به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

شبکه های عصبی و یادگیری عمیق

Mini Project 2

نام و نام خانوادگی	مسعود رحیمی
شماره دانشجویی	۸۱۰۱۹۸۱۶۱
تاریخ ارسال گزارش	۹۹/۳/۲۰

نام و نام خانوادگی	زهرا قاسمی نژاد
شماره دانشجویی	۸۱۰۱۹۸۲۲۵
تاریخ ارسال گزارش	۹۹/۳/۲۰

فهرست

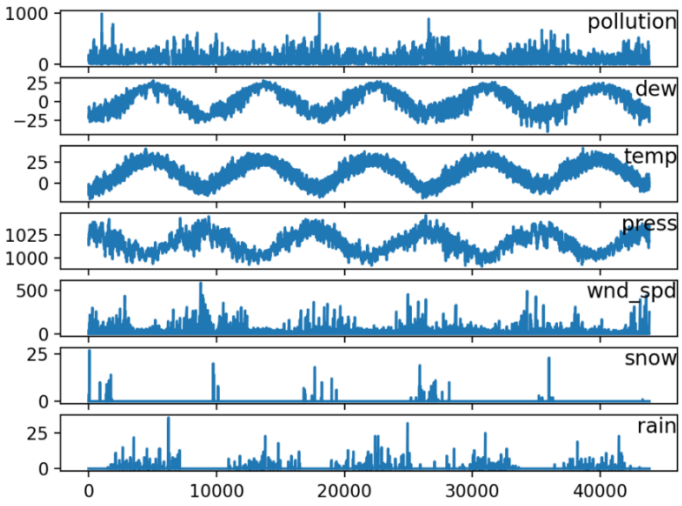
سوال 1

هدف از انجام این پروژه آشنایی با شبکه ی بازگشتی است ازشبکه های بازگشتی مهم میتوان به RNN,LSTM,GRU اشاره کرد در شبکه ها از حافظه برای استفاده از دادهای گذشته در ورودی های جدید استفاده میشود اما طراحی RNN دارای مشکل حافظه کوتاه مدت است اگر طول سری طولانی باشد این شبکه در تشخیص ارتباطات طولانی به مشکل برمیخورد. زیرا دادهای یک مرحله قبل از خود را نگه می دارد. در طول back propagation نیز دارای مشکل gradient vanishing است . درنتیجه اگر در مساله ای نیاز به تشخیص یک ارتباط طولانی بود این شبکه کارکرد مناسبی نخواهد داشت . در شبکه LSTM ,GRU راه حلی برای مشکل حافظه های کوتاه داده شد. این شبکه ها مکانیسمی به نام گیت دارند که جریان داده را تنظیم میکند. این گیت ها میتوانند یادبگیرند که کدام جریان از داده مهم است و نیاز به نگهداری دارد و یا حذف شود. با انجام این کار میتوان داده های مرتبط را از ارتباطات طولانی تر برای پیشبینی استفده کرد.

شبکه GRU نیز مانند شبکه LSTM است با اندکی تغییرات در طراحی گیت ها به این صورت که cell state حذف شده و از hidden state برای جابه جایی داده ها استفاده شده است

با توجه به این که ساختار این شبکه ها شبیه یکدیگر بوده و یک هدف را دنبال میکنند در نتیجه در مراحل آماده سازی داده ها برای شبکه و همچنین فرایند نرمال سازی داده ها مشابه هستند و میتوان از توابع مشترک برای این شبکه ها استفاده کرد درنتیجه بجز مرحله تعریف شبکه و طراحی لایه ها با هم تفاوت ندارند ابتدا این مراحل مشترک را توضیح داده و سپس برای هر کدام از سوالات پرسیده شده توابع را اجرا میکنیم.

در این سوال از دیتاست pollution استفاده کرده ایم در ابتدا میخواهیم با استفاده از داده های ۱۱ ساعت گذشته مقدار آلودگی را برای ساعت ۱۲و۲۴ پیش بینی کنیم در نتیجه به یک سری زمانی ۱۱ ساعته نیاز داریم همچنین در این پروژه قصد داریم برای پیش بینی آب و هوای شهر Beijing از شبکه های مختلف استفاده کنیم و همچنین حالت های مختلف را در نظر بگیریم در این پروژه از مقاله ی ضمیمه شده استفاده کرده ایم که به خوبی در مورداثر این شبکه ها و همچنین شماتیک شبکه ها توضیح داده است در ابتدا دیتا را لود میکنیم و برای درک بهتر داده ها میتوان از مقادیر هر ستون از بردارهای دیتا را که یک مولفه از هوا را نشان میدهد در نمودار زیر کمک گرفت :



چنانچه میبینیم یک الگو با توالی منظم در هر مولفه مشاهده میشود و به نظر میرسد پیش بینی با درک این توالی انجام شدنی خواهد بود .

برای همه ی شبکه ها که در سه بخش اول با آنها مواجه هستیم از ۱۲۰۰۰داده برای آموزش و ارزیابی همچنین ۳۰۰۰داده برای تست استفاده خواهیم کرد.

۱) نمودارتست و ترین و مقدار حقیقی و پیش بینی شده برای شبکه های RNN,GRU,LSTM

با توجه به پیوستگی و ارتباط گزینه ۱ با گزینه ۳ نمودار تست و ترین و همچنین مقدارحقیقی و پیش بینی شده در قسمت ۳ به تفصیل توضیح داده شده است

۲) مقایسه سرعت و دقت شبکه های RNN,GRU,LSTM

شبکه های طراحی شده برای همه ی قسمت ها صورت زیر هستند :

ساختار RNN :

```
Model: "sequential_76"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_rnn_8 (SimpleRNN)     (None, 50)                2950
_________________________________________________________________
dense_77 (Dense)             (None, 1)                 51
=================================================================
Total params: 3,001
Trainable params: 3,001
Non-trainable params: 0
_________________________________________________________________
```

شکل ۱) ساختار شبکه RNN

ساختار LSTM :

```
Model: "sequential_75"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_36 (LSTM)               (None, 50)                11800
_________________________________________________________________
dense_76 (Dense)             (None, 1)                 51
=================================================================
Total params: 11,851
Trainable params: 11,851
Non-trainable params: 0
_________________________________________________________________
```

شکل ۲) ساختار شبکه LSTM

ساختار GRU :

```
Model: "sequential_77"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_33 (GRU)                 (None, 50)                8850
_________________________________________________________________
dense_78 (Dense)             (None, 1)                 51
=================================================================
Total params: 8,901
Trainable params: 8,901
Non-trainable params: 0
_________________________________________________________________
```

شکل ۳) ساختار شبکه GRU

همچنین در مورد سرعت و دقت شبکه ها در قسمت بعد بحث شده است

۳) نحوه ی عملکرد شبکه برای تابع های هزینه متفاوت و روش های بهینه سازی متفاوت

- **تابع خطای MSE**

در این مرحله قصد داریم از تابع خطای mse استفاده کنیم و درضمن روش بهینه سازی را (Adam,adagrad,RMSProp) در نظر میگیریم ابتدا کمی در مورد عملکرد این تابع خطا توضیح میدهیم این تابع به صورت میانگین مجذور فاصله بین خروجی تولید شده توسط شبکه و خروجی واقعی است،همیشه نامنفی است و مقدار بهینه آن صفر است . مشکلی که در استفاده از این تابع ممکن است رخ دهد ،حضور داده های پرت است (outlier) چون حضور داده های پرت مقدار تابع هزینه را خیلی زیاد میکند (به دلیل مجزور شدن ، مقادیر با اندازه های بزرگ، بزرگ تر هم میشوند) و در این حالت MSE خوب کار نمیکند و بهتر است به جای آن از MAE استفاده کرد که مجموع قدر مطلق اختلاف ها است .

در ابتدا به سراغ بهینه ساز adam میرویم و تاثیر این بهینه ساز رابه همراه تابع خطای mse در میزان خطا و زمان آموزش بررسی میکنیم .

- **بهینه ساز Adam :**

Adam

Adam stands for adaptive moment estimation, and is another way of using past gradients to calculate current gradients. Adam also utilizes the concept of momentum by adding fractions of previous gradients to the current one. This optimizer has become pretty widespread, and is practically accepted for use in training neural nets.

It's easy to get lost in the complexity of some of these new optimizers. Just remember that they all have the same goal: minimizing our loss function. Even the most complex ways of doing that are simple at their core.

ین بهینه ساز ایده های هیوریستیک Momentum و RMSProp را ترکیب میکند Momentum جستجو را در جهت کمینه سرعت میبخشد ، در حالیکه RMSProp مانع از جستجو در جهت نوسانات میشود همچنین معادلات این بهینه ساز به شکل زیر است :

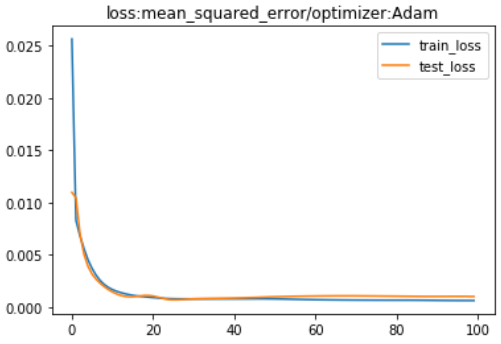
$$w_{t+1} = w_t + \Delta w_t$$

$$\Delta w_t = -\eta \frac{v_t}{\sqrt{v_t + \varepsilon}} * g_t$$

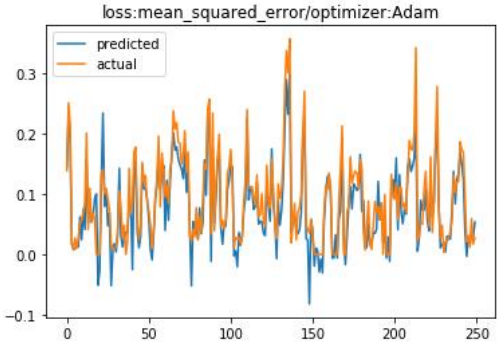
$$v_t = \beta_1 * v_{t-1} - (1-\beta_1) * g_t$$

$$s_t = \beta_2 * s_{t-1} - (1-\beta_2) * g_t^{\,2}$$

که در آن η، نرخ یادگیری اولیه ، g_t، گرادیان در زمان t در امتداد w، v_tمیانگین نمایی گرادیان در امتداد w ، و s_t،میانگین نمایی مربعات گرادیان خواهد بود . شکل های زیر نمودار خطای تست و ترین و همچنین مقدار پیش بینی و حقیقی را برای شبکه ی RNN که در قسمت های قبل توضیح داده شده نشان میدهد :



شکل ۴) نمودار loss برای شبکه RNN با آپتیمایزرAdam



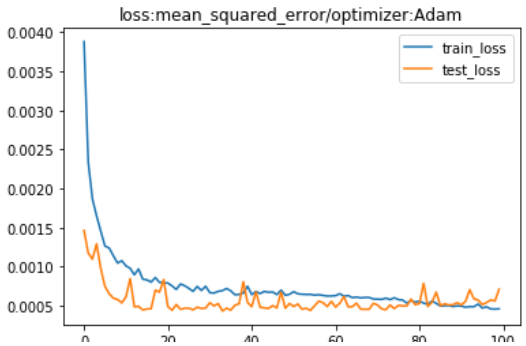
شکل ۵) نمودار predict برای شبکهRNN با آپتیمایزرAdam

همانطور که مشاهده میشود نمودارهای تست و ترین به خوبی همدیگر را دنبال میکنند و هیچ گونه اعوجاجی در این نمودارها دیده نمیشود که نشان دهنده ی این است که این بهینه ساز به همراه تابع خطای mse بخوبی عمل میکند همچنین زمان آموزش را در زیر میتوانیم مشاهده کنیم .

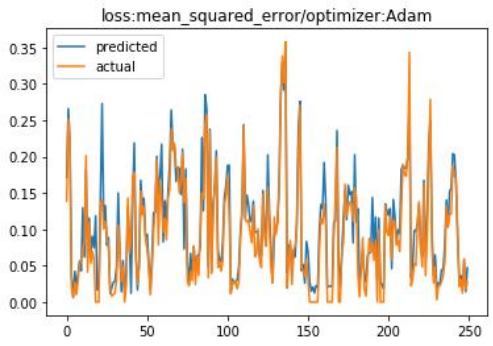
```
Epoch 100/100
 - 1s - loss: 6.3252e-04 - val_loss: 0.0010
RNN learning time( mean_squared_error and Adam ) is:65.45
RMSE for test data is : 0.032
```

طبق نتیجه بالا به دلیل سادگی شبکه RNN زمان آموزش کم است (65.45S)

در شکل زیر میتوانیم شاهد نمودارها برای یک شبکهLSTM باشیم که راجع به شبکهLSTM به تفصیل در قسمت های قبل بحث کردیم همچنین مانند مرحله ی قبل از بهینه سازadam به همراه تابع خطای mseاستفاده میکنیم :



شکل ۶)نمودار loss برای شبکه LSTM با آپتیمایزرAdam

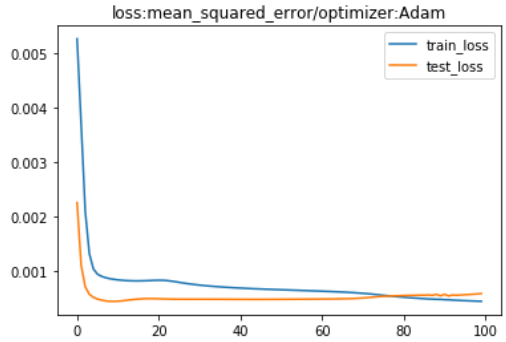


شکل ۷)نمودار predict برای شبکه LSTM با آپتیمایزرAdam

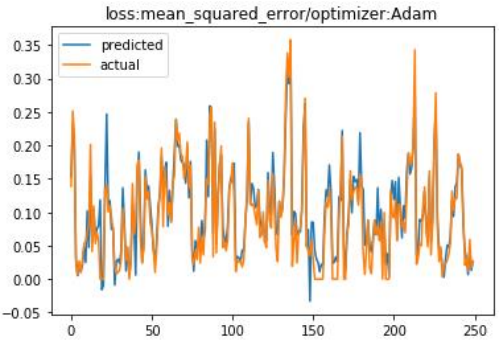
همانطور که مشاهده میشود زمان آموزش برای شبکه LSTMدر مقایسه با RNN افزایش یافته ولی خطا کاهش یافته است . همچنین در نمودار خطا شاهد اعوجاج هایی هستیم

```
Epoch 100/100
 - 1s - loss: 4.5905e-04 - val_loss: 7.0906e-04
LSTM learning time( mean_squared_error and Adam ): 120.90427303314209
RMSE for test data is : 0.027
```

حال تاثیر شبکه ی GRU را برای بهینه ساز Adam و تابع MSE مورد بررسی قرار میدهیم :



شکل ۸) نمودار loss برای شبکه GRU با آپتیمایزر Adam



شکل۹) نمودار predict برای شبکه GRU با آپتیمایزرAdam

همانطور که مشاهده میشود زمان آموزش شبکه GRU با اختلاف کمی از شبکه RNNبیش تر و از شبکهLSTM کمتر است همچنین خطا برای این شبکه نسبت به دو شبکه ی قبل کمتر شده است

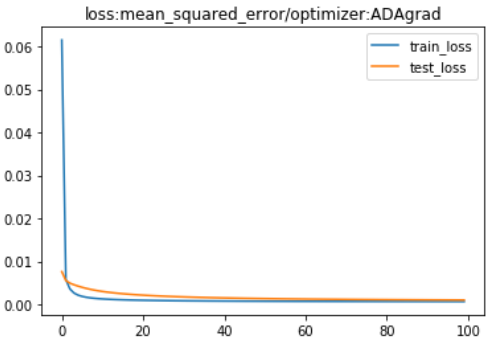
```
Epoch 100/100
 - 1s - loss: 4.4953e-04 - val_loss: 5.9043e-04
GRU learning time( mean_squared_error and Adam ) is:83.67
```

بعد از بهینه ساز adam به سراغ بهینه ساز adagrad میرویم و تاثیر این بهینه ساز را به همراه mse برای ۳ شبکه rnn,lstm.gru بررسی میکنیم .

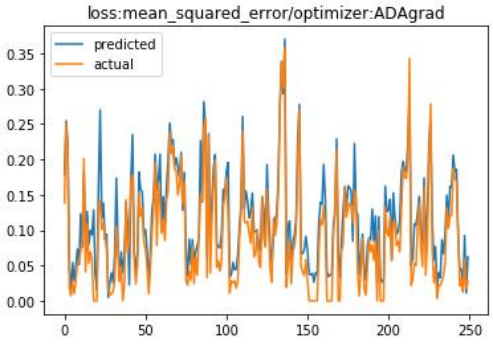
- **بهینه ساز ADagrad :**

Adagrad

Adagrad adapts the learning rate specifically to individual features: that means that some of the weights in your dataset will have different learning rates than others. This works really well for sparse datasets where a lot of input examples are missing. Adagrad has a major issue though: the adaptive learning rate tends to get really small over time. Some other optimizers below seek to eliminate this problem.



شکل ۱۰) نمودار loss برای شبکه RNN با آپتیمایزر Adgrad



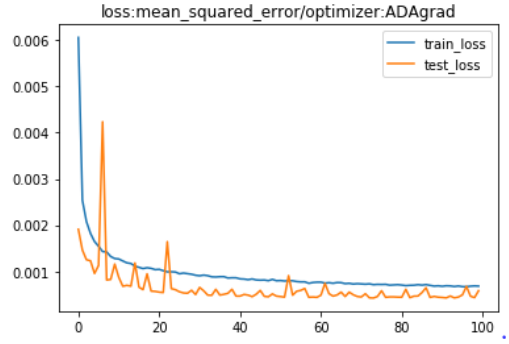
شکل۱۱)نمودار predict برای شبکه RNN با آپتیمایزر Adgrad

همانطور که مشاهده میشود نمودار تست به خوبی و بدون اعوجاج نمودار ترین را دنبال میکند و شکل سمت راست نیز بیانگرنمودار مقدار حقیقی که در دیتاها موجود است و مقدار پیش بینی شده توسط شبکه را نشان میدهد که اختلاف این دو مقدار loss در هر مقدار را نشان میدهد .

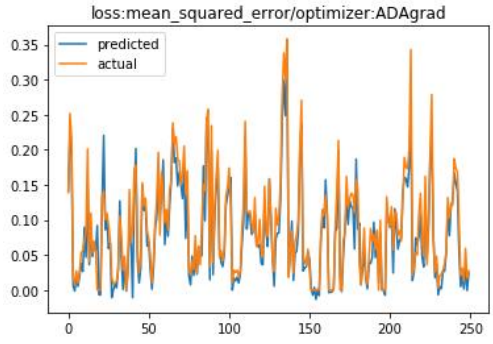
```
Epoch 100/100
 - 1s - loss: 6.7597e-04 - val_loss: 0.0010
RNN learning time( mean_squared_error and ADAgrad ) is:67.51
RMSE for test data is : 0.032
```

نتیجه بالا نشان دهنده ی این است که در شبکه RNN بهینه ساز Adgrad نسبت به Adam زمان آموزش افزایش یافته و مقدار خطا تغییر نکرده است .

شکل زیر نشان دهنده ی شبکه LSTM هنگام استفاده از بهینه ساز Adgrad است .



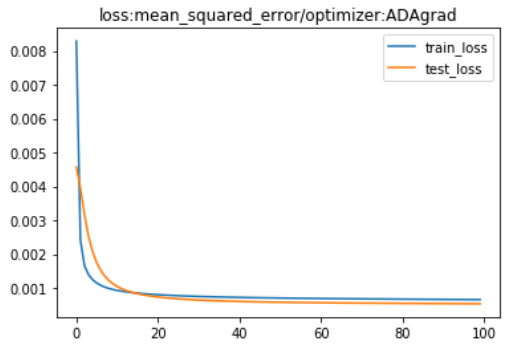
شکل۱۲) نمودار loss برای شبکه LSTM با آپتیمایزرAdagrad



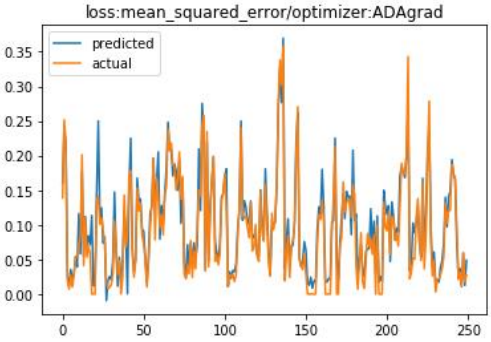
شکل۱۳)نمودار predict برای شبکه LSTM با آپتیمایزرAdagrad

```
Epoch 100/100
 - 1s - loss: 6.8246e-04 - val_loss: 5.8295e-04
LSTM learning time( mean_squared_error and ADAgrad ): 119.43060159683228
RMSE for test data is : 0.001
```

همانطور که مشاهده میکنیم در شبکه LSTM با بهینه ساز Adagrad نسبت به Adam زمان آموزش افزایش یافته و خطا با تفاوت ناچیزی کمتر شده است . در نموار حطا در ایپاک های اول شامل اعوجاج های با پیک بزرگ هستیم که با افزایش ایپاک ها این پیک کاهش میابد .



شکل۱۴) نمودار loss برای شبکه GRU با آپتیمایزر Adgrad



شکل۱۵) نمودار predict برای شبکه GRU با آپتیمایزر Adgrad

```
Epoch 100/100
 - 1s - loss: 6.5641e-04 - val_loss: 5.3587e-04
GRU learning time( mean_squared_error and ADAgrad ) is:70.53
RMSE for test data is : 0.023
```

این نتیجه نیز نشان دهنده ی این است که وقتی از بهینه ساز Adgrad استفاده میکنیم زمان آموزش بیشتر (در مقایسه با بهینه ساز Adam) میشود و خطا نیز تغییر نکرده است . همچنین در شبکه ی gru برخلاف lstm مشاهده میکنیم که در نمودار خطا خطای تست و ترین به خوبی همدیگر را دنبال میکنند که این نتیجه در rnn نیز مشاهده میشد .

- **بهینه ساز RMSProp :**

RMSprop

RMSprop is a special version of Adagrad developed by Professor Geoffrey Hinton in his neural nets class. Instead of letting all of the gradients accumulate for momentum, it only accumulates gradients in a fixed window. RMSprop is similar to Adaprop, which is another optimizer that seeks to solve some of the issues that Adagrad leaves open.

این بهینه ساز نیاز به تنظیم نرخ یادگیری را حذف و آن را به صورت خودکار انجام میدهد همچین این بهینه ساز به ازای هر پارامتر یک نرخ یاد گیری متفاوت را اتخاذ خواهد کرد . در RMSProp به ازای هر پارامتر ، هر بروزرسانی بصورت جداگانه و مطابق با معادلات زیر انجام میگیرد :

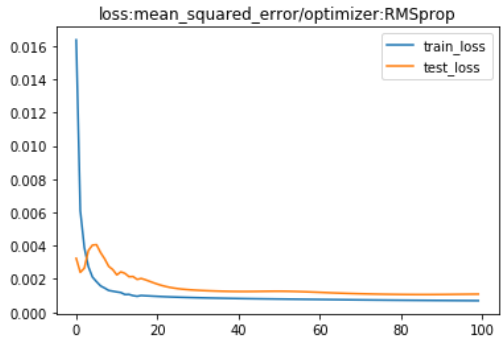
$$w_{t+1} = w_t + \Delta w_t$$

$$\Delta w_t = -\frac{\eta}{\sqrt{v_t + \varepsilon}} * g_t$$

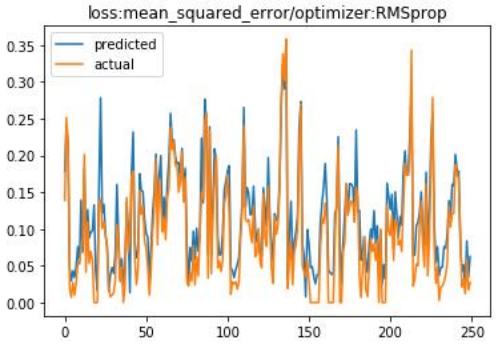
$$v_t = p * v_{t-1} + (1-p) * g_t^{\,2}$$

که در آن η، نرخ یادگیری اولیه ، v_tمیانگین نمایی مربعات گرادیان و g_t، گرادیان در زمان t در امتداد w خواهد بود .

حال در این مرحله بهینه سازRMSProp را برای هر سه شبکه مورد بررسی قرار میدهیم که شکل های زیر برای شبکه RNN است:



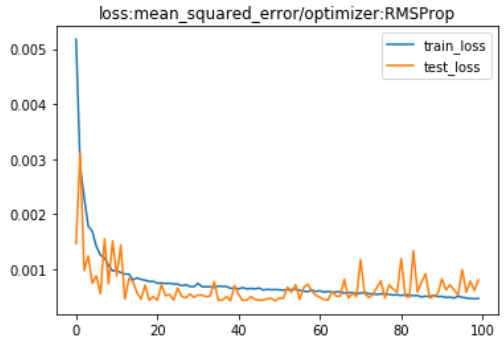
شکل۱۶)نمودار loss برای شبکه RNN با آپتیمایزر RMSProp



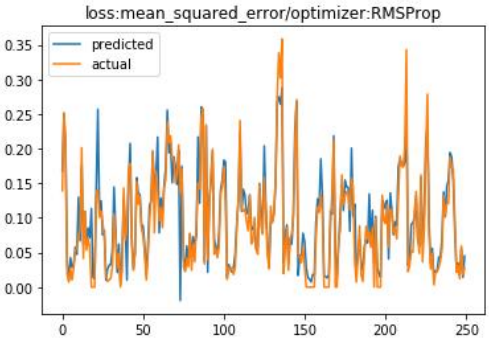
شکل۱۷)نمودار predict برای شبکه RNN با آپتیمایزر RMSProp

```
Epoch 100/100
 - 1s - loss: 6.9411e-04 - val_loss: 0.0011
RNN learning time( mean_squared_error and RMSprop ) is:69.76
RMSE for test data is : 0.033
```

برای شبکه RNN خطای بهینه ساز RMSProp نسبت به دو بهینه ساز قبلی بیشتر شده است



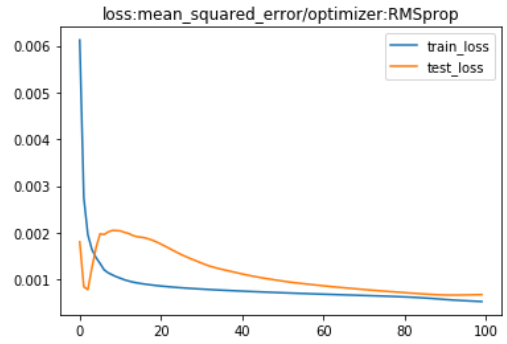
شکل ۱۸)نمودار loss برای شبکه LSTM با آپتیمایزر RMSProp



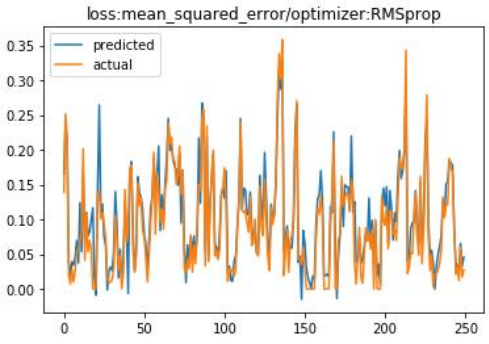
شکل۱۹)نمودار predict برای شبکه LSTM با آپتیمایزر RMSProp

```
Epoch 100/100
 - 1s - loss: 4.7090e-04 - val_loss: 8.0632e-04
LSTM learning time( mean_squared_error and RMSProp ): 134.44413781166077
```

برای شبکه LSTM هنگام استفاده از بهینه ساز RMSProp زمان آموزش در مقایسه با دو بهینه ساز قبل بیشتر شده ولی خطا با اختلاف ناچیزی تقریبا میتوان گفت تغییری نکرده است .



شکل۲۰)نمودار loss برای شبکه GRU با آپتیمایزر RMSProp



شکل ۲۱)نمودار predict برای شبکه GRU با آپتیمایزر RMSProp

```
Epoch 100/100
 - 1s - loss: 4.5089e-04 - val_loss: 9.7376e-04
GRU learning time( mean_squared_error and RMSprop ) is:85.59
```

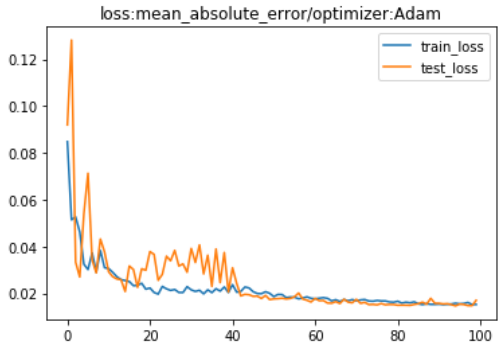
مطابق شکل بالا برای شبکه GRU هنگام استفاده از بهینه ساز RMSProp هم خطا و هم زمان آموزش افزایش یافته است .

- **تابع خطای MAE**

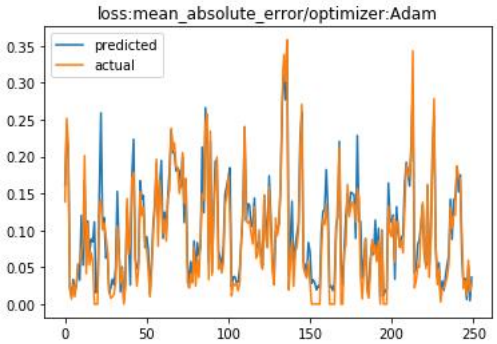
در اینجا قدرمطلق اختلاف ها بین مقدار واقعی و پیش بینی شده محاسبه میشود تا از بروز خطای منفی جلوگیری شود . باید به این نکته توجه کرد که اگر داده ها شامل نویز و داده های پرت نباشد در این صورت استفاده ازMSE خوب است ولی اگر داده های شما دارای نویز یا داده های پرت باشد ، در کل MSEتقویت می شود که خوب نیست. در این حالت بهتر است از MAE استفاده شود.

حال در این قسمت تابع خطای mae را به ازای سه بهینه ساز ذکر شده و همچنین شبکه های مختلف مورد بحث قرار میدهیم در ابتدا بهینه ساز Adam را در نظر میگیریم

- **بهینه ساز Adam :**



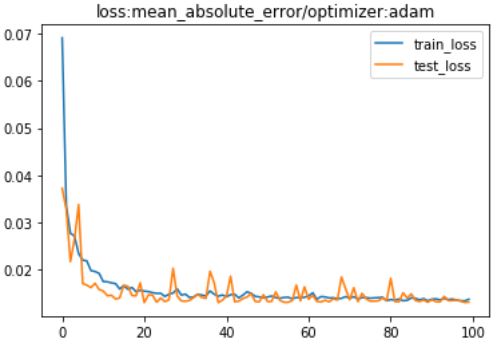
شکل۲۲) نمودار loss برای شبکه RNN با آپتیمایزر Adam



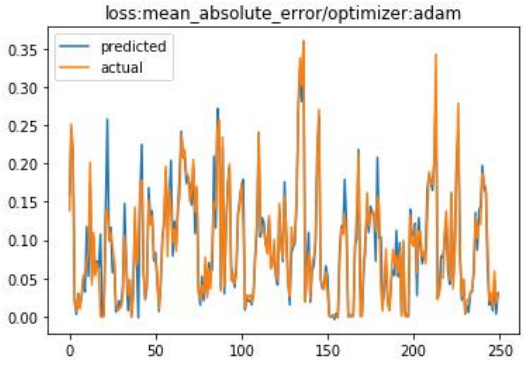
شکل ۲۳) نمودار predict برای شبکه RNN با آپتیمایزر Adam

از نمودار خطا در ابتدا شاهد نوسانات زیادی هستیم که با افزایش ایپاک ها این نوسانات به مرور زمان میرا میشود همچنین خطا در هنگام استفاده از تابع خطای mae در مقایسه با mse با روش بهینه سازی یکسان افزایش یافته است ولی زمان اجرا کمتر شده است .

```
Epoch 100/100
 - 1s - loss: 0.0152 - val_loss: 0.0170
RNN learning time( mean_absolute_error and Adam ) is:56.28
```



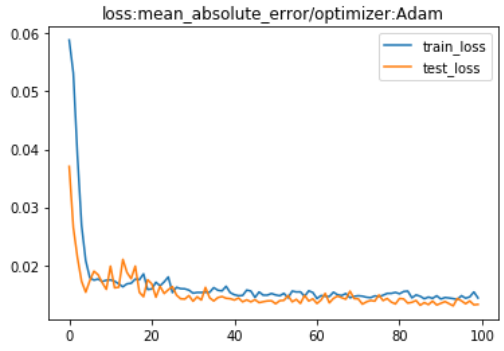
شکل ۲۴) نمودار loss برای شبکه LSTM با آپتیمایزر Adam



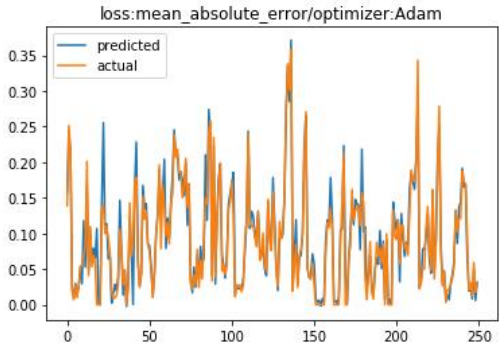
شکل۲۵) نمودار predict برای شبکه LSTM با

در LSTM نیز هنگام استفاده از MAE خطا افزایش یافته ولی زمان کماکان تغییری نکرده است همچنین نوساناتی در داده های تست مشاهده میشود

```
Epoch 100/100
 - 1s - loss: 0.0137 - val_loss: 0.0131
LSTM learning time( mean_absolute_error and adam ): 120.59618473052979
```



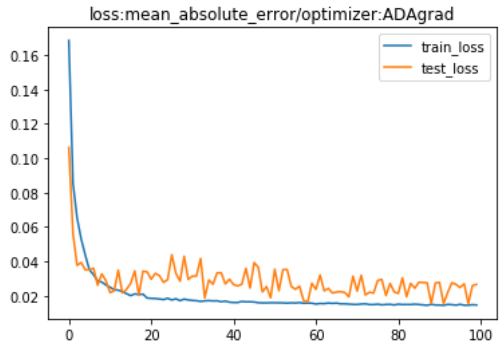
شکل ۲۶)نمودار loss برای شبکه GRU با آپتیمایزر Adam



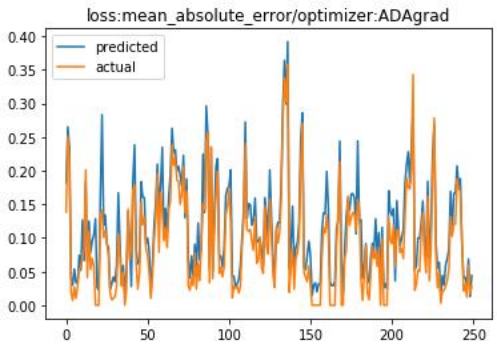
شکل۲۷) نمودار predict برای شبکه GRU با آپتیمایزر Adam

```
Epoch 100/100
 - 1s - loss: 0.0143 - val_loss: 0.0132
GRU learning time( mean_absolute_error and Adam ) is:72.27
```

- **بهینه ساز Adgrad :**

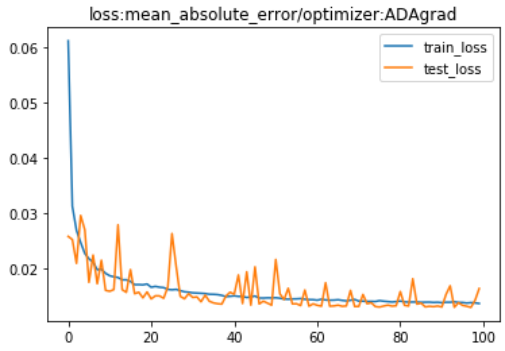


شکل ۲۸)نمودار loss برای شبکه GRU با آپتیمایزر Adgrad

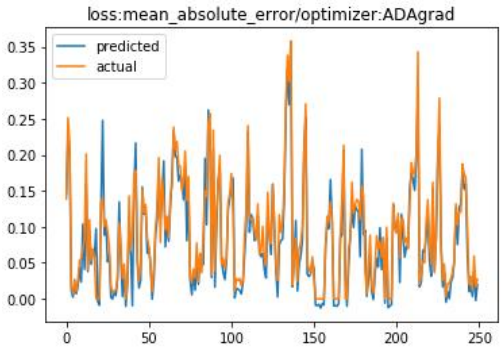


شکل ۲۹) نمودار predict برای شبکه GRU با آپتیمایزر Adgrad

```
Epoch 100/100
 - 1s - loss: 0.0146 - val_loss: 0.0266
RNN learning time( mean_absolute_error and ADAgrad ) is:54.35
```

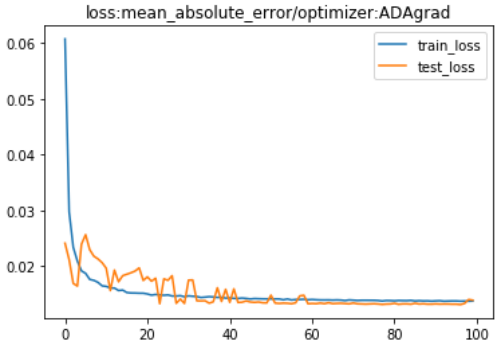


شکل ۳۰) نمودار loss برای شبکه LSTM با آپتیمایزر Adgrad

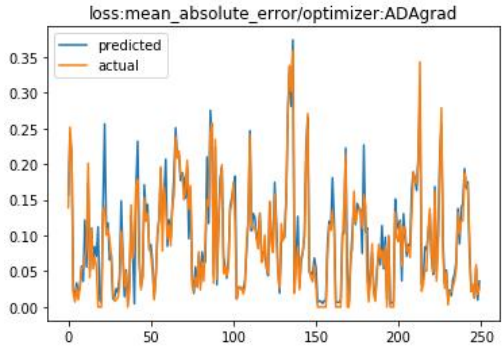


شکل ۳۱) مودار predict برای شبکه LSTM با آپتیمایزر Adgrad

```
Epoch 100/100
 - 1s - loss: 0.0136 - val_loss: 0.0163
LSTM learning time( mean_absolute_error and ADAgrad ): 134.90626072883606
```



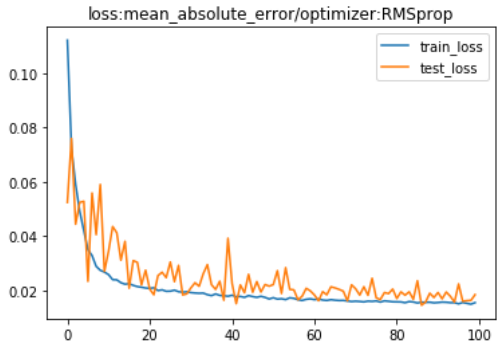
شکل۳۲)نمودار loss برای شبکه GRU با آپتیمایزر Adgrad



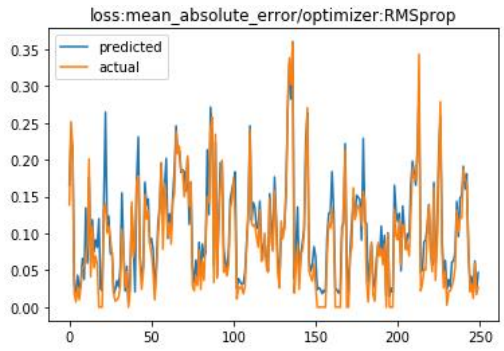
شکل۳۳)نمودار predict برای شبکه GRU با آپتیمایزر Adgrad

```
Epoch 100/100
 - 1s - loss: 0.0137 - val_loss: 0.0139
GRU learning time( mean_absolute_error and ADAgrad ) is:58.51
```

- **بهینه ساز RMSProp :**

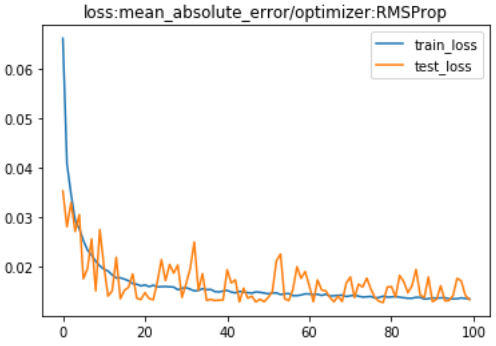


شکل۳۲)نمودار loss برای شبکه RNN با آپتیمایزرRMSprop

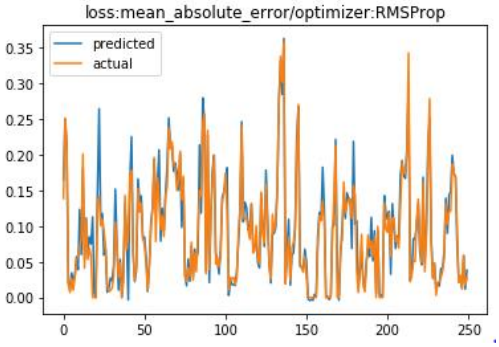


شکل۳۳)نمودار predict برای شبکه RNN با آپتیمایزرRMSprop

```
Epoch 100/100
 - 1s - loss: 0.0155 - val_loss: 0.0184
RNN learning time( mean_absolute_error and RMSprop ) is:55.29
```

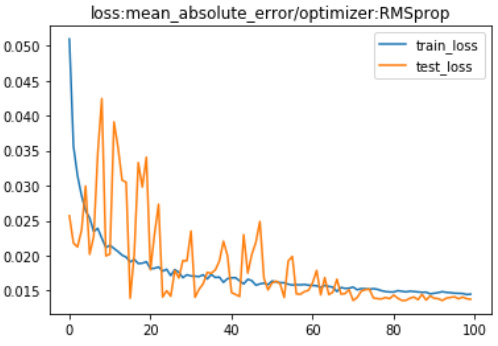


شکل۳۴)نمودار loss برای شبکه LSTM با آپتیمایزرRMSprop

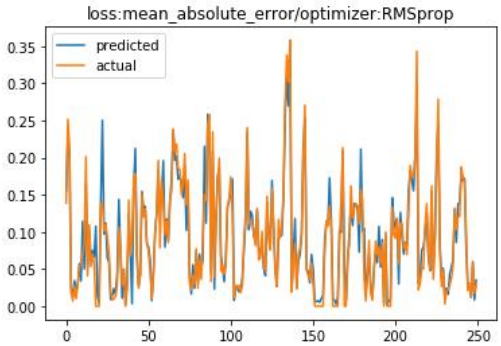


شکل۳۵)نمودار predict برای شبکه LSTM با آپتیمایزرRMSprop

```
Epoch 100/100
 - 1s - loss: 0.0136 - val_loss: 0.0134
LSTM learning time( mean_absolute_error and RMSProp ): 124.25345134735107
```



شکل۳۶)نمودار loss برای شبکه GRU با آپتیمایزرRMSprop



شکل۳۷)نمودار predict برای شبکه GRU با آپتیمایزر RMSprop

```
Epoch 100/100
 - 1s - loss: 0.0145 - val_loss: 0.0138
GRU learning time( mean_absolute_error and RMSprop ) is:56.13
```

همانطور که میدانیم از این توابع برای آپدیت کردن وزن ها پس از هر دور استفاده میشود (بسته به اندازه ی batchsize) هر کدام از این روش ها به نوعی سعی میکنند وزن ها را آپدیت کنند و شبکه را در مسیر درست قرار دهند همانطور که در جدول زیر مشاهده میشود در میان این توابع بهینه ساز Adam نتیجه ی بهتری گرفته است در این روش سعی میشود که از نوسانات در جهات غیر از جهت حرکت به سمت نقطه بهینه کاسته و به تقویت حرکت سریع تر به سمت محل بهینه بپردازد . جدول زیر خلاصه ای از نتایج زمان آموزش و مقدار خطا را به ازای هر سه شبکه هنگام استفاده از سه بهینه ساز مختلف نشان میدهد .

	optimization	Network	Time	Loss
MSE	Adam	RNN	65.45	0.00063
		LSTM	120.9	0.00045
		GRU	83.66	0.00044
	ADagrad	RNN	67.51	0.00068
		LSTM	119.43	0.00068
		GRU	70.53	0.00065
	RMSProp	RNN	69.76	0.00069
		LSTM	134.4	0.00047
		GRU	85.59	0.00045
MAE	Adam	RNN	56.28	0.015
		LSTM	120.59	0.013
		GRU	72.27	0.0143
	ADagrad	RNN	54.35	0.0146
		LSTM	134.9	0.0136
		GRU	58.51	0.0137
	RMSProp	RNN	55.29	0.0155
		LSTM	124.25	0.0136
		GRU	56.13	0.0145

همانطور که در جدول بالا مشاهده میشود دلیل نزدیکی نتایج سه شبکه بخاطر کوتاه بودن طول برگشتی است که در نظرگرفته شده و اگر این مقدار افزایش یابد انتظار داریم که شبکه RNN دیگر به خوبی دو شبکه دیگر کار نکند

- **نتیجه نهایی :**

با توجه به نتایج جدول و تحلیل های صورت گرفته نتیجه میگیریم که نحوه عملکرد شبکه برای روش های بهینه سازی متفاوت (Adam,ADagrad,RMSProp) به این صورت است که برای هر سه روش بهینه سازی کمترین زمان مربوط به شبکه RNN میباشد که دلیل آن نیز ساختار ساده تر این شبکه نسبت به دو شبکه دیگر و در نتیجه محاسبات کمتر و پیرو آن یادگیری سریع تراین شبکه میتواند باشد.

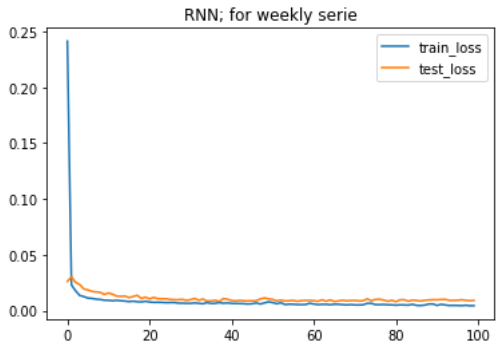
همچنین زمان شبکه GRU نیز کمتر از شبکه LSTM میباشد و این نیز به دلیل ساده کردن طراحی LSTM و حذف یکی از گیت های LSTM در این شبکه است . همچنین مشاهده میشود که مقدار Loss بدست آمده برای شبکه GRU کمتر از دو شبکه دیگر است و آن هم به دلیل تغییراتی است که در این شبکه داده شده و برخی از مشکلات دو شبکه ی دیگر را برطرف کرده است .

۴)

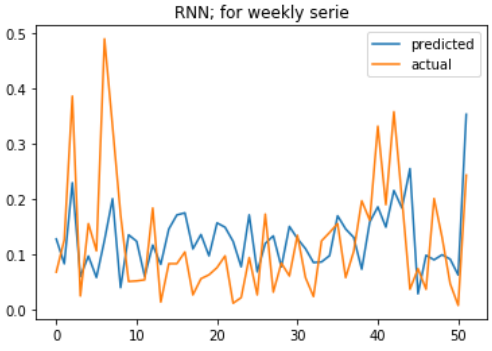
در این قسمت می‌خواهیم سری‌های زمانی هفتگی و ماهانه را بررسی کنیم. سری هفتگی بدین معنا که با استفاده از اطلاعات یک ساعت مشخص(ساعت ۱ تا ۲۴) ۶ روز پیاپی سعی کنیم الودگی همان ساعت روز هفتم را پیش‌بینی کنیم. در این حالت چون تعداد داده‌های اموزش کم است، پیش‌بینی خوبی نخواهیم داشت، اما نمودار loss کاهشی خواهد بود. توجه شود که برای حل این مرحله از بهینه ساز adam و تابع خطای mse استفاده کردیم همچنین کل داده ها را در نظر گرفتیم .

- **سری زمانی هفتگی :**

RNN



شکل۳۸)نمودار loss برای شبکه rnn برای سری هفتگی

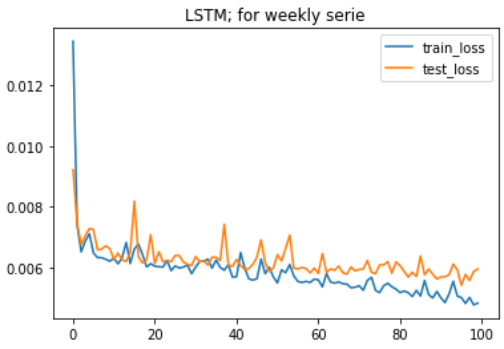


شکل۳۹)نمودار predict برای شبکه rnn برای سری هفتگی

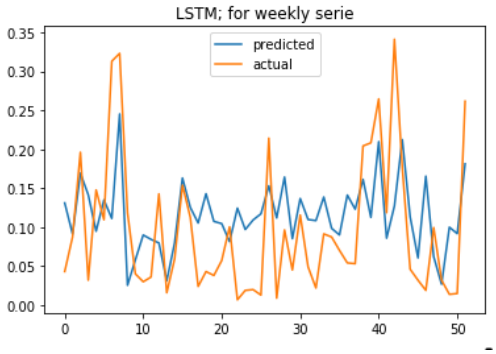
```
Epoch 100/100
 - 0s - loss: 0.0044 - val_loss: 0.0091
learning time is:7.69
RMSE for test data is : 0.095
```

همانطور که در شکل های بالا نشان داده شده است نمدار خطا به خوبی همدیگر را دنبال میکنند و مشکل اورفیتینگ نیز ندارند و پیش بینی هم خوب نشان داده شده است

LSTM



شکل۴۰)نمودار loss برای شبکه lstm برای سری هفتگی

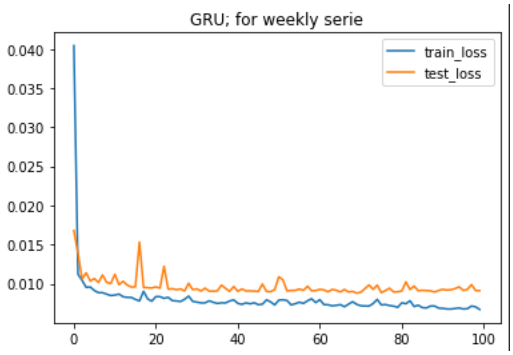


شکل۴۱)نمودار predict برای شبکه lstm برای سری هفتگی

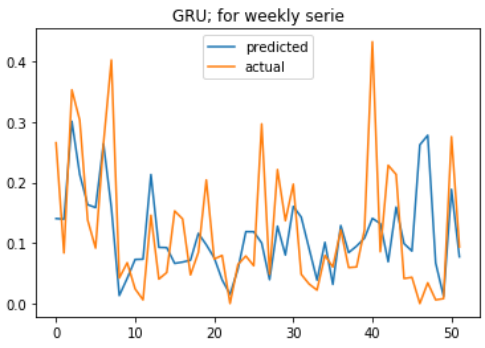
```
Epoch 100/100
 - 0s - loss: 0.0048 - val_loss: 0.0060
learning time is:15.20
RMSE for test data is : 0.077
```

زمان آموزش شبکه lstm از rnn بیشتر شده است و در بین سه شبکه بیشترین زمان آموزش را به خود اختصاص داده است همچنین نمودار خطای آن نزولی هست هر چند که در اپیاکهای آخر فاصله بین دو نمودار افزایش میابد ولی باز به دلیل اسکیل و اختلاف ناچیز باعث مشکل اورفیتینگ میشود

GRU



شکل۴۲)نمودار loss برای شبکه gru برای سری هفتگی



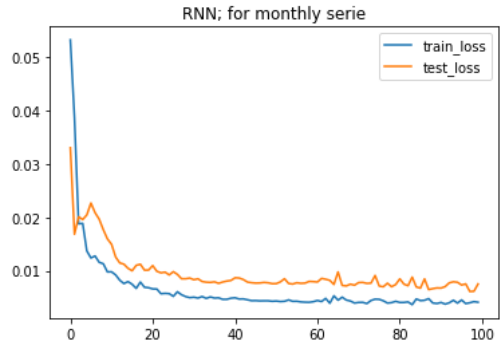
شکل۴۳)نمودار predict برای شبکه gru برای سری هفتگی

```
Epoch 100/100
 - 0s - loss: 0.0067 - val_loss: 0.0091
 learning time is:14.86
RMSE for test data is : 0.095
```

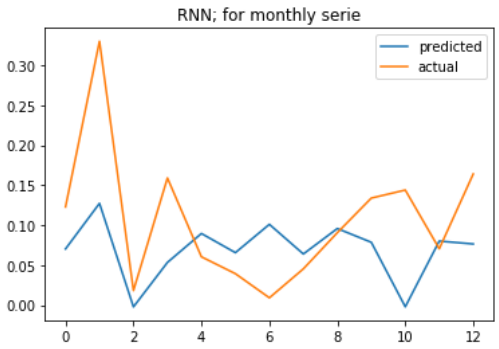
در مرحله ی آخر سری زمانی هفتگی را برای شبکه gru بررسی کردیم که مقدار rmse آن با شبکه rnn برابر ولی در مقایسه با lstm افزایش یافته است .

- **سری زمانی ماهانه :**

RNN



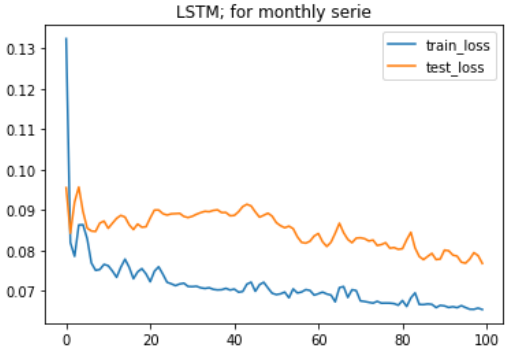
شکل۴۴)نمودار loss برای شبکه rnn برای سری ماهانه



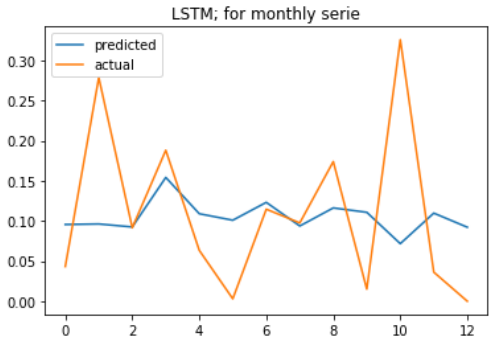
شکل۴۵)نمودار predict برای شبکه rnn برای سری ماهانه

```
Epoch 100/100
 - 0s - loss: 0.0041 - val_loss: 0.0075
learning time is:2.53
RMSE for test data is : 0.087
```

LSTM



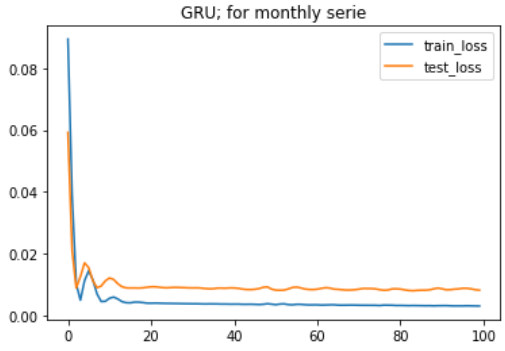
شکل۴۶)نمودار loss برای شبکه lstm برای سری ماهانه



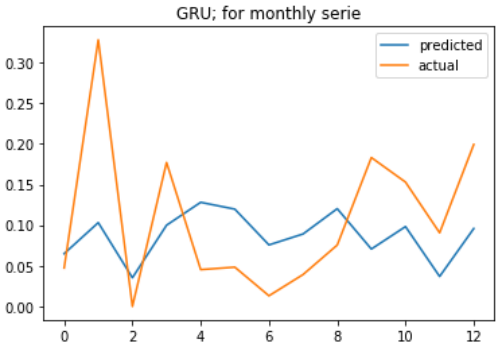
شکل۴۷)نمودار predict برای شبکه lstm برای سری ماهانه

```
Epoch 100/100
 - 0s - loss: 0.0655 - val_loss: 0.0768
 learning time is:4.29
RMSE for test data is : 0.104
```

GRU



شکل۴۸)نمودار loss برای شبکه gru برای سری ماهانه



شکل۴۹)نمودار predict برای شبکه gru برای سری ماهانه

```
Epoch 100/100
 - 0s - loss: 0.0031 - val_loss: 0.0083
learning time is:4.41
RMSE for test data is : 0.091
```

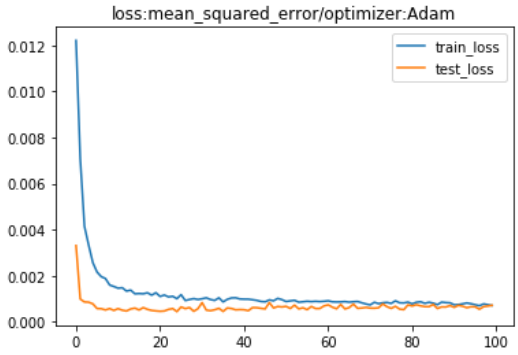
برای سری زمانی ماهانه نیز با دقت در شکل ها و نتایج حاصل شده در میابیم که شبکه lstm بیشترین زمان را دارد که بخاطر ساختار پیچیده آن است و شبکه rnn با توجه به ساختار ساده ای که دارد زمان آموزش کمتری دارد همچنین نمودار خطا برای هر سه شبکه شیب نزولی دارد که برای gru برخلاف دو شبکه ی دیگر نرم تر است و اعوجاج ندارد . در این مرحله شبکه rnn و gru بهتر عمل کرده و پیش بینی بهتری داشتند .

۵) تاثیر لایه دراپ اوت بر روی یک شبکه دلخواه

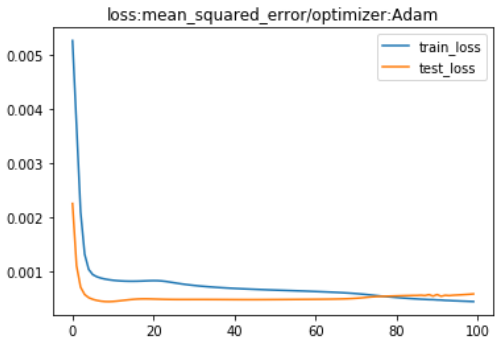
در این مرحله تاثیر لایه دراپ اوت را بر روی شبکه GRU بررسی میکنیم : از بهینه ساز Adam و mse استفاده میکنیم همینطور ساختار شبکه به صورت زیر است

```
Model: "sequential_47"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_31 (GRU)                 (None, 50)                8850
_________________________________________________________________
dropout_7 (Dropout)          (None, 50)                0
_________________________________________________________________
dense_47 (Dense)             (None, 1)                 51
_________________________________________________________________
```

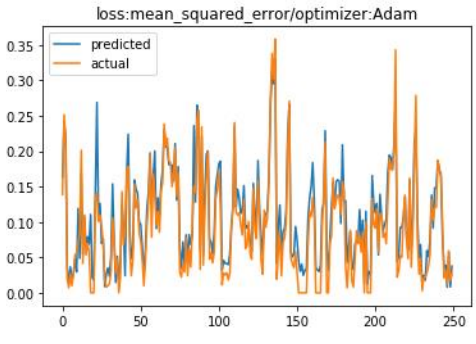
شکل های زیر شبکه GRU را در دو حالت (با دراپ اوت و بدون دراپ اوت مقایسه میکند)



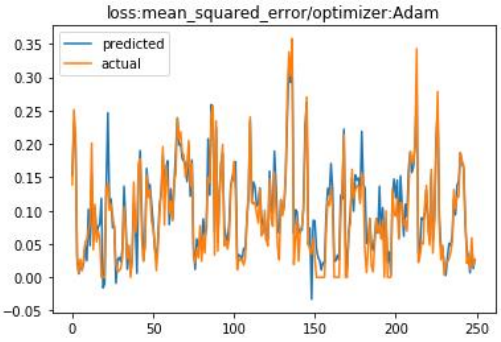
شکل ۵۰) نمودار loss برای شبکه GRU با دراپ اوت



شکل ۵۱) نمودار predict برای شبکه GRU بدون دراپ اوت



شکل۵۲) نمودار loss برای شبکه GRU با دراپ اوت



شکل ۵۳) نمودار predict برای شبکه GRU بدون دراپ اوت

```
Epoch 100/100
 - 1s - loss: 7.1321e-04 - val_loss: 7.2276e-04
GRU learning time( mean_squared_error and Adam ) is:63.43
```

چنانچه مشاهده میکنیم با این کار مقدار Loss درصد خیلی کمی تغییر میکند (بیشتر میشود) بنظر میرسد دراپ اوت نمیتواند شبکه را دچار بهبود کند.

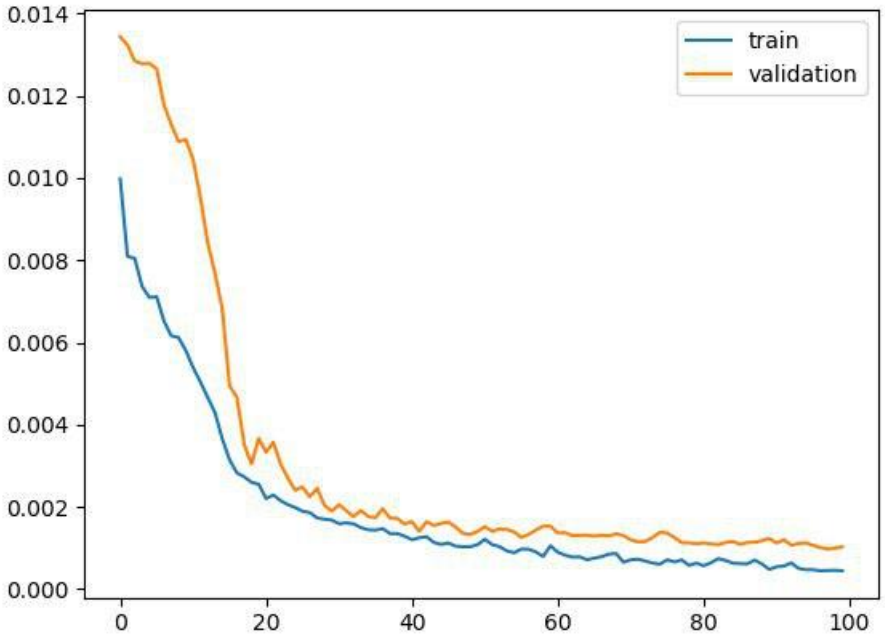
۶)

در این قسمت از ما خواسته شده است که سه شبکه ی بازگشتی به موازات هم بسازیم پنجره ی زمانی را بگونه ای تنظیم میکنیم که هر سه شبکه یه ساعت را پیش بینی کنند درنتیجه ما پیش بینی روزانه و هفتگی و ماهانه از یک سلول LSTM را مورد بررسی قرار میدهیم .بعد از خروجی هر سه میانگین میگیریم خلاصه ساختار مدل به صورت زیر است :

Layer (type)	Output Shape	Param #	Connected to
input_4 (InputLayer)	[(None, 11, 8)]	0	
input_5 (InputLayer)	[(None, 6, 8)]	0	
input_6 (InputLayer)	[(None, 3, 8)]	0	
gru (GRU)	(None, 50)	9000	input_4[0][0]
gru_1 (GRU)	(None, 50)	9000	input_5[0][0]
gru_2 (GRU)	(None, 50)	9000	input_6[0][0]
dense_4 (Dense)	(None, 1)	51	gru[0][0]
dense_5 (Dense)	(None, 1)	51	gru_1[0][0]
dense_6 (Dense)	(None, 1)	51	gru_2[0][0]
average_1 (Average)	(None, 1)	0	dense_4[0][0] dense_5[0][0] dense_6[0][0]
dense_7 (Dense)	(None, 1)	2	average_1[0][0]

```
Total params: 27,155
Trainable params: 27,155
Non-trainable params: 0
```

نتیجه نهایی در شکل زیر نشان داده شده است :



شکل ۵۴)نتیجه فیوژن خروجی

همان طور که مشاهده میکنیم در صورتی که از فیوژن استفاده کنیم،دقت تخمین نسبت به حالتی که فقط از سری هفتگی و یا فقط از سری ماهانه استفاده شود، بیشتر است. در واقع مقدار خطا کمتر است . اما در حالتی که سری روزانه استفاده میکنیم، چون تعداد داده‌ها برای اموزش شبکه زیاد است بدون استفاده از فیوژن به نتیجه مطلوبی میرسیم

۷)

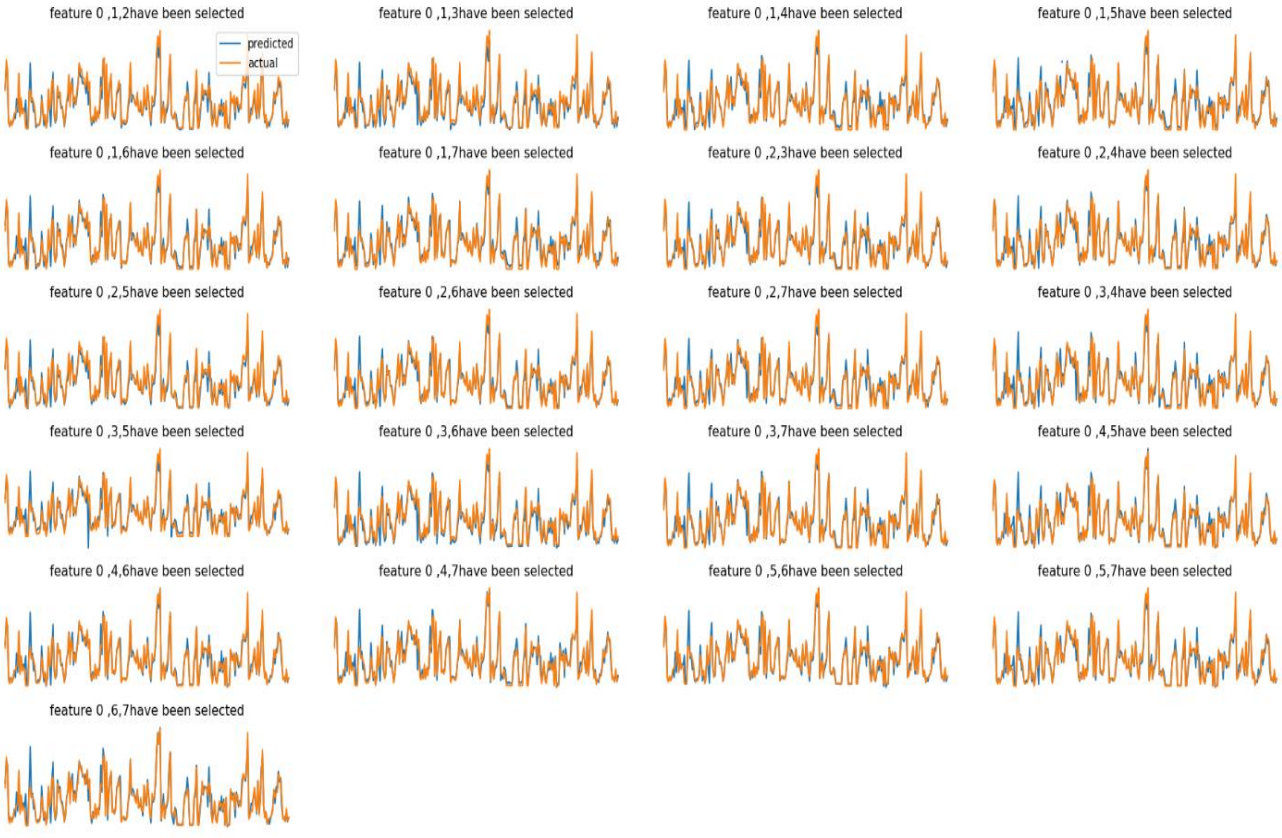
دراین سوال از ما خواسته شده است سه ویژگی با بیشترین تاثیر روی prediction را مشخص کنیم، در واقع می‌خواهیم feature selection انجام دهیم. روش‌های متعددی برای freature selection وجود دارد از جمله : Exhustive search و forward selectionو stepwize selection و backward elimination و...

روش exhustive search یعنی همه‌ی حالت‌های ممکن را ازمایش کنیم. بدین وسیله به optimum global می‌رسیم. اگر ابعاد بردار ویژگی بزرگ باشد به دلیل وجود محدودیت‌های پردازشی نمی‌توان از این روش استفاده کرد و مجبوریم از دیگر روش‌های فوق‌الذکر بهره بگیریم که ما را sub-optimal می‌رساند.اما در این سوال چون exhustive search ممکن است; بدین روش می‌پردازیم.

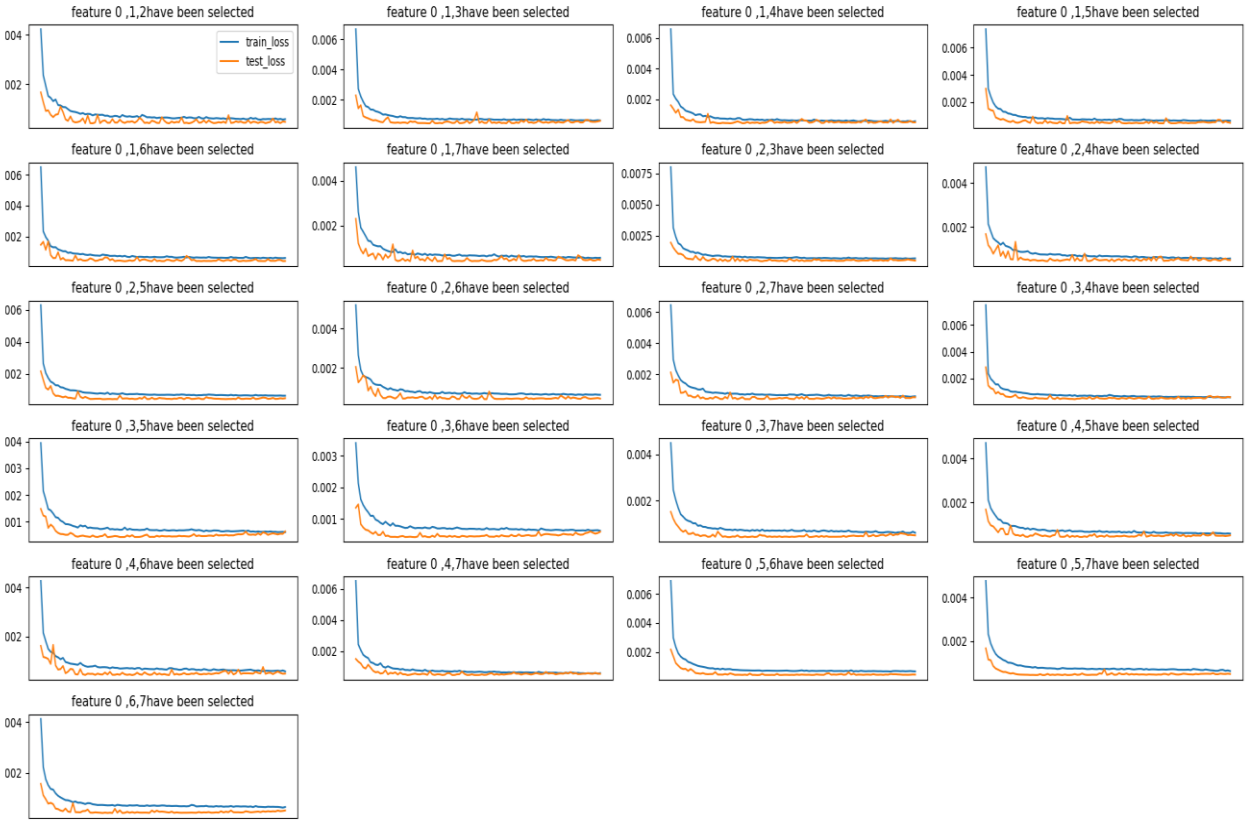
همه‌ی حالت‌های ممکن یعنی انتخاب ویژگی 0,1,2 و 0,1,3 و 0,1,4 و .... که در مجموع ۲۱ حالت مختلف می‌شود را چک می‌کنیم. همینطور این کار را با شبکه‌های LSTM , RNN , GRU انجام میدهیم.

انتخاب سه ویژگی موثر تر (می‌دانیم که ویژگی آلودگی ثابت است) به وسیله شبکه LSTM :

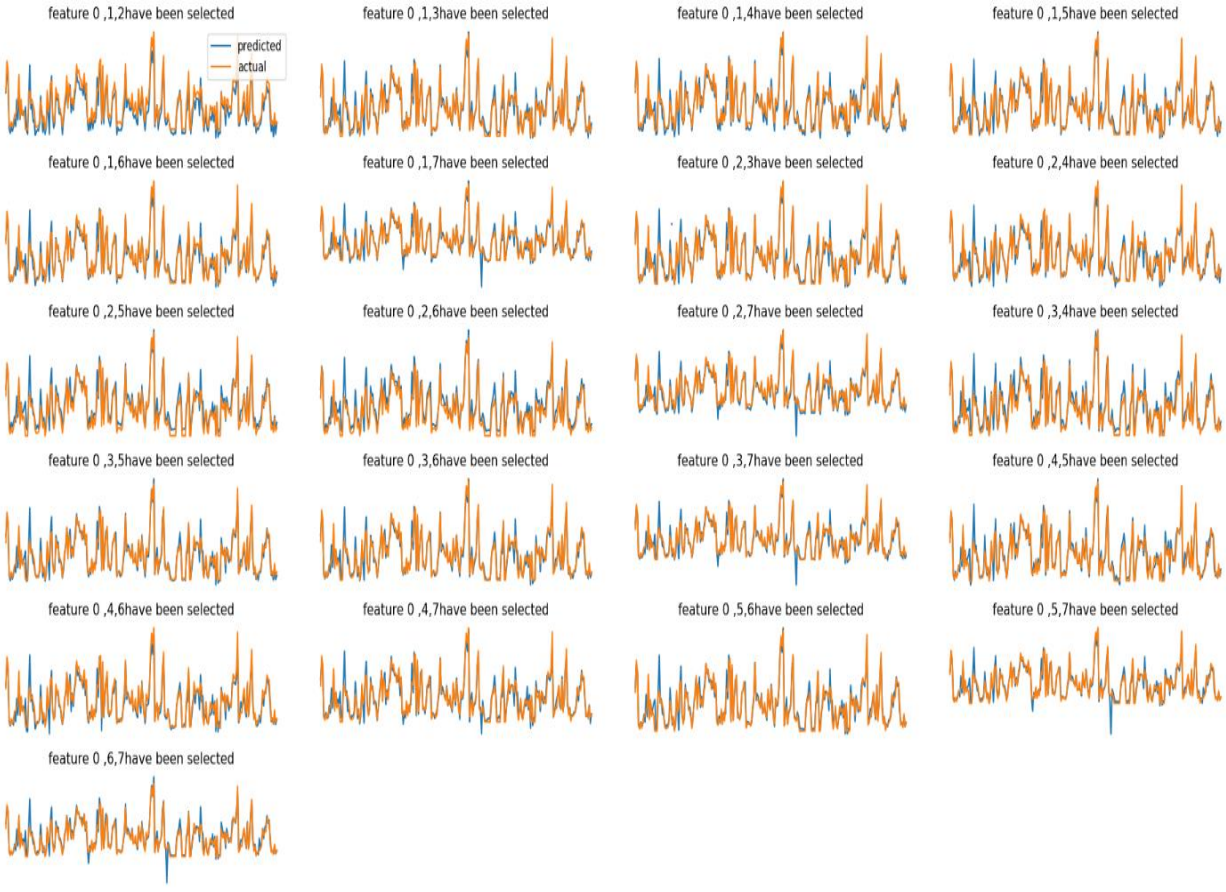
زمان مورد نیاز ۸۳۰.۰۵ ثانیه



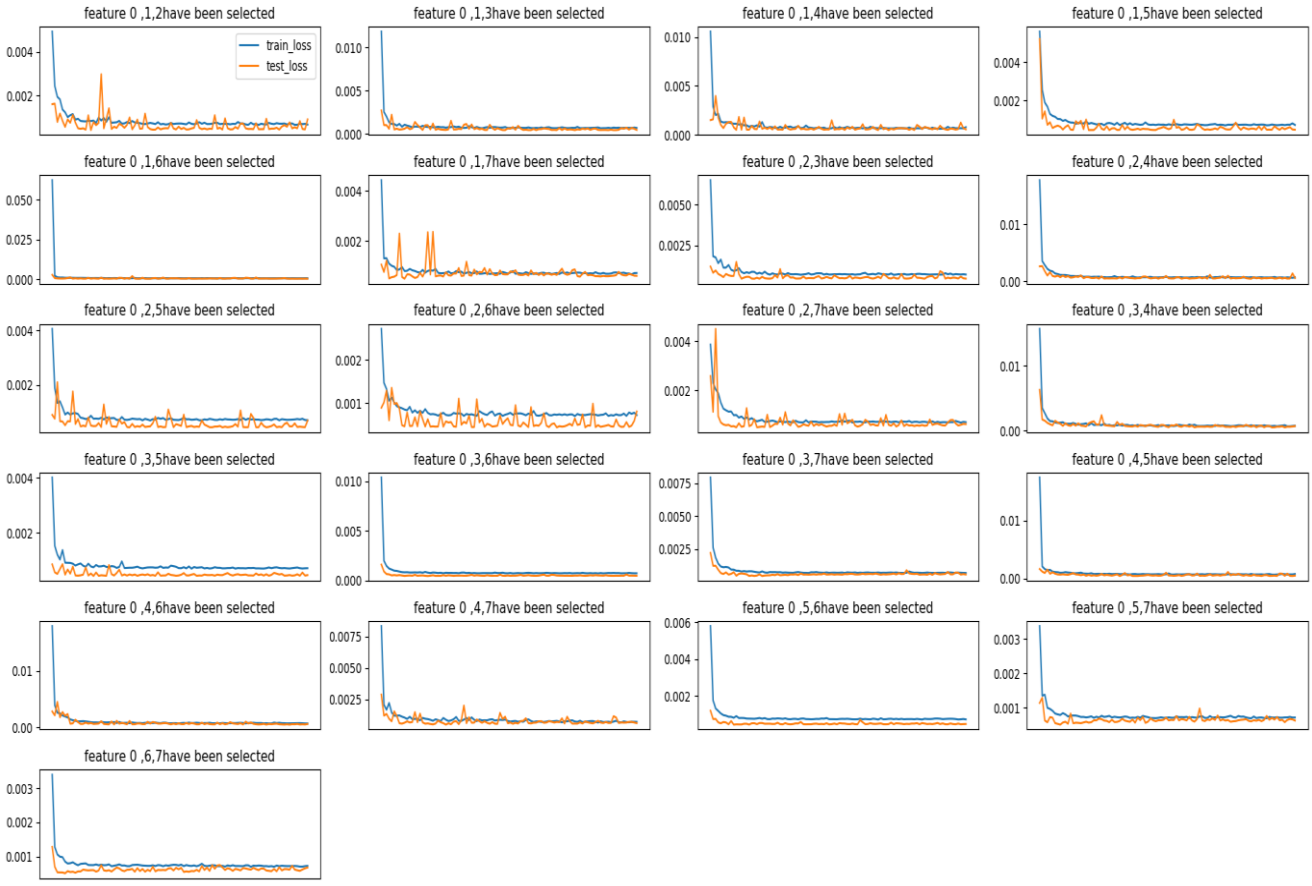
شکل۵۵) پیش‌بینی الودگی به ازای هر حالت ممکن برای شبکه LSTM



شکل ۵۶) خطای داده‌های test و train به ازای هر حالت شبکه LSTM

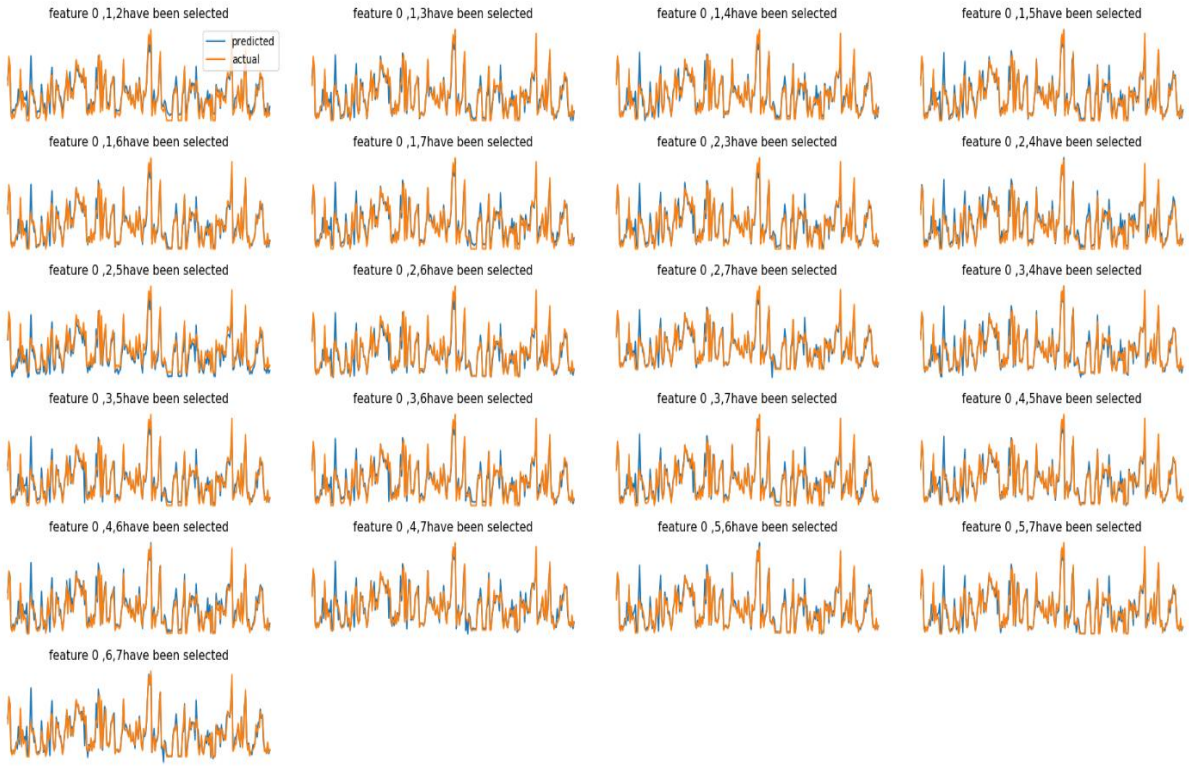


شکل ۵۷) پیش‌بینی الودگی به ازای هر حالت ممکن برای شبکه RNN

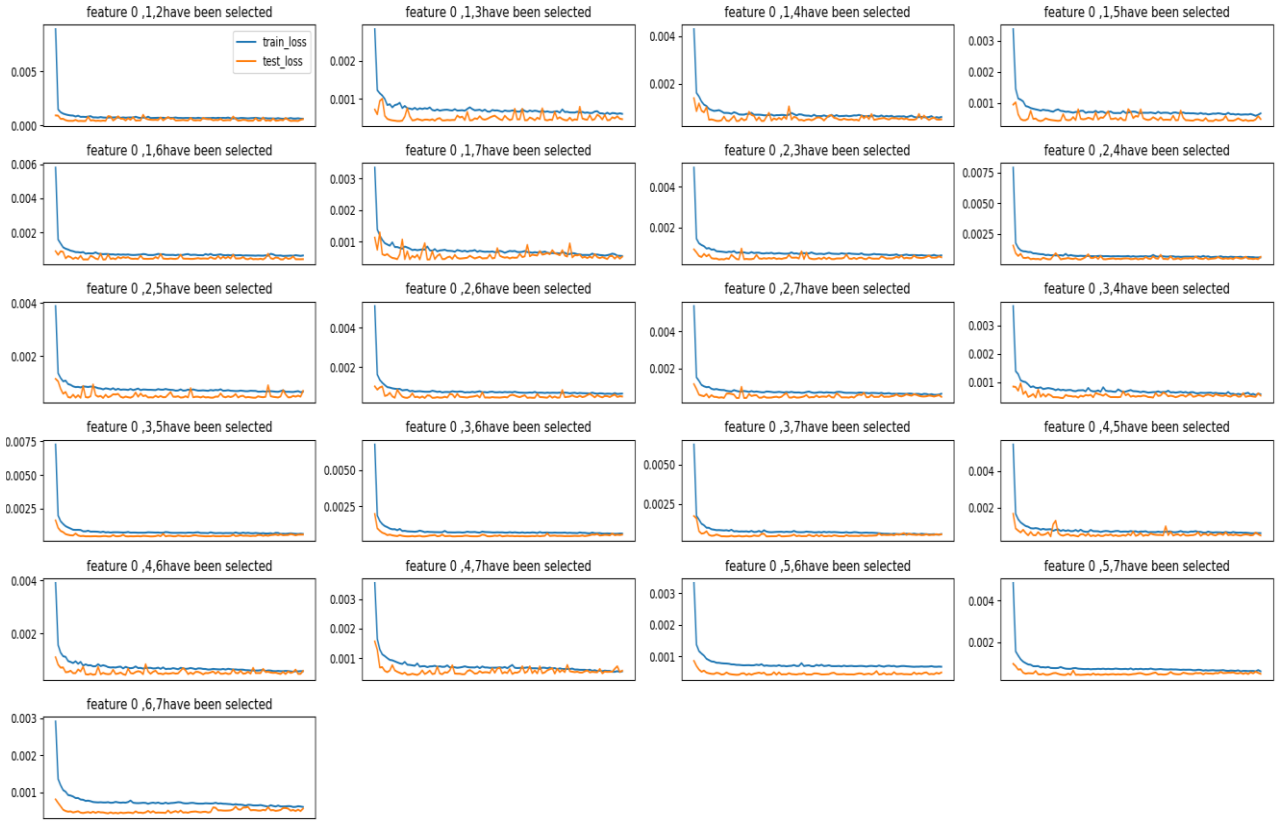


شکل۵۸) خطای داده‌های test و train به ازای هر حالت شبکه RNN

GRU



شکل ۵۹) پیش‌بینی الودگی به ازای هر حالت ممکن برای شبکه GRU



شکل ۴۰) خطای داده‌های test و train به ازای هر حالت شبکه GRU

این مراحل انتخاب سه ویژگی موثرتر را برای شبکه RNN و GRU با فرض این که میدانیم ویژگی آلودگی ثابت است انجام میدهیم و درنهایت نتیجه هر سه شبکه در جدول زیر نشان داده شده است :

شماره ۳ ویژگی انتخاب شده	LSTM	RNN	GRU
۰ , ۱ , ۲	۰/۰۲۱	۰/۰۳۰	۰/۰۲۵
۰ , ۱ , ۳	۰/۰۲۴	۰/۰۲۱	۰/۰۲۱
۰ , ۱ , ۴	۰/۰۲۲	۰/۰۲۰	۰/۰۲۲
۰ , ۱ , ۵	۰/۰۲۱	۰/۰۲۱	۰/۰۲۰
۰ , ۱ , ۶	۰/۰۲۰	۰/۰۲۱	۰/۰۲۵
۰ , ۱ , ۷	۰/۰۲۱	۰/۰۲۴	۰/۰۲۳
۰ , ۲ , ۳	۰/۰۲۱	۰/۰۲۱	۰/۰۲۹
۰ , ۲ , ۴	۰/۰۲۲	۰/۰۲۴	۰/۰۲۴
۰ , ۲ , ۵	۰/۰۲۱	۰/۰۲۶	۰/۰۲۰
۰ , ۲ , ۶	۰/۰۲۰	۰/۰۲۸	۰/۰۲۲
۰ , ۲ , ۷	۰/۰۲۳	۰/۰۲۵	۰/۰۲۱
۰ , ۳ , ۴	۰/۰۲۴	۰/۰۲۶	۰/۰۲۵
۰ , ۳ , ۵	۰/۰۲۵	۰/۰۲۱	۰/۰۲۳
۰ , ۳ , ۶	۰/۰۲۴	۰/۰۲۱	۰/۰۲۳
۰ , ۳ , ۷	۰/۰۲۲	۰/۰۲۴	۰/۰۲۲
۰ , ۴ , ۵	۰/۰۲۲	۰/۰۲۳	۰/۰۲۲
۰ , ۴ , ۶	۰/۰۲۲	۰/۰۲۴	۰/۰۲۲
۰ , ۴ , ۷	۰/۰۲۳	۰/۰۲۴	۰/۰۲۲
۰ , ۵ , ۶	۰/۰۲۱	۰/۰۲۱	۰/۰۲۲
۰ , ۵ , ۷	۰/۰۲۱	۰/۰۲۵	۰/۰۲۴
۰ , ۶ , ۷	۰/۰۲۲	۰/۰۲۶	۰/۰۲۴

مقادیر Mean Squared Error

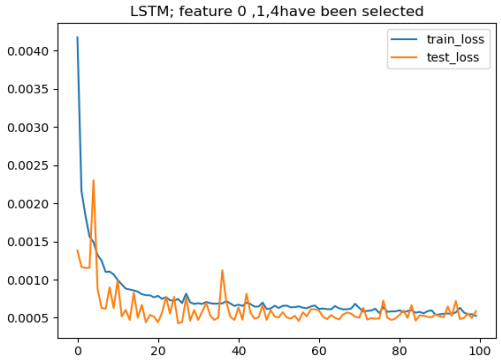
بدین ترتیب ویژگی ۰ و ۱ و ۴ را انتخاب میکنیم.(pollution, dew, wind_dir)

همچنین برای درستی انتخاب انجام شده از کتابخانه sklearn.feature_selection استفاده کردیم که همین ۳ ویژگی را انتخاب کرد.

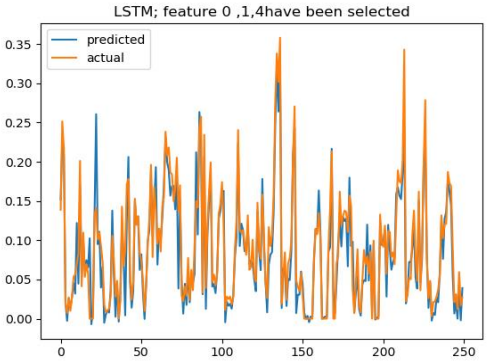
۸)

حال با سه ویژگی انتخاب شده در قسمت قبل سه شبکه LSTM , RNN , GRU را بررسی میکنیم. و با استفاده از این سه ستون میزان آلودگی روزانه را برای هر سه سلول بررسی میکنیم همچنین برای این قسمت از آپتیمایزر adam و تابع خطا mse استفاده کردیم که نتایج به صورت زیر میشود :

LSTM :

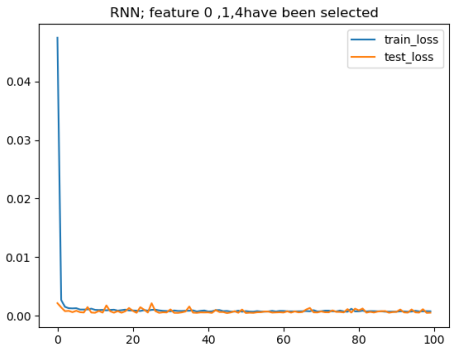


شکل۶۱)نمودار loss برای شبکه LSTM با آپتیمایزرAdam

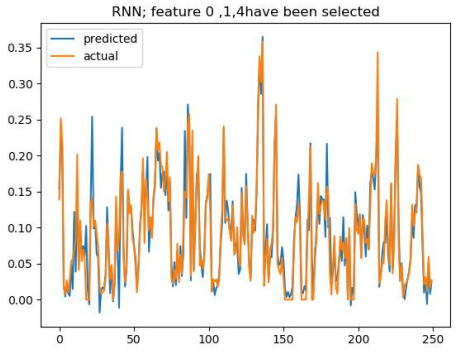


شکل۶۲)نمودار predict برای شبکه LSTM با آپتیمایزرAdam

RNN:

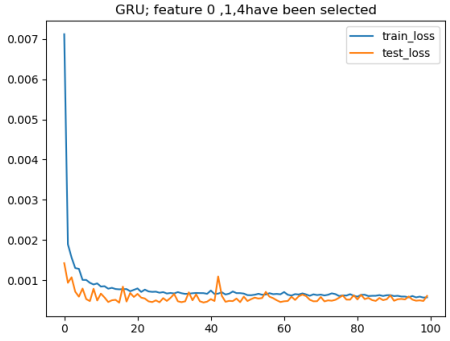


شکل۶۳)نمودار loss برای شبکه RNN با آپتیمایزرAdam

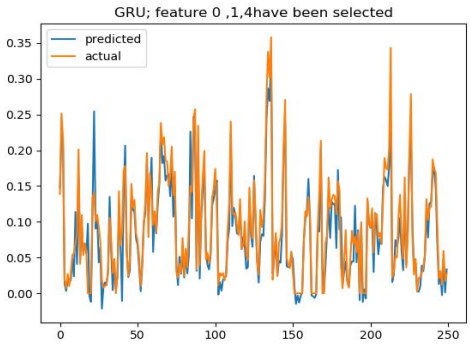


شکل۶۴)نمودار predict برای شبکه RNN با آپتیمایزرAdam

GRU:



شکل۶۵)نمودار loss برای شبکه GRU با آپتیمایزر Adam



شکل۶۶)نمودار predict برای شبکه GRU با آپتیمایزرAdam

همان‌طور که می‌بینیم با انتخاب ۳ ویژگی موثرتر می‌توانیم تقریبا به نتیجه‌ی مشابهی با زمانی که از هر ۸ ویژگی استفاده کردیم، برسیم. یعنی نمودار loss و predict مشابه می‌شوند.

سوال ۲

نقصان دادگان

یکی از مسائل مهم در شبکه های عصبی نقصان دادگان است در تحلیل داده ها گاهی برخی از مشاهدات به دلایل گوناگون و روش های متفاوت ، گمشده محسوب میشوند. چگونگی برخورد با این مشاهدات در تحلیل داده ها به دلیل اهمیت نتایج حاصل از آنها به ویژه در تصمیم گیری های حساس ، از اهمیت به سزایی برخوردار است این داده های از بین رفته گاهی برای حل مشکلات ضروری هستند بنابراین نمیتوان به سادگی این داده های مفقود شده را در مجموعه ی داده ها نادیده گرفت لذا بهتر است همیشه قبل از مدل کردن شبکه داده های گمشده (nan) را برای هر ستون در داده های ورودی خود ، شناسایی و جایگزین کنیم

یک روش برای رویارویی با این کار این است که با قسمتی از داده ها که در دسترس است مقادیر گمشده را استنتاج کرد در ابتدا برای غلبه بر مشکل داده های گمشده مرسوم ترین روش ، حذف داده های گمشده بود که منجر به داده هایی با کیفیت پایین و به تبع آن تحلیل و استخراج نتایج دارای سوگیری میشد همچنین استفاده از این حالت موجب کاهش حجم نمونه شده و باعث میشود که برآورد پارامترها اریب شودهمچنین در این روش حذف مقادیر گمشده ، میتواند به دور ریختن اطلاعات باارزش منجر شود و داده های باقیمانده نمونه ی خوبی برای کل داده ها نباشند همچنین ساده ترین و ابتدایی ترین راه در برخورد با این مسائل جایگزینی مقادیر nan با صفر است که نتیجه قابل قبولی ندارد و در پیش بینی اثر منفی دارد

۱) حذف ۲۰ درصد از دادگان برای هر ستون از قسمت دادگان آموزش

همانطور که میدانیم فهمیدن دلیل این‌که چرا داده‌ها گم‌شده‌اند برای مدیریت صحیح سایر داده‌ها مهم است. اگر داده‌ها کاملاً به‌تصادف گم‌شده باشند، نمونه ی داده‌ها احتمالاً هنوز نماینده‌ای جامعه خواهد بود. اما اگه داده‌ها به شکل سیستماتیک گم‌شده باشند. تحلیل ممکن است اریب باشد . در این قسمت ابتدا برای هرستون بطور مجزا ۲۰درصد از داده ها را nan قرار میدهیم تا در مراحل بعد از این داده های گمشده در پیش بینی استفاده کنیم .

```
> 0, Missing: 2229 (20.3%)
> 1, Missing: 2219 (20.2%)
> 2, Missing: 2234 (20.3%)
> 3, Missing: 2200 (20.0%)
> 4, Missing: 2229 (20.3%)
> 5, Missing: 2220 (20.2%)
> 6, Missing: 2214 (20.1%)
> 7, Missing: 2231 (20.3%)
```

۲) اتخاذ روشی برای حل نقصان دادگان

امروزه با پیشرفت های علمی در حوزه های گوناگون و پیدایش روش های توانمند آماری ، میتوان پیش از مدلسازی داده های ناکامل ، مقادیر گمشده را با مقادیر مناسب جایگذاری یا براورد کرد روش های گوناگونی برای این حل نقصان دادگان وجود دارد که در قسمت بعد مفصل توضیح داده خواهد شد .

۳) شرح ۳ روش برطرف کردن نقصان دادگان

الگوریتم EM

این الگوریتم را در اواخر دهه ۱۹۷۰روبین ، دمپستر و لارد معرفی کردند و توسعه دادند . با توجه به اینکه در مقادیر مشاهده شده اطلاعاتی در خصوص احتمال مقادیر گمشده وجود دارد ، این الگوریتم از سایر متغیر ها برای جایگذاری مقدار گمشده در یک متغیر استفاده میکند و بررسی میکند که آیا این مقدار محتمل ترین مقدار است و اگر نباشد مقدار دیگری جایگزین میشود و این روند تا رسیدن به محتمل ترین مقدار ادامه پیدا میکند . انتخاب نام EM نیز به علت یک مرحله امید ریاضی گیری و سپس ماکسیمم سازی در تکرار الگوریتم است . این الگوریتم از داده های کامل برای محاسبه ی میانگین، واریانس و کواریانس استفاده میکند . پس از آن برای بدست آوردن خطوط رگرسیون ارتباط هر متغیر به سایر متغیر ها ، روش ماکسیمم درست نمایی (ML) بکار میرود . در این مرحله به تعداد متغیر ها، معادله خواهیم داشت .ML این اطمینان را به ما میدهد که این معادله ها دقیق ترین میانگین ، واریانس و کواریانس را ارائه میدهند. با استفاده از این معادلات مقادیر گمشده برآورد میشوند و مجموعه داده های ما در این مرحله کامل می شود . سپس با استفاده از این مجموعه داده کامل ، دوباره میانگین ، واریانس و کواریانس برآورد میشوند که ممکن است با مقادیر قبلی کمی متفاوت باشند، چرا که در این مرحله با استفاده از مجموعه داده های کامل براورد شده اند . دوباره معادلات رگرسیون با استفاده از ML محاسبه میشوند و مجددا

مقادیر برآورد میشوند . این سه مرحله تا رسیدن به همگرایی تکرار میشوند . با جایگذاری مقادیر گمشده با استفاده از EM ارتباط بین متغیر ها حفظ میشود . یکی از پیچیدگی های الگوریتم EM آن است که نیازمند مدل بندی پارامترهای مزاحم متغیرهای کمکی است .در برخی مواقع فقط با تعدلد کمی از متغیرهای طبقه ای توزیع چند جمله ای اشباع شده میتواند برازش داده شود . زمانی که متغیرهای بیش تری وجوددارند ، اغلب انجام برخی از ساده سازی ها برای توزیع توام ضروری است .

الگوریتم داده افزایی :

این روش نیز مانند الگوریتم EM محاسبه ای تکراری است که به طور متناوب داده های گمشده را جای گذاری میکند و سپس پارامترهای ناشناخته را با روندی تصادفی پیش بینی میکند در این روش نخست جایگذاری ابتدایی برای داده های گمشده براساس مقادیر فرضی پارامترها در نظر گرفته میشود سپس پارامترهای جدید با استفاده از توزیع پسین بدست آمده از داده های کامل برآورد میشود . فرآیند شبیه سازی پارامترها و داده های گمشده یک زنجیر مارکف تولید میکند که سرانجام تثبیت شده یا در توزیع همگرا میشود . این روش میتواند با تکرار مراحل زیر حاصل شود :

گام I ام جانهی - (جانهی به پرکردن جای داده‌های گم‌شده می‌گویند) : با بردار میانگین و ماتریس کواریانس برآورد شده مقادیر گمشده برای هر مشاهده به طور مستقل شبیه سازی میشوند . به این معنی که اگر شما متغیرهای با مقادیر گمشده را برای مشاهده ی i ام با $Y_{i(mis)}$ نشان دهید و متغیرهای با مقادیر مشاهده شده را با $Y_{i(obs)}$ ، بنابراین مرحلهی اول ، مقادیر را برای $Y_{i(mis)}$ از توزیع شرطی $Y_{i(mis)}\left|Y_{i(obs)}\right.$ تولید میکند .

گام p ام پسین : بردار میانگین و ماتریس کواریانسپسین جامعه از برآوردهای نمونه کامل شبیه سازی شده اند . سپس این برآوردهای جدید در گام I استفاده میشوند . بدون اطلاع قبلی درباره ی پارامترها ، یک پیشین ناآگاهی بخش استفاده میشود .

این دو مرحله به اندازه ی کافی برای نتایج قابل اعتماد برای مجموعه داده های جانهی چندگانه موثر هستند . اغلب تعداد کمی از جانهی ها در جانهی چندگانه مناسب هستند .

روش حذف داده گمشده

این روش در گذشته مرسوم ترین روش برخوردبا داده گمشده بوده است . اما استفاده از آن موجب کاهش حجم نمونه شده و باعث میشود که براورد پارامترها اریب شود . همچنین در این روش حذف مقادیر گمشده میتواند به دور ریختن اطلاعات با ارزش منجر شود و داده های باقیمانده نمونه ی خوبی برای کل داده ها نباشند بنابراین از روش های بهتری مانند استفاده از میانگین گیری و روش های آماری استفاده میکنیم .

۴) پیش بینی دادگان از بین رفته با یک روش دلخواه

بهتر است که مقادیر گمشده برای هر ستون در داده های ورودی خود را قبل از مدل کردن کار پیش بینی، شناسایی و جایگزین کنیم همانطور که قبل از این سوال نیز ذکر شد راه های متفاوتی برای این کار وجود دارد یک رویکرد رایج برای تعویض داده ها محاسبه ی یک مقدار آماری (مانند میانگین یا میانه) برای هر ستون و جایگزین کردن تمام مقادیر گمشده برای آن ستون با آمار است این کار بدیل اینکه اغلب منجر به عملکرد خوب میشود و همچنین بدلیل راحتی یک رویکرد رایج است همانطور که در قسمت اول نشان دادیم برای هر ستون از داده های آموزش ۲۰ درصد از داده ها را به طور تصادفی nan میکنیم حال در این قسمت با رویکرد آماری مقادیر nan را پر میکنیم و در نهایت برای آموزش شبکه در قسمت های بعدی از این داده ها استفاده میکنیم .

۵) میزان خطای موارد پیش بینی شده برای دادگان از دست رفته برای هر ستون با استفاده از روش خطای MSE

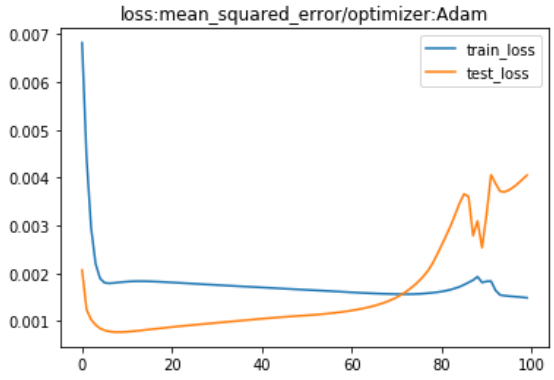
شکل زیر مقدار خطای پیش بینی شده برای دادگان از دست رفته برای هر ستون را نشان میدهد . که از روش میانگین گیری استفاده کرده ایم :

```
RMSE for column1 data is : 0.043
RMSE for column2 data is : 0.093
RMSE for column3 data is : 0.091
RMSE for column4 data is : 0.081
RMSE for column5 data is : 0.131
RMSE for column6 data is : 0.046
RMSE for column7 data is : 0.017
RMSE for column8 data is : 0.019
```

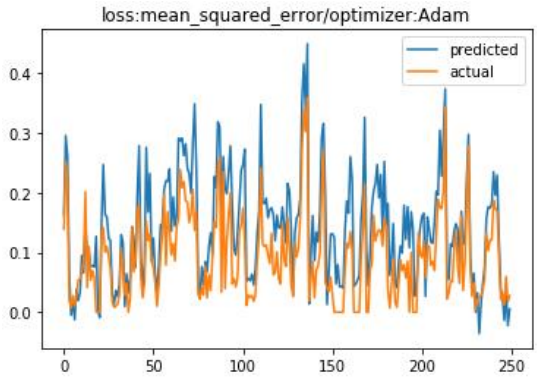
همانطور که مشاهده میشود ستون هفتم کمترین خطا و ستون پنجم بیشترین خطا را دارد .

۶) پیش بینی میزان آلودگی روزانه با استفاده از دادگان پیش بینی شده برای سلول های GRU و LSTM, و مقایسه نتایج با دادگان بی نقص

حال با داده های گمشده شبکه را پیش بینی میکنیم شکل زیر میزان آلودگی هوا برای شبکه GRU با بهینه ساز Adam و تابع خطای mse را نشان میدهد .



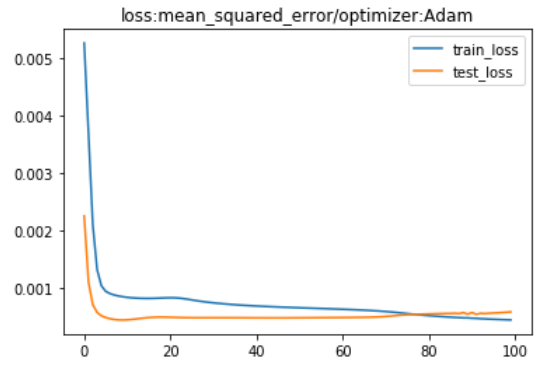
شکل۶۷) نمودار loss دادگان miss شده در شبکه gru



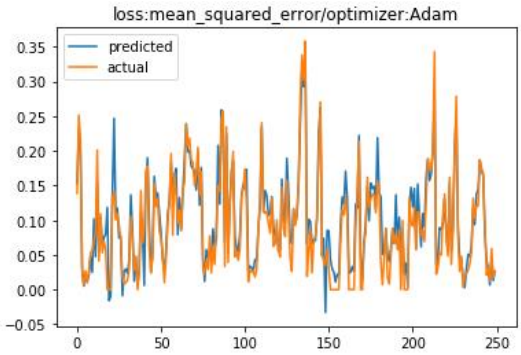
شکل۶۸) نمودار predict دادگان miss شده در شبکه gru

```
Epoch 100/100
 - 1s - loss: 0.0015 - val_loss: 0.0041
GRU learning time( mean_squared_error and Adam ) is:100.15
RMSE for test data is : 0.064
```

حال نتایج را با دادگان بی نقص مقایسه میکنیم .



شکل۶۹) نمودار loss دادگان بی نقص در شبکه gru

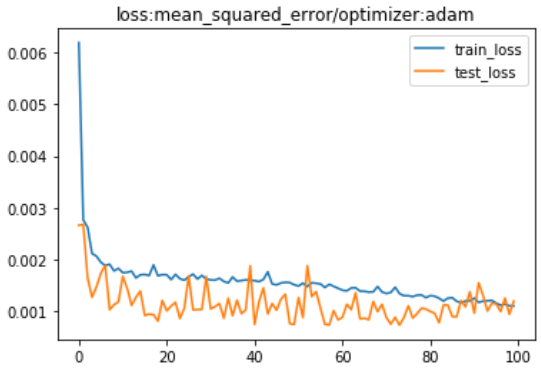


شکل ۷۰) نمودار predict دادگان بی نقص در شبکه gru

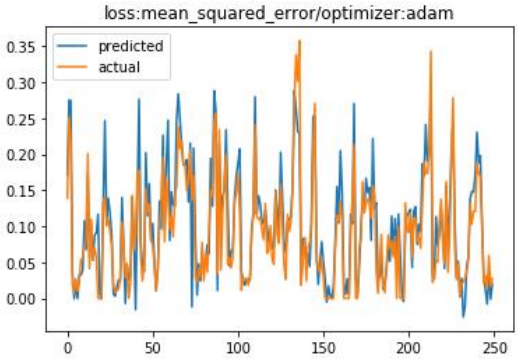
```
Epoch 100/100
 - 1s - loss: 4.4953e-04 - val_loss: 5.9043e-04
GRU learning time( mean_squared_error and Adam ) is:83.67
```

همانطور که مشاهده میشود در مقایسه نتایج با دادگان بی نقص که در سوال اول بررسی شد زمان آموزش افزایش یافته و همچنین خطا نیز بیشتر میشود همانطور که در نمودار خطا مشاهده میکنیم برای دادگان گمشده در ایپاکهای نهایی فاصله نمودار تست و ترین افزایش میابد که این افزایش ممکن است منجر به اورفیتینگ نیز بشود در حالی که در دادگان بی نقص همچین مشکلی وجود ندارد نمودار پیش بینی نیز همانطور که به وضوح مشخص است نتایج بهتری را نشان میدهد .

حال شبکه lstm را با دادگان گمشده مورد بررسی قرار میدهیم



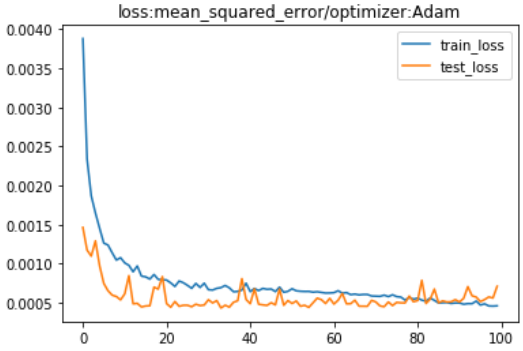
شکل۷۱) نمودار loss دادگان miss شده در شبکه lstm



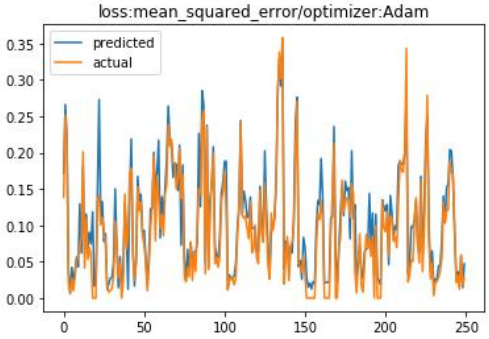
شکل۷۲) نمودار predict دادگان miss شده در شبکه lstm

```
Epoch 100/100
 - 2s - loss: 0.0011 - val_loss: 0.0012
LSTM learning time( mean_squared_error and adam ): 184.23953890800476
RMSE for test data is : 0.035
```

و در پایان نتایج آن را با دادگان بی نقص مقایسه میکنیم :



شکل۷۳) نمودار loss دادگان بی نقص در شبکه lstm



شکل۷۴) نمودار predict دادگان بی نقص در شبکه lstm

```
Epoch 100/100
 - 1s - loss: 4.5905e-04 - val_loss: 7.0906e-04
LSTM learning time( mean_squared_error and Adam ): 120.90427303314209
RMSE for test data is : 0.027
```

برای شبکه lstm نیز نیز زمان آموزش و همچنین مقدار خطای شبکه افزایش یافته است . ولی با مقایسه نمودار خطای آنها با یکدیگر مشاهده میکنیم که در هر دوحالت خطای تست شامل نوساناتی است ولی در ایپاک های نهایی نمودار تست و ترین به هم خیلی نزدیک میشوند نمودار پیش بینی نیزدر هر دوحالت خوب عمل کرده است و در حالت کلی این شبکه نسبت به gru در مقابل دادگان گمشده بهتر عملکرده است .