نام خدا

تمرین چهارم درس هوش مصنوعی

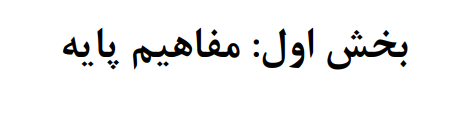
محمد امین سلطانی چم حیدری

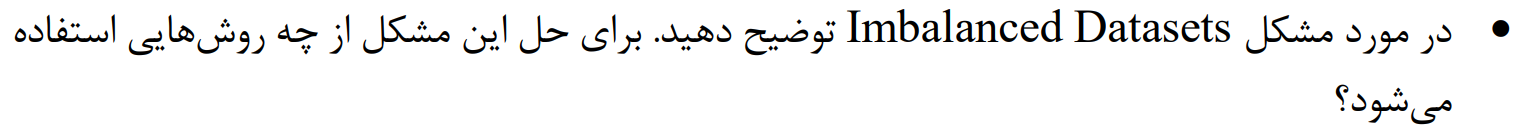
810601081

استاد مربوطه:

دکتر مسعود شریعت پناهی

بهار1402

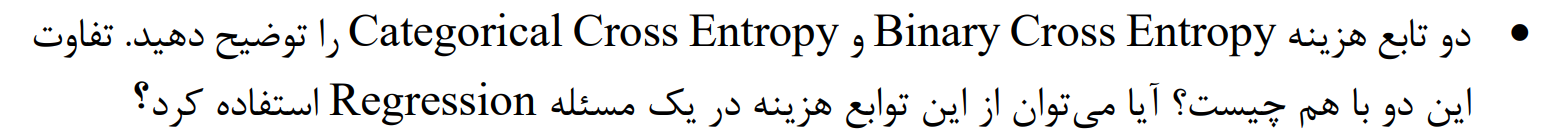




داده های نامتعادل یا Imbalanced Datasets به آن دسته از مجموعه داده ها اشاره دارد که در آن کلاس هدف دارای توزیع نابرابر مشاهدات است، یعنی یک برچسب کلاس تعداد مشاهدات بسیار بالایی دارد و دیگری تعداد مشاهدات بسیار کمی دارد. با یک مثال می توانیم مدیریت نامتعادل مجموعه داده را بهتر درک کنیم.

بیایید فرض کنیم که مثلا بانک ملی برای مشتریان خود کارت اعتباری صادر می کند. اکنون بانک نگران است که برخی تراکنش‌های تقلبی در حال انجام است و وقتی بانک داده‌های آنها را بررسی می‌کند متوجه شد که برای هر 2000 تراکنش فقط 30 شماره کلاهبرداری ثبت شده است. بنابراین، تعداد کلاهبرداری در هر 100 تراکنش کمتر از 2٪ است یا می توان گفت بیش از 98٪ تراکنش ماهیت "بدون کلاهبرداری" دارد. در اینجا، کلاس "بدون تقلب" را طبقه اکثریت، و کلاس "تقلب" بسیار کوچکتر، طبقه اقلیت نامیده می شود. عدم توازن کلاس معمولاً در مسائل طبقه بندی طبیعی است. اما، در برخی موارد، این عدم تعادل در جایی که حضور طبقه اکثریت بسیار بیشتر از طبقه اقلیت است، کاملاً حاد است. در موارد نادری مانند تشخیص تقلب یا پیش‌بینی بیماری، شناسایی صحیح طبقات اقلیت حیاتی است. بنابراین مدل نباید برای تشخیص فقط طبقه اکثریت مغرضانه باشد، بلکه باید به طبقه اقلیت نیز وزن یا اهمیت مساوی بدهد. در اینجا برخی از چند تکنیکی را که می توانند با این مشکل مقابله کنند، ذکر شده اند.

1. **نسبتی که با آن داده های test و train جدا شده اند را تغییر داد. مثلا یک بار داده ها به نسبت 0.8 و 0.2 جدا شوند و یکبار با نسبت 0.7 و 0.3 .**
2. **استفاده از K-fold Cross-Validation به روش درست.**
3. **تغییر داده های train یا به اصطلاح resample آن ها برای آموزش.**
4. **استفاده از معیارهای ارزیابی درست و مناسب برای مسئله. مثلا برای مسائل دسته بندی با داده های نامتوازن معیار accuracy مناسب نیست زیرا این معیار کلاسی که نمونه های آن بیشتر است را بهتر از سایر کلاس ها در نظر می گیرد و انتخاب می کند.**

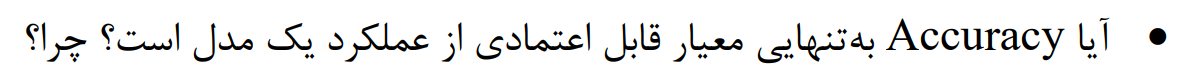


(categorical cross entropy) و (binary cross entropy) دو تابع خطا بسیار مرسوم در شبکه‌های عصبی هستند. categorical cross entropy برای مسائل طبقه‌بندی چند دسته‌ای استفاده می‌شود. این تابع خطا، تفاضل بین توزیع احتمالاتی پیش‌بینی شده توسط شبکه و توزیع احتمالاتی واقعی برچسب‌های داده‌ها را محاسبه می‌کند. با استفاده از این تابع خطا، شبکه عصبی به شیوه‌ای که بتواند توزیع احتمالی برچسب‌های داده‌های جدید را حساب کند، آموزش می‌یابد.

binary cross entropy برای مسائل طبقه‌بندی دو دسته‌ای (binary classification) استفاده می‌شود. این تابع خطا، تفاضل بین احتمال پیش‌بینی شده توسط شبکه برای دسته‌ی مثبت و دسته‌ی منفی و احتمالات واقعی برچسب‌های داده‌ها است. با استفاده از این تابع خطا، شبکه عصبی به شیوه‌ای که بتواند احتمال اینکه یک داده برای دسته‌ی مثبت باشد یا نباشد را حساب کند، آموزش می‌یابد.

تفاوت این دو تابع در این است که از categorical cross entropy برای مسائلی استفاده می‌شود که تعداد دسته‌های طبقه‌بندی بیشتر از ۲ باشد، در حالی که از binary cross entropy برای مسائلی استفاده می‌شود که تنها دو دسته در آن‌ها وجود داشته باشد. به عنوان مثال، اگر بخواهیم تصویری را به کلاس‌های "سگ" و "گربه" و "پرنده" تقسیم کنیم، از تابع خطای categorical cross entropy استفاده خواهیم کرد. اگر بخواهیم تصویری را به کلاس‌های "بد" و "خوب" تقسیم کنیم، از تابع binary cross entropy استفاده خواهیم کرد.

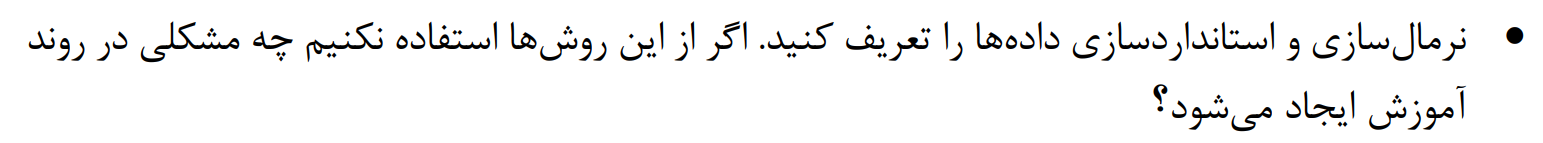
هم چنین از این دوتابع خطا برای مسائل رگرسیون استفاده نمی شود زیرا خروجی رگرسیون پیوسته است در حالی که این دوتابع خطا یک بردار احتمالاتی تعیین میکنند و خروجی آن ها گسسته است.



در بسیاری از مسائل یادگیری ماشین، معیار دقت (Accuracy) به تنهایی معیار قابل اعتمادی از عملکرد یک مدل نیست و نباید به عنوان تنها معیار برای ارزیابی عملکرد مدل استفاده شود. در بسیاری از مسائل، احتمال اینکه داده‌های ورودی به هر کدام از کلاس‌ها تعلق دارند، نامتوازن است. به عبارت دیگر، تعداد داده‌های مربوط به یکی از کلاس‌ها نسبت به کلاس‌های دیگر بسیار بیشتر است. در چنین حالتی، دقت به تنهایی معیاری قابل اعتماد برای ارزیابی عملکرد مدل نیست، زیرا این معیار تمایل دارد کلاسی که تعداد نمونه‌های آن بیشتر است را بهتر از کلاس‌های دیگر تشخیص دهد.

در بسیاری از مسائل، نوع خطا نیز بسیار مهم است. به عنوان مثال، در یک مسئله پزشکی، تشخیص درست یک بیماری خطرناک می‌تواند بسیار مهم باشد و خطا در این موضوع می‌تواند عواقب جبران ناپذیری داشته باشد. در این حالت، دقت یا accuracy به تنهایی نمی‌تواند به عنوان معیار قابل اعتمادی برای ارزیابی عملکرد مدل استفاده شود.

بنابراین، برای ارزیابی عملکرد یک مدل، بهتر است از چندین معیار ارزیابی استفاده کرد و نه تنها از دقت یا Accuracy. معیارهای دیگری مانند حساسیت (Sensitivity)، دقت پیش‌بینی مثبت (Positive Predictive Value) و دقت پیش‌بینی منفی (Negative Predictive Value) نیز برای ارزیابی عملکرد مدل مورد استفاده قرار می‌گیرند.

نرمالسازی و استانداردسازی داده‌ها دو روش است که در پیش‌پردازش داده‌ها برای استفاده در مدل‌های یادگیری ماشین استفاده می‌شوند. استفاده از نرمالسازی و استانداردسازی داده‌ها در بسیاری از مدل‌های یادگیری ماشین مفید است. این روش‌ها به عنوان یک پیش‌پردازش داده‌ها می‌توانند بهبود کیفیت و سرعت آموزش مدل‌ها را بهبود بخشند.

**نرمالسازی داده‌ها:** در این روش، داده‌ها به گونه‌ای تغییر داده می‌شوند که در بازه‌ی صفر تا یک باشند. به طوری که بزرگترین داده برابر یک و کوچک ترین داده برابر صفر و سایر داده ها در بازه بین این دو عدد باشند.

**استانداردسازی داده‌ها:** در این روش، داده‌ها به گونه‌ای تغییر داده می‌شوند که میانگین آنها صفر و واریانس آنها یک شود. به این ترتیب داده ها بین یک و منفی یک می باشند.

اگر از این روش‌ها در پیش‌پردازش داده‌ها استفاده نشود، مشکلاتی مانند :

\_ مقیاس‌پذیری ناصحیح داده‌ها

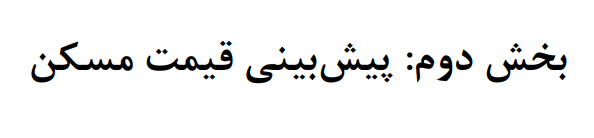
\_ تغییرات زیاد و عدم پایداری در مقادیر ورودی مدل

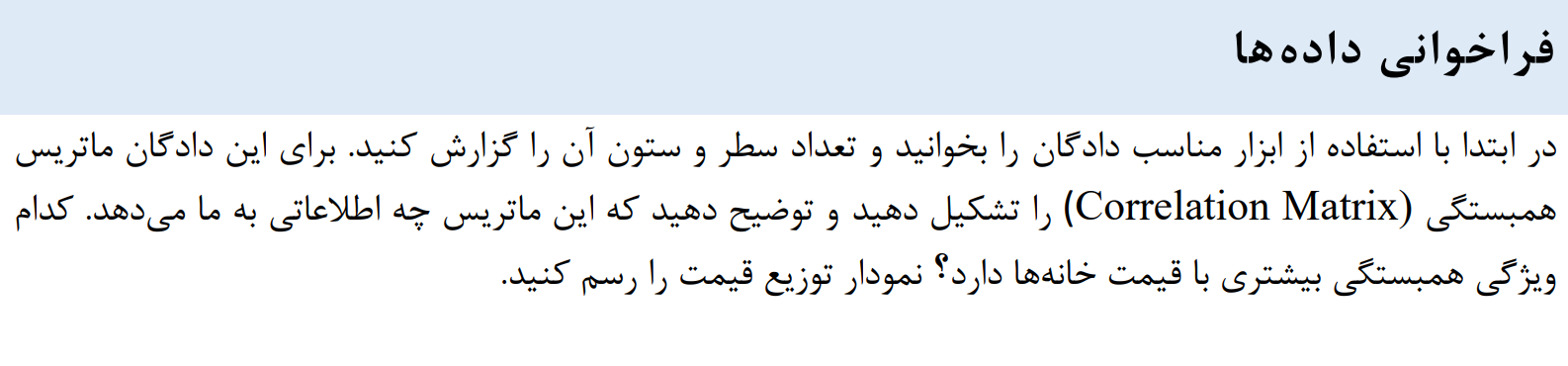
\_ کاهش سرعت آموزش

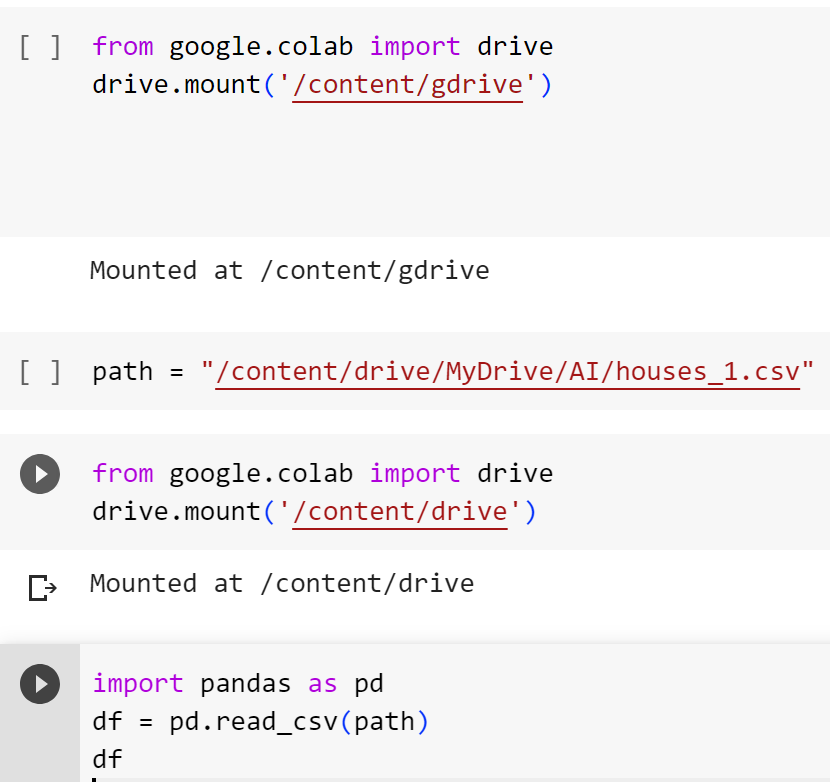
\_ حساسیت شبکه به مقادیر بزرگ و کوچک

ممکن است در تربیت مدل بوجود بیاید. همچنین در شبکه‌های عمیق، ممکن است مشکلاتی مانند اضافه شدن مقادیر بزرگ و کوچک به ورودی‌های لایه‌های بعدی وجود داشته باشد که باعث شود پارامترهای شبکه عصبی به طور کلی بیشتر از حد معمول بوده و موجب شود تا شبکه به دقت بسیار پایینی برسد.

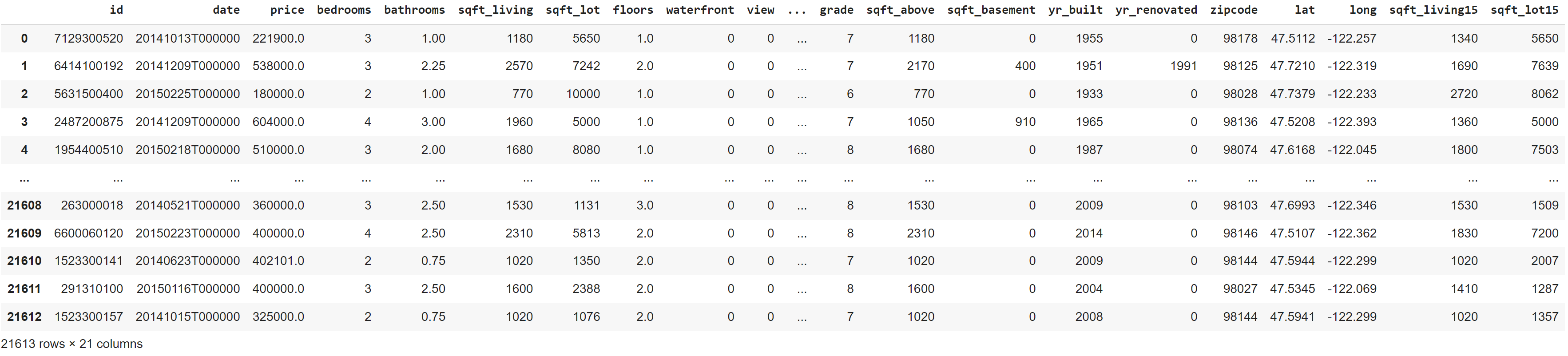
بنابراین، نرمال‌سازی داده‌های ورودی به شبکه عصبی مصنوعی برای بهبود عملکرد شبکه بسیار مهم است و می‌تواند باعث بهبود دقت و سرعت آموزش شود.



