

بسم الله الرحمن الرحيم

## کاربرد شبکه عصبی مصنوعی جهت تخمین پارامترهای مختلف در مهندسی تونل و مهندسی معدن

در این مقاله آموزشی به نکات زیر پرداخته شده است:

✓ مبانی شبکه عصبی مصنوعی

✓ برنامه نویسی شبکه عصبی با استفاده از نرم افزار MATLAB

✓ آنالیز رگرسیون چندگانه و مقایسه نتایج آن با نتایج شبکه عصبی

✓ مقایسه نتایج حاصل با مقادیر واقعی اندازه گیری شده

این گزارش طوری تهیه شده است که دوستان عزیز می توانند بعد از مطالعه این آن در موارد مشابه از شبکه های عصبی جهت تخمین پارامترهای اندازه گیری شده واقعی و مقایسه نتایج آن با مقدار برآورد شده با شبکه عصبی بپردازند.

همچنین این مقاله طوری تنظیم شده است که شما قادر خواهید بود برای مواردی همچون:

✓ تخمین نشست تونل ها با استفاده از شبکه عصبی

✓ تخمین پارامترهای اندازه گیری شده در آزمایشگاه مکانیک سنگ با استفاده از شبکه عصبی

✓ تخمین میزان جابجایی ویا سایر پارامترهای مرتبط در تونل ها با استفاده از شبکه عصبی

✓ تخمین میزان عقب زدگی در معادن روباز با استفاده از شبکه عصبی

✓ ...

و به طور کلی در حیطه مهندسی تونل و مهندسی معدن شما می‌توانید پارامترهای مختلف مجهول که با استفاده از ابزار اندازه گیری برآورد شده اند را با استفاده از شبکه عصبی به اصطلاح "صحت آرایی" یا verify کنید.

## مقدمه

در این مقدمه لازم ندانستم که تکرار مکررات یا کپی مطالب موجود در کتاب های پرمحتوای شبکه عصبی را با ترجمه ای دست و پا شکسته ارائه دهم. عناوین کتاب های خیلی خوب در مورد شبکه عصبی رو قرار وبلاگ قرار دادم. (به وبلاگ مهندسی تونل <http://tunnel-iran.blogfa.com>) مراجعه کنید.

اما به صورت خیلی خلاصه برای دوستان عزیز که تازه می‌خواهند وارد حوزه ی شبکه عصبی شوند باید عرض کنم که در مسائل مربوط به شبکه عصبی همواره ما با یک گمشده طرفیم. حال این گمشده ما می‌تواند قابلیت نفوذ یک دستگاه حفار یا یک پارامتر مهم سنگ که قصد داریم در آزمایشگاه مکانیک سنگ آن را برآورد کنیم باشد. حتی در بعضی از موارد ممکن است شما با استفاده از نرم افزارهای عددی، که برخی از آن ها در وبلاگ مهندسی تونل قابل دانلود است، اقدام به تخمین یک پارامتر مثلا جابجایی در سقف تونل و یا میزان گشتاور بر روی سیستم نگهداری (مثل شاتکریت) کنید و بعد از آن نیاز به تعیین صحت پارامتر مود برآورد داشته باشید.

در تمامی این موارد شبکه عصبی ابزار قدرتمندی است که به کمک شما خواهد آمد، اما چگونه؟

