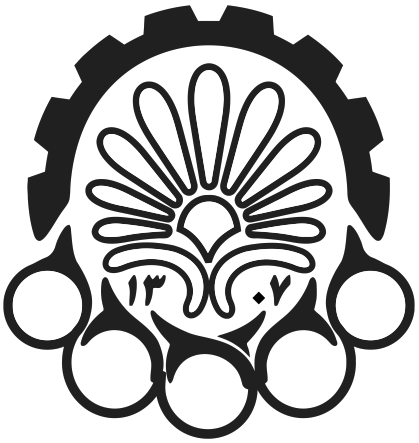
شبکه‌های عصبی و یادگیری عمیق
دکتر صفابخش



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین هفتم

شبکه Transformer

۲۱ تیر ۱۴۰۳

شبکه‌های عصبی و یادگیری عمیق

تمرین هفتم

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

دانشکده مهندسی کامپیوتر

سوال اول - تئوری

یکی از دلایل نیاز به مکانیزم توجه، گلوگاهی بود که بین رمزگذار و رمزگشا در مدل های seq2seq به وجود می‌آمد. این مشکل را توضیح دهید و نشان دهید چطور مکانیزم توجه این مشکل را حل کرد. یکی دیگر از مشکلات، عدم توجه مدل به گذشته دور بود. به طور مثال در یک متن به کلمات نزدیک‌تر اهمیت بیشتری داده می‌شد تا کلمات دورتر و وزن کلمات دورتر به صورت نمایی کاهش پیدا می‌کرد. آیا استفاده از lstm و یا lstm دوطرفه می‌تواند این مشکل را به طور کامل رفع کند؟ توضیح دهید.

پاسخ

در مدل‌های Seq2Seq، انکودر دنباله ورودی را پردازش کرده و آن را به یک بردار متنی با طول ثابت تبدیل می‌کند. این بردار متنی باید تمام اطلاعات مربوط به دنباله ورودی را در خود ذخیره کند. سپس دیکودر این بردار با طول ثابت را می‌گیرد و دنباله خروجی را تولید می‌کند. برای دنباله‌های ورودی طولانی، فشرده‌سازی تمام اطلاعات به یک بردار با طول ثابت دشوار است. این منجر به از دست رفتن اطلاعات می‌شود، بردار متنی با طول ثابت ممکن است نتواند تمام جزئیات لازم برای تولید دنباله خروجی منسجم و دقیق را ذخیره کند همین موضوع به عنوان یکی از چالش ها م مشکلات مدل‌های RNN مطرح می‌شود.

مکانیزم توجه برای کاهش مشکل گلوگاه در شبکه‌های RNN معرفی شد. مکانیزم توجه بر خلاف روش‌های قبلی، به جای اتکا به یک بردار متنی با طول ثابت، به دیکودر این اجازه را می‌دهد که برای هر خروجی یک بردار متنی پویا ایجاد کند که این بردار متنی پویا یک جمع وزنی از تمام وضعیت‌های پنهان (از گذشته‌های دور تا الان)انکودر است.

مسئله دیگری در مدل‌های Seq2Seq، به‌ویژه با RNNها، دشواری در پردازش وابستگی‌های بلندمدت بود. RNNهای سنتی و حتی LSTMها تمایل دارند که به ورودی‌های جدید، بیشتر از ورودی‌های دورتر اهمیت دهند.

LSTM ها برای کاهش مشکل محو شدن گرادیان طراحی شده‌اند که به ضبط وابستگی‌های طولانی‌تر نسبت به RNNهای معمولی کمک می‌کند. با این حال، تأثیر ورودی های دورتر همچنان تمایل دارد که با گذشت زمان کاهش یابد، هرچند نه به اندازه‌ای که در RNNهای استاندارد دیده می‌شود.

در BiLSTMها دنباله را در هر دو جهت جلو و عقب پردازش می‌کنند و بنابراین اطلاعات را از هر دو زمینه گذشته و آینده فراهم می‌کنند. این رویکرد دوطرفه توانایی مدل را در ضبط وابستگی‌ها در هر دو جهت بهبود می‌بخشد. با این وجود، BiLSTMها همچنان به بردارهای با طول ثابت متکی هستند و با وابستگی‌های بسیار طولانی مشکل دارند.

در عمل این موضوع به عنوان یکی از ضعف‌های این نوع شبکه‌ها محسوب می‌شود و شبکه Transformer و به‌ویژه مکانیزم توجه این مشکل را حل نموده و وابستگی‌های طولانی مدت را در دنباله سیگنال ورودی، بیشتر از سایر شبکه‌ها درک می‌کند.

مکانیزم توجه به دیکودر اجازه می‌دهد تا به هر قسمت از دنباله ورودی به‌طور مستقیم دسترسی داشته باشد، بدون توجه به موقعیت آن. این دسترسی مستقیم به این معنی است که ورودی دور نیز می‌توانند بر ورودی فعلی تاثیرگذار باشد.

سوال دوم - تئوری

در شبکه‌های بازگشتی ما می‌توانستیم از خروجی مرحله قبل در ورودی، تاریخچه و گذشته را مدل کنیم. اما باتوجه به اینکه مدل‌های ترنسفورمر از شبکه‌های بازگشتی استفاده نمی‌کنند، چطور می‌توانند بهتر از شبکه‌های بازگشتی گذشته را درنظر بگیرند (نشان دهید). مشکلات ترنسفورمر را در مقایسه با شبکه‌های بازگشتی بیان کنید.

