



دانشگاه صنعتی امیرکبیر  
(پای تکنیک تهران)

## پاسخ تمرین ششم شبکه های عصبی

استاد درس جناب آقای دکتر صفابخش

تدریس یار جناب آقای مهندس اسدی

نیمسال دوم سال تحصیلی ۱۴۰۰-۱۴۰۱

محسن عبادپور

شماره دانشجویی: ۴۰۰۱۳۱۰۸۰

ایمیل: [m.ebadpour@aut.ac.ir](mailto:m.ebadpour@aut.ac.ir)

## فهرست پاسخ ها

۲	توضیحات اولیه، بارگزاری داده ها و پیش پردازش
۳	مسئله اول
۶	مسئله دوم: قسمت اول
۸	مسئله دوم: قسمت دوم
۱۰	مسئله دوم: قسمت سوم
۱۱	مسئله سوم: قسمت اول
۱۴	مسئله سوم: قسمت دوم

## توضیحات

### توضیحات اولیه، بارگزاری داده ها و پیش پردازش

برای پیاده سازی و پاسخ گویی به این سری از تمرینات از زبان پایتون و کتابخانه keras برای ایجاد و آموزش مدل ها استفاده شده است. در مجموعه داده ضمیمه شده تعدادی دیتاست خالی وجود داشت که آنها حذف شدند. سپس هر کدام از فایل های csv. خوانده شده و سپس ستون های اضافی پاک شدند که شامل اسمی و... بودند. در مرحله بعدی مجموعه داده ها بر اساس تاریخ با هم join و بر اساس تاریخ با هم جوین شدند تا ویژگی های شاخص های گوناگون در یک روز کنار هم قرار گیرند. در گام بعد داده های یک سال اخیر برای آموزش و آزمون مدل انتخاب شدند و در نهایت داده ها بر اساس روز مرتب شدند.

برای سری کردن داده ها نیز یکتابع به نام create\_time\_steps پیاده سازی شده است که در آن به ازای پارامتر ارسالی n، هر n داده متوالی از یکسال گذشته یک رکورد جدید را تشکیل می دهد. برچسب مربوط به رکورد جدید نیز برابر با صعودی بودن یا نزولی بودن شاخص در روز  $n+1$  می باشد. همچنین برای همگرایی سریع و بهبود روند آموزش شبکه و دور از تاثیر گذاری ویژگی هایی که رنج بزرگ دارند، ویژگی ها نرمال شده اند.

هر رکورد از مجموعه دادهی حاصل به شکل (15,129) در آمد که ۱۵ نشانگر گام زمانی(تعداد روز های متوالی مورد نیاز برای پیش بینی شاخص برای روز بعد) و ۱۲۹ نیز نشانگر ویژگی های هر گام زمانی می باشد(ویژگی های مربوط با شاخص ها). مقصود از انتخاب ۱۵ گام زمانی برای پیش بینی شاخص، معادل بودن با سه هفته معاملاتی در بورس می باشد. همچنین مجموعه داده به با نسبت های ۲۰، ۷۰ و ۱۰ درصد به گروه های آموزش، آزمون و اعتبار سنجی بصورت تصادفی و طبقه ای تقسیم می شود.

## مسئله اول

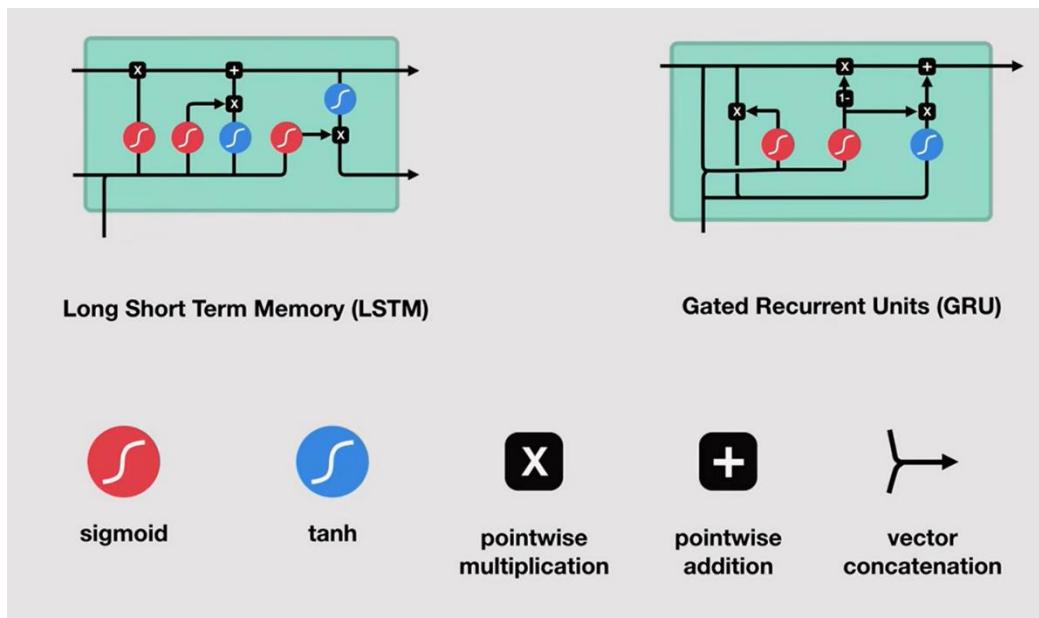
### مسئله اول

چالش اساسی شبکه های عصبی بازگشتی حافظه کوتاه در ذخیره و قدرت محدود در انتقال داده های قدیمی آن است که در بلندمدت این شبکه امکان یادگیری اطلاعاتی را که در زمان بسیار قبل تر به شبکه داده شده است را از دست می دهد. دلیل این موضوع Vanishing Gradient است. ابتدا معماری LSTM و سپس GRU معرفی شدند که حالتی از RNN بوده و توانایی یادگیری دنباله های بلندمدت هستند و مشکل Vanishing Gradient را در دنباله های طولانی حل میکنند. شبکه‌ی LSTM و GRU کنترل‌های داخلی بصورت Gate دارند که این گیت‌ها جریان اطلاعات را کنترل می‌کنند؛ به عبارتی دیگر این گیت‌ها مسئول تعیین داده های مهم برای حفظ و انتقال در گام‌های سری زمانی هستند و مشخص می‌کنند چه داده‌هایی باید حفظ/حذف شود، ورودی/خروجی داده شود یا خیر. به این صورت، شبکه‌ی داده های مهم را در طول عمر سری عبور می‌دهد تا خروجی مدنظر حاصل شود.

برای بررسی شباهت‌ها، تفاوت‌ها و کاربرد مدل‌های LSTM و GRU ابتدا هر کدام را شرح داده و سپس اقدام به تحلیل جزئی تر می‌کنیم؛ ابتدا شبکه LSTM را مورد ارزیابی قرار می‌دهیم. در معماری LSTM ما دارای سه گیت ورودی، خروجی و فراموشی می‌باشیم. گیت فراموشی تصمیم می‌گیرد کدام داده‌های در هر مرحله ذخیره و کدام داده‌ها حذف یا به عبارتی فراموش شود. اطلاعات ورودی گام جدید به همراه اطلاعات Hidden State گام قبلی به این گیت وارد شده و از تابع فعالیت Sigmoid عبور می‌کنند. خروجی این تابع بین صفر تا یک حاصل می‌شود. به هر میزان که مقدار خروجی به عدد صفر نزدیک‌تر باشد، یعنی باقی‌ماندی داده‌های وضعیت قبلی فراموش شود و هر چه به عدد یک نزدیک‌تر باشد یعنی باید حفظ شود.

