

گزارش تمرین اول

در این گزارش به بررسی نکات و کد تمرین اول پرداخته و نتایج مشاهده شده تحلیل می‌گردد. این تمرین شامل دو بخش بوده که در بخش اول به طراحی مدلی برای تشخیص اعداد دست‌نویس پرداخته و در بخش دوم با حذف بخشی از کلاس‌ها از داده‌های Train، نتیجه حدس زده شده و مشاهده می‌شود.

بخش اول

در این تمرین و برای ساخت مدل، از مجموعه داده‌های MNIST یا Modified National Institute of Standards and Technology استفاده شده است. این Dataset دارای ۷۰ هزار نمونه ارقام دست‌نویس (به فرمت زبان انگلیسی) بوده که از ۶۰ هزار نمونه برای Train و از ۱۰ هزار نمونه باقی‌مانده برای Test استفاده می‌شود. تمامی نمونه‌ها به صورت سیاه و سفید بوده و در تصویری با سایزهای مشخص شده ۲۸ در ۲۸ هستند.

علت عدم تقسیم داده‌ها به صورت Train-Validation-Test امکان استفاده از Cross-study Comparison است چراکه عدم استفاده از داده‌هایی به عنوان Validation باعث مقایسه بهتر مدل‌های مختلف و عملکردشان نسبت به یکدیگر می‌شود و همچنین تقسیم داده‌ها به ۳ بخش باعث کمتر شدن داده‌ها برای بخش Train می‌شود. موارد گفته شده با تقسیم داده‌ها به ۳ بخش مخالفتی ندارد اما در این تمرین و در کاربردهای اصلی این مجموعه داده (کاربردهای آموزشی و تمرینی) تقسیم به ۲ بخش Train-Test منطقی‌تر است.

برای ساخت مدل از کتابخانه Keras استفاده شده و Sequential Classification Model با استفاده از لایه‌های Fully connected طراحی شده است. به علت بررسی مدل در حالت‌های مختلف به منظور مشاهده بهترین حالت، ابتدا اطلاعات و نتیجه هر بررسی، مشاهده و در ادامه بهترین حالت بررسی می‌شود. حالت ابتدایی توسط حالت مشخص شده در [وبسایت Kaggle](#) تعیین شده است.

جدول ۱

مقایسه Accuracy و Loss برای مدل‌های مختلف جهت تعیین بهترین حالت.

مقدار Accuracy در Epoch آخر	مقدار Loss در Epoch آخر	
۰.۹۹۷۱	۰.۰۰۹۰	حالت ابتدایی
۰.۹۹۸۰	۰.۰۰۷۷	کم کردن لایه‌ها
۰.۹۹۷۰	۰.۰۰۸۲	زیاد کردن لایه‌ها
۰.۹۹۷۱	۰.۰۰۸۹	کم کردن نورون‌های لایه‌های میانی
۰.۹۹۸۰	۰.۰۰۶۰	زیاد کردن نورون‌های لایه‌های میانی
۰.۹۱۴۳	۰.۳۰۶۷	تغییر Optimizer به Adagrad
۰.۹۹۶۸	۰.۰۰۹۷	تغییر Optimizer به Nadam
۰.۹۹۹۲	۰.۰۰۶۳	تغییر Optimizer به Adamax
۰.۹۹۹۵	۰.۰۰۱۵	تغییر Optimizer به RMSprop

با توجه به نتایج بالا، بهترین مدل تغییر Optimizer به RMSprop است و آن را می‌توان به صورت زیر توصیف کرد:

– ۳ لایه Dense با Activation از نوع relu و تعداد نورون ۲۵۶.

– ۱ لایه Dense با Activation از نوع Softmax و تعداد نورون‌های ۱۰ (برای هر کلاس).

– با Optimizer از نوع RMSprop و Batch Size به اندازه ۱۲۸ و Epoch به اندازه ۲۰.

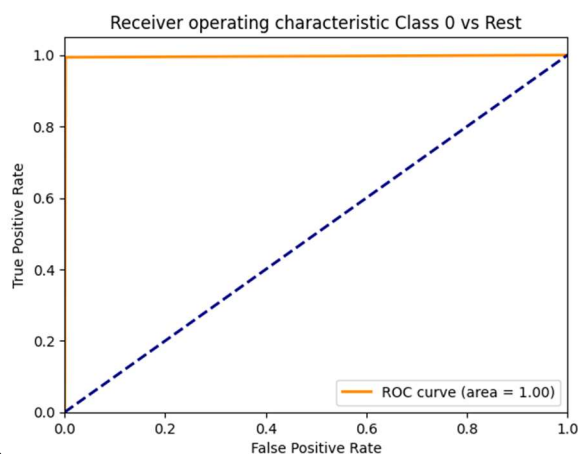
در ادامه نیز پارامترهای زیر برای داده‌های Test روی بهترین مدل بررسی شده و قابل مشاهده هستند.

– Accuracy برای بهترین مدل به اندازه ۰.۹۸۲ دیده می‌شود.

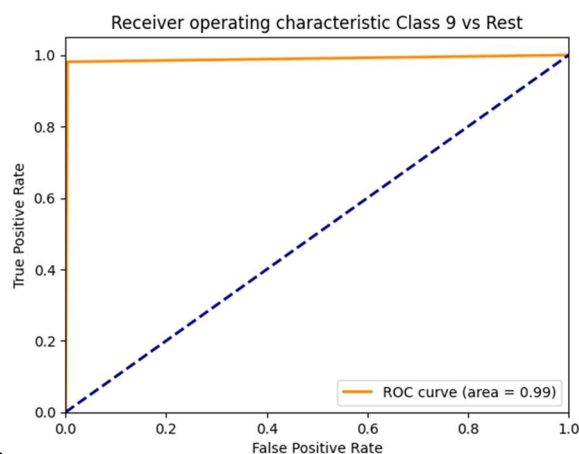
– Recall برای بهترین مدل به اندازه ۰.۹۸۲۰۲۵۸۷۷۸۱۷۹۶ دیده می‌شود.

– Precision برای بهترین مدل به اندازه ۰.۹۸۱۹۵۰۸۲۵۴۱۳۰۱۷۷ دیده می‌شود.

– ROC Curve برای هر کلاس به صورت One vs Rest به صورت زیر مشخص می‌شود. (تنها برای کلاس‌های ارقام ۰ و ۹ نمایش داده شده است).



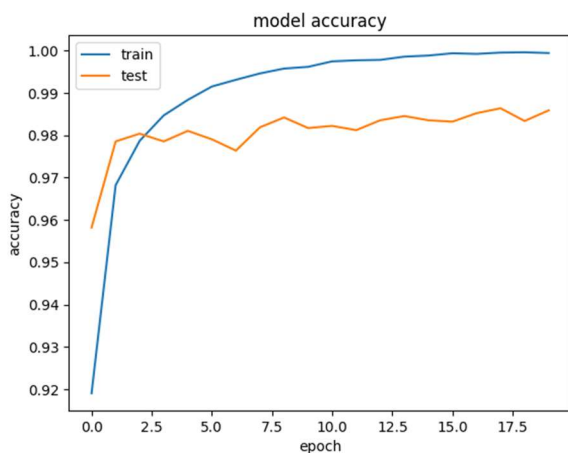
a



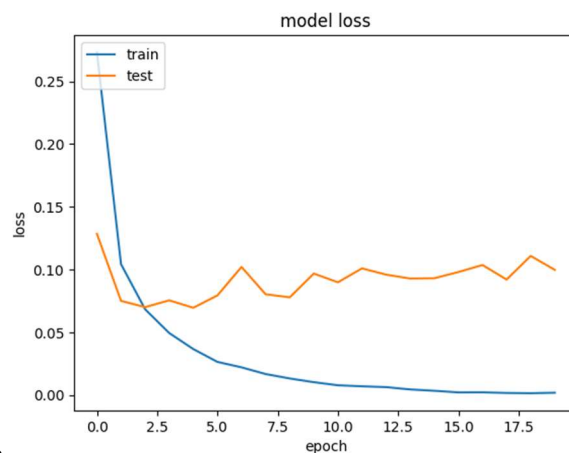
b

نمودار ۱ a نمودار ROC Curve برای رقم ۰ نسبت به بقیه ارقام. B نمودار ROC Curve برای رقم ۹ نسبت به بقیه ارقام.

– Learning Curve به صورت زیر دیده می‌شود. این نمودار برای Loss و Accuracy در هر Epoch مشخص می‌شود.



a



b

نمودار ۲ a نمودار Learning Curve دارای مقادیر Accuracy در هر Epoch. b نمودار Learning Curve دارای مقادیر Loss در هر Epoch.

