

گزارش تمرین پنجم شبکه عصبی

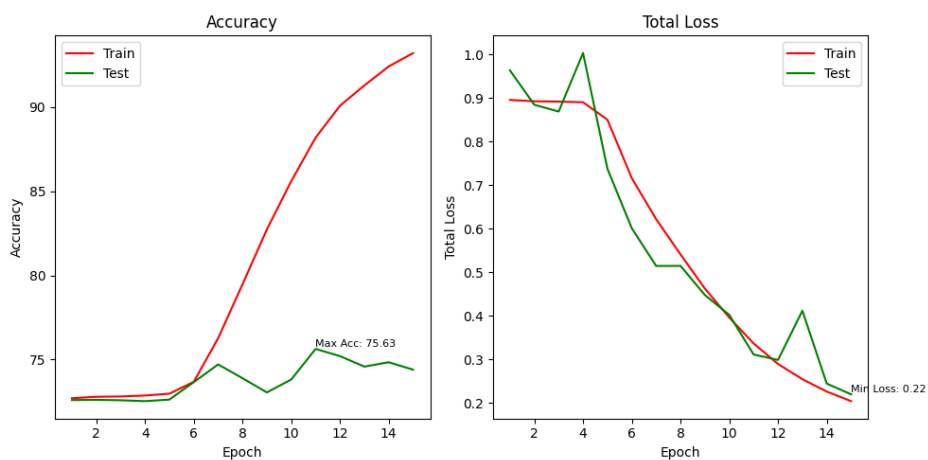
علیرضا امینی

۱ بخش اول

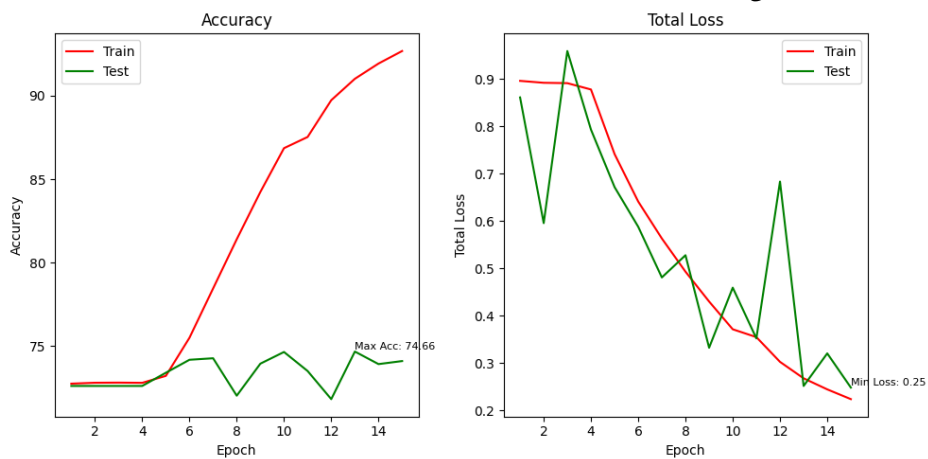
در این تمرین قصد داریم تا یک سیستم طبقه بند sentiment را برای داده های مربوط به نظرات کالاهای amazoon را با استفاده از RNN تعلیم دهیم. دیتاست مربوط به لوازم Industrial and Scientific از زیرمجموعه داده های 5-core دیتاست Amazon Review Data (2018) به عنوان دیتاست هدف انتخاب شد. از این دیتاست قسمت reviewText به عنوان متن اصلی نظرات (داده) و قسمت overall به عنوان لیبل آن داده که عددی بین ۱ تا ۵ است، استخراج شدند.

برای اینکه بتوان داده های متن دار را به شبکه های موردنظر به عنوان ورودی داد، ابتدا باید به عدد تبدیل شود. برای تبدیل رشته های ورودی به عدد، ابتدا باید رشته ها tokenize شوند. برای tokenize شدن از spacy و یکی از لغت نامه های آن استفاده می کنیم. همچنین برای کلماتی که در لغت نامه وجود ندارند از توکن <UNK> استفاده شده است.

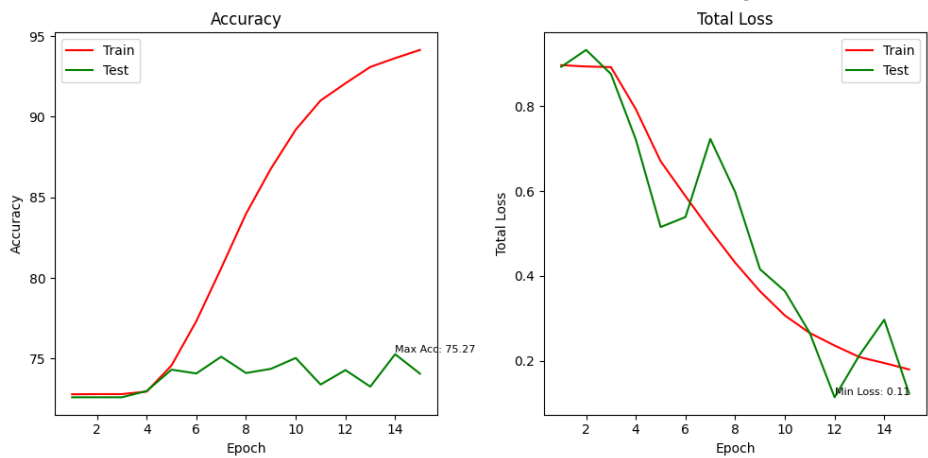
در ادامه عملکرد انواع شبکه های RNN، LSTM و GRU با تعداد لایه های مختلف ارزیابی شده است. نتایج نمودار تغییرات دقت و خطا شبکه LSTM با تعداد لایه های ۱، ۲ و ۳ در شکل های ۱، ۲ و ۳ قابل مشاهده است. همچنین نتایج مربوط به RNN در شکل های ۴، ۵ و ۶ و نتایج مربوط به GRU به شبکه نیز در شکل های ۷، ۸ و ۹ نشان داده شده است.



شکل ۱: نمودار تغییرات دقت و خطا شبکه LSTM یک لایه



شکل ۲: نمودار تغییرات دقت و خطا شبکه LSTM دو لایه



شکل ۳: نمودار تغییرات دقت و خطا شبکه LSTM سه لایه

همان طور که مشاهده شبکه RNN با هیچ کدام از تعداد لایه ها قادر نیست یادگیری مناسبی را بر روی این داده ها انجام دهند. همچنین شبکه های LSTM و مخصوصا GRU بر روی داده ها یادگیری خوبی داشتند و توانستند تابع هزینه را هم برای داده های آموزشی و هم برای داده های تست کاهش دهند. اما همان طور که مشاهده می شوند مدل بر روی داده های آموزشی overfit شده است و بعد از ایپاک های ابتدایی بهبود دقتی برای داده های تست مشاهده نمی شود. علت آن نیز می تواند کمبود داده های آموزشی به نسبت داده های تست و validation و یا نبودن داده های مربوط به کلاس های مختلف باشد که باعث می شود خروجی نسبت به یک کلاس با داده های بیشتر سوق پیدا کند.

مقایسه عملکرد شبکه ها با تعداد لایه های مختلف بر اساس بهترین دقت داده های validation در جدول ۱ نشان داده شده است. در شبکه LSTM، بهترین عملکرد مربوط به شبکه ۱ لایه است و بعد از آن شبکه ۳ لایه به بهترین دقت رسیده است. در شبکه RNN، تغییر تعداد لایه های تأثیری در عملکرد شبکه نداشته است. در شبکه GRU، ۲ لایه به بهترین دقت رسیده است و عملکرد شبکه ۳ لایه نیز از شبکه ۱ لایه بهتر بوده است.

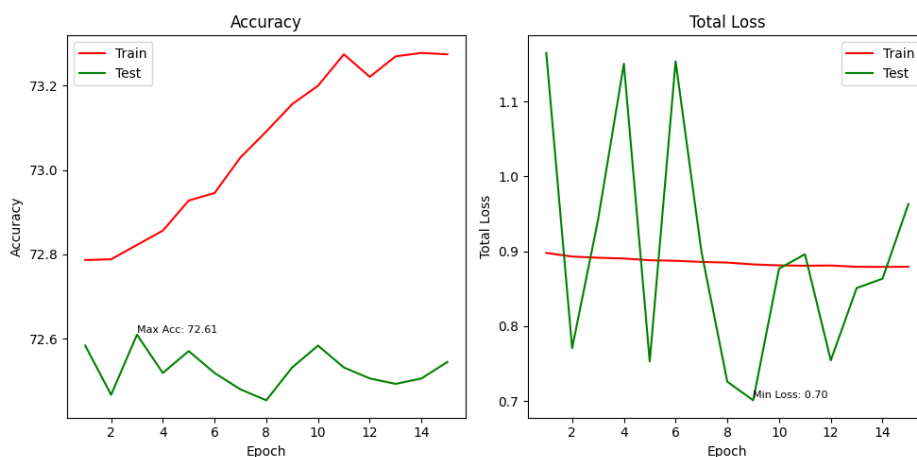
در نهایت بهترین عملکرد مربوط به شبکه GRU با ۲ لایه است که به دقت 77.01 بر روی داده های validation رسیده است.

شبکه	تعداد لایه ها	دقت تست
LSTM	۱	75.63
LSTM	۲	74.66
LSTM	۳	75.27
RNN	۱	72.61
RNN	۲	72.62
RNN	۳	72.61
GRU	۱	76.20
GRU	۲	77.01
GRU	۳	76.42

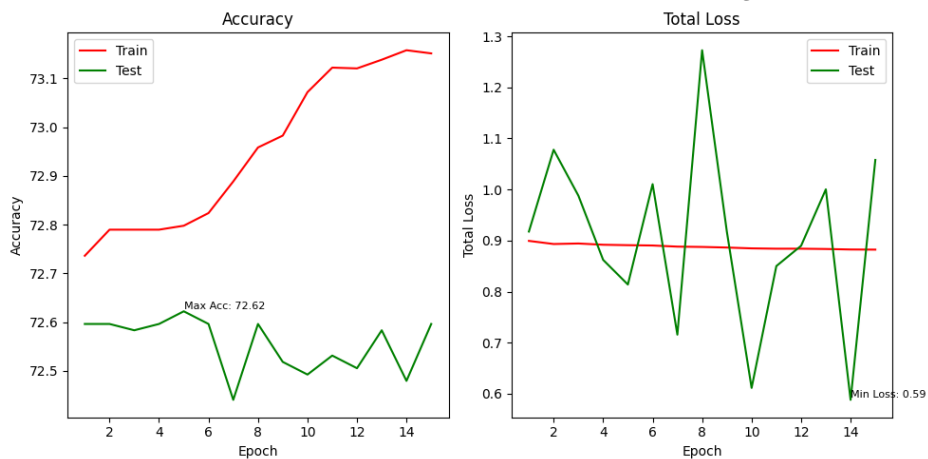
جدول ۱: مقایسه عملکرد شبکه ها با تعداد لایه های مختلف

۲ بخش دوم

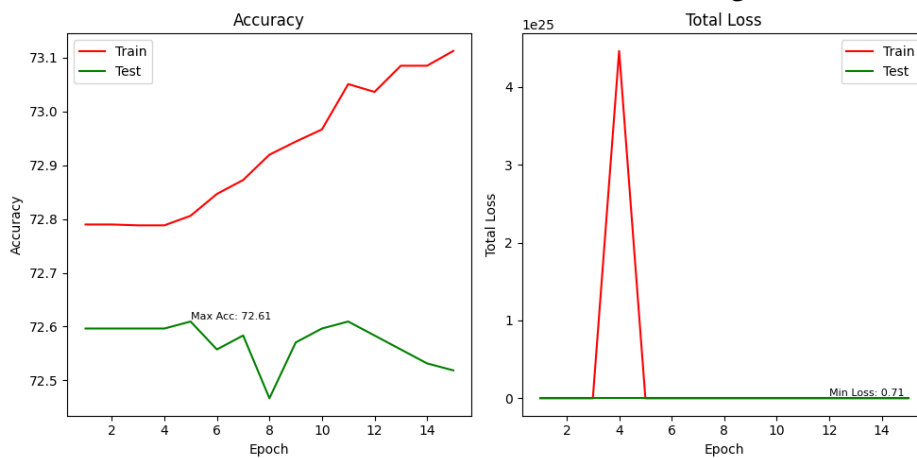
در این قسمت معماری شبکه ای که بهترین عملکرد را در قسمت اول داشت یعنی شبکه GRU دو لایه را انتخاب می کنیم. سپس داده های مربوط به کلاس ۲ و ۴ را از دیتاست آموزش حذف می کنیم و شبکه را مجدد از ابتدا آموزش می دهیم. همان طور که مشاهده می شود چون شبیه در فرآیند آموزش هیچ کدام از داده های کلاس ۲ و ۴ را ندیده است، بنابراین نمی تواند score و لیل صحیح متناظر با آن ها را برگرداند، بنابراین در فرآیند آموزش برای هر ایپاک دقت داده های تست به صفر می رسد. نمودار تغییرات دقت و خطای شبکه به ازای داده های آموزش و تست در شکل ۱۰ مشاهده می شود.



