

به نام خدا

دوره کارآموزی هوش مصنوعی پردازش زبان طبیعی

بخش سوم پروژه - طبقه‌بندی با روش یادگیری عمیق
محدثه رهنما - زینب تقوی

آبان الی دی ۹۹

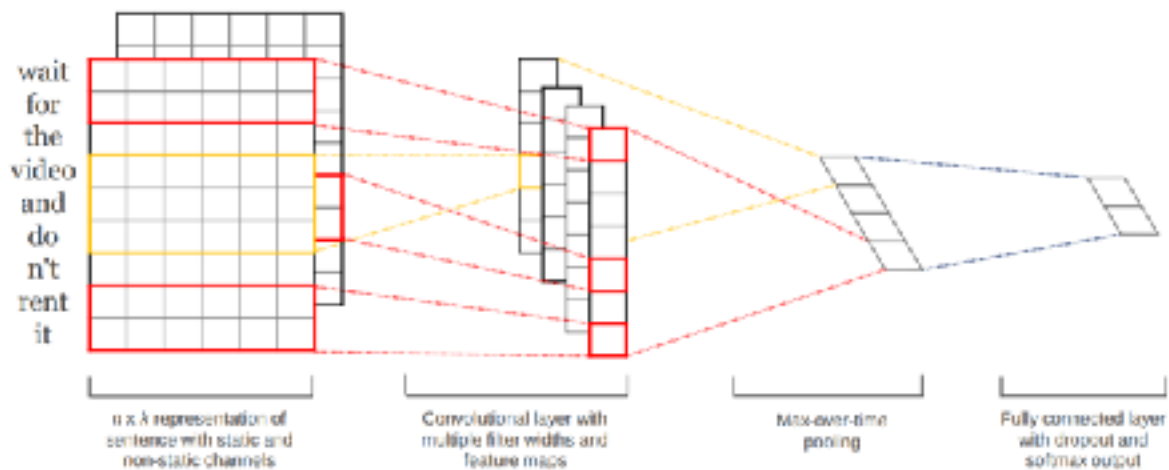
تغییرات در پیش‌پردازش

انواع معماری‌های یادگیری عمیق

در این فاز چند معماری مورد بررسی قرار گرفت. معماری این شبکه‌ها مبتنی بر شبکه‌های کانولوشن، شبکه‌های بازگشتی یا ترکیبی از هر دو نوع بودند. در ادامه درباره‌ی انواع شبکه‌های بررسی شده توضیح داده خواهد شد.

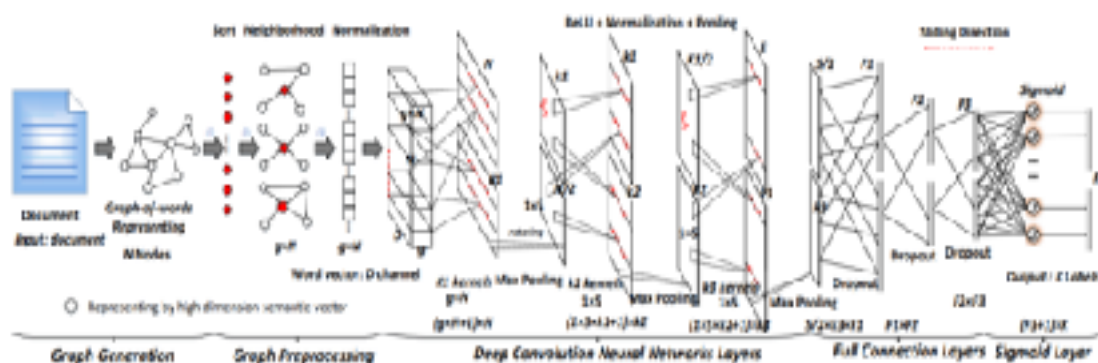
شبکه‌های مبتنی بر کانولوشن

۱. اولین شبکه‌ی مورد نظر بر اساس مقاله (Kim, 2014) پیاده‌سازی شده است:



برای بهبود نتایج، به این معماری چند لایه‌ی تماماً متصل نیز افزوده شده که تأثیر خوبی داشته است. از آن‌جا که این شبکه نتایج بهتری نسبت به سایر شبکه‌ها داشته، با کم و زیاد کردن لایه‌های کانولوشن و تماماً متصل تلاش شده تا تعداد لایه‌های مناسبی یافت شود.

۲. معماری دیگری نیز در (Peng, et al., 2018) ارائه شده که برای مسائلی که به صورت گراف بازنمایی می‌شوند مناسب است. صرف نظر از لایه‌های مربوط به برای بازنمایی گراف، لایه‌های کانولوشن و تماماً متصل تقریباً مشابه معماری قبلی است:



این شبکه نیز پیاده سازی شد اما موجب کرش کردن رم در کولب شد.

