مدل های تولیدی

نیم‌سال اول ۱۴۰۳-۱۴۰۴

دانشکده‌ی علوم کامپیوتر مدرس: دکتر سیدصالحی

تمرین سری اول موعد تحویل: ۲۶ آبان

- مهلت ارسال پاسخ تا ساعت ۵۹ : ۲۳ روز مشخص‌شده است.
- در طول ترم امکان ارسال با تاخیر پاسخ همه‌ی تمارین (به استثنای هفته‌ی امتحان میانترم) تا سقف هفت روز وجود دارد. پس از گذشت این مدت، پاسخ‌های ارسال‌شده پذیرفته نخواهندبود.
- هم‌کاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد.
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.

مسئله‌ی ۱. (۴۰ نمره)

در این سوال، شما با مدل‌های مولد مبتنی بر شبکه‌های عصبی پیچشی و به طور خاص مدل PixelCNN آشنا خواهید شد. هدف این سوال پیاده‌سازی مدل PixelCNN بر روی مجموعه داده MNIST است که یک مجموعه از تصاویر دست‌نوشته اعداد است. شما باید مدل PixelCNN را طوری طراحی کنید که بتواند تصاویر MNIST را به صورت مولد (یعنی پیکسل به پیکسل) تولید کند.

۱. **آماده‌سازی داده‌ها**
مجموعه داده MNIST را با استفاده از کتابخانه‌های رایج یا به صورت دستی بارگذاری کرده و داده‌ها را به شکل مناسب برای ورودی مدل PixelCNN تبدیل کنید.

۲. **پیاده‌سازی مدل PixelCNN**
یک شبکه عصبی پیچشی طراحی کنید که به صورت پیکسل به پیکسل کار کند. برای این کار از کانولوشن‌های ماسک‌دار استفاده کنید تا مدل فقط بتواند پیکسل‌های قبلی را برای پیش‌بینی پیکسل فعلی ببیند.

- برای پیاده سازی این شبکه عصبی، از دو نوع ماسک متفاوت A و B باید استفاده کنید. تفاوت این دو نوع را شرح دهید..
- برای پیاده سازی مدل می توانید از تنظیمات زیر استفاده کنید. استفاده از این تنظیمات اجباری نبوده و درصورتی که با هایپرپارامترهای دیگر جواب مناسبی می گیرید، میتوانید این مقادیر را تغییر دهید.
  - به طور کلی از ۸ لایه پیچشی استفاده کنید به گونه ای که اندازه کرنل ۶ لایه اول ۷ باشد و اندازه کرنل دو لایه آخر ۱ باشد.
  - تعداد کانال ها را ۶۴ درنظر بگیرید.
  - بعد از هر لایه پیچشی از تابع فعال ساز ReLU استفاده کنید. بعد از آخرین لایه از تابع Sigmoid بهره ببرید.

۳. **آموزش مدل**
مدل را روی مجموعه داده آموزش دهید. هایپرپارامترها از قبیل نرخ یادگیری و تعداد دوره را به گونه ای تنظیم کنید که مدل آموزش ببیند. سپس نمودار خطای داده آموزش و تست را بر حسب دوره را رسم کرده و در گزارش خود نمایش دهید.

۴. **تولید نمونه**
پس از آموزش مدل ۱۰۰ نمونه تصویر تولید کرده و در گزارش نمایش دهید.

تمام قسمت ها باید توسط خودتان پیاده سازی شود. در صورت استفاده از کدهای آماده و موجود در اینترنت نمره ای به شما تعلق نمی گیرد. در صورتی که در قسمت های غیر اصلی از کدهای موجود در اینترنت استفاده می کنید، باید منبع خود را ذکر کنید.

مسئله‌ی ۲. (۶۰ نمره)

در این سوال، شما با مدل‌های مولد که از شبکه‌های RNN برای تولید متن استفاده می‌کنند، آشنا خواهید شد. هدف این سوال پیاده‌سازی یک مدل Decoder برای تولید متن فارسی است که با استفاده از داده‌های ویکی‌پدیا فارسی آموزش دیده است. همچنین، در این سوال با متد Transfer Learning و استفاده از مدل‌های از پیش آموزش‌دیده نیز آشنا خواهید شد.

۱. آماده سازی و پیش پردازش دادگان

در این تمرین، ابتدا لازم است داده‌های ویکی‌پدیا فارسی را از پلتفرم Kaggle بارگذاری کنید. برای این کار می‌توانید از کد زیر استفاده کنید:

```
۱
۲ mkdir ~/.kaggle
۳ cp ./kaggle.json ~/.kaggle/
۴ chmod 600 ~/.kaggle/kaggle.json
۵ kaggle datasets download miladfa7/persian-wikipedia-dataset -f Persian-WikiText-1.txt
۶
```

پس از بارگذاری داده‌ها، باید پیش‌پردازش‌های لازم را انجام دهید و آن‌ها را به شکلی مناسب برای ورودی مدل RNN تبدیل کنید. توجه داشته باشید که انجام صحیح این مرحله در عملکرد مدل بسیار اهمیت دارد. داده‌های ویکی‌پدیا معمولاً شامل نویز و اطلاعات غیرضروری هستند. بنابراین، کاراکترها، کلمات، یا هر قسمتی که باید از متن حذف شود را شناسایی کنید. به‌عنوان مثال، stop wordها معمولاً از داده‌های ورودی مدل حذف می‌شوند. توضیح دهید چرا بهتر است این کلمات از متن حذف شوند و همچنین هر مورد مشابه دیگر را با ذکر دلیل حذف کنید.

سپس با استفاده از کتابخانه hazm، متن را نرمال کنید. از متد lemmatize در این کتابخانه استفاده کرده و علت استفاده از آن را توضیح دهید.

در مرحله آخر، داده‌ها را توکنایز کنید. این یکی از مهم‌ترین مراحل پیش‌پردازش داده‌های متنی است. Byte-Pair Encoding (BPE) یکی از متدهای رایج برای توکنایز کردن داده‌های متنی است. در این بخش، درباره نحوه عملکرد این الگوریتم و مزایا و معایب آن توضیح دهید.

۲. آموزش مدل

در این تمرین، هدف ما آموزش یک مدل با استفاده از متد n-gram است. ابتدا در مورد نحوه عملکرد این متد تحقیق کنید.

سپس با استفاده از کتابخانه PyTorch، Dataset و Dataloader را به گونه‌ای پیاده‌سازی کنید که برای آموزش مدل به صورت n-gram مناسب باشد. یک مدل RNN طراحی کنید که برای تسک تولید توکن بعدی (Next Token Generator) مناسب باشد. شما می‌توانید از هر یک از انواع مدل‌های RNN استفاده کنید. تعیین معماری و هایپرپارامترهای مدل نیز بر عهده شماست. توجه داشته باشید که تعداد پارامترهای مدل باید با توجه به نوع تسک و حجم داده‌های آموزشی به‌درستی انتخاب شود.

سپس مدل طراحی‌شده را با استفاده از یک تابع loss مناسب آموزش دهید. نمودار کاهش loss در طول مراحل آموزش را رسم کرده و تمامی پارامترهای استفاده‌شده در مدل را در گزارش خود ذکر کنید.

۳. ارزیابی مدل

درباره معیار Perplexity برای ارزیابی مدل های زبانی تحقیق کرده و از آن برای ارزیابی مدل آموزشی خود استفاده کنید.

۴. **تولید متن**
در این بخش، از مدل آموزش‌دیده خود استفاده کنید. یک قسمت از یک متن را به عنوان ورودی به مدل بدهید و ادامه متن را با توجه به این ورودی تولید کنید.

۵. **استفاده از ترنسفرمرها**
در این بخش، به منظور تولید متن و ارتقاء کیفیت متن تولید شده در قسمت قبل با استفاده از از دیتاست آماده شده در قسمت قبل , یک مدل ترنسفورمر از ابتدا (فرام‌اسکرچ) طراحی کرده و خروجی‌های آن را با نتایج بخش قبلی مقایسه کنید .

۶. **بهینه‌سازی محاسبات توجه در ترانسفورمرها**

فرض کنید یک مدل ترانسفورمر با طول توالی ورودی $n = ۴۰۹۶$ و ابعاد فضای نمایشی $d = ۵۱۲$ دارید. پیچیدگی محاسباتی مدل ترانسفورمر معمولی برابر $O(n^۲ \cdot d)$ است که برای ورودی‌های طولانی بسیار پرهزینه می‌شود. اکنون به سوالات زیر با دقت و محاسبات کامل پاسخ دهید:

(آ) تعداد کل عملیات محاسباتی در محاسبات توجه استاندارد را برای این مدل محاسبه کنید.

(ب) فرض کنید از تکنیک KV Cache استفاده می‌شود که فقط نیاز به محاسبه توجه برای ۲۰% از توالی‌های جدید است. تعداد کل عملیات محاسباتی را با این فرض دقیقاً محاسبه کنید.

(ج) حال فرض کنید از تکنیک Grouped Query Attention نیز استفاده می‌شود، به طوری که محاسبات برای این ۲۰% از توالی‌ها به ۶۴ گروه تقسیم شود و به‌صورت موازی انجام شود. تعداد عملیات محاسباتی را با دقت محاسبه کنید.

(د) فرض کنید علاوه بر استفاده از KV Cache و Grouped Query Attention، به دلیل محدودیت‌های سخت‌افزاری و حافظه، هزینه سربار ناشی از مدیریت حافظه پنهان و تقسیم‌بندی گروه‌ها باعث افزایش ۳۵ درصدی کل زمان محاسبات شود. تعداد کل عملیات نهایی (با در نظر گرفتن سربار) را دقیقاً محاسبه کنید.

(ه) اگر به دلیل تنگنای حافظه، پردازنده تنها بتواند ۷۵% از عملکرد معمول خود را ارائه دهد، چند عملیات اضافی ناشی از این محدودیت به محاسبات اضافه می‌شود؟

(و) در نهایت، با فرض اینکه از روش بهینه‌سازی محاسباتی مانند Linformer استفاده می‌شود که پیچیدگی زمانی را به $O(n \cdot \log(n) \cdot d)$ کاهش می‌دهد، تعداد عملیات نهایی را برای شرایطی که از KV Cache، Grouped Query Attention و محدودیت‌های سخت‌افزاری استفاده می‌شود، محاسبه کنید. آیا این بهینه‌سازی در شرایط خاص (مانند داده‌های بسیار طولانی) موثرتر از روش‌های قبلی خواهد بود؟ توضیح دهید چرا یا چرا نه.

تمام قسمت ها باید توسط خودتان پیاده سازی شود. در صورت استفاده از کدهای آماده و موجود در اینترنت نمره ای به شما تعلق نمی گیرد. در صورتی که در قسمت های غیر اصلی از کدهای موجود در اینترنت استفاده می کنید، باید منبع خود را ذکر کنید.

موفق باشید :)