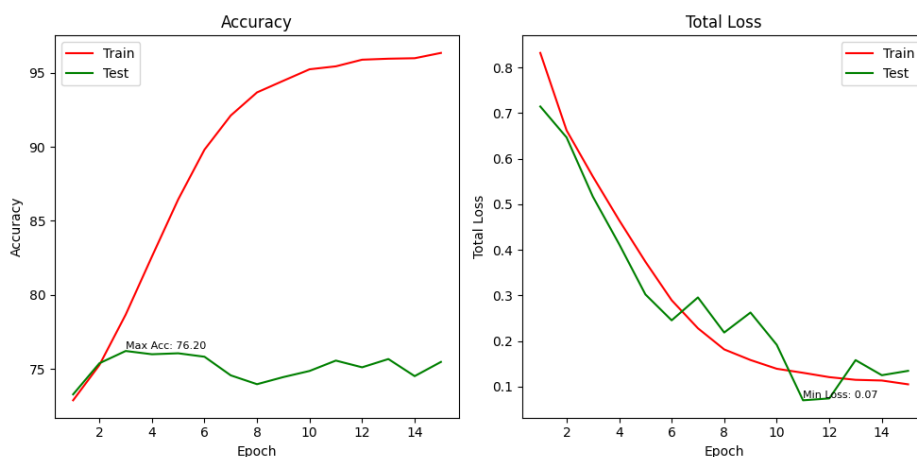
شکل ۴: نمودار تغییرات دقت و خطا شبکه RNN یک لایه



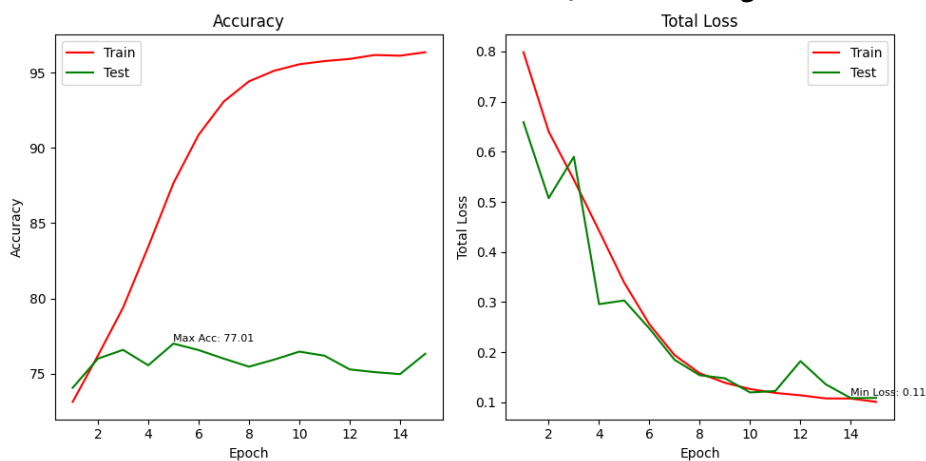
شکل ۵: نمودار تغییرات دقت و خطا شبکه RNN دو لایه



