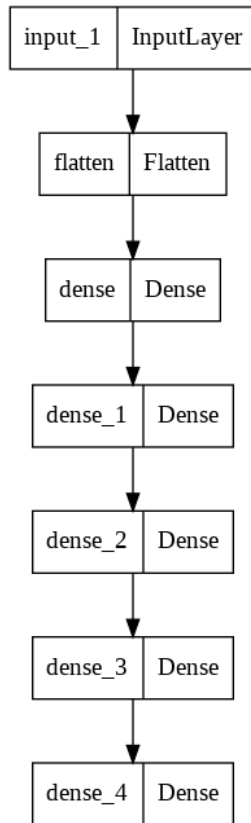


سوال دوم) زمان: ۳ ساعت

مدل fully connected:



Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
flatten (Flatten)	(None, 3072)	0
dense (Dense)	(None, 128)	393344
dense_1 (Dense)	(None, 128)	16512
dense_2 (Dense)	(None, 256)	33024
dense_3 (Dense)	(None, 256)	65792
dense_4 (Dense)	(None, 10)	2570
=====		
Total params: 511,242		
Trainable params: 511,242		
Non-trainable params: 0		
=====		

مدل کانولوشنی:

Model: "sequential_19"

Layer (type)	Output Shape	Param #
zero_padding2d_81 (ZeroPadding2D)	(None, 34, 34, 3)	0
conv2d_95 (Conv2D)	(None, 30, 30, 16)	1216
max_pooling2d_60 (MaxPooling2D)	(None, 15, 15, 16)	0
zero_padding2d_82 (ZeroPadding2D)	(None, 17, 17, 16)	0
conv2d_96 (Conv2D)	(None, 13, 13, 32)	12832
max_pooling2d_61 (MaxPooling2D)	(None, 7, 7, 32)	0
zero_padding2d_83 (ZeroPadding2D)	(None, 9, 9, 32)	0
conv2d_97 (Conv2D)	(None, 5, 5, 64)	51264
max_pooling2d_62 (MaxPooling2D)	(None, 3, 3, 64)	0
zero_padding2d_84 (ZeroPadding2D)	(None, 5, 5, 64)	0
conv2d_98 (Conv2D)	(None, 5, 5, 64)	102464
max_pooling2d_63 (MaxPooling2D)	(None, 3, 3, 64)	0
flatten_7 (Flatten)	(None, 576)	0
dense_17 (Dense)	(None, 128)	73856
dense_18 (Dense)	(None, 10)	1290
Total params: 242,922		
Trainable params: 242,922		
Non-trainable params: 0		

این مدل تقریباً نصف مدل قبلی پارامتر دارد.

تعداد epoch را ۱۵ در نظر گرفتیم.

(الف)

برای مدل fully connected:

دقت و خطا در آخرین epoch روی داده ی train:

Epoch 15/15

1563/1563 [=====] - 5s 3ms/step - loss: 1.5487 - accuracy: 0.4383

دقت و خطا روی داده ی تست:

Loss and Accuracy on Test set :

313/313 [=====] - 1s 3ms/step - loss: 1.7115 - accuracy: 0.3893

برای مدل کانولوشنی:

دقت و خطا در آخرین epoch روی داده ی train:

Epoch 15/15

1563/1563 [=====] - 7s 5ms/step - loss: 0.5941 - accuracy: 0.7896

دقت و خطا روی داده ی تست:

Loss and Accuracy on Test set :

313/313 [=====] - 1s 3ms/step - loss: 1.3093 - accuracy: 0.6333

ما برای آپدیت کردن وزن های شبکه از تابع loss استفاده میکنیم که معمولا تابعی از اختلاف خروجی درست موردنظر برای ورودی و خروجی داده شده توسط شبکه است و با توجه به نوع شبکه روابط مختلفی برای آن وجود دارد. در واقع با بدست آوردن مقدار خطا و پخش کردن آن در لایه های قبلی سعی میکنیم این خطا را کاهش دهیم و اختلاف خروجی شبکه با خروجی درست را کم کنیم. حالا هر چه قدر که بتوانیم این خطا را کاهش بدهیم (با افزایش تعداد تکرار، پیچیده تر کردن شبکه، افزایش لایه ها و ..) جواب شبکه به ورودی به پاسخ درست نزدیک تر میشود و میتوان گفت دقت افزایش می یابد.

ممکن است در حالت هایی با کاهش یافتن مقدار loss دقت افزایش پیدا نکند مثلا هر چه قدر که خطا کاهش پیدا نکند برای یک ورودی، باز هم جواب شبکه درست نمیشود در این حالت ها ممکن است دقت افزایش پیدا نکند. دقت دیتای تست یعنی نسبت تعداد جواب های درست شبکه به ورودی ها به روی تعداد کل جواب ها یا کل دیتا و رابطه ی ریاضی خاصی با خود جواب ها ی شبکه و جواب های درست و خطا ندارد. اما loss چون مستقیما با اختلاف جواب ها رابطه دارد حتی ممکن است در صورتی که شبکه دقت خوبی دارد و ۱۰۰ درصد جواب ها درست است (مثلا شبکه هایی که لایه ی آخر با sigmoid کار میکنند و با احتمال بیشتر پاسخ نهایی انتخاب میشود)، مقدار loss ۰ نشود.

منبع کورس NLP در کورسرا

(ب)

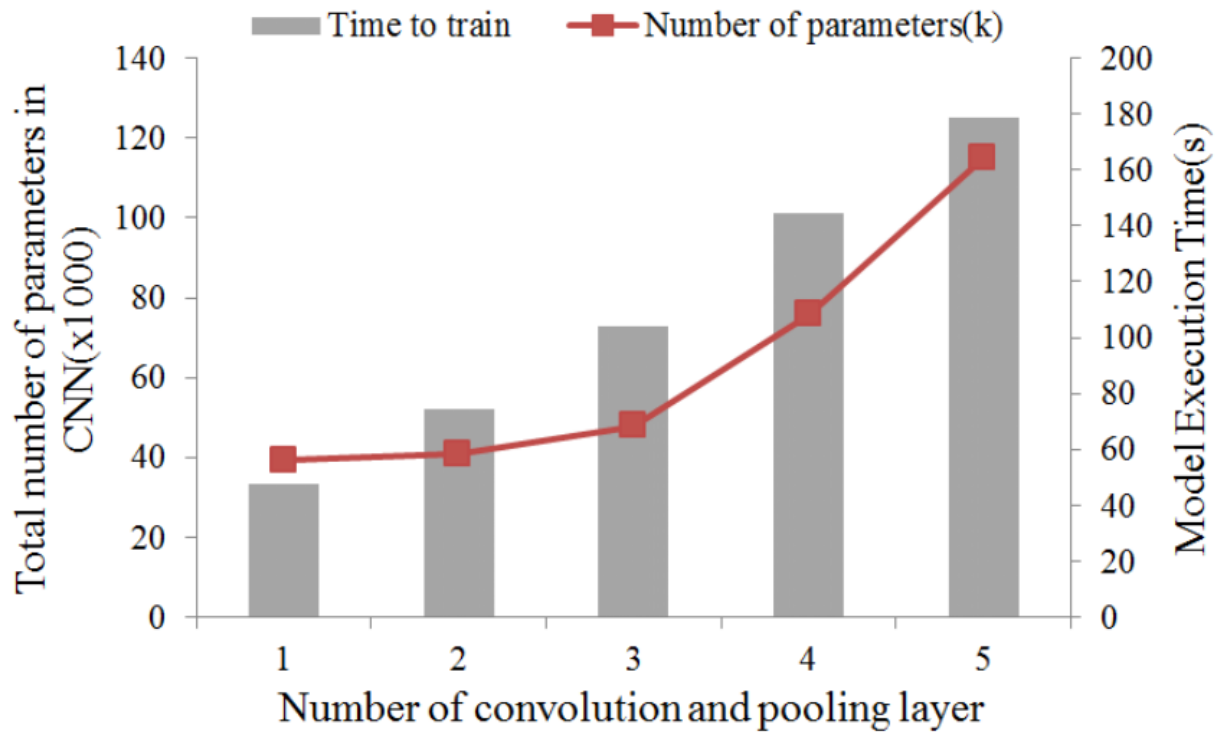
مدت زمان همه ی epoch ها در مدل کاملا متصل : ۵ ثانیه

در مدل کانولوشنی: ۷ و ۸ ثانیه که در ۳ تای اول ۸ و ۹ و ۸ و در بقیه ی تکرار ها ۷ ثانیه بوده است.

(پ)

خیر ارتباطی ندارد. اگر این طور بود مدل اول تعداد پارامتر بیشتری دارد و باید اجرای هر ایپاک آن بیشتر طول میکشید اما زمان اجرای هر ایپاک در مدل دوم بیشتر است. از طرفی در مدل دوم ورودی های هر لایه دو بعدی هستند و این هم ممکن است روی پیچیدگی محاسبات و طولانی تر شدن تاثیر گذاشته باشد. درواقع هرچه پارامتر مشترک بیشتر باشد و هرچه محاسبات شبکه برای تابع هزینه و آپدیت کردن ها پیچیده تر باشد (مثلا قابل انجام به صورت موازی نباشد و حتما باید به صورت متوالی انجام شود چون جواب بعدی به قبلی بستگی دارد). بیشتر طول میکشد.

اما به طور کلی در مقایسه ی یک نوع از شبکه های عصبی، با افزایش تعداد پارامتر ها زمان اجرا افزایش پیدا خواهد کرد.



منبع:

https://www.researchgate.net/figure/Change-in-the-number-of-model-parameters-and-training-time-with-increased-number-of_fig3_335865710