تمرین 3 یادگیری عمیق

توضیحات سوال 1

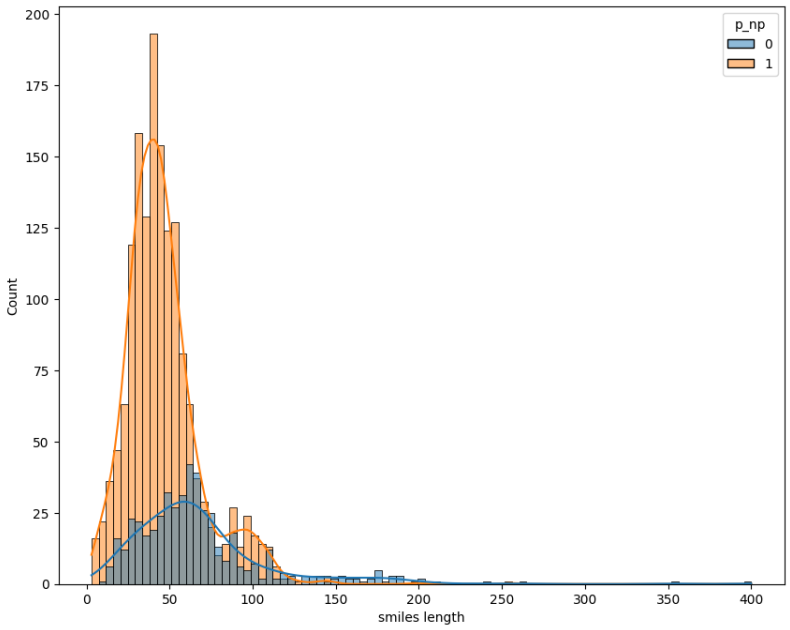
امیرعباس افضلی 400100662

الف)

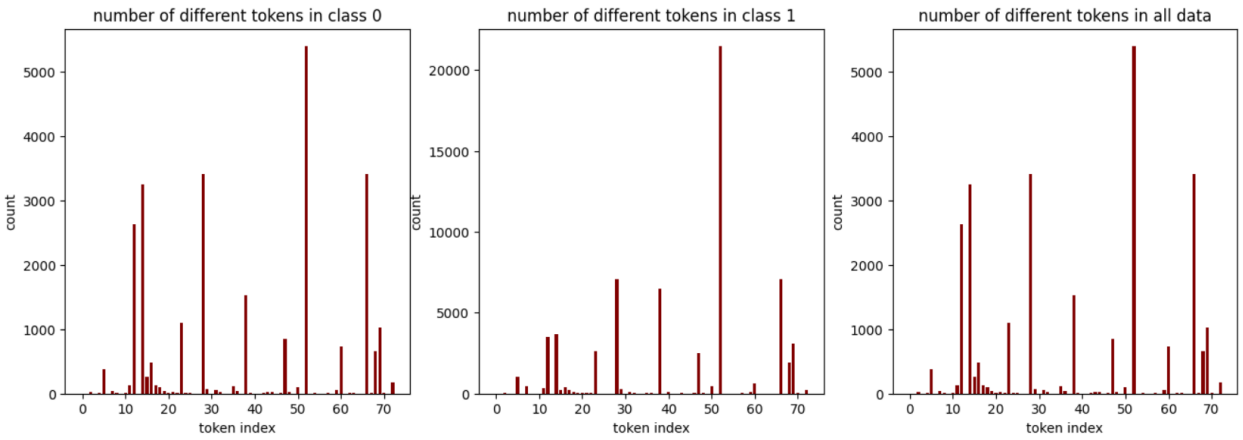
ابتدا دیتاست داده شده را لود می کنیم و آن را به DataFrame تبدیل می کنیم.

	num	name	p_np	smiles
0	1	Propanolol	1	[Cl].CC(C)NCC(O)COc1cccc2ccccc12
1	2	Terbutylchlorambucil	1	C(=O)(OC(C)(C)C)CCCc1ccc(cc1)N(CCCl)CCCl
2	3	40730	1	c12c3c(N4CCN(C)CC4)c(F)cc1c(c(C(O)=O)cn2C(C)CO...
3	4	24	1	C1CCN(CC1)Cc1cccc(c1)OCCCNC(=O)C
4	5	cloxacillin	1	Cc1onc(c2ccccc2Cl)c1C(=O)N[C@H]3[C@H]4SC(C)(C)...

که در آن نام ملکول، نفوذ پذیر یا ناپذیر بودن آن و SMILES مربوط به هر ملکول آمده است. هیستوگرام زیر توزیع طول SMILES دو کلاس مختلف را نمایش می دهد.



