

مقدمه

کتابخانه TensorFlow و Keras

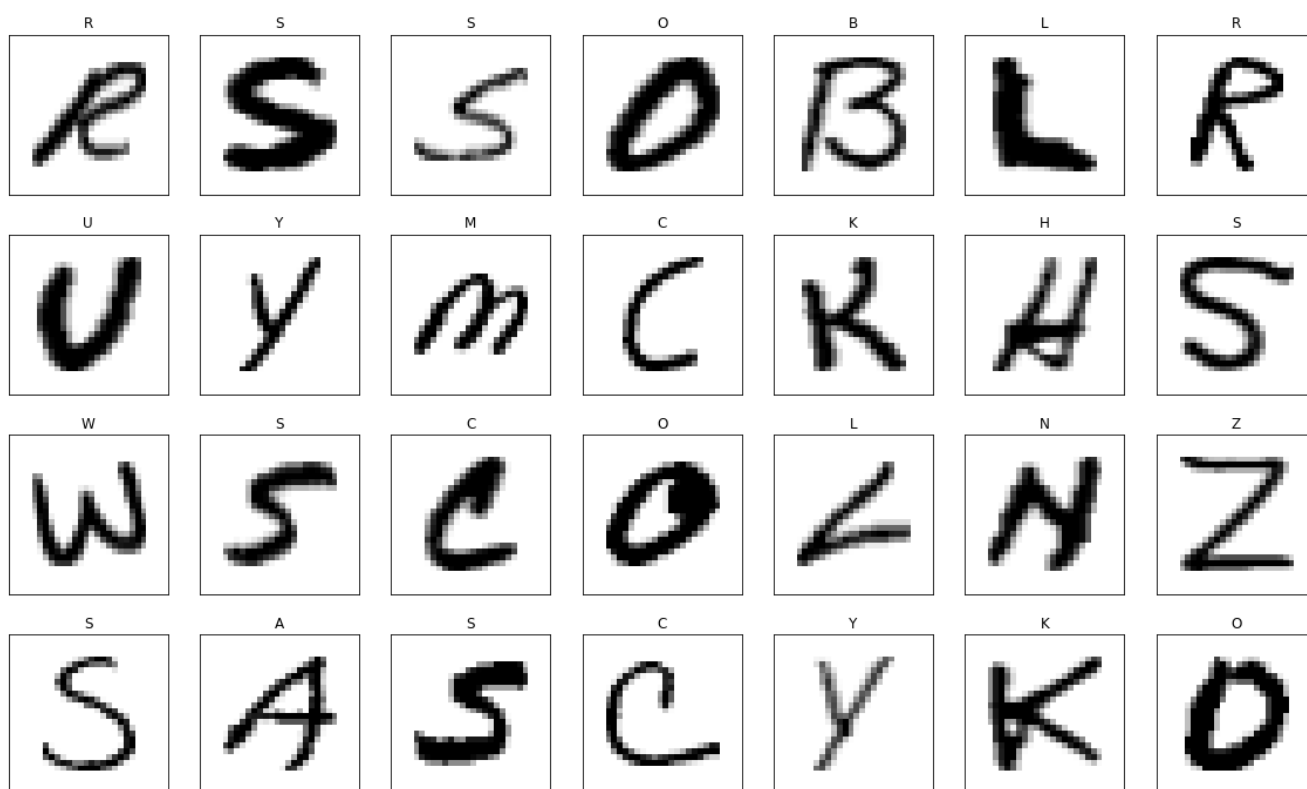
شبکه‌های عصبی یکی از قدرتمندترین ساختارهای یادگیری ماشین است که در سال‌های اخیر با توجه به افزایش قدرت محاسباتی پردازنده‌ها، کاربردهای بسیاری در حوزه‌های مختلف علمی و صنعتی پیدا کرده است. از مزایای این الگوریتم‌ها آن است که امکان ساخت هر تابع مشتق‌پذیر دلخواهی را با تنها استفاده از دو لایه مخفی از نورون‌ها ممکن می‌سازد. بنابراین امکان پیاده‌سازی و یادگیری توابع غیرخطی نیز با کمک آن‌ها ممکن است. با توجه به پیشرفت‌های اخیر این ساختار و کاربرد روز افزون آن در صنعت، فریم ورک‌های بسیار قدرتمندی برای سهولت در ایجاد شبکه‌های عصبی بسیار پیچیده و با کارایی بسیار بالا عرضه شده است. در این پروژه قصد داریم با کتابخانه TensorFlow و با استفاده از رابط سطح بالای Keras برخی از مسائل مربوط به شبکه‌های عصبی را بررسی نماییم.

