تمرین سوم پایتون
شبکه‌های عصبی بهار ۱۴۰۲

آریا ابراهیمی ۹۸۲۲۷۶۲۱۷۵

هدف از این تمرین، بررسی شبکه‌های عصبی بازگشتی (RNN) بر روی داده های متنی می‌باشد. در این تمرین از دادگان مربوط به نظرهای کالاهای Amazon استفاده شده است. در ابتدا یک طبقه‌بندی با پنج کلاس را با استفاده از LSTM و GRU انجام می‌دهیم و آن‌ها را با یکدیگر مقایسه می‌کنیم. در ادامه تاثیر افزایش unit ها را بررسی کرده و درنهایت استفاده از بازنمایی‌های پیش یادگیری شده با استفاده از GloVe را بررسی می‌کنیم.

در بخش بعدی، به جای آموزش بر روی تمامی کلاس‌های مربوط به داده‌های Amazon، روی تعداد کمتری از کلاس‌ها آموزش را انجام می‌دهیم و پیش‌بینی شبکه را برای کلاس‌هایی که در یادگیری مشاهده نکرده‌است را بررسی می‌کنیم.

بخش اول

۱- پیش‌پردازش دادگان

در ابتدا نیاز است تا دادگان را در محیط colab بارگذاری کنیم. از آنجایی که دادگان مربوط به نظرهای Amazon بزرگ است، از بخش کوچکی از آن که مربوط به دسته Toys and Games است استفاده شده است که تقریبا شامل ۱۷۰۰۰۰۰ نظر می‌باشد که شامل قسمت های زیادی می‌شود. دو قسمت اصلی داده ها که ما نیاز داریم ۱- متن نظر و ۲- امتیاز هستند. متن نظر شامل یک string می‌شود که اندازه آن نامشخص است و برای داده های متفاوت، فرق می‌کند و امتیاز می‌تواند شامل پنج مقدار از عدد ۱ تا عدد ۵ شود که ۱ کمترین امتیاز و ۵ بیشترین امتیاز است.

بعد از لود کردن دادگان، نیاز به پیش پردازش آن‌ها است. فرمت این داده ها json است که با استفاده از gzip فشرده‌سازی شده‌اند. تابع read_data پیاده سازی شده است که ابتدا داده‌ها را خوانده و فقط قسمت‌های مربوط به امتیاز و متن نظر را تبدیل به یک dataframe می‌کند. در ادامه برای پاک‌سازی داده‌ها، مراحلی طی می‌شود که شامل حذف کردن تگ‌های html، حذف stopword ها، استفاده از حروف کوچک و تبدیل کلمات به یک شکل واحد با استفاده از setmmer می‌شود.

بعد از دریافت دادگان، باید آن‌ها را برای ورودی دادن به شبکه آماده کرد. از آنجایی که فقط داده‌های عددی را می‌توان به شبکه داد، باید یک دیکشنری از کلمات موجود ساخته شود که یک عدد را به هر کلمه نگاشت کند. برای این کار می‌توان از Tokenizer استفاده کرد که هر کلمه‌را به یک عدد نگاشت می‌کند.

از آنجایی که داده ها اندازه های متفاوتی دارند، نیاز است تا آن‌ها را به یک اندازه واحد تبدیل کنیم. برای این کار میانگین طول نظرات را به دست آورده و تمامی داده ها را به آن اندازه تبدیل می‌کنیم. برای داده‌هایی که اندازه کوچک‌تر از این مقدار دارند، عدد ۰ قرار می‌دهیم تا به اندازه مورد نظر برسند و برای داده‌هایی که بزرگتر هستند، یک برش به اندازه مورد نظر انجام می‌دهیم. بعد از انجام این مرحله داده ها برای استفاده در شبکه آماده هستند.

۲- پیاده‌سازی شبکه

برای این قسمت ابتدا یک لایه Embedding اضافه می‌کنیم که سایز دیکشنری ایجاد شده توسط Tokenizer و اندازه بردارهای Embedding را به عنوان ورودی دریافت می‌کند. در ادامه یک لایه Bidirectional LSTM اضافه شده است. لایه Bidirectional این امکان را فراهم می‌کند که علاوه بر استفاده از اطلاعات کلمات قبل برای کلمات بعد، از اطلاعات کلمات آینده نیز برای کلمات قبل‌تر استفاده کرد. برای جلوگیری از overfit شدن یک لایه Dropout با مقدار ۰.۲ نیز ایجاد شده است.

در ادامه از آنجایی که داده‌های Amazon دارای پنج کلاس هستند، یک لایه Dense با تعداد ۵ نورون نیز اضافه می‌شود که از softmax به عنوان تابع فعال‌ساز استفاده می‌کند.

برای compile کردن مدل طراحی شده، از RMSprop با نرخ یادگیری ۰.۰۰۱ استفاده شده است. دلیل عدم استفاده از Adam این است که در تست‌های انجام شده در مقایسه با RMSprop عمل‌کرد ضعیف‌تری داشت. برای تابع هزینه از categorical crossentropy استفاده شده است زیرا طبقه‌بندی چندکلاسه داریم.

۳- انجام آزمایش‌ها

در گام اول استفاده از GRU و LSTM را بررسی می‌کنیم. برای استفاده از GRU کافیست به جای لایه LSTM از لایه GRU استفاده کنیم و تغییر خاص دیگری نیاز نیست. در شرایط مساوی این دو لایه تقریبا عمل‌کرد مشابهی داشتند.

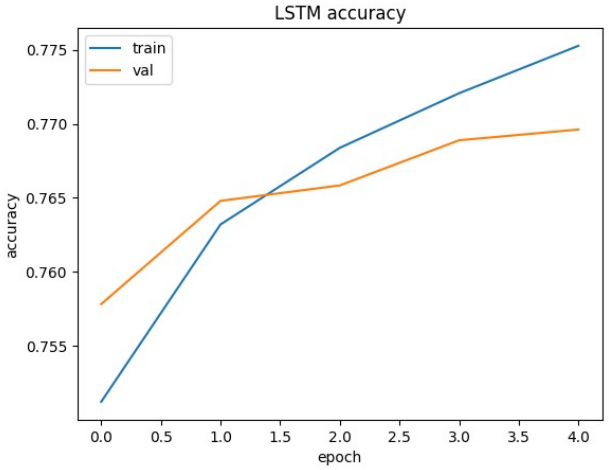

Figure ۱: نتایج مربوط به استفاده از Bidirectional LSTM برای آموزش شبکه.

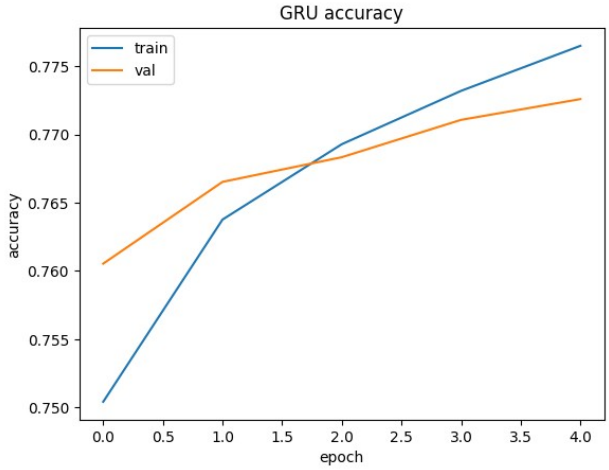

Figure ۲: نتایج مربوط به استفاده از Bidirectional GRU برای آموزش شبکه.

از آنجایی که تقریبا عمل‌کرد مشابهی دارند، برای ادامه تمرین از LSTM استفاده می‌کنیم.

در گام بعد تاثیر افزایش تعداد یونیت های LSTM را بررسی می‌کنیم. برای این آزمایش دو نسخه متفاوت از LSTM که یکی ۱۲۸ یونیت و دیگری ۵۱۲ یونیت دارد بررسی شده‌اند.

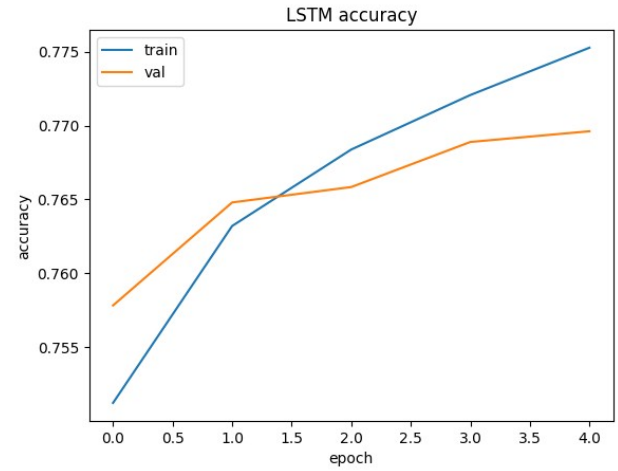

Figure ۳: نتایج مربوط به استفاده از Bidirectional LSTM برای آموزش شبکه.

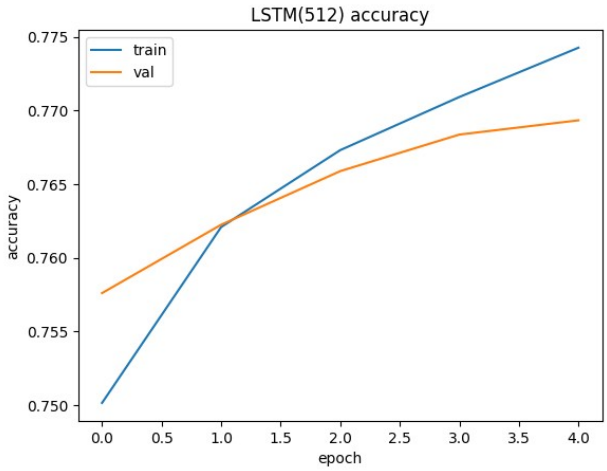

Figure ۴: نتایج مربوط به استفاده از Bidirectional LSTM(512) برای آموزش شبکه.

مشاهده می‌شود که این دو شبکه نیز تفاوت زیادی با یکدیگر ندارند. اگر شبکه ها را با داده های کمتری یادگیری کنیم(برای مثال ۴۰۰۰۰۰ نظر)، میزان دقت آن‌ها از ۶۵ درصد بیشتری نمی‌شود و خیلی سریع overfit می‌شوند. این باعث به وجود آمدن فرضیه‌ای می‌شود که bottleneck اصلی برای یادگیری در اینجا معماری شبکه نیست، بلکه میزان داده‌ها است. با بیشتری کردن تعداد epoch ها مشاهده می‌شود که همان رفتاری که با داده‌های کمتر رخ داد اتفاق میفتد و شبکه overfit می‌شود به صورتی که دقت train افزایش پیدا می‌کند ولی دقت validation در همان حدود ۷۷ درصد باقی می‌ماند و حتی کمتر نیز می‌شود. برای حل این مشکل نیاز است تا دادگان بزرگتری را استفاده کنیم که حافظه محدود colab این اجازه را نمی‌دهد و کار را با همین دادگان ادامه می‌دهیم.

در آزمایش بعدی، تاثیر استفاده از مدل های بازنمایی از پیش یادگیری شده را بررسی می‌کنیم. برای این کار مدل GLoVe-Twitter-200 استفاده شده است.به این صورت عمل می‌کنیم که یک ماتریس embedding تشکیل می‌دهیم و با پیمایش روی کلمات داخل دیکشنری که در Tokenizer قابل دسترس است، بردار مربوط به هر کلمه را از مدل GloVe دریافت می‌کنیم و داخل ماتریس embedding قرار می‌دهیم. سپس در شبکه از این ماتریس به عنوان embeddings_initializer استفاده می‌کنیم. مشاهده می‌شود که استفاده از این مدل کمی دقت مدل را بهبود داده است ولی بازهم این تغییر چشم‌گیر نیست و در ادامه از این مدل استفاده نمی‌کنیم.

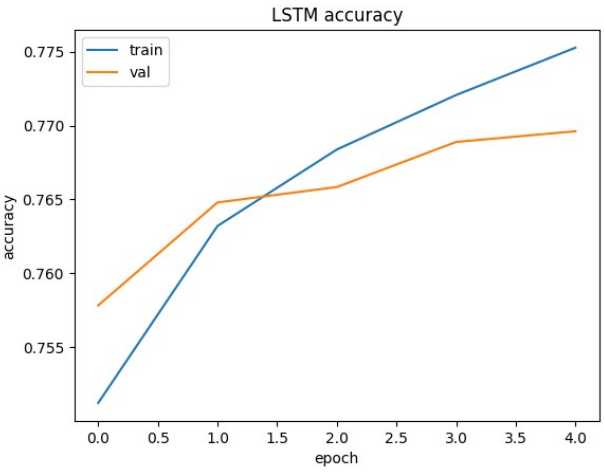

Figure ۵: نتایج مربوط به استفاده از Bidirectional LSTM برای آموزش شبکه بدون استفاده از GloVe.

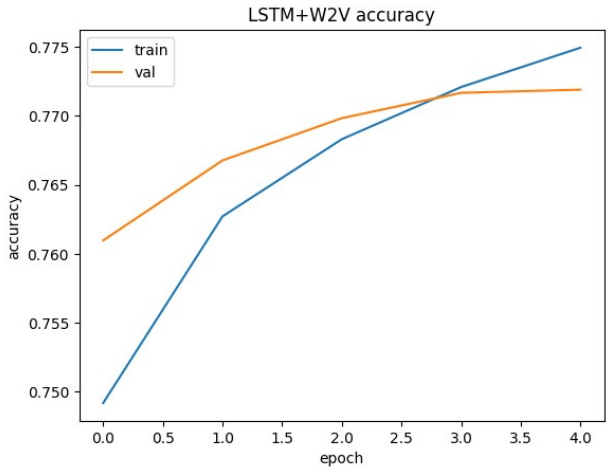

Figure ۶: نتایج مربوط به استفاده از مدل GLoVe در embedding های شبکه.

بخش دوم

۱- آموزش با استفاده از کلاس‌های ۱و ۳و ۵

برای این بخش در ابتدا یک تابع ایجاد شده است که داده‌های مربوط به کلاس های ۱و ۳و ۵ را از داده‌های کلاس‌های ۲و ۴ جدا می‌کند. در ادامه همانند بخش اول داده‌های کلاس های ۱و ۳و ۵ را به عنوان داده های train پردازش می‌کنیم و شبکه را با استفاده از آنها آموزش می‌دهیم. مشاهده می‌شود که این شبکه به دقت بسیار بالاتری از شبکه های بخش قبل دست پیدا می‌کند (دقت validation برابر با ۹۳ درصد) در نتیجه فرضی کم بودن داده‌ها قوی‌تر می‌شود زیرا بیشتر داده ها از کلاس ۵ هستند و حذف کردن دوکلاس باعث ساده‌تر شدن مسئله می‌شود و برای این مسئله ساده تر تقریبا سه چهارم داده ها را در اختیار داریم.

در ادامه مدل یادگیری شده با استفاده از کلاس‌های ۱و ۳و ۵ را با داده‌های مربوط به کلاس‌های ۲و ۴ که به عنوان داده‌های test ذخیره کردیم، تست می‌کنیم. مشاهده می‌شود که داده های کلاس ۲ به کلاس های ۱ و ۳ نگاشت می‌شوند و داده های مربوط به

کلاس ۴، غالبا به کلاس های ۳ و ۵ نگاشت می‌شوند که نتیجه جالبی است و همانطوری است که انتظار می‌رود زیرا کلاس های ۴ نظرات مثبتی هستند که ممکن است قسمت های منفی نیز داشته باشند که بر اساس میزان مثبت یا منفی بودنشان شبکه آنها را به کلاس ۵ یا ۳ نگاشت می‌کند. در مقابل کلاس ۲ نیز وضعیتی مشابه دارد، یعنی شامل نظرات غالبا منفی است که نکات مثبت کمی نیز دارد. در نتیجه اگر یک نظر خیلی منفی باشد شبکه آنرا به ۱ نگاشت می‌کند و اگر نکات مثبتی نیز در نظر وجود داشته باشد به کلاس ۳ نگاشت می‌شود.

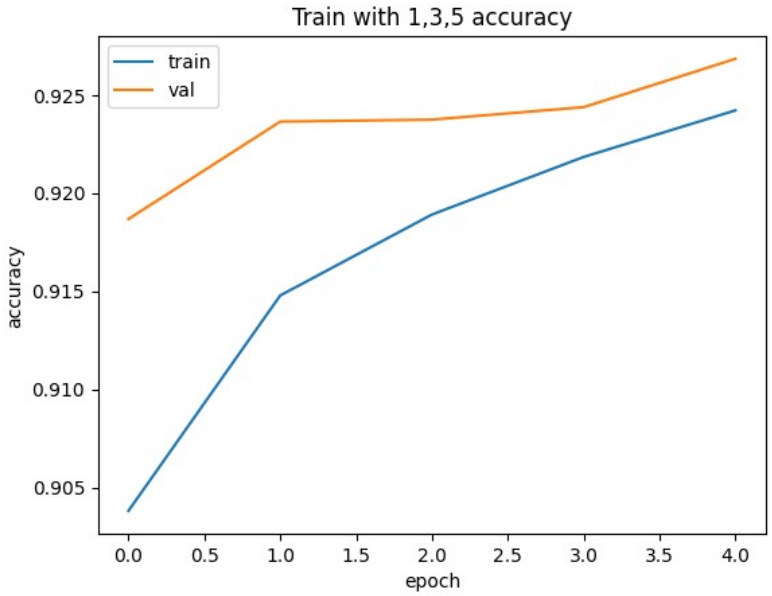

Figure ۷: نتیجه یادگیری شبکه با استفاده از داده‌های مربوط به کلاس های ۱و ۳و ۵.

Table ۱: مقایسه لیبل پیش‌بینی شده و لیبل واقعی در شبکه آموزش دیده شده با استفاده از سه کلاس.

متن پیش‌پردازش شده	لیبل واقعی	لیبل پیش‌بینی شده
['the', 'stain', 'glass', 'page', 'pretti', 'cool', 'and', 'nice', 'black', 'outlin', 'super', 'dark', 'thick', 'and', 'dragon', 'fight', 'wizard']	۴	۵
['i', 'love']	۲	۳
this', 'game', 'amaz', 'if', 'play', 'like', 'balderdash', 'tri', 'either', 'way','] 'game', 'adapt', 'well', 'differ', 'group', 'peopl', 'differ', 'tast', 'strength', 'almost', 'nobodi', 'heard', 'introduc', 'i', 'get', 'frequent', 'request', 'bring', 'parti', 'board', 'game', 'night', 'access', 'often', 'hilari', 'if', ["stock', 'grab', 'game', 'hard', 'find	۲	۳
i', 'rememb', 'mousetrap', 'labori', 'i', 'kid', 'i', 'sure', 'game', '20','] 'year', 'ago', 'parent', 'warn', 'sit', 'hour', 'complic', 'board', 'setup', 'requir', 'read', 'skill', 'my', 'son', 'love', 'i', 'admit', 'i', 'tri', 'avoid', 'unless', 'i', 'absolut', 'finish', 'daili', 'duti', 'afford', 'commit', 'time', ["take', 'complet', 'game	۲	۱

۲- تست داده های IMDB

در این بخش، شبکه یادگیری شده توسط کلاس های ۱و ۳و ۵ را با استفاده از داده‌های IMDB تست می‌کنیم. برای این کار ابتدا داداگان IMDB را در colab بارگذاری کرده و پیش پردازش را روی آنها انجام می‌دهیم. این داده ها شامل ۵۰۰۰۰ داده در ۲ کلاس

مثبت و منفی می‌شوند. مشاهده می‌شود که داده های مربوط به positive در IMDB اکثرا خروجی ۵ می‌دهند و داده‌های مربوط به negative بیشتر خروجی ۱ می‌دهند.

Table ۲: مقایسه لیبل پیش‌بینی شده برای داده ها IMDB و لیبل واقعی در شبکه آموزش دیده شده با استفاده از سه کلاس.

متن پیش‌پردازش شده	لیبل واقعی	لیبل پیش‌بینی شده
['bad', 'plot', 'bad', 'dialogue', 'bad', 'acting', 'idiotic', 'directing', 'annoying', 'porn', 'groove', 'soundtrack', 'ran', 'continually', 'overacted', 'script', 'crappy', 'copy', 'vhs', 'cannot', 'redeemed', 'consuming', 'liquor', 'trust', 'i', 'stuck', 'turkey', 'end', 'it', 'pathetically', 'bad', 'i', 'figure', 'fourth', 'rate', 'spoof', 'springtime', 'hitler', 'the', 'girl', 'played', 'janis', 'joplin', 'faint', 'spark', 'interest', 'could', 'sing', 'better', 'original', 'if', 'want', 'watch', 'something', 'similar', 'thousand', 'times', 'better', 'watch', 'beyond', 'the', 'valley', 'the', 'dolls']	negative	۱
['firstly', 'title', 'relevance', 'whatsoever', 'movie', 'it', 'started', 'fine', 'good', 'development', 'got', 'annoying', 'tell', 'girlfriend', 'happened', 'even', 'attempt', 'tell', 'police', 'failed', 'added', 'annoyance', 'value', 'there', 'many', 'pregnant', 'pauses', 'movie', 'seemed', 'like', 'filler', 'anything', 'worthwhile', 'the', 'plot', 'never', 'revealed', 'crime', 'although', 'good', 'plot', 'would', 'allowed', 'disclosure', 'the', 'ending', 'nothing', 'short', 'hey', 'run', 'budget', 'let', 'stop', 'now', 'if', 'i', 'written', 'novel', 'ended', 'way', 'i', 'top', 'trash', 'trash', 'trash']	negative	۳
['one', 'reviewers', 'mentioned', 'watching', 'oz', 'episode', 'hooked', 'they', 'right', 'exactly', 'happened', 'the', 'first', 'thing', 'struck', 'oz', 'brutality', 'unflinching', 'scenes', 'violence', 'set', 'right', 'word', 'go', 'trust', 'show', 'faint', 'hearted', 'timid', 'this', 'show', 'pulls', 'punches', 'regards', 'drugs', 'sex', 'violence', 'its', 'hardcore', 'classic', 'use', 'word', 'it', 'called', 'oz', 'nickname', 'given', 'oswald', 'maximum', 'security', 'state', 'penitentary', 'it', 'focuses', 'mainly', 'emerald', 'city', 'experimental', 'section', 'prison', 'cells', 'glass', 'fronts', 'face', 'inwards', 'privacy', 'high', 'agenda', 'em', 'city', 'home', 'many', 'aryans', 'muslims', 'gangstas', 'latinos', 'christians', 'italians', 'irish', 'scuffles', 'death', 'stares', 'dodgy', 'dealings', 'shady', 'agreements', 'never', 'far', 'away', 'i', 'would', 'say', 'main', 'appeal', 'show', 'due', 'fact', 'goes', 'shows', 'dare', 'forget', 'pretty', 'pictures', 'painted', 'mainstream', 'audiences', 'forget', 'charm', 'forget', 'romance', 'oz', 'mess', 'around', 'the', 'first', 'episode', 'i', 'ever', 'saw', 'struck', 'nasty', 'surreal', 'i', 'say', 'i', 'ready', 'i', 'watched', 'i', 'developed', 'taste', 'oz', 'got', 'accustomed', 'high', 'levels', 'graphic', 'violence', 'not', 'violence', 'injustice', 'crooked', 'guards', 'sold', 'nickel', 'inmates', 'kill', 'order', 'get', 'away', 'well', 'mannered', 'middle', 'class', 'inmates', 'turned', 'prison', 'bitches', 'due', 'lack', 'street', 'skills', 'prison', 'experience', 'watching', 'oz', 'may', 'become', 'comfortable', 'uncomfortable', 'viewing', 'thats', 'get', 'touch', 'darker', 'side']	positive	۵

مشاهده می‌شود که برای نظر دوم که در جدول ثبت شده است، با اینکه نظر negative بوده است، ولی شامل کلمات مثبتی نیز می‌باشد و برای همین شبکه کلاس ۳ را پیش‌بینی کرده است.