نمودار تکرار توکن های مختلف در دو کلاس:



درمورد SMILES:

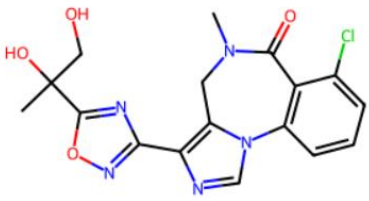
فرمت SMILES (Simplified Molecular Input Line Entry System) یک روش نمایش ساختار سازماندهی نشده مولکول‌ها است. این فرمت از کاراکترها و اعداد به‌عنوان یک رشته متنی برای نمایش اطلاعات ساختار مولکولی استفاده می‌کند. این نمایش مولکول‌ها به‌صورت خطی و ساده است و امکان تشخیص سریع ساختار مولکول را فراهم می‌کند.

SMILES می‌تواند به‌عنوان یک روش استاندارد برای انتقال اطلاعات ساختار مولکولی در پایگاه داده‌ها و نرم‌افزارهای شیمی مورد استفاده قرار بگیرد. این فرمت از استانداردهای گسترده‌ای برخوردار بوده و قابلیت ترسیم و تحلیل ساختار مولکول‌ها را فراهم می‌کند.

این روش معمولاً برای توصیف ساختارهای شیمیایی در محیط‌هایی همچون شیمی‌سنجی محاسباتی، پایگاه داده‌های شیمیایی و نرم‌افزارهای تحلیل ساختار مولکولی استفاده می‌شود.

همچنین با استفاده از این SMILES می توانیم شکل هندسی ملکول را نشاهده کنیم، مثلا:

OCC(C)(O)c1onc(c2ncn3c2CN(C)C(c4c3cccc4Cl)=O)n1



با مطالعه ی این لیترچر یافتیم که عبارت ها (توکن ها) یی که اغلب در SMILES ملکول ها وجود دارد، توکن های زیر هستند: (115 توکن)

```
{'[c-]', '[SeH]', '[N]', '[C@@]', '[Te]', '[OH+]', 'n', '[AsH]',
 '[B]', 'b', '[S@@]', 'o', ')', '[NH+]', '[SH]', 'O', 'I', '[C@]',
'-', '[As+]', '[Cl+2]', '[P+]', '[o+]', '[C]', '[C@H]', '[CH2]',
'\\', 'P', '[O-]', '[NH-]', '[S@@+]', '[te]', '[s+]', 's', '[B-]',
'B', 'F', '=', '[te+]', '[H]', '[C@@H]', '[Na]', '[Si]', '[CH2-]',
'[S@+]', 'C', '[se+]', '[cH-]', '6', 'N', '[IH2]', '[As]', '[Si@]',
'[BH3-]', '[Se]', 'Br', '[C+]', '[I+3]', '[b-]', '[P@+]', '[SH2]',
'[I+2]', '%11', '[Ag-3]', '[O]', '9', 'c', '[N-]', '[BH-]', '4',
'[N@+]', '[SiH]', '[Cl+3]', '#', '(', '[O+]', '[S-]', '[Br+2]',
'[nH]', '[N+]', '[n-]', '3', '[Se+]', '[P@@]', '[Zn]', '2', '[NH2+]',
'%10', '[SiH2]', '[nH+]', '[Si@@]', '[P@@+]', '/', '1', '[c+]',
'[S@]', '[S+]', '[SH+]', '[B@@-]', '8', '[B@-]', '[C-]', '7', '[P@]',
'[se]', 'S', '[n+]', '[PH]', '[I+]', '5', 'p', '[BH2-]', '[N@@+]',
'[CH]', 'Cl'}
```

(Adopted from DeepChem)

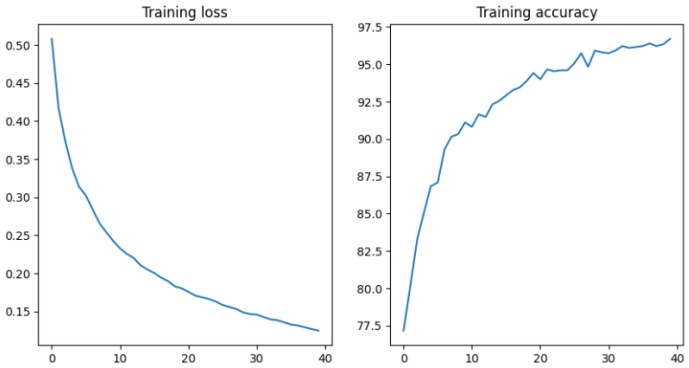
حال به کمک regex دیتاست خودمان را توکنایز می کنیم. مشاهده می شود که در دیتاست کلن 73 توکن وجود دارد که در توکن های بالا هم وجود دارند.

