

# شبکه‌های عصبی و یادگیری عمیق

## دکتر صفابخش



**دانشگاه صنعتی امیرکبیر**  
( پلی تکنیک تهران )  
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین هفتم  
شبکه Transformer

۲۰ تیر ۱۴۰۳



## 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 چالش‌ها 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.

LSTM‌ها برای کاهش مشکل محو شدن گرادین طراحی شده‌اند که به ضبط وابستگی‌های طولانی‌تر نسبت به RNN‌های معمولی کمک می‌کند. با این حال، تأثیر ورودی‌های 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 می‌تواند بر ورودی فعلی تأثیرگذار باشد.

## سوال دوم - تئوری

در شبکه‌های بازگشتی ما می‌توانستیم از خروجی مرحله قبل در ورودی، تاریخچه و گذشته را مدل کنیم. اما با توجه به اینکه مدل‌های ترنسفورمر از شبکه‌های بازگشتی استفاده نمی‌کنند، چطور می‌توانند بهتر از شبکه‌های بازگشتی گذشته را در نظر 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 GPU ها 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 RNN 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 می‌پردازیم:

## 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 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، RNN ها 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 seq2seq 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 ترنسفورمرها 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. ترنسفورمرها

اگر مجدداً 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 بتوانیم 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. لطفاً به مقاله که برای انتخاب کرنل مراجعه کردید، ارجاع دهید.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

پاسخ

بر اساس رابطه ۱ مکانیزم Self-attention شامل محاسبه ضرب داخلی  $Q$  و  $K$  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 هسته‌هایی 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 تقریب‌های 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:

$$\text{Attention}(Q, K, V) \approx (\phi(Q)\phi(K)^T) V$$

۲. کاهش پیچیدگی

- تبدیلات: نقشه ویژگی تصادفی  $\phi$  به طور معمول ابعاد کمتری  $r$  دارد. تبدیل  $Q$  و  $K$  به  $\phi(Q)$  و  $\phi(K)$  به ترتیب شامل عملیات  $O(Nd_k r)$  است.
  - ضرب داخلی: ضرب داخلی  $\phi(Q)\phi(K)^T$  شامل عملیات  $O(N^2 r)$  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:  $O(Nd_k r)$
  - ضرب داخلی:  $O(N^2 r)$
  - محصول نهایی:  $O(Nrd_v)$
- با ترکیب اینها، پیچیدگی زمانی کلی:

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

با  $r \ll N$ ، عبارت غالب  $O(N^2 r)$  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. در این سوال می‌خواهیم یک مدل ترنسفورمر را از ابتدا برای این وظیفه آموزش دهیم. به این منظور مراحل زیر را دنبال کنید.

۱. مجموعه داده‌ای 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. مجموعه داده 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 این جا 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 مجموعه 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 (8, 20069) 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	...نران! اثر زدی بر ریشهام، جوانه رویید جای زخم	0	0	8	0	['سین_کف']	Twitter for Android	anger
2	...پدر سوخته ای که با بام بهم میگه دو معنی داره که	1	0	11	0	['پدر_ایرانی']	Twitter for Android	anger
3	...با خود مواجه شوید و احم نکنید. اقتدار در نگاه	0	0	1	0	['جدیه', 'احم']	Twitter for iPhone	anger
4	...با این همه [مدح] تو را در شادی و در غم نوشتند	4	6	36	0	['نیا_عظیم']	Twitter Web App	anger
...	...	...	...	...	...	...	...	...
20064	...پرخاشگری پلیس نیروی غیرحرفه‌ای یا	0	1	5	0	['افغانستان']	Twitter for iPhone	anger
20065	...فروید میگوید: «تشنه از آنجا آغاز شد که انسان	0	3	23	0	['کلمات']	Twitter for Android	anger
20066	...صبح امروز (۱۴۰۳/۰۱/۰۵) راننده خطی ایستگاه (۱) رشتو #	3	0	3	0	['رشتو']	Twitter for Android	anger
20067	...وقتی میگیم ملقمه بیشعوری فرومایگی و بی شرفی	2	0	3	0	['پالافارسیم', 'مجوی']	Twitter Web App	anger
20068	...یکی از مهم ترین مزیت های آشکارتر هواس آشکارتر کر	1	3	50	0	['کلاب_هواس', 'لاشخورهای_رسانه']	Twitter Web App	anger

20069 rows x 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 زخمرا ندی مرا
2      ...پدر سوخته ای که با بام بهم میگه دو معنی داره که
3      ...با خود مواجه شوید و احم 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 امروز ۱۴۰۳/۰۱/۰۵ 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, زخمرا,
2      ...پدر, سوخته, ای, که, با, بام, بهم, میگه, دو, معن[
3      ...با, خود, مواجه, شوید, و, احم, 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, امروز, ۱۴۰۳, ۰۱, ۰۵, 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.



Size of the dataset: (20069, 15)

```
0 ... خواب, دیدم, بمبی, زدن, نورش, سبزه, کورمال, کو]
1 ... تبر, زدی, ریشهام, جوانه, روید, زخم‌راندی, دل]
2 ... پدر, سوخته, یابام, بهم, میگه, معنی, شرایط, مع]
3 ... مواجه, اخم, اقتدار, ات, چشمانت, بگذار, واقعیت]
4 ... مدح, شادی, غم, نوشتندبا, نوشتند, خنده, لبث, ت]
...
20064 ... پرخاشگری, پلیس, نیروی, غیرحرفهای, گرگهای, ممن]
20065 ... , فروید, میگوید, تمدن, آغاز, انسان, سنگ, کلمه]
20066 ... صبح, امروز ۱۱۵۰۰۰, راننده, خطی, ایستگاه, میدان]
20067 ... میگیرم, ملقمه, بیشعوری, فرومایگی, شرفی, ست, پا]
20068 ... مهم, مزیت, آشکارتر, چهره, مشمنزکننده, دیکتا تو]
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 ... , خوابم, دیدم, دست, اخم, قاسم, سبزه, عین, زدن]
1 ... دل, روید, لبخند, زخم‌راندی, جوانه, ریشهام, زد]
2 ... شرایط, سوخته, معنی, معنیشو, باره, پدر, بشم_پ]
3 ... , بگذار, مواجه, عکس, اخم, چشمانت, واقعیت, کپی]
4 ... غم, نوشتندسید, اخم, مدح, خنده, تصنیف, لبث, نو]
...
20064 ... ممنونم, پلیس, غیرحرفهای, نیروی, وقتتان, پرخاش]
20065 ... سنگها, انسان, فروید, متمدن, دردناک, انسان, می]
20066 ... , آزادیس, درخواست, ایستگاه, شکایت, محض, ماسک]
20067 ... , کونت, کارتل, گیو, یارس, زیرخواب, میکیا_ینهمه]
20068 ... , پرخاشگری, دیکتا_تورما_بانه, مهم, دنیا_ی, نقاب]
Name: shuffled_tokens, Length: 20069, dtype: object
```

شکل ۵: کلمات بهم ریخته در جمله

از میان پارامترهای این دیتاست، طول جمله و تعداد هشتک‌های هر توییت را به عنوان ویژگی‌های مدل استخراج کردیم و در نهایت ابعاد داده ما (15, 20069) شد. و آن را در فایل به نام processed\_anger.csv ذخیره می‌کنیم تا از آن در مرحله بعد استفاده کنیم.