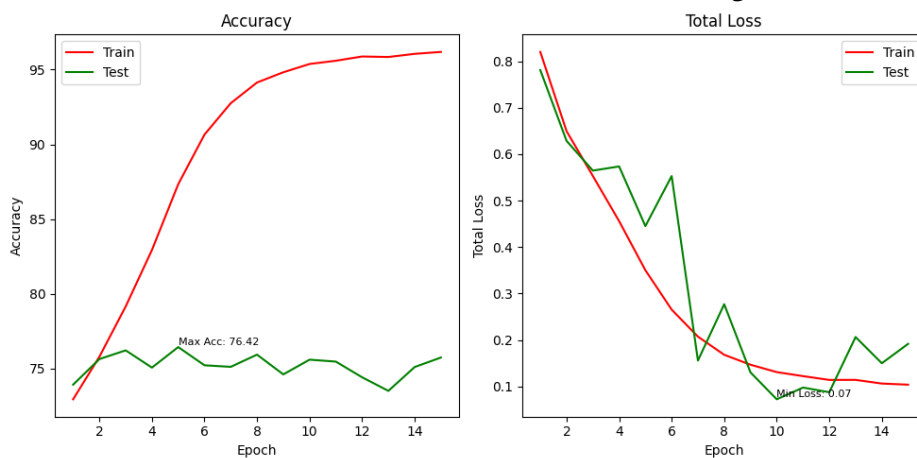
شکل ۶: نمودار تغییرات دقت و خطا شبکه RNN سه لایه



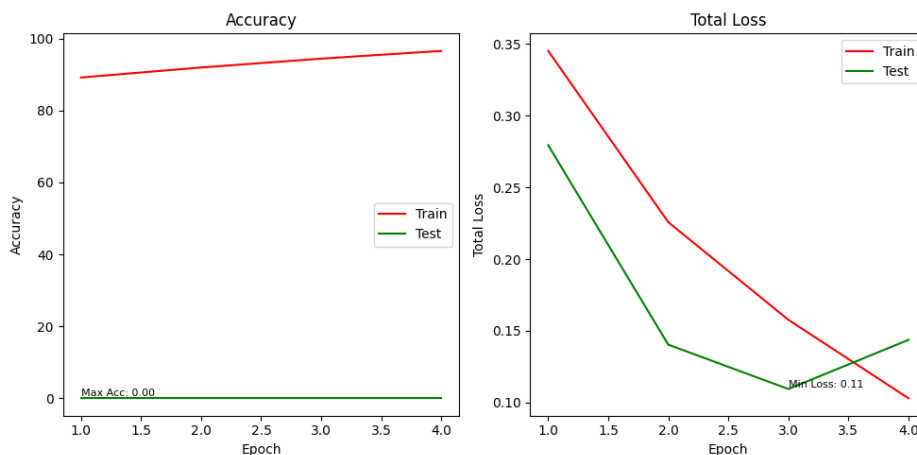
شکل ۷: نمودار تغییرات دقت و خطا شبکه GRU یک لایه



شکل ۸: نمودار تغییرات دقت و خطا شبکه GRU دو لایه



شکل ۹: نمودار تغییرات دقت و خطا شبکه GRU سه لایه



شکل ۱۰: نمودار تغییرات دقت و خطا شبکه آموزش داده شده با داده کلاس های ۱، ۳ و ۵ و تست شده با داده کلاس های ۲ و ۴

همچنین عملکرد مجموعه شبکه آموزش داده شده با داده های سه کلاس ۱ و ۳ و ۵ بر روی دیتاست IMDB ارزیابی می کنیم. این دیتاست شامل نظرات IMDB است که لیبل های آن شامل positive و negative است. برای لود داده ها به صورت batch لیبل های مثبت به 1 و لیبل های منفی به 0 تبدیل شده اند.

برای ارزیابی عملکرد شبکه در مقابل داده های دیتاست IMDB، یک batch از این داده ها رو انتخاب می کنیم و از شبکه عبور می دهیم و خروجی شبکه را برای هر کدام می گیریم و در نهایت با لیبل اصلی آن ها مقایسه می کنیم. این کار را برای مدل های آموزش داده شده در اپیک های مختلف تست شد و بهترین عملکرد متعلق به مدل ذخیره شده در اپیک چهارم بود که نتایج آن در شکل ۱۱ قابل مشاهده می باشد.

```
[0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0]
[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 4, 4, 2, 0, 3]
```

شکل ۱۱: نتایج مدل بر روی داده IMDB - آرایه بالایی، برچسب های داده ها و آرایه پایینی خروجی مدل

همان طور که مشاهده می شود خروجی برای تمامی داده های مثبت کلاس ۵ (۴ در پیاده سازی) می باشد که خروجی موردانتظار است ولی برای داده های منفی تعدادی از خروجی ها ۴، و تعدادی از آن ها ۰ یا ۲ می باشد. علت خطا نیز می تواند این باشد که مدل مجدداً به علت بالانس نبودن داده ها، overfit شده است.