## مراحل تخمین یک پارامتر مجهول (گمشده!) با استفاده از شبکه عصبی

### الف) تعیین پارامترهای موثر

همانطور که از طبیعت بسیاری از گمشده ها ی ما در حوزه مهندسی تونل، معدن، مکانیک سنگ، خاک و... بر می‌آید، میتوان این گمشده ها را با پارامترهای دیگر مربوط دانست. به طور مثال در تعیین قابلیت نفوذ کله حفار با اندکی مبالغه می‌توان گفت که میزان چسبندگی خاک یا سنگ تاثیر به سزایی در نرخ نفوذ دارد. در اینجا این به تجربه مهندسی شما و همچنین نظارت علمای گرامی (اساتید راهنما) بر میگردد که پارامترهای تاثیرگذار دیگر را نیز با تلاش و تامل کافی تعیین کنید. توجه داشته باشید که این مرحله از حساس ترین مراحل مربوط به پروژه های شبکه عصبی خواهد بود. چراکه در نظر گرفتن پارامترهای خنثی و یا کم تاثیر مدل شبکه عصبی شما را تا حدی غیرواقعی خواهد کرد. لذا سعی کنید این مرحله را با حساسیت ودقت تمام پیش برید.

### ب) تهیه مدل شبکه عصبی

تهیه مدل شبکه عصبی به معنای نوشتن یک برنامه چند خطی در نرم افزار MATLAB و یا استفاده از TOLLBOX های آماده (یا لقمه های آماده) این نرم افزار می‌باشد. لذا نگران نباشد این مرحله به راحتی خوردن یک لیوان آب می باشد. حتی می‌توانید در

حالی که در سایت دانشکده مشغول خوردن چای هستید، مدل شبکه عصبی را هم تهیه کنید. فقط این نکته را خدمت دوستان عزیز بگویم که در کل فرآیند تهیه شبکه عصبی بهینه زمان بر یا به قول فرنگی ها (time consuming) می باشد. پس حتما قند و چای فراوان از خوابگاه به دانشکده بیاورید!



شکل ۱) نمونه یک چایی قند پهلوی که در حین تهیه مدل شبکه عصبی به آن نیاز دارید!!

داده هایی را که جمع آوری کرده اید را در این مرحله باید به چند قسمت نامساوی تقسیم کنید، فرآیند تقسیم را مفصل به عرضتان خواهم رسانید. داده های شما می تواند پارامترهای اندازه گیری شده در آزمایشگاه، محل پروژه های تونل سازی یا معدنی و یا خدایی نکرده ساختگی !!! باشد. (البته بیشتر فرنگی ها داده هایشان ساختگی است!)

(ج) مقایسه نتایج

برای اینکه اعتبار (authenticity) شبکه عصبی که در مرحله قبل تهیه کرده اید را متوجه شوید و برای لحظاتی به خودتان افتخار کنید! نیاز است که نتایج حاصل از شبکه عصیتان را با داده های اندازه گیری شده واقعی مقایسه کنید. در بعضی موارد (کاری است مرسوم در بسیاری از پروژه هایی که در قالب مقالات ISI ارایه میشوند) از نمودارهای مقایسه ای استفاده می شود. به این صورت که نتایج شبکه عصبی و مقادیر اندازه گیری شده واقعی را بر روی یک نمودار رسم کرده و نزدیک بودن یا نبودن نتایج را با هم مشاهده کنید. در صورت عدم تطبیق می توانید مرحله (ب) را دوباره تکرار کنید (البته با یک مدل متفاوت، حالا متوجه میشوید که چقدر دیگه قند و چایی لازمه!). بعد از  $n$  بار تکرار سرانجام موفق به کسب نتایج رضایت بخش خواهید شد. (مقدار  $n$  بستگی به شرایط مسئله و پروژه ای که دارید روی آن کار میکند دارد، ولی در کل می توانید پیچیدگی پروژه خود را با مقدار قند و چایی که مصرف می کنید بسنجید!)

سپس نیاز است تا ضرایب همبستگی (Coefficient of Determination & Correlation Coefficient) را نیز برآورد کنید. تعجب نکنید، این دو مفهوم آماری هستند که در کتابی که با عنوان "آمار و احتمال برای مهندسين" در وبلاگ مهندسی تونل قرار داده ام به راحتی قابل درک است. (این کتاب به زبان انگلیسی است).

**هشدار:** خواهشمندم حتما با مفاهیم آماری آشنا باشید که صرفا تعیین یک عدد همینجوری ارزش خاصی ندارد. ناسلامتی شما دارید مهندس میشوید!

