه شده است.

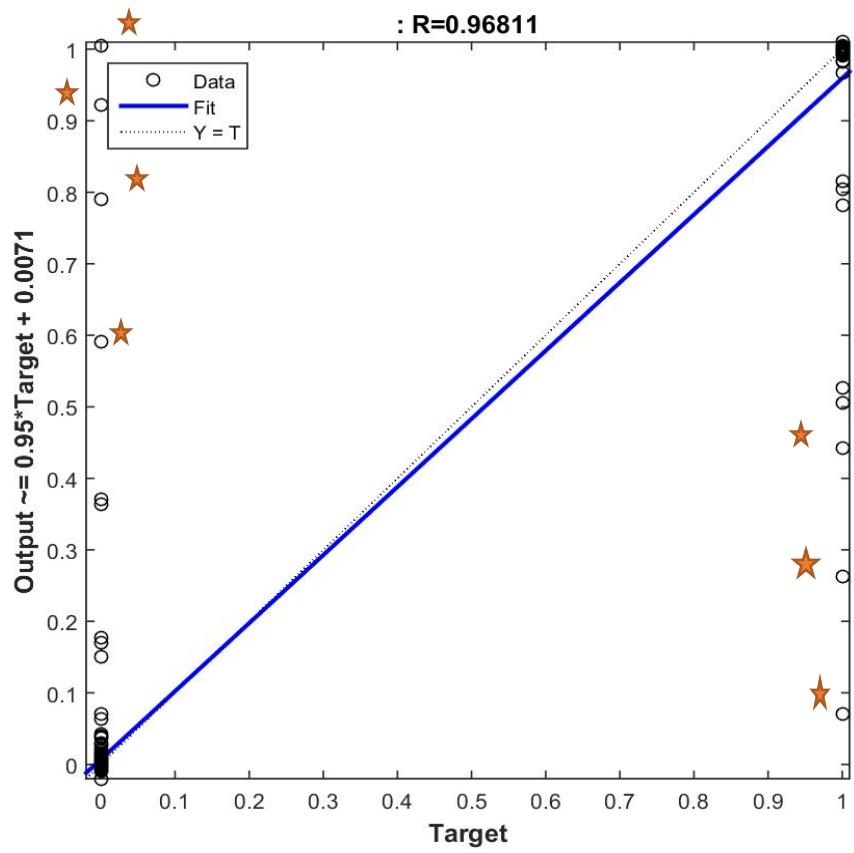
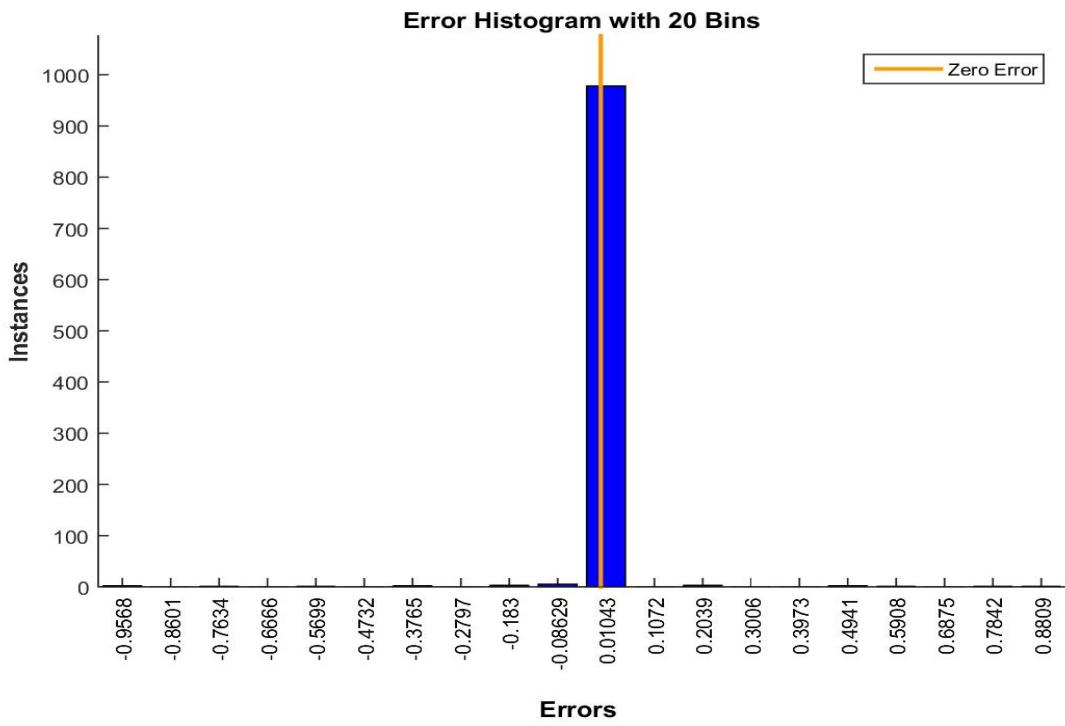
با توجه صفر و یک بودن خروجی، اگر عدد خروجی شبکه عصبی بزرگتر از ۰.۵ بود آن را ۱ در نظر می گیریم و اگر کوچکتر از ۰.۵ بود آن را صفر در نظر می گیریم. در ادامه نتایج شبکه عصبی برای هر گروه (اعداد ۰ تا ۹) ارائه شده است.

ابتدا ساختار مناسب برای آموزش یک گروه با صحیح و خطأ مشخص شده است. البته ساختارهای دقیق تری را می‌توان برای این امر استفاده نمود، اما با توجه به افزایش مدت نتیجه‌گیری از ساختار دولایه‌ای با ۱۵ و ۵ نورون استفاده شده است. برای بقیه گروه‌ها نیز از همین ساختار استفاده می‌شود.

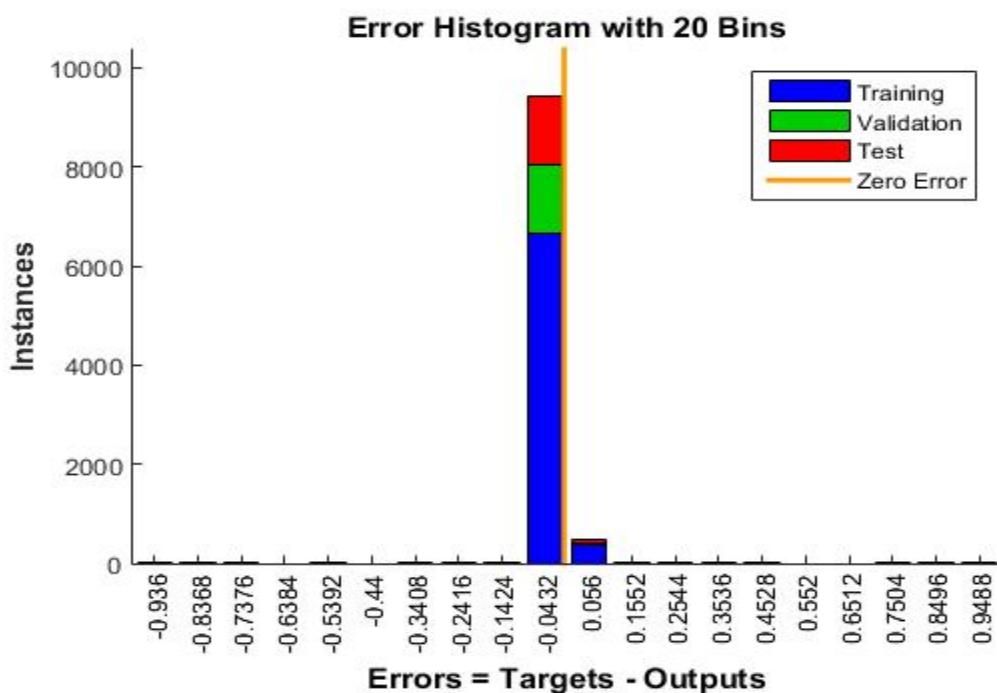
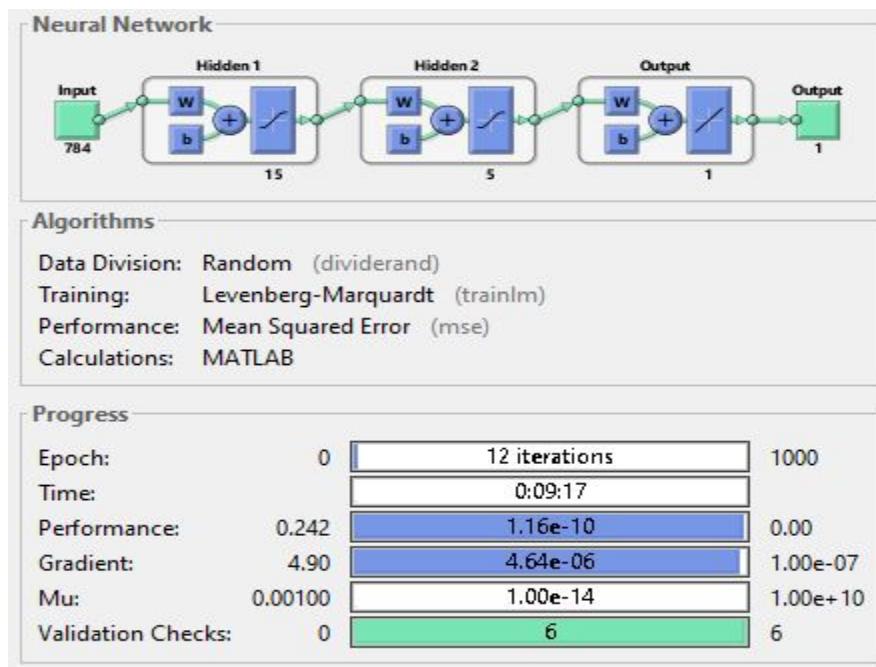
در ادامه ساختار و نتایج گروه‌های ۰ تا ۹ آورده شده است. ساختار تمامی شبکه‌های عصبی یکسان است. برخی از آنها آورده شده است. همچنانی نمودار رگرسیون و خطأ در هر گروه استخراج شده از شبکه عصبی نیز ارائه شده است. همانطور که گفته شد با فرض انجام شده، خطاهای زیر ۰.۵ موجب اشتباہ در گروه بندی تصاویر نمی‌شود. تعداد تصاویر برای آموزش ۱۰۰۰ در نظر گرفته شده است. این تعداد تقریب خوبی را نسبت به زمان انجام به ما می‌دهد. البته برای گروه ۰ تعداد ۱۰۰۰۰ تصویر هم آورده شده که در ادامه می‌بینیم.

گروه عدد ۰: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۰.۵ باشد، یعنی آن تصویر عدد ۰ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است. همانطور که در نمودار خطاهای دیده می‌شود، تعداد نمونه‌هایی که خطای آنها بیشتر از ۰.۵ باشد نسبت به کل نمونه‌ها کمتر از ۱۰ درصد است. موارد ستاره دار در نمودار رگرسیون هم نشان دهنده تعداد خطاهاست که در تعداد ۱۰۰۰ نمونه تعداد قابل قبولی است.

