تمرین سری چهارم (امتیازی)

سید سروش مجد

۴۰۰۴۴۳۱۸۱

درس یادگیری عمیق | بهار ۱۴۰۱

استاد درس: جناب آقای دکتر حامد ملک

تیر ماه ۱۴۰۱

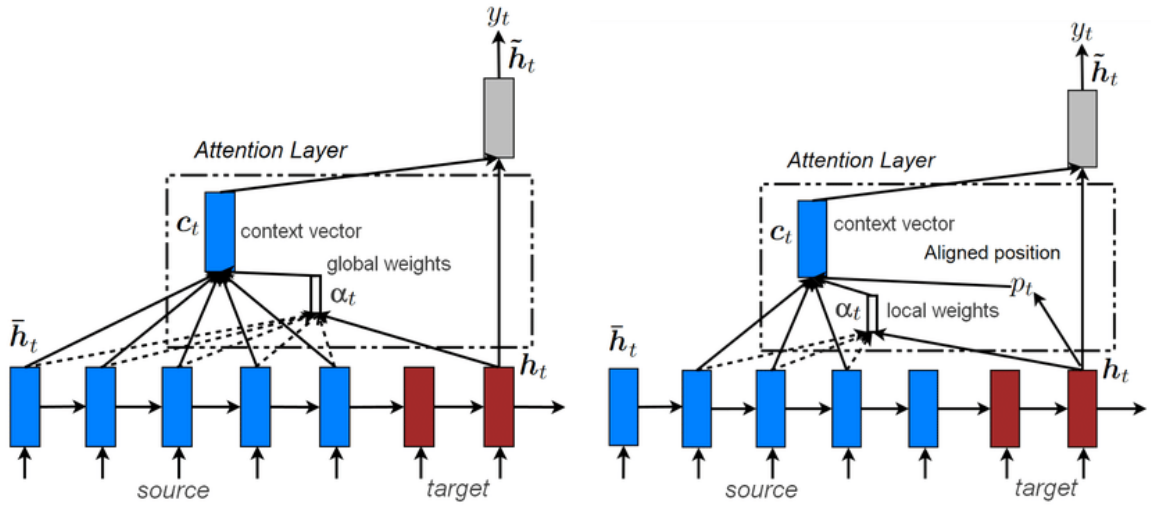
بخش اول سوالات:

۱- تفاوت دو حالت Attention Soft و Attention Hard را توضیح دهید.

در Soft Attention بردار Context با میانگین وزن‌داری از hidden state ها محاسبه می‌شود ولی در Hard Attention به جای این کار از Attention score برای انتخاب یکی از hidden state ها استفاده می‌شود. در Soft attention برخلاف Hard Attention می‌توان از GD استفاده کرد .

۲- تفاوت attention global و attention local را به طور کامل و با رسم شکل توضیح دهید.

در global attention مکانیزم attention بر روی کل دنباله ورودی از انکودر اعمل می‌شود ولی در local attention بر روی زیر مجموعه‌ای از ورودی اعمال و بردار context از آن محاسبه می‌شود. local attention دارای receptive field کمتر و global attention به دلیل تعداد وزن‌های بیشتر دارای پیچیدگی محاسباتی بالاتری است. مشکل receptive field کمتر در local attention می‌تواند با stack کردن لایه‌ها تا حدی برطرف شود و receptive field افزایش یابد. در شکل زیر در حالت global a_t وزن‌های attention هستند که از هر encoder step و decoder step قبلی محاسبه می‌شود. سپس با ضرب a_t ها context vector به دست می‌آید. ولی در local attention ابتدا p_t (Aligned position) پیدا شده و سپس با پنجره‌ای از کلمات و h_t بردار context به دست می‌آید.

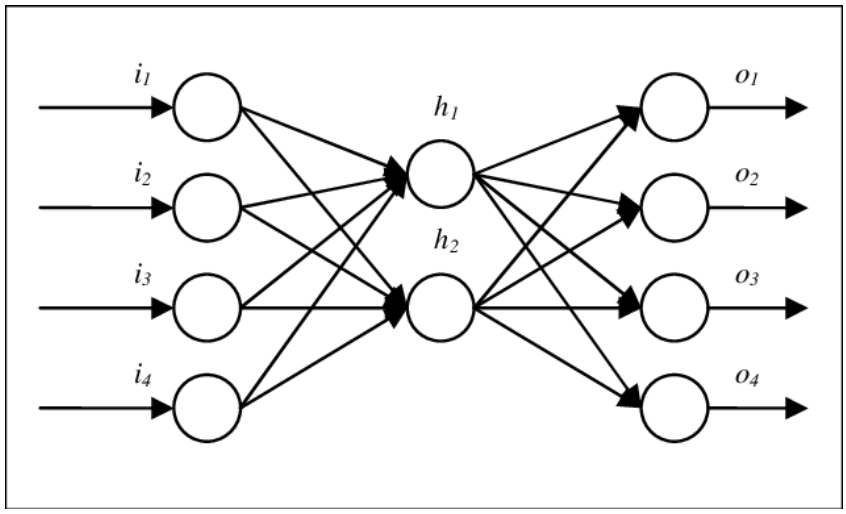


۳- توضیح دهید چرا auto encoder یک روش خود تظارتی است.

برای آموزش auto encoderها لازم نیست کار زیادی انجام شود و فقط دیتای خام وارد شبکه می‌شود و به لیبل برای دیتا نیز احتیاجی نیست. auto encoder ها لیبل‌ها را خودشان تولید می‌کنند و از این جهت خود نظارتی هستند. در واقع auto encoderها یک representation فشرده‌ای از دیتای ورودی یاد می‌گیرند.

۴- آیا auto encoder یک روش کاهش بعد است؟چرا؟

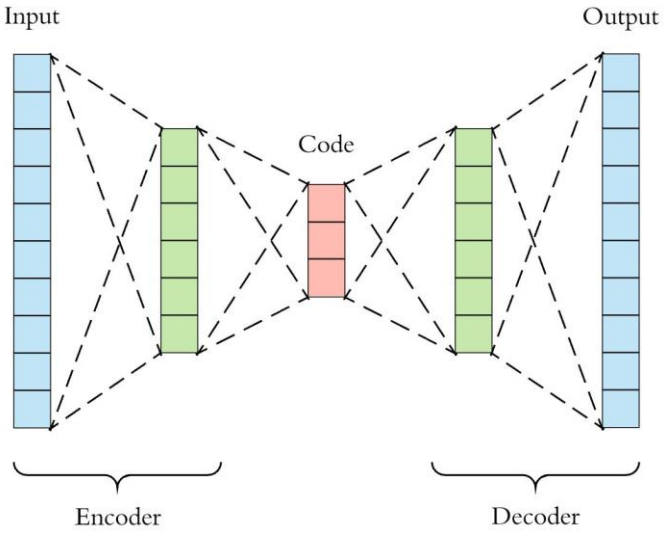
بله. در auto encoder دیتا توسط encoder به بعد کوچکتر encode می‌شود که سایز آن برابر با سایز لایه Bottleneck می‌باشد. این عملیات می‌تواند Feature extraction نیز نام‌گذاری شود. برای مثال در شکل زیر دیتا از ۴ بعد به ۲ بعد در لایه Bottleneck توسط encoder کاهش بعد داریم. البته ابعاد خروجی و ورودی در auto encoder یکسان است و از این لحاظ کاهش بعد نداریم.



۵- معماری قسمت encoder و decoder در یک autoencoder از چه جهات شباهت دارد؟ یکسانی چه پارامترها یا ویژگی‌هایی در این دو قسمت الزامی است؟

encoder و decoder هر دو لایه‌های Fully connected و Feed forward هستند. معمولا decoder مانند معکوس لایه encoder در نظر گرفته می‌شود اما این یک ضرورت نیست. ولی تساوی ابعاد لایه ورودی در encoder و خروجی در decoder ضروری می‌باشد. شکل زیر

معماری decoder و encoder با تعداد لایه‌ها و نورون‌های برابر در هرکدام از لایه‌های decoder و encoder را نشان می‌دهد.



۶- مدل‌های auto encoder در چه تسک‌هایی کاربرد دارند؟ مختصر توضیح دهید.

۱- کاهش ابعاد : در سوال ۴ توضیح داده شد.

۲- فشرده‌سازی تصویر: تصویر توسط انکودر فشرده شده و توسط decoder بازسازی می‌شود و وزن‌های شبکه می‌تواند با بازسازی تصاویر توسط decoder از دیتای فشرده شده توسط encoder اموزش داده شود.

۳- حذف نویز تصویر : ورودی تصویر با نویز و خروجی تصویر بدون نویز

۴- پیشبینی seq2seq

۵- سیستم recommendation

۶- جست و جوی تصویر: مجموعه داده تصاویر و تصویری که می‌خواهیم جست و جو کنیم توسط دوتا encoder فشرده شده و با یکدیگر مقایسه می‌شوند.

۷- anomaly detection: برای مثال شبکه با دیتای سمپل یک کلاس آموزش ببیند و یک داده از کلاسی دیگر loss بیشتری نتیجه دهد.

۸- جایگذاری missing value: به صورت رندوم داده missing در دیتای ورودی قرار دهیم و شبکه باید تلاش کند داده اصلی را بازسازی کند.

بخش دوم پیاده‌سازی:

۱) در این بخش مدلی برای پیشبینی داده‌های بورس با استفاده از واحدهای LSTM به صورت پشته‌ای برای بازه زمانی ۳۰ روزه پیاده‌سازی شد. الگوریتم بهینه‌سازی، تابع فعالیت و تابع زیان به ترتیب از الگوریتم ،Adam تابع tanhو تابع MSE استفاده شد. در نهایت با این مدل، قیمت Closeدر مجموعه داده را پیشبینی کردم. برای رسیدن به بهترین مدل، عملکرد سه شبکه با تعداد واحدهای LSTMمختلف را مقایسه کردیم. مجموعه داده google stock دارای ۶ فیچر است.

