به نام خدا

دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر

شبکه های عصبی و یادگیری عمیق

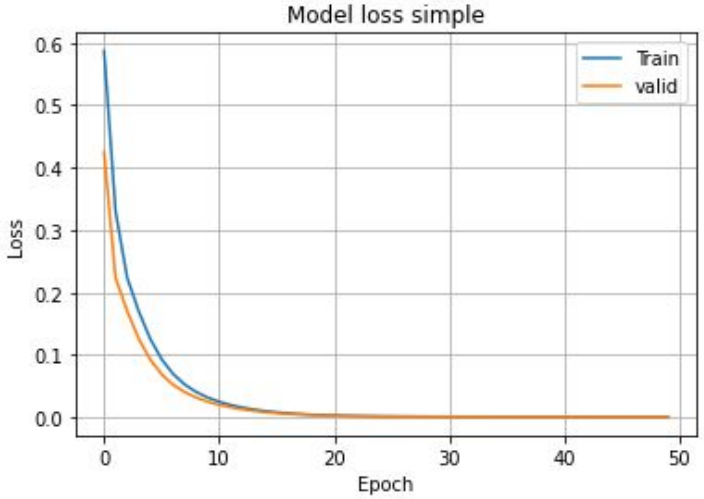
مینی پروژه۲

نام و نام خانوادگی	فاطمه حقیقی سجاد پاکدامن ساوجی
شماره دانشجویی	۸۱۰۱۹۵۳۸۵ ۸۱۰۱۹۵۵۱۷
تاریخ ارسال گزارش	۱۳ خرداد

فهرست گزارش سوالات

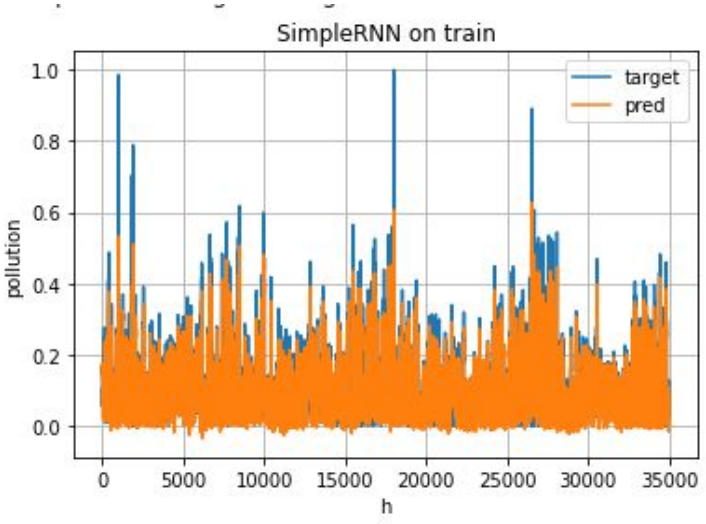
سوال ۱.۱-

در این قسمت پس از طراحی شبکه، داده های train یا آموزشی را به شبکه داده و شبکه را با این داده ها آموزش دادیم. نمودار loss به ازای داده های train و validation در این شبکه به صورت زیر می باشد:

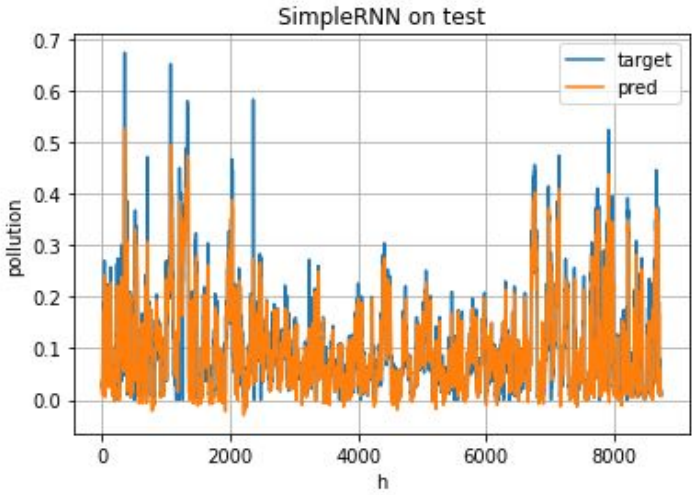


شکل ۱: نمودار loss برای داده های train و validation

پس از آن به وسیله ی شبکه طراحی شده خروجی شبکه یا مقدار تخمین زده شده توسط شبکه به ازای داده های test و train بدست آوردیم. نمودار های مقدار تخمینی و واقعی برای این دو دسته داده به صورت زیر می باشد:



شکل ۲: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی

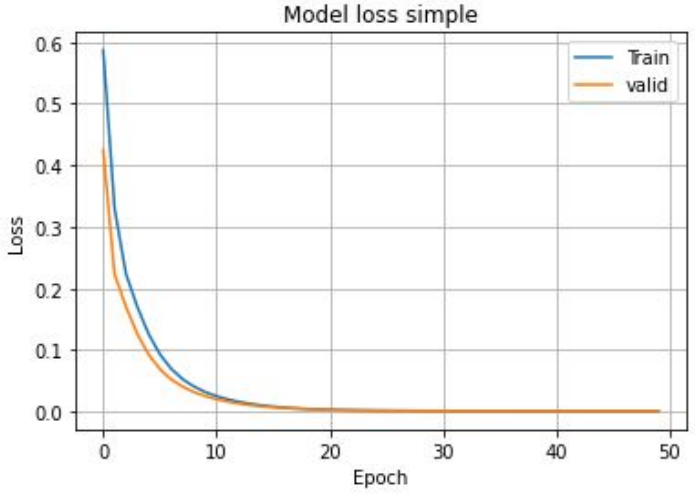


شکل ۳: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های تست

سوال ۱.۲ –

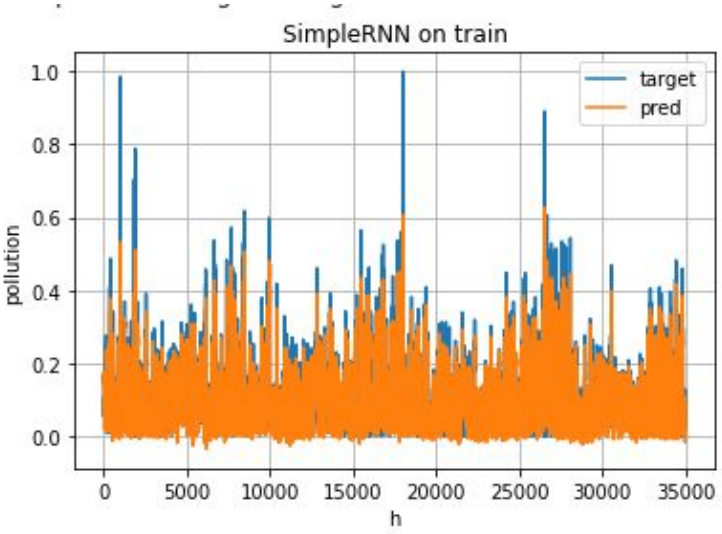
در شبکه ی RNN:

نمودار loss برای داده های trainو (validation(test به صورت زیر می باشد:



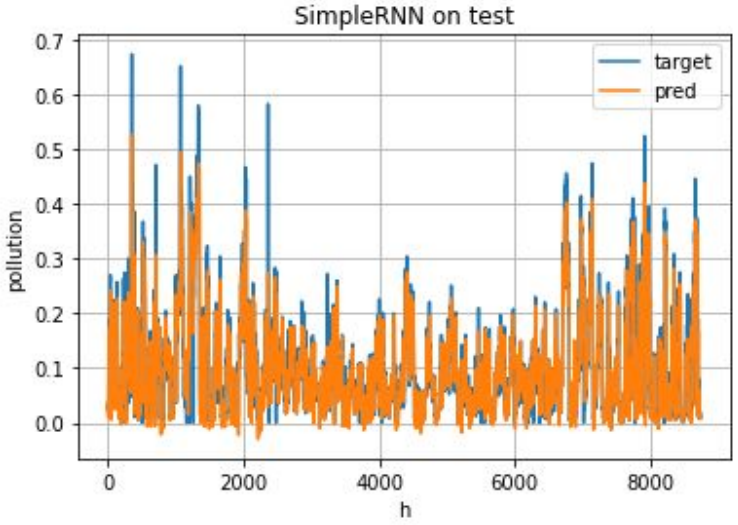
شکل ۴: نمودار loss برای داده های train و validation در شبکه RNN

نمودار مقادیر واقعی و مقادیر تخمین زده شده برای داده های آموزشی به صورت زیر می باشد:



شکل ۵: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی در شبکه ی RNN

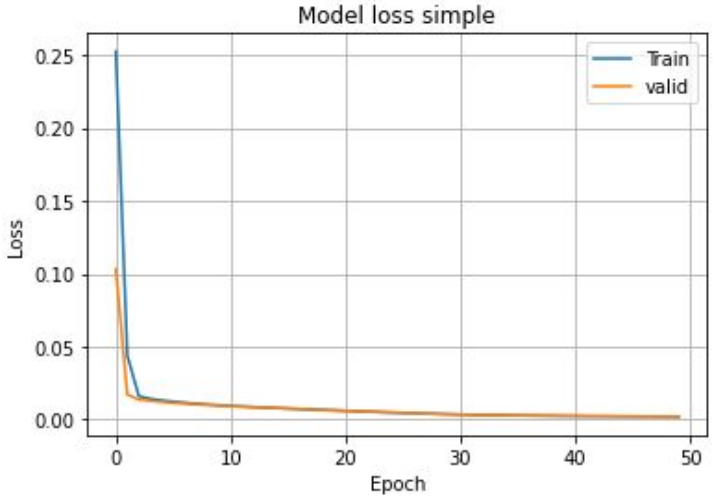
نمودار مقادیر واقعی و مقادیر تخمین زده شده برای داده های تست به صورت زیر می باشد:



شکل ۶: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های تست در شبکه ی RNN

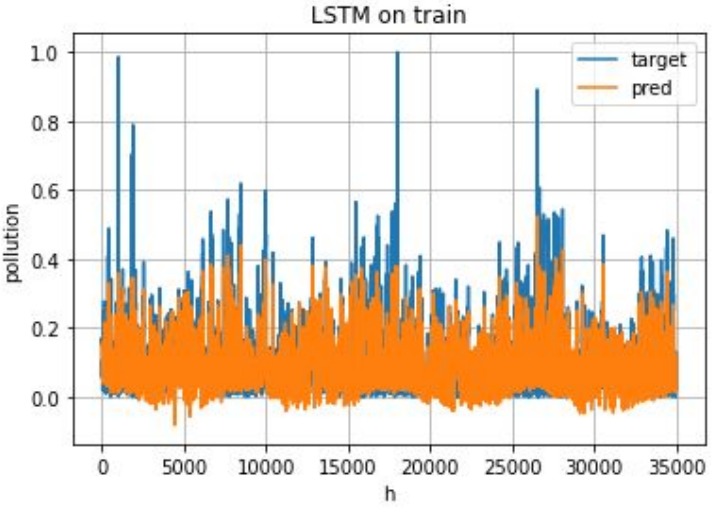
در شبکه ی LSTM:

نمودار loss برای داده های trainو validation(test) به صورت زیر می باشد:



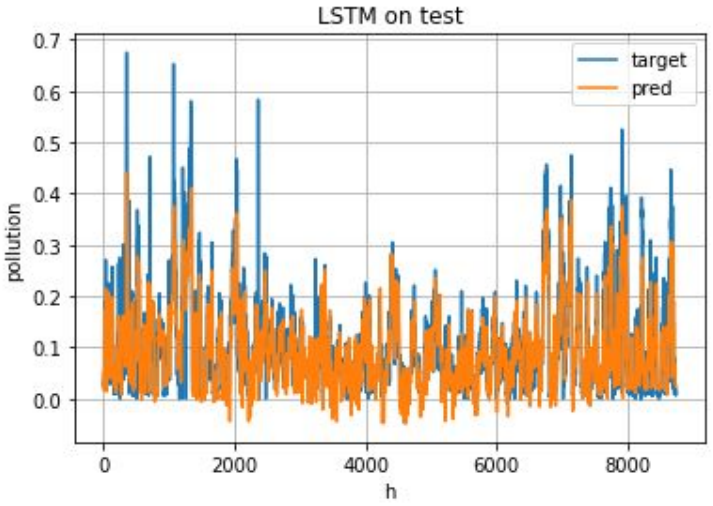
شکل ۷: نمودار loss برای داده های train و validation در شبکه LSTM

نمودار مقادیر واقعی و مقادیر تخمین زده شده برای داده های آموزشی به صورت زیر می باشد:



شکل ۸: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی در شبکه ی LSTM

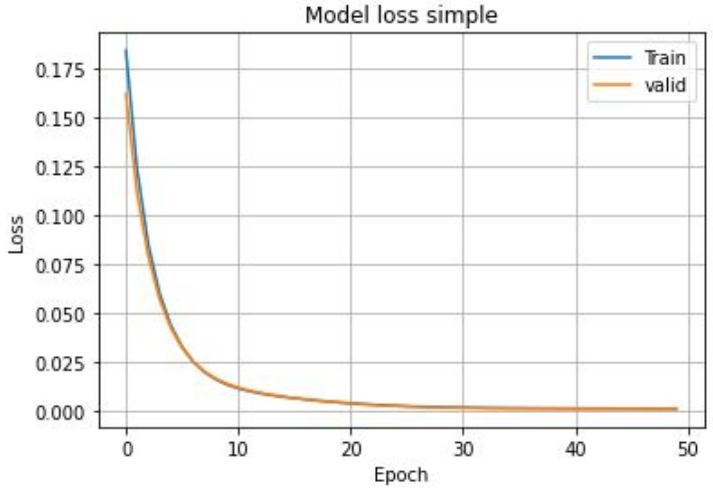
نمودار مقادیر واقعی و مقادیر تخمین زده شده برای داده های تست به صورت زیر می باشد:



شکل ۹: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های تست در شبکه ی LSTM

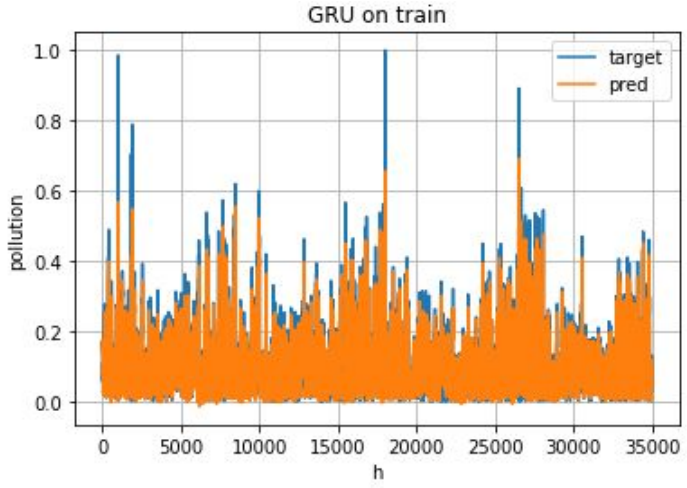
در شبکه ی GRU:

نمودار loss برای داده های trainو validation(test) به صورت زیر می باشد:



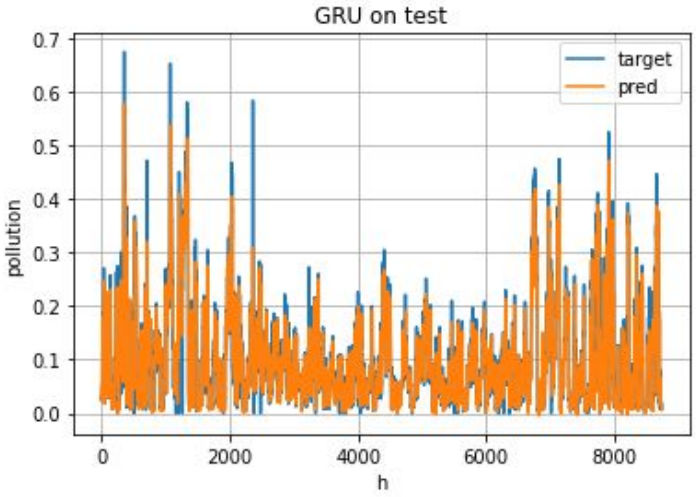
شکل ۱۰: نمودار loss برای داده های train و validation در شبکه GRU

نمودار مقادیر واقعی و مقادیر تخمین زده شده برای داده های آموزشی به صورت زیر می باشد:



شکل ۱۱: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی در شبکه ی GRU

نمودار مقادیر واقعی و مقادیر تخمین زده شده برای داده های تست به صورت زیر می باشد:



شکل ۱۲: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های تست در شبکه ی GRU

در صورتی که بخواهیم سرعت سه شبکه ی ذکر شده را با یکدیگر مقایسه کنیم، با توجه به نتیجه ی آموزش بر روی هر شبکه در کد موجود است می توان گفت سرعت شبکه ی RNN از شبکه ی GRU بیشتر و سرعت شبکه ی GRU از شبکه ی LSTM بیشتر می باشد.

برای مقایسه ی دقت سه شبکه ی ذکر شده با توجه به میزان خطا در هر ایپاک برای داده های validation می توان گفت دقت شبکه ی LSTM از شبکه ی GRU بیشتر و دقت شبکه ی GRU از شبکه ی RNN بیشتر است.

