به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین امتیازی

طراحان تمرین:

سوال یک: بابک حسینی محتشم

سوال دو: محمد جواد رنجبر

سوال سه: یوسف قادری

سوال چهار: امیرحسین کمیجانی

مهلت ارسال: ۲۸ دی ۱۴۰۴

فهرست

سوال ۱. تولید توضیحات متنی برای تصاویر 6

۱-۱. آماده‌سازی دادگان (۱۵ امتیاز) 6

۱-۲. پیاده‌سازی مدل 7

۱-۲-۱. قسمت پردازش تصویر (۵ امتیاز) 7

۱-۲-۲. قسمت تولید متن (۱۵ امتیاز) 7

۱-۳. آموزش و ارزیابی 9

۱-۳-۱. آموزش مدل (۱۰ امتیاز) 9

۱-۳-۲. ارزیابی مدل (۱۵ امتیاز) 9

۱-۴. یادگیری خود نظارت شده مقایسه‌ای 9

۱-۵. پیاده‌سازی مدل 10

۱-۵-۱. قسمت پردازش تصویر (۵ امتیاز) 10

۱-۵-۲. قسمت تولید متن (۵ امتیاز) 10

۱-۵-۳. ادغام دو قسمت (۵ امتیاز) 10

۱-۶. آموزش (۱۰ امتیاز) 11

۱-۷. ارزیابی retrieve (۱۵ امتیاز) 11

سوال ۲. تشخیص اصوات شهری با استفاده از مدلWav2Vec 12

۲-۱. سوال‌های نظری (۳۰ امتیاز) 12

۲-۱-۱. عملکرد مدل‌های خودنظارتی 12

۲-۱-۲. تکنیک‌های افزایش داده صوتی 13

۲-۲. آموزش و ارزیابی مدل تشخیص صوت (۷۰ امتیاز) 13

۲-۲-۱. آموزش مدل‌های Wav2Vec و CNN 14

۲-۲-۲. ارزیابی مدل‌های Wav2Vec و CNN 15

سوال ۳. Fine tuning LLM با کمک LoRa در تشخیص احساسات 16

۳-۱. مفاهیم و تعاریف اولیه‌ی Full Fine-Tuning و LoRA Fine-Tuning (۱۰ امتیاز) 16

۳-۲. تعریف مسئله 17

۳-۲-۱. آشنایی با Hugging Face وDatasets 17

۳-۳. پیاده سازی 17
۳-۳-۱ .نمونه‌گیری Stratified (۱۰ امتیاز) 18
۳-۳-۲ . بارگذاری مدل و Tokenizer (۱۰ امتیاز) 18
۳-۳-۳. پیاده‌سازی تابع format_prompt (۱۰ امتیاز) 18
۳-۳-۴. انجام Tokenization و Encoding (۱۰ امتیاز) 19
۳-۴. آموزش LoRA (۴۰ امتیاز) 20
۳-۵. ارزیابی مدل (۱۰ امتیاز) 21
سوال ۴. حملات متخاصم 22
۴-۱. پیاده‌سازی حملات FGSM و PGD (۳۰ امتیاز) 22
۴-۱-۱. بارگذاری دیتاست CIFAR-10 و آموزش مدل CNN 22
۴-۱-۲. پیاده‌سازی حملات FGSM و PGD 22
۴-۱-۳. ارزیابی مدل تحت حمله 23
۴-۱-۴. نمودارها 23
۴-۲. مقایسه مقاومت دو معماری (۳۰ امتیاز) 23
۴-۲-۱. آموزش دو مدل و اعمال حملات FGSM و PGD 23
۴-۲-۲. مقایسه نهایی 23
۴-۳. مقاوم‌سازی مدل با Adversarial Training (۴۰ امتیاز) 23
۴-۳-۱. adversarial training 24
۴-۳-۲. ارزیابی پس از دفاع 24
۴-۳-۳. نمودارها 24

شکل ها

شکل 1. تاثیر وجود و عدم وجود Teacher Forcing 8

شکل 2 .معماری مدل Wav2vec 12

شکل 3. تفاوت Full Fine Tuning و Fine Tuning Using Lora 16

جدول‌ها

جدول 1 . مقادیر هایپرپارامترها 14
جدول 2 . روش‌های آموزش 15

سوال ۱. تولید توضیحات متنی برای تصاویر

یکی از کاربردهای مهم مدل‌های متنی و تصویری در یادگیری عمیق تولید توضیحات متنی برای تصاویر[1] است. برای مثال می‌توان از مدل‌های آموزش دیده برای این تسک برای ایجاد زیرنویس برای ویدیوها، یا کمک به افراد نا بینا، امکان خوشه‌بندی و جستجوی تصاویر و در زمینه‌های دیگر استفاده کرد.

مدل‌های بینایی ماشین معمولا برای طبقه‌بندی، تشخیص یا بخش‌بندی تصاویر آموزش می‌بینند. از طرفی مدل‌های زبانی برای تولید و درک جملات زبان انسانی آموزش دیده‌اند. با ترکیب این دو جوزه می‌توان به مدل‌هایی رسید که تصویری را به عنوان ورودی دریافت می‌کنند و توضیح متنی کوتاهی مرتبط با تصویر خروجی می‌دهند.

۱-۱. آماده‌سازی دادگان (۱۵ امتیاز)

دو دادگان رایج در این زمینه COCO و Flickr هستند. در این تمرین باید از این لینک دادگان را دریافت کنید که شامل بخشی از دادگان COCO و Flickr است که توضیحات متنی به زبان فارسی ترجمه شده‌اند. فایل دانلود شده را unzip کنید. تصاویر در پوشه images قرار گرفته‌اند و توضیحات متنی تصاویر در فایل‌هایی با فرمت csv برای سه داده آموزش، اعتبارسنجی و ارزیابی وجود دارند.

پس از دریافت دادگان، باید توضیحات متنی را پیش‌پردازش کنید. برای این کار، باید یک کلاس Tokenizer تشکیل دهید که متون را به عنوان ورودی دریافت می‌کند و ابتدا نمادها و emoji ها را از متون حذف می‌کند.

در مورد سایر روش‌های پیش‌پردازش متون برای این تسک تحقیق و گزارش کنید.

