یادگیری ژرف
نیم‌سال اول ۰۲-۰۱
استاد: دکتر حمید بیگی
گردآورندگان: علی سطوتی - محمدعلی صدرایی جواهری - امیرحسین عاملی - محدثه میربیگی
بررسی و بازبینی: مهران حسین‌زاده
دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

تمرین چهارم | یادگیری بازنمایی و مکانیزم توجه | مهلت ارسال: ۹ دی ۱۴۰۱

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص‌شده است.
- در طول ترم امکان ارسال با تاخیر تمرین‌ها بدون کسر نمره تا سقف ۱۰ روز (تا سقف ۳ روز برای هر تمرین) وجود دارد. محل بارگزاری جواب تمرین‌ها بعد از ۵ روز بسته خواهد شد و پس از گذشت این مدت، پاسخ‌های ارسال‌شده پذیرفته نخواهند شد. 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 پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد.
- 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.
- پاسخ تمامی سوالات (تئوری و عملی) را در یک فایل فشرده به صورت $DL\_Hw4\_[firstName]\_[lastName]\_[StudentId]$ نام‌گذاری کرده و ارسال کنید.

سوالات نظری (۶۰ نمره)

۱. Autoencoder - 1 (۱۰ نمره)

فرض کنید مدل های خطی با دو وزن ماتریس داریم: یک رمزگذار $W_1 \in \mathbb{R}^{K\times m}$ و یک رمزگشا $W_2 \in \mathbb{R}^{m\times k}$.
مدل سنتی autoencoder بازنمایی با ابعاد کم n نقطه را با داده‌های آموزشی $X \in \mathbb{R}^{m\times n}$ یاد می گیرد. این یادگیری با کمینه‌سازی تابع زیر صورت می گیرد.

$$L(W_1, W_2; X) = \frac{1}{n}\|X - W_2W_1X\|_F^2$$

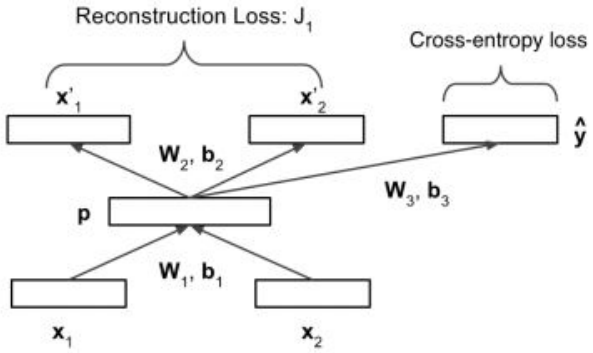
فرض کنید که برای ماتریس $1/nXX^T$ مقادیر ویژه $\sigma_1 > ... > \sigma_k > \sigma_{k+1} >= 0$ وجود دارند.
حال L2-regularized linear autoencoder را با تابع هدف زیر در نظر بگیرید:

$$L(W_1.W_2; X) = \frac{1}{n}\|X - W_2W_1X\|_F^2 + \lambda\|W_1\|_F^2 + \lambda\|W_2\|_F^2$$

برای حل این رابطه گفته شده اگر لاندا مقدار غیر صفر کوچک باشد، آیا ما یک inductive bias برای پیداکردن ماتریس W_2 داریم؟ بحث نمایید.

۲. Autoencoder - 2 (+Word Embedding) (۱۴ نمره)

یکی از راه‌های ترکیب بردارها جمعشان با هم است. در این سوال می خواهیم راه دیگری با استفاده از autoencoder امتحان کنیم. شکل زیر نشان دهنده این راه حل است.



در این autoencoder دو بردار ورودی $x_2 \in \mathbb{R}^{D_x \times 1}$ و x_1 به هم متصل می‌شوند و یک بردار $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ می‌سازند. بردار والد و ماتریس W به صورت زیر محاسبه می‌شوند.

$$p = ReLU(W_1 x + b_1) \in \mathbb{R}^{D_p \times 1}$$

$$W_1 = \begin{bmatrix} W_{11} & W_{12} \end{bmatrix}$$

در حین آموزش از بردار والد برای بازسازی بردار ورودی استفاده می شود.

$$x' = \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = W_2 p + b_2 \in \mathbb{R}^{2D_x \times 1}$$

بردارهای x'_1 ، x'_2 بردار های بازسازی شده هستند. تابع هزینه بازسازی برابر با فاصله اقلیدسی بین ورودی ها و بازسازی ها در حین آموزش است.

$$J_1 = \frac{1}{2}||x' - x||^2 \in \mathbb{R}$$

برای تحلیل احساسات از بردار والد برای پیش بینی استفاده می شود تا کلاس مورد نظر $\hat{y}$ پیش بینی شود.

$$\hat{y} = W_3 p + b_3 \in \mathbb{R}^{D_c \times 1}$$

برای تحلیل احساسات ما سه کلاس داریم. شبکه با تابع هزینه entropy cross آموزش می بیند.

$$J_2 = CE(y, \hat{y} \in \mathbb{R})$$

Y یک بردار برچسب one-hot است. شبکه با جمع دو تابع هزینه گفته شده آموزش می بیند: $J = J_1 + J_2$

الف) چگونه بردارهای بازسازی شده و تابع هزینه بازسازی به یادگیری بردار والد کمک می کنند؟
ب) ابعاد هر یک از وزن ها و بایاس های شبکه را بیابید؟ تعداد کل پارامترهای مدل چند تا است؟
ج) چگونه می توان شبکه‌ای فقط با کپی از این ماژول autoencoder داشت که بتواند برچسب برای تحلیل احساسات برای یک جمله کامل را انجام دهد؟

۳. مکانیزم توجه (۱۸ نمره)

مکانیزم توجه برای از بین بردن گلوگاه اطلاعات بین رمزگذار و رمزگشا معرفی شده است. به این صورت که به جای آخرین بردار نهان رمزگذار، رمزگشا به تمام بردارهای نهان رمزگذار دسترسی دارد. این مکانیزم به صورت زیر فرموله می‌شود و در هر گام شبکه‌ی تکرارشونده‌ی رمزگشا مورد استفاده قرار می‌گیرد:

$$a_t(s) = \frac{\exp score\left(h_d^{(t)}, h_e^{(s)}\right)}{\sum_{s'} \exp score\left(h_d^{(t)}, h_e^{(s')}\right)}$$

$$c_t = \sum_{s'} a_t(s') h_e^{(s')}$$

$$\hat{h} = \tanh W_c \left[c_t; h_d^{(t)}\right]$$

$$y_t = softmax\left(W_s \hat{h}\right)$$

که در آن $h_d^{(i)}$ بردار نهان رمزگشا، $h_e^{(i)}$ بردار نهان رمزگذار و y_t خروجی گام t ام رمزگشا می‌باشد. تابع $score\left(h_d^{(t)}, h_e^{(s)}\right)$ را می‌توان به سه روش زیر تعریف کرد:

$$score\left(h_d^{(t)}, h_e^{(s)}\right) = \begin{cases} h_d^{(t)^T} h_e^{(s)} & dot \\ h_d^{(t)^T} W_a h_e^{(s)} & general \\ v_a^T \tanh W_a \left[h_d^{(t)}; h_e^{(s)}\right] & \tanh layer \end{cases}$$

(آ) این سه تابع را از نظر توان مدل کردن، هزینه‌ی محاسباتی و عبور گرادیان در مرحله بازانتشار خطا مقایسه کنید. شما کدام یک را برای یک شبکه Seq2Seq انتخاب می‌کنید؟

(ب) در ادبیات یادگیری عمیق، دو کار پژوهشی دو مکانیزم توجه ارائه داده‌اند که جز رایج‌ترین کارهای این حوزه می‌باشد: ۱ - مکانیزم۱ ۲ - مکانیزم۲ . این دو ساختار را با هم مقایسه کنید و تفاوت‌های آن را ذکر کنید. کدام یک توانایی مدل کردن بیشتری دارد؟

(ج) یکی از مشکلات رایج مکانیزم توجه، مخصوصا هنگامی که متن ورودی در طرف رمزگذار طولانی باشد، عدم توانایی این مکانیزم در پرداختن به تکه‌های مختلف متن ورودی است. به طور مثال ممکن است در تمامی گام‌های رمزگشا، مکانیزم توجه فقط به یک یا دو کلمه ی خاص امتیاز بسیار بالایی بدهد و فقط آن‌ها را در نظر بگیرد. در این صورت مدل قادر نخواهد بود که از تمامی متن ورودی استفاده کند. برای حل این مشکل چه راهکاری پیشنهاد میدهید؟ توضیح دهید.

۴. Transformer (۱۸ نمره)

یکی از مشکلاتی که transformerها دارند این است که مرتبه هزینه محاسباتی و هزینه ذخیره‌سازی عملیات self-attention دارای عبارت N^2 می‌باشد. این مرتبه باعث می‌شود که آموزش این شبکه روی داده‌های طولانی مانند کتاب مشکل‌زا باشد. دلیل این امر عملگر Softmax می‌باشد که برای محاسبه شباهت دو بردار استفاده می‌شود. در این تمرین قصد داریم به بررسی یک راهکار جایگزین برای این مورد بپردازیم. یکی از این راهکارها استفاده از مکانیزم‌های توجهی کرنلی می‌باشد.

اگر ورودی را $x \in \mathbb{R}^{N\times F}$ و ماتریس‌های مکانیزم توجه را $W_Q \in \mathbb{R}^{F\times D}, W_K \in \mathbb{R}^{F\times D}$ و $W_V \in \mathbb{R}^{F\times M}$ در نظر بگیریم. می‌توان این عملیات را به صورت زیر نوشت:

$$Q = xW_Q, K = xW_K, V = xW_V$$
$$V' = softmax\left(\frac{QK^T}{\sqrt{D}}\right)V$$

حال با تعریف $sim\left(Q_i, K_j\right) = \exp\left(\frac{Q_i^T K_j}{\sqrt{D}}\right)$ می‌توان این عبارت را به فرم زیر بازنویسی کرد:

$$V_i' = \frac{\sum_{j=1}^{N} sim\left(Q_i, K_j\right) V_j}{\sum_{j=1}^{N} sim\left(Q_i, K_j\right)}$$

(آ) مرتبه زمانی و حافظه مورد نیاز برای محاسبه عملگر self-attention بالا را براساس پارامترهای N,D,M محاسبه کنید.

(ب) یکی از توابعی که می‌توان جایگزین $sim\left(Q_i, K_j\right)$ کرد، کرنل توجه چندجمله‌ای می‌باشد. عبارت جایگزین را برای حالت درجه دو (Quadratic) بنویسید.

(ج) برای کرنل مرتبه بخش قبل، بردار ویژگی $\phi\left(\cdot\right)$ را بنویسید

(د) حال باتوجه به رابطه $\mathrm{K}\left(q,k\right) = \phi(q)^T\phi\left(k\right)$ ، رابطه نهایی V_i^t (طبق توضیحات بالا) را بازنویسی کنید و مرتبه زمانی رابطه و مرتبه حافظه مورد نیاز را برای آن محاسبه کنید. با مقایسه این مرتبه‌ها با مرتبه‌های رابطه قبلی، در چه شرایطی استفاده از این رابطه بهتر از رابطه قبلی می‌باشد؟

سوالات عملی (۴۰+۱۰ نمره)

---

۱. بردارهای معنایی (۲۵ نمره)

در این تمرین عملی قرار است شما با استفاده از بردارهای glove چهار کار جالب در زمینه NLP را انجام دهید. در بخش اول و دوم شما با استفاده از روابط ریاضی برداری چند ویژگی جالب در بردارهای کلمات مشاهده

می‌کنید.
در بخش سوم و چهارم قرار است یک طبقه‌بندی را آموزش دهید. داده ورودی شما متون نقد فیلم کاربران است و هر نقد بر اساس محتوایش می‌تواند نقد مثبت یا منفی باشد. در این تمرین شما باید به دو روش مختلف مدلی را برای طبقه‌بندی آموزش دهید و این دو روش را با هم مقایسه کنید.
در بخش سوم مدل شما باید با روش میانگیرگیری بین بردارهای کلمات موجود در نقد، بردار معادل هر نقد را پیدا کند و با استفاده از مدل MLP آن را طبقه‌بندی کند. بخش زیادی از کد بخش سوم زده شده است و فقط یکی از تابع‌های آن نیاز به تکمیل دارد.
در بخش چهارم شما باید یک مدل LSTM برای این موضوع آموزش دهید. در زمینه تعداد لایه‌ها یا اندازه بردار مدل انتخاب دست خودتان است. همچنین می‌توانید با دیدن کدهای روش اول برای زدن کدهای روش دوم ایده بگیرید. همچین شما باید مدل بخش چهارم را با دو تنظیم مختلف آموزش دهید. در تنظیم اول لایه embedding فریز می‌شود و در تنظیم دوم خیر و در نهایت باید عملکرد مدل نهایی را در این دو حالت مقایسه کنید و در نوتبوک خود توضیح دهید.

برای حل این سوال به نوتبوک ضمیمه شده explore_word_embeddings.ipynb مراجعه کنید و همه‌ی پاسخ‌های خواسته شده را داخل همان نوتبوک ارائه دهید.

۲. طبقه‌بندی اسپم (۲۵ نمره)

در این تمرین به پیاده‌سازی یک طبقه‌بند هرزنامه خواهیم پرداخت. دادگان موجود مجموعه دادگان پیامک‌های هرزنامه هستند که دارای برچسب مناسب میباشند. برای پیاده‌سازی طبقه‌بند از مدل‌های زبانی از پیش آموزش دیده و یک نوع خاص از خانواده ترنسفورمر به نام برت (BERT) استفاده خواهیم کرد. برای حل این سوال به نوتبوک ضمیمه شده spam_classification.ipynb مراجعه کنید و همه‌ی پاسخ‌های خواسته شده را داخل همان نوتبوک ارائه دهید.