مجید ادیبیان 4۰۰۱۳۱۰۷۸ تمرین اول

بخش اول)

در این قسمت همانطور که خواسته شده است فایل های گفته شده دانلود شده است.

بخش دوم)

در ابتدا متن موجود در فایل train.txt را می خوانیم و روی کلمات آن پیمایش میکنیم و هر دو کلمه و سه کلمه متوالی را جدا میکنیم و در یک دیکشنری قرار می دهیم و تعداد تکرار آن دو کلمه یا سه کلمه را نیز برابر مقدار آن در دیکشنری قرار می دهیم. در این حالت تعداد تکرار unigram ها و bigram ها را داریم.

سپس باید smoothing را انجام دهیم. توابعی پیاده سازی شده است که smoothing را برای unigram و bigram انجام میدهد. نحوه هموار سازی نیز با استفاده از روش absolute discounting و با استفاده از فرمول های زیر است:

$$unigram: P(w_i) = \frac{\max(\#(w_i) - \delta, 0)}{N} + \frac{\delta}{N}$$

$$bigram: P(w_i|w_{i-1}) = \frac{\max(\#(w_{i-1}, w_i) - \delta, 0)}{\#(w_{i-1})} + \frac{\delta}{\#(w_{i-1})} B * P_{unigram}$$

که در این فرمول ها N برابر تعداد کل کلمات داده آموزشی و B برابر تعداد دوتایی هایی است که $\#(w_{i-1}, w_i)$ بزرگتر از صفر است.

سپس هر بار دلتا را تغییر می دهیم و مقدار perplexity را بر روی داده های ارزیابی به دست می آوریم تا بهترین مقدار دلتا یافته شود. مقدار perplexity را از روی مقدار احتمال هر کلمه از یک سطر از داده ارزیابی به دست می آوریم و سپس از فرمول زیر استفاده می کنیم:

$$perplexity = P(w_1 w_2 \dots w_n)^{-\frac{1}{N}}$$

البته در این پیاده سازی برای جلوگیری ضرب احتمال ها که به صفر میل میکند از لگاریتم آن ها استفاده شده که در نتیجه ضرب به جمع تبدیل می شود و توان به ضریب تبدیل می شود و در نهایت e به توان مقدار به دست آمده را بر می گردانیم. حال مقدار perplexity را برای هر مصرع محاسبه می کنیم و سپس میانگین میگیریم.

همچنین دلتا را از صفر تا یک با گام های ۰.۰۵ تغییر می دهیم که مقدار perplexity بر روی داده های validation در هر بار به صورت زیر است:

```
delta: 0.05
       unigram perplexity: 1360.7482165830145
       bigram perplexity: 896.5963299880291
delta: 0.1
       unigram perplexity: 1255.3646534193872
       bigram perplexity: 812.9647001352185
delta: 0.15
       unigram perplexity: 1204.0645156351343
       bigram perplexity: 773.4331753816832
delta: 0.2
       unigram perplexity: 1171.4896572487612
       bigram perplexity: 749.1201087894766
delta: 0.25
       unigram perplexity: 1148.1562951505282
       bigram perplexity: 732.3567453005434
delta: 0.3
       unigram perplexity: 1130.2369272973428
       bigram perplexity: 720.0751862088389
delta: 0.35
       unigram perplexity: 1115.8340926881492
       bigram perplexity: 710.7718366609752
delta: 0.4
       unigram perplexity: 1103.8795086957416
       bigram perplexity: 703.6156941274498
delta: 0.45
       unigram perplexity: 1093.7168573124022
       bigram perplexity: 698.1127853932402
delta: 0.5
       unigram perplexity: 1084.9163131460652
       bigram perplexity: 693.9589784069473
delta: 0.55
       unigram perplexity: 1077.182274391375
       bigram perplexity: 690.969174997163
delta: 0.6
       unigram perplexity: 1070.3034307584817
       bigram perplexity: 689.0424890280086
delta: 0.65
       unigram perplexity: 1064.1239114179427
       bigram perplexity: 688.1476969407297
delta: 0.7
       unigram perplexity: 1058.525712435501
       bigram perplexity: 688.3240066509783
delta: 0.75
       unigram perplexity: 1053.417520931823
       bigram perplexity: 689.6997506383678
delta: 0.8
       unigram perplexity: 1048.727345333057
       bigram perplexity: 692.5431973426023
delta: 0.85
       unigram perplexity: 1044.397503571306
```

```
	bigram perplexity: 697.3918206541689
delta: 0.9
	unigram perplexity: 1040.3811232834391
	bigram perplexity: 705.4352635062517
delta: 0.95
	unigram perplexity: 1036.6396408409998
	bigram perplexity: 720.1470084676779
```