#### د) بحث و تحلیل نتایج

در این مرحله حتماً نظر افراد متخصص (مانند اساتید راهنما یا مشاور) در این زمینه را جویا شوید، و صحت و درستی نتایج به دست آمده را مورد بحث قرار دهید. به طور کلی شبکه عصبی قدرت فراوانی در تخمین روابط بین پارامترهایی که مخصوصاً با هم رابطه غیر خطی دارند، دارد. لذا هرچه تعداد پارامترهای وابسته ی موثر شما بیشتر باشد، با اطمینان بالاتری میتوانید نتایج شبکه عصبی را قبول کنید.

در قسمت بعد به تشریح موارد "الف" تا "د" همراه با حل یک مثال نمونه خواهیم پرداخت.

**نکته:** داده های مورد استفاده در این مسئله فرضی و فقط جهت آموزش به شما ارائه شده اند.

#### تعریف مسئله:

میخواهیم قابلیت حفاری یک دستگاه حفاری TBM را که در یک پروژه تونل سازی مورد استفاده قرار گرفته است را با استفاده از شبکه عصبی تخمین بزنیم. برای این منظور مهندسین مشاور پارامترهایی را از قبیل چسبندگی سنگ، سرعت چرخش تیغه ها و همچنین نیروی محوری معادل که بر صفحه برش اعمال می شوند را پیشنهاد داده اند که تاثیر بسیار زیادی را بر روی نرخ نفوذ TBM گذاشته اند. با استفاده از نرم افزار MATLAB مدل شبکه عصبی تهیه کنید که به بهترین نحو قادر به تخمین میزان نرخ نفوذ دستگاه TBM باشد.

**نکته:** پارامترهای زیادی بر روی نرخ نفوذ تاثیر دارند چون مسئله فرضی است همین پارامترها را قبول کنید!

#### آنالیز مسئله:

مجهول یا گم شده ی مسئله ما در این جا نرخ نفوذ دستگاه TBM میباشد. همچنین پارامترهایی که جهت برآورد مجهول در دست است توسط مهندس مشاور ارائه شده است. در جدول ۱ مقادیر این داده ها را مشاهده میکنید.

جدول ۱ – پارامترهای موثر در تخمین نرخ نفوذ TBM

C	PS	Th	Cu-Field
22.8	27	850	4.26
22.8	28	850	4.73
22.8	29	850	5.08
22.8	30	850	4.55
22.8	27	930	5.03
22.8	28	930	5.41

22.8	29	930	5.48
22.8	30	930	5.34
22.8	27	1010	6.7
22.8	28	1010	6.84
22.8	29	1010	7.12
22.8	30	1010	6.78
25.8	27	850	4.03
25.8	28	850	4.44
25.8	29	850	4.3
25.8	30	850	4.18
25.8	27	930	4.86
25.8	28	930	5.23
25.8	29	930	5.11
25.8	30	930	4.86
25.8	27	1010	6.5
25.8	28	1010	6.71
25.8	29	1010	6.59
25.8	30	1010	6.44
29	26	850	3.86
29	27	850	4.31
29	28	850	4.18
29	29	850	4.02
29	26	930	4.14
29	27	930	4.52
29	28	930	4.27
29	29	930	4.2
29	26	1010	5.87
29	27	1010	6.2
29	28	1010	6.02
29	29	1010	5.89

پس از تعیین پارامترها نیاز است تا با استفاده از یک مدل شبکه عصبی میزان نرخ نفوذ را تخمین بزنیم در اینجا یک نمونه از برنامه های شبکه عصبی ارائه شده است که در ادامه به توضیح آن خواهیم پرداخت.

ساختار مدل شبکه عصبی که برای حل این مسئله در نظر گرفته شده است به شرح زیر است:

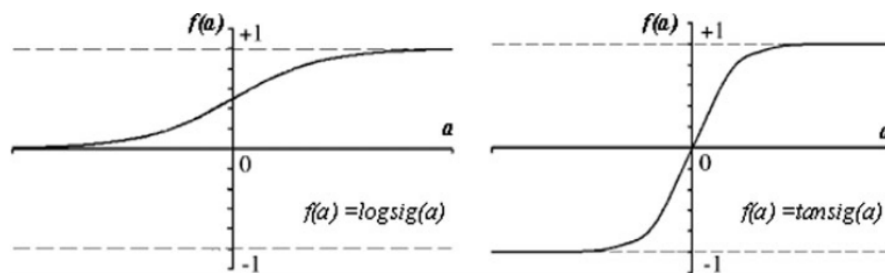
Network type	Multi-layer feed forward backdrop
Adaptive Learning function	Gradient descent method
Training algorithm	Levenberg–Marquardt
Transfer function	Sigmoid
No. of input neurons	03
No. of output neurons	02
No. of hidden layers	01
No. of hidden neurons	06
No. of training epochs	500
No. of datasets	36

در مورد back propagation NET ها باید توجه داشت که پس از اینکه داده های ورودی به شبکه داده شد، با توجه به معماری شبکه خروجی ها محاسبه شده و سپس با مقادیر واقعی مقایسه میشوند. با تعیین مقدار خطای موجود میتوان با تغییر تعداد نرون ها یا تعداد لایه های پنهان کاربری شبکه عصبی را به صورت بهینه تعیین کرد.

در مورد gradient descent که تابع یادگیری در این شبکه در نظر گرفته شده است در نظر گرفتن یک نکته ضروری است. فرض کنید شما در یک فضای مه آلود بر روی دامنه یک تپه ایستاده اید و قصد دارید به بالاترین نقطه تپه دست پیدا کنید. تنها مشکل شما این است که هوا مه آلود است و تا چند قدمی شما قابل دید نیست. یک راه حل این است که شما منتظر بمانید تا هوا صاف شود و به راه خود ادامه دهید، اما عقلانی ترین راه حل این است که شما در جهتی که شیب بیشتری به سمت بالا دارد شروع به قدم برداشتن کنید (فرض کنید در بالای تپه فرد مهمی در انتظار شماست). اما زمانی که با یک نقطه رسیدید این سوال مطرح میشود که آیا این نقطه بالاترین نقطه تپه است؟ متأسفانه بدون جستجوی نقاط دیگر و مقایسه آنها پاسخ این سوال غیر ممکن است. اگر تنها برداشتن قدم به سمت پایین در این مسئله اجازه داده شود، شما ابتدا به یک نقطه پایینی میرسید که به اصطلاح آن را local minimum میگویند. همچنین پایین ترین نقطه را global minimum معرفی کرده اند. همواره رسیدن به global minimum در مسائل مربوط به شبکه عصبی مد نظر نیست. حال فرض کنید به ما یک شاخص بدهند و تعیین کنند که به هر نقطه که رسیدید خود را با آن نقطه مقایسه کنید و اگر آن نقطه دلخواه است به عنوان local minimum معرفی شود. زیرا یافتن global minimum در این حالت تقریباً غیر ممکن است. در شبکه عصبی که از gradient descent استفاده میکند شبیه همین فرآیند تکرار می شود یعنی اینکه تا جایی که با آموزش بیشتر شبکه مقدار خطا کاهشی نداشته باشد. اما توجه داشته باشید که این در صورتی قابل قبول است که مقدار خطایی که در آن خطا شما به local minimum دست پیدا کرده اید قابل قبول باشد.

اما training algorithm که در اینجا از Levenberg Marquardt یا LM استفاده شده است یک تکنیک تکرار کننده یا iterative technique است که مقدار minimum را برای یک تابع پیدا میکند. این تابع باید به صورت مجموع مربع خطاهای توابع غیر خطی با مقادیر حقیقی تعریف شود. این متد روش خوبی برای مسائل غیر خطی کمترین مربعات می باشد. برای توضیحات بیشتر به مقاله ای که در همین زمینه در فایل های ضمیمه قرار داده شده است مراجعه کنید.

نهایتاً transfer function تابعی است که رفتار نرون های ورودی-خروجی را تعیین میکند، در زیر سیگنوئیدال نشان داده شده است.



برای اینکه بتوان مدل شبکه عصبی را برای این مسئله تهیه کرد نیاز است تا برنامه مورد نیاز برای این مدل نوشته شود در زیر یکی از نمونه های برنامه نویسی شده در محیط MATLAB آورده شده است. که قابل استفاده برای مسائل مهندسی میباشد.

```
1-load cu.txt ;
2-P=cu(1:30,1:3);
3-T=cu(1:30,4);
4-a=cu(31:36,1:3);
5-s=cu(31:36,4);
6-[pn,minp,maxp,tn,mint,maxt]=premnmx(P',T);
7-[an,mina,maxa,sn,mins,maxs]=premnmx(a',s);
8-net=newff(minmax(pn),[6 1],{'tansig','purelin'},'trainlm');
9-net.trainParam.epochs=500;
10-net.trainParam.lr=0.3;
11-net.trainParam.mc=0.6;
12-net=train (net,pn,tn);
```

برای اینکه بهتر بتوان هر سطر برنامه را توضیح داد، از شماره گذاری استفاده شد.

۱- دستور load فایلی که به صورت txt در مسیری که نرم افزار نصب شده است قرار گیرد اکثرا در آدرسی شبیه زیر:

C:\Users\mahdi\Documents\MATLAB

برای اینکار داده های جدول ۱ را در نرم افزار Microsoft Excel قرار دهید سپس داده ها را با پسوند Text Tab delimited ذخیره کنید به آدرسی که در بالا داده شد. البته در اینجا نام فایل cu گذاشته شد.

۲- این دستور برای تعیین ورودی های شبکه است که همان متغیرهای تاثیر گذار در تعیین نرخ نفوذ TBM هستند. برای اینکار از مجموع ۳۶ داده، ۳۰ داده اول برای آموزش شبکه انتخاب شدند.

۳- این دستور مقادیر خروجی که مقادیر واقعی نرخ نفوذ هستند را تعیین میکند. باید توجه داشت که باید ۳۰ داده متناظر با داده های ورودی نیز برای این قسمت انتخاب شوند که اگر دستورات را طبق آن چیزی که اینجا آورده شده است انجام دهید به نتیجه دلخواه خواهید رسید.

۴- این دستور برای تعیین داده هایی که برای تست شبکه انتخاب شده اند. این همان قسمتی است که در قبل به آن "صحت آرایی" گفتیم. که نیاز است با داده های مشابه به تست شبکه عصبی پردازیم که ببینیم عملکرد شبکه بعد از آموزش چگونه است؟ این دستور وردی های داده ها برای تست را نشان میدهد. ۶ داده ی باقیمانده برای تست در نظر گرفته شدند.

۵- این دستور داده های واقعی نرخ نفوذ را تعیین میکند. داده های متناظر با ورودی. این داده ها هم تعداد ۶ تا هستند که برای تست شبکه نیاز است.

۶- برای اینکه شبکه عصبی بتواند بهتر داده ها را آنالیز کرده و آموزش ببیند نیاز است تا داده ها را به صورت نرمالایز شده به آن دهیم. چون بعضی از مواقع با رنج های متفاوتی از داده ها مثلاً بین ۱ تا ۵۰۰ مواجه هستیم که با نرمالایز کردن میتوان آنها را بین ۰ تا ۱ یا ۱- تا ۱ قرار داد. تابع معرفی شده در اینجا یعنی premnmx داده ها را بین ۱- و ۱ قرار میدهد.

۷- مشابه ۶ است برای داده هایی که می خواهیم برای تست شبکه از آنها استفاده کنیم.

۸- در این دستور معماری شبکه یعنی تعداد نرون ها در لایه های پنهان و همچنین نوع transfer function ها و الگوریتم آموزش نوشته میشود.

نکته: اگر میخواهید بیش از یک لایه پنهان استفاده کنید نیاز است تا تعداد نرون ها و transfer function آن را به دستور بالا اضافه کنید. به طور مثال اگر میخواهید یک شبکه ای با دو لایه پنهان که در لایه اول ۶ نرون و در لایه دوم ۳ نرون باشند و هر دو از transfer function نوع tansig استفاده کند میتوانید از دستور زیر استفاده کنید:



```
net=newff(minmax(pn),[6 3 1],{'tansig','tansig','purelin'},trainlm);
```

توجه کنید که trainlm همان الگوریتم آموزش LM است.

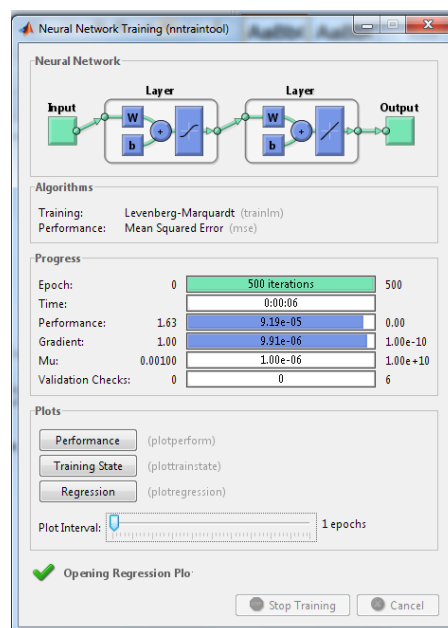
۹- تعداد epoch ها یا تعداد مراحل که شبکه برای آموزش پیشروی کند را اینجا تعیین میکنیم. البته خطای مورد نظر هم در تعیین تعداد epoch نیاز است.

۱۰- برای سرعت بخشیدن به فرآیند یادگیری از learning rate استفاده میشود.

۱۱- این دستور به تسریع فرآیند رسیدن شبکه به مقدار local minimum کمک میکند.

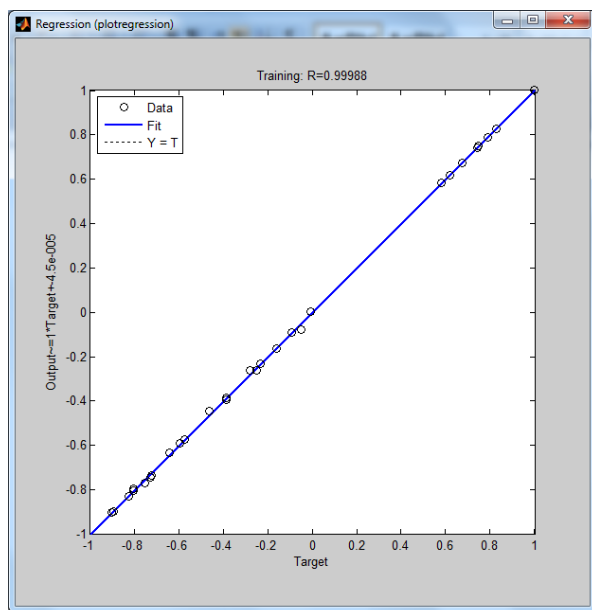
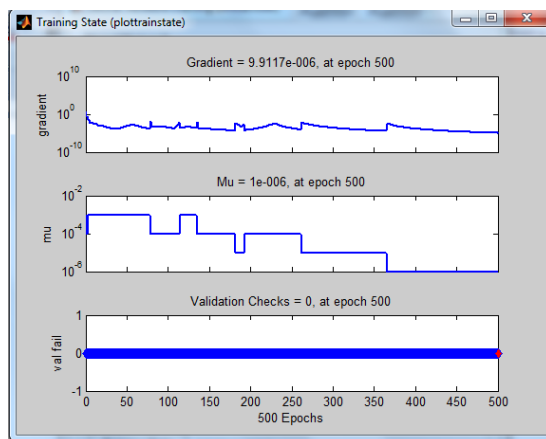
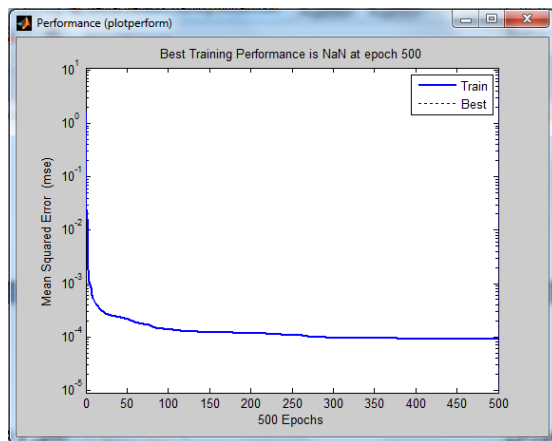
۱۲- نهایتاً با استفاده از این دستور آموزش شبکه آغاز میشود.

نتیجه ای که پس از اعمال این دستورات در نرم افزار MATLAB خواهید گرفت به صورت زیر است. که CORRELATION COEFFICIENT یا R استفاده شده است که مقدار آن 0.99 است. پس از اعمال دستورات بالا پنجره زیر ابتدا باز میشود:



در قسمت بالا (Neural Network) معماری شبکه به صورت نمایشی نشان داده شده است. در قسمت بعدی الگوریتم مورد استفاده برای آموزش و کنترل خطاها دیده میشوند. سپس فرآیندی که داده ها برای آموزش طی کرده اند در قسمت progress قابل مشاهده است. در قسمت Plot شما میتوانید performance یا همان میزان کاهش خطاها را در هر epoch ،

training state یا فرآیند آموزش شبکه و نهایتاً regression که R را بین خروجی شبکه عصبی و داده های واقعی (نرخ نفوذ TBM) نشان میدهد، مشاهده کنید.



پس از انجام فرآیند آموزش نیاز است که مدل شبکه عصبی که تهیه کرده اید را تست کنید. برای اینکار از دستورات زیر استفاده کنید:

```
y=sim(net,an);
```

برای simulation یا شبیه سازی شبکه از این دستور استفاده میشود.

```
t=postmnmx(y',mins,maxs);
```

عکس فرآیند نرمالایز سازی یعنی برگرداندن داده ها به حالت اصلیشان توسط این دستور قابل اجراست.

```
[t s]
```

```
plot(t,'r') ;
```

```
hold;
```

```
plot(s);
```

```
title('Comparison between actual targets and predictions');
```

این دستورات هم که از دستورات ابتدایی برای plot کردن نمودارهاست در نرم افزار MATLAB که امیدوارم شما این دستورهایی ساده را بدانید!! اگر هم مشکلی با هر کدام دارید دستور را کپی کرده و در قسمت help نرم افزار قرار دهید. توضیحات کامل را خواهید یافت.

```
d=[t-s].^2;
```

