به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین امتیازی

پرسش ۱	نام دستیار طراح	میلاد محمدی
	رایانامه	miladmohammadi@ut.ac.ir
	نام دستیار طراح	محمد اسدزاده
	رایانامه	mo.asadzadeh76@gmail.com
پرسش ۲	نام دستیار طراح	محمد سلمانی
	رایانامه	m.salmani@ut.ac.ir
	نام دستیار طراح	حامد خادمی خالدی
	رایانامه	Hamed.khaledi@ut.ac.ir
	مهلت ارسال پاسخ	۱۴۰۳/۱۰/۲۱

فهرست

شکل‌ها

قوانین

قبل از پاسخ دادن به پرسش‌ها، موارد زیر را با دقت مطالعه نمایید:

- از پاسخ‌های خود یک گزارش در قالبی که در صفحه‌ی درس در سامانه‌ی Elearn با نام ***REPORTS_TEMPLATE.docx*** قرار داده شده تهیه نمایید.
- پیشنهاد می‌شود تمرین‌ها را در قالب گروه‌های دو نفره انجام دهید. (بیش از دو نفر مجاز نیست و تحویل تک نفره نیز نمره‌ی اضافی ندارد) توجه نمایید الزامی در یکسان ماندن اعضای گروه تا انتهای ترم وجود ندارد. (یعنی، می‌توانید تمرین اول را با شخص A و تمرین دوم را با شخص B و ... انجام دهید)
- **کیفیت گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است**؛ بنابراین، لطفا تمامی نکات و فرض‌هایی را که در پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید در گزارش ذکر کنید.
- در گزارش خود مطابق با آنچه در قالب نمونه قرار داده شده، برای شکل‌ها زیرنویس و برای جدول‌ها بالانویس در نظر بگیرید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست، اما باید نتایج بدست آمده از آن را گزارش و تحلیل کنید.
- **تحلیل نتایج الزامی می‌باشد، حتی اگر در صورت پرسش اشاره‌ای به آن نشده باشد.**
- **دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند**؛ بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- **کدها حتما باید در قالب نوت‌بوک با پسوند .ipynb تهیه شوند، در پایان کار، تمامی کد اجرا شود و خروجی هر سلول حتما در این فایل ارسالی شما ذخیره شده باشد.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده‌اید، این نمودار باید هم در گزارش هم در نوت‌بوک کدها وجود داشته باشد.
- **در صورت مشاهده‌ی تقلب امتیاز تمامی افراد شرکت‌کننده در آن، 100- لحاظ می‌شود.**
- تنها زبان برنامه نویسی مجاز **Python** است.
- **استفاده از کدهای آماده برای تمرین‌ها به هیچ وجه مجاز نیست. در صورتی که دو گروه از یک منبع مشترک استفاده کنند و کدهای مشابه تحویل دهند، تقلب محسوب می‌شود.**

- نحوه محاسبه تاخیر به این شکل است: پس از پایان رسیدن مهلت ارسال گزارش، حداکثر تا یک هفته امکان ارسال با تاخیر وجود دارد، پس از این یک هفته نمره آن تکلیف برای شما صفر خواهد شد.
  - سه روز اول: بدون جریمه
  - روز چهارم: ۵ درصد
  - روز پنجم: ۱۰ درصد
  - روز ششم: ۱۵ درصد
  - روز هفتم: ۲۰ درصد
- حداکثر نمره‌ای که برای هر سوال می‌توان اخد کرد ۱۰۰ بوده و اگر مجموع بارم یک **سوال** بیشتر از ۱۰۰ باشد، در صورت اخد نمره بیشتر از ۱۰۰، اعمال نخواهد شد.
  - برای مثال: اگر نمره اخذ شده از سوال ۱ برابر ۱۰۵ و نمره سوال ۲ برابر ۹۵ باشد، نمره نهایی تمرین ۹۷.۵ خواهد بود و نه ۱۰۰.
- لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه‌ی Elearn بارگذاری نمایید:

HW[Number] _[Lastname]_[StudentNumber]_[Lastname]_[StudentNumber].zip

(مثال: HW1_Ahmadi_810199101_Bagheri_810199102.zip)

- برای گروه‌های دو نفره، بارگذاری تمرین از جانب یکی از اعضا کافی است ولی پیشنهاد می‌شود هر دو نفر بارگذاری نمایند.

پرسش ۱. تنظیم دقیق[۱] مدل‌های زبانی بزرگ[۲] برای گفتگو[۳] در زبان فارسی

در این تمرین با فرآیند *instruction fine-tuning* مدل‌های زبانی بزرگ (LLM) آشنا خواهید شد. هدف این تمرین، درک و پیاده‌سازی روش‌های مختلف تنظیم مدل بر روی یک مجموعه دادگان مکالمه‌ای زبان فارسی است. شما با سه رویکرد اصلی تنظیم مدل کار خواهید کرد و آزمایش‌های مرتبط را انجام خواهید داد.

توجه: در هیچ‌کدام از زیربخش‌های این تمرین مجاز به استفاده از ماژول‌ها، ابزارها و piplineهایی که به صورت کاملا آماده فرایند fine-tuning را انجام می‌دهند نیستید. کتابخانه‌ی مورد توصیه و مجاز انجام این تمرین PEFT و Transformers از HuggingFace است.

۱-۱. دادگان و انتخاب مدل

(۱۰ نمره)

معرفی دادگان slimOrca

دادگان SlimOrca نسخه‌ای کوچک‌تر و بهینه‌شده از مجموعه داده‌های OpenOrca است که به‌صورت کارآمد، عملکردی نزدیک به مجموعه‌های بزرگ‌تر ارائه می‌دهد. این مجموعه شامل حدود ۵۰۰ هزار نمونه مکالمه تکمیل‌شده توسط GPT-4 است.

ویژگی کلیدی این دادگان، یک مرحله اضافی پالایش با استفاده از GPT-4 است که در آن پاسخ‌هایی که بر اساس حاشیه‌نویسی انسانی نادرست تلقی می‌شوند، حذف شده‌اند. این فرآیند اندازه دادگان را کاهش داده و امکان دستیابی به کیفیتی مشابه نسخه‌های بزرگ‌تر با هزینه‌ی محاسباتی کمتر را فراهم کرده است.

برای این تمرین، مجموعه دادگانی از ترجمه‌ی انگلیسی به فارسی با استفاده از مدل GPT4o-mini از نسخه‌ی 50k از دادگان اصلی آماده شده و در اختیار شما قرار گرفته است.

---

[۱] fine-tuning

[۲] large language models

[۳] chat

بررسی فرمت دادگان

در این مرحله ابتدا باید دادگان ارائه‌شده را از طریق لینک داده شده از سایت HuggingFace بارگذاری کنید. با بررسی بخشی از داده‌ها، به ساختار کلی سؤالات و مکالمات موجود در دادگان پی ببرید. سپس تحلیل کنید که چرا فرمت کلی دادگان به این صورت طراحی شده و هدف از این طراحی چیست.

وظایف:

- دادگان را بارگذاری کنید و ساختار آن را بررسی کنید.
- نمونه‌هایی از داده‌ها را مشاهده کرده و نوع سؤالات و پاسخ‌ها را توصیف کنید.
- دلیل طراحی این فرمت کلی برای دادگان را تحلیل و توضیح دهید.

انتخاب مدل

مدلی که در این تمرین استفاده می‌شود، Gemma2 با اندازه ۲ میلیارد پارامتر است. این مدل زبانی بزرگ چندزبانه به‌صورت پیش‌آموزش‌یافته و تنظیم‌شده ارائه شده و شامل نسخه‌هایی با اندازه‌های ۲ میلیارد، ۹ میلیارد و ۲۷ میلیارد پارامتر است. برای استفاده از این مدل، ابتدا باید در سایت Hugging Face درخواست دسترسی دهید.

وظایف:

- بیان کنید دو نسخه‌ی base و instruct چه تفاوتی باهم دارند؟
- بین دو نسخه‌ی توضیح داده شده یک مدل را انتخاب و دلیل خود را شرح دهید.

همچنین شما مجاز هستید که بجای Gemma2-2b از مدل Llama3.2-3B استفاده کنید .در انتخاب مدل بین این دو مدل کاملا مختار هستید.

آماده‌سازی دادگان

پس از انتخاب مدل، توکنایزر مرتبط با آن را دانلود و نصب کنید. سپس دادگان را پردازش کرده و به فرمت مناسب برای آموزش مدل تبدیل کنید. این کار باید به‌گونه‌ای انجام شود که داده‌ها با مدل و توکنایزر انتخاب‌شده سازگار باشند.

وظایف:

- توکنایزر مدل انتخابی را دانلود و نصب کنید.
- دادگان را پردازش کرده و فرمت آن را برای آموزش مدل آماده کنید.

تست مدل

پس از آماده‌سازی مدل و دادگان، در این مرحله مدل انتخابی را بر روی چند ورودی تست کنید تا با عملکرد اولیه آن آشنا شوید. این تست‌ها در بخش‌های بعدی نیز مورد استفاده قرار خواهند گرفت، بنابراین سؤالاتی را انتخاب کنید که بتوانند عملکرد مدل را به خوبی سنجش کنند.

وظایف:

- چند نمونه تست (ورودی/خروجی) به زبان فارسی بنویسید.
- مدل را بر روی نمونه‌های طراحی شده اجرا و خروجی‌ها را چاپ کنید.
- خروجی‌های فعلی مدل را مورد ارزیابی و تحلیل قراردهید.

۱-۲. روش‌های SoftPrompts

(۳۵ نمره)

معرفی روش‌های SoftPrompts

در این بخش، با مفهوم Soft Prompts و کاربردهای آن آشنا خواهید شد. همچنین سه روش اصلی این رویکرد شامل Prompt Tuning، Prefix Tuning و P-Tuning به‌صورت مختصر بررسی می‌شوند. در پایان، یکی از این روش‌ها را انتخاب کرده و دلیل انتخاب خود را بیان خواهید کرد.

وظایف:

- درباره مفهوم کلی Soft Prompts تحقیق کرده و توضیح مختصری بنویسید.
- سه روش Prompt Tuning، Prefix Tuning و P-Tuning را به‌صورت خلاصه توضیح دهید.
- یکی از این روش‌ها را انتخاب کرده و دلایل انتخاب خود را توضیح دهید.

معرفی روش‌های SoftPrompts

در این بخش، با استفاده از روش انتخابی خود، مدل را با کمک کتابخانه PEFT در Hugging Face Fine-Tune خواهید کرد.

وظایف:

- محیط کدنویسی خود را برای استفاده از کتابخانه PEFT آماده کنید.
- با استفاده از روش انتخابی، مدل را بر روی دادگان ارائه‌شده آموزش دهید.
- اطمینان حاصل کنید که تنها پارامترهای مربوط به پرامپت‌ها در حال به‌روزرسانی هستند و پارامترهای اصلی مدل بدون تغییر باقی می‌مانند.
- فرآیند آموزش را مستند کرده و چالش‌ها یا مشکلاتی که با آنها مواجه شدید را توضیح دهید.
- نمودارهایی از فرآیند یادگیری، مانند نمودار خطای آموزش و اعتبارسنجی، تهیه کرده و آنها را تحلیل کنید.

ارزیابی مدل پس از آموزش

پس از پایان آموزش، از مجموعه داده‌های تست برای ارزیابی مدل استفاده کنید. هدف این بخش، بررسی تأثیر روش انتخابی بر عملکرد مدل است و مشخص می‌شود که آیا خروجی‌های مدل دقیق‌تر یا مرتبط‌تر شده‌اند.

وظایف:

- مدل آموزش‌دیده را با استفاده از ورودی‌های تست ارزیابی کنید.
- خروجی‌های جدید مدل را با خروجی‌های اولیه مقایسه کنید.
- تحلیل کنید که آیا پاسخ‌های مدل دقیق‌تر، کامل‌تر یا مرتبط‌تر شده‌اند.
- نتایج را مستند کرده و بهبودهای مشاهده‌شده را توضیح دهید.

۱-۳. روش‌های مبتنی بر LoRA

(۳۵ نمره)

معرفی روش LoRA

در این بخش، با روش LoRA (Low-Rank Adaptation) و کاربردهای آن آشنا خواهید شد LoRA مانند روش SoftPrompts یک روش Parameter-Efficient Fine-Tuning (PEFT) است که با تجزیه ماتریس‌های بزرگ لایه‌های توجه به دو ماتریس کوچک با رتبه پایین، تعداد پارامترهایی را که نیاز به تنظیم دارند، به‌طور چشمگیری کاهش می‌دهد. این روش کارایی محاسباتی را افزایش داده و منابع مورد نیاز برای Fine-Tuning را کاهش می‌دهد.

وظایف:

- درباره روش LoRA تحقیق کرده و توضیح مختصری بنویسید.
- توضیح دهید LoRA بر روی کدام لایه‌ها و اجزای مدل زبانی بزرگ ترنسفرمری باید اعمال شود؟
- **بخش امتیازی (۵ نمره امتیازی):** دیگر متدهای مبتنی بر LoRA مانند DoRA، LoHa، و RsLoRA را مطالعه کنید. در صورت تمایل، می‌توانید به جای LoRA یکی از این متدها را (به شرطی که توسط کتابخانه PEFT پشتیبانی شود) برای این بخش از تمرین (کل بخش ۳) انتخاب کنید. اگر متد دیگری را انتخاب کردید، دلایل انتخاب خود را توضیح داده و بیان کنید که چرا این روش را به LoRA ترجیح می‌دهید.

توجه: صرفا در صورتی که با خطاهای مربوط به حافظه پردازشی مواجه شدید می‌توانید بجای روش LoRA از روش QloRA استفاده کنید.

آموزش مدل با استفاده از روش انتخابی

در این بخش، مدل خود را با استفاده از روش انتخابی (مانند *LoRA* یا متد جایگزین) با استفاده از کتابخانه PEFT در Hugging Face آموزش خواهید داد.

وظایف:

- با استفاده از روش انتخابی مدل را بر روی دادگان ارائه‌شده آموزش دهید.
- اطمینان حاصل کنید که تنها پارامترهای مربوط به لایه‌های انتخابی که در زیربخش قبل توضیح دادید اعمال می‌شود.
- فرآیند آموزش را به‌صورت کامل مستند کنید و هرگونه چالش یا مسئله‌ای که با آن مواجه شدید را توضیح دهید.
- نمودارهایی از فرآیند یادگیری، مانند نمودار خطای آموزش تهیه کنید و آنها را تحلیل کنید.

ارزیابی مدل پس از آموزش

به مانند زیربخش پایانی بخش ۲، پس از پایان آموزش، از مجموعه داده‌های تست برای ارزیابی مدل استفاده کنید. هدف این بخش، بررسی تأثیر روش انتخابی بر عملکرد مدل و مقایسه با نتایج بخش ۲ است.

وظایف:

- مراحلی که در زیربخش ارزیابی مدل بخش ۲ انجام شد را تکرار کنید.
- نتایج آموزش مدل در این بخش را با بخش ۲ مقایسه و تحلیل کنید.

۱-۴. تغییر وزن برخی از لایه‌ها

(۱۵ نمره)

در این بخش، با روش‌های سنتی‌تر *Fine-Tuning* مدل‌ها که در تمرینات قبلی درس نیز با آن‌ها آشنا شده‌اید، کار خواهید کرد. برخلاف بخش‌های پیشین، در این تمرین از کتابخانه PEFT استفاده نخواهید کرد. می‌توانید با استفاده از کتابخانه‌ی transformers شرکت HuggingFace استفاده کنید. (همچنین می‌توانید از ابزارهای پایه مانند PyTorch یا TensorFlow این بخش از تمرین را انجام دهید.)

ارزیابی مدل پس از آموزش

ابتدا ساختار مدل انتخابی خود را بررسی و استخراج کنید. سپس، تنها دو لایه‌ی اول و آخر مدل را برای Fine-Tuning تنظیم کنید (Unfreeze) و دیگر لایه‌ها را قفل (Freeze) کنید تا تغییر نکنند.

وظایف:

- با دستور کد مناسب، ساختار و معماری مدل را استخراج کرده و به‌صورت خلاصه توضیح دهید.
- دو لایه‌های اول و آخر مدل را باز کنید و دیگر لایه‌ها را قفل کنید.
- آموزش مدل را انجام دهید.
- فرآیند آموزش را مستند کنید و چالش‌ها یا مشکلاتی که با آن مواجه شدید را توضیح دهید.
- نمودارهایی از فرآیند یادگیری (مانند نمودار خطای آموزش) تهیه کرده و تحلیل کنید.

ارزیابی مدل پس از آموزش

پس از اتمام فرایند آموزش مانند زیربخش آخر بخش ۲ و ۳ مدل را ارزیابی کنید.

۱-۵. جمع‌بندی و تحلیل مقایسه‌ای

(۵ نمره)

در این بخش، روش‌های مختلف تنظیم مدل را از جنبه‌های مختلف تحلیل و مقایسه خواهید کرد. این تحلیل‌ها شامل میزان منابع مصرفی، عملکرد و نتایج حاصل از هر روش است. در پایان، نتیجه‌گیری کلی خود را ارائه دهید و بهترین روش را با توجه به نتایج مشخص کنید.

وظایف:

- روش‌های مختلف را از نظر منابع مورد نیاز (زمان آموزش، حافظه مصرفی، تعداد پارامترهای قابل آموزش و غیره) مقایسه کنید.
- عملکرد هر روش را بر اساس نتایج به‌دست‌آمده تحلیل کنید.
- مزایا و معایب هر روش را بیان کنید.
- بر اساس تحلیل‌های انجام‌شده، بهترین روش را با توجه به شرایط خاص تمرین معرفی و نتیجه‌گیری کنید.

پرسش ۲ – تولید کپشن برای تصاویر (Image Captioning)

۲-۱. مقدمه

Image-captioning یک تکنولوژی در حوزه پردازش تصویر و یادگیری ماشین است که به سیستم‌ها این امکان را می‌دهد تا به صورت خودکار توضیحاتی متنی برای تصاویر تولید کنند. این فرایند معمولاً شامل تحلیل ویژگی‌های بصری تصویر توسط مدل‌های یادگیری عمیق است که سپس این ویژگی‌ها را به جملات قابل فهم ترجمه می‌کنند. مدل‌های image-captioning اغلب از شبکه‌های عصبی پیچیده مانند شبکه‌های عصبی کانولوشنی (CNN) برای استخراج ویژگی‌ها و شبکه‌های عصبی بازگشتی (RNN) یا ترنسفورمرها برای تولید توضیحات متنی استفاده می‌کنند. این تکنولوژی در کاربردهایی مانند دسترسی به محتوای تصویری برای افراد نابینا، جستجوی تصاویر و تجزیه و تحلیل محتوا در شبکه‌های اجتماعی مفید است.

در این پروژه، هدف تولید کپشن‌های متنی برای تصاویر موجود در دیتاست Flickr8k با استفاده از مدل‌های متنوع encoder-decoder است. این فرآیند شامل دو مرحله اصلی است:

- استخراج ویژگی‌های بصری تصاویر با کمک مدل‌های پیشرفته.
- تبدیل این ویژگی‌ها به توضیحات متنی معنادار با استفاده از شبکه‌های عصبی.

۲-۲. آماده سازی دیتاست

(۲۰ نمره)

انتخاب مجموعه داده

- ابتدا دیتاست flicker 8k را دانلود کرده و چند نمونه تصویر را به همراه متن متناظر آن نمایش دهید.

پیش‌پردازش تصاویر

- اندازه تصاویر را به ابعاد ثابت و مناسب تغییر دهید تا با ورودی مدل CNN سازگار باشند.
- نرمال‌سازی (Normalization):
  - مقادیر پیکسل تصاویر را به محدوده دلخواه نرمال‌سازی کنید.
  - میتوانید از میانگین و انحراف معیار استاندارد استفاده کنید. تا ویژگی‌های استخراج‌شده توسط CNN بهینه شوند.

پیش‌پردازش متن (Captions)

- تمام متن‌ها را به حروف کوچک تبدیل کنید تا حساسیت به حروف بزرگ حذف شود.
- می‌توانید علائم نگارشی، نمادهای خاص و اعداد غیرضروری را نیز حذف کنید.
- تبدیل کلمات به شناسه عددی (Tokenization):
  - هر کلمه را به یک شناسه عددی منحصربه‌فرد تبدیل کنید.
  - یک دیکشنری بسازید که هر کلمه را به یک عدد نگاشت کند.
  - توجه داشته باشید که special tokenهای مورد نیاز را نیز به این دیکشنری اضافه نمایید. (مانند <pad>، <sos>، <eos> و <unk>) کاربرد هرکدام را بیان کنید.
  - دیکشنری را در یک فایل Json ذخیره کنید. در ادامه سوالات لازم است از این فایل به عنوان tokenizer استفاده کنید.
- طول ثابت برای کپشن‌ها:
  - توضیح دهید که چرا باید کپشن‌ها را به طول ثابت تبدیل کنیم.
  - اگر طول کپشن کمتر از مقدار ثابت تعیین‌شده باشد، از توکن <pad> برای پر کردن فضای خالی استفاده کنید.

تقسیم داده‌ها

- دیتاست را به نسبت ۱۰/۱۰/۸۰ تقسیم‌بندی کنید تا به سه مجموعه آموزش (Train)، اعتبارسنجی (Validation)، و تست (Test) دست پیدا کنید. توجه داشته باشید که این تقسیم‌بندی باید بر اساس عکس‌ها انجام شود، به‌طوری‌که هیچ عکس تکراری میان این مجموعه‌ها وجود نداشته باشد.

نمایش داده‌های پردازش‌شده

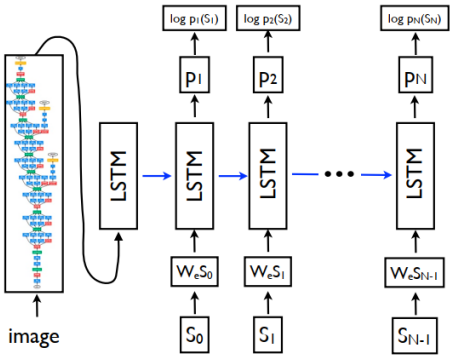
- برای بررسی دیتاست، به‌صورت تصادفی ۵ عکس را انتخاب کرده و همراه با یکی از کپشن‌های مربوط به آن‌ها نمایش دهید.
- نمودار پراکندگی (Scatter Plot) طول کپشن‌ها را رسم کنید تا تنوع طول توصیف‌ها در دیتاست مشخص شود.
- در نهایت، یک هیستوگرام از ۲۰ کلمه پرتکرار در تمامی کپشن‌ها ایجاد کرده و نمایش دهید.

۲-۳. پیاده‌سازی CNN-RNN

(۲۵ نمره)

مدل‌های CNN-RNN یکی از روش‌های برجسته در زمینه یادگیری عمیق هستند که برای حل مسائل Multimodal نظیر توضیح تصاویر (Image Captioning) به کار می‌روند. در این معماری، از شبکه عصبی CNN به‌عنوان رمزگذار (Encoder) برای استخراج ویژگی‌های بصری تصویر استفاده می‌شود و سپس شبکه عصبی بازگشتی (RNN) یا مدل‌های پیشرفته‌تر آن (مانند LSTM یا GRU) به‌عنوان رمزگشا (Decoder) برای تولید متون توصیفی به کار گرفته می‌شود. در این بخش میخواهیم معماری CNN-RNN را با استفاده از پارامترها و طراحی‌های الهام گرفته از این مقاله[1] پیاده‌سازی کنیم.

[1] Show and Tell: A Neural Image Caption Generator



شکل ۱- معماری مدل **CNN-RNN**

طراحی مدل

پیاده سازی بخش رمزگذار (Encoder):

- از یک مدل CNN از پیش‌آموزش‌داده‌شده استفاده کنید. (مانند مدل EfficientNet-B0)
- لایه Fully Connected آخر را حذف کنید تا بردار ویژگی‌های تصویر استخراج شود.
- ابعاد خروجی رمزگذار را بررسی کنید.

پیاده سازی بخش رمزگشا (Decoder):

- از یک لایه embedding برای کلمات استفاده کنید. دلیل این کار را توضیح دهید. چرا استفاده از تعبیه کلمات (Word Embedding) به‌جای بردارهای One-hot مناسب‌تر است؟
- از LSTM برای تولید کلمات استفاده کنید.
- بردار ویژگی تصویر به‌عنوان حالت اولیه به LSTM داده شود.
- از یک لایه Linear با softmax برای پیش‌بینی کلمه بعدی استفاده کنید.

اتصال رمزگذار و رمزگشا (Encoder-Decoder):

- چگونه رمزگذار و رمزگشا را به یک مدل End-to-End تبدیل کنیم که قابلیت آموزش داشته باشد؟ یک کلاس سفارشی ImageCaptioningModel بنویسید که هر دو کامپوننت را یکپارچه کند.

آموزش مدل

- از تابع هزینه مناسب برای محاسبه خطا استفاده کنید و padding را در محاسبه هزینه در نظر نگیرید.
- در بخش Encoder، همه یا بجز چند لایه آخر CNN را ثابت نگه دارید.

- بررسی کنید مقاله چه تکنیک‌هایی برای جلوگیری از بیش‌برازش (Overfitting) را بکار می‌گیرد؟

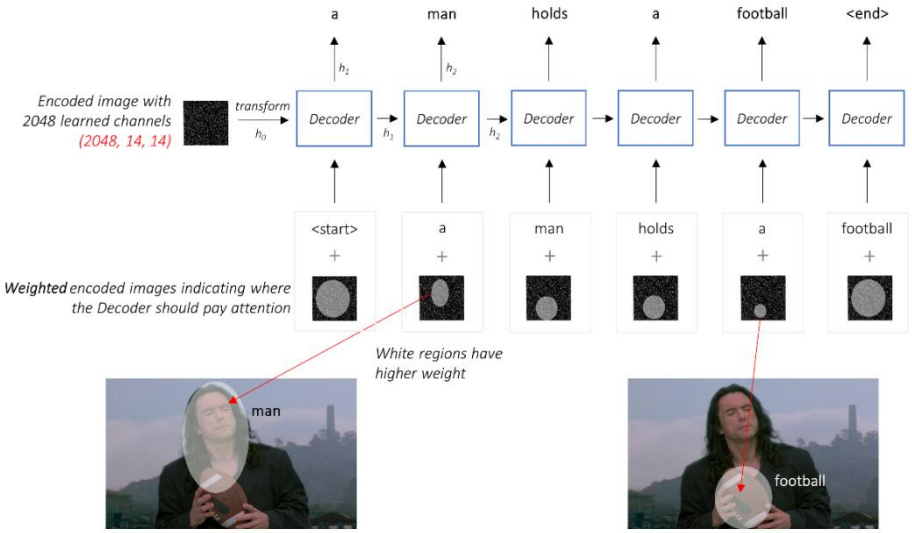
نکته: در صورت نیاز به وقفه در طول آموزش، پیشنهاد می‌شود نقاط بازیابی (Checkpoints) را برای ادامه آموزش ذخیره کنید.

ارزیابی مدل

- نمودار خطای داده آموزش و ارزیابی را در طول هر دوره (Epoch) گزارش کنید.
- می‌توانید در پایان هر دوره یک نمونه تصویر و خروجی آن را نمایش دهید.
- در پایان آموزش، ۵ تا از تصاویر و توضیحات تولیدشده آن‌را در کنار یکدیگر نمایش دهید.
- برخی از خطاهای مدل را شناسایی و مشخص کنید (مانند عدم تشخیص اشیاء یا روابط)

۲-۴. پیاده‌سازی Attention based CNN-RNN

(۳۰ نمره)



شکل ۲- مکانیزم **attention**

اضافه کردن مکانیزم توجه به معماری CNN-RNN یک گام پیشرفته است که به مدل اجازه می‌دهد در هنگام تولید هر کلمه از توضیح تصویر، روی بخش‌های خاصی از تصویر تمرکز کند. این روش به بهبود دقت و کیفیت جملات تولیدشده کمک می‌کند. در واقع به جای استفاده از یک بردار ثابت برای ویژگی‌های تصویر، مکانیزم توجه یک بردار وزن‌دار تولید می‌کند که مناطق مهم تصویر را برجسته می‌سازد. در این بخش میخواهیم معماری CNN-RNN با مکانیزم توجه را با استفاده از پارامترها و طراحی‌های الهام گرفته از این مقاله[1] پیاده‌سازی کنیم.

طراحی مدل

پیاده سازی بخش رمزگذار (Encoder):

- از مدل CNN مشابه که در قسمت قبلی استفاده شده است بهره ببرید.
- در این بخش نیاز است که ویژگی‌های تصویر شامل اطلاعات مکانی نیز باشند به همین دلیل علاوه بر لایه های Fully Connected، لایه‌های Pooling آخر نیز حذف می‌شوند.
- خروجی رمزگذار را بررسی کرده و ابعاد آن را گزارش دهید.

[1] Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

پیاده سازی مکانیزم توجه (Attention)

- مکانیزم توجه بر اساس ترکیب اطلاعات خروجی رمزگذار (Encoder) و حالت مخفی رمزگشا (Decoder) عمل می‌کند. به اینصورت که برای هر منطقه از تصویر، وزنی محاسبه می‌شود که نشان می‌دهد آن منطقه تا چه حد برای تولید کلمه جاری مهم است.
- فرمول محاسبه وزن:

$$e_t^i = f(W_h h_{t-1}, W_a a_i)$$

  - h_{t-1}: حالت مخفی RNN در گام قبلی.
  - a_i: ویژگی‌های منطقه i-ام از نقشه ویژگی.

- وزن‌های نرمال‌شده:

$$\alpha_t^i = softmax(e_t^i)$$

- ویژگی‌های وزن‌دار تصویر:

$$z_t = \sum_t \alpha_t^i . a_i$$

پیاده سازی بخش رمزگشا (Decoder)

- در هر گام بردار وزن‌دار z_t به همراه بردار embedding کلمه قبلی به LSTM داده می‌شود.

$$input_t = concat(x_t, z_t)$$

نکته: برای تبدیل کردن ابعاد $input_t$ به ابعاد مورد نظر، می‌توانید آن را از یک لایه خطی نیز عبور دهید.

- به‌روزرسانی وضعیت LSTM:

$$h_t, c_t = LSTM\big(input_t, (h_{t-1}, c_{t-1})\big)$$

- پیش‌بینی کلمه بعدی: h_t به یک لایه خطی (Fully Connected) داده می‌شود تا احتمال کلمات موجود در دیکشنری محاسبه شود:

$$p_t = softmax(W\ h_t + b)$$

- انتخاب کلمه بعدی: کلمه با بالاترین احتمال را انتخاب کرده و به‌عنوان ورودی گام بعدی به مدل داده بدهید.

آموزش مدل

- مشابه قسمت قبل از تابع هزینه مناسب برای محاسبه خطا استفاده کرده و padding را در محاسبه هزینه در نظر نگیرید.
- مدل را با تعداد Epochهای یکسان با مرحله قبل آموزش دهید.

ارزیابی مدل

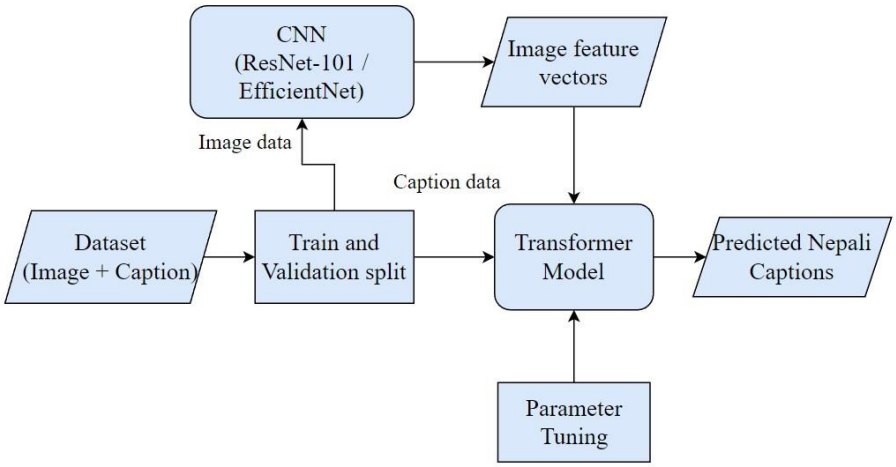
- نمودار خطای داده آموزش و ارزیابی را در طول هر دوره (Epoch) گزارش کنید.
- می‌توانید در پایان هر دوره یک نمونه تصویر و خروجی آن را نمایش دهید.
- در پایان آموزش، ۵ تا از تصاویر و توضیحات تولیدشده آن‌را در کنار یکدیگر نمایش دهید.
- برای یک نمونه خروجی تولید شده از داده تست، نقشه حرارتی (Heatmap) وزن‌های توجه در هر مرحله از تولید کلمه را رسم کنید تا ببینید مدل روی کدام بخش‌های تصویر تمرکز کرده است. آیا مدل در تمام مراحل تولید کلمه، به بخش‌های منطقی تصویر توجه کرده است؟
- برخی از خطاهای مدل را شناسایی و مشخص کنید و با قسمت قبل مقایسه نمایید.

۲-۵. پیاده‌سازی CNN-Transformer

(۲۵ نمره)

هدف این بخش، پیاده‌سازی سیستم تولید کپشن برای تصاویر با استفاده از معماری Transformer، مشابه مقاله[1] مرجع است. روند کلی این روش در شکل زیر قابل مشاهده است. یک فایل notebook جهت راهنمایی آماده‌سازی شده است که در صورت نیاز می‌توانید از آن بهره ببرید.

[1] CNN-Transformer based Encoder-Decoder Model for Nepali Image Captioning



شکل ۳- مراحل تولید متن از عکس

پیاده سازی بخش Tokenizer

- برای این مرحله، ابتدا باید به tokenizer پیاده‌سازی‌شده در بخش‌های قبل قابلیت masking اضافه کنید. برای توکن‌هایی که در ادامه جمله اضافه می‌شوند، مقدار mask را برابر True قرار دهید. سپس در گزارش توضیح دهید که masking کلمات چه تأثیری در فرآیند آموزش دارد؟

پیاده سازی بخش رمزگذار (Encoder)

- Encoder را با استفاده از مدلی مانند EfficientNet-B0 پیاده‌سازی کنید. مطمئن شوید که در هنگام آموزش، لایه‌های این مدل فریز شده باشند و وزن‌های آن تغییر نکنند. و فقط لایه آخر آن را تغییر دهید تا ویژگی‌های تصویر با اندازه مطلوب در خروجی به دست بیاید.

پیاده سازی بخش رمزگشا (Decoder)

- هدف از این بخش پیاده سازی attention بین کلمات و بردار استخراج شده از تصویر است. برای اینکار بخش Decoder را با استفاده از لایه Transformer موجود در PyTorch پیاده‌سازی کنید. این قسمت نیازمند استفاده از Word Embedding و Positional Embedding است. در گزارش به‌طور خلاصه توضیح دهید که Positional Embedding چه نقشی در مدل ایفا می‌کند؟

آموزش و ارزیابی مدل

- با توجه به تعداد زیاد پارامترهای مدل، برای دستیابی به نتایج مطلوب، لازم است که مدل را برای تعداد Epoch کافی آموزش دهید. برای تسریع آموزش و بهره‌مندی از سخت‌افزار مناسب می‌توانید از پلتفرم‌هایی مانند Kaggle استفاده کنید.
- همچنین در صورت نیاز به ارزیابی‌های بیشتر مطمئن شوید که مدل را پس از انجام آموزش ذخیره نموده باشید.
- پس از اتمام آموزش، تغییرات Loss را برای داده‌های آموزش و اعتبارسنجی در قالب یک نمودار رسم کنید. سپس برای ۵ عکس تصادفی از مجموعه داده تست، کپشن تولید کنید و نتایج را بررسی کنید.

۲-۶. بخش امتیازی

(۱۰ نمره)

- تحقیق کنید چه معیارهایی برای ارزیابی این مدل‌ها بکار می‌رود.
- مطالعه‌ای درباره Bleu Score انجام داده و به‌طور مختصر توضیح دهید که این معیار چگونه عملکرد مدل را ارزیابی می‌کند. سپس Bleu Score (از یک تا چهار) را روی داده‌های تست محاسبه کرده و نتایج آن را برای بخش‌های مختلف گزارش کنید و مقایسه‌ای انجام دهید.