به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی برق و کامپیوتر

درس یادگیری عمیق با کاربرد در بینایی ماشین و پردازش صوت

تمرین شماره ۳

اردیبهشت ۱۴۰۰

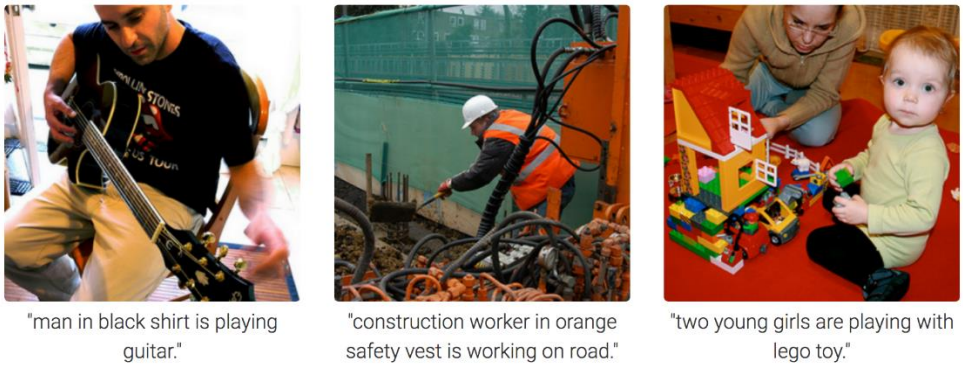
❖ مقدمه

شبکه عصبی بازگشتی یا Recurrent Neural Network (RNN)، نوعی از شبکه عصبی مصنوعی است که در تشخیص گفتار، پردازش زبان طبیعی (NLP) ، پردازش احساسات و موارد بسیار زیاد دیگری استفاده می‌شود. بسیاری از شبکه‌های عمیق مانند CNN شبکه‌های پیش خور (Feed Forward) هستند یعنی سیگنال در این شبکه‌ها فقط در یک جهت از لایه ورودی، به لایه‌های مخفی و سپس به لایه خروجی حرکت می‌کند و داده‌های قبلی به حافظه سپرده نمی‌شوند. اما شبکه‌های عصبی بازگشتی (RNN) یک لایه بازخورد دارند که در آن خروجی شبکه به همراه ورودی بعدی، به شبکه بازگرداننده می‌شود. RNN می‌تواند به علت داشتن حافظه داخلی، ورودی قبلی خود را به خاطر بسپارد و از این حافظه برای پردازش دنباله‌ایی از ورودی‌ها استفاده کند. به بیان ساده، شبکه‌های عصبی بازگشتی شامل یک حلقه بازگشتی هستند که موجب می‌شود اطلاعاتی را که از لحظات قبلی بدست آورده‌ایم از بین نروند و در شبکه باقی بمانند. در این سری از تمرینات، هدف آشنایی با کاربردهای این شبکه می‌باشد و از آنجایی که پیاده‌سازی این شبکه‌ها با پیچیدگی‌های زیادی همراه می‌باشد، بنابراین شما مجاز هستید که در این سری از تمرینات از امکانات کتابخانه Pytorch استفاده نمایید.

❖ سوالات

• سوال اول : Image Captioning

یکی از حوزه‌های جذاب در یادگیری ماشین، توصیف یک عکس توسط یک جمله می‌باشد. در واقع، هدف ما ایجاد و آموزش یک مدل می‌باشد که بتواند یک تصویر را به عنوان ورودی بگیرد و در نهایت یک جمله جهت توصیف آن عکس در خروجی خود تولید کند. تصویر زیر نمونه‌ای از خروجی این شبکه را نشان می‌دهد.



شکل ۱: خروجی یک مدل آموزش دیده جهت Image Captioning

حال در این تمرین قصد داریم یک مدل جهت رسیدن به این هدف پیاده‌سازی نماییم. ساختار کلی این مدل‌ها به این صورت است که یک شبکه CNN جهت تولید ویژگی‌های تصاویر وجود دارد و در کنار آن روش‌های

مختلفی برای Embedding جملات موجود است که در نهایت بردار ویژگی تصاویر و متن در کنار هم قرار گرفته و به عنوان ورودی یک شبکه بازگشتی اعمال می‌شود تا در نهایت جمله نهایی را تولید نماید. در ادامه بیشتر با بخش‌های مختلف آن آشنا خواهید شد.

- **بخش اول: مجموعه دادگان و پیش‌پردازش آن‌ها**

مجموعه دادگان مورد استفاده در این تمرین flickr8K می‌باشد که جز مجموعه دادگان با سایز کوچک در حوزه Image Captioning می‌باشد. این مجموعه را می‌توانید از لینک زیر دانلود نمایید.

https://www.kaggle.com/adityajn105/flickr8k

این مجموعه از دو بخش به نام Images و Caption.txt تشکیل شده‌است که پوشه Images شامل ۸۰۹۱ تصویر می‌باشد و Caption.txt شامل 40455 جمله می‌باشد که برای هر تصویر ۵ جمله مختلف توسط افراد مختلف جمع‌آوری شده‌است. در کنار هر جمله نام تصویر مورد نظر نیز آورده شده‌است.

با آماده‌سازی تصاویر برای اعمال به شبکه‌های کانولوشنی پیش‌تر آشنا شدید، در اینجا جملات نیز باید پیش‌پردازش شوند تا به بردارهایی از اعداد تبدیل شوند. ما در اینجا برای سادگی کار از لایه Embedding در پایتورچ استفاده می‌کنیم که نحوه کار با این لایه را در لینک زیر مشاهده می‌کنید.

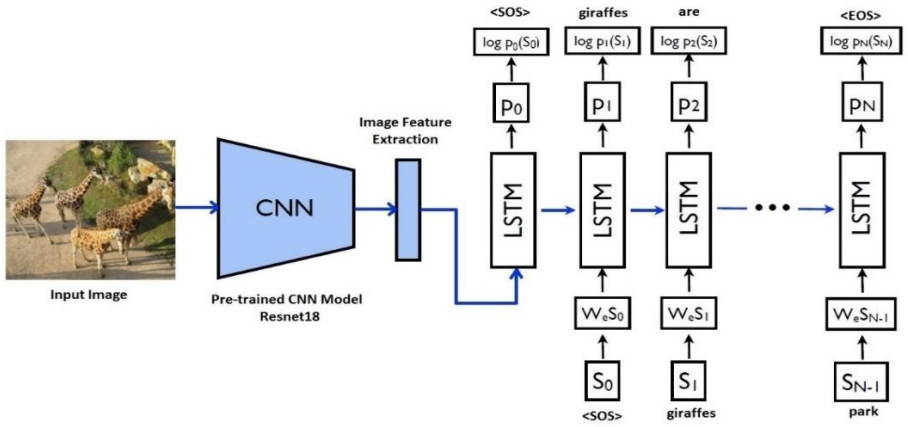
https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html

پارامتری به نام Embedding_dim در آن وجود دارد که می‌توانید آن را ۳۰۰ در نظر بگیرید که البته انتخاب آن در اختیار شما می‌باشد که درواقع این عدد مشخص می‌کند که برای هر کلمه یک بردار عددی با طول ۳۰۰ در نظر بگیرد. نکته مهمی که در پیش‌پردازش داده‌ها باید توجه نمایید، این است که باید برای هر جمله از توکن‌های شروع و پایان جمله یعنی <SOS> و <EOS> استفاده نماییم. که توکن‌های خاصی می‌باشند که توسط خود شما تعریف می‌شوند. همچنین باید مجموعه لغات موجود در دیتاست خود را پردازش و به هر کدام از آن‌ها یک Index نسبت دهید. بهتر است علامت‌های نگارشی از جملات حذف شوند. همچنین از آنجایی که جملات Captionها طول‌های متفاوتی دارند باید طول آن‌ها باهم یکسان شوند، که این کار را با Padding مناسب می‌توانید انجام دهید که می‌توان یک طول مشخص ثابت را درنظر گرفت یا یکسان‌سازی را در هر مینی‌بچ انجام داد.

- **بخش دوم: مدل شبکه**

در شکل شماره۲ مدل کلی مد نظر این سوال را مشاهده می‌کنید. همانطور که مشاهده می‌کنید، بخشی از مدل جهت استخراج ویژگی تصاویر مورد استفاده قرار می‌گیرد. در این مسئله ما قصد داریم از یک مدل از

پیش‌آموزش داده شده Resnet18 استفاده نماییم. این مدل در کتابخانه پایتورچ قابل دسترس می‌باشد و از آخرین لایه شبکه کانولوشنی آن ویژگی‌های تصویر استخراج می‌شود که در نهایت نیاز است به یک لایه خطی جهت استخراج ویژگی‌های مورد نظر با ابعاد مناسب جهت ورود به شبکه بازگشتی، استفاده نمود.

