به نام خالق رنگین کمان

ستاره باباجانی – گزارش تمرین سری 6

سوال 1: هنگام آموزش مدلی مانندBERT ، انتخاب وزنه های اولیه می تواند به طور قابل توجهی بر روند تمرین و عملکرد نهایی مدل تأثیر بگذارد. حال به بررسی هرکدام از سناریوها میپردازیم:

1. وزن های از پیش آموزش دیده از مدلBERT:

- فرآیند آموزشی:

  - راه اندازی اولیه: مدل با وزن هایی شروع می شود که قبلاً روی یک مجموعه داده بزرگ برای وظایف مختلف تنظیم شده اند. این بدان معناست که مدل از قبل ساختارهای زبان، عبارات رایج و روابط معنایی را درک کرده است.
  - همگرایی: این مدل احتمالاً سریعتر همگرا می شود زیرا از یک حالت کاملاً آگاه شروع می شود. Fine-tuning یک کار خاص در مقایسه با شروع از صفر(scratch) به زمان کمتر و epochs کمتری نیاز دارد.
  - بهینه سازی: تابع گرادیان و loss احتمالاً روان تر عمل می کند و خطر مواجهه با vanishing یا exploding gradients را کاهش می دهد.

- عملکرد بعد از آموزش:
  - دقت و تعمیم: انتظار می رود که مدل در target task بهتر عمل کند، به خصوص اگر کار مشابه کارهایی باشد که مدل از قبل آموزش داده شده است. درک مدل از زبان از پیش‌آموزش به تعمیم بهتر آن کمک می‌کند.
  - Fine-Tuning: افزایش عملکرد در درجه اول به دلیل انتقال دانش از مدل از پیش آموزش دیده به کار جدید است. این یادگیری انتقال اغلب منجر به دقت بالاتر و عملکرد قوی تر می شود.

2. مقدار اولیه وزن تصادفی:

- فرآیند آموزشی:
  - راه اندازی: مدل با وزن های تصادفی شروع می شود، به این معنی که هیچ دانش قبلی از ساختارهای زبان یا معناشناسی ندارد.
  - همگرایی: آموزش از صفر معمولاً بیشتر طول می کشد. این مدل باید الگوهای زبان و روابط را از پایه بیاموزد که به epochs و منابع محاسباتی بیشتری نیاز دارد.
  - بهینه سازی: فرآیند آموزش می تواند ناپایدارتر باشد، با خطر بیشتری برای مواجهه با مسائلی مانند vanishing یا انفجار

گرادیان ها. تنظیم دقیق فراپارامترها (مانند نرخ یادگیری) ضروری است.

- عملکرد بعد از آموزش:
  - دقت و تعمیم: عملکرد نهایی به طور کلی در مقایسه با استفاده از وزن های از پیش آموزش دیده کمتر است. این مدل ممکن است به خوبی تعمیم نیابد، به خصوص اگر مجموعه داده آموزشی به اندازه کافی بزرگ نباشد که تنوع زبان را پوشش دهد.
  - دانش: این مدل فاقد مزیت دانش انتقال یافته از پیش آموزش گسترده است، که توانایی آن را برای اجرای خوب وظایف زبانی پیچیده محدود می کند.

سوال 2: Catastrophic Forgetting یک چالش مهم در آموزش شبکه های عصبی است، به ویژه هنگامی که وظایف را به صورت متوالی یاد می گیریم. زمانی اتفاق می‌افتد که یک مدل، در حین یادگیری یک کار جدید، به سرعت و به شدت کارهایی را که قبلاً آموخته‌اند فراموش می‌کند. این به این دلیل اتفاق می‌افتد که وزن‌هایی که برای کارهای قبلی بهینه شده بودند، در طول آموزش کارهای جدید بازنویسی می‌شوند و منجر به از دست دادن عملکرد در کارهای قدیمی‌تر می‌شوند.

هنگامی که یک شبکه عصبی بر روی چندین کار یکی پس از دیگری آموزش می بیند، بدون هیچ مکانیزمی برای حفظ اطلاعات مربوط به کارهای قبلی، با

یادگیری وظایف جدید، تمایل دارد وظایف قبلی را فراموش کند. این به این دلیل است که وزن های یکسان برای همه کارها استفاده می شود و به روز رسانی آنها برای یک کار جدید می تواند دانش آموخته شده قبلی را مختل کند.

تکنیک کاهش: تثبیت وزن الاستیک(EWC): یکی از روش های موثر برای کاهش Catastrophic Forgetting، تثبیت وزن الاستیک است. EWC به مدل کمک می کند تا وزن های مهم وظایف قبلی را با افزودن یک عبارت منظم سازی به تابع ضرر حفظ کند. این اصطلاح تغییرات قابل توجهی را در وزن‌هایی که برای کارهای قدیمی‌تر حیاتی هستند جریمه می‌کند. نحوه کار آن به شرح زیر است:

1. شناسایی وزن های مهم: EWC اهمیت هر وزن را با استفاده از ماتریس اطلاعات Fisher محاسبه می کند که میزان حساسیت خروجی شبکه به تغییرات هر وزن را اندازه می گیرد.
2. Regularization Term: هنگام یادگیری یک کار جدید، EWC یک عبارت منظم سازی را به تابع ضرر اضافه می کند که تغییرات وزن های مهم را جریمه می کند، بنابراین دانش وظایف قبلی را حفظ می کند.

با استفاده ازEWC ، مدل می‌تواند وظایف جدید را بدون تأثیر قابل‌توجهی بر عملکرد وظایف قبلی بیاموزد و به طور موثر Catastrophic Forgetting را کاهش دهد.

منابع:

- Overcoming Catastrophic Forgetting in Neural Networks
- https://en.wikipedia.org/wiki/Catastrophic_interference
- https://neurosciencenews.com/ai-continuous-learning-23671/

سوال 3:

- Transfer Learning: یادگیری انتقالی یک تکنیک یادگیری ماشینی است که در آن یک مدل توسعه یافته برای یک کار خاص به عنوان نقطه شروع برای یک مدل در یک کار دوم مجددا استفاده می شود. این رویکرد از دانش به دست آمده از اولین کار برای بهبود عملکرد در کار جدید استفاده می کند. یادگیری انتقال به ویژه زمانی مفید است که وظیفه جدید داده های محدودی داشته باشد. روند آن به شرح زیر است:
  - استخراج ویژگی: از یک مدل از پیش آموزش دیده (مثلاً یک مدل آموزش دیده درImageNet) به عنوان استخراج کننده ویژگی ثابت استفاده میکنیم. وزن های مدل از پیش آموزش دیده فریز شده و تنها لایه های نهایی مخصوص کار جدید آموزش داده می شوند.

- آموزش: فقط لایه های جدید را در مجموعه داده جدید آموزش میدهیم در حالی که لایه های مدل از قبل آموزش داده شده را ثابت نگه میداریم.

Transfer Learning زمانی که وظیفه جدید مجموعه داده کوچکی دارد و یا هنگامی که کار جدید شبیه به کاری است که مدل در ابتدا روی آن آموزش داده شده است (به عنوان مثال، هر دو کار شامل طبقه بندی تصویر هستند)، استفاده میشود. مزایای آن به شرح زیر است:

- به منابع محاسباتی و زمان آموزش کمتری نیاز داریم.
- برای کارهایی با داده های محدود مفید است.

- Fine-Tuning: گسترش یادگیری انتقالی است. در تنظیم دقیق، پس از انجام استخراج ویژگی، مدل با بازکردن برخی یا همه لایه‌های مدل از پیش آموزش‌دیده شده، بیشتر آموزش داده می‌شود. این به مدل اجازه می دهد تا به طور خاص با کار جدید سازگار شود. روند آن به شرح زیر است:

- استخراج ویژگی: مشابه یادگیری انتقالی، با یک مدل از پیش آموزش دیده شروع کرده و لایه های نهایی را با لایه های جدید مخصوص کار جدید جایگزین میکنیم.
- Unfreeze کردن لایه ها: برخی یا همه لایه های از پیش آموزش دیده را از حالت انجماد خارج میکنیم.