که بهترین مقدار دلتا مقداری است که perplexity را کمتر کند که در نتیجه بهترین مقدار دلتا با بررسی نتایج بر روی داده های validation به صورت زیر است:

```
best results:
unigram:
	best delta: 0.95
	perplexity: 1036.6396408409998
bigram:
	best delta: 0.65
	perplexity: 688.1476969407297
```

حال مقدار perplexity بر روی داده های فایل test.txt را به دست می آوریم که نتایج به صورت زیر است. همان طور که دیده می شود نتایج مشابه بهترین حالت در قسمت validation است و در bigram نسبت به unigram نتایج مناسب تر بوده است:

```
best results in test:
unigram:
	best delta: 0.95
	perplexity: 1040.813171510543
bigram:
	best delta: 0.65
	perplexity: 677.4662636794408
```

بخش سوم)

در ادامه با استفاده از مدل های آموزش دیده شده مصرع های تستی که قرار داده شده است را کامل می کنیم.

با استفاده از مدل unigram نتایج تکمیل مصرع ها به صورت زیر است:

شماره تست	جمله ناقص	جمله کامل	جمله پیشبینی شده
۱	این سخن حقست اگر نزد سخن گستر	این سخن حقست اگر نزد سخن گستر برند	این سخن حقست اگر نزد سخن گستر و
۲	آنکه با یوسف صدیق چنین خواهد	آنکه با یوسف صدیق چنین خواهد کرد	آنکه با یوسف صدیق چنین خواهد و
۳	هیچ دانی چکند صحبت او با	هیچ دانی چکند صحبت او با دگران	هیچ دانی چکند صحبت او با و
۴	سرمه دهی بصر بری سخت خوش است	سرمه دهی بصر بری سخت خوش است تاجری	سرمه دهی بصر بری سخت خوش است و
۵	آتش ابراهیم را	آتش ابراهیم را نبود زیان	آتش ابراهیم را و و
۶	من که اندر سر	من که اندر سر جنونی داشتم	من که اندر سر و و
۷	هر شیر شرزه را که به نیش	هر شیر شرزه را که به نیش سنان گزید	هر شیر شرزه را که به نیش و و
۸	هرکه از حق به	هرکه از حق به سوی او نظریست	هرکه از حق به و و و
۹	گفت این از	گفت این از خدای باید خواست	گفت این از و و و
۱۰	کلاه لاله که لعل است	کلاه لاله که لعل است اگر تو بشناسی	کلاه لاله که لعل است و و و

با استفاده از مدل bigram نتایج تکمیل مصرع ها به صورت زیر است:

شماره تست	جمله ناقص	جمله کامل	جمله پیشبینی شده
۱	این سخن حقست اگر نزد سخن گستر	این سخن حقست اگر نزد سخن گستر برند	این سخن حقست اگر نزد سخن گستر و
۲	آنکه با یوسف صدیق چنین خواهد	آنکه با یوسف صدیق چنین خواهد کرد	آنکه با یوسف صدیق چنین خواهد کرد
۳	هیچ دانی چکند صحبت او با	هیچ دانی چکند صحبت او با دگران	هیچ دانی چکند صحبت او با تو
۴	سرمه دهی بصر بری سخت خوش است	سرمه دهی بصر بری سخت خوش است تاجری	سرمه دهی بصر بری سخت خوش است و
۵	آتش ابراهیم را	آتش ابراهیم را نبود زیان	آتش ابراهیم را به دست
۶	من که اندر سر	من که اندر سر جنونی داشتم	من که اندر سر و از
۷	هر شیر شرزه را که به نیش	هر شیر شرزه را که به نیش سنان گزید	هر شیر شرزه را که به نیش و از
۸	هرکه از حق به	هرکه از حق به سوی او نظریست	هرکه از حق به دست و از
۹	گفت این از	گفت این از خدای باید خواست	گفت این از آن را به
۱۰	کلاه لاله که لعل است	کلاه لاله که لعل است اگر تو بشناسی	کلاه لاله که لعل است و از آن

همان طور که دیده می شود نتایج به دست آمده از مدل unigram در پیش بینی کلمه بعدی نامناسب است و صرفا پرتکرارترین کلمه جایگزین می‌شود که با توجه به روش آن منطقی است و در نتیجه مدل زبانی مناسبی نمی سازد. در حالی که مدل bigram برای پیش بینی کلمه بعدی کلمات متنوع‌تری را پیش بینی می‌کند که هرچند دقیق نیستند و با کلمه واقعی در بیشتر موارد متفاوت است ولی دنباله بی معنی نمی سازد و مثلا بعد از کلمه "خواهد" تشخیص داده شده که کلمه "کرد" پیش بینی شود که در نتیجه مدلی بهتر از unigram برای مدل زبانی فراهم کرده است.

بخش چهارم)

در این قسمت با استفاده از شبکه عصبی می خواهیم مدل زبانی را بسازیم. در پیاده سازی شبکه عصبی از کتابخانه keras استفاده شده است. در ابتدا یک لایه embedding استفاده میکنیم تا بتوانیم از integer encoding استفاده کنیم و با جلوگیری از استفاده از one-hot encoding میتوان از داده های بیشتری در آموزش مدل استفاده کرد. سپس دو لایه dense استفاده میکنیم که به ترتیب دارای ابعاد ۲۵۶ و تعداد واژگان هستند. همچنین به دلیل استفاده از integer encoding از تابع لاس sparse_categorical_crossentropy استفاده میکنیم.

برای فراهم سازی داده های آموزشی مدل، بر روی متن پیمایش میکنیم و هر بار به اندازه یک کلمه به پیش میرویم و دو کلمه اول را به عنوان داده ورودی و بعدی را به عنوان برچسب آن در نظر میگیریم و همچنین سه کلمه را به عنوان داده ورودی و بعدی را به عنوان برچسب آن در نظر میگیریم. در نتیجه ۷۶۱۷۵۵ داده ورودی bigram و ۶۱۰۶۴۰ داده آموزشی trigram داریم. سپس به هر یک از کلمات واژگان یک عدد اختصاص می دهیم و وکتور داده ها را می سازیم.

در ادامه از تمام داده های به دست آمده از train.txt در آموزش مدل استفاده می کنیم و مدل ها را آموزش می دهیم و از داده های فایل valid.txt در validation مدل استفاده می کنیم.

پس از چند epoch اول دقت بر روی داده های آموزشی و ارزیابی افزایش می یابد ولی بعد از آن با وجود بهبود دقت بر روی داده های آموزشی دقت بر روی داده های ارزیابی کاهش دارد که نشان overfit شدن مدل بر روی داده های آموزشی است که آموزش را تا همین مرحله متوقف می کنیم به همین دلیل آموزش هر دو مدل تنها تا ۷ epoch ادامه یافته است. فرایند آموزش مدل بر روی داده‌ها به صورت زیر است که برای این آموزش از colab و GPU استفاده شده است تا فرایند آموزش وقت کمتری نیاز داشته باشد:

bigram:

```
Epoch 1/7
744/744 [==============================] - 67s 89ms/step - loss: 7.7561 - accuracy: 0.0530 - val_loss: 7.3871 - val_accuracy: 0.0651
Epoch 2/7
744/744 [==============================] - 66s 89ms/step - loss: 7.1900 - accuracy: 0.0734 - val_loss: 7.2345 - val_accuracy: 0.0776
Epoch 3/7
744/744 [==============================] - 67s 90ms/step - loss: 6.9188 - accuracy: 0.0853 - val_loss: 7.1805 - val_accuracy: 0.0813
Epoch 4/7
744/744 [==============================] - 68s 92ms/step - loss: 6.6511 - accuracy: 0.0956 - val_loss: 7.1849 - val_accuracy: 0.0830
Epoch 5/7
744/744 [==============================] - 69s 92ms/step - loss: 6.3885 - accuracy: 0.1052 - val_loss: 7.2288 - val_accuracy: 0.0834
Epoch 6/7
744/744 [==============================] - 69s 92ms/step - loss: 6.1357 - accuracy: 0.1154 - val_loss: 7.3007 - val_accuracy: 0.0839
Epoch 7/7
744/744 [==============================] - 69s 92ms/step - loss: 5.8947 - accuracy: 0.1281 - val_loss: 7.3927 - val_accuracy: 0.0834
```