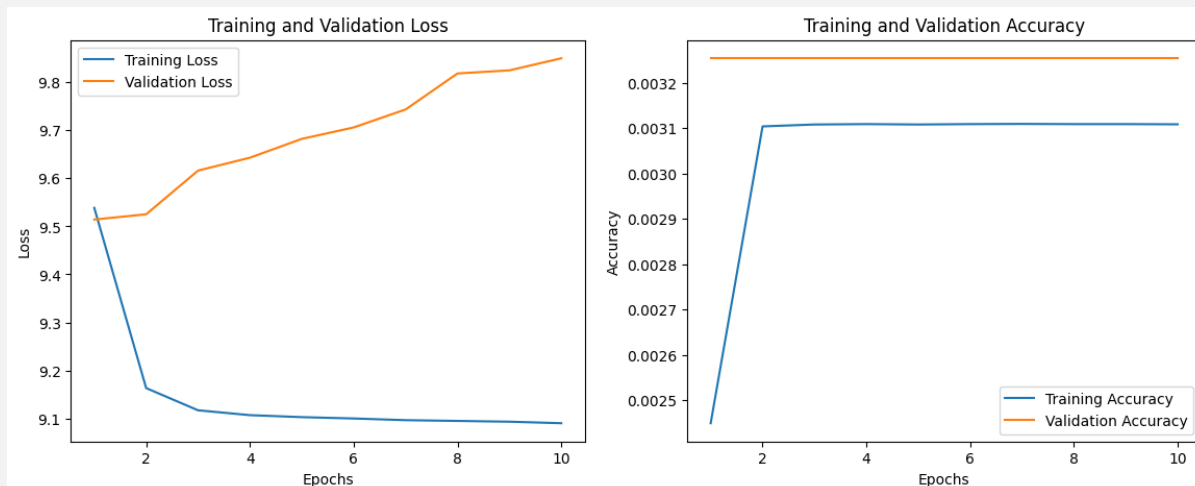
tweet	replyCount	retweetCount	likeCount	quoteCount	hashtags	sourceLabel	emotion	cleaned_tweet	tokens	tweet_length	num_hashtags	sourceLabel_encoded	emotion_encoded	shuffled_tokens
0	0	3	2	0	[NoZBR]	Twitter Web App	anger	ایستگاه خواب دیدم بمبی زدن نورش سبزه کورمال کو	خواب دیدم بمبی زدن نورش سبزه کورمال کو	260	1	0	0	... خوابم دیدم دست اخم قاسم سبزه عین زدن
1	0	0	8	0	[چین_کتاب]	Twitter for Android	anger	تبر زدی ریشهام جوانه روید زخم‌راندی دل	تبر زدی ریشهام جوانه روید زخم‌راندی دل	76	1	0	0	... تبر زدی ریشهام جوانه روید زخم‌راندی دل
2	1	0	11	0	[پرس_اوق]	Twitter for Android	anger	پدر سوخته یابام بهم میگه معنی شرایط مع	پدر سوخته یابام بهم میگه معنی شرایط مع	232	1	1	0	... پدر سوخته یابام بهم میگه معنی شرایط مع
3	0	0	1	0	[مجنه_عصر]	Twitter for iPhone	anger	بگذار مواجه عکس اخم چشمانت واقعیت کپی	بگذار مواجه عکس اخم چشمانت واقعیت کپی	115	2	2	0	... بگذار مواجه عکس اخم چشمانت واقعیت کپی
4	4	6	36	0	[بنا_طنز]	Twitter Web App	anger	غم نوشتندسید اخم مدح خنده تصنیف لبث نو	غم نوشتندسید اخم مدح خنده تصنیف لبث نو	142	1	0	0	... غم نوشتندسید اخم مدح خنده تصنیف لبث نو
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
20064	0	1	5	0	[مصلحتان]	Twitter for iPhone	anger	ممنونم پلیس غیرحرفهای نیروی وقتتان پرخاش	ممنونم پلیس غیرحرفهای نیروی وقتتان پرخاش	65	1	2	0	... ممنونم پلیس غیرحرفهای نیروی وقتتان پرخاش
20065	0	3	23	0	[مصلحتان]	Twitter for Android	anger	سنگها انسان فروید متمدن دردناک انسان می	سنگها انسان فروید متمدن دردناک انسان می	195	1	1	0	... سنگها انسان فروید متمدن دردناک انسان می
20066	3	0	3	0	[زند]	Twitter for Android	anger	آزادیس درخواست ایستگاه شکایت محض ماسک	آزادیس درخواست ایستگاه شکایت محض ماسک	249	1	1	0	... آزادیس درخواست ایستگاه شکایت محض ماسک
20067	2	0	3	0	[آنتی_سوس]	Twitter Web App	anger	کونت کارتل گیو یارس زیرخواب میکیا_ینهمه	کونت کارتل گیو یارس زیرخواب میکیا_ینهمه	258	2	0	0	... کونت کارتل گیو یارس زیرخواب میکیا_ینهمه
20068	1	3	50	0	[کتاب_مصلحتان]	Twitter Web App	anger	پرخاشگری دیکتا_تورما_بانه مهم دنیا_ی نقاب	پرخاشگری دیکتا_تورما_بانه مهم دنیا_ی نقاب	185	2	0	0	... پرخاشگری دیکتا_تورما_بانه مهم دنیا_ی نقاب

شکل ۶: دیتاست پردازش شده

در مرحله بعد که مرحله آموزش باشد، داده‌های پیش‌پردازش شده را از فایل processed\_anger.csv می‌خوانیم و توکن‌ها را به string تبدیل می‌کنیم:

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#). Adjust cell output [settings](#)...

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 محدودیت‌های سخت‌افزاری، Colab در افزایش اندازه داده‌های ورودی، نمی‌توانیم ورودی‌های زیادی را در اندازه‌های LLM ها 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 ورودی‌های دیتاست بهم‌ریخته را به شبکه می‌دهیم و انتظار داریم که شبکه کلمات را مرتب کند. برای مثال ورودی زیر را به شبکه می‌دهیم:

Input: <pad> <unk> سوال خورد کتان صبا مختصر حکایت آیدکه دشنام بریزد غم چراغ جواب شکر استکه دانی سیراباگر گوش مهتابدعات داردو ای سهل گر تشنه بمبرد گل  
Expected Output: <pad> سوال خورد کتان صبا مختصر حکایت آیدکه دشنام بریزد غم چراغ جواب شکر استکه دانی سیراباگر گوش مهتابدعات داردو ای سهل گر تشنه بمبرد گل

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 به نظر 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

مجموعه داده 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 مجموعه داده پیاده‌سازی کنیم. موارد زیر را دنبال کنید:

۱. دو فایل `in_domain_train.tsv` و `out_of_domain_dev.tsv` در اختیار شما قرار گرفته است. این فایل‌ها را در محیط برنامه‌نویسی خود بارگذاری 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 تنظیم-دقیق 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 (4, 8551) 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 x 4 columns

شکل ۱۱: داده‌های آموزش دیتاست `in_domain_train.tsv`

پاسخ

همانند سوال قبل، پیش‌پردازش‌های لازم مانند Tokenize کردن و حذف کلمات Stop، و اضافه نمودن [CLS] و [SEP] به توکن‌ها را انجام می‌دهیم.

برای مثال، خروجی Tokenize شده یکی از ورودی‌ها به صورت 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 تعریف 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 به صورت 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 در نهایت 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 به صورت زیر به دست 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 epoch took: 0:01:18

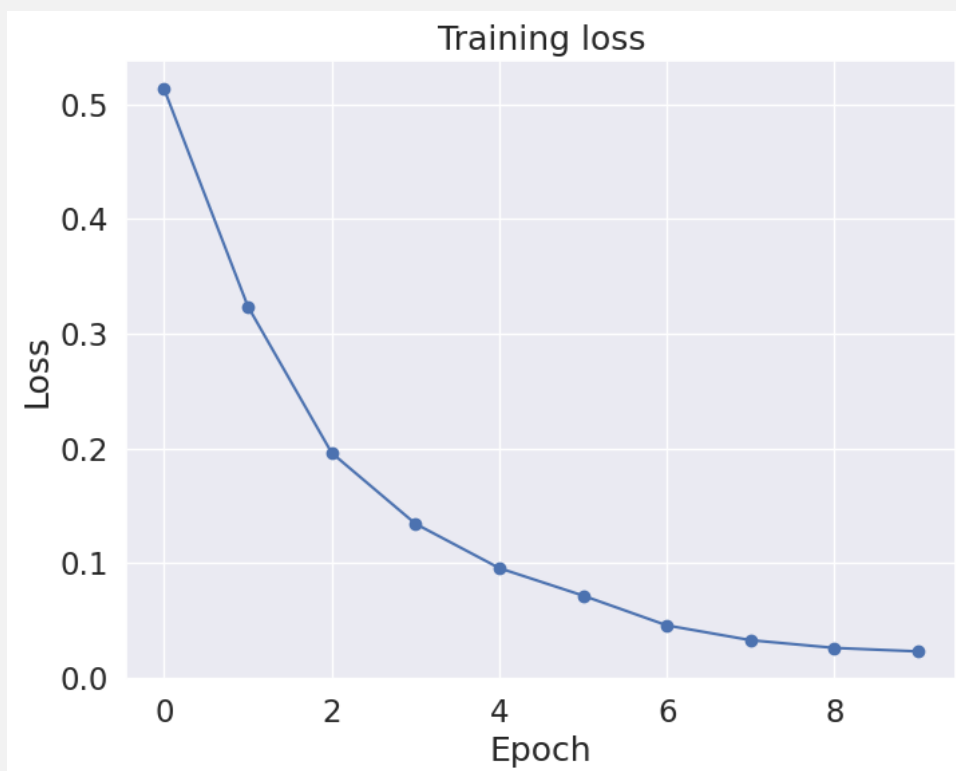
Running Validation...
Accuracy: 0.83
Validation took: 0:00:03

Training complete!
```

شکل ۱۵: مقدار دقت و خطا در پایان آموزش

پاسخ

نمودار خطای آموزش برحسب تعداد Epoch نیز به صورت زیر بدست می‌آید:



شکل ۱۶: نمودار خطای آموزش