تمرین دوم

نام درس: یادگیری عمیق

استاد درس: دکتر محمدرضا محمدی

نام: محمد حقیقت

شماره دانشجویی: 403722042

گرایش: هوش مصنوعی

دانشکده: مهندسی کامپیوتر

نیم سال دوم 1403-1404

سوال اول

آ)

هدف اصلی استفاده از روش‌های تنظیم دقیق با پارامتر بهینه (PEFT) در مدل‌های زبانی، سفارشی‌سازی مدل‌های زبانی از پیش‌آموزش‌دیده (PLMs) با هزینه حافظه و ذخیره‌سازی کمتر و عملکرد رقابتی است. در PEFT، یك ماژول سبك‌وزن برای هر مجموعه داده یاد گرفته می‌شود در حالی که مدل زبانی از پیش‌آموزش‌دیده زیربنایی بدون تغییر باقی می‌ماند. این امر منجر به ایجاد چندین ماژول فشرده می‌شود که مهارت‌های متنوعی را هنگام اعمال بر روی دامنه‌ها و وظایف مختلف نشان می‌دهند.

دو روش PEFT که در این مقاله بررسی شده‌اند عبارتند از:

LoRA (Low-Rank Adaptation): این روش، ماتریس‌های وزن در ترانسفورمر را با افزودن حاصل‌ضرب دو ماتریس کوچك‌تر (با رتبه پایین) تغییر می‌دهد ($h \leftarrow h+BAx$). در عمل، LoRA در ماتریس‌های پروجکشن پرس‌وجو (query) و مقدار (value) در ماژول توجه استفاده می‌شود. ماتریس A با توزیع گوسی تصادفی و ماتریس B با صفر مقداردهی اولیه می‌شوند تا در ابتدای آموزش، مدل از پیش‌آموزش‌دیده بازیابی شود. ماژول‌های پارامتر-بهینه در LoRA شامل $\theta lora = \{A, B\}$ هستند.

$(IA)^3$ (Infused Adapter by Inhibiting and Amplifying Inner Activations): این روش بردارهای قابل آموزشی به نام‌های l_k، l_v و l_{ff} را معرفی می‌کند تا به ترتیب کلیدهای توجه، مقادیر توجه و فعال‌سازی‌های داخلی در شبکه‌های فیدفورواد را تغییر مقیاس دهد ($h \leftarrow l \odot h$).

این بردارها در ابتدا با مقدار یك مقداردهی اولیه می‌شوند تا مدل در شروع آموزش بدون تغییر بماند. ماژول پارامتر-بهینه در $(IA)^3$ شامل $\theta_{ia3} = \{l_k, l_v, l_{ff}\}$ است.

ویژگی متمایز اصلی این روش‌ها نسبت به روش‌های سنتی finetuning کامل (full finetuning) این است که تنها تعداد کمی از پارامترها را تنظیم می‌کنند و بیشتر پارامترهای از پیش‌آموزش‌دیده را ثابت نگه می‌دارند. این امر منجر به کاهش قابل توجه هزینه‌های حافظه و ذخیره‌سازی می‌شود و ماژول‌های حاصل فشرده و به راحتی قابل مدیریت و انتقال هستند.

ب)

مقاله نشان می‌دهد که ترکیب ماژول‌های PEFT (PEMs) از طریق عملیات حسابی خطی در فضای پارامترها می‌تواند به بهبود عملکرد منجر شود. دلایل این بهبود عبارتند از:

ادغام قابلیت‌ها: با جمع کردن پارامترهای ماژول‌های مختلف که هر کدام مهارت یا دانش خاصی را از داده‌های متفاوت کسب کرده‌اند، ماژول حاصل می‌تواند ترکیبی از این قابلیت‌ها را به ارث ببرد. به عنوان مثال، ترکیب ماژول‌های آموزش‌دیده بر روی زیرمجموعه‌های مختلف داده با توزیع‌های متفاوت می‌تواند به تعمیم‌پذیری بهتر بر روی توزیع کلی داده‌ها منجر شود. یا ترکیب ماژول‌های آموزش‌دیده بر روی وظایف مختلف می‌تواند یك یادگیرنده چندوظیفه‌ای ایجاد کند.