پاسخ

مدل‌های ارنسفرمر برخلاف شبکه‌های (RNN)، تمام توکن‌ها را به صورت موازی پردازش می‌کنند که منجر به افزایش قابل توجهی در کارایی محاسباتی، به ویژه در سخت‌افزارهای مدرن مانند GPUها می‌شود.
این پردازش موازی به ارنسفرمرها اجازه می‌دهد تا دنباله‌های طولانی را به طور موثرتری مدیریت کنند، زیرا از مشکل گلوگاه پردازش مرحله به مرحله در هاRNN اجتناب می‌شود.
به طور دقیق این مکانیزم توجه است که این امکان را به ما می‌دهد تا برخلاف شبکه‌های RNN بدون هیچ کامپوننت بازگشتی و فیدبک‌ای اطلاعات و وابستگی‌های گذشته را درک کنیم و آن را مدل کنیم.
اما در برابر همه این مزایا، ترنسفرمرها معایبی نیز دارند که در ادامه به معرفی چند مورد از آنها می‌پردازییم:

۱. **حجم محاسبات و حافظه زیاد:**
مکانیزم توجه خود دارای پیچیدگی زمانی و حافظه‌ای زیاد نسبت به طول دنباله است. این امر باعث می‌شود ترنسفرمرها برای دنباله‌های بسیار طولانی منابع زیادی مصرف کنند. در مقابل، RNN ها دارای پیچیدگی محاسباتی کمتر نسبت به ترنسفرمرها هستند که می‌تواند برای برخی کاربردها کارآمدتر باشد.

۲. **مدیریت دنباله‌های بسیار طولانی:**
با اینکه ترنسفرمرها در درک وابستگی‌های طولانی بهتر هستند، عملکرد آنها با دنباله‌های بسیار طولانی می‌تواند به دلیل پیچیدگی بیش از حد به صورت توان دوم کاهش یابد.

۳. **نیاز به داده‌های زیاد برای آموزش:**
ترانسفورمرها به طور کلی به مقادیر زیادی از داده‌های آموزشی و منابع محاسباتی قابل توجهی برای آموزش موثر نیاز دارند. در مقابل، RNNها می‌توانند از نظر داده‌ای کارآمدتر و ارزان‌تر برای آموزش باشند، و با مجموعه داده کوچکتری در مقایسه با ترنسفرمرها آموزش ببینند.

سوال سوم - تئوری

ترنسفورمرها نسبت به شبکه‌های sqe2seq قابلیت موازی‌سازی بیشتری دارند. با ذکر جزئیات توضیح دهید.

پاسخ

مدل‌های Seq2Seq معمولاً از RNN یا LSTM استفاده می‌کنند که پردازش ترتیبی دارند. اما در مقابل ترنسفرمرها به صورت موازی داده‌ها را پردازش می‌کنند.

۱. **در مدل‌های seq2seq**
اگر فرض شود دنباله ای با طول n داریم، هر گام زمانی در RNN یا LSTM باید به ترتیب پردازش شود و برای هر گام زمانی، یک پردازش زمانی $O(1)$ انجام می‌شود. بنابراین، کل پیچیدگی زمانی برای پردازش توالی ورودی $O(n)$ است.

هر گام زمانی باید منتظر تکمیل گام قبلی باشد، بنابراین پردازش‌ها نمی‌توانند به‌طور موازی انجام شوند. در بهترین حالت، هر گام زمانی می‌تواند به‌طور موازی با پردازش داخلی خود (مثل محاسبات داخل سلول‌های LSTM) انجام شود، اما این قابلیت موازی‌سازی به‌طور کلی محدود است.

در مقابل در

۲. **ترنسفرمرها**
اگر مجددا فرض شود توالی ای ورودی با طول n داریم، در لایه Self Attention، هر نشانه می‌تواند به تمام نشانه‌های دیگر در توالی توجه کند. این کار با محاسبه ماتریس توجه انجام می‌شود که پیچیدگی زمانی $O(n^2)$ دارد. محاسبات ماتریسی برای محاسبه توجه به‌طور موازی قابل انجام است. بنابراین، پیچیدگی زمانی یک لایه خود توجهی $O(n^2)$است.

محاسبات ماتریسی در لایه Self Attention (مثل محاسبه ماتریس KQ^T برای توجه) به‌طور کامل به‌صورت موازی انجام می‌شوند.

سوال چهارم - تئوری

یکی از مشکلات ترنسفورمرها مرتبه هزینه محاسباتی و هزینه ذخیره‌سازی عملیات self-attention است که از مرتبه $O(N^2)$ می‌باشد. تلاش‌هایی برای کاهش این مشکل انجام شد. مقالاتی نشان دادند که عملکرد softmax باعث می‌شود تا نتوانیم این پیچیدگی را کاهش دهیم. توضیح دهید چرا عملکرد softmax باعث وجود این مسئله می‌شود. همچنین یکی از پیشنهادها برای حل این مشکل استفاده از مکانیزم‌های توجه کرنلی است. در مورد این مکانیزم تحقیق کنید و نشان دهید چطور این روش منجر به کاهش پیچیدگی می‌شود. یک کرنل به دلخواه انتخاب کنید و عبارت «۱» را بازنویسی کنید و مرتبه زمانی و حافظه مورد نیاز برای عملکرد self-attention را محاسبه کنید. لطفا به مقاله که برای انتخاب کرنل مراجعه کردید، ارجاع دهید.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (۱)$$

پاسخ

بر اساس رابطه ۱ مکانیزم Self-attention شامل محاسبه ضرب داخلی Qو K می‌شود که منجر به تولید ماتریسی با ابعاد $N \times N$ می‌شود. تابع softmax به هر سطر این ماتریس اعمال می‌شود که مقادیر توجه را نرمالیزه می‌کند. پیچیدگی درجه دوم از اینجا ناشی می‌شود که:

۱. **ضرب ماتریسی**: محاسبه QK^T شامل عملیات $O(N^2 d_k)$ است.

