باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

**۲۵۶۴۷-یادگیری عمیق-ترم پاییز ۱۴۰۰**

**تمرین سری سوم**

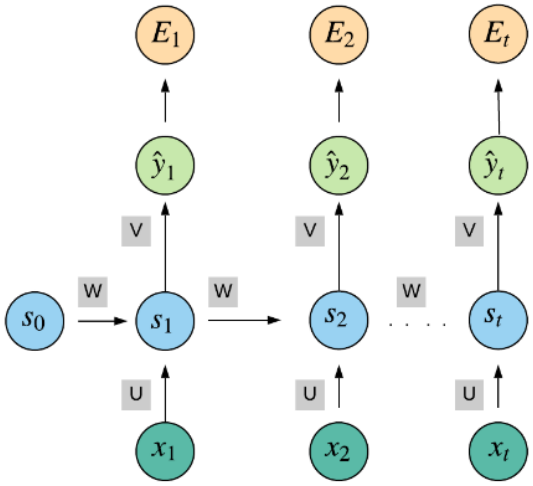
**موعد تحویل: ۱۲ دی ۱۴۰۰**

## نحوه تحویل:

- آپلود در CW در قالب یک فایل واحد با نام HW_03_stdnum.zip که stdnum شماره دانشجویی شما در دانشگاه صنعتی شریف می‌باشد.
- کدهای سوالات کامپیوتری را به صورت فایل‌های جداگانه و در فرمت ipynb تحویل‌دهید.
- دربرخی از قسمت‌ها نیاز به استفاده از تنسوربورد بوده, از این رو فایل Summary خود را حتما تحویل‌دهید.
- درصدهای طبقه بندی شبکه شما می‌بایست, در حدنرمالی قراربگیرد, قطعا درصدهای بالاتر از حدنرمال و پایین‌تر, به ترتیب نمره اضافه و کسری نمره را به همراه خواهد داشت.
- ازآنجایی که در طراحی‌های خود, مقید به محدودیتی نیستید, که چنین رویکردی منجر به خروجی‌های متفاوتی در همه ابعاد خواهد بود, از این رو مشاهده تقلب در تمارین باعث از دست‌دادن کل یا بخشی از نمره تمرین هر دو طرف می‌شود.
- در تکالیف شبیه‌سازی سهم عمده نمره تکلیف را تحلیل و دریافت شما از نتایج کدهای نوشته شده، دارد.
- از اجرای کدهای خود اطمینان حاصل فرمایید، ترجیحا برای هماهنگی از version پکیج‌های مورد استفاده از فایل requirements.txt پکیج منیجر pip استفاده‌کنید. برای اطلاعات بیشتر به PEP ۴۹۶ یا PEP ۵۰۸ مراجعه‌کنید.
- برای تمارین کامپیوتری، استفاده از Git نمره‌ی امتیازی دارد، سعی‌کنید حتماً تعدادی Branch، Commit و Merge در Version Control خود داشته‌باشید. Branchهای Merge شده را Remove نکنید. به نکاتی مانند قرارندادن تصاویر و فایل‌هایی که به سورس کدتان مربوط‌نمی‌شود نیز دقت‌کنید.

# تمرینات تئوری (۳۰نمره):

۱. شبکه بازگشتی زیر را در نظر بگیرید که در آن $s_i$ و $y_i$ به ترتیب حالت مخفی و خروجی می باشند. U، V و W هم وزن هایی هستند که ماژول های مختلف را به یکدیگر وصل می کنند.




فرض کنید E خطای کل شبکه است و $E_i$ خطای خروجی i ام می باشد. مقادیر $\frac{\partial E_3}{\partial W}$ و $\frac{\partial E_3}{\partial V}$ را محاسبه کنید (همراه با مراحل میانی). همچنین ابعاد آن ها و تمامی مراحل میانی را نیز مشخص کنید. (ورودی شبکه را n بعدی و بردارهای نهان را d بعدی و تعداد کلاس های خروجی را k می باشد.)

۲. این مقاله را مطالعه نمایید و به سوالات زیر پاسخ دهید:

الف) یک تابع Attention چیست؟ شکل ورودی آن به چه صورت می‌باشد؟

ب) علت اسکیل‌کردن ضرب داخلی در واحد Attention چیست؟

ج) مقاله به چه شکل‌هایی از Attention چندسَر در ماژول پیشنهادی خود استفاده می‌کند؟

د) پیشنهاد مقاله برای استفاده از خاصیت ترتیبی ورودی به جای کانولوشن چیست؟ مقایسه‌ی انجام‌شده در مقاله بین روش‌های ممکن را شرح دهید.

ه) بر مبنای بخش ۴ مقاله، مقایسه‌ای بین Self-Attention و لایه‌های بازگشتی و کانولوشن انجام دهید.

و) حال معماری ترانسفورمر، بخش اصلی مقاله که منجر به محبوبیت آن شد را با عنایت به قسمت‌های قبلی شرح دهید. از چه تابع بهینه‌سازی استفاده شده‌است؟ از چه روشی برای Regularization بهره گرفته‌شد؟

(امتیازی) ز) استفاده از ترانسفورمر منتهی به کاربردهای NLP نشده‌است. برای مثال در سال ۲۰۲۰، تیمی از گوگل یک ساختار مبتنی بر ترانسفورمر برای طبقه‌بندی تصاویر به نام Vision Transformer (ViT) ارائه داد که جزئیات آن در این مقاله آمده‌است. خلاصه‌ای از روند پیشنهادی این تیم و ساختار ViT ارائه دهید. نیازی به ورود به جزئیات نمی‌باشد.

## تمرینات کامپیوتری(۷۰+۱۰نمره):

۱. در این سوال هدف پیاده سازی تسک تحلیل احساسات (sentiment analysis) است و در آن از شبکه های بازگشتی (RNN) و شبکه BERT استفاده خواهید کرد. مجموعه داده مورد استفاده در این سوال، دیتاست Sentiment140 می باشد که می توانید آن را از اینجا دانلود کنید و با جزییات آن آشنا شوید. کتابخانه مورد استفاده در این تمرین، PyTorch یا TensorFlow می باشد.

ابتدا نیاز است تا یک سری مراحل پیش پردازشی بر روی داده ها انجام دهید. بدین منظور، ابتدا کارهای زیر را در ابتدا انجام دهید:

- هر یک از نمونه های این دیتاست یک توییت است. هشتگ ها و منشن ها را با یک عبارت یا کلمه جایگزین کنید.

- توییت های حاوی لینک را حذف کنید تا در آموزش مدل مشکلی ایجاد نکنند.

- تمام علایم نگارشی را از توییت ها حذف کنید.

- برچسب داده ها را به صفر (منفی)، یک (خنثی) و دو (مثبت) تغییر دهید.

حال باید به هر کلمه به کمک یک word embedding درحالت ساده یک عدد یکتا یا یک بردار one-hot نسبت دهیم، برای این منظور از glove42b با بعد ۳۰۰ استفاده کنید. این embedding هر واژه را به یک بردار ویژگی با بعد ۳۰۰ نگاشت می کند و می توانید آن را از اینجا دانلود کنید. (با توجه به اینکه طول کلمات متفاوت است قبل از اعمال embedding بر روی آنها می توانید از padding مناسب استفاده کنید برای مثال می توانید طول همه ورودی ها را ۲۸۰ بکنید.)

الف) مدل LSTM یک طرفه با یک لایه و بعد مخفی ۱۵۰ و یک لایه خطی با ۳ خروجی بدهید. از تابع فعالساز Softmax در لایه خروجی استفاده کنید. بهینه ساز خود را Adam و تابع هزینه را Cross entropy در نظربگیرید. مدل را آموزش دهید و نمودار دقت و خطا را برای داده های آموزش، اعتبارسنجی و تست رسم کنید همچنین ماتریس confusion را گزارش کنید. (اگر یک لغت در word embedding وجود نداشته باشد در این حللت به جای بردار آن لغت، میانگین تمامی بردارهای ویژگی موجود در glove42b را در نظربگیرید)

ب) در قسمت الف، LSTM دوطرفه را جایگزین کنید و مجدد نمودار دقت و خطا را برای داده های آموزش، اعتبارسنجی و تست رسم کنید همچنین ماتریس confusion را گزارش کنید. نتایج را با قسمت قبل مقایسه و تحلیل کنید.

ج) مدل LSTM یک طرفه بخش اول را با یک ساختار هرمی (Pyramid) با ۴ سطح جایگزین کنید. ساختار هرمی به این صورت است که در هر سطح، ورودی هر سلول از اتصال خروجی دو سلول سطح پایین تر ساخته می شود. برای آشنایی بیشتر با ساختار هرمی می توانید به این مقاله مراجعه کنید. بعد مخفی پایین ترین سطح را ۶۴ درنظر بگیرید و نمودار دقت و خطا را برای داده های آموزش، اعتبارسنجی و تست رسم کنید همچنین ماتریس confusion را گزارش کنید.

نتایج را با قسمت های قبل مقایسه و تحلیل کنید. (در صورت طولانی شدن زمان آموزش می توانید تنها از نیمی از داده های آموزش استفاده کنید.)

*د) (امتیازی) در این قسمت با استفاد از یکی از مدل BERT از قبل آموزش داده شده، تحلیل احساسات باید انجام دهید و مدل را بر روی دادگان سوال fine-tune خواهید کرد. برای آشنایی با ساختار شبکه BERT می توانید از مراجع زیر استفاده کنید:

مرجع یک مرجع دو مرجع سه

در این قسمت از کتابخانه pytorch_pretrained_bet یا برای افرادی که از TensorFlow استفاده میکنند از این لینک استفاده کنید.

ابتدا داده ها را به tekenizer دهید. وظیفه tokenizer مشابه یک word embedding است. در واقع، tokenizer هر جمله را به توکن های از پیش تعریف شده تقسیم می کند و سپس به هر کدام از آن ها یک آیدی نسبت می دهد. در نهایت، این آیدی ها ورودی مدل BERT خواهند بود. طول ورودی ها را ۳۰۰ درنظر بگیرید اگر طول آیدی استخراج شده توسط tokenizer کمتر از ۳۰۰ باشد به انتای آن صفر اضافه کنید تا طول آن برابر با ۳۰۰ شود و اگر طول آن بزرگتر از ۳۰۰ باشد تنها ۳۰۰ توکن ابتدایی را در نظر بگیرید. در این قسمت از مدل و tokenizer آموزش داده شده bert-base-uncase استفاده کنید (این مدل دارای ۱۲ لایه با بعد مخفی ۷۶۸ هست). خروجی مدل BERT را به یک لایه خطی با ۳ خروجی بدهید و در نهایت تابع Softmax را بر روی آن اعمال کنید. مدل را آموزش دهید و نمودار دقت و خطا را برای داده های آموزش، اعتبارسنجی و تست رسم کنید همچنین ماتریس confusion را گزارش کنید. نتایج را با قسمت های قبل مقایسه و تحلیل کنید.

۲. در این تمرین به رفع نویز سیگنال‌های مختلف با شبکه‌های بازگشتی می‌پردازیم. در هر بخش حتما از نمایش سیگنال‌ها غافل نشوید.

الف) سیگنال‌های سینوسی، مثلثی و دندان اره‌ای به طول ۱۰ تناوب تولید کنید و ۷ دوره تناوب را برای آموزش و ۱ تناوب برای اعتبارسنجی و مابقی را برای تست اختصاص دهید.

ب) نویزگوسی با میانگین صفر و یونیفرم تولید نموده و جداگانه به هر سیگنال اضافه کنید.

ج) یک بار با شبکه RNN ساده و یک بار با LSTM مدل خود را روی سیگنال آموزش ترین نمایید و با استفاده از خطای کمترین مربعات، دقت هر شبکه را بسنجید. همچنین SNR خروجی را گزارش دهید. آماده‌سازی داده‌ها برای ترین و تنظیم پارامترهای شبکه بر عهده خودتان است و هدف دستیابی به بهترین دقت بازسازی است. (بررسی مدل پایه برای پارامترهای مختلف کفایت می‌کند و در صورت نیاز می‌توانید از لایه‌ی Dense استفاده نمایید. نیازی به پیاده‌سازی ساختارهای ذکر شده از پایه نیست.)

د) اثرات Learning Rate، Batch Size، Number of Epochs را با نمودارهای مناسب روی تابع هزینه آموزش به ازای واریانس‌های مختلف نشان دهید. تا چه واریانسی سیگنال اصلی به طور کیفی قابل بازیابی است؟

ه)نویز اضافه‌شده در قسمت ب ایستان می‌باشد اما اگر واریانس یا میانگین آن در زمان تغییر می‌کرد، آیا آموزش شبکه‌های RNN همچنان سرراست بود؟ این موضوع را با اضافه کردن نویز گوسی غیر ایستان که واریانس و میانگین آن متغیر در زمان است بررسی کنید و قسمت د را تکرار نمایید. به عنوان مثال می‌توانید سیگنال اصلی را به چندین بخش تقسیم کرده و در هر بخش از واریانس و میانگین متفاوت با بخش‌های دیگر استفاده نمایید.

و) حال به بررسی عملکرد شبکه‌ها در فضای دو بعدی می‌پردازیم. بدین منظور خم‌های پارامتریک زیر را در نظر گرفته و قسمت‌های ب الی ه را برای هر کدام از این خم‌ها تکرار نمایید. برای راحتی بگیرید $0 \le t \le 10 \cdot 2\pi$

- دایره به شعاع ۱ که مرکز آن در مبدا است. آیا لازم بود ده بار دایره را دور بزنیم؟
- $Epitrochoid$ با پرمایش زیر:

$$x(t) = (a+b)\cos t - c\cos\left(\left(\frac{a}{b}+1\right)t\right),$$

$$y(t) = (a+b)\sin t - c\sin\left(\left(\frac{a}{b}+1\right)t\right), a = c = 5, b = 3$$

- $Nephroid$ با پرمایش زیر:

$$x(t) = a(\cos 3t - 3\cos t), \qquad y(t) = a(\sin 3t - 3\sin t), a = 2$$

و) به دلخواه روی صدای ضبط‌شده خود یا یک قطعه موسیقی بخش‌های الف الی ه را تکرار نمایید. آیا نتایج مطلوب است؟ می‌توانید طول سیگنال را افزایش داده و آموزش را دوباره اجرا کنید. یک روش مورد استفاده برای افزایش کیفیت بازسازی صدا، استفاده از طیف سیگنال در حوزه‌ی زمان-فرکانس است به طوری که با اعمال فیلتر ضربی روی تبدیل زمان-فرکانس به دنبال جداکردن مولفه‌های نویز جمع شونده هستیم و شبکه‌های مختلفی برای تخمین این فیلتر پیشنهاد شده‌اند. یک روش به نام EHNET که برمبنای ترکیب CNN و RNN می‌باشد در این مقاله آمده است. ابتدا در مورد Bidirectional RNN تحقیق نمایید و سپس روش استفاده‌شده در مقاله مورد نظر را توضیح دهید. از چه تابع هزینه‌ای استفاده شده‌است؟ شکل ورودی‌های هر لایه چگونه است؟

ز) حال تصویر پیوست‌شده را به صورت Grayscale در نظر بگیرید و روند قسمت‌های الف الی ه را این بار با نویزهای گوسی و فلفل نمکی تکرار نمایید. آیا کیفیت بازسازی به نحوه‌ی انتخاب پیکسل‌های تصویر وابسته است؟ آیا آموزش مدل و دقت بازسازی همانند سیگنال یک بعدی است؟ در این قسمت بیشتر تلاش و توضیحات شما مدنظر می‌باشد.

(امتیازی: روی تصویر سه کاناله تکرار کنید)

حال این مقاله مطالعه نمایید و خلاصه‌ای از روش ارائه شده و مقایسه‌ی آن با MemNet و DRCN, DRRN (از پاراگراف قبل از Experiments می‌توانید استفاده کنید) شرح دهید. Non-Locality به چه معناست؟

چه نتیجه‌ای از این تمرین در مورد کاربرد مدل‌های پایه‌ای LSTM, RNN برای رفع نویز سیگنال‌ها می‌گیرید؟

۳. در این تمرین قصد داریم تا با استفاده از شبکه های بازگشتی و کانولوشنی، یک سیستم برای کپشن زدن بر روی تصاویر را ایجاد کنیم.

الف) جمع آوری داده: در این تمرین قصد داریم تا از داده های flickr8k استفاده کنیم که میتوانید از اینجا آن را دانلود کنید و یا مستقیما از نوتبوک کگل استفاده کنید تا بتوانیم بر روی تصاویر کپشن بزنیم. این دیتاست شامل ۸۰۰۰ تصویر که هر یک دارای ۵ کپشن است میباشد. پس از دریافت فایل سعی کنید تا شهود کلی از داده ها را بدست آورده و سپس داده های خود را بر اساس آی دی تصویر و کپشن در دیکشنری یا هر داده ساختار مناسبی ذخیره کنید.

ب) تمیز کردن داده های کپشن: برای کار با داده های متنی ابتدا باید بر روی داده ها پیش پردازش انجام دهیم. این پیش پردازش بسته به کاربرد میتواند انواع خاصی داشته باشد. در اینجا صرفا چند پیش پردازش ساده انتظار میرود. سعی شود تا تمام لغت های موجود در کپشن ها به فرم lowercase باشد. Punctuation ها و لغت هایی که شامل عدد در کپشن را حذف کنید و سعی کنید تا از کلمات تک حرفی نیز پرهیز کنید. (این کار با استفاده از کتابخانه string قابل انجام میباشد. طبیعتا پیش پردازش های بیشتر و مفیدتر شامل نمره امتیازی خواهد بود) بهتر است تا این کپشن هارا در فایلی در هارد دیسک ذخیره کنید و در ادامه کار هر جا لازم شد این فایل را لود کنید تا حافظه cache سیستم را اشغال نکند.

پ) برای کار با داده های متنی لازم است تا یک لغت نامه(vocabulary) داشته باشیم. لغت نامه از مجموعه لغت های یکتا در کپشن ها تشکیل میشود. سعی کنید تا در این بخش تمامی لغت های یکتا در کپشن ها استخراج کنید. در ادامه لغت هایی که کمتر از ۱۰ مرتبه در کپشن ها استخراج شده اند را دور ریخته و تعداد تمامی لغت ها را نشان دهید.

ت) حال در این قسمت لازم است تا داده های train را بسازیم. فایل flickr_8k.trainImages.txt دارای اسامی تصاویری که متعلق به داده های ترین است میباشد. ۶۰۰۰ هزار تصویر به عنوان داده ترین موجود است و سپس از فایلی که ذخیره کرده اید کپشن ها را نیز میتوانید بازیابی کنید. در اینجا به ابتدا و انتهای هر کدام از کپشن ها باید توکن های <START> و <END> نیز اضافه کنید. این توکن ها برای مشخص کردن ابتدا و انتهای جمله ضروری هستند.

ث) در این قسمت نیاز به نمایش برداری از تصاویر داریم. بنابراین سعی میکنیم تا از یک شبکه از پیش یادگیری شده کار transfer learning را انجام دهیم. این کار میتواند با هرشبکه ای که با دیتاست imagenet ترین شده است، انجام شود. توجه داشته باشید که در اینجا هدف ما دسته بندی نیست و صرفا یک طول ثابت برداری مدنظر است. بنابراین در اینجا با استفاده از فانکشن encode (که خودتان پیاده سازی میکنید) تصاویر ترین را به یک نمایش d بعدی تبدیل میکنید. به صورت مشابه میتوانید تصاویر تست را نیز به همین صورت به d بعد انتقال داده و سپس در دیسک ذخیر کنید.

ج) در نظر داریم که کپشن ها را میخواهیم کلمه به کلمه پیش بینی کنیم. بنابراین به صورت مشابه نیز نیاز داریم تا لغات را نیز به صورت یک نمایش برداری ثابت دربیاوریم. حال انتظار داریم تا دو دیکشنری بسازید. دیکشنری word2idx و idx2word. با اینکار میخواهیم تا لغات موجود در لغت نامه را با یک عدد integer نشان دهیم. توجه داشته باشید که توکنایزرهای از پیش آماده ای موجود هستند و اینکار را برای زبان های مختلف انجام میدهند اما در این تمرین انتظار میرود تا خودتان این بخش را پیاده سازی کنید. در ادامه سعی کنید تا بیشترین طول هر کدام از کپشن ها را نیز بدست آوردید. این کار در ادامه برای کار padding مهم میباشد.

چ) در این قسمت سعی داریم تا با استفاده از دیتا جنریتور داده هایمان را آماده سازی کنیم. در نظر داشته باشید که داده ها را به صورت mini batch به شبکه میدهیم بنابراین تمام کپشن ها را با استفاده از توکن <PAD> به بزرگترین طول مدنظر افزایش دهید (در این قسمت بهتر است از متود pad_sequence استفاده کنید). سپس یک دیتاجنریتور در نظر بگیرید که مانند iterator عمل میکند. این دیتا جنریتور در هر مرحله دو ورودی و یک خروجی دارد. خروجی (که همان هدف در lstm است) کلمه بعدی است و ورودی ها تصویر و تمامی لغاتی که قبل از کلمه هدف آمده اند میباشد.

ح) حال میخواهیم تمامی کلمات را به حالت وکتور نمایش دهیم. در اینجا نیز سعی میکنیم تا از یک نمایش برداری آماده و قوی برای بردار ها استفاده کنیم. در اینجا سعی میکنیم تا از نمایش از پیش یادگیری شده GLOVE استفاده کنیم. (میتوان از نمایش های دیگر نیز استفاده کرد) حال برای تمام لغت های یکتا ، embedding_matrix را پیاده سازی میکنیم. در ادامه در نظر داشته باشید که پس از استفاده از نمایش از پیش یادگیری شده، لایه embedding را حتمن freeze کنید. (استفاده از GLOVE صرفا برای دقت کار توصیه میشود و میتوان آن را در فرایند یادگیری نیز فراگرفت.)

خ) در این قسمت میخواهیم توجهی به architecture ای که باید پیاده سازی شود داشته باشیم. در مسئله image captioning دو موضوع مدنظر است. ابتدا encode کردن تصاویر و سپس decode کردن آن با الحاق لغات embed شده در کنار فیچر های به دست آمده از مرحله قبل. حال شما میتوانید این معماری را به هر شکلی که از نظر خودتان مطلوب است پیاده سازی کنید.

د) پس از آموزش مدل، سعی کنید تا مدل را بر روی چند تصویر تست اعمال کنید و کپشن های تولید شده را در گزارش بیاورید. در نظر داشته باشید که پس از به دست آمدن <END> کار تولید کپشن را متوقف کنید.