Date :تاریخ ثبت شدن هر رکورد

Open : قیمت در زمان باز شدن بازار سهام

High : بالاترین قیمت در آن تاریخ

Low : پایین ترین قیمت در آن تاریخ

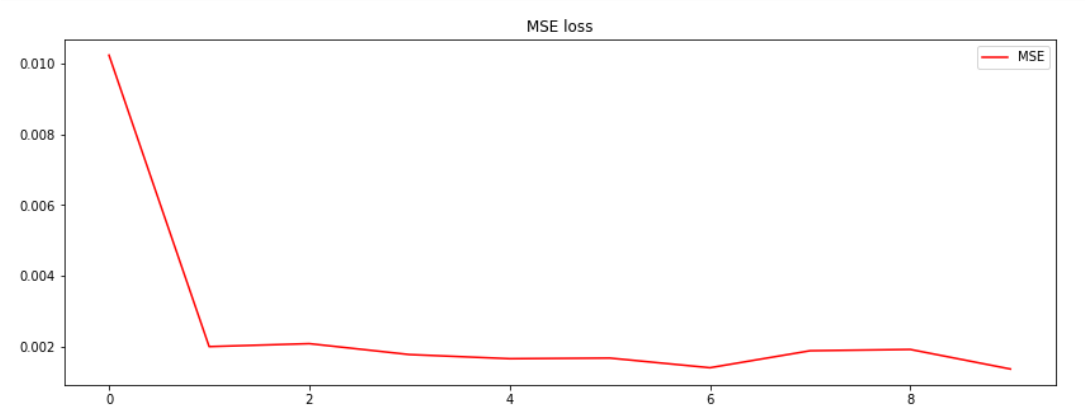
Volume : فروش کل سهام در آن تاریخ

برای پیاده‌سازی از کتاب‌خانه sklearn استفاده کردیم. و قبل از اینکه داده‌ها را به شبکه بدهیم با minmax scaler مقدار فیچرهای open، high، low و volume را نرمالایز کردیم. در شبکه‌ها افزایش تعداد لایه‌ها و نورون‌ها باعث می‌شود شبکه پیچیده‌تر شده و بتواند پترن‌های پیچیده‌تری یاد بگیرد و برای جلوگیری از بیش‌برازش و فیت نشدن شبکه بر روی اطلاعات outlier از drop out استفاده کردیم تا تعمیم‌پذیری را نیز افزایش دهیم. البته از جایی به بعد مشاهده کردیم با پیچیده‌تر شدن شبکه دقت آنچنان بهتر نمی‌شود. تلاش شد تا با صحیح و خطا تلاش کردیم بهترین پارامترها را برای شبکه‌ها با تعداد لایه‌های ۲، ۳ و ۴ به دست بیاوریم. سپس ۱۵۰ نمونه آخر (۵ ماه آخر) را برای prediction به شبکه دادیم و نتیجه را نمایش دادیم و مقدار MSE برای آن‌ها را نیز محاسبه کردیم.

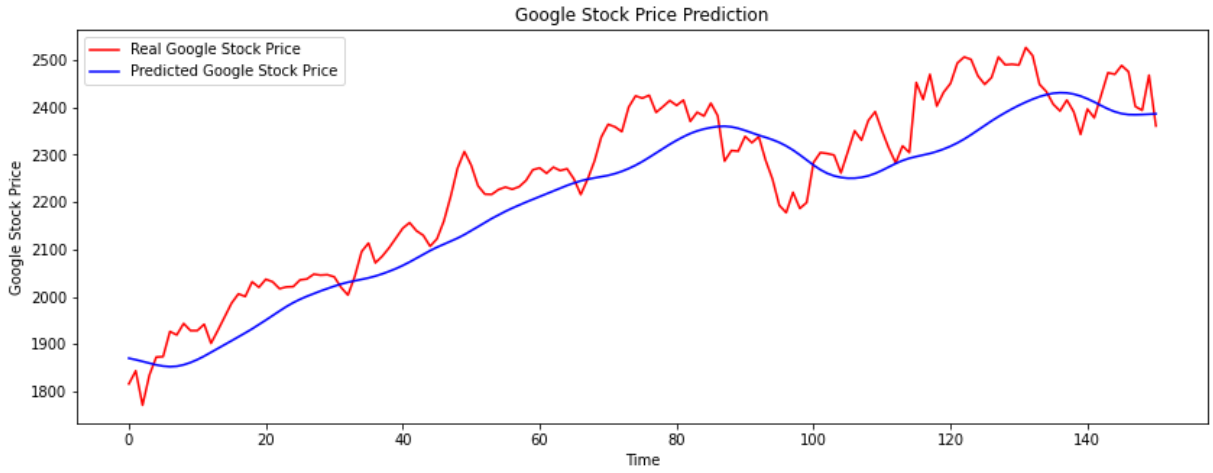
مدل اول **LSTM** با ۴ لایه به تعداد یونیت ها از راست به چپ برابر با **[۶۰ ۷۰ ۸۰ ۱۰۰]**

```
Model: "sequential_85"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_233 (LSTM)             (None, 30, 60)            15840

 dropout_283 (Dropout)       (None, 30, 60)            0

 lstm_234 (LSTM)             (None, 30, 70)            36680

 dropout_284 (Dropout)       (None, 30, 70)            0

 lstm_235 (LSTM)             (None, 30, 80)            48320

 dropout_285 (Dropout)       (None, 30, 80)            0

 lstm_236 (LSTM)             (None, 100)               72400

 dropout_286 (Dropout)       (None, 100)               0

 dense_85 (Dense)            (None, 1)                 101

=================================================================
Total params: 173,341
Trainable params: 173,341
Non-trainable params: 0
_________________________________________________________________
```

نمودار آموزش شبکه (محور افقی ایپاک می‌باشد)



بخش پیشبینی و نمودار واقعی بورس ستون close:

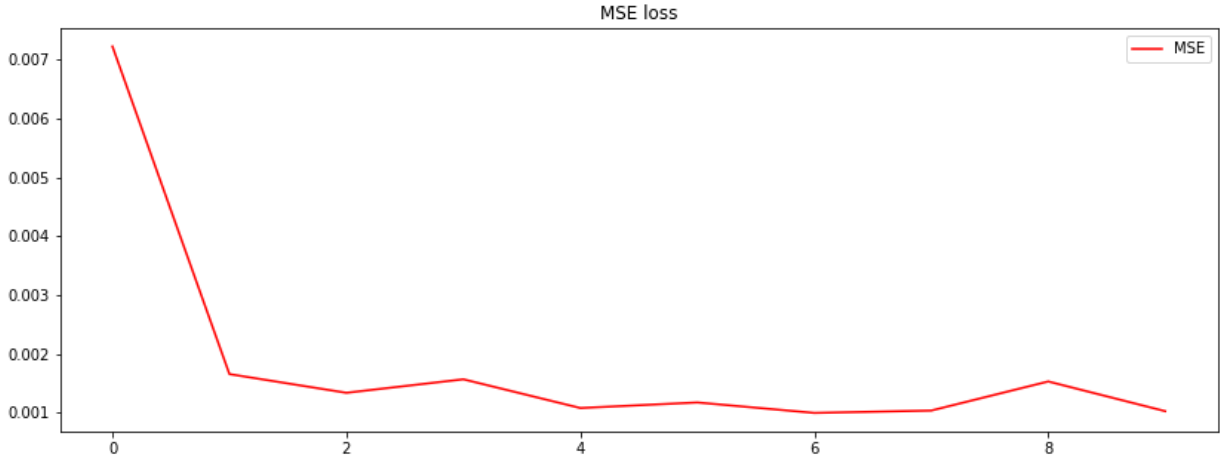


مقدار mse برای داده تست: ۰.۱۸۲

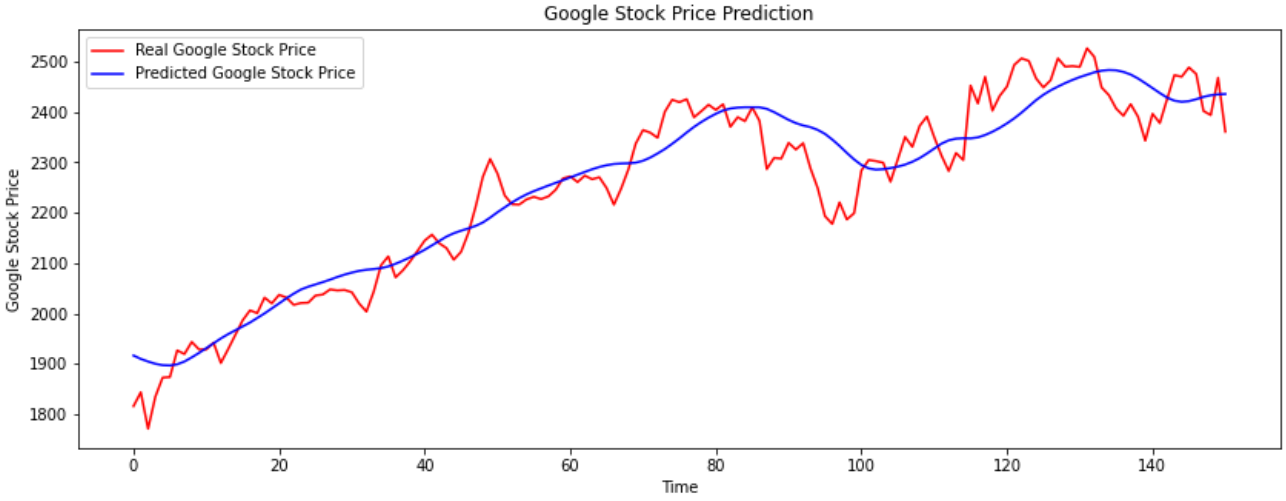
مدل دوم LSTM با ۳ لایه به تعداد یونیت ها از راست به چپ برابر با [۷۰ ۸۰ ۹۰]:

```
Model: "sequential_19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_23 (LSTM)              (None, 30, 70)            21280

 dropout_49 (Dropout)        (None, 30, 70)            0

 lstm_24 (LSTM)              (None, 30, 80)            48320

 dropout_50 (Dropout)        (None, 30, 80)            0

 lstm_25 (LSTM)              (None, 90)                61560

 dropout_51 (Dropout)        (None, 90)                0

 dense_19 (Dense)            (None, 1)                 91

=================================================================
Total params: 131,251
Trainable params: 131,251
Non-trainable params: 0
_________________________________________________________________
```

نمودار آموزش شبکه (محور افقی ایپاک می‌باشد)



بخش پیشبینی و نمودار واقعی بورس ستون close:

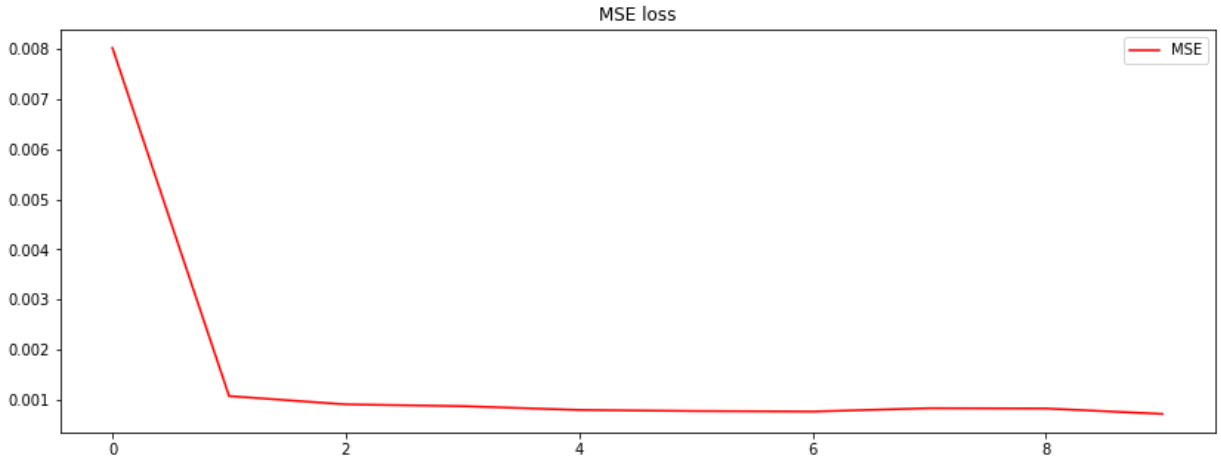


مقدار mse برای داده تست: ۰.۰۱۷۲

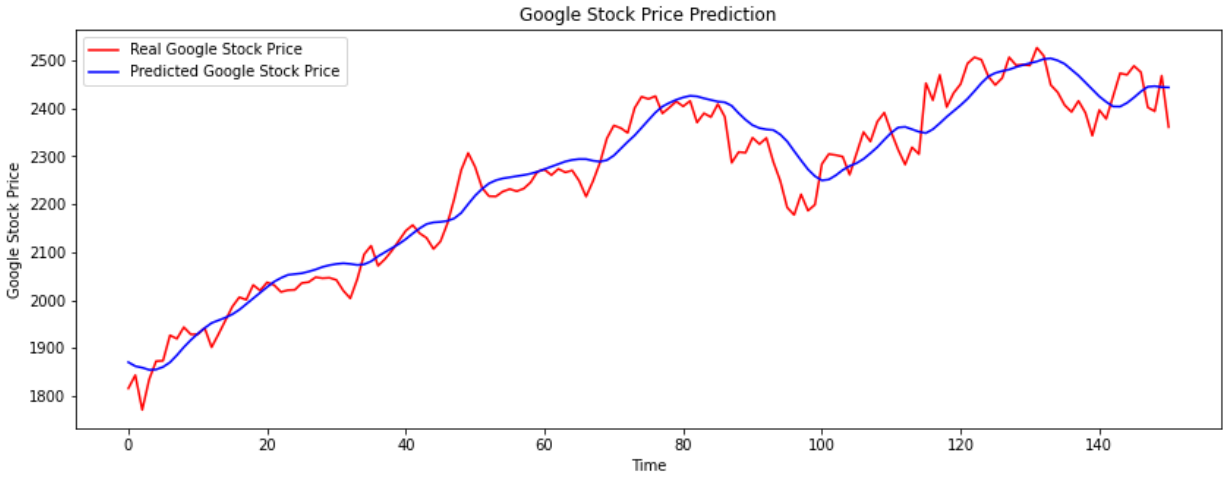
مدل سوم LSTM با ۲ لایه به تعداد یونیت ها از راست به چپ برابر با [۷۰ ۱۰۰]

```
Model: "sequential_31"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_48 (LSTM)              (None, 30, 70)            21280

 dropout_74 (Dropout)        (None, 30, 70)            0

 lstm_49 (LSTM)              (None, 100)               68400

 dropout_75 (Dropout)        (None, 100)               0

 dense_31 (Dense)            (None, 1)                 101

=================================================================
Total params: 89,781
Trainable params: 89,781
Non-trainable params: 0
_________________________________________________________________
```

نمودار آموزش شبکه (محور افقی ایپاک می‌باشد)



بخش پیشبینی و نمودار واقعی بورس ستون close:



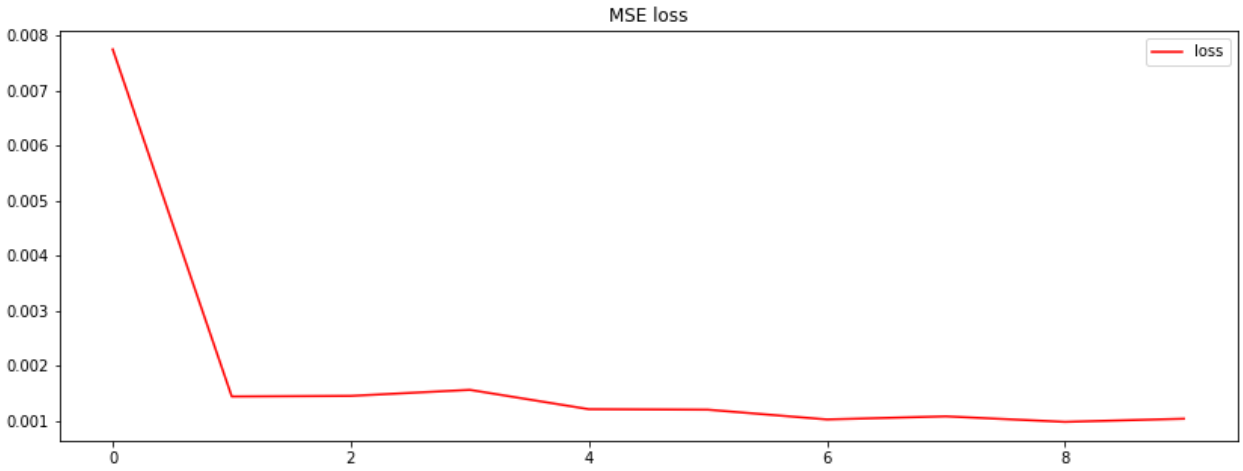
مقدار mse برای داده تست: ۰.۰۱۸۳

۲) بخش (۱) با واحدهای GRU پیاده‌سازی شد.

