به نام خدا

تمرین شماره چهار درس پردازش زبانهای طبیعی توحید عابدینی ۹۹۷۲۲۱۵۳

سوال ۱)

بخش a)

تفاوت اصلی RNN و LSTM در این است که به دلیل وجود مشکل ناپدید شدن گرادیان در RNN اقدام به ارائه مدل LSTM شده است.

مشکل ناپدید شدن گرادیان: در RNN ها هنگامی که طول پنجره ما زیاد باشد و به تعداد گام های بیشتری از قبل و یا بعد نگاه کنیم به دلیل مشتق گیری های زنجیره ای متوالی در نهایت گرادیان بسیار کوچک میشود و مقدای که به لایه مورد نظر میرسد عملا نمیتواند بروز رسانی چشمگیری داشته باشد. به این دلیل که حافظه RNN ها کوتاه مدت است داده های قبلی اثر کمتری روی گام فعلی دارند و همین اثر کم یعنی مقدار عددی کمتر و با ضرب شدن اینها در هم در نهایت عدد گرادیان کوچک تر میشود. اما LSTM با داشتن گیتها حافظه بلند مدت هم دارد و یاد میگیرد که چه بخش هایی را بیشتر یاد بگیرد و اثر دهد و همین مفهوم یعنی اثر عددی بیشتر و افزایش میزان گرادیان و رفع مشکل ناپدید شدن (صفر شدن یا خیلی کوچک شدن) گرادیان.

منبع:

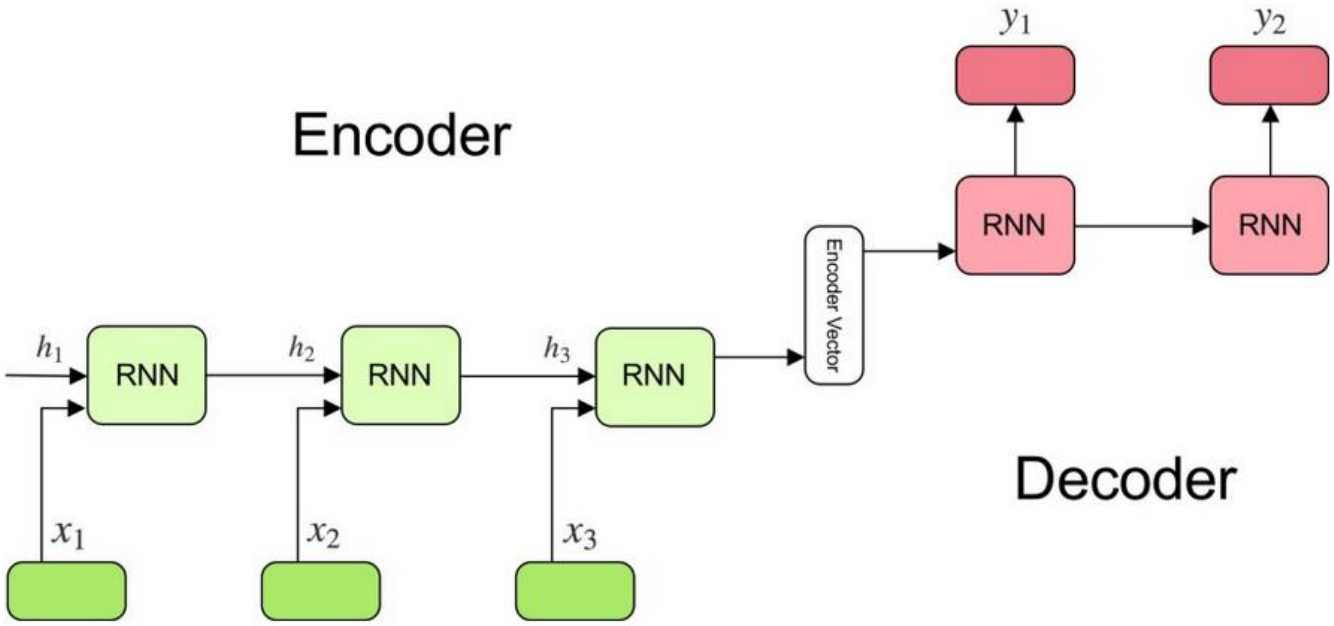
https://ashutoshtripathi.com/2021/07/02/what-is-the-main-difference-between-rnn-and-lstm-nlp-rnn-vs-lstm/
https://www.youtube.com/watch?v=AK9IN5uvaok

بخش b)

مدل های ترتیبی به دلیل ویژگی ای که دارند مناسب برای استفاده در تسک های متنی هستند.

