

یادگیری عمیق

تمرین ششم

نیکی نزاکتی

98522094

1.

برای محاسبه batch normalization میانگین و واریانس سطرها را پیدا کرده، از هر سطر میانگین را کم کرده و بر واریانس تقسیم میکنم:

```
for i in range(0,5):  
    layer_mean = np.mean(layer[i])  
    layer_std = np.std(layer[i])  
    batch_normalization = (layer[i] - layer_mean) / layer_std  
    output.append(batch_normalization)
```

خروجی:

```
[-0.87559 , -1.0484039 , -0.18433477, 0.39171126, 1.7166171 ],  
[-1.2120533 , 0.50998974, 1.5365924 , 0.11259517, -0.94712365],  
[-0.89006555, 0.46766162, 1.2973838 , -1.4180706 , 0.54309094],  
[-0.88133436, 1.8990659 , 0.01049212, -0.30427018, -0.72395325],  
[-1.3094535 , 0.8184084 , 1.3094535 , 0.12276126, -0.94116974],
```

برای محاسبه layer normalization این عملیات را بر روی هر ستون انجام میدهیم.

خروجی:

```
[-0.25839037, -0.8613012 , 1.8948627 , 0. , -0.7751711 ],  
[-1.9522605 , 0.36725685, 0.27061027, 0.8988129 , 0.41558012],  
[-1.3241694 , 1.3241694 , 0.41380295, -0.99312705, 0.5793241 ],  
[-0.17220359, 1.4063282 , -0.74621516, -1.3202267 , 0.8323167 ],  
[ 1.2455118 , -0.9291913 , 1.1960868 , -0.68206596, -0.83034116],
```

2.

BERT یک شبکه عمیق unsupervised برای language representation است که بازنمایی های متنی را از متن بدون ساختار می آموزد و چون با داده بدون لیبل آموزش میبیند، پس از Deep unsupervised representation learning استفاده می کند. BERT بازنمایی ها را بر اساس context ی که کلمات در آن ظاهر می شوند، یاد می گیرد. در نتیجه، BERT میتواند بازنمایی های معنایی غنی تری را بیاموزد که معانی متفاوتی از کلمات را بسته به context آنها دریافت میکند. در واقع BERT با نوع خاصی از self-supervised task بهینه شده است که به داده های manually annotated نیاز ندارد. یعنی، در طول

آموزش، درصدی از token های انتخاب شده به طور تصادفی از جمله input قبل از عبور از رمزگذار Transformer پوشانده میشوند. رمزگذار جمله ورودی را به یک سری از embedding vectors (یکی برای هر کلمه در جمله) نگاشت می کند. این بردارها از یک لایه softmax عبور می کنند که احتمالات را در کل واژگان محاسبه می کند تا محتمل ترین کلمات شانس بیشتری برای انتخاب داشته باشند. به عبارت دیگر، این یک وظیفه پر کردن جای خالی در جمله است که در آن BET قصد دارد سیگنال ورودی خراب را از context نیمه موجود بازسازی کند.

3.

(الف)

نتیجه آموزش مدل با استفاده از داده های آموزشی دارای برچسب :

loss: 0.2233 - accuracy: 0.9350

نتیجه ارزیابی بر روی داده های تست :

Accuracy on Test Set : 10.000000149011612

همانطور که مشاهده می شود به علت کم بودن داده ها، مدل بر روی داده های آموزش overfit شده و عملکرد خوبی روی داده های تست ندارد.

(ب)

نتیجه آموزش مدل با استفاده از داده های آموزشی بدون برچسب و سپس دارای برچسب :

loss: 0.3290 - accuracy: 0.8772

loss: 1.3069 - accuracy: 0.6250

نتیجه ارزیابی بر روی داده های تست :

16.06000065803528

مشاهده می شود که عملکرد بهبود یافته است.

(پ)

ضریب دسته بندی ۱ و زاویه ۵:

loss: 1.7362 - rotation_output_loss: 0.3411 - classification_output_loss: 0.0308 -
rotation_output_accuracy: 0.8699 - classification_output_accuracy: 0.9964 -
val_loss: 23.7016 - val_rotation_output_loss: 2.4674 -
val_classification_output_loss: 11.3644 - val_rotation_output_accuracy: 0.6775 -
val_classification_output_accuracy: 0.1000

در این حالت عملکرد آموزشی مدل در هر دو تسک خوب بوده اما مدل در ارزیابی بر تسک classification خوب عمل نکرده است.