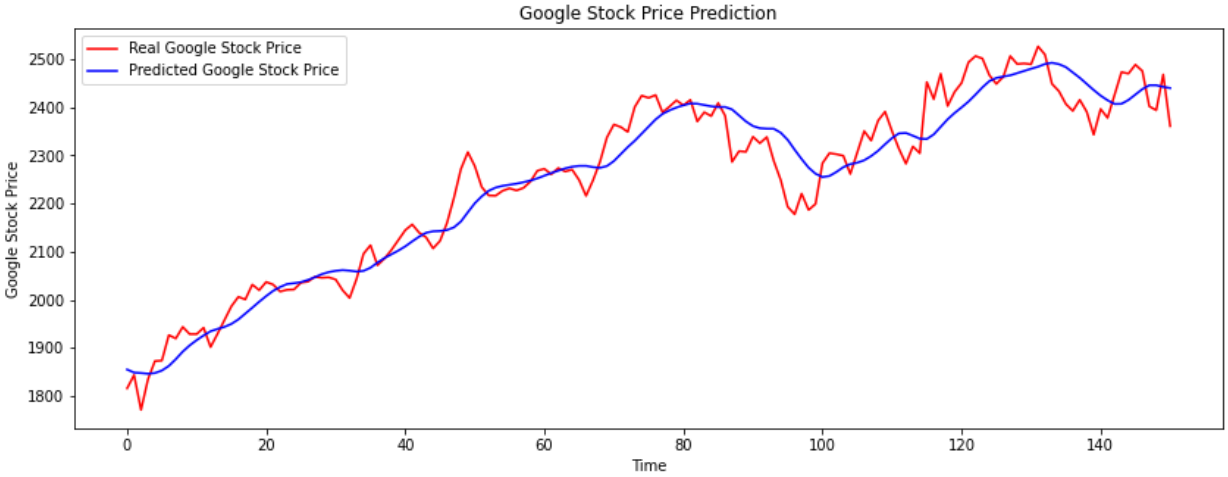
مدل اول GRU با ۴ لایه به تعداد یونیت ها از راست به چپ برابر با [۶۰ ۷۰ ۸۰ ۱۰۰]:

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru_4 (GRU)                 (None, 30, 60)            12060

 dropout_12 (Dropout)        (None, 30, 60)            0

 gru_5 (GRU)                 (None, 30, 70)            27720

 dropout_13 (Dropout)        (None, 30, 70)            0

 gru_6 (GRU)                 (None, 30, 80)            36480

 dropout_14 (Dropout)        (None, 30, 80)            0

 gru_7 (GRU)                 (None, 100)               54600

 dropout_15 (Dropout)        (None, 100)               0

 dense_5 (Dense)             (None, 1)                 101

=================================================================
Total params: 130,961
Trainable params: 130,961
Non-trainable params: 0
_________________________________________________________________
```

نمودار آموزش شبکه (محور افقی ایپاک می‌باشد)



بخش پیشبینی و نمودار واقعی بورس ستون close:

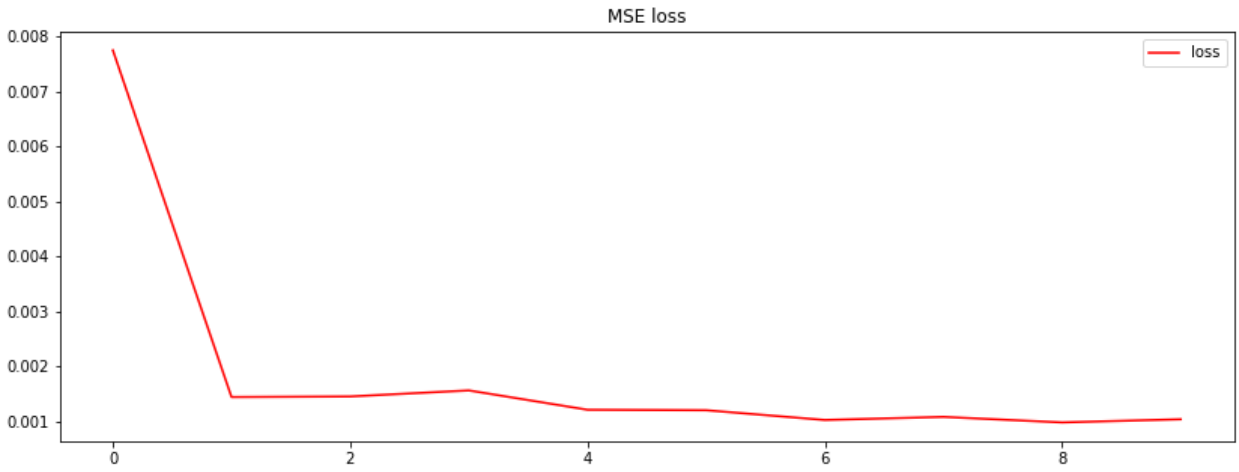


مقدار mse برای داده تست: ۰.۰۱۸۳

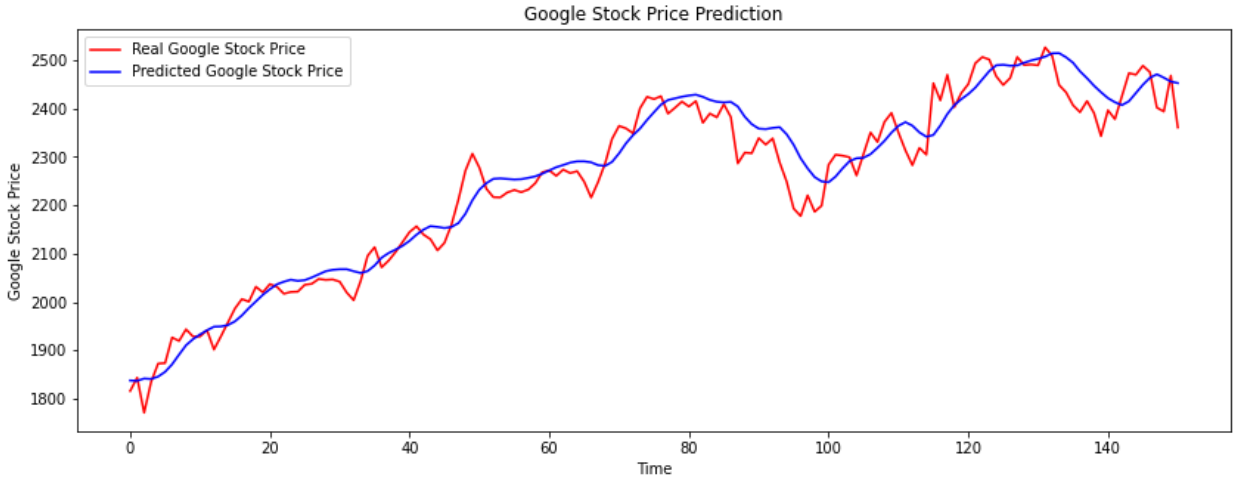
مدل دوم GRU با ۳ لایه به تعداد یونیت ها از راست به چپ برابر با [۷۰ ۸۰ ۹۰]:

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru_8 (GRU)                 (None, 30, 70)            16170

 dropout_16 (Dropout)        (None, 30, 70)            0

 gru_9 (GRU)                 (None, 30, 80)            36480

 dropout_17 (Dropout)        (None, 30, 80)            0

 gru_10 (GRU)                (None, 90)                46440

 dropout_18 (Dropout)        (None, 90)                0

 dense_6 (Dense)             (None, 1)                 91

=================================================================
Total params: 99,181
Trainable params: 99,181
Non-trainable params: 0
_________________________________________________________________
```

نمودار آموزش شبکه (محور افقی ایپاک می‌باشد):



بخش پیشبینی و نمودار واقعی بورس ستون close:

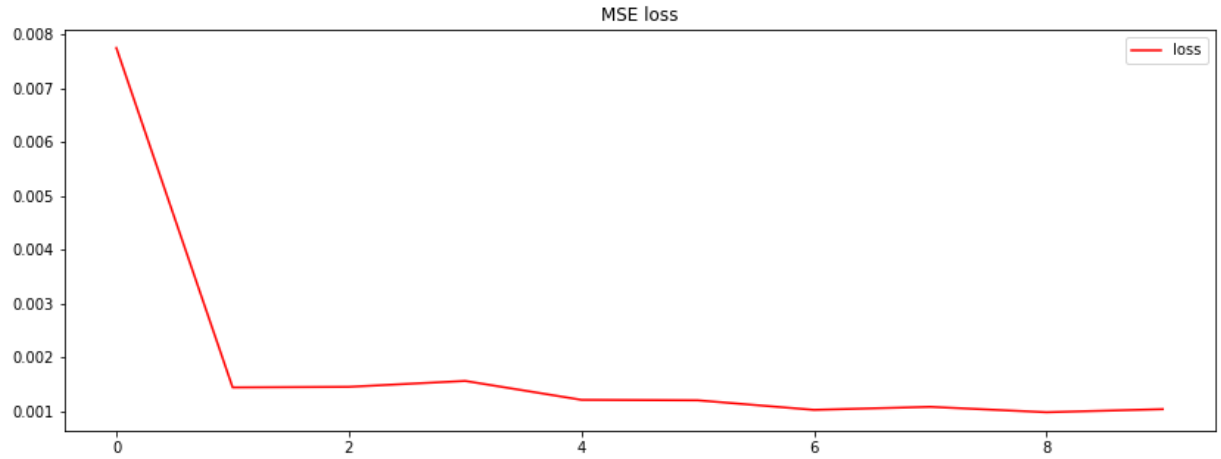


مقدار mse برای داده تست: ۰.۰۱۸۷

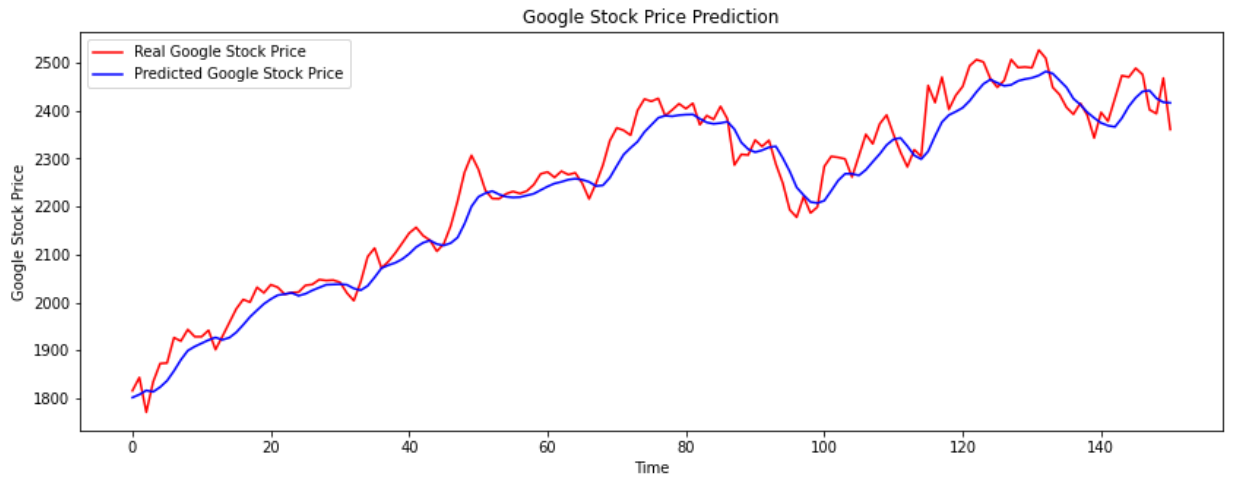
مدل سوم GRU با ۲ لایه به تعداد یونیت ها از راست به چپ برابر با [۱۳۰ ۷۰]

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru_24 (GRU)                (None, 30, 70)            16170

 dropout_32 (Dropout)        (None, 30, 70)            0

 gru_25 (GRU)                (None, 130)               78780

 dropout_33 (Dropout)        (None, 130)               0

 dense_13 (Dense)            (None, 1)                 131

=================================================================
Total params: 95,081
Trainable params: 95,081
Non-trainable params: 0
_________________________________________________________________
```

نمودار آموزش شبکه (محور افقی ایپاک می‌باشد):



بخش پیشبینی و نمودار واقعی بورس ستون close:



مقدار mse برای داده تست: ۰.۰۱۸۴

۳) نتایج هر دو بخش را با هم مقایسه کنید و تحلیل کنید کدام مدل بهتر است و چرا؟

قبل از بررسی نتایج مروری مختصر بر معماری gru و lstm خواهیم داشت.

معماری gru از دروازه‌هایی بنام Update gate وReset gate استفاده می‌کند. با این دو دروازه تصمیم گرفته می‌شود چه اطلاعاتی به خروجی منتقل شود. آنها را می‌توان آموزش داد تا اینطور اطلاعات مربوط به گام های زمانی بسیار قبل را بدون آنکه در گذر زمان (طی گام های زمانی مختلف) تغییر کنند حفظ کند. Reset Gate مانند سویچی کار می‌کند که شبکه با کمک آن می‌تواند مشخص کند چه میزان از اطلاعات گذشته فراموش شود و در گام فعلی از چه میزان از اطلاعات گام قبل استفاده شود. update gate نیز مشخص می‌کند در یک گام زمانی حالت قبلی مورد استفاده قرار گیرد یا ورودی و یا ترکیبی از هر دو باهم. در نتیجه شبکه قادر خواهد بود تا المانهایی را از گذشته دور در حافظه خود نگهداشته و از آن استفاده کند.

در شبکه LSTM سه دروازه وجود دارد که از طریق آن‌ها شبکه جریان داده درون خود را کنترل می‌کند. دروازه Forget gate وظیفه کنترل جریان اطلاعات از گام زمانی قبلی را دارد. این دروازه مشخص میکند آیا اطلاعات حافظه از گام زمانی قبل استفاده شود یا خیر و اگر باید از گام زمانی قبل چیزی وارد شود به چه قدر باشد. دروازه Update gate مشخص می‌کند که گام زمانی فعلی از اطلاعات جدید استفاده شود یا خیر و اگر استفاده می‌شود مقدارش مشخص شود. دروازه Output gate نیز مشخص می‌کند چه میزان از اطلاعات گام زمانی قبل با اطلاعات گام زمانی فعلی به بعد منتقل بشود. مشکل گرادیان کاهشی که در شبکه عصبی بازگشتی وجود داشت نیز در این شبکه‌ها حل شده.

واضح است که LSTM از GRU پیچیده‌تر است و در تسک‌های مختلف ثابت شده است که نسبت به GRU به علت پیچیدگی بیشتر دقت بیشتری نیز دارد. ولی GRU به دلیل داشتن پارامتر کمتر از LSTM حدود ۳۰ درصد سریع‌تر می‌باشد و حجم مموری کمتری نیز اشغال می‌کند. ولی به دلیل اینکه دیتاست ما آنقدر بزرگ نیست و مجموعا ۲۰۱۴ تا داده دارد در نتایجی که در بخش‌های قبل نشان داده شده است GRU عملکرد و دقت بسیار نزدیکی به LSTM داشته است. ولی اگر دیتاست بزرگتر با دنباله بلندتری داشتیم LSTM عملکرد خیلی بهتری نسبت به GRU نشان می‌داد. بهترین شبکه (از نظر کمترین MSE برای prediction) برای LSTM با سه لایه به دست آمد. (MSE = ۱.۰۱۷) ولی شبکه‌های GRU مخصوصا GRU با دو لایه نیز عملکرد کاملا قابل قبولی داشتند و برای دستاست ما بهینه‌تر می‌باشند.