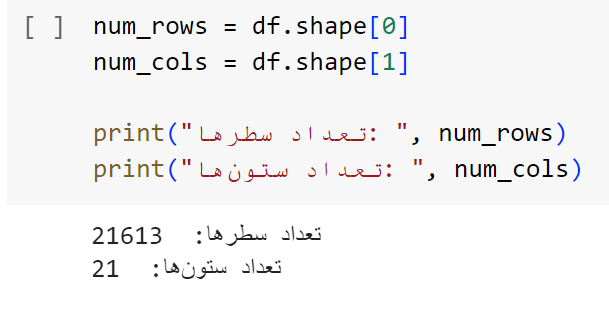




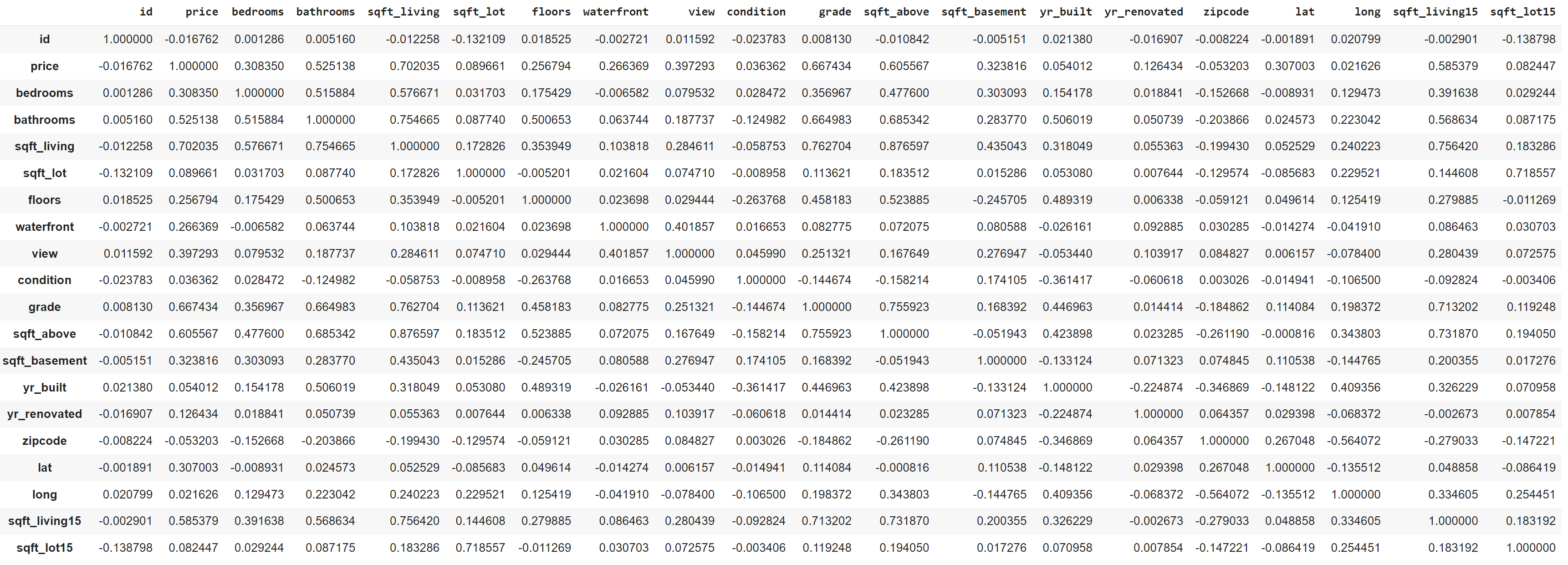
با استفاده از دستور بالا داده ها از گوگل درایو فراخوانی شده اند.



داده های فراخوانی شده به صورت بالا قابل مشاهده هستند.

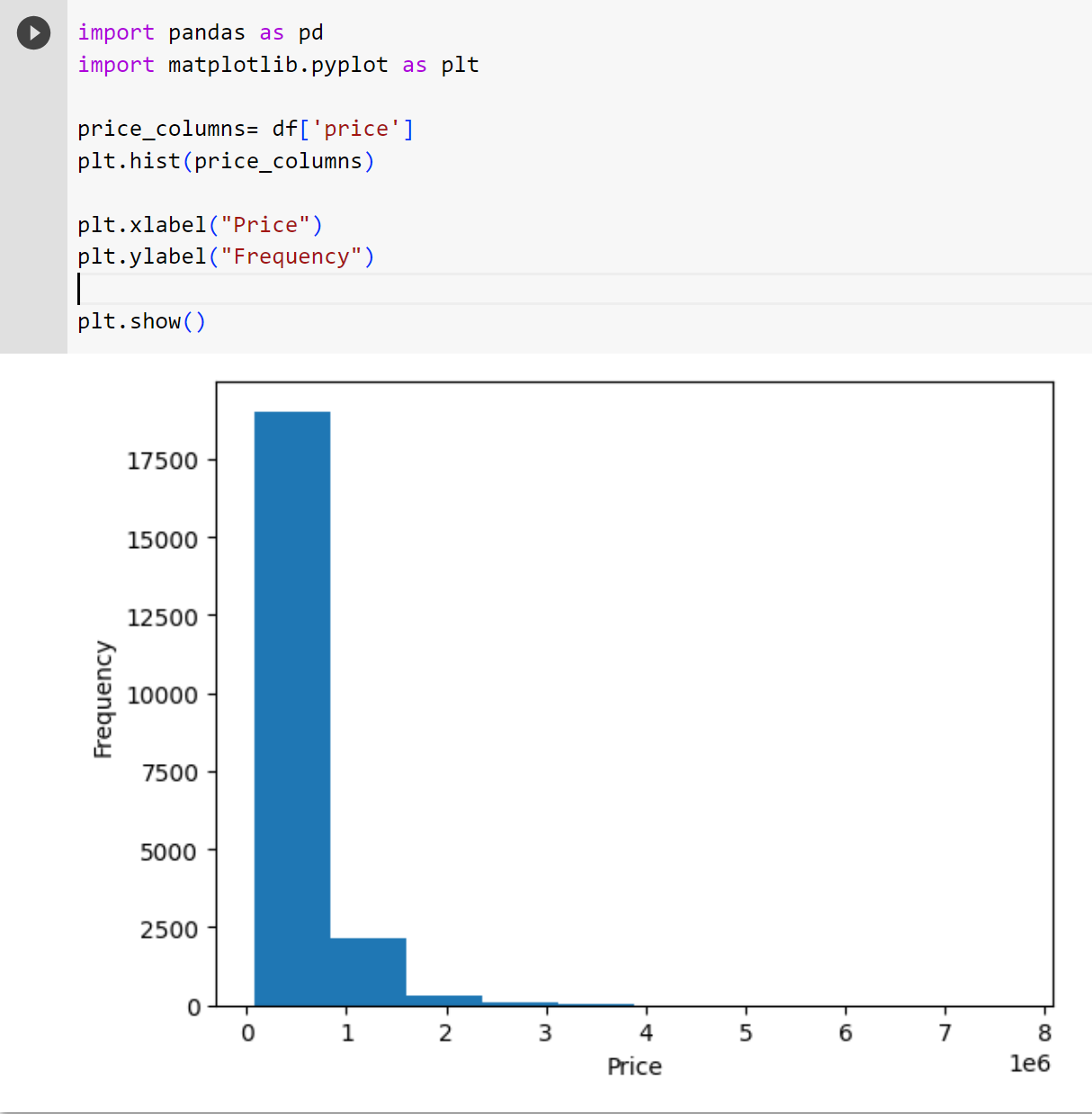


**Corelation matrix:**

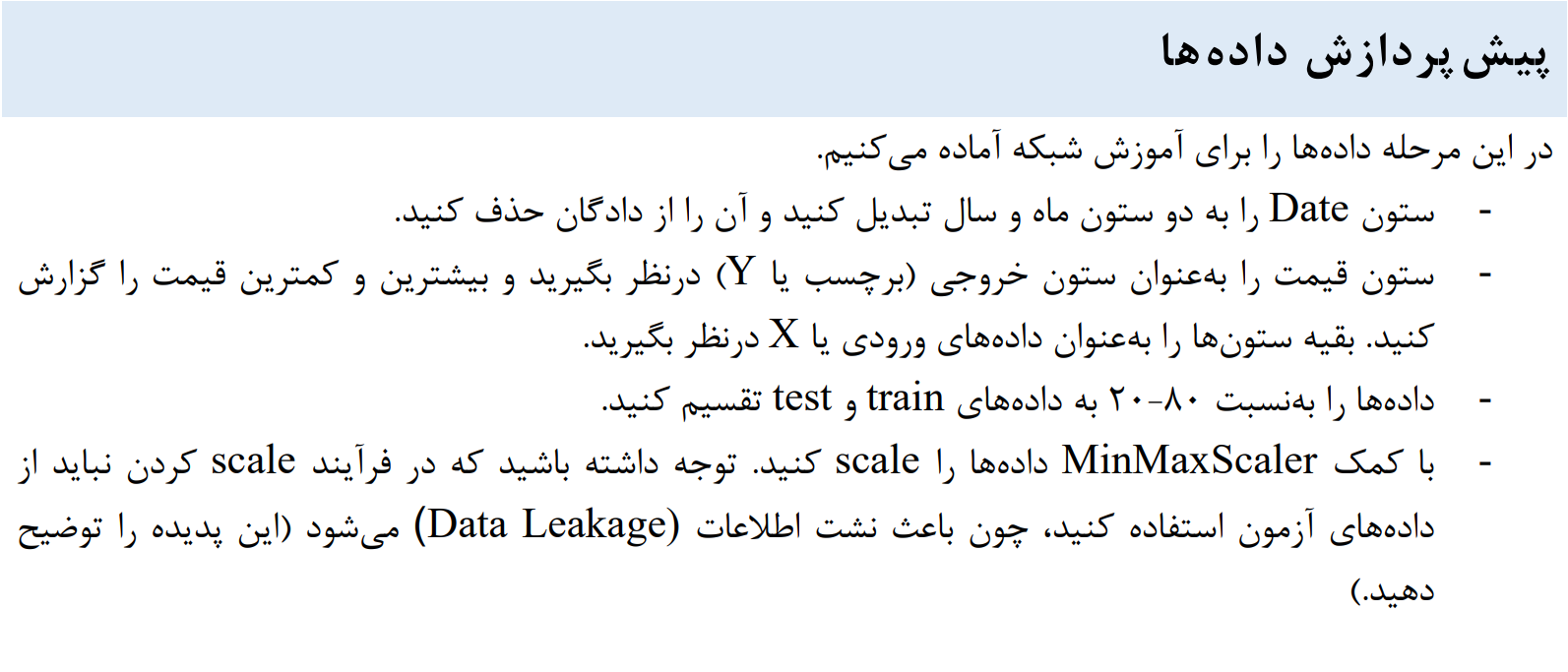


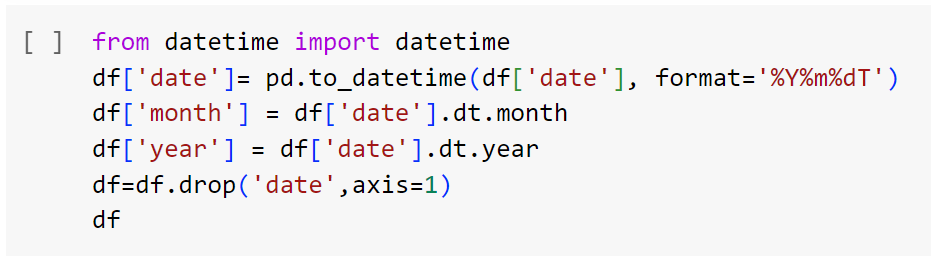
ماتریس correlation نشان‌دهنده قدرت و جهت رابطه بین دو متغیر می باشد. هر درایه در ماتریس correlation، نشان می‌دهد که دو متغیر چقدر با یکدیگر رابطه دارند و این رابطه به چه جهتی است. اگر مقدار correlation بین دو متغیر برابر با یک باشد، این به این معنی است که دو متغیر با یکدیگر رابطه خطی دارند و افزایش یکی، باعث افزایش دیگری خواهد شد. اگر مقدار correlation بین دو متغیر برابر با منفی یک باشد، این به این معنی است که دو متغیر با یکدیگر رابطه خطی دارند، اما افزایش یکی، باعث کاهش دیگری خواهد شد. اگر مقدار correlation بین دو متغیر برابر با صفر باشد، این به این معنی است که دو متغیر با یکدیگر رابطه خطی ندارند. این به معنی این است که تغییر در یکی از متغیرها، تغییری در مقدار دیگری از متغیرها ایجاد نمی‌کند. همچنین هم بستگی هر متغیر با خودش برابر یک می باشد.

برای بررسی اینکه کدام ویژگی همبستگی بیشتری با قیمت خانه ها دارد باید ستون مربوط به price بررسی شود. با بررسی این ستون مشاهده می شود که ویژگی **sqft\_living** بیشترین همبستگی را با ستون price دارد.

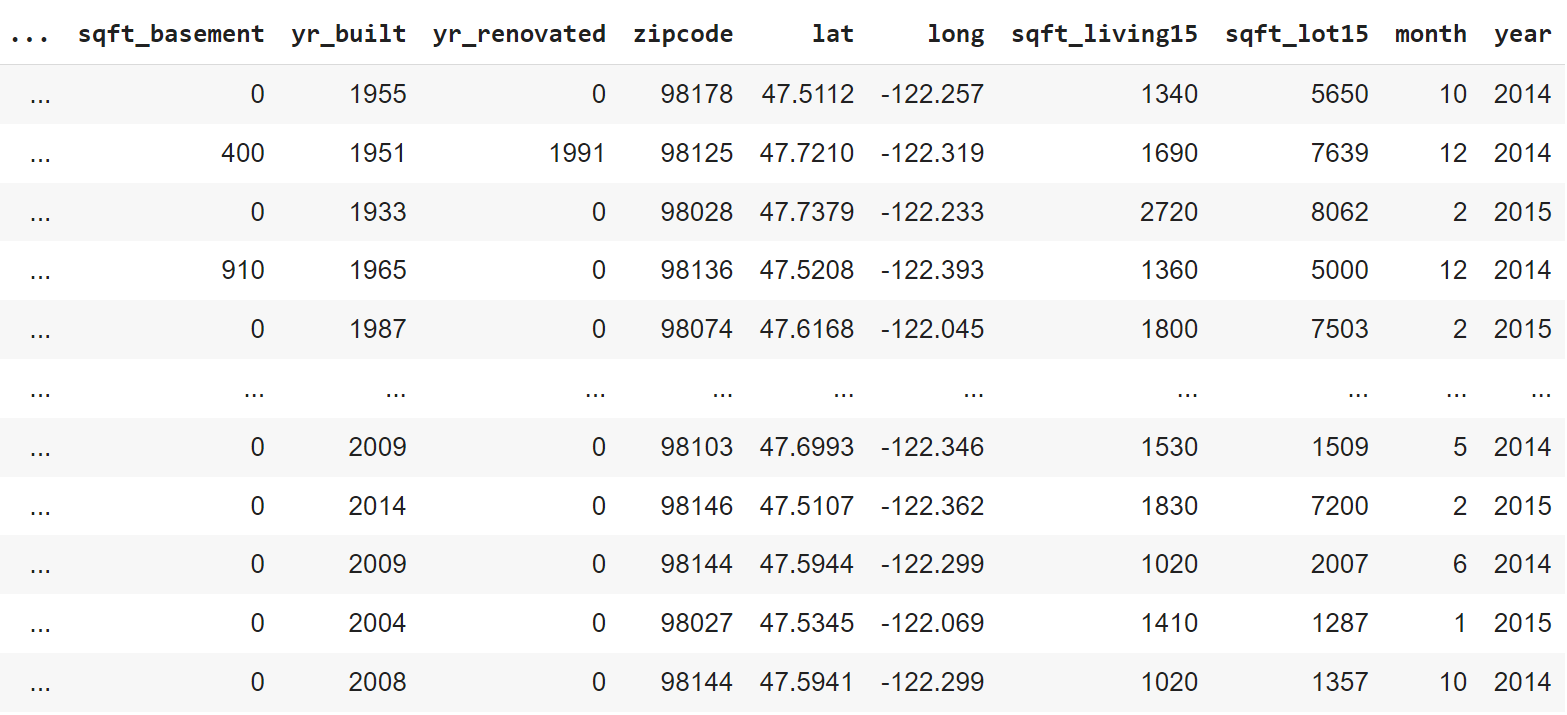


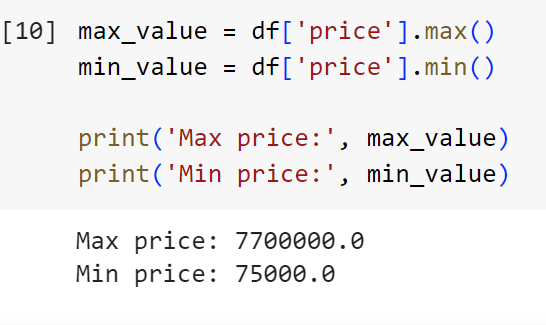
نودار توزیع قیمت



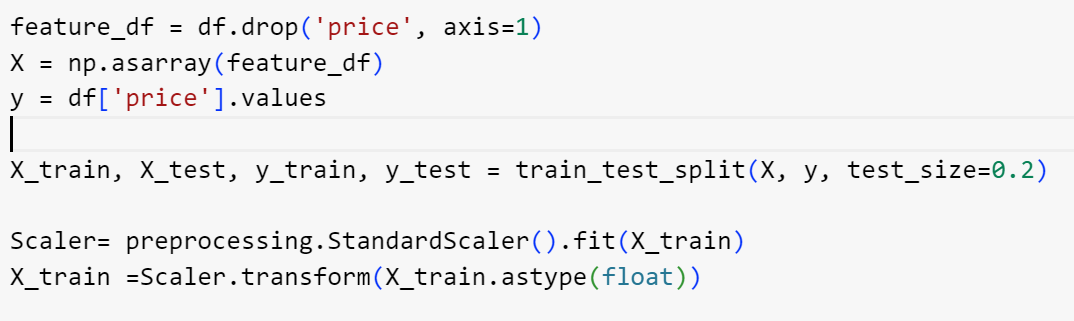


با استفاده از دستور datetime داده های ستون date به دو ستون year و month تقسیم می شوند و در فایل داده ها ذخیره می شوند.





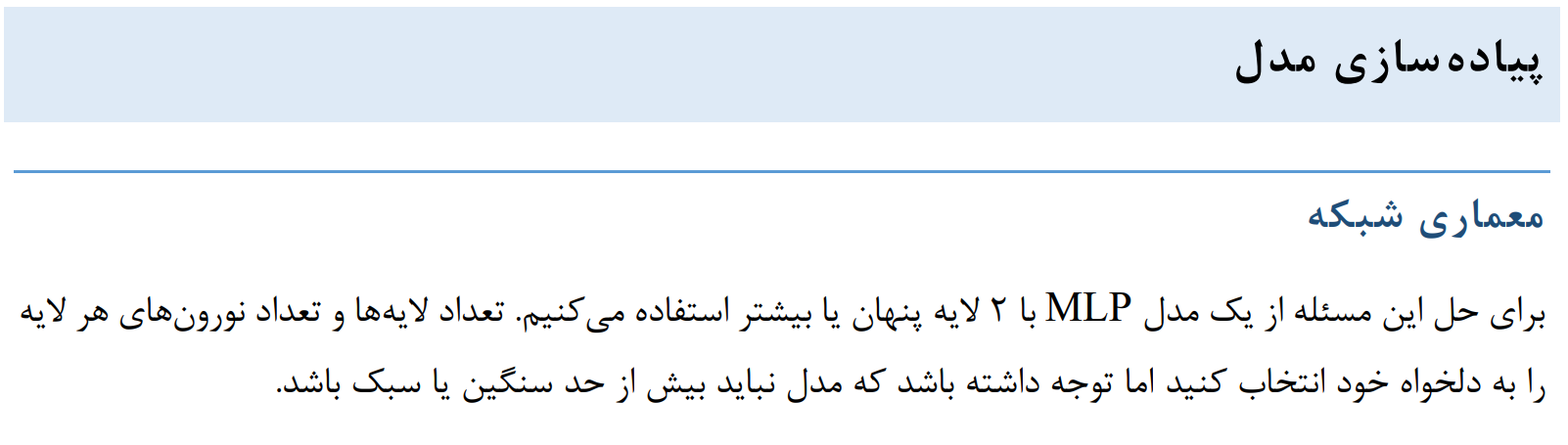
کمترین و بیشترین قیمت در ستون price با کد بالا گزارش شده است.

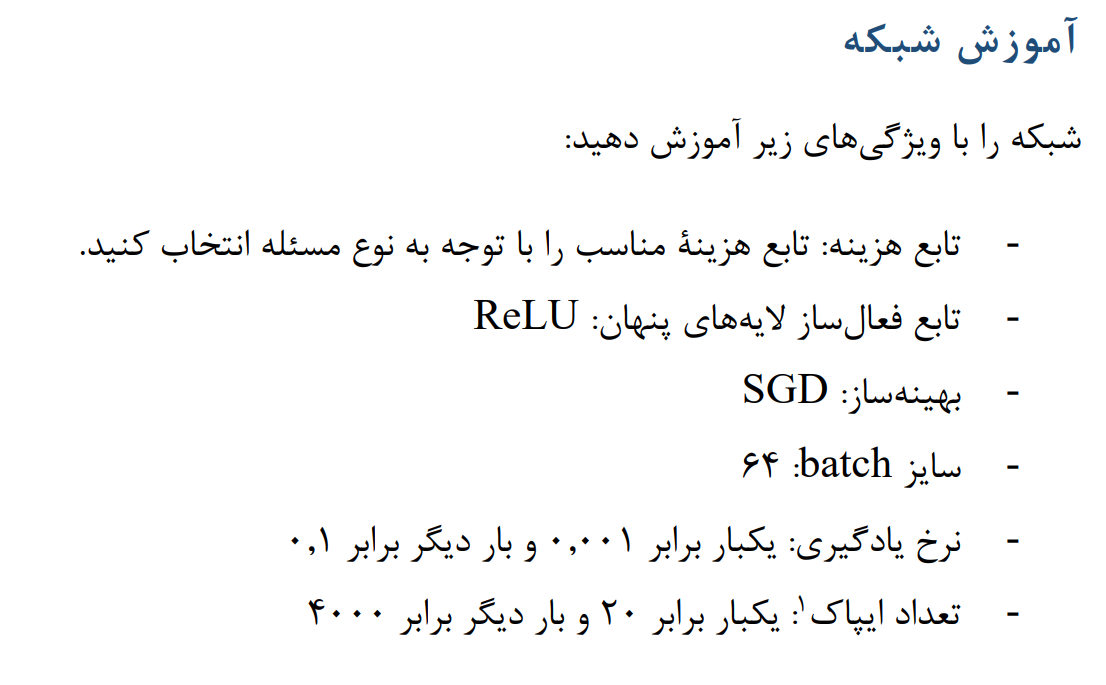


سپس ستون price به عنوان خروجی در نظر گرفته شده. همچنین سایر ستون ها به عنوان ویژگی در نظر گرفته شده اند و در متغیر های X\_train , X\_test , y\_train , y\_test با نسبت 80-20 ریخته می شوند.

سپس داده های train با استفاده از دستور minmaxscaler داده ها scale می شوند. باید توجه شود که برای جلوگیری از data leakage فقط باید داده های train را scale کرد.

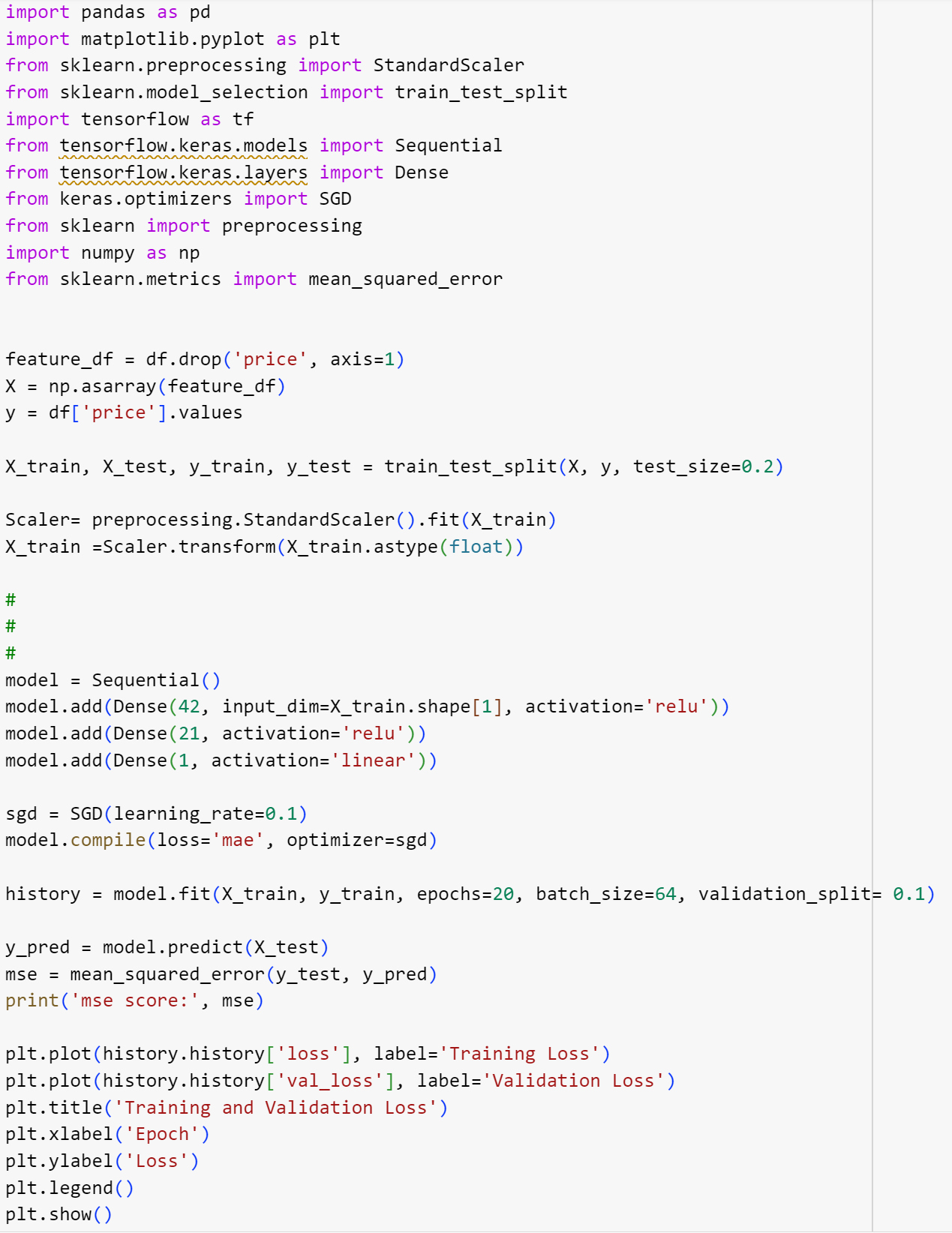
**Data Leakage** به هر نوع انتقال اطلاعات بین مجموعه داده‌های آموزشی و مجموعه داده‌های تست اطلاق می‌شود که باعث می‌شود عملکرد مدل در مجموعه داده‌های تست بهتر از واقعیت باشد. به عبارت دیگر، این پدیده زمانی به وجود می‌آید که اطلاعاتی از مجموعه داده‌های تست به مجموعه داده‌های آموزشی دسترسی دارد و از آن‌ها استفاده می‌کند، عملکرد مدل در مجموعه داده‌های تست به طور اشتباهی بهبود پیدا می‌کند. از آنجا که Data Leakage می‌تواند باعث بهبود موهومی عملکرد مدل در مجموعه داده‌های تست شود و باعث کاهش دقت و اعتبار مدل در محیط واقعی شود، باید از آن پرهیز کرد و در طراح مدل به دقت و اعتبار لازم در محیط واقعی رسید.





تابع هزینه: برای این مدل تابع هزینه min square error یا mse در نظر گرفته شده است.

همچنین از آنجایی **که تعداد نورون لایه پنهان حداقل باید به اندازه تعداد ویژگی های ورودی باشد** و در این مسئله با اضافه کردن ستون year و month در کل 21 ویژگی وجود دارد تعداد نورون لایه پنهان اول برابر با 42 و تعداد نورون لایه پنهان دوم برابر 21 در نظر گرفته شده است.



در ابتدا کتابخانه های مورد نیاز برای تربیت شبکه عصبی فراخوانی می شوند.

matplotlib 🡨 نمایش data و رسم نمودار

pandas 🡨 کار با داده های ساختار یافته مانند جداول،فایل csv و excel و...

Numpy 🡨 انجام عملیات ریاضی،آماری و عددی

Sickit\_learn 🡨 یادگیری ماشین و تحلیل داده

Pylab 🡨 تجسم داده های علمی و عملی.ترکیبی از matplotlib و numpy می باشد.

Tensorflow 🡨 تربیت مدل شبکه عصبی

Dense 🡨 تعریف تعداد لایه ها و نورون های شبکه

SGD 🡨 بهینه ساز شبکه از نوع کاهش گرادیانی.

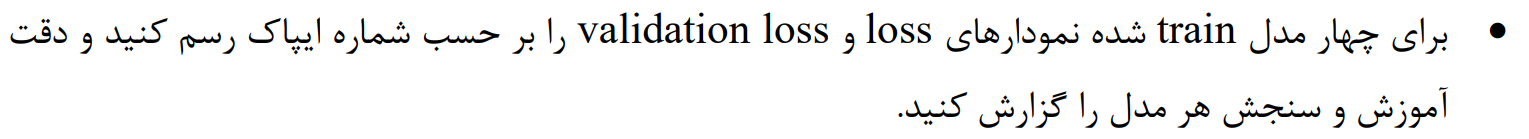
برای ساخت مدل شبکه عصبی با استفاده از دستور sequential یک مدل تعریف شده است که در آن سه لایه وجود دارد. دولایه اول لایه های مخفی با تعداد نورون 42 و 21 می باشند و لایه سوم که لایه خروجی است دارای یک نورون می باشد.

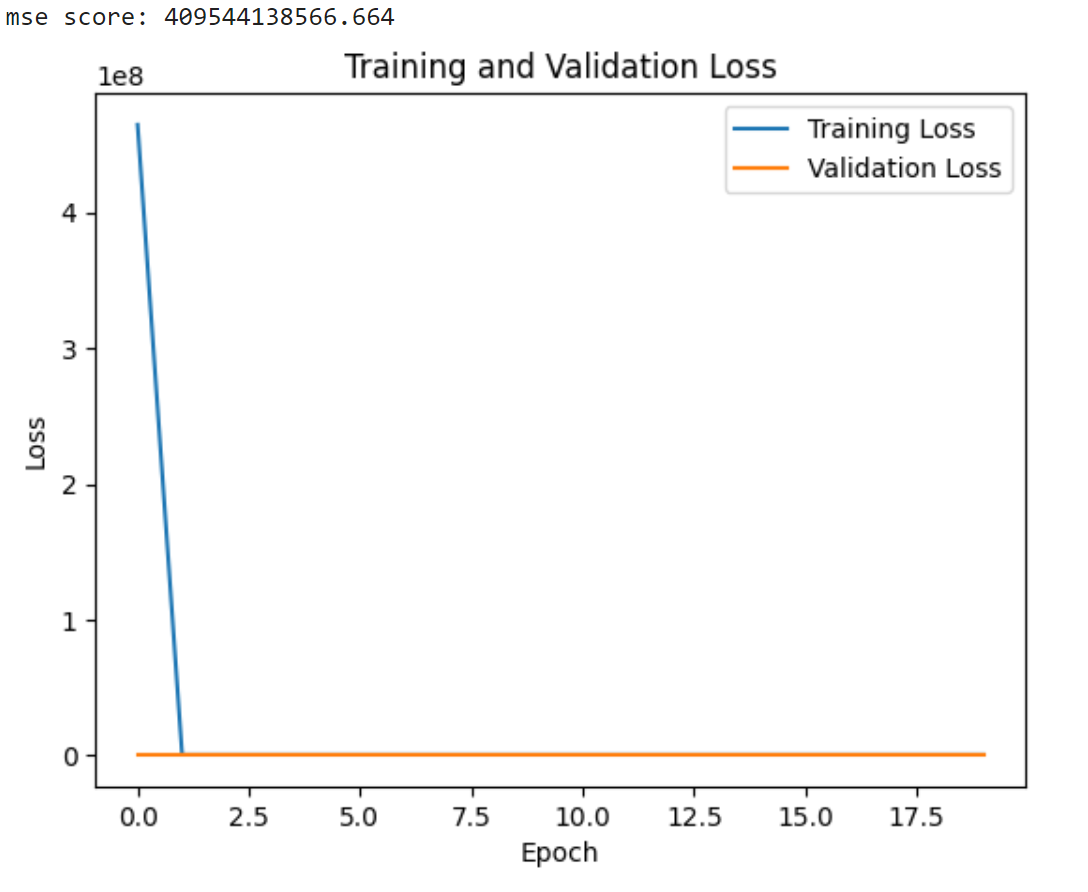
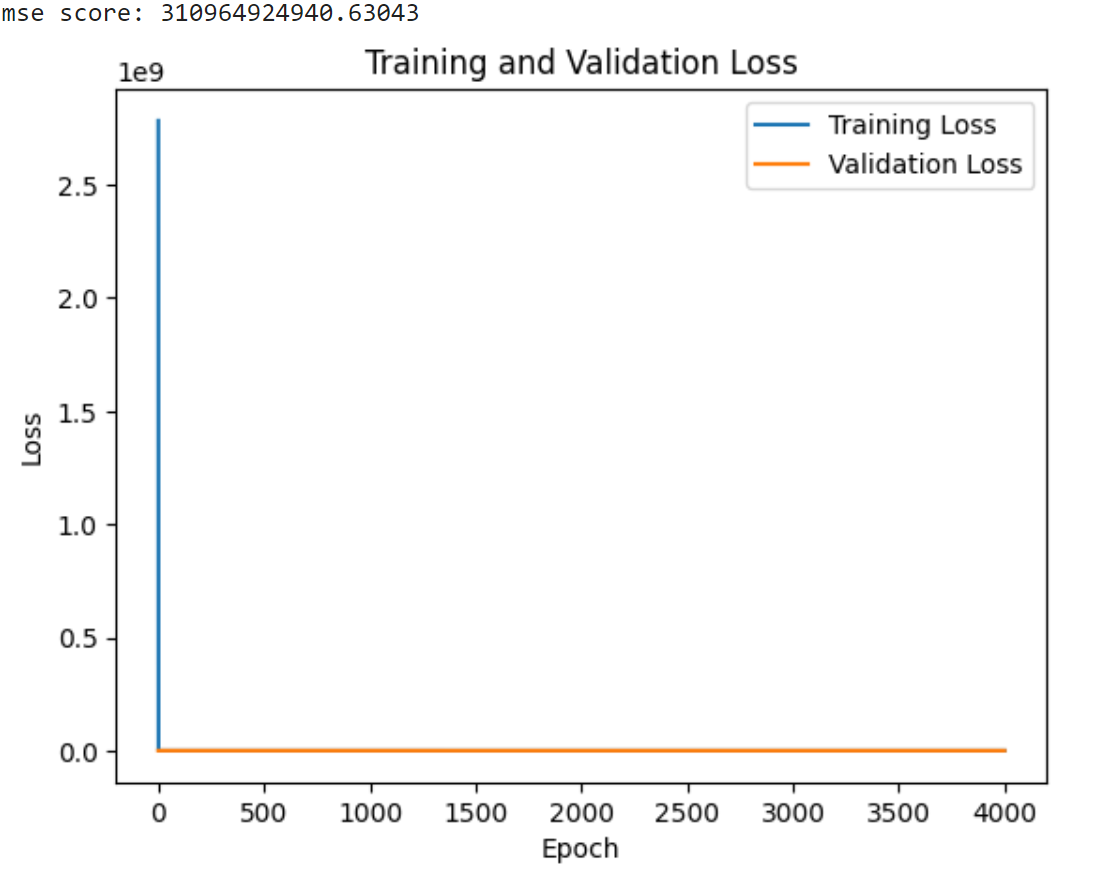
دستور sequential از کتابخانه Transflow و با دستور keras فراخوانی شده است.

تابع فعال ساز برای لایه های پنهان مطابق خواسته ی سوال relu در نظر گرفته شده است و تابع فعال سازی لایه خروجی از نوع linear می باشد.