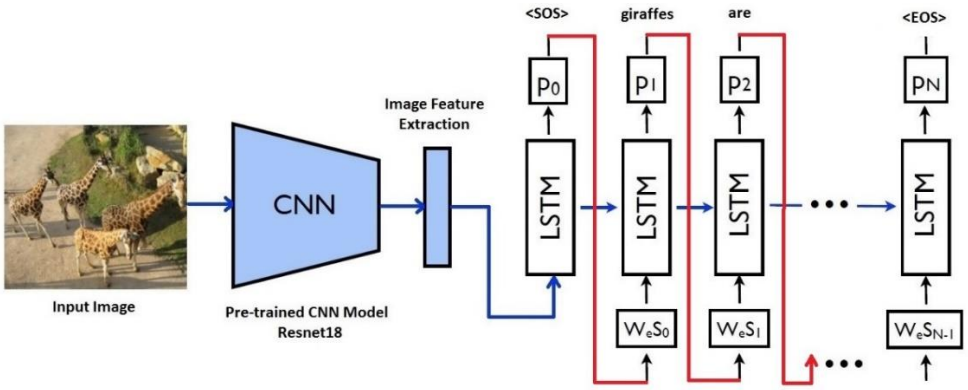


شکل۲: تصویر مدل مورد بررسی در سوال اول

در این قسمت از یک لایه شبکه LSTM با تعداد ۲۵۶ لایه پنهان استفاده می‌نماییم و بردارهای Embed شده جملات در کنار بردار تصویر به آن داده شده و خروجی آن به یک لایه خطی به سایز ورودی Hidden State و سایز خروجی تعداد کلمات موجود در دیتاست اعمال می‌شود و به این ترتیب به محاسبه خطا و پیش‌بینی مدل می‌پردازیم.

- **بخش سوم: پیش‌بینی شبکه**

بعد از آموزش شبکه، نیاز دارید تا شبکه را ارزیابی نمایید. جهت ارزیابی شبکه باید به صورت شکل زیر از شبکه استفاده نماییم.



شکل۳: نحوه استفاده از مدل در زمان تست جهت تولید جمله

همانطور که می‌دانیم در زمان تست شبکه آموزش داده شده، Caption وجود ندارد و ما باید برای یک تصویر Caption تولید نماییم. برای این منظور روش‌های مختلفی وجود دارد ولی ما در اینجا مدل بالا را

پیشنهاد می‌دهیم. در یک تابع به عنوان ورودی، تصویر تست و مدل آموزش داده شده را جهت پیش‌بینی کلمات اعمال می‌کنیم. قطعه کد زیر الگوریتم این شبکه را نمایش داده‌است.

```
input_data = Trained_Model.CNN(image)
states = None  #(Hn, Cn)

for _ in range(max_length):
    hiddens, states = Trained_Model.lstm(input_data, states)

    output = Trained_Model.linear(hiddens)

    predicted_index = output.argmax()

    input_data = Trained_Model.Embedding(predicted_index)

    caption_prediction.append(predicted_index)

    if predicted_index.item() == "<EOS>":
        break
```

شکل۴: الگوریتم بازگوکننده شبکه شکل۳ جهت تولید جمله

در نهایت caption_prediction مجموعه indexهای کلمات می‌باشد که در نهایت به کمک دایره لغات موجود در مجموعه دادگان قابل تبدیل به کلمات می‌باشد. توجه داشته باشید که الگوریتم فوق فقط مراحل کار را نشان داده‌است و نیاز به بازنویسی درست، رعایت ابعاد تنسورها و غیره دارد که بر عهده شما می‌باشد. البته استفاده از هر شیوه دیگری جهت تست و تولید جملات بلامانع می‌باشد.

○ بخش چهارم: سوالات

۱) در مرحله اول، از یک مدل از پیش‌آموزش داده شده Resnet18 به عنوان شبکه CNN استفاده نمایید و به جز لایه خطی آخر تمامی لایه‌های آن را Freeze نمایید تا در عملیات بروزرسانی وزن‌ها شرکت نداشته باشند. سپس خروجی آن را در کنار بردارهای Embed شده جملات به یک لایه شبکه LSTM یکطرفه اعمال کرده و نمودار خطای آموزش و تست را در طول یادگیری گزارش نمایید. از تابع خطای CrossEntropy و تابع بهینه‌ساز Adam می‌توانید استفاده نمایید. بعد از فرآیند آموزش، ۳ عدد عکس از دادگان تست را جهت پیش‌بینی مدل، به آن اعمال کرده و خروجی آن را در گزارش کار خود ذکر نمایید.

۲) با حفظ موارد قبلی حال تمامی لایه‌های شبکه Resnet18 را Unfreeze نمایید و مجددا موارد خواسته شده در سوال قبل را بررسی نمایید و نتایج بدست آمده را با مرحله قبل مقایسه کنید.

