Neural Networks and Deep Learning — Assignment 4

Instructors: **Dr. Bahrak**

TA: ریحانه خوارزمی

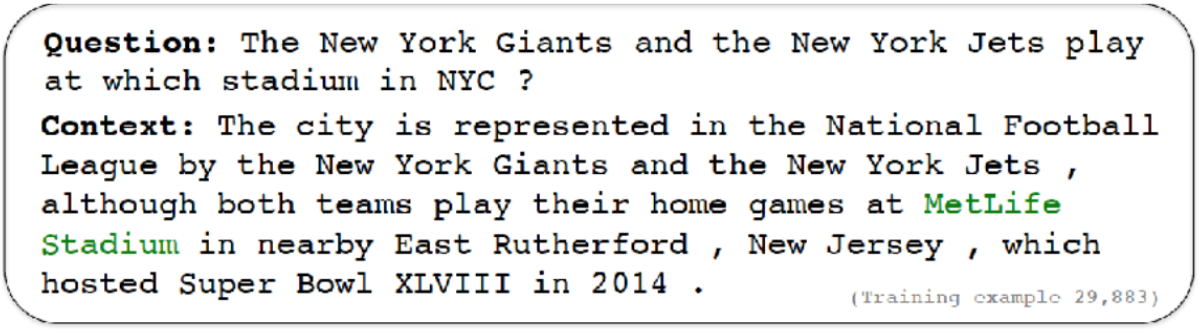
Deadline: 1404/03/01

پرسش 1: سامانه پرسش و پاسخ (Q&A)

مقدمه

سامانه‌های پرسش و پاسخ (QA) در سال‌های اخیر به‌دلیل توانایی‌شان در استخراج خودکار پاسخ‌ها از متن، توجه زیادی را جلب کرده‌اند. این سیستم‌ها در حوزه‌های مختلفی مانند بازیابی اطلاعات، پشتیبانی مشتری و دستیاران مجازی نقش کلیدی دارند.

هدف از این تمرین طراحی و پیاده سازی یک سیستم QA استخراجی(extractive question answering) مبتنی بر ترانسفورمرها میباشد که با دریافت یک متن و سوال مربوط به آن، بهترین پاسخ مناسب را از متن استخراج می‌نماید. برخلاف سیستمهای QA مولد که پاسخ‌ها را از ابتدا تولید میکنند، سیستمهای QA استخراجی برای تولید پاسخها به اطلاعات موجود در متن زمینه متکی هستند.(مانند تصویر1) با استفاده از قدرت یادگیری عمیق و مدل‌های زبانی از پیش آموزش‌دیده‌شده مانند BERT، این سیستم‌ها میتوانند به طور موثر ساختارهای پیچیده زبان، روابط متنی و تفاوت‌های معنایی لازم برای استخراج دقیق پاسخ را درک کرده و بهترین پاسخ را استخراج نمایند.



تصویر ۱- نمونه **pair** سوال و متن و جواب مشخص شده در متن ، برگرفته از دیتاست **squad**

در این تمرین، شما با طراحی و پیاده‌سازی یک مدل BERT روی مجموعه داده PQuad، به دنیای سیستم‌های پرسش و پاسخ در زبان فارسی وارد خواهید شد. این تمرین شامل پیش‌پردازش داده‌ها، طراحی مدل، مدیریت استثناها و ارزیابی عملکرد با معیارهای امتیاز تطابق دقیق (EM) و امتیاز F1 است.

1-1. معرفی مقاله

برای آشنایی و فهم عملکرد مدل ترنسفورمری BERT، مقاله BERT اصلی (Devlin et al., 2018) را از این لینک بخوانید تا معماری و تسک‌ها وتوابع هدفی که BERT بر روی آن‌ها پیش آموزش شده‌است (pretrain) را درک کنید. اجزای کلیدی مدل BERT، از جمله معماری ترانسفورمری، نمایش ورودی و اهداف pre training آن را شرح دهید.

