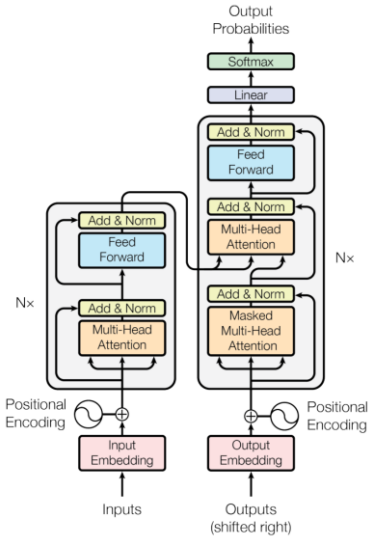
رایانش تکاملی

تمرین سوم

جستجوی معماری عصبی یا Neural Architecture Search روشی برای اتوماسیون طراحی شبکه‌های عصبی عمیق است که به طور گسترده در زمینه یادگیری ماشین مورد استفاده قرار می‌گیرد. از NAS معمولا برای طراحی شبکه‌هایی استفاده می‌شود که از معماری‌هایی با طراحی دستی بهتر عمل می‌کنند. هدف NAS جستجوی معماری شبکه عصبی مقاوم و با کارایی مناسب است که به وسیله انتخاب و ترکیب اعمال مختلف پایه از قبل تعریف شده در فضای جستجو صورت می‌گیرد. یکی از اصلی‌ترین روش‌ها در NAS، الگوریتم‌های تکاملی می‌باشند. الگوریتم‌های تکاملی می‌توانند عملیات جستجوی معماری را به صورت هوشمند طی کنند و دستیابی به معماری بهینه را سرعت ببخشند. در این تمرین با استفاده از این الگوریتم‌ها به دنبال یک معماری مناسب مبتنی بر ترنسفورمرها برای دسته‌بندی متون هستیم. در بخش ۱ معماری ترنسفورمر توضیح داده شده است. در بخش ۲ مسئله طراحی شبکه و مقادیر ابرپارامترهای ممکن برای شبکه معرفی شده است. در بخش ۳ ملاحظات لازم برای حل مسئله و در بخش ۴ مواردی که باید تحویل داده شوند، مشخص شده است. مهلت تحویل این تمرین پایان روز یکشنبه ۱۳ آذر ۱۴۰۱ خواهد بود.

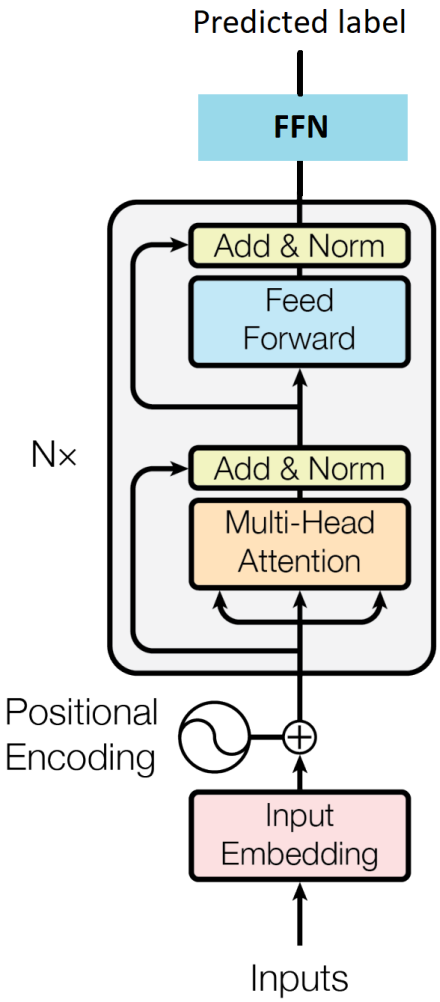
۱ – تشریح معماری ترنسفورمر

شکل ۱ معماری اصلی شبکه عصبی ترنسفورمر را که برای اولین بار در سال ۲۰۱۷ توسط تیمی از محققان گوگل در مقاله‌ای با نام Attention is all you need ارائه شد، نشان می‌دهد. این مقاله در ابتدا مدل ترنسفورمر را برای مسئله ترجمه ماشینی معرفی کرد ولی امروزه این مدل برای مسائل زیادی در همه زمینه‌های متن، تصویر، صوت و ... استفاده می‌شود. شبکه ترنسفورمر همچون بسیاری از مدل‌های sequential دیگر از یک انکودر (بخش سمت چپ شکل) و یک دیکودر (بخش سمت راست شکل) تشکیل می‌شود. وظیفه انکودر گرفتن دنباله ورودی و نگاشت آن به یک فضای پیوسته مخفی است. دیکودر هم با گرفتن خروجی انکودر به دنبال تولید دنباله خروجی مناسب می‌باشد.



شکل ۱: معماری اصلی ترنسفورمر

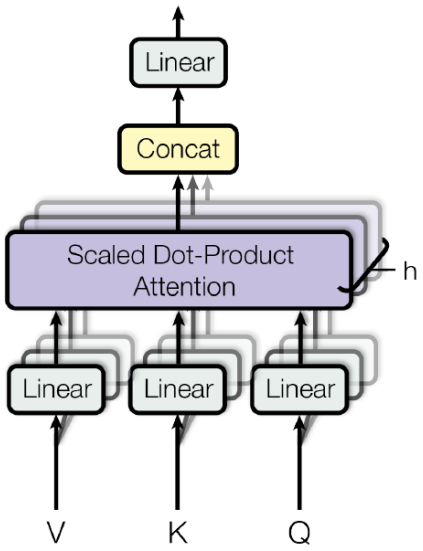
از آن جا که در مسئله دسته‌بندی متن، تولید دنباله خروجی معنا ندارد، به بخش دیکودر شکل ۱ نیازی نیست و فقط از بخش انکودر آن استفاده می‌شود. شکل ۲ معماری مد نظر برای مسئله دسته‌بندی متن را نشان می‌دهد. این شکل همان انکودر مدل اصلی ترنسفورمر است که یک شبکه fully connected به بالای آن به منظور انجام نگاشت نهایی و عملیات دسته‌بندی افزوده شده است. این مدل از N لایه ترنسفورمری تشکیل می‌شود که خروجی هر لایه به عنوان ورودی لایه بعدی استفاده می‌گردد. در ابتدا دنباله ورودی که شامل کل کلمات متن است، توکن‌بندی می‌شود و پس از عبور از المان‌های Input Embedding و Positional Encoding وارد لایه اول ترنسفورمری می‌شود. وظیفه المان Input Embedding در واقع تبدیل هر کلمه به یک بردار است. روش‌های مختلفی برای تبدیل کلمه به بردار وجود دارد که ساده‌ترین آن‌ها بردار one hot است. در این روش هر کلمه در دیکشنری یک اندیس منحصر به فرد دارد و اگر به عنوان مثال اندیس یک کلمه ۵ باشد، پنجمین المان بردار منتسب به آن کلمه ۱ و بقیه المان‌های آن صفر است. وظیفه المان Positional Encoding هم کدگذاری جایگاه نسبی هر کلمه در دنباله ورودی است که این کار را با الحاق یک بردار اضافی به بردار embedding هر کلمه انجام می‌دهد.



شکل ۲: معماری مورد استفاده برای مسئله دسته‌بندی متن

بردار ورودی به لایه اول ترنسفورمر، یک بردار d_{model} بُعدی است. در ادامه این بردار باید از بلوک MultiHead Attention عبور کند و خروجی این بلوک به همراه ورودی آن در بلوک Add & Norm اولی جمع و سپس نرمالیزه شوند. بردار حاصل از یک شبکه Feed Forward (که یک شبکه fully connected است) عبور می‌کند و خروجی این شبکه به همراه ورودی آن در بلوک Add & Norm دومی جمع و نرمالیزه می‌شوند. خروجی هر لایه ترنسفورمری همچنان یک بردار d_{model} بُعدی است. بردار خروجی از لایه‌های ترنسفورمری در ادامه به یک شبکه FFN (که یک شبکه fully connected است) داده می‌شود تا دسته‌بندی نهایی برای آن انجام گردد.

بلوک MultiHead Attention: شکل ۳ جزییات این بلوک را نشان می‌دهد. این بلوک سه بردار Q، K و V با بُعد d_{model} را به عنوان ورودی دریافت می‌کند و هر یک را با استفاده از یک لایه fully connected (که در شکل با بلوک Linear نشان داده شده است)، به بردارهایی به ترتیب d_{k} ، d_{k} و d_{v} بُعدی تبدیل می‌کند. این سه بردار در ادامه از h عملگر Attention عبور می‌کنند که خروجی هر یک از این عملگرها برداری d_{v} بُعدی است. در انتها این بردارها الحاق (Concatenate) می‌گردند تا برداری $h\, d_{\text{v}}$ بُعدی تشکیل شود و پس از عبور از یک لایه fully connected خروجی d_{model} بُعدی تولید می‌شود.

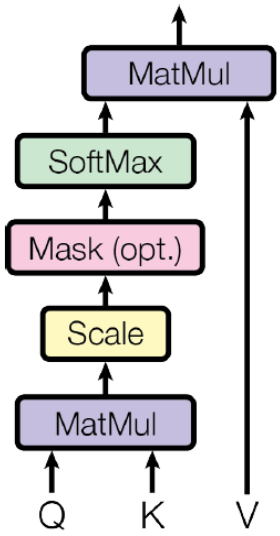


شکل ۳: بلوک MultiHead Attention

عملگر Attention: شکل ۴ جزییات عملگر Attention را نشان می‌دهد. این عملگر سه ورودی Q، K و V را که به ترتیب d_{k} ، d_{k} و d_{v} بُعدی هستند را دریافت و بر اساس رابطه زیر خروجی را محاسبه می‌کند:

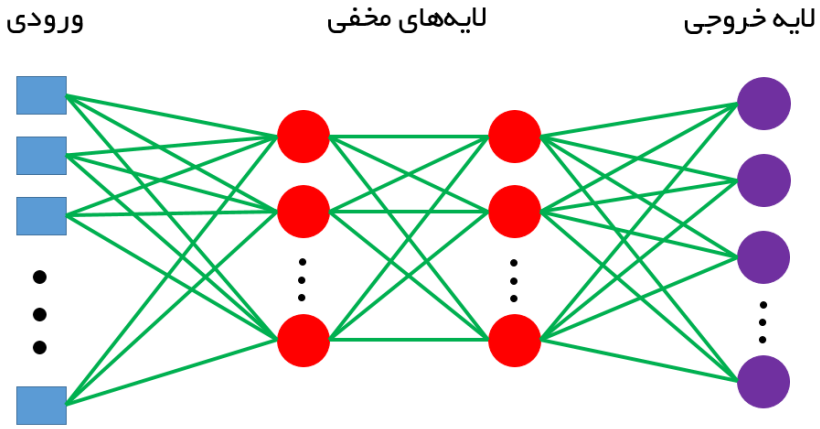
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

در واقع عملگر Attention تعیین می‌کند که دنباله ورودی Q و K به چه صورت و چه میزان به هم توجه کنند و این توجه را به صورت ضرایب V تبدیل می‌کند.



شکل ۴: عملگر Attention مورد استفاده در بلوک MultiHead Attention

بلوک‌های Feed Forward و FFN: این دو بلوک هر دو یک شبکه fully connected هستند که ورودی آن‌ها یک بردار d_{model} بُعدی است. لایه خروجی Feed Forward از تابع فعال‌سازی Linear استفاده می‌کند و d_{model} تا نورون دارد. لایه خروجی FFN هم ۲ (به تعداد دسته‌ها) نورون دارد و تابع فعال‌سازی آن Softmax است. شکل ۵ شماتیک این شبکه‌ها را نشان می‌دهد.



شکل ۵: شبکه مورد استفاده در بلوک‌های Feed Forward (درون ترنسفورمر) و FFN (خارج از ترنسفورمر)

۲ – مسئله طراحی شبکه ترنسفورمر

می‌خواهیم شبکه معرفی شده در بخش قبل را بر روی مجموعه داده imdb که متشکل از نظرات کاربران در مورد فیلم‌های سینمایی است، آموزش دهیم. این نظرات به دو دسته مثبت و منفی تقسیم می‌شوند و شبکه آموزش داده شده با دریافت نظرات جدید (داده آزمایشی) قادر است آن‌ها را برچسب‌گذاری کند. پارامترهای این شبکه (وزن‌ها و بایاس‌ها) در فرایند آموزش، تعیین

می‌شوند ولی ابرپارامترهای آن باید قبل از فاز آموزش توسط طراح تعیین گردند. معمولا طراح شبکه با استفاده از تجربه خود و قوانین سرانگشتی که برای طراحی شبکه وجود دارد، ابرپارامترهای شبکه را تعیین می‌کند. به منظور اتوماسیون فرایند طراحی شبکه عصبی به خصوص در مسائلی که تعداد ابرپارامترها زیاد است، از NAS استفاده می‌گردد.

شما باید با استفاده از الگوریتم‌های تکاملی، ابرپارامترهای بهینه شبکه معرفی شده در بخش قبل را به منظور دسته‌بندی نظرات مجموعه داده imdb به دست آورید. شما می‌توانید از هر الگوریتم تکاملی دلخواه (چه داخل درس و چه خارج از درس) استفاده کنید. یک راه حل در واقع یک معماری کامل ترنسفورمری با مشخص بودن تمام ابرپارامترهای شبکه است. معیار برازندگی در این مسئله Accuracy روی مجموعه آزمایش است. بنابراین راه حل بهینه آن معماری است که به بیشترین Accuracy روی مجموعه آزمایشی منجر شود. جدول ۱ مقادیر ممکن برای ابرپارامترهای شبکه را نشان می‌دهد. شما باید از بین این مقادیر مجاز بهترین ترکیب ممکن را با استفاده از الگوریتم‌های تکاملی به دست آورید.

جدول ۱: ابرپارامترهای موجود در شبکه و مقادیر مجاز برای هر کدام

ابرپارامتر	مقادیر ممکن
تعداد لایه‌های ترنسفورمری (N)	۱-۲-۳
تعداد head های attention در هر لایه ترنسفورمری	۱-۲-۴-۸
تعداد لایه‌های مخفی Feed Forward داخل هر لایه ترنسفورمری	۰-۱-۲
تعداد لایه‌های مخفی FFN خارج از ترنسفورمر	۰-۱
تابع فعال‌سازی در هر لایه مخفی Feed Forward و FFN	ReLU-Sigmoid
تعداد نورون‌ها در هر لایه مخفی Feed Forward و FFN	۵-۱۰-۲۰-۳۰
احتمال Dropout در هرلایه مخفی Feed Forward و FFN	۰ تا ۱۰۰ درصد
d_{model}	۱۶-۳۲-۶۴-۱۲۸
هر یک از Norm ها در هر لایه ترنسفورمری	وجود یا عدم وجود

جدول ۲ نیز تنظیمات مورد نیاز برای حل مسئله را نشان می‌دهد که باید حتما آن‌ها را رعایت کنید.

جدول ۲: تنظیمات لازم برای حل مسئله

تعداد epoch	۵
تعداد نسل‌های الگوریتم تکاملی	۲۰
تعداد افراد جمعیت در هر نسل (popSize)	۱۰
تعداد اجرا برای هر ارزیابی هر عضو از جمعیت	۱۰

۳ – ملاحظاتی که در حل مسئله باید در نظر گرفته شوند

الف) شما باید نوع مدل‌سازی و جزییات روش نمایش خود را به طور شفاف مشخص کنید.

ب) شما باید به صورت کامل و صریح عملگرهای انتخاب و تغییر و نیز تاثیر آن‌ها بر بهترین پاسخ به دست آمده را توضیح دهید.

ث) با توجه به راه حل‌های به دست آمده در نسل‌های مختلف و برازندگی آن‌ها، چشم انداز برازندگی این مسئله را توصیف کنید.

ج) با استفاده از معماری‌های به دست آمده و برازندگی آن‌ها تعیین کنید که کدام یک از المان‌ها و بلوک‌های موجود در مسئله تاثیر بیشتری در دقت به دست آمده برای مسئله دسته‌بندی متن دارند.

توجه: برای ارزیابی هر فرد باید میانگین Accuracy آن در ۱۰ اجرا به عنوان برازندگی آن در نظر گرفته شود.

۴ – مواردی که باید تحویل داده شود

- فایل(های) کد برنامه مورد استفاده برای پیاده‌سازی تمرین در یک پوشه به نام Code
- فایل گزارش با نام Doc.pdf شامل موارد زیر:
  - نتایج حل مسئله NAS به همراه ملاحظات مشخص شده در بخش ۳
  - تشریح و تحلیل نتایج به دست آمده از نظر شما
  - هر گونه توضیح اضافی در مورد نحوه انجام تمرین

* دقت کنید که گزارش شما حتما باید به صورت یک گزارش فنی باشد.

- کد یک راه حل ممکن برای معماری شبکه عصبی در پیوست این تمرین به صورت فایل ژوپیتر موجود است.

فایل‌های کد و گزارش را به صورت یک فایل فشرده در قالب ZIP و با نام EC_Name_Family_HW3 در سایت کوئرا بارگذاری کنید (به جای Name نام و به جای Family نام خانوادگی خود را قرار دهید).

مهلت تحویل این تمرین تا پایان روز یکشنبه ۱۳ آذر خواهد بود.

موفق باشید

کارشناس