

تمرین سوم یادگیری عمیق

زهرآ ختن لو ۹۲۰۴۳۹۸۲

سوال های تشریحی

۱. مدل های LSTM و Transformer چه تفاوت هایی باهم دارند؟

در واقع LSTM ها از دسته شبکه های بازگشتی هستند. ترنسفورمرها قابلیت موازی سازی دارند و به همین دلیل، بر روی GPU ها می توان چنین مدل هایی را train کرد و به نتایج خوبی از لحاظ پرفورمنس رسید.

شبکه های LSTM برای کاربردهایی که به طریقی sequential هستند و یک ترتیب دارند، مثلاً تسک هایی که وابسته به زمان هستند، استفاده می شوند.

۲. چه مزایایی در مدل Bert باعث میشود تا روی کاربردهای پردازش متن، عملکرد مناسبی داشته باشد؟

از آنجایی که BERT برای پیش بینی کلمات ماسکه در متن آموزش دیده، و از آنجا که هر جمله را بدون جهت خاصی تجزیه و تحلیل می کند، نسبت به روش های قبلی NLP، مانند روش های جاسازی، در درک معنای همنام ها بهتر عمل می کند.

به طور گسترده گزارش شده است که این مدل جدید کندتر از مدل های قبلی خود است. با این حال، اکثر متخصصان در این زمینه موافق هستند که این مدل از سایر مدل ها بهتر عمل می کند، زیرا آگاهی قوی تری از زمینه هر متنی که تحلیل می کند دارد.

به طور خلاصه، این ۳ دلیل است که چرا BERT احتمالاً یک تغییر دهنده در NLP است:

- دو طرفه است
- این مدل زبان ماسکه (MLM) و پیش بینی جمله بعدی (NSP) را ترکیب می کند.
- تا کنون، این بهترین روش در NLP برای درک متون سنگین زمینه است
- برای زبان های غیر از انگلیسی نیز از قبل آموزش دیده و قابل استفاده است.

۳. تفاوت مدل BERT با مدل GPT ۲ چیست؟ این دو مدل را با هم مقایسه کرده و توضیح دهید.

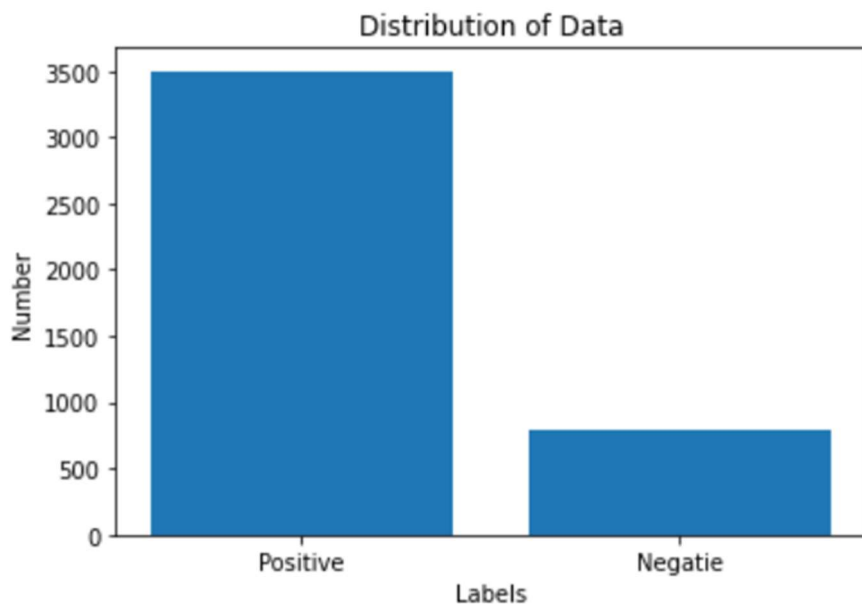
Bert فقط انکدر دارد ولی GPT2 فقط دیکدر دارد،
Bert پریترن میشه و در تسک های مختلف قابل استفاده است. ولی GPT2 تنها برای شرکت های بزرگ و مطرح قابل استفاده است.
در مدل bert می توان برای تسک هایی که بعد از کلمات ماسکه شده را نیاز دارد ببیند اما در GPT2 تنها می تواند قبل از آن را ببیند.
مدل bert دو طرفه است اما GPT2 یک طرفه است.

توضیح کد و بررسی نتایج

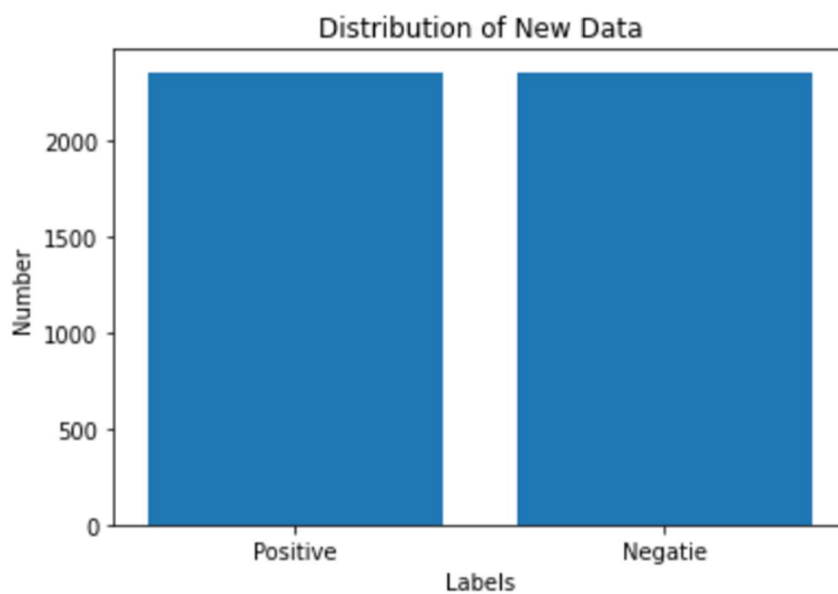
در ابتدا پس از نصب و import کردن پکیج ها و کتابخانه های مورد نیاز، باید دیتاست را آماده کنیم. در ابتدا طبق کد آماده دیتاست را لود کرده و در مرحله normalization ابتدا دیتاستمان را برچسب گذاری می کنیم. (بر اساس توضیحات آمده اگر rate کمتر از ۳ باشد لیبل منفی می دهیم و در غیر اینصورت لیبل مثبت).
در مرحله بعد باید دیتا ستما را پیش پردازش را انجام می دهیم برای مثال تگ های html، لینک های URL، ایموجی و... را حذف می کنیم. و در نهایت در دیتاست، کامنت های پردازش شده را مرتب می کنیم.
از جمله پردازش های دیگر اینکه ابتدا تعداد کلمات کامنت ها را محاسبه کرده و به عنوان یک ستون به دیتاست اضافه می کنیم و در مرحله بعد کامنت هایی که طول آن ها کمتر از ۳ یا بیشتر از ۲۵۶ بوده را حذف می کنیم.

در مرحله بالانس کردن با توجه به اینکه داده های به صورت یکسان نیستند و داده های برچسب زده شده مثبت و منفی در یک اندازه نیستند. اندازه داده کمتر را ۲ یا ۳ برابر کرده به این معنی که بعضی داده ها را به صورت رندوم تکرار می کنیم و داده بیشتر را نیز به نسبت و رندوم حذف می کنیم.

قبل از اینکه این مرحله انجام شود نسبت داده های مثبت و منفی بالانس شود نسبت داده ها به شکل زیر بوده:



بعد از انجام این مرحله و بالانس کردن تعداد داده ها نسبت به هم نمودار زیر بدست می آید:



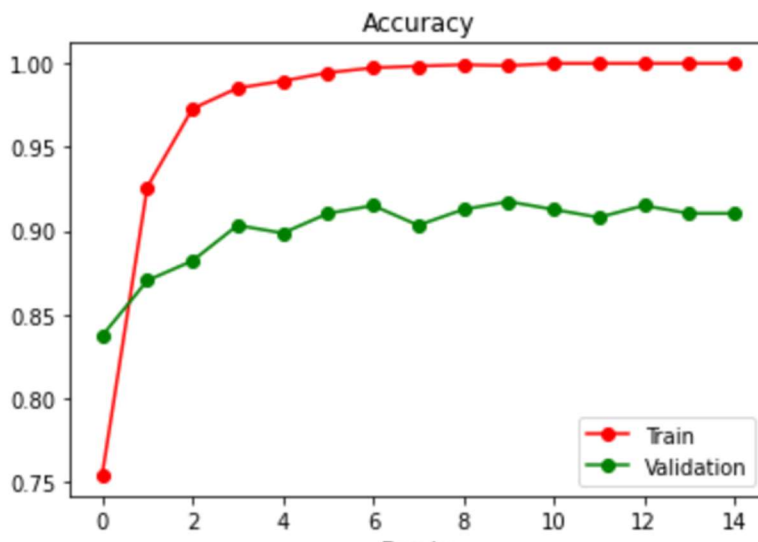
در مرحله ساختن مدل از مدل pretrained شده hooshvareLab استفاده می کنیم مدل را با فانکشن bert و attention می سازیم.

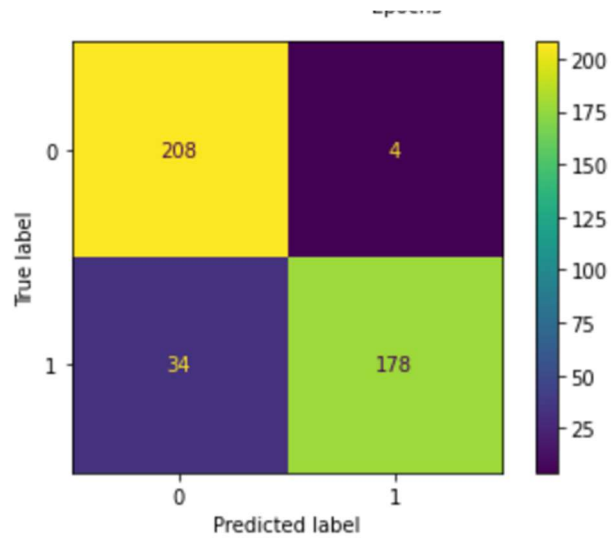
تابع acc_and_f1 را با استفاده از اختلاف y پیش بینی شده و y تارگت و توابع mean و f1 محاسبه می کنیم.

در بخش **training** ابتدا باید پارامتر های `input_ids`, `attention_mask`, `token_type_ids`, `targets` را تعریف می کنیم. در ادامه از مدلی که ساختیم و با استفاده از پارامتر های تعریف شده برای پیش بینی خروجی استفاده می کنیم. در انتها باید احتمال بدست آمده را به کلاس هایی که داریم تبدیل کنیم تا بتوانیم تابع `loss` را محاسبه کنیم.

در بخش `optimizer, scheduler & loss function` از تابع `AdamW` برای `optimize` کردن و `get_linear_schedule_with_warmup` به عنوان `scheduler` و از `crossEntropyLoss` به عنوان تابع `loss` استفاده می کنیم.

در نهایت و در بخش **Complete Training & Plot Loss and Accuracy Diagram** یک سری پارامتر ها را تعریف می کنیم و از توابعی که پیاده سازی کرده بودیم برای سنجش مدل روی دادههای آموزش و `validation` استفاده می کنیم. نتایج به شکل زیر بدست می آید:





در نهایت از مدل ساخته شده برای پیش بینی داده های تست نیز استفاده می کنیم تا خروجی های داده تست نیز بدست آید و از **f1-score & Precision & Recall** برای ارزیابی پیش بینی انجام شده استفاده می کنیم:

	precision	recall	f1-score	support
Neg	0.88	0.96	0.92	235
Pos	0.96	0.87	0.91	236
accuracy			0.92	471
macro avg	0.92	0.92	0.92	471
weighted avg	0.92	0.92	0.92	471