

بسم الله الرحمن الرحيم

دانشگاه علم و صنعت ایران

بهار ۱۴۰۰

تحويل: شنبه ۱۱ اردیبهشت

تمرین سری هفتم

مبانی یادگیری عمیق

۱. معماری ارائه شده در مقاله زیر را توضیح دهید. این معماری چگونه منجر به بهبود عملکرد مدل در مسئله ترجمه ماشینی می‌شود. همچنین نحوه بدست آوردن بردار زمینه (context vector) برای هر کلمه را توضیح دهید. برای پاسخ به این سوال مطالعه بخش سوم مقاله کافیست اما توصیه می‌شود برای درک بهتر مسئله مقاله را به طور کامل مطالعه نمایید.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "[Neural machine translation by jointly learning to align and translate](#)." *arXiv preprint arXiv:1409.0473* (2014).

۲. در این تمرین شما با معماری seq2seq و نحوه پیاده‌سازی اجزای مختلف آن برای مسئله ترجمه ماشینی آشنا می‌شوید. درک مفهوم مقاله‌ی قسمت قبل به شما برای حل این مسئله کمک می‌کند. همچنین سوالاتی در رابطه با قسمت‌های مختلف پیاده‌سازی در **نوت‌بوک** تمرین مطرح شده است که پاسخ به آن‌ها را در همان قسمت ارائه کنید. دقت کنید که برای شروع لازم است ابتدا نوت‌بوک را در درایو خود ذخیره کنید.

* لایه‌ی تعبیه یا Embedding layer:

در پردازش متن لازم است کلمات به عددی تبدیل شوند تا برای مدل قابل پردازش باشند. برای این کار می‌توان به هر کلمه در دایره لغات (دایره لغات مجموعه‌ای از تمام کلمات منحصر به فرد تمام متن‌های ورودی مدل است) به طور دستی برداری از اعداد ثابت را اختصاص داد (مانند فرمت one-hot). در اینجا درمورد جزئیات این روش صحبتی نمی‌کنیم اما خودتان می‌توانید به معایب این روش فکر کنید. در روش دوم، اختصاص بردار اعداد کلمات را به عهده مدل می‌گذاریم تا در زمان آموزش مدل، بردار اعداد کلمات را یاد بگیرد. وظیفه این کار به عهده‌ی لایه Embedding است که در ابتدای مدل‌های متنی به عنوان لایه‌ی اول اضافه می‌شود. یادگیری بردار اعداد کلمات مشابه یادگیری وزن‌های یک لایه‌ی Dense ساده است. لایه Embedding در کتابخانه تانسورفلو به این شکل است:

```
tf.keras.layers.Embedding(  
    input_dim, output_dim, embeddings_initializer='uniform',  
    embeddings_regularizer=None, activity_regularizer=None,  
    embeddings_constraint=None, mask_zero=False, input_length=None, **kwargs  
)
```

برخی از پارامترهای مهم این لایه را به اختصار توضیح می‌دهیم:

Input_dim : اندازه دایره لغات یا همان کلمات منحصر به فرد در متن را مشخص می‌کند.

Output_dim : طول برداری که برای هر کلمه تولید می‌شود را مشخص می‌کند.

Input_length : طول هر جمله (هر توالی) در ورودی را مشخص می‌کند.

```
model = tf.keras.Sequential()  
model.add(tf.keras.layers.Embedding(1000, 64, input_length=10))
```

در اینجا اندازه دایره لغات (کلمات منحصر به فرد) ۱۰۰۰ است.
طول بردار هر کلمه 64 است و طول هر توالی ۱۰ است.
بنابراین خروجی این مدل به شکل (None, 10, 64) است که همان Batch_size است.

نکته دیگری که به آن توجه باید داشته باشید این است که ورودی این لایه به شکل (None, 10) است. که به هر کلمه منحصر به فرد یک index از 0 تا 999 داده شده است.

[این](#) و [این](#) لینک می‌تواند به شما در فهم بهتر این لایه، وظیفه و ضرورت استفاده از آن در مدل‌های پردازش متن کمک کند.

نکات تکمیلی

- (۱) لطفاً پاسخ سوالات (تئوری و توضیحات پیاده‌سازی) را به طور گویا و به زبان فارسی و در صورت امکان تایپ همراه با سورس کدهای نوشته شده، در یک فایل فشرده شده به شکل HW7_YourStudentID.zip قرار داده و بارگذاری نمایید.
- (۲) منابع استفاده شده را به طور دقیق ذکر کنید.
- (۳) برای سهولت در پیاده‌سازی‌ها و منابع بیشتر، زبان پایتون پیشنهاد می‌شود. لطفاً کدهای مربوطه را به طور جداگانه در فرمت .py یا .ipynb ارسال نمایید.
- (۴) ارزیابی تمرین‌ها براساس صحیح بودن راه حل‌ها، گزارش مناسب، بهینه بودن کدها و کپی نبودن می‌باشد.
- (۵) در مجموع تمام تمرین‌ها، تنها ۷۲ ساعت تاخیر در ارسال پاسخ‌ها مجاز است اما پس از آن به صورت خطی از نمره کسر خواهد شد (معادل با روزی ۵۰ درصد).
- (۶) در رابطه با پرسش و پاسخ در رابطه با تمرین‌ها می‌توانید در گروه مربوطه مطرح کنید.

موفق باشید.