۲. **محاسبه softmax**: اگرچه softmax برای هر سطر $O(N)$ است، به همه N سطر اعمال می‌شود که منجر به $O(N^2)$ در کل می‌شود.

برای حل مشکل پیچیدگی درجه دوم، مکانیزم‌های توجه مبتنی بر هسته پیشنهاد شده‌اند. این روش‌ها با استفاده از توابع هسته، تابع softmax را تقریب می‌زنند که می‌تواند پیچیدگی محاسبات توجه را کاهش دهد.
یکی از این روش‌ها استفاده از نقشه ویژگی تصادفی برای تقریب تابع هسته softmax است. ایده اصلی این است که ورودی را با استفاده از نقشه ویژگی به فضایی تبدیل کنیم که در آن ضرب داخلی، تقریب تابع هسته اصلی را ارائه دهد.
حال سوال پیش می‌آید که از چه هسته هایی می‌توان استفاده نمود؟

۱. **هسته RBF**
هسته RBF یکی از انتخاب‌های محبوب برای چنین تقریب‌هایی است. هسته RBF به صورت زیر تعریف می‌شود:

$$k(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$$

اما برای کارایی محاسباتی، می‌توانیم از تقریب های سری فوریه تصادفی استفاده کنیم.

۲. **تقریب softmax با هسته RBF**
هسته RBF می‌تواند با استفاده از ویژگی‌های سری فوریه تصادفی به صورت زیر تقریب زده شود:

$$k(x, y) \approx \phi(x)^T \phi(y)$$

که در آن $\phi(x)$ یک نقشه ویژگی تصادفی از x است.

پاسخ

۱. **بازنویسی توجه مبتنی بر هسته**
با استفاده از این تقریب، مکانیزم توجه می‌تواند به صورت زیر بازنویسی شود:

$$Attention(Q, K, V) \approx \left(\phi(Q)\phi(K)^T\right) V$$

۲. **کاهش پیچیدگی**

- **تبدیلات:** نقشه ویژگی تصادفی ϕ به طور معمول ابعاد کمتری r دارد. تبدیل Q و K به $\phi(Q)$ و $\phi(K)$ به ترتیب شامل عملیات $O(Nd_kr)$ است.
- **ضرب داخلی:** ضرب داخلی $\phi(Q)\phi(K)^T$ شامل عملیات $O(N^2r)$ است.
- **ضرب نهایی با V:** ضرب نتیجه با V شامل عملیات $O(Nrd_v)$ است.

با انتخاب $r \ll N$، پیچیدگی به طور قابل توجهی کاهش می‌یابد.

۳. **پیچیدگی زمانی و حافظه**

- **پیچیدگی زمانی:**
  - نقشه ویژگی: $O(Nd_kr)$
  - ضرب داخلی: $O(N^2r)$
  - محصول نهایی: $O(Nrd_v)$

  با ترکیب اینها، پیچیدگی زمانی کلی:

$$O(Nd_kr + N^2r + Nrd_v)$$

  با $r \ll N$، عبارت غالب $O(N^2r)$ است.
- **پیچیدگی حافظه:**
  - ذخیره $\phi(Q)$ و $\phi(K)$: $O(Nr)$

*

References

[1] "Rethinking Attention with Performers" by Choromanski et al. (2021), which introduces the Performer model using kernel-based approximations to reduce the complexity of self-attention.

سوال پنجم - عملی

در دوران ابتدایی برای اینکه درک بهتری از جملات و جایگاه کلمات در جمله داشته باشیم تمرینی تحت عنوان "با کلمات زیر جمله بسازید" داشتیم. دراین سوال می‌خواهیم یک مدل ترنسفورمر را از ابتدا برای این وظیفه آموزش دهیم. به این منظور مراحل زیر را دنبال کنید.

۱. مجموعه‌داده‌ای فارسی به انتخاب خودتان از اینترنت دانلود کنید.

۲. جملات هر متن را جدا کنید. (ممکن است چالش‌هایی داشته باشید. ایده این قسمت را بطور کامل بیان کنید. در صورتی که بتوانید تا حد خوبی جملات هر متن را جدا کنید، نمره اضافه برای شما در نظر گرفته می‌شود.)

۳. مجموعه‌داده مربوط به این سوال را بسازید. ستون اول جمله‌ای که به صورت تصادفی کلماتش جابجا شدند و ستون دوم مرتب شده آن جمله است.

۴. مدل ترنسفورمر خود را پیاده‌سازی کنید و مدل را آموزش دهید. دقت کنید برای رسیدن به صحت مناسب به دیتا زیادی نیاز دارید و ممکن است منابع شما محدود باشد. در این‌جا با توجه به منابع خودتان این موضوع را مدیریت کنید. یک دقت حداقلی برای این سوال کافی است.

۵. مدل را با داده‌های آزمون ارزیابی کرده. ۵ نمونه از داده‌های آزمون را به صورت تصادفی انتخاب کرده، کلمات آن را جابجا کنید و به مدل بدهید. قبل و بعد این ۵ نمونه را در گزارش خود بیاورید.

۶. توضیح دهید در مرحله قبل با چه روشی مدل را ارزیابی کردید و دلایل خود را بیان کنید.

پاسخ

دیتاستی که در این سوال از آن استفاده کردیم، دیتاست مجوعه توییت‌های فارسی است که می‌توانید آن را از اینجا دانلود کنید.
این دیتاست شامل دسته‌های زیر است:

۱. `anger.csv`
۲. `disgust.csv`
۳. `fear.csv`
۴. `joy.csv`
۵. `sad.csv`
۶. `surprise.csv`

که ما از مجموعه داده `anger.csv` برای آموزش و تست شبکه استفاده کرده‌ایم.
ابعاد این مجموعه $(20069, 8)$ است. که نمونه‌هایی از آن را می‌توانید در شکل «۱» ببینید.

پاسخ

	tweet	replyCount	retweetCount	likeCount	quoteCount	hashtags	sourceLabel	emotion
0	...دیشب خواب دیدم بمبی چیزی زدن نورش خیلی خیره کن	0	3	2	0	['No2IR']	Twitter Web App	anger
1	...ر\nتبر زدی بر ریشه‌ام، جوانه رویید جای زخم	0	0	8	0	['سین_کاف']	Twitter for Android	anger
2	...پدر سوخته ای که بابام بهم میگه دو معنی داره که	1	0	11	0	['پدر_ایرانی']	Twitter for Android	anger
3	... با خود مواجه شوید و اخم نکنید. اقتدار در نگاه	0	0	1	0	['جذبه', 'اخم']	Twitter for iPhone	anger
4	...یا این همه ا\nمدح تو را در شادی و در غم نوشتند	0	4	6	36	['نبأ_عظیم']	Twitter Web App	anger
...	...	...	...	...	...	...	...	...
20064	... پرخاشگری پلیس #افغانستان؛ نیروی غیرحرفه‌ای یا	0	1	5	0	['افغانستان']	Twitter for iPhone	anger
20065	... فروید می‌گوید: «تمدن از آنجا آغاز شد که انسان	0	3	23	0	['کلمات']	Twitter for Android	anger
20066	...صبح امروز (۰۰/۰۱/۱۵) راننده خطی ایستگ (۱۱)ر\nشتو #	0	3	0	0	['رشتو']	Twitter for Android	anger
20067	...وقتی میگیم #پالانفارسیسم ملقمه بیشعوری فروما	0	2	0	3	['پالانفارسیسم', 'مجوس']	Twitter Web App	anger
20068	...یکی از مهم ترین مزیت های #کلاب_هاوس آشکارتر کر	1	3	50	0	['کلاب_هاوس', 'لاشخورهای_رسانه']	Twitter Web App	anger

20069 rows × 8 columns

شکل ۱: دیتاست anger.csv

ابتدا در فاز Preprocessing علائم های نگارشی را از جملات حذف کرده و بخشی خروجی به صورت زیر می‌شود:

```
Size of the dataset: (20069, 9)

0        ...دیشب خواب دیدم بمبی چیزی زدن نورش خیلی خیره کن
1        ...تبر زدی بر ریشهام جوانه رویید جای زخمراندی مرا
2        ...پدر سوخته ای که بابام بهم میگه دو معنی داره که
3        ...با خود مواجه شوید و اخم نکنید اقتدار در نگاه ا
4        ...مدح تو را در شادی و در غم نوشتندبا این همه اما
                              ...
20064    ... پرخاشگری پلیس  نیروی غیرحرفهای یا گرگهای تنها
20065    ... فروید میگوید تمدن از آنجا آغاز شد که انسان به
20066    ...صبح امروز۰۰۱۱۵ راننده خطی ایستگاه میدان آزا ۱
20067      ...وقتی میگیم  ملقمه  بیشعوری فرومایگی و بی شرفی
20068    ... یکی از مهم ترین مزیت های  آشکارتر کردن چهره ی
Name: cleaned_tweet, Length: 20069, dtype: object
```

شکل ۲: دیتاست پس از حذف علائم نگارشی

سپس با استفاده از کتابخانه `nltk` کلمات را از داخل جملات tokenize می‌کنیم. و خروجی به صورت زیر می‌شود:

```
Size of the dataset: (20069, 10)

0        [...دیشب, خواب, دیدم, بمبی, چیزی, زدن, نورش, خیلی
1        [...تبر, زدی, بر, ریشهام, جوانه, رویید, جای, زخمر
2        [...پدر, سوخته, ای, که, بابام, بهم, میگه, دو, معن
3        [... ,با, خود, مواجه, شوید, و, اخم, نکنید, اقتدار
4        [...مدح, تو, را, در, شادی, و, در, غم, نوشتندبا, ا
                              ...
20064    [...,پرخاشگری, پلیس, نیروی, غیرحرفهای, یا, گرگهای
20065    [... ,فروید, میگوید, تمدن, از, آنجا, آغاز, شد, که
20066    [...صبح, امروز۰۰۱۱۵, راننده, خطی, ایستگاه, می, ۱,
20067    [...وقتی, میگیم, ملقمه, بیشعوری, فرومایگی, و, بی
20068    [...,یکی, از, مهم, ترین, مزیت, های, آشکارتر, کردن
Name: tokens, Length: 20069, dtype: object
```

شکل ۳: جملات Tokenize شده

سپس کلمات Stop را حذف می‌کنیم.

پاسخ

```
Size of the dataset: (20069, 15)

0        [کو, کورمال, سبزه, نورش, زدن, بمبی, دیدم, خواب...
1        [دل, زخمراندی, رویید, جوانه, ریشهام, زدی, تبر,...
2        [مع, شرایط, معنی, میگه, بهم, بابام, سوخته, پدر...
3        [واقعیت, بگذار, چشمانت, ات, اقتدار, اخم, مواجه...
4        [ت, لبت, خنده, نوشتند, نوشتندبا, غم, شادی, مدح...
                              ...
20064    [ممن, گرگهای, غیرحرفهای, نیروی, پلیس, پرخاشگری...
20065    [کلمه, سنگ, انسان, آغاز, تمدن, میگوید, فروید, ...
20066    [میدان, ایستگاه, خطی, راننده, ۰۰۰۱۱۵امروز, صبح...
20067    [پا, ست, شرفی, فرومایگی, بیشعوری, ملقمه, میگیم...
20068    [دیکتاتو, مشمئزکننده, چهره, آشکارتر, مزیت, مهم...
Name: tokens, Length: 20069, dtype: object
```

شکل ۴: حذف کلمات Stop

و در مرحله بعد کلمات موجود در هر جمله را به صورت رندوم Shuffle می‌کنیم:

```
0        [زدن, عین, سبزه, قاسم, اخم, دست, دیدم, خوابم, ...
1        [زد, ریشهام, جوانه, زخمراندی, لبخند, رویید, دل...
2        [پ_بشم۱, پدر, پاره, معنیشو, معنی, سوخته, شرایط...
3        [کپی, واقعیت, چشمانت, اخم, عکس, مواجه, بگذار, ...
4        [نو, لبت, تصنیف, خنده, مدح, اخم, نوشتندسید, غم...
                              ...
20064    [پرخاش, وقتتان, نیروی, غیرحرفهای, پلیس, ممنونم...
20065    [می, انسان, دردناک, متمدن, فروید, انسان, سنگها...
20066    [ماسک, محض, شکایت, ایستگاه, درخواست, آزادیسه, ...
20067    [میکنیاینهمه, زیرخواب, پارس, گیو, کارتل, کونت,...
20068    [نقاب, دنیایی, مهم, دیکتاتورمآبانه, پرخاشگری, ...
Name: shuffled_tokens, Length: 20069, dtype: object
```

شکل ۵: کلمات بهم ریخته در جمله

از میان پارامترهای این دیتاست، طول جمله و تعداد هشتک های هر توییت را به عنوان ویژگی های مدل استخراج کردیم و درنهایت ابعاد داده ما $(20069, 15)$ شد. و آن را در فایلی به‌نام `processed_anger.csv` ذخیره می‌کنیم تا از آن در مرحله بعد استفاده کنیم.

	tweet	replyCount	retweetCount	likeCount	quoteCount	hashtags	sourceLabel	emotion	cleaned_tweet	tokens	tweet_length	num_hashtags	sourceLabel_encoded	emotion_encoded	shuffled_tokens
0	دیشب خواب دیدم بمبی چیزی زدن نورش خیلی خیره کن...	0	3	2	0	['No2IR']	Twitter Web App	anger	دیشب خواب دیدم بمبی چیزی زدن نورش خیلی خیره کن...	[خواب, دیدم, بمبی, زدن, نورش, سبزه, کورمال, کو...	260	1	0	0	[خوابم, دیدم, دست, اخم, قاسم, سبزه, عین, زدن, ...
1	...ران\nتبر زدی بر ریشه‌ام، جوانه رویید جای زخم	0	0	8	0	['سین_کاف']	Twitter for Android	anger	تبر زدی بر ریشهام جوانه رویید جای زخمراندی مرا...	[تبر, زدی, ریشهام, جوانه, رویید, زخمراندی, دل...	76	1	1	0	[دل, رویید, لبخند, زخمراندی, جوانه, ریشهام, زد...
2	...پدر سوخته ای که بابام بهم میگه دو معنی داره که	1	0	11	0	['پدر_ایرانی']	Twitter for Android	anger	...پدر سوخته ای که بابام بهم میگه دو معنی داره که	[پدر, سوخته, بابام, بهم, میگه, معنی, شرایط, مع...	232	1	1	0	[شرایط, سوخته, معنی, معنیشو, پاره, پدر, بشم۱_پ...
3	... با خود مواجه شوید و اخم نکنید. اقتدار در نگاه	0	0	1	0	['جذبه', 'اخم']	Twitter for iPhone	anger	...با خود مواجه شوید و اخم نکنید اقتدار در نگاه ا	[مواجه, اخم, اقتدار, ات, چشمانت, بگذار, واقعیت...	115	2	2	0	[بگذار, مواجه, عکس, اخم, چشمانت, واقعیت, کپی...
4	...با این همه \nمدح تو را در شادی و در غم نوشتند	4	6	36	0	['تبا_عظیم']	Twitter Web App	anger	...مدح تو را در شادی و در غم نوشتندبا این همه اما	[مدح, شادی, غم, نوشتندبا, نوشتند, خنده, لبت, ت...	142	1	0	0	[غم, نوشتندسید, اخم, مدح, خنده, تصنیف, لبت, نو...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20064	پرخاشگری پلیس #افغانستان؛ نیروی غیرحرفه‌ای یا ...	0	1	5	0	['افغانستان']	Twitter for iPhone	anger	پرخاشگری پلیس نیروی غیرحرفهای یا گرگهای تنها...	[پرخاشگری, پلیس, نیروی, غیرحرفهای, گرگهای, ممن...	65	1	2	0	[ممنونم, پلیس, غیرحرفهای, نیروی, وقتتان, پرخاش...
20065	... فروید می‌گوید: «تمدن از آنجا آغاز شد که انسان	0	3	23	0	['کلمات']	Twitter for Android	anger	... فروید میگوید تمدن از آنجا آغاز شد که انسان به	[فروید, میگوید, تمدن, آغاز, انسان, سنگ, کلمه, ...	195	1	1	0	[سنگها, انسان, فروید, متمدن, دردناک, انسان, می...
20066	صبح امروز (۰۰/۰۱/۱۵) راننده خطی (\n۱رشتو# ...ایستگ	3	0	3	0	['رشتو']	Twitter for Android	anger	صبح امروز ۰۰۰۱۱۵ راننده خطی ایستگاه میدان ۱ ازا...	[صبح, امروز۰۰۰۱۱۵, راننده, خطی, ایستگاه, میدان...	249	1	1	0	[آزادیسه, درخواست, ایستگاه, شکایت, محض, ماسک, ...
20067	...وقتی میگیم #پالانفارسیسم ملقمه بیشعوری فروما	2	0	3	0	['پالانفارسیسم', 'مجوس']	Twitter Web App	anger	وقتی میگیم ملقمه بیشعوری فرومایگی و بی شرفی...	[میگیم, ملقمه, بیشعوری, فرومایگی, شرفی, ست, پا...	258	2	0	0	[کونت, کارتل, گیو, پارس, زیرخواب, میکنیاینهمه...
20068	یکی از مهم ترین مزیت های #کلاب_هاوس آشکارتر کر...	1	3	50	0	['کلاب_هاوس', 'لاشخورهای_رسانه']	Twitter Web App	anger	یکی از مهم ترین مزیت های آشکارتر کردن چهره ی ...	[مهم, مزیت, آشکارتر, چهره, مشمئزکننده, دیکتاتو...	185	2	0	0	[پرخاشگری, دیکتاتورمآبانه, مهم, دنیایی, نقاب, ...

20069 rows × 15 columns

شکل ۶: دیتاست پردازش شده

در مرحله بعد که مرحله آموزش باشد، داده‌های پیش‌پردازش شده را از فایل `processed_anger.csv` می‌خوانیم و توکن ها را به string تبدیل می‌کنیم:

پاسخ

شکل ۷: توکن های string شده

۸۰ درصد داده‌ها را به‌عنوان داده‌های آموزش و ۲۰ درصد باقی مانده را به عنوان داده‌های اعتبارسنجی درنظر می‌گیریم. مدلی که برای این شبکه درنظر گرفتیم به‌صورت زیر است:

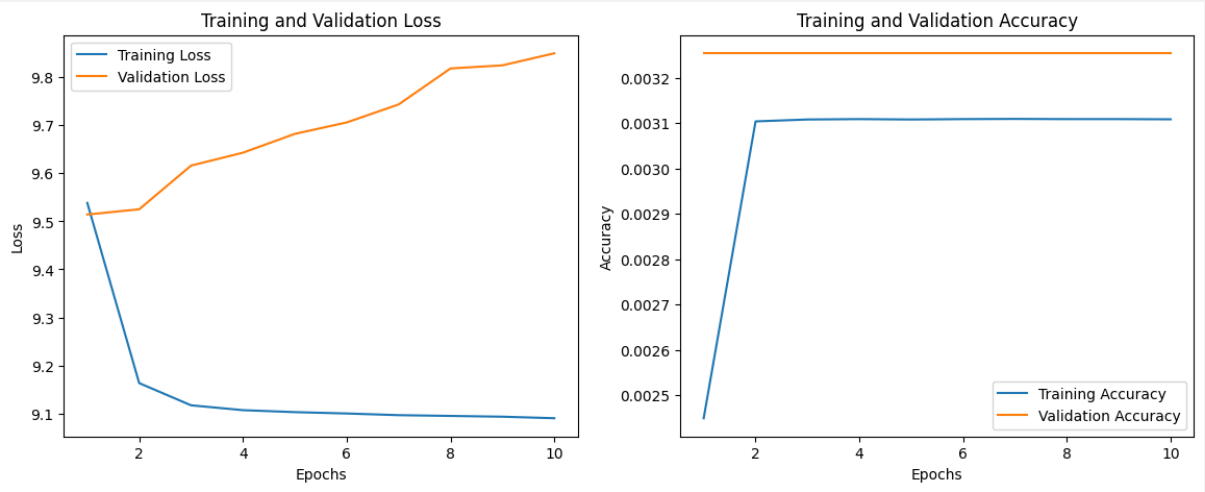
```
class TransformerModel(nn.Module):
def __init__(self, vocab_size, d_model, nhead, num_encoder_layers,
num_decoder_layers, dim_feedforward, max_seq_length):
super(TransformerModel, self).__init__()
self.embedding = nn.Embedding(vocab_size, d_model)
self.positional_encoding = nn.Parameter(torch.zeros(1, max_seq_length,
d_model))
self.transformer = nn.Transformer(d_model, nhead, num_encoder_layers,
num_decoder_layers, dim_feedforward)
self.fc_out = nn.Linear(d_model, vocab_size)
```

پارامترهای مدل نیز به‌صورت زیر تنظیم شده است:

1. `d_model = 512`
2. `nhead = 8`
3. `num_encoder_layers = 6`
4. `num_decoder_layers = 6`
5. `dim_feedforward = 2048`
6. `max_seq_length = maxlen`
7. `learning_rule = 0.001`
8. `batch_size = 32`

درنهایت شبکه را آموزش می‌دهیم و نمودارهای خطا و دقت برای داده‌های آموزش و اعتبارسنجی به‌صورت زیر به‌دست آمده است:

پاسخ



شکل ۸: نمودارهای دقت و خطا

علت اینکه شبکه نتوانسته است به خوبی آموزش ببیند و مقدار خطا بالاست، این است که با توجه به محدودیت های سخت‌افزاری، Colab در افزایش اندازه داده‌های ورودی، نمی‌توانیم ورودی‌های زیادی را در اندازه‌های LLMها به شبکه بدهیم. به همین دلیل مدل نمی‌تواند با این دیتای محدود به خوبی آموزش ببیند و خطای خود را مینیمم کند.
برای ارزیابی شبکه، یکی از ورودی های دیتاست بهم‌ریخته را به شبکه می‌دهیمِ و انتظار داریم که شبکه کلمات را مرتب کند.
برای مثال ورودی زیر را به شبکه می‌دهیم:

```
Input: سؤال خورد کتان صبا مختصر حکایت آیدکه دشنام بریزد غم چراغ جواب شکر استکه دانی سیرا باگر گوش مهتا بدعات داردو اي سهل گر تشنه بمیرد گل <pad> <unk>
Expected Output: سؤال خورد کتان صبا مختصر حکایت آیدکه دشنام بریزد غم چراغ جواب شکر استکه دانی سیرا باگر گوش مهتا بدعات داردو اي سهل گر تشنه بمیرد گل <pad> <unk>
```

شکل ۹: جمله ورودی و خروجی مطلوب

ذکر این نکته الزامیست که به دلیل حذف کاراکترهای اضافی و کلمات Stop ممکن است جمله اصلی (مرتب) نیز بی معنی به‌نظر برسد.
خروجی شبکه برای مرتب کردن دو کلمه از جمله به صورت زیر شده است:

```
Generated Output: خورد رای
```

شکل ۱۰: خروجی مرتب شده

سوال دوم - عملی

مجموعه‌داده CoLA (Corpus of Linguistic Acceptability) یک مجموعه داده مهم در زمینه پردازش زبان طبیعی (NLP) است که برای ارزیابی مقبولیت زبانی جملات استفاده می‌شود. مقبولیت زبانی به این معنی است که آیا یک جمله از نظر دستوری و نحوی توسط گویش‌وران بومی یک زبان درست است یا نه. در این سوال قصد داریم تا با تنظیم دقیق مدل BERT، یک طبقه‌بند دو کلاسه برای این مجموعه‌داده پیاده‌سازی کنیم. موارد زیر را دنبال کنید:

۱. دو فایل `in_domain_train.tsv` و `out_of_domain_dev.tsv` در اختیار شما قرارگرفته است. این فایل‌ها را در محیط برنامه‌نویسی خود بارگزاری کنید. پیش پردازش‌های لازم (مانند اضافه کردن کارکترهای خاص [SEP] و ...) به جملات، توکنایز کردن و ...

۲. ۱۰ درصد از داده‌های `"in_domain_train.tsv"` را به برای اعتبارسنجی در نظر بگیرید.

۳. مدل BERT را بارگذاری و پیکره‌بندی کنید. (پیشنهاد می‌شود از کتابخانه transformers) استفاده کنید.

۴. مدل را آموزش دهید. در هر epoch، خطا و صحت را برای داده‌های اعتبارسنجی چاپ کنید. همچنین بعد از اتمام آموزش نمودار خطا را به ازای هر دسته (batch) آموزش رسم کنید. (هر epoch می‌تواند شامل چندین دسته باشد).

۵. از داده‌های `out_of_domain_dev.tsv` برای ارزیابی مدل تنظیم‌ـدقیق شده خود استفاده کنید. برای این قسمت از معیار F1 و MCC1 استفاده کنید. این معیار را توضیح دهید و بگویید چرا استفاده از این معیار در اینجا نسبت به F1 بهتر است.

۶. معیار MCC شما برای داده‌های `out_of_domain_dev.tsv` نباید کوچکتر از ۵.۰ باشد.

پاسخ

ابتدا دیتاست را لود می‌کنیم. اندازه داده‌های آموزش $(8551, 4)$ است:

```
Shape of dataset: (8551, 4)
```

	sentence_source	label	label_notes	sentence
0	gj04	1	NaN	our friends wo n't buy this analysis , let alo...
1	gj04	1	NaN	one more pseudo generalization and i 'm giving...
2	gj04	1	NaN	one more pseudo generalization or i 'm giving ...
3	gj04	1	NaN	the more we study verbs , the crazier they get .
4	gj04	1	NaN	day by day the facts are getting murkier .
...	...	...	...	...
8546	ad03	0	*	poseidon appears to own a dragon
8547	ad03	0	*	digitize is my happiest memory
8548	ad03	1	NaN	it is easy to slay the gorgon .
8549	ad03	1	NaN	i had the strangest feeling that i knew you .
8550	ad03	1	NaN	what all did you get for christmas ?

8551 rows × 4 columns

شکل ۱۱: داده‌های آموزش دیتاست `in_domain_train.tsv`

پاسخ

همانند سوال قبل، پیش‌پردازش های لازم مانند Tokenize کردن و حذف کلمات Stop، و اضاف نمودن [CLS] و [SEP] به توکن ها را انجام می‌دهیم.
برای مثال، خروجی Tokenize شده یکی از ورودی ها به‌صورت زیر است:

```
Original:  our friends wo n't buy this analysis , let alone the next one we propose .
Token IDs: [101, 2256, 2814, 24185, 1050, 1005, 1056, 4965, 2023, 4106, 1010, 2292, 2894, 1996, 2279, 2028, 2057, 16599, 1012, 102]
```

شکل ۱۲: خروجی Tokenize شده

برای آموزش شبکه از مدل از پیش آموزش دیده Bert استفاده می‌کنیم. مدل به صورت زیر تعرریف شده است:

```
model = BertForSequenceClassification.from_pretrained(
"bert-base-uncased",
num_labels = 2,
output_attentions = False,
output_hidden_states = False, )
```

```
BertForSequenceClassification(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSdpaSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
...
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (classifier): Linear(in_features=768, out_features=2, bias=True)
)
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

شکل ۱۳: معماری مدل

پاسخ

همچنین پارامترهای شبکه به‌صورت زیر است:

```
The BERT model has 201 different named parameters.

==== Embedding Layer ====

bert.embeddings.word_embeddings.weight                  (30522, 768)
bert.embeddings.position_embeddings.weight                (512, 768)
bert.embeddings.token_type_embeddings.weight                (2, 768)
bert.embeddings.LayerNorm.weight                              (768,)
bert.embeddings.LayerNorm.bias                                (768,)

==== First Transformer ====

bert.encoder.layer.0.attention.self.query.weight          (768, 768)
bert.encoder.layer.0.attention.self.query.bias                (768,)
bert.encoder.layer.0.attention.self.key.weight            (768, 768)
bert.encoder.layer.0.attention.self.key.bias                  (768,)
bert.encoder.layer.0.attention.self.value.weight          (768, 768)
bert.encoder.layer.0.attention.self.value.bias                (768,)
bert.encoder.layer.0.attention.output.dense.weight        (768, 768)
bert.encoder.layer.0.attention.output.dense.bias              (768,)
bert.encoder.layer.0.attention.output.LayerNorm.weight        (768,)
bert.encoder.layer.0.attention.output.LayerNorm.bias          (768,)
bert.encoder.layer.0.intermediate.dense.weight           (3072, 768)
bert.encoder.layer.0.intermediate.dense.bias                (3072,)
bert.encoder.layer.0.output.dense.weight                 (768, 3072)
...
bert.pooler.dense.weight                                  (768, 768)
bert.pooler.dense.bias                                        (768,)
classifier.weight                                           (2, 768)
classifier.bias                                                 (2,)
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

شکل ۱۴: پارامترهای شبکه

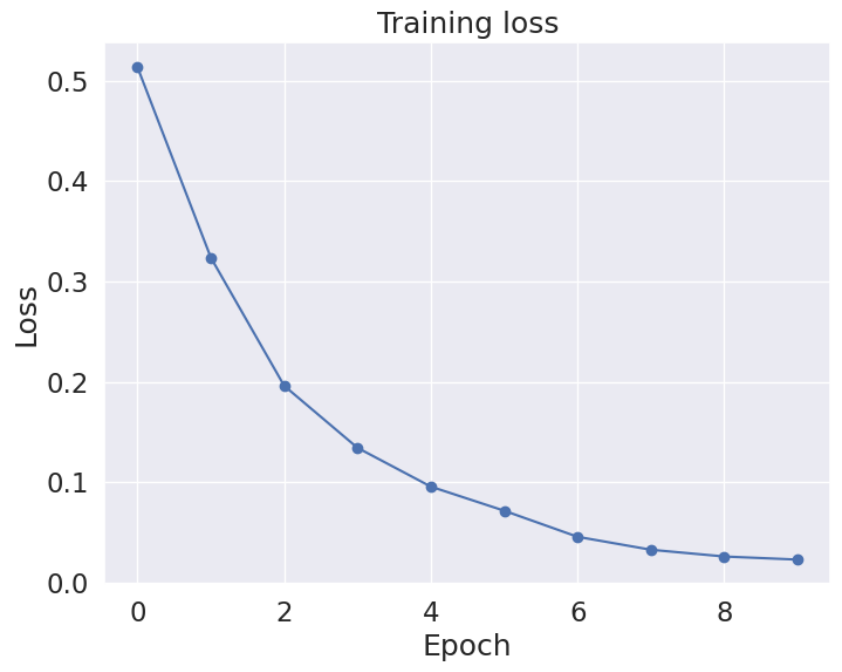
و درنهایت شبکه را Fine tune می‌کنیم. از آنجایی که مدل از پیش آموزش دیده است، نیازی به آموزش طولانی نیست. در حد ۵ الی ۱۰ دوره برای آموزش مناسب می‌باشد. در اینجا ما تعداد دوره های آموزشی را ۱۰ انتخاب کردیم.
در نهایت پس از اتمام آموزش، مقدار خطا و دقت به‌صورت زیر به‌دست می‌آید:

```
======== Epoch 10 / 10 ========
Training...
  Batch    40  of    241.    Elapsed: 0:00:13.
  Batch    80  of    241.    Elapsed: 0:00:26.
  Batch   120  of    241.    Elapsed: 0:00:39.
  Batch   160  of    241.    Elapsed: 0:00:52.
  Batch   200  of    241.    Elapsed: 0:01:05.
  Batch   240  of    241.    Elapsed: 0:01:18.

  Average training loss: 0.02
  Training epcoh took: 0:01:18

Running Validation...
  Accuracy: 0.83
  Validation took: 0:00:03

Training complete!
```

شکل ۱۵: مقدار دقت و خطا در پایان آموزش

پاسخ

نمودار خطای آموزش برحسب تعداد Epoch نیز به‌صورت زیر بدست می‌آید:



شکل ۱۶: نمودار خطای آموزش

در نهایت در فاز اعتبار سنجی را با مجموعه داده `out_of_domain_dev.tsv` انجام می‌دهیم و معیار MCC برای این مدل را 0.415 به‌دست آورده‌ایم. که همانطور که در صورت سوال گفته شده است، زیر 0.5 نیست.
اغلب در کاربردهای NLP معیار MCC به F1 ترجیح داده می‌شود چرا که معیار MCC همه چهار دسته ماتریس سردرگمی (TP, TN, FP, FN) را در نظر می‌گیرد و ارزیابی جامع‌تری از کیفیت طبقه‌بندی ارائه می‌دهد. امتیاز F1 بر تعادل بین precision و Recall تمرکز می‌کند که می‌تواند مفید باشد اما TN را در نظر نمی‌گیرد. در مواردی که TN ها مهم هستند، MCC ارزیابی متعادل‌تری ارائه می‌دهد.