پروژه نهایی درس یادگیری عمیق

مقایسه عملکرد و دقت شبکه‌های CNN و Transformer بر روی مجموعه داده WELFake

سید سروش مجد

۴۰۰۴۴۳۱۸۱

درس یادگیری عمیق | تیر ۱۴۰۱

استاد درس: جناب آقای دکتر حامد ملک

فهرست مطالب

صفحه

مقدمه

در این پروژه می‌خواهیم عملکرد و دقت شبکه‌های CNN و Transformer را در حوزه دسته‌بندی متن بررسی و مقایسه کنیم. شبکه‌های CNN و Transformerهای معروف BERT و RoBERTa بر روی مجموعه‌داده WELFake که کسی عملیات دسته‌بندی بر روی آن انجام نداده بود به طور کامل پیاده‌سازی شد و نتایج و دقت آن‌ها مورد ارزیابی و مقایسه قرار گرفت.

معرفی مجموعه‌داده

قبل از اینکه مدلمان را تولید کنیم، تلاش می‌کنیم یک دید کلی از مجموعه‌داده به دست آوریم. مجموعه‌داده (WELFake) مجموعه‌ای از ۷۲۱۳۴ مقاله خبری که ۳۵۰۲۸ تای آن‌ها خبر واقعی و ۳۷۱۰۶ تا خبر جعلی با فرمت CSV است. جمع‌آوری‌کنندگان چهار مجموعه‌داده خبری محبوب مانند Kaggle، McIntire، رویترز، Political و BuzzFeed را ادغام کردند تا از Overfitting بیش از حدClassifierها جلوگیری کنند و داده‌های متنی بیشتری برای آموزش بهتر ارائه کنند. مجموعه‌داده شامل چهار ستون است: شماره سریال (شروع از ۰)، عنوان متن خبری، (content) محتوای اخبار و برچسب (۱ برابر با متن غیر جعلی و ۰ متن جعلی) منتشر شده در:

IEEE Transactions on Computational Social Systems: pp. 1-13 (doi: 10.1109/TCSS.2021.3068519).

پیش‌پردازش و نرمال‌سازی داده‌ها

در این قسمت متن‌های هر سمپل از داده‌های موجود در ستونه content را با استفاده از تابع text_clean که در آن اعمالی چون حذف کردنpunctuation ها، اموجی‌ها، اعداد، فاصله‌های اضافی و کوچک کردن حروف بزرگ انگلیسی موجود در متن انجام می‌شود، clean کردم. از بین دادگان موجود در این مجموعه‌داده‌ها به دلیل سنگین بودن مجموعه‌داده ۱۰۰۰۰ تا از آن‌ها را به صورت رندوم انتخاب کردم و ۲۰ درصد آن را به عنوان داده آزمون و ۸۰ درصد آن را به عنوان داده آموزش جدا و با فرمت csv ذخیره کردم.

پیاده‌سازی مدل‌ها

شبکه CNN

قبل از وارد کردن داده‌ها به شبکه آن‌را Tokenize کردم. با استفاده از فریم‌ورک کراس شبکه عصبی کانولوشنی برای متن را پیاده‌سازی کردم. در لایه اول word embedding انجام شده، در لایه دوم از یک لایه کانولوشن با

