به نام خدا

دانشگاه تهران

دانشکده مهندسی

برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین پنجم

نام و نام خانوادگی	فاطمه نائینیان – محمد عبائیانی
شماره دانشجویی	810198432 - 810198479
تاریخ ارسال گزارش	1401-10-22

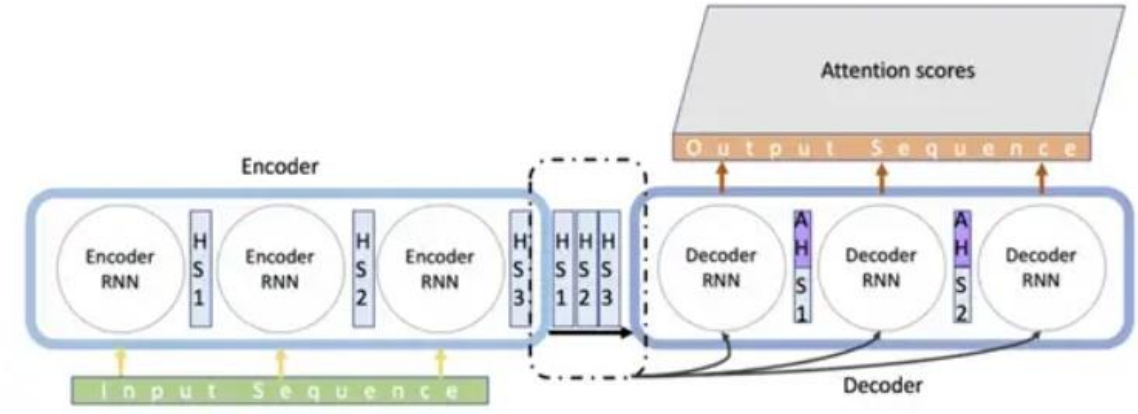
فهرست

پاسخ ۱ – آشنایی با مفهوم توجه و پیاده سازی مدلBERT

(1-1

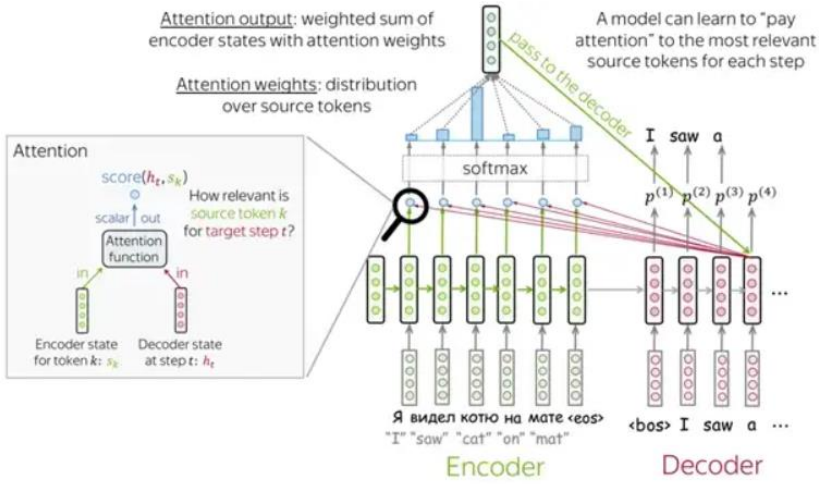
1) در روانشناسی توجه به معنی این است که در حالی که به یک یا چند موضوع دقت میکنید، بقیه موضوعات را نادیده بگیرید. این مفهوم در یادگیری عمیق به بر پایه encoder-decoder تعریف می شود. علت به وجود امدن این مدل این است که شبکه RNN هنگامی که مجموعه ورودی ها از حدی طولانی تر شوند دقت خوبی ندارد. همچنین در LSTM ها که حافظه طولانی تری داشتند دیدیم که این مدل میتواند با دقت بیشتری اطلاعات را در خود نگه دارد و دقت خوبی بگیرد. اما در این نوع پروژه هایی که نیاز به ترجمه داریم، طول ورودی ثابت نیست و متغیر هستند پس نیاز به مدل جدیدی داریم که در کنار ورودی گرفتن و خروجی دادن با طول متغیر، به صورت تدریجی اطلاعات را فراموش نکند. شبکه های encoder-decoder در این راستا کمک کنند هستند. این شبکه ها با بهره گرفتن از مفهوم توجه سعی میکند تا در هر بخش اطلاعات مهم را شناسایی کند و به انها وزن کمتری دهد بدین ترتیب میتواند با دقت و سرعت بیشتری به پیش بینی درست تری برسد.

Encoder ها شبکه هایی هستند که ویژگی های داده های ورودی را استخراج میکنند. به شکلی که sequence ورودی را میگیرد و آن را خلاصه میکند. Decoder ها از Encoder ورودی میگیرند و آن را تفسیر میکند.



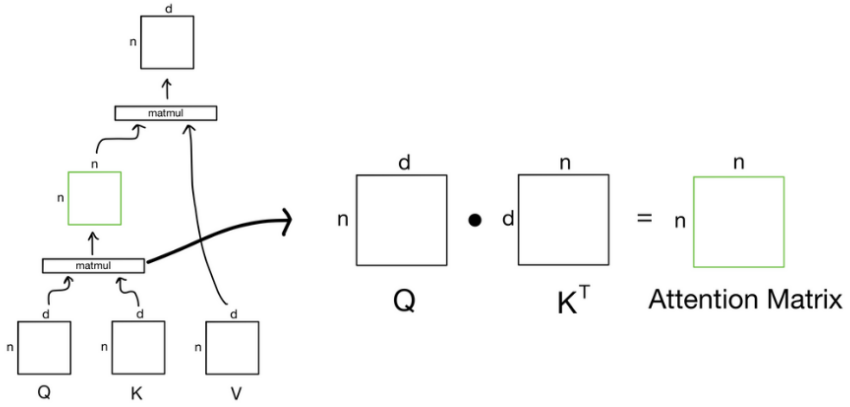
شکل 1 نحوه عملکرد مکانیزم توجه

مکانیزم توجه با فراهم کردن وزن هایی برای اطلاعات encoder سبب می شود تا decoder بتواند تشخیص دهد به کدام بخش ها توجه بیشتری بکند و پیش بینی را بر اساس ان توجه در هر بازه زمانی انجام دهد.



شکل 2 : عملکرد **encoder-decoder**

2) در یک Single-Head attention هر بار فقط یکی از حالات توجه بررسی می شود اما زمانی که از Multi-head attention استفاده میکنیم میتوانیم head هایی داشته باشیم که به صورت موازی از هم کار میکنند و این موضوع سبب می شود تا حالات مختلف توجه بررسی شود بدین ترتیب شبکه بسیار بهبود پیدا میکند و مدل میتواند بر موقعیت های مختلف توجه کند و تاثیر ان را ببیند.

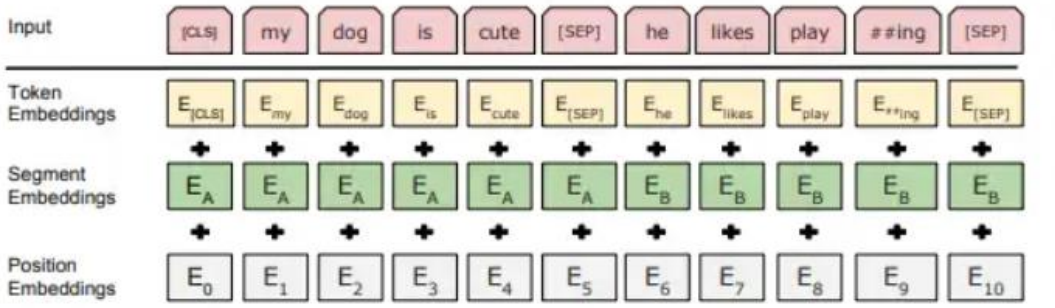


شکل 3: نحوه اعمال مکانیزم توجه

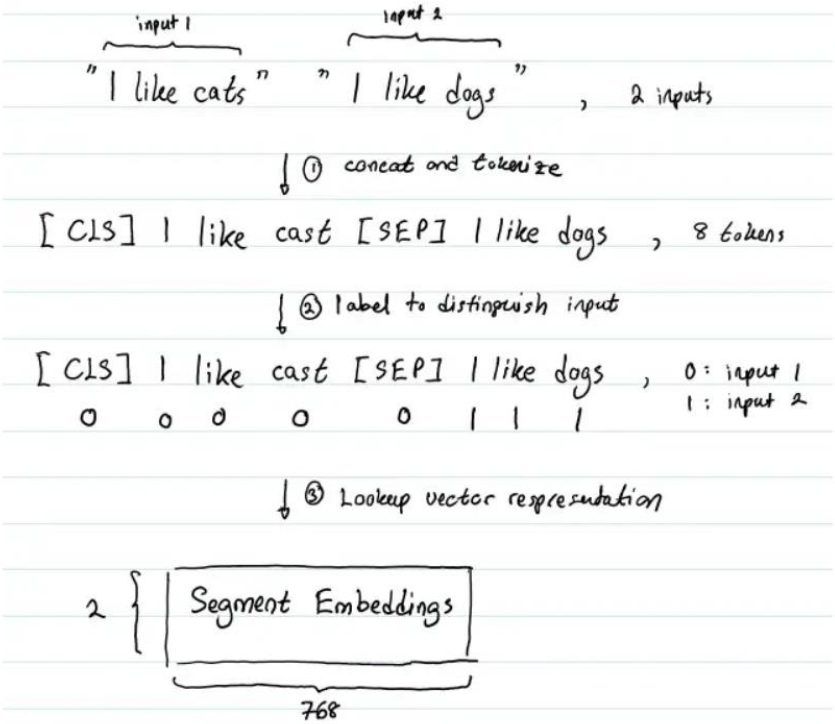
بنابراین Multi-head attention به شبکه عصبی این ویژگی را میدهد تا اطلاعات بین چند بخش ورودی را ترکیب کند و نمایش غنی تری از اطلاعات به دست اورد بدین ترتیب عملکرد شبکه نیز بهتر می شود.

(2-1

1) در segmentation embedding دو توکن خاص وجود دارد که CLS و SEP هستند. توکن CLS هنگامی استفاده می شود که یک ورودی وارد می شود و این توکن به ابتدای ان اضافه می شود. هنگام پایان هر جمله یا زیردنباله توکن SEP به آن اضافه می شود که در واقع مرز بین جملات را نشان میدهد. در کل BERT سه مکانیزم Token و positional و segmentation می باشد. ترکیب این سه مکانیزم سبب می شود تا مدل بتواند آسان تر مکانیزم توجه را پیاده سازی کند و بهتر ورودی بگیرد و پیش بینی ها دقیق تر شود.



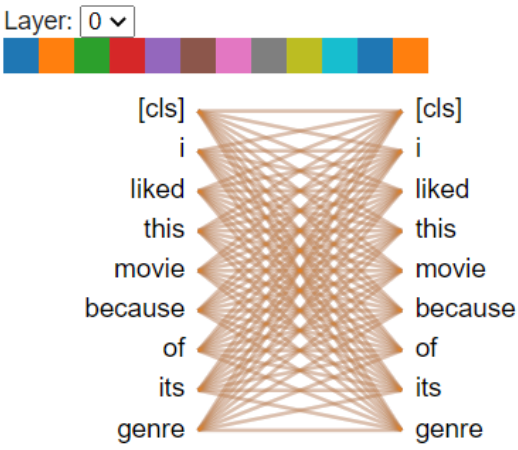
شکل 4: نحوه عملکرد **Embedding**



شکل 5: یک مثال از عملکرد **Embedding**

2) نتایج زیر را برای جمله "I liked this movie because of its genre" داریم. میبینیم مدل توانسته خروجی خوبی بدهد و دقت ان برای این جمله 75٪ است. این موضوع تایید دیگری بر اهمیت و کاربرد این نوع شبکه است.

```
sentence = "I liked this movie because of its genre"
toks, atts = get_att_tok(model, sentence.lower())
call_html()
head_view(atts, toks, layer=0)
```




```
model(np.array([encode_sentence(sentence.lower(), MAXLEN)], dtype=np.int64))
```

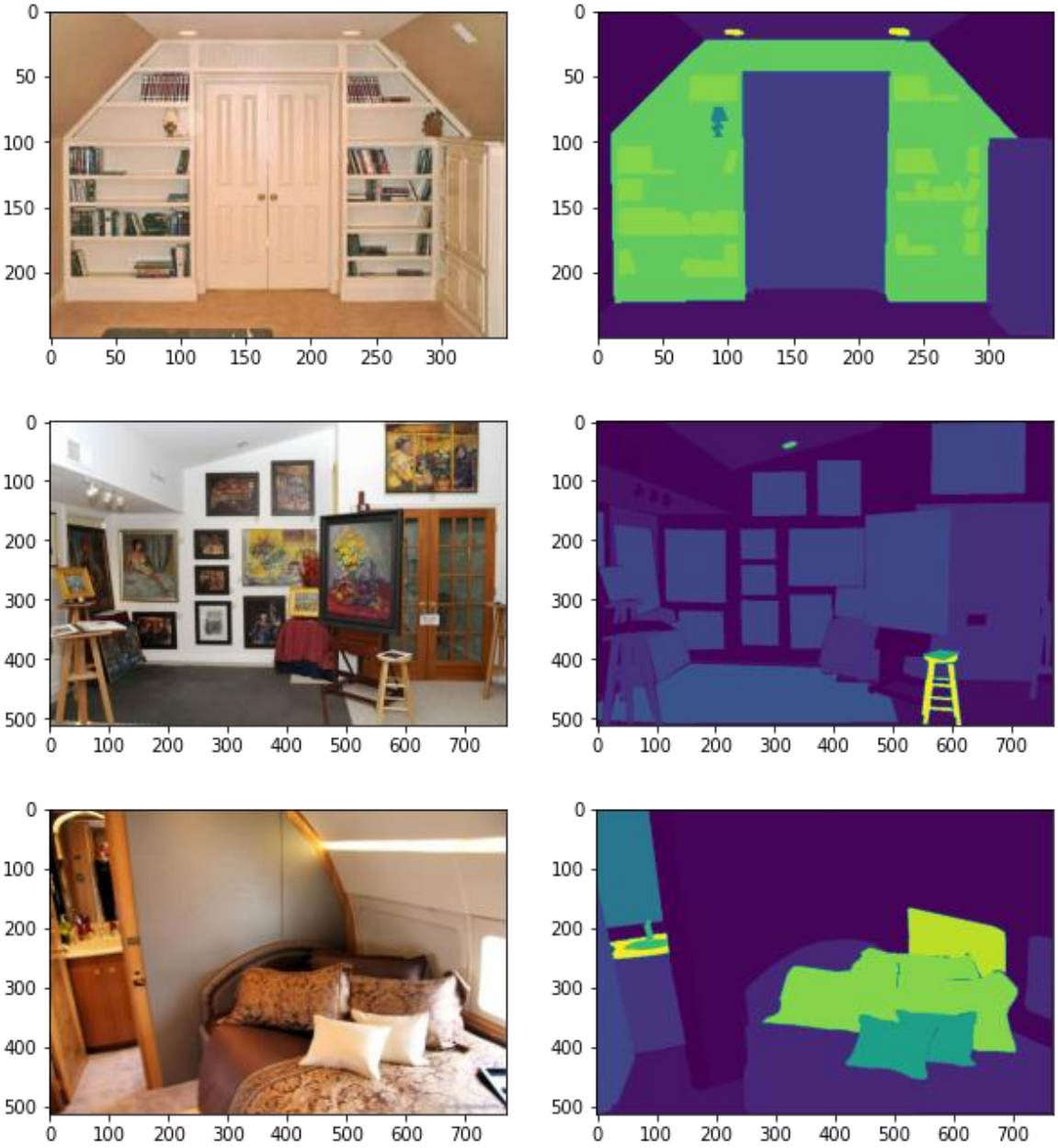
```
<tf.Tensor: shape=(1, 1), dtype=float32, numpy=array([[0.7530225]], dtype=float32)>
```

شکل 6: خروجی مدل **Bert** برای جمله دلخواه

پاسخ ۲ – آشنایی با کاربرد تبدیل کننده ها در تصویر

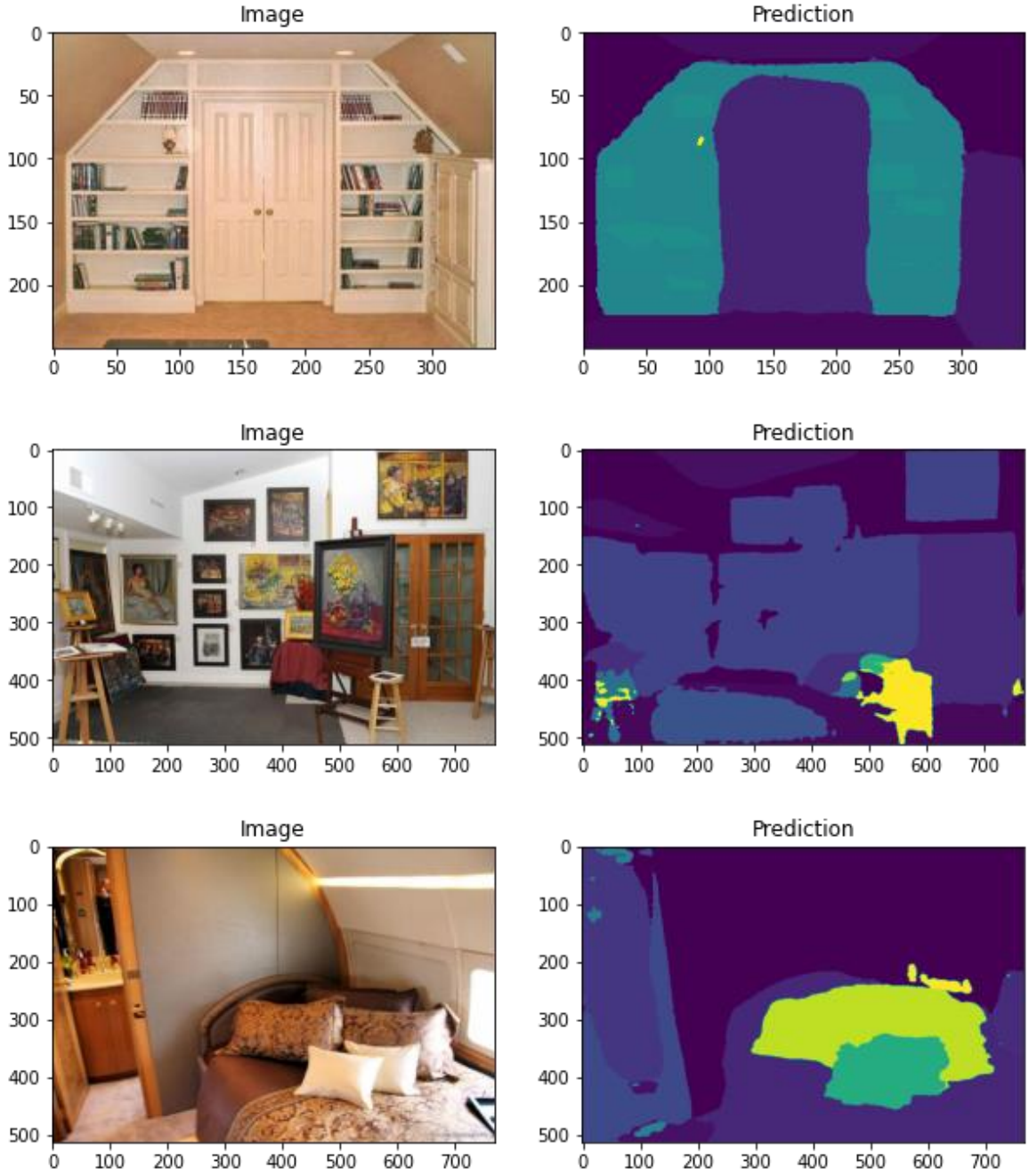
2-2)

تصاویر اصلی که در دیتا ست وجود دارد به همراه خروجی آنها که همان label انها می باشد.



شکل 7: برچسب های تصاویر دیتاست

حال خروجی انها که از مدل beit به دست آمده را مشاهده میکنیم.



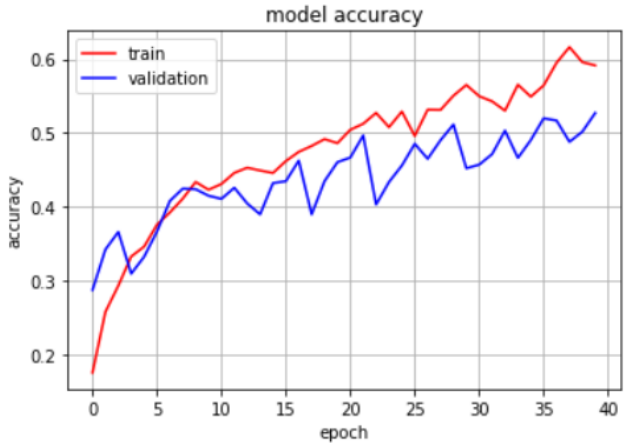
شکل 8: خروجی **Beit Segmentation**

میبینیم مدل beit با دقت خیلی خوبی توانسته segmentation را انجام دهد. این موضوع نشان دهنده اهمیت و کارایی این مدل است.

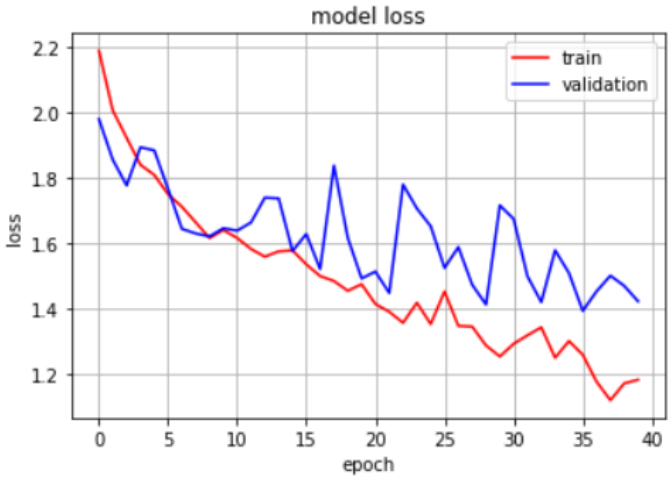
3-2)

برای مدل MLP لایه هایی با تعداد نورون 3072 و 4096 و 2048 و 1024 و 512 و 256 و 64 و 10 در نظر میگیریم و سپس آن را آموزش میدهیم.

نتایج آن به شکل زیر می شود.



شکل 9: نمودار دقت مدل **MLP** بر حسب **epoch**

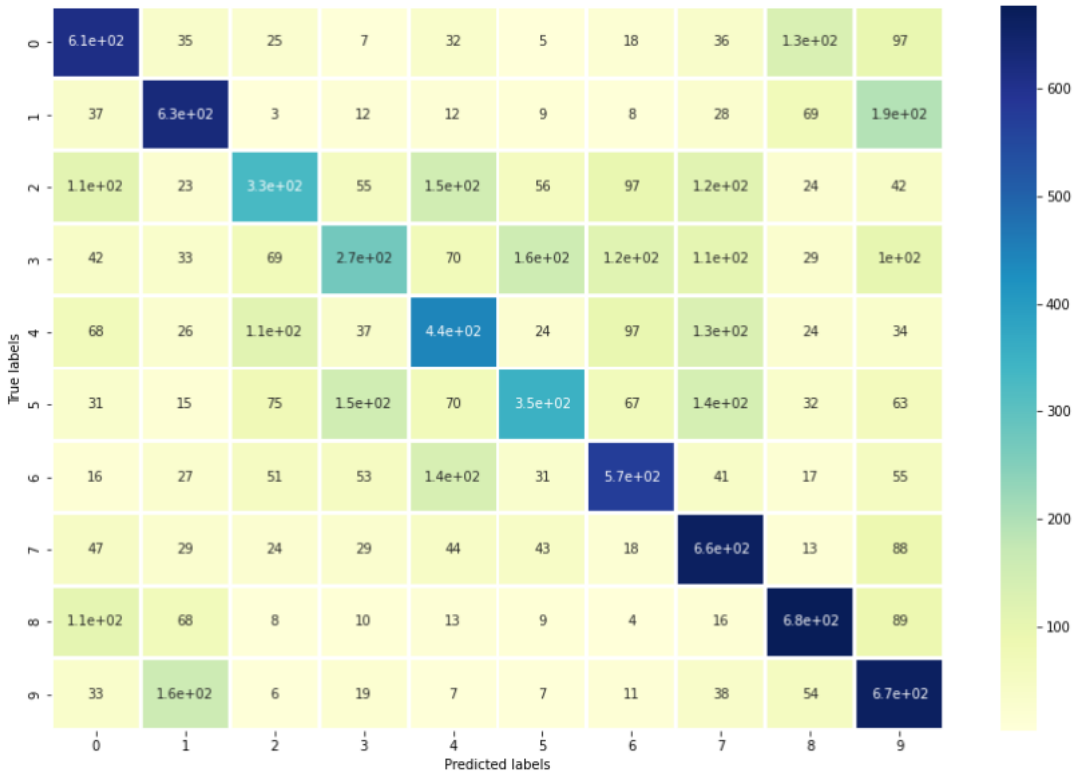


شکل 10: نمودار **loss** مدل **MLP** بر حسب **epoch**

```
Accuracy: 0.522300
Precision: 0.518176
Recall: 0.522300
F1 score: 0.512910
```

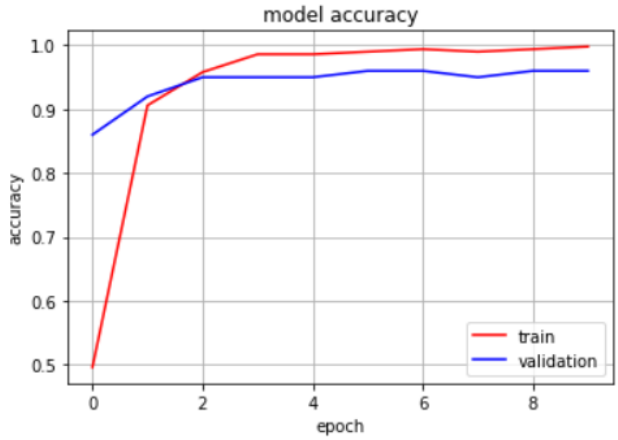
	precision	recall	f1-score	support
airplane	0.56	0.61	0.58	1000
automobile	0.60	0.63	0.61	1000
bird	0.47	0.33	0.39	1000
cat	0.42	0.27	0.33	1000
deer	0.45	0.44	0.45	1000
dog	0.51	0.35	0.42	1000
frog	0.57	0.57	0.57	1000
horse	0.50	0.67	0.57	1000
ship	0.63	0.68	0.65	1000
truck	0.47	0.67	0.55	1000
accuracy			0.52	10000
macro avg	0.52	0.52	0.51	10000
weighted avg	0.52	0.52	0.51	10000

شکل 11: **calssification report** برای مدل **MLP**

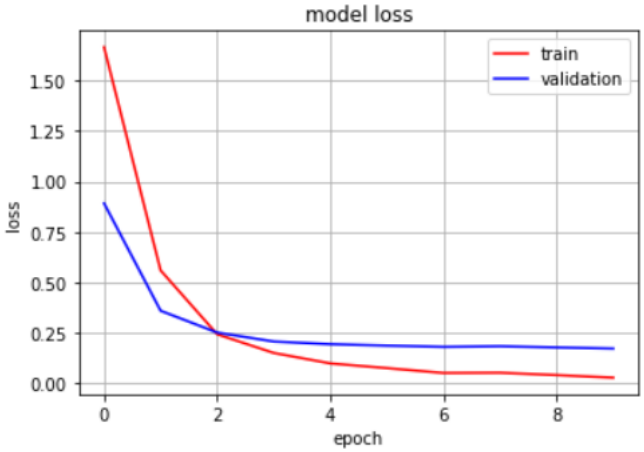


شکل 12: ماتریس آشفتگی برای مدل **MLP**

حال مدل دیگری میزنیم که شامل Beit شود. نتایج آن به صورت زیر می شود.



شکل 13: نمودار دقت برای مدل **Beit**



شکل 14: نمودار **loss** برای مدل **Beit**

شبکه Beit قدرت بیشتری نسبت به MLP دارد اما زمان طولانی تری را نیاز دارد تا وزن ها را اپدیت کند. این دو مدل در واقع قابل مقایسه نیستند زیرا کارایی متفاوتی دارند.

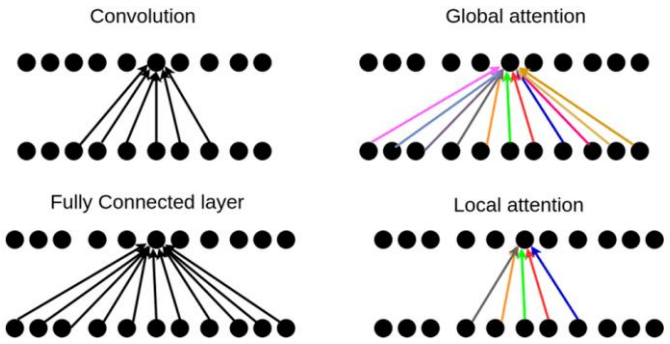
(4-2

پرسش ها

1. در شبکه های کانوولوشنی فیلتر یکسانی بر روی کل ورودی ها انجام می شود و بدون توجه به اطلاعات درون داده ها ، همه را از یک فیلتر عبور میدهد و وزن را آپدیت میکند. این مفهوم مشابه مفهوم توجه است. شبکه های کانوولوشنی میتواند مفهوم توجه را پیاده سازی کند اما برای اینکار به کانال ها و لایه های زیادی احتیاج دارد تا بتواند وزن های بخشی از اطلاعات را پایین بیاورد و

ان ها را نادیده بگیرد. تفاوت کانوولوشن و توجه در این است که کانوولوشن فیلتر یکسانی به همه داده ها اعمال می شود و در توجه این وزن ها به ازای هر بخش ورودی متفاوت است.

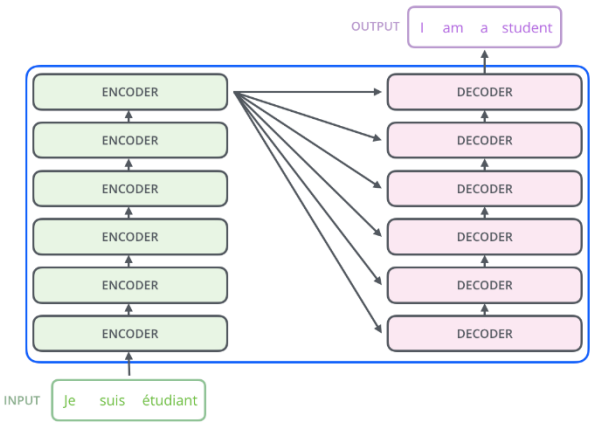
2. همانطور که در درس دیدیم شبکه های fully connected هر نورون از همه نورون های لایه قبل ورودی میپذیرد پس همه نورون ها بهم واسته هستند این یعنی این شبکه از نوعی توجه همگانی استفاده میکند. اما در CNN هر نورون به واسطه عمل کانوولوشن به همه نورون های قبلی وابسته نیست و فقط از تعدادی از آنها ورودی میپذیرد. این موضوع همانند توجه محلی است. البته باید دقت کرد که در توجه محلی و همگانی وزن ها ثابت نیستند و با توجه ورودی داده شده میتواند تغییر کند اما در CNN و fully connected این وزن ها بعد از آموزش ثابت میماند .



شکل 15: توجه همگانی و محلی

درست یا نادرست

1. در بخشی از لایه های تبدیل کننده ی Vanilla از شبکه ی LSTM استفاده شده است. نادرست است زیرا transformer ها ساختار بازگشتی ندارند و نیازی به استفاده از شبکه بازگشتی LSTM نیست و استفاده از شبکه بازگشتی در transformer ها سبب می شود تا این شبکه ها به درستی کار نکنند و به جای استفاده از LSTM از attention استفاده می شود.
2. یک تبدیل کننده از چند بلوک رمزگذار و چند بلوک رمزگشا تشکیل شده است. درست است. زیرا در درس و در همین تمرین این موضوع را دیدیم و پیاده سازی کردیم. برای مثال داریم:



شکل 16: یک نمونه از عملکرد رمزگذار و رمزگشا

3. multi head attention از یک بخش توجه و چند لایه ی تمام متصل موازی تشکیل شده است. نادرست است زیرا در multi head attention از چند Scaled dot-product attention استفاده می شود که چندین head به صورت موازی attention مربوط به خودشان را دارند و درنهایت همه انها concat می شوند و به سایز ورودی اصلی تبدیل می شوند و یک نمای کلی از Attention را می سازند.
4. وجود Positional Encoding در ساختار یک تبدیل کننده حیاتی است و بدون آن شبکه از کار می افتد. درست است زیرا در transformer ها که لایه کانولوشن و بازگشتی ندارند، برای اینکه مدل ترتیب کلمات بتواند در نظر بگیرد باید یک موقعیت مکانی از کلمات، به دنبال اضافه کنیم. به همین دلیل Positional Encoding اهمیت خاصی پیدا میکند.