در ادامه با شکستن متون به توکن‌ها، باید لغت‌نامه‌ای از توکن‌ها تشکیل دهید به طوریکه با دریافت اندیس، توکن متناظر را بازگرداند. همچنین کلاس شما باید بتواند با دریافت متن، لیستی از توکن‌ها و با دریافت لیستی از توکن‌ها، متن متناظر را خروجی دهد.

همچنین لغت‌نامه باید شامل چهار توکن ویژه <unk>، <sos>، <pad> و <eos> باشد و توابع نوشته شده در کلاس Tokenizer باید به درستی از این توکن‌ها هنگام تبدیل متن به توکن‌ها استفاده کند.

در مورد این توکن‌ها تحقیق کنید و اهمیت وجود هر یک را توضیح دهید و بیان کنید از هر یک از این توکن‌ها در کجای متن استفاده می‌شود.

با کمک داده‌های آموزش، شیء Tokenizer را تشکیل دهید تا لغت‌نامه مناسبی ایجاد شود. سپس یک نمونه تصویر به همراه متن توضیح و توکن‌های استخراج شده را رسم کنید. همچنین اطلاعات آماری مانند توکن‌ها با بیشترین و کمترین تکرار و میانگین طول کپشن‌ها را گزارش کنید.

دیتاستی تشکیل دهید که تصویر را به همراه لیست آیدی توکن‌ها برگرداند. سپس دیتالودرها را تشکیل دهید و از یک تابع collate استفاده کنید تا حداکثر طول متون یک بچ را به اندازه ثابتی مثلا 40 توکن کاهش دهد و برای متون کوتاه‌تر از توکن‌های ویژه برای رساندن به طول مورد نظر استفاده کند.

---

[1] Image Captioning

۱-۲. پیاده‌سازی مدل

شبکه‌های عصبی پیچشی[2] از رایج‌ترین مدل‌های پردازش تصاویر به شمار می‌روند که در این تمرین برای استحراج ویژگی‌های تصویر از آن‌ها استفاده می‌کنید. در حوزه متن، مدل‌های LSTM از جمله معماری‌های موفق به شمار می‌روند که با افزودن مکانیزم توجه به آن‌ها، می‌توان به عملکرد بهتری رسید. پس از پیاده‌سازی هر یک از قسمت‌های انکودر و دیکودر، تعداد پارامترهای آن بخش را گزارش کنید.

به نظر شما از چه معماری‌های جایگزین می‌توان برای این تسک استفاده کرد؟ مزایای و معایب هر یک را برای این تسک توضیح دهید.

۱-۲-۱. قسمت پردازش تصویر (۵ امتیاز)

در این تمرین از قسمت استخراج ویژگی‌های مدل VGG16 از پیش آموزش دیده استفاده کنید. برای افزایش سرعت آموزش می‌توانید پارامترهای آموزشی این مدل را freeze کنید. برای پیش‌پردازش تصاویر، از پیش‌پردازش استفاده شده برای آموزش مدل روی دادگان IMAGENET1K_V1 استفاده کنید که نحوه استفاده از آن در این لینک آمده است.

۱-۲-۲. قسمت تولید متن (۱۵ امتیاز)

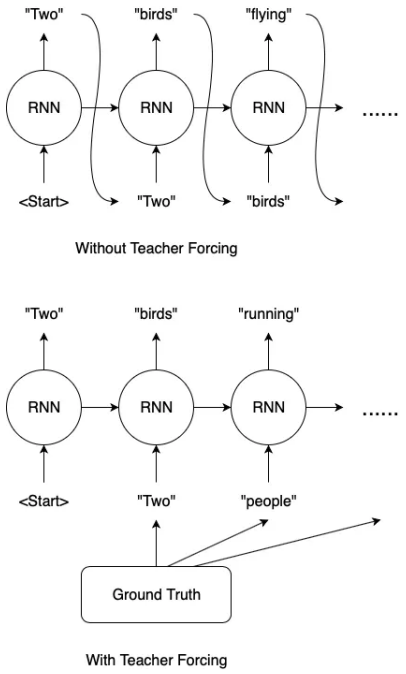
برای تبدیل توکن‌ها به بردارهای قابل آموزش، از یک لایه embedding استفاده کنید. برای تولید توضیحات متنی، باید از معماری LSTM به همراه مکانیزم توجه معرفی شده در این مقاله استفاده کنید.

تفاوت مکانیزم توجه معرفی شده در مقاله را با مکانیزم توجه مورد استفاده در معماری Transformer بیان کنید.

در روش تولید متن حریصانه، در هر مرحله محتمل‌ترین توکن، به عنوان خروجی در نظر گرفته می‌شود.

---

[2] Convolutional Neural Networks



شکل 1. تاثیر وجود و عدم وجود teacher forcing

همان طور که در شکل مشخص است، در هنگام آموزش معمولا از روش teacher forcing استفاده می‌کنیم تا با دادن توکن‌های متن هدف به مدل باعث افزایش سرعت آموزش شویم. دلیل آن این است که در ابتدا مدل در تولید متن ضعیف است و با تولید یک توکن اشتباه، توکن‌های بعدی هم باید از روی آن حدس زده بشوند و این باعث تجمیع خطاها و سخت‌تر شدن یادگیری مدل می‌شود. پس از آموزش مدل، چون متن هدف را نداریم و قصد داریم تنها با دادن تصویر به عنوان ورودی به توضیح متنی برسیم، از روش teacher forcing نمی‌توانیم استفاده کنیم.

بدین ترتیب قسمت دیکودر با دریافت ویژگی‌های استخراج شده در انکودر، باید به صورت حریصانه[3] و با روش teacher forcing توضیح متنی مناسب را تولید کند. توجه کنید پس از اتمام آموزش و هنگام ارزیابی نباید از روش teacher forcing استفاده کنید برای این کار می‌توانید یک تابع در مدل داشته باشید که با دریافت تصویر متن را با روش حریصانه تولید و خروجی دهد.

برای پیاده‌سازی این بخش می‌توانید از این هایپرپارامترها استفاده کنید: طول بردار امبدینگ برابر با 300، طول بردار وزن‌های LSTM و مکانیزم توجه هم برابر 512 در نظر بگیرید.

[3] Greedy

۱-۳. آموزش و ارزیابی

۱-۳-۱. آموزش مدل (۱۰ امتیاز)

توابع مورد نیاز برای آموزش مدل را بنویسید و مدل را آموزش دهید. از داده اعتبارسنجی استفاده کنید و با چاپ کردن متن تولیدی برای حداقل یکی از تصاویر در هنگام آموزش پس از هر ایپاک، بهبود متن تولیدی را بررسی کنید. همچنین بهترین مدل را در طول آموزش از لحاظ کمینه بودن خطا ذخیره و پس از پایان آموزش از آن استفاده کنید. برای آموزش ازتابع خطای Cross Entropy استفاده کنید.

۱-۳-۲. ارزیابی مدل (۱۵ امتیاز)

ابتدا تغییرات خطای برای داده آموزش و اعتبارسنجی را هنگام آموزش بررسی کنید. سپس متن تولید شده برای تعدادی از تصاویر داده ارزیابی را رسم و تحلیل کنید. برای مثال بررسی کنید که آیا اشتباهات موجود در متن تولیدی به طور شهودی قابل توجیه است؟

در مورد نحوه کارکرد معیار BLEU تحقیق کنید.

در نهایت مقادیر BLEU-1 تا BLEU-4 را بر روی داده ارزیابی حساب کنید. دریافت حداقل مقدار 0.03 برای BLEU-4 برای دریافت نمره کامل ضروری است.

۱-۴. یادگیری خود نظارت شده مقایسه‌ای

یادگیری خود نظارت شده مقایسه‌ای[4] یکی از روش‌های رایج برای آموزش مدل‌هایی با فضای اشتراکی متنی-تصویری است. یکی از معروف‌ترین مدل‌های آموزش دیده با این روش CLIP است. از این مدل‌ها می‌توان برای جستجوی متن به کمک تصویر یا جستجوی تصویر به کمک متن و طبقه‌بندی بدون نمونه[5] استفاده کرد.

هدف آن است که دو مدل داشته باشیم که با یکی با دریافت متن و دیگری با دریافت تصویر، برداری خروجی بدهند که شباهت کسینوسی این دو بردار بالا باشد. در این روش‌ها فرض می‌شود داده انبوهی از کلاس‌های زیاد در اختیار داریم بدین ترتیب در هر بچ، احتمال آنکه داده‌هایی از دو کلاس یکسان بیفتد پایین می‌شود و می‌توان با فرض متفاوت بودن کلاس‌ها تابع خطایی را معرفی کرد که تا حد امکان بردارهای مربوط به متن و تصویر را به هم نزدیک و بردار آن‌ها را از بقیه تصاویر و متون موجود در بچ دور کند. یکی از توابع رایج که به همین صورت عمل می‌کند و در این تمرین پیاده می‌کنید، InfoNCE است.

---

[4] Contrastive self-supervised learning
[5] Zero-shot Clasification

۱-۵. پیاده‌سازی مدل

۱-۵-۱. قسمت پردازش تصویر (۵ امتیاز)

برای قسمت استخراج ویژگی تصویر از مدل از پیش آموزش دیده ViT small patch16 224 یا مدل‌های بزرگتر ViT از کتابخانه timm استفاده کنید و به انتهای آن یک لایه کاملا متصل اضافه کنید تا به بردار خروجی با طول مورد نظر برسید. همچنین بردار ویژگی‌های مدل را نرمالایز کنید. از تبدیلاتی که برای آموزش مدل ViT استفاده شده است در تصاویر خود، پیش از ورودی دادن به مدل استفاده کنید. برای این کار می‌توانید از کد زیر استفاده کنید:

```
data_config = timm.data.resolve_model_data_config(model)
transforms = timm.data.create_transform(**data_config,
is_training=False)
```

اگر مدل توضیح داده شده را به طور کامل freeze کنیم آیا با آموزش بخش پردازش متن می‌توان پیشرفتی مشاهده کرد؟

۱-۵-۲. قسمت تولید متن (۵ امتیاز)

برای تبدیل متن به بردار ویژگی‌ها از مدل LSTM آموزش داده شده در بخش قبل استفاده کنید و تنها به آن یک لایه کاملا متصل برای رسیدن به ابعاد مورد نظر از ویژگی برسید. بردار ویژگی خروجی از این مدل را نیز نرمالایز کنید.

۱-۵-۳. ادغام دو قسمت (۵ امتیاز)

دو قسمت پردازش متن و تصویر را ادغام کنید و به مدلی برسید که با دریافت تصاویر و متن، ابتدا بردارهای ویژگی‌ها را به دست آورد و سپس ضرب داخلی دو بردار را برگرداند.

در مورد تاثیر اضافه کردن دما [6] به حاصل ضرب داخلی بردارها تحقیق کنید و از دما با مقدار مناسب در مدل خود استفاده کنید.

---

[6] Temperature

۱-۶. آموزش (۱۰ امتیاز)

تابع خطای InfoNCE را به همراه بقیه توابع مورد نیاز برای آموزش مدل را بنویسید و مدل را با داده آموزش قسمت قبل، آموزش دهید. از داده اعتبارسنجی در طول آموزش استفاده کنید تا بهترین مدل را در طول آموزش از لحاظ کمینه بودن خطا ذخیره و پس از پایان آموزش از آن استفاده کنید.

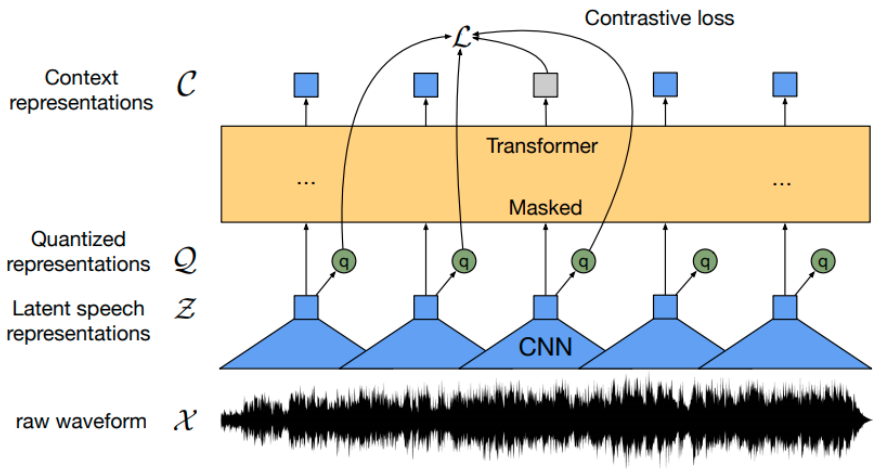
۱-۷. ارزیابی retrieve (۱۵ امتیاز)

ابتدا تغییرات خطای برای داده آموزش و اعتبارسنجی را هنگام آموزش بررسی کنید. سپس به ازای هر یک از تصاویر ارزیابی، شباهت آن به تمام متون داده ارزیابی را به دست آورید. با کمک اعداد به دست آمده، حساب کنید به طور میانگین برای چند درصد تصاویر، متن مربوط به تصویر، جزو ۵ تا از شبیه‌ترین متون بود یا جزو ۱۰ تا از شبیه‌ترین متون بود. برای حداقل ۴ نمونه از تصاویر، ۵ تا از شبیه‌ترین متون به همراه متن هدف را نمایش دهید.

همین مقادیر Top 1/5/10 accuracy را برای متون با جستجوی تصویر به دست آورید و برای حداقل ۴ نمونه از متون، پنج تا از شبیه‌ترین تصاویر به همراه تصویر هدف را نمایش دهید.

سوال ۲. تشخیص اصوات شهری با استفاده از مدل Wav2Vec

مدل‌های صوتی نظیر Wav2Vec 2.0 انقلابی در پردازش گفتار ایجاد کرده‌اند. این مدل‌ها با استفاده از یادگیری خودنظارتی[7] بر روی حجم عظیمی از داده‌های بدون برچسب آموزش می‌بینند. در این سوال قصد داریم با مکانیزم داخلی این مدل آشنا شویم. مقاله wav2vec 2.0 را مطالعه کنید.



شکل 2 .معماری مدل Wav2Vec

۲-۱. سوال‌های نظری (۳۰ امتیاز)

۲-۱-۱. عملکرد مدل‌های خودنظارتی

الف) تابع هزینه در این مدل از نوع Contrastive Loss است. نحوه عملکرد این تابع هزینه و نقش "Masking" در فرآیند آموزش را به دقت تشریح کنید. مدل چگونه بدون داشتن متن یاد می‌گیرد که بازنمایی‌های معناداری تولید کند؟

ب) علاوه بر Wav2Vec 2.0، مدل‌های دیگری مانند HuBERT و Whisper نیز مطرح هستند. تفاوت اصلی در رویکرد آموزش HuBERT نسبت به Wav2Vec 2.0 چیست؟

پ) مدل Wav2Vec 2.0 را با یک مدل استاندارد Mel-Spectrogram + CNN مقایسه کنید. مزیت اصلی استفاده از Raw Waveform در Wav2Vec چیست؟

[7] Self-supervised learning (SSL)

۲-۱-۲. تکنیک‌های افزایش داده صوتی

الف) با توجه به ماهیت داده‌های صوتی و نیاز به افزایش مقاومت مدل در برابر نویز و تنوع، تکنیک‌های افزایش داده[8] بسیار اهمیت دارند. حداقل سه تکنیک رایج افزایش داده که در دامنه زمانی[9] عمل می‌کنند را نام برده و تأثیر آن‌ها بر سیگنال خام را به اختصار توضیح دهید.

ب) ویژگی‌های مهندسی‌شده‌ای[10] هنوز برای بسیاری از وظایف صوتی مورد استفاده قرار می‌گیرند. سه ویژگی استخراجی صوتی مهم و متداول را نام ببرید. برای هر یک از این سه ویژگی، یک وظیفه یادگیری ماشینی را مشخص کنید که در آن ویژگی مورد نظر نقش کلیدی دارد (مثلاً Pitch برای تشخیص جنسیت یا MFCC برای ASR). همچنین توضیح دهید که چرا این ویژگی برای وظیفه دیگری (مثلاً Pitch برای ASR) مناسب نیست یا اهمیت کمتری دارد.

۲-۲. آموزش و ارزیابی مدل تشخیص صوت (۷۰ امتیاز)

مجموعه‌داده UrbanSound8K یکی از معتبرترین مجموعه‌داده‌های عمومی برای تحقیقات در حوزه پردازش سیگنال‌های صوتی شهری است. این مجموعه داده شامل ۸۷۳۲ کلیپ صوتی کوتاه‌مدت با حداکثر طول ۴ ثانیه است که از محیط‌های شهری مختلف جمع‌آوری شده‌اند. هر فایل صوتی به یکی از ۱۰ کلاس متمایز تعلق دارد که عبارتند از: کولر گازی، بوق خودرو، بازی کودکان، پارس سگ، صدای دریل، موتور در حال کار، شلیک گلوله، چکش‌بتن‌شکن، آژیر، و موسیقی خیابانی. این مجموعه داده به دلیل تنوع بالای صداها، حضور نویزهای پس‌زمینه طبیعی و چالش‌های واقعی محیط‌های شهری، به یک benchmark استاندارد در تحقیقات Audio Scene Classification تبدیل شده است.

---

[8] Augmentation

[9] Time Domain

[10] Hand-Engineered Features

جدول 1 . مقادیر هایپرپارامترها

دسته‌بندی	پارامتر	مقدار
پارامترهای داده	نرخ نمونه‌برداری	16 kHz
	حداکثر طول کلیپ	4ثانیه
	تعداد کلاس‌های خروجی	10
	تقسیم‌بندی داده	Train: 80% Validation: 10% Test: 10%
	Learning rate	1e-4
	Epoch	5
	Patience (Early Stopping)	3
	Optimizer	AdamW

۲-۲-۱. آموزش مدل‌های Wav2Vec و CNN

در این بخش، قصد داریم چهار رویکرد متفاوت را برای حل مسئله طبقه‌بندی صوت شهری را پیاده‌سازی و مقایسه کنیم. هدف اصلی، درک عمیق‌تر از تفاوت‌های عملکردی بین آموزش از صفر[11] و تکنیک‌های مختلف یادگیری انتقالی[12] است. همچنین قصد داریم تأثیر فریز کردن لایه‌های مختلف شبکه را بر سرعت همگرایی، دقت نهایی و تعداد پارامترهای قابل آموزش بررسی کنیم. در جدول ۲ مدل‌هایی که قصد آموزش و مقایسه‌ی آن‌ها را داریم نشان داده شده است.

[11] From Scratch
[12] Transfer Learning

جدول 2 . روش‌های آموزش

نام آزمایش	وضعیت پارامترها	توضیح کوتاه
شبکه عصبی پیچشی پایه	همه پارامترها قابل آموزش	یک مدل CNN با حداقل دو لایه طراحی کرده و برای این وظیفه آموزش دهید.
تنظیم دقیق کامل Wav2Vec 2.0	همه پارامترها قابل آموزش	از مدل پیش‌آموزش‌دیده facebook/wav2vec2-base استفاده کنید و تمام پارامترها را آموزش دهید.
آموزش لایه طبقه‌بند	فقط لایه طبقه‌بند قابل آموزش	تمام لایه‌های مدل Wav2Vec را فریز کرده و تنها لایه طبقه‌بند را آموزش دهید.
فریز جزئی لایه‌ها	تعدادی از لایه‌های مدل و همچنین لایه طبقه‌بند قابل آموزش	شش لایه اول transformer فریز کنید، بقیه لایه‌ها و طبقه‌بند را آموزش دهید.

۲-۲-۲. ارزیابی مدل‌های Wav2Vec و CNN

الف) برای هر مدل، معیارهای عملکرد و پیچیدگی را برای داده‌های آزمون محاسبه کنید:

- معیارهای عملکرد: دقت، F1 Score (Macro)، خطا
- معیارهای پیچیدگی: تعداد پارامترهای قابل آموزش، درصد پارامترهای قابل آموزش، زمان آموزش در هر epoch

پس از انجام آزمایش‌ها، موارد زیر را تولید کنید:

- جدول مقایسه مدل‌ها بر اساس معیارهای عملکرد و پیچیدگی
- نمودار سایز مدل در مقابل دقت
- نمودار درهم‌ریختگی[13]
- نمودار دقت و خطا در طول آموزش

پ) تحلیل و مقایسه:

- مدل‌ها و عملکردشان را با یکدیگر مقایسه کنید. توضیح دهید که هر مدل چرا این عملکرد را داشت و در چه شرایطی استفاده از هر یک از روش‌های بالا توصیه می‌شود.
- آیا افزایش تعداد پارامترهای قابل آموزش لزوما باعث بهتر شدن عملکرد مدل می‌شود؟ چرا؟

---

[13] Confusion Matrix

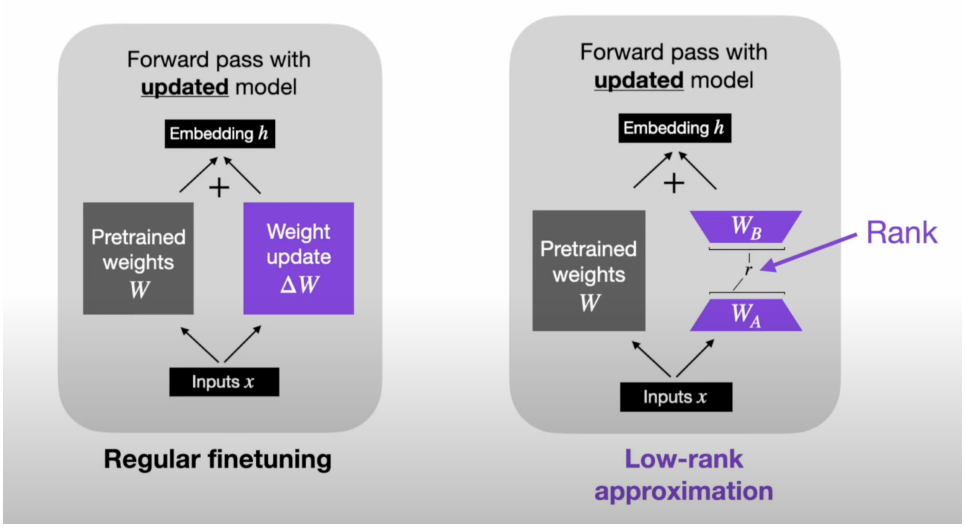
سوال ۳. Fine tuning LLM با کمک LoRa در تشخیص احساسات

در این سؤال، هدف ما بررسی مفهوم Fine-Tuning است؛ با این تفاوت که منظور ما Full Fine-Tuning یا همان به‌روزرسانی همه‌ی وزن‌های مدل نیست، بلکه رویکرد LoRA Fine-Tuning مدنظر است. در روش LoRA (Low-Rank Adaptation)، وزن‌های اصلی مدل از پیش‌آموزش‌دیده فریز شده و ثابت می‌مانند و به‌جای آن، یکسری ماتریس‌های کم‌رتبه‌ی افزوده به‌نام LoRA adapters به لایه‌ها اضافه می‌شود. در فرآیند آموزش، تنها پارامترهای این ماتریس‌های افزوده به‌روزرسانی می‌شوند و در نهایت، وزن مؤثر هر لایه به صورت ترکیب وزن اصلی به‌علاوه‌ی تصحیح کم‌رتبه‌ی ناشی از LoRA اعمال می‌گردد.

۳-۱. مفاهیم و تعاریف اولیه‌ی Full Fine-Tuning و LoRA Fine-Tuning (۱۰ امتیاز)

در یادگیری عمیق، Fine Tuning یعنی ما یک مدل از پیش آموزش‌دیده را برمی‌داریم، تقریباً همه یا بخش بزرگی از وزن‌های آن را روی داده‌های جدید دوباره آموزش می‌دهیم تا مدل دقیقاً با مسئله و دامنه‌ی جدید سازگار شود. در مقابل، LoRA Fine Tuning روشی کارآمدتر است که در آن وزن‌های اصلی مدل (مثلاً وزن‌های لایه‌های ترنسفورمر) را فریز می‌کنیم و به‌جای تغییر مستقیم آن‌ها، تعدادی ماتریس کم‌مرتبه (Low-Rank) کوچک به مدل اضافه می‌کنیم که تنها همین پارامترهای اضافه‌شده آموزش داده می‌شوند. در شکل ۱ نحوه عملکرد نمایش داده شده است.

مزیت‌ها و معایب هر یک از این دو روش را بیان کنید همینطور در توضیحات خودتان به دنبال یک مثال عددی باشید که واقعا مزیت اصلی LoRa را نشان بدهد.



شکل 3. تفاوت Full Fine tuning و Fine tuning using LoRa

۳-۲. تعریف مسئله

در این سوال قرار هست مدل زبانی Llama-3.2-1B را برروی یک دیتاست تشخیص احساسات تنظیم دقیق کنیم اما در ابتدا در خصوص دیتاست و مدل زبانی اشاره شده و نحوه دسترسی به آن نکاتی را بیان می کنیم.

۳-۲-۱. آشنایی با Hugging Face وDatasets

Hugging Face یک پلتفرم متن‌باز برای دسترسی و مدیریت مدل‌های یادگیری ماشین و هوش مصنوعی است که در ابتدا تمرکز اصلی آن روی حوزه‌ی NLP بوده، اما امروزه دامنه‌ی آن به سایر حوزه‌ها نیز گسترش یافته است. در این پلتفرم می‌توان به انواع مدل‌ها و دیتاست‌ها دسترسی داشت؛ برخی از آن‌ها برای استفاده نیازمند مجوز و تأیید هستند، در حالی که تعداد زیادی نیز به صورت عمومی (public) در دسترس قرار دارند. در این سؤال، ما از یک مدل زبانی و دیتاست عمومی استفاده خواهیم کرد و تنها پیش‌نیاز کار، دسترسی به این منابع بدون نیاز به اخذ مجوز خاص است. یک access token در این وب سایت از طریق اکانت خودتان ایجاد کنید.

برای دسترسی به دیتاست سوال می توانید قطعه کد زیر را اجرا کنید.

```
from datasets import load_dataset
from transformers import AutoModel
dataset = load_dataset('emotion')
```

دیتاست Emotion یک مجموعه داده متنی زبان انگلیسی برای تسک تشخیص احساسات (emotion classification) است که از پیام‌های واقعی توییتر جمع‌آوری شده و به شش احساس اصلی برچسب‌گذاری شده است. این دیتاست در سه بخش تقسیم شده است: آموزش: حدود 16000 نمونه، اعتبارسنجی: حدود 2000 نمونه، و آزمون: حدود 2000 نمونه. برای دریافت اطلاعات بیشتر درمورد دیتاست، به این لینک مراجعه کنید.

برای برچسب‌های دیتاست از این mapping استفاده کنید:

{0 : sadness, 1 : joy, 2 : love, 3 : anger, 4 : fear, 5 : surprise}

۳-۳. پیاده سازی

در این قسمت قدم به قدم توضیح میدهیم که چطور میتوانیم یک مدل را به شیوه LoRa تنظیم دقیق کنیم با دقت تمامی مراحل را طی کنید و در هر مرحله نتیجه را در گزارشتان قرار دهید.

۳-۳-۱ .نمونه‌گیری Stratified (۱۰ امتیاز)

یک تابع بنویسید به نحوی که از دیتاست نام برده شده به صورت لایه بندی شده نمونه گیری کند توزیع برچسب داده‌های شما باید توضیح اصلی دیتاست را حفظ کند تعداد داده‌ها در هر دسته به طریق زیر باشد. در نهایت توزیع داده‌ی منتخب را نمایش دهید و با توزیع دیتاست اصلی مقایسه کنید.

```
DS_NAME = 'emotion'
DS_TRAINING_SIZE = 1500
DS_TEST_SIZE = 100
DS_VALIDATION_SIZE = 50
```

۳-۳-۲. بارگذاری مدل و Tokenizer (۱۰ امتیاز)

meta-llama/Llama-3.2-1B یک مدل decoder هست که شامل ۱ میلیارد پارامتر بوده و قابل اجرا در محیط colab و kaggle خواهد بود. از طریق این لینک میتوانید در خصوص ویژگی های این مدل مطالعه کنید. از طریق قطعه کد زیر مدل را به همراه توکنایزر مورد نظر Load کنید.

```
from transformers import AutoTokenizer, AutoModelForCausalLM
tokenizer = AutoTokenizer.from_pretrained("meta-llama/Llama-3.2-1B")
model = AutoModelForCausalLM.from_pretrained("meta-llama/Llama-3.2-
1B")
```

۳-۳-۳. پیاده‌سازی تابع format_prompt (۱۰ امتیاز)

یک تابع بنویسید به نحوی که فرمت پرامپت مناسب را شکل بدهد پارامتر ورودی شامل یک system instruction , user input , assistant outputمناسب باشد برای راهنمایی بیشتر در قسمت system instructionوظیفه مدل زبانی را به همراه برچسب هایی که میتواند خروجی بدهد مشخص کنید در قسمت user inputمتن مورد نظر از دیتاست را قرار بدهید و assistant output هم صرفا شامل برچسب خروجی از مدل باشد.فرمت پرامپت ورودی بهتر است به فرمت زیر باشد:

```
prompt= f"<s>[INST] {system_instruction} [/INST] {user_input} </s>
[ASSISTANT] {assistant_output} </s>"
```

۳-۳-۴. انجام Tokenization و Encoding (۱۰ امتیاز)

در نهایت با با استفاده از توکنایزر مدل بارگذاری شده داده ها را tokenize کنیم تا پارامتر های مدل روی embeddingهای این توکنایزر آموزش ببیند. برای پارامترهای tokenization این موارد را در نظر داشته باشید:

```
truncation=True,
padding="max_length",
max_length=128
```

در نهایت بعد از انجام tokenization برای یکی از داده ها نشان دهید با decode کردن tokenization به همان جمله اولیه میرسید. برای مثال:

```
[128000, 45147, 31868, 65562, 60, 38527, 3059, 279, 27065, 315, 279,
2768, 1495, 13, 7974, 9382, 527, 25, 364, 83214, 2136, 518, 364,
4215, 518, 364, 31153, 518, 364, 4091, 518, 364, 69, 686, 518, 364,
20370, 9868, 4527, 66028, 65562, 60, 602, 2011, 2019, 430, 420,
85152, 706, 1027, 682, 35208, 34356, 449, 1063, 3682, 4442, 520,
990, 84055, 3515, 24869, 323, 602, 2733, 1093, 602, 617, 1027, 264,
39332, 1285, 17240, 694, 82, 29, 510, 5045, 3931, 2891, 60, 51978,
694, 82, 29, 128001, 128001, 128001, 128001, 128001, 128001, 128001,
128001, 128001, 128001, 128001, 128001, 128001, 128001, 128001, 128001,
128001, 128001, 128001, 128001, 128001, 128001, 128001, 128001, 128001,
128001, 128001, 128001, 128001, 128001, 128001, 128001, 128001, 128001,
128001, 128001, 128001, 128001, 128001, 128001, 128001, 128001, 128001,
128001, 128001, 128001, 128001]
 === Verification ===
Original Tokenized Text:
<s>[INST] Analyze the sentiment of the following text. Valid labels
are:'sadness', 'joy', 'love', 'anger', 'fear','surprise'. [/INST] i
must say that this makeover has been all consuming coupled with some
major changes at work coworkers having babies and i feel like i have
been a neglectful lady </s> [ASSISTANT] sadness </s> Labels (Tokens
Being Predicted): sadness </s> Assistant Text Extracted: [ASSISTANT]
sadness </s>
```

۳-۴. آموزش LoRA (۴۰ امتیاز)

در تنظیمات LoRa پارامترهای مختلفی وجود دارد که میتوان آن‌ها را قبل از شروع آموزش تنظیم کرد. در مورد پارامترهای rank value، scaling factor، lora drop out، و target modules تحقیق کنید و هرکدام را توضیح دهید.

برای آموزش مدل تنظیمات را طبق کد زیر اعمال کنید و تعداد پارامتر های قابل آموزش را در گزارشتان ذکر کنید.

```
r_values = 8
lora_alpha_values = 32
target_modules_values = ["q_proj", "k_proj", "v_proj", "o_proj",
"gate_proj", "up_proj", "down_proj"]
lora_dropout_values = 0.1
```

TrainingArguments یک کلاس در کتابخانه Transformers است که تنظیمات و هایپرپارامترهای مربوط به روند آموزش، ارزیابی و ذخیره‌سازی مدل را به صورت متمرکز تعریف می‌کند. یکی از پارامترهای این کلاس lr_scheduler_type می‌باشد که نوع تغییر نرخ یادگیری در طول آموزش را مشخص می‌کند و با مقدار cosine نرخ یادگیری به‌تدریج طبق منحنی کسینوسی کاهش می‌یابد. همچنین per_device_train_batch_size تعداد نمونه‌هایی که در هر گام آموزش پردازش می‌شوند را مشخص می‌کند.

مدل را طبق کد زیر، با تعریف TrainingArguments مناسب برای آموزش تعریف کنید. در فایل گزارش هر کدام از پارامترهای ورودی این کلاس که در کدتان استفاده کردید را در حد یک خط تعریف کنید.

```
from transformers import TrainingArguments
training_args = TrainingArguments(
lr_scheduler_type="cosine",
per_device_train_batch_size=4,
gradient_accumulation_steps=4,
per_device_eval_batch_size=4,
num_train_epochs=10,
fp16=True,
load_best_model_at_end=True,
weight_decay=0.01,
)
```

در نهایت مدل را آموزش دهید و نمودار مربوط به تغییرات Loss را در حین آموزش و ارزشیابی رسم کنید.

۳-۵. ارزیابی مدل (۱۰ امتیاز)

بعد از آموزش مدل به روش LoRa می بایست به سراغ مقایسه مدل ها میرویم سه مدل:

- مدل Fine tune شده شما به وسیله LoRa
- مدل Base اولیه: meta-llama/Llama-3.2-1B
- مدل Instruction: meta-llama/Llama-3.2-1B-Instruct

داده‌های تستِ دیتاست را روی هر سه مدل اعمال کنید و برای هر مدل، confusion matrix را استخراج و گزارش کنید. در نهایت، مقادیر به‌دست‌آمده برای دو معیار Accuracy و F1-score را برای هر سه مدل رسم و با یکدیگر مقایسه کنید.

سوال ۴. حملات متخاصم

یکی از چالش‌های بنیادین در حوزه یادگیری عمیق، آسیب‌پذیری مدل‌های شبکه‌های عصبی در برابر حملات متخاصم (Adversarial Attacks) است. این حملات با ایجاد تغییرات بسیار کوچک و نامحسوس در ورودی، به‌گونه‌ای که برای انسان قابل تشخیص نباشد، می‌توانند تصمیم مدل را به‌طور کامل تغییر دهند. این پدیده نشان می‌دهد که حتی مدل‌هایی با دقت بالا و میلیون‌ها پارامتر نیز ممکن است در برابر تغییرات بسیار ظریف، رفتار غیرمنتظره‌ای داشته باشند.

در سال‌های اخیر، حملاتی مانند FGSM و PGD به ابزارهای اصلی برای بررسی امنیت مدل‌ها تبدیل شده‌اند. این روش‌ها با بهره‌گیری از گرادیان مدل، تغییراتی هدایت‌شده در داده‌ها اعمال می‌کنند تا شبکه را وادار به خطا کنند. بررسی نحوه عملکرد مدل در برابر این حملات، نه‌تنها اهمیت robustness را آشکار می‌سازد، بلکه باعث آشنایی با مفاهیمی مانند حساسیت ویژگی‌ها و رفتار معماری‌های مختلف می‌شود.

در این تمرین، هدف این است که حملات متخاصم را پیاده‌سازی و تحلیل کنید، تفاوت مقاومت معماری‌های مختلف را مقایسه کنید، و در نهایت با adversarial training مقاومت مدل را افزایش دهید.

۴-۱. پیاده‌سازی حملات FGSM و PGD (۳۰ امتیاز)

در این بخش یک مدل CNN ساده را روی دیتاست CIFAR-10 آموزش داده‌شده و سپس حملات FGSM و PGD روی آن پیاده‌سازی خواهند شد.

۴-۱-۱. بارگذاری دیتاست CIFAR-10 و آموزش مدل CNN

ابتدا دیتاست CIFAR-10 را بارگذاری کرده و چهار نمونه‌ی تصادفی از تصاویر به همراه برچسب صحیح آن‌ها نمایش دهید. سپس یک شبکه‌ی عصبی کانولوشنی ساده شامل ۲ تا ۳ لایه‌ی کانولوشن طراحی و پیاده‌سازی کنید. مدل باید برای حداقل ۱۰ ایپاک آموزش داده شود. در پایان فرآیند آموزش، نمودار تغییرات دقت (accuracy) و خطا (loss) برای داده‌های آموزش و اعتبارسنجی رسم و گزارش شود.

۴-۱-۲. پیاده‌سازی حملات FGSM و PGD

پس از آموزش مدل، حمله‌ی FGSM را برای سه مقدار مختلف ε شامل 0.01، 0.05 و 0.1 پیاده‌سازی کنید. همچنین حمله‌ی PGD را با تعداد تکرارهای k برابر با 5 و 10 اجرا نمایید. مقدار گام به‌روزرسانی α را متناسب با ε انتخاب کنید و توضیح دهید که این انتخاب چه تأثیری بر شدت و پایداری حمله دارد.

۴-۱-۳. ارزیابی مدل تحت حمله

برای هر مقدار ε، دقت مدل را در سه حالت مختلف محاسبه و گزارش کنید. این سه حالت شامل عملکرد مدل روی تصاویر clean، عملکرد مدل تحت حمله‌ی FGSM و عملکرد مدل تحت حمله‌ی PGD هستند. نتایج باید به‌صورت عددی و قابل مقایسه ارائه شوند.

۴-۱-۴. نمودارها

نمودار دقت مدل برحسب ε را رسم کنید. در این نمودار باید چهار منحنی شامل دقت روی داده‌هایclean ، دقت تحت حمله‌ی FGSM و دقت تحت حمله‌ی PGD با k=5 و k=10 نمایش داده شوند.

در گزارش تحلیل کنید که کدام حمله مخرب‌تر است و چرا، پارامتر k در حمله‌ی PGD چه نقشی دارد و افزایش آن چه اثری بر قدرت حمله می‌گذارد، و همچنین با افزایش ε دقت مدل چگونه تغییر می‌کند و علت این رفتار چیست.

۴-۲. مقایسه مقاومت دو معماری (۳۰ امتیاز)

در این بخش آزمایش می‌کنیم که معماری شبکه چه تأثیری در مقاومت در برابر حملات adversarial دارد.

۴-۲-۱. آموزش دو مدل و اعمال حملات FGSM و PGD

دو مدل مختلف را روی دیتاست CIFAR-10 در نظر بگیرید. مدل اول همان CNN ساده‌ی بخش قبل است که نیازی به آموزش مجدد آن نیست. مدل دوم ResNet18 بوده که با استفاده از وزن‌های pretrained آن را 10 ایپاک آموزش دهید و مانند بخش قبل نمودارهای مورد نیاز را رسم کنید. برای هر دو مدل، حملات FGSM و PGD را دقیقاً با همان تنظیمات و پارامترهای بخش ۱-۱ اجرا کنید تا مقایسه‌ای منصفانه حاصل شود.

۴-۲-۲. مقایسه نهایی

مانند بخش قبل، نمودار دقت مدل برحسب ε را رسم کنید. با این تفاوت که این بار نتایج هر دو مدل باید در نمودار حضور داشته باشند. برای هر مدل، چهار منحنی شامل دقت clean، دقت تحت FGSM و دقت تحت PGD با k=5 و k=10 رسم می‌شود، بنابراین در مجموع هشت منحنی در نمودار نهایی وجود خواهد داشت.

در گزارش تحلیل کنید کدام مدل مقاوم‌تر است و چرا، و نقش عواملی مانند عمق شبکه، وجود skip connection و smoothness تابع تصمیم‌گیری در معماری ResNet را توضیح دهید.

۴-۳. مقاوم‌سازی مدل با Adversarial Training (۴۰ امتیاز)

در این بخش باید نشان دهید که adversarial training چه تاثیری بر روی بهبود robustness چیست.

۴-۳-۱. adversarial training

در این مرحله، هر دو مدل CNN و ResNet18 باید به‌مدت ۱۰ ایپاک با استفاده از نمونه‌های adversarial تولیدشده توسط حمله‌ی PGD آموزش داده شوند. به این معنا که در فرآیند آموزش، به‌جای استفاده‌ی صرف از داده‌های clean، ورودی‌های آموزش شامل تصاویر متخاصمی باشند که با PGD تولید شده‌اند تا مدل به‌طور مستقیم در برابر این حمله مقاوم‌سازی شود. هدف از این نوع آموزش، افزایش robustness مدل نسبت به حمله‌ی PGD است.

در طول آموزش، برای هر ایپاک باید دو مقدار برای دقت و خطا گزارش شود؛ یکی مربوط به ارزیابی روی داده‌های اعتبارسنجی clean و دیگری مربوط به ارزیابی روی داده‌های اعتبارسنجی تحت حمله‌ی PGD. در نمودارهای نهایی آموزش، روند آموزش، عملکرد مدل روی داده‌های clean و عملکرد آن روی داده‌های adversarial باید به‌طور هم‌زمان نمایش داده شوند.

۴-۳-۲. ارزیابی پس از دفاع

پس از پایان adversarial training، عملکرد مدل‌ها را در سه حالت مختلف ارزیابی و مقایسه کنید: تصاویر clean، تصاویر تحت حمله‌ی FGSM و تصاویر تحت حمله‌ی PGD. این مقایسه باید برای هر دو مدل انجام شود.

۴-۳-۳. نمودارها

مانند بخش‌های قبل، نمودار دقت مدل برحسب ε را رسم کنید، با این تفاوت که این بار نتایج مدل‌های قبل و بعد از adversarial training به‌طور هم‌زمان مقایسه می‌شوند. در مجموع سه نمودار مورد نیاز است. نمودار اول عملکرد مدل‌ها را قبل و بعد از adversarial training روی داده‌های clean مقایسه می‌کند. نمودار دوم مقایسه‌ی مشابهی را تحت حمله‌ی FGSM نشان می‌دهد. نمودار سوم عملکرد مدل‌ها را تحت حمله‌ی PGD با مقادیر k=5 و k=10 نمایش می‌دهد.

در گزارش خود توضیح دهید که adversarial training چه تأثیری بر دقت مدل روی داده‌های clean دارد و چرا ممکن است این دقت کاهش یا ثابت بماند. همچنین adversarial training را به‌عنوان شکلی از data augmentation سخت تحلیل کنید و توضیح دهید که آیا مقاوم‌سازی مدل نسبت به یک حمله (PGD) می‌تواند بر عملکرد آن در برابر حملات دیگر مانند FGSM تأثیرگذار باشد.