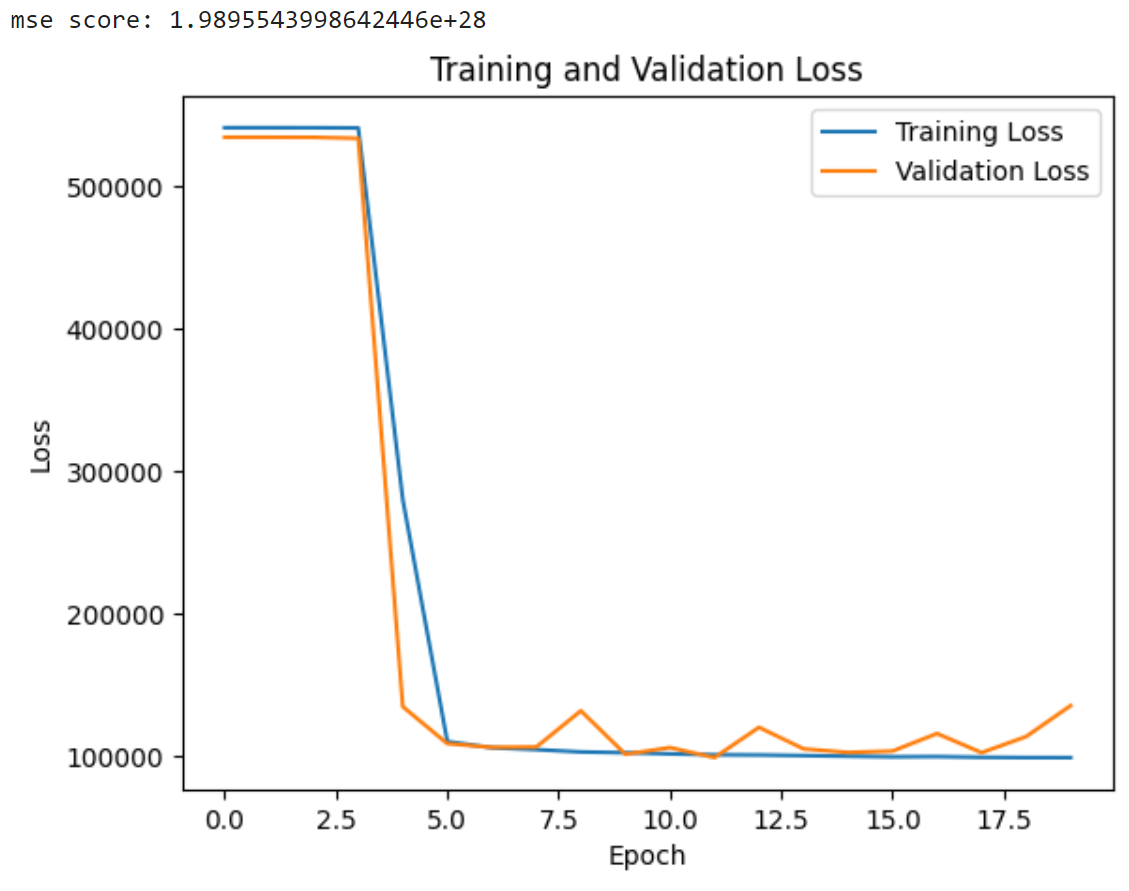
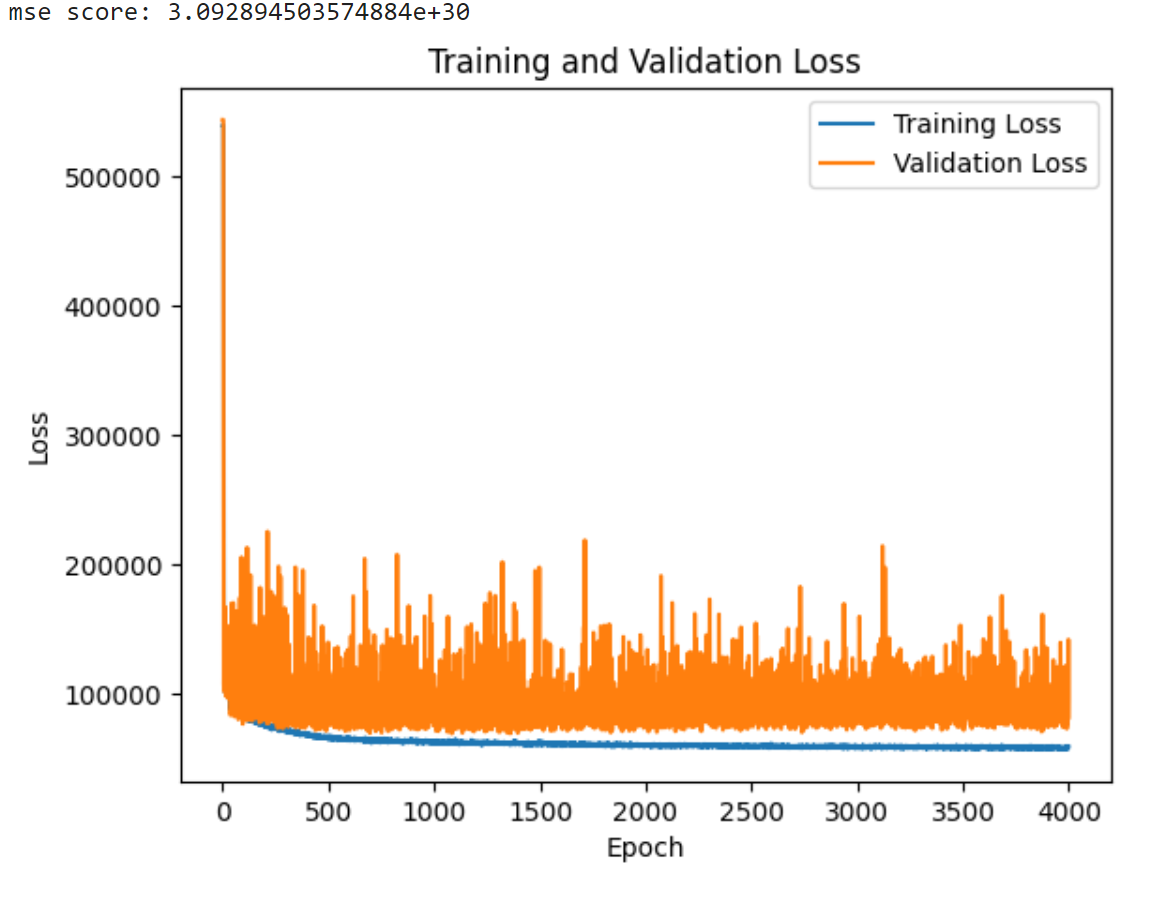
سپس بهینه ساز از نوع SGD که بهینه ساز با روش کاهش گرادیانی می باشد تعریف می شود و نرخ یادگیری برابر با 0.1 در نظر گرفته می شود.

در پایان مدل شبکه عصبی تعریف شده بر روی داده های train اعمال می شود تا شبکه تربیت شود و نمودار های loss و validation loss برای هر حالت رسم شود.



Learning rate = 0.1 with 20 epoch Learning rate = 0.1 with 4000 epoch

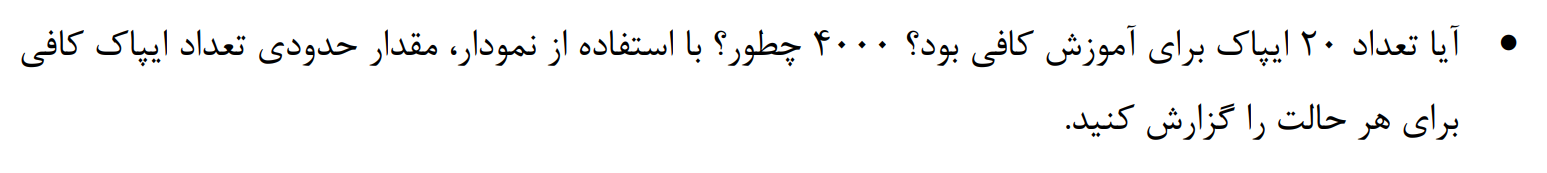
Learning rate = 0.001 with 20 epoch Learning rate = 0.001 with 4000 epoch

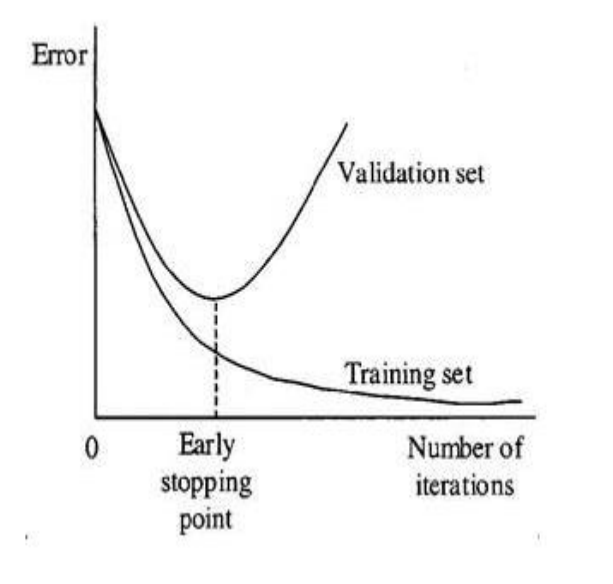
**مقایسه مجموع خطای 4 مدل:**

|  |  |  |
| --- | --- | --- |
| mse | epoch | Learning rate |
| 4\*e11 | 20 | 0.1 |
| 3\*e11 | 4000 | 0.1 |
| 3\*e30 | 20 | 0.001 |
| 2\*e28 | 4000 | 0.001 |

Mean square error در چهار حالت مطابق بررسی قرار گرفت که بهترین مدل مربوط به نرخ یادگیری 0.1 و تعداد ایپاک 4000 می باشد زیرا کمترین خطا را دارند.

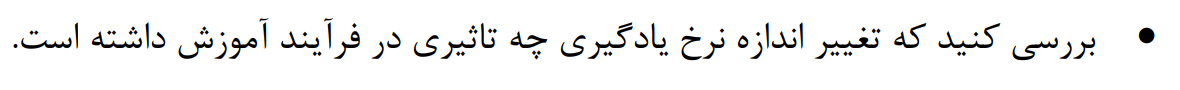
همانگونه که مشاهده می شود با کاهش نرخ یادگیری مقدار خطا همواره افزایش می یابد این به ان معناست که شبکه بهبود نمی یابد.



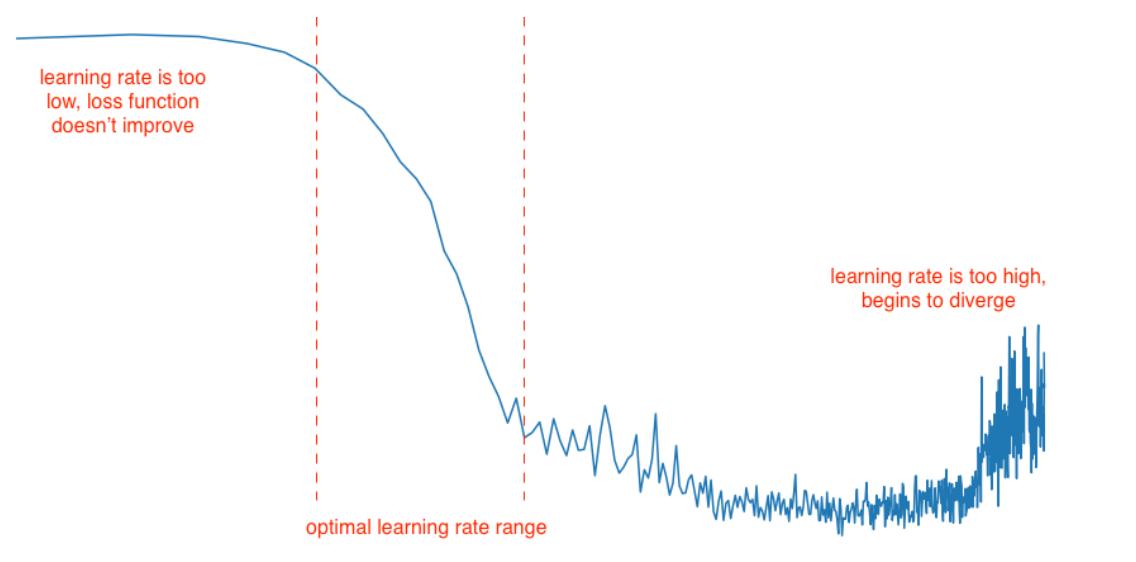


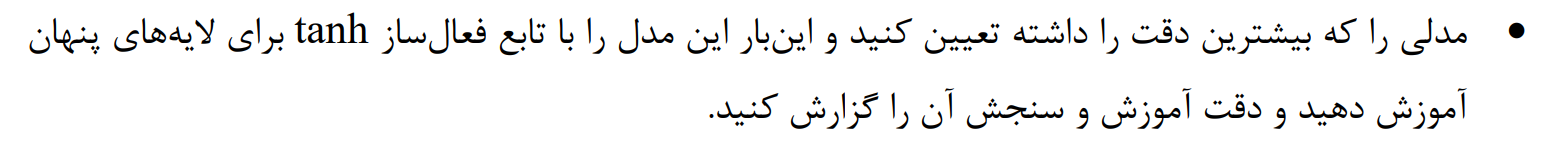
**اگر معیار توقف شبکه تعداد ایپاک باشد باید توجه داشت که هم شبکه دچار بیش بردازش نشود و هم به میزان کافی تربیت شود که پیش بینی های آن صحیح باشد**. لذا چون در این جا برای پایان تربیت از شرط توقف early stopping استفاده نشده است باید از روی نمودارهای بدست آمده محل واگرا شدن نمودار loss و validation loss را پیدا کرد و تعداد ایپاک در آن نقطه را یاد داشت کرد.این تعداد ایپاک تعداد ایپاک بهینه برای تربیت مدل می باشد که در جدول زیر تعداد ایپاک مناسب برای هر حالت نشان داده شده است.

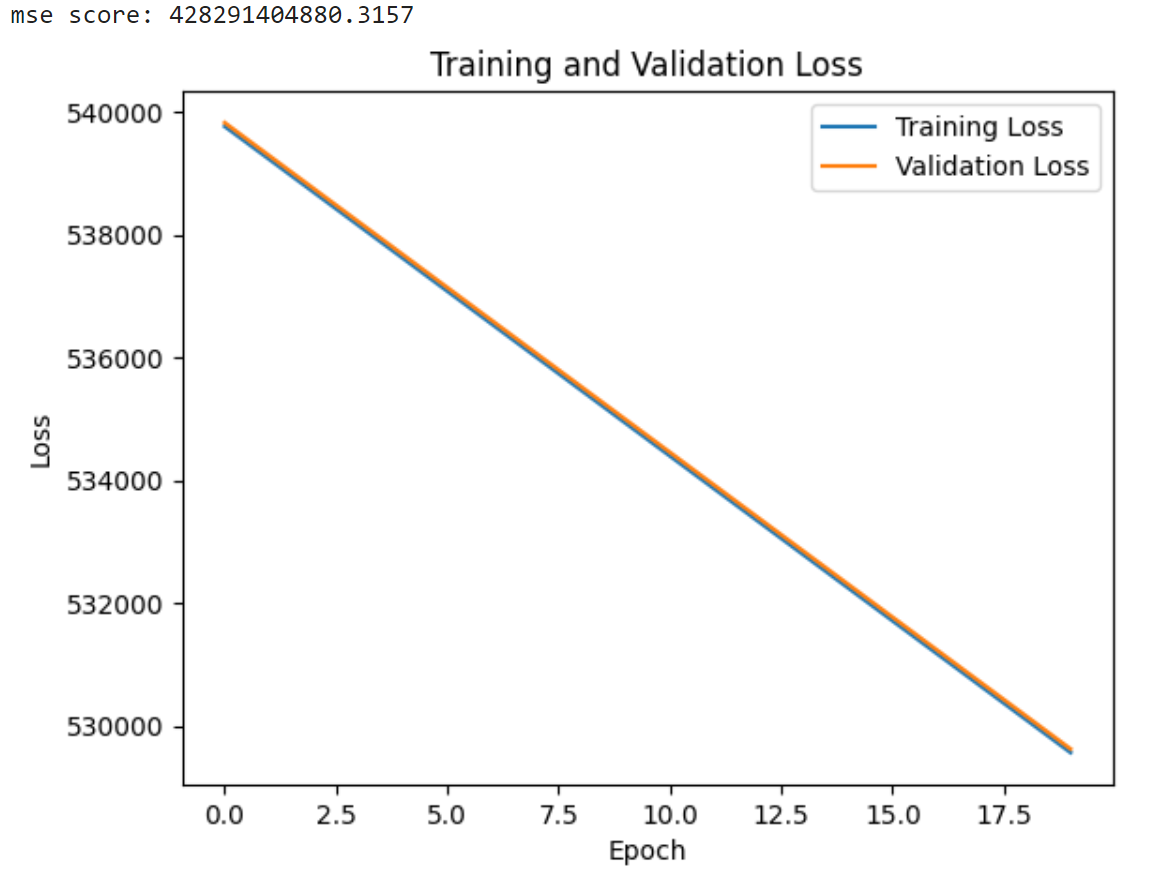
|  |  |
| --- | --- |
| Epoch enough | Learning rate |
| More than 4000 | 0.1 |
| About 200 | 0.001 |



همانگونه که در جدول گزارش مجموع خطا قابل مشاهده است با کاهش نرخ یادگیری در تعداد ایپاک مشخص شبکه بهبود نمی یابد و مجموع خطای آن افزایش می یابد. با افزایش بیش زیاد نرخ یادگیری نیز نمودار به صورت نامتناوب و ناپایدار بدست می اید به اینصورت که با اضافه شدن تاثیر هرضریب وزنی مقدار خطا تغییر زیادی می کند(مثال معروف غوره و کشمش).