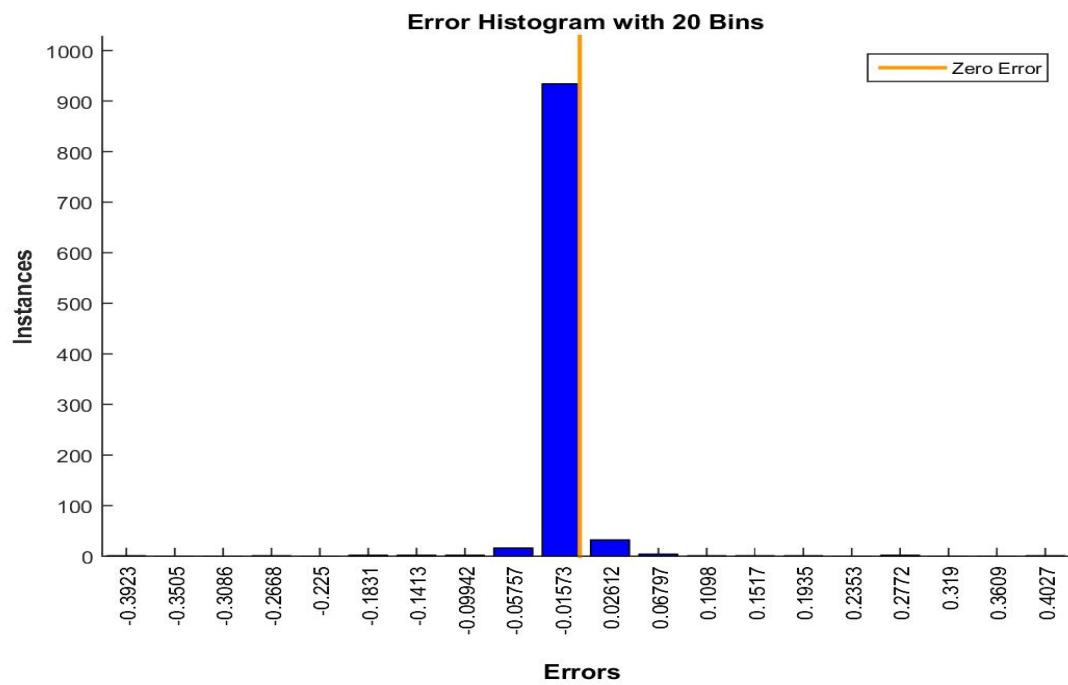
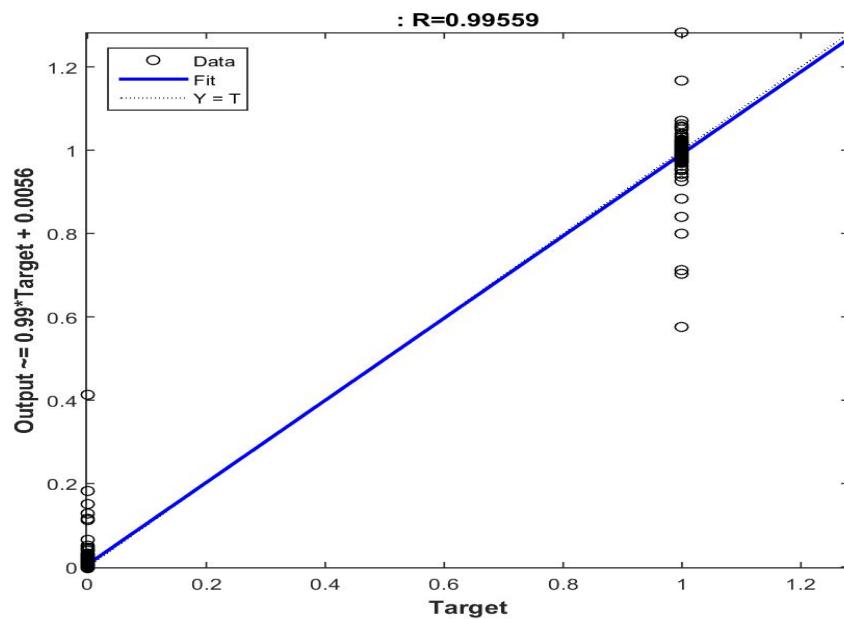




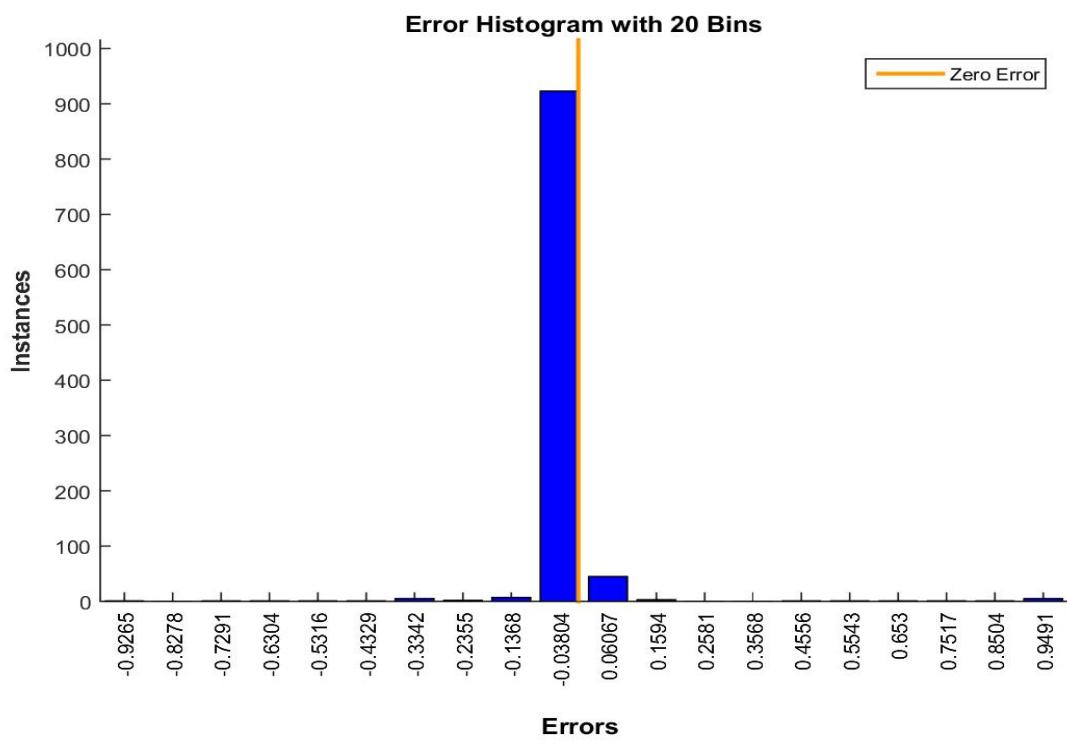
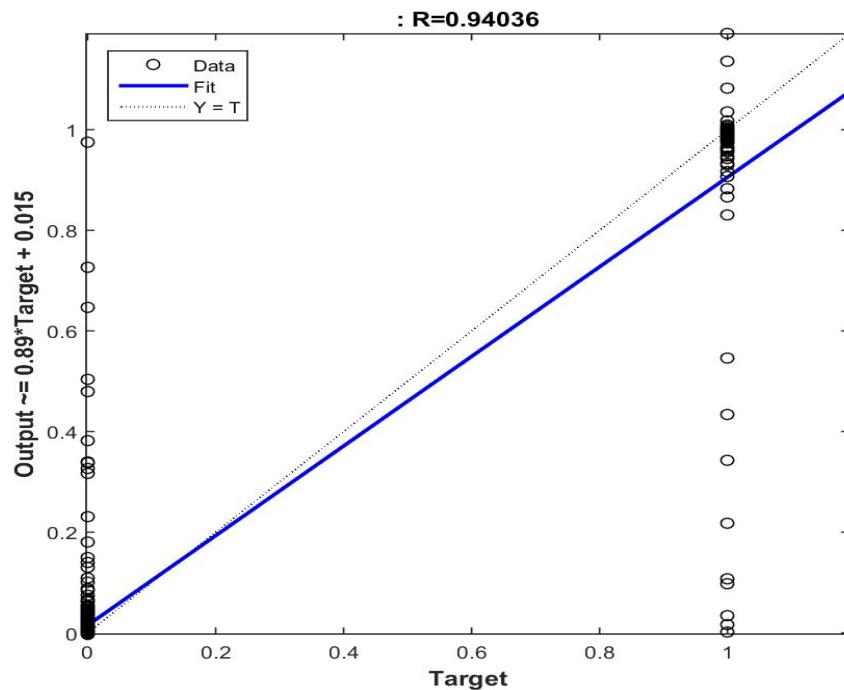
گروه عدد ۰ با ۱۰۰۰۰ نمونه: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۵۰ باشد، یعنی آن تصویر عدد ۱ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.



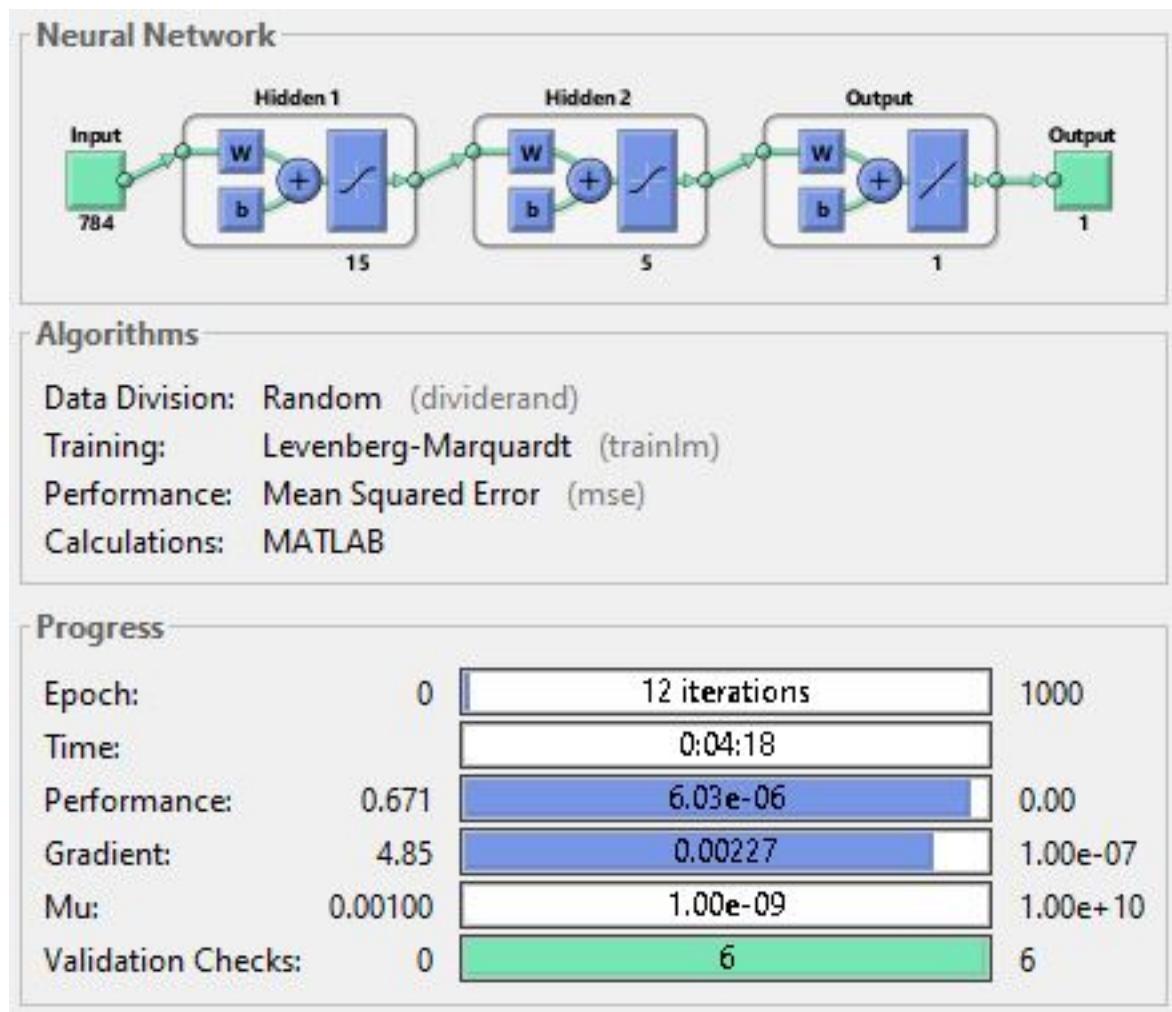
گروه عدد ۱: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۰.۵ باشد، یعنی آن تصویر عدد ۱ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.