گیت ورودی نیز برای به روزرسانی مقادیر موجود در cell state پیاده سازی شده است. اطلاعات ورودی گام جدید، به همراه اطلاعات hidden state گام قبلی، به این گیت ورودی داده شده و از تابع فعالیت Sigmoid عبور داده می‌شوند و خروجی بین صفر و یک این تابع تصمیم می‌گیرد که چه داده‌هایی نادیده گرفته شده و چه داده‌هایی به روزرسانی شوند. همچنین داده ورودی در این گام زمانی به همراه اطلاعات hidden state گام قبلی به تابع فعالیت Tanh وارد می‌شوند. در نهایت خروجی دو تابع فعالیت Sigmoid و Tanh در هم ضرب می‌شوند تا تابع Sigmoid تصمیم بگیرد چه مقادیری از خروجی تابع Tanh باید حفظ شوند. همچنین گیت خروجی نیز تصمیم می‌گیرد که بعدی چه مقداری را به خود گیرد. همان‌طور که از شبکه‌های عصبی بازگشتی به خاطر داریم hidden state اطلاعات ورودی‌های قبلی را حفظ و انتقال می‌دهد. اول اطلاعات ورودی گام کنونی به همراه اطلاعات hidden state گام قبلی به تابع فعالیت Sigmoid وارد می‌شوند. مقدار آپدیت‌شده‌ی cell state به تابع Tanh وارد می‌شود. خروجی این دو تابع با هم ضرب می‌شود تا تصمیم

گرفته شود hidden state چه اطلاعاتی را با خودش به گام بعدی ببرد. درنهایت cell state جدید و hidden state جدید به گام زمانی بعدی منتقل می‌شوند.



شبکه عصبی GRU نیز بسیار شبیه شبکه LSTM می‌باشد با این تفاوت که به جای سه گیت، فقط دو گیت Update Gate و Reset Gate دارد. علاوه بر آن که شبکه GRU مسیری انتقالی برای Cell State ندارد؛ برای انتقال اطلاعات از Hidden State را در ساختار خود دارد. همینطور در شبکه GRU هم مانند شبکه LSTM توابع Sigmoid و TanH وجود دارد. گیت آپدیت در شبکه استفاده می‌کند. همینطور در شبکه GRU عمل می‌کند؛ این گیت تصمیم می‌گیرد چه مقدار از اطلاعات گذشته که در گام‌های قبلی زمانی داشته ایم، به شبکه وارد شود. در این گیت مقدار ورودی جدید به همراه مقدار Hidden State گام قبلی در وزن دقیقاً مانند دو گیت فراموشی و ورودی در شبکه LSTM عمل می‌کند؛ این گیت تصمیم می‌گیرد چه مقدار از اطلاعات گذشته که در گام‌های قبلی زمانی داشته ایم، به شبکه وارد شود. در این گیت مقدار ورودی جدید به همراه مقدار Hidden State گام قبلی در وزن متناظر خود ضرب و سپس با هم جمع می‌شوند و به تابع فعالیت Sigmoid وارد می‌شوند تا خروجی در رنج صفر الی یک قرار بگیرد. در زمان آموزش شبکه، این وزن‌ها هر بار به روزرسانی می‌شوند تا فقط اطلاعات مفید به شبکه اضافه و منتقل شوند. درنهایت خروجی گیت آپدیت با گام قبلی ضرب می‌شوند تا در مرحله بعد برای محاسبه خروجی استفاده شود.

گیت بازنشانی نیز تصمیم می‌گیرد چه مقدار از اطلاعات داده‌های گام‌های زمانی قبلی فراموش شود. در این مرحله مقدار ورودی جدید به همراه مقدار hidden state گام قبلی، در وزن متناظر خود ضرب و سپس با هم جمع می‌شوند و به تابع فعالیت Sigmoid وارد می‌شوند تا خروجی در رنج صفر تا یک حاصل شود. تفاوتی که این گیت با گیت آپدیت دارد در این است که وزن‌هایی که مقدار ورودی و گام قبلی در آن ضرب می‌شوند متفاوت است و این یعنی بردارهای حاصل در این گیت با بردارهای حاصل که در گیت به روزرسانی داریم، متفاوت می‌باشد. سپس خروجی گیت بازنشانی با hidden state گام قبلی ضرب نقطه‌ای می‌شود. همچنین ورودی جدید ابتدا در

وزن متناظر مربوط به خود ضرب و سپس با خروجی (ضرب گیت بازنشانی با hidden state گام قبلی) جمع می‌شود؛ درنهایت این بردار خروجی به یک تابع  $\text{Tanh}$  وارد می‌شود تا خروجی حاصل شود.

همانطور که ملاحظه می‌شود شبکه‌های عصبی GRU و LSTM به جهت کارکرد بسیار شبیه همدیگر است مثلاً در شبکه GRU مانند شبکه LSTM توابع فعالیت یکسانی به کار گرفته می‌شود و در ساختار و هدف مشابه یکدیگر می‌باشند که هر دو می‌خواهند کنترلی در مسیر جریان داده‌های انتقالی و تاثیر آن در ورودی جدید داشته باشند اما در عین حال تفاوت‌های جزئی نیز دارند. ما در شبکه GRU به جای سه گیت، فقط دو گیت آپدیت و بازنشانی داریم و در شبکه GRU مسیری به نام Cell State وجود ندارد و برای انتقال اطلاعات از همان Hidden State استفاده می‌کند. همچنین GRU حافظه داخلی ندارد و بر عکس LSTM از گیت خروجی نیز استفاده نمی‌کند. در LSTM گیت ورودی و گیت خروجی توسط یک گیت متصل می‌شوند و در GRU گیت reset مستقیماً به Hidden state قبلی اعمال می‌شود. همینطور در LSTM مسئولیت مسیر بازنشانی گیت بر عهده دو گیت ورودی و خروجی است.

همانگونه که می‌بینیم، پیچیدگی و اتصالات LSTM برای حفظ و انتقال داده بین گام‌ها بسیار بیشتر از GRU می‌باشد لذا وزن‌های مورد نیاز برای یادگیری و زمان آموزش آن نیز بیشتر است؛ بنابراین می‌توان نتیجه گرفت که معماری GRU برای دیتاست‌هایی بهتر است که چندان بزرگ نبوده و پیچیدگی کمی بین گام‌های زمانی برای انتقال داده داشته باشند و الگوی موجود پیچیدگی بسیار بالایی نداشته باشند؛ همچنین روش LSTM نیز برای دیتاست‌های بزرگ که ممکن است پیچیدگی بالایی بین گام‌های زمانی داشته باشند می‌تواند نتیجه بهتری حاصل نماید. (مراجع)

## مسئله دوم

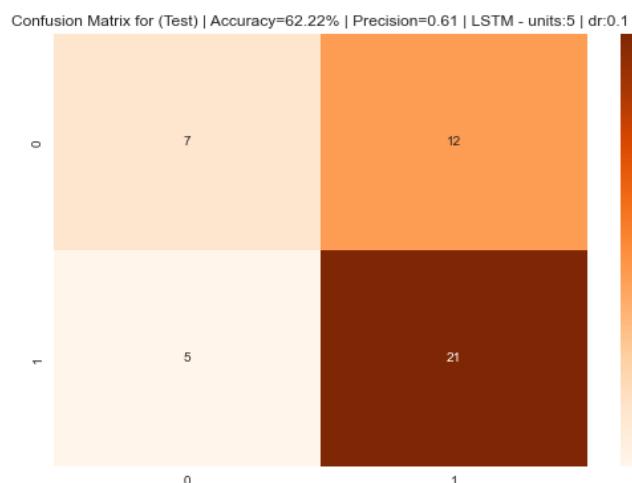
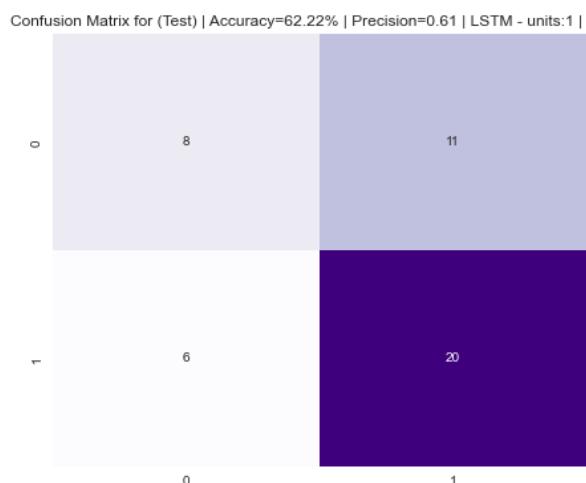
### مسئله دوم: قسمت اول

همانگونه که در توضیحات اولیه اشاره شده است در این سری از تمرینات بندۀ برای آموزش شبکه و پیش‌بینی شاخص از داده‌های ۱۵ روز استفاده می‌کنم که معادل با سه هفته معاملاتی می‌باشد. برای ایجاد مدل‌های LSTM و GRU یک تابع با امضای get\_lstm\_gru پیاده‌سازی شده است که با گرفتن یک لیستی شامل تعداد نورون‌های هر لایه مدل نظر را ایجاد می‌کند. اگر لیست شامل یک عنصر باشد که یک لایه مخفی از نوع خواسته شده و به تعداد تعیین شده ایجاد و باز می‌گرداند. اگر لیست شامل تعدادی عنصر باشد به تعداد طول لیست لایه مخفی از نوع خواسته شده (GRU یا LSTM) به طور پشت‌هه ایجاد می‌کند.

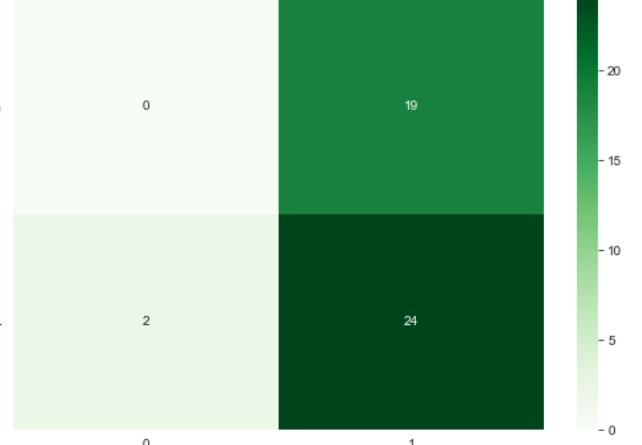
برای این قسمت، چهار آزمایش طراحی و انجام شده است که ابعاد لایه مخفی هر یک به ترتیب ۱، ۵، ۱۰، و ۱۵ واحد بوده و برای جلوگیری از بیش برآذش در همه موارد ۱۰٪ drop outing اعمال شده است. همچنین از جایی که تعداد داده‌های آموزشی بسیار کم است، تعداد داده‌های هر batch در آموزش ۱۶ تعیین شده و بر اساس آموخته‌ها از تمارین قبلی از بهینه‌ساز Adam با نرخ یادگیری پایین ۰.۰۰۰۷۵ و از تابع هزینه binary crossentropy بهره گرفته شده است.

طول آموزش ۳۰۰ تکرار تعیین شده است که محدود کننده‌ی آن Early stopping می‌باشد که در صورتی که در ۲۵ تکرار اخیر مقدار خطای اعتبار سنجی کاهش نداشته باشد آموزش را متوقف کرده و وزن‌ها را به بهترین حالت بازنمانی می‌کند تا از بیش برآذش جلوگیری شده و وزن‌های ایده‌آل تری حاصل شود.

نتایج آزمایش‌های انجام شده در زیر آمده است و در قسمت title هر یک جزئیات آموزش انجام شده درج شده است. از نتایج حاصل و نمودار‌های حین یادگیری و افزایش فاصله بین خطوط نمودار loss ضمن افزایش واحد‌ها می‌توانیم استدلال کنیم که با افزایش ابعاد لایه‌ی مخفی، قدرت و پیچیدگی شبکه افزایش یافته و به سمت بیش برآذش حرکت می‌کند.



Confusion Matrix for (Test) | Accuracy=53.33% | Precision=0.28 | LSTM - units:10 | dr:0.1



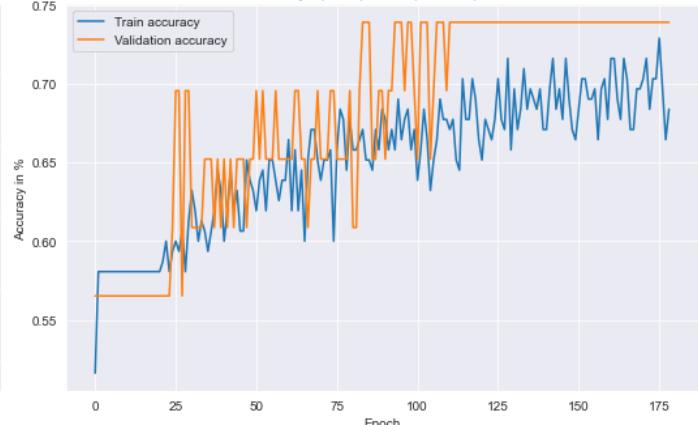
Confusion Matrix for (Test) | Accuracy=55.56% | Precision=0.52 | LSTM - units:15 | dr:0.1



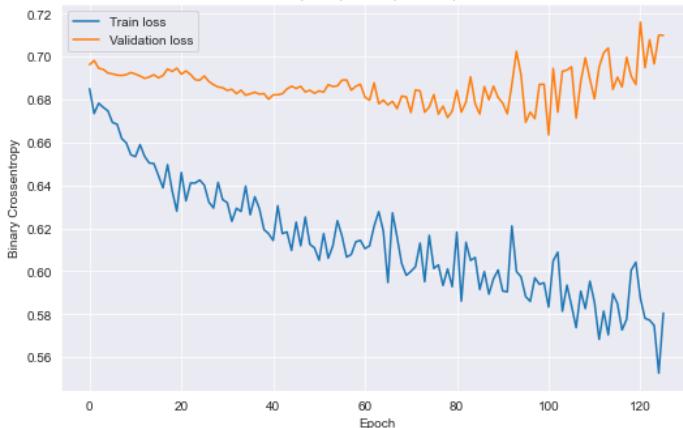
Loss/Epoch | LSTM | units:1 | dr:0.1



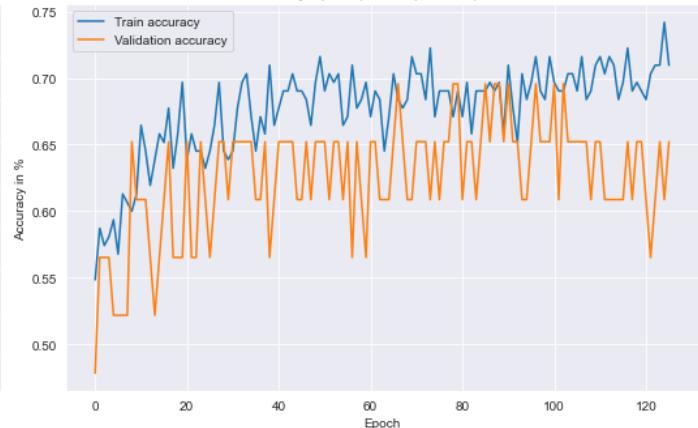
Accuracy/Epoch | LSTM | units:1 | dr:0.1



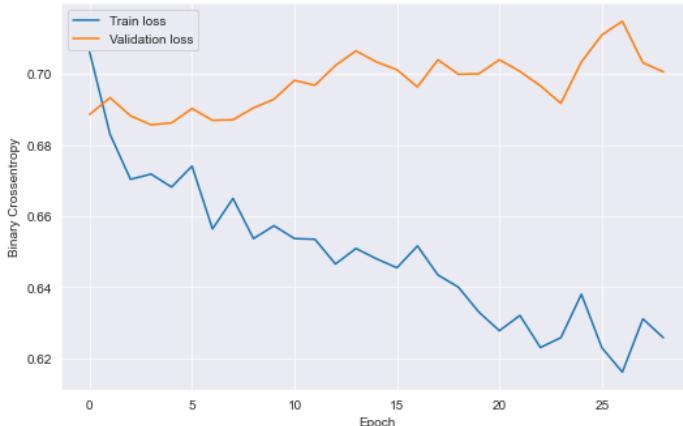
Loss/Epoch | LSTM | units:5 | dr:0.1



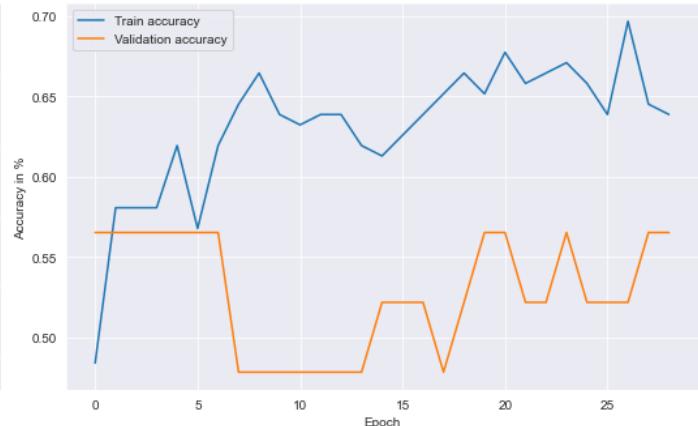
Accuracy/Epoch | LSTM | units:5 | dr:0.1

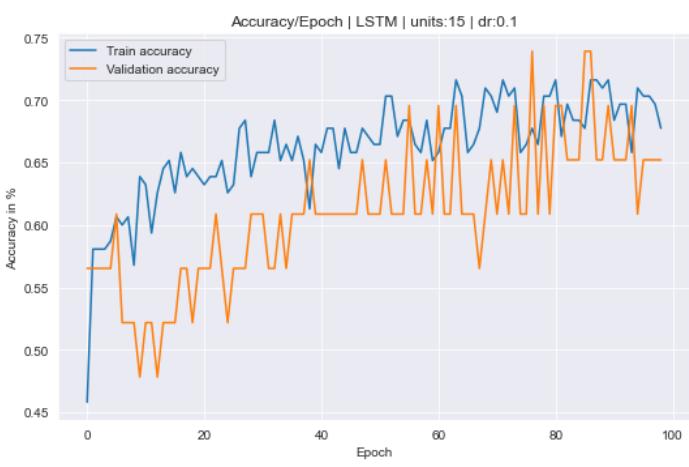


Loss/Epoch | LSTM | units:10 | dr:0.1



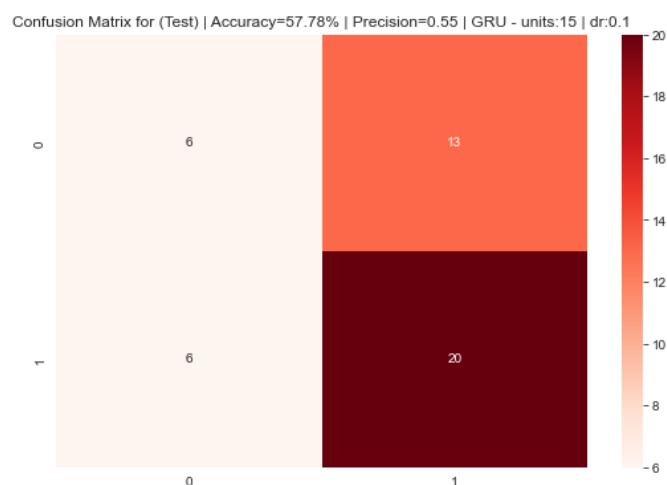
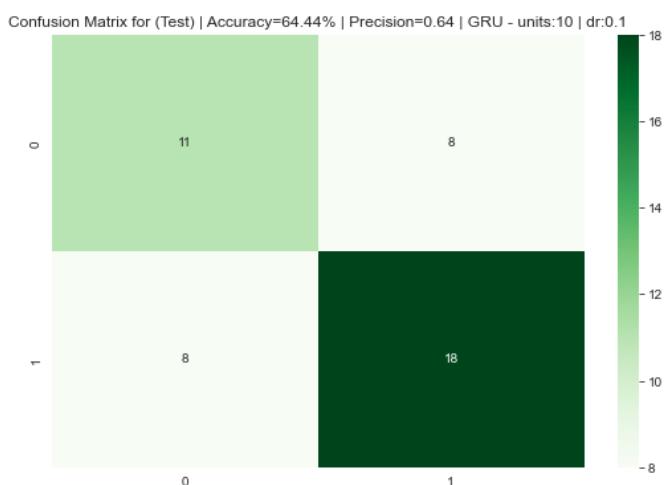
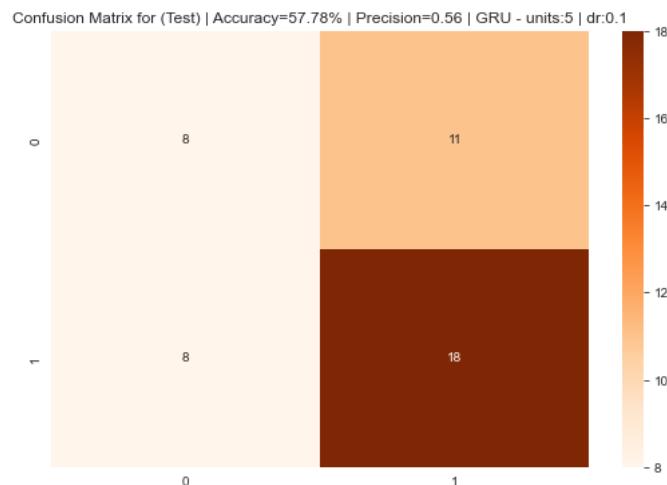
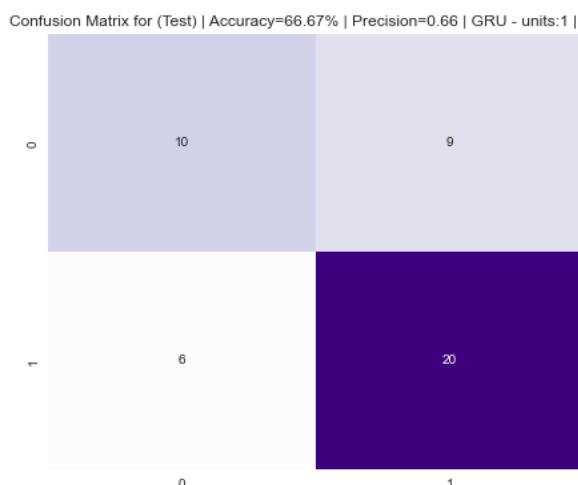
Accuracy/Epoch | LSTM | units:10 | dr:0.1

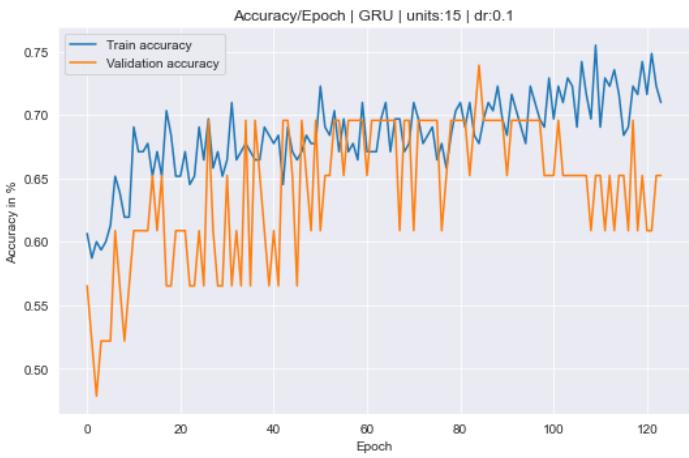
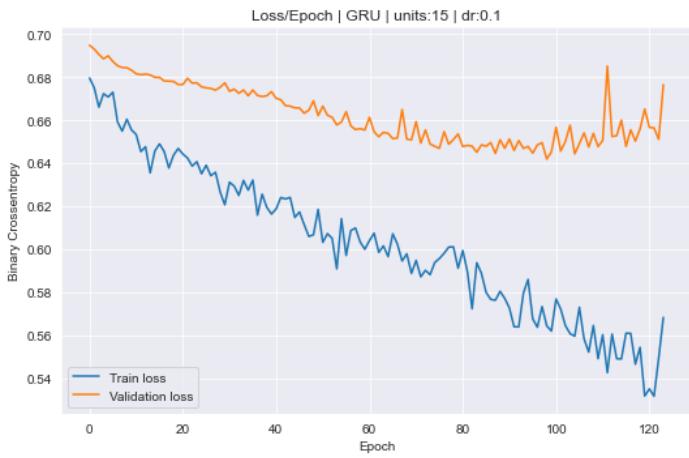
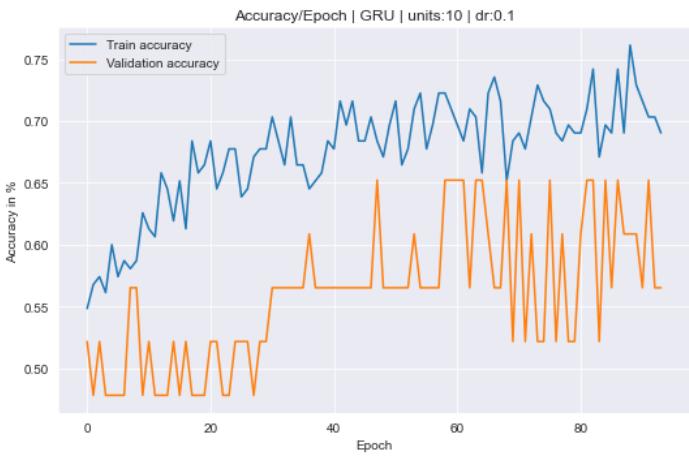
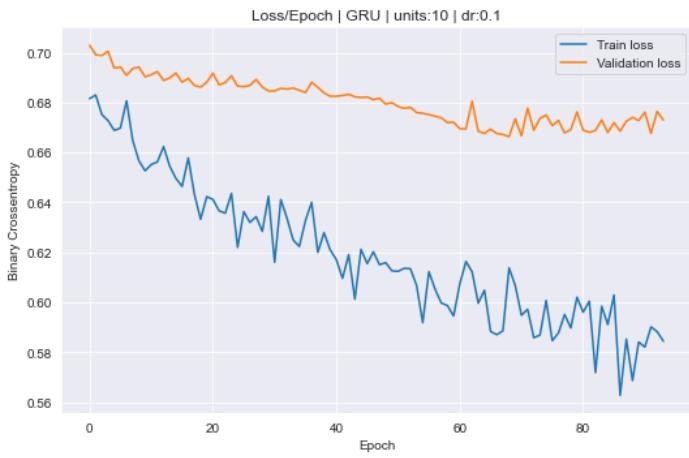
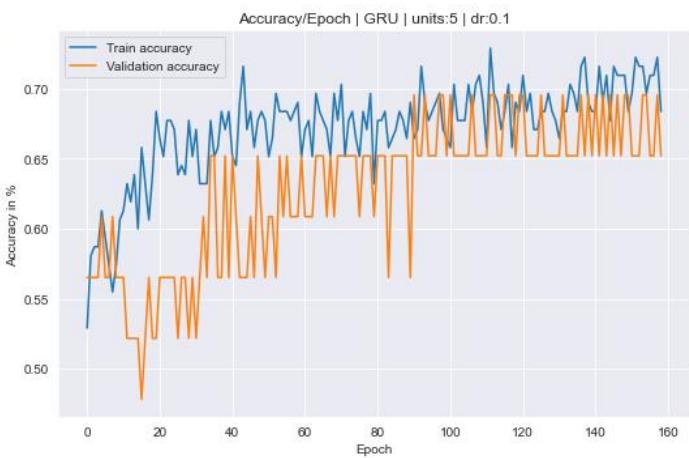
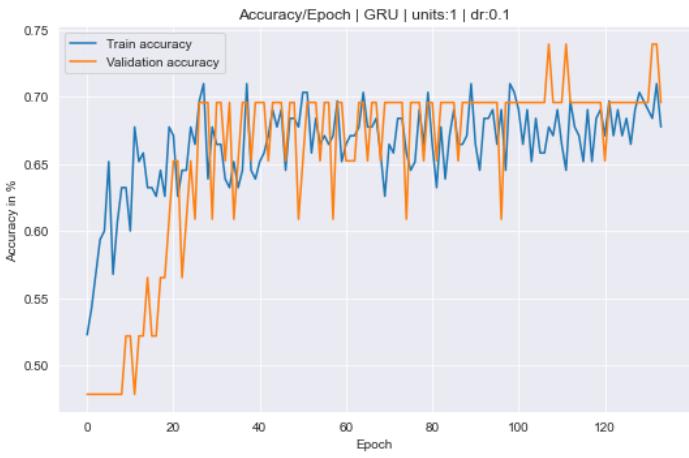




## مسئله دوم: قسمت دوم

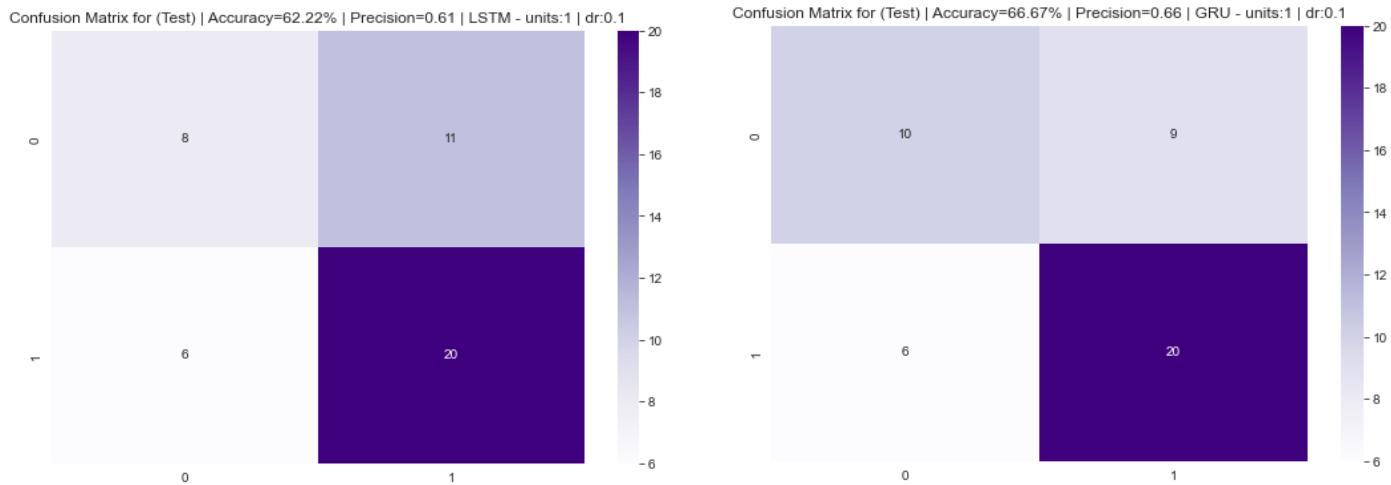
برای پاسخ گویی به این بخش، دقیقاً فرآیند مشابه با قسمت قبل را به ازای لایه‌ی GRU تکرار می‌کنیم. نتایج آزمایش‌های انجام شده در زیر آمده است و در قسمت title هر یک جزئیات آموزش انجام شده درج شده است. همانند نتایج شبکه LSTM، از نتایج حاصل و نمودارهای حین یادگیری و افزایش فاصله بین خطوط نمودار loss ضمن افزایش واحد‌ها می‌توانیم استدلال کنیم که با افزایش ابعاد لایه‌ی مخفی، قدرت و پیچیدگی شبکه GRU نیز افزایش یافته و به سمت بیش برآش حرکت می‌کند.





## مسئله دوم: قسمت سوم

بهترین عملکرد مربوط به هر یک از شبکه های LSTM و GRU در زیر آورده شده است:



مالحظه می شود که بهترین عملکرد GRU در صحت پیش‌بینی ۴۴٪ بهتر از بهترین عملکرد LSTM حاصل شده است؛ همچنین به طور میانگین و در مقایسه تمامی آزمایش ها به ازای تعداد واحد های مختلف نیز GRU توانسته است بهتر از LSTM ظاهر شود لذا می توان گفت عملکرد کلی GRU برای این مجموعه داده بهتر از LSTM بوده است؛ هر چند در هر دو نشانه های بیش برآش وجود داشته و با drop outing یکسان تلاش شده است این مورد کنترل و محدود شود.

دلیل مورد فوق را می توان بدین گونه توجیه نمود که با توجه به پیچیدگی بالای LSTM در اتصالات و استعداد بالا در بیش برآش برای مجموعه داده های کوچک، نمیتوان آن را گزینه مناسبی برای پیش‌بینی روند رشد شاخص بورس در این مسئله دانست. در نقطه مقابل، GRU که پیچیدگی کمتری نسبت به LSTM دارد، کمتر دچار بیش برآش شده و میتواند در مقابله با داده های Unseen بهتر عمل کردد و به دقت پیش‌بینی بالاتری دست یابد لذا تفاوت حاصل ناشی از سطح پیچیدگی شبکه ها برای آموزش می باشد.

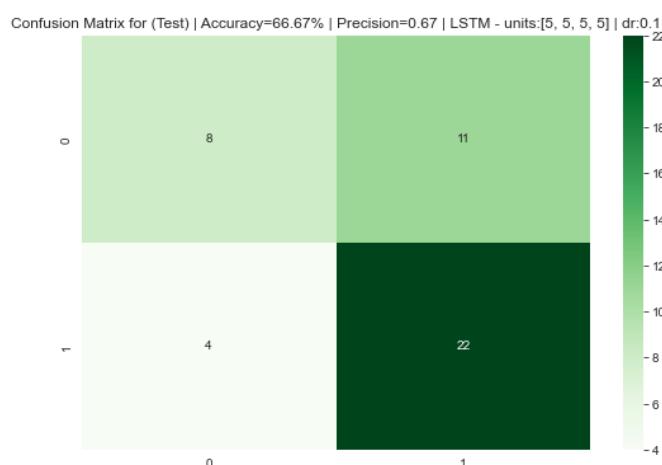
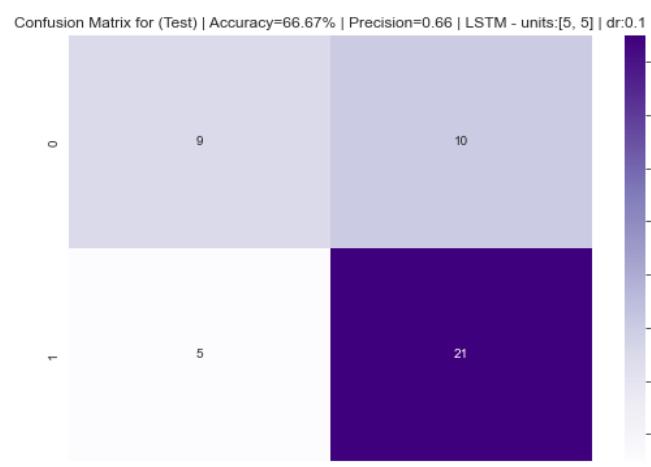
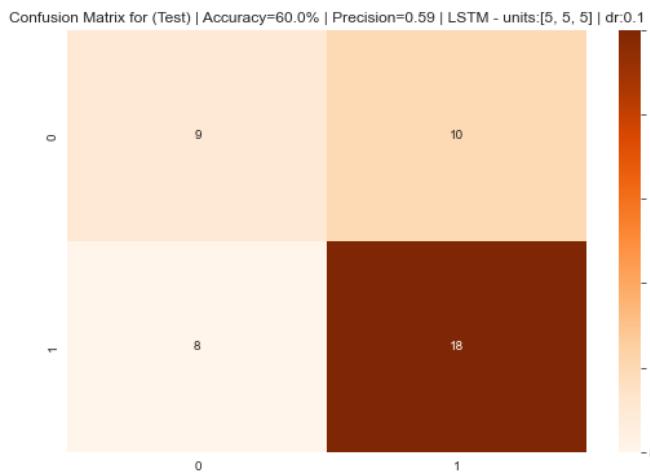
## مسئله سوم