در نهایت هر توکن را به صورت one hot تبدیل می کنیم و سایز دیتاست پراسس شده به صورت زیر می باشد:

```
data shape:  (2050, 120, 73)
```

ب)

نتایج پردیکشن با FC model :



و در نهایت ترینینگ نتایج k-fold validation:

```
Train accuracy: 93.66586538461539 ± 0.7045361004688822
Test accuracy: 87.94748520710058 ± 1.6464121565972247
```

و)

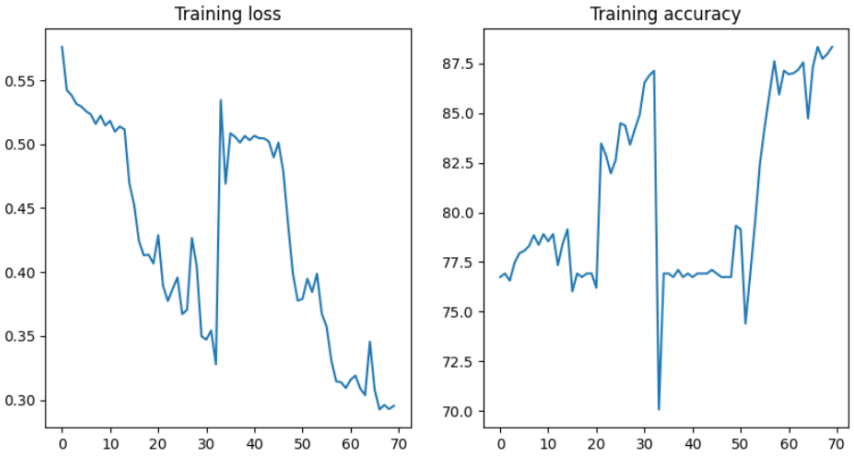
LSTM و BILSTM هر دو از مدل‌های شبکه‌های عصبی بازگشتی (RNN) هستند که برای پردازش داده‌های دنباله‌ای مانند متون و صوت‌ها استفاده می‌شوند. اما اختلاف اصلی بین آنها در روش پردازش داده‌ها است.

LSTM یک نوع از RNN است که برای حل مشکل از بین رفتن اطلاعات در زمان طولانی استفاده می‌شود.

BILSTM یک نوع دیگر از مدل‌های RNN است که با داشتن دو لایه LSTM، یکی به صورت معکوس از دیگری عمل می‌کند. این نوع از مدل به ما اجازه می‌دهد تا از اطلاعات قبل و بعد از زمان جاری همزمان استفاده کنیم و می‌تواند الگوهای پیچیده‌تری را شناسایی کند.

به طور کلی، LSTM و BILSTM هر دو برای پردازش داده‌های دنباله‌ای مفیدند، اما BILSTM از قابلیت‌های بیشتری برای درک زمان وابستگی دارد.حال با معماری های LSTM و BILSTM این کار را انجام می دهیم.

نتایج LSTM :

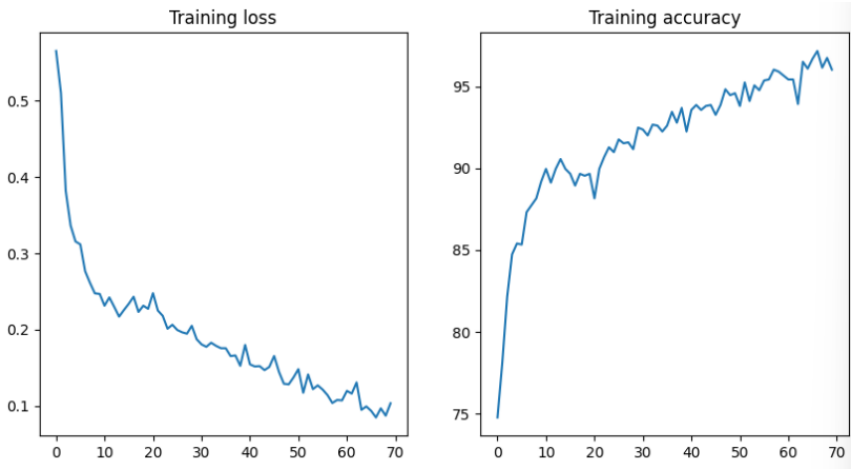


5-fold validation:

```
Train accuracy: 84.11057692307693 ± 4.0573761497320175
Test accuracy: 83.22855029585799 ± 2.6038497087786525
```

البته این معماری در ترین کردن هم مشکلاتی بوجود میارد و در گاهی مواقع مدل بایاس می شود و صرفا یک کلاس را پردیکت می کند. که برای کم کردن این مشکل از dropout در آن استفاده کردیم. ولی همچنان بدلیل تعداد پارامتر های زیاد و کم بودن دیتای ترین ، دقت این مدل از مدل FC کمتر می باشد.