۳. شبکه‌ای با لایه‌های lstm و لایه‌های تماماً متصل مورد بررسی قرار گرفت که خلاصه‌ی پارامترهای آن در شکل زیر مشاهده می‌شود:

Layer (type)	Output Shape	Param #
SentenceInput (InputLayer)	[(None, 300)]	0
shared_embd (Embedding)	(None, 300, 300)	67643500
lstm_1 (LSTM)	(None, 300, 128)	219448
lstm_2 (LSTM)	(None, 64)	49408
drop_out_2 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 256)	16640
dense_2 (Dense)	(None, 78)	20046
out (Softmax)	(None, 78)	0
Total params: 67,909,242		
Trainable params: 305,742		
Non-trainable params: 67,603,500		
None		

۴. معماری دیگر (Kim, 2014) بررسی شد.

نکات و چالش‌ها

در تمام حالات از الگوریتم word2vec برای word embedding استفاده شد و برای این کار کتابخانه‌ی genism به کار گرفته شده است و از داده‌های موجود برای ساخت این مدل استفاده شده است. همچنین از کتابخانه‌ی keras برای پیاده سازی شبکه‌ها استفاده شده است.

یکی از مواردی که باید در هنگام بازیابی داده‌ها از فایل csv توجه داشت این است که هر سطر از این فایل حاوی لیستی از توکن‌هاست. اگر فایل با کتابخانه‌ی pandas و با تابع read_csv خوانده شود، محتوای هر سطر به صورت رشته خوانده می‌شود. برای تبدیل محتوای یک ستون به لیست، می‌توان پارامتر converters={'column_name': eval} را به تابع read_csv اضافه کرد.

برای تعیین epoch های لازم در اجرای شبکه ی عمیق، می‌توان از توابع call back در keras استفاده کرد. به این صورت که با تابع EarlyStopping می‌توان تعیین کرد تا اگر پس از n تا epoch تغییری در مقدار loss دیده نشد، یادگیری متوقف شود. همچنین با تابع ModelCheckpoint می‌توان در پایان هر epoch مدل به دست آمده را ذخیره نمود.

برای تعیین metrics در مدل، استفاده ی همزمان های accuracy و binary_accuracy و categorical_accuracy پیشنهاد شده بود. پس از آموزش مدل ها و مشاهده ی نتایج در هر epoch مشخص شد که binary_accuracy بسیار بالا و accuracy و categorical_accuracy بسیار پایین است. علت بالا بودن دقت باینری این است که به دلیل ذات مسئله‌ی چند کلاسه، احتمال دارد که نمونه‌ای در چند کلاس قرار بگیرد. در محاسبه ی دقت باینری، هر کلاس با کلاس دیگری در نظر گرفته می‌شود. برای مثال فرض کنیم در کل ۴ کلاس متفاوت داشته باشیم و برچسب واقعی نمونه‌ای برابر با رشته

ی [1,0,0,0] باشد (یعنی نمونه برچسب کلاس ۱ را دارد). حال اگر برچسب پیش بینی شده برابر با [0,0,0,0] باشد، در محاسبه ی دقت باینری عددی برابر با ۷۵ درصد به دست می آید. در حالی که برچسب تخمینی به وضوح نادرست است. به همین دلیل این معیار کنار گذاشته شد.

نتایج

در ادامه یافته های به دست آمده از معماری های اول، سوم و چهارم خلاصه شده است. به دلیل این که معماری اول – شبکه کانولوشن و لایه های تماماً متصل – در مقایسه با شبکه های دیگر نتایج بسیار بهتری داشتند، در این شبکه با کم کردن یا اضافه کردن لایه ها تأثیر این مورد بررسی شد.

نتایج حاصل در معماری های ۱ و ۳:

در اجرای این شبکه ها ترکیب معیارهای دقت، دقت باینری و دقت categorical هر کدام در محاسبه ی نتایج آموزش شبکه مورد استفاده قرار گرفتند.

Models	F1-score	metric		
		Accuracy	Accuracy	Accuracy
		Binary_accuracy	Categorical_accuracy	
CNN-Base	micro avg	0.54	0.42	0.55
	macro avg	0.50	0.40	0.53
	weighted avg	0.52	0.43	0.56
	samples avg	0.48	0.45	0.56
LSTM-Base	micro avg	0.06	0.07	0.07
	macro avg	0.05	0.02	0.03
	weighted avg	0.06	0.03	0.04
	samples avg	0.06	0.06	0.06

با توجه به یافته‌های حاصل، واضح است که معماری ۱ یعنی شبکه کانولوشن و لایه‌های تماماً متصل نتایج بهتری داشته است. همچنین معماری ۴ یعنی افزودن لایه‌های bilstm به معماری ۱ نیز تأثیر چندانی نداشته و در این مورد به نظر می‌رسد شبکه‌های بازگشتی کمکی به بهبود نتایج نکرده‌اند.

