به نام خدا

درس مبانی یادگیری عمیق

گزارش پروژه پایانی

استاد درس : دکتر مرضیه داوودآبادی

دستیاران : مرتضی حاجی‌آبادی، سحر سرکار، فائزه صادقی، مهسا موفق بهروزی، الناز رضایی، پریسا ظفری، حسن حماد، سید محمد موسوی، کمیل فتحی، شایان موسوی نیا، امیررضا ویشته

دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر

نیمسال اول تحصیلی ۱۴۰۲ - ۱۴۰۳

موضوع:

تحلیل احساسات در متن فارسی

ردیف	نام ونام خانوادگی	شماره دانشجویی
۱	دانشجوی شماره ۱:محمد صادق پولایی	
۲	دانشجوی شماره ۲: فاطمه عسکری	

جدول ۱: مشخصات اعضای گروه

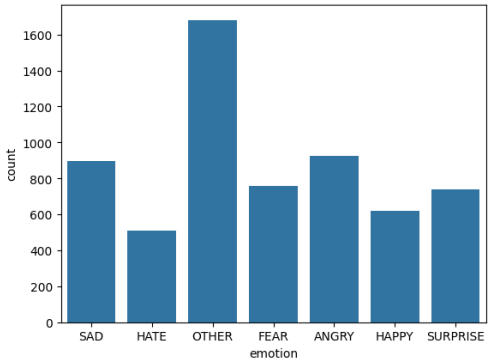
۱ شرح موضوع و مجموعه دادگان

شرح موضوع:

در این پروژه از دیتاست ArmanEmo که یک دیتاست از مجموعه متن های زبان فارسی است که به هر متن یک لیبل از حالات مختلف نسبت داده شده است.دیتاست به دو قسمت train,test تقسیم شده است و هدف پروژه این است با انتخاب مدل مناسب بتوانیم دقت و سایر معیارهای ارزیابی بر روی دیتا تست به مقدار مناسبی برسانیم و بتواند متن هایی که به ان میدهیم را به خوبی دست بندی کند.

شرح مجموع دادگان:

همانطور که در بالا گفته شد دیتاست که استفاده کردیم ArmanEmo استکه شامل ۷ کلاس با توزیع زیر است:



فایل های train,test به فرمت tsv هستند و به هر جمله فارسی یکی از ۷ لیبل بالا نسبت داده شده است.متن ها از شبکه های اجتماعی،نظرات دیجیکالا و .... جمع آوری شده است.

۲ پیش‌پردازش داده ها

به دلیل اینکه جمله ها از جاهای مختلف از جمله نظرات در شبکه های اجتماعی جمع آوری شده اند ممکن بعضی از کلمات شامل حروف اضافه مثلا کلمه "خیییللی" باشند که حالت رسمی ندارند.یا شامل یکسری حروف غیر فارسی باشند باید یک پیش پردازشی داشته باشیم که بتوانیم شکل درست تر کلمات را داشته باشیم.ما طبق مقاله ای که در لینک داک پروژه بود پیش بردازش را انجام دادیم.مراحل زیر را طی کردیم:

۱.ابتدا فایل های tsv را به csv تبدیل کردیم.

۲.حروف انگلیسی را از جملات فارسی حذف کردیم.

۳.اگر یک حرف بیش از دوبار پشت سر هم ظاهر شود دو حرف ان را حذف میکنیم مثلا کلمه "خیییللی" تبدیل میشه به "خیلی"

۴.حروف عربی را از جملات حذف کردیم.

۵.تگ "#" و "_" را از جملات حذف کردیم.

۶.اعداد فارسی را از جملات حذف کردیم.

۷.و در نهایت با استفاده از parsivar جملات را normalize کردیم.

۳ انتخاب مدل

از hugging face استفاده کردیم که یک پلتفرم قدرتمند هست که اجازه دسترسی به مدل های قدرتمندی را می دهد که میشه از library های این پلتفرم مثل tokenize کردن و ... استفاده کرد.از مدل های زیادی استفاده کردیم تا به مدل مطلوب رسیدیم که به صورت زیر است:

طبق مقاله که در داک پروژه بود مدل ها را انتخاب کردیم.

۱.ParsBert :

مدل برت یک انکودر دو جهته از خانواده ترسنفرمر ها هست که توسط گوگل برای تسک mask رو حجم زیاد داده ترین شده.

نتایج به صورت زیر شد:

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.118050	0.630756
2	1.169400	1.045267	0.648132
3	0.616600	1.329898	0.608167
4	0.308900	1.574990	0.618593
5	0.308900	1.742687	0.634231
6	0.126600	2.194758	0.612511
7	0.064200	2.325243	0.636838
8	0.039900	2.550015	0.622068
9	0.039900	2.837571	0.602954
10	0.027600	3.081619	0.594266
11	0.019100	2.917941	0.622937
12	0.023200	2.933458	0.616855
13	0.023200	2.835371	0.634231
14	0.012300	3.020250	0.620330
15	0.013100	3.002439	0.629018
16	0.005900	2.981993	0.626412
17	0.010000	2.978277	0.638575

۲.ALBERT :

یه نسخه سبک تر از برت هست. قابلیت اشتراک گذاری پارامتر بین لایه هارو داره. با استفاده از قابلیت Factorized Embedding Parameterizationکمتر مموری استفاده میکنه. خطای Sentence Order Prediction داره که باعث فهمیدن رابطه بین جمله ها در مدل میشه.

[1915/1915 05:0

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.371873	0.475239
2	1.276700	1.303931	0.528236
3	0.838200	1.384774	0.507385
4	0.581800	1.425434	0.523023
5	0.581800	1.456195	0.526499

۳.roberta_facebook :

[1915/1915 10:45

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.496247	0.421373
2	1.652100	1.225842	0.576890
3	1.095400	1.178461	0.612511
4	0.918900	1.195764	0.605560
5	0.918900	1.172159	0.631625

۴. roberta-base-ft-udpos28:

[1915/1915 10:39, Epoch 5/5]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.505347	0.452650
2	1.602200	1.278498	0.571677
3	1.024100	1.211697	0.600348
4	0.818100	1.171851	0.621199
5	0.818100	1.189166	0.620330

مدل نهایی:

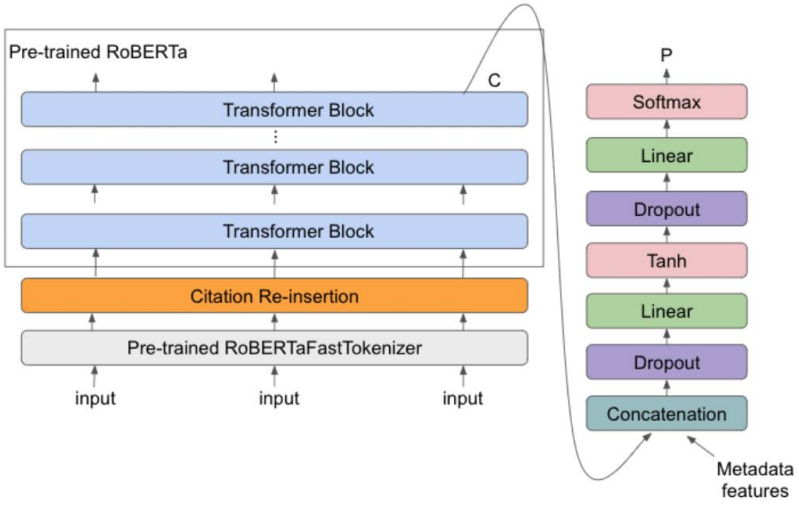
persian_xlm_roberta_large:

یه نسخه ای از برت هست که توسط فیس بوک توسعه داده شده. از Dynamic Masking استفاده میکند در حالی که برت از static masking استفاده میکند یعنی الگوی مسک کردن برای هر مینی بچ تغییر میکند و اجازه میدهد مدل تمامی توکن هارا در هر ایپوک ببیند.رابرتا از مینی بچ های سایز بزرگتر از برت استفاده کرده است و برای مدت طولانی تری و رو حجم بزرگتری از داده ها ترین شده و پرفرمنس مدل رو افزایش داده

xlm-roberta:

یک نسخه چند زبانه از رابرتا هست که روی ۱۰۰ زبان مختلف و ۲.۵ ترابایت دیتا ترین شده است.

ساختار مدل Roberta به صورت زیر است:



و در نهایت با این مدل به بهترین دقت بر روی داده تست رسیدیم

[769/1440 36:20 < 31:47, 0.35 it/s, Epoch 8/15]

Epoch	Training Loss	Validation Loss	Accuracy
1	No log	1.968679	0.172893
2	No log	1.666397	0.361425
3	No log	1.247379	0.589922
4	No log	1.134480	0.635969
5	No log	1.049189	0.650738
6	1.361600	0.992559	0.674196
7	1.361600	1.014308	0.667246
8	1.361600	0.998116	0.683753

وقتی تا ۱۵ اپک هم پیش رفتیم به دقت ۶۹ درصد هم رسیدیم.

۴ اقدامات انجام شده

در مقاله ای که در داک گذاشته شده است ابتدا مرحله preprocess مشابه ما انجام داده است و بعد بر روی یکسری مدل تست کرده است و معیارهای ارزیابی آن f1,recall,precision است و بهترین عملکرد را مدل XLM_Roberta_large گرفته است.

Model	Precision (Macro)	Recall (Macro)	F1 (Macro)
FastText [42]	54.82	46.37	47.24
HAN [43]	49.56	44.12	45.10
RCNN [44]	50.53	48.11	47.95
RCNNVariant	51.96	48.96	49.17
TextAttBiRNN [45, 46]	54.66	46.26	47.09
TextBiRNN	51.45	47.16	47.14
TextCNN [47]	58.66	51.09	51.47
TextRNN [48]	49.39	47.20	46.79
ParsBERT	67.10	65.56	65.74
XLM-Roberta-base	72.26	68.43	69.21
XLM-Roberta-large	**75.91**	**75.84**	**75.39**
XLM-EMO-t	70.05	68.08	68.57

و بعد تعدادی نمونه که اشتباهی پیش بینی کرده اند را نمایش داده است.

کار های انجام شده:

۱.مقاله خوندیم که لینک های آن در مراجع هست.

۲.دیتاهای تست و train را preprocess کردیم.

۳.رو مدل های مختلف تست کردیم و بهترین مدل را انتخاب کردیم.

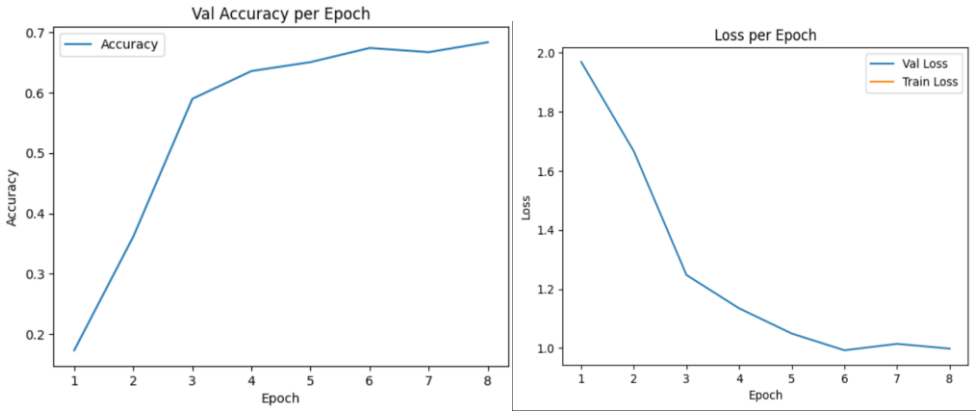
۴.بهترین مدل را save کردیم.

۵.معیار های accuracy,recall,precision,f1 را تست کردیم.

یکی از چالش هایی که با آن مواجه بودیم تنظیم هایپرپارامترها بود که بر روی دقت مدل بسیار تاثیر گذار بود مخصوصا لرنینگ ریت که با آزمون و خطاهای بسیار به اعداد مناسب برای هایپرپارامترها رسیدیم.

۵ ارزیابی مدل

همانطور که در بالاتر گفته شد ما از مدل **persian_xlm_roberta_large** استفاده کردیم که بر روی معیارهای مختلف از جمله accuracy,f1,recall,precision تست شد که نتایج و نمودار ها به صورت زیر است:



و نتایج بر روی معیارهای ارزیابی نیز به صورت زیر شد:

```
Precision:  0.732781028240936
Recall:  0.7115551694178974
F1 score:  0.7094612321075684
```

و همچنین confusion matrix نیز به صورت زیر شد:

حالا به مثال هایی که در تابع predict به عنوان ورودی دادیم می پردازیم:

```
sentence: امروز خیلی روز غم انگیزی بود امیدوارم هیچ وقت تکرار نشه
result: [{'label': 'SAD', 'score': 0.9440757632255554}, {'label': 'HATE'
class : SAD

sentence: چرا همچین اشتباهی کردی من ازت متنفرم
result: [{'label': 'SAD', 'score': 0.004411958623677492}, {'label': 'HAT
class : HATE

sentence: چه جمله خنده داری
result: [{'label': 'SAD', 'score': 0.016603756695985794}, {'label': 'HAT
class : HAPPY
```

به عنوان مثال سه جمله بالا را به عنوان ورودی دادیم و کلاس هایی که پیش بینی کرده نیز درست است.حالا به مثال هایی میپردازیم که اشتباه پیش بینی کرده است:

"با آرزوی موفقیت و پیروزی ایران جام جهانی بورس سهام ده درصد"

HAPPY :True

OTHER: Pred

" طوفان یه جوری منشن میده داشتم اشک میریختم حالا خنده‌ام قطع نمیشه . خدا خیرت بده "

HAPPY:True

SAD:Pred

" این راننده خطی انقدر باشعوره که صدای آهنگ رو کم میکنه که بقیه اذیت نشن . آفرین مرد "

HAPPY:True

ANGRY:Pred

"در صف اول انتقاد از نتیجه‌گرایی قرار دارم اما خودم خدای نتیجه گرایی‌ام یعنی چی که " که چی " مرگ " که چی " گوساله درد " که چی " مثل بقیه زندگیتو کن"

ANGRY: True

OTHER :Pred

"فن مترو خراب‌بود . ملت داشتن غر میزدن . آقای میانسالی گفت : وقتی تو مدرسه به بچهها تجاوز میکنن توقع دارین مترو همهچیزش سالم باشه ! مترو "

SURPRISE:True

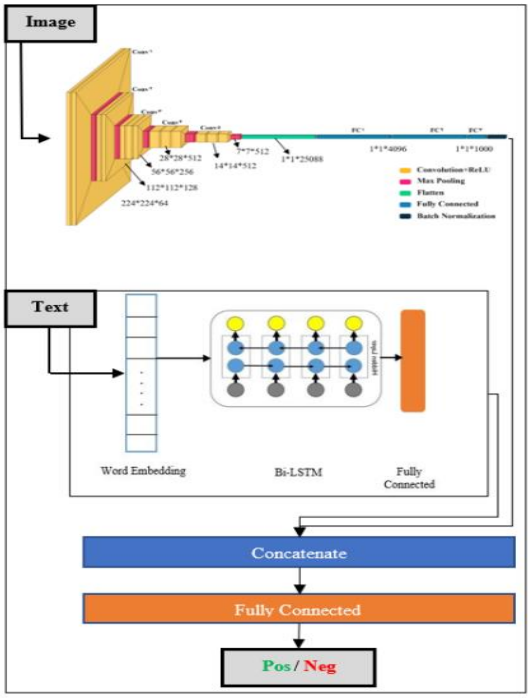
SAD:Pred

همانطور که مشاهده میکنیم سه جمله بالا که اشتباه پیش بینی شده واقعا جملات لبه مرزی هستند یعنی خیلی دقیق نمیشه گفت که به چه کلاسی تعلق دارند.

۶ بخش امتیازی

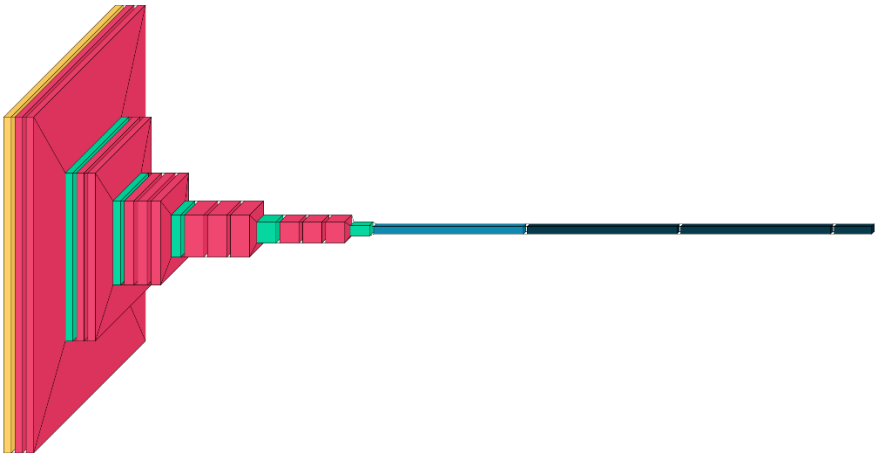
ما در این بخش علاوه بر متن باید دیتاستی پیدا می کردیم که شامل عکس و متن هر دو با هم باشد بعد از سرچ کردن مقاله ای را پیدا کردیم که Sentiment Analysis of Persian Instagram Post: a Multimodal Deep Learning Approach از پست

های اینستاگرام استفاده میکرد که شامل متن و عکس هست در واقع شامل یک فایل اکسل و یک فایل rar که شامل تصاویر بود تعداد کل داده ها هم 1000 بود که لیبل های ان positive,negative بودند مدلی که استفاده شده بود به صورت زیر است:

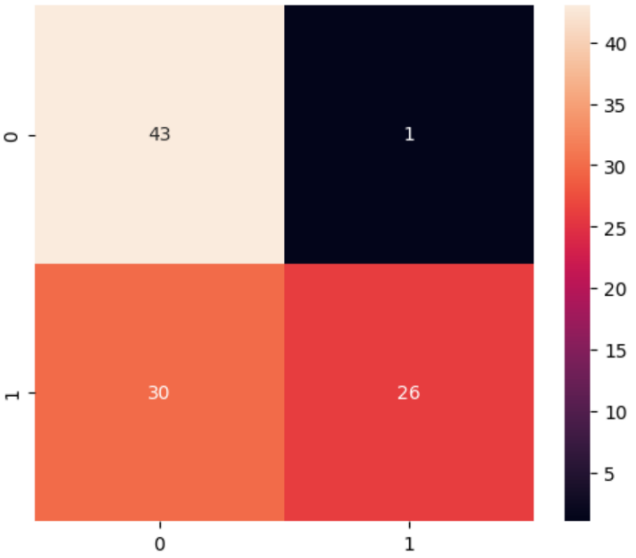


در واقع عکس را به یک شبکه CNN داده و Text را به یک LSTM و بعد این دو را concate کرده است.

در ابتدا عکس را به یک شبکه pretrain VGG16 دادیم که ساختار شبکه آن به صورت زیر است:



و text را نیز به یک شبکه *bi-lstm* دادیم و به accuracy ۶۹ درصد رسیدیم.



۷ مراجع

https://ieeexplore.ieee.org/document/9443026
https://arxiv.org/pdf/2207.11808.pdf
https://huggingface.co/docs/transformers/tasks/sequence_classification
https://github.com/Arman-Rayan-Sharif/arman-text-emotion