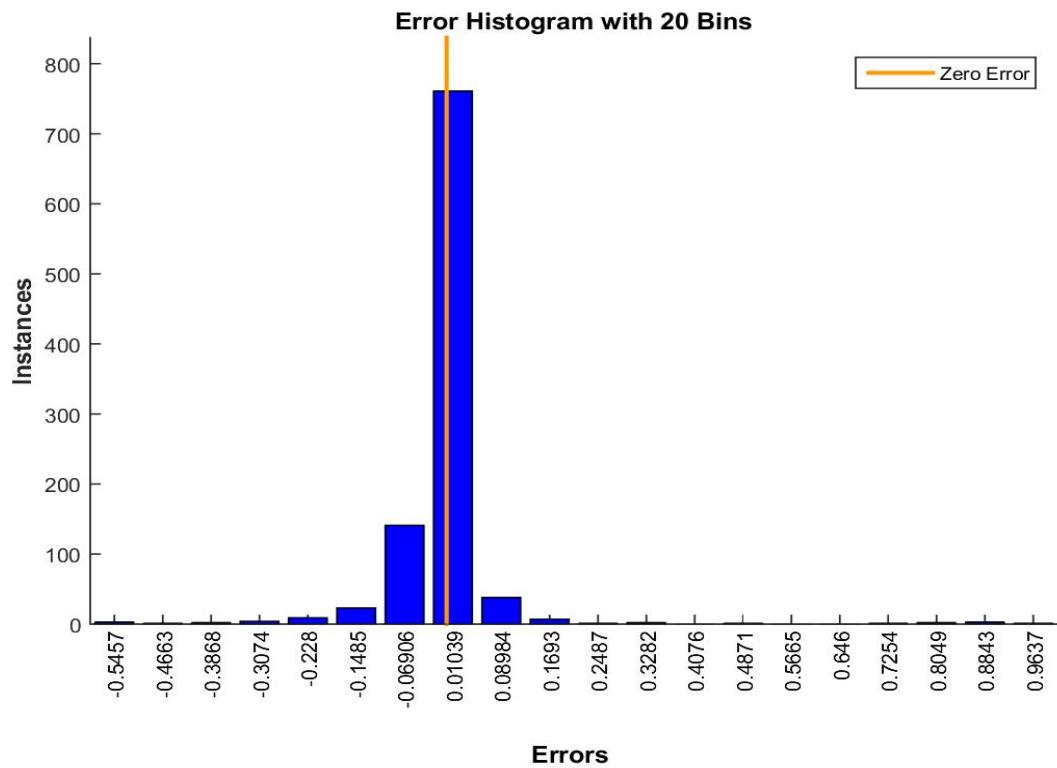
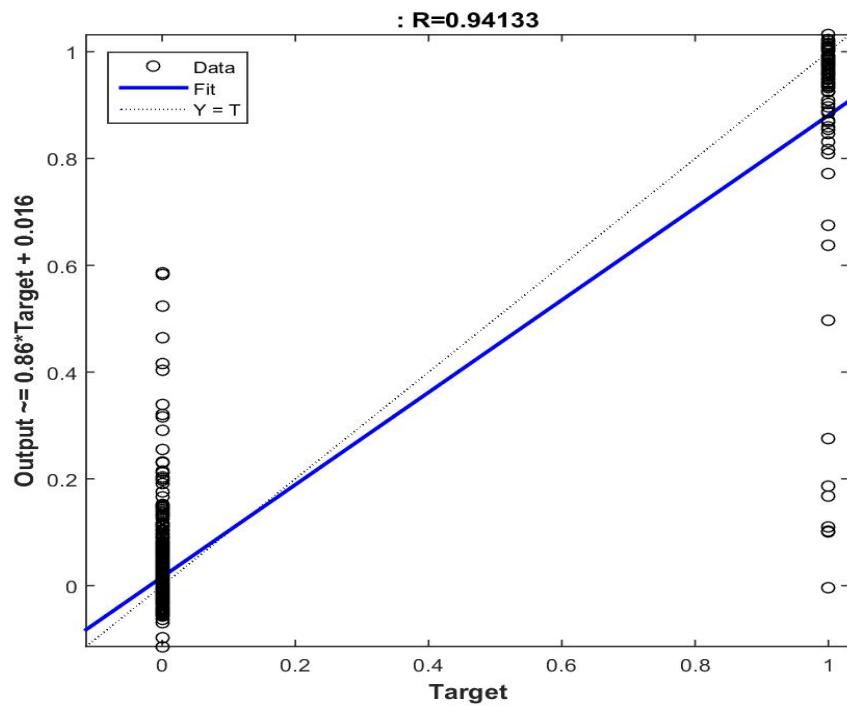


گروه عدد ۲: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۵۰ باشد، یعنی آن تصویر عدد ۲ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.

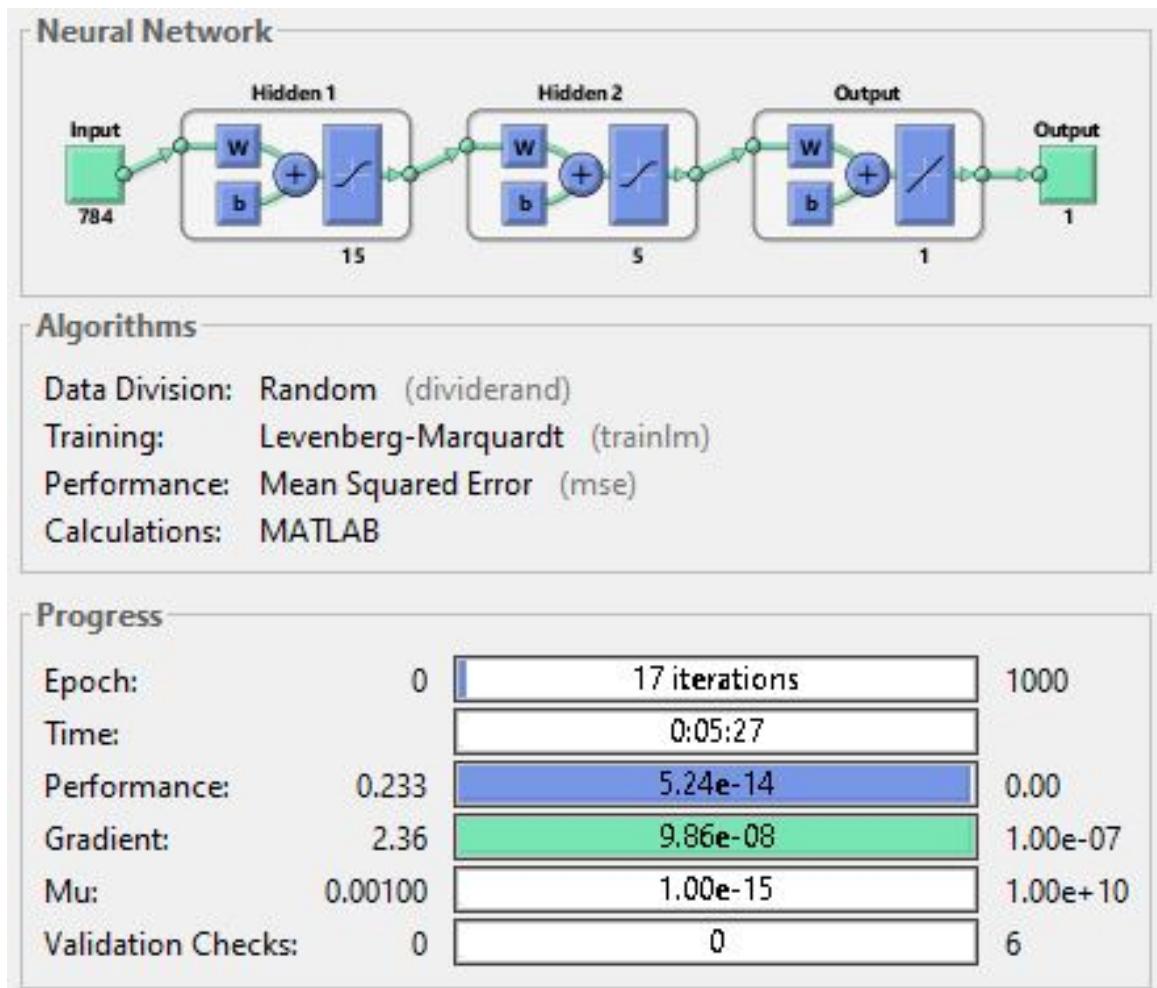


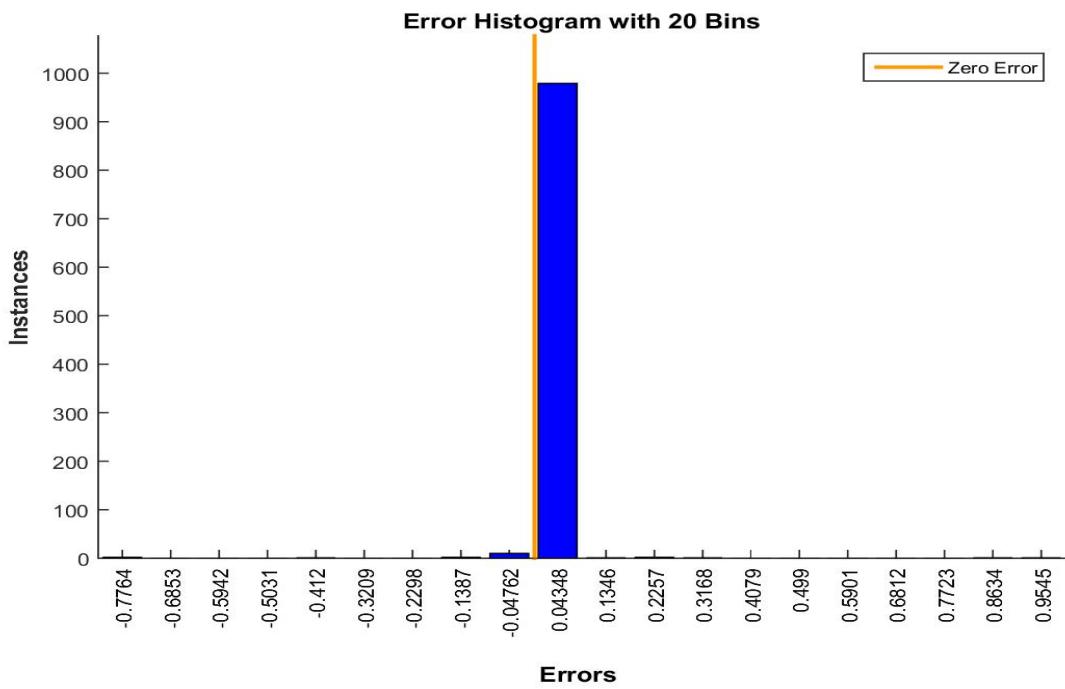
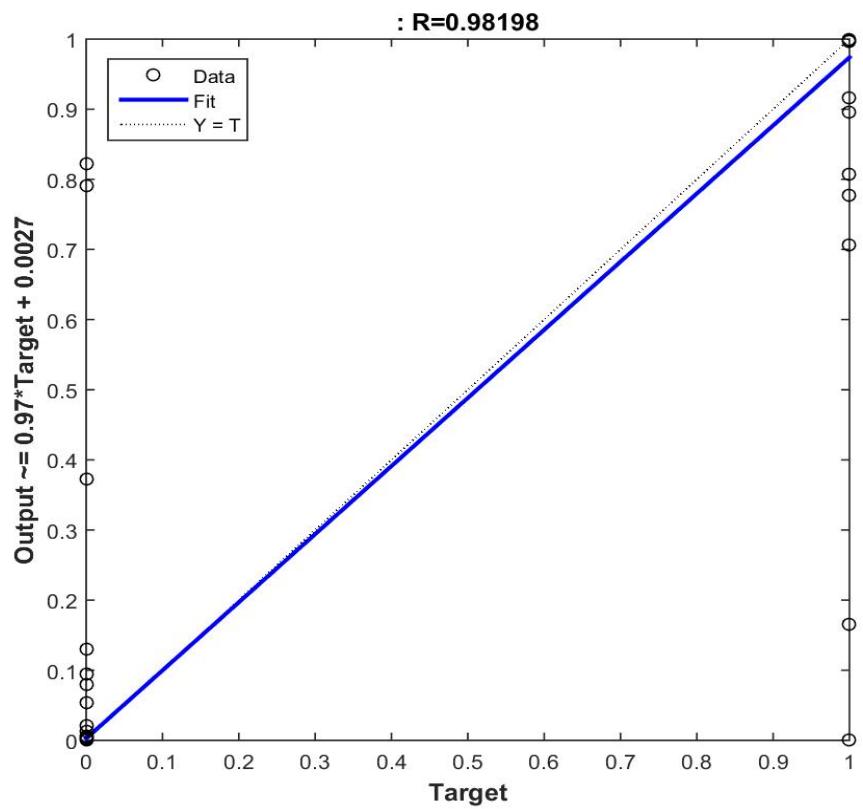
گروه عدد ۳: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۰.۵ باشد، یعنی آن تصویر عدد ۳ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.



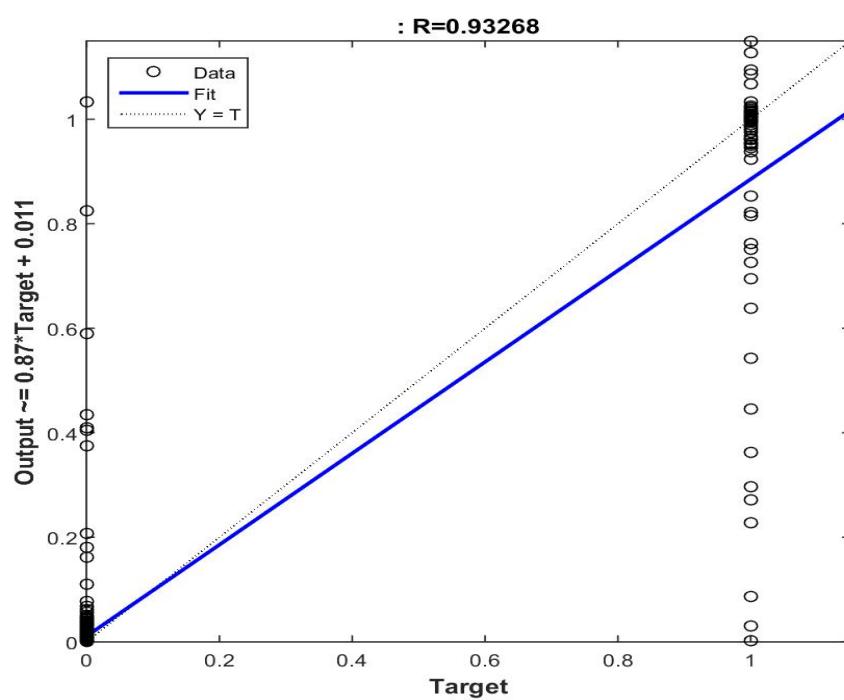
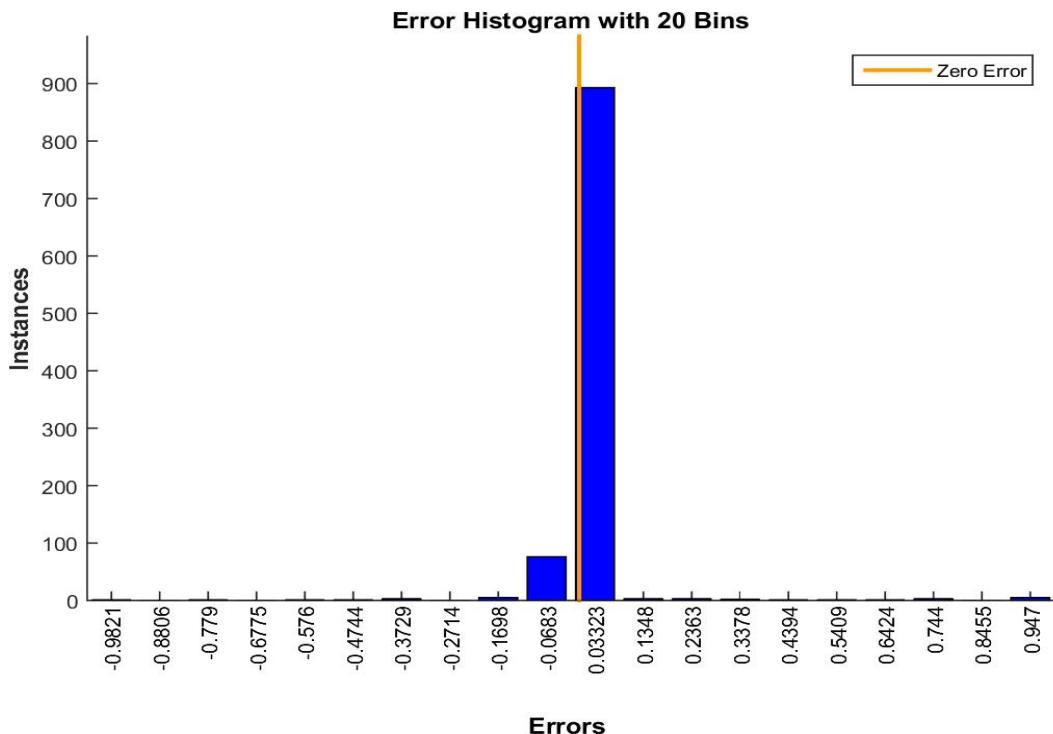


گروه عدد ۴: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۰.۵ باشد، یعنی آن تصویر عدد ۴ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.





گروه عدد ۵: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۵۰ باشد، یعنی آن تصویر عدد ۵ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.



گروه عدد ۶: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۰.۵ باشد، یعنی آن تصویر عدد ۶ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.

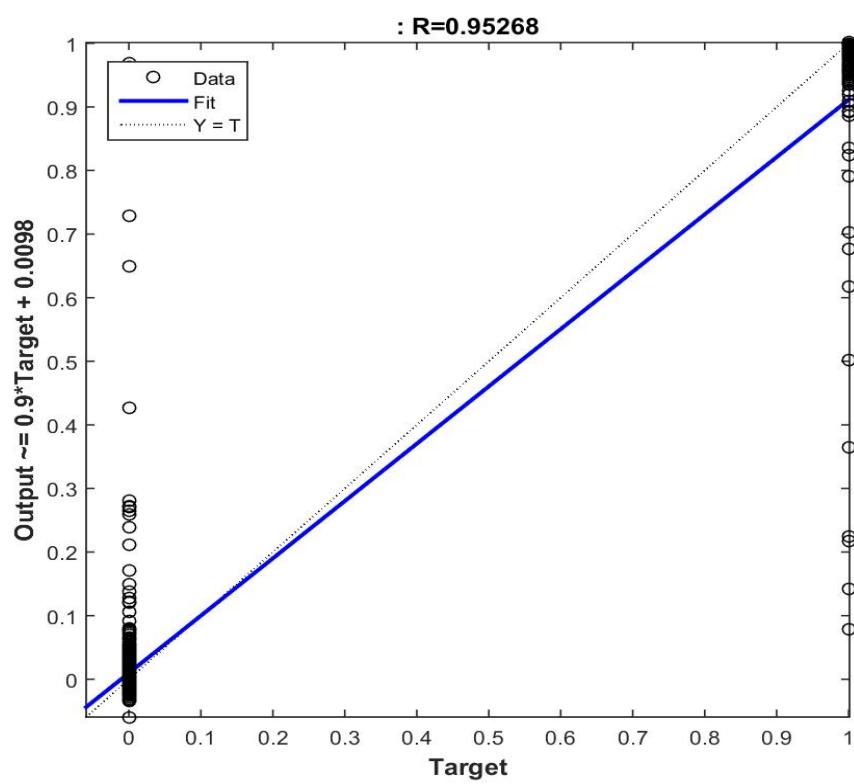
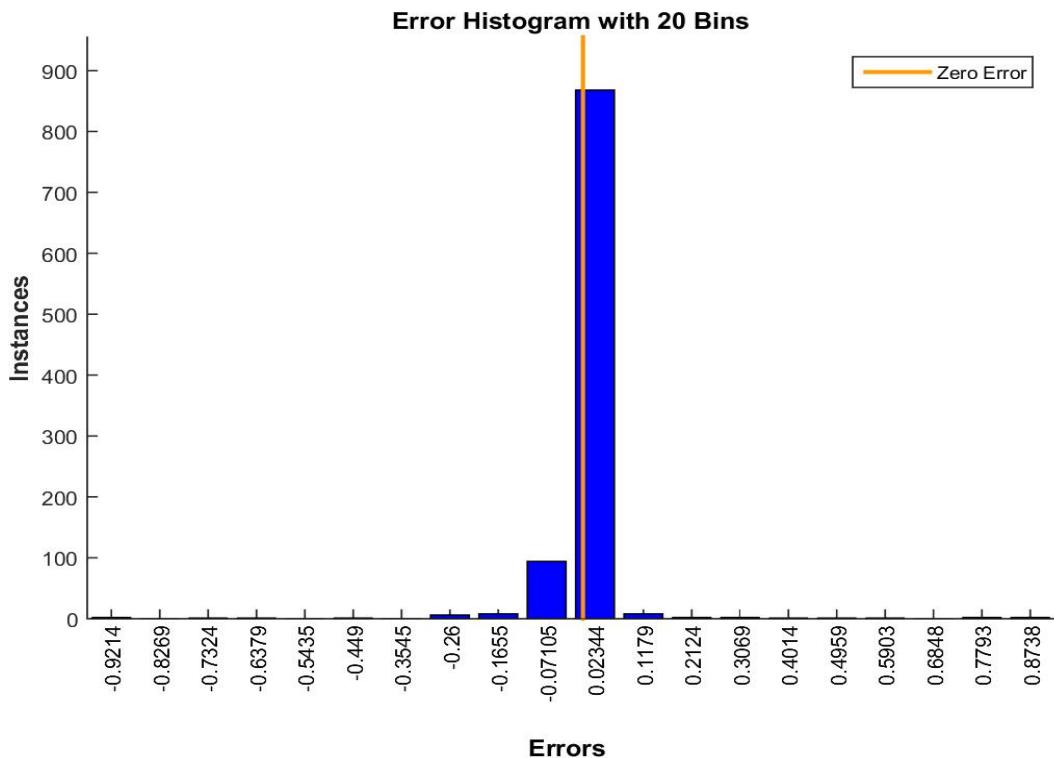
Neural Network

Algorithms

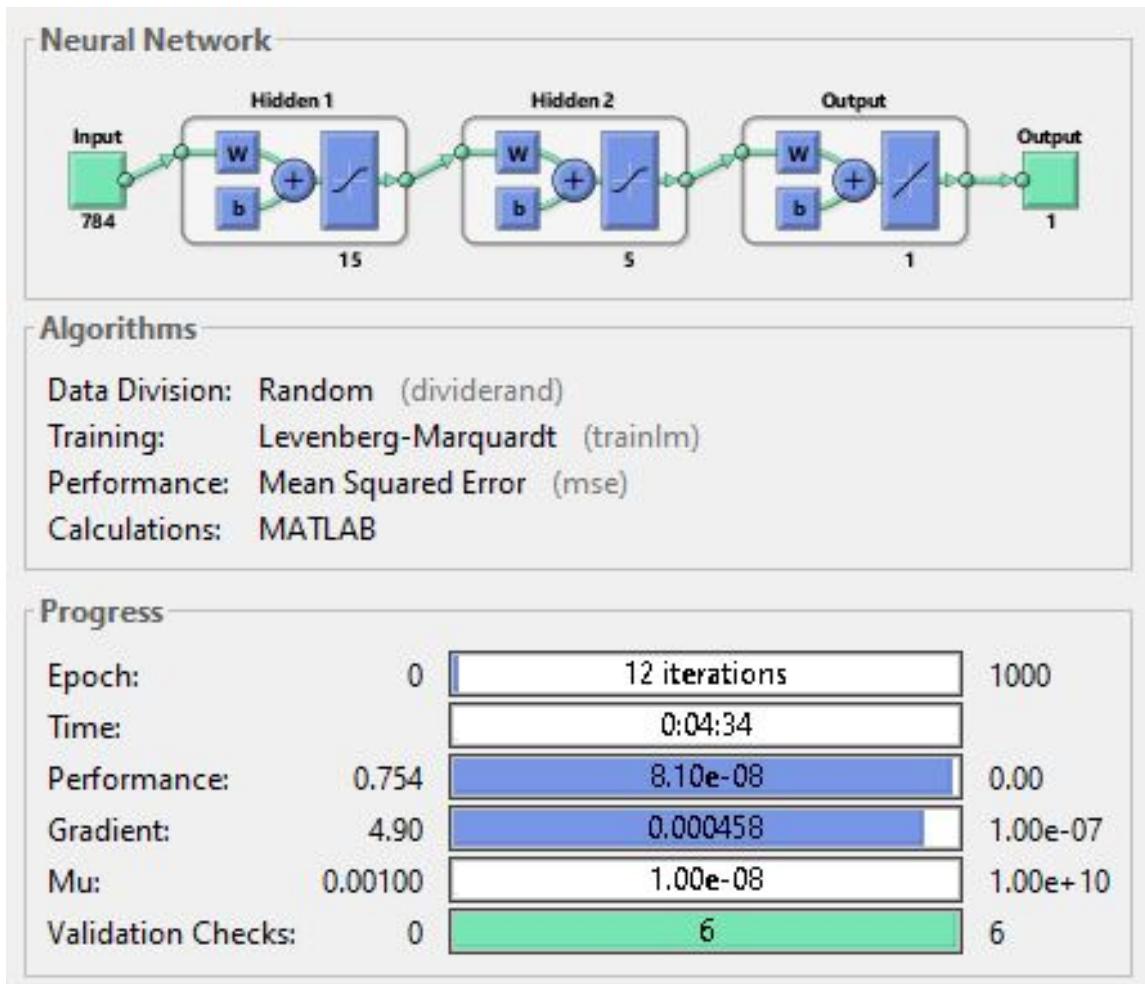
Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MATLAB

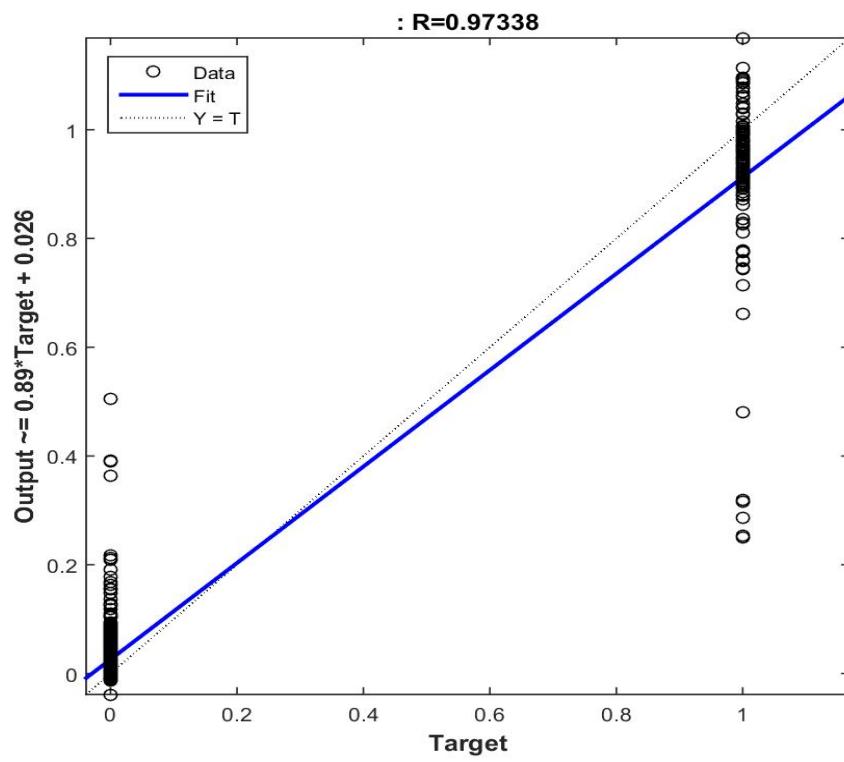
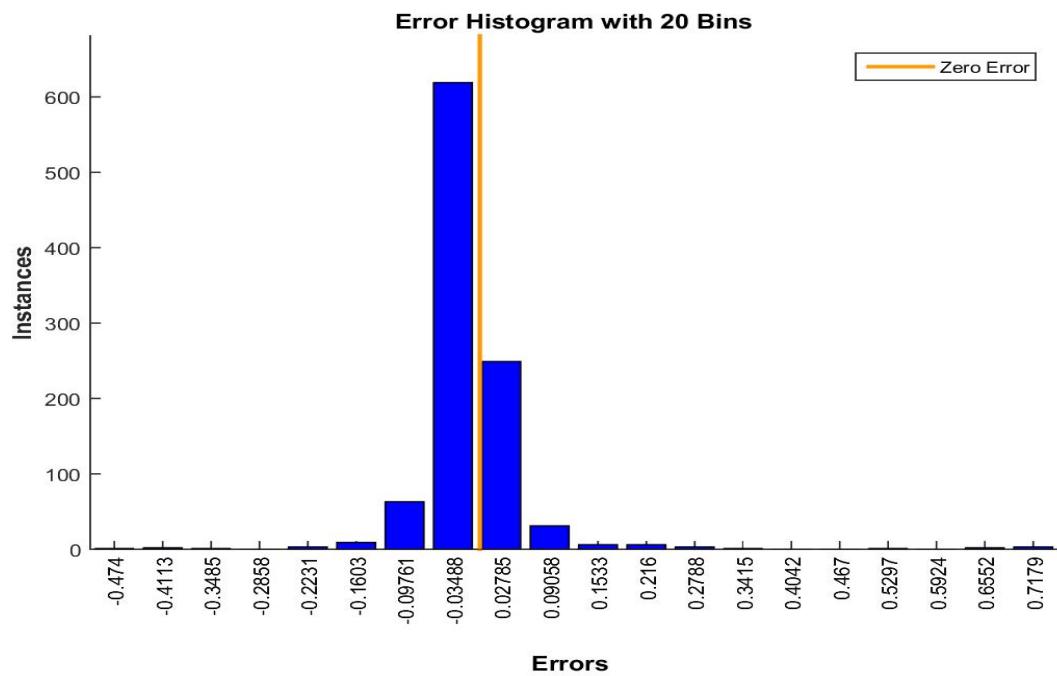
Progress

Epoch:	0	10 iterations	1000
Time:		0:02:35	
Performance:	0.402	4.12e-09	0.00
Gradient:	4.71	5.39e-05	1.00e-07
Mu:	0.00100	1.00e-13	1.00e+10
Validation Checks:	0	6	6