به طور کلی در خارج از مدل داده را پس از اینکه به سایز مورد نظر (توکن کلمه ای یا کاراکتر – بسته به کاربرد) تبدیل کردیم یک دیکشنری از تمامی توکن ها ایجاد کرده و برای هر یک ایندکس تعریف میکنیم. سپس ایندکس ها را عوض توکن ها به مدل داده و پردازش را روی آن انجام میدهیم. میتوان از معماری های متفاوتی استفاده کرد که بهترین و پرکابرد ترین معماری، معماری encoder-decoder است. که با هر نوع سایز ورودی و خروجی کار میکند و در حالت خاص اگر سایز ورودی و خروجی یکسان باشید میتوان از شبکه بازگشتی ساده استفاده کرد. پس از اتمام محاسبات توسط مدل در نهایت خروجی را که ایندکس های ما هستند مجددا به صورت معکوس با استفاده از دیکشنری "توکن – ایندکس" به کلمات تبدیل میکنیم.

در شکل زیر یک نمونه مدل با معماری encoder-decoder مشاهده میکنیم.



Encoder-decoder sequence to sequence model

همانطور که از شکل هم مشخص است هر کدام از قسمت های encoder و decoder متشکل از چند نورون بازگشتی استک شده هستند. یک بخش میانی هم در مدل با نام بردار میانی یا intermediate vector است که خروجی حالت نهایی بخش encoder مدل است و شامل تمامی اطلاعات ورودی ما به صورت فشرده است و به عنوان حالت مخفی ابتدایی برای بخش decoder مدل استفاده میشود. نحوه اتصال ورودی ها و دریافت خروجی ها در تصویر بالا قابل مشاهده است.

منبع:

https://towardsdatascience.com/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346

سوال ۲)

پیاده سازی در نوت بوک Bert_finetune تحویل داده شده است.

بخش a)

امبدینگ های ثابت به دلیل اینکه به زمینه متن توجه نمیکنند نمیتوانند معنای سطح بالا را در نظر بگیرند و در بعضی شرایط هم ابهام دارند و برخی کلمات به خودی خود چندی معنا دارند که بدون توجه به زمینه امکان رفع ابهام از معنا وجود ندارد که با در نظر گرفتن زمینه میتوان معنای واضح تری از کلمه در متن داشت. همچنین سایز کل لغات محدود هست و در هر حال امکان دارد با کلماتی که جدید هستند روبرو شویم و به مشکل بخوریم اما در امبدینگ های پویا این مشکل به این شدت وجود ندارد و با در نظر گرفتن زمینه معنای حدودی بدست میاورد. همچنین در تسک های معنایی به دلیل اینکه بخش زیادی از اطلاعات را در نظر نمیگیریم عملکرد خوبی روی این تسک ها با امبدینگ های ثابت نخواهیم داشت. مورد بعدی این است که امبدینگ هر کلمه بسته به اینکه دیتاست مورد نظر چه پیش زمینه ای داشته باشد میتواند متغیر باشد و با عوض شدن دیتاست تغییرات زیادی در امبدینگ همان کلمه با دیتاست جدید داشته باشیم، که این هم یکی از ضعف های امبدینگ پویا است.

بخش b)

با توجه به عملکرد بسیار خوب مدل BERT و قرار داشتن در مرز دانش به لحاظ عملکردی، در تسک های متفاوتی از این مدل استفاده میشود. اما میدانیم این مدل بسیار بزرگ بوده و عمل آموزش آن از ابتدا بسیار زمان بر خواهد بود. پس با استفاده از انتقال یادگیری، مدل را آموزش داده و بعد در تسک های دیگر از آن استفاده میشود (تسک های down stream)

و میتوان با fine tune کردن مدل عملکرد مناسبی را با توجه به داده ها و مساله داشت.

دو objective در حالت از پیش آموزش داده شده برای این مدل به شرح زیر است:

مدل سازی زبانی mask شده(Masked Language Modeling): برای پیش بینی کلمه مخفی در یک جمله است.

مثال:

[CLS]Machine [MASK] is Super Cool[SEP]Machine Learning Is Super Cool[EOS]

در جمله بالا باید مدل پیش بینی کند که چه کلمه ای mask شده است.

پیش بینی جمله بعدی (Next Sequence Prediction): با دادن یک جمله مدل باید جمله بعدی را پیش بینی کند.

مثال:

[CLS]Sentence A [SEP] Sentence B[EOS]

جمله A را داریم و باید مدل جمله B را پیش بینی کند.

منبع:

https://medium.com/analytics-vidhya/an-overview-of-the-various-bert-pre-training-methods-c365512342d8