به نام خدا

تكليف سري پنجم NLP بهاره كاوسي نژاد – 99431217

سوالات تئوری

1. دو سناریوی آموزش مدلی مانند BERT را در نظر بگیرید که در اولی وزن های اولیه ی آموزش برابر با وزن های یک مدل BERT از قبل آموزش دیده شده (Pretrain) بر روی مجموعه تسک های مختلف می باشد. و در دومی وزن های اولیه مقادیری تصادفی می باشند. توضیح دهید هر کدام از مدل ها در فرآیند آموزش به طور تقریبی چگونه عمل خواهند کرد و همینطور انتظار داریم عملکرد مدل ها پس از فرآیند آموزش چگونه باشد؟

تفاوت‌های نحوه عملکرد مدل‌ها در طول train و عملکرد مورد انتظار بعد از train:

1. **وزن های مدل BERT از پیش آموزش دیده به عنوان وزن های train اولیه:**

در این سناریو، وزن‌های اولیه مدل بر روی وزن‌های یک مدل BERT از پیش آموزش‌دیده تنظیم می‌شوند که روی یک مجموعه بزرگ برای مجموعه‌ای از taskها (مانند مدل‌سازی زبان (Language Modeling)، طبقه‌بندی جملات (sentence classification) و غیره) آموزش داده شده است. مدل از پیش آموزش دیده، representationهای غنی از زبان را آموخته و اطلاعات زمینه ای (contextual) را به دست آورده است.

در طول فرآیند training:

* این مدل با یک عملکرد اولیه قوی شروع می شود زیرا قبلاً دانشی از مدل از پیش آموزش دیده دارد.
* این مدل با تنظیم دقیق وزن های از پیش آموزش دیده، به سرعت با task خاص سازگار می شود. از دانش عمومی آموخته شده از مدل از پیش آموزش دیده بهره خواهد برد، که می تواند به همگرایی (converge) سریعتر و دستیابی به عملکرد بالاتر کمک کند.

پس از طی مراحل training:

* انتظار می رود که این مدل در کاری که برای آن تنظیم شده است عملکرد خوبی داشته باشد، به خصوص اگر کار مشابه نمونه های از پیش آموزش دیده باشد.
* دانش و representationهای مدل از پیش آموزش دیده به مدل تنظیم شده منتقل می شود و به آن امکان می دهد به خوبی تعمیم یابد و بهتر از یک مدل اولیه تصادفی اولیه عمل کند، به خصوص زمانی که داده های آموزشی موجود محدود است.

1. **وزن های اولیه به طور تصادفی به عنوان وزن های train اولیه:**

در این سناریو، وزن‌های اولیه مدل به‌طور تصادفی مقادیری را بدون هیچ دانش قبلی از یک مدل از پیش آموزش‌دیده، تخصیص می‌دهند.

در طول فرآیند training:

* مدل بدون دانش خاص یا سوگیری (bias) های مرتبط با کار شروع می شود، که ممکن است منجر به همگرایی (convergence) کندتر و عملکرد اولیه کمتر در مقایسه با سناریوی مدل از پیش آموزش دیده شود.
* مدل باید الگوهای task-specific را از ابتدا (from scratch) و تنها با تکیه بر داده های آموزشی و فرآیند بهینه سازی (optimization) بیاموزد.

پس از طی مراحل training:

* عملکرد مدل به کیفیت و کمیت داده های آموزشی و همچنین پیچیدگی task بستگی دارد.
* ممکن است training iterationهای بیشتری طول بکشد تا مدل به عملکرد قابل مقایسه با سناریوی مدل از پیش آموزش دیده دست یابد زیرا نیاز به یادگیری تمام اطلاعات task-specific (به تنهایی از داده های آموزشی) دارد.

به طور خلاصه، هنگام استفاده از وزن‌های مدل BERT پیش‌آموزش‌شده به‌عنوان وزنه های آموزشی اولیه، مدل از دانش قبلی و representation‌های آموخته‌شده از یک مجموعه بزرگ سود می‌برد که منجر به هم‌گرایی (convergence) سریع‌تر و عملکرد بالقوه بهتر می‌شود. از سوی دیگر، هنگام شروع با وزنه‌های تصادفی، مدل باید همه چیز را از ابتدا یاد بگیرد، که ممکن است منجر به همگرایی کندتر شود و برای دستیابی به عملکرد قابل مقایسه به training iterationهای بیشتری نیاز داشته باشد.

1. یکی از چالش های قابل توجه در فرآیند آموزش یک مدل شبکه عصبی بخصوص در فرآیند Fine-Tuning، Catastrophic Forgetting می باشد. این چالش را به طور کامل توضیح دهید و روشی ارائه دهید که بتوانیم این چالش را کاهش دهیم و یا برطرف کنیم. (منابعی که در آن ها جست و جو کردید را ارائه دهید.)