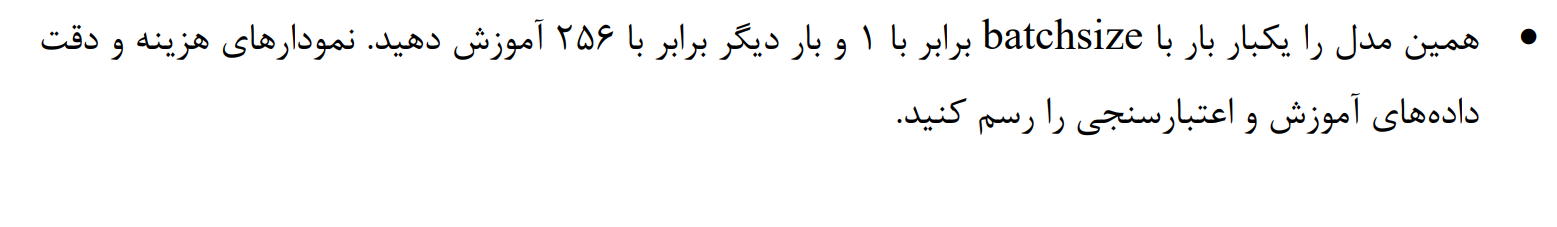


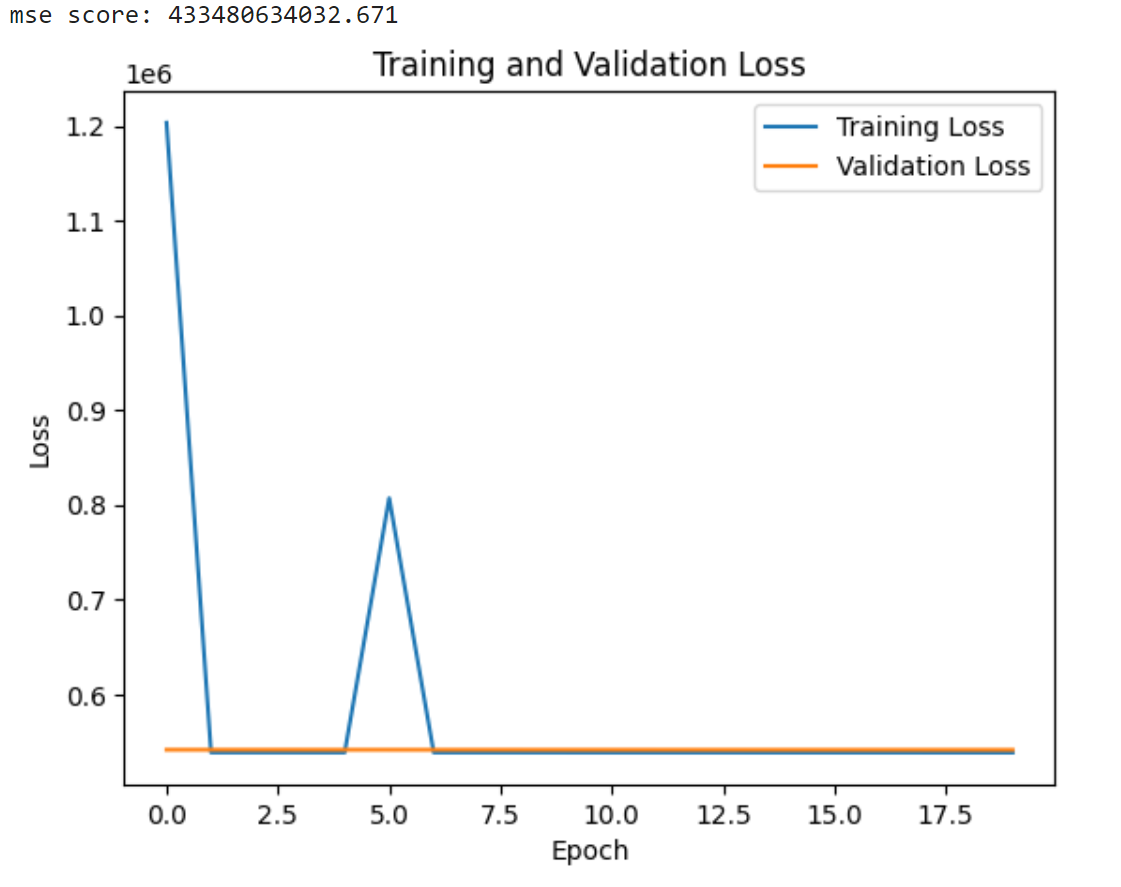




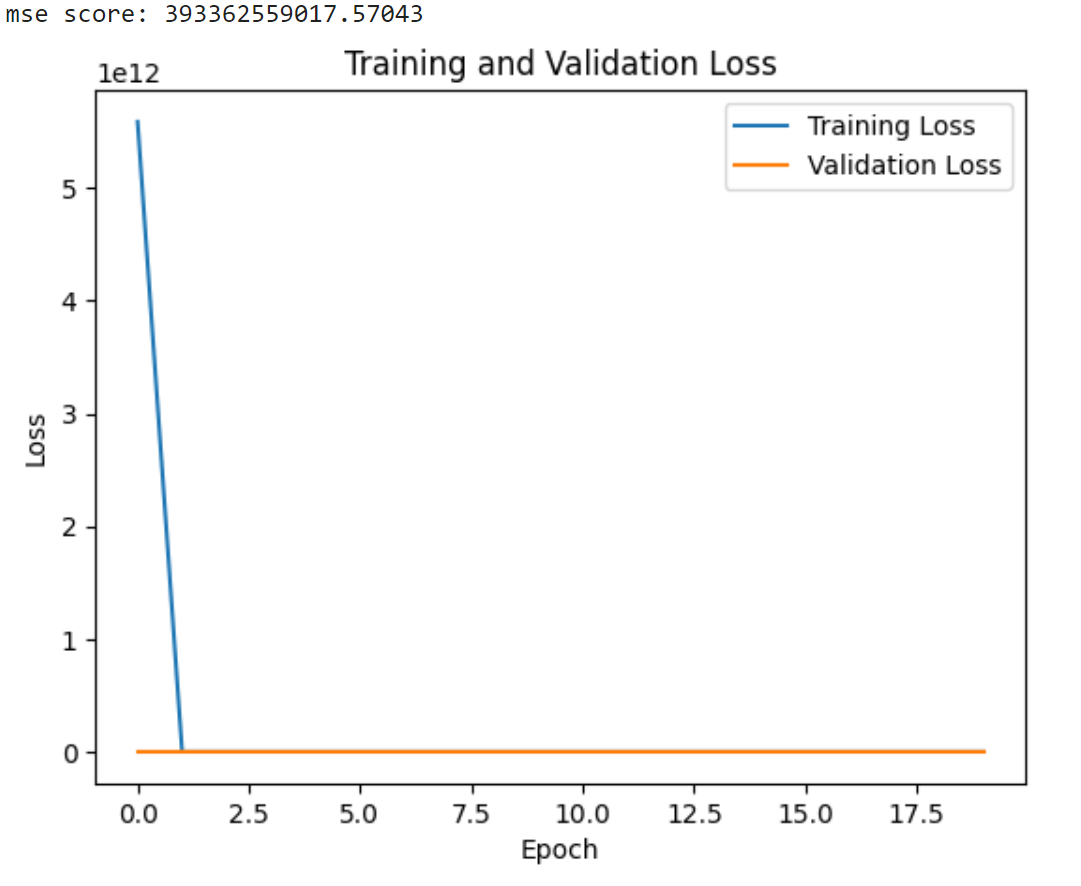
با توجه به اینکه تابع فعال ساز tanh یک فعال ساز صفر و یک است و مقادیر ورودی را به صفر و یک تصویر می کند این مدل با تابع فعال ساز tanh خطای بیشتری را حاصل می کند یا به عبارتی دقت شبکه عصبی با تابع فعال ساز tanh کاهش می یابد.

مجموع خطای محاسبه شده با این تابع فعال ساز در حدود 4.2\*e11 می باشد.





**Batch size = 256**

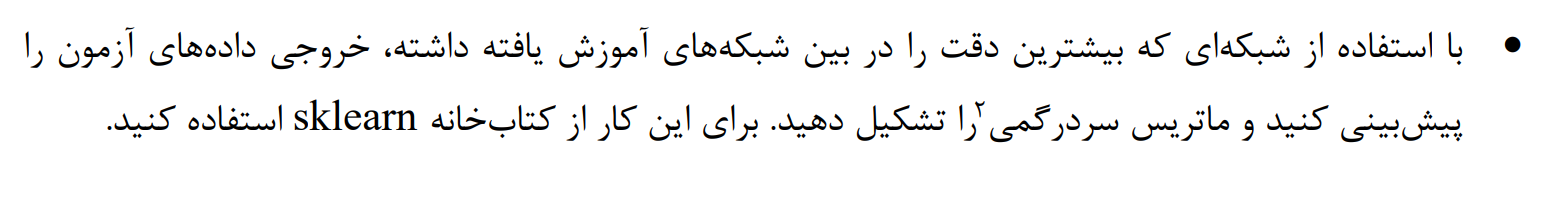


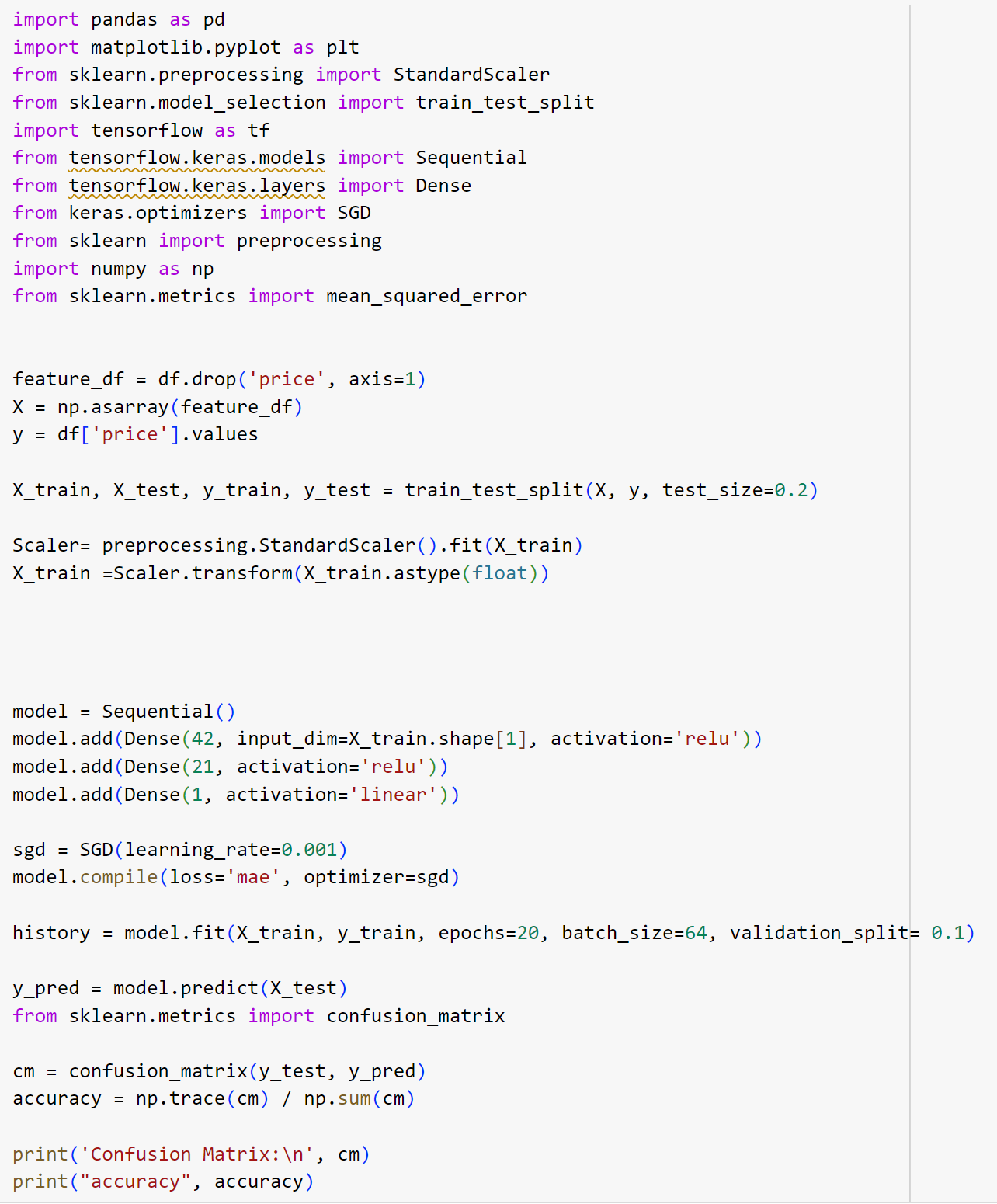
**Batch size = 1**

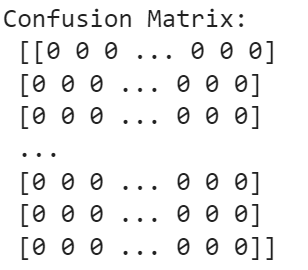
Batch size مربوط به اندازه دسته ها در روش کاهش گرادیانی یا gradient descent می باشد.

کاهش مقدار batch size موجب پایداری بیشتر نمودار خطا یا به عبارتی پایداری بیشتر در آموزش شود. همچنین batch size کوچک تر تعداد iteration ها و در نتیجه زمان حل برای تربیت مدل را بسیار افزایش می دهد.

از طرفی batch size بزرگتر موجب افزایش سرعت تربیت و کاهش دقت مدل شبکه عصبی می شود.



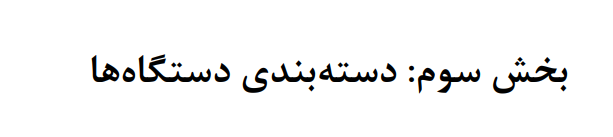


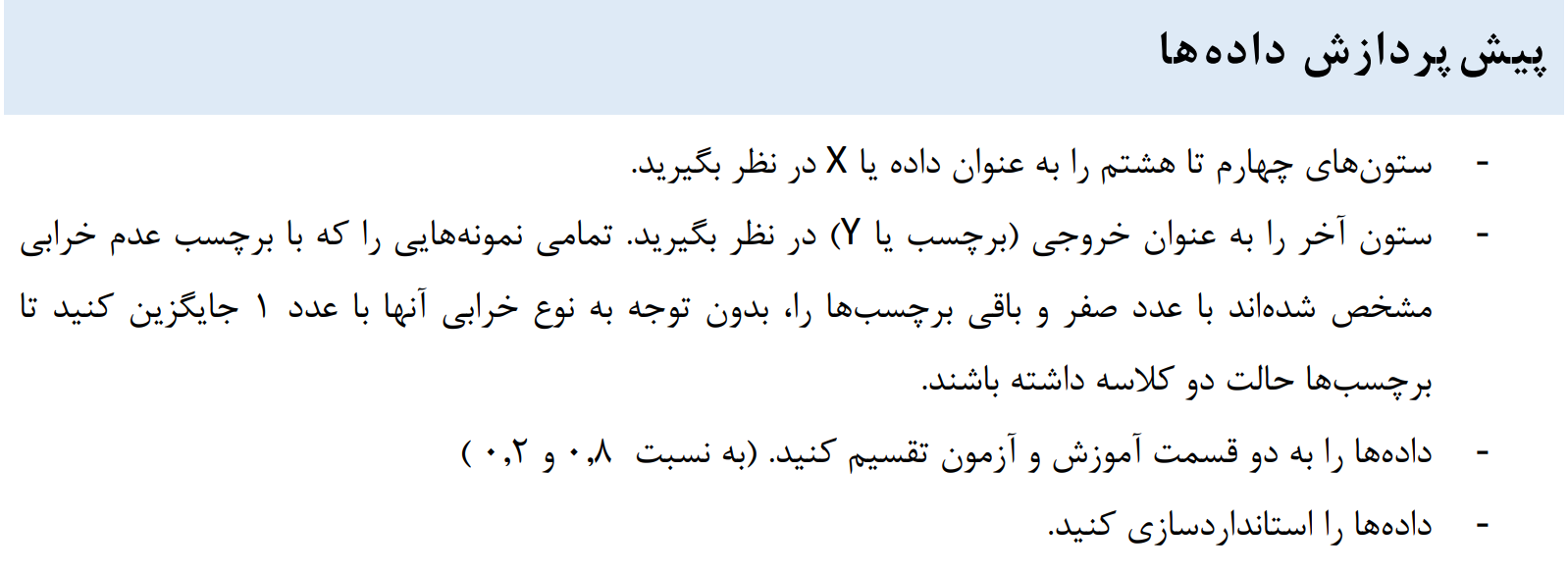


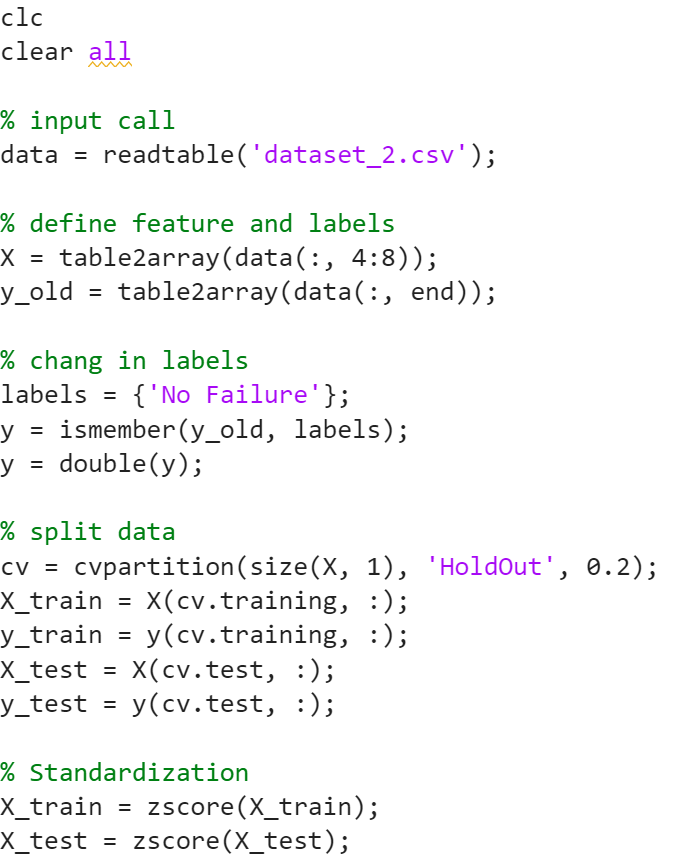
ماتریس سردرگمی برای بهترین مدل محاسبه شده است و درایه های آن قابل مشاهده است که با توجه به زیاد بودن تعداد سطر و ستون آن در شکل بالا تصادفا فقط درایه های صفر نمایش داده شده است.

برای سنجش دقت این مدل accuracy معیار مناسبی نمی باشد زیرا داده های ما توزیع یکسانی ندارند و به اصطلاح ما یک imbalanced dataset داریم.