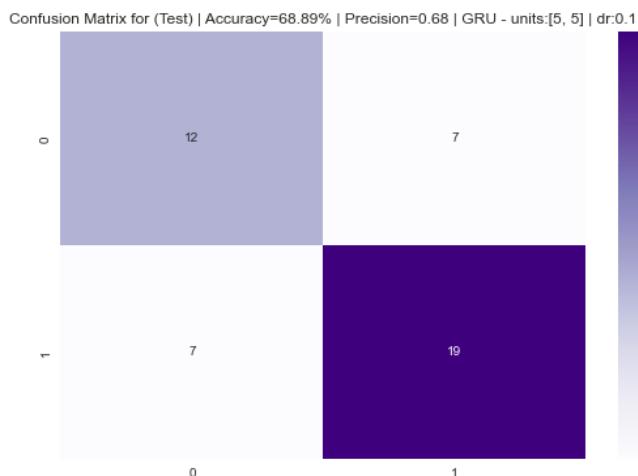
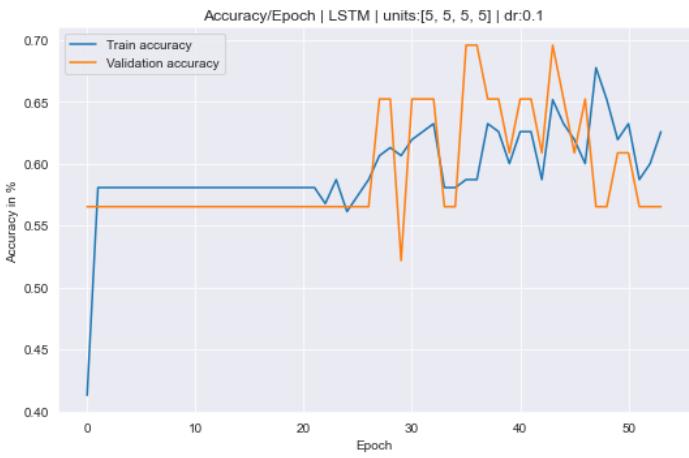
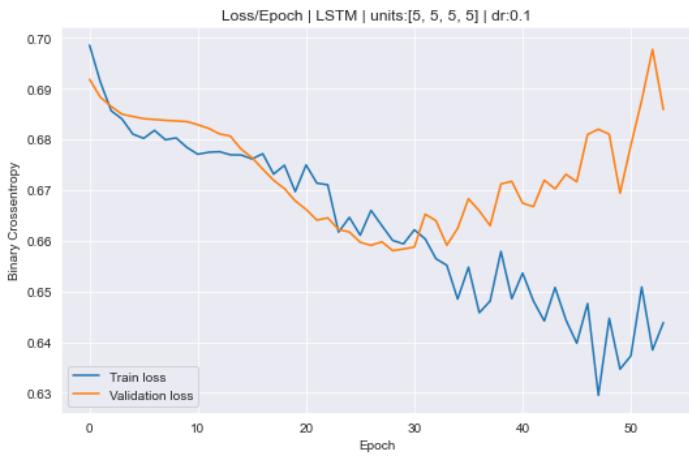
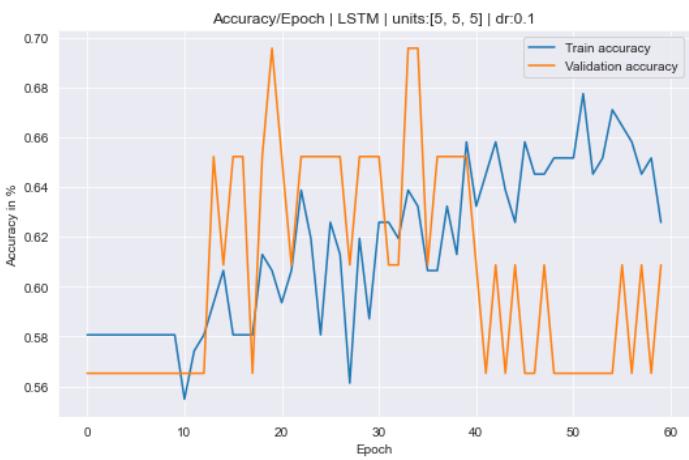
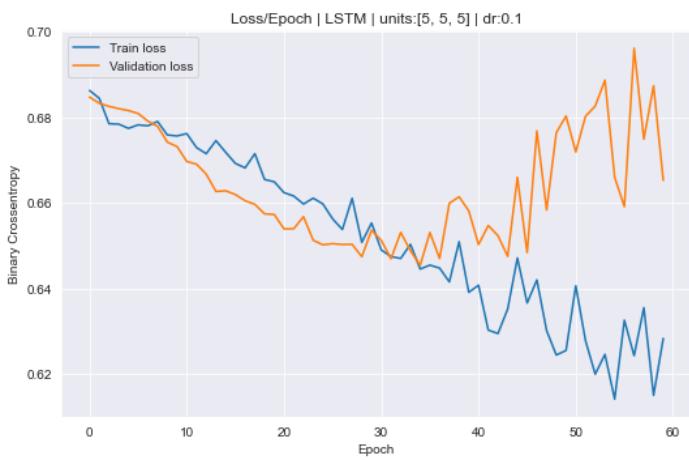
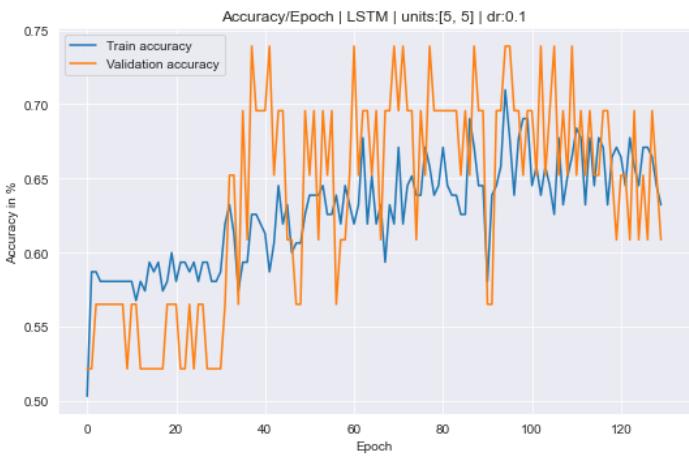
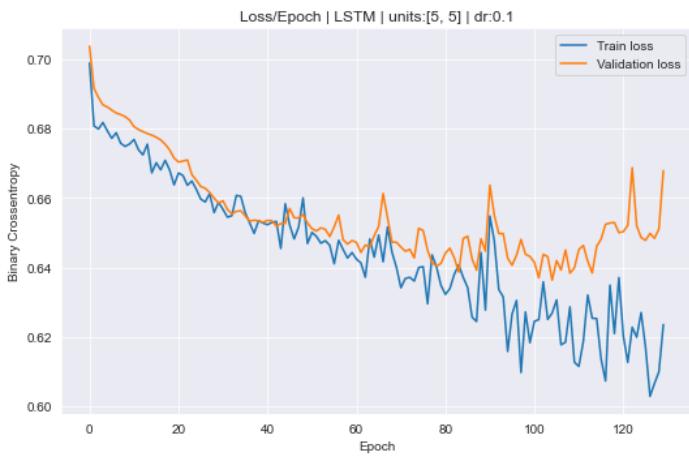
### مسئله سوم: قسمت اول

برای پارامتر های آموزشی این قسمت همانند توضیحات ارائه شده در مسئله دوم اقدام و استفاده می کنیم؛ برای پشته کردن هر یک از دو شبکه مورد نظر LSTM و GRU طی ۳ آزمایش شبکه هایی با عمق ۲، ۳، ۴ را آموزش می دهیم که در هر عمق ۵ واحد قرار گرفته شده است.

