بسمه تعالی

دانشکده مهندسی کامپیوتر

یادگیری عمیق

نام استاد: دکتر محمدی

تمرین یازدهم

آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

آذر ۱۴۰۰

پاسخ سوال اول

با استفاده از این روابط میتوانیم به راحتی برای هر نوع از لایه، تعداد پارامترها را بشماریم:

$$Dense: units * (input_{size}+1)$$

$$RNN : (input_{size}+1) * units + units_2$$

$$Conv2D: filters_{num}*(height_{kernel} * width_{kernel} * channels_{num} + 1)$$

که +۱ ها به علت bias است. اگر width و height ورودی به لایه کانوولوشن برابر باشد که در این سوال هست، رابطه تعداد خروجی به این صورت خواهد بود:

$$Out\ put\ of\ Conv2D : \frac{(X - kernel_{size} + 2 * Padding_{size})}{Stride_{size}} + 1$$

$$Out\ put\ of\ MaxPooling2D: \frac{X - Pooling_{size}}{Stride_{size}} + 1$$

پس طبق این روابط میتوانیم لایه به لایه جلو برویم و تعداد پارامتر و ابعاد خروجی را حساب کنیم. به ترتیب از چپ شروع میکنیم:

layer	input	output	Parameters
Conv2D	30*30*3	24*24*64	64*(7*7*3+1)
MaxPooling2D	24*24*64	12*12*64	0
Conv2D	12*12*64	8*8*128	128*(5*5*64+1)
MaxPooling2D	8*8*128	4*4*128	0
Conv2D	4*4*128	2*2*256	256*(3*3*128+1)
MaxPooling2D	2*2*256	1*1*256	0
Flatten	1*1*256	256	0
Dense	256	128	128*(256+1)
RNN	256	128	128*(256+1) + 128*128
Dense	128	128	128*(128+1)

Concatenate	256	256	0
Dense	256	128	128*(256+1)
Dense	128	1000	1000*(128+1)

منابع:

https://kegui.medium.com/how-to-calculate-the-output-size-when-using-conv2dtranspose-layer-19124c79aa15

https://medium.com/deep-learning-with-keras/lstm-understanding-the-number-of-parameters-c4e087575756

پاسخ سوال دوم

بخش تئوری:

- Stop words: کلمات توقف مجموعه ای از کلمات پرکاربرد در یک زبان هستند. نمونه هایی از کلمات توقف در زبان انگلیسی که در این سوال هم حذف شده اند عبارتند از and، a، is، the،in ، will، be.
کلمات توقف معمولا در متن کاوی و پردازش زبان طبیعی (NLP) برای حذف کلماتی که رایج هستند، استفاده می شود. زیرا اطلاعات مفید بسیار کمی را حمل میکنند.
پیش پردازش هایی که روی داده اعمال شد شامل: حذف کلمه هایی که در آنها عدد وجود داشت، حذف اعداد و رقم ها، تبدیل whitespace به فاصله عادی، تبدیل همه کاراکتر ها به lowercase ، تبدیل جمله به لسیتی از کلمات، split بر اساس فاصله و حذف رشته های خالی و جداسازی کلمه های stop word از لیست کلمات به دست آمده بود.
- پس از پاک کردن متن از مواردی که در قسکت قبل توضیح دادیم، بعد لیستی از همه کلمات غیر تکراری در همه ی خبر ها درست میکنیم. بعد یک دیکشنری از کلمات غیر تکراری به کار رفته در خبر ها درست کرده به این صورت که کلید آن مقدار خود کلمه است و مقدار value ان ایندکس آن کلمه در لییست میباشد. این دیکشنری را بعضا inverted index مینامند.
سپس لیستی از tuple هایی به صورت (focus word, context word) میسازیم. و باید آن را به صورت one hot coding بکنیم (مشابه تابع to categorical در برپسب های داده ها). پس در نهایت تعداد فیچر های ما برابر تعداد کلمات غیر یکسان موجود در کل داده های آموزشی (۱۰۰ داده ای که در ابتدا جدا کردیم و از کتگوری crude بودند) است.
در نهایت میتوانیم مدلی را با دو نورون (چون صورت سوال اندازه word embedding را ۲ تعیین کرده است) بسازیم.
- مزایای افزایش سایز window: بهبود بخشیدن رابطه بین کلمات
معایب افزایش سایز window: در نظر گرفتن context word بی ربط به focus word – افزایش زمان آموزش به دلیل افزایش tuple ها
پنجره های بزرگتر تمایل دارند اطلاعات موضوع/دامنه بیشتری را جذب کنند: چه کلمات دیگری (از هر نوع) در بحث های مرتبط استفاده می شود؟ پنجره‌های کوچک‌تر تمایل دارند اطلاعات بیشتری درباره خود کلمه داشته باشند: چه کلمات دیگری از نظر عملکردی مشابه هستند؟ (بسط خود آنها، تعبیه‌های مبتنی بر

وابستگی، برای یافتن بیشتر کلمات مشابه، مترادف‌ها یا بدیهی‌های بدیهی که می‌توانند جایگزین کلمه اصلی شوند، بهترین به نظر می‌رسد.)

به عنوان مثال "stackoverflow great website for programmars" با ۵ کلمه، اگر اندازه پنجره ۲ باشد، بردار کلمه "stackoverflow" مستقیماً تحت تأثیر کلمه "great" و "website" قرار می گیرد، اگر اندازه پنجره ۵ باشد، "stackoverflow" می تواند مستقیماً تحت تأثیر دو کلمه دیگر "for" و "programmers" قرار گیرد. تاثیر در اینجا به این معنی است که بردار دو کلمه را نزدیکتر می کند.

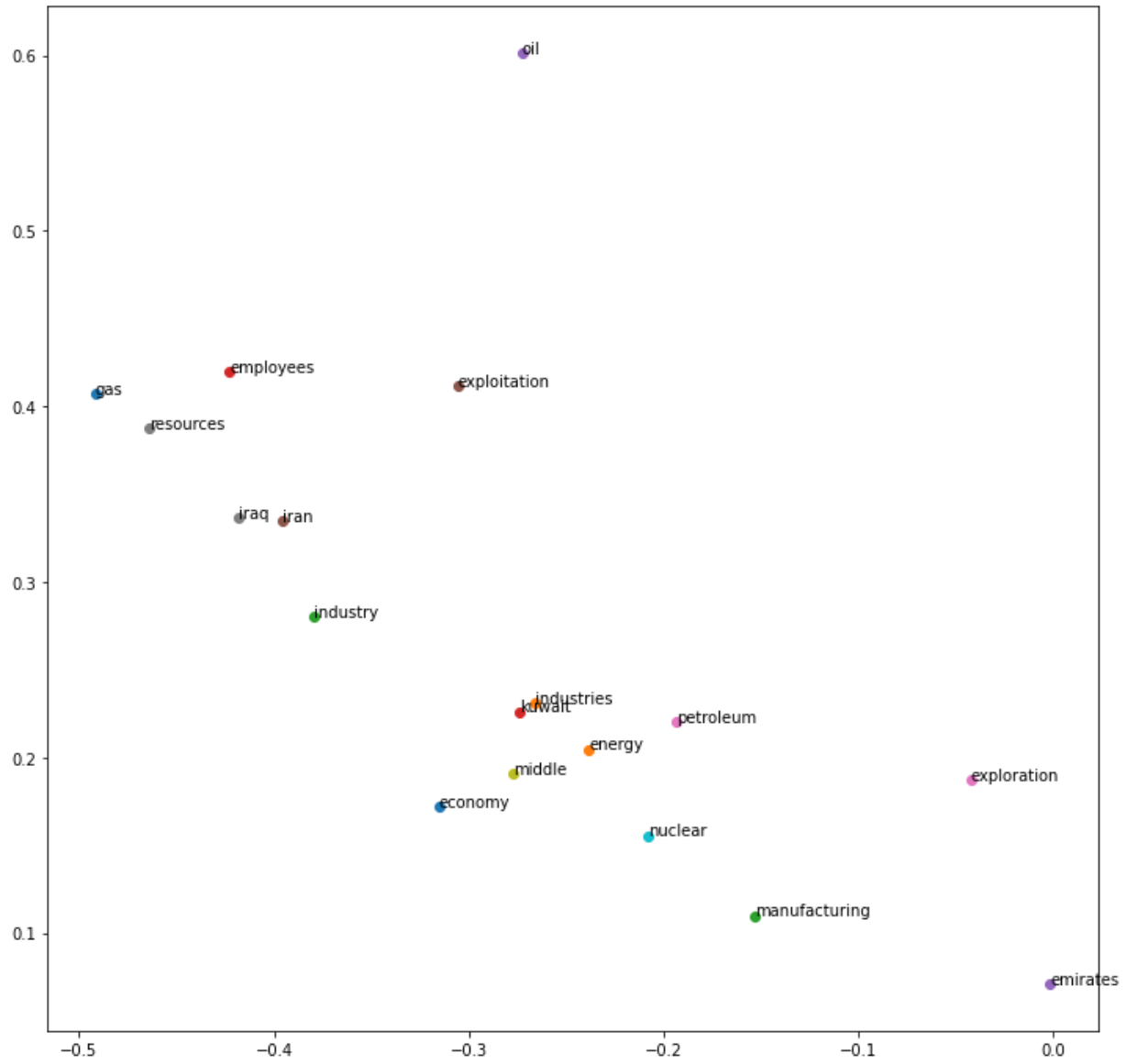
پس انتخاب ابعاد پنجره به نوع متن و استفاده ربط دارد، نه خیلی زیاد مناسب است و نه خیلی کم.

- همانطور که گفتیم ورودی آرایه ای است به طول تعداد کلمات غیر تکراری. که از روش one hot coding استفاده کرده ایم و به همین دلیل یکی از خونه ها ۱ و بقیه صفر هستند. پس از این که مدل آموزش داده شد وزن های لایه اول را درنظر میگیریم. که بصورت یک ماتریس است که هر ردیف آن یک tuple از دو وزن است. که وزن های حاصل از اتصال نورون به دو نورون لایه مخفی میباشد. میدانیم هر نورن نماینده یک کلمه از کلمات موجود در داده های آموزشی میباشد. بنابراین این وزن های دوگانه یادشده همان word embedding برای کلمه هستند.

سوالات تحلیلی:

پس از قسمت های پیاده سازی که در نوتبوک HW11.ipynb قسمت Question2 پیوست شده است، و به کمک ریپازیتوری داده شده انجام شد، دو قسمت باید به کد اضافه میکردیم که به این صورت است:

- کلمات داده شده در صورت سوال را به عنوان ورودی میدهیم و به چنین شکلی میرسیم:



که مشخص است روابطی را بین کلمات تشخیص داده است. مثلا کلمات "انرژی" و "هسته ای" و "نفت" که از نظر معنی نزدیک هستند را به درستی نزدیک هم تشخیص داده است. همچنین "عراق" و "ایران" را. البته نکته نامطلوبی مانند دور بودن زیاد "گاز" از "نفت" و "روغن" هم نشاندهنده کامل نبودن مدل است. چون به هرحال آموزش را زیاد ادامه نداده ام و فقط 20 epoch انجام شد. و دیتا هم آنچنان بزرگ نیست. بردار بین دو کلمه "نفت" و "بهره برداری" تقریبا متناظر با بردار بین "اقتصاد" و "صنعت" به دست می آید که از نکات جالب این تحلیل است.

- کلمات دلخواه زیر هم به عنوان ورودی دادم:

```
['japan', 'decrease', 'lake', 'shell', 'kilolitres', 'iranians', 'year']
```

و ۵ کلمه مشابه با هر کدام را به ترتیب به دست آوردم که به این صورت بودند:

```
similar with word 'japan' :
['search', 'reflect', 'oreffice', 'cancelled', 'monday']
**********************************************************************

similar with word 'decrease' :
['needs', 'megalitres', 'predicted', 'cgp', 'costing']
**********************************************************************

similar with word 'lake' :
['produced', 'courts', 'their', 'slight', 'addition']
**********************************************************************

similar with word 'shell' :
['gas', 'adherence', 'espinosa', 'nine', 'ago']
**********************************************************************

similar with word 'kilolitres' :
['properties', 'supplied', 'pzl', 'saturday', 'enhance']
**********************************************************************

similar with word 'iranians' :
['far', 'colon', 'puerto', 'exported', 'demonstrated']
**********************************************************************

similar with word 'year' :
['noting', 'employees', 'modified', 'comment', 'stabilization']
**********************************************************************
```

که میتوان حس کرد هر کدام از این کلمات به نوعی در خانواده کلمات حدس زده شده هستند. هرچند در صورت آموزش بیشتر مدل و دیتاست بزرگتر و متنوع تر، قطعا به نتایج بهتری دست میافتیم.

منابع:

https://www.opinosis-analytics.com/knowledge-base/stop-words-explained/#:~:text=Stop%20words%20are%20a%20set,carry%20very%20little%20useful%20information

https://towardsdatascience.com/creating-word-embeddings-coding-the-word2vec-algorithm-in-python-using-deep-learning-b337d0ba17a8

https://newbedev.com/word2vec-effect-of-window-size-used

پاسخ سوال سوم

کدهای مربوط به این سوال در نوتبوک HW11.ipynb پیوست شده است. (لینک گوگل کولب)

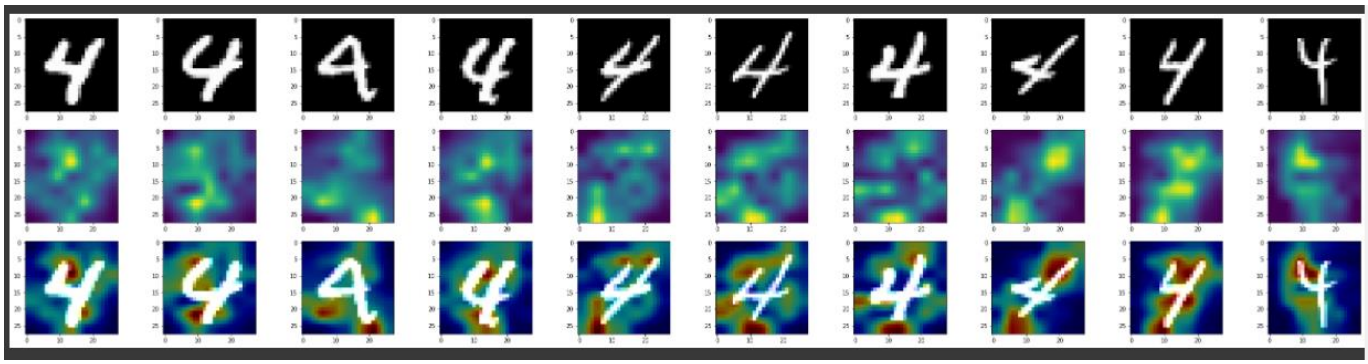
مراحل مختلف گفته شده شامل دریافت دیتاست، shuffle کردن، پیش پردازش داده ها، نرمال کردن (تقسیم بر ۲۵۵) و تبدیل کردن برچسب ها به فرمت one-hot coding یا درواقع to categorical کردن آن ها در سلول های مختلف پیاده سازی شده اند. سپس مدل را دقیقا با شرایط گفته شده پیاده کرده و به دقت های عالی روی داده آموزشی و تست میرسد:

```
1875/1875 [==============================] - 7s 4ms/step - loss: 0.0023 - accuracy: 0.9993
accuracy on train: 99.93 %
313/313 [==============================] - 2s 5ms/step - loss: 0.0289 - accuracy: 0.9925
accuracy on test: 99.25 %
```

تابعی برای خروجی heatmap گرفتن از عکس میسازیم که این این الگوریتم گرادیان خروجی برای داده ی کلاس مشخص را نسبت به اخرین لایه کانولوشنی محسابه میکند تا ناحیه های مهم در تصویر را برای پیشبینی برجسته کند.

تابعی هم برای superimpose تعریف کرده ایم که به منظور نمایش هیت مپ روی عکس اصلی است.

خروجی آخرین سلول جواب مسئله است که برای هرکدام از کلاس های صفر تا ۹، ۱۰ تا از داده ها را به همراه heatmap مربوط به آن و همچنین عکس superimpose به صورت زیرهم نمایش داده شده است. برای نمونه خروجی های کلاس عدد ۴ به این صورت بوده است (برای ۱۰ نمونه اول داده آموزشی نمایش داده ام):



منابع:

https://towardsdatascience.com/understand-your-algorithm-with-grad-cam-d3b62fce353