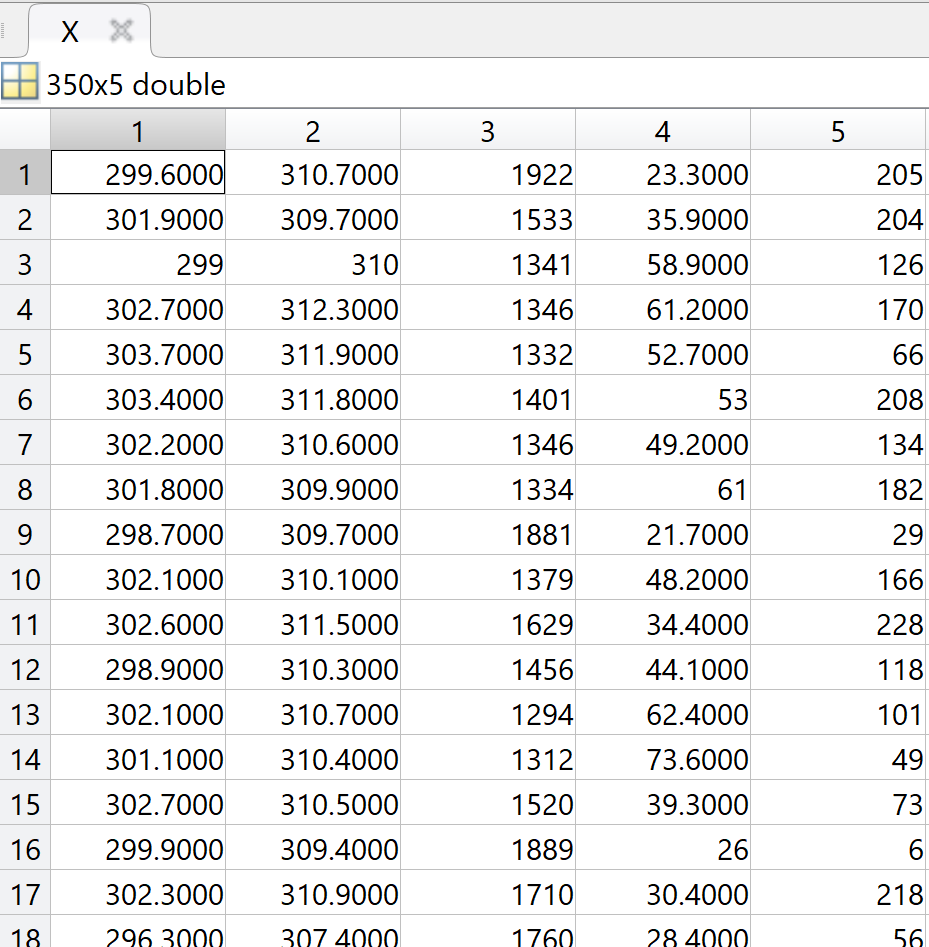
لذا محاسبه accuracy برای این مدل مقداری بسیار نزدیک به صفر یا صفر مطلق را حاصل می کند.



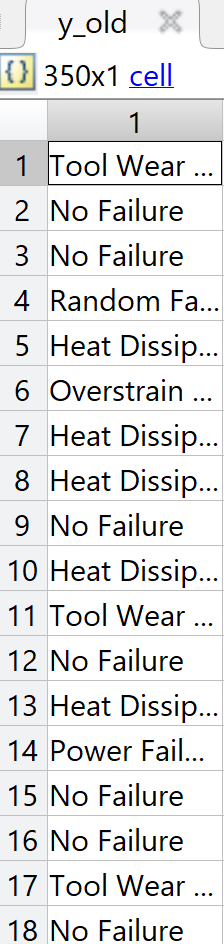
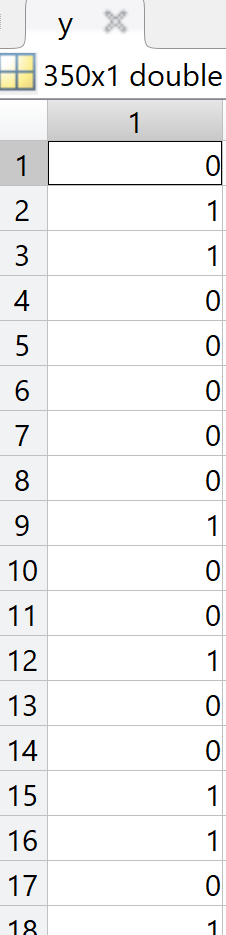




ابتدا با استفاده از دستور Read table داده ها فراخوانی شده اند. سپس ستون چهارم تا هشتم جدا شده اند و در متغیر x ریخته شده اند که 18 ستون اول آن ها در شکل زیر قابل مشاهده است و ابعاد آن 350\*5 می باشد.

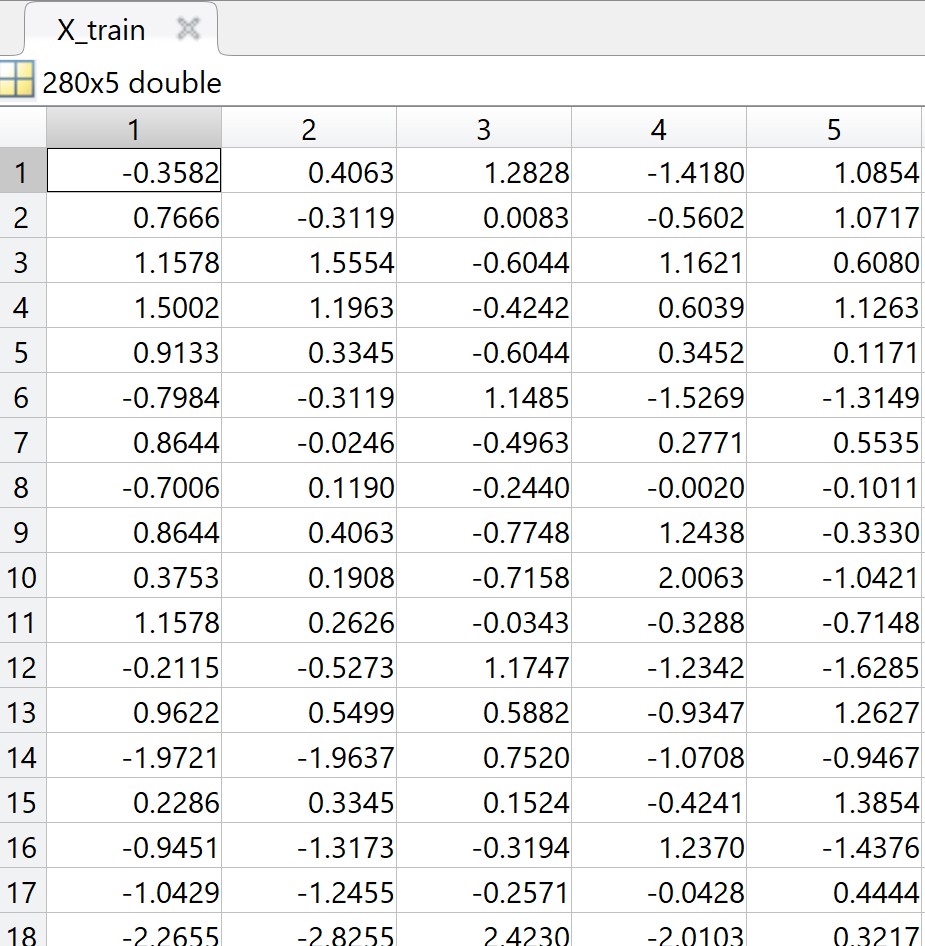


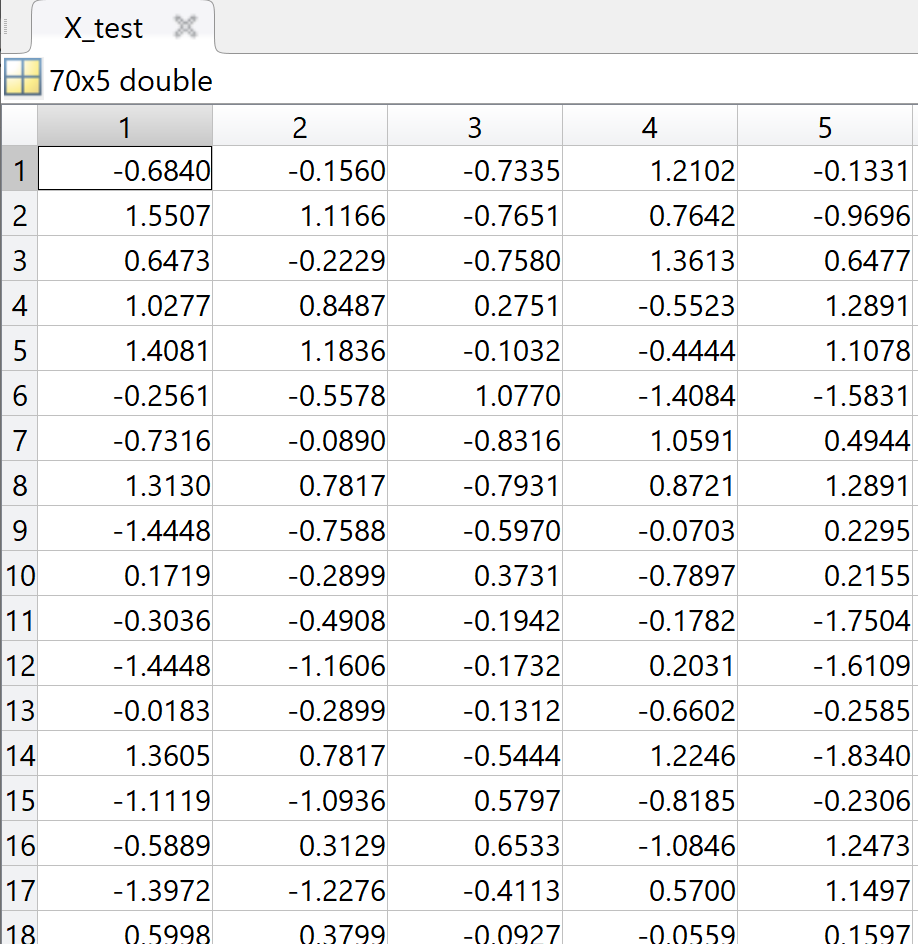
سپس ستون آخر در متغیر y\_old ریخته شده است و مطابق خواسته ی صورت سوال مقادیر آن با صفر و یک برچسب زده شده اند.( برای عبارت no failure عدد صفر و سایر عبارات خرابی عدد یک lael زده شده است).

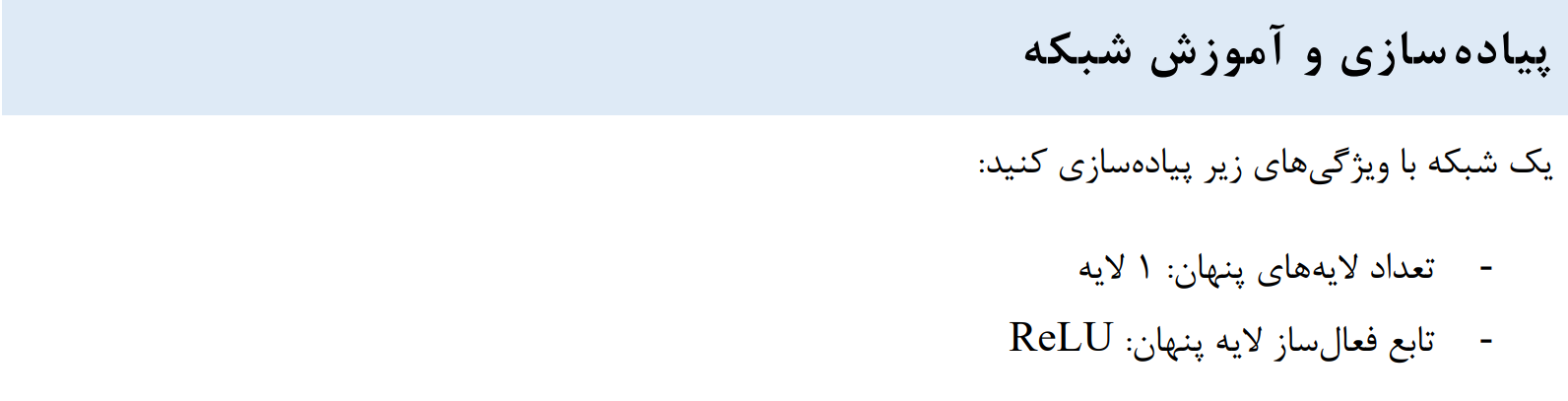
 

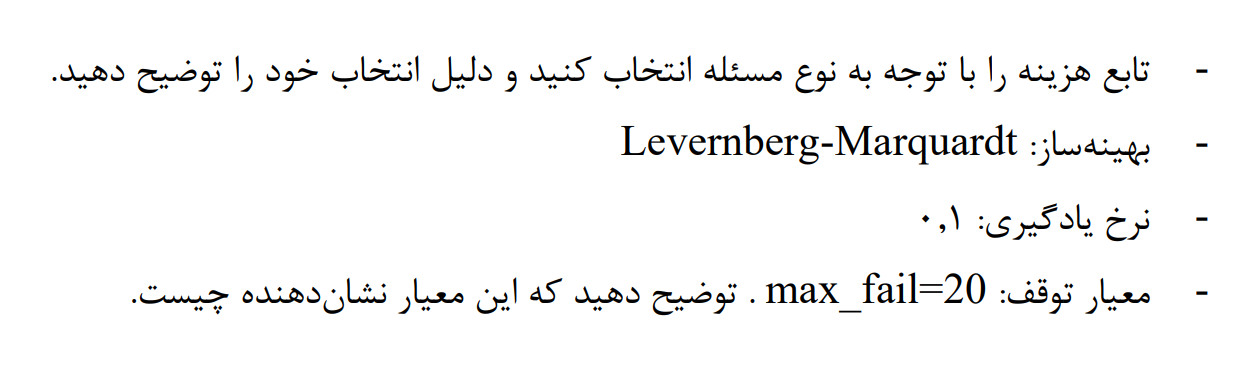
Y\_old y

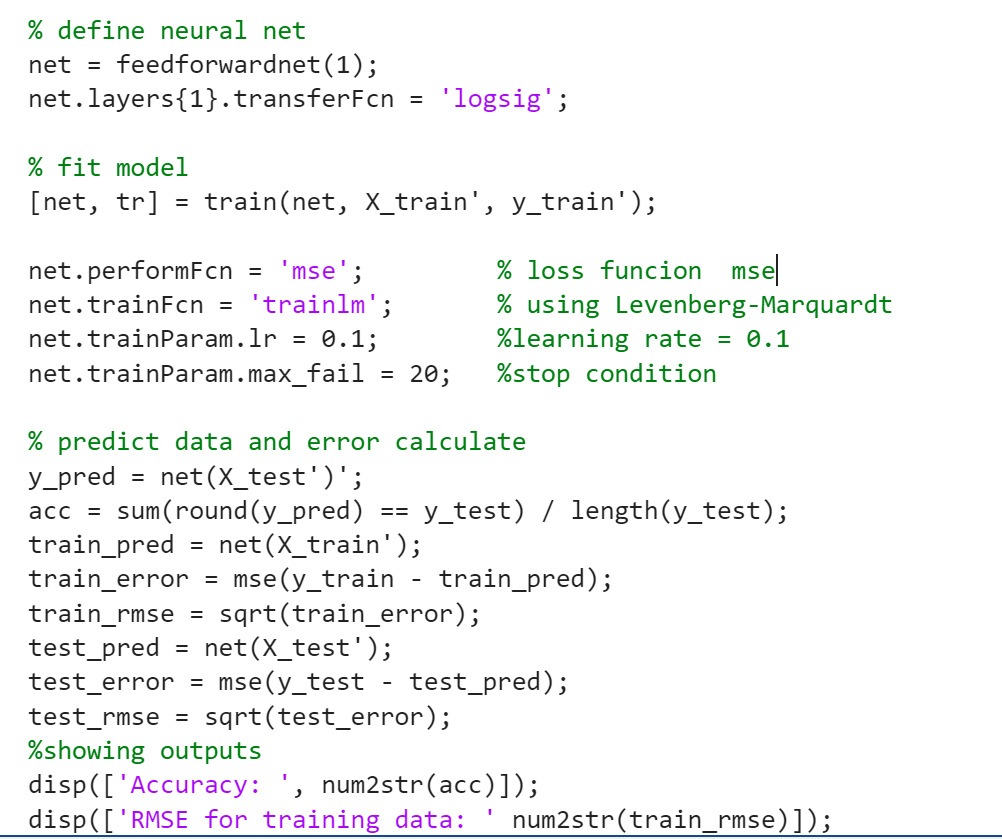
سپس در قسمت split data با استفاده از دستور cvpartition داده ها به داده های train و test تقسیم بندی شده اند و در مرحله بعد با استفاده از دستور zscor داده های ورودی استانداردسازی شده اند که ویژگی ها یا ورودی های استاندارد شده در زیر آورده شده اند:







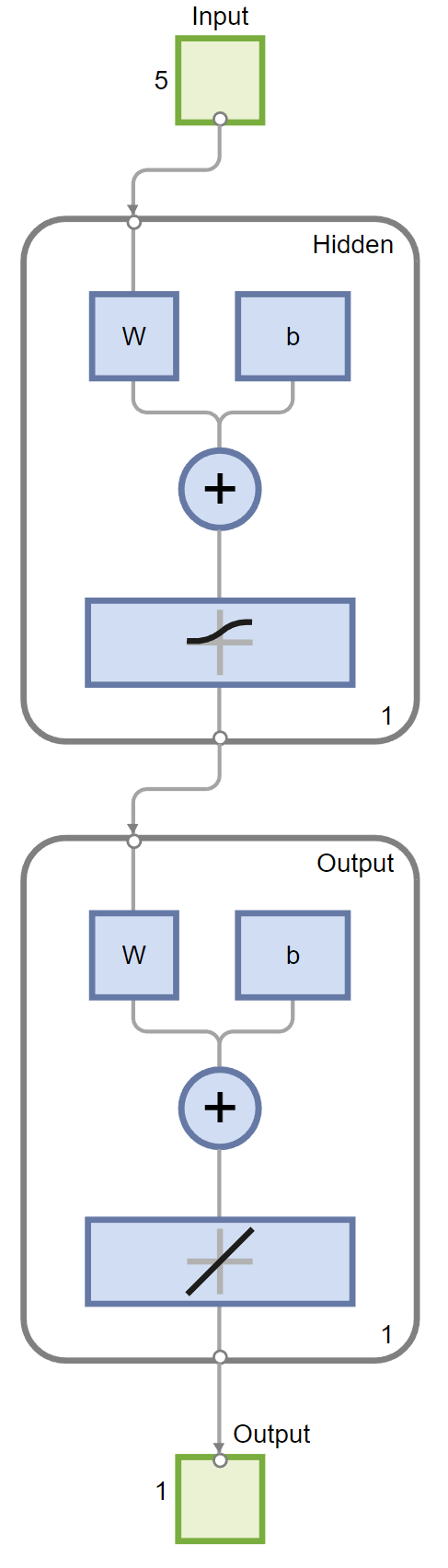
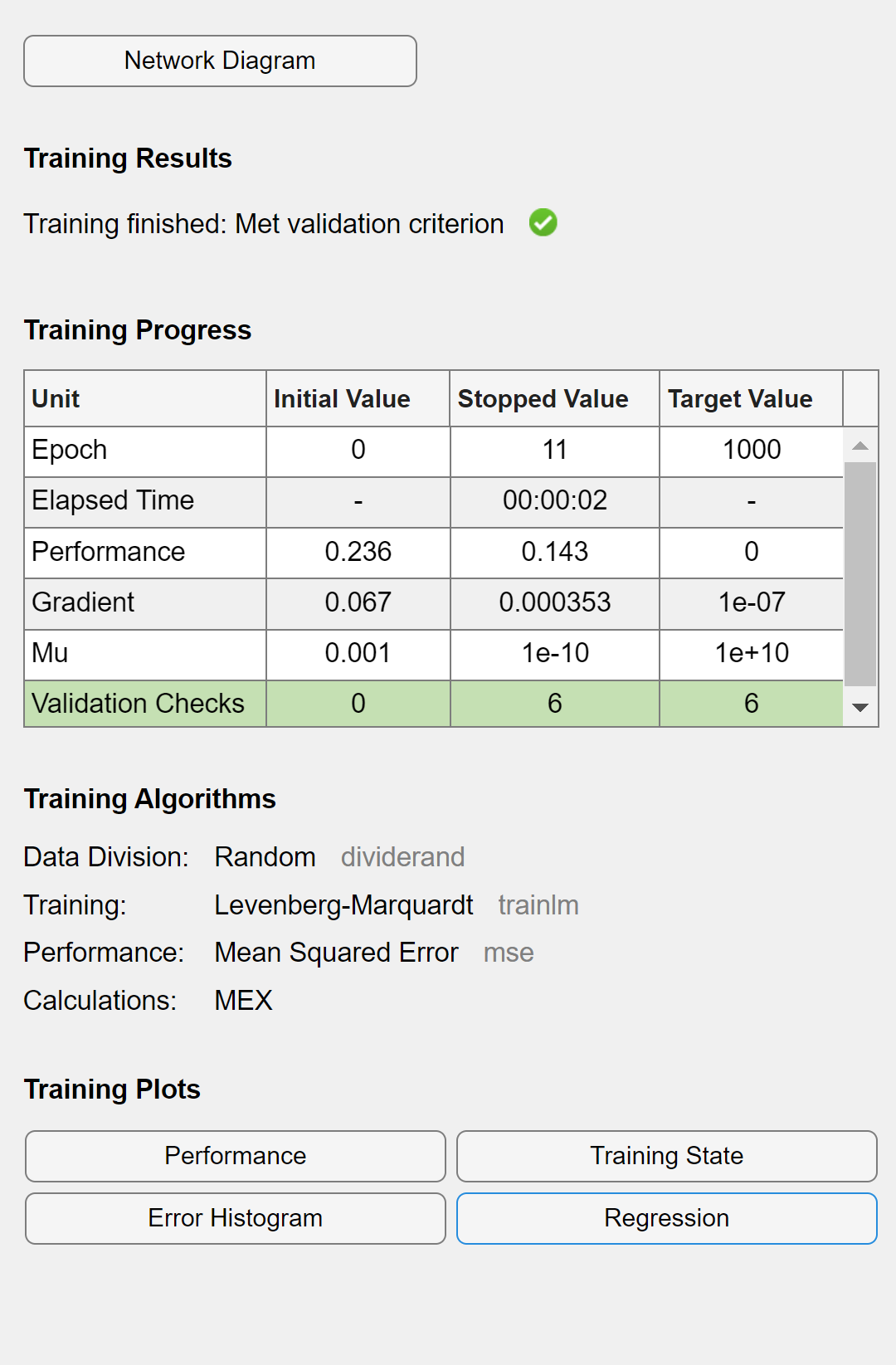




مطابق خواسته سوال یک شبکه عصبی با یک لایه پنهان و یک نورون (در قسمت های بعد تعداد نورون تغییر داده می شود) با استفاده از دستور feedforwardnet تربیت شده است.

برای تابع هزینه از mse استفاده شده و مطابق خواسته صورت سوال بهینه ساز levenerg-marquardt با نرخ یادگیری 0.1 و max-fail =20 بکار گرفته شده اند.

**باتوجه به اینکه متلب از نام Relu برای فعال ساز استفاده نمی کند با سرچ doc transfer FCN در documantion متلب برای فعال ساز تابع logsig به کار گرفته شده است**.

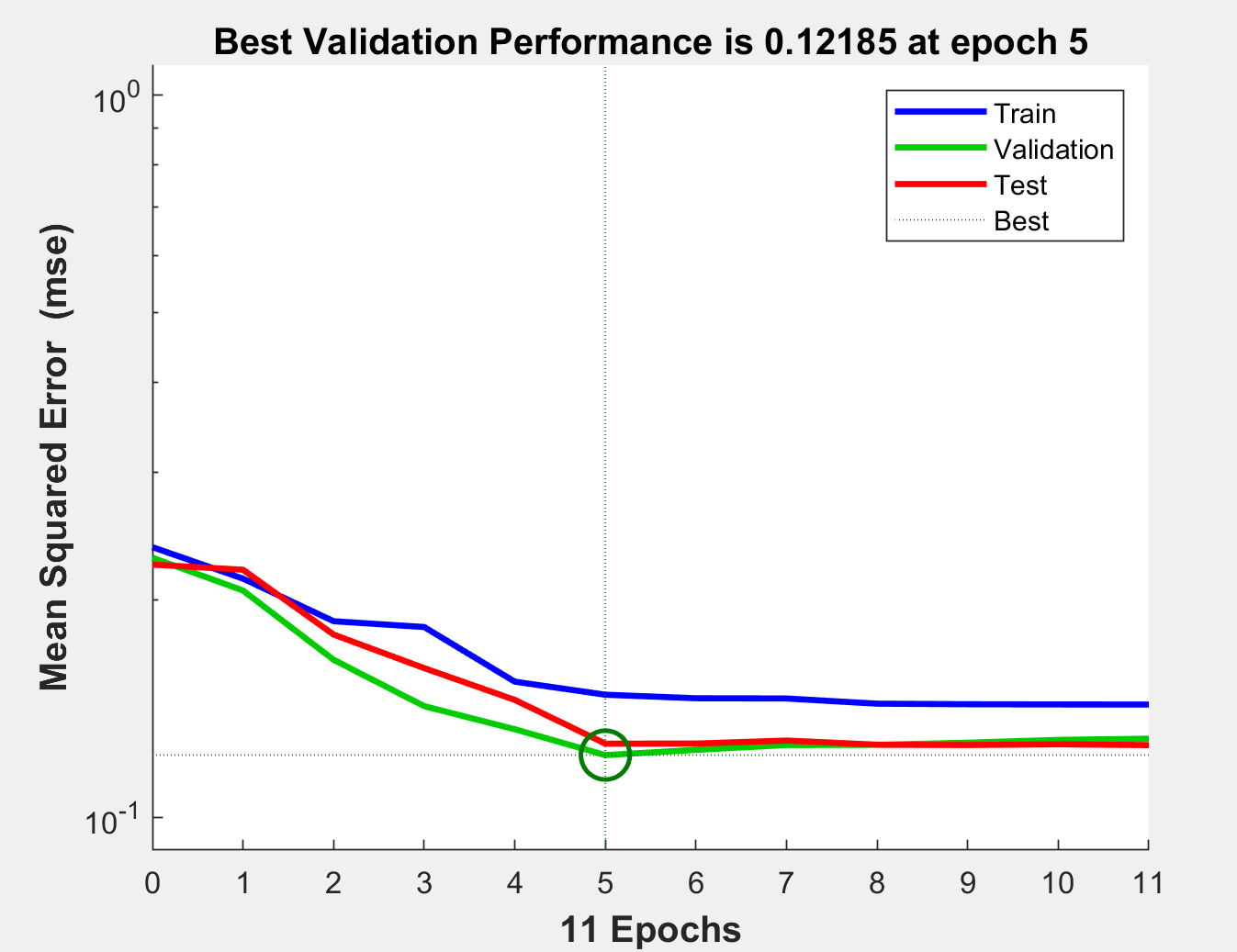
 

دیاگرام شبکه عصبی تربیت شده **نتایج تربیت مدل( پایان تربیت با رسیدن به معیار validation)**

**معیار max fail :**

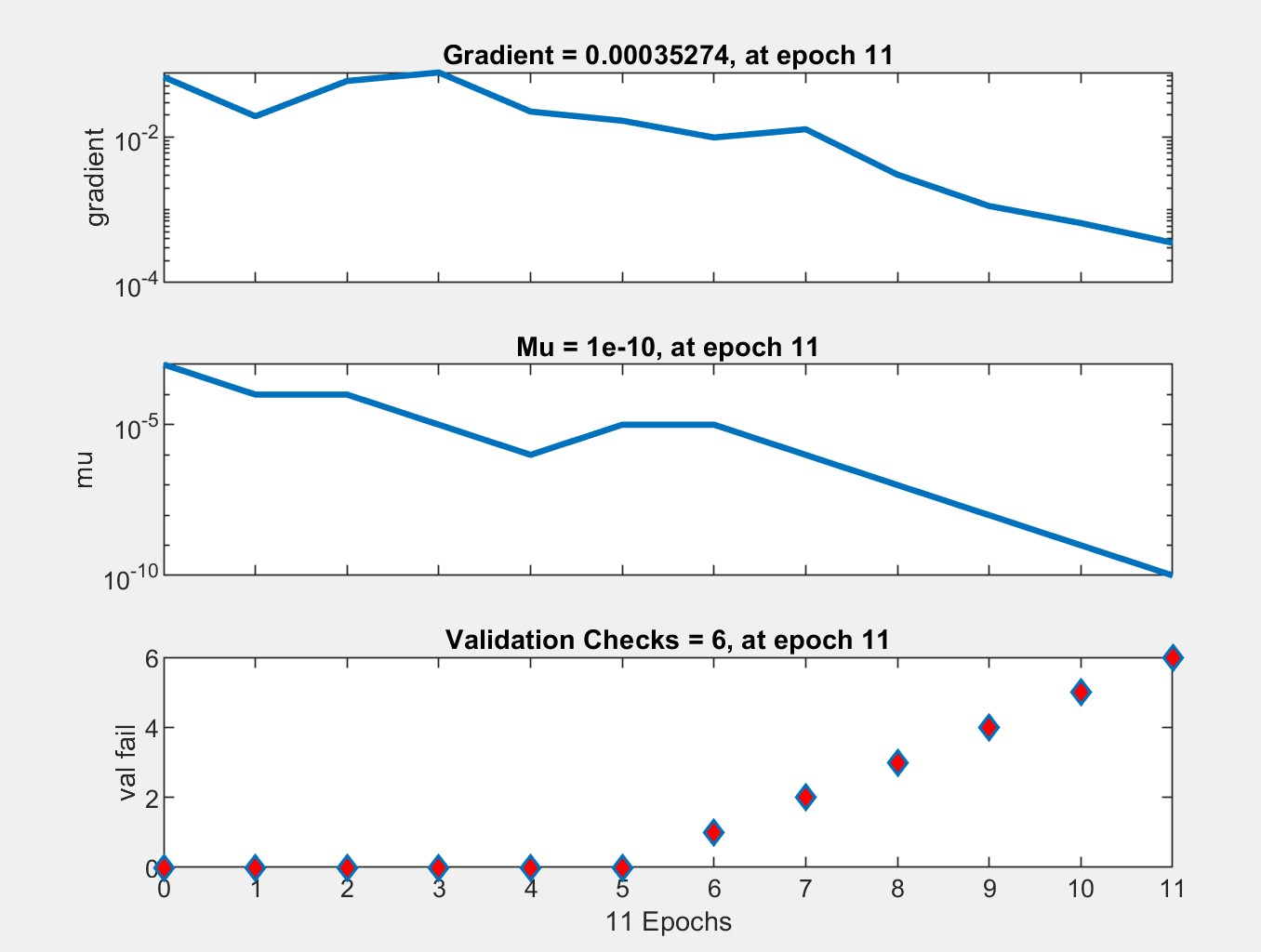
در شبکه‌های عصبی، یک پارامتر کنترلی است که برای تعیین شرایط اتمام آموزش شبکه عصبی استفاده می‌شود. این پارامتر به تعداد بارهایی اشاره دارد که شبکه عصبی می‌تواند در آنها بهبودی نداشته باشد تا آموزش متوقف شود. به طور کلی، در هر مرحله از آموزش شبکه عصبی، عملکرد شبکه با استفاده از یک معیار ارزیابی مشخص می‌شود. اگر عملکرد شبکه در مراحل متوالی بهبود نیابد و مقدار معیار ارزیابی آن به یک مقدار ثابت و مشخص نزدیک شود، آموزش شبکه به پایان می‌رسد. در این حالت، با توجه به پارامتر max fail، تعداد بارهایی که شبکه بدون بهبود به ادامه آموزش ادامه می‌دهد، تعیین می‌شود.

به عنوان مثال، فرض کنید پارامتر max fail برابر با 6 باشد. در این صورت، اگر شبکه عصبی در 6 بار آموزش متوالی بهبودی نداشته باشد، آموزش متوقف می‌شود و شبکه به عنوان خروجی نهایی تحویل داده می‌شود. با توجه به پارامتر max fail، می‌توان بهبود عملکرد آموزش شبکه را بهبود داد و زمان آموزش را بهینه کرد.

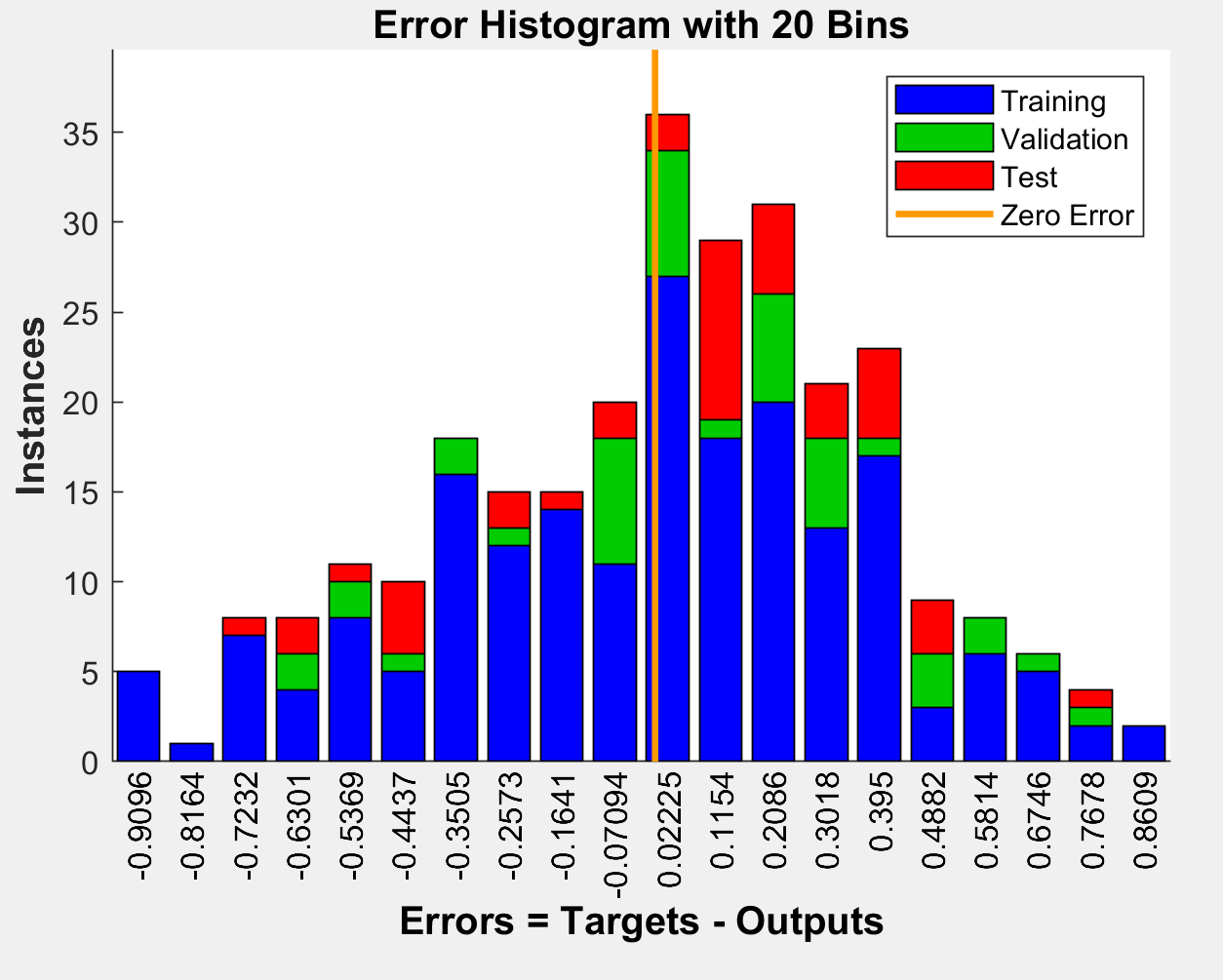


**نمودار mse برحسب تعداد ایپاک برای داده های تست و آموزش و ارزیابی.در ایپاک 5 ام معیار validation با توجه به ثابت شدن شیب نمودار و اینکه از این به بعد دیگه خطا کاهش پیدا نمی کنه آموزش رو متوقف کرده**

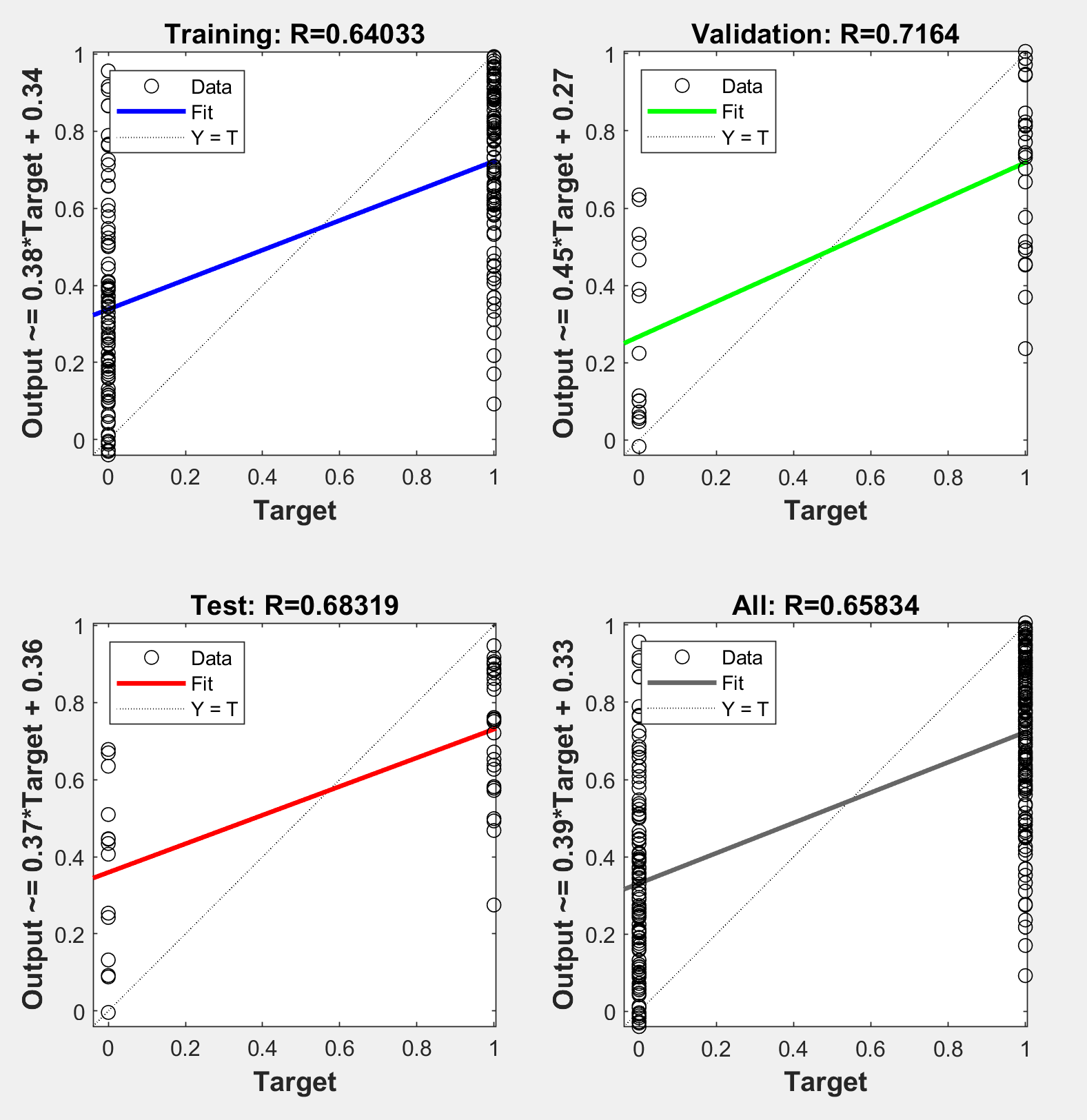
ضریب Mu : در هر ایپاک learning rate را اصلاح و به روزآوری می کند.



