به نام خدا

درس یادگیری عمیق

تمرین سری چهارم

استاد درس : دکتر محمدرضا محمدی

دستیاران : رضا علیدوست ، علیرضا حقانی

و امیرحسین نمازی

دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر

نیمسال دوم تحصیلی ۱۴۰۳ - ۱۴۰۴

مهلت تحویل : ۱۴۰۴/۰۲/۲۳

لطفا به نکات موجود در سند قوانین انجام و تحویل تمرین ها دقت فرمایید.

سوالات تئوری

۱.  بر اساس مقاله لطفا به سوالات زیر پاسخ دهید (۱۰ نمره):

(آ) چالش‌های اصلی در زمینه مدیریت حافظه که سیستم‌های خدمات‌دهی LLM موجود مواجه هستند، چیست؟

(ب) PagedAttention چگونه به این چالش‌ها پاسخ می‌دهد؟

(ج) اهمیت اشتراک‌گذاری حافظه کش KV در سیستم‌های خدمات‌دهی LLM را مورد بحث قرار دهید. vLLM چگونه اشتراک‌گذاری حافظه را تسهیل می‌کند و این موضوع چه پیامدهایی برای توان عملیاتی کلی سیستم دارد؟ پاسخ خود را با جزئیات موجود در مقاله بیان کنید.

ویدیوی ارائه نویسندگان مقاله در یک کنفرانس

۲.  با توجه به Multi-Head Attention به پرسش‌های زیر پاسخ دهید (۱۰ نمره):

(آ) چرا در مدل‌های ترنسفورمر از توجه چندسری (Multi-Head Attention) استفاده می‌شود؟ و این سرهای توجه چه نوع اطلاعاتی را می‌توانند یاد بگیرند؟

(ب) فرض کنید یک مدل آموزش‌دیده داریم که بر پایه‌ی توجه چندسری (Multi-Head Attention) ساخته شده است و می‌خواهیم برای افزایش سرعت پیش‌بینی، سرهای توجه کم‌اهمیت‌تر را

حذف (Prune) کنیم. چگونه می‌توانیم آزمایش‌هایی طراحی کنیم تا اهمیت هر سر توجه را اندازه‌گیری کنیم؟

(ج) حذف سرهای توجه چه اثری روی وظایف پایین‌دستی (مثل طبقه‌بندی یا ترجمه) دارد؟ از چه معیارهایی برای ارزیابی تأثیر حذف سرها استفاده کنیم؟

(د) آیا می‌توان از یادگیری تقویتی (Reinforcement Learning) برای انتخاب دینامیک سرهای توجه استفاده کرد؟

(اختیاری: می‌توانید از مقاله بهره بگیرید.)

۳.  در رابطه با Additive Attention به پرسش‌های زیر پاسخ دهید (۱۰ نمره):

(آ) آیا ایده‌ی خوبی است که در مدل ترنسفورمر، توجه ضرب نقطه‌ای مقیاس‌شده (Scaled Dot-Product Attention) را با توجه جمعی (Additive Attention) جایگزین کنیم؟ چرا؟

(ب) آیا می‌توان ترکیبی از این دو نوع توجه استفاده کرد؟

(ج) یک توجه چند سر additive با ۳ سر را در نظر بگیرید. ابعاد query، key و value را به ترتیب ۱۰، ۲۰، ۳۰ در نظر بگیرید فرض کنید هر کدام از سرها به ابعاد ۱۰۰ تبدیل شوند. همچنین در نظر داشته باشید که خروجی نهایی ۵۰ میباشد. با فرض اینکه دنباله ورودی ۶۴ تایی باشد، تعداد پارامترها را مشخص کنید.

۴.  در رابطه با کاربرد مدل‌های transformer در سری‌های زمانی به سوالات زیر پاسخ دهید (۲۰ نمره):

(آ) چه زمانی استفاده از ترنسفورمر در سری زمانی مناسب‌تر از استفاده از LSTM است؟

(ب) چگونه داده‌های سری زمانی باید برای ورودی به ترنسفورمر پیش‌پردازش شوند؟

(ج) چه تفاوتی بین ترنسفورمر استاندارد و ترنسفورمر مخصوص سری زمانی (مانند Time Series Transformer یا Informer) وجود دارد؟

(د) چگونه می‌توان از ترنسفورمر برای پیش‌بینی چند مرحله‌ای (multi-step forecasting) در سری‌های زمانی استفاده کرد؟

(ه) نحوه‌ی عملکرد مدل iTransformer جهت وظیفه‌ی Time Series Forecasting را توضیح دهید. (میتوانید از مقاله بهره بجویید.)

سوالات عملی

۵.  در این تمرین با مدل ViT برای دسته‌بندی تصاویر آشنا خواهید شد. شما یک مدل pretrained را با استفاده از Hugging Face بارگذاری می‌کنید، وزن‌های attention آن را تجزیه و تحلیل می‌کنید و برای درک بهتر مکانیزم توجه و رفتار آن در مدل پیش آموخته شده، وزن‌های یادگیری شده را نمایش خواهید داد. در بخش بعدی آن را روی دیتاست CIFAR-10 آموزش خواهید داد (fine-tune) و در نهایت نقش attention headها را بررسی می‌کنید. در این سوال از نوتبوک Q5.ipynb استفاده کنید.(۲۵ نمره)

(آ) از کتابخانه transformers در Hugging Face برای بارگذاری مدل pretrained استفاده کنید (ترجیحا مدل google/vit-base-patch16-224). در این بخش پس از انجام پیش‌پردازش تصویر مورد نظر، با استفاده از مدل پیش‌آموزش دیده خروجی مدل را بدست آورده و ۵ کلاس برتر پیش‌بینی شده را همراه با احتمالات پیش‌بینی محاسبه کنید.

(ب) با فراخوانی مدل می‌توانید به وزن‌های مکانیزم توجه برای تصویر موردنظر دسترسی داشته باشید. در این بخش attention weights مربوط به توکن [CLS] را استخراج کنید و نقشه‌های توجه این توکن را برای تمام لایه و headها به صورت جداگانه نمایش دهید.

(ج) Attention Rollout روشی برای مصورسازی و تفسیر مکانیزم توجه در مدل‌های Transformer است. در این روش با ضرب تجمعی ماتریس‌های attention در لایه‌ها، مسیر توجه از ورودی تا خروجی مدل به‌صورت یکپارچه نمایش داده می‌شود. با استفاده از روش attention rollout و توضیحات داخل نوتبوک جریان تاثیر هر patch از تصویر روی توکن cls را در طول لایه‌ها نمایش دهید.

(د) مدل پیش‌آموخته شده را بر روی دیتاست CIFAR-10 آموزش (fine-tune) دهید و دقت آن را گزارش کنید.

(ه) در این بخش پس از آموزش روی دادگان بررسی کنید که دور ریختن یک یا چند head چه تاثیری در عملکرد مدل ایجاد می‌کند. با استفاده از داده validation تحلیل کنید کدام headها نقش مهمتری در تصمیم‌گیری مدل دارند.

۶.  در این تمرین، مدل ترجمه ماشینی مبتنی بر توجهی را آموزش خواهید داد تا کلمات را از انگلیسی به Pig-Latin ترجمه کنید.Pig-Latinیک بازی زبانی است که در آن قوانین به صورت مستقل برای هر کلمه اعمال می شود: (۲۵ نمره)

- اگر اولین حرف یک کلمه، حرف بی صدای انگلیسی باشد، آن حرف به انتهای کلمه منتقل شده و حروف ayبه انتهای کلمه اضافه می شوند: team → eamtay .
- اگر اولین حرف، یک حرف صدادار انگلیسی باشد، کلمه بدون تغییر باقی می ماند و حروف way به انتهای کلمه اضافه می شوند: impress → impressway.
- برخی از جفت حروف مانند sh به عنوان یک بلوک در نظر گرفته میشوند و به صورت کل به انتهای رشته منتقل میشوند: shopping → oppingshay

هدف این است که مدل ترجمه ماشینی قوانین را به طور ضمنی از طریق جفت های کلمات (English، Pig-Latin) که source کلمه انگلیسی و target ترجمه آن به Pig-Latin است، یاد بگیرد.

داده ها:

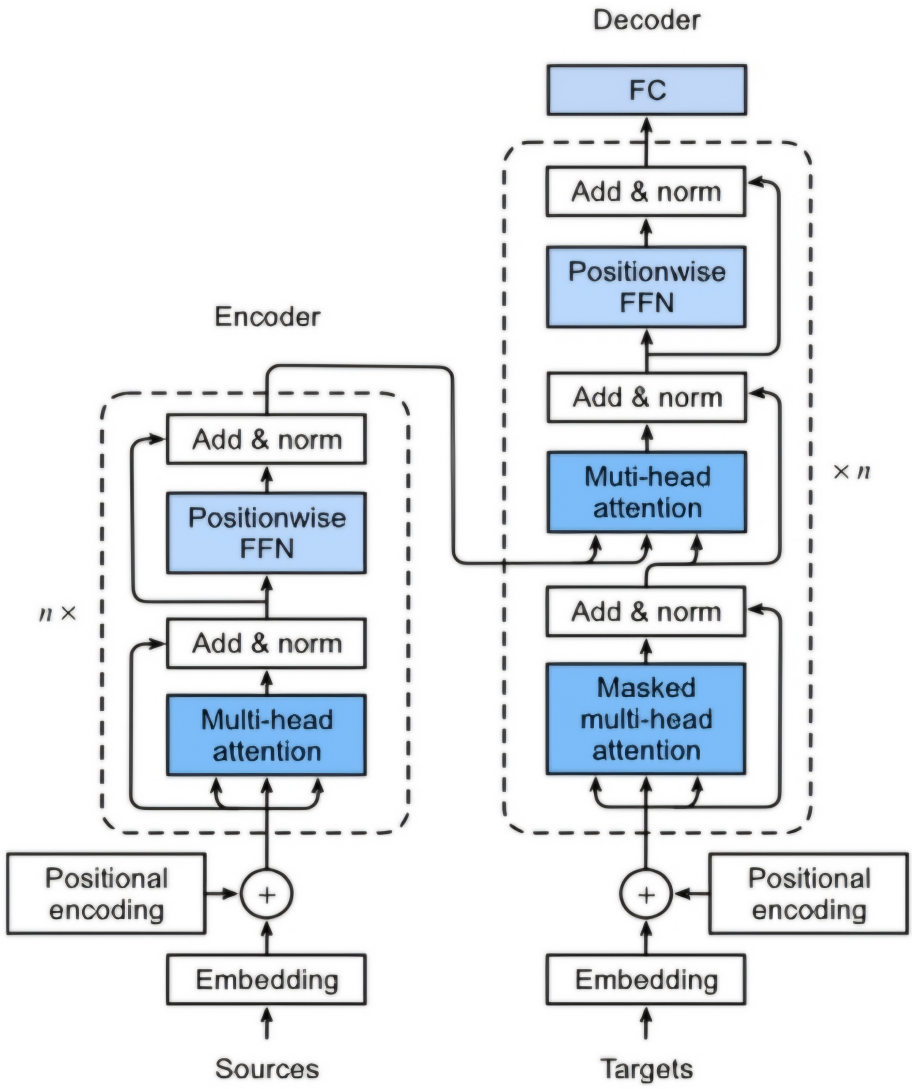
در این تمرین از دو مجموعه داده استفاده خواهید کرد:

- واژگان مجموعه داده کوچک شامل ۲۹ نشانه است: ۲۶ حرف استاندارد الفبا (همه با حروف کوچک)، نماد خط تیره "-" و دو نشانه <SOS> و <EOS> که به ترتیب شروع و پایان یک دنباله را نشان می‌دهند. مجموعه داده شامل ۳۱۹۸ جفت (English، Pig-Latin) منحصر به فرد است.
- مجموعه داده بزرگ‌تر، شامل ۲۰،۰۰۰ کلمه انگلیسی پرکاربردتر است که با مجموعه داده قبلی ترکیب می‌شود و ۲۲۴۰۲ کلمه منحصر به فرد به دست می‌آید

(آ) به بخش scaled dot product attention در نوتبوک pigLatin مراجعه کرده و بخش‌های مشخص‌شده را تکمیل کنید.

(ب) مدل Transformer را با استفاده از hidden size های ۳۲ و ۶۴ و با استفاده از مجموعه داده کوچک و بزرگ (در مجموع ۴ اجرا) اجرا کنید و اثرات افزایش ظرفیت مدل از طریق hidden size و افزایش اندازه مجموعه داده را گزارش کنید.

(ج) به معماری Transformer در شکل زیر نگاه کنید. در هر لایه ابتدا CausalScaledDotAttention را به ورودی‌های decoder و سپس ScaledDotAttention را به encoder annotations اعمال می‌کنیم. __init__ بخش decoder را طوری تغییر دهید که فقط از ScaledDotAttention استفاده کند. نتایج خود را حالت قبلی مقایسه کنید.


Decoder
FC
Add & norm
Positionwise FFN
Add & norm
Muti-head attention
Add & norm
Masked multi-head attention
× n
Encoder
Add & norm
Positionwise FFN
Add & norm
Multi-head attention
n ×
Positional encoding
+
Embedding
Sources
+
Positional encoding
Embedding
Targets