trigram:

```
Epoch 1/7
597/597 [==============================] - 53s 89ms/step - loss: 7.8852 - accuracy: 0.0506 - val_loss: 7.5319 - val_accuracy: 0.0545
Epoch 2/7
597/597 [==============================] - 53s 89ms/step - loss: 7.3189 - accuracy: 0.0647 - val_loss: 7.3536 - val_accuracy: 0.0730
Epoch 3/7
597/597 [==============================] - 53s 89ms/step - loss: 7.0198 - accuracy: 0.0801 - val_loss: 7.2686 - val_accuracy: 0.0793
Epoch 4/7
597/597 [==============================] - 54s 91ms/step - loss: 6.6955 - accuracy: 0.0920 - val_loss: 7.2619 - val_accuracy: 0.0809
Epoch 5/7
597/597 [==============================] - 55s 92ms/step - loss: 6.3604 - accuracy: 0.1032 - val_loss: 7.3150 - val_accuracy: 0.0817
Epoch 6/7
597/597 [==============================] - 55s 92ms/step - loss: 6.0324 - accuracy: 0.1164 - val_loss: 7.4126 - val_accuracy: 0.0812
Epoch 7/7
597/597 [==============================] - 55s 92ms/step - loss: 5.7172 - accuracy: 0.1349 - val_loss: 7.5443 - val_accuracy: 0.0805
```

در ادامه بر روی داده آزمون مقدار perplexity را به دست می آوریم. به این صورت که هر سه کلمه یا چهار کلمه متوالی را در نظر میگیریم و کلمه آخر را به عنوان برچسب در نظر میگیریم و بقیه را به عنوان ورودی به مدل می دهیم و در وکتور خروجی که احتمال هر کلمه مشخص شده است احتمال برچسب جدا شده را به دست می آوریم و این کار را برای پیش بینی کل یک سطر از داده ارزیابی انجام میدهیم و در فرمول perplexity مانند قسمت قبل استفاده میکنیم. سپس با محاسبه perplexity بر روی هر سطر این مقادیر را میانگین می گیریم تا perplexity کل مدل زبانی به دست آید. مقدار perplexity برای دو مدل unigram و bigram شبکه عصبی به صورت زیر است:

```
perplexity of neural bigram LM: 547.7307027241197
perplexity of neural trigram LM: 473.6301467049531
```

بخش پنجم)

جملات ناقص را با استفاده از مدل های زبانی ساخته شده با شبکه عصبی کامل می کنیم. در این فرایند برای مدل bigram دو کلمه پایانی جمله ناقص را به مدل می دهیم و محتمل ترین کلمه بعدی را به دست می‌آوریم

و برای پیش بینی کلمه بعدی نیز همین فرایند را تکرار می کنیم و برای مدل trigram همین کار را برای سه کلمه پایانی انجام می دهیم. خروجی ها به صورت زیر است:

کامل کردن جملات با استفاده از مدل شبکه عصبی bigram:

شماره تست	جمله ناقص	جمله کامل	جمله پیشبینی شده
۱	این سخن حقست اگر نزد سخن گستر	این سخن حقست اگر نزد سخن گستر برند	این سخن حقست اگر نزد سخن گستر و
۲	آنکه با یوسف صدیق چنین خواهد	آنکه با یوسف صدیق چنین خواهد کرد	آنکه با یوسف صدیق چنین خواهد که
۳	هیچ دانی چکند صحبت او با	هیچ دانی چکند صحبت او با دگران	هیچ دانی چکند صحبت او با تو
۴	سرمه دهی بصر بری سخت خوش است	سرمه دهی بصر بری سخت خوش است تاجری	سرمه دهی بصر بری سخت خوش است و
۵	آتش ابراهیم را	آتش ابراهیم را نبود زیان	آتش ابراهیم را به جان
۶	من که اندر سر	من که اندر سر جنونی داشتم	من که اندر سر و پا
۷	هر شیر شرزه را که به نیش	هر شیر شرزه را که به نیش سنان گزید	هر شیر شرزه را که به نیش و گل
۸	هرکه از حق به	هرکه از حق به سوی او نظریست	هرکه از حق به جز تو نیست
۹	گفت این از	گفت این از خدای باید خواست	گفت این از آن زلف و
۱۰	کلاه لاله که لعل است	کلاه لاله که لعل است اگر تو بشناسی	کلاه لاله که لعل است و در دل

کامل کردن جملات با استفاده از مدل شبکه عصبی trigram:

شماره تست	جمله ناقص	جمله کامل	جمله پیشبینی شده
۱	این سخن حقست اگر نزد سخن گستر	این سخن حقست اگر نزد سخن گستر برند	این سخن حقست اگر نزد سخن گستر بود
۲	آنکه با یوسف صدیق چنین خواهد	آنکه با یوسف صدیق چنین خواهد کرد	آنکه با یوسف صدیق چنین خواهد کرد
۳	هیچ دانی چکند صحبت او با	هیچ دانی چکند صحبت او با دگران	هیچ دانی چکند صحبت او با دل
۴	سرمه دهی بصر بری سخت خوش است	سرمه دهی بصر بری سخت خوش است تاجری	سرمه دهی بصر بری سخت خوش است و
۵	آتش ابراهیم را	آتش ابراهیم را نبود زیان	آتش ابراهیم را در جهان
۶	من که اندر سر	من که اندر سر جنونی داشتم	من که اندر سر آن که
۷	هر شیر شرزه را که به نیش	هر شیر شرزه را که به نیش سنان گزید	هر شیر شرزه را که به نیش می کند
۸	هرکه از حق به	هرکه از حق به سوی او نظریست	هرکه از حق به دست حقورست و
۹	گفت این از	گفت این از خدای باید خواست	گفت این از آن روی تو
۱۰	کلاه لاله که لعل است	کلاه لاله که لعل است اگر تو بشناسی	کلاه لاله که لعل است و به هر

همان طور که در جداول بالا دیده می شود نتایج پیشبینی کلمه های بعدی با استفاده از مدل های bigram و trigram هر دو توانسته اند کلمه های بعدی را به گونه ای پیش بینی کنند که ادامه مصرع با معنا است و مدل trigram توانسته کلمات مناسب تری را پیش بینی کند. به عنوان مثال در تست شماره ۱ مدل bigram "و" را پیشبینی کرده در حالی که مدل tigram کلمه "بُوَد" را پیشبینی کرده که کلمه ای مناسب تر است و دلیل این بهبود هم مشخصا به دلیل استفاده از سه کلمه قبل به جای دو کلمه قبل برای پیشبینی کلمه بعدی است که باعث می شود مدل محتوای بیشتری در اختیار داشته باشد و بتواند بهتر کلمه بعد را تشخیص دهد.

بخش ششم:

استفاده از شبکه عصبی در ساخت مدل زبانی نسبت به روش آماری باعث می شود برای هر کلمه از واژگان وکتوری ساخته شود و این وکتور آموزش ببیند و مشکلاتی محاسبات احتمال ها و صفر شدن احتمال ها را نداریم. همچنین روش شبکه عصبی بهتر می تواند محتوا را یادبگیرد و نسبت به روش آماری تعمیم پذیری بهتری دارد.

با مقایسه نتایج دو روش آماری و شبکه عصبی در ساخت مدل زبانی می بینیم که در پیش بینی کلمه بعدی روش شبکه عصبی بهتر عمل کرده است و کلمات پیشنهادی آن مناسب تر هستند که دلیل آن همان دلایلی است که گفته شد. همچنین مقدار perplexity در حالت استفاده از شبکه عصبی نسبت به حالت آماری مقدار کمتری دارد که نشان دهنده بهتر بودن مدل زبانی شبکه عصبی است.

به عنوان مثال تست اول جداول بالا را اگر مقایسه کنیم در حالت bigram آماری پیش بینی کرده است " نزد سخن گستر و " و bigram شبکه عصبی پیش بینی کرده است " نزد سخن گستر بود " که نشان می دهد هر دو مقدار معناداری پیش بینی کرده اند ولی در شبکه عصبی کلمه بهتری پیش بینی کرده است.