1-2. پیش‌پردازش داده‌ها (30 نمره)

دیتاست مورد استفاده در این تمرین دیتاست PQuad می‌باشد که از طریق این لینک قابل دسترسی است. در ابتدا لازم است اطلاعات آماری (شامل تعداد، نوع و طول متن‌ها و سوالات ...) دیتاست مورد نظر را بعد از اینکه در دیتا فریم جای گرفتند نمایش دهید. این کار به شما کمک می‌کند تا از کیفیت داده‌ها مطمئن شوید و هرگونه داده غیرمنطقی یا گمشده را شناسایی کنید.

سپس پیش پردازش های لازم از جمله Tokenization و Embedding داده‌ها برای انجام تسک مورد نظر را پیاده سازی کنید.

1-2-1. Tokenization

BERT برای پردازش متون از توکن‌ها استفاده می‌کند. به این معنا که متن‌های طولانی به واحدهای کوچکتری به نام توکن تقسیم می‌شوند. این توکن‌ها معمولاً شامل کلمات، نمادها و حتی بخشی از کلمات هستند. برای انجام توکن‌سازی در مدل BERT، می‌توانید از توکنایزر مخصوص BERT استفاده کنید که توسط کتابخانه‌هایی مانند `transformers` ارائه شده است. این توکنایزر به صورت خودکار ورودی‌ها را به توکن‌های BERT تبدیل می‌کند. این توکنایزر به طور همزمان عملیات‌هایی مانند پوشش (Padding) و قطع (Truncation) را نیز انجام دهد تا اندازه ورودی‌ها برای مدل مناسب شود.

1-2-2. امبدینگ (Embedding)

امبدینگ‌ها فرایندی هستند که توکن‌ها را به وکتورهای عددی تبدیل می‌کنند. این وکتورها نمایانگر ویژگی‌های معنایی و دستوری کلمات و جملات هستند. در مدل‌های BERT، هر کلمه یا توکن به یک بردار عددی در فضای امبدینگ تبدیل می‌شود که نمایانگر ویژگی‌های آن کلمه است.

BERT از امبدینگ‌های خود برای هر توکن استفاده می‌کند که این امبدینگ‌ها شامل سه نوع اصلی هستند:

Token Embeddings: که به هر توکن یک وکتور اختصاص داده می‌شود.
Segment Embeddings: برای نشان دادن تفاوت بین دو بخش (متن و سوال).
Position Embeddings: برای نمایش موقعیت هر توکن در متن (یعنی توکن‌های اول، دوم و...).
این امبدینگ‌ها به همراه هم در مدل BERT برای پردازش داده‌ها و تولید نتایج استفاده می‌شوند.
در هنگام توکن‌سازی، توکن‌ها به صورت عددی به مدل وارد می‌شوند. این توکن‌ها سپس به امبدینگ‌های عددی تبدیل می‌شوند که شامل اطلاعات معنایی هر کلمه و موقعیت آن در جمله است.

2-1-3. مدیریت ورودی‌ها

برای مدل BERT، ورودی‌ها باید به دو قسمت اصلی تقسیم شوند:

1. متن سوال: که باید به مدل وارد شود.
2. متن زمینه (context): که شامل متنی است که مدل باید از آن پاسخ را استخراج کند.

در مدل‌های استخراجی، سوال و متن زمینه باید به هم متصل شوند تا مدل بتواند بهترین پاسخ را پیدا کند. این کار معمولاً با استفاده از یک ترکیب از توکن‌های سوال و زمینه انجام می‌شود.
در طول پیش‌پردازش، سوال و متن زمینه باید به یک توکن واحد تبدیل شوند. مدل BERT به طور خاص از جداکننده‌ها (separator token) برای تفکیک سوال از متن زمینه استفاده می‌کند.

2-1-4. ایجاد ماسک توجه (Attention Mask)

ماسک توجه یکی دیگر از اجزای مهم در پردازش داده‌ها است که به مدل می‌گوید کدام بخش از ورودی‌ها باید پردازش شود و کدام بخش‌ها باید نادیده گرفته شوند. معمولاً این ماسک برای داده‌هایی که طول متفاوتی دارند استفاده می‌شود. به طور معمول، مدل‌های BERT از ماسک توجه برای جلوگیری از پردازش پدها (padding tokens) استفاده می‌کنند.