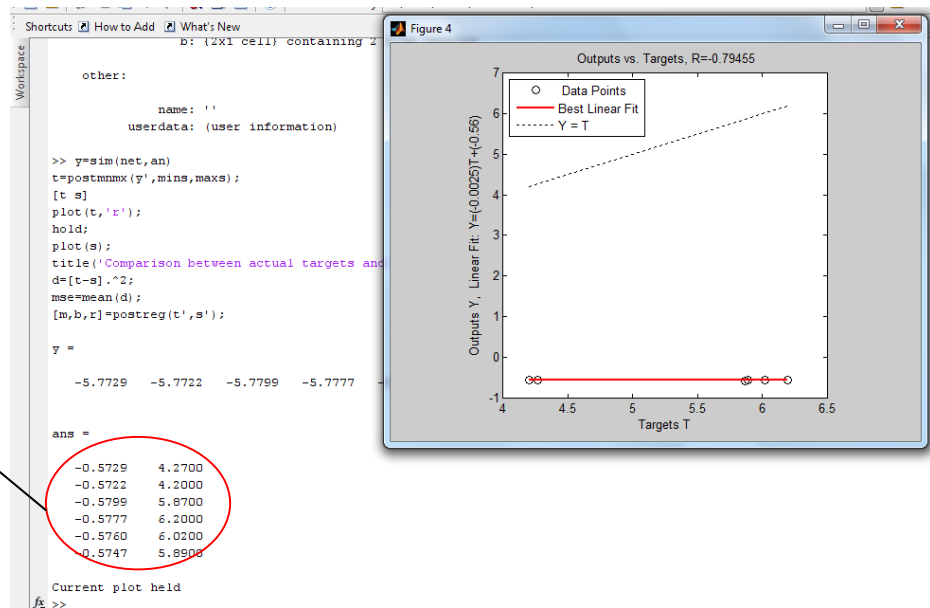
```
mse=mean(d);
```

این دو دستور برای تعیین mean square error استفاده میشود. که ساده است.

```
[m,b,r]=postreg(t',s');
```

نهایتاً برای اینکه ببینیم عملکرد شبکه چگونه بده است نیاز به تعیین مقدار R برای داده های تخمین زده شده توسط ANN و داده های واقعی داریم. که در شکل زیر مشاهده میشود. مقدار  $R=-0.79455$  نشان از smoth بودن داده ها دارد اما اگر به خرجی نگاهی بیندازید و آنها را با هم مقایسه کنید خواهید دید تفاوت زیادی بین خروجی شبکه و مقدر واقعی وجود دارد. پس نیاز است تا تعداد نرون ها در لایه های پنهان یا تعداد لایه های پنهان تغییر کنید. از اینجا به بعد به خسته کننده ترین فرایند کارهای شبکه عصبی میرسیم که یافتن شبکه بهینه است. توجه کنید که شبکه بهینه معما نیست که قابل حل باشد فقط با سعی وخطا قابل دستیابی است.

داده های خروجی شبکه  
عصبی پس از تست شبکه و  
داده های واقعی متناظر با  
آنها



در قسمت workspace نیز میتوانید مقادیر عددی هر پارامتری که خواستید را مشاهده کنید. کافیت روی آن دوبار کلیک کنید.

MATLAB 7.8.0 (R2009a)

File Edit View Graphics Debug Parallel Desktop Window Help

Shortcuts How to Add What's New

Workspace

Name	Value	Min	Max
P	<30x3 double>	22.8000	1010
T	<30x1 double>	3.8600	7.1200
a	<6x3 double>	26	1010
an	<3x6 double>	-1	29
ans	<6x2 double>	-0.5799	6.2000
b	<3x6 double>	-0.5619	-0.5619
cu	<36x4 double>	3.8600	1010
d	[23.4532;22.7744;41.6...	22.7744	45.9373
m	-0.0025	-0.0025	-0.0025
maxa	[29;29;1010]	29	1010
maxp	[29;30;1010]	29	1010
maxs	6.2000	6.2000	6.2000
maxt	7.1200	7.1200	7.1200
mina	[29;26;930]	26	930
minp	[22.8000;26;850]	22.8000	850
mins	4.2000	4.2000	4.2000
mint	3.8600	3.8600	3.8600
mte	36.5109	36.5109	36.5109
net	<1x1 network>		
pn	<3x30 double>	-1	1
r	-0.7945	-0.7945	-0.7945
s	[4.2700;4.2000;5.8700;...	4.2000	6.2000
sn	[-0.9300;-1.0;6700;1.0...	-1	1
t	[-0.5729;-0.5722;-0.57...	-0.5799	-0.5722
tn	<1x30 double>	-1	1
y	[-5.7729;-5.7722;-5.77...	-5.7799	-5.7722

Start