

به نام خدا
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

یادگیری ماشین

تکلیف هشتم

استاد درس: دکتر صفابخش

امیرحسین کاشانی

۴۰۰۱۳۱۰۷۱

amkkashani@gmail.com

نیم سال اول ۱۴۰۱-۱۴۰۲

فهرست

سوال ۱).....	۳
الف).....	۳
ب).....	۴
سوال ۲).....	۴
ج).....	۴
د).....	۵
سوال ۳).....	۸
ه).....	۸

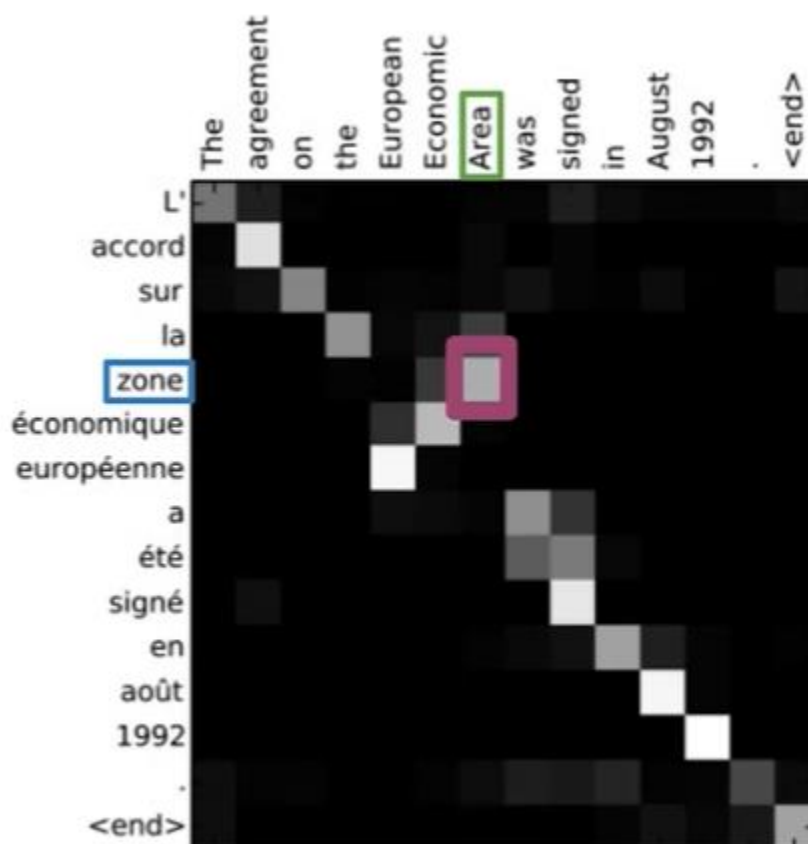
سوال ۱)

(الف)

مکانیسم توجه در ترانسفورماتور یک جزء کلیدی است که مدل را قادر می‌سازد تا وابستگی هایی که در دراز مدت در داده‌ها وجود دارد را درک و توالی‌های ورودی را به صورت موازی پردازش کند. مکانیسم توجه به مدل این امکان را می‌دهد که در زمان‌های مختلف هنگام پیش‌بینی، روی بخش‌های مختلف توالی ورودی تمرکز کند.

مکانیسم توجه، با توجه به یک بردار **query**، یک بردار **key** و یک بردار **value**، مجموع وزنی بردارهای مقدار را محاسبه می‌کند، جایی که وزن‌ها بر اساس شباهت بین بردارهای **query** و **key** محاسبه می‌شوند. سپس از این امتیاز توجه برای وزن دادن سهم هر بردار مقدار در خروجی نهایی استفاده می‌شود.

برای مثال تصویر زیر رابطه **query**, **key** در ترجمه ماشینی فرانسوی به انگلیسی می‌باشد. خانه‌هایی که سفید شده اند به معنای تاثیر بیشتر و خانه‌هایی که سیاه تر هستند به معنی تاثیر کمتر می‌باشند. (این تصویر زیر نمونه ای از **cross attention** می‌باشد)



در مدل ترانسفورمر از **self-attention** بهره گرفته شده است به این معنی که **key** , **query** , **value** همگی از یک لایه ایجاد شده اند و مکانیزم توجه ما به جای تمرکز بر روی خروجی بر روی ورودی تمرکز دارد و میزان اهمیت مکان‌های مختلف دنباله ورودی را محاسبه می‌نماید.

(ب)

تفاوت اصلی میان self-attention و cross-attention در بردارهای ورودی آنها مشخص می‌شود. در self-attention بردارهای توجه در همان دنباله ایجاد می‌شوند. به عبارت دیگر، مکانیسم توجه امتیازات توجه بین عناصر را در یک دنباله محاسبه می‌کند. این نوع مکانیسم توجه اغلب برای سنجش اهمیت بخش‌های مختلف یک دنباله برای یک کار پیش‌بینی خاص استفاده می‌شود.

	Hi	how	are	you
Hi	98	27	10	12
how	27	89	31	67
are	10	31	91	54
you	12	67	54	92

یک نمونه از self-attention

از سوی دیگر، cross-attention به مکانیزم توجهی اشاره دارد که امتیاز توجه را بین عناصر از دو دنباله متفاوت محاسبه می‌کند. این نوع مکانیسم توجه اغلب زمانی استفاده می‌شود که بخواهیم رابطه یا وابستگی‌های بین دو دنباله جداگانه را محاسبه کنیم، مانند ترجمه ماشینی، جایی که مکانیسم توجه برای سنجش اهمیت کلمات در زبان مبدأ هنگام ترجمه به کار می‌رود.

(سوال ۲)

(ج)

BERT (Bidirectional Encoder Representations from Transformers)

BERT یک رویکرد پیش‌آموزشی برای شبکه‌های مبتنی بر ترانسفورماتور در پردازش زبان طبیعی (NLP) است. ایده اصلی پشت BERT این است که یک شبکه بزرگ ترانسفورماتور را بر روی مجموعه بزرگی از متن از قبل آموزش دهیم، به طوری که شبکه نمایش‌های غنی از زبان را یاد بگیرد. سپس این شبکه از پیش آموزش دیده را می‌توان روی یک مجموعه داده کوچک‌تر و مختص

کار تنظیم کرد تا طیف گسترده‌ای از وظایف NLP مانند تجزیه و تحلیل احساسات، شناسایی موجودیت نام‌گذاری شده و پاسخ‌گویی به سؤال را انجام دهد.

در رابطه با Bert می‌بایست به ۳ نوآوری آن اشاره کرد.

۱. استفاده از مدل bidirectional که امکان استنتاج معانی کلمه در جمله را با توجه به کلمات قبل و بعد هموار می‌کند

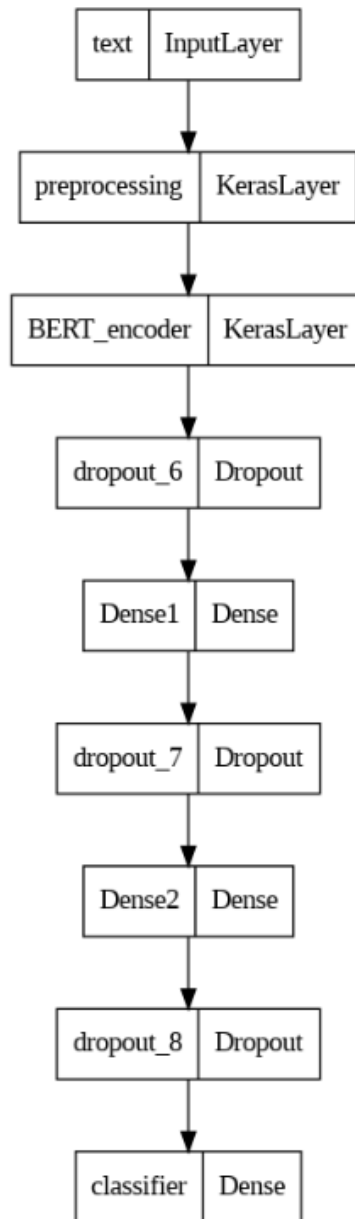
۲. مدل BERT برپایه مدل از پیش آموزش دیده شده (pre trained) از تسکی که به آن masked language modeling گفته می‌شود ایجاد شده است در این تسک می‌بایست مدل کلمات خالی رندوم که در جمله بوجود آمده است را پر کند.

۳. Bert، Next Sentence Prediction، با استفاده از وظیفه‌ای به نام پیش‌بینی جمله بعدی، از قبل آموزش داده شده است که در آن مدل باید احتمال وقوع دو جمله را بصورت متوالی پیش‌بینی کند. این کار به مدل کمک می‌کند تا رابطه بین جملات و معنای یک جمله را در زمینه یک سند بزرگتر بیاموزد.

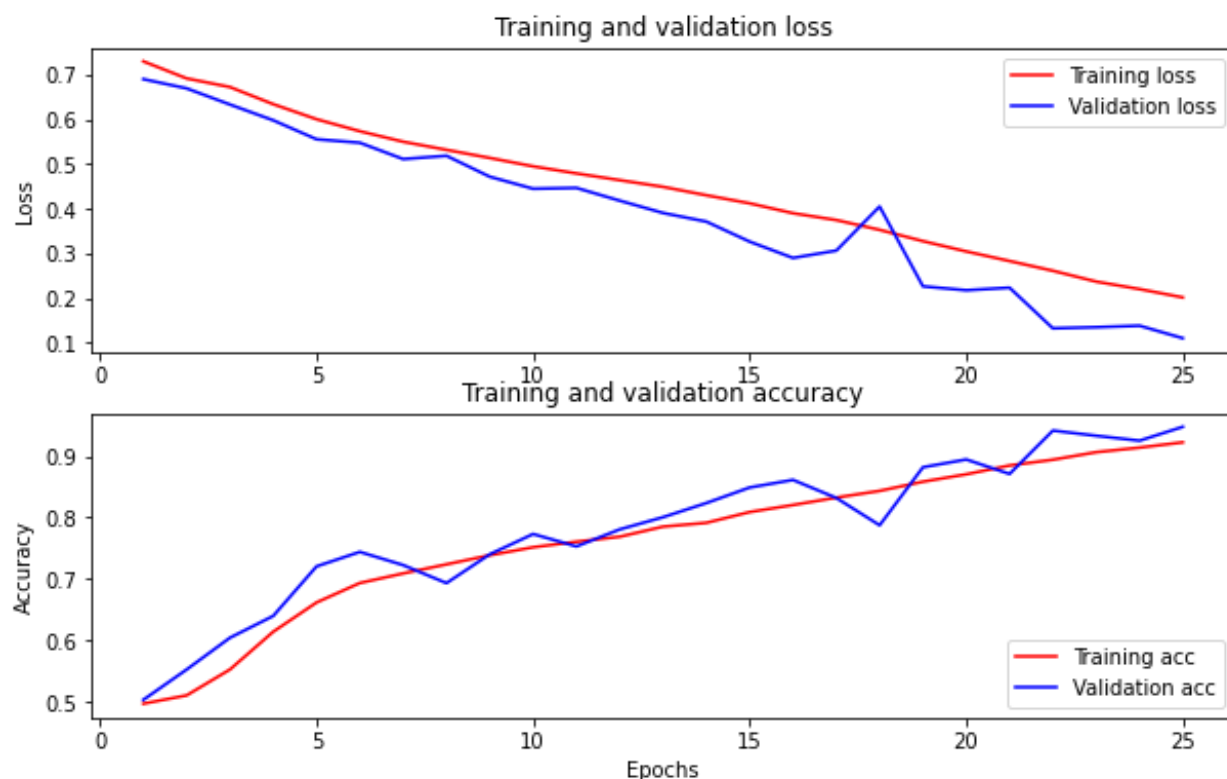
(د)

در این بخش از مدل small_bert/bert_en_uncased_L-۶_H-۵۱۲_A-۸ استفاده شده است و همچنین از bert_en_uncased_preprocess به عنوان بخش پیش پردازش استفاده شده است. برای بهبود عملیات دسته بندی چند لایه ی Dense به همراه Dropout به مدل اضافه گردیده است. همچنین برای کمک به بهبود عملکرد مدل پیش پردازش‌هایی که در تکلیف ۶ بر روی داده‌های اعمال شده است در این مساله نیز تکرار گردیده است با این تفاوت که با توجه به نیاز مدل ما به اینکه داده‌ها بصورت یک رشته باشند توکن‌های استخراج شده برای هر جمله دوباره به یک دیگر متصل شده اند.

نتایج و معماری استفاده شده را در ادامه مشاهده می‌کنید.

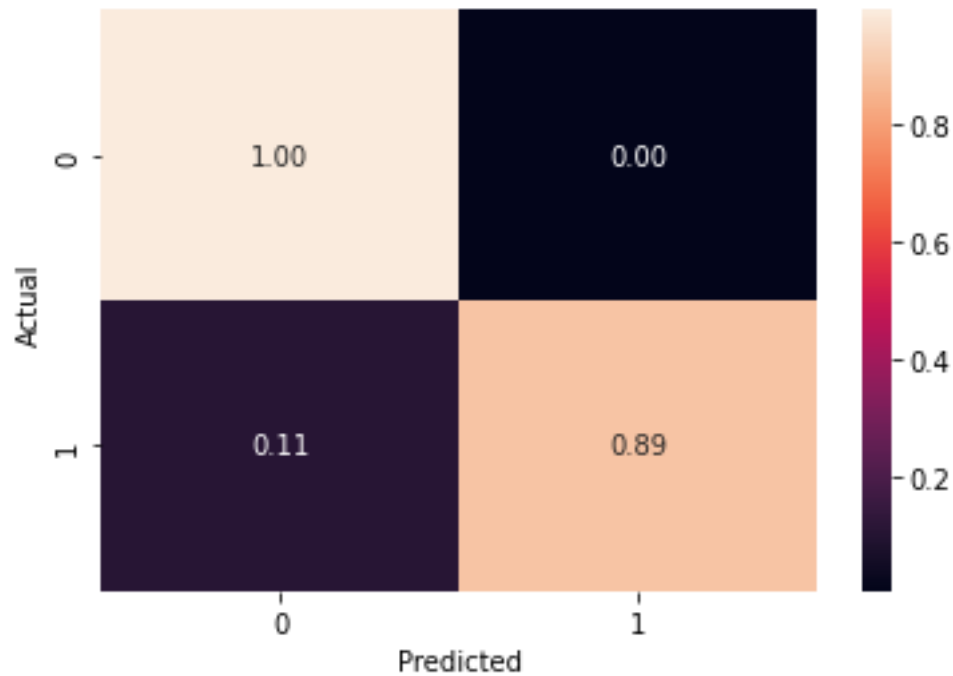


Loss: 0.11396937817335129
Accuracy: 0.9451733231544495



	precision	recall	f1-score	support
0	0.90	1.00	0.95	9355
1	1.00	0.89	0.94	9395
accuracy			0.95	18750
macro avg	0.95	0.95	0.95	18750
weighted avg	0.95	0.95	0.95	18750

accuracy model 94.51733333333333 %



سوال ۳)

۵)

دو مساله انتخاب شده در این بخش عبارت اند از ترجمه ماشینی و ایجاد متن که یک نمونه بسیار ساده از استفاده از این مدل‌ها در فایل ضمیمه قرار داده شده است.

لازم به ذکر است از آنجا که این تمرین به طور کامل در colab، توسعه داده شده است ممکن در صورتی که با Vscode یا PyCharm کدها را باز کنید مشکل در نمایش داشته باشند.