Neural Networks and Deep Learning **Final Project**

Instructors: **Dr. Bahrak** TA: محمد امانلو

Deadline: دهم شهریورماه

سوال اول: آشنایی با Fine-Tuning دستورمحور (Instruction Fine-Tuning) در مدل‌های زبانی بزرگ

این تمرین با هدف یادگیری و درک عمیق فرآیند تنظیم دقیق مدل‌های زبانی بزرگ (Large Language Models - LLM) با رویکرد دستورمحور طراحی شده است. در این فرآیند، مدل نه صرفاً برای پیش‌بینی متن بعدی، بلکه برای پاسخ‌گویی دقیق و هماهنگ با نیازهای خاص کاربر آموزش داده می‌شود. شما در طول این تمرین با سه روش مهم برای تنظیم مدل آشنا می‌شوید، هر کدام را پیاده‌سازی می‌کنید و نتایج آنها را مقایسه خواهید کرد.

نکته مهم: در این تمرین اجازه استفاده از ابزارها یا Pipeline های آماده که به‌صورت کامل فرآیند fine-tuning را انجام می‌دهند ندارید. کتابخانه‌های مجاز برای استفاده، Transformers و PEFT از HuggingFace هستند.

بخش اول – داده‌ها و انتخاب مدل (۱۰ نمره)

معرفی داده‌ها: SlimOrca

SlimOrca نسخه‌ای کوچک و بهینه‌شده از مجموعه داده OpenOrca است. این مجموعه شامل حدود ۵۰۰ هزار نمونه مکالمه است که توسط GPT-4 تولید شده‌اند. تفاوت مهم SlimOrca با نسخه اصلی این است که یک مرحله پالایش اضافه روی آن انجام شده است، به این شکل که پاسخ‌های نادرست یا کم‌کیفیت که توسط حاشیه‌نویسی انسانی مشخص شده‌اند، حذف شده‌اند. این کار باعث می‌شود حجم داده کاهش یابد ولی کیفیت حفظ شود.

برای این تمرین، نسخه‌ای شامل ۵۰ هزار نمونه از SlimOrca، با ترجمه فارسی تولید شده توسط مدل GPT4o-mini، در اختیار شما قرار داده شده است. این داده‌ها مکالمه‌محور هستند و قالب خاصی برای پرسش و پاسخ دارند.

وظایف این بخش:

۱. داده‌ها را از لینک HuggingFace بارگذاری کنید.

۲. چند نمونه از داده‌ها را مشاهده کنید و ساختار پرسش و پاسخ را توضیح دهید.

۳. تحلیل کنید که چرا این قالب داده انتخاب شده و چه مزایایی دارد.

انتخاب مدل:

مدل اصلی این تمرین Gemma2 با ۲ میلیارد پارامتر است. این مدل چندزبانه بوده و دو نسخه Base و

Instruct دارد.
نسخه Base صرفاً پیش‌آموزش داده شده است و دانش عمومی دارد، اما نسخه Instruct علاوه بر پیش‌آموزش، روی داده‌های دستورمحور نیز تنظیم شده و توانایی بهتری در دنبال کردن دستورات دارد.
شما می‌توانید بین Gemma2-2B و Llama3.2-3B یکی را انتخاب کنید. (در سایت هاگینگ فیس درخواست دسترسی بدهید.)

وظایف انتخاب مدل:
۱. تفاوت نسخه‌های Base و Instruct را توضیح دهید.
۲. یک نسخه را انتخاب کرده و دلیل انتخاب خود را بیان کنید.

آماده‌سازی داده‌ها:
۱. توکنایزر مدل انتخابی را دانلود و نصب کنید.
۲. داده‌ها را به قالبی که با مدل و توکنایزر سازگار باشد، پردازش کنید. این شامل حذف یا تغییر بخش‌های اضافی و اطمینان از هماهنگی ورودی و خروجی است.