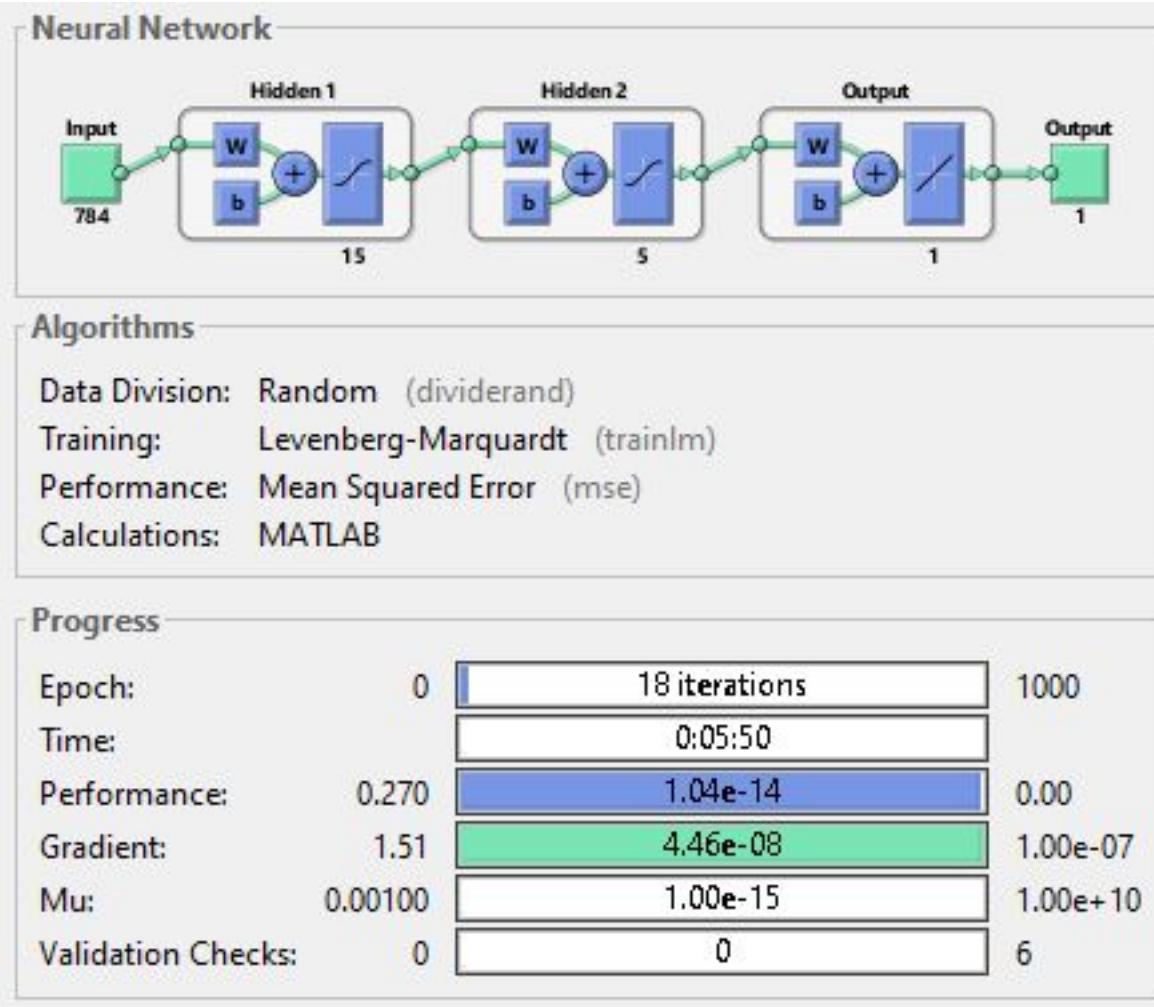


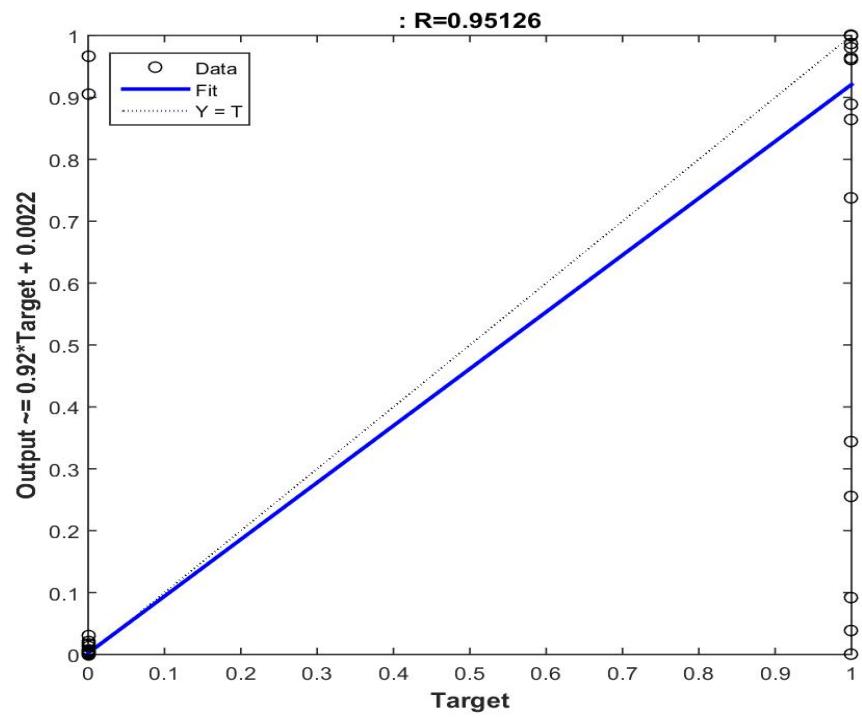
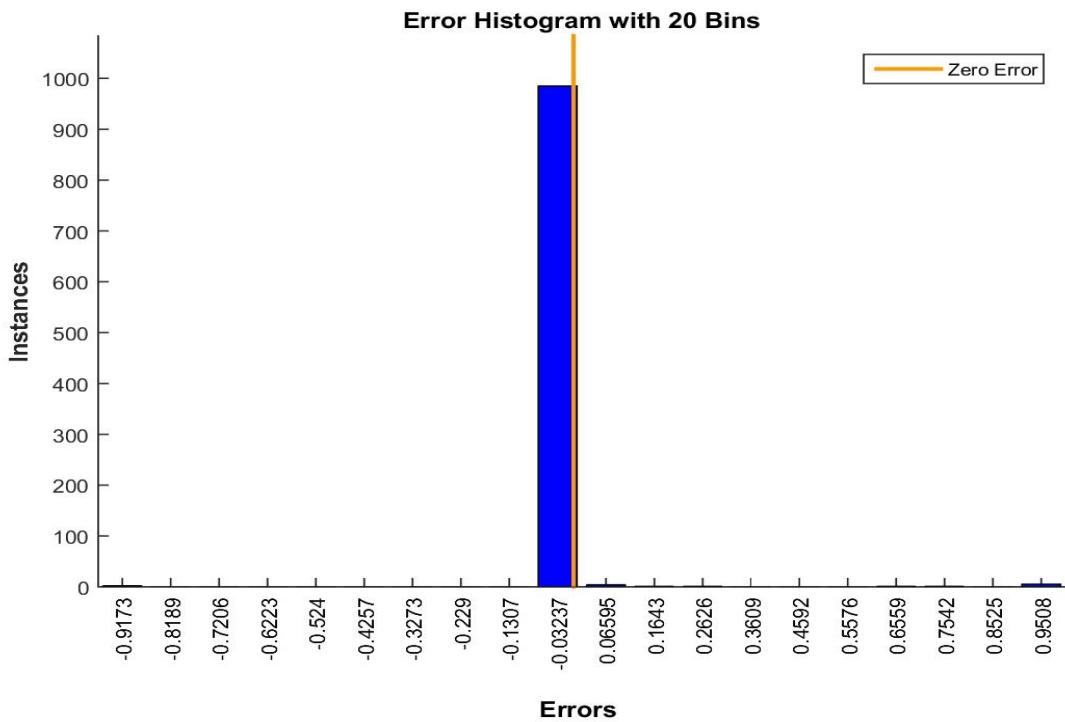
گروه عدد ۷: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۰.۵ باشد، یعنی آن تصویر عدد ۷ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.



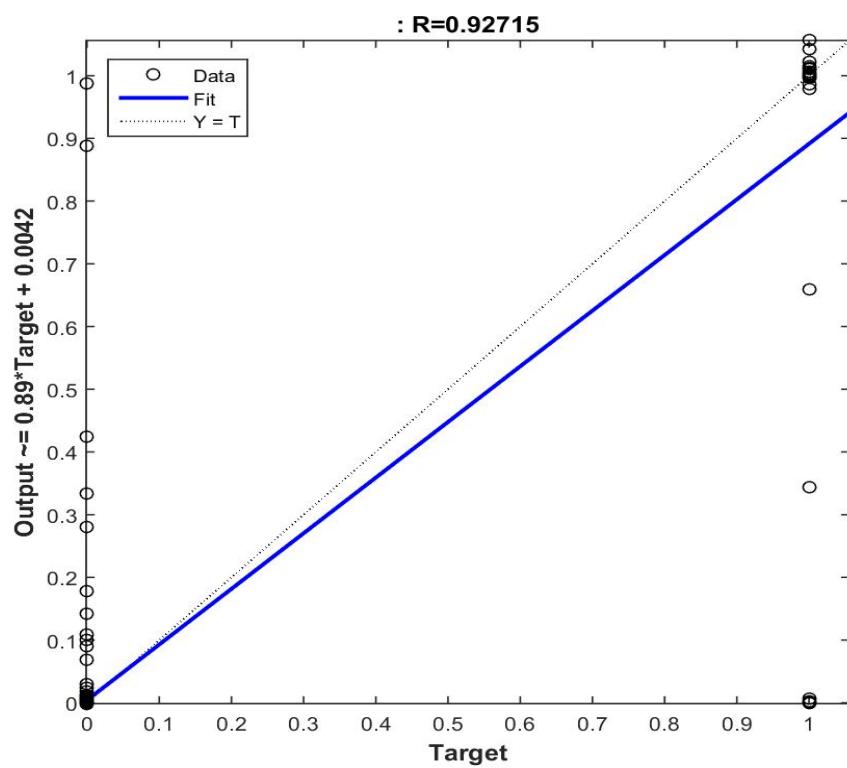
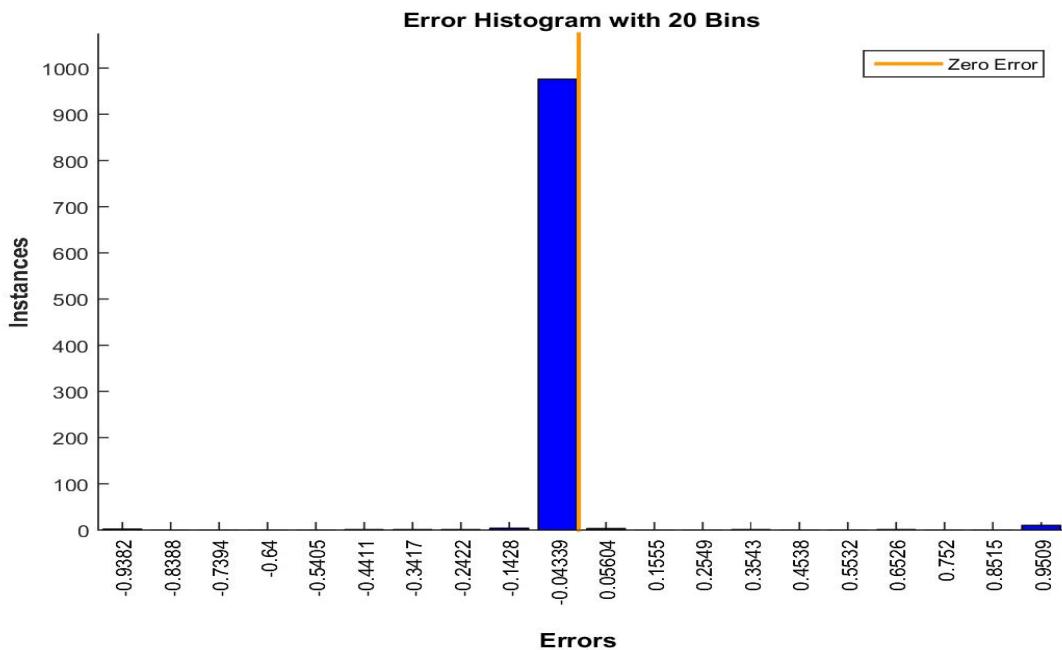


گروه عدد ۸: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۰.۵ باشد، یعنی آن تصویر عدد ۸ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.





گروه عدد ۹: اگر خروجی در شبکه عصبی برای یک عکس بزرگتر از ۰.۵ باشد، یعنی آن تصویر عدد ۹ را نشان می‌دهد و در غیر این صورت تصویر عدد دیگری است.