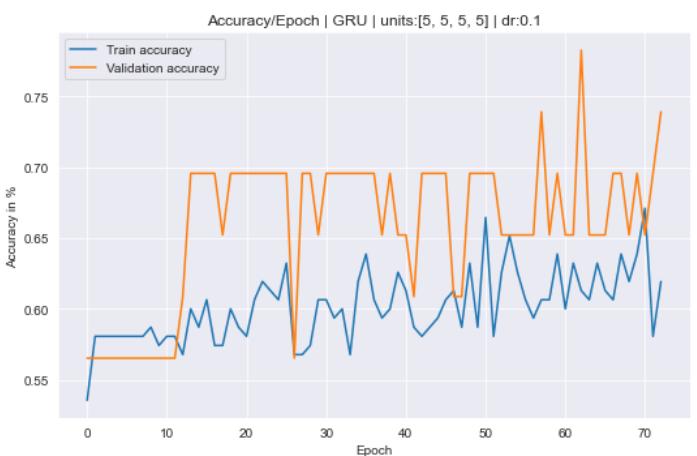
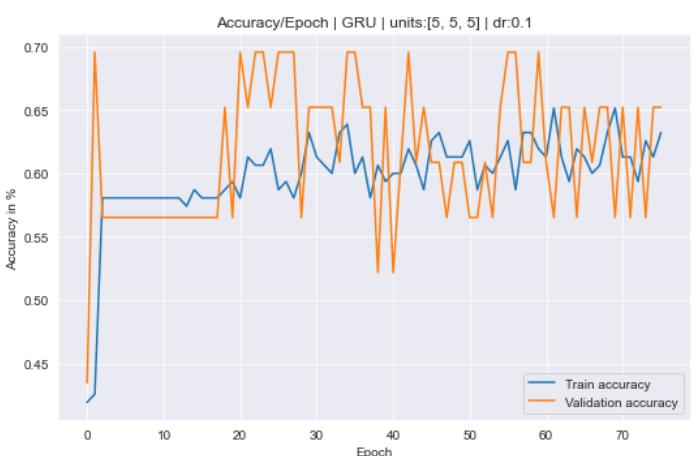
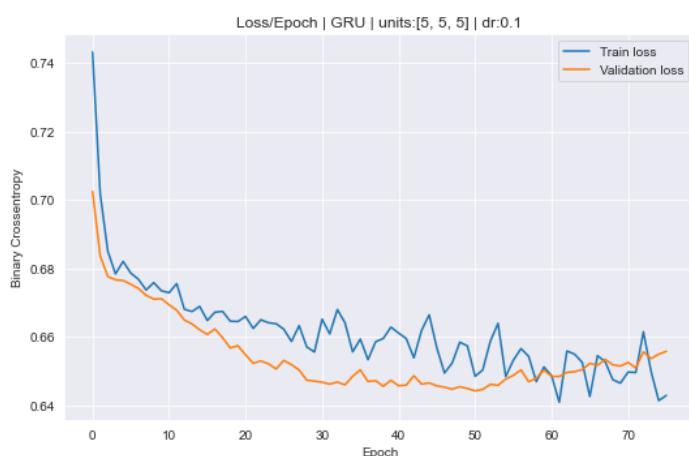
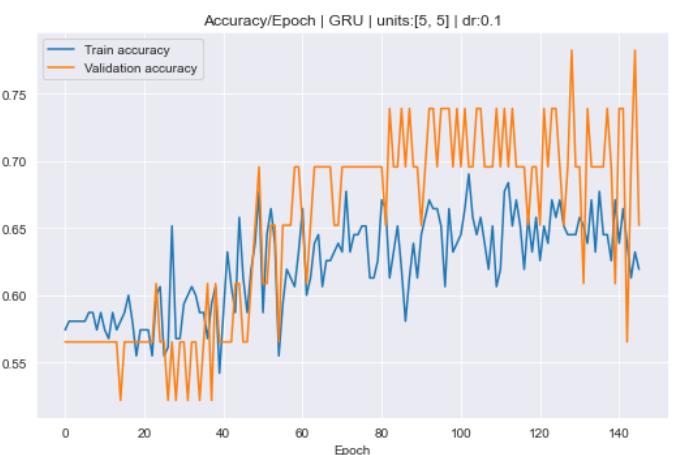
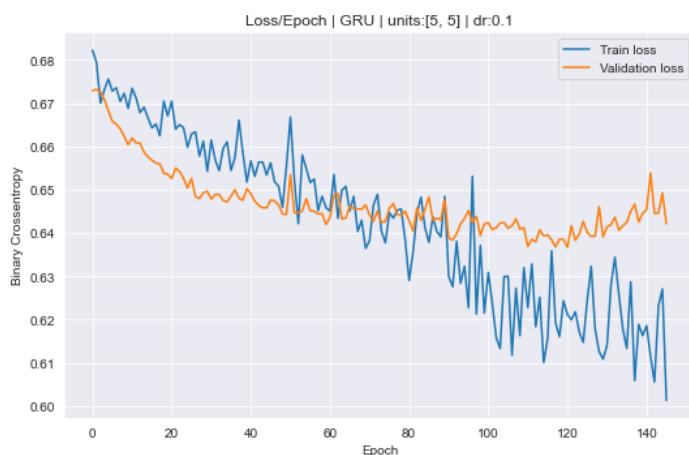
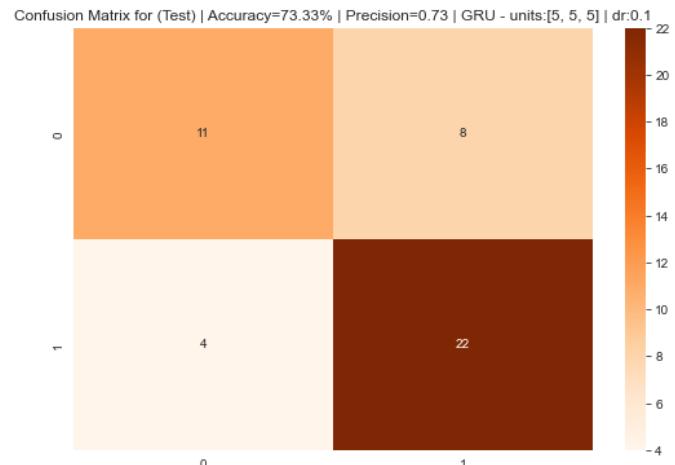
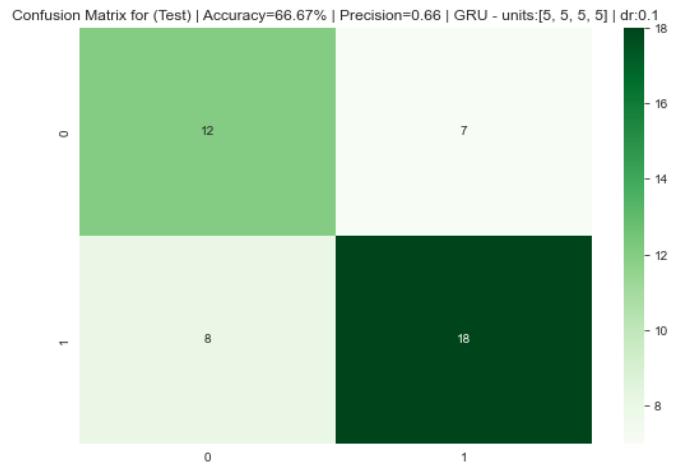
نتایج آزمایش های انجام شده در زیر آمده است و در قسمت title هر یک جزئیات آموزش انجام شده درج شده است. با توجه به نتایج حاصل قابل مشاهده است که افزایش عمق در هر دو مدل LSTM و GRU باعث افزایش تقریبی عملکرد مدل ها و دقت پیش بینی برای داده های Unseen نسبت به حالت پایه‌ی خود شده است. همچنین سرعت سوق به بیش برآذش در حالت عمیق نسبت به قسمت قبل تا حدود اندکی کمتر و کنترل شده است. قابل ذکر است در این حالت نیز عملکرد GRU بهتر از LSTM بوده و با اختلاف تقریبی ۶.۷٪ توانسته است دقت بالاتری برای پیش بین داده های آزمون به ارمغان آورد.

آزمایش های پشته ای مربوط به شبکه : LSTM



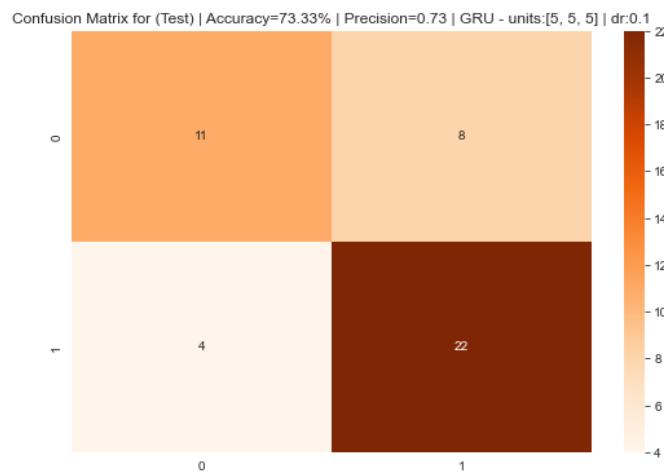
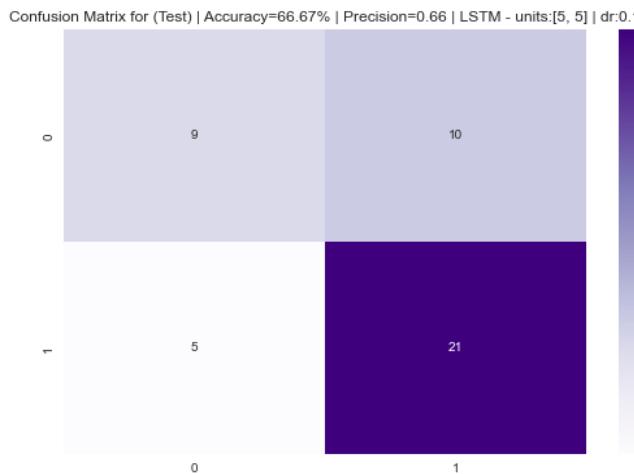


آزمایش های پشتۀ ای مربوط به شبکه GRU :



## مسئله سوم: قسمت دوم

برای پاسخگویی به این پرسش که عمق کدام یک از شبکه های LSTM و GRU را میتوان بیشتر افزایش داد، بهترین عملکرد هر یک را که در قسمت قبل حاصل شده است مجدد در این قسمت می آوریم:



اگر معیار ارزیابی برای افزایش عمق در شبکه های فوق را دقت پیش بینی داده های unseen قرار دهیم، نتیجه میگیریم که پشتنهای GRU توانسته است با اختلاف تقریبی ۷٪ عملکرد بهتری را از خود به جا گذارد. اگر معیار ارزیابی را عدم بیش برازش در نمودار loss قرار دهیم، همچنان پشتنهای GRU می تواند با افزایش عمق کمتر دچار بیش برازش شده و فاصله خطوط آن از هم زیاد نشود در حالی که در LSTM با افزایش عمق، سرعت بیش برازش بیشتر شده و خطوط نمودار سریع تر از هم واگرا می شوند؛ لذا با عنایت به موارد فوق به GRU و در این مجموعه داده میتوان عمق بیشتری دارد.

دلیل مورد فوق را میتوان پیچیدگی پایین ذات GRU و اتصالات کم آن دانست که باعث می شود پشتنه کردن آن ضمن انتقال اطلاعات بین گام های زمانی، ویژگی های استخراجی و hidden state منتقل شده به عمق بعدی شفاف تر و حاوی اطلاعات مفید و غیر اضافی برای بیش برازش باشد.