تست اولیه مدل:
۱. چند نمونه پرسش و پاسخ به زبان فارسی طراحی کنید که بتوانند توانایی مدل را ارزیابی کنند.
۲. این نمونه‌ها را روی مدل اجرا کنید و خروجی را مشاهده نمایید.
۳. خروجی‌ها را تحلیل کنید و نقاط قوت و ضعف مدل را قبل از آموزش ثبت کنید.

بخش دوم – روش‌های Soft Prompts (۳۵ نمره)

معرفی Soft Prompts:
در روش Soft Prompt، به جای تغییر وزن‌های اصلی مدل، یک مجموعه پارامتر ورودی (پرومپت) قابل یادگیری تعریف می‌شود که در ابتدای توالی ورودی اضافه می‌گردد. این پارامترها در طول آموزش به‌روزرسانی می‌شوند و باعث تغییر رفتار مدل می‌گردند، بدون اینکه نیاز باشد کل مدل دوباره آموزش داده شود. این روش از نظر منابع محاسباتی بسیار کارآمدتر از Fine-Tuning کامل مدل است.

سه روش اصلی:
۱. Prompt Tuning – اضافه کردن تعدادی توکن قابل یادگیری به ابتدای ورودی.
۲. Prefix Tuning – اضافه کردن بردارهای قابل یادگیری به تمام لایه‌های ترنسفورمر به عنوان پیشوند.
۳. P-Tuning – نسخه پیشرفته‌تر که توکن‌های یادگیری را با مکانیزم embedding به مدل اضافه می‌کند.

وظایف:
۱. یک توضیح مختصر درباره Soft Prompts ارائه دهید.
۲. سه روش فوق را توضیح دهید.
۳. یکی را انتخاب کرده و دلیل انتخاب خود را بنویسید.

آموزش با روش انتخابی:
۱. محیط کدنویسی را برای استفاده از کتابخانه PEFT آماده کنید.
۲. مدل را با روش انتخابی روی داده‌های فارسی آموزش دهید.

۳. مطمئن شوید فقط پارامترهای مربوط به پرومپت‌ها تغییر کنند و باقی مدل ثابت بماند.
۴. روند آموزش، چالش‌ها، و مشکلات را مستند کنید.
۵. نمودار خطای آموزش و اعتبارسنجی را رسم کرده و تحلیل کنید.

ارزیابی پس از آموزش:
۱. مدل آموزش‌دیده را روی داده تست اجرا کنید.
۲. خروجی‌های جدید را با خروجی‌های اولیه مقایسه کنید.
۳. بهبودها و تغییرات را توضیح دهید.

بخش سوم – روش‌های مبتنی بر LORA (۳۵ نمره)

معرفی :LoRA
LoRA یا Low-Rank Adaptation یکی از روش‌های PEFT است که با کاهش رتبه ماتریس‌های بزرگ در لایه‌های توجه، تعداد پارامترهای قابل آموزش را کم می‌کند. به جای تغییر کل ماتریس، دو ماتریس کوچک‌تر آموزش داده می‌شود که حاصل‌ضرب آنها اثر مشابه دارد. این باعث صرفه‌جویی در حافظه و زمان می‌شود.

وظایف:
۱. LoRA را توضیح دهید.
۲. مشخص کنید LoRA باید روی کدام لایه‌ها اعمال شود.
۳. در صورت تمایل می‌توانید از روش‌های مشابه مانند LoHa، DoRA، یا RsLoRA استفاده کنید (در صورتی که PEFT پشتیبانی کند). اگر روش دیگری انتخاب کردید، دلیل خود را توضیح دهید.

آموزش مدل با LoRA یا روش جایگزین:
۱. مدل را با روش انتخابی و PEFT روی داده‌ها آموزش دهید.
۲. اطمینان حاصل کنید که فقط لایه‌های مشخص‌شده به‌روزرسانی می‌شوند.
۳. فرآیند آموزش و چالش‌ها را مستند کنید.
۴. نمودار خطای آموزش را رسم و تحلیل کنید.

ارزیابی پس از آموزش:
همانند بخش قبل، مدل را ارزیابی کنید و نتایج را با روش Soft Prompts مقایسه نمایید.

