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

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 $\phi(K)$: $O(Nr)$

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.

پاسخ

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.)

پاسخ

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.

پاسخ

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.

پاسخ

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.

پاسخ

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

پاسخ

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 ...

پاسخ

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

پاسخ

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

پاسخ

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).

پاسخ

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.

پاسخ

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