در شبکه ی LSTM به نسبت شبکه ی RNN، ساختارهایی داریم که به واسطه ی آن ها بهتر می توان جریان کار را کنترل کرده و به ازای هر وزن آموزش دیده ورودی ها را با یکدیگر ترکیب کنیم. پس توانایی کنترل در شبکه ی LSTM بیشتر است و نتیجه ی بهتری را به ما می دهد اما در آن پیچیدگی و هزینه ی مربوط به محاسبات بیشتر می باشد. بنابراین با توجه به این نکات می توان بیشتر بودن دقت LSTM از RNN و کمتر بودن سرعت LSTM از RNN را توجیه کرد.

شبکه ی GRU مدل ساده تری از شبکه ی LSTM می باشد اما از شبکه ی RNN قوی تر است به عبارت دیگر با توجه به آنکه این مدل (همانند مدل LSTM) پارامترهای بیشتری نسبت به RNNدارد، سرعت آن از RNNکمتر است اما چون مدل ساده تری از LSTMاست ، سرعتش نیز از آن بیشتر است. با همین منطق می توان توجه کرد که دقت شبکه ی GRU از شبکه ی LSTM کمتر و از شبکه ی RNNبیشتر می باشد.

سوال ۱.۳ -

بررسی اثرات مربوط به optimizerها و loss function های مختلف را بر روی شبکه ی RNN

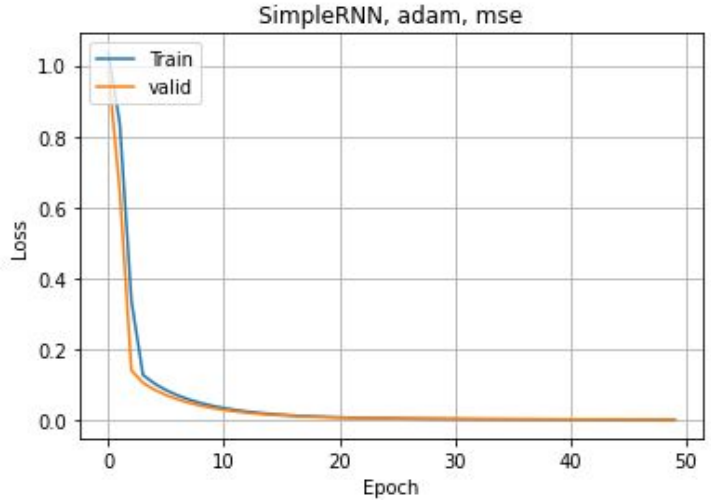
در صورتی که در این شبکه از optimizer adam استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

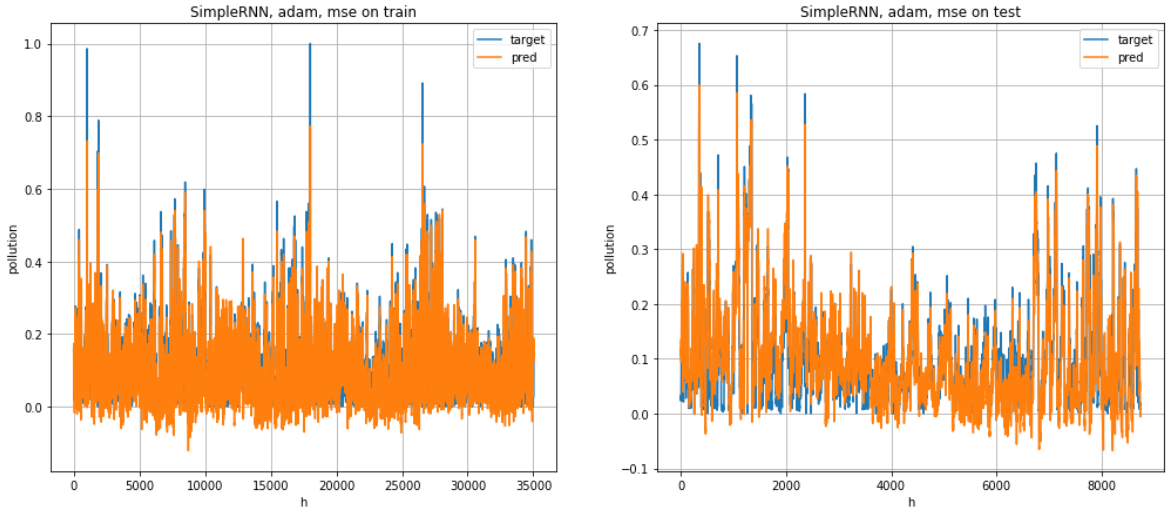
mse on train: 606.8964847654131

mse on test: 155.3268339436942

و نیز نمودارهای بدست آمده به ازای loss function = mse , optimizer = adam، برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۱۳: نمودار loss برای داده های train و validation در شبکه RNN



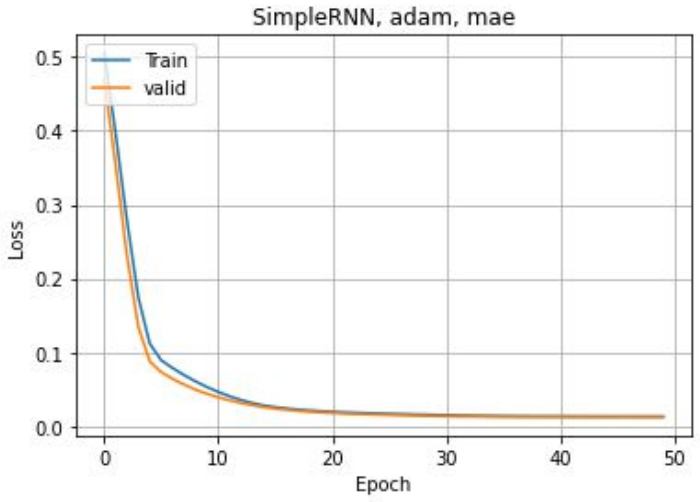
شکل ۱۴: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی RNN

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

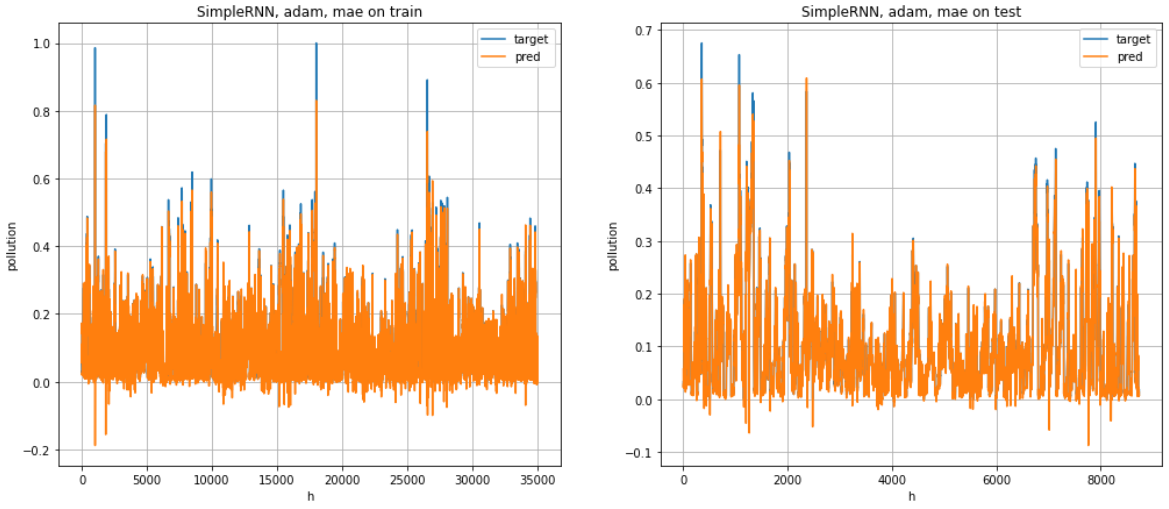
mse on train: 590.4975584682601

mse on test: 152.46882867933166

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = adam، برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۱۵: نمودار loss برای داده های train و validation در شبکه RNN



شکل ۱۶: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی RNN

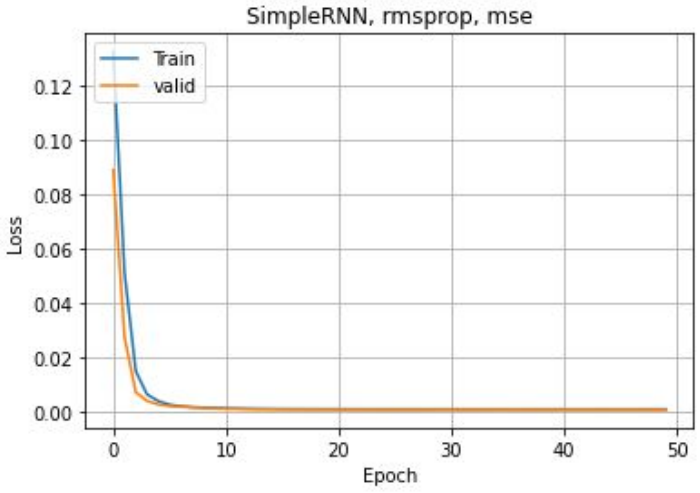
در صورتی که در این شبکه از optimizer RMSProp استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

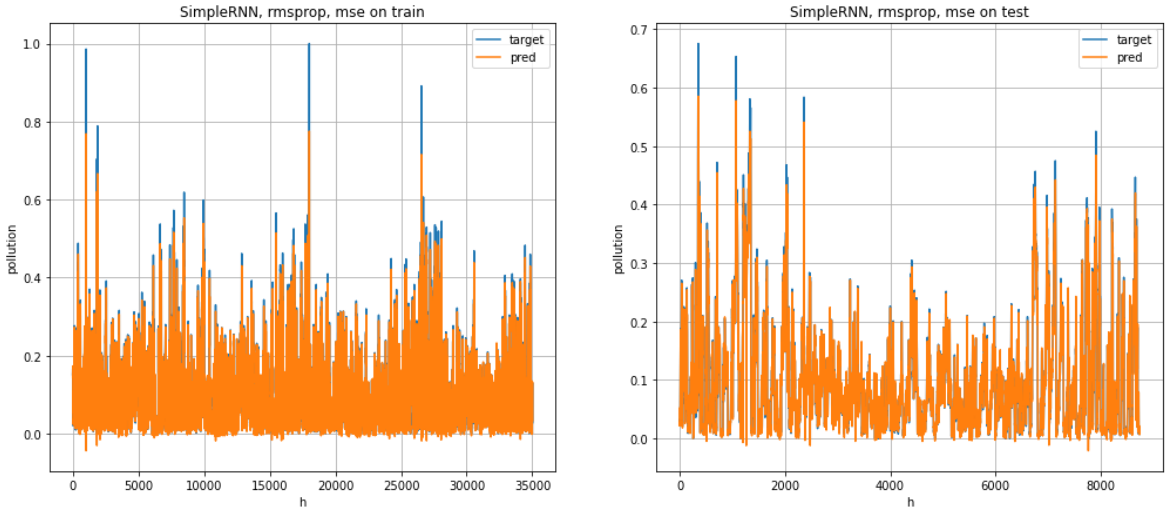
mse on train: 574.619206716354

mse on test: 148.58940833142594

و نیز نمودارهای بدست آمده به ازای loss function = mse , optimizer = RMSProp، برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۱۷: نمودار loss برای داده های train و validation در شبکه RNN



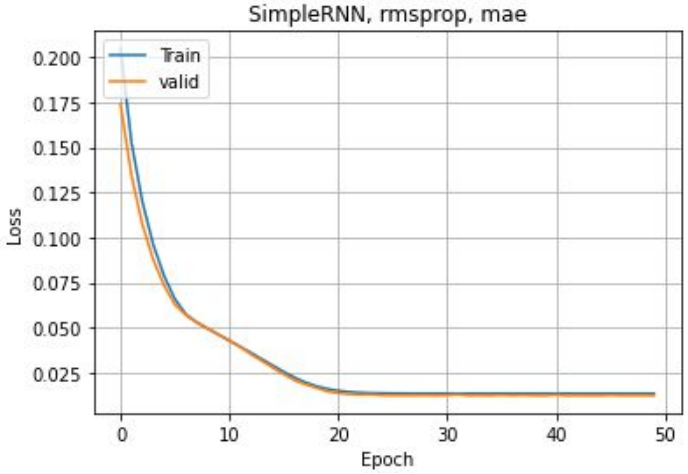
شکل ۱۸: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی RNN

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

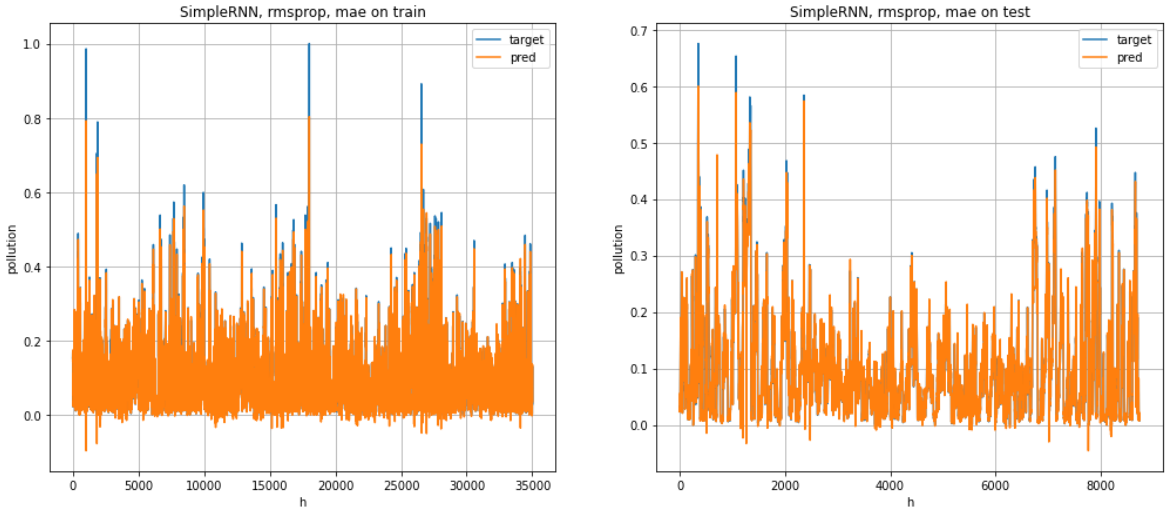
mse on train: 584.5772866158371

mse on test: 151.08656157015335

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = RMSProp، برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۱۹: نمودار loss برای داده های train و validation در شبکه RNN



شکل ۲۰: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی RNN

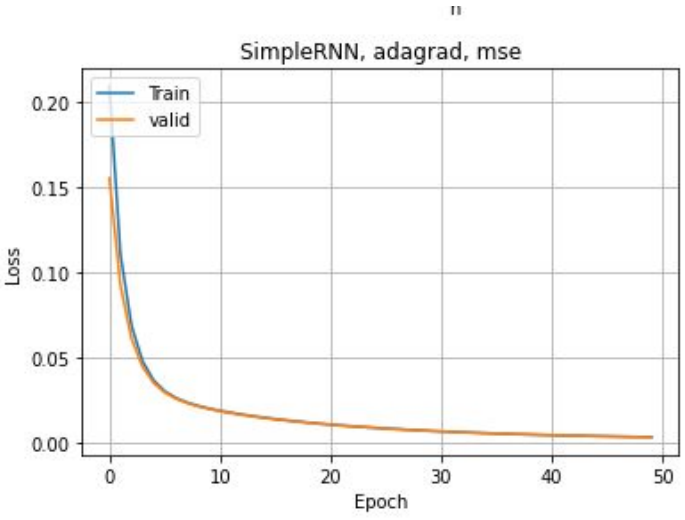
در صورتی که در این شبکه از optimizer ADAgrad استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

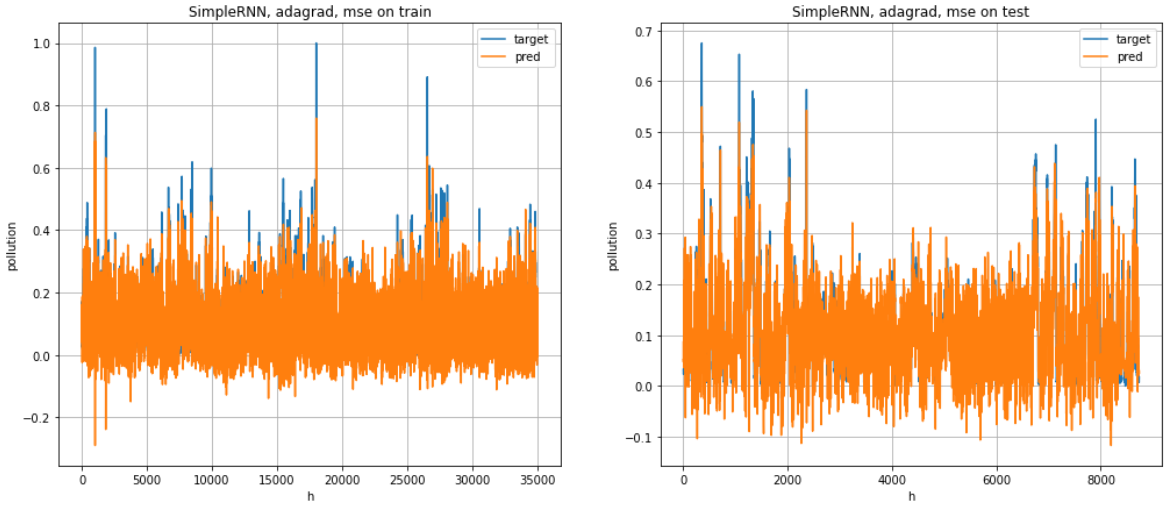
mse on train: 511.6192352722808

mse on test: 132.13997768368048

و نیز نمودارهای بدست آمده به ازای loss function = mse , optimizer = ADAgrad، برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۲۱: نمودار loss برای داده های train و validation در شبکه RNN



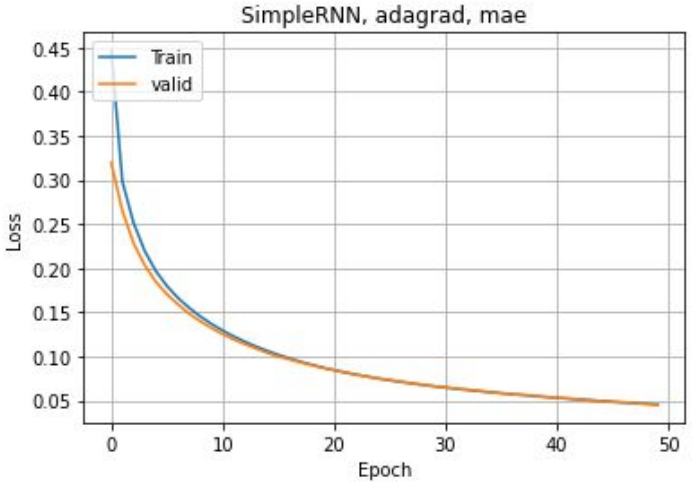
شکل ۲۲: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی RNN

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

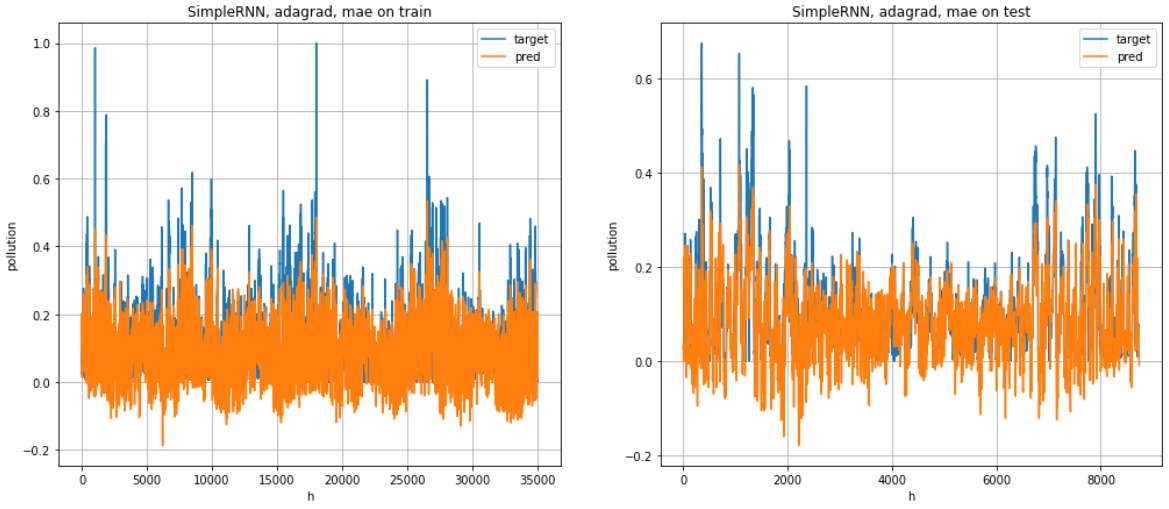
mse on train: 555.4857077586453

mse on test: 144.95363641565575

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = ADAgrad، برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۲۳: نمودار loss برای داده های train و validation در شبکه RNN



شکل ۲۴: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی RNN

بررسی اثرات مربوط به optimizerها و loss function های مختلف را بر روی شبکه ی LSTM

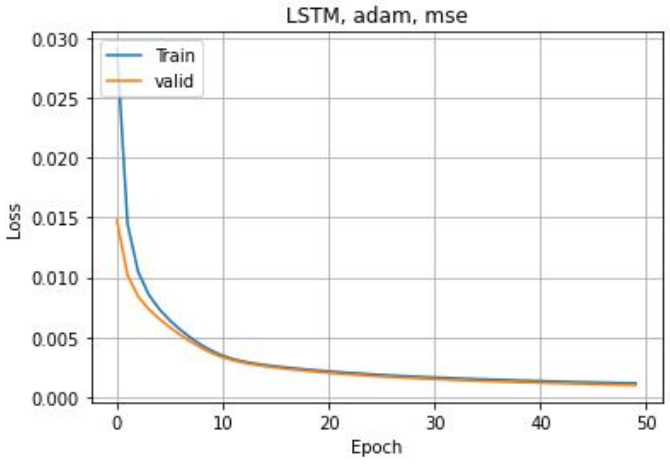
در صورتی که در این شبکه از optimizer adam استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

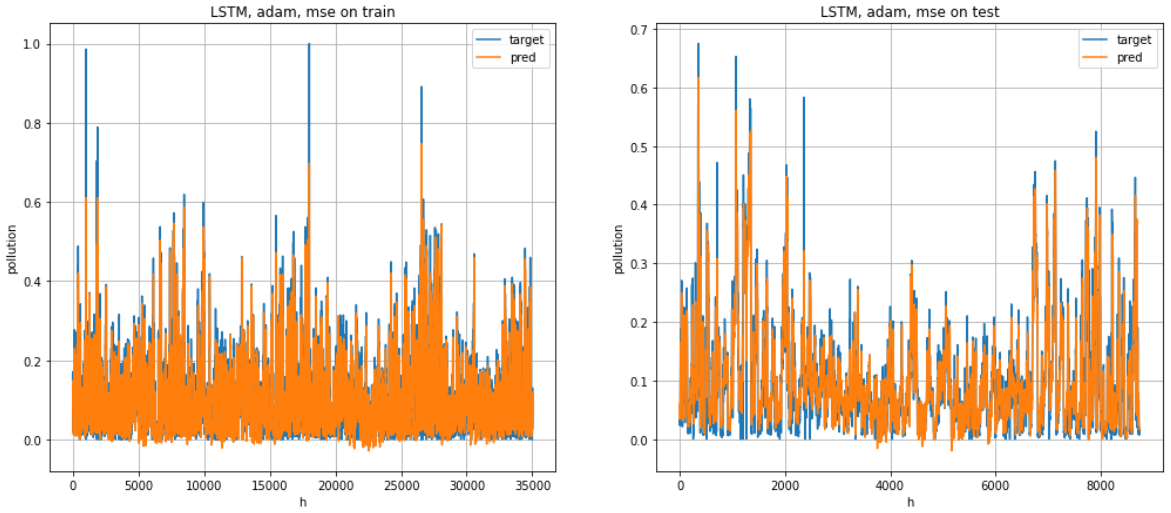
mse on train: 547.1483480396356

mse on test: 140.48761158159763

و نیز نمودارهای بدست آمده به ازای loss function = mse , optimizer = adam ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۲۵: نمودار loss برای داده های train و validation در شبکه LSTM



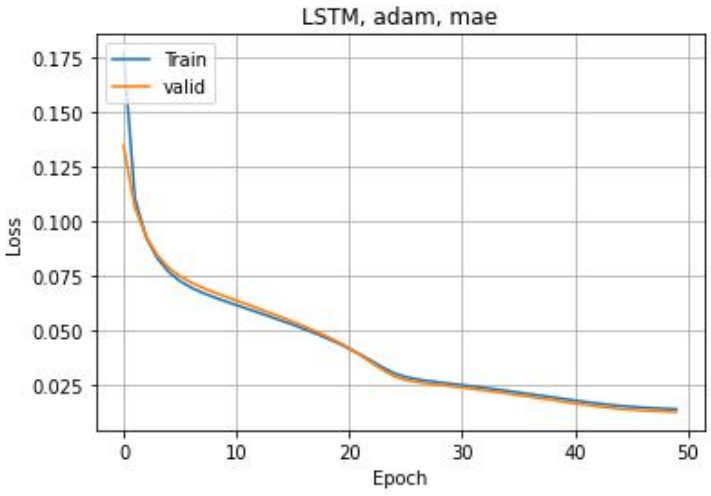
شکل ۲۶: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی LSTM

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

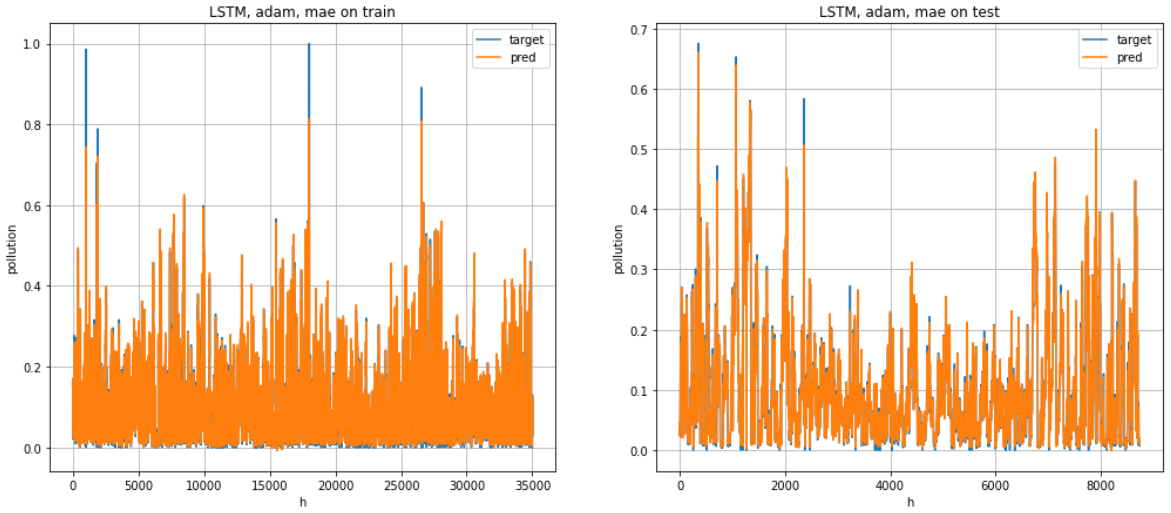
mse on train: 595.3684571232759

mse on test: 154.46980144197758

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = adam ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۲۷: نمودار loss برای داده های train و validation در شبکه LSTM



شکل ۲۸: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی LSTM

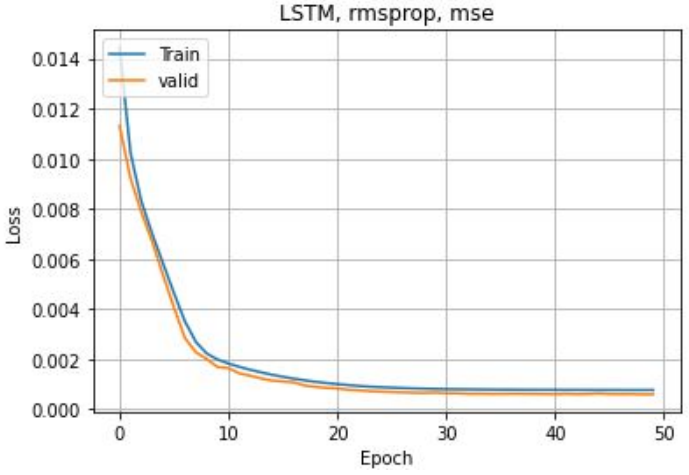
در صورتی که در این شبکه از optimizer RMSProp استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

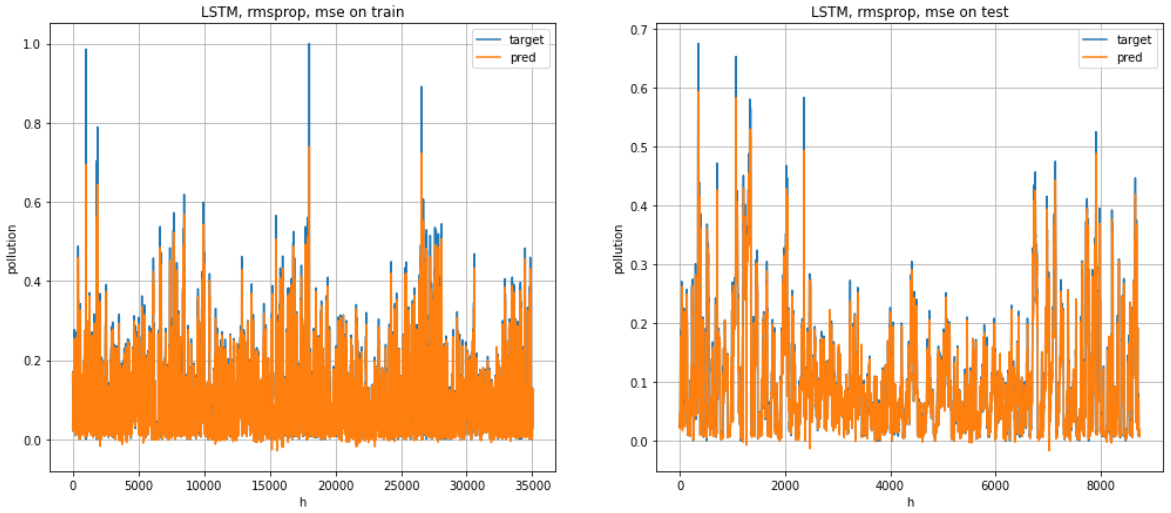
mse on train: 570.6229760986893

mse on test: 147.86964980544747

و نیز نمودارهای بدست آمده به ازای loss function = mse , optimizer = RMSProp ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۲۹: نمودار loss برای داده های train و validation در شبکه LSTM



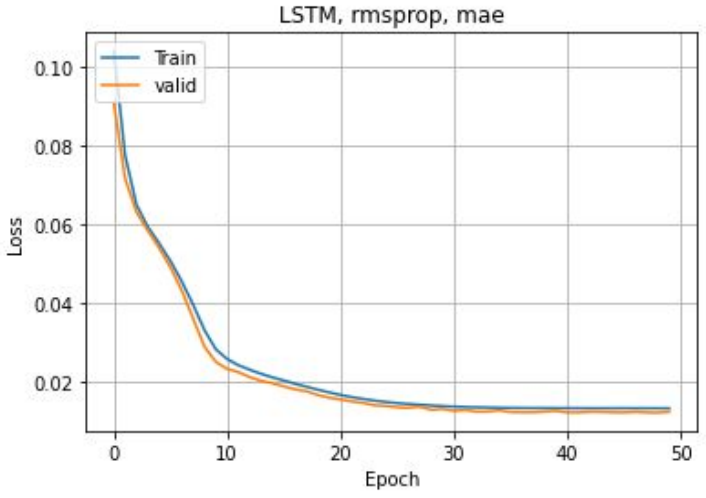
شکل ۳۰: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی LSTM

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

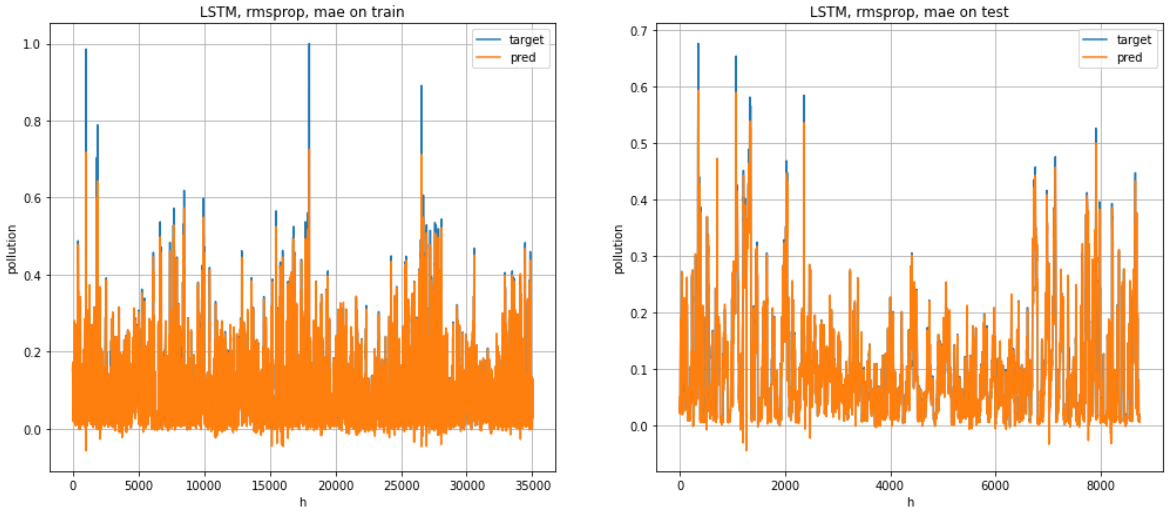
mse on train: 590.7945401067992

mse on test: 152.8655441748684

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = RMSProp ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۳۱: نمودار loss برای داده های train و validation در شبکه LSTM