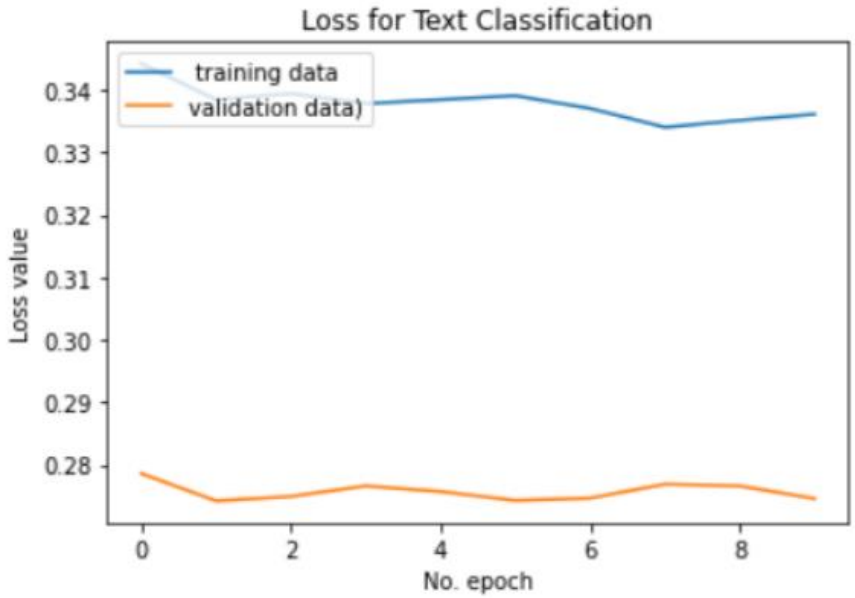
تابع فعال‌سازی رلو استفاده کردم. سپس یک لایه max pooling برای کاهش ابعاد Feature Mapها و در لایه بعدی از Dropout استفاده شده است. و در نهایت لایه dense با دو خروجی و تابع فعالساز sigmoid را برای عمل classification در نظر گرفتیم. (این بخش مربوط به مایل استون اول بوده است و در آن فایل اطلاعات تکمیلی موجود می‌باشد)

Layer (type)	Output Shape	Param #
embedding_3 (Embedding)	(None, 1000, 64)	1280064
conv1d_3 (Conv1D)	(None, 998, 8)	1544
global_max_pooling1d_3 (GlobalMaxPooling1D)	(None, 8)	0
dropout_3 (Dropout)	(None, 8)	0
dense_3 (Dense)	(None, 2)	18

Total params: 1,281,626
Trainable params: 1,281,626
Non-trainable params: 0

شکل ۱ شبکه عصبی استفاده شده

در شکل زیر مشاهده می‌شود شبکه cnn از جایی به بعد توانایی آموزش و فیت ششدن بیشتر ندارد.



شکل ۲ مقادیر loss برای داده‌ای آموزش و ولیدیشن

مدل BERT TRANSFORMER

مدل BERT با مجموعه آموزشی بزرگی آموزش دیده و Pre Trained شده است. شبکهBERT دو تا کار را بر روی ورودی‌ها اعمال می‌کند. اولین کار Masked Language می‌باشد که در آن ٪۱۵ لغات متن ماسکه شده و به مدل داده می‌شوند. همچنین یک لایه Classification با اندازه تعداد لغات با تابع فعالساز softmax به خروجیِ انکودر اضافه شده است. در این روش باید لغات حذف شده توسط مدل پیشبینی شود. کار بعدی Next Sentence Prediction است که در این روش دو جمله با توکن SEP از یکدیگر جدا می‌شوند. شبکه تلاش می‌کند تشخیص دهد این دو جمله متوالی هستند یا نیستند. با استفاده از BERT می‌توان عملیات fine-tuning را انجام داد. در این مدل قسمت بزرگی از متن یکجا به شبکه داده می‌شود و برای تولید بردار هر کلمه، کلمه های قبلی و بعدی هم در نظر گرفته می‌شود. این ویژگی باعث می‌شود word embedding بهتری داشته باشد و دقت مدل بهتر شود.

با استفاده از فریم‌ورک پایتورچ این مدل را پیاده‌سازی کردم. برای پیاده‌سازی مدل ازGPU استفاده شد تا با سرعت بیشتری مدل آموزش داده شود. برای پیاده‌سازی اول داده‌های آموزشی و آزمون را توکن بندی کردم. برای توکن بندی از کتابخانه Transformers استفاده شد و با مدلbert-base-uncased ورودی ها توکن بندی شدند.

خروجی Tokenizer دوتا دیکشنری است که شامل دو کلید است، Input_ids که دنباله‌هایی از اعداد صحیح از جملات ورودی است و مقادیر ۱۰۱و ۱۰۲ را نیز به هر جمله اضافه می‌کند. به دلیل اینکه متن‌ها در سمپل‌ها دارای طول متفاوتی هستند و طول جملات ورودی باید برابر باشند، از padding استفاده شد تا طول همه متن‌ها یکسان شود برای padding از ۱۲۸توکن اول هر نمونه استفاده کردم. همچنین یکی از محدودیت-های مدل BERT این است که حداکثر طول دنباله ۵۱۲ توکن می‌باشد. اگر max_lenght را عددی بزرگتر از طول اکثر جملات انتخاب کنیم تمامی دنباله‌های ورودی با استفاده از توکن padding به این مقدار می‌رسند و این باعث می‌شود مدل نتواند اطلاعات مفیدی از آن‌ها یادبگیرد و همچنین عملیات آموزش کند شود پس ۱۲۸ به عنوان max_length انتخاب شد.

در مرحله بعد دنباله‌های عددی را بهtensor تبدیل کرده و بعد Dataloader را برای مجموعه آموزشی و آزمون فراخوانی کردم. Dataloader دسته ای از داده‌های آموزشی و آزمون را به عنوان ورودی در مرحله آموزش به مدل منتقل می‌کند. سپس با استفاده از مدل BertForSequanceClassification(که یک مدل pretrain شده است و از ۱۲بلوک استفاده کرده است و یک لایه کلسیفایر در بالای آن است) مدل را آموزش دادم. همچنین ازAdamW برای بهینه‌ساز استفاده شد. در آخر دو تابع برایfine tune کردن مدل

روی داده‌ها و ارزیابی آن تعریف شده است. نتایج مدل BERTبرای دادگانمان را در قسمت نتایج گزارش شده است.

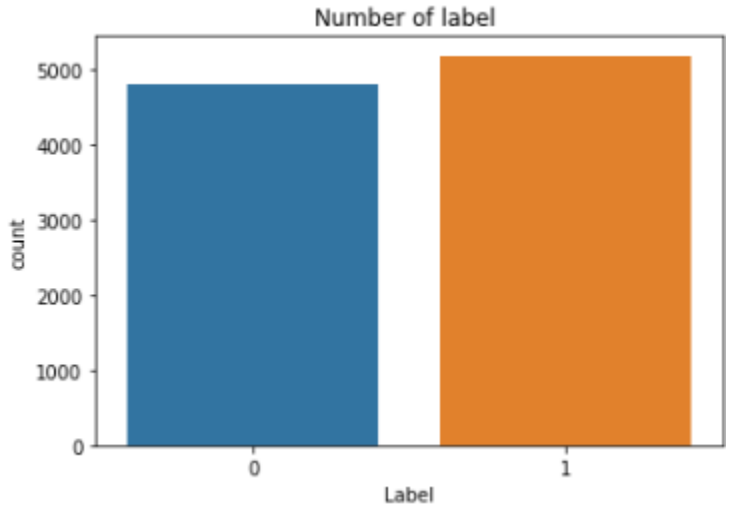
مدل RoBERTa Transformer

مدل pre-trained شده ی RoBERTa بر روی مجموعه‌داده بزرگتری آموزش داده شده است. قسمت پیشبینی کردن جمله بعدی که در BERT بود در این مدل نیس و تعداد minibatch ها برای آموزش مدل در RoBERTa بیشتر از مدل BERT است و اموزش را هر بار روی جمله‌های بلندتر انجام می‌شود. مدل RoBERTa بر خلاف BERT الگوی ماسکه کردن دنباله‌های ورودی اموزشی را هر بار تغییر می‌دهد.

