Sobhan ranjbar

در این پروژه قرار است یکی از مهم ترین موارد NLP یعنی تحلیل احساسات را انجام دهیم و من برای راحتی کار مراحل انجام شده را در بلوک های مختلف گذاشتم که راحت تر قابل توضیح باشد.

قسمت اول

در قسمت اول تمام کتابخونه های مورد نیاز را اضافه کردم مواردی که از کتابخونه توزچ نیاز داشتم اوردم و از هزم برای قسمت پیش پردازش استفاده کردم.

قسمت دوم

لود کردن داده می باشد که من قصد داشتم به دو روش داده ها رو بارگذاری کنم که متاسفانه به دلایلی نشد و مستقیم از کگل نگذاشتم و فقط در اول فایل نشون دادم که چگونه می توان از اون روش استفاده کرد و در روش دوم از حافظه موقت کولب استفاده کردم و فایل ها را لود کردم من برای راحتی کار اگر امتیاز موردی بیشتر از 50 بود مثبت گرفتم و کمتر منفی به این دلیل که داکیومنت این گونه نوشته بود که می خواهد.

و بعد پیش پردازی را انجام دادم و کلمات را به توکن تبدیل کردم.

قسمت سوم

برای این که کلمات را داشته باشیم باید دیکشنری از کلمات بسازیم و بدانیم چه کلماتی داریم و همه بردار ها نیز به این شکل نرمالیز می شوند یعنی اندازه هم می شوند.

در ادامه طول هر بدار کلمات را 100 تنظیم می کنیم و به داده ها اعمال می کنیم.

از مواردی مثل pad استفاده کردیم بعضی اوقات جمله ای که داریم مثلا 100 کلمه پر نمی کنه به همین دلیل هرچقدر خالی موند pad می ذاریم که مدل می فهمه این ها رو نخونه جمله تموم شده.

قسمت چهارم

داده ها را به دو بخش امورزش و ارزیابی می شکنیم اندازه ارزیابی را 20 درصد در نظر گرفته و از عدد تصادفی 42 استفاده می کنیم به این دلیل که هر بار اگر این عدد را وارد کردیم عدد های انتخابی مواردی باشد که قبلا انتخاب کردیم

نمی دونم چقدر این واضح گفتم یا نه 😂

قسمت پنجم

از اون جایی که از ما خواسته شده و ما هم مجبوریم اومدیم از مواردی که گفتند استفاده کردیم از اون جایی که از ما خواسته شده و ما هم مجبوریم اومدیم از مواردی که گفتند استفاده کردیم الله الموزش کنه و اگر رم پر شد در مرحله اموزش به مشکل نخوریم یعنی داریم دسته های 64 تایی می کنیم و تیکه تیکه می دیم مدل بخوره.

قسمت ششم

ساخت مدل

به همون شکل که گفته شد در داک از امبدینگ و Istm ولایه fully connected استفاده کردیم (من وسطش انگلیسی می نویسم فارسیش واقعیت نمی دونم چی می شه)

قسمت هفتم

هایپریارامتر ها را ست می کنیم.

قسمت هشتم

از loss function

criterion = nn.BCEWithLogitsLoss()

استفاده کردیم واقعیتش چیزی در موردش نمی دونم داک گفته بود استفاده کردم

از بهینه ساز ادام استفاده کردیم با نرخ یادگیری 0.001 بر اساس داک.

قسمت نهم

اموزش مدل

بسته های همون بچ ها را هر دفعه می گیریم و مدل را اپدیت می کنیم

هر دفعه گرادینت را ریست می کنیم و پیش بینی رو بتج گرفته شده را انجام می دهیم

بعد از ان لاس فانكشن را حساب مي كنيم

و دقت بدست امده را اندازه گیری می کنیم چون هم لیبل واقعی را داریم و هم پیش بینی این دو رو هم از کم کنیم می تونیم ببینیم چقدر اشتباه کردیم

بعد از اون مدل را باک وارد اپدیت می کنیم

و در اخر میزان لاس و دقت را جمع می کنیم و در قسمت خروجی تابع دقت و لاس کلی را بر می گردانیم.

قسمت دهم

ارزیابی تابع با استفاده از از داده های validation

قسمت يازدهم

حلقه آموزش ، تابع اموزش و ارزیابی که قبلا نوشتیم را این جا استفاده می کنیم.

هر دفعه لاس وليو چک مي کنيم اگه بهتر باشه اون مدل ذخيره مي کنيم.

قسمت دوازدهم

بهترین مدل که از قبل یازدهم داریم را بارگذاری می کنیم و رو تستی که الان داریم انجام می دهیم ببینم نتیجه چگونه است.

قسمت سيزدهم

محاسبه متریک هایی که هم داک بود و هم باید بینیم واقعا مدل خوب اموزش دیده که استفاده کنیم یا خیر.

قسمت های اضافه تر برای نمره بیشتر را نوشته ام بسیار ساده و واضح نوشتم که از روش سنتی TF-IDF استفاده کردم. استفاده کردم و برای قسمت بعدی نیز از Bert استفاده کردم.