شکل ۳۲: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی LSTM

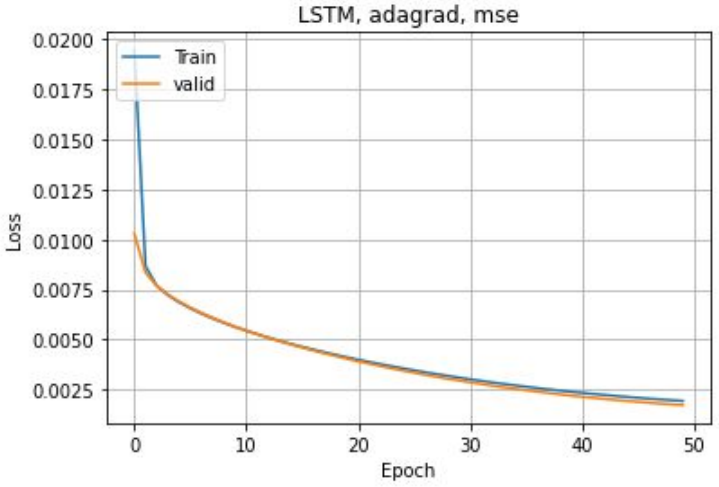
در صورتی که در این شبکه از optimizer ADAgrad استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

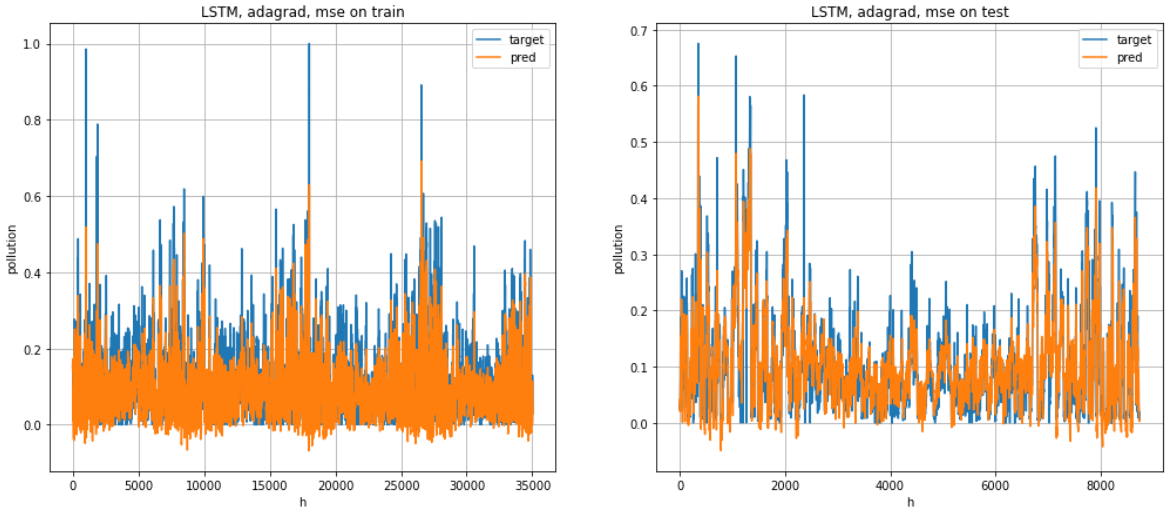
mse on train: 488.69316656672095

mse on test: 128.84152552071413

و نیز نمودار های بدست آمده به ازای loss function = mse , optimizer = ADAgrad ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۳۳: نمودار loss برای داده های train و validation در شبکه LSTM



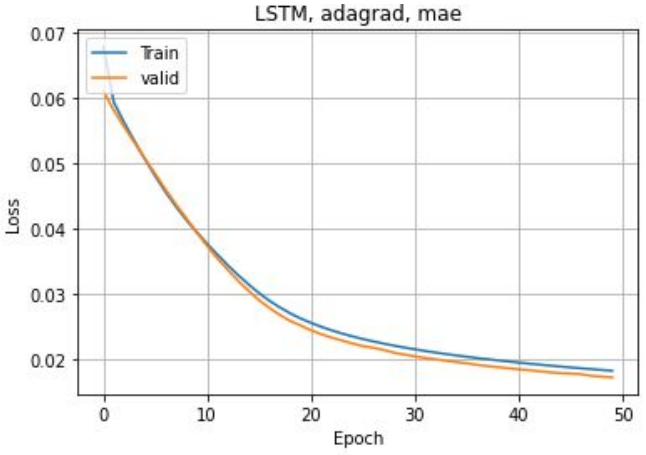
شکل ۳۴: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی LSTM

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

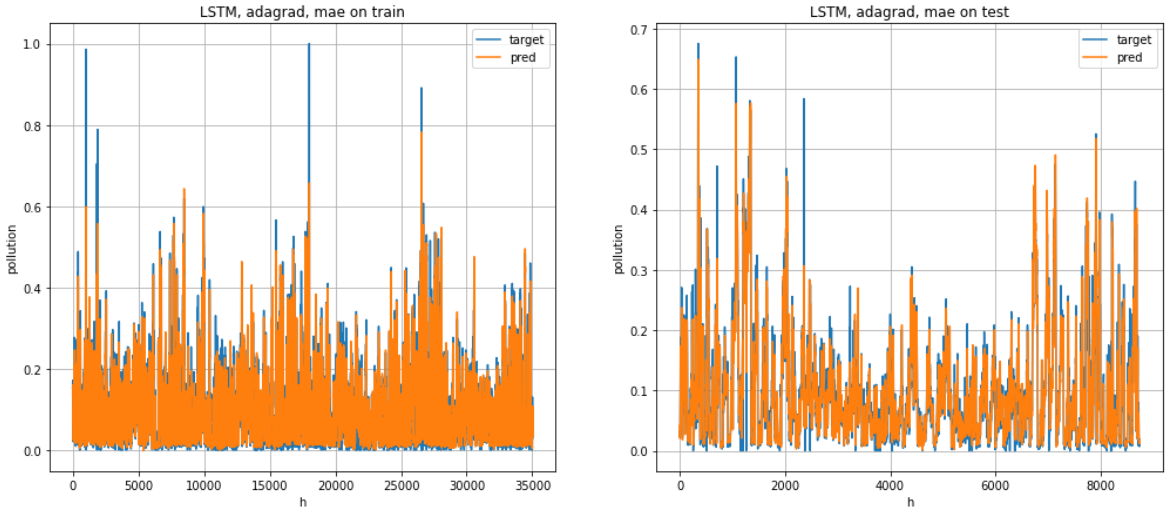
mse on train: 575.243267940261

mse on test: 149.6371452277409

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = ADAgrad ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۳۵: نمودار loss برای داده های train و validation در شبکه LSTM



شکل ۳۶: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی LSTM

بررسی اثرات مربوط به optimizerها و loss function های مختلف را بر روی شبکه ی GRU

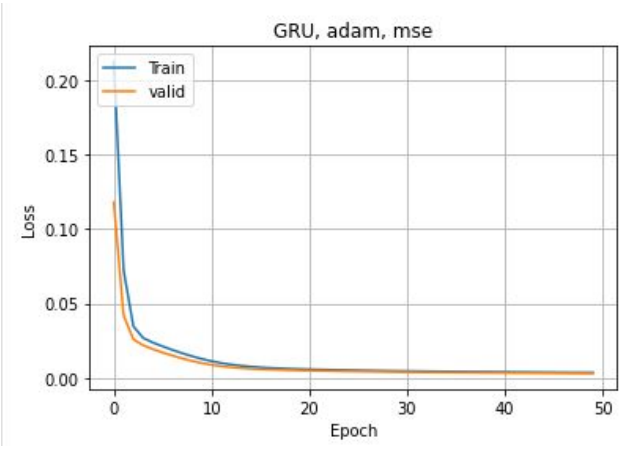
در صورتی که در این شبکه از optimizer adam استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

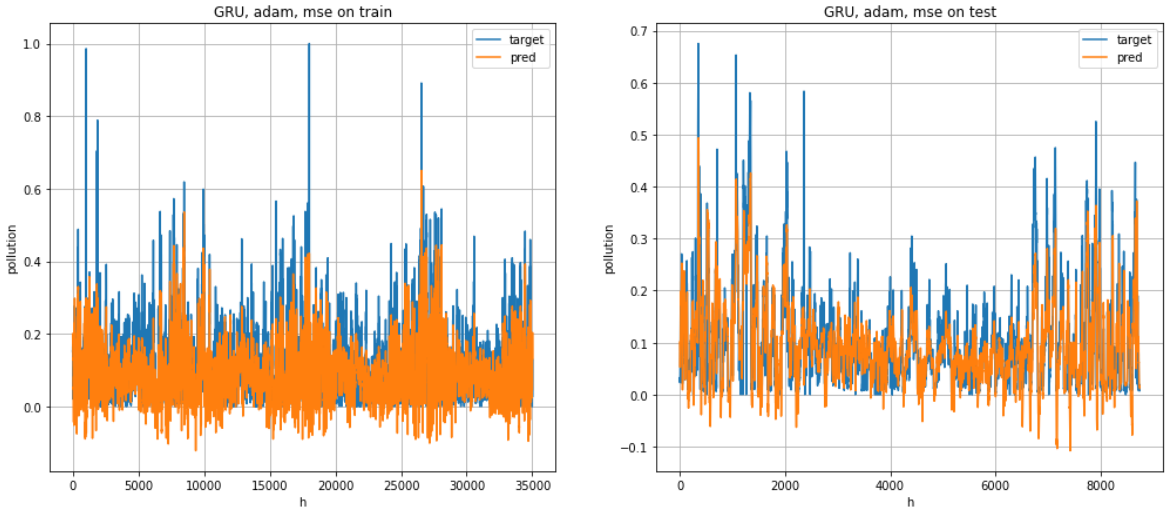
mse on train: 522.3088037922271

mse on test: 129.7831311512932

و نیز نمودارهای بدست آمده به ازای loss function = mse , optimizer = adam ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۳۷: نمودار loss برای داده های train و validation در شبکه GRU



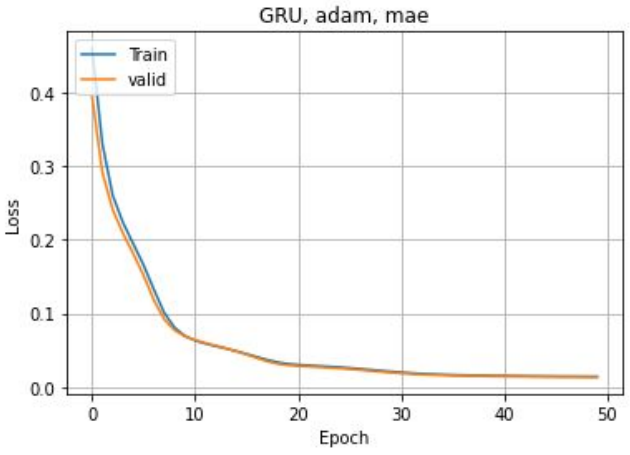
شکل ۳۸: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

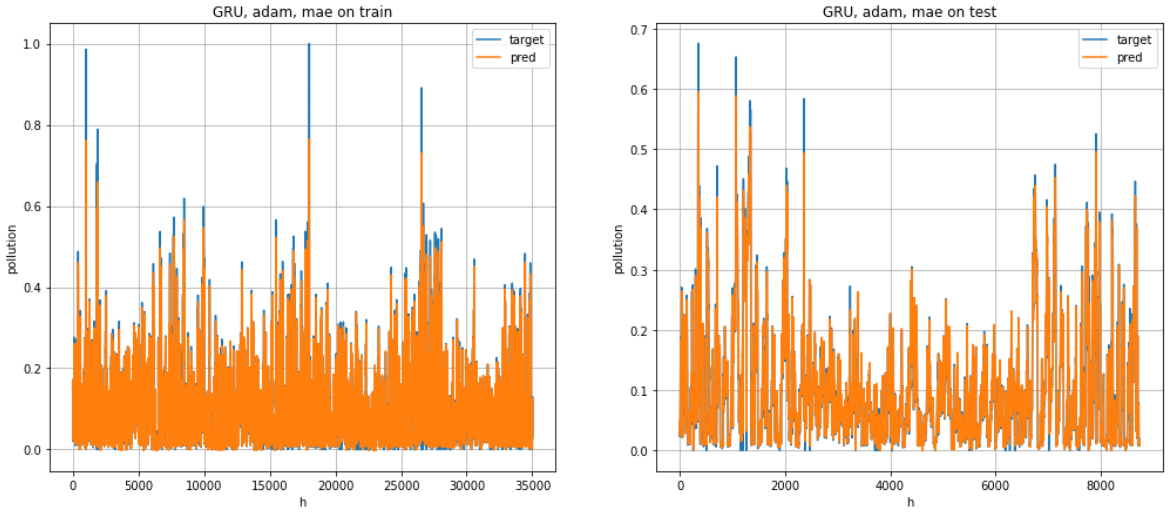
mse on train: 582.5780861817871

mse on test: 150.8055333028153

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = adam ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۳۹: نمودار loss برای داده های train و validation در شبکه GRU



شکل ۴۰: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

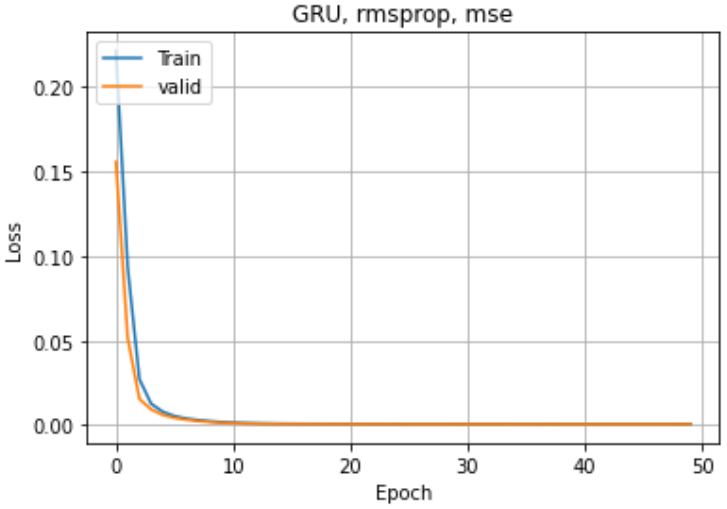
در صورتی که در این شبکه از optimizer RMSProp استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

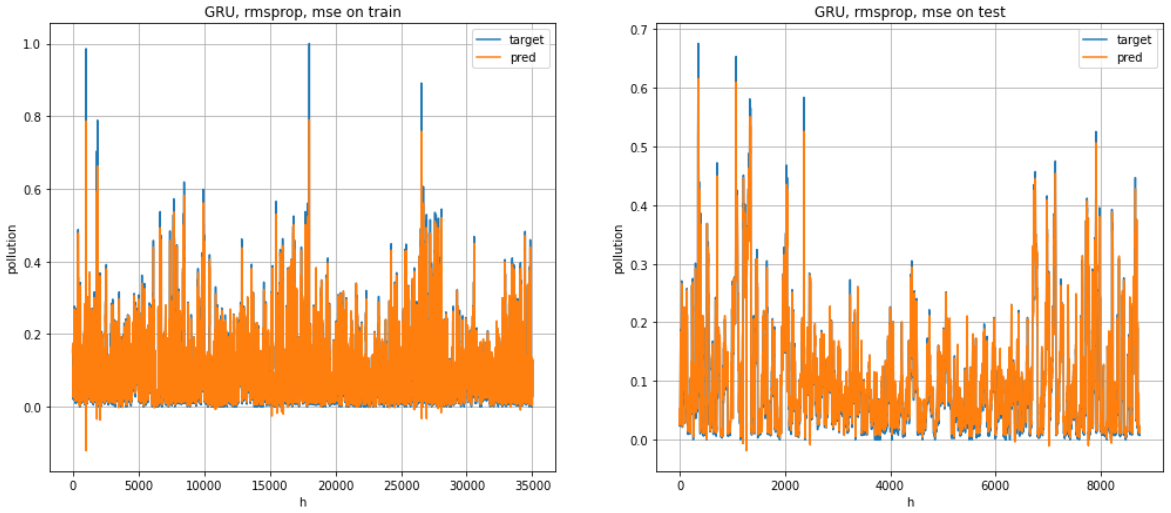
mse on train: 574.300008566778

mse on test: 148.77667944609752

و نیز نمودارهای بدست آمده به ازای loss function = mse , optimizer = RMSProp ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۴۱: نمودار loss برای داده های train و validation در شبکه GRU



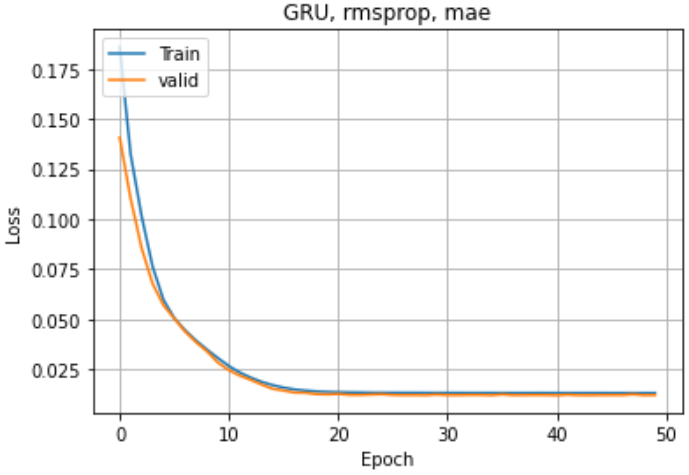
شکل ۴۲: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

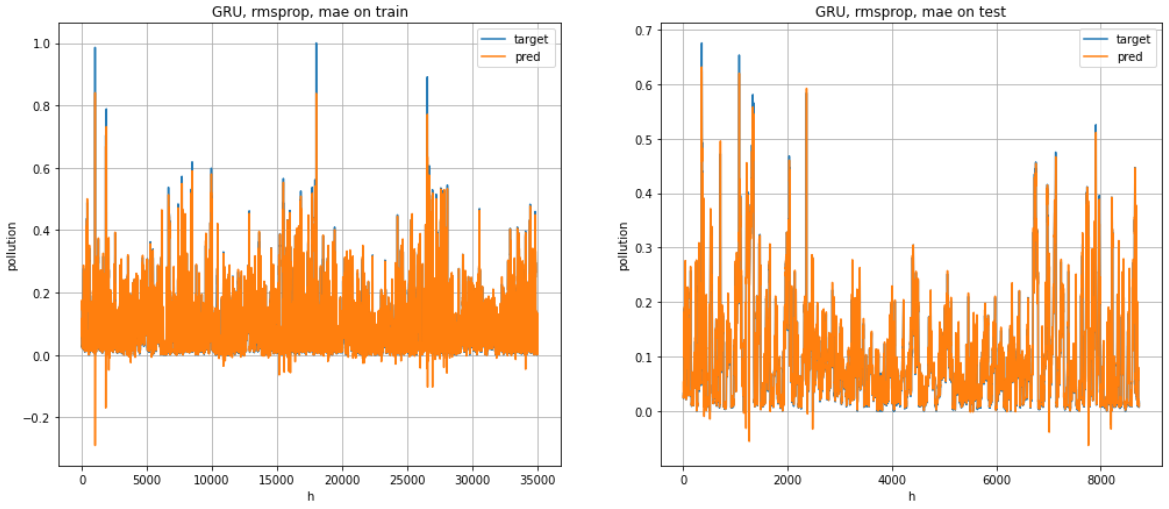
mse on train: 593.275193466404

mse on test: 153.25806820782788

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = RMSProp ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۴۳: نمودار loss برای داده های train و validation در شبکه GRU



شکل ۴۴: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

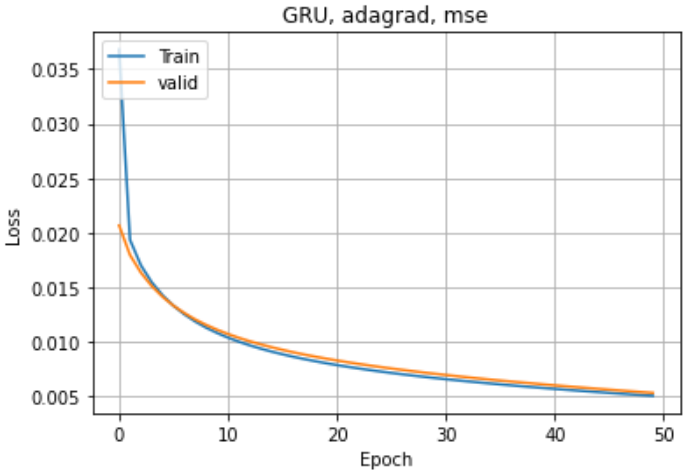
در صورتی که در این شبکه از optimizer ADAgrad استفاده کنیم:

مقدار تابع خطا mse برای داده های تست و train به صورت زیر است:

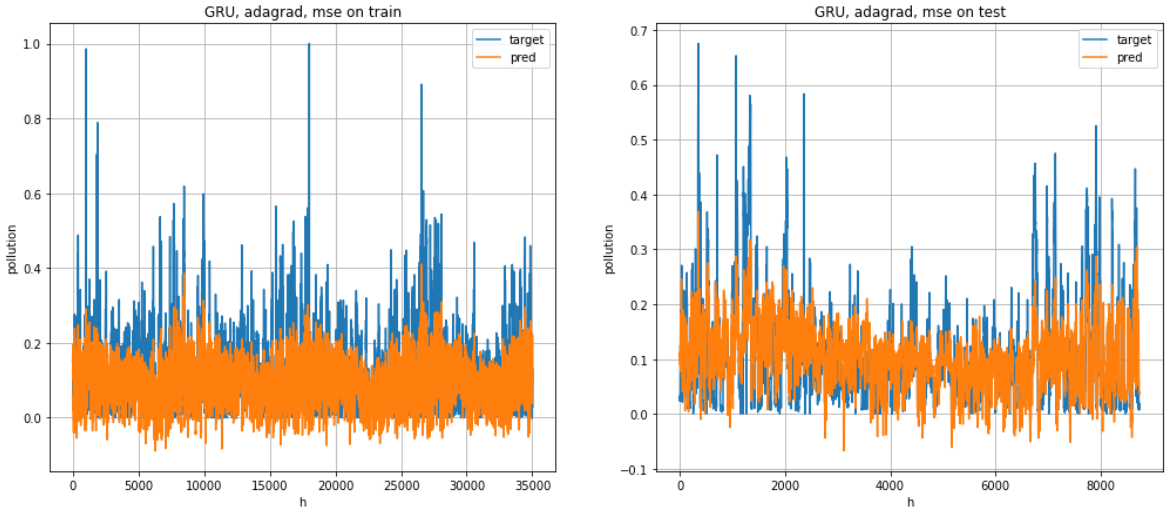
mse on train: 420.1489191581713

mse on test: 107.56446698329137

و نیز نمودارهای بدست آمده به ازای loss function = mse , optimizer = adagrad ،برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۴۵: نمودار loss برای داده های train و validation در شبکه GRU



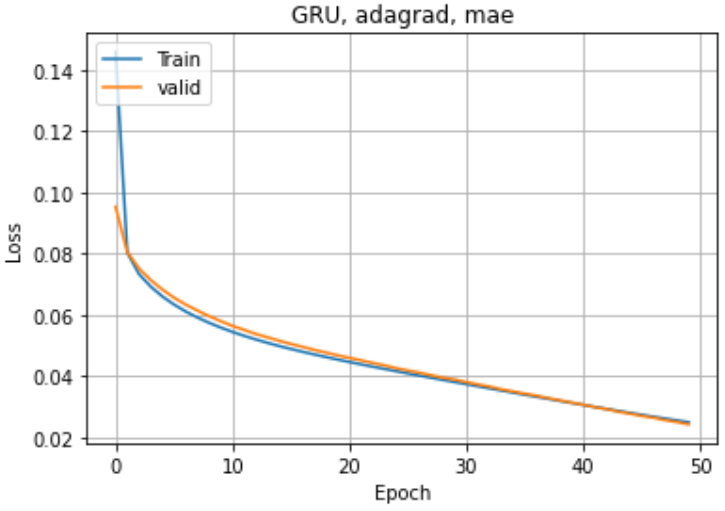
شکل ۴۶: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

مقدار تابع خطا mae برای داده های تست و train به صورت زیر است:

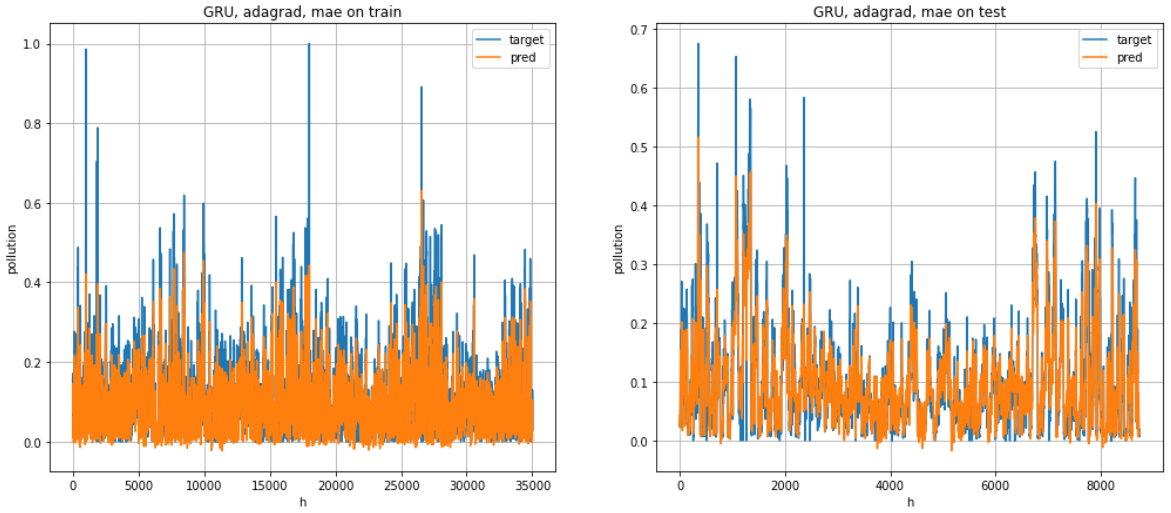
mse on train: 483.99977155258574

mse on test: 125.46550984206912

و نیز نمودارهای بدست آمده به ازای loss function = mae , optimizer = adagrad، برای loss و مقدار واقعی و تخمینی داده های تست و train به صورت زیر می باشد:



شکل ۴۷: نمودار loss برای داده های train و validation در شبکه GRU

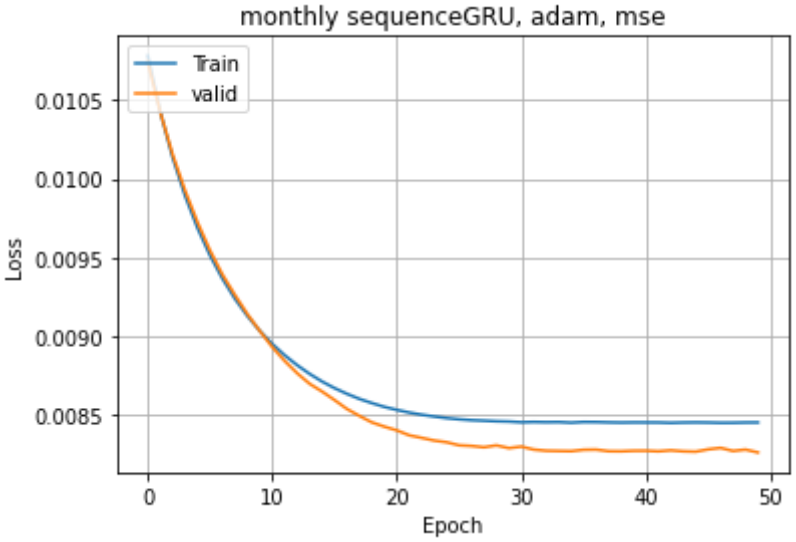


شکل ۴۸: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

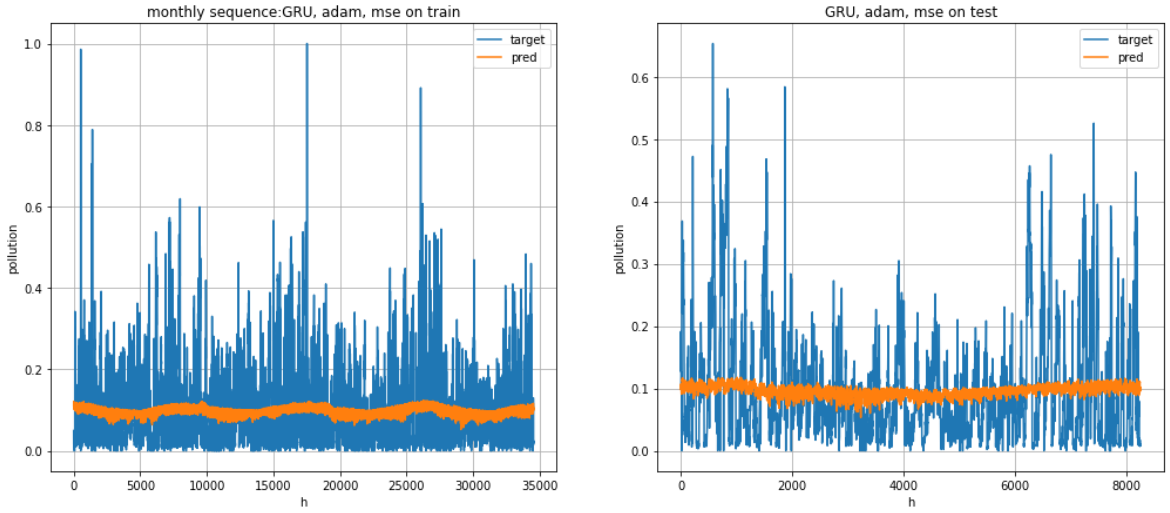
سوال ۱.۴ -

بررسی عملکرد شبکه به ازای سری زمانی های ماهیانه:

که در این شبکه نمودارهای loss برای داده های تست و آموزشی و نمودارهای مقدار واقعی و مقدار تخمینی برای داده های آموزشی و تست به صورت زیر می باشد:



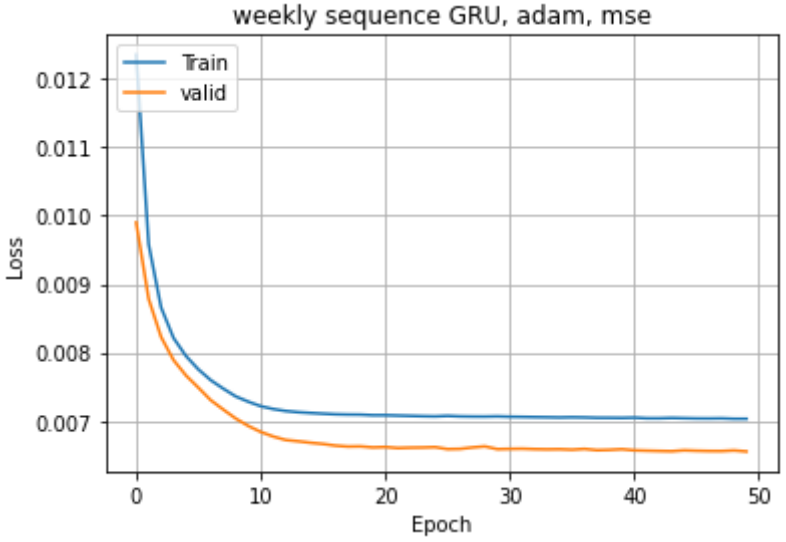
شکل ۴۹: نمودار loss برای داده های train و validation در شبکه GRU



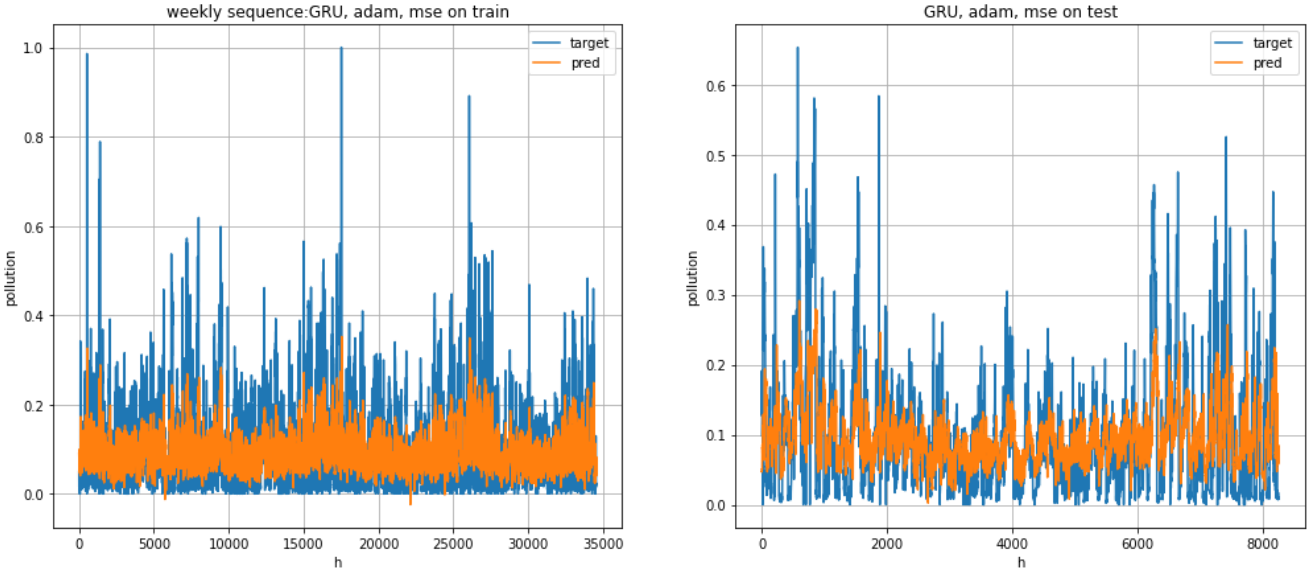
شکل ۵۰: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

بررسی عملکرد شبکه به ازای سری زمانی های هفتگی:

که در این شبکه نمودارهای loss برای داده های تست و آموزشی و نمودارهای مقدار واقعی و مقدار تخمینی برای داده های آموزشی و تست به صورت زیر می باشد:

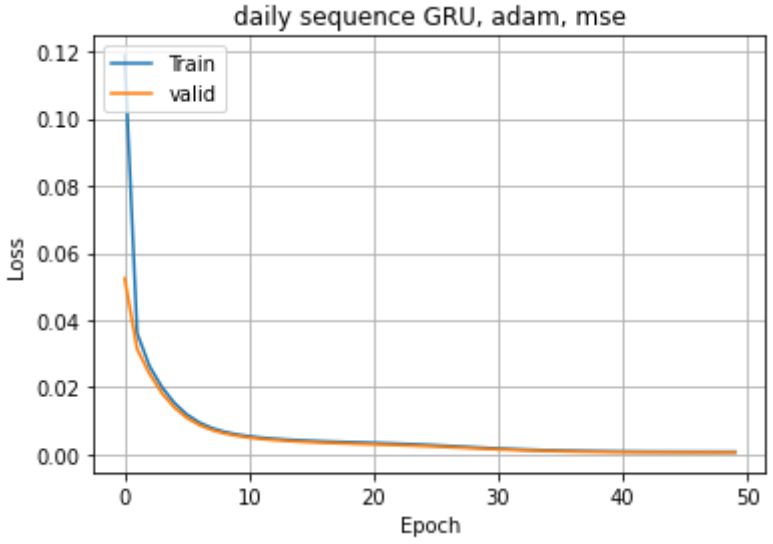


شکل ۵۱: نمودار loss برای داده های train و validation در شبکه GRU

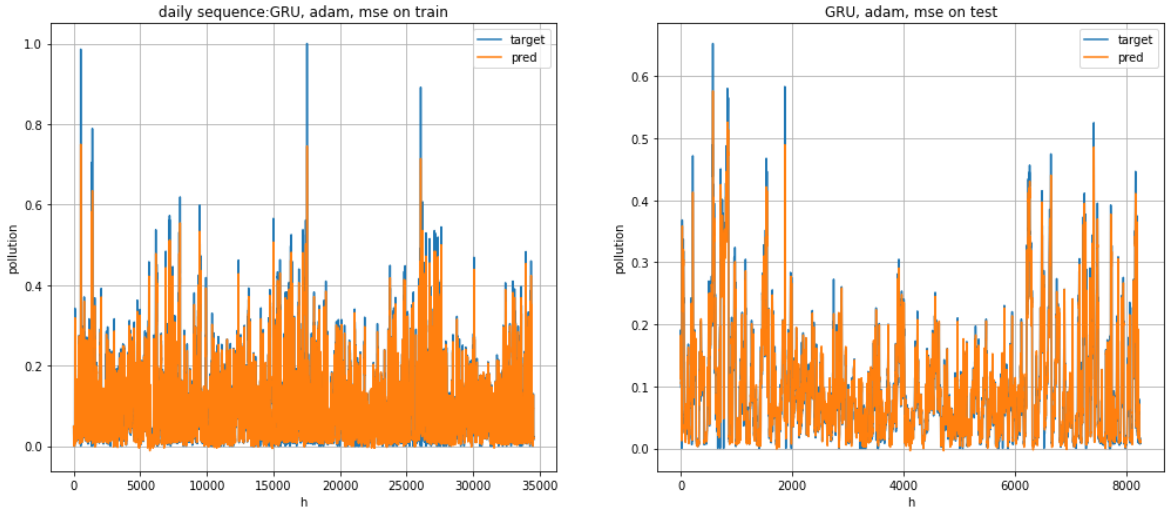


شکل ۵۲: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

بررسی عملکرد شبکه به ازای سری زمانی های روزانه:



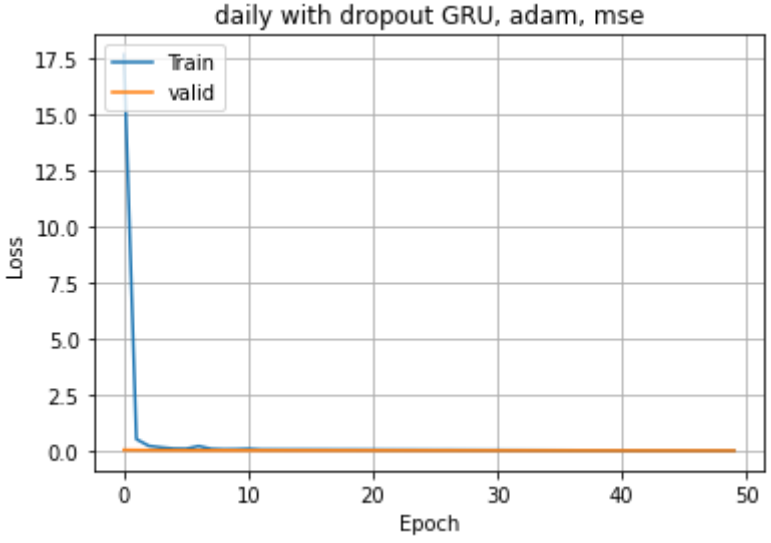
شکل ۵۳: نمودار loss برای داده های train و validation در شبکه GRU



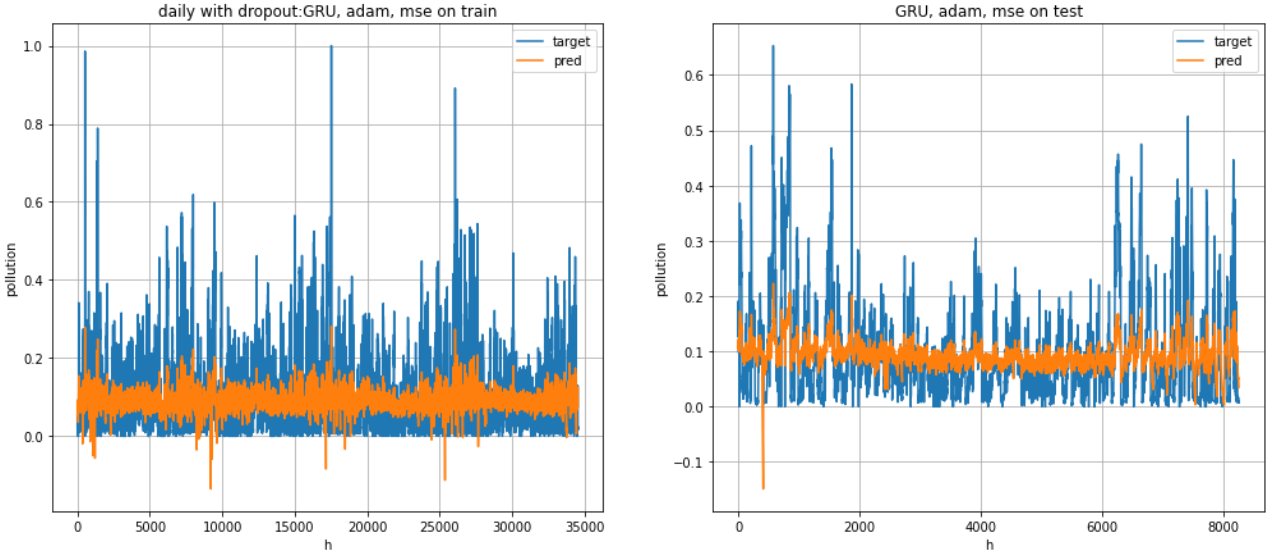
شکل ۵۴: نمودار مقدار واقعی و مقدار تخمین زده شده برای داده های آموزشی و تست در شبکه ی GRU

سوال ۱.۵ : dropout layer

در حالت کلی تاثیر لایه dropout جلوگیری از overfit شدن روی داده ها است. در مدل طراحی شده برای این مسئله مدل را بعلاوه یک لایه dropout=0.3 برای 50 epoch آموزش دادیم. تاثیر این لایه در مسئله های regression به صورت low pass filtering است که همان دوگان پیوسته overfitting برای مسئله classification است. پس از اعمال این لایه پیش بینی های انجام شده فورم نرم تری خواهند داشت. در شکل ۵۵ نمودار آموزش مدل و در شکل ۵۶ نمودار عملکرد مدل آورده شده است.



شکل۵۵: نمودار آموزش مدل برای ۵۰ epoch و یک لایه dropout



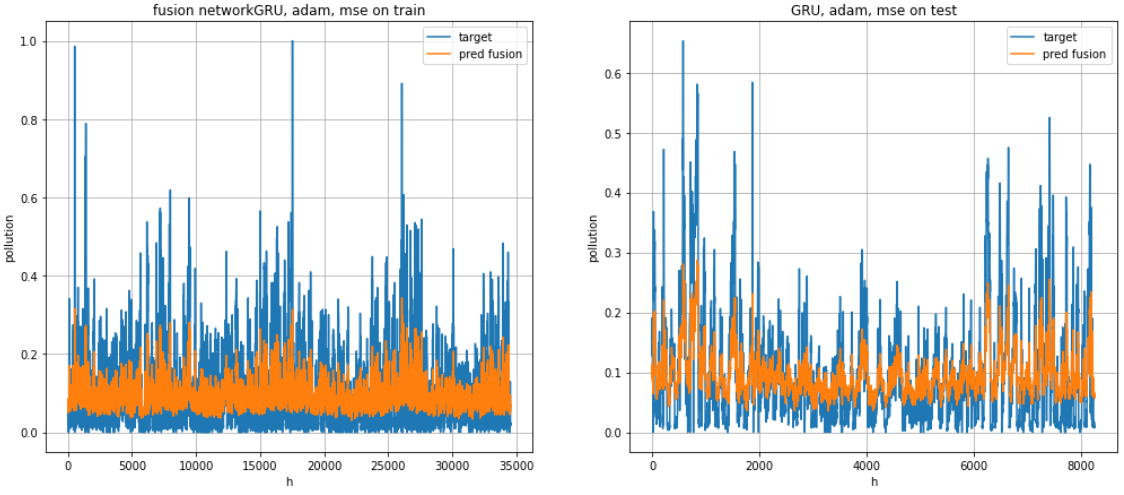
شکل۵۶: عملکرد مدل همراه با لایه dropout

سوال ۱.۶- fusion layer

در این قسمت ۳ شبکه پیشین را که بر روی انواع متفاوتی از دنباله زمانی آموزش داده شده اند را با یک دیگر ترکیب می‌کنیم. برای این کار از یک لایه average استفاده می‌کنیم. شهود ما در مورد این ترکیب این است که در یک شبکه پترن بلند مدت (ماهانه) ، در شبکه میانی پترن میان مدت(هفتگی) و در شبکه دیگر پترن کوتاه مدت(روزانه) یادگرفته می‌شود و با روش های مختلف می‌توان خروجی های مختلف این شبکه ها را با یک دیگر ترکیب کرد. ساده ترین روش میانگین هم وزن می‌باشد.

روش بهتر این بود که وزن خروجی ها را با توجه به خطای آنها روی داده های آموزش انتخاب می‌کردیم. با این روش مدلی که دقت بیشتری دارد تاثیر بیشتری نیز خواهد داشت. از این ترکیب نوعی regularization نیز می‌توان برداشت کرد که خطای بایاس را زیاد می‌کند و خطای واریانس را کم می‌کند.

شکل ۵۷: عملکرد ترکیب ۳ شبکه خواسته شده



سوال ۱.۷ - feature selection

روش های مختلفی برای feature selection وجود دارد که برخی از آن ها به اختصار توضیح داده می‌شوند.

Forward Selection

در این روش ویژگی ها تک تک به مسئله اضافه می‌شود و بهبود عملکرد مدل با هر افزایش اندازه گیری می‌شود. در هر مرحله آن ویژگی انتخاب می‌شود که با اضافه کردن آن بیشتر بهبود در عملکرد مشاهده شد.

Backward Elimination

در این روش مخالف روش اول عمل می‌کنیم و ویژگی ها را تک تک از داده حذف می‌کنیم و آن ویژگی که کمترین کاهش را در عملکرد داشت برای حذف انتخاب می‌شود. این فرایند را تا آنجایی ادامه می‌دهیم که حذف ویژگی برای دقت مدل خطای غیر قابل تحمل ایجاد کند.

Statistical Correlation

در این روش همبستگی میان ویژگی های مختلف و ویژگی هدف بررسی می‌شود و آن ویژگی هایی که ضریب همبستگی آن ها از حدی بالاتر بود انتخاب می‌شوند. برای محاسبه همبستگی از روش های مختلفی از جمله روش pearson می‌توان استفاده کرد.

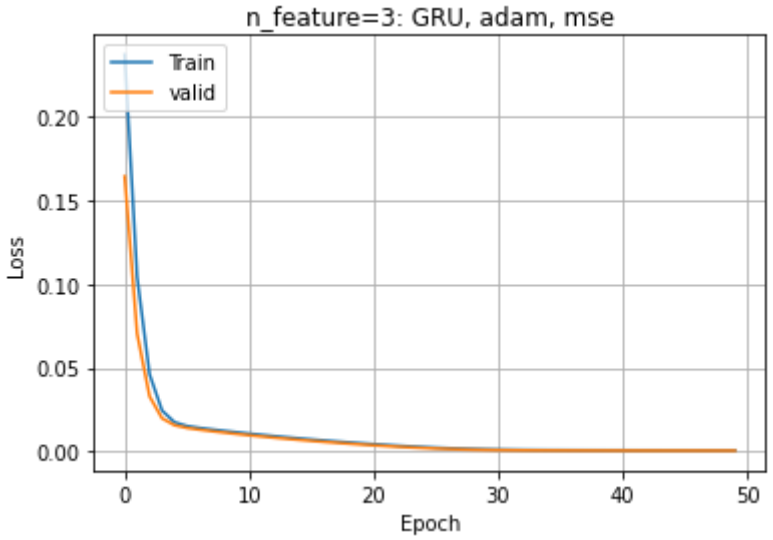
سوال ۱.۸ - dimensionality reduction

برای انتخاب ویژگی ها از روش Pearson's Coefs استفاده شد. ماتریس ضرایب همبستگی در زیر آمده است. با استفاده از این روش دو ویژگی temp و wind_dir بیشترین وابستگی خطی را با pollution داشتند و آن ها انتخاب شدند.

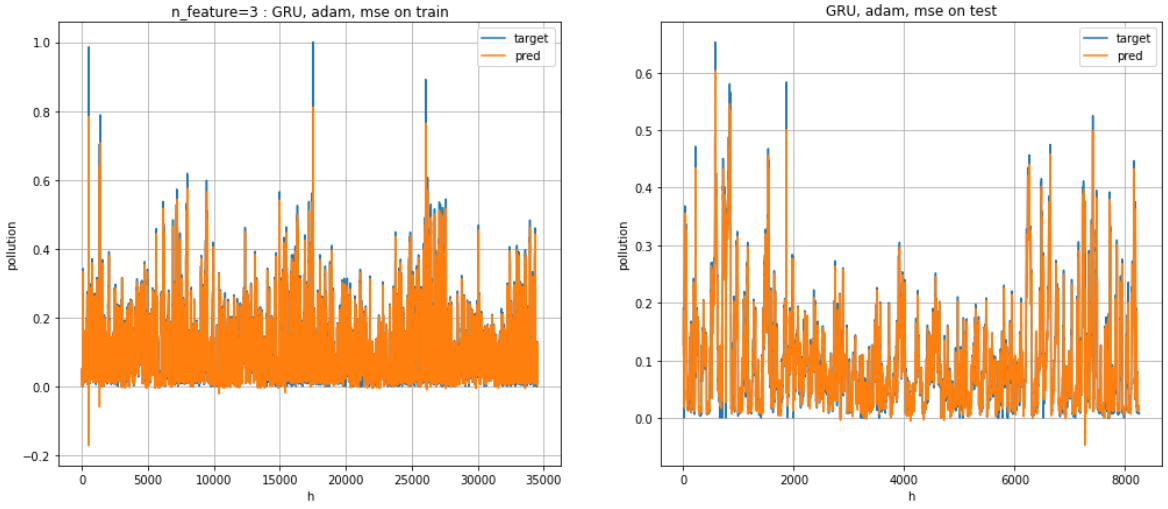
```
[1.00 0.54 0.81 0.56 0.81 0.55 0.81 0.53]
```

آموزش مدل با سلول GRU:

در این قسمت مدل با سلول GRU را بر روی داده های جدید آموزش دادیم. نمودار آموزش و عملکرد شبکه در شکل ۵۸ و شکل ۵۹ آمده است.