بخش چهارم – تغییر وزن برخی لایه‌ها (۱۵ نمره)

در این بخش از روش سنتی‌تر Fine-Tuning استفاده می‌کنید و کتابخانه PEFT استفاده نمی‌شود. شما باید با استفاده از Transformers یا ابزارهای پایه مثل PyTorch یا TensorFlow، تنها دو لایه اول و دو لایه آخر مدل را Unfreeze کرده و بقیه را Freeze کنید.

وظایف:
۱. ساختار مدل را استخراج و توضیح مختصر دهید.
۲. لایه‌های مشخص‌شده را آزاد کنید.
۳. مدل را آموزش دهید.

۴. روند آموزش و چالش‌ها را مستند کنید.
۵. نمودار خطای آموزش را رسم و تحلیل کنید.

ارزیابی:
پس از آموزش، مدل را مانند مراحل قبل ارزیابی کنید.

بخش پنجم – جمع‌بندی و تحلیل مقایسه‌ای (۵ نمره)

در این بخش، نتایج سه روش را مقایسه می‌کنید. معیارها شامل زمان آموزش، حافظه مصرفی، تعداد پارامترهای آموزش‌دیده، کیفیت خروجی، و میزان بهبود نسبت به مدل اولیه است.

وظایف:
۱. روش‌ها را از نظر منابع محاسباتی مقایسه کنید.
۲. عملکرد هر روش را تحلیل کنید.
۳. مزایا و معایب هر روش را بیان کنید.
۴. بهترین روش را با توجه به شرایط تمرین معرفی کنید.

پرسش 2 - تولید کپشن برای تصاویر

تکنولوژی Captioning یکی از شاخه‌های مهم پردازش تصویر و یادگیری ماشین است که به سیستم‌ها این توانایی را می‌دهد تا به‌طور خودکار توضیحات متنی معنادار برای تصاویر تولید کنند. این فرآیند معمولاً با تحلیل ویژگی‌های بصری تصویر توسط مدل‌های یادگیری عمیق آغاز می‌شود. ابتدا مدل ویژگی‌های مهم تصویر را شناسایی می‌کند، سپس این ویژگی‌ها به کمک یک مدل زبانی به جملات قابل فهم تبدیل می‌شوند.
مدل‌های Image Captioning اغلب ترکیبی از شبکه‌های عصبی کانولوشنی (CNN) برای استخراج ویژگی‌های تصویری و شبکه‌های عصبی بازگشتی (RNN) یا معماری‌های پیشرفته‌تر مانند ترنسفورمرها برای تولید متن هستند. این فناوری کاربردهای گسترده‌ای دارد، از جمله کمک به افراد نابینا در دسترسی به محتوای بصری، بهبود سیستم‌های جستجوی تصویر و تحلیل محتوای شبکه‌های اجتماعی.
در این پروژه هدف ما تولید توضیحات متنی برای تصاویر دیتاست Flickr8k است، با استفاده از مدل‌های متنوع Encoder-Decoder. این کار شامل دو گام اصلی است:
۱. استخراج ویژگی‌های بصری تصویر با کمک مدل‌های پیشرفته.
۲. تبدیل این ویژگی‌ها به جملات توصیفی با استفاده از شبکه‌های عصبی.

بخش اول: آماده‌سازی دیتاست (۲۰ نمره)

برای آموزش مدل، ابتدا باید داده‌ها را به شکل مناسبی آماده کنید. در این مرحله تصاویر و توضیحات متنی باید پیش‌پردازش شوند تا بتوانند مستقیماً وارد مدل شوند. تصاویر معمولاً به اندازه ثابت تغییر داده می‌شوند و سپس نرمال‌سازی می‌شوند تا ورودی شبکه CNN استاندارد شود. متن‌ها نیز باید به فرمت عددی تبدیل شوند تا شبکه عصبی بتواند با آن‌ها کار کند.
این فرآیند شامل انتخاب دیتاست، نمایش نمونه‌ها، انجام پیش‌پردازش روی تصاویر و متن‌ها، ساخت دیکشنری کلمات، تعیین طول ثابت کپشن‌ها و تقسیم داده‌ها به مجموعه‌های آموزشی، اعتبارسنجی و تست است.

وظایف:

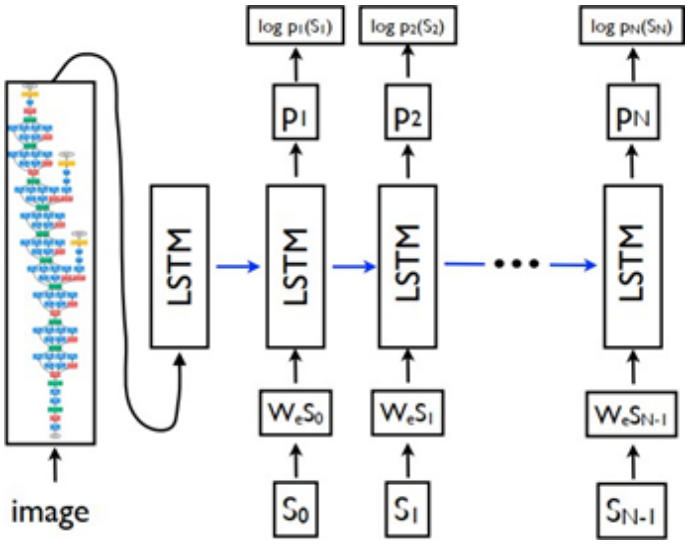
- دیتاست Flickr8k را دانلود کنید و چند نمونه تصویر همراه با کپشن را نمایش دهید.
- اندازه تصاویر را به ابعاد مناسب ورودی مدل CNN تغییر دهید.
- مقادیر پیکسل‌ها را با استفاده از میانگین و انحراف معیار استاندارد نرمال‌سازی کنید.
- متن کپشن‌ها را پیش‌پردازش کنید:
  - همه حروف را به حروف کوچک تبدیل کنید.
  - علائم نگارشی، نمادها و اعداد غیرضروری را حذف کنید.
  - متن را توکنایز کنید و کلمات را به شناسه عددی تبدیل کنید.
  - دیکشنری کلمات بسازید و توکن‌های ویژه <unk>، <eos>، <sos>، <pad> را اضافه کنید و کاربرد هرکدام را توضیح دهید.
  - دیکشنری را به صورت فایل JSON ذخیره کنید.
- طول ثابت برای کپشن‌ها تعیین کنید و از <pad> برای پر کردن استفاده کنید.
- دیتاست را به نسبت ۸۰٪ آموزش، ۱۰٪ اعتبارسنجی و ۱۰٪ تست تقسیم کنید و مطمئن شوید که تصاویر تکراری در مجموعه‌ها وجود نداشته باشد.
- ۵ تصویر تصادفی همراه با کپشن پردازش‌شده را نمایش دهید.

- نمودار پراکندگی طول کپشن‌ها را رسم کنید.
- هیستوگرام ۲۰ کلمه پرتکرار را ترسیم کنید.

بخش دوم: پیاده‌سازی مدل CNN-RNN (۲۵ نمره)

مدل‌های CNN-RNN مطابق این مقاله از ترکیب دو نوع شبکه عصبی ساخته می‌شوند: CNN برای استخراج ویژگی‌های تصویری و RNN (یا LSTM/GRU) برای تولید متن. در این پروژه از یک مدل CNN پیش‌آموزش‌داده‌شده مثل EfficientNet-B0 به عنوان Encoder استفاده می‌کنیم تا ویژگی‌های تصویر استخراج شود. سپس این ویژگی‌ها به LSTM داده می‌شوند که به عنوان Decoder وظیفه تولید کلمه به کلمه کپشن را بر عهده دارد.

برای پیش‌بینی کلمات بعدی از یک لایه Linear همراه با Softmax استفاده می‌شود. در آموزش مدل باید از تابع هزینه مناسب استفاده کرد و مراقب بود که Padding تأثیری بر محاسبه خطا نگذارد. همچنین برای جلوگیری از Overfitting می‌توان بخشی از لایه‌های CNN را ثابت نگه داشت یا از تکنیک‌های منظم‌سازی استفاده کرد.



معماری مدل CNN-RNN

وظایف:

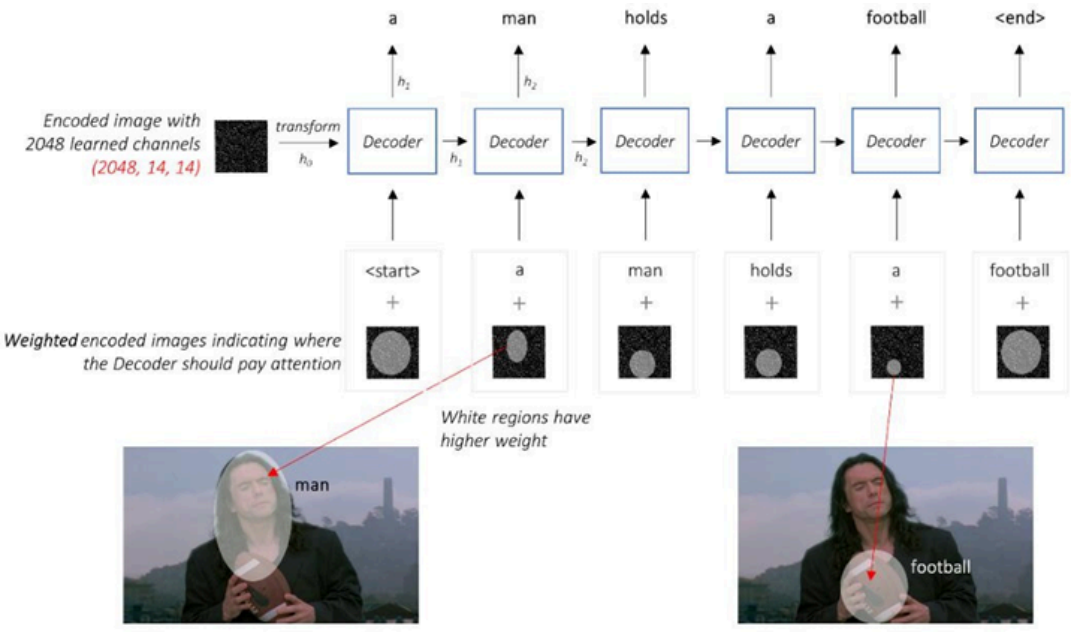
- بخش Encoder را پیاده‌سازی کنید:
  - از مدل CNN پیش‌آموزش‌داده‌شده EfficientNet-B0 استفاده کنید.
  - لایه Fully Connected نهایی را حذف کنید.
  - ابعاد خروجی ویژگی‌ها را بررسی کنید.
- بخش Decoder را پیاده‌سازی کنید:
  - از لایه Embedding برای نمایش برداری کلمات استفاده کنید و مزیت آن نسبت به One-hot را توضیح دهید.
  - از LSTM برای تولید کپشن استفاده کنید.
  - بردار ویژگی تصویر را به عنوان حالت اولیه LSTM وارد کنید.
  - از لایه Linear همراه با Softmax برای پیش‌بینی کلمه بعدی استفاده کنید.

- Encoder و Decoder را در قالب یک مدل End-to-End با نام ImageCaptioningModel ترکیب کنید.
  - مدل را با تابع هزینه مناسب آموزش دهید و در محاسبه خطا تأثیر Padding را حذف کنید.
  - همه یا بخشی از لایه‌های CNN را ثابت نگه دارید.
  - از تکنیک‌های پیشنهادی مقاله مرجع برای جلوگیری از Overfitting استفاده کنید.
  - Checkpoint مدل را ذخیره کنید تا در صورت توقف بتوان آموزش را ادامه داد.
- مدل را ارزیابی کنید:
  - نمودار خطای آموزش و اعتبارسنجی را در طول Epochها رسم کنید
  - خروجی مدل برای یک تصویر را در پایان هر Epoch نمایش دهید.
  - ۵ تصویر همراه با کپشن تولیدی نهایی را نمایش دهید.
  - خطاها را تحلیل کنید (مثلاً عدم تشخیص اشیاء یا روابط).

بخش سوم: پیاده‌سازی مکانیزم توجه (Attention) در مدل CNN-RNN (۳۰ نمره)

مکانیزم توجه یک گام پیشرفته در بهبود معماری CNN-RNN است که به مدل اجازه می‌دهد هنگام تولید هر کلمه از کپشن، روی بخش‌های مهم تصویر تمرکز کند. به جای اینکه مدل فقط یک بردار ثابت از ویژگی‌های تصویر دریافت کند، مکانیزم توجه به آن امکان می‌دهد وزن‌های متفاوتی به مناطق مختلف تصویر اختصاص دهد و این کار باعث افزایش دقت و کیفیت توضیحات تولید شده می‌شود.

در این بخش معماری CNN-RNN با مکانیزم توجه را بر اساس این مقاله پیاده سازی می‌کنیم.



مکانیزم اتنشن

بخش Encoder:

- بخش Encoder همانند بخش قبلی است ولی لایه Fully Connected نهایی و لایه‌های Pooling آخر حذف می‌شوند تا ویژگی‌ها شامل اطلاعات مکانی (Spatial) نیز باشند.
- خروجی Encoder را بررسی و ابعاد آن را گزارش کنید.
- وزن‌های توجه برای هر منطقه تصویر با ترکیب حالت مخفی گام قبلی Decoder و ویژگی‌های آن منطقه محاسبه می‌شوند.
- وزن‌های محاسبه شده از softmax عبور داده شده تا نرمال شوند.
- ویژگی‌های وزندار تصویر به صورت مجموع وزندار مناطق مختلف محاسبه می‌شود.

پیاده سازي 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:

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

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

وزنهای نرمالشده:

$$\alpha^i = softmax(e^i)$$

ویژگیهای وزندار تصویر:

$$z_t = \sum \alpha^i . a_i$$

بخش Decoder:

- در هر گام تولید کلمه، بردار وزندار تصویر (tz) همراه با embedding کلمه قبلی به LSTM داده می‌شود.
- برای تطابق ابعاد می‌توانید این ترکیب را از یک لایه خطی عبور دهید.
- وضعیت LSTM به‌روزرسانی می‌شود و خروجی آن به لایه خطی متصل می‌شود تا احتمال کلمات بعدی پیش‌بینی شود.
- کلمه با بیشترین احتمال انتخاب و به عنوان ورودی گام بعدی استفاده می‌شود.

آموزش مدل:

- از تابع هزینه مناسب استفاده کنید و padding را در محاسبه خطا لحاظ نکنید.
- تعداد Epochها مشابه مرحله قبل باشد.

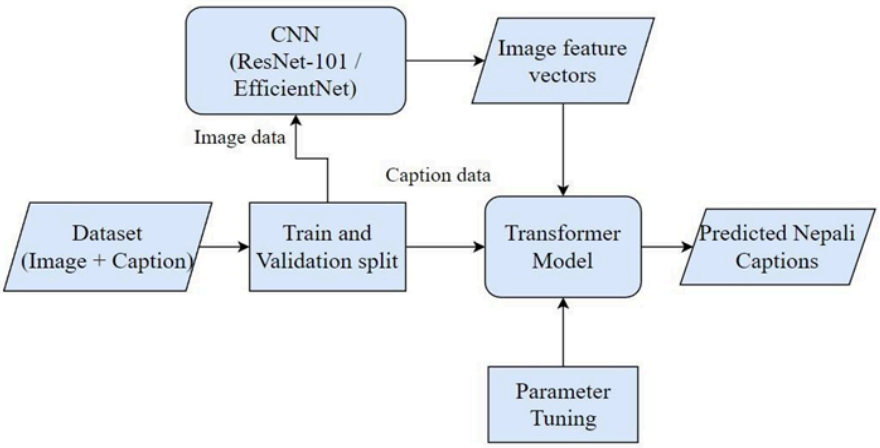
ارزیابی مدل:

- نمودار خطای آموزش و اعتبارسنجی را در طول هر Epoch رسم و گزارش کنید.

- در پایان هر دوره، نمونه‌ای از تصویر و کپشن تولید شده را نمایش دهید.
- ۵ تصویر و کپشن تولید شده نهایی را نمایش دهید.
- برای یکی از نمونه‌های داده تست، نقشه حرارتی (Heatmap) وزن‌های توجه در هر مرحله تولید کلمه را رسم کنید و تحلیل کنید که مدل در کدام بخش‌های تصویر تمرکز کرده است.
- خطاهای مدل را شناسایی کرده و آن‌ها را با مرحله قبل مقایسه کنید.

بخش چهارم: پیاده‌سازی مدل CNN-Transformer برای تولید کپشن (۲۵ نمره)

هدف این بخش استفاده از معماری Transformer در بخش Decoder برای تولید کپشن است که در کنار یک Encoder مبتنی بر CNN کار می‌کند. این روش مدرن‌تر و توانمندتر است و امکان مدل‌سازی بهتر روابط بین کلمات را فراهم می‌کند. در این بخش ما طبق این مقاله پیشروی خواهیم کرد.



مراحل تولید متن از عکس

پیاده‌سازی Tokenizer:

- Tokenizer را با قابلیت Masking تنظیم کنید، به‌طوری که توکن‌های اضافه شده در ادامه جمله دارای مقدار mask برابر True باشند.
- در گزارش توضیح دهید که چرا Masking کلمات در فرآیند آموزش اهمیت دارد.

بخش Encoder:

- از مدل EfficientNet-B0 به عنوان Encoder استفاده کنید.
- تمام لایه‌های Encoder را Freeze کنید (وزن‌ها ثابت بمانند) به جز لایه آخر که برای تنظیم ابعاد خروجی اصلاح می‌شود.

بخش Decoder:

- بخش Decoder را با استفاده از لایه Transformer پیاده کنید.
- از Embedding کلمات و Positional Embedding استفاده کنید.

- در گزارش به طور خلاصه توضیح دهید که Positional Embedding چه نقشی در مدل دارد.

آموزش و ارزیابی:

- مدل را برای تعداد Epoch کافی آموزش دهید تا به نتایج مطلوب برسید.
- در صورت نیاز از سخت‌افزار مناسب یا پلتفرم‌هایی مانند Kaggle استفاده کنید.
- پس از آموزش، مدل را ذخیره کنید.
- تغییرات Loss برای داده‌های آموزش و اعتبارسنجی را رسم کنید.
- برای ۵ تصویر تصادفی از داده تست، کپشن تولید کنید و نتایج را بررسی نمایید.

بخش پنجم: معیارهای ارزیابی مدل‌ها (۱۰ نمره)

ارزیابی کیفیت مدل‌های تولید کپشن اهمیت زیادی دارد و معیارهای مختلفی برای این کار استفاده می‌شود. یکی از معیارهای مهم، امتیاز BLEU است که برای اندازه‌گیری شباهت بین متن تولید شده مدل و متن‌های مرجع استفاده می‌شود.

- معیارهای مختلف ارزیابی مدل‌های Captioning را بررسی کنید.
- مطالعه‌ای مختصر درباره امتیاز BLEU انجام داده و توضیح دهید که چگونه عملکرد مدل را ارزیابی می‌کند.
- امتیاز BLEU (از BLEU-1 تا BLEU-4) را روی داده‌های تست محاسبه کنید.
- نتایج به‌دست‌آمده را برای بخش‌های مختلف مدل‌ها گزارش و مقایسه نمایید.