تعریف مسئله

در این پروژه قصد داریم با استفاده از شبکه‌های عصبی، الفبای انگلیسی را بر اساس تصویر آن‌ها تشخیص دهیم. برای راحتی استفاده از کتابخانه‌ها و انجام محاسبات، می‌توانید از سرویس **Google Colab** استفاده کنید.

معرفی مجموعه داده

مجموعه داده [A-Z Handwritten Alphabets](#) یک مجموعه داده از تصاویر دست‌نویس حروف الفبای انگلیسی است. این مجموعه داده شامل 372,450 تصویر همراه با label متناظر آن‌هاست. (برای دانلود یا استفاده از مجموعه داده در Colab، می‌توانید از سایت [kaggle](#) یا این [لینک](#) استفاده کنید)



نکات پیاده‌سازی

در این پروژه باید به کمک رابط Keras یک شبکه Feed Forward ایجاد کنید که حداقل شامل دو لایه مخفی باشد (با در نظر گرفتن لایه‌های ورودی و خروجی (softmax) باید شبکه شما حداقل دارای ۴ لایه باشد). دقت کنید به هیچ عنوان نباید از لایه‌های شبکه‌های Convolution و Recurrent استفاده کنید.

فاز اول: بررسی و پیش‌پردازش داده

- با استفاده از کتابخانه pandas داده csv را بخوانید.
- داده‌ها و برچسب متناظر هریک از آن‌ها را از یکدیگر جدا کرده و به طور مناسبی آن‌ها را ذخیره کنید.
- داده‌ها را به تصاویر با مقیاس ۲۸ در ۲۸ تغییر دهید.
- تعداد داده‌های خوانده شده و تعداد کلاس‌های موجود را چاپ کنید.
- تعداد تصاویر هر کلاس را محاسبه کنید و برای آن نمودار میله‌ای رسم کنید.
- از هر کلاس موجود، فقط یک تصویر همراه با نام کلاس آن را نشان دهید.
- دیتاست را به دو بخش train و test و با مقیاس مناسب تقسیم کنید.
- برای برچسب‌ها One Hot Encoding انجام دهید و دلیل این کار را شرح دهید. ارتباط این موضوع با طراحی ساختار لایه‌ی آخر شبکه عصبی خود را بیان کنید.

فاز دوم: طراحی شبکه عصبی

یک شبکه‌ی عصبی با شرایط زیر طراحی کنید:

- بهینه‌ساز شما باید از نوع SGD باشد.
- مقدار learning rate باید 0.01 باشد.
- تعداد epoch ها باید 10 باشد.
- اندازه batch_size باید 32 باشد.
- تابع فعال‌سازی تمام لایه‌ها غیر از لایه آخر باید RELU باشد.

★ نکته ۱: مقدار batch_size در تمامی مراحل بعدی 32 است مگر آنکه به طور مستقیم ذکر شود که آن را تغییر دهید.

★ نکته ۲: در تمامی مراحل تمرین، پس از اتمام تمامی epoch ها باید نمودار مقدار loss و accuracy در هر epoch را رسم کنید.

★ نکته ۳: در تمامی مراحل تمرین، پس از اتمام تمامی epoch ها باید معیارهای F1, recall, precision برای

داده‌های تمرین و تست چاپ شود.

پس از ایجاد ساختار شبکه عصبی مناسب با توجه به شرایط داده شده تعداد پارامترهای هر لایه را چاپ کرده و اعداد چاپ شده را توجیه کنید (برای چاپ تعداد پارامترها میتوانید از رابط Keras استفاده کنید و نیازی به پیاده سازی توسط خودتان نیست).