اتصال خطی (Linear Connectivity): این رویکرد از این فرضیه الهام گرفته شده است که مدل‌هایی که از یك نقطه شروع (checkpoint) یکسان از پیش‌آموزش‌دیده، تنظیم دقیق شده‌اند، اغلب در یك "حوضه خطا" (error basin) قرار می‌گیرند. بنابراین، پارامترهای آن‌ها را می‌توان به طور مستقیم با هم ترکیب (مثلاً جمع) کرد. از آنجایی که PEFTها تنها تغییرات کوچکی در مدل‌های از پیش‌آموزش‌دیده ایجاد می‌کنند، این خاصیت ممکن است برای پارامترهای PEFT نیز صادق باشد، به خصوص وقتی مقداردهی اولیه آن‌ها یکسان باشد.

روش پیشنهادی در مقاله نیازی به آموزش مجدد ماژول‌ها ندارد زیرا:

عملیات ساده حسابی: ترکیب ماژول‌ها از طریق عملیات حسابی خطی ساده مانند جمع و تفریق وزن‌دار پارامترهای ماژول‌ها انجام می‌شود. به عنوان مثال، عملگر جمع به سادگی به صورت $\theta add = \theta(1) + \theta(2)$ تعریف می‌شود.

عدم وجود پارامترهای قابل یادگیری جدید: در فرآیند ترکیب، هیچ پارامتر جدیدی که نیاز به یادگیری داشته باشد، معرفی نمی‌شود. تنها یك هایپرپارامتر وزنی λ وجود دارد که برای تعیین نسبت مشارکت هر ماژول در ترکیب نهایی استفاده می‌شود و این هایپرپارامتر بر روی یك مجموعه داده اعتبارسنجی (validation set) تنظیم می‌شود، نه از طریق آموزش گرادیانی.

ج)

طبق یافته‌های مقاله، ترکیب PEMهایی که با مقداردهی اولیه متفاوت (مثلا با استفاده از seedهای تصادفی متفاوت برای مقداردهی اولیه ماتریس A در LoRA) آموزش دیده‌اند، ممکن است منجر به کاهش جزئی در بهبود عملکرد نسبت به ترکیب ماژول‌هایی با مقداردهی اولیه یکسان شود.

دلیل این امر این است که مقداردهی اولیه متفاوت می‌تواند باعث شود که ماژول‌های PEFT پس از آموزش در حوضه‌های خطای (loss basins) متفاوتی قرار گیرند. وقتی ماژول‌ها در فضاهای پارامتری متفاوتی همگرا می‌شوند، ترکیب خطی ساده پارامترهای آن‌ها ممکن است به یك نقطه بهینه مشترك منجر نشود و در نتیجه کارایی مدل ترکیبی کمتر از حالت ایده‌آل باشد. با این حال، مقاله اشاره می‌کند که این کاهش عملکرد فاجعه‌بار نیست و ترکیب PEMها با مقداردهی اولیه متفاوت همچنان می‌تواند عملکرد بهتری نسبت به ماژول‌های منفرد اولیه ارائه دهد. شکل 3 در مقاله نیز این موضوع را با نشان دادن شباهت منحنی‌های عملکرد برای ترکیب PEMها با مقداردهی اولیه یکسان و متفاوت، تایید می‌کند

د)

مقاله نشان می‌دهد که ترکیب PEMها در فضای وزن می‌تواند منجر به عملکردی فراتر از حالت تك‌وظیفه‌ای (single-task) شود و تعمیم به وظایف یا داده‌های جدید را بهبود بخشد. این امر از طریق چندین آزمایش و نتایج آن‌ها نشان داده شده است:

تعمیم توزیع (Distribution Generalization):

در این آزمایش، دو PEM جداگانه بر روی دو زیرمجموعه از داده‌های یك وظیفه که دارای توزیع‌های متفاوت و نامتعادل از نظر برچسب هستند، آموزش داده می‌شوند. سپس این دو PEM با هم ترکیب می‌شوند $(\theta(1) \oplus \theta(2))$

