گزارش تمرین سری سوم

اصول بینایی کامپیوتر

رضا اکبریان بافقی - ۹۵۱۰۰۰۶۱

۱

با استفاده از سرویس آنلاین Colaboratory و با استفاده از لایبری tensorflow شبکه عصبی‌ای ساختم که دقت آن پس از ۵۰ epoch برابر **۹۹,۳۷٪** می‌شود. می‌توانید پارامترهای این شبکه را در جدول زیر مشاهده کنید.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_4 (Conv2D)            (None, 28, 28, 128)       1280
_________________________________________________________________
activation_6 (Activation)    (None, 28, 28, 128)       0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 14, 14, 128)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 14, 14, 64)        73792
_________________________________________________________________
activation_7 (Activation)    (None, 14, 14, 64)        0
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
flatten_2 (Flatten)          (None, 3136)              0
_________________________________________________________________
dropout_2 (Dropout)          (None, 3136)              0
_________________________________________________________________
dense_4 (Dense)              (None, 100)               313700
_________________________________________________________________
batch_normalization_2 (Batch (None, 100)               400
_________________________________________________________________
dense_5 (Dense)              (None, 10)                1010
_________________________________________________________________
activation_8 (Activation)    (None, 10)                0
=================================================================
Total params: 390,182
Trainable params: 389,982
Non-trainable params: 200
```

در این مدل، ابتدا یک لایه کانولوشنال با ۱۲۸ فیلتر روی تصویر اعمال می‌شود، برای هر فیلتر ۱۰ پارامتر داریم که می‌شود ۱۲۸۰ پارامتر در این مرحله. سپس از تابع فعال‌ساز ReLU استفاده می‌شود و با استفاده از MaxPooling با اندازه ۲ در ۲، اندازه تصویر نصف می‌شود و هم‌چنین داده‌های اضافی را پاک می‌کند.

در لایه بعدی با این تفاوت که از ۶۴ فیلتر استفاده می‌کنیم و باقی مراحل را مانند لایه قبلی انجام می‌دهیم. حال عکس‌های ما تبدیل به مربع‌های ۷ در ۷ شده‌اند که از هرکدام آن‌ها ۶۴ فیلتر مختلف گذر کرده‌است. بنابراین با استفاده از Flatten آن‌ها را تبدیل به یک بردار می‌کنیم.

حال یک بردار ۳۱۲۶ بعدی داریم که می‌خواهیم با استفاده از یک لایه fully connected به ۱۰۰ نورون در لایه بعد متصل‌شان کنیم. که از Dence استفاده می‌کنیم. اما قبل از آن برای جلوگیری از overfitting و هم‌چنین به‌دست آوردن شبکه‌ای با دقت بیش‌تر از Dropout استفاده می‌کنیم که در هر مرحله با احتمال ۰,۵ هر یک از نورون‌ها از آن ۳۱۲۶ نورون را خاموش می‌کند.

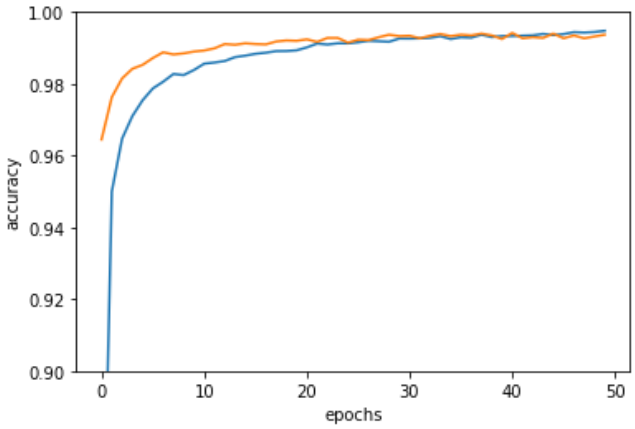
سپس بعد از آن دوباره ۱۰ نورون مشخص می‌کنیم و با استفاده دوباره از لایه fully connected این ۱۰۰ نورون را به این ۱۰ نورون وصل می‌کنیم. ولی قبل از آن عمل BatchNormalization را انجام می‌دهیم. چون ممکن است بر اثر Dropout در

لایه قبلی مقادیر این لایه دچار تغییرات زیادی شود. این کار باعث شد تا دقت داده آموزشی و تست با هم بیشتر شود. ضمن این‌که سریع‌تر هم‌گرا شدند.

در نهایت با استفاده از تابع فعال ساز softmax احتمال هر کلاس برای داده مورد نظر را به‌دست می‌آوریم. سپس تابع cross entropy را به عنوان تابع loss function و بهینه‌ساز Adam را با نرخ یادگیری اولیه ۰,۰۰۱ در نظر می‌گیریم.

ما ابتدا داده‌ها را نرمال می‌کنیم و از ۰ تا ۲۵۵ به ۰ تا ۱ می‌بریم تا فرآیند آموزش شبکه سریع‌تر اتفاق بیافتد. سپس labelهای این تصاویر را تبدیل به بردار one-hot می‌کنیم. سپس مدل را مانند کاری که در بالا گفتیم تعریف می‌کنیم.

سپس برای فیت کردن داده‌ها، اندازه batchها را ۲۵۶ در نظر می‌گیریم. سایز این batch را طوری در نظر گرفتیم که به مشکل حافظه برنخوریم. می‌توانیم نمودار دقت این مدل را در epochهای مختلف در نظر بگیریم به شکل زیر در می‌آید.



شکل ۱ همان‌طور که می‌توانید مشاهده کنید بدون رخ دادن overfitting خطای داده تست به مرور کم‌تر شده‌است. و دقت داده تست (نارنجی) و دقت داده آموزشی (نارنجی) به مقدار معینی همگرا شده‌است.

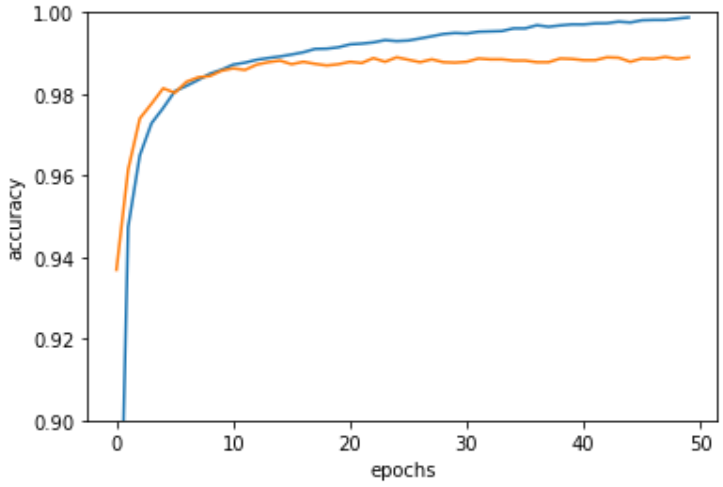
هم‌چنین می‌توانید مقدار خطای روی داده تست و خطا روی داده آموزشی را در تعدادی از epochها ببینید.

```
Epoch 1/50
235/235 [==============================] - 8s 36ms/step - loss: 0.5783 -
accuracy: 0.8241 - val_loss: 1.0962 - val_accuracy: 0.9645
Epoch 2/50
235/235 [==============================] - 8s 34ms/step - loss: 0.1818 -
accuracy: 0.9500 - val_loss: 0.1881 - val_accuracy: 0.9763
Epoch 3/50
235/235 [==============================] - 8s 34ms/step - loss: 0.1291 -
accuracy: 0.9647 - val_loss: 0.0770 - val_accuracy: 0.9815
.
.
.
Epoch 48/50
235/235 [==============================] - 8s 33ms/step - loss: 0.0188 -
accuracy: 0.9943 - val_loss: 0.0212 - val_accuracy: 0.9927
Epoch 49/50
235/235 [==============================] - 8s 33ms/step - loss: 0.0178 -
accuracy: 0.9944 - val_loss: 0.0194 - val_accuracy: 0.9932
Epoch 50/50
235/235 [==============================] - 8s 33ms/step - loss: 0.0167 -
accuracy: 0.9948 - val_loss: 0.0196 - val_accuracy: 0.9937
```

در نهایت وقتی تابع evaluate را روی این مدل اعمال می‌کنیم، حاصل زیر را به عنوان دقت و خطای خروجی می‌دهد.

[0.01958615519106388, 0.9937000274658203]

من خواستم از Data Augmentation استفاده کنم ولی در نتیجه این شبکه نتوانست تاثیر بهتری بگذارد. ولی قبل از این‌که از Dropout استفاده کنم، دقت داده تست در مقداری کم‌تر از دقت آموزشی ثابت می‌شد، که می‌توایند دقت این شبکه قبل از اعمال لایه Dropout در شکل زیر مشاهده کنید. هم‌چنین قبل از استفاده از BatchNormalization دقت شبکه در مقدار کمتر ۹۹,۱۸٪ هم‌گرا می‌شد که با اعمال این لایه دقت تا ۹۹,۳۷٪ افزایش یافت.



شکل ۲ دقت داده‌های تست (نارنجی) در مقابل دقت داده‌های آموزشی (آبی)، قبل از اعمال لایه Dropout

مدل در فایل model.h5 ذخیره شده‌است. هم‌چنین در فایل جوپیتر Q1.ipynb جزئیات کار آورده شده است.

بخش اضافی – Naive Bayes

من قبلا با این داده mnist کار کرده بودم و توانسته‌بودم با استفاده از مدل Naive Bayes و استفاده از Gaussian mixture model به دقت ۹۶,۲۷٪ برسم. که فایل جوپیتر آن را در فایل جداگانه‌ای به نام Q1_NaiveBayes_Optional.ipynb قرار داده‌ام.