شکل۵۸: نمودار آموزش مدل بر داده های کاهش بعد یافته و سلول GRU

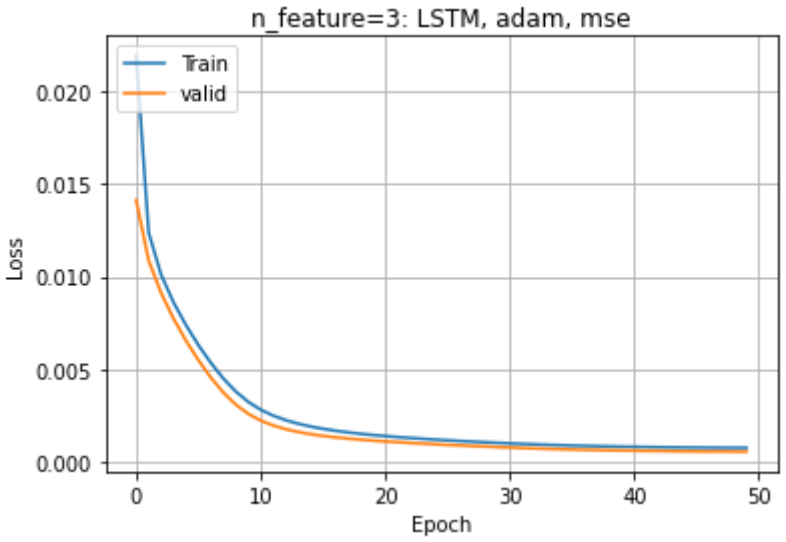


شکل ۵۹: عملکرد مدل آموزش داده شده بر روی داده های کاهش بعد یافته

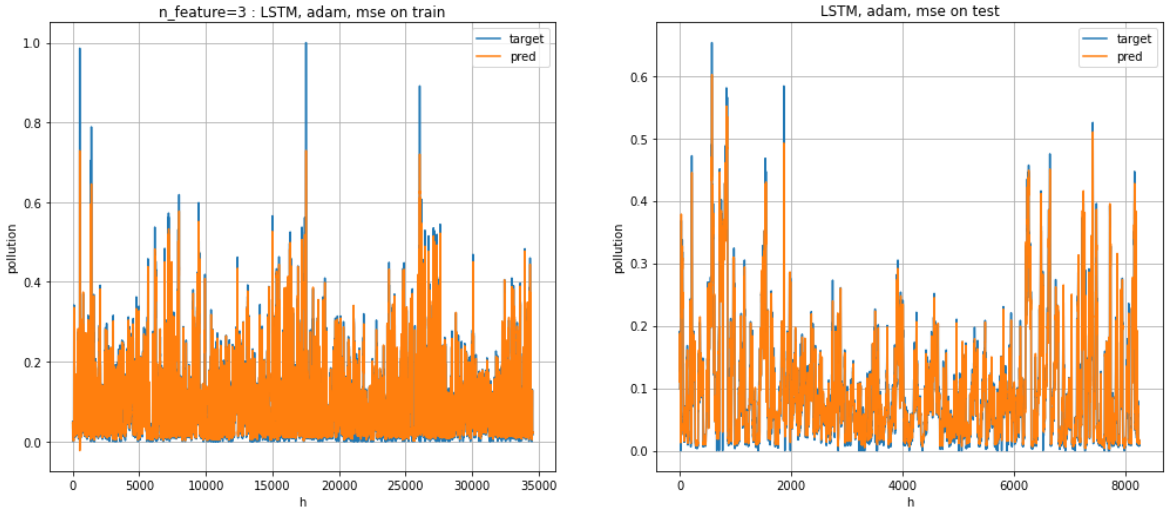
```
mse on train: 0.0007809150652502l5
mse on test:  0.0005653024947260 59
```

آموزش مدل با سلول LSTM:

در این قسمت مدل با سلول LSTM را بر روی داده های جدید آموزش دادیم. نمودار آموزش و عملکرد شبکه در شکل ۶۰ و شکل ۶۱ آمده است.



شکل ۶۰: نمودار آموزش سلول LSTM بر روی داده های کاهش بعد یافته

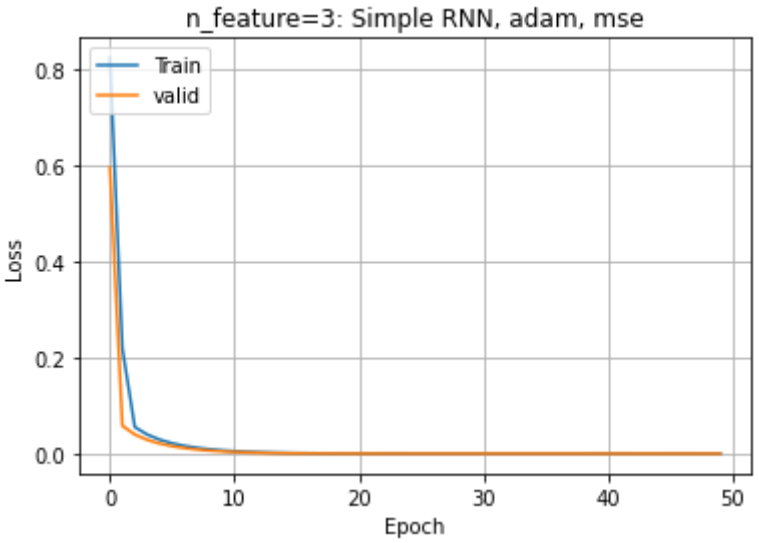


شکل ۶۱: عملکرد مدل بر روی داده های کاهش بعد یافته

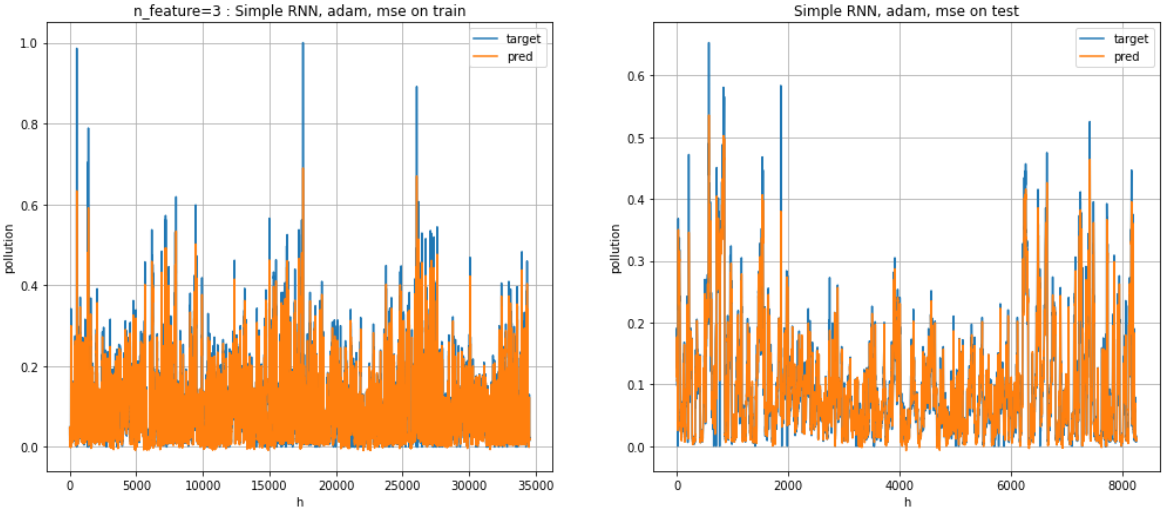
```
mse on train: 0.0007626322758868800
mse on test:  0.0005595340647890950
```

آموزش مدل با سلول RNN:

در این قسمت مدل با سلول RNN را بر روی داده های جدید آموزش دادیم. نمودار آموزش و عملکرد شبکه در شکل ۶۲ و شکل ۶۳ آمده است.



شکل ۶۲: نمودار آموزش سلول RNN بر داده های کاهش بعد یافته



شکل ۶۳: عملکرد مدل بر روی داده های کاهش بعد یافته

```
mse on train: 0.000892586748937448
mse on test:  0.000674810415322242
```

سوال ۲ - نقصان دادگان

قسمت های ۲.۱ ، ۲.۲، ۲.۳ و ۲.۴:

برای حل مشکل missing values در مجموعه داده از روش های متعددی می‌توان استفاده کرد. تعدادی از روش های معروف در زیر توضیح داده شده است. (منبع)

Listwise or case deletion

در این روش نمونه هایی که از هر جهت نقصان دارند حذف می‌شوند. این روش یکی از رایج ترین روش‌ها برای حل مشکل نقصان داده است و در ابزار های آماری مختلف استفاده می‌شود. ایرادی که به این روش وارد است بایاس شدن پارامتر ها است، که اگر فرض MCAR: missing completely at random در داده ها وجود داشته باشد، این مشکل به وجود نخواهد آمد.

Pairwise deletion

در این روش که معمولا برای statistical testing استفاده می‌شود. در این روش اگر قسمتی از داده از بین رفته باشد تنها در زمانی نمونه حذف می‌شود که در تست به صورت مستقیم به داده نیاز داشته باشیم.

Mean/Median substitution

در این روش داده های گم شده را با استفاده از یک شاخص تمرکز همانند میانگین یا میانه پر می‌کنیم. اگر داده ها چند بعدی باشند هم می‌توان برای تک‌تک feature ها این کار را جداگانه انجام داد.

Regression imputation

اگر داده ها دنباله دار باشند، و یا در یکی از ویژگی ها هیچ گونه نقصانی نباشد، می‌توان با حل یک مسئله regression داده های گم شده را تخمین زد. برای regression هم می‌توان از مدل های مختلف linear و یا polynomial استفاده کرد.

Last observation carried forward

روش دیگری که برای داده هایی با ترتیب زمانی استفاده می‌شود، تکرار آخرین داده موجود است. با فرض این که در طبیعت تغییرات خیلی شدید نداریم و تعداد و دقت نمونه برداری بالا است، می‌توان آخرین نمونه موجود را برای داده های از دست رفته تکرار کرد.

Maximum likelihood

در روش هایی که بر مبنای ML کار می‌کنند، معمولا فرض می‌شود که داده ها از یک چگالی مشخص مثلا multivariate normal distribution نمونه برداری شده اند. با این فرض و با حذف داده های دارای نقص پارامتر های مدل تخمین زده می‌شود و با استفاده از مدل داده های دارای نقص جایگزین می‌شوند.

Expectation-Maximization

این روش بر مبنای ML است. در این روش پس از این که پارامتر های مدل تخمین زده شد، داده ها مجددا از مدل محاسبه می‌شود. در مرحله بعد پارامتر های مدل مجددا روی مجموعه داده جدید فیت می‌شوند و این کار را آنقدر ادامه می‌دهیم تا پارامتر های مدل (یا دیتا ست تولیدی)به اعداد خاصی همگرا شوند. مشکل اصلی در این روش این است که تضمینی برای همگرایی وجود ندارد.

KNN

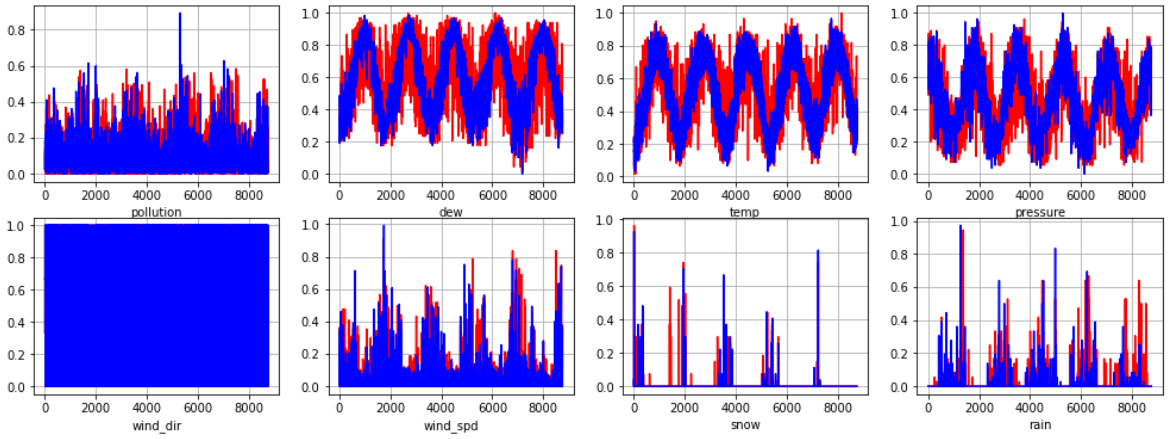
در این روش برای هر نمونه ، K تا نزدیک ترین همسایه آن پیدا می‌شود. برای محاسبه فاصله تنها از feature هایی استفاده می‌شود که نقصان داده ندارند.

قسمت ۲.۵ :

برای اصلاح داده ها از روش Expectation Maximization استفاده شده است. یکی از مراحل این الگوریتم انتخاب یک رگرسور مناسب است. به همین جهت ۴ مورد از رگرسور های معروف امتحان شدند و نتایج عملکرد آن ها آمده است.

۱) DecisionTreeRegressor: non-linear regression

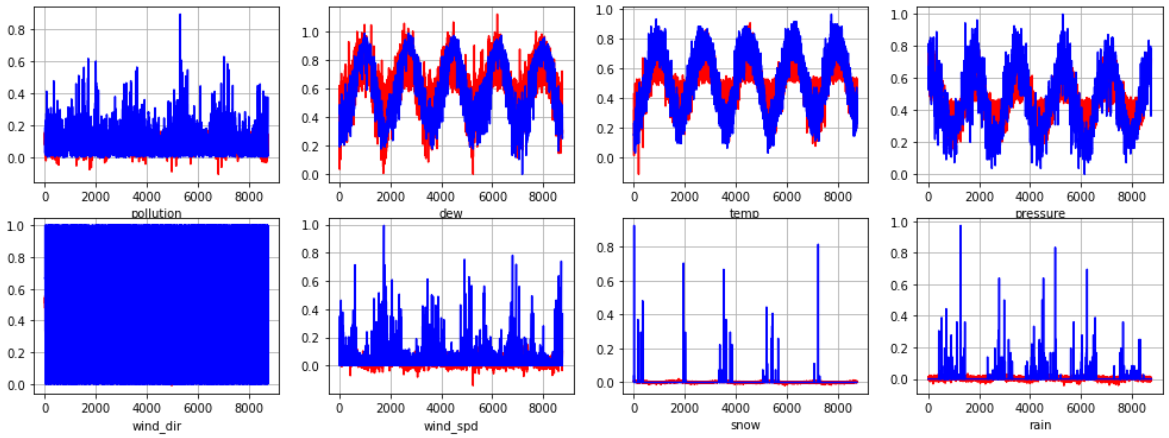
برای این رگرسور خطای MSE روی تمامی داده های miss شده برابر با `0.0244` شد. عملکرد این رگرسور بر روی داده های miss شده در شکل ۶۴ آمده است.



شکل۶۴: عملکرد EM با استفاده از DT Regressor

۲) BayesianRidge: regularized linear regression

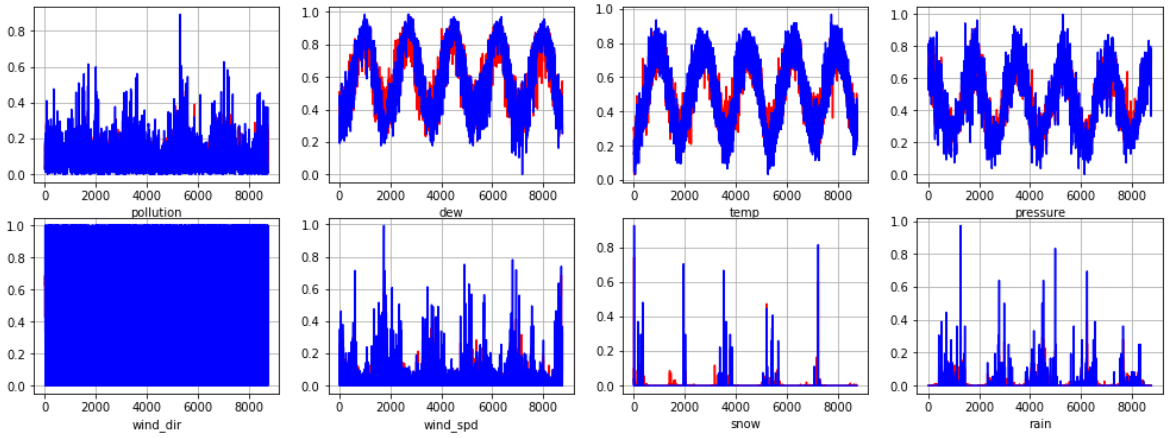
برای این رگرسور خطای MSE روی تمامی داده های miss شده برابر با `0.0179` شد. عملکرد این رگرسور بر روی داده های miss شده در شکل ۶۵ آمده است.



شکل ۶۵ : عملکرد EM با استفاده از Linear Regression

۳) ExtraTreesRegressor: similar to missForest in R

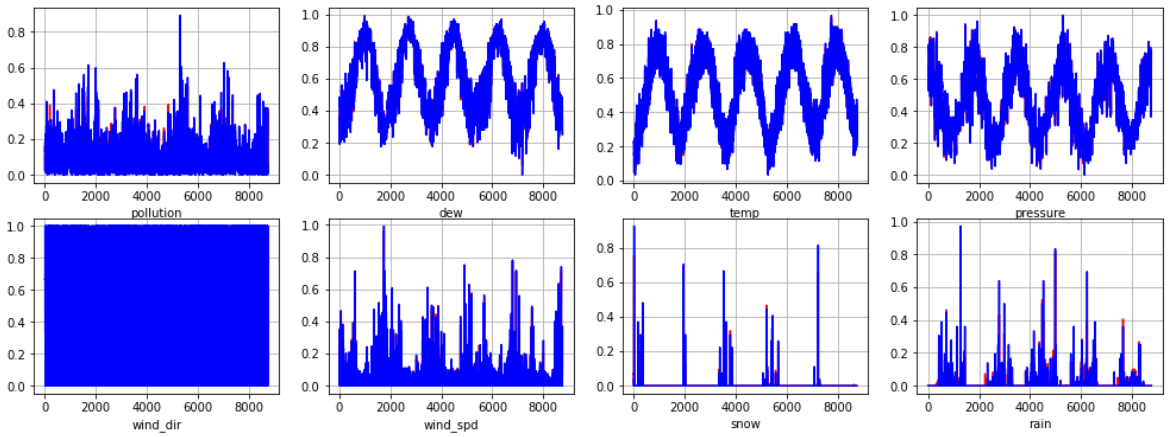
برای این رگرسور خطای MSE روی تمامی داده های miss شده برابر با `0.0123` شد. عملکرد این رگرسور بر روی داده های miss شده در شکل ۶۶ آمده است.



شکل ۶۶ : عملکرد EM با استفاده از ExtraTreeRegressor

۴) KNeighborsRegressor

برای این رگرسور خطای MSE روی تمامی داده های miss شده برابر با `0.0101` شد. عملکرد این رگرسور بر روی داده های miss شده در شکل ۶۷ آمده است.



شکل ۶۷ : عملکرد EM با استفاده از KNNRegressor

بهترین دقت کلی بر روی مدل KNNRegressor بدست آمد و MSE برای داده های miss شده به تفکیک feature ها نیز برای این مدل مطابق خواسته سوال محاسبه شده است.

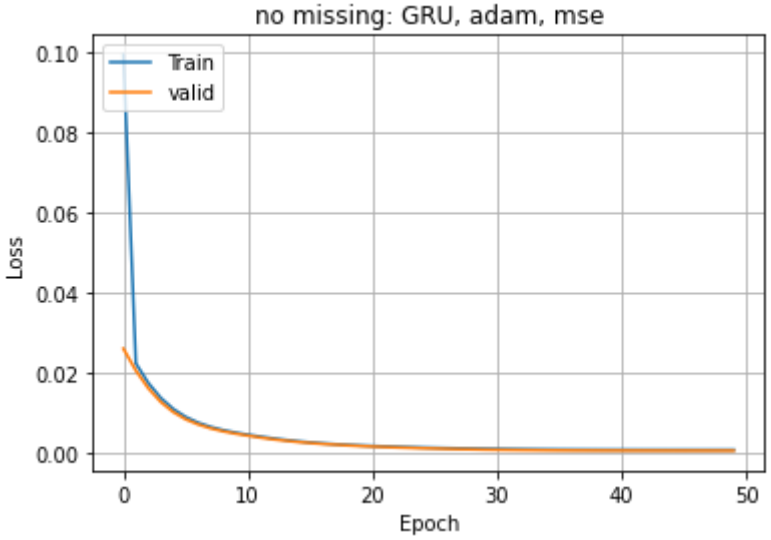
```
pollution : 0.0006129148162046969
dew       : 0.00035319678571542123
temp      : 0.0005739906026470082
pressure  : 0.00015332378423673734
wind_dir  : 0.0784194026980097
wind_spd  : 0.0005745162238887405
snow      : 0.00013304285509994288
rain      : 0.00035946800805429865
```

قسمت ۲.۶ :

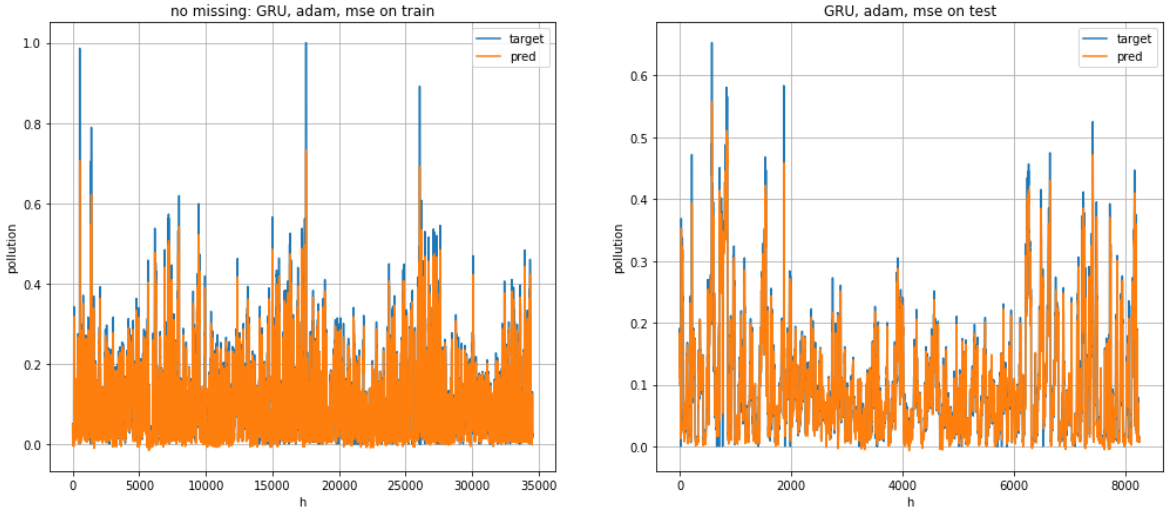
برای مقایسه درست و معقول دو حالت، تنها ۲۰ درصد از هر بعد بردار ویژگی نمونه های آموزش تغییر پیدا کرد و داده های آزمایش بدون تغییر باقی ماند.

۱) تاثیر imputation بر روی عملکرد سلول های GRU

ابتدا در شکل ۶۸ و ۶۹ نمودار هزینه هنگام آموزش برای سلول GRU آورده شده است و سپس پیشبینی شبکه بر روی داده های آموزش و آزمایش نشان داده شده است. در این نمودار های شبکه بر روی داده اصلی آموزش داده شده است.



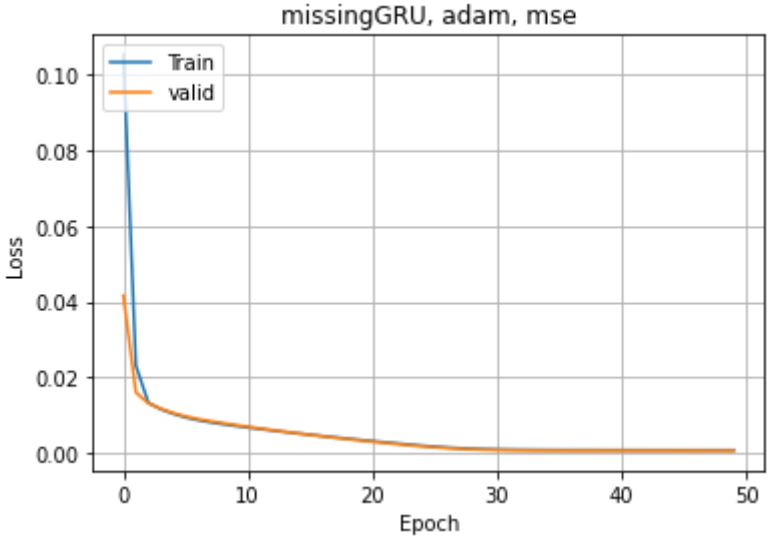
شکل ۶۸: نمودار هزینه هنگام آموزش شبکه با سلول GRU بر داده ها بدون نقصان



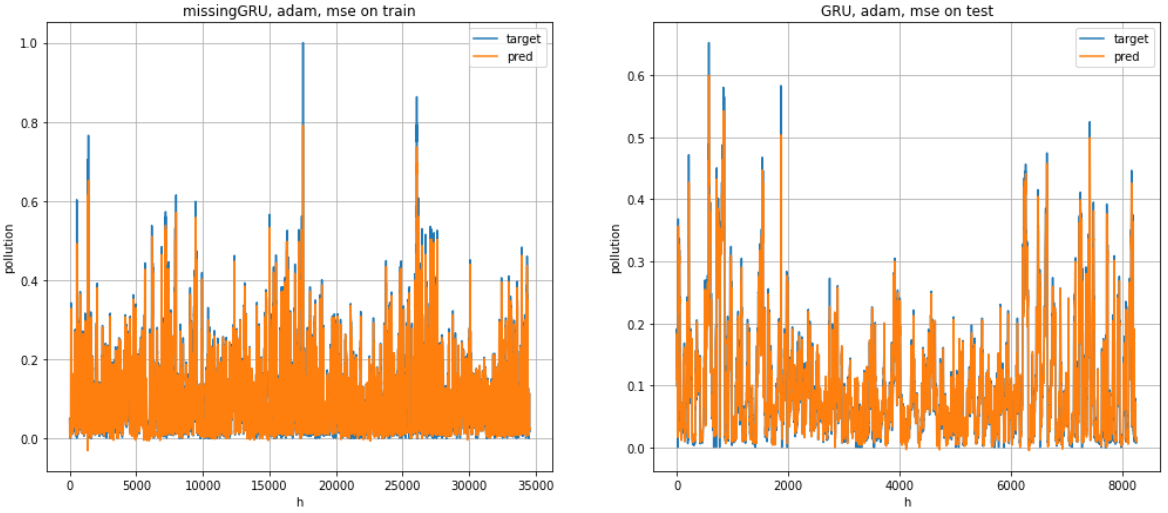
شکل ۶۹ : عملکرد شبکه آموزش داده شده بر داده های آموزش و آزمایش

```
mse on train: 0.0008137297900926467
mse on test: 0.0006051393366813082
```

سپس شبکه با سلول مشابه بر روی داده های imputed آموزش داده شد و نمودار های هزینه و عملکرد آن در شکل های ۷۰ و ۷۱ آمده است.



شکل ۷۰: نمودار هزینه هنگام آموزش بر روی داده های imputed



شکل ۷۱: عملکرد شبکه بر روی داده های آزمایش و آموزش

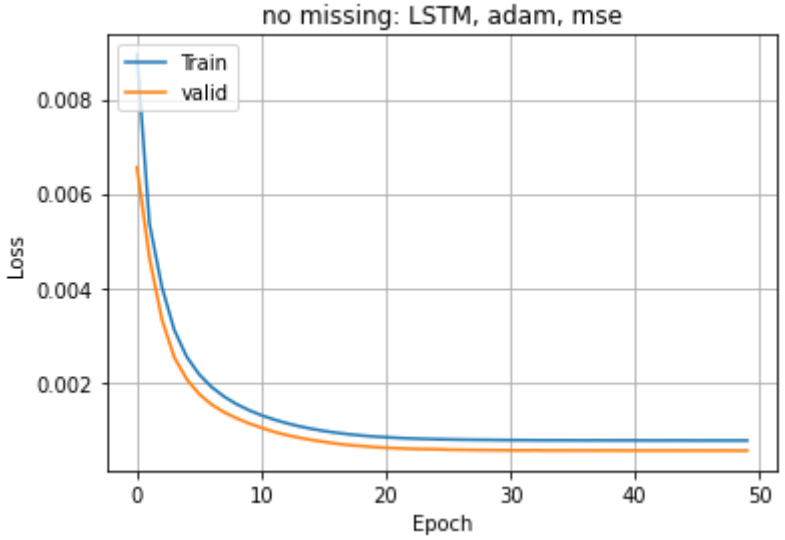
```
mse on train: 0.0007046574598554939
```

```
mse on test: 0.000564276500588688
```

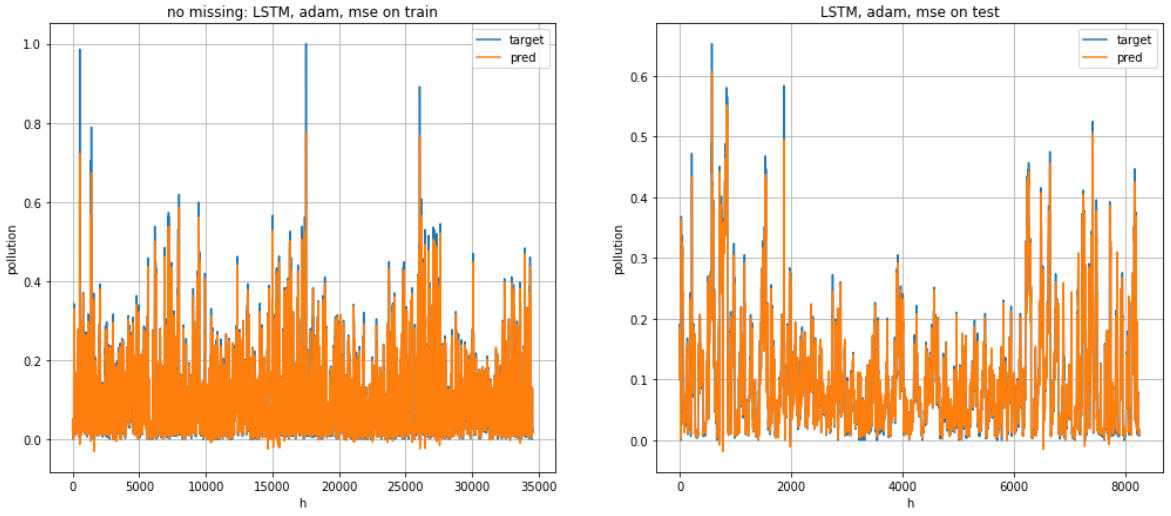
از مقایسه MSE داده های تست به این نتیجه می‌رسیم که با impute کردن داده های missing به دقت قابل قبولی می رسیم و گاهی خطا کمتر از حد معمول می‌شود. توجیه این اتفاق آن است که فرایند imputation قالب با استفاده از مدل هایی انجام می‌شود که همانند lowpass filter عمل می‌کنند بنابراین داده های آموزش چگالی نرم تری پیدا کرده و درک پترن درون آن ها برای شبکه راحت تر می‌شود. همچنین به خاطر این filtering مشکل overfitting نیز کمتر خواهد بود و دقت بالا بر داده های آزمایش در حالت دوم سندی بر این حرف است.

۲) تاثیر imputation بر روی عملکرد سلول های LSTM

برای سلول‌های LSTM نیز مشابه قسمت قبل عمل میکنیم. ابتدا در شکل ۷۲ و ۷۳ نمودار هزینه هنگام آموزش برای سلول LSTM آورده شده است و سپس پیشبینی شبکه بر روی داده های آموزش و آزمایش نشان داده شده است. در این نمودار های شبکه بر روی داده اصلی آموزش داده شده است.



شکل ۷۲: نمودار هزینه هنگام آموزش برای داده های بدون نقصان

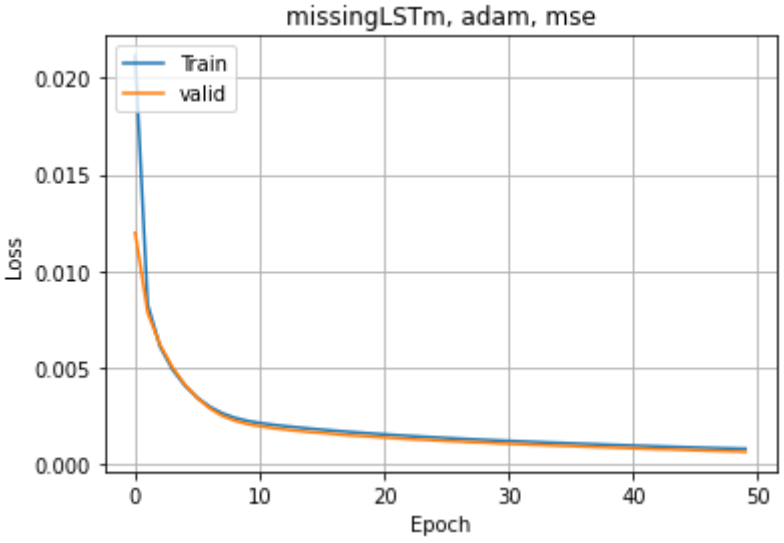


شکل ۷۳: عملکرد شبکه بر داده های آموزش و آزمایش

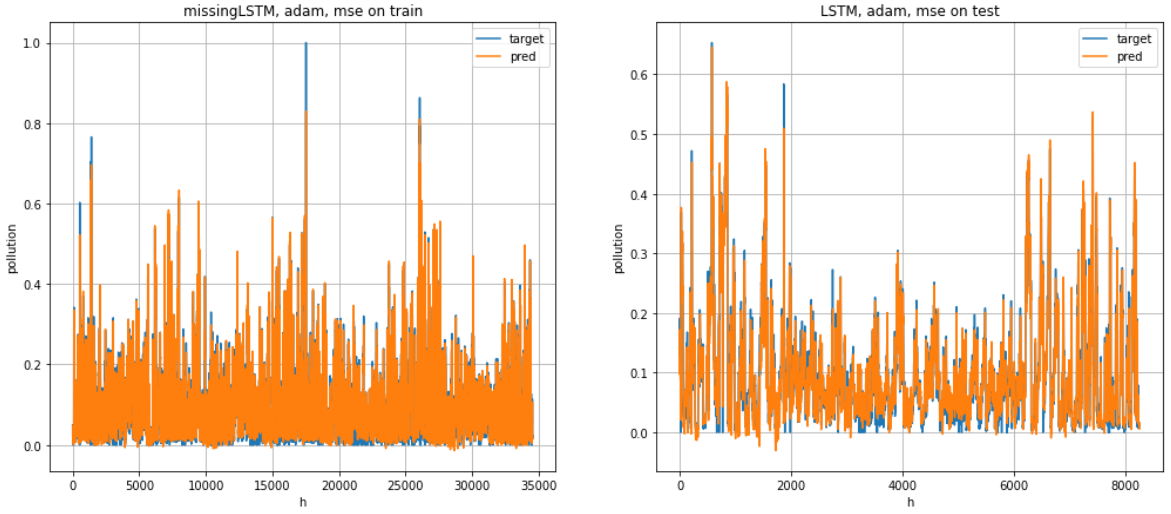
```
mse on train: 0.0007616502144614497
```

```
mse on test: 0.0005495717783973555
```

سپس شبکه با سلول مشابه بر روی داده های imputed آموزش داده شد و نمودار های هزینه و عملکرد آن در شکل های ۷۴ و ۷۵ آمده است.



شکل ۷۴: نمودار هزینه هنگام آموزش برای داده های imputed



شکل ۷۵: عملکرد شبکه بر روی دادهای آموزش و آزمایش

```
mse on train: 0.0007872557181684624
mse on test: 0.0006449183574378455
```

در این نوع سلول هم mse به حالت بدون نقصان بسیار نزدیک است، هم برای داده های آموزش و هم برای داده های آزمایش. این نزدیکی نشان گر این است که روش پیشنهادی برای imputation توزیع داده ها را به‌صورت چشم گیری تغییر نمیدهد و در نتیجه به نتایج یکسانی می‌رسیم.

نحوه اجرای کدها

کد های سوال ۱ در فایل NNDL_miniproj2_1.inpy و کدهای سوال ۲ در فایل NNDL_miniproj2_2.inpy قرار دارد.