فاز سوم: طبقه‌بندی داده‌ها

در این بخش باید شبکه عصبی ایجاد شده را روی داده‌ها تمرین دهید.
در این بخش باید داده‌های خوانده‌شده را نرمال کنید. برای این کار هر پیکسل داده خوانده شده باید به 255 تقسیم گردد.
در این قسمت باید ساختار شبکه عصبی خود را به گونه‌ای تغییر دهید که بیشترین مقدار F1 برای نتیجه روی داده‌ی تست ایجاد شود. برای بقیه بخش‌ها از همین ساختار بهینه استفاده نمایید. (برای به دست آوردن یک شبکه عصبی مناسب می‌توانید تابع‌های فعال‌ساز دیگری مثل tanh و sigmoid و Leaky RELU را هم بررسی کنید)

بررسی تاثیر تغییرات متخلف و سوالات:

قسمت اول) تاثیر optimizer:

- تحقیق کنید که momentum چیست و چرا استفاده از آن در مرحله train مفید است؟
- مدل را با momentum برابر 0.5 و 0.9 و 0.98 تمرین دهید و نتایج را تفسیر کنید.
- آیا همواره با افزایش momentum نتیجه بهبود می‌یابد؟ توضیح دهید.
- یکی از مشهورترین optimizer هایی که در فرایند تمرین دادن مدل‌ها بسیار استفاده می‌شود، Adam است. مدل خود را با استفاده از Adam تمرین دهید و نتیجه را با SGD مرحله‌ی قبل مقایسه کنید. در قسمت‌های بعدی از Adam استفاده کنید (از نرخ اولیه پیش‌فرض برای train به کمک Adam استفاده کنید که مقدار آن 0.001 است).

قسمت دوم) تاثیر تعداد epoch:

- مدل را به تعداد epoch = 20 تمرین دهید.
- آیا در همه مسائل نیاز به آن است که شبکه عصبی در چندین epoch تمرین نماید؟ دلیل این مسئله چیست؟
- آیا همواره استفاده از تعداد epoch های بیشتر برای تمرین مفید است؟ اگر جواب شما مثبت است، دلیل خود را توضیح دهید و اگر جواب شما منفی است، راه حل های مقابله با اتفاق نامطلوبی که روی می دهد را بیان کنید.

قسمت سوم) تاثیر Loss Function:

- شبکه عصبی را با استفاده از تابع هزینه مشهور به MSE^1 تمرین دهید و نتیجه را با categorical cross entropy مقایسه نمایید.
- چرا استفاده از تابعی مثل MSE برای classification مناسب نیست؟ چه زمانی از این تابع استفاده می کنیم؟

قسمت چهارم) تاثیر regularization:

- یکی از مشکلات جدی در تمرین دادن شبکه های عصبی، مسئله overfitting است. در این قسمت به بررسی راه حل های این موضوع می پردازیم.
- با استفاده از رابط keras برای هر لایه L2 regularization را با مقدار 0.0001 اعمال کنید و نتیجه را مشاهده کنید و توضیح دهید.
- با استفاده از رابط keras برای هر لایه dropout با مقدار 1.0 قرار دهید. تاثیر استفاده از dropout در حین تمرین و تست شبکه را توضیح دهید. نتیجه استفاده از این لایه را تفسیر کنید.

منابع

- برای یادگیری شروع کار با Keras میتوانید از این [لینک](#) کمک بگیرید.
- برای آموزش شیوه استفاده از Google Colab می توانید از این [لینک](#) یا این [لینک](#) استفاده نمایید.
- در این [لینک](#) درباره ی Adam Optimizer مطالعه کنید.
- در این [لینک](#) درباره ی روش های مختلف regularization مطالعه کنید.

¹ Mean squared error

نکات پایانی

- دقت کنید که هدف پروژه تحلیل نتایج است؛ بنابراین از ابزارهای تحلیل داده بطور مثال نمودارها استفاده کنید و توضیحات مربوط به هر بخش از پروژه را بطور خلاصه و در عین حال مفید در گزارش خود ذکر کنید.
- نتایج و گزارش خود را در یک فایل فشرده با عنوان `AI-CA5-p2-<#SID>.zip` تحویل دهید. محتویات پوشه باید شامل فایل `jupyter-notebook`، خروجی `html` و فایل‌های مورد نیاز برای اجرای آن باشد. توضیح و نمایش خروجی‌های خواسته شده بخشی از نمره این تمرین را تشکیل می