جدول 2 نتایج این آزمایش را نشان می‌دهد. به عنوان مثال، برای وظیفه RTE، ترکیب ماژول LoRA و ماژول $(IA)^3$ به ترتیب بهبود مطلق 5.2 و 4.0 درصدی را نسبت به میانگین عملکرد دو PEM منفرد نشان می‌دهد. این نتایج نشان می‌دهد که PEM ترکیب‌شده توانایی تعمیم بهتری بر روی توزیع کلی داده‌ها دارد و این فراتر از عملکرد هر یك از PEMها بر روی زیرمجموعه خاص خود است.

چندوظیفه‌ای (Multi-tasking):

در این سناریو، PEMهای آموزش‌دیده بر روی وظایف مختلف (MNLI و RTE) با هم ترکیب می‌شوند $(\theta(1) \oplus \theta(2))$ تا یك PEM چندوظیفه‌ای ایجاد شود.

جدول 3 نشان می‌دهد که اگرچه ممکن است افت عملکرد جزئی در هر یك از وظایف منفرد نسبت به PEM آموزش‌دیده اختصاصی برای آن وظیفه وجود داشته باشد، اما PEM ترکیب‌شده LoRA بهبودی

در میانگین دقت دو وظیفه نشان می‌دهد که شاخصی از توانایی چندوظیفه‌ای مدل است. به عنوان مثال، میانگین دقت برای LoRA از 81.3 (بهترین مدل منفرد) به 82.5 افزایش یافته است.

شکل 2 نیز تغییرات دقت اعتبارسنجی MNLI و RTE را با مقادیر مختلف ضریب ترکیب λ برای LoRA نشان می‌دهد و چگونگی موازنه بین دو وظیفه را به تصویر می‌کشد.

انتقال دامنه (Domain Transfer):

ر اینجا، هدف انتقال دانش از یك دامنه منبع به یك دامنه هدف است که داده‌های برچسب‌دار در آن موجود نیست. این کار با استفاده از یك معادله قیاسی انجام می‌شود

مثلا $\theta yelp\_cls = \lambda\theta amazon\_cls \oplus (1-\lambda)(\theta yelp\_lm \ominus \theta amazon\_lm)$

جدول 5 نتایج را نشان می‌دهد. به عنوان مثال، برای انتقال به Yelp با استفاده از مدل T5-small و روش LoRA، دقت از 94.76% (منبع) به 95.83% (ترکیب) افزایش می‌یابد که بهبود معناداری است. این نشان می‌دهد که ترکیب PEMها می‌تواند به طور موثر دانش را بین دامنه‌ها منتقل کند، کاری که یك PEM منفرد آموزش‌دیده بر روی دامنه منبع به تنهایی نمی‌تواند به این خوبی انجام دهد.

این مثال‌ها نشان می‌دهند که با ترکیب هوشمندانه PEMها، می‌توان ماژول‌های جدیدی با قابلیت‌های گسترده‌تر یا تعمیم‌یافته‌تر ایجاد کرد که از توانایی‌های اجزای سازنده خود فراتر می‌روند.

۵)

با وجود مزایای نشان داده شده، مقاله به چند محدودیت در به‌کارگیری این روش در کاربردهای واقعی مدل‌های زبانی بزرگ (LLMs) اشاره می‌کند:

محدودیت به معماری و مقداردهی اولیه یکسان: در بیشتر آزمایش‌ها، ترکیب PEMها محدود به مواردی بوده است که معماری PEMها یکسان بوده و از مقداردهی اولیه مشابهی استفاده کرده‌اند. کاوش در ترکیب PEMها با معماری‌های متفاوت یا مقداردهی‌های اولیه متنوع به عنوان کارهای آتی ذکر شده است.

نیاز به تنظیم هایپرپارامتر وزنی λ: روش پیشنهادی نیازمند تنظیم هایپرپارامتر وزنی λ است که نسبت مشارکت هر ماژول را در ترکیب نهایی مشخص می‌کند. اگرچه این تنظیم بدون آموزش مجدد و بر روی مجموعه اعتبارسنجی انجام می‌شود، اما همچنان یك مرحله اضافی است. یافتن روش‌های خودکار برای محاسبه این هایپرپارامتر به عنوان کار آتی مطرح شده است.