۳) (**امتیازی**) بجای یک لایه LSTM یکطرفه، از یک لایه LSTM دوطرفه استفاده نمایید و مجددا مراحل قبل را تکرار کنید.

- **سوال دوم**
  - **مقدمه**

در این سوال میخواهیم از شبکه های عمیق بازگشتی برای کاربرد استخراج روابط یا Relation Extraction استفاده کنیم.

  - **تعریف مسئله**

در میان مسائل اصلی پردازش زبان طبیعی، استخراج روابط یا Relation Extraction نقش اساسی در بسیاری از کاربرد ها از جمله اخبار، متون وبسایت ها و زمینه های بیوپزشکی دارد. در نتیجه استخراج روابط یک موضوع داغ تحقیقاتی در سالهای اخیر شده است.

استخراج روابط مسئله یافتن ارتباطات معنایی بین جفت موجودیت ها یا entity ها (کلمات، عبارات و حتی جمله ها) در یک متن و تخصیص طبقه مشخصی از این رابطه است. برای مثال نمونه زیر را در نظر بگیرید:

هفته گذشته ما به پارک چیتگر که در غرب تهران واقع است رفتیم.

در جمله بالا "پارک چیتگر" موجودیت یا عبارت اول ما و "غرب تهران" موجودیت دوم ما هستند و بین این دو رابطه قرارگیری-در برقرار است.

  - **دادگان**

در این سوال ما از دادگان محبوب SemEval-2010 Task 8 استفاده میکنیم که شامل ۸۰۰۰ جمله اموزش و ۲۷۱۷ جمله تست است. جملات این دیتاست در ۹ دسته اصلی زیر قرار میگیرند.

(1) Cause-Effect

(2) Instrument-Agency

(3) Product-Producer

(4) Content-Container

(5) Entity-Origin

(6) Entity-Destination

(7) Component-Whole

(8) Member-Collection

(9) Message-Topic

هر سمپلی که در ۹ دسته بالا قرار نگیرد در طبقه بندی لیبل Other (۱۰) به آن تعلق میگیرد. به دلیل عدم توازن در توزیع جملات این دیتاست. تنها از ۷۱۱۰ جمله ابتدایی آن که از هر ۹ دسته گرفته شده اند استفاده کنید. (۸۹۱ جمله انتهایی تنها از پنج دسته اول هستند) برای دستیابی به این دیتاست از این لینک استفاده کنید. داده ای آموزش را به دو بخش آموزش و اعتبارسنجی تقسیم کنید.

- **ساختار مدل**

برای حل مسئله RE یکی از انواع شبکه های بازگشتی یعنی شبکه Bidirectional LSMT را استفاده خواهید کرد. از آنجایی که با داده های متنی کار میکنید ابتدا نیاز است تا این داده ها را بخش بخش (tokenize) کنید. برای استفاده از شبکه LSTM باید هر token از جملات ورودی پس از tokenization به یک بردار ویژگی نظیر شود (دقت کنید که به یک مرحله padding نیز نیاز خواهید داشت تا بردار های متناظر با جملات یک batch طول برابر داشته باشند. راهنمایی: برای padding باید این نکته را نیز در نظر بگیرید که در مسئله RE موجودیت ها ممکن است در هر محل از جمله ظاهر شوند). برای این کار نیاز است تا پیش از اعمال ورودی به شبکه Bi-LSTM یک لایه Embedding قرار داده شود که هر جمله را به برداری از بردار token های embed شده تبدیل کند. برای آموزش شبکه های هر سه بخش زیر از negative log-likelihood استفاده کنید.

نمودار های خطا و دقت (loss , accuracy) را برای دادگان آموزش و اعتبار سنجی رسم کنید. معیار های Precision, Recall و F1-score را برای دادگان تست گزارش کنید.

- **بخش اول**

ابتدا شبکه Bi-LSTM را با لایه Embedding که وزن های آن به صورت تصادفی انتخاب شده اند ترکیب کنید. در این بخش می بایست تا وزن های لایه ی Embedding را نیز در کنار وزن های LSTM آموزش دهید. برای طبقه بندی، خروجی اخرین لایه مخفی از مسیر forward و backward را از شبکه Bi-LSTM بگیرید و با هم concatenate کرده و به یک شبکه خطی fully-connected بدهید. خروجی لایه خطی را از یک لایه softmax.

ساختار کلی شبکه به صورت جدول زیر است.

Embedding	embedding dimension: 100
Bi-LSTM	number of layers: 2 hidden size: 150
Linear	
Softmax	

- **بخش دوم**

در این بخش به جای مقدار دهی تصادفی از مقادیر بردار های ویژگی آموزش دیده GloVe استفاده کنید. به این منظور از این لینک فایل glove.42B.300d.zip را دانلود کنید. وزن های Embedding را با وزن های GloVe اما آنها آموزش ندهید و ثابت نگه دارید. مراحل بخش اول را اجرا کنید و نتایج را با حالتی که مقدار دهی تصادفی به بردار های embedding داشتید مقایسه کنید. زمان آموزش شبکه و نرخ همگرایی را نیز جز معیار های مقایسه خود قرار دهید.

- **بخش سوم**

همانطور که در دادگان این سوال مشاهده کردید، هر جمله دارای دو موجودیت (entity) و رابطه میان آنها است. در مسائلی مانند RE که نمیخواهیم تمام جمله را طبقه بندی کنیم، بهترین راه ممکن است تنها استفاده از خروجی اخرین لایه token در جمله نباشد. به این منظور در این بخش ابتدا برای هر token از موجودیت اول و دوم خروجی لایه مخفی متناظر با آن را در مسیر forward و backward گرفته و از روی این دو بردار max-pooling اعمال کنید تا برداری با ابعاد برابر با لایه مخفی LSTM حاصل شود. سپس روی بردارهایی که برای token های موجودیت اول بدست آورید average-pooling انجام دهید. این کار را برای token های موجودیت دوم نیز انجام دهید. در نهایت باید برای هر کدام از موجودیت اول و دوم یک بردار بازنمایی با ابعاد برابر با ابعاد لایه مخفی LSTM بدست اوردید. حال این دو بردار را با هم concatenate کنید و به عنوان ورودی به شبکه خطی بدهید و در نهایت امتیاز ها را توسط softmax محاسبه کنید و هر رابطه را طبقه بندی کنید. در این بخش نیز مانند بخش دوم وزن های لایه Embedding را با GloVe جایگزین کرده و آنها را آموزش ندهید. برای این بخش علاوه بر مقادیر recall, precision و F1-score ماتریس در هم ریختگی (confusion matrix) را رسم و آن را تحلیل کنید

نکات:

- مهلت تحویل این تمرین، تا پایان روز پنجشنبه ۶ خرداد ماه می‌باشد.
- انجام این تمرین به صورت انفرادی می‌باشد.
- برای انجام این تمرین استفاده از امکانات کتابخانه Pytorch بلامانع می‌باشد.
- داخل کدها کامنت‌های لازم را قرار دهید و تمامی موارد مورد نیاز برای اجرای صحیح کد را ارسال کنید.
- **در صورت مشاهدهٔ موارد تشابه بین دو یا چند فرد در گزارش کار و یا کد ، به طرفین تقلب نمره صفر داده خواهد شد و هیچ گونه عذر و بهانه‌ای از جمله ارسال کد به دوست خود و عدم آگاهی از کپی برداری کد شما پذیرفته نخواهد شد، بنابراین به هیچ عنوان کدهای خود را در اختیار دیگران قرار ندهید در غیر این صورت مسئولیت تقلب بر عهده شما نیز می‌باشد. همچنین کپی برداری از کدهای آماده موجود در اینترنت و یا استفاده از کدهای افراد ترم‌های گذشته تفاوت چندانی با تقلب ندارد و در چنین مواردی نیز نمره صفر به فرد تعلق می‌گیرد و جای هیچگونه اعتراضی وجود ندارد.**
- اگر بخشی از کد را از کدهای آماده اینترنتی استفاده می‌کنید که جزء قسمت‌های اصلی تمرین نمی‌باشد، حتما باید لینک آن در گزارش و کد ارجاع داده شود.
- گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است و نیمی از نمره شما را دربرخواهد گرفت. لطفاً تمامی نکات و فرض‌هایی که برای پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید را در گزارش ذکر کنید و تمامی اصول نگارشی را مطابق با فایل ارسالی در صفحه درس رعایت بفرمایید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست. اما باید نتایج بدست آمده را گزارش و تحلیل کنید.
- برای پیاده‌سازی می‌توانید از محیط Colab استفاده نمایید.
- لطفا گزارش (درقالب PDF) ، فایل کدها و سایر ضمائم مورد نیاز را با فرمت زیر در صفحه درس در سامانه یادگیری الکترونیکی بارگذاری نمائید.

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

- در صورت وجود هرگونه ابهام یا مشکل می‌توانید بر اساس شماره سوال از طریق رایانامه‌های زیر با دستیاران آموزشی مربوطه در تماس باشید.
  - **سوال اول:**
    - حسین پورمهرانی : h.pourmehrani@gmail.com
  - **سوال دوم:**
    - پیمان باقرشاهی : p.baghershahi@ut.ac.ir