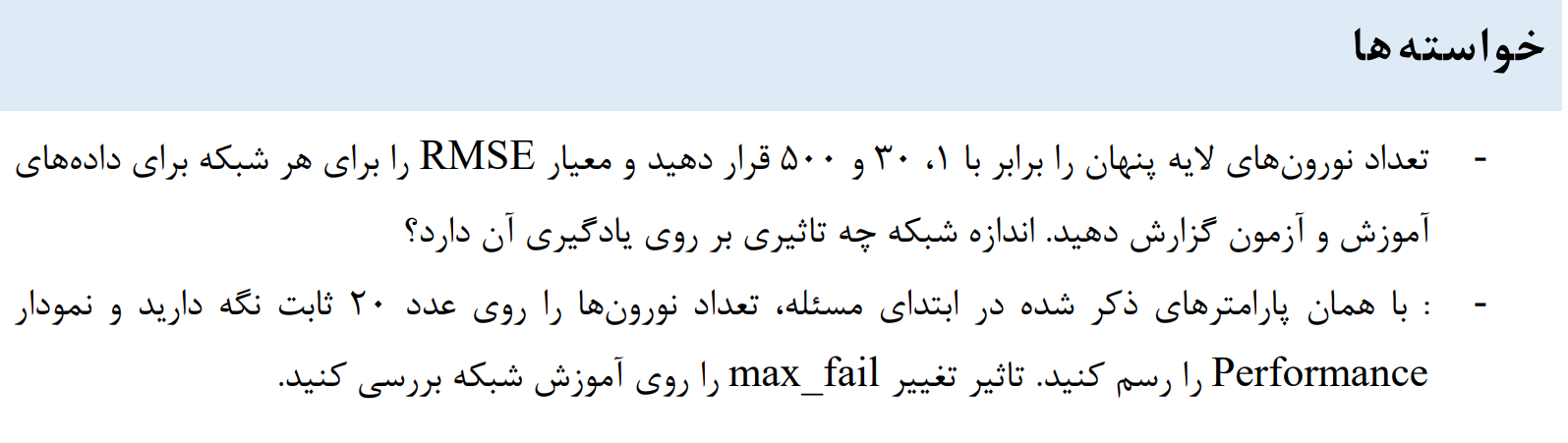
**مشاهده می شود که مقادیر mu و گرادیان از ایپاک 5 به بعد کاهش می یابند.**



**نمودار هیستوگرام خطا برای داده های train, test, validation**

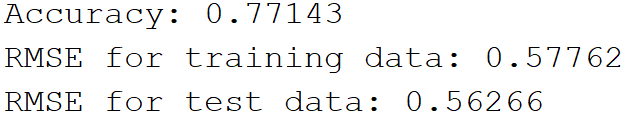


نمودار پیشبینی یا regression برای داده های تست و آموزش و ارزیابی که تابع جداسازی regression را برای هر حالت نشان داده و مقایسه ای بین مقادیر واقعی خروجی و مقدار ارزیابی شده به صورت جدا گانه دارد و در نمودار آخر این مقدار را برای جمیع مقادیر داده نشان داده است.

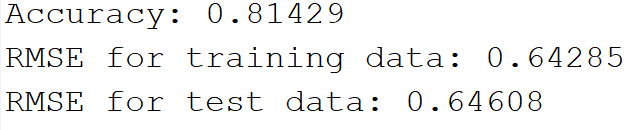


مقایسه ی RMSE برای تعداد نورون 1 , 30 , 500 :

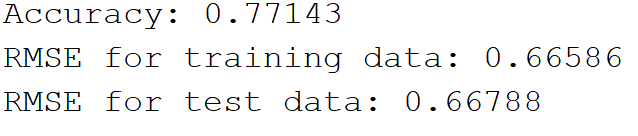
تعداد نورون = 1 :



تعداد نورون = 30 :



تعداد نورون = 500 :

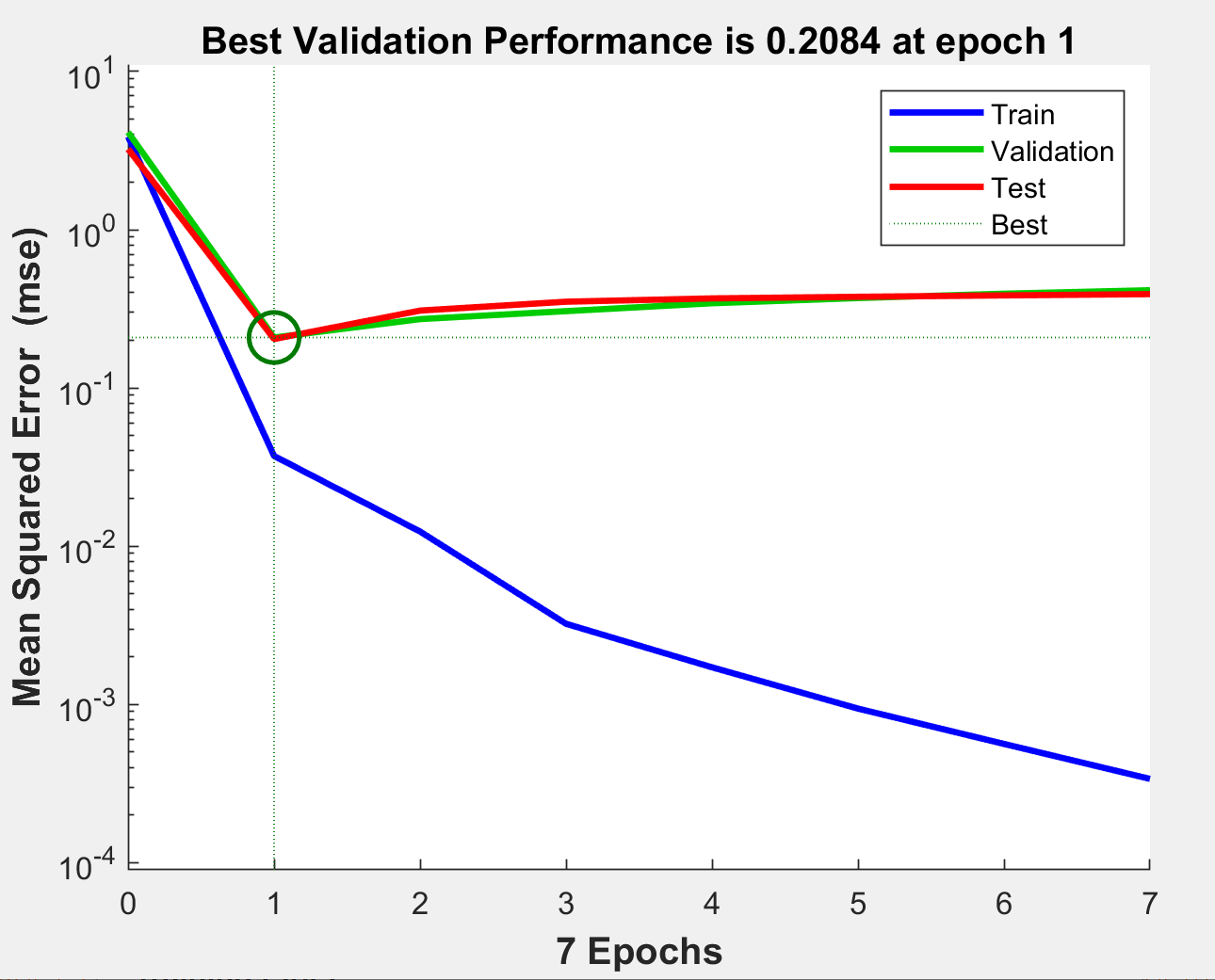


با افزایش تعداد نورون یا به عبارتی افزایش اندازه شبکه از 1 به 30 مقدار RMSE افزایش یافته است.همچنین با افزایش بیشتر اندازه شبکه از30 به 500 نورون نیز مقدار RMSE افزایش یافته است.البته میزان افزایش آن از حالت قبل کمتر است.به عبارت کلی با افزایش تعداد نورون RMSE نیز افزایش می یابد.

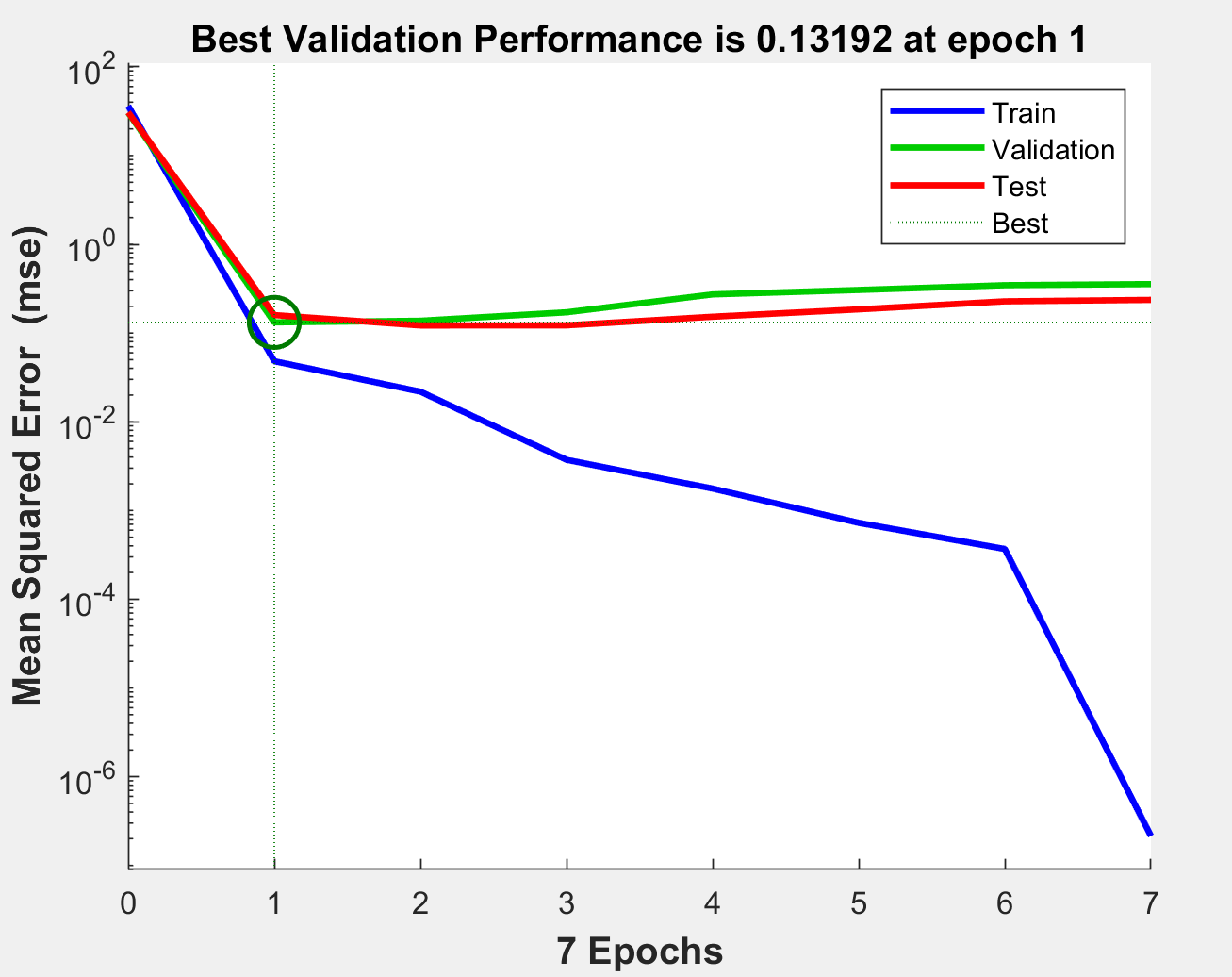
البته برای Accuracy این موضوع صادق نیست. یعنی accuracy در تعداد نورون مشخصی بیشینه است و با افزایش و کاهش نورون این مقدار کاهش می یابد.

**تاثیر تغییرات max fail روی آموزش :**

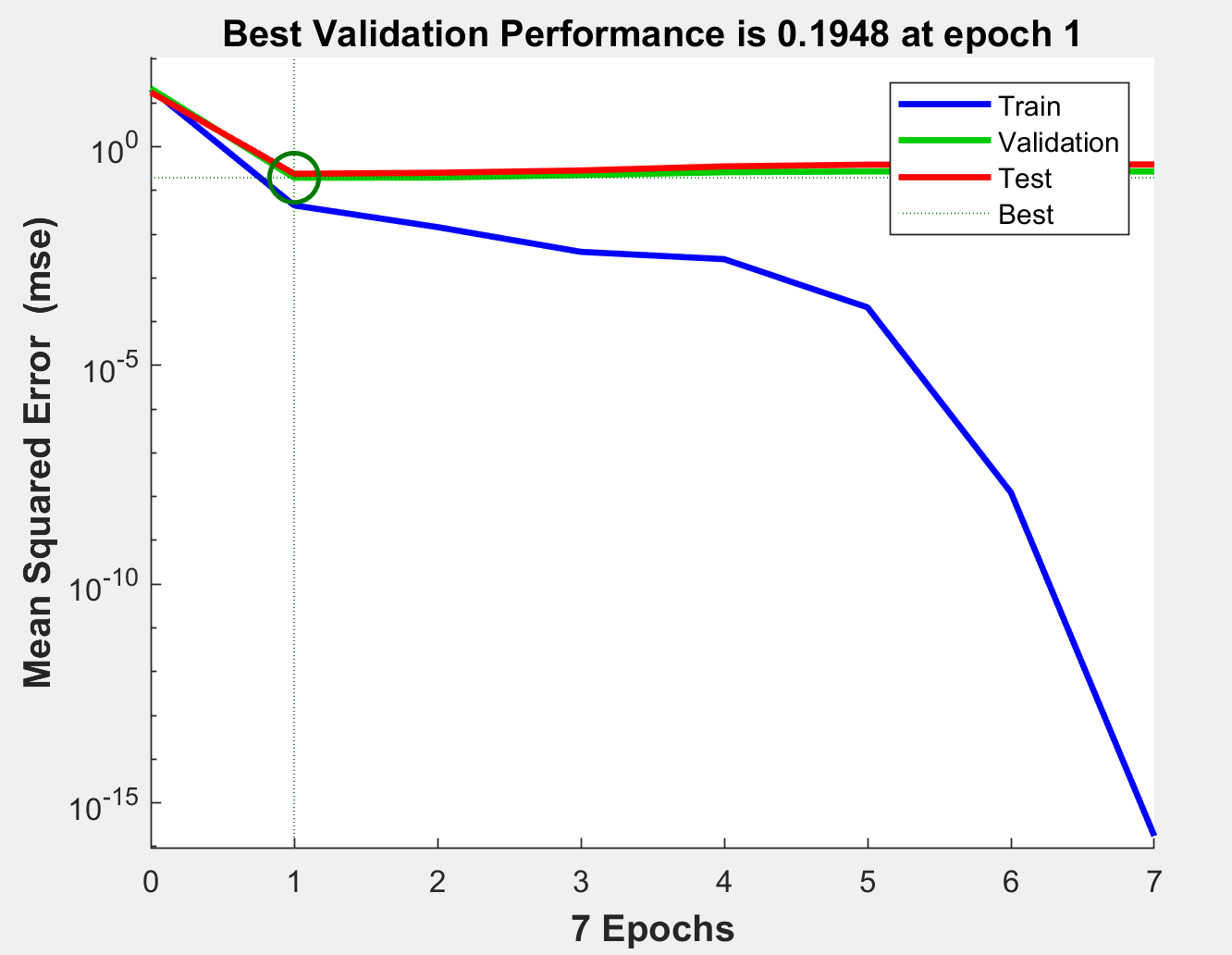
**Max fail = 1 :**



**Max fail = 10**



**Max fail = 1000**



در این مثال چون معیار توقف شبکه max fail نمی باشد و معیار توقف validation است تاثیر این پارامتر تا قبل از توقف شبکه خیلی مشهود نیست. همانطور که مشاهده می شود با افزایش max fail همگرایی داده های train به validation و test بیشتر می شود و به دنبال آن دقت نیز روی داده های test و validation بیشتر می شود.البته باید دقت کرد که با افزایش بیش از حد max fail اگر شرط توقف دیگری در مدل نباشد شبکه دچار بیش بردازش می شود.

منابع:

1. ویکی پدیا
2. <https://www.w3schools.com/>
3. Youtub.com
4. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>
5. Chat GPT