معماری شبکه‌ی ۱، همانطور که بیان شد از یک لایه کانولوشن به همراه max pooling تشکیل شده که در نهایت مقادیر لایه‌ی pooling به لایه‌ی آخر که تماماً متصل بوده هدایت می‌شوند. در این پروژه مشخص شد که با افزودن لایه‌های بیشتر نتایج بهتری حاصل می‌شود و در نهایت اضافه کردن ۴ لایه‌ی تماماً متصل مناسب به نظر رسید. در لایه‌های تماماً متصل میانی از تابع فعال کننده‌ی relu استفاده شد و همچنین برای تعیین برچسب‌ها در لایه‌ی آخر از تابع فعال کننده‌ی sigmoid استفاده شد تا احتمال تعلق داشتن به هر کلاس با عددی بین ۰ تا ۱ به دست آید. در ادامه نتایج حاصل از تابع فعال کننده softmax در آخرین لایه با تابع sigmoid مقایسه شده است:

Models	F1-score	Accuracy	Accuracy	Accuracy
		Binary_accuracy	Categorical_accuracy	
CNN-Base	micro avg	0.54	0.42	0.55
	macro avg	0.50	0.40	0.53
	weighted avg	0.52	0.43	0.56
	samples avg	0.48	0.45	0.56
CNN-Base With soft-max Layer at end	micro avg	0.10	0.13	0.18
	macro avg	0.10	0.14	0.16
	weighted avg	0.12	0.16	0.20
	samples avg	0.09	0.12	0.17

همان طور که از نتایج به دست آمده مشخص است، تابع فعال کننده sigmoid مناسب تر است. لازم به ذکر است که در تعیین برچسب هر نمونه، لازم است تا با تعیین آستانه‌ای در احتمال تعلق به هر کلاس، نوع برچسب (ها) هر نمونه مشخص شود. برای نمونه اگر میزان احتمال از ۰.۵ بیشتر باشد به این معنا است که نمونه به کلاس مورد نظر تعلق دارد. نسبت به هر مدل آموزشی، میزان آستانه متفاوت است. در نتایج به دست آمده تلاش شده تا مقدار آستانه مناسب به دست آید.

در ادامه نتایج به دست آمده از کم و زیاد کردن لایه‌ها خلاصه شده است:

نتایج حاصل از یک شبکه کانولوشن و ۴ لایه تماماً متصل (بهترین نتایج در این فاز):

Models		F1-score
CNN-Base	micro avg	0.55
	macro avg	0.53
	weighted avg	0.56
	samples avg	0.56

تأثیر لایه‌های تماماً متصل

نتایج حاصل از یک شبکه کانولوشن و ۳ لایه تماماً متصل:

Models		F1-score
CNN-Base +3 Dense	micro avg	0.41
	macro avg	0.38
	weighted avg	0.40
	samples avg	0.42

نتایج حاصل از یک شبکه کانولوشن و ۵ لایه تماماً متصل

Models		F1-score
CNN-Base +5 Dense	micro avg	0.41
	macro avg	0.38
	weighted avg	0.40
	samples avg	0.42

تأثیر تعداد شبکه‌ی کانولوشن

نتایج حاصل از بدون شبکه کانولوشن و ۴ لایه تماماً متصل (MLP):

Models		F1-score
No-CNN	micro avg	0.30
	macro avg	0.26
	weighted avg	0.29
	samples avg	0.28

نتایج حاصل از دو شبکه کانولوشن و ۴ لایه تماماً متصل:

Models		F1-score
CNN-Base: 2 CNNs	micro avg	0.30
	macro avg	0.30
	weighted avg	0.32
	samples avg	0.34

نتایج حاصل از سه شبکه کانولوشن و ۴ لایه تماماً متصل:

Models		F1-score
CNN-Base: 3 CNNs	micro avg	0.18
	macro avg	0.17
	weighted avg	0.19
	samples avg	0.19

همان طور که پیشتر اشاره شد بهترین نتایج در یک شبکه کانولوشن و ۴ لایه تماماً متصل به دست آمد. در ادامه مقادیر سایر معیارهای ارزیابی ارائه شده است.

- Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification.
- Peng, H., Li, J., He, Y., Liu, Y., Mengjiao, B., Wang, L., . . . Yang, Q. (2018). Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN.