نام خدا

تمرین چهارم درس هوش مصنوعی

محمد امین سلطانی چم حیدری

۸۱۰۶۰۱۰۸۱

استاد مربوطه:

دکتر مسعود شریعت پناهی

بهار۱۴۰۲

بخش اول: مفاهیم پایه

- در مورد مشکل Imbalanced Datasets توضیح دهید. برای حل این مشکل از چه روش‌هایی استفاده می‌شود؟

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 ۲۰۰۰ تراکنش فقط ۳۰ شماره کلاهبرداری ثبت شده است. بنابراین، تعداد کلاهبرداری در هر ۱۰۰ تراکنش کمتر از ۲٪ است یا می توان گفت بیش از ۹۸٪ 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.

۱- **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 .**
۲- **استفاده از K-fold Cross-Validation به روش درست.**
۳- **تغییر داده های train یا به اصطلاح resample آن ها برای آموزش.**
٤- **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.**

- دو تابع هزینه Binary Cross Entropy و Categorical Cross Entropy را توضیح دهید. تفاوت این دو با هم چیست؟ آیا می‌توان از این توابع هزینه در یک مسئله Regression استفاده کرد؟

(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.

- آیا Accuracy به‌تنهایی معیار قابل اعتمادی از عملکرد یک مدل است؟ چرا؟

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 را تعریف کنید. اگر از این روش‌ها استفاده نکنیم چه مشکلی در روند آموزش ایجاد می‌شود؟

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.

بخش دوم: پیش‌بینی قیمت مسکن

فراخوانی داده‌ها

در ابتدا با استفاده از ابزار مناسب دادگان را بخوانید و تعداد سطر و ستون آن را گزارش کنید. برای این دادگان ماتریس همبستگی (Correlation Matrix) را تشکیل دهید و توضیح دهید که این ماتریس چه اطلاعاتی به ما می‌دهد. کدام ویژگی همبستگی بیشتری با قیمت خانه‌ها دارد؟ نمودار توزیع قیمت را رسم کنید.

```
[ ] from google.colab import drive
    drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
[ ] path = "/content/drive/MyDrive/AI/houses_1.csv"
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
df = pd.read_csv(path)
df
```

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

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	...	7	1180	0	1955	0	98178	47.5112	-122.257	1340	5650
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	...	7	2170	400	1951	1991	98125	47.7210	-122.319	1690	7639
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	...	6	770	0	1933	0	98028	47.7379	-122.233	2720	8062
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	...	7	1050	910	1965	0	98136	47.5208	-122.393	1360	5000
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	...	8	1680	0	1987	0	98074	47.6168	-122.045	1800	7503
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
21608	263000018	20140521T000000	360000.0	3	2.50	1530	1131	3.0	0	0	...	8	1530	0	2009	0	98103	47.6993	-122.346	1530	1509
21609	6600060120	20150223T000000	400000.0	4	2.50	2310	5813	2.0	0	0	...	8	2310	0	2014	0	98146	47.5107	-122.362	1830	7200
21610	1523300141	20140623T000000	402101.0	2	0.75	1020	1350	2.0	0	0	...	7	1020	0	2009	0	98144	47.5944	-122.299	1020	2007
21611	291310100	20150116T000000	400000.0	3	2.50	1600	2388	2.0	0	0	...	8	1600	0	2004	0	98027	47.5345	-122.069	1410	1287
21612	1523300157	20141015T000000	325000.0	2	0.75	1020	1076	2.0	0	0	...	7	1020	0	2008	0	98144	47.5941	-122.299	1020	1357

21613 rows × 21 columns

داده های فراخوانی شده به صورت بالا قابل مشاهده هستند.

```
[ ]  num_rows = df.shape[0]
     num_cols = df.shape[1]

     print("تعداد سطرها: ", num_rows)
     print("تعداد ستون‌ها: ", num_cols)
```

```
تعداد سطرها:  21613
تعداد ستون‌ها:  21
```

Corelation matrix:

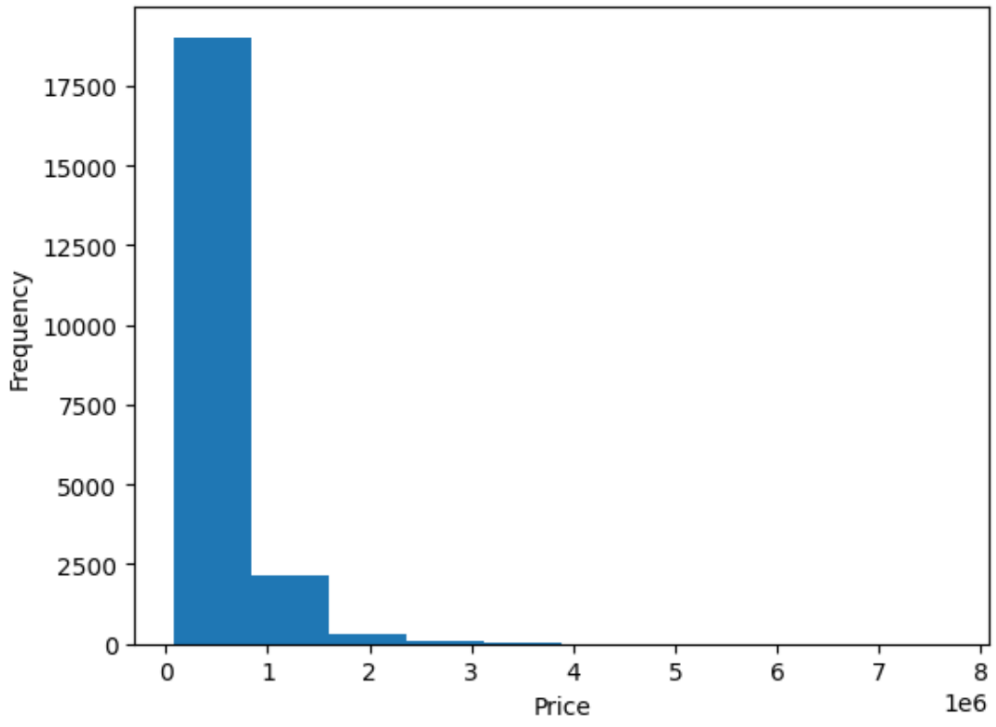
	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15
id	1.000000	-0.016762	0.001286	0.005160	-0.012258	-0.132109	0.018525	-0.002721	0.011592	-0.023783	0.008130	-0.010842	-0.005151	0.021380	-0.016907	-0.008224	-0.001891	0.020799	-0.002901	-0.138798
price	-0.016762	1.000000	0.308350	0.525138	0.702035	0.089661	0.256794	0.266369	0.397293	0.036362	0.667434	0.605567	0.323816	0.054012	0.126434	-0.053203	0.307003	0.021626	0.585379	0.082447
bedrooms	0.001286	0.308350	1.000000	0.515884	0.576671	0.031703	0.175429	-0.006582	0.079532	0.028472	0.356967	0.477600	0.303093	0.154178	0.018841	-0.152668	-0.008931	0.129473	0.391638	0.029244
bathrooms	0.005160	0.525138	0.515884	1.000000	0.754665	0.087740	0.500653	0.063744	0.187737	-0.124982	0.664983	0.685342	0.283770	0.506019	0.050739	-0.203866	0.024573	0.223042	0.568634	0.087175
sqft_living	-0.012258	0.702035	0.576671	0.754665	1.000000	0.172826	0.353949	0.103818	0.284611	-0.058753	0.762704	0.876597	0.435043	0.318049	0.055363	-0.199430	0.052529	0.240223	0.756420	0.183286
sqft_lot	-0.132109	0.089661	0.031703	0.087740	0.172826	1.000000	-0.005201	0.021604	0.074710	-0.008958	0.113621	0.183512	0.015286	0.053080	0.007644	-0.129574	-0.085683	0.229521	0.144608	0.718557
floors	0.018525	0.256794	0.175429	0.500653	0.353949	-0.005201	1.000000	0.023698	0.029444	-0.263768	0.458183	0.523885	-0.245705	0.489319	0.006338	-0.059121	0.049614	0.125419	0.279885	-0.011269
waterfront	-0.002721	0.266369	-0.006582	0.063744	0.103818	0.021604	0.023698	1.000000	0.401857	0.016653	0.082775	0.072075	0.080588	-0.026161	0.092885	0.030285	-0.014274	-0.041910	0.086463	0.030703
view	0.011592	0.397293	0.079532	0.187737	0.284611	0.074710	0.029444	0.401857	1.000000	0.045990	0.251321	0.167649	0.276947	-0.053440	0.103917	0.084827	0.006157	-0.078400	0.280439	0.072575
condition	-0.023783	0.036362	0.028472	-0.124982	-0.058753	-0.008958	-0.263768	0.016653	0.045990	1.000000	-0.144674	-0.158214	0.174105	-0.361417	-0.060618	0.003026	-0.014941	-0.106500	-0.092824	-0.003406
grade	0.008130	0.667434	0.356967	0.664983	0.762704	0.113621	0.458183	0.082775	0.251321	-0.144674	1.000000	0.755923	0.168392	0.446963	0.014414	-0.184862	0.114084	0.198372	0.713202	0.119248
sqft_above	-0.010842	0.605567	0.477600	0.685342	0.876597	0.183512	0.523885	0.072075	0.167649	-0.158214	0.755923	1.000000	-0.051943	0.423898	0.023285	-0.261190	-0.000816	0.343803	0.731870	0.194050
sqft_basement	-0.005151	0.323816	0.303093	0.283770	0.435043	0.015286	-0.245705	0.080588	0.276947	0.174105	0.168392	-0.051943	1.000000	-0.133124	0.071323	0.074845	0.110538	-0.144765	0.200355	0.017276
yr_built	0.021380	0.054012	0.154178	0.506019	0.318049	0.053080	0.489319	-0.026161	-0.053440	-0.361417	0.446963	0.423898	-0.133124	1.000000	-0.224874	-0.346869	-0.148122	0.409356	0.326229	0.070958
yr_renovated	-0.016907	0.126434	0.018841	0.050739	0.055363	0.007644	0.006338	0.092885	0.103917	-0.060618	0.014414	0.023285	0.071323	-0.224874	1.000000	0.064357	0.029398	-0.068372	-0.002673	0.007854
zipcode	-0.008224	-0.053203	-0.152668	-0.203866	-0.199430	-0.129574	-0.059121	0.030285	0.084827	0.003026	-0.184862	-0.261190	0.074845	-0.346869	0.064357	1.000000	0.267048	-0.564072	-0.279033	-0.147221
lat	-0.001891	0.307003	-0.008931	0.024573	0.052529	-0.085683	0.049614	-0.014274	0.006157	-0.014941	0.114084	-0.000816	0.110538	-0.148122	0.029398	0.267048	1.000000	-0.135512	0.048858	-0.086419
long	0.020799	0.021626	0.129473	0.223042	0.240223	0.229521	0.125419	-0.041910	-0.078400	-0.106500	0.198372	0.343803	-0.144765	0.409356	-0.068372	-0.564072	-0.135512	1.000000	0.334605	0.254451
sqft_living15	-0.002901	0.585379	0.391638	0.568634	0.756420	0.144608	0.279885	0.086463	0.280439	-0.092824	0.713202	0.731870	0.200355	0.326229	-0.002673	-0.279033	0.048858	0.334605	1.000000	0.183192
sqft_lot15	-0.138798	0.082447	0.029244	0.087175	0.183286	0.718557	-0.011269	0.030703	0.072575	-0.003406	0.119248	0.194050	0.017276	0.070958	0.007854	-0.147221	-0.086419	0.254451	0.183192	1.000000

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.

```
import pandas as pd
import matplotlib.pyplot as plt

price_columns= df['price']
plt.hist(price_columns)

plt.xlabel("Price")
plt.ylabel("Frequency")

plt.show()
```



نودار توزیع قیمت

پیش پردازش داده‌ها

در این مرحله داده‌ها را برای آموزش شبکه آماده می‌کنیم.

- ستون Date را به دو ستون ماه و سال تبدیل کنید و آن را از دادگان حذف کنید.
- ستون قیمت را به‌عنوان ستون خروجی (برچسب یا Y) درنظر بگیرید و بیشترین و کمترین قیمت را گزارش کنید. بقیه ستون‌ها را به‌عنوان داده‌های ورودی یا X درنظر بگیرید.
- داده‌ها را به‌نسبت ۸۰-۲۰ به داده‌های train و test تقسیم کنید.
- با کمک MinMaxScaler داده‌ها را scale کنید. توجه داشته باشید که در فرآیند scale کردن نباید از داده‌های آزمون استفاده کنید، چون باعث نشت اطلاعات (Data Leakage) می‌شود (این پدیده را توضیح دهید.)

```
[ ]  from datetime import datetime
     df['date']= pd.to_datetime(df['date'], format='%Y%m%dT')
     df['month'] = df['date'].dt.month
     df['year'] = df['date'].dt.year
     df=df.drop('date',axis=1)
     df
```

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.

...	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15	month	year
...	0	1955	0	98178	47.5112	-122.257	1340	5650	10	2014
...	400	1951	1991	98125	47.7210	-122.319	1690	7639	12	2014
...	0	1933	0	98028	47.7379	-122.233	2720	8062	2	2015
...	910	1965	0	98136	47.5208	-122.393	1360	5000	12	2014
...	0	1987	0	98074	47.6168	-122.045	1800	7503	2	2015
...	...	...	...	...	...	...	...	...	...	...
...	0	2009	0	98103	47.6993	-122.346	1530	1509	5	2014
...	0	2014	0	98146	47.5107	-122.362	1830	7200	2	2015
...	0	2009	0	98144	47.5944	-122.299	1020	2007	6	2014
...	0	2004	0	98027	47.5345	-122.069	1410	1287	1	2015
...	0	2008	0	98144	47.5941	-122.299	1020	1357	10	2014

```
[10] max_value = df['price'].max()
     min_value = df['price'].min()

     print('Max price:', max_value)
     print('Min price:', min_value)
```

```
Max price: 7700000.0
Min price: 75000.0
```

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

```
feature_df = df.drop('price', axis=1)
X = np.asarray(feature_df)
y = df['price'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

Scaler= preprocessing.StandardScaler().fit(X_train)
X_train =Scaler.transform(X_train.astype(float))
```

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.

پیاده‌سازی مدل

معماری شبکه

برای حل این مسئله از یک مدل MLP با ۲ لایه پنهان یا بیشتر استفاده می‌کنیم. تعداد لایه‌ها و تعداد نورون‌های هر لایه را به دلخواه خود انتخاب کنید اما توجه داشته باشد که مدل نباید بیش از حد سنگین یا سبک باشد.

آموزش شبکه

شبکه را با ویژگی‌های زیر آموزش دهید:

- تابع هزینه: تابع هزینهٔ مناسب را با توجه به نوع مسئله انتخاب کنید.
- تابع فعال‌ساز لایه‌های پنهان: ReLU
- بهینه‌ساز: SGD
- سایز batch: ۶۴
- نرخ یادگیری: یکبار برابر ۰,۰۰۱ و بار دیگر برابر ۰,۱
- تعداد ایپاک[1]: یکبار برابر ۲۰ و بار دیگر برابر ۴۰۰۰

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 yearو month در کل ۲۱ ویژگی وجود دارد تعداد نورون لایه پنهان اول برابر با ۴۲ و تعداد نورون لایه پنهان دوم برابر ۲۱ در نظر گرفته شده است.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from keras.optimizers import SGD
from sklearn import preprocessing
import numpy as np
from sklearn.metrics import mean_squared_error


feature_df = df.drop('price', axis=1)
X = np.asarray(feature_df)
y = df['price'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

Scaler= preprocessing.StandardScaler().fit(X_train)
X_train =Scaler.transform(X_train.astype(float))

#
#
#
model = Sequential()
model.add(Dense(42, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(21, activation='relu'))
model.add(Dense(1, activation='linear'))

sgd = SGD(learning_rate=0.1)
model.compile(loss='mae', optimizer=sgd)

history = model.fit(X_train, y_train, epochs=20, batch_size=64, validation_split= 0.1)

y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print('mse score:', mse)

plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

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 ۴۲ و ۲۱ می باشند و لایه سوم که لایه خروجی است دارای یک نورون می باشد.

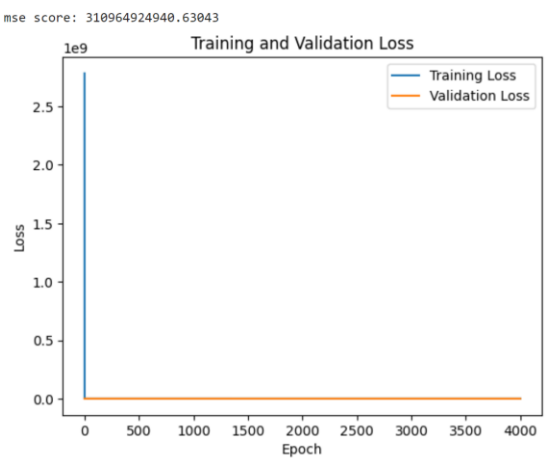
دستور sequential از کتابخانه Transflow و با دستور keras فراخوانی شده است.

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.

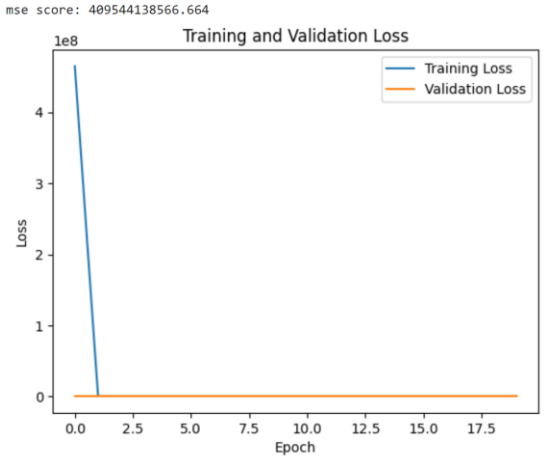
سپس بهینه ساز از نوعSGD که بهینه ساز با روش کاهش گرادیانی می باشد تعریف می شود و نرخ یادگیری برابر با ۰.۱ در نظر گرفته می شود.

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.

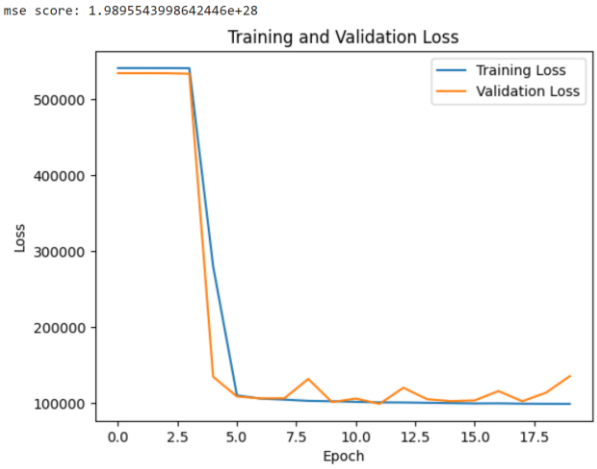
- برای چهار مدل train شده نمودارهای loss و validation loss را بر حسب شماره ایپاک رسم کنید و دقت آموزش و سنجش هر مدل را گزارش کنید.



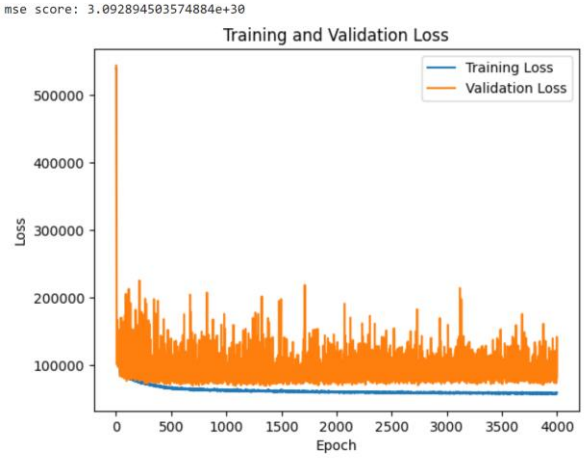
Learning rate = 0.1 with 4000 epoch



Learning rate = 0.1 with 20 epoch



Learning rate = 0.001 with 4000 epoch



Learning rate = 0.001 with 20 epoch

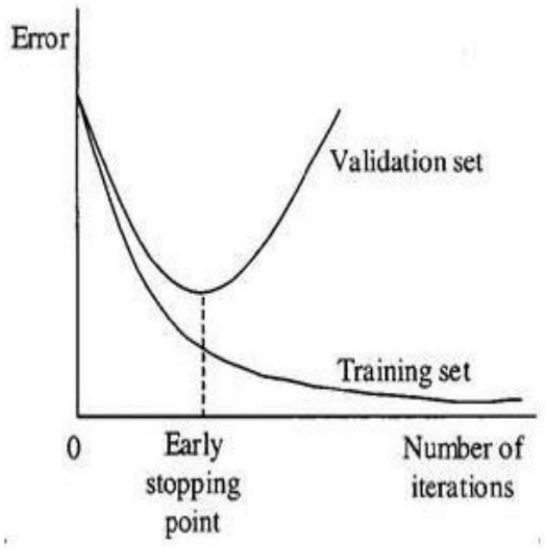
مقایسه مجموع خطای ۴ مدل:

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

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.

- آیا تعداد ۲۰ ایپاک برای آموزش کافی بود؟ ۴۰۰۰ چطور؟ با استفاده از نمودار، مقدار حدودی تعداد ایپاک کافی برای هر حالت را گزارش کنید.

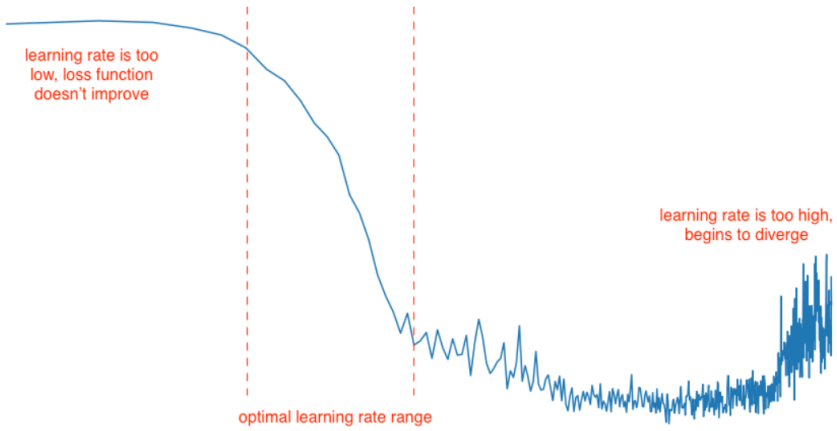


اگر معیار توقف شبکه تعداد ایپاک باشد باید توجه داشت که هم شبکه دچار بیش برداز‌ش نشود و هم به میزان کافی تربیت شود که پیش بینی های آن صحیح باشد. لذا چون در این جا برای پایان تربیت از شرط توقف early stopping استفاده نشده است باید از روی نمودارهای بدست آمده محل واگرا شدن نمودار loss و validation loss را 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.

Learning rate	enough Epoch
0.1	More than 4000
0.001	About 200

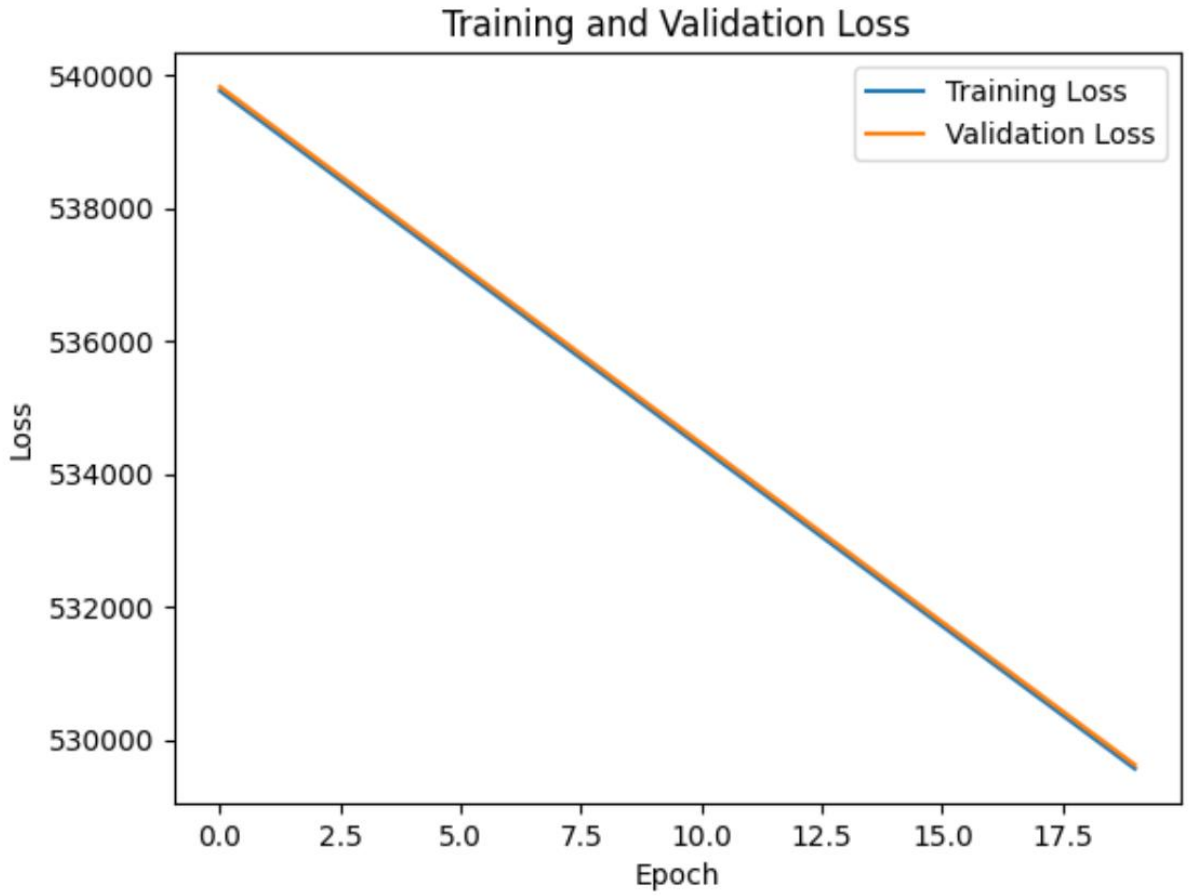
- بررسی کنید که تغییر اندازه نرخ یادگیری چه تاثیری در فرآیند آموزش داشته است.

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).



- مدلی را که بیشترین دقت را داشته تعیین کنید و این‌بار این مدل را با تابع فعال‌ساز tanh برای لایه‌های پنهان آموزش دهید و دقت آموزش و سنجش آن را گزارش کنید.

```
mse score: 428291404880.3157
```

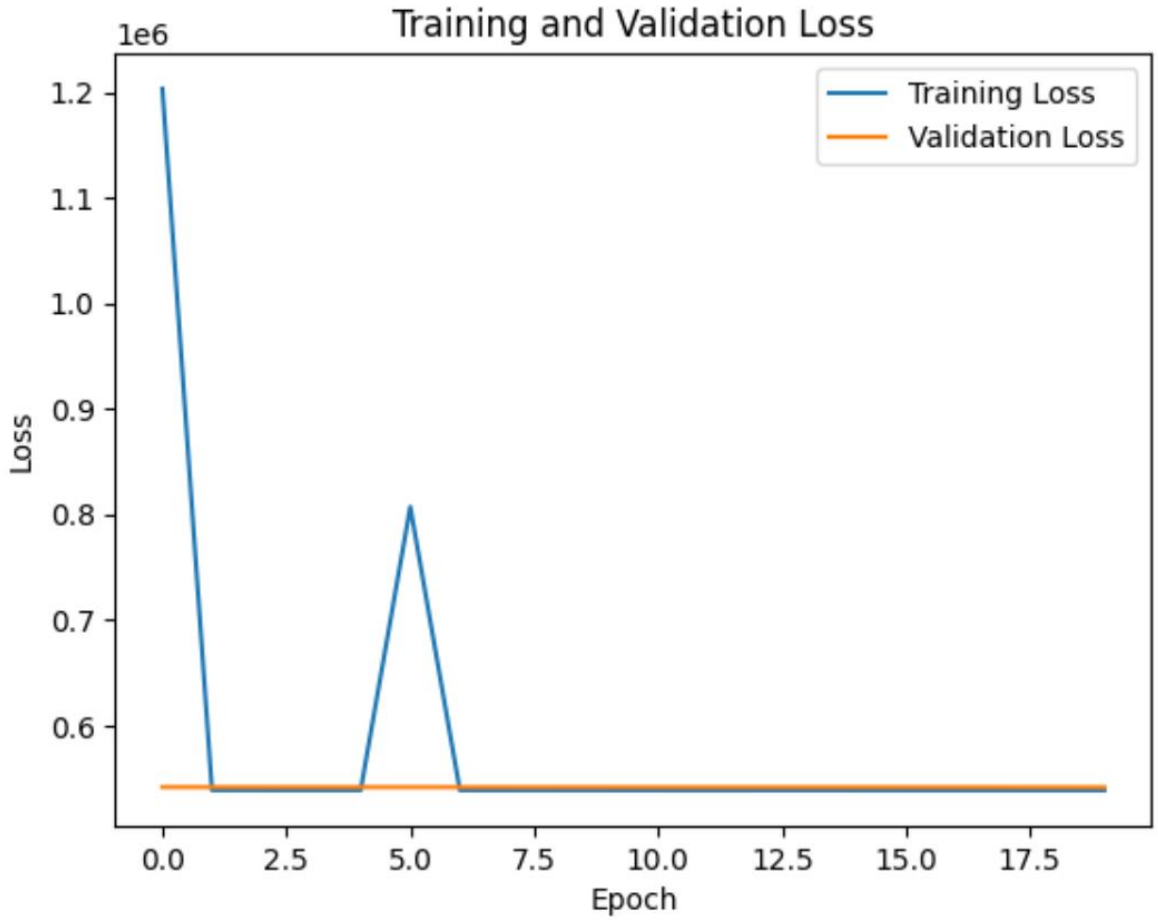


با توجه به اینکه تابع فعال ساز tanh یک فعال ساز صفر و یک است و مقادیر ورودی را به صفر و یک تصویر می کند این مدل با تابع فعال ساز tanh خطای بیشتری را حاصل می کند یا به عبارتی دقت شبکه عصبی با تابع فعال ساز tanh کاهش می یابد.

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

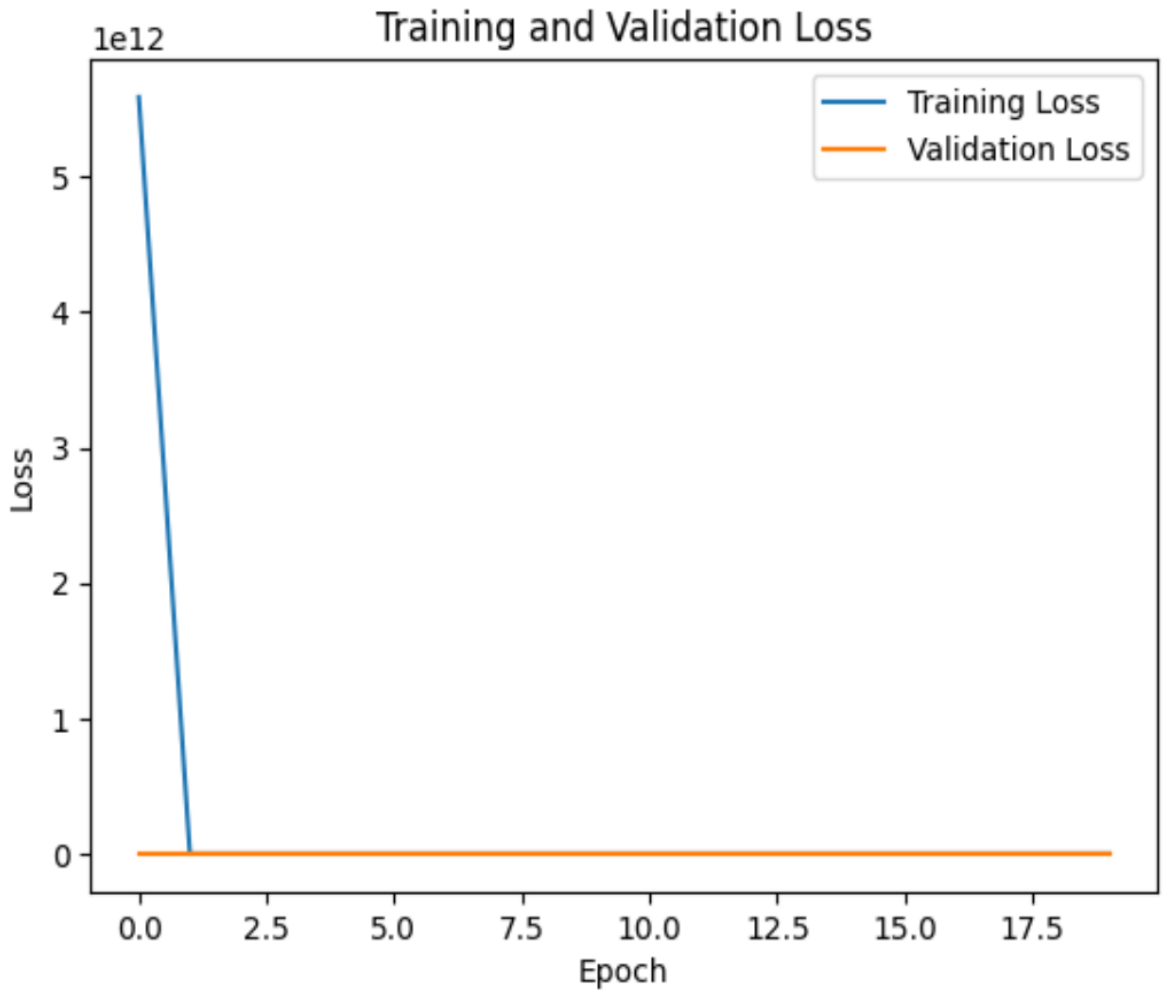
- همین مدل را یکبار بار با batchsize برابر با ۱ و بار دیگر برابر با ۲۵۶ آموزش دهید. نمودارهای هزینه و دقت داده‌های آموزش و اعتبارسنجی را رسم کنید.

```
mse score: 433480634032.671
```



Batch size = 256

```
mse score: 393362559017.57043
```



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.

- با استفاده از شبکه‌ای که بیشترین دقت را در بین شبکه‌های آموزش یافته داشته، خروجی داده‌های آزمون را پیش‌بینی کنید و ماتریس سردرگمی[۲] را تشکیل دهید. برای این کار از کتاب‌خانه sklearn استفاده کنید.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from keras.optimizers import SGD
from sklearn import preprocessing
import numpy as np
from sklearn.metrics import mean_squared_error


feature_df = df.drop('price', axis=1)
X = np.asarray(feature_df)
y = df['price'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

Scaler= preprocessing.StandardScaler().fit(X_train)
X_train =Scaler.transform(X_train.astype(float))




model = Sequential()
model.add(Dense(42, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(21, activation='relu'))
model.add(Dense(1, activation='linear'))

sgd = SGD(learning_rate=0.001)
model.compile(loss='mae', optimizer=sgd)

history = model.fit(X_train, y_train, epochs=20, batch_size=64, validation_split= 0.1)

y_pred = model.predict(X_test)
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)
accuracy = np.trace(cm) / np.sum(cm)

print('Confusion Matrix:\n', cm)
print("accuracy", accuracy)
```

```
Confusion Matrix:
 [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
```

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.

بخش سوم: دسته‌بندی دستگاه‌ها

پیش پردازش داده‌ها

- ستون‌های چهارم تا هشتم را به عنوان داده یا X در نظر بگیرید.
- ستون آخر را به عنوان خروجی (برچسب یا Y) در نظر بگیرید. تمامی نمونه‌هایی را که با برچسب عدم خرابی مشخص شده‌اند با عدد صفر و باقی برچسب‌ها را، بدون توجه به نوع خرابی آنها با عدد ۱ جایگزین کنید تا برچسب‌ها حالت دو کلاسه داشته باشند.
- داده‌ها را به دو قسمت آموزش و آزمون تقسیم کنید. (به نسبت ۰,۸ و ۰,۲)
- داده‌ها را استانداردسازی کنید.

```
clc
clear all

% input call
data = readtable('dataset_2.csv');

% define feature and labels
X = table2array(data(:, 4:8));
y_old = table2array(data(:, end));

% chang in labels
labels = {'No Failure'};
y = ismember(y_old, labels);
y = double(y);

% split data
cv = cvpartition(size(X, 1), 'HoldOut', 0.2);
X_train = X(cv.training, :);
y_train = y(cv.training, :);
X_test = X(cv.test, :);
y_test = y(cv.test, :);

% Standardization
X_train = zscore(X_train);
X_test = zscore(X_test);
```

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 ۱۸ ستون اول آن ها در شکل زیر قابل مشاهده است و ابعاد آن 5*350 می باشد.

X

350x5 double

	1	2	3	4	5
1	299.6000	310.7000	1922	23.3000	205
2	301.9000	309.7000	1533	35.9000	204
3	299	310	1341	58.9000	126
4	302.7000	312.3000	1346	61.2000	170
5	303.7000	311.9000	1332	52.7000	66
6	303.4000	311.8000	1401	53	208
7	302.2000	310.6000	1346	49.2000	134
8	301.8000	309.9000	1334	61	182
9	298.7000	309.7000	1881	21.7000	29
10	302.1000	310.1000	1379	48.2000	166
11	302.6000	311.5000	1629	34.4000	228
12	298.9000	310.3000	1456	44.1000	118
13	302.1000	310.7000	1294	62.4000	101
14	301.1000	310.4000	1312	73.6000	49
15	302.7000	310.5000	1520	39.3000	73
16	299.9000	309.4000	1889	26	6
17	302.3000	310.9000	1710	30.4000	218
18	296.3000	307.4000	1760	28.4000	56

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

350x1 double

	1
1	0
2	1
3	1
4	0
5	0
6	0
7	0
8	0
9	1
10	0
11	0
12	1
13	0
14	0
15	1
16	1
17	0
18	1

y_old

350x1 cell

	1
1	Tool Wear ...
2	No Failure
3	No Failure
4	Random Fa...
5	Heat Dissip...
6	Overstrain ...
7	Heat Dissip...
8	Heat Dissip...
9	No Failure
10	Heat Dissip...
11	Tool Wear ...
12	No Failure
13	Heat Dissip...
14	Power Fail...
15	No Failure
16	No Failure
17	Tool Wear ...
18	No Failure

y

Y_old

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:

X_train

280x5 double

	1	2	3	4	5
1	-0.3582	0.4063	1.2828	-1.4180	1.0854
2	0.7666	-0.3119	0.0083	-0.5602	1.0717
3	1.1578	1.5554	-0.6044	1.1621	0.6080
4	1.5002	1.1963	-0.4242	0.6039	1.1263
5	0.9133	0.3345	-0.6044	0.3452	0.1171
6	-0.7984	-0.3119	1.1485	-1.5269	-1.3149
7	0.8644	-0.0246	-0.4963	0.2771	0.5535
8	-0.7006	0.1190	-0.2440	-0.0020	-0.1011
9	0.8644	0.4063	-0.7748	1.2438	-0.3330
10	0.3753	0.1908	-0.7158	2.0063	-1.0421
11	1.1578	0.2626	-0.0343	-0.3288	-0.7148
12	-0.2115	-0.5273	1.1747	-1.2342	-1.6285
13	0.9622	0.5499	0.5882	-0.9347	1.2627
14	-1.9721	-1.9637	0.7520	-1.0708	-0.9467
15	0.2286	0.3345	0.1524	-0.4241	1.3854
16	-0.9451	-1.3173	-0.3194	1.2370	-1.4376
17	-1.0429	-1.2455	-0.2571	-0.0428	0.4444
18	-2.2655	-2.8255	2.4230	-2.0103	0.3217

X_test

70x5 double

	1	2	3	4	5
1	-0.6840	-0.1560	-0.7335	1.2102	-0.1331
2	1.5507	1.1166	-0.7651	0.7642	-0.9696
3	0.6473	-0.2229	-0.7580	1.3613	0.6477
4	1.0277	0.8487	0.2751	-0.5523	1.2891
5	1.4081	1.1836	-0.1032	-0.4444	1.1078
6	-0.2561	-0.5578	1.0770	-1.4084	-1.5831
7	-0.7316	-0.0890	-0.8316	1.0591	0.4944
8	1.3130	0.7817	-0.7931	0.8721	1.2891
9	-1.4448	-0.7588	-0.5970	-0.0703	0.2295
10	0.1719	-0.2899	0.3731	-0.7897	0.2155
11	-0.3036	-0.4908	-0.1942	-0.1782	-1.7504
12	-1.4448	-1.1606	-0.1732	0.2031	-1.6109
13	-0.0183	-0.2899	-0.1312	-0.6602	-0.2585
14	1.3605	0.7817	-0.5444	1.2246	-1.8340
15	-1.1119	-1.0936	0.5797	-0.8185	-0.2306
16	-0.5889	0.3129	0.6533	-1.0846	1.2473
17	-1.3972	-1.2276	-0.4113	0.5700	1.1497
18	0.5998	0.3799	-0.0927	-0.0559	0.1597

پیاده‌سازی و آموزش شبکه

یک شبکه با ویژگی‌های زیر پیاده‌سازی کنید:

- تعداد لایه‌های پنهان: ۱ لایه
- تابع فعال‌ساز لایه پنهان: ReLU

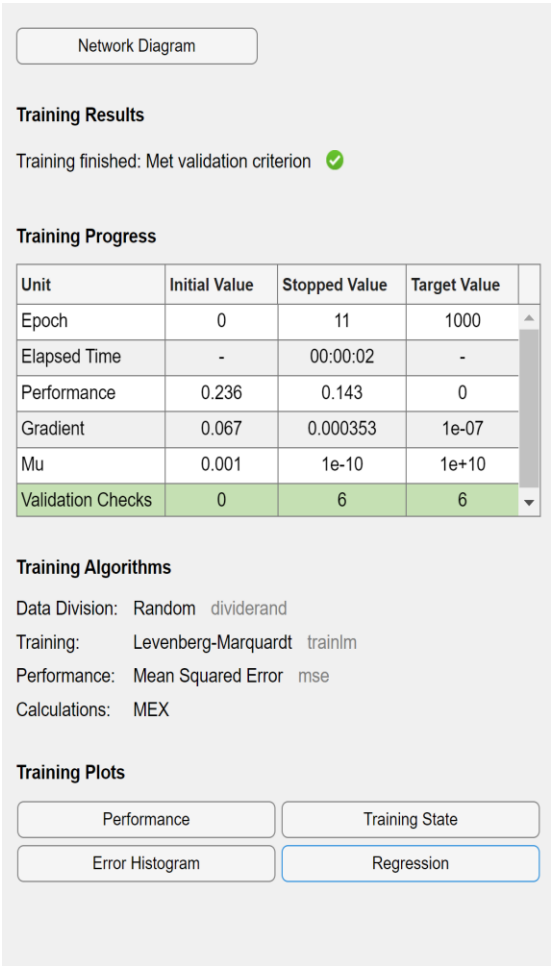
- تابع هزینه را با توجه به نوع مسئله انتخاب کنید و دلیل انتخاب خود را توضیح دهید.
- بهینه‌ساز: Levernberg-Marquardt
- نرخ یادگیری: ۰,۱
- معیار توقف: max_fail=20 . توضیح دهید که این معیار نشان‌دهنده چیست.

```
% define neural net
net = feedforwardnet(1);
net.layers{1}.transferFcn = 'logsig';

% fit model
[net, tr] = train(net, X_train', y_train');

net.performFcn = 'mse';         % loss funcion  mse
net.trainFcn = 'trainlm';       % using Levenberg-Marquardt
net.trainParam.lr = 0.1;        %learning rate = 0.1
net.trainParam.max_fail = 20;   %stop condition

% predict data and error calculate
y_pred = net(X_test')';
acc = sum(round(y_pred) == y_test) / length(y_test);
train_pred = net(X_train');
train_error = mse(y_train - train_pred);
train_rmse = sqrt(train_error);
test_pred = net(X_test');
test_error = mse(y_test - test_pred);
test_rmse = sqrt(test_error);
%showing outputs
disp(['Accuracy: ', num2str(acc)]);
disp(['RMSE for training data: ' num2str(train_rmse)]);
```

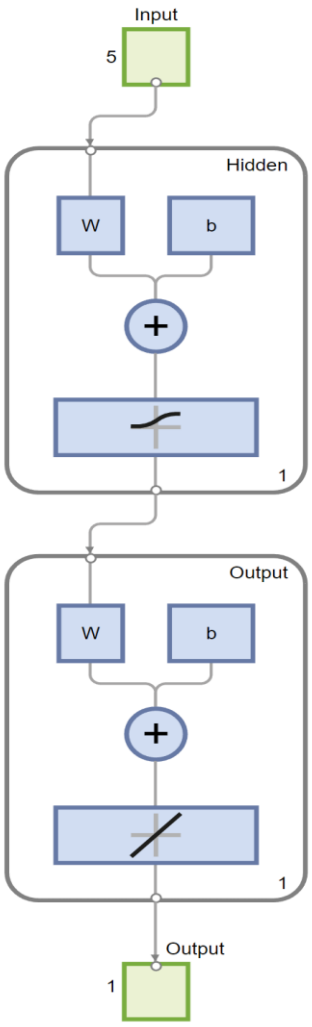
مطابق خواسته سوال یک شبکه عصبی با یک لایه پنهان و یک نورون (در قسمت های بعد تعداد نورون تغییر داده می شود) با استفاده از دستور feedforwardnet تربیت شده است.

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.



نتایج تربیت مدل(پایان تربیت با رسیدن به معیار validation)

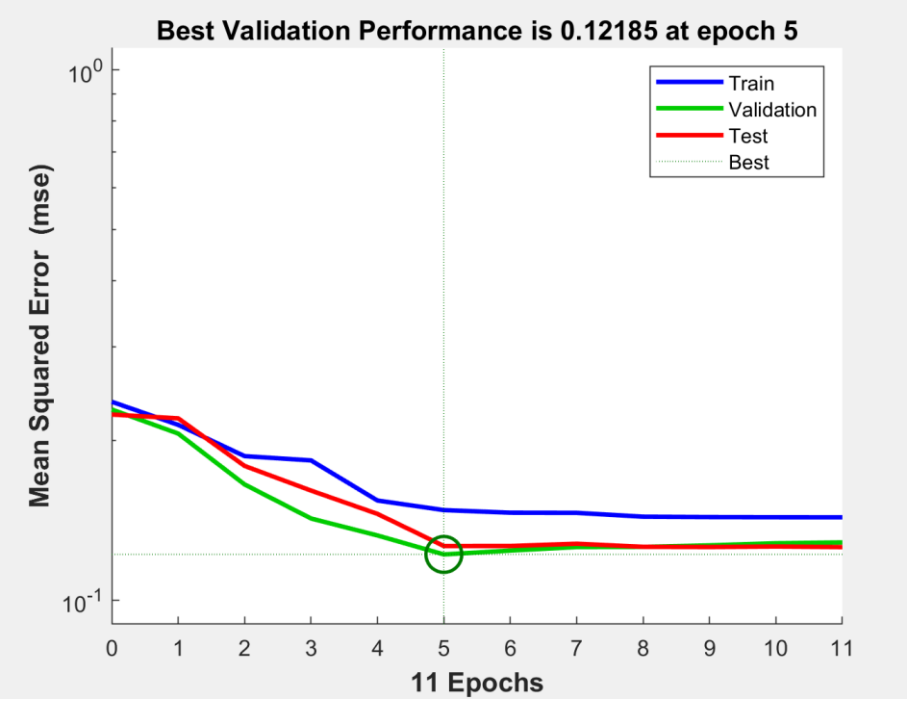


دیاگرام شبکه عصبی تربیت شده

معیار max fail :

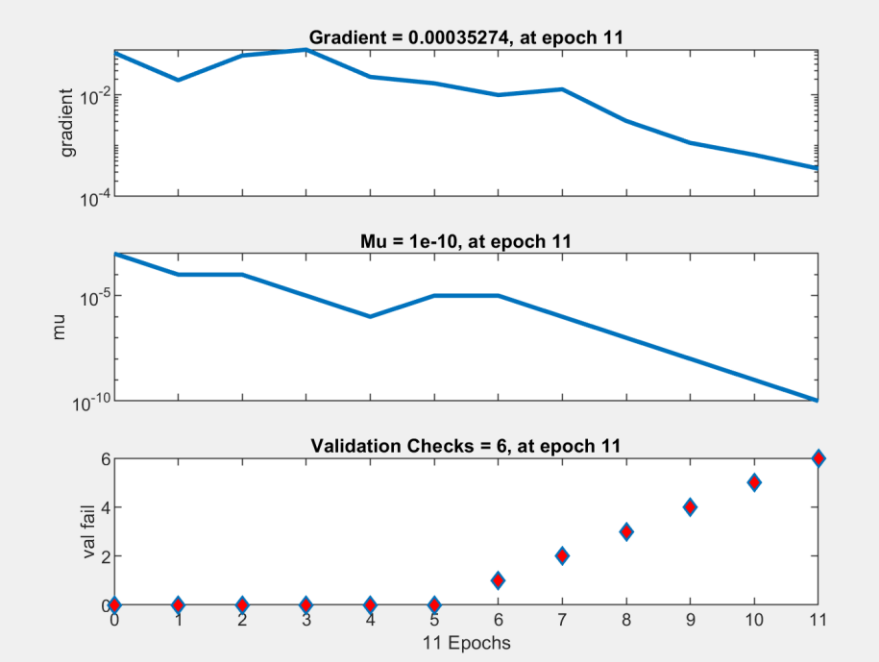
در شبکه‌های عصبی، یک پارامتر کنترلی است که برای تعیین شرایط اتمام آموزش شبکه عصبی استفاده می‌شود. این پارامتر به تعداد بارهایی اشاره دارد که شبکه عصبی می‌تواند در آنها بهبودی نداشته باشد تا آموزش متوقف شود. به طور کلی، در هر مرحله از آموزش شبکه عصبی، عملکرد شبکه با استفاده از یک معیار ارزیابی مشخص می‌شود. 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 fail max، تعداد بارهایی که شبکه بدون بهبود به ادامه آموزش ادامه می‌دهد، تعیین می‌شود.

به عنوان مثال، فرض کنید پارامتر fail max برابر با ۶ باشد. در این صورت، اگر شبکه عصبی در ۶ بار آموزش متوالی بهبودی نداشته باشد، آموزش متوقف می‌شود و شبکه به عنوان خروجی نهایی تحویل داده می‌شود. با توجه به پارامتر fail max، می‌توان بهبود عملکرد آموزش شبکه را بهبود داد و زمان آموزش را بهینه کرد.

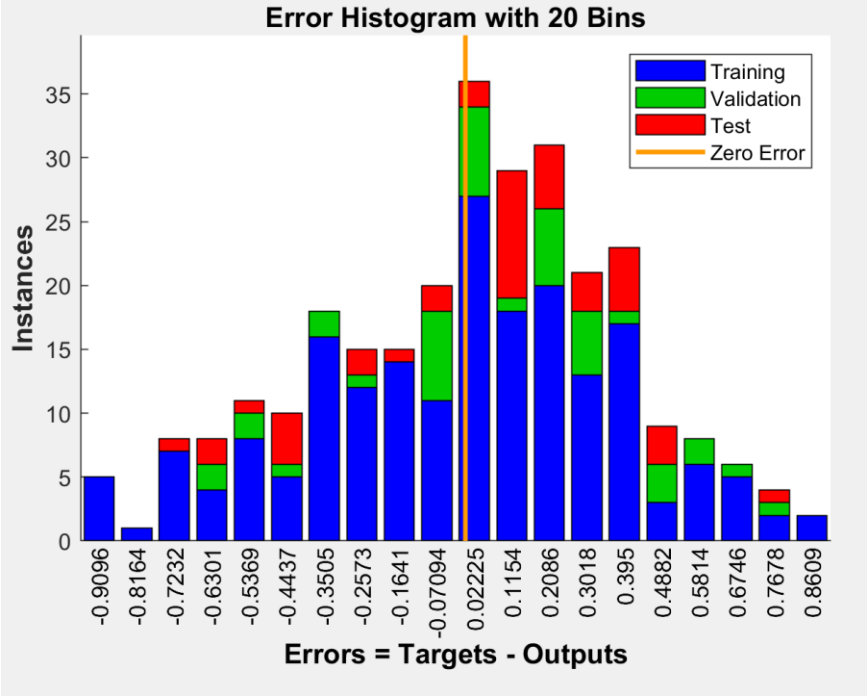


نمودار mse برحسب تعداد ایپاک برای داده های تست و آموزش و ارزیابی.در ایپاک ۵ ام معیار validation با توجه به ثابت شدن شیب نمودار و اینکه از این به بعد دیگه خطا کاهش پیدا نمی کنه آموزش رو متوقف کرده

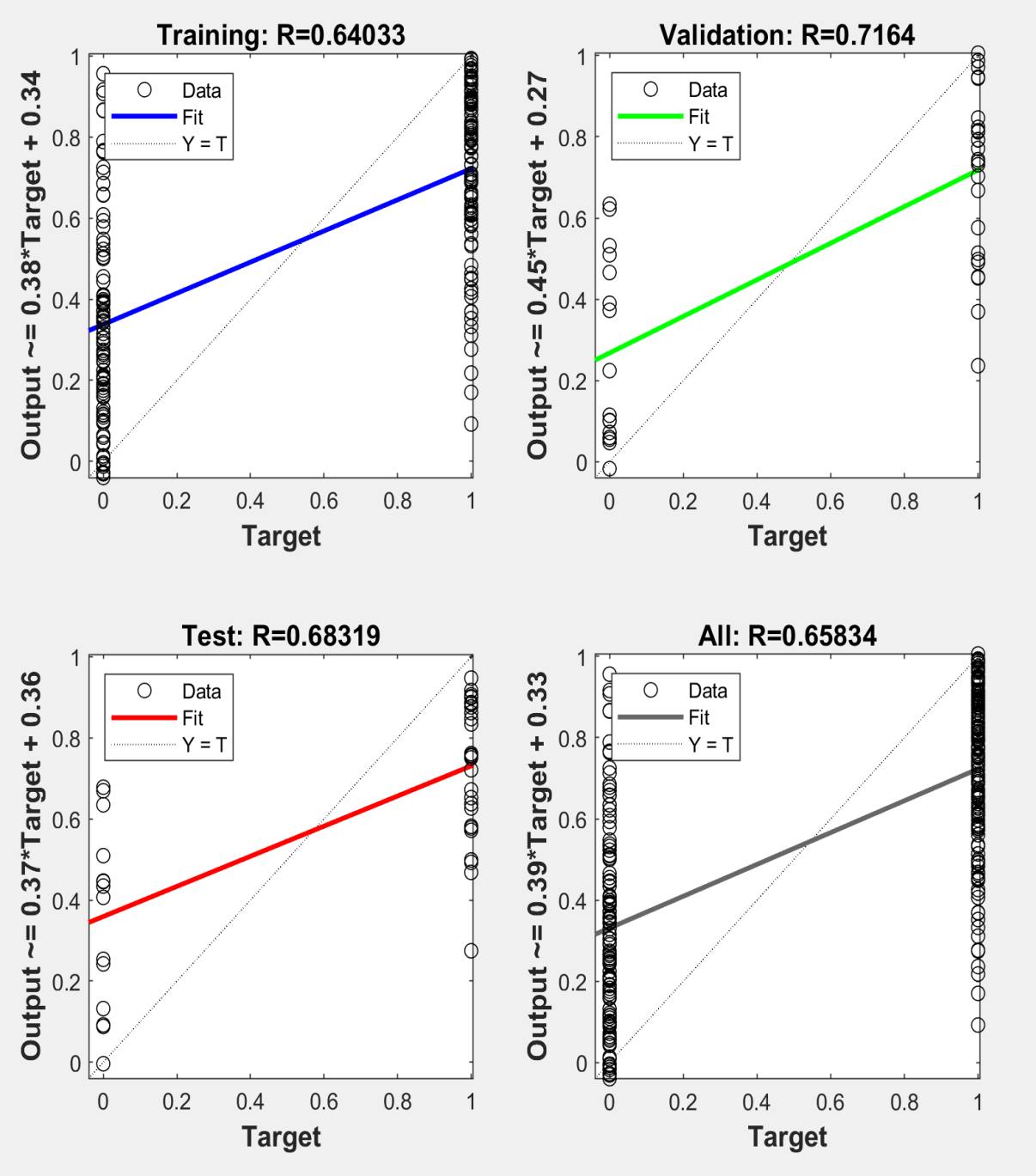
ضریب Mu : در هر ایپاک learning rate را اصلاح و به روزآوری می کند.



مشاهده می شود که مقادیر mu و گرادیان از ایپاک ۵ 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.

خواسته ها

- تعداد نورون‌های لایه پنهان را برابر با ۱، ۳۰ و ۵۰۰ قرار دهید و معیار RMSE را برای هر شبکه برای داده‌های آموزش و آزمون گزارش دهید. اندازه شبکه چه تاثیری بر روی یادگیری آن دارد؟
- : با همان پارامترهای ذکر شده در ابتدای مسئله، تعداد نورون‌ها را روی عدد ۲۰ ثابت نگه دارید و نمودار Performance را رسم کنید. تاثیر تغییر max_fail را روی آموزش شبکه بررسی کنید.

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

تعداد نورون = 1 :

```
Accuracy: 0.77143
RMSE for training data: 0.57762
RMSE for test data: 0.56266
```

تعداد نورون = 30 :

```
Accuracy: 0.81429
RMSE for training data: 0.64285
RMSE for test data: 0.64608
```

تعداد نورون = 500 :

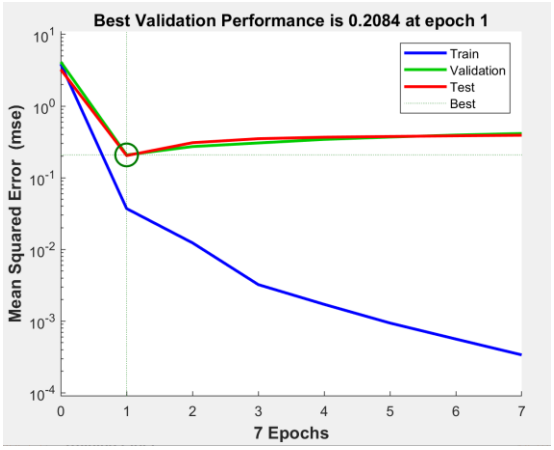
```
Accuracy: 0.77143
RMSE for training data: 0.66586
RMSE for test data: 0.66788
```

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 از 30 به 500 نورون نیز مقدار RMSE افزایش یافته است.البته میزان افزایش آن از حالت قبل کمتر است.به عبارت کلی با افزایش تعداد نورونRMSE نیز افزایش می یابد.

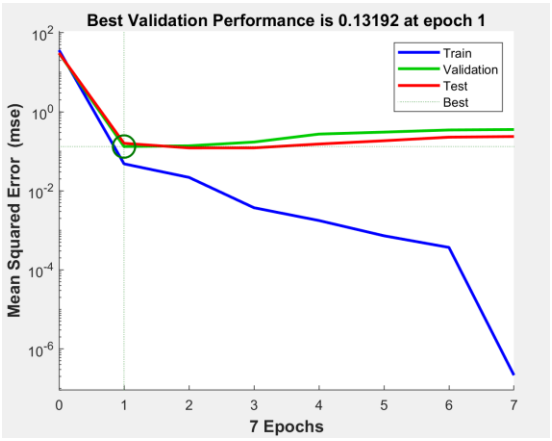
البته برای Accuracy این موضوع صادق نیست. یعنی accuracy در تعداد نورون مشخصی بیشینه است و با افزایش و کاهش نورون این مقدار کاهش می یابد.

تاثیر تغییرات max fail روی آموزش :

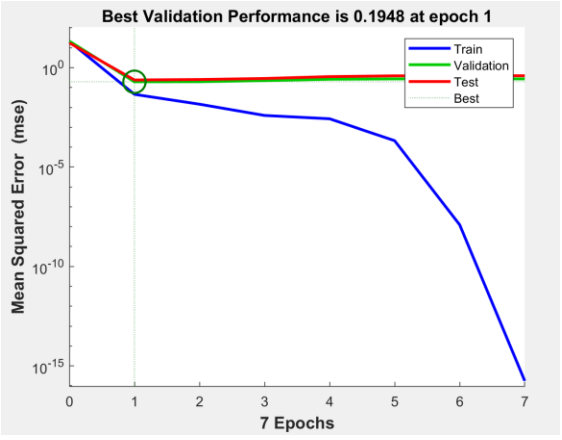
Max fail = 1 :



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 برداز می شود.

منابع:

۱- ویکی پدیا

۲- https://www.w3schools.com/

۳- Youtub.com

۴- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

۵- Chat GPT