



یادگیری عمیق

پاییز ۱۴۰۱
استاد: دکتر فاطمی زاده

گردآورندگان:

مهلت ارسال: ۷ بهمن

NLP و RNN

تمرین چهارم

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص شده است.
- در طول ترم امکان ارسال با تاخیر پاسخ همه‌ی تمرین تا سقف ۲۳ روز وجود دارد. پس از گذشت این مدت، پاسخ‌های ارسال‌شده پذیرفته نخواهند بود. همچنین، به ازای هر روز تأخیر غیر مجاز ۱۰ درصد از نمره تمرین به صورت ساعتی کسر خواهد شد.
- همکاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ‌های ارسال‌شده باید توسط خود او نوشته شده باشد. (دقت کنید در صورت تشخیص مشابهت غیرعادی برخورد جدی صورت خواهد گرفت.)
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.
- نتایج و پاسخ‌های خود را در یک فایل با فرمت zip به نام HW۴-Name-StudentNumber در سایت [Quera](#) قرار دهید. برای بخش عملی تمرین نیز لینک گیت‌هاب که تمرین و نتایج را در آن آپلود کرده‌اید قرار بدهید. دقت کنید هر سه فایل نوتبوک تکمیل شده بخش عملی را در گیت‌هاب قرار دهید.
- لطفا تمامی سوالات خود را از طریق کوثرای درس مطرح بکنید (برای اینکه تمامی دانشجویان به پاسخ‌های مطرح شده به سوالات دسترسی داشته باشند و جلوی سوالات تکراری گرفته شود، به سوالات در بسترهای دیگر پاسخ داده نخواهد شد).
- دقت کنید کدهای شما باید قابلیت اجرای دوباره داشته باشند، در صورت دادن خطا هنگام اجرای کدتان، حتی اگر خطا بدلیل اشتباه تایپی باشد، نمره صفر به آن بخش تعلق خواهد گرفت.

سوالات نظری (۲۵ نمره)

۱. (۲۵ نمره)

در زیر روابط سلول پایه RNN عادی و سلول شبکه XNN داده شده است. (نمادگذاری‌ها منطبق بر آنچه در درس گفته شده است.)

:RNN Vanilla

$$h_t = \tanh(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

:XNN

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$$

$$\tilde{h}_t = \tanh(W^{(hx)}x_t + r_t \circ W^{(hh)}h_{t-1})$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

(آ) اثر و عملکرد گیت‌های z_t ، r_t در خروجی سلول را شرح دهید.

(ب) با محاسبه ریاضی بررسی نمایید که چگونه سلول XNN دارای مشکل Vanish Gradient کمتری نسبت به RNN عادی است.

سوالات عملی (۳۰۰ نمره)

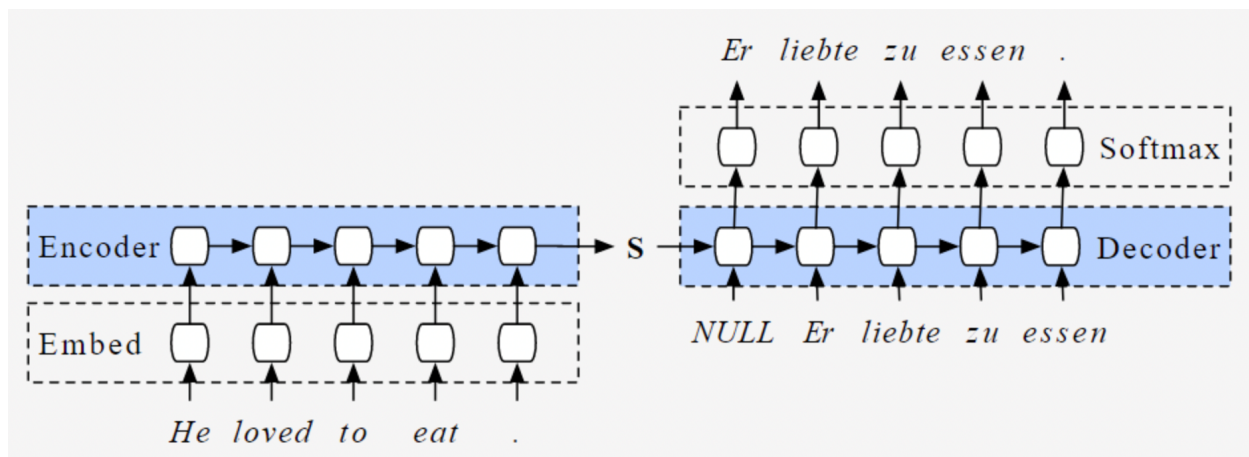
۱. (۱۰۰ نمره) برای این سوال کلیه اشعار ابوالقاسم فردوسی در اختیار شما قرار گرفته است. هدف از این سوال پیاده سازی یک مدل sequence to sequence با استفاده از شبکه های عمیق و روش های NLP می باشد. ابتدا مجموعه ی دادگان را پردازش کنید و دادگان مناسب برای آموزش شبکه مورد نظر را بدست آورید. (برای آموزش بهتر مدل، دادگان داده شده را به بیت های مجزا تقسیم کنید و یک بیت را برای ورودی در نظر بگیرید و بیت پس از آن را خروجی قرار دهید)

(آ) از معماری Encoder - Decoder با واحدهای LSTM استفاده کنید، مدل را آموزش دهید و نمودار دقت و خطا را رسم کنید.

(ب) از معماری Encoder - Decoder با واحدهای GRU دوطرفه استفاده کنید، مدل را آموزش دهید و نمودار دقت و خطا را رسم کنید.

برای طراحی مدل های بالا ابتدا کلمات را به فضای embedding مناسب ببرید (با توجه به اینکه طول ابیات متفاوت است قبل از اعمال embedding بر روی آنها می توانید از padding مناسب استفاده کنید)
برای آموزش مدل از تابع فعالساز Softmax در لایه خروجی decoder استفاده کنید و بهینه ساز آموزش را Adam و تابع هزینه را Cross Entropy در نظر بگیرید.

شکل ۱: شمای مدل sequence to sequence مورد اشاره



به توضیحات زیر توجه نمایید:

- برای آموزش مدل از روش **Teacher Forcing** استفاده کنید.
- دقت کنید مدل باید خاصیت مولد^۱ داشته باشد و پس از اتمام آموزش، مدل باید قابلیت این را داشته باشد که برای هر ورودی خاص، خروجی های متفاوتی تولید نماید.

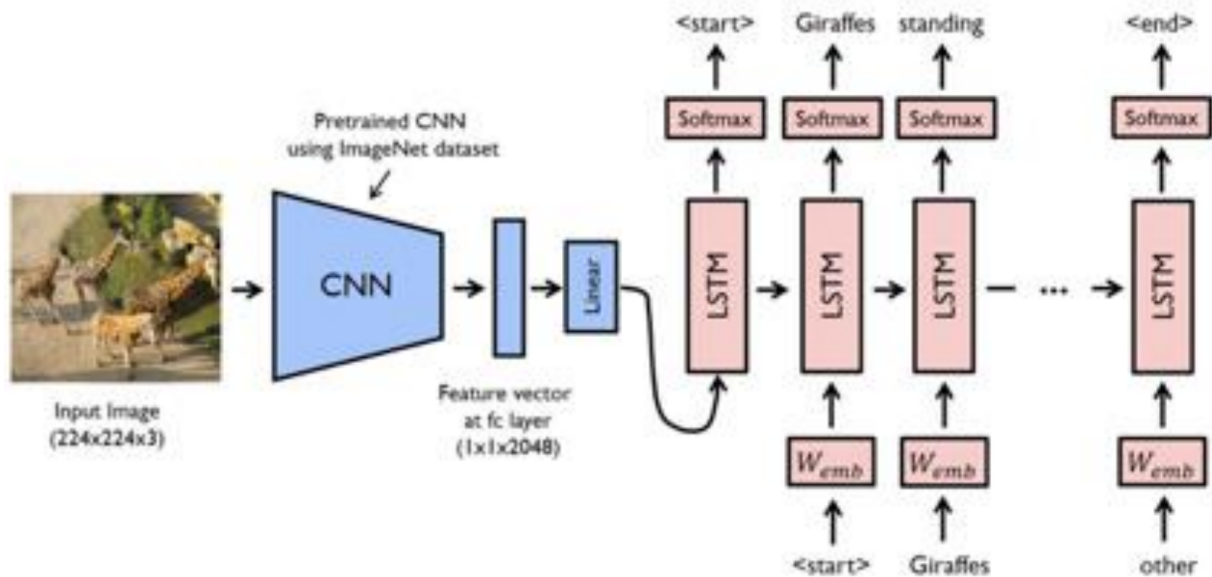
۲. (۱۰۰ نمره) زیرنویسی تصویر^۲ تولید زیرنویس برای تصاویر است. در این تمرین می خواهیم با استفاده از ترکیبی از یک شبکه ی پیچشی^۳ و یک شبکه ی LSTM برای یک تصویر داده شده، زیرنویس انگلیسی تولید کنیم.

^۱Generative

^۲Image Captioning

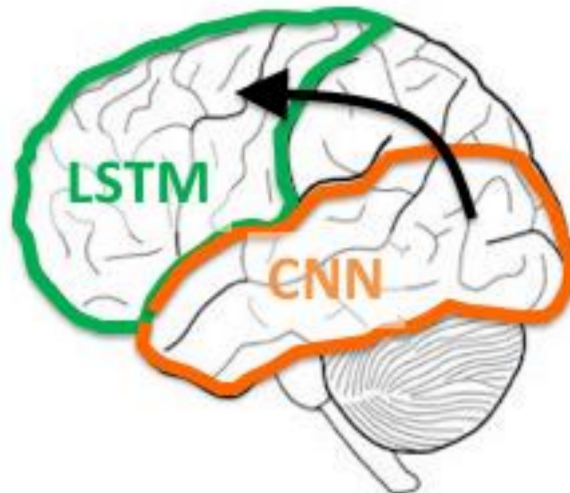
^۳Convolutional

شکل ۲: شمای کلی نمونه مدل مورد نظر



شمایی از شبکه‌ای که بایستی در این سوال پیاده‌سازی کنید در شکل ۲ مشاهده می‌کنید. جالب است بدانید که معماری مذکور شبیه به فهم کلی ما از شیوهی کارکرد مغز در نقش مغز است. برای مثال، بسیاری از مدل‌های شبکه‌های عصبی از یک شبکه‌ی پیچشی در ابتدا (برای تبدیل پیکسل‌های تصویر به بازنمایی مورد نیاز) استفاده می‌کنند و خروجی آن را برای ترکیب با زبان طبیعی به لایه‌های متنوع LSTM می‌دهند؛ شبیه به فهم ما از اینکه مغز چگونه تصویر را در مسیر ventral پردازش کرده و اطلاعات را به قشر پیش‌پیشانی منتقل می‌کند که برای مثال با درک/تولید گفتار سر و کار دارد (شکل ۳).

شکل ۳: نگاه مشابه مدل زیرنویسی تصویر به آنچه در مغز روی می‌دهد



در این تمرین از دادگان COCO استفاده می‌شود. چارچوب‌های مختلف توسعه‌ی هوش مصنوعی مانند TensorFlow و PyTorch ابزارهایی برای کار با این دادگان دارند. احتمالاً پیش از استفاده، نیاز به نصب این API خواهید داشت. تنها بر روی قسمت train دادگان آموزش را انجام دهید.

تصویر ورودی شبکه را به اندازه‌ی 224×224 تبدیل نمایید. برای بخش کدکنده‌ی تصویر، از یک مدل پیچشی آموزش‌دیده بر روی دادگان ImageNet استفاده نمایید. چارچوب‌های مختلف شبکه‌های متنوع

پیش‌آموزش دیده را در بر دارند. توصیه می‌شود از معماری ResNet استفاده نمایید.
 برای بخش تولید کننده‌ی زیرنویس از معماری LSTM یا انواع دیگر RNN ها استفاده نمایید.
 موارد زیر را برای پردازش زیرنویس‌ها در نظر داشته باشید:

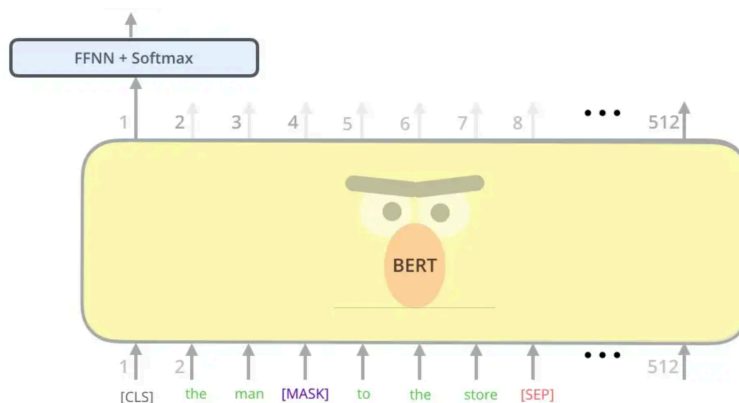
- علائم نگارشی و کلمات شامل عدد را حذف نمایید و تمام حروف را به صورت کوچک در آورید.
- برای پردازش متن می‌توانید از کتاب‌خانه‌ی NLTK استفاده نمایید.
- یک لغت‌نامه از کلمات موجود در زیرنویس‌ها تهیه کنید و هر کلمه را به یک اندیس نگاشت نمایید.
- به ابتدا و انتهای زیرنویس‌ها کلمات منحصر به فردی برای مشخص کردن آغاز و پایان اضافه کنید.
- بیشترین طول زیرنویس‌ها در دادگان بیابید و با اضافه کردن یک کلمه‌ی منحصر به فرد، طول تمام زیرنویس‌ها را یکسان نمایید. این مورد برای آنکه بتوانید زیرنویس‌ها را در یک batch استفاده کنید لازم است.

پیشنهاد می‌شود موارد انجام شده در هر بخش، به ویژه قسمت پردازش متن را ذخیره نمایید تا در زمان اجرای دوباره نیاز به پردازش مجدد نباشد.

در نهایت، بایستی تابعی داشته باشید تا با دریافت یک تصویر ورودی در ابعاد 224×224 زیرنویسی برای آن تولید کند.

۳. (۱۰۰ نمره) در این تمرین قصد داریم با استفاده از مدل‌های دارای مکانیزم توجه، یک طبقه‌بندی‌کننده متن برای تشخیص سبک شاعر استفاده کنیم. دیتاست اشعار فارسی به تفکیک شاعر در این لینک موجود است. ۱۰ شاعر را به دلخواه انتخاب کنید و تمرین را با استفاده از اشعار آن‌ها انجام دهید. توجه کنید که مانند همیشه می‌بایست ابتدا داده موجود را به دو یا سه بخش train، test و در صورت نیاز validation تقسیم کنید.

(آ) یکی از مدل‌های از پیش آموزش دیده BERT بر روی زبان فارسی را انتخاب کنید (برای مثال) و با استفاده از لاجیت توکن طبقه‌بندی^۴ (که در BertPooler ظاهر می‌شود) یک مدل طبقه‌بندی‌کننده تولید کنید. (یک لایه خطی کفایت می‌کند!) سپس تنها مازول طبقه‌بندی را آموزش دهید سپس خسارت، دقت و نمره میانگین micro F1 را گزارش کنید و ماتریس درهم‌ریختگی را رسم کنید.



(ب) این بار تمام مدل را روی داده fine-tune کنید سپس خسارت، دقت و نمره میانگین micro F1 را گزارش کنید و ماتریس درهم‌ریختگی را رسم کنید. این کار را یک بار با استفاده از بهینه‌ساز Adam و یک بار با استفاده از SGD انجام دهید.

(ج) حال می‌خواهیم برای بررسی تغییرات مدل، می‌خواهیم **Perplexity** را برای مدل، قبل و بعد از fine-tuning بررسی کنیم. یک سمپل نارایب شامل ۱۰۰۰ بیت از شعرای مختلفی که انتخاب کرده‌اید را برای مدل قبل و بعد از تیونینگ بررسی کنید و نمره را به دست آورید.

^۴Classification Token