2-1-5. تقسیم داده‌ها به دسته‌ها (Batching)

پس از انجام تمام مراحل پیش‌پردازش، داده‌ها باید به دسته‌هایی تقسیم شوند تا مدل بتواند آن‌ها را به طور همزمان پردازش کند. این کار به افزایش کارایی کمک می‌کند و معمولاً از یک `DataLoader` در PyTorch یا مشابه آن برای این کار استفاده می‌شود.

برای validation، train و test از داده های با همین اسامی در دیتاست استفاده کنید.

1-3. پیاده‌سازی مدل (50 نمره)

ساختار کلی یک سامانه پرسش و پاسخ شامل ورودی‌هایی نظیر سوالات کاربران و متون مرجع است، که به مدل اجازه می‌دهد اطلاعات مرتبط را استخراج کند. در این دیتاست نیز متن، سوال و جواب را داریم. خروجی‌ها معمولاً شامل پاسخ‌های تولید شده و نمره اعتبار آن‌ها بر اساس جواب‌هایی که داریم هستند. مدل‌های استفاده‌شده معمولاً شامل مدل‌های پردازش زبان طبیعی مانند BERT هستند که برای درک متن و شناسایی روابط بین سوال و پاسخ طراحی شده‌اند. توابع خطا مانند Cross-Entropy Loss و Mean Squared Error برای آموزش مدل به کار می‌روند تا درک سوالات، استخراج اطلاعات صحیح و تولید پاسخ‌های منطقی را یاد بگیرند. در نهایت، هدف مدل این است که توانایی خود را در شناسایی و تولید پاسخ‌های دقیق و مرتبط به سوالات بهبود بخشد.

برای پیاده‌سازی، از دو مدل مبتنی بر ParsBERT و ALBERT استفاده نمایید. مدل های از پیش آموزش دیده شده آن‌ها در huggingface از این لینک و این لینک قابل دسترسی میباشند. ParsBERT همان ساختار مدل BERT را داشته که روی متون فارسی آموزش دیده شده است. ALBERT یک مدل مبتنی بر ترانسفورمرها است که بر اساس معماری BERT (مدلی مبتنی بر ترنسفورمر که با استفاده از توجه دوطرفه به تحلیل و درک زمینه‌ای متن‌ها می‌پردازد) ساخته شده است و مدل آموزش دیده شده‌ی آن بر روی دیتای فارسی قابل دسترس میباشد.

شبکه طراحی شده را با استفاده از یکی از این دو مدل پیاده سازی نمایید. (توجه فرمایید استفاده از کلاس AutoModelForQuestionAnswering مجاز نمی باشد.)

1-4. ارزیابی و پس‌پردازش (20 نمره)

در طول انجام تسک توسط مدل استثناهایی در حین بارگیری داده‌ها، پیش پردازش و پس پردازش به دلیل طولانی بودن متون زمینه و محدودیت ورودی مدل‌های ترانسفورمری بوجود می‌آید، که نیاز است آنها را مدیریت کنید، اطمینان حاصل کنید که مدل شما میتواند اینگونه استثنائات را مدیریت کند، آنها را گزارش دهید.

در نهایت پس از مدیریت استثنائات، دو مدل آموزش دیده خود را بر روی مجموعه داده تست با استفاده از دو معیار EM و F1-score ارزیابی کنید و نتایج خود را با نتایج ذکر شده در مقاله مقایسه نمایید. (برای ارزیابی می توانید از ابزار های آماده استفاده نمایید.)

پرسش 2: تنظیم دقیق (Fine-Tuning) مدل زبانی GPT2 برای تحلیل احساسات کاربران IMDb

مقدمه

سیستم‌های تحلیل احساسات[1] به عنوان یکی از مهم‌ترین کاربردهای پردازش زبان طبیعی، نقش حیاتی در درک خودکار عواطف و نظرات موجود در متون ایفا می‌کنند. این سیستم‌ها امروزه در حوزه‌های متنوعی مانند نظرسنجی‌های اجتماعی، تحلیل بازخورد مشتریان، مدیریت شهرت برندها، و حتی پژوهش‌های بازار به کار گرفته می‌شوند.

مدل‌های زبانی بزرگ[2] مانند GPT[3]، تحولی شگرف در حوزه پردازش زبان طبیعی ایجاد کرده‌اند. این مدل‌ها با معماری مبتنی بر ترانسفورمر و آموزش روی حجم عظیمی از داده‌های متنی، توانایی درک پیچیده‌ترین الگوهای زبانی را کسب می‌کنند. خانواده مدل‌های GPT، از GPT-1 تا GPT-4، با بهره‌گیری از یادگیری خودنظارتی[4] و توجه یک‌طرفه[5]، توانایی تولید متن‌های منسجم و پاسخ به سوالات را دارند.

در این تمرین، از GPT-2 (نسخه‌ای متعادل از نظر اندازه و کارایی) برای انجام تحلیل احساسات روی مجموعه داده IMDb استفاده می‌کنیم. برخلاف روش‌های سنتی که از طبقه‌بندهای اختصاصی مانند BERT استفاده می‌کنند، در اینجا مسئله را به شکل تولید متن شرطی فرمول‌بندی می‌کنیم؛ به این معنا که مدل پس از خواندن نظر، کلمه «مثبت» یا «منفی» را به عنوان پاسخ تولید می‌کند.

هدف این تمرین، آشنایی با تنظیم دقیق[6] مدل‌های ازپیش‌آموزش‌دیده، درک تفاوت‌های رویکردهای تولیدی در مقابل تمایزی، و بررسی چالش‌های استفاده از GPT برای وظایف طبقه‌بندی است. در پایان، مدل آموزش‌دیده را با معیارهای Accuracy و Perplexity (که میزان اطمینان مدل در پیش‌بینی‌ها را می‌سنجد) ارزیابی کرده و نتایج را با روش‌های پایه مقایسه خواهیم کرد.

---

[1] Sentiment Analysis
[2] LLMs
[3] Generative Pre-trained Transformer
[4] Self-Supervised Learning
[5] Unidirectional Attention
[6] Fine-Tuning

2-1. معرفی مقاله (10 نمره)

برای آشنایی با مبانی نظری مدل‌های ترانسفورمر و به ویژه معماری GPT، مقاله را از این لینک مطالعه کنید و در گزارش خود به موارد زیر پاسخ دهید:

1. معماری مبتنی بر توجه یک‌طرفه (Unidirectional Attention) در GPT چگونه کار می‌کند؟
2. تفاوت‌های کلیدی بین معماری GPT و BERT چیست؟
3. روش‌های پیش‌آموزش و تنظیم دقیق در GPT چگونه انجام می‌شود؟

2-2. پیش‌پردازش داده‌ها (30 نمره)

در این بخش از مجموعه داده IMDb که شامل 50,000 نظر فیلم با برچسب‌های مثبت و منفی است استفاده می‌کنیم. این مجموعه داده از طریق کتابخانه datasets قابل دسترسی است:

```
# Load and prepare dataset
dataset = load_dataset("imdb")
```

الف) توزیع کلاس‌ها را بررسی کنید و نمایش دهید، طول متون را با معیارهای میانگین، میانه، بیشینه و کمینه تحلیل کنید و نمونه‌ای از داده‌های خام را نشان دهید. برای ساده‌تر شدن از نظرات با طول کوتاه‌تر (حداقل ۵۰۰ کاراکتر) می‌توانید استفاده کنید.

ب) برای قالب‌بندی داده‌ها به این صورت عمل کنید:

"Review: {متن نظر}\nSentiment: {label}"

برچسب‌های عددی را به متن تبدیل کنید:

1 → “positive”, 0 → “negative”

پ) برای توکنایز کردن داده‌ها از توکنایزر GPT-2 استفاده کنید:

```
tokenizer = GPT2Tokenizer.from_pretrained("gpt2")
tokenizer.pad_token = tokenizer.eos_token
```

ت) یک دیتا لودر[7] بسازید، اندازه دسته[8] مناسب را پیدا کنید و با صدا کردن ستون train و test می‌توانید به این داده‌ها دسترسی پیدا کنید. (برای سرعت بخشیدن به کار ۵۰۰۰ نمونه از داده‌های train و ۱۰۰۰ نمونه داده‌ی تست را استفاده کنید.)

[7] Data Loader
[8] Batch

2-3. پیاده‌سازی مدل (50 نمره)

در این بخش از پروژه، شما موظف به پیاده‌سازی یک مدل مبتنی بر GPT-2 هستید. این مدل باید با توجه به معماری خاص آن و نکات مهم زیر پیاده‌سازی شود:

```
model = GPT2LMHeadModel.from_pretrained("gpt2")
```

مدل GPT-2 یکی از مدل‌های معروف و قدرتمند در پردازش زبان طبیعی است. برای شروع، شما باید از یک نسخه پیش‌آموزش دیده این مدل استفاده کنید. این کار به شما این امکان را می‌دهد که از دانش از پیش آموخته‌شده مدل بهره ببرید و تمرکز بیشتری بر روی آموزش لایه‌های بالایی داشته باشید.

- یکی از نکات کلیدی در این پیاده‌سازی، فریز[9] کردن لایه‌های پایه مدل است. با این کار، شما تنها لایه‌های بالایی (لایه‌های نازک‌تر) را آموزش خواهید داد. این روش معمولاً به افزایش سرعت آموزش و جلوگیری از فرایند Overfitting کمک می‌کند. با فریز کردن لایه‌های پایه، می‌توانید مطمئن شوید که ویژگی‌های اساسی مدل حفظ شده و تنها جنبه‌های خاص‌تری از داده‌ها یاد گرفته می‌شود.
- برای بهینه‌سازی عملکرد مدل، تنظیم هایپرپارامترهایی مانند نرخ یادگیری،[10] تعداد ایپاک‌ها و اندازه دسته‌ها بسیار حائز اهمیت است. شما باید با انجام آزمایش‌های اولیه، بهترین ترکیب این پارامترها را پیدا کنید.
- در این پیاده‌سازی، لازم است که تابع loss تنها برای labelها محاسبه شود. این بدین معنی است که شما باید فقط بر روی داده‌های هدف (target) تمرکز کنید و اطمینان حاصل کنید که مدل در یادگیری این داده‌ها بهینه عمل می‌کند.

2-4. ارزیابی و تحلیل نتایج (10 نمره)

حال با معیارهای ارزیابی دقت[11] (که نشان می‌دهد چه میزان از پیش‌بینی‌های مدل با برچسب‌های واقعی تطابق دارند) و پرپلکسیتی[12] (که میزان اطمینان مدل در پیش‌بینی‌ها را اندازه‌گیری می‌کند و هر چه پایین‌تر باشد نشانگر اطمینان بالاتر است) مدل را بسنجید. هم‌چنین پس از محاسبه Recall و Precision جدول کانفیوژن ماتریکس[13] آن را نمایش دهید.

توضیح دهید که تفاوت این روش با استفاده از مدل‌هایی مانند BERT که نوعی از آن به طور خاص برای طبقه‌بندی ساخته شده است چیست؟ و به بررسی نقاط ضعف و قوت هر روش بپردازید.

---

[9] Freeze
[10] Learning rate
[11] Accuracy
[12] Perplexity
[13] Confusion matrix

2-5. امتیازی (10 نمره)

از دیگر رویکردها برای آموزش مدل‌های از پیش آموزش دیده GPT استفاده از روش لورا[14](LoRA) با منابع کمتر و حفظ عملکرد است. LoRA ماتریس‌های کم‌رتبه قابل آموزش را به وزن‌های اصلی مدل اضافه می‌کند و به مدل این امکان را می‌دهد که بدون تغییر مستقیم وزن‌های پیش‌آموزش دیده تنظیم شود و تعداد پارامترهایی که باید به‌روز شوند را کاهش دهد. به جای فریزکردن لایه‌ها و استفاده محدود از دو لایه آخر، سعی کنید تمرین را با این تکنیک پیاده‌سازی کنید.

[14] Low-Rank Adaptation (LoRA)