ضریب دسته بندی ۱ و زاویه ۱:

loss: 0.2230 - rotation_output_loss: 0.1922 - classification_output_loss: 0.0308 -
rotation_output_accuracy: 0.9285 - classification_output_accuracy: 0.9964 -
val_loss: 11.8536 - val_rotation_output_loss: 2.1950 -
val_classification_output_loss: 9.6585 - val_rotation_output_accuracy: 0.6860 -
val_classification_output_accuracy: 0.1000

در این حالت عملکرد آموزشی مدل در هر دو تسک خوب بوده اما مدل در ارزیابی بر تسک classification خوب عمل نکرده است.

ضریب دسته بندی ۴ و زاویه ۱:

loss: 0.2256 - rotation_output_loss: 0.1567 - classification_output_loss: 0.0172 -
rotation_output_accuracy: 0.9427 - classification_output_accuracy: 0.9968 -
val_loss: 48.4179 - val_rotation_output_loss: 2.0522 -
val_classification_output_loss: 11.5914 - val_rotation_output_accuracy: 0.6673 -
val_classification_output_accuracy: 0.1010

در این حالت عملکرد آموزشی مدل در هر دو تسک خوب بوده و مدل در ارزیابی بر تسک classification بهتر عمل کرده است.

در کل انجام موازی این دو تسک باعث بهبود نتایج شده است.

4.

(الف)

بعد از ۵ trial ، مقادیر بهینه به صورت بدست آمدند:

Search: Running Trial #5		
Value	Best Value So Far	Hyperparameter
4	4	n_blocks
0.5	0.1	rate
256	256	n_nodes
0.00086799	0.00091908	learning_rate

(ب)

همانطور که مشاهده می شود هاپیر پارامترهای مناسب ۴ بلاک کانولوشنی، احتمال Dropout با مقدار 0.1، تعداد نوروں های لایه Dense ماقبل آخر 256، و learning rate با مقدار 0.0009 است. تعداد بلاک های کانولوشنی بیشتر باعث می شود که مدل ویژگی های با معناتری را یاد بگیرد، Dropout کمتر احتمال غیرفعال کردن نوروں ها را کاهش می دهد و یادگیری با نوروں های بیشتر افزایش می یابد، تعداد نوروں های لایه dense به اندازه ای زیاد شده اند که یادگیری بیشتر شود ولی نه به اندازه ای که مدل overfit شود، و learning rate هم کاهش یافته تا مدل explore کردن را کاهش دهد.

(پ)

نتیجه حاصل از آموزش بهترین مدل بدست آمده:

accuracy: 0.8805 - val_accuracy: 0.8161

نتایج حاصل از مقادیر precision و recall و f1-score به صورت زیر است:

Precision Score : 0.8161

Recall Score : 0.8161

F1 Score : 0.8161

با توجه به نتایج حاصل از متریک ها می بینیم که عملکرد مدل مناسب بوده است. دلیل اصلی استفاده از این متریک ها این است که مطمئن شویم مدل موارد تکراری را به درستی شناسایی کند و از مثبت کاذب جلوگیری کند. precision F1 و recall کلاس مثبت را متعادل می کند در حالی که accuracy به مشاهدات درست طبقه بندی شده مثبت و منفی نگاه می کند. پس در این دیتاست نیازی به این متریک ها نداریم و accuracy برای بررسی عملکرد مدل کافی است.

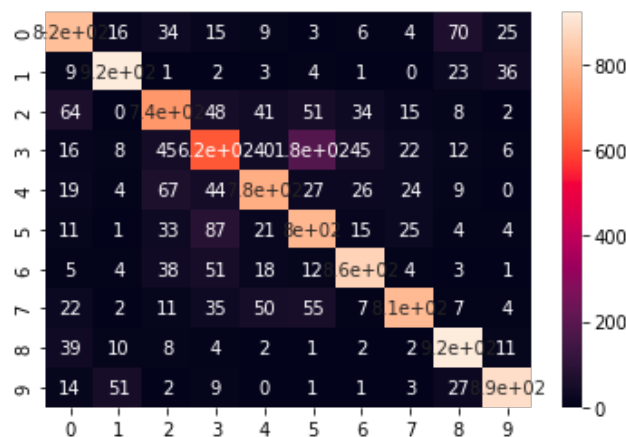
(پ)

TP : شما مثبت پیش بینی کردید و درست است.

TN : شما منفی پیش بینی کردید و درست است.

FP : شما مثبت پیش بینی کردید و نادرست است.

FN : شما منفی پیش بینی کردید و نادرست است.



مدل ما False positive های بیشتری دارد پس در این قسمت ضعیف عمل کرده است.

Resources:

<https://www.geeksforgeeks.org/>

<https://towardsdatascience.com/>