- آموزش: هم لایه های جدید و هم لایه های از پیش آموزش دیده فریز نشده را روی مجموعه داده جدید آموزش میدهیم.

Fine-Tuning زمانی که مجموعه داده جدید به اندازه کافی بزرگ باشد که از به روز رسانی لایه های از پیش آموزش دیده بهره مند شود یا زمانی که کار جدید ارتباط نزدیکی با کار اصلی دارد اما همچنان به ویژگی های خاص اضافی نیاز دارد، استفاده میشود. مزایای آن به شرح زیر است:

- می تواند با تنظیم دقیق مدل برای گرفتن ویژگی های خاص تر، عملکرد بهتری در کار جدید داشته باشد.
- به مدل اجازه می دهد تا دانش عمومی را در حین تطبیق با ظرایف جدید و خاص کار حفظ کند.

سوال 4: MLMها مانند BERT با پوشاندن درصد معینی از نشانه‌های ورودی و سپس پیش‌بینی نشانه‌های ماسک‌شده آموزش می‌بینند. این فرآیند به مدل اجازه می‌دهد تا بازنمایی‌های bidirectional context را بیاموزد، که برای درک معنای کلمات در متن بسیار مهم هستند.

Masking Methods:

1. Random Masking: شامل انتخاب تصادفی زیرمجموعه ای از نشانه ها در توالی ورودی برای ماسک کردن است. این روش تضمین می کند که همه توکن ها شانس مساوی برای ماسک شدن دارند و سیگنال آموزشی متنوعی ارائه می دهد. همچنین به مدل کمک می کند تا

بازنمایی هایی را برای طیف گسترده ای از کلمات و زمینه ها بیاموزد. اما اگر توکن های حیاتی (مثلاً آنهایی که کمیاب هستند یا آموزنده هستند) خیلی مکرر یا اصلاً ماسک نشوند، گاهی اوقات ماسک تصادفی می تواند منجر به بی ثباتی شود. موارد استفاده آن به شرح زیر است:

- به طور کلی برای ایجاد مدل های زبان قوی که در انواع وظایف downstream به خوبی انجام می شود، موثر است.
- با این حال، ممکن است به اندازه کافی بر ساختارهای نحوی یا معنایی تمرکز نکند، و به طور بالقوه عملکرد را در وظایفی که نیاز به درک زبانی عمیق تری دارند، محدود می کند.

2. Part of Speech Masking: این روش شامل ماسک کردن نشانه ها بر اساس بخشی از گفتار آنها، مانند اسم، افعال، یا صفت است. که با تمرکز بر بخش های خاص گفتار، مدل می تواند در مورد روابط و نقش این کلمات در جملات اطلاعات بیشتری کسب کند. که درک مدل از نحو و دستور زبان را افزایش می دهد، زیرا می تواند الگوهای مربوط به انواع خاصی از کلمات را یاد بگیرد. موارد استفاده آن به شرح زیر است:

- می تواند منجر به عملکرد بهتر در کارهایی شود که نیاز به درک نحوی دارند، مانند تجزیه یا شناسایی موجودیت نامگذاری شده.

- ممکن است در وظایفی که نیاز به coverage واژگانی گسترده دارند، عملکرد ضعیفی داشته باشد، زیرا به جای کل واژگان، بر انواع کلمات خاص تمرکز دارد.

تعیین مقدار Maskable Tokens:

1. رویکرد استاندارد (15٪ پوشش): معمولاً 15 درصد از توکن‌ها در توالی ورودی در طول تمرین ماسک می‌شوند. این تعادل به مدل اجازه می‌دهد تا زمینه کافی را بیاموزد در حالی که هنوز نشانه‌های قابل مشاهده کافی برای درک ساختار جمله دارد.
2. نسبت ماسک کردن بالاتر یا پایین تر:

- نسبت های بالاتر از 15٪: پوشش تهاجمی تر می تواند مدل را وادار کند که بیشتر به context تکیه کند و به طور بالقوه درک زمینه ای آن را بهبود بخشد. که می تواند توانایی مدل را برای پر کردن اطلاعات از دست رفته افزایش دهد، اما در صورت پراکندگی زمینه ممکن است عملکرد کلی را کاهش دهد.
- نسبت های پایین تر از 15٪: پوشش تهاجمی کمتر به این معنی است که مدل context بیشتری برای کار دارد و به طور بالقوه ثبات و کارایی یادگیری را بهبود می بخشد. که ممکن است عملکرد را در کارهایی که نیاز به دقت بالایی در توکن‌های

unmasked دارند، بهبود بخشد، اما می‌تواند توانایی مدل را برای مدیریت اطلاعات از دست رفته محدود کند.

سوال 5:

1. Sequence-to-Sequence: مدل‌های Seq2Seq که اغلب از معماری‌هایی مانند LSTM یا Transformer استفاده می‌کنند، برای مدیریت توالی‌های ورودی و تولید توالی‌های خروجی طراحی شده‌اند. آنها به طور گسترده در کارهایی مانند ترجمه ماشینی، خلاصه سازی متن و برنامه های چت بات استفاده می شوند. مزایای آنها به شرح زیر است:

- تطبیق پذیری: می توانند وظایف مختلف مربوط به تبدیل دنباله را انجام دهند.
- درک Context: قادر به گرفتن وابستگی های دوربرد و Context در دنباله ها.
- پردازش موازی: با مدل های مبتنی برTransformerها ، Seq2Seq می توانند از پردازش موازی، سرعت بخشیدن به آموزش و استنتاج استفاده کنند.

و معایب آنها به شرح زیر است:

- پیچیدگی: می تواند از نظر محاسباتی گران باشند، به خصوص برای دنباله های طولانی.

- نیازمندی به داده: به مقادیر زیادی داده موازی برای آموزش موثر نیاز دارند.
- Exposure Bias: مدل ممکن است از Exposure Bias رنج ببرد، جایی که در طول آموزش عملکرد خوبی دارد اما در طول استنتاج به دلیل تفاوت در نحوه تولید دنباله ها ضعیف است.

مثال آنها نیز به شرح زیر است:

- Google Translate: از مدل Seq2Seq برای ترجمه متن از یک زبان به زبان دیگر استفاده می کند.

2. Masked Language Models: MLMها، مانندBERT ، با پوشاندن برخی از نشانه‌ها در توالی ورودی و پیش‌بینی این نشانه‌های پوشانده شده بر اساس زمینه اطراف آموزش می‌بینند. این رویکرد پیش‌آموزشی مدل را قادر می‌سازد تا bidirectional context را درک کند. مزایای آن به شرح زیر است:

- Bidirectional Context: می تواند context را از هر دو جهت اعمال کند، و برای کارهایی که نیاز به درک عمیق متن دارند (مانند پاسخ به سؤال، شناسایی موجودیت نامگذاری شده) بسیار مؤثر است.

- Transfer Learning: MLMهای از پیش آموزش دیده را می توان در کارهای مختلف downstream تنظیم کرد و نیاز به داده های خاص کار را کاهش داد.

و معایب آن به شرح زیر است:

- هزینه آموزش: آموزشMLM ها از نظر محاسباتی گران و وقت گیر است.
- سرعت استنتاج: در طول استنتاج، این مدل ها به دلیل پیچیدگی می توانند در مقایسه با مدل های یک طرفه کندتر باشند.

مثال آن نیز به شرح زیر است:

- BERT: برای کارهای مختلف NLP مانند تجزیه و تحلیل احساسات، طبقه بندی متن و پاسخ به سؤال استفاده می شود.

3. Causal Language Models: CLMها، مانندGPT ، با پیش بینی توکن بعدی در یک توالی بر اساس توکن های قبلی، متن تولید می کنند. آنها یک طرفه هستند، به این معنی که آنها فقط context سمت چپ (نشان های گذشته) را برای پیش بینی در نظر می گیرند. مزایای آنها به شرح زیر است:

- قابلیت تولید: در تولید متن منسجم و مرتبط با متن عالی هستند، و آنها را برای کارهایی مانند تولید متن، سیستم های گفتگو و تکمیل خودکار ایده آل می کنند.
- کارایی: معمولاً در طول استنتاج سریعتر هستند زیرا آنها توکن ها را به ترتیب از چپ به راست پیش بینی می کنند.

و معایب آنها به شرح زیر است:

- Context محدود: فقط می توانند از context گذشته استفاده کنند، که ممکن است درک را در مقایسه با مدل های دو طرفه محدود کند.
- خطر Overfitting: بدون تنظیم دقیق، CLMها می توانند بر روی داده های آموزشی بیش از حد قرار بگیرند و متن تکراری یا بی معنی تولید کنند.

مثال آنها نیز به شرح زیر است:

- GPT-3: برای طیف گسترده ای از برنامه ها از جمله چت بات ها، ایجاد محتوا و تولید کد استفاده می شود.

Model Type	Advantages	Disadvantages	Example
Seq2Seq	Versatile, good at context understanding, parallel processing with Transformers	Computationally expensive, requires large datasets, exposure bias	Google Translate
MLM	Utilizes bidirectional context, effective transfer learning	High training cost, slower inference	BERT
CLM	Excellent generative capabilities, efficient inference	Limited to past context, risk of overfitting	GPT-3

سوال 6: مدل‌های زبان ماسک‌شده (MLM) مانند BERT اساساً برای درک و تکمیل متن با پیش‌بینی نشانه‌های ماسک‌دار طراحی شده‌اند. با این حال، تولید دنباله‌ای از متن مستقیماً با MLM به تکنیک‌های بیشتری نیاز دارد، زیرا MLM ها ذاتاً مدل‌های مولد مانند مدل‌های زبان علّی (CLM) نیستند. حال به بررسی چند استراتژی برای استفاده از MLMها برای تولید متن میپردازیم:

1. Iterative Masking and Prediction: یکی از رویکردها، ماسک کردن و پیش‌بینی مکرر tokens برای تولید متن است. که مراحل آن به شرح زیر است:
   - Initialization: با یک دنباله اولیه حاوی نشانه‌های خاصی مانند «[CLS]» و «[SEP]» شروع کنیم.
   - ماسک کردن: به طور تصادفی زیر مجموعه ای از tokens که باید تولید شوند را ماسک میکنیم.

- پیش بینی: از MLM برای پیش بینی نشانه های ماسک شده استفاده میکنیم.
- به روز رسانی: توکن های ماسک شده را با tokens پیش بینی شده جایگزین میکنیم.
- تکرار: فرآیند ماسک کردن و پیش بینی را تا زمانی که کل دنباله تولید شود تکرار میکنیم.

با این حال، این رویکرد می‌تواند محاسباتی فشرده باشد و ممکن است به دلیل فقدان بازخورد autoregressive در طول تولید، همیشه متن منسجمی تولید نکند.

2. Fusion with Autoregressive Models: ترکیب MLM با مدل های autoregressive می تواند از نقاط قوت هر دو معماری استفاده کند. طرح کلی آن به شرح زیر است:

- درک Context: از MLM برای درک و پیش بینی context ماسک شده در یک زمینه خاص استفاده میکنیم.
- تولید توالی: از یک مدل autoregressive مانند GPT برای تولید متن به صورت متوالی استفاده میکنیم و از انسجام و ارتباط متنی اطمینان حاصل میکنیم.

این ادغام امکان تولید دنباله‌هایی را فراهم می‌کند که در آن MLM درک متنی را تضمین می‌کند، در حالی که مدل autoregressive تولید روان متن را تضمین می‌کند.

3. استفاده از مدل زبان ماسک شده برای Inpainting: Text inpainting شامل تولید متن برای پر کردن قسمت‌های از دست رفته در یک context خاص است. این می تواند برای برنامه هایی مانند تکمیل جملات یا پاراگراف ها مفید باشد:
   - تدارک Context : زمینه MLM را در اطراف قسمت گم شده فراهم کنیم.
   - پیش بینی ماسک: قسمتی که باید تولید شود را ماسک میکنیم و از MLM برای پیش بینی و پر کردن متن از دست رفته استفاده میکنیم.
4. تکنیک های Constrained Generation: MLMها همچنین می توانند در سناریوهای تولید متن محدود استفاده شوند که در آن خروجی باید شرایط یا محدودیت های خاصی را برآورده کند:
   - تولید مبتنی بر الگو: از قالب هایی با قسمت های ثابت و متغیر استفاده میشود. بخش های ثابت ساختار را ارائه می دهند و MLM قسمت های متغیر را پیش بینی می کند.

- تولید متن Conditional: شرایط یا اعلان های خاصی را ارائه داده و از MLM برای تولید متن متناسب با آن شرایط استفاده میکنیم.

سوال 7: بررسی سوالات پرسیده شده در فایل کد:

- Understanding the Masking Strategy in Masked Language Models: استراتژی masking در MLMها مانند BERT شامل سه جزء است:
  1. ٪80 از توکن های ماسک شده با توکن "[MASK]" جایگزین می شوند.
  2. ٪10 با کلمات تصادفی جایگزین می شود.
  3. 10% بدون تغییر باقی میماند.

این رویکرد روشمند تضمین میکند که مدل به طور موثر از context یاد می گیرد و در برابر تغییرات قوی می شود. حال به منطق پشت هر جزء از این استراتژی میپردازیم:

1. 80% با توکن «[MASK]» ماسک شده است:
   - هدف اصلی MLM ها پیش بینی توکن های ماسک شده بر اساس context آنهاست. با جایگزینی 80 درصد از توکن های ماسک شده با نشانه «[MASK]»، به مدل به صراحت گفته می شود که

کدام توکن ها را باید پیش بینی کند. این دستورالعمل واضح به مدل کمک می کند تا بر درک و استفاده از زمینه های اطراف برای پیش بینی های دقیق تمرکز کند.

- این درصد بالا تضمین می کند که مدل به طور مکرر با نشانه "[MASK]" در طول آموزش مواجه شود، بنابراین یاد می گیرد که از سرنخ های متنی از نشانه های اطراف برای استنتاج اطلاعات گم شده استفاده کند. این فرآیند توانایی مدل را برای درک زمینه و روابط بین کلمات افزایش می دهد.

با جایگزینی 80٪ از نشانه های ماسک شده با "[MASK]"، مدل یاد می گیرد که به context ارائه شده توسط نشانه های دیگر توجه دقیق داشته باشد. این توانایی آن را برای پرکردن دقیق شکاف ها تیز می کند.

2. ٪10 با کلمات تصادفی جایگزین شده است:

- جایگزینی 10٪ از نشانه های ماسک شده با کلمات تصادفی باعث ایجاد نویز و تغییرپذیری در داده های آموزشی می شود. این استراتژی از تطبیق بیش از حد مدل با نشانه خاص «[MASK]» جلوگیری می‌کند و

به آن کمک می‌کند تا ورودی‌های غیرمنتظره یا جدید را در طول استنتاج قوی‌تر مدیریت کند.

- با دیدن گهگاه کلمات تصادفی به جای نشانه های ماسک شده، مدل یاد می گیرد که با نویز و عدم قطعیت مقابله کند، که در برنامه های کاربردی دنیای واقعی که داده های ورودی ممکن است نویز یا ناقص باشد رایج هستند.

این مولفه مدل را در برابر تغییرات و ورودی های غیرمنتظره انعطاف پذیرتر می کند. با تکیه نکردن صرفاً به وجود نشانه «[MASK]» تعمیم بهتری را می آموزد.(Enhanced Robustness)

معرفی کلمات تصادفی تضمین می کند که مدل در پیش بینی نشانه های ماسک شده به روشی بسیار قابل پیش بینی خیلی تخصصی نمی شود و در نتیجه استحکام کلی آن را افزایش می دهد.(Preventing Overfitting)

3. 10% بدون تغییر باقی میماند:

- بدون تغییر باقی گذاشتن 10 درصد از نشانه‌های ماسک‌شده به جلوگیری از وابستگی بیش از حد مدل به نشانه «[MASK]» برای پیش‌بینی کمک

می‌کند. این مؤلفه تضمین می کند که مدل نیز از کلمات واقعی در دنباله یاد می گیرد.

- با مشاهده گاه توکن های اصلی در جای خود، مدل یاد می گیرد که لازم نیست همه نشانه ها به صورت ماسک شده پیش بینی شوند، که به تعمیم بهتر کمک می کند. این به مدل کمک می‌کند بفهمد که همه نشانه‌ها در یک دنباله در معرض تغییر نیستند، که برای کارهایی که شامل دنباله‌هایی هستند که در آن تغییرات حداقل هستند، بسیار مهم است.

این جنبه از استراتژی، تعادلی را بین یادگیری پیش بینی توکن های ماسک دار و درک توالی واقعی نشانه ها فراهم می کند. این به مدل کمک می‌کند تا دید واقعی از داده‌های متنی را حفظ کند، جایی که اکثر نشانه‌ها بدون تغییر باقی می‌مانند.(Balanced Learning)

با دیدن گهگاه توکن های اصلی، مدل از سوگیری نسبت به انتظار توکن های ماسک دار جلوگیری می کند که منجر به عملکرد بهتر در وظایف

مختلف NLP می شود.(Improved Generalization)

- Improving the performance: بهبود عملکرد مدل زبان ماسک شده (MLM) شامل چندین مرحله است. در اینجا به برخی از آنها بطور خلاصه اشاره میکنیم:

1. پیش آموزش در مجموعه داده های بزرگ: مدل هایی مانند BERT در مجموعه داده های وسیعی مانند Wikipedia و BookCorpus از قبل آموزش داده شده اند تا طیف گسترده ای از ساختارها و الگوهای زبانی را درک کنند.

مراحل: MLM را روی مجموعه داده های متنی بزرگ و متنوع جمع آوری کرده و از قبل آموزش میدهیم. این مرحله پیش‌آموزشی گسترده به مدل کمک می‌کند تا بازنمایی‌های زبانی قوی را بیاموزد.

2. تنظیم دقیق مجموعه داده های خاص: تنظیم دقیق مجموعه داده های خاص دامنه، دانش مدل را با context و واژگان خاص برنامه تطبیق می دهد.

مراحل: داده های متنی را جمع آوری کرده که نماینده دامنه ای است که قصد داریم مدل را در آن استقرار دهیم. سپس از این داده های خاص دامنه برای آموزش بیشتر مدل، اصلاح

درک آن و بهبود عملکرد آن در وظایف مربوط به آن دامنه استفاده میکنیم.

3. افزایش داده و پیش پردازش: افزایش داده های آموزشی با تبدیل های مختلف، استحکام مدل را افزایش می دهد.
مراحل: از روش هایی مانند جایگزینی مترادف، درج تصادفی، حذف تصادفی و ترجمه برگشتی برای ایجاد نمونه های آموزشی متنوع تر استفاده میتوان کرد.
4. افزایش پیچیدگی مدل: مدل های بزرگتر با پارامترهای بیشتر می توانند الگوها و وابستگی های پیچیده تری را در داده ها ثبت کنند.
مراحل: میتوانیم تعداد لایه ها، سرهای توجه و ابعاد تعبیه شده را افزایش دهیم. به عنوان مثال، حرکت از BERT-base به BERT-large.
5. تنظیم فراپارامتر: هایپرپارامترهای Fine-tuning مانند نرخ یادگیری، batch size و نرخ dropout می توانند به طور قابل توجهی عملکرد و کارایی آموزش مدل را بهبود ببخشند.
مراحل: از تکنیک هایی مانند جستجوی شبکه ای، جستجوی تصادفی یا بهینه سازی بیزی برای یافتن بهترین هایپرپارامترها استفاده کنیم.