بخش دوم

در این بخش، کلاس‌های اعداد ۵ تا ۹ از داده‌هایی که برای Train استفاده می‌شوند حذف شده و به صورت داده‌های Test به مدل داده شده و نتیجه برای کلاس‌هایی که مدل تا به حال ندیده است بررسی می‌شود.

با توجه به توضیحات صورت تمرین، باید پیش‌بینی درباره نتیجه استفاده از این مدل در تشخیص اعداد کلاس‌های ۵ تا ۹ بررسی شود. با توجه به اینکه مدل هیچ کدام از این اعداد را در بخش Train ندیده است، بنابراین نمی‌تواند هیچ کدام را به صورت درست تشخیص دهد اما می‌توان پیش‌بینی کرد که هر رقم ۵ تا ۹ که بیشتر به ارقام ۰ تا ۴ شباهت دارند، در کلاس آن دسته قرار می‌گیرند.

برای تشخیص این دسته‌ها، حالت‌هایی از مدل بخش اول که به اشتباه در دسته دیگری قرار گرفته‌اند را بررسی کرده و با توجه به بیشترین تعداد اشتباه می‌توان حدودی حدس زد که کدام اعداد بیشترین شباهت را به هم دارند. توجه شود که علت اینکه اعدادی در مدل بخش اول اشتباه در دسته دیگری قرار گرفته‌اند، این است که دست خط خاص بوده یا کمی از پیکسل‌ها به صورت خاصی تغییر کرده‌اند و باعث اشتباه مدل شده است ولی اعدادی که به صورت واضح و معمولی نوشته شده‌اند در دسته مورد نظر خود قرار گرفته‌اند و این حدس‌ها فقط در حد پیش‌بینی بوده و نمی‌توان به صورت کامل به آنها اکتفا کرد. نتیجه این حدس به صورت زیر قابل مشاهده است. (تصاویر تنها برای تشخیص‌های اعداد ۶ نمایش داده شده است).

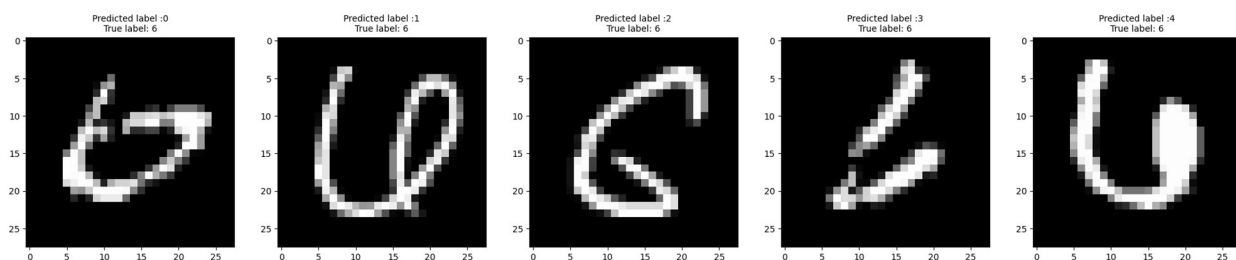
- عدد ۵ بیشترین بار به صورت عدد ۳ تشخیص داده شده است.

- عدد ۶ بیشترین بار به صورت عدد ۱ تشخیص داده شده است.

- عدد ۷ بیشترین بار به صورت عدد ۲ تشخیص داده شده است.

- عدد ۸ بیشترین بار به صورت عدد ۲ تشخیص داده شده است.

- عدد ۹ بیشترین بار به صورت عدد ۴ تشخیص داده شده است.



تصویر ۱ تصاویر حالتی که عدد ۶ به صورت اشتباه در کلاس‌های دیگر تشخیص داده شده است.

برای ساخت مدل ابتدا تمام نمونه‌ای ارقام ۰ تا ۴ و ۵ تا ۹ از داده‌های Train و Test به صورت جداگانه تفکیک شده و داده‌های ۰ تا ۴ به عنوان داده Train و داده‌های ۵ تا ۹ به عنوان داده Test به مدل قبلی داده شده و مدل روی داده‌های جدید Fit می‌شود. به علت تقریباً برابر شدن سائز این مجموعه داده‌ها و به همان نسبت کمتر شدن داده‌های Train، استفاده از تفکیک مقداری مشابه داده‌ها برای مدل‌های دیگر پیشنهاد نمی‌شود و این تقسیم‌بندی در این مثال فقط به منظور عدم مشاهده برخی کلاس‌ها توسط مدل، صورت گرفته است.

