به نام خدا

امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

پاسخ سوالات تئوری تمرین سری ۶

۱)

استفاده از وزن های اولیه pretrained به این صورت است که چون از قبل با تسک های مختلف آموزش دیده؛ دانش پایه ای در رابطه با تسکی که الان به آن محول کردیم دارد و نیاز ندارد همه چیز را به تازگی آموزش ببیند در نتیجه میتواند همگرایی سریعتر و داده کمتری را طلب کند و در کل برای تسک هایی که قبلا آموزش دیده عملکرد خوب و قابل توجهی داشته باشد.
البته همین مورد میتواند معایبی را نیز به همراه داشته باشد؛ مثلا انعطاف پذیری مدل کمتر میشود چون یک دانش پایه ای را از قبل دارد و چون از وزن های قبل استفاده میکند ممکن است در برابر تغییرات و وفق دادن خود با داده های جدید برای تسک جدید کمی مقاومت کند.یا اینکه ممکن است وزن های به دست آمده برای تسک های مختلفی باشند و تسک فعلی ما متفاوت باشد که آن وزن ها بیشتر میتوانند گمراه کننده باشند.یا اینکه اگر مشکل overfitting در آموزش قبلی وجود داشته باشد ممکن است به این تسک نیز انتقال یابد.

استفاده از وزن های اولیه تصادفی انعطاف بیشتری دارند و مشکل وفق پیدا کردن با تسک جدید را مانند حالت قبل ندارند.اما چون فرآیند آموزش از ابتدا در حال انجام است و دانش پایه دیگر وجود ندارد منطقا زمان بیشتری سپری خواهد شد و به داده های بیشتری نیاز داریم.

۲) پدیده catastrophic forgetting زمانی ایجاد میشود که به طور مثال ما مدل را برای تسک A آموزش داده ایم و وزن ها طوری آپدیت و نهایی شده اند که عملکرد مدل در تسک A بهینه شود و وقتی به تسک بعدی مثلا تسک B میرویم و میخواهیم مدل را برای این تسک

آموزش دهیم باید وزن ها را طوری آپدیت کنیم که برای این تسک عملکرد مناسبی(یعنی تابع ضرر ما مینیمم شود) داشته باشد و وزن هایی که در تسک A داشتیم فراموش میشود و وقتی به سراغ تسک های بعدی میرویم نیز این اتفاق برای وزن های یافت شده در تسک قبلی نیز میفتد.و در کل وقتی به صورت sequential میخواهیم تسک های مختلف را آموزش دهیم این اتفاق میفتد و به صورت کلی ماتریس وزن هایی که داشتیم در هر مرحله مقادیر موجود در ماتریس وزنهای حاصل از مرحله قبل را حذف میکند.

برای مقابله با این پدیده روش هایی مانند Regularization Methods, Dynamic Architectures, Rehearsal Methods, Dual Memory Systems, Multi-Task Learning, Optimization معرفی شده اند که شرح کامل هر کدام در لینک مربوط به سایت medium که در منابع گذاشته ام موجود هست.اما برخی را به صورت خلاصه شرح میدهم.

Dual memory systems : برای این است که یک حافظه ای را به صورت جداگانه برای وزن های قبلی در نظر بگیریم.

Multi task learning : به این صورت است که مدل به جای اینکه به صورت sequential آموزش ببیند به صورت همزمان تسک های مختلف را آموزش ببیند و برای هر تسک وزن ها را به صورت جداگانه به دست آورند.

Regularization : برای این که تغییرات وزن را محدود تر کنیم تا وزن های به دست آمده در تسک های قبلی کاملا از بین نروند و فاصله معناداری بین وزن های جدید و وزن های فعالی وجود نداشته باشد.

Dynamic architecture : استفاده کردن از معماری داینامیک برای هر شبکه مطابق با هر تسک.

۳) Transfer Learning به این صورت است که دانشی که از یک تسک به دست آمده را برای افزایش performance در یک تسک مرتبط از آن استفاده میکنیم.

تفاوت Transfer learning , fine-tuning در این است که در transfer learning ما با تسک های مشابه سر و کار و مرتبط سر و کار داریم مناسب هستند اما در fine-tuning میتوانیم در تسک های متفاوت نیز از مدل استفاده کنیم.
به صورت کلی در transfer learning دیتاست جدید باید کوچکتر از حالت قبلش باشد اما در fine-tuning با توجه به اینکه تسک میتواند متفاوت نیز باشد دیتاست جدید باید بزرگ باشد و بهتر است شبیه به دیتاستی باشد که مدل طبق آن pretrained باشد.
با توجه به مورد بالا هزینه محاسباتی و زمان آموزش کمتری در transfer learning داریم.
همچنین در fine-tuning بیشتر به دنبال این هستیم که مدل را با تغییراتی که میدهیم بهبود ببخشیم.

به صورت کلی در شرایطی از transfer learning استفاده میکنیم که دیتاست کوچک باشد و تسک مشابه و مرتبط با تسک قبلی باشد و از fine-tuning زمانی استفاده میکنیم که دیتای خوب و کافی داشته باشیم و تسک جدید نیز اگر نامرتبط بود میتوانیم استفاده کنیم.

Use Case	When task-specific data is available and computational resources allow full retraining	When limited labeled data or computational resources are available, and tasks share similarities

۴) روش masking ، در این روش، تعدادی از توکن‌های جمله به طور تصادفی انتخاب و جایگزین می‌شوند، مدل سپس وظیفه دارد که توکن‌های پنهان را با توجه به متن اطراف پیش‌بینی کند .این فرآیند به مدل کمک می‌کند تا روابط آماری بین کلمات را در یک جمله یاد بگیرد و درک عمیق‌تری از زبان پیدا کند.

روش های مختلف masking :

- **Masking تصادفی :** در این روش، توکن‌ها به طور تصادفی با احتمال ثابتی انتخاب و جایگزین می‌شوند.

- **مبتنی بر :Speech of part** در این روش، توکن‌ها با توجه به نقش نحوی یا معنایی آنها در جمله انتخاب و جایگزین می‌شوند .به عنوان مثال، اسم‌ها با احتمال بیشتری نسبت به حروف اضافه جایگزین می‌شوند.

تعداد توکن‌های قابل Mask در جمله می‌تواند بر عملکرد مدل تأثیر بگذارد .اگر تعداد توکن‌های قابل Mask خیلی کم باشد، مدل ممکن است به اندازه کافی چالش برانگیز نشود و به درستی یاد نگیرد .اگر تعداد توکن‌های قابل Mask خیلی زیاد باشد، مدل ممکن است گیج شود و عملکرد ضعیفی داشته باشد.

تعداد توکن‌های قابل Mask در جمله می‌تواند بر عملکرد مدل تأثیر بگذارد .اگر تعداد توکن‌های قابل Mask خیلی کم باشد، مدل ممکن است به اندازه کافی چالش برانگیز نشود و به درستی یاد نگیرد .اگر تعداد توکن‌های قابل Mask خیلی زیاد باشد، مدل ممکن است گیج شود و عملکرد ضعیفی داشته باشد.

به طور کلی عملیات masking میتواند سرعت آموزش و عملکرد مدل را بهبود ببخشند.
برای استفاده از روش رندوم اگر حجم داده کم باشد مناسب است و اگر توان محاسباتی خوبی داریم میتوانیم از روش pos استفاده کنیم.
به طور کلی اگر این دو روش را بخواهیم مقایسه کنیم به این صورت است که روش رندوم به صورت تصادفی یکسری از توکن ها را مَسک میکند و چون رندوم است و نیاز به پیاده سازی یا یادگیری خاصی ندارد سادگی و سرعت بهتری دارد اما چون تصادفی است منطقا نسبت به روش های هوشمند تر مثل pos احتمال زیاد دارای کیفیت خوبی نیست یا به کیفیت آن نیست.
در روش pos توکن‌ها با توجه به نقش نحوی یا معنایی آنها در جمله انتخاب و جایگزین می‌شوند .به عنوان مثال، اسم‌ها با احتمال بیشتری نسبت به حروف اضافه جایگزین می‌شوند . این روش می‌تواند به مدل کمک کند تا بر روی قسمت‌های مهم‌تر جمله تمرکز کند و درک عمیق‌تری از ساختار نحوی و معنایی جمله پیدا کند. در این روش کیفیت افزایش پیدا میکند اما پیچیدگی بیشتر است و زمان بیشتری میطلبد.

Maskingمبتنی بر part of speech می‌تواند به طور قابل توجهی دقت مدل را در وظایف مختلف NLP مانند ترجمه ماشینی، خلاصه‌سازی متن و پرسش و پاسخ بهبود بخشد.

این روش به مدل کمک می‌کند تا بر روی قسمت‌های مهم‌تر جمله تمرکز کند و درک عمیق‌تری از ساختار نحوی و معنایی جمله پیدا کند.

اگر سرعت آموزش مدل مهمتر از دقت آن است روش تصادفی مناسب تر است.

۵)

عملکرد معماری :

- **Seq2seq** : این معماری از دو شبکه عصبی تشکیل شده است :یک **رمزگذار** (Encoder)و یک **رمزگشا** .(Decoder)رمزگذار توالی ورودی را به یک نمایش برداری تبدیل می‌کند و رمزگشا آن نمایش را به توالی خروجی مورد نظر تبدیل می‌کند.
- **MLM** : در این معماری، تعدادی از توکن‌های جمله به طور تصادفی انتخاب و جایگزین می‌شوند، مدل سپس وظیفه دارد که توکن‌های پنهان را با توجه به متن اطراف پیش‌بینی کند.
- **CLM** : در این معماری، دو نسخه از جمله به مدل ارائه می‌شود و از مدل خواسته می‌شود که تشخیص دهد کدام نسخه از جمله صحیح‌تر است.

مزایا :

- **Seq2seq** :

انعطاف‌پذیری :به طور کلی برای طیف گسترده‌ای از وظایف NLP مانند ترجمه ماشینی، خلاصه‌سازی متن و پرسش و پاسخ قابل استفاده است.

کنترل دقیق :به مدل اجازه می‌دهد تا بر روی قسمت‌های خاص توالی ورودی تمرکز کند و اطلاعات بیشتری را از آنها استخراج کند.

درک متنی :قادر به درک زمینه‌های طولانی مدت و پیچیده در دنباله‌های ورودی و خروجی است.

- **MLM** :

سادگی :آموزش و پیاده‌سازی آن به دلیل وجود یک شبکه عصبی ساده‌تر، آسان‌تر است.

نیاز به داده :به حجم کمتری از داده‌های خام نیاز دارد.

درک متنی قوی :به دلیل استفاده از ماسک‌های دوجهته، مدل می‌تواند زمینه‌های گسترده‌ای را درک کند.

قابلیت تعمیم :می‌تواند به طور بالقوه برای وظایف مختلف NLP مانند درک مطلب و خلاصه‌سازی متن تعمیم داده شود.

- **CLM** :

دقت :می‌تواند به دقت بالایی در وظایف مختلف NLP مانند درک مطلب و پاسخ به سوال دست یابد.

قابلیت تعمیم :می‌تواند به طور بالقوه برای وظایف مختلف NLP تعمیم داده شود.

تولید متن طبیعی :به دلیل پیش‌بینی خودبازگشتی، می‌تواند متن‌های بسیار طبیعی و روان تولید کند.

معایب :

- **Seq2seq** :

پیچیدگی :آموزش و پیاده‌سازی آن به دلیل وجود دو شبکه عصبی مجزا، پیچیده‌تر است.

نیاز به داده :به حجم قابل توجهی از داده‌های جفت‌شده ورودی و خروجی نیاز دارد.

- **MLM** :

کنترل محدود :به مدل اجازه می‌دهد تا بر روی تمام توالی ورودی به طور یکسان تمرکز کند و اطلاعات خاصی را از آن استخراج کند.
کیفیت :ممکن است به اندازه Seq2Seq دقت بالایی نداشته باشد.

- **CLM** :

پیچیدگی :آموزش و پیاده‌سازی آن به دلیل نیاز به دو نسخه از جمله، پیچیده‌تر است.
نیاز به داده :به حجم قابل توجهی از داده‌های جفت‌شده نیاز دارد.

مثال ها :

MLM = Bert,RoBerta
CLM = GPT-2 , GPT-3
Seq2Seq = wu-dao 2.0,Transformers

۶)

در روش اول ، ابتدا مدل توکن‌های قبلی در دنباله را به عنوان ورودی دریافت می‌کند. سپس مدل احتمال هر توکن ممکن را در موقعیت بعدی پیش‌بینی می‌کند. بر اساس این احتمالات، مدل مجموعه‌ای از توکن‌های بعدی با بالاترین احتمال را انتخاب می‌کند. سپس مدل برای هر توکن در این مجموعه، توکن بعدی را پیش‌بینی می‌کند و این فرآیند را تکرار می‌کند. در نهایت، دنباله‌ای از متن تولید می‌شود که از توکن‌هایی تشکیل شده است که بالاترین احتمال را در هر مرحله داشته‌اند.

روش دیگر این است توکن های بعدی به صورت رندوم و تصادفی انتخاب شوند. در این روش، مدل به طور تکراری توکن‌های بعدی را در دنباله پیش‌بینی می‌کند. برای انجام این کار، ابتدا مدل توکن‌های قبلی در دنباله را به عنوان ورودی دریافت می‌کند. سپس مدل احتمال هر توکن ممکن را در موقعیت بعدی پیش‌بینی می‌کند. بر اساس این احتمالات، مدل توکن بعدی را به طور تصادفی نمونه‌گیری می‌کند. این فرآیند به طور تکراری تکرار می‌شود تا دنباله متن مورد نظر تولید شود.