نتایج BILSTM :



5-fold validation:

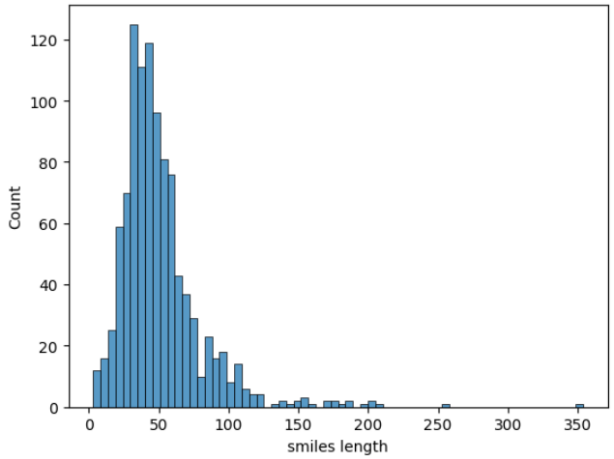
```
Train accuracy: 87.86057692307693 ± 4.004767291328213
Test accuracy: 87.40754437869823 ± 4.8290518918980085
```

در این معماری بدلیل پراسس دوطرفه ی یک sequence ، مشکل قبلی گفته شده در LSTM بوجود نیامد و نتایج هم طبق انتظار بهتر شد.

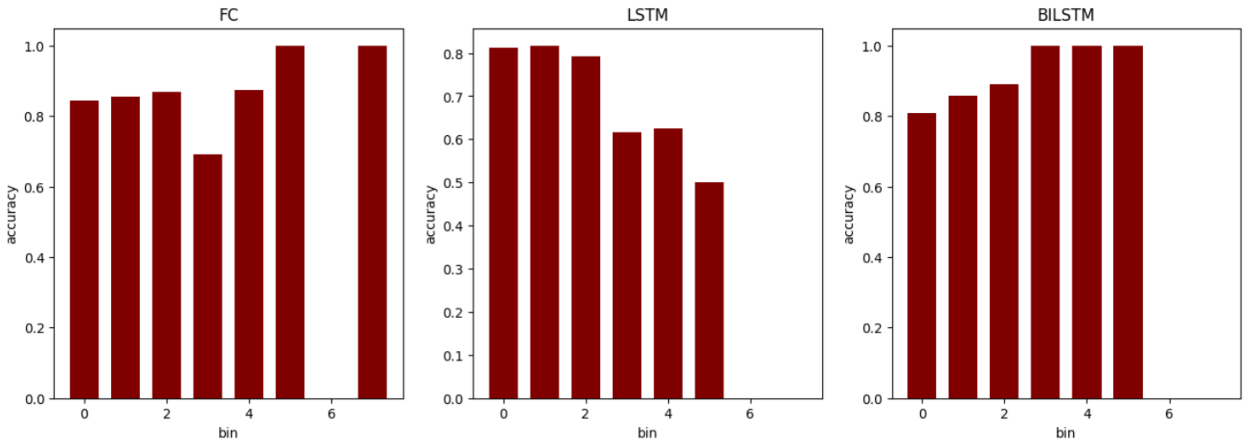
ز)

در این بخش بدلیل این که در بخش های قبلی از kfold استفاده کردیم و دیتای تست یکتایی برای سه مدل وجود نداشت، دوباره بر روی کل دیتاست ترین تست اسپلیت انجام می دهیم و سه مدل را دوباره ترین می کنیم. همچنین برای وجود تنوع در length دیتای تست ، با نسبت 0.5 ترین تست اسپلیت را انجام می دهیم.

توزیع طول SMILES دیتای تست:



دیتای تست را به 10 بین تقسیم می کنیم و اکیبورسی هر مدل را در هر بین بدست میاوریم که به صورت زیر است : (در دو بین اصلا دیتایی وجود نداشته)



در نمودار های بالا از چپ به راست طول SMILES افزایش می یابد. همچنین در دو bin آخر در هر کدام یک دیتا وجود داشته که هم LSTM و هم BILSTM اشتباه پردیکت کرده اند.

همچنین طبق انتظار با افزایش طول، نتیجه ی LSTM افت می کند و دقت آن کمتر می شود. ولی این مشکل در BILSTM بدلیل پراسسینگ دو طرفه بسیار کمتر هست که در نمودار بالا هم مشهود است.

این مشکل کاهش دقت با افزایش طول دنباله ، یک مشکل رایج در مدل های recurrence می باشد که امروزه باعث ترجیح مدل های ترنسفورمری بر این مدل ها شده است.