وراثت سوگیری‌ها و نگرانی‌های ایمنی: ترکیب PEMهای موجود ممکن است منجر به وراثت سوگیری‌ها (biases) یا نگرانی‌های ایمنی شود که به طور ذاتی در این PEMها وجود دارند.

سم‌زدایی و طبیعت جعبه-سیاه: اگرچه آزمایش‌های سم‌زدایی (detoxification) نتایج مثبتی نشان داده‌اند، اما طبیعت جعبه-سیاه شبکه‌های عصبی ممکن است در برخی سناریوها به طور ضمنی سمیت را در مدل بگنجاند، حتی اگر در تنظیمات آزمایشی مشاهده نشده باشد.

این محدودیت‌ها نشان می‌دهد که اگرچه روش پیشنهادی بسیار امیدبخش است، اما برای کاربردهای عملی گسترده‌تر، به خصوص با LLMهای مدرن، نیاز به تحقیقات و توسعه بیشتری وجود دارد

سوال 2

آ)

ضریب تکانه (β) در به‌روزرسانی کدگذار کلید (key encoder) در MoCo نقش بسیار حیاتی دارد. دلیل اهمیت آن به شرح زیر است:

حفظ پایداری و سازگاری دیکشنری (صف نمونه‌های منفی):

در MoCo، کدگذار کلید، بازنمایی (representation) نمونه‌های منفی را که در یک صف (queue) ذخیره می‌شوند، تولید می‌کند. این صف به عنوان یک "دیکشنری" عمل می‌کند که کدگذار پرس‌وجو (query encoder) باید بتواند بازنمایی نمونه مثبت خود را از بازنمایی‌های موجود در این دیکشنری تمیز دهد.

برای اینکه یادگیری مقابله‌ای (contrastive learning) به طور مؤثر انجام شود، بازنمایی‌های موجود در دیکشنری باید نسبتاً پایدار و سازگار باشند. اگر کدگذار کلید خیلی سریع تغییر کند (یعنی با هر بچ (batch) از داده‌ها، وزن‌هایش به شدت به‌روز شوند)، بازنمایی نمونه‌هایی که در ابتدای صف وارد شده‌اند با بازنمایی نمونه‌هایی که اخیراً وارد شده‌اند، بسیار متفاوت خواهد بود. این ناسازگاری فرآیند یادگیری را مختل می‌کند، زیرا مدل نمی‌تواند یاد بگیرد که کدام ویژگی‌ها برای تمایز مهم هستند وقتی "معیار" مقایسه (یعنی بازنمایی‌های کلید) دائماً در حال تغییر شدید است.

ضریب تکانه بالا (مثلاً 0.999) تضمین می‌کند که کدگذار کلید به آرامی و به عنوان یک میانگین متحرک نمایی (exponential moving average) از وزن‌های کدگذار پرس‌وجو به‌روز می‌شود:

پارامترهای_کدگذار_کلید = β * پارامترهای_کدگذار_کلید + (1 - β) * پارامترهای_کدگذار_پرس‌وجو

این به‌روزرسانی آهسته باعث می‌شود که کدگذار کلید به تدریج تکامل یابد و بازنمایی‌های موجود در صف، حتی اگر از بچ‌های مختلف آمده باشند، از یک کدگذار نسبتاً مشابه و پایدار تولید شده باشند.

مشکلات انتخاب نامناسب ضریب تکانه:

اگر ضریب تکانه بیش از حد کوچک باشد (مثلاً نزدیک به 0):

کدگذار کلید تقریبا به طور کامل با وزن‌های کدگذار پرس‌وجو در هر مرحله جایگزین می‌شود (پارامترهای_کدگذار_کلید ≈ پارامترهای_کدگذار_پرس‌وجو).

این امر پایداری دیکشنری را از بین می‌برد. دیکشنری به سرعت تغییر می‌کند و مدل نمی‌تواند الگوهای معناداری را یاد بگیرد. در واقع، این حالت شبیه به روش‌های end-to-end بدون مکانیزم خاصی برای حفظ پایداری دیکشنری بزرگ می‌شود که MoCo سعی در بهبود آن دارد.

ممکن است منجر به نوسانات شدید در فرآیند یادگیری و عدم همگرایی به بازنمایی‌های خوب شود.

اگر ضریب تکانه بیش از حد بزرگ باشد (مثلاً بسیار نزدیک به 1، مانند 0.99999):

کدگذار کلید بسیار بسیار آهسته به‌روز می‌شود.

در حالی که پایداری بالاست، اگر بیش از حد آهسته باشد، ممکن است کدگذار کلید نتواند به اندازه کافی سریع با ویژگی‌های جدیدی که کدگذار پرس‌وجو در طول آموزش یاد می‌گیرد، تطبیق پیدا کند. این امر می‌تواند سرعت یادگیری کلی را کاهش دهد یا باعث شود مدل در یک نقطه بهینه محلی گیر کند، زیرا کدگذار کلید همیشه "عقب" است.

با این حال، در عمل، انتخاب یک ضریب تکانه بالا (مانند 0.99 یا 0.999) معمولاً نتایج خوبی به همراه دارد و مشکل اصلی بیشتر مربوط به کوچک بودن این ضریب است.

ب)

صف نمونه‌های منفی در الگوریتم MoCo نقش کلیدی در افزایش کارایی و اثربخشی یادگیری مقابله‌ای دارد:

امکان استفاده از تعداد زیادی نمونه منفی: یادگیری مقابله‌ای زمانی بهتر عمل می‌کند که مدل بتواند نمونه مثبت را از تعداد زیادی نمونه منفی متمایز کند. اگر فقط از نمونه‌های منفی موجود در همان بچ استفاده شود (همانطور که در برخی روش‌های اولیه بود)، تعداد نمونه‌های منفی به اندازه بچ محدود می‌شود که ممکن است برای یادگیری بازنمایی‌های قوی کافی نباشد.

صف به MoCo اجازه می‌دهد تا یک دیکشنری بزرگ و پویا از نمونه‌های منفی را حفظ کند (مثلاً با اندازه‌هایی مانند 4096، 8192 یا حتی بیشتر)، بدون اینکه نیاز به افزایش اندازه بچ باشد. اندازه بچ بزرگتر، حافظه GPU بسیار بیشتری مصرف می‌کند.

جداسازی اندازه بچ از تعداد نمونه‌های منفی:

با استفاده از صف، می‌توان اندازه بچ را نسبتاً کوچک نگه داشت (که از نظر محاسباتی بهینه است) و همزمان از مزایای داشتن تعداد زیادی نمونه منفی بهره‌مند شد.

ارائه نمونه‌های منفی سازگار (به کمک کدگذار کلید با به‌روزرسانی تکانه‌ای):

نمونه‌های موجود در صف، بازنمایی‌هایی هستند که توسط کدگذار کلید (که به آرامی به‌روز می‌شود) تولید شده‌اند. این تضمین می‌کند که حتی اگر نمونه‌ها از بچ‌های مختلف آمده باشند، بازنمایی‌های آن‌ها نسبتاً سازگار است و مدل می‌تواند روی یادگیری ویژگی‌های تمایزدهنده تمرکز کند.

مکانیزم عملکرد صف:

در هر مرحله آموزش، بازنمایی‌های (کلیدهای) نمونه‌های موجود در بچ فعلی (که به عنوان نمونه منفی برای یکدیگر عمل می‌کنند) به انتهای صف اضافه می‌شوند.

همزمان، قدیمی‌ترین بازنمایی‌ها از ابتدای صف حذف می‌شوند (مکانیزم FIFO: First-In, First-Out). این باعث می‌شود دیکشنری پویا باشد و نمونه‌های خیلی قدیمی که ممکن است دیگر نماینده خوبی از توزیع داده نباشند، حذف شوند.

به طور خلاصه، صف نمونه‌های منفی یک راهکار هوشمندانه برای داشتن یک دیکشنری بزرگ، پویا و سازگار از نمونه‌های منفی است که به MoCo کمک می‌کند بازنمایی‌های با کیفیت‌تری را با کارایی محاسباتی بالا یاد بگیرد.

ج)

تاثیر داده‌افزایی بر یادگیری بازنمایی با MoCo در مسئله تصاویر MRI با داده محدود

فرض کنید برای دسته‌بندی تصاویر MRI، از روش MoCo با تنها 2000 نمونه بدون برچسب استفاده می‌کنیم و یادگیری بازنمایی به خوبی انجام نمی‌شود. حال اگر با داده‌افزایی (data augmentation) از هر نمونه 10 نمونه جدید بسازیم و MoCo را با 22000 نمونه (2000 نمونه اصلی + 20000 نمونه افزوده شده) آموزش دهیم، تاثیرات زیر را می‌توان انتظار داشت:

بهبود کیفیت یادگیری بازنمایی:

افزایش تنوع داده‌ها: MoCo و سایر روش‌های یادگیری مقابله‌ای به شدت به داده‌افزایی متکی هستند. داده‌افزایی به مدل کمک می‌کند تا یاد بگیرد که کدام ویژگی‌ها تحت تبدیلات مختلف (مانند چرخش، برش، تغییر روشنایی، اضافه کردن نویز و غیره در تصاویر MRI) ثابت و پایدار (invariant) باقی می‌مانند. با 2000 نمونه، تنوع داده‌ها محدود است و مدل ممکن است ویژگی‌های سطحی یا بیش‌برازش (overfit) شده به این مجموعه کوچک را یاد بگیرد.

جفت‌های مثبت قوی‌تر: در MoCo، جفت‌های مثبت از دو نسخه متفاوت داده‌افزایی شده از یک تصویر یکسان ایجاد می‌شوند. با افزایش تعداد نمونه‌ها از طریق داده‌افزایی، مدل با جفت‌های مثبت متنوع‌تری مواجه می‌شود که به آن کمک می‌کند تا بازنمایی‌های کلی‌تر و معنادارتری را یاد بگیرد.

جلوگیری از فروریختن (Collapse): با داده‌های بسیار کم، خطر فروریختن مدل (یعنی یادگیری یک بازنمایی بدیهی برای همه ورودی‌ها) بیشتر است. داده‌افزایی با افزایش موثر حجم داده‌ها به کاهش این خطر کمک می‌کند.

تاثیر بر دقت نهایی مسئله دسته‌بندی:

افزایش قابل توجه دقت: اگر بازنمایی‌های یادگرفته شده توسط MoCo با کیفیت‌تر باشند، دقت مدل دسته‌بندی نهایی (که معمولاً یک طبقه‌بند خطی یا یک شبکه عصبی کوچک است که روی این بازنمایی‌های "منجمد" یا "تنظیم دقیق شده" آموزش می‌بیند) به احتمال بسیار زیاد به طور قابل توجهی افزایش خواهد یافت.

عمومیت‌پذیری بهتر: بازنمایی‌های یادگرفته شده با داده‌های بیشتر (حتی اگر از طریق داده‌افزایی باشند) معمولاً عمومیت‌پذیری بهتری به داده‌های دیده‌نشده نشان می‌دهند. این به معنای عملکرد بهتر مدل دسته‌بندی روی مجموعه تست (test set) خواهد بود.

نکته مهم در مورد داده‌افزایی MRI: تکنیک‌های داده‌افزایی باید متناسب با نوع داده (تصاویر MRI) و مسئله باشند. برای مثال، چرخش‌های بسیار شدید یا تغییرات رنگی که در دنیای واقعی MRI رخ نمی‌دهند، ممکن است مفید نباشند یا حتی مضر باشند. اما تکنیک‌هایی مانند تغییرات جزئی در شدت، کنتراست، برش‌های تصادفی، چرخش‌های خفیف، وارون‌سازی افقی (اگر معنی‌دار باشد) و دگرشکلی‌های الاستیک (elastic deformations) می‌توانند بسیار موثر باشند.

برای رفع برخی ایرادات و ابهامات از AI استفاده شده است.