نتیجه اعمال مدل روی مجموعه داده انتخابی را می‌توان در ادامه مشاهده کرد. (تصاویر تنها برای تشخیص‌های اعداد ۶ نمایش داده شده است.) به علت عدم وجود هیچ داده‌ای از کلاس‌های انتخابی در بخش داده‌های Train، مقدار دقت در Predict کردن در بخش

Test، کاملاً صفر است. (اگر از بخشی از داده‌های کلاس‌های ۰ تا ۴ در داده‌های Test نیز استفاده می‌شد، مقدار دقت کمی بالاتر از صفر قرار می‌گرفت اما به علت عدم وجود هیچ کدام از ارقام ۵ تا ۹، این مقدار باز هم مقدار کمی بود.)

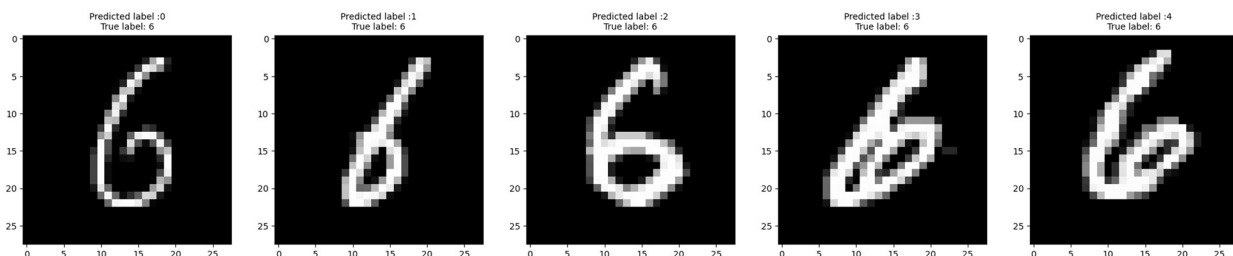
- عدد ۵ بیشترین بار به صورت عدد ۳ تشخیص داده شده است.

- عدد ۶ بیشترین بار به صورت عدد ۴ تشخیص داده شده است.

- عدد ۷ بیشترین بار به صورت عدد ۳ تشخیص داده شده است.

- عدد ۸ بیشترین بار به صورت عدد ۳ تشخیص داده شده است.

- عدد ۹ بیشترین بار به صورت عدد ۴ تشخیص داده شده است.



تصویر ۲ تصاویر حالتی که عدد ۶ در کلاس‌های دیگر تشخیص داده شده است.

مشکل اصلی مدل بررسی شده در عدم تشخیص کلاس‌های ۵ تا ۹ به درستی نبوده و مشکل این است که مدل نمی‌تواند تشخیص دهد که این داده‌ها را تا به حال مشاهده نکرده است و با قطعیت مشخص می‌کند که این رقم مشابه رقمی که در هنگام Train شدن دیده است می‌باشد. برای رفع این مشکل و همچنین تشخیص و اعمال اعلام کردن عدم قطعیت در تشخیص یک کلاس، می‌توان کلاس دیگری را نیز بجز کلاس‌های مشخص شده در نظر گرفت و اگر داده‌های جدید در هیچ کدام از کلاس‌ها با قطعیت بالا قرار نمی‌گرفت (می‌توان از یک Threshold برای تشخیص قرار گرفتن یا نگرفتن یک داده در یک کلاس استفاده کرد)، آن را در کلاس آخری به عنوان داده ای که در بخش Train ندیده است و نمی‌داند در کدام دسته قرار می‌گیرد، جایگذاری کرده و با تغییر مدل به این حالت می‌توان به درستی مدل در بخش‌هایی که دیده است نیز اطمینان بیشتری داشت چراکه داده‌های کلاس‌های خارج از داده‌های Train نیز در مقدار دقت تاثیر داشته و در صورت تشخیص به عنوان کلاسی جدا، مقدار دقت قابل اطمینان می‌شود و خوبی یا بدی مدل نیز می‌تواند به درستی بررسی شود.