توجه:

همچنین همراه همین گزارش کد پایتون دیتاست MINIST نیز در ژوپیتر با نام `mnist_PART5.ipynb` نیز آپلود شده است که در آن نتایج و نمودار ها به خوبی قابل مشاهده اند.

فایل پایتون آن نیز با نام `mnist_PART5.py` آپلود شده است.

بخش ششم:

در این بخش سعی داریم تا با استفاده از شبکه عصبی نویز روی تصاویر را حذف نماییم. به این منظور از دیتابست بخش قبلی استفاده می کنیم. ابتدا با استفاده از دستور زیر نویز گوسی به آن اضافه می نماییم.

```
imnoise(Trainx(i,:), 'gaussian', 0.1, 0.05);
```

دو عکس زیر به ترتیب تصویر اصلی و تصویر با نویز را نشان می دهد که با استفاده از دستور بالا به آن نویز اضافه شده است.

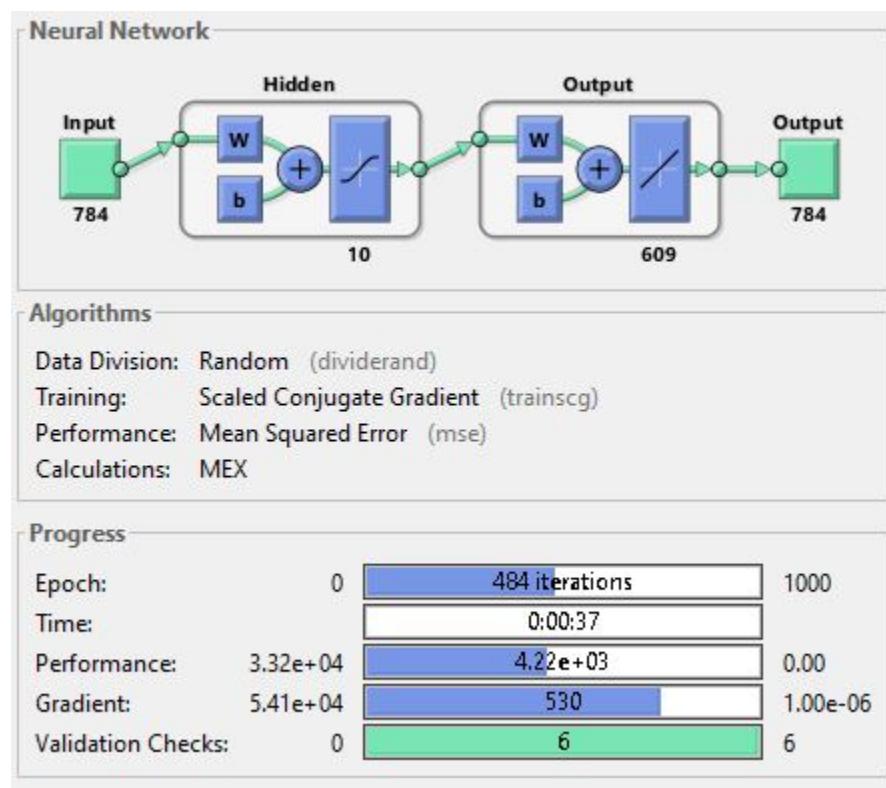


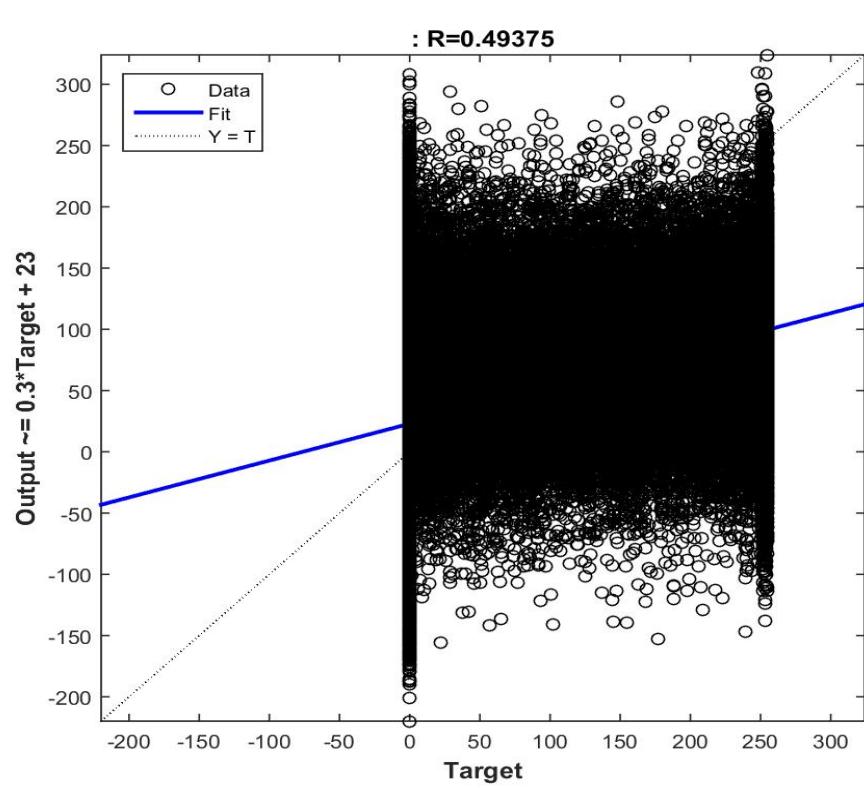
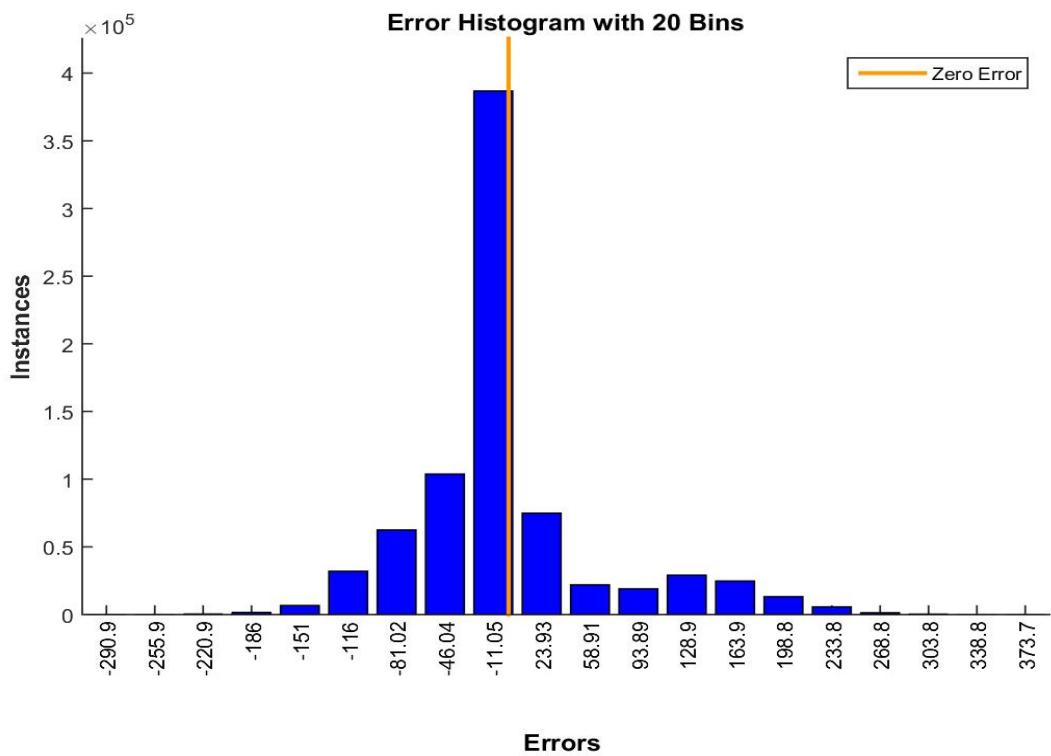
بنابراین شبکه های عصبی برای این منظور آموزش داده می شود که ورودی آنها تصاویر با نویز و خروجی آنها تصاویر بدون نویز است. در ادامه ۴ شبکه عصبی با به ترتیب ۱۰، ۵۰، ۱۰۰ و ۵۰۰ نوروون در یک لایه آموزش داده می شوند که نتایج هر یک آورده شده است.

خروجی هر یک از شبکه های عصبی نیز برای تصویر زیر آورده شده است.



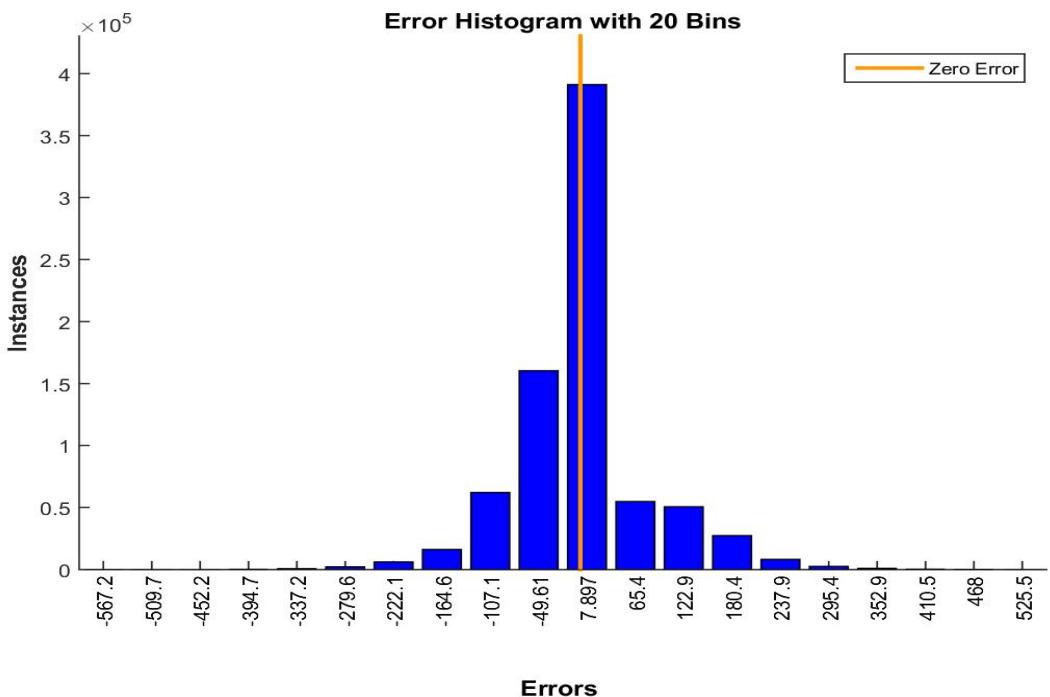
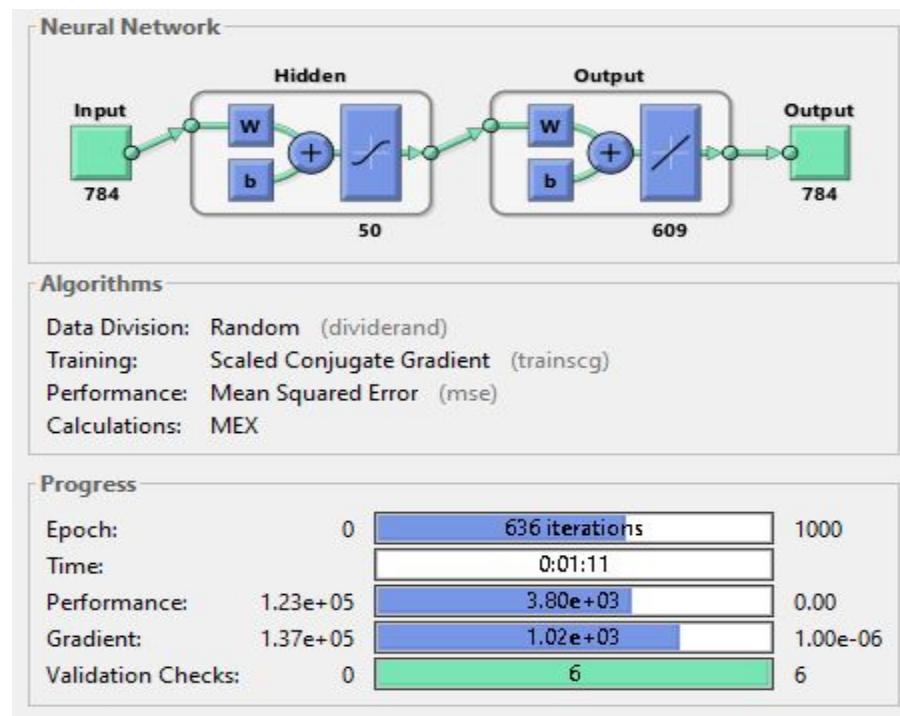
شبکه عصبی با یک لایه پنهان و ۱۰ نورون:

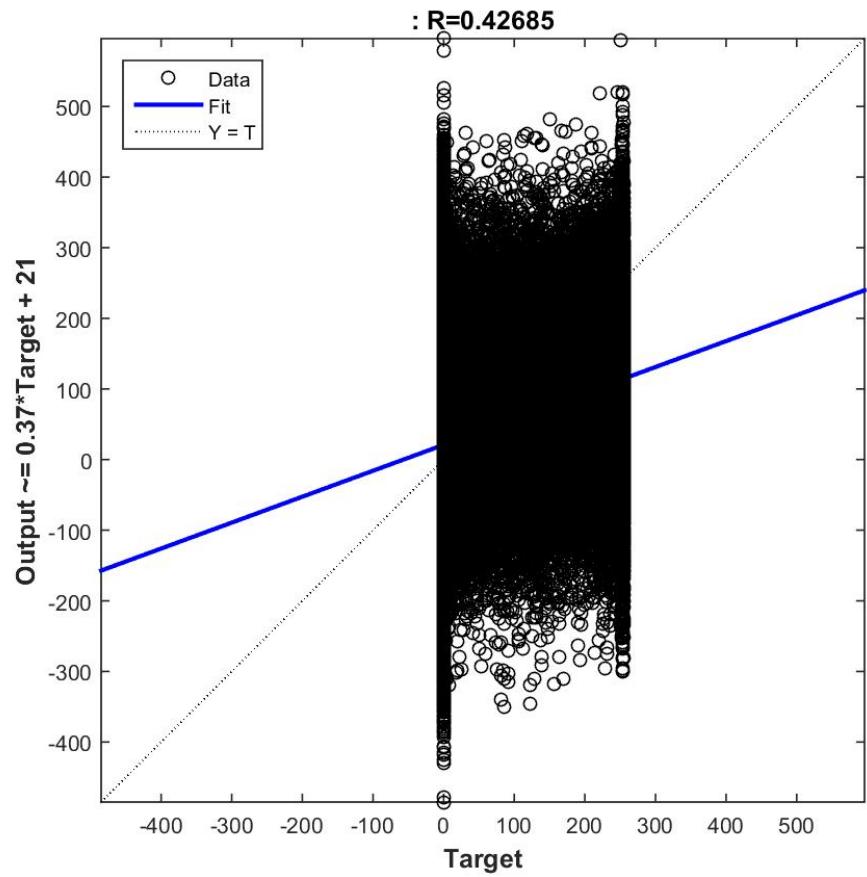




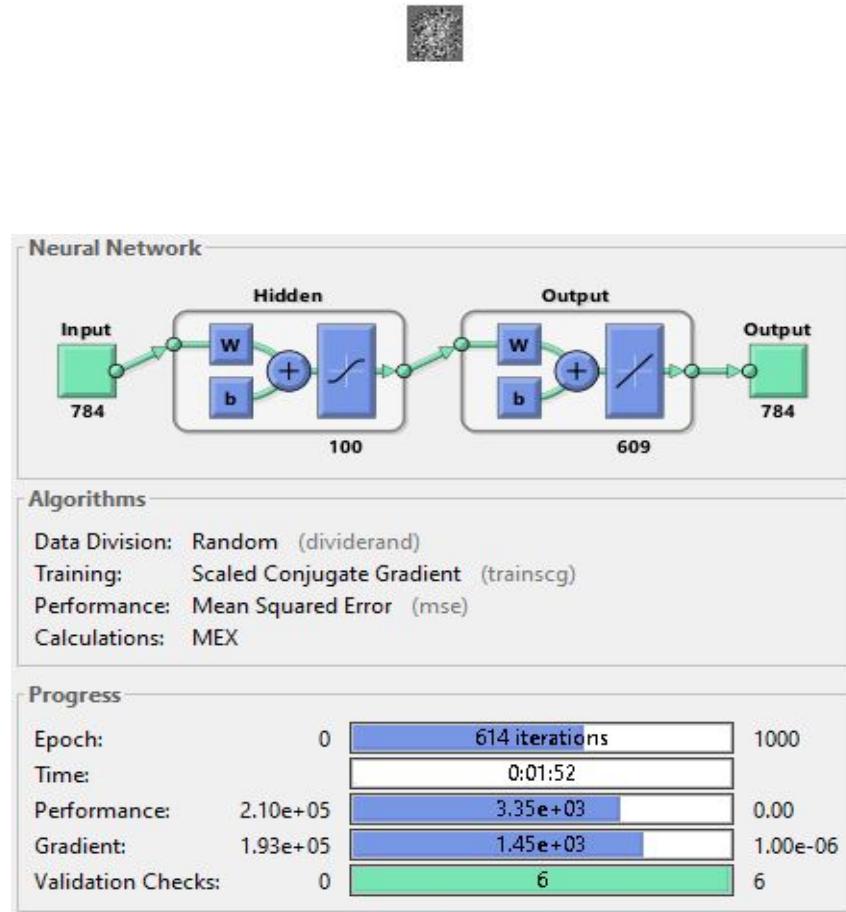
شبکه عصبی با یک لایه پنهان و ۵۰ نوروون:

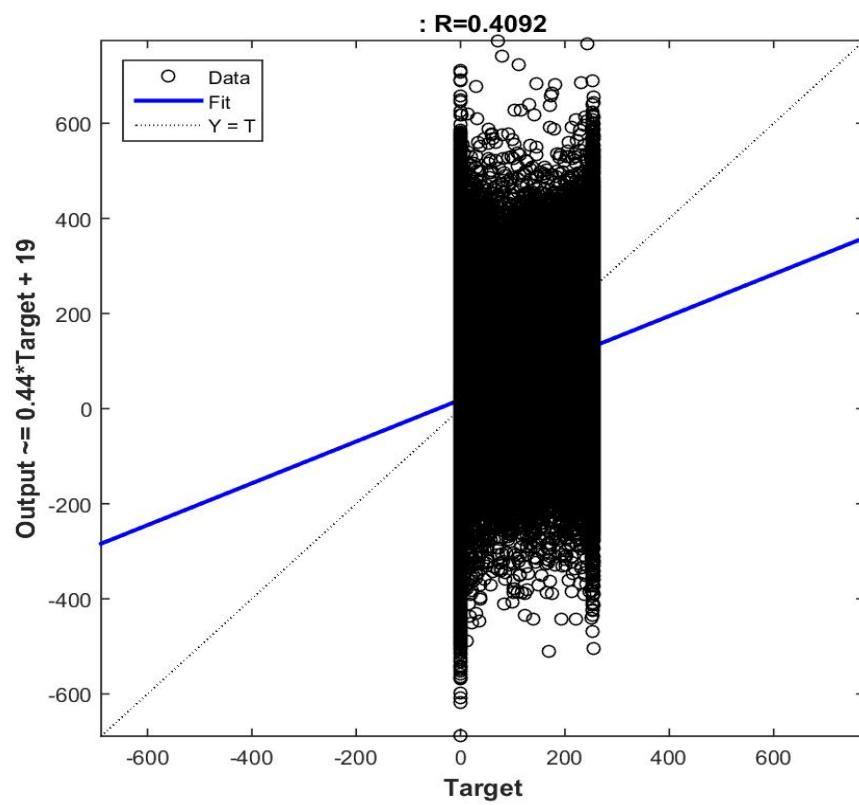
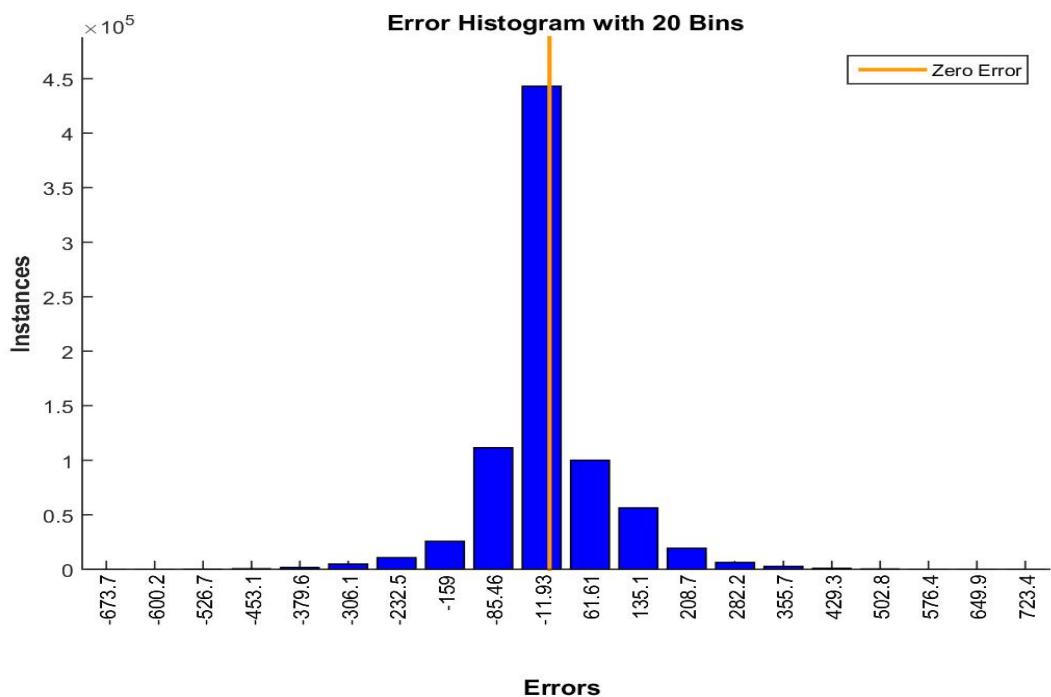






شبکه عصبی با یک لایه پنهان و ۱۰۰ نورون:





شبکه عصبی با یک لایه پنهان و ۵۰۰ نورون:

