



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیووتر
شبکه های عصبی و یادگیری عمیق

مینی پروژه شماره ۲۵

امیرمحمد رنجبر پازکی، هرا موسوی موحد	نام و نام خانوادگی
۸۱۰۱۹۶۶۲۹، ۸۱۰۱۹۹۳۴۰	شماره دانشجویی
۱۴۰۰ /۴ /۴	تاریخ ارسال گزارش

در این تمرین سوالات از بین سوالات ۱ تا ۳ سوال ۱ و ۳ برای حل انتخاب شده است.

فهرست گزارش سوالات

- سوال ۱ – آشنایی با کاربرد شبکه‌های عصبی بازگشتی در سری زمانی..... 3
- سوال ۳ – آشنایی با کاربرد شبکه‌های عصبی بازگشتی در متن..... 33
- سوال ۴ – آشنایی با مقالات مرتبط 36

سوال 1 – آشنایی با کاربرد شبکه‌های عصبی بازگشتی در سری زمانی

۱ - در ابتدا برای این سوال داده‌های موردنظر از `pdr.get_data_yahoo` گرفته شد. سپس، `index` آن‌ها حذف شد و ستون‌های `Date` و `Adj Close` از ستون‌ها حذف شدند.

سپس، `label`‌ها جدا شدند و اولین `label` نیز حذف شد چراکه داده‌های روزهای قبل آن وجود نداشتند. آخر سطر ویژگی‌ها نیز حذف شد چرا که برچسب روز بعد آن وجود نداشت.

دو ویژگی `Open` و `Close` انتخاب شدند. دلیل این موضوع مشخص شدن روند حرکتی `bitcoin` بود. سپس، از `StandardScaler` برای `standardization` داده‌ها استفاده شد تا میانگین آن‌ها صفر و واریانس آن‌ها یک شود تا یک توزیع نرمال و هم مقیاس در همه ویژگی‌ها داشته باشیم.

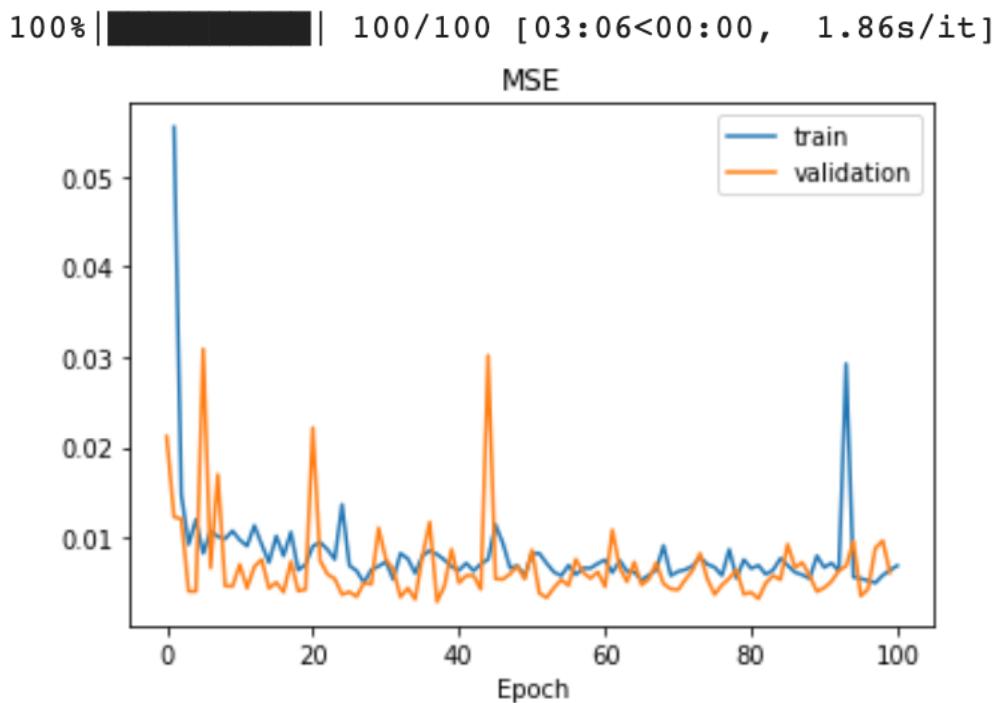
برای درست کردن `dataloader`‌ها ابتدا داده‌ها تست کنار گذاشته شدند. تعداد این داده‌ها بیشتر از ۷ بازه‌ی ۲۹ تایی بود. دلیل این موضوع آن بود که طول `sequence` ۲۸ درنظر گرفته شد. ما نیاز به داده‌ی ۲۸ روز به علاوه یک روز به عنوان هدف داریم. پس، ۷ بازه‌ی ۲۹ تایی برای تست انتخاب می‌کنیم که مدل در هنگام آموزش آن را نبیند. سپس، `sequence`‌های ۲۸ تایی پشت سر هم تولید می‌کنیم. به عنوان مثال روز اول تا ۲۸ام، روز دوم تا ۲۹ام و به همین ترتیب...

حال این `sequence`‌ها را با استفاده از `train_test_split` به نسبت ۱ به ۹ به داده‌ی اعتبار سنجی و آموزش تقسیم می‌کنیم. داده‌ی تست نیز پیش از این جدا شده است. حال برای این که بتوان از این داده‌ها `dataloader`‌های مناسب ساخت، یک `dataset` خودمان می‌نویسیم. حال با استفاده از این `dataset` و ویژگی‌ها و `label`‌ها، `dataset` مربوط به داده‌های آموزش، اعتبار سنجی و آزمون را به `dataloader` با سایز `batch` ۴ تبدیل می‌کنیم. در حقیقت، شکل نهایی `batch` به صورت (4, 28, 2) برای ویژگی‌ها و `batch` به صورت (1, 4) برای برچسب‌هاست.

پس از آن، تابع‌هایی برای آموزش و اعتبار سنجی `epoch` به `epoch` و رسم نمودار خط‌آ و تست و رسم مقادیر حقیقی به همراه مقادیر پیش‌بینی شده زده شده است.

۲ - سه شبکه‌ی LSTM، GRU و RNN طراحی و پیاده‌سازی شده‌اند. این شبکه‌ها از یک سلول خودشان به همراه یک لایه fully connected تشکیل شده‌اند. برای این شبکه‌های تابع هزینه و فعال‌ساز یکسان در نظر گرفته شده است. برای هر شبکه زمان آموزش، نمودار خطاب بر حسب epoch برای داده‌ی آموزش و اعتبارسنجی، مقادیر پیش‌بینی شده و واقعی برای داده‌تست به همراه میزان MAE گزارش شده است.

شبکه‌ی RNN:



شکل ۱- زمان آموزش و نمودار خطاب بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی RNN



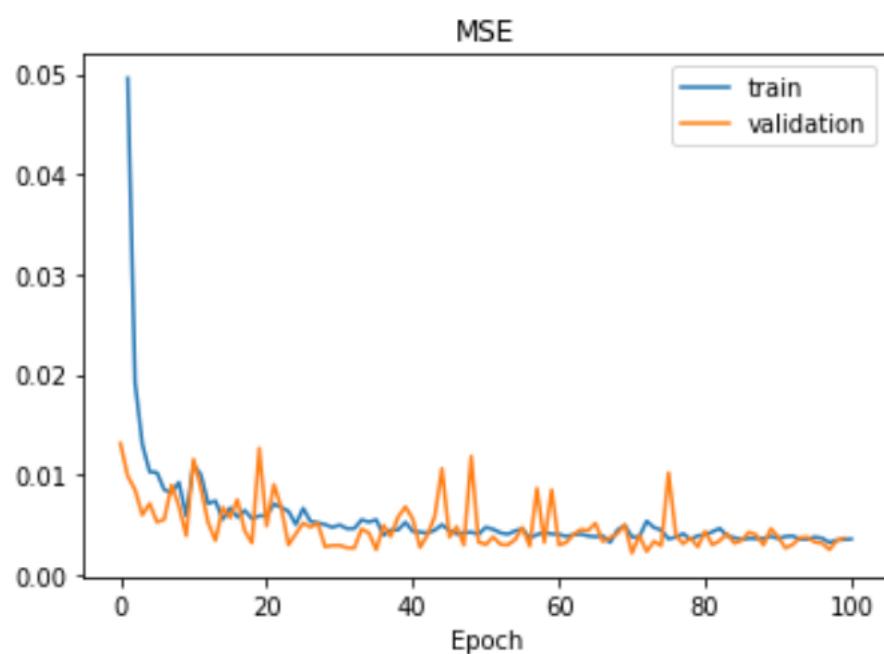
MAE: 628.5429894538341

MSE: 1184451.806555702

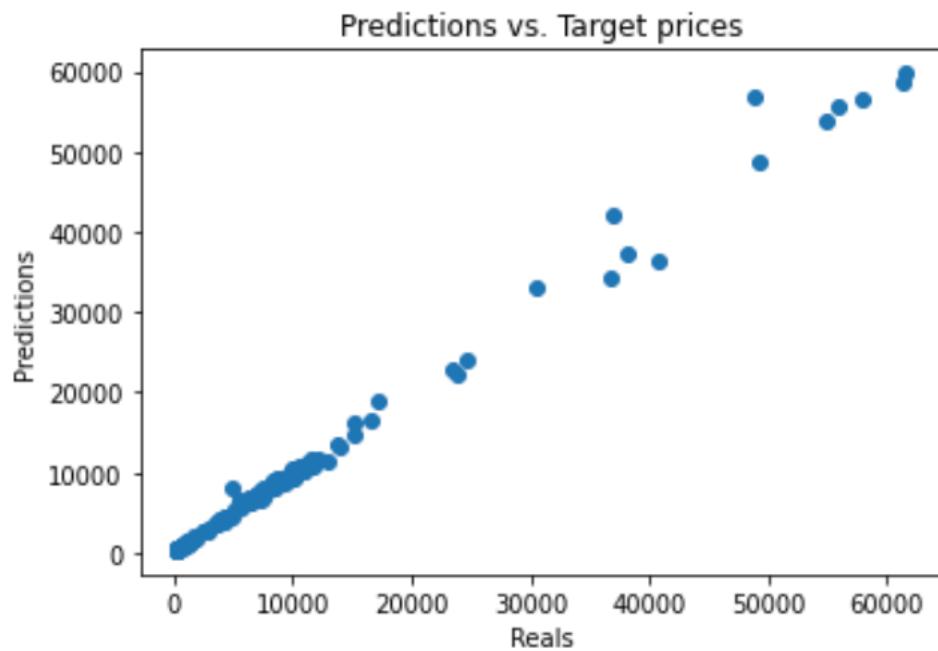
شکل ۲- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE شبکه RNN

شبکه‌ی LSTM

100% |██████████| 100/100 [03:23<00:00, 2.03s/it]



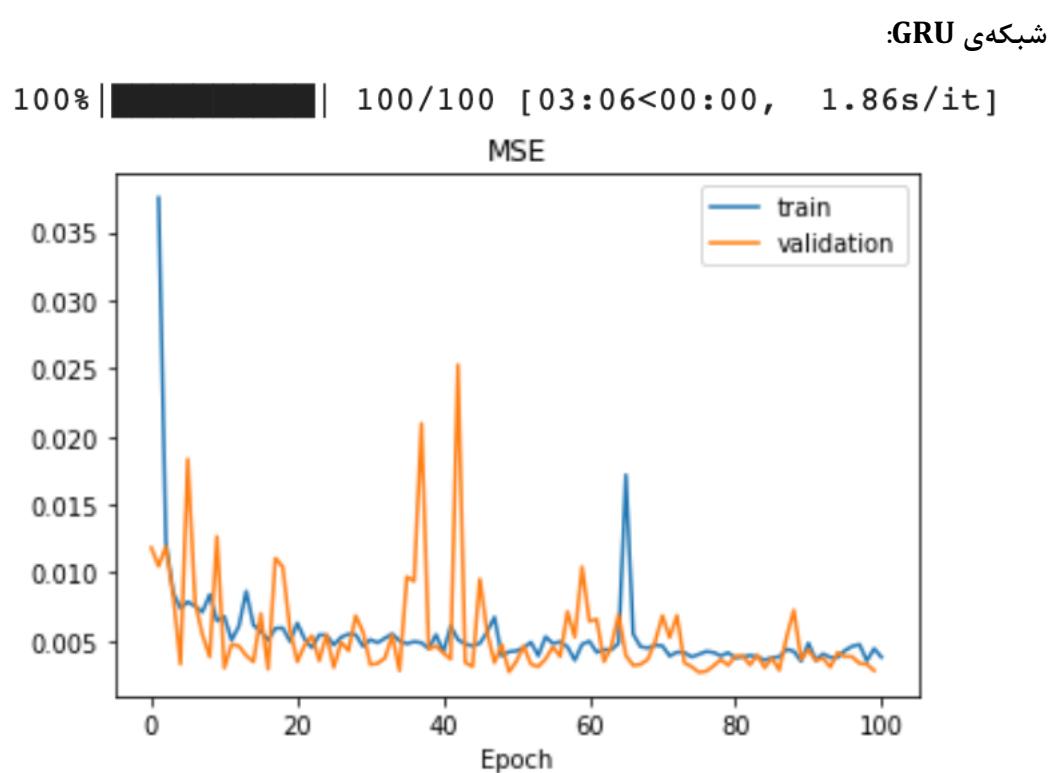
شکل ۳- زمان آموزش و نمودار خطأ بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی LSTM



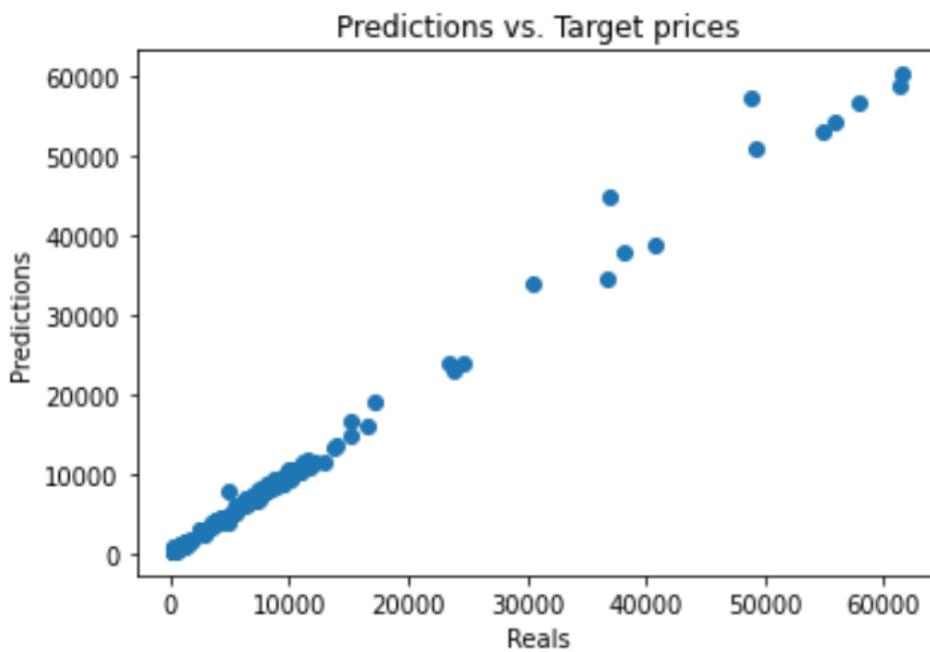
MAE: 362.81503568150697

MSE: 712785.8823894638

شکل ۴- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE شبکه LSTM



شکل ۵- زمان آموزش و نمودار خطای بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی GRU



MAE: 383.8349741702259

MSE: 871977.026220915

شکل ۶- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE شبکه GRU

همهی شبکه‌ها به میزان ۱۰۰ epoch آموزش داده شده‌اند. جدول زیر نشان‌دهنده مقایسه نتایج این سه شبکه است.

جدول ۱- جدول مقایسه سه شبکه RNN و LSTM و GRU

MSE	MAE	زمان آموزش(ثانیه)	نوع شبکه
۱۱۸۴۴۵۱.۸	۶۲۸.۵	۱۸۶	RNN
۷۱۲۷۸۵.۹	۳۶۲.۸	۲۰۳	LSTM
۸۷۱۹۷۷	۳۸۳.۸	۱۸۶	GRU

همانطور که مشاهده می‌کنید شبکه‌های RNN و GRU نسبت به LSTM سرعت بیشتری دارند. این به دلیل سنگین تر بودن شبکه‌ی LSTM است که با توجه به اتصالات پیچیده‌تر آن زمان بیشتری برای آموزش می‌خواهد.

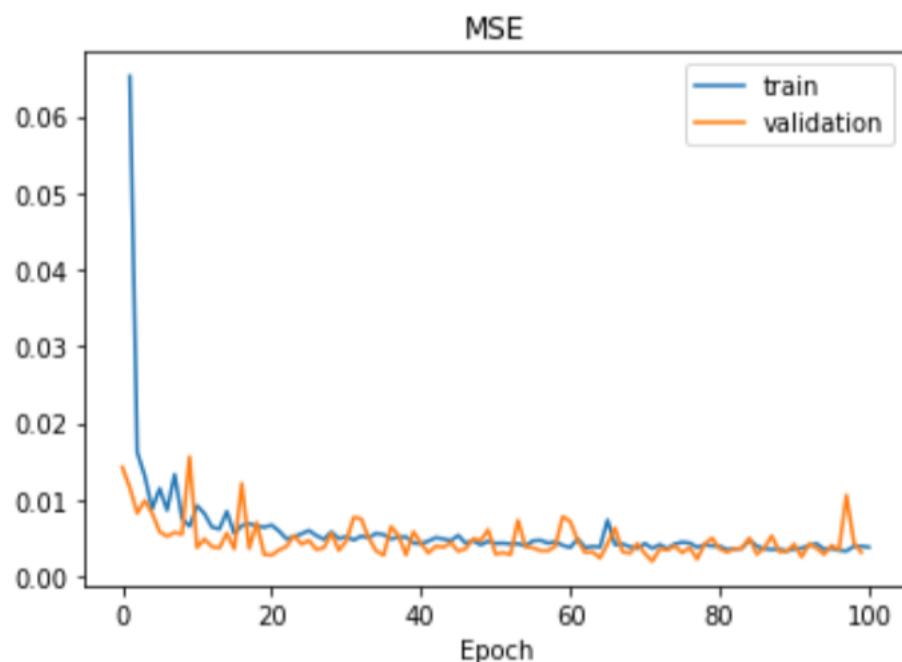
شبکه‌ی LSTM بهترین دقت یا همان کمترین میزان خطاهای را دارد و همین طور، نمودار قابل قبولی نیز برای مقادیر پیش‌بینی شده‌دارد. این موضوع به دلیل قدرت بیشتر شبکه‌ی LSTM در بیرون کشیدن رابطه‌ها به نسبت RNN در این مسئله است.

روند پیش‌بینی LSTM نسبت به دو شبکه‌ی دیگر کمنوسان‌تر است و پرش در خطا نداشته است. همچنانی، خطای زیادی در پیش‌بینی نداشته است.

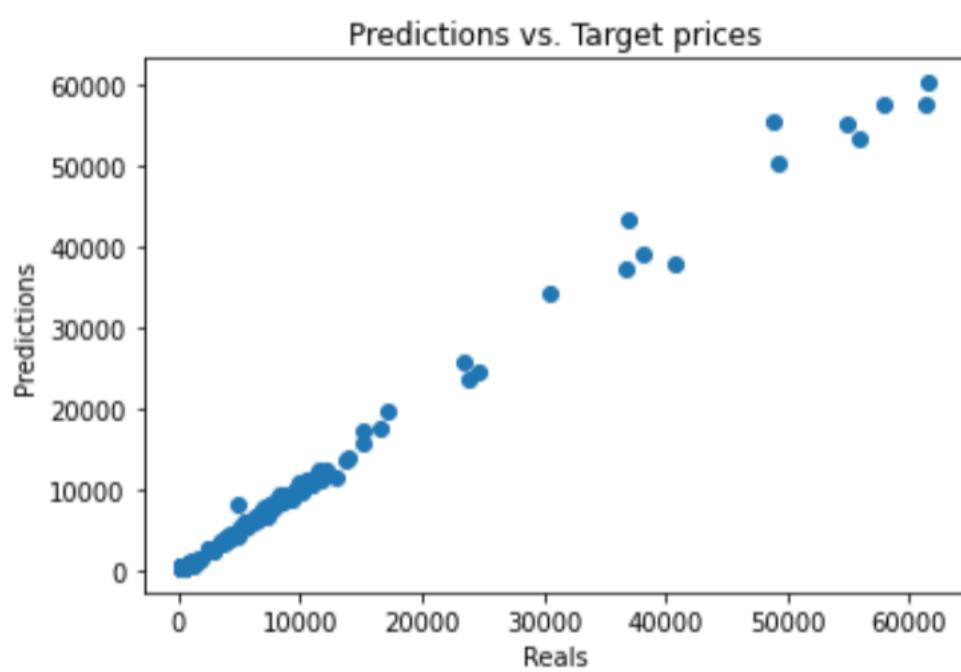
بهترین شبکه‌ی انتخابی در این مرحله LSTM است.

۳ - همانطور که مشاهده می‌کنید MSE عملکرد بهتری داشته است. زیرا این تابع هزینه باعث می‌شود که شبکه مقادیر پرتو پیش‌بینی نکند. با توجه به بازه‌ی بزرگ نوسانات بیت‌کوین امکان پیش‌بینی داده‌ی پرتو بالاست. به همین دلیل استفاده از این داده بهتر از MAE است. نتایج این دو تابع در زیر قابل مشاهده است.

MSE - معیار

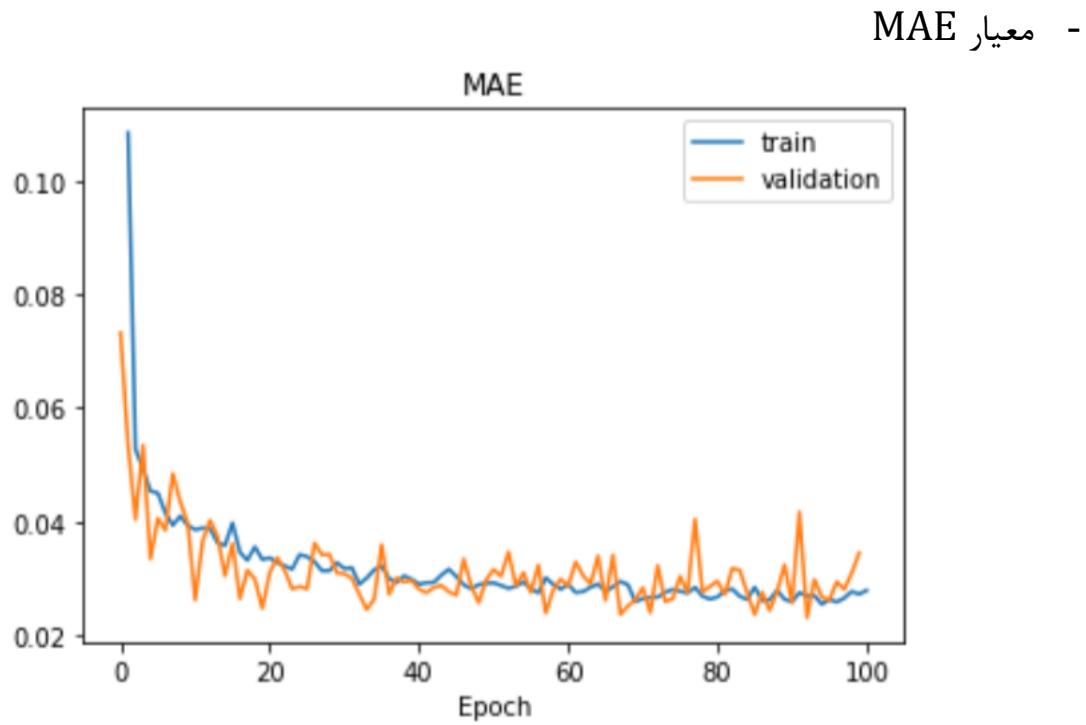


شکل ۷ - نمودار خط‌الاطلاع بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای mse

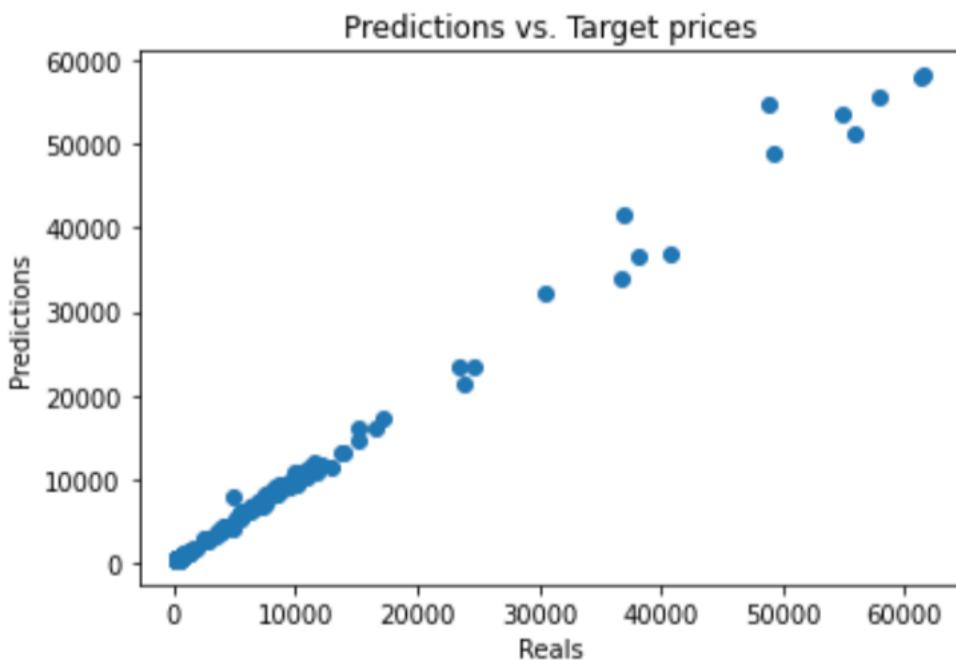


MAE: 355.11601573896604
 MSE: 726030.4655282064

شکل ۸ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای mse



شکل ۹ - نمودار خطاب بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای mae



MAE: 400.6003307168415
MSE: 718052.1300879266

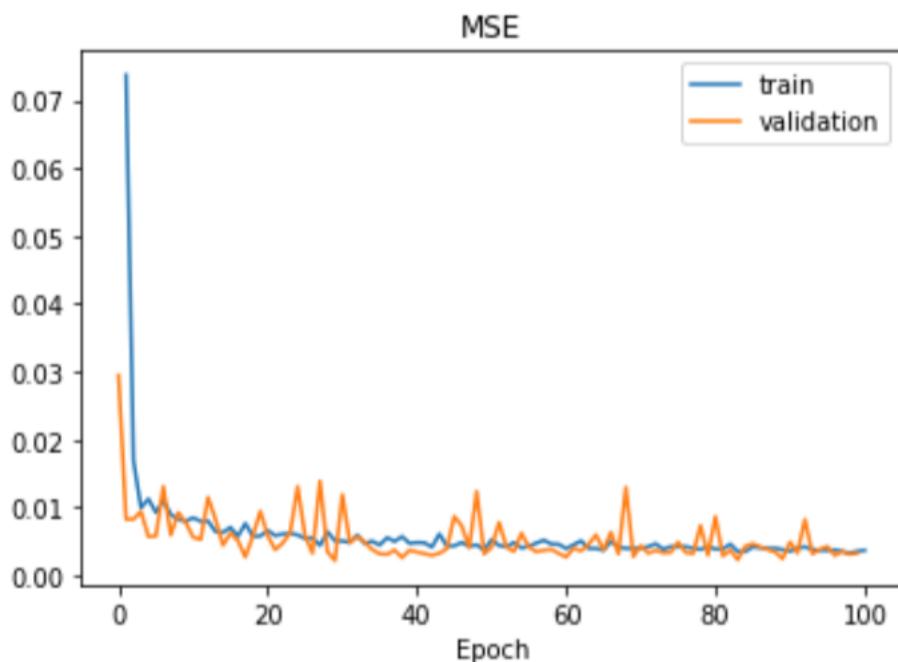
شکل ۱۰- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه **MAE** و **MSE** برای **mae**

نتایج روش‌های بهینه‌سازی مختلف در زیر قابل مشاهده است.

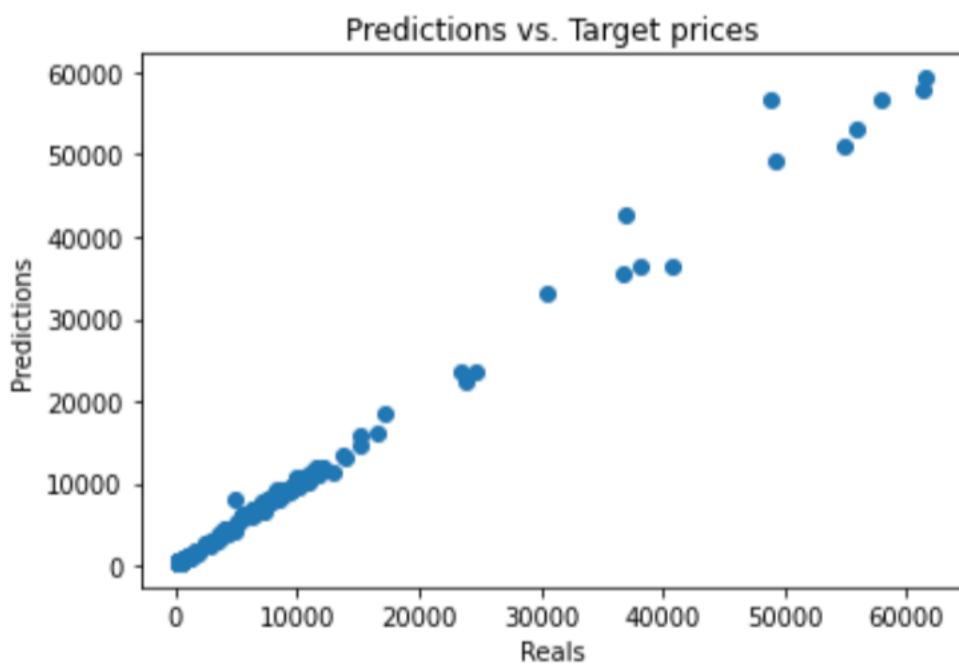
همانطور که مشاهده می‌کنید RMSProp بهترین عملکرد را در میان سایرین داشته است. این تابع کمترین میزان MSE و MAE را داشته است. نمودار این بهینه ساز نیز بسیار بهتر از مابقی است. دلیل بهتر بودن این بهینه ساز استفاده هوشمندانه آن از میانگین متحرک مربع گرادیان برای نرمالایز کردن خود گرادیان است. به دلیل سری زمانی بودن داده‌ی ما استفاده از این میانگین متحرک مفید بوده است و به نتایج بهتری رسیده است.

در میان این توابه بهینه ساز SGD سریع‌تر از بقیه است چراکه تابع بهینه سازی سریع‌تری دارد. عملکرد Adagrad نیز بعد از آن قابل قبول بوده است.

Adam optimizer -



شکل ۱۱ - نمودار خطای بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای Adam

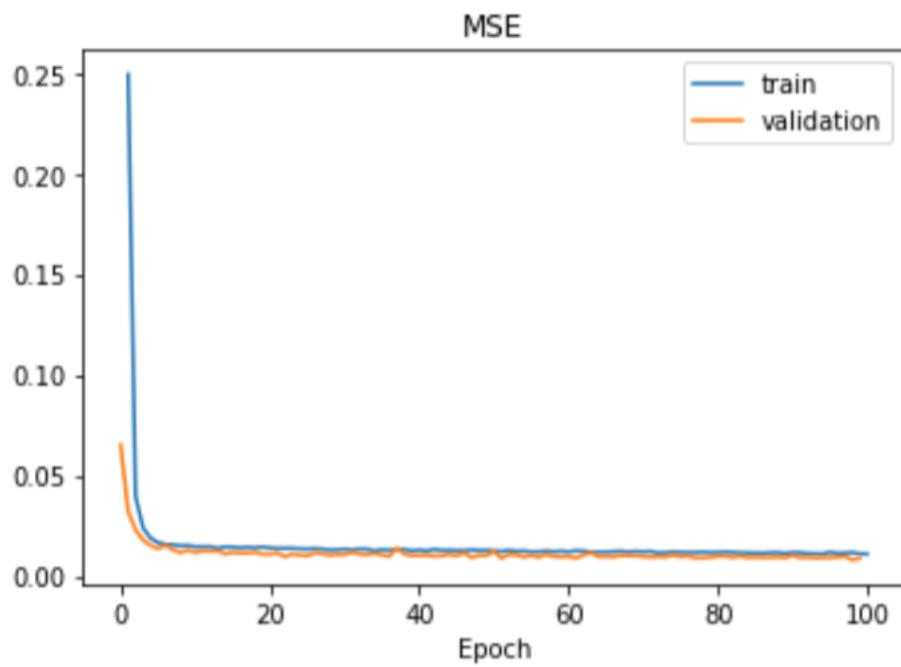


MAE: 388.2973291547467

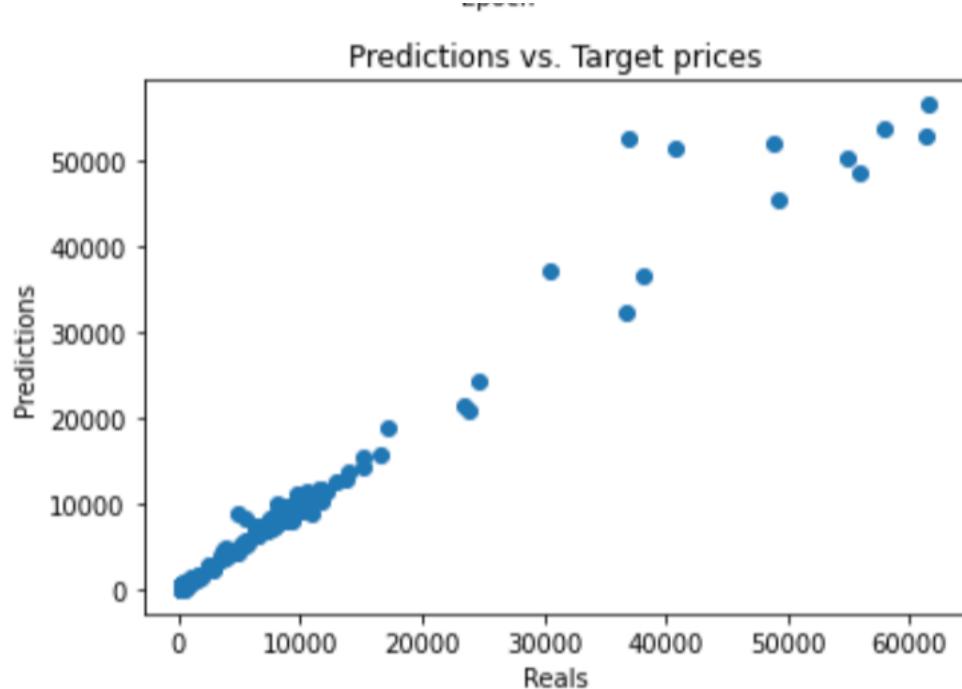
MSE: 822431.1741875098

شکل ۱۲ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای Adam

ADAgrad optimizer -

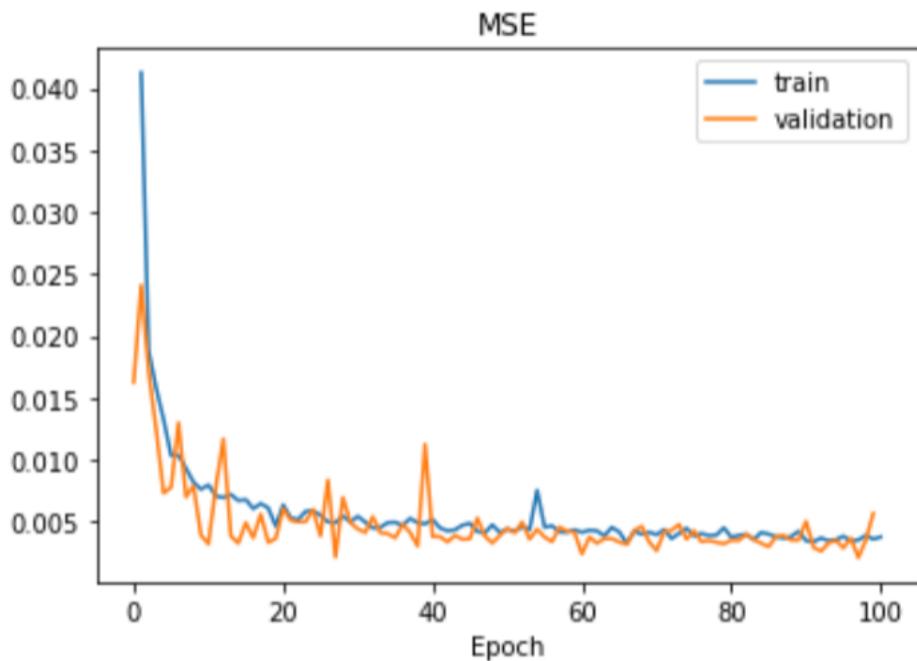


شکل ۱۳ - نمودار خطای میانگین مربعی (MSE) بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای ADAgrad

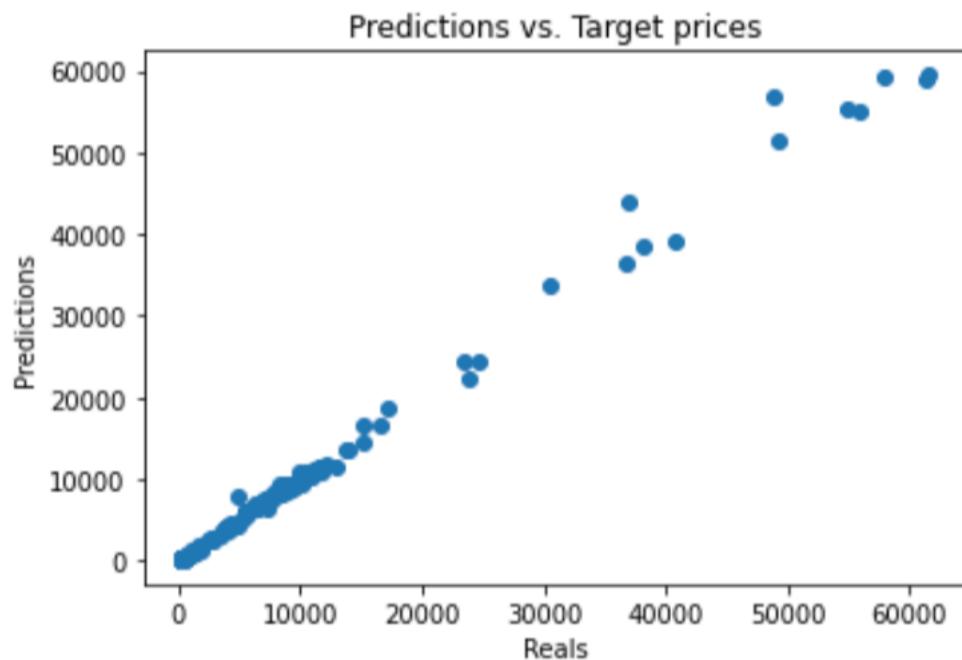


شکل ۱۴ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای ADAgrad

RMSprop optimizer -

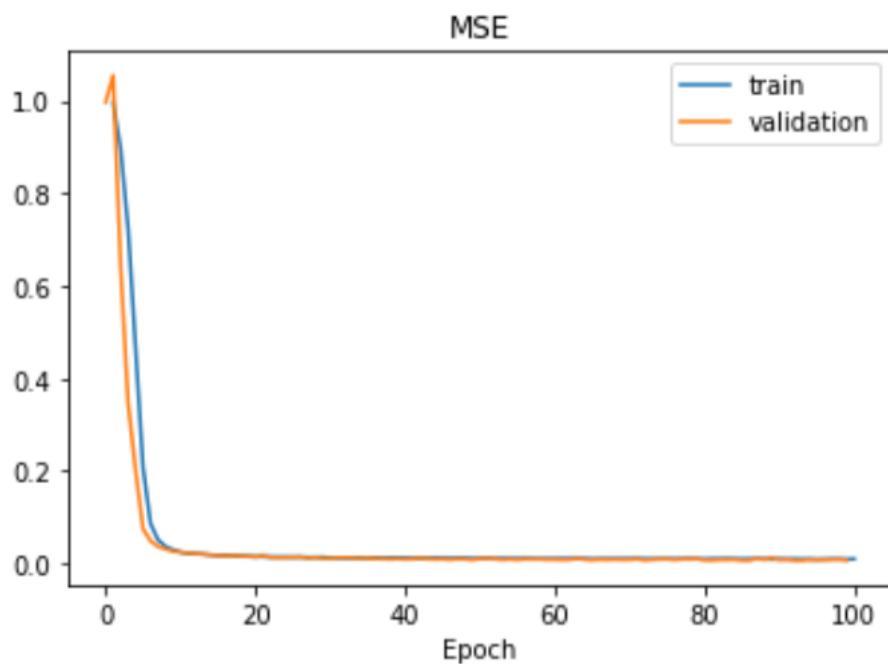


شکل ۱۵ - نمودار خطای خطا بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای RMSprop

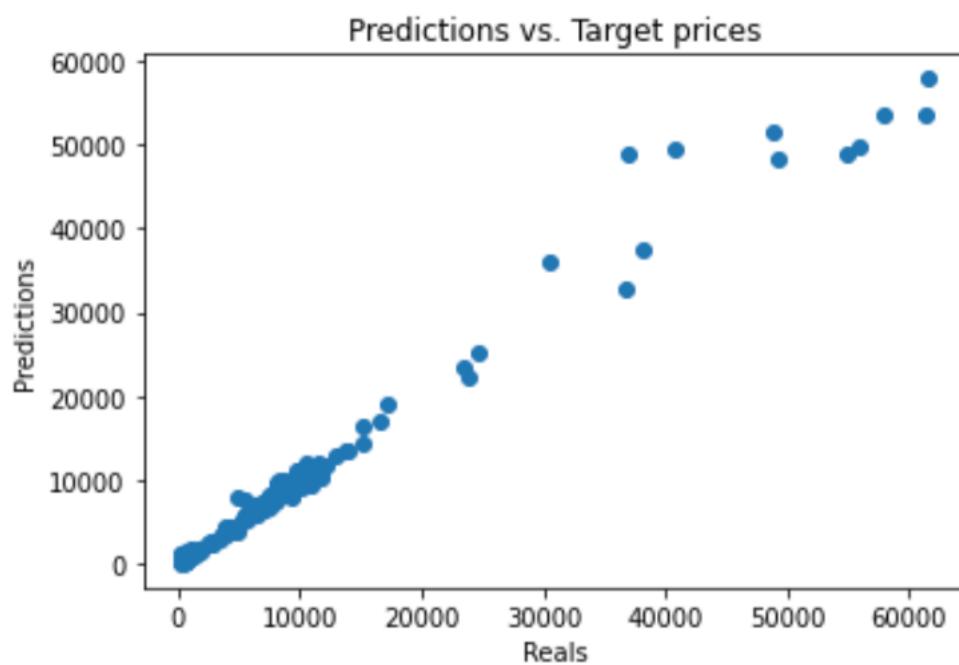


شکل ۱۶ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای RMSprop

SGD optimizer -



شکل ۱۷ - نمودار خطأ بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای SGD



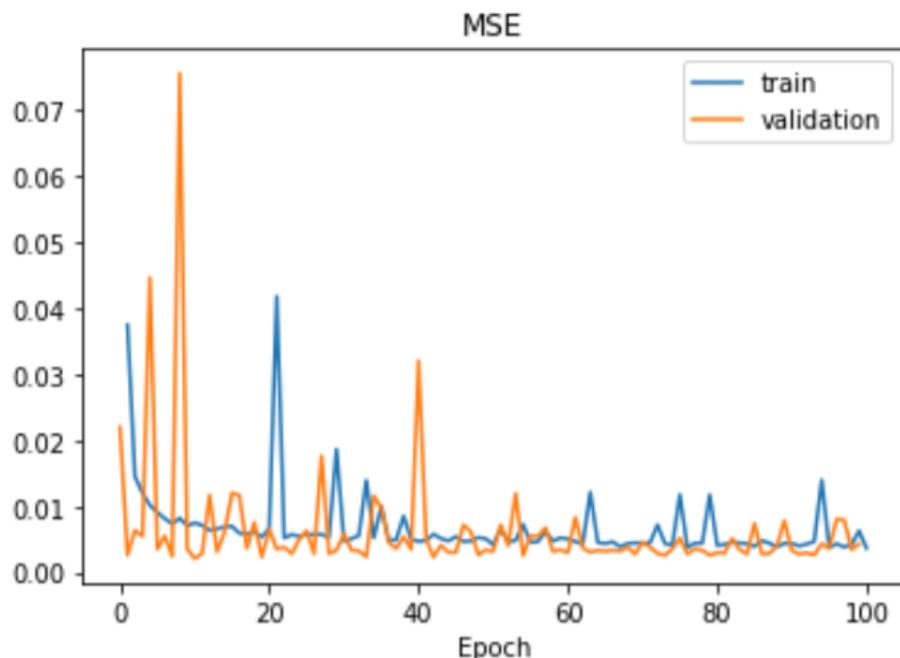
MAE: 619.0514009087907

MSE: 2104038.793558243

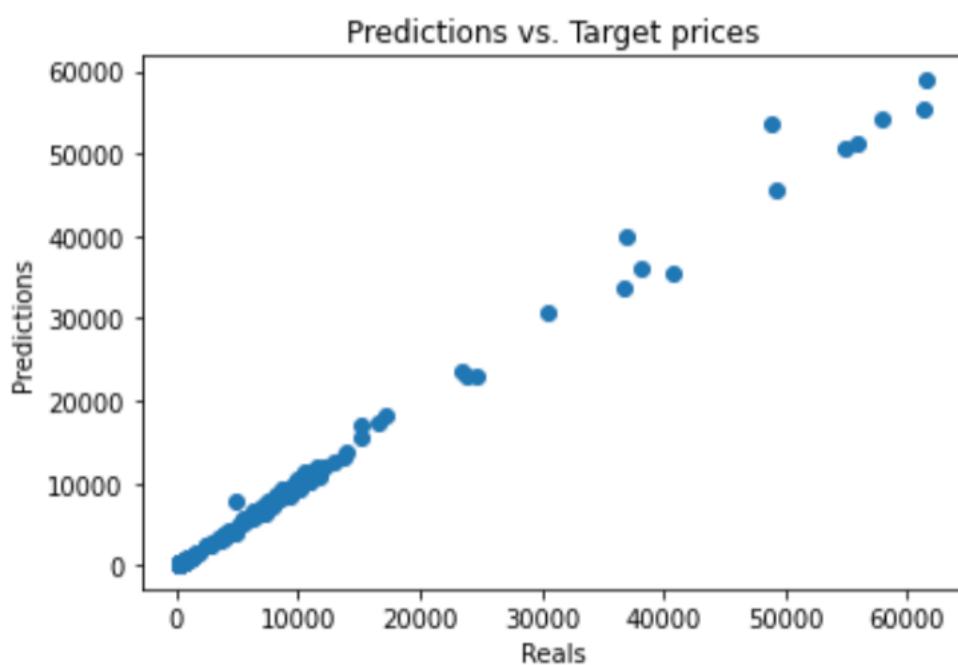
شکل ۱۸ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای SGD

۴ - همانطور که در نتایج زیر می‌بینید، افزودن dropout به هر شبکه خطای آن را بالاتر برده است. دلیل این موضوع پایین بودن پیچیدگی شبکه است. به گونه‌ای که افزودن dropout باعث خطای bias در آن و مشکل under fitting می‌شود. به همین دلیل، دقت شبکه کاهش و خطا افزایش می‌یابد.

RNN - شبکه



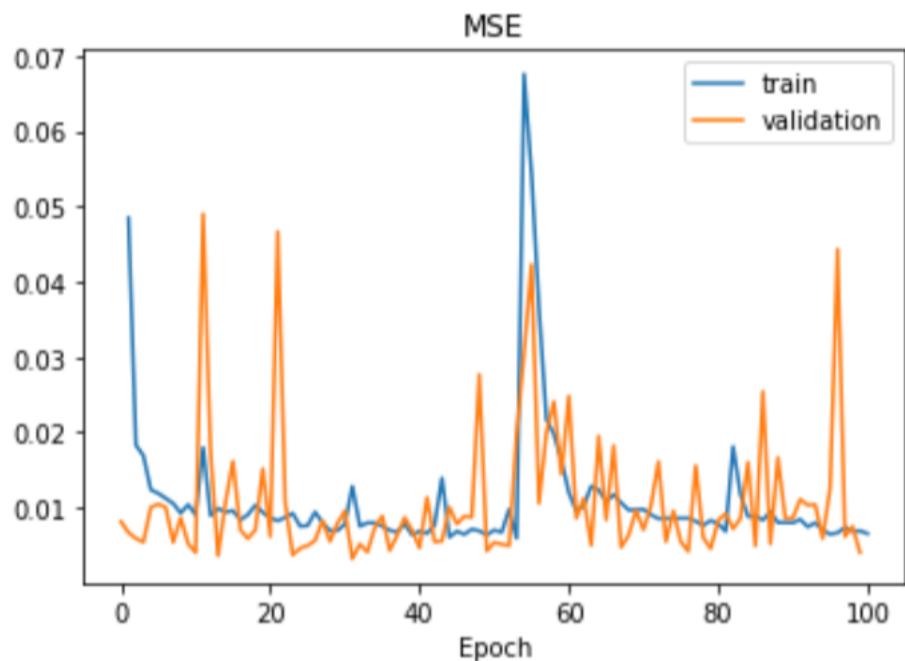
شکل ۱۹ - نمودار خطای بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای RNN بدون dropout



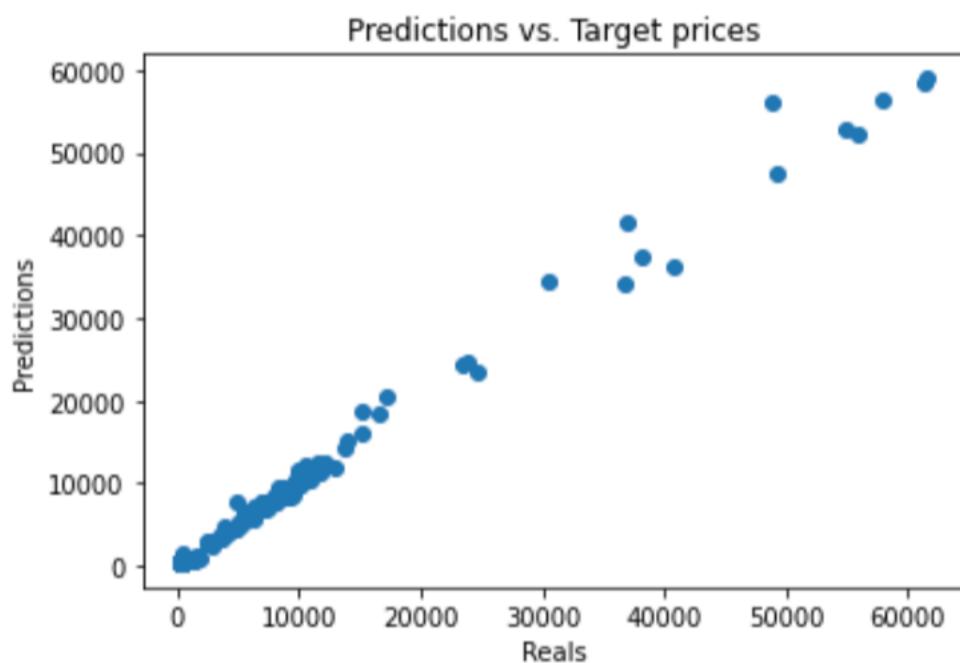
MAE: 391.57262050264615

MSE: 866552.6694797262

شکل ۲۰- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه **MSE** و **MAE** برای RNN بدون **dropout**

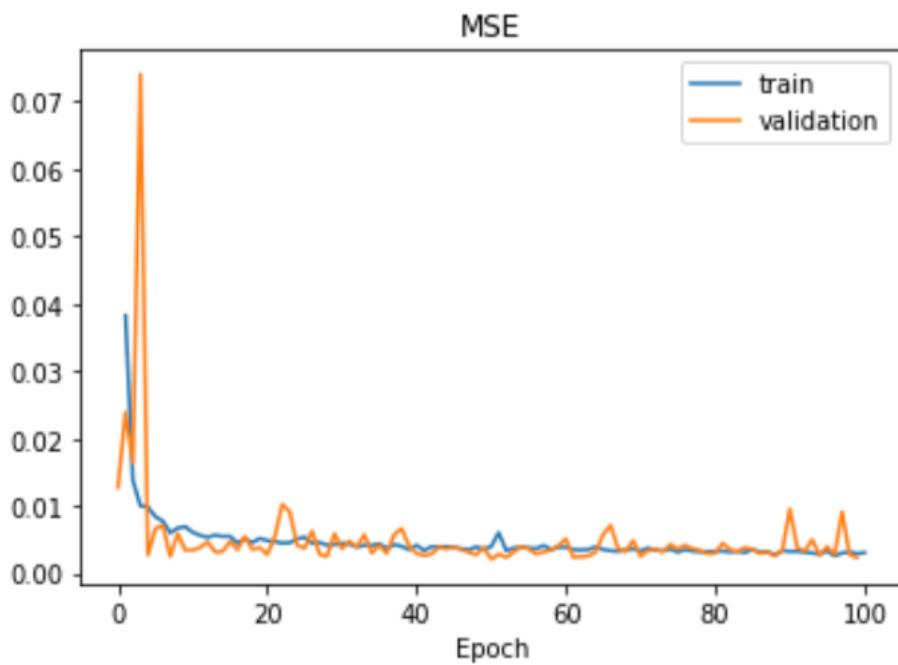


شکل ۲۱ - نمودار خطا بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای RNN با dropout

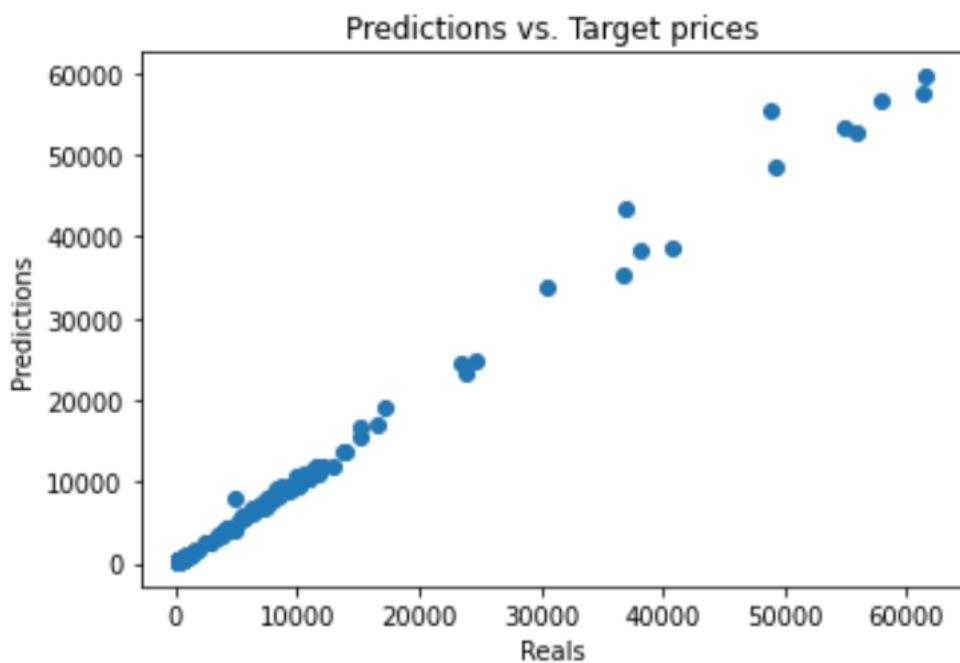


شکل ۲۲- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه **MSE** و **MAE** با **RNN** و **dropout**

LSTM شبکه -



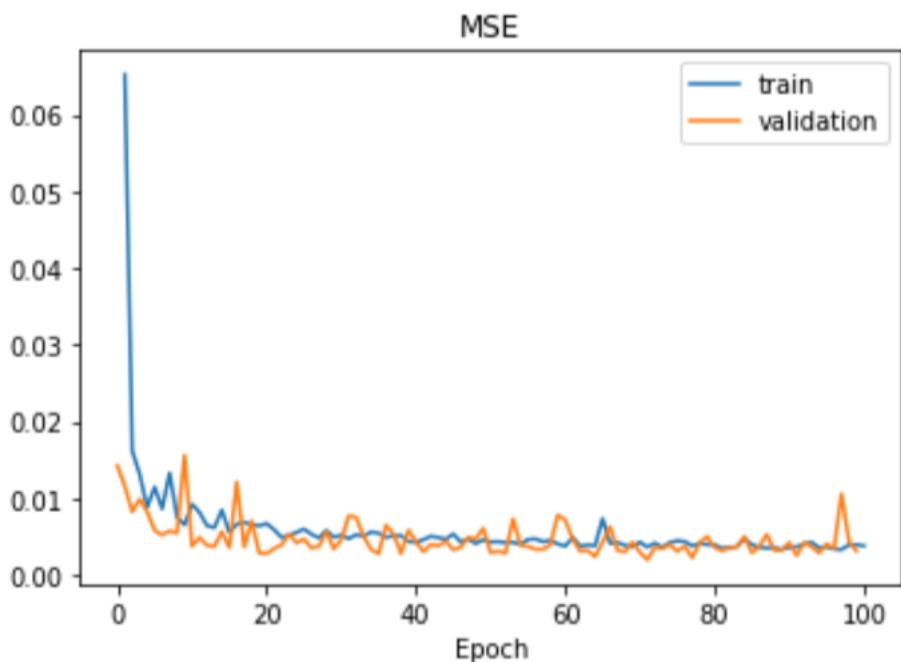
شکل ۲۳ - نمودار خطای آموزش و اعتبارسنجی برای LSTM بدون dropout



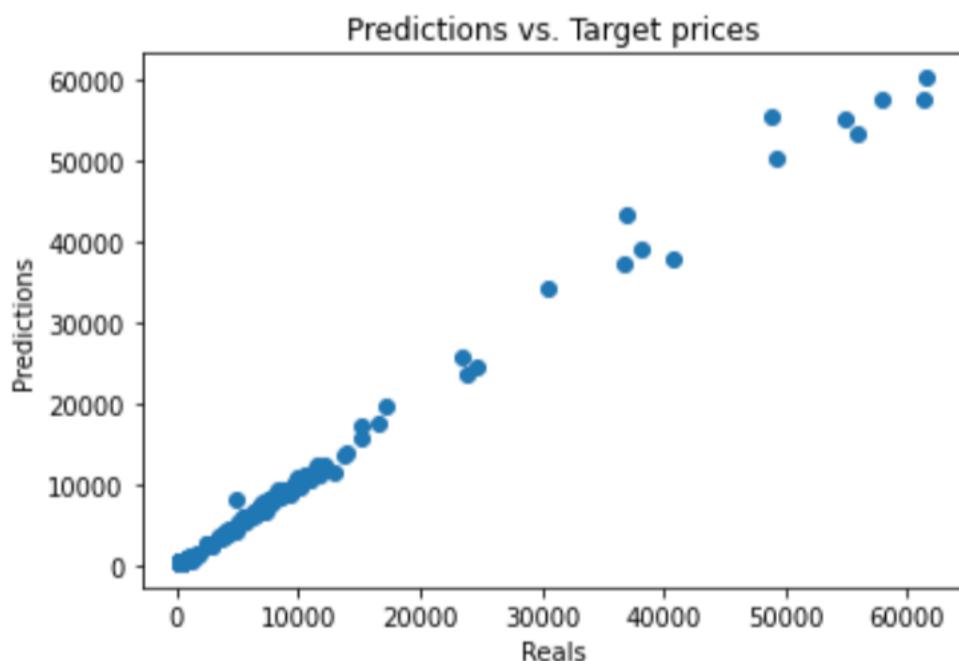
MAE: 283.45390269172646

MSE: 684272.9202275348

شکل ۲۴ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای LSTM بدون dropout

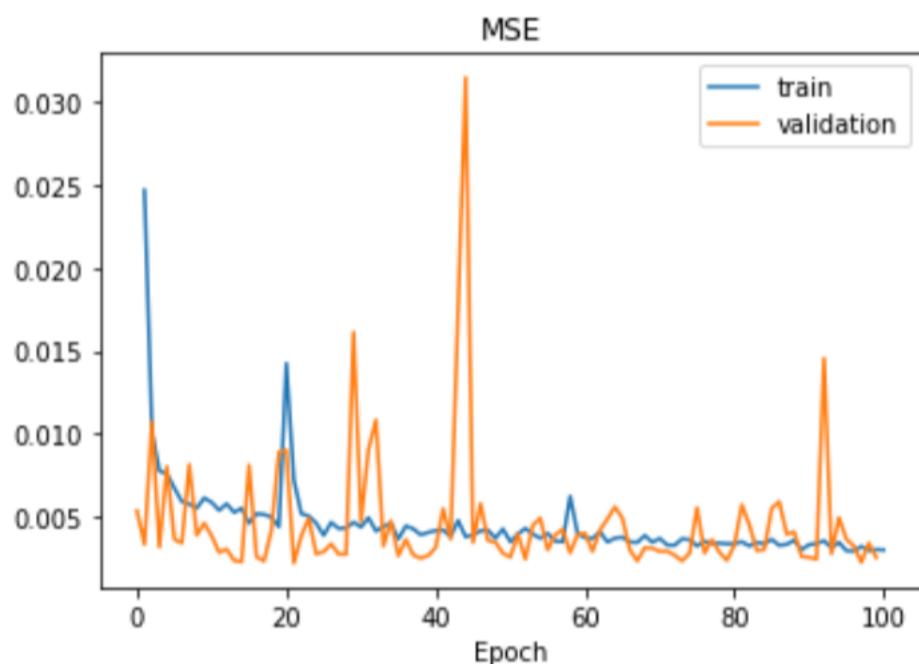


شکل ۲۵ - نمودار خطای بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای LSTM با dropout



شکل ۲۶ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای LSTM با dropout

شبکه GRU -



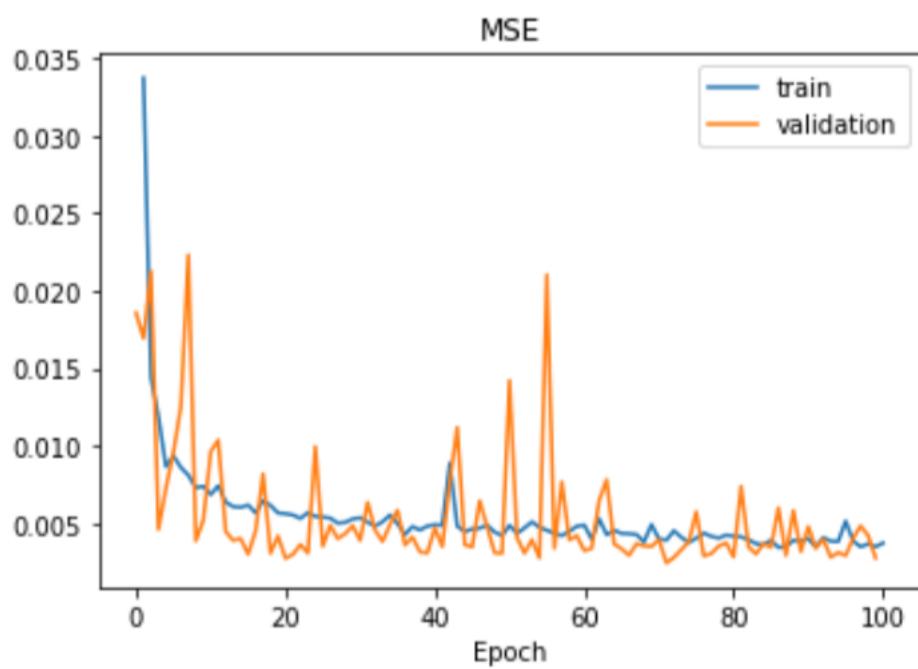
شکل ۲۷ - نمودار خطای بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای GRU بدون dropout



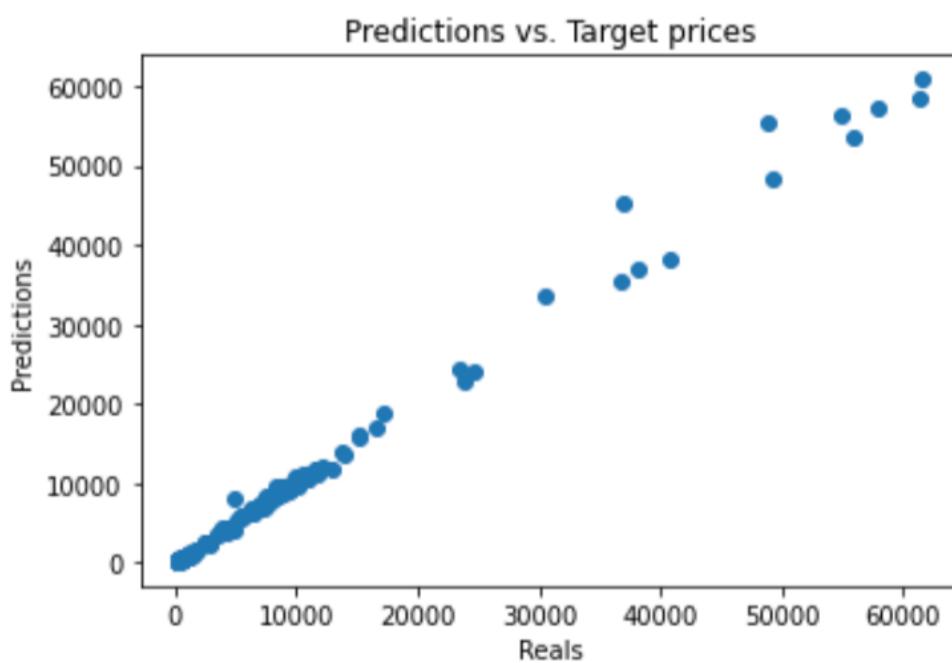
MAE: 317.9872828440053

MSE: 933055.2383456726

شکل ۲۸- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای GRU بدون dropout



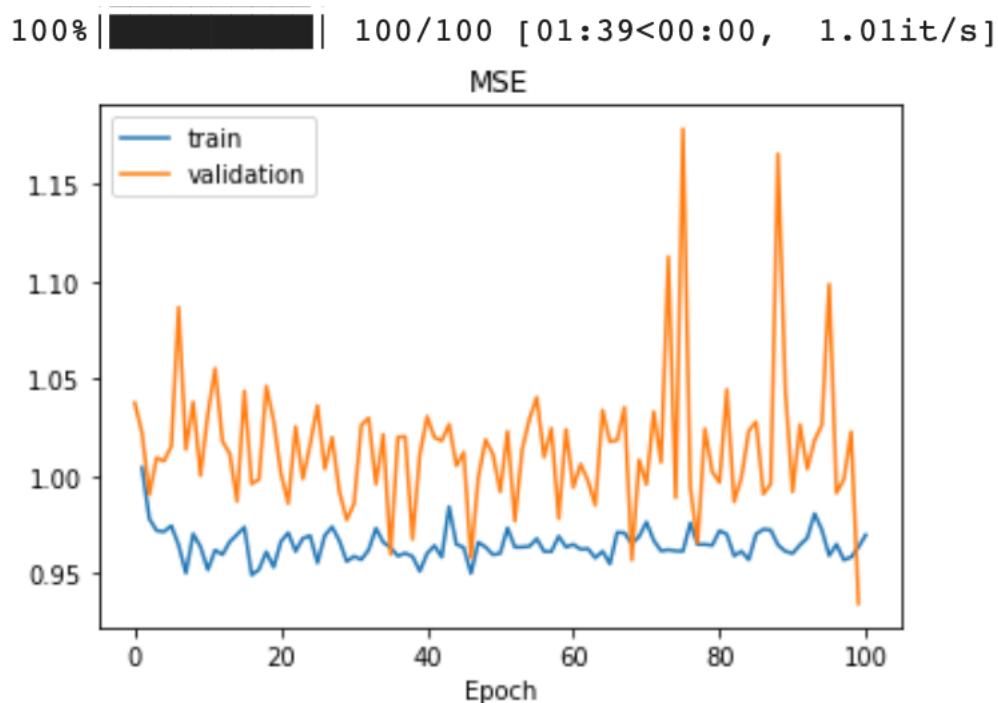
شکل ۲۹ - نمودار خطای خطا بر حسب epoch برای داده‌ی آموزش و اعتبارسنجی برای GRU با dropout



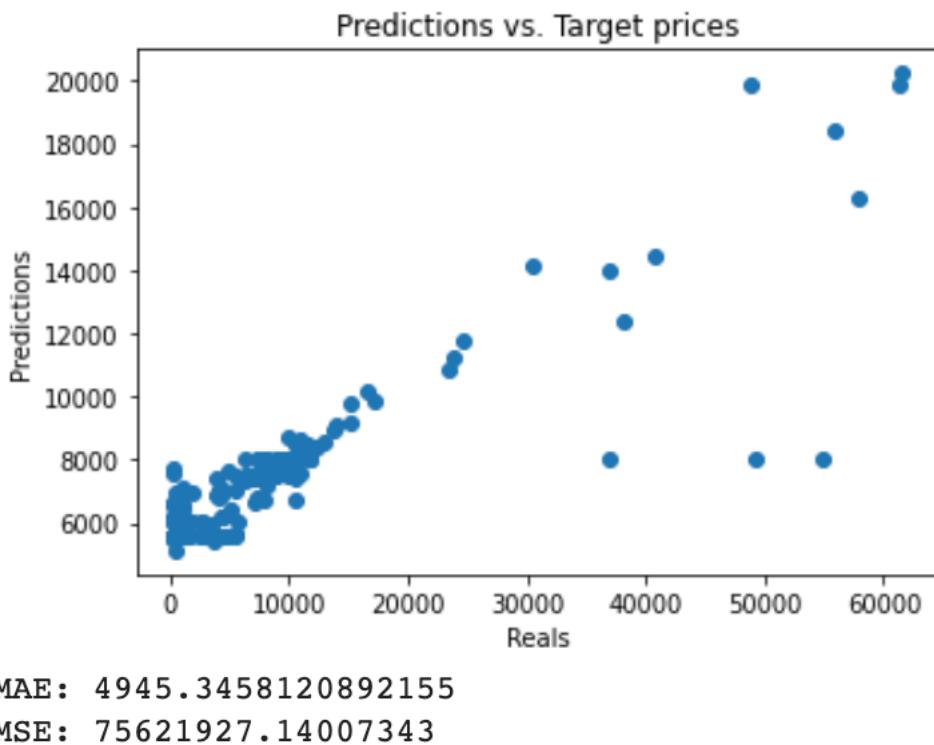
شکل ۳۰- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE با GRU

در این سوال ابتدا یک شبکه‌ی MLP پیاده‌سازی شده‌است که تعداد لایه‌ها و سایز آن‌ها در هنگام آموزش مشخص می‌گردد. این شبکه در کلاس MLP پیاده‌سازی شده‌است.

این شبکه‌ی MLP پنج لایه با سایزهای تعداد ویژگی‌ها(۵۶)، ۳۲، ۱۶، ۸ و ۱ دارد. نتایج این شبکه به صورت زیر است.



شکل ۳۱- زمان آموزش و نمودار خطاب بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی MLP



شکل ۳۲- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه **MAE** و **MSE** شبکه **MLP**

همانطور که مشاهده می‌کنید این شبکه عملکرد خوبی نداشته است. دلیل این موضوع می‌تواند مربوط به نوع شبکه باشد که نمی‌تواند سیر زمانی را ببیند و الگو را کشف کند. این شبکه انگار ۵۶ ویژگی ورودی دارد که هیچ ارتباط با یکدیگر ندارند و می‌خواهند هدف را حدس بزنند.

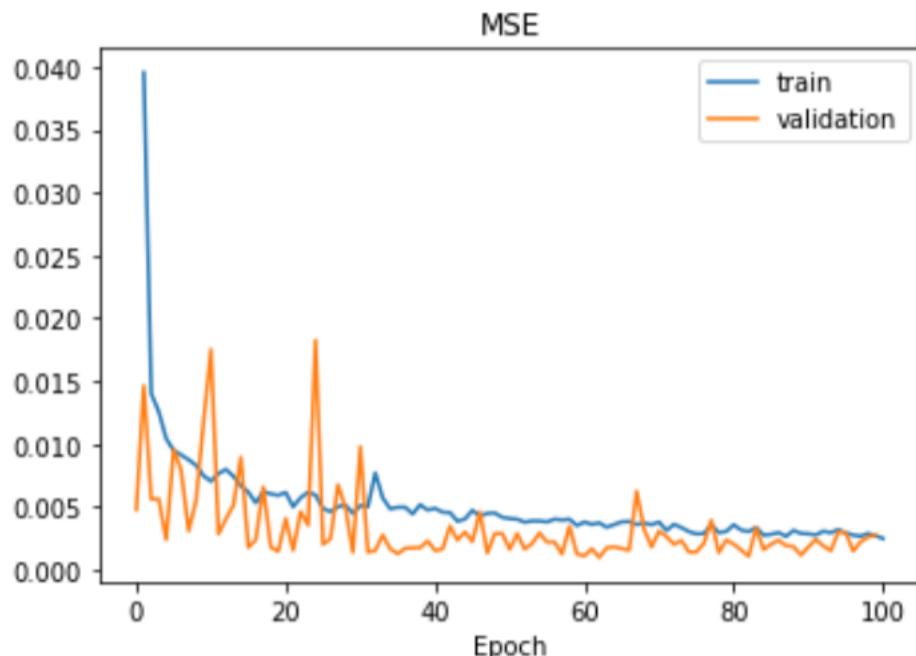
شبکه‌های بازگشتی امکان استفاده از سیر داده‌ها را دارند و نسبت به این شبکه‌ها برتری ارتباط دادن داده‌ها مختلف را دارند.

راهکاری که می‌توانست نتیجه را بهتر کند، آموزش به تعداد epoch بیشتری بود. چرا که تازه خطای validation رو به کاهش گذاشته است. از طرفی شاید می‌شد یک CNN autoencoder یا روی داده‌ها زد تا ویژگی‌هایشان بهتر استخراج شود و اطلاعات روند در آن‌ها لحاظ شود.

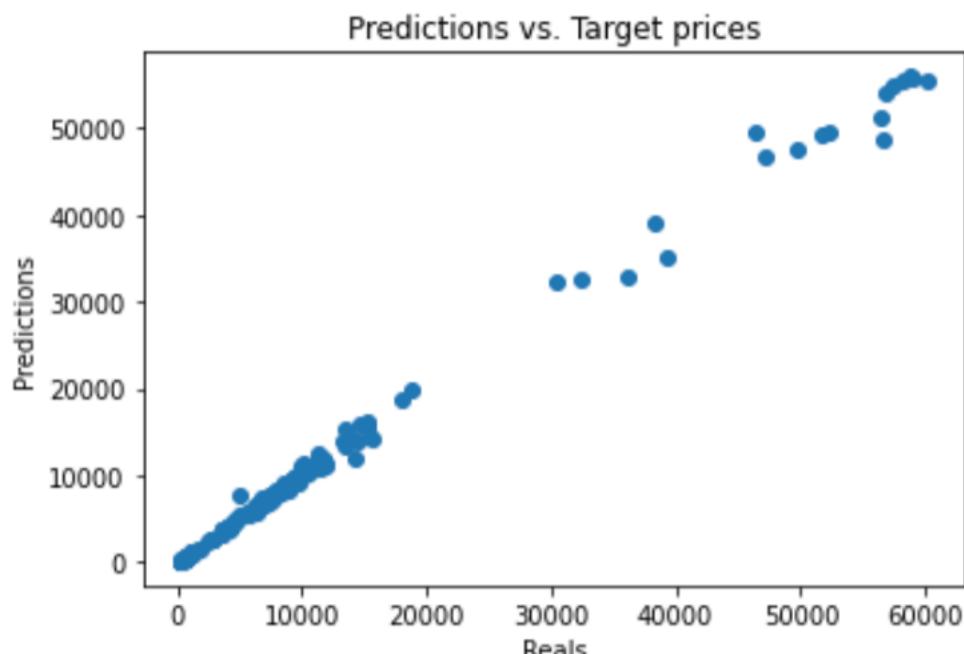
جدول ۲- جدول مقایسه چهار شبکه **MLP**, **GRU**, **LSTM**, **RNN** و

MSE	MAE	زمان آموزش(ثانیه)	نوع شبکه
۱۱۸۴۴۵۱.۸	۶۲۸.۵	۱۸۶	RNN
۷۱۲۷۸۵.۹	۳۶۲.۸	۲۰۳	LSTM
۸۷۱۹۷۷	۳۸۳.۸	۱۸۶	GRU
۷۵۶۲۱۹۲۷.۱	۴۹۴۵.۳	۹۹	MLP

خبر. افزودن تمام ویژگی‌ها باعث بهتر شدن نتیجه نشد. دلیل این موضوع می‌تواند وجود همبستگی میان ویژگی‌های مختلف روزهای متوالی باشد. این کار صرفا باعث می‌شود افزونگی به مدل افزوده شود و قدرت تعمیم مدل پایین بیاید. نتایج این کار در زیر مشاهده می‌شود.



شکل ۳۳ - نمودار خطأ بر حسب epoch برای داده‌ی آموزش و اعتبارسنجی برای تمام ویژگی‌ها



MAE: 416.25466051536966

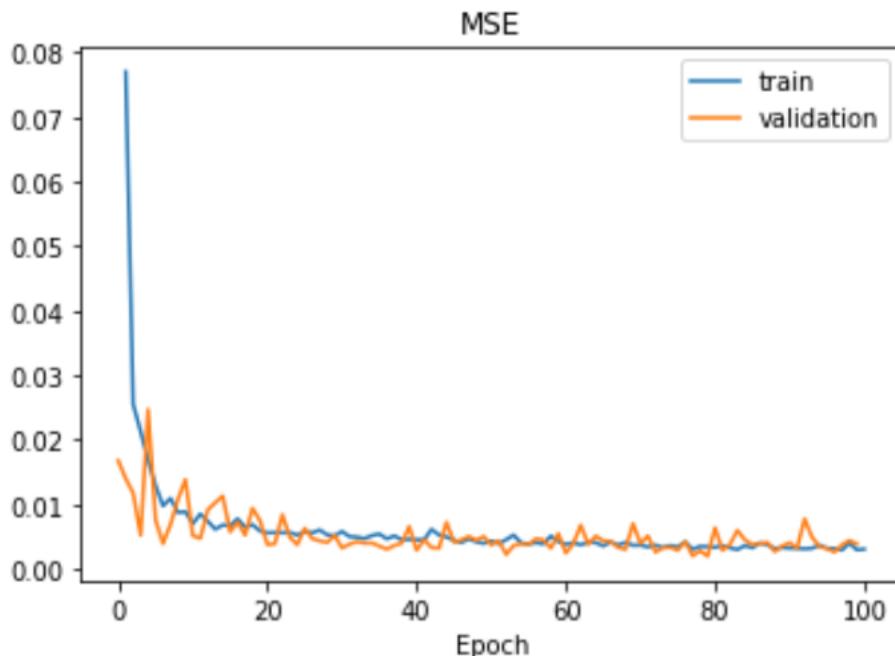
MSE: 1049637.915139839

شکل ۳۴- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای تمام ویژگی‌ها

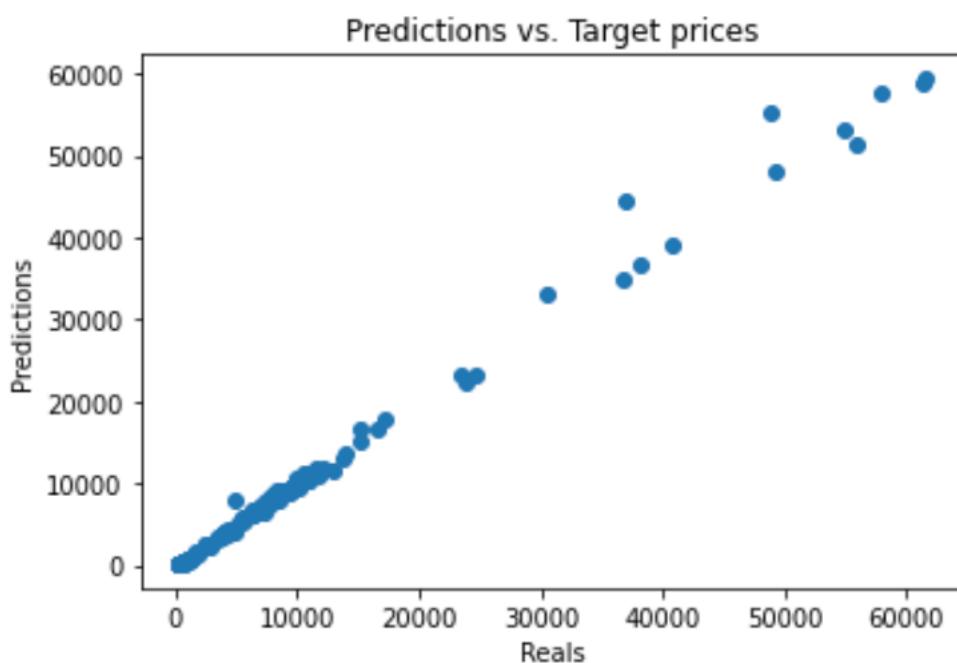
برای تشخیص ویژگی‌های مفیدتر می‌توانیم از feature selection مختلف استفاده کنیم. به عنوان مثال، forward selection یا Backward elimination

همچنین، می‌توان از روش‌های dimensionality reduction LDA و PCA استفاده کرد تا ویژگی‌های مستقل و مفید را بیرون کشید تا دچار curse of dimensionality نشویم.

۷ - افزودن لایه‌های مخفی (۲ لایه) تاثیر مثبتی در شبکه نداشته است چه با dropout و چه بدون آن. نتایج این دو حالت در زیر آورده شده است. دلیل این موضوع می‌تواند عدم طولانی بودن sequence و کم بودن ویژگی‌ها باشد که با افزودن لایه‌ی مخفی پیچیدگی شبکه به اشتباه بالا می‌رود و دچار over parametrization می‌شویم.



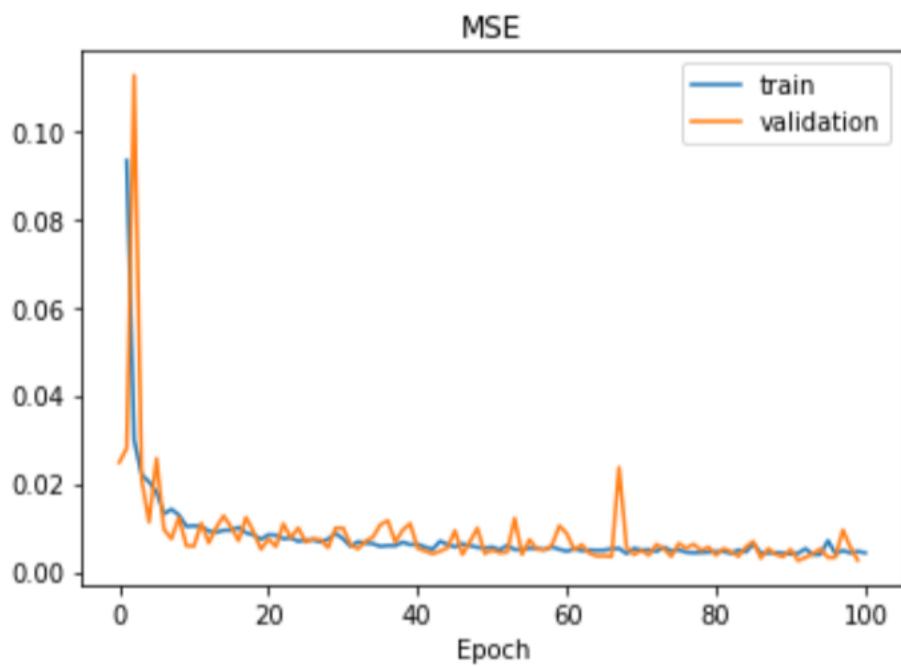
شکل ۳۵ - نمودار خطای بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی با لایه مخفی و بدون dropout



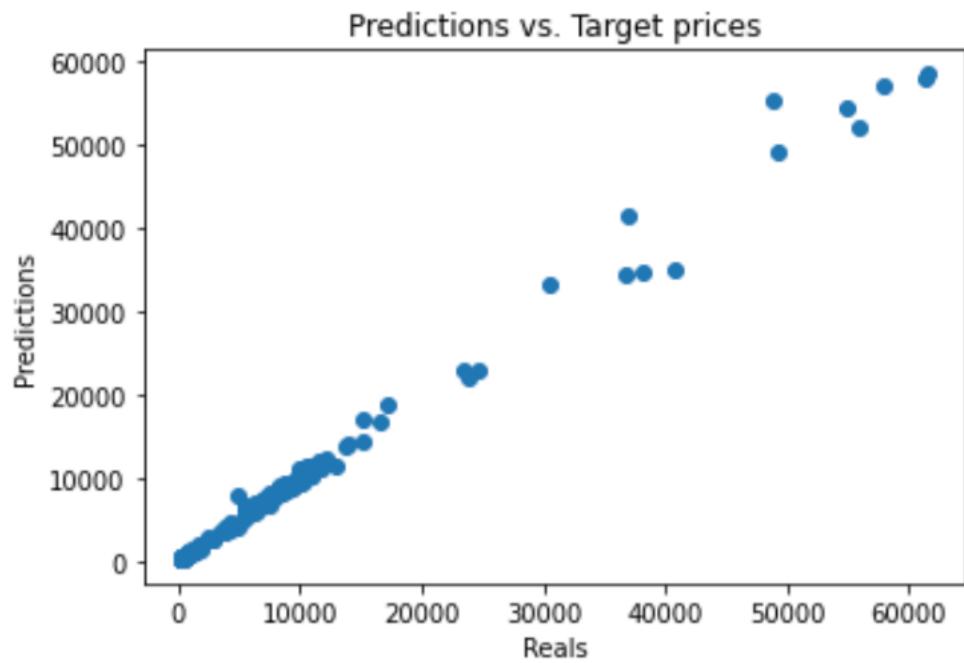
MAE: 367.1653496200119

MSE: 763673.7685463899

شکل ۳۶ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE با لایه مخفی و بدون dropout



شکل ۳۷ - نمودار خطای بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی با لایه مخفی و dropout



MAE: 399.8362636328733

MSE: 835083.9383313429

شکل ۳۸ - نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE با لایه مخفی و dropout

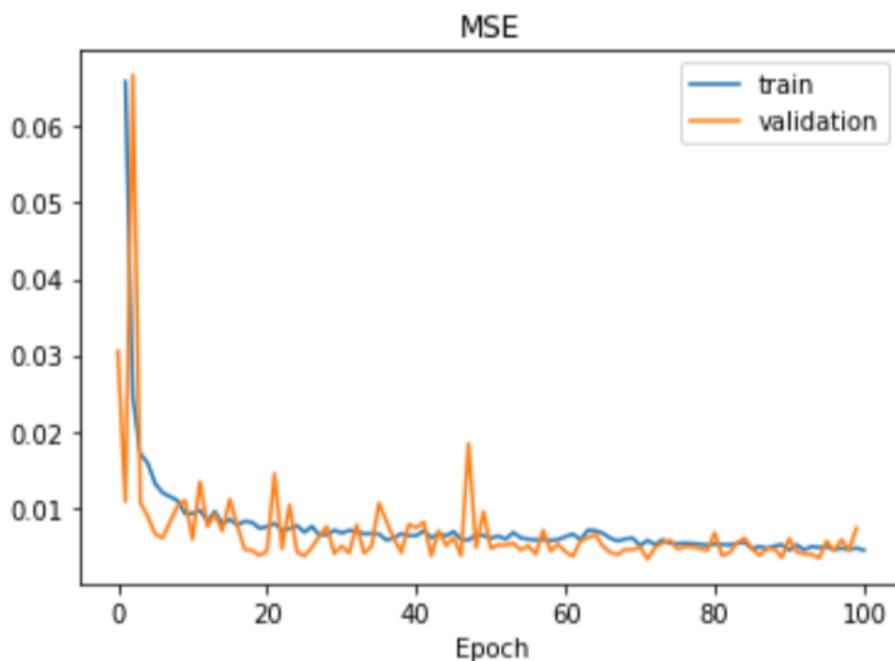
- ۸

بین این سوال و سوال ۹، سوال ۹ برای پاسخگویی انتخاب شده است.

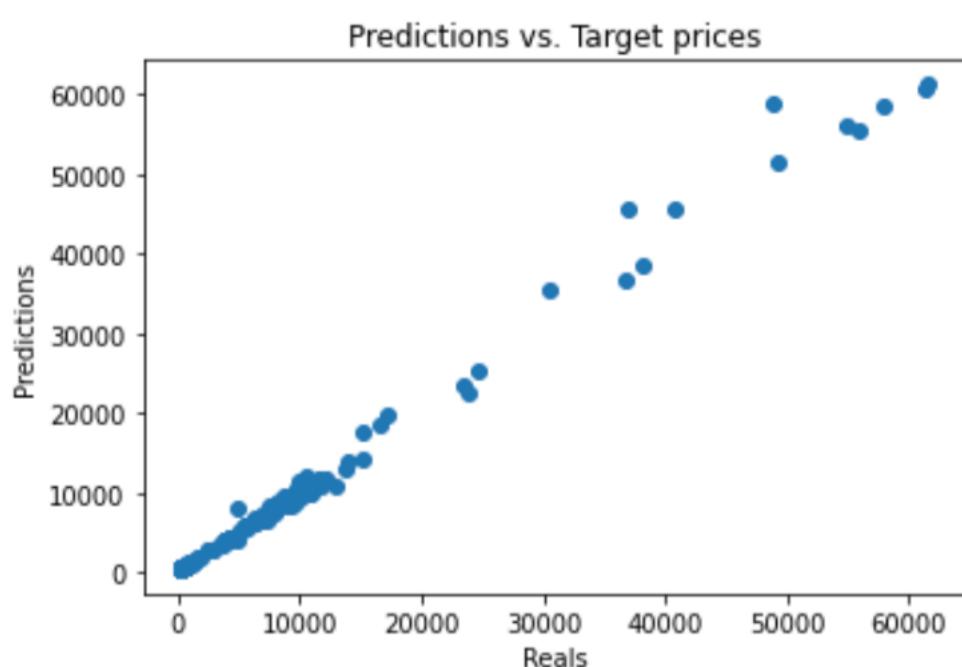
- ۹

در این سوال شبکه‌ی CNN-LSTM پیاده‌سازی شده است که ترکیب شبکه‌های کانولوشنی با شبکه‌های بازگشتی است. پیش از دادن داده به LSTM یک لایه conv و یک pool متناسب با سایز داده روی داده زده شده است. با این کار شبکه CNN ویژگی‌ها را بهتر استخراج می‌کند و به شبکه‌ی LSTM می‌دهد تا رابطه‌ی زمانی را استخراج کند.

همانطور که در نتایج زیر می‌بینید، این ساختار نتوانسته است بهبودی در عملکرد شبکه ایجاد کند. دلیل آن می‌تواند پیچیدگی کم شبکه (under fitting) و میزان کم داده‌ها باشد. یعنی اگر تعداد لایه‌ها کانولوشنی کمی بیشتر شود ویژگی‌ها می‌توانند بهتر استخراج شوند. همچنین، تعداد بیشتر داده‌ها نیز می‌تواند استخراج ویژگی با استفاده از CNN را مفیدتر کنند. تعداد داده‌های تست بیشتر در نظر گرفته شده است. یک پیشنهاد دیگر می‌تواند آموزش شبکه برای تعداد بیشتر epoch باشد تا دقیق آن بتواند کمی بالاتر برود. چراکه شبکه پیچیده‌تر شده است و نیاز به آموزش بیشتری دارد.



شکل ۳۹ - نمودار خطاب بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی CNN-LSTM



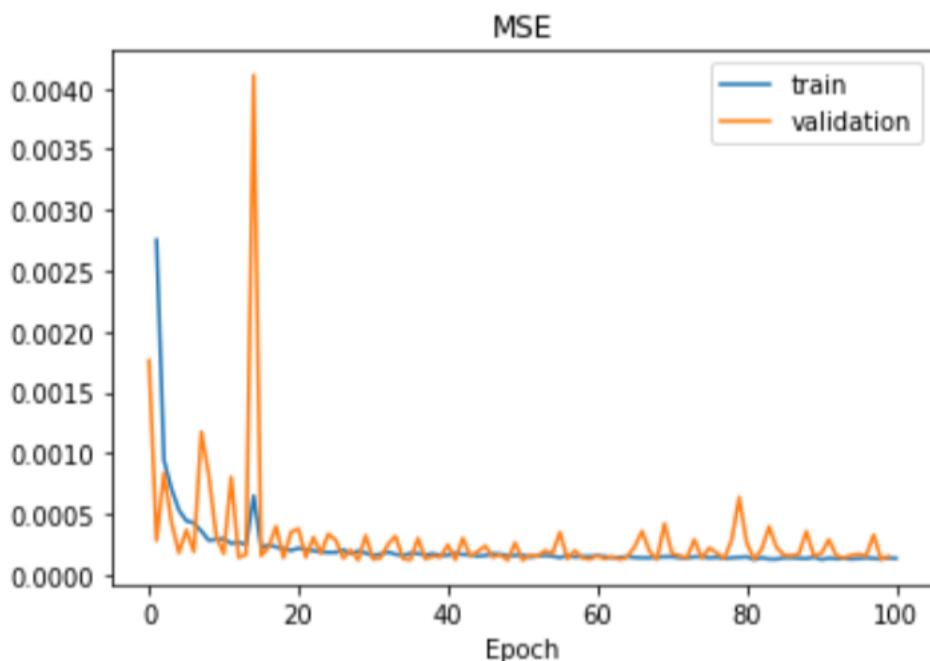
MAE: 437.55651697182554

MSE: 1226336.2782341596

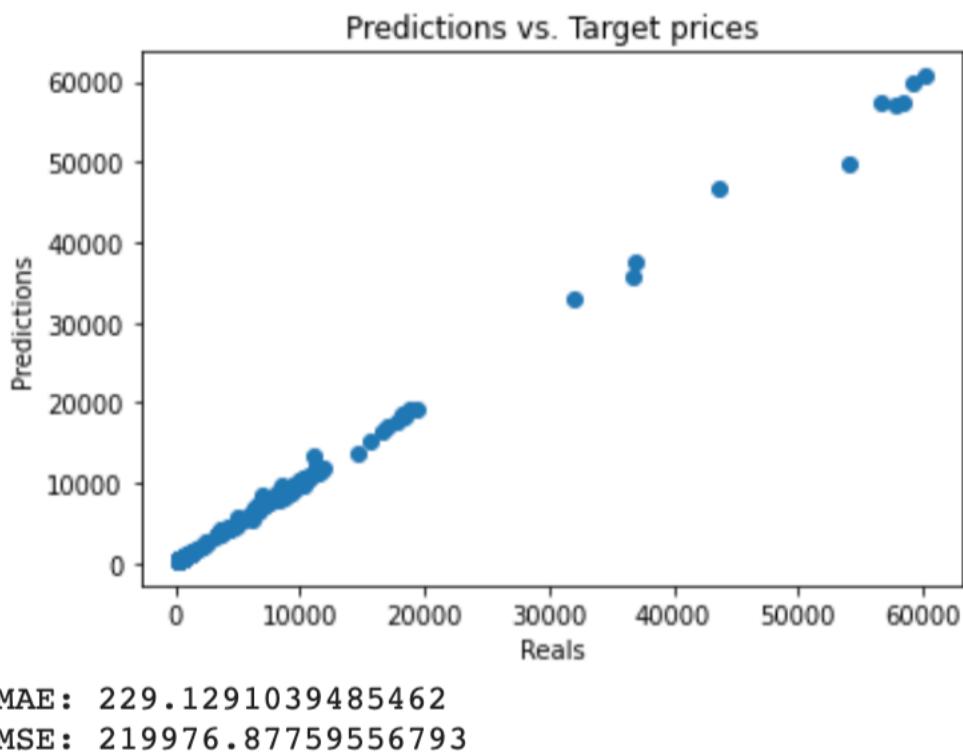
شکل ۴۰- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE CNN-LSTM

در روند یادگیری از standardization استفاده شد. دلیل استفاده از آن نرمال کردن توزیع همه ویژگی‌ها بود تا آن‌ها هم مقیاس شوند. با این کار شبکه‌ی ما قادر تعمیم دهنده بیشتری پیدا می‌کند چراکه وزن‌های ویژگی‌های مختلف بسیار بزرگ و کوچک نمی‌شوند که شبکه دیگر پیش نرود و در یک نقطه گیرکند.

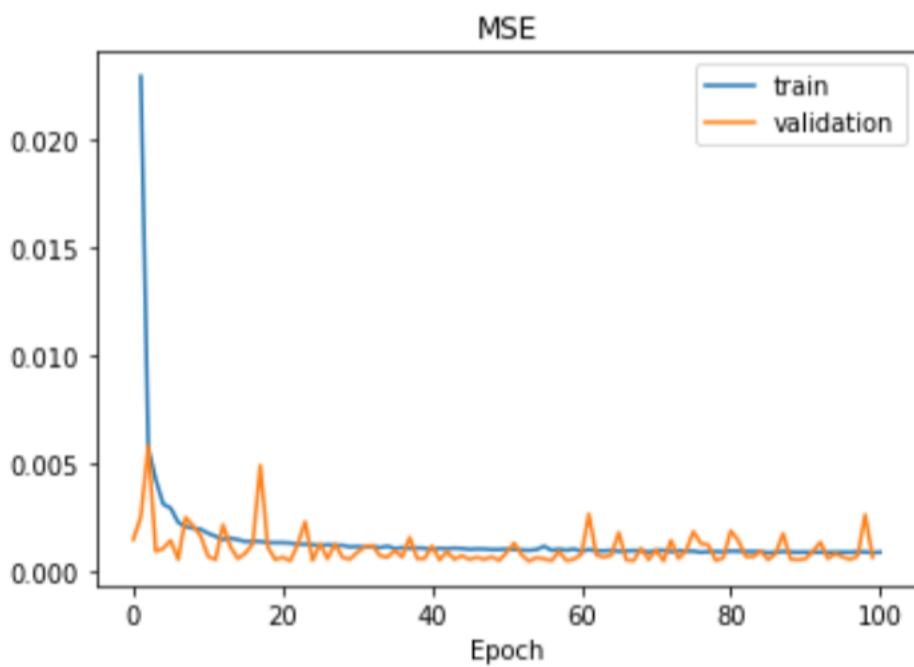
از دو تبدیل power transform و normalization نیز استفاده شد که نتایج آن در زیر قابل مشاهده است.



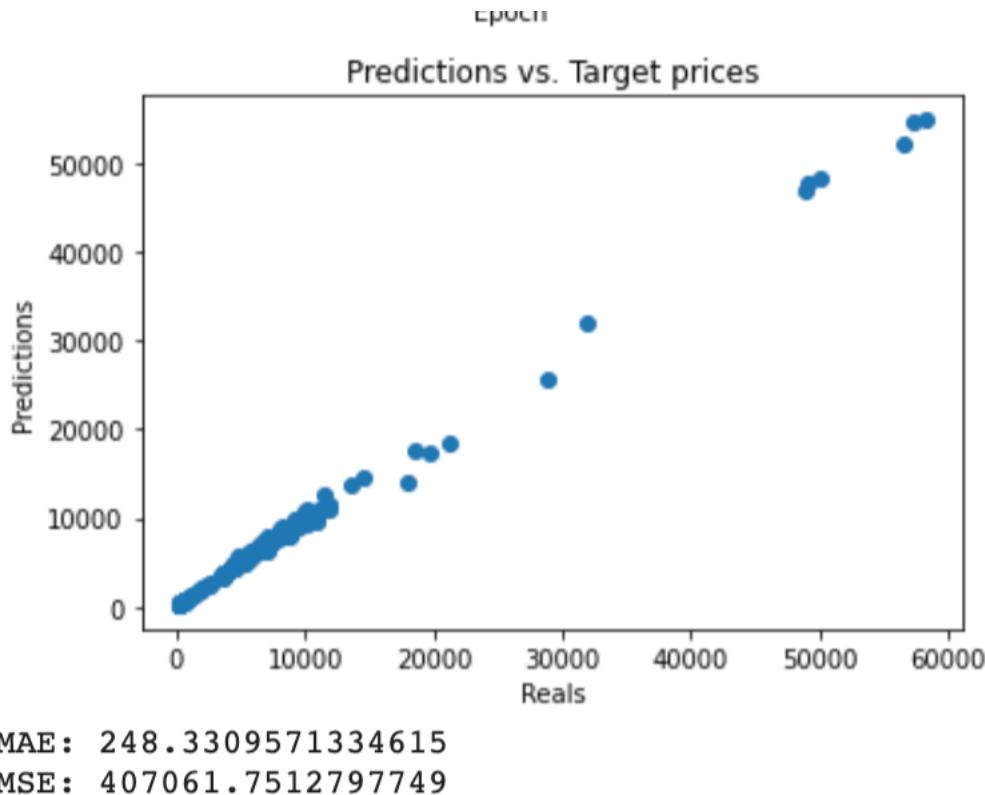
شکل ۴۱ - نمودار خطاب بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای Normalization



شکل ۴۲- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE برای Normalization



شکل ۴۳ - نمودار خطای بر حسب epoch برای داده‌ی آموزش و اعتبار سنجی برای power transform



شکل ۴۴- نمودار مقادیر پیش‌بینی شده و هدف برای داده تست به همراه MAE و MSE را power transform

نتایج با استفاده از این دو تبدیل بهتر شد. بهترین نتیجه با استفاده از **normalization** به دست آمد. دلیل این موضوع نیز بازه بسیار بزرگ قیمت‌ها در بیت کوین است که با MinMaxScaler می‌توان داده‌ها بزرگ را نیز کاهش جدی داد. همچنین، داده‌های بیت کوین چندان از توزیع نرمال پیروی نمی‌کنند که بتوان آن‌ها را standard کرد و به توزیع نرمال واحد رساند.

شبکه‌ی برنده:

LSTM شبکه:

MSE تابع هزینه:

RMSProp تابع فعالساز:

میزان dropout: صفر

تعداد ویژگی‌ها: ۲

تعداد لایه‌های بازگشتی: ۲

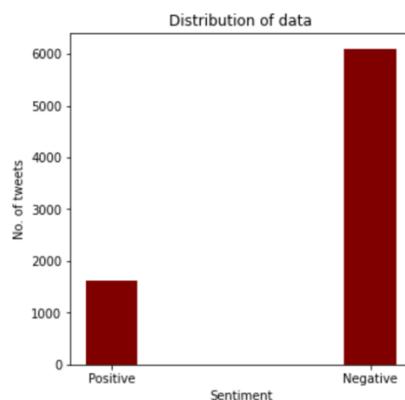
تبديل مورد استفاده: normalization

برای پیش‌بینی ۷ روز، حداقل به ۳۵ روز داده نیاز داریم. پس نیاز است که شبکه ما این ۲۸ روز متوالی را پیش‌بینی کرده باشد و ما از این پیش‌بینی‌ها برای ۷ روز بعدی استفاده کنیم. نتایج به دست آمده را در زیر می‌توانید ببینید.

```
[array([59110.43180576]),  
 array([58675.23365846]),  
 array([58334.32215226]),  
 array([58073.7316918]),  
 array([57885.83781693]),  
 array([57767.27368681]),  
 array([57716.96734763])]
```

سوال ۳ – آشنایی با کاربرد شبکه‌های عصبی بازگشتی در متن

۱ – برای اینکه بتوانیم تعداد رکوردهای داده موجود در کلاس‌های مختلف را متوازن کنیم راهکارهای مختلفی از جمله کاهش و افزایش تعداد نمونه‌ها داریم. در روش کاهش تعداد، با کم کردن تعداد رکوردهای موجود در کلاس‌هایی که تعداد داده بیشتری دارند تعداد کلاس‌هارا متوازن می‌کنیم. این روش در حالتی که تعداد نمونه‌ها کم باشد باعث از دست رفتن داده‌ها و ضعف عملکرد می‌شود. در روش افزایش تعداد باید تعداد رکوردهای موجود در کلاسی با تعداد کمتر را افزایش دهیم تا تعداد داده‌های کلاس‌ها متوازن شود. پس نیاز است که از داده‌های موجود داده‌های جدید تولید کنیم. ساده‌ترین روش این کار تکرار رندوم برخی داده‌های کلاس می‌باشد (البته باید دقیق کنیک‌ها را فقط بر روی داده‌های آموزش انجام دهیم)، روش دیگر برای افزایش داده‌های کلاس اعمال تغییرات بر روی برخی از داده‌های موجود می‌باشد. تغییراتی چون تغییر برخی کاراکترها، جایگزینی برخی از کلمات با کلمات هم‌معنی آنها، جایه‌جایی چینش جملات، ترکیب یک یا چند داده و ... که هر کدام وابسته به نوع مسئله قابل استفاده هستند.



شکل ۴۵ – توزیع داده‌های آموزش

در اینجا ما از روش جایگزینی تعدادی از کلمات با مترادف‌شان استفاده کردیم. همچنین جهت بهبود عملکرد مدل به تابع LOSS وزن دادیم، تا برای کلاس با تعداد کمتر وزن کمتری نیز در نظر بگیرد.

– ۲

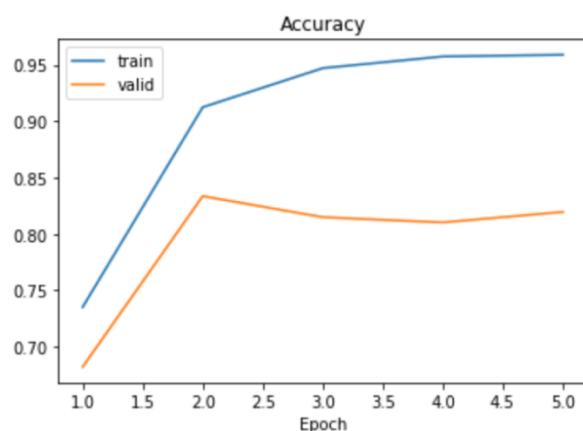
پیش‌پردازش‌های استفاده شده در سوال:

- تبدیل حروف به حروف کوچک، برای اینکه مدل دو کلمه با حروف یکسان ولی متفاوت در کوچک یا بزرگ بودن را یکسان تشخیص دهد.
- حذف URL‌ها، \$، RT، @ و علائم نگارشی چون احساس جمله را تغییری نمی‌دهند.

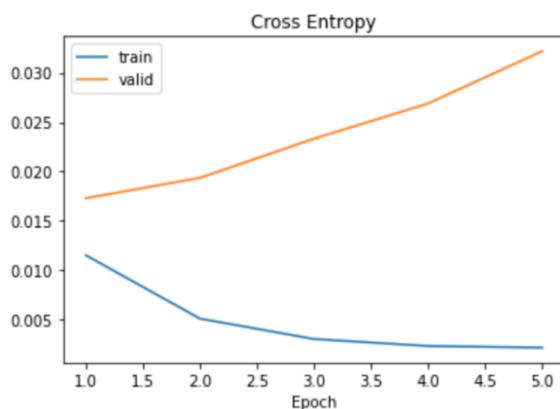
- تبدیل ایموجی‌های موجود به کلمه معادل آنها جهت افزودن بار معنایی آن به متن
- Lemmatize کردن کلمات، با این کار کلمات با مصدر یکسان و اما تفاوت‌های ساختاری متفاوت یکسان می‌شوند و مدل متوجه شباهت آنها می‌شود.
- تصحیح اشتباهات نوشتاری

۳ - برای این بخش از کلمات تعبیه وابسته به متن Bert استفاده شد.

۴ و ۵ - برای این بخش از شبکه‌های عمیق متفاوت با پارامترهای متنوع استفاده شد و در زیر بهترین شبکه را می‌بینیم. در نمودار مربوط به loss افزایش این مقدار برای داده ارزیابی به دلایلی چون کم بودن داده و همینطور کم بود ایپاک‌های آموزش است. همینطور که می‌بینیم دقت مدل هم برای داده‌های آموزش و هم داده‌های اعتبارسنجی بالای ۸۰ درصد است و در ابتدا روندی صعودی داشته است.



شکل ۴۶ - نمودار دقت برای داده‌های آموزش و ارزیابی

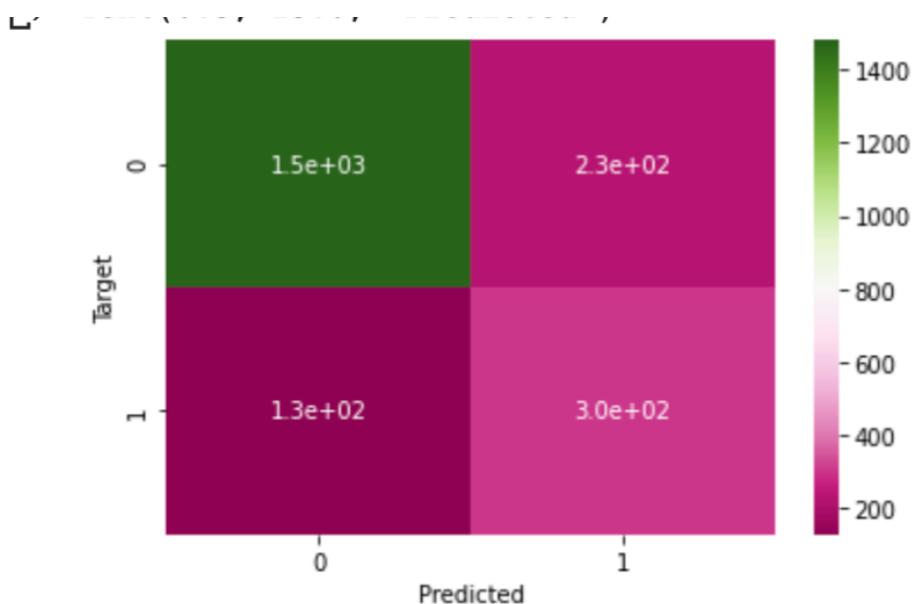


شکل ۴۷ - نمودار loss برای داده‌های آموزش و ارزیابی

پیش از برقراری توازن در داده‌های آموزش معیار $f1$ مدل برای داده آزمون، درصد بود و اما با استفاده از تکنیک‌های مربوط به بالانس داده این مقدار افزایش یافت همانطور که می‌بینیم تعداد داده‌هایی از کلاس ۰ که به درستی تشخیص داده شده اند بسیار زیاد است. کم بودن تعداد کلاس‌هایی که به درستی ۱ تشخیص داده شده اند نسبت به کلاس ۰ به علت عدم توازن اولیه داده‌هاست.

	precision	recall	f1-score	support
0	0.92	0.86	0.89	1713
1	0.57	0.70	0.63	433
accuracy			0.83	2146
macro avg	0.74	0.78	0.76	2146
weighted avg	0.85	0.83	0.84	2146

شکل ۴۸ – ارزیابی مدل بر روی داده آزمون



شکل ۴۹ – ماتریس آشتفتگی خروجی مدل برای داده آزمون

سوال ۴ – آشنایی با مقالات مرتبط

مقالات انتخابی در این بخش:

- **Understanding and Controlling Memory in Recurrent Neural Network**
- **A long video caption generation algorithm for big video data retrieval**