این تغییرات درRoBERTa باعث شده تا در بسیاری از موارد عملکرد بهتری داشته باشد.. ابتدا با استفاده از کتابخانه transformers و مدلRoberta-base دنباله‌های ورودی را توکن بندی کردم و _training Input_ids و training_ Attention_mask را بدست آوردم. برای padding نیز از ۱۲۸توکن اول هر نمونه استفاده شد. تعدادbatch را برابر ۸ و از بهینه سازAdamW استفاده شد. درنهایت، مدل-Roberta base را برای دسته‌بندی روی داده‌های خود fine tune کردم. دقت و مقایسه این مدل نیز در قسمت بعدی ذکر شده است. . با استفاده از فریم‌ورک پایتورچ مدل ربرتا را پیاده‌سازی سد.

نتایج و جمع‌بندی

اول چک می‌کنیم تعداد داده‌ها با برچسب‌های صفر و یک بالانس باشند تا نتایج به سمت برچسب خاصی بایاس نشود (چه برای آموزش و چه برای پیشبینی):



شکل ۳ تعداد برچسب‌های داده‌ها

نتایج شبکه کانولوشن:

نتایج بر روی داده‌های Test(Prediction)

	precision	recall	f1-score	support
0	0.65	0.72	0.68	7080
1	0.69	0.62	0.66	7207
accuracy			0.67	14287
micro avg	0.67	0.67	0.67	14287
weighted avg	0.67	0.67	0.67	14287

نتایج پیشبینی مدل BERT آموزش با سه ایپاک و سایز بچ برابر با هشت:

Accuracy	Precision	Recall	f1-score
۰.۹۶۷۱۲	۰.۹۵۹۴۱	۰.۹۷۵۸۳	۰.۹۶۷۵۵

:Confusion matrix

٩٤٣	٤١
٢٤	٩٦٩

نتایج پیشبینی مدل RoBERTa آموزش با سه ایپاک و سایز بچ برابر با هشت:

Accuracy	Precision	Recall	f1-score
۰.۹۶۸۱۳	۰.۹۷۹۱	۰.۹۵۶۲۱	۰.۹۶۷۵۴

:Confusion matrix

٩٧٥	٢٠
٤٣	٩٣٩

در نهایت مقدار Accuracy برای CNN برابر با ۰.۶۷، برای مدل BERT برابر با ۰.۹۶۷ و برای RoBERTa برای با ۰.۹۶۸ به دست آمد. شبکه‌های CNN برای دیتاهای Spatial مناسب هستند و می بینیم که برای دیتای متن دقت خوبی نمی‌دهند و شبکه‌های CNN نمی‌توانند داده‌های Sequential را به خوبی مدل کنند و پیوستگی کلمات یک جمله را درک نمی‌کنند . از طرفی شبکه‌های transformer به دلیل استفاده از Attention Self می توانند هر کلمه را با توجه به بقیه کلماتی که در متن آمده است encode کنند پس encoding بهتری خواهند داشت و پیوستگی کلمات را در جمله بهتر درک می‌شود. مدل RoBERTa وBERT نیز در مجموعه‌داده ما نتایج تقریبا یکسانی دارند. تمامی این نتایج در فایل‌های نوت‌بوک ذخیره شده قابل مشاهده و اجرا می‌باشند.

منابع، مراجع و کدها

منابع و مراجع:

https://www.youtube.com/watch?v=MsL79ZIqWpg

مقاله ربرتا:https://arxiv.org/abs/1907.11692

مقاله برت: https://arxiv.org/abs/1810.04805

کدها:

کد ربرتا: https://colab.research.google.com/drive/1Zm09j3sj-rE7ie7FYqSlyGZa4jPKNeOl#scrollTo=_WY7cKdGZe3O

کد برت:

https://colab.research.google.com/drive/1Jq1n11w4LZBdIbba9iYwy1JwHY_GB4O8

کد شبکه عصبی:

https://colab.research.google.com/drive/1NEqWEt4BP3SQpHwzdxjodsMVofafAYOG#scrollTo=D-_MyxSmQqGJ