



نکات زیر را رعایت کنید:  
فایل گزارش را به همراه تمامی کدها در یک فایل فشرده و با عنوان HW2\_STD# در سایت Quera.ir بارگذاری نمایید.  
سوالات خود را از طریق Quera مطرح کنید.

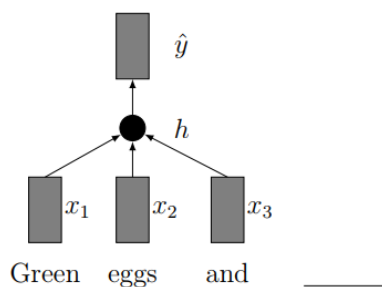
## مسئله ۱. Neural network Language Models (NNLM)

برای آموزش بردارهای کلمات<sup>۱</sup> می توانیم از معماری مدل های زبانی شبکه عصبی<sup>۲</sup> استفاده کنیم. این مدل سعی می کند با توجه به N کلمه قبلی (کلمات زمینه<sup>۳</sup>)، کلمه بعد از آنها را (کلمه هدف<sup>۴</sup>) پیش بینی کند. برای این کار بردارهای کلمات قبلی را با هم متصل<sup>۵</sup> می کنیم و حاصل را از یک لایه مخفی با تابع فعال ساز Tanh رد می کنیم و در آخر خروجی را به یک لایه Softmax می دهیم تا کلمه هدف را پیش بینی کند. با در نظر گرفتن فرض های زیر، به سوالات پاسخ دهید.

- سائز لایه مخفی برابر با H و تعداد لغات منحصر به فرد<sup>۶</sup> داخل پایگاه داده V است.
- تابع هزینه مورد استفاده برای آموزش مدل، Cross Entropy است.
- بردار کلمه ی N لغات زمینه برابر با ماتریس x است که هر کدام از  $x_i$  ها یک بردار D بعدی هستند.
- بردار y بردار Onehot کلمه هدف است.
- ابعاد پارامترها و متغیرها برابر است با:

$$x \in \mathbb{R}^{(N \cdot D)}, W \in \mathbb{R}^{H \times (N \cdot D)}, b \in \mathbb{R}^H, h \in \mathbb{R}^{V \times H}, d \in \mathbb{R}^V, \hat{y} \in \mathbb{R}^V$$

- روابط ریاضی در شبکه مورد نظر:



$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

$$\mathbf{h} = \tanh(W\mathbf{x} + \mathbf{b})$$

$$\hat{\mathbf{y}} = \text{softmax}(U\mathbf{h} + \mathbf{d})$$

$$J = \text{CE}(\mathbf{y}, \hat{\mathbf{y}})$$

$$CE = - \sum_i y_i \log(\hat{y}_i).$$

<sup>1</sup>Word Vectors

<sup>2</sup>neural network language models

<sup>3</sup>Context

<sup>4</sup>Target

<sup>5</sup>Concatenate

<sup>6</sup>Vocabulary Size

(آ) دو تفاوت اصلی و عمده بین NNLM و CBOW که در درس آموخته‌اید را بیان کنید و برای هرکدام یک مثال (به زبان فارسی یا انگلیسی) بزنید. (۹ نمره)

(ب) پیچیدگی محاسباتی را در بدست آوردن خروجی شبکه NNLM با روش انتشار رو به جلو<sup>۷</sup> برای یک عدد داده آموزشی محاسبه کنید. توجه کنید که برای گرفتن نمره کامل این بخش باید مرحله به مرحله پیچیدگی محاسباتی را حساب کنید و سپس تجمیع آن‌ها را بدست آورید. (۷ نمره)

(ج) اگر بتوانیم تابع هزینه را تغییر دهیم، یک روش ارائه کنید تا بتوانیم پیچیدگی محاسباتی بخش قبل را کاهش دهیم. نام بردن روش به صورت مختصر کافی است و نیازی به توضیح نیست. (۲ نمره)

(د) گرادینان  $J$  نسبت به  $x$  را بدست آورید. (۵ نمره)

## مسئله ۲. Word2Vec and Glove

به سوالات زیر پاسخ دهید.

(آ) یکی دیگر از روش‌های یادگیری بردار کلمات، روش‌های وقوع همزمان مبتنی بر شمارش<sup>۸</sup> است. دو مزیت و دو عیب در استفاده از روش Word2Vec نسبت به این روش را بیان کنید. (۴ نمره)

(ب) دو نفر می‌خواهند از روش Word2Vec برای بدست آوردن تعبیه کلمات در یک Vocabulary یکسان با سایز  $V$  استفاده کنند. به طور دقیق‌تر نفر اول بردار کلمات زمینه  $u_w^A$  و بردار کلمه هدف  $v_w^A$  را برای هر  $w \in V$  و نفر دوم بردار کلمات زمینه  $u_w^B$  و بردار کلمه هدف  $v_w^B$  را برای هر  $w \in V$  می‌خواهند به وسیله این روش، بدست آورند. فرض کنید برای هر جفت کلمه  $w, w' \in V$  ضرب داخلی بردار کلمات در مدل هر دو نفر یکسان باشد یعنی

$$(u_w^A)^T v_{w'}^A = (u_w^B)^T v_{w'}^B$$

آیا می‌توان نتیجه گرفت که برای هر کلمه  $w \in V$  داریم  $v_w^A = v_w^B$ ؟ چرا؟ (۴ نمره)

(ج) چرا استفاده از تابع Softmax برای Word2Vec کار درستی نیست؟ راهکار پیشنهادی شما برای حل این مشکل چیست؟ (۳ نمره)

(د) مقدار حافظه مصرفی الگوریتم‌های Word2Vec و Glove را با ذکر دلیل مقایسه کنید. (۲ نمره)

(ه) دو مورد از نواقص مشترک Word2Vec و Glove را نام ببرید. (۴ نمره)

(و) در تابع هزینه زیر که متعلق به الگوریتم Glove است، اگر ماتریس‌های  $R$  و  $\tilde{R}$  تعبیه‌های کلمات  $\{r_i\}, \{\tilde{r}_j\}$  را در خود داشته باشند، نشان دهید که این تابع هزینه محدب<sup>۹</sup> نیست. برای حل این سوال، نیازی به اثبات ریاضی نیست و فقط کافی است توضیح منطقی‌ای ارائه کنید که محدب نبودن را تصدیق کند.

$$J(R, \tilde{R}) = \sum_{i,j} f(x_{ij})(r_i^T \tilde{r}_j - \log x_{ij})^2$$

راهنمایی: از مفاهیم Swap-invariance و Permutation-invariance استفاده کنید. (۷ نمره)

<sup>7</sup>Forward Propagation

<sup>8</sup>Cooccurrence Count-based Methods

<sup>9</sup>Convex

## مسئله‌ی ۳. Transformers and Attention models

به سوالات زیر پاسخ دهید.

۱. یکی از مراحل مقدماتی در پیش پردازش داده‌ها برای حوزه پردازش متن، توکن‌سازی<sup>۱۰</sup> از جمله‌های پایگاه داده است. یکی از چالش‌های این مرحله، این است که ترتیب کلمات در جمله با این کار از بین می‌رود. برای حل این مشکل رویکرد Positional Embedding مطرح می‌شود. به صورت کلی و کوتاه توضیح دهید که چگونه این رویکرد می‌تواند مشکل را برطرف کند. (۳ نمره)

۲. یکی از کاربردهای رویکرد گفته شده در قسمت قبل، استفاده از Transformer ها است. این معماری چگونه مشکل گفته شده (بهم خوردن ترتیب کلمات) را برطرف می‌کند؟ (۴ نمره)

۳. محدودیت معماری Encoder-Decoder ای که از مکانیزم توجه استفاده نمی‌کند را در حوزه ترجمه ماشینی بیان کنید و سپس به صورت خلاصه توضیح دهید که چگونه مکانیزم توجه می‌تواند این مشکل را برطرف کند. (۶ نمره)

## مسئله‌ی ۴. Autoencoders (عملی)

خودکدگذارها<sup>۱۱</sup> یک خانواده از شبکه‌های عصبی هستند که برای یادگیری بدون نظارت مناسب‌اند. این معماری روشی مناسب برای تشخیص الگوهای ذاتی در یک مجموعه داده است، همچنین می‌تواند برای برچسب زدن به الگوهای بدست آمده استفاده شود. در این مسئله می‌خواهیم یک خودکدگذار را با استفاده از Pytorch پیاده‌سازی کنیم و برای این منظور از مجموعه داده معروف Mnist استفاده خواهیم کرد. در فایل نوتبوک<sup>۱۲</sup> که در اختیار شما قرار گرفته است، تمامی مراحل پیاده‌سازی مشخص شده است. قسمت هایی که باید توسط شما تکمیل شوند در فایل نوتبوک به صورت ToDo مشخص شده‌اند. وظیفه شما تکمیل این قسمت‌ها می‌باشد و سایر قسمت‌های کد نباید تغییری کند. (۲۳ نمره)

## مسئله‌ی ۵. Attention Models (عملی)

شبکه‌ی RNN نوعی شبکه‌ی عصبی است که ساختار آن به شکلی طراحی شده است که امکان مدل‌کردن داده‌های ترتیبی را بدهد. با اضافه کردن attention به این شبکه‌ها، مدل می‌تواند علاوه بر حالت مخفی کنونی، بر روی حالت مخفی قبلی نیز براساس خروجی رمزگشای<sup>۱۳</sup> قبلی تمرکز کند. روش های مختلفی برای پیاده‌سازی Attention Model ها وجود دارد. یکی از این راه‌ها PyTorch است که توجه به هر ورودی را بر اساس وضعیت پنهان رمزگشا و جاسازی<sup>۱۳</sup> کلمه قبلی خروجی محاسبه می‌کند. در این مسئله می‌خواهیم این مکانیزم را پیاده‌سازی کنیم. توضیحات مورد نیاز در فایل نوتبوک آمده است. وظیفه شما کامل کردن قسمت های مشخص شده در کد می‌باشد. (۱۷ نمره)

<sup>10</sup>Tokenization

<sup>11</sup>Autoencoders

<sup>12</sup>Decoder

<sup>13</sup>embedding