Catastrophic Forgetting به پدیده‌ای اشاره دارد که در آن یک مدل، زمانی که در معرض داده‌های جدید قرار می‌گیرد، تمایل دارد دانش قبلی را فراموش کند. این موضوع به ویژه در سناریوهایی که یک مدل نیاز به یادگیری و انطباق با جریانی از وظایف در طول زمان دارد برجسته می شود.

ماهیت Catastrophic Forgetting

شبکه‌های عصبی، ستون فقرات بسیاری از مدل‌های یادگیری ماشینی مدرن، با تنظیم وزن اتصالات بین نورون‌ها در طول train، یاد می‌گیرند. هنگامی که یک مدل بر روی داده های جدید آموزش می بیند، این وزن ها به روز می شوند تا الگوها و روابط موجود در اطلاعات جدید را به تصویر بکشند. با این حال، چالش زمانی ایجاد می‌شود که این به‌روزرسانی‌ها منجر به کاهش عملکرد در وظایفی شود که مدل قبلاً بر آنها مسلط شده بود.

سناریویی را در نظر بگیرید که در آن یک شبکه عصبی در ابتدا برای تشخیص ارقام دست نویس آموزش دیده است. اگر مدل پس از آن وظیفه یادگیری شناسایی حیوانات را داشته باشد، تنظیماتی که در وزن‌ها برای تشخیص حیوانات انجام می‌شود ممکن است با نمایش‌های آموخته‌شده قبلی برای ارقام تداخل داشته باشد و منجر به کاهش عملکرد در کار تشخیص رقم شود. این تغییر Catastrophic Forgetting است.

پیامدهای Catastrophic Forgetting

Catastrophic Forgetting می تواند پیامدهای عمیقی داشته باشد، به ویژه در کاربردهای دنیای واقعی که انتظار می رود مدل ها به طور مداوم یاد بگیرند و تطبیق دهند. برخی از مفاهیم کلیدی عبارتند از:

1. چالش های یادگیری مادام العمر: در سناریوهایی که مدل ها باید از یک جریان مستمر از taskها یاد بگیرند، Catastrophic Forgetting می تواند مانع حفظ دانش به دست آمده از وظایف قبلی شود. این امر به ویژه در برنامه هایی مانند سیستم های autonomous، که در آن مدل نیاز به تطبیق مداوم با محیط ها و وظایف جدید دارد، مشکل ساز است.
2. ناکارآمدی منابع (Resource Inefficiency): نیاز به آموزش مجدد مدل‌ها در مورد taskهایی که قبلاً آموخته‌اند می‌تواند از نظر محاسباتی گران و ناکارآمد باشد. در شرایطی که منابع محاسباتی محدود هستند، کاهش Catastrophic Forgetting برای اطمینان از عملکرد بهینه ضروری است.
3. Inhibited Autonomous Learning: برای سیستم‌های مستقل و روبات‌هایی که از تجربه یاد می‌گیرند، Catastrophic Forgetting می‌تواند مانع از توانایی آن‌ها در انباشت دانش در طول زمان شود. این محدودیت ممکن است بر سازگاری و استقلال چنین سیستم هایی تأثیر بگذارد.

کاهش Catastrophic Forgetting: رویکردها و راهبردها

پرداختن به فراموشی فاجعه آمیز یک حوزه فعال تحقیقاتی است و چندین استراتژی برای کاهش اثرات آن پیشنهاد شده است. به برخی از رویکردهای برجسته می پردازیم:

1. تکنیک های Rehearsal:

* بافر داده های گذشته: یک رویکرد ساده این است که یک بافر از داده های گذشته را حفظ کنید و به طور دوره ای آن را در فرآیند آموزش قرار دهید. این به مدل کمک می کند تا دانش taskهای قبلی را مجدداً بررسی و تقویت کند.
* بازپخش (Replay کردن) تجربه: این شامل نمونه برداری از بافر replay در طول آموزش، شکستن همبستگی (correlation) های متوالی و افزایش ثبات در حین یادگیری است.

1. تکنیک های Regularization:

* تثبیت وزن الاستیک (Elastic Weight Consolidation): EWCیک عبارت جریمه را در تابع ضرر بر اساس اهمیت پارامترها برای taskهایی که قبلاً آموخته شده اند معرفی می کند. این به محدود کردن تغییرات در وزن‌ها که برای حفظ دانش قدیمی حیاتی است، کمک می‌کند.
* هوش سیناپسی: مشابه EWC، این روش اهمیت وزن‌های individual را در نظر می‌گیرد و برای جلوگیری از تغییرات شدید در پارامترهای حیاتی، شرایط جریمه را معرفی می‌کند.

1. راه حل های معماری:

* معماری modular: طراحی شبکه‌های عصبی با اجزای ماژولار برای taskهای مختلف به جداسازی تأثیر داده‌های جدید بر روی بخش‌های خاص شبکه کمک می‌کند و تداخل با representationهای آموخته‌شده برای کارهای قبلی را کاهش می‌دهد.
* شبکه‌های عصبی پیشرونده (Progressive Neural Networks): PNN‌ها شبکه‌های جدیدی را برای هر task جدید اضافه می‌کنند در حالی که شبکه‌های قدیمی ثابت نگه می‌دارند. در طول intference، خروجی های همه شبکه ها با هم ترکیب می شوند.

1. شبکه های Memory-Augmented:

* ماشین‌های تورینگ عصبی (Neural Turing Machines) و شبکه‌های حافظه: مدل‌هایی که اجزای حافظه خارجی را در خود جای داده‌اند، امکان نگهداری انتخابی اطلاعات مهم از taskهای قبلی را فراهم می‌کنند.
* حافظه Gradient Episodic (GEM): GEM، memoryهای episodic را در طول train ذخیره می کند و از فراموشی هنگام یادگیری taskهای جدید جلوگیری می کند.

1. معماری پویا:

* میانگین وزن پویا (DWA): Dynamic Weight Average به صورت پویا سهم وزنه های قدیمی و جدید را در طول train تنظیم می کند تا به task فعلی اهمیت بیشتری بدهد.
* شبکه‌های پیشرو (Progressive Networks): ساخت لایه‌های شبکه عصبی جدید برای هر کار جدید امکان انطباق با حفظ دانش از taskهای قبلی را فراهم می‌کند.

1. روش های Ensemble:

* جنگل های یادگیری مادام العمر (Lifelong Learning Forests): درخت های تصمیم گیری یا جنگل های تصادفی برای هر کار آموزش داده می شوند و مجموعه ای از این درختان برای inference استفاده می شود. درخت های جدید برای کارهای جدید اضافه می شوند.
* شبکه‌های مدولار وظیفه محور (Task-Driven Modular Networks): شبکه‌ها به شکل ماژولار ساختار یافته‌اند و هر ماژول به یک کار خاص اختصاص دارد و به حفظ دانش مربوط به کار کمک می‌کند.

منابع در انتهای فایل آورده شده اند.

1. در رابطه با Transfer Learning و تفاوت های آن با Fine-Tuning تحقیق کنید و توضیح دهید در چه شرایطی از هرکدام استفاده می شود؟

Fine-tunning، یک مدل از پیش آموزش‌دیده را روی یک task خاص با تغییرات جزئی تطبیق می‌دهد، در حالی که Transfer Learning از دانش به‌دست‌آمده از یک مدل از پیش آموزش‌دیده برای بهبود عملکرد در یک task متفاوت اما مرتبط استفاده می‌کند.

در حدول زیر این مقایسه به صورت دقیق تر آمده است:

|  |  |  |
| --- | --- | --- |
| جنبه | Fine-Tuning | Transfer Learning |
| هدف | تطبیق مدل از پیش آموزش دیده با یک task جدید | بهبود عملکرد با استفاده از یک مدل از پیش آموزش دیده در یک task مرتبط |
| رویکرد training | Train کردن کل مدل با داده جدید | فریز کردن برخی از لایه های مدل از پیش آموزش دیده و train لایه های خاص روی task جدید |
| داده مورد نیاز | نیازمند داده بیشتر مخصوص task جدید | موثر با مجموعه داده کوچکتر به دلیل استفاده از دانش از قبل آموزش دیده |
| استفاده | زمانی که داده های task-specific در دسترس باشند و منابع محاسباتی امکان train دوباره کامل را فراهم کنند | هنگامی که داده های برچسب گذاری شده یا منابع محاسباتی محدود باشند و taskها دارای شباهت هایی با هم هستند |
| پیچیدگی | پیچیده تر است زیرا شامل بازآموزی کل مدل می شود. | پیچیدگی کمتری دارد زیرا اغلب شامل فریز کردن برخی از لایه ها و train کردن فقط لایه های خاص است. |
| مثال | Fine-tune کردن BERT برای sentiment analysis روی دیتاست جدید | استفاده از یک مدل ImageNet از پیش آموزش دیده برای بهبود image classification روی دیتاست جدید |

1. راجع به تاثیرات روش های مختلف masking و همینطور تعیین میزان توکن های قابل mask بر روی فرآیند آموزش و عملکرد مدل های MLMs توضیح دهید. (روش رندوم و روش مبتنی بر part of speech )

در Masked Language Modelها (MLMs)، masking یک مرحله مهم است که در آن tokenهای خاصی در متن ورودی با یک mask token ویژه جایگزین می‌شوند. سپس مدل آموزش داده می شود تا masked tokenهای اصلی را پیش بینی کند. روش‌های مختلفی برای mask کردن توکن‌ها وجود دارد، و انتخاب روش می‌تواند بر روند آموزش و عملکرد MLM تأثیر بگذارد.

1. Random Masking:

* Method: در این رویکرد، درصد مشخصی از توکن های موجود در متن ورودی به صورت تصادفی انتخاب شده و با mask جایگزین می شوند. token های انتخاب شده می توانند اسم، افعال، صفت یا هر نوع token دیگری باشند.
* Training Procedure: Random masking، randomness را وارد فرآیند آموزش می کند، زیرا مدل باید توکن های اصلی را بر اساس اطلاعات ناقص پیش بینی کند. این مدل را تشویق می‌کند تا بازنمایی‌های قوی را بیاموزد که بتواند اطلاعات از دست رفته را مدیریت کند. همچنین به جلوگیری از اتکای بیش از حد مدل به الگوها یا token های خاص در ورودی کمک می کند.
* عملکرد MLMها: Random masking به MLM اجازه می دهد تا الگوهای زبان عمومی و وابستگی ها را یاد بگیرد، زیرا نیاز به پیش بینی انواع مختلف tokenها دارد. مدل یاد می گیرد که context را درک کند و بدون توجه به part of speech خاصی یا نوع tokenای که mask شده است، پیش بینی کند.

2. Part-of-Speech Based Masking:

* Method: در این رویکرد، انواع خاصی از token ها بر اساس part of speechشان انتخاب شده و با mask token جایگزین می شوند. به عنوان مثال، تمام اسم ها یا تمام افعال در متن ورودی می توانند ماسک شوند.
* Training Procedure: Masking مبتنی بر part-of-speech، مدل را قادر می سازد تا بیشتر بر جنبه های زبانی خاص، مانند اسم ها یا افعال تمرکز کند. با mask کردن token‌های part-of-speech بخصوص، مدل تشویق می‌شود تا روابط نحوی و معنایی مربوط به آن token‌های خاص را به تصویر بکشد. این می تواند در taskهایی که درک یک part-of-speech بخصوص مهم است، مفید باشد.
* عملکرد MLMها: Masking بر اساس Part-of-Speech می تواند به MLM کمک کند تا در جنبه های زبانی خاصی تخصص پیدا کند. برای مثال، اگر مدلی عمدتاً با افعال mask دار آموزش داده شود، ممکن است در درک روابط مربوط به فعل و گرفتن الگوهای خاص فعل در زبان ماهرتر شود. این تخصص می تواند در کارهای پایین دستی که به شدت به اطلاعات مربوط به فعل متکی هستند مفید باشد.

در هر دو روش، انتخاب توکن‌های maskable بر رویه آموزشی و عملکرد MLM تأثیر می‌گذارد. Random masking، randomness را معرفی می‌کند و مدل را تشویق می‌کند تا بازنمایی‌های زبانی قوی و کلی را بیاموزد. Masking بر اساس Part-of-Speech به مدل اجازه می دهد تا بر جنبه های زبانی خاص تمرکز کند و در گرفتن الگوهای مربوط به Part-of-Speech انتخابی تخصص داشته باشد.

1. عملکرد معماری های CLM، MLM و Seq2Seq را با یکدیگر مقایسه کنید و مزایا و معایب هرکدام را بیان کنید. و از هرکدام نمونه هایی را مثال بزنید.
2. با توجه به عمکرد مدل های MLM (پیشبینی توکن های mask شده)، چگونه می توان از آن ها برای تولید یک دنباله ای از متن استفاده کرد؟

سوالات عملی

1. در نوت بوک q7.ipynb می خواهیم معماری یک مدل MLM را پیاده سازی کنیم. آن را تکمیل کنید. (سوالات تشریحی درون نوت بوک را در پی دی اف پاسخ سوالات تئوری قرار دهید)

**منابع سوال دوم:**

* <https://medium.com/@navarai/understanding-and-mitigating-catastrophic-forgetting-in-machine-learning-d5caa93d375e>
* <https://en.wikipedia.org/wiki/Catastrophic_interference>
* <https://neurosciencenews.com/ai-continuous-learning-23671/>

**منابع دیگر:**

* <https://www.geeksforgeeks.org/what-is-the-difference-between-fine-tuning-and-transfer-learning/>
* <https://www.techtarget.com/searchenterpriseai/definition/masked-language-models-MLMs>
* <https://satoricyber.com/data-masking/data-masking-8-techniques-and-how-to-implement-them-successfully/>
* <https://medium.com/@tom_21755/understanding-causal-llms-masked-llm-s-and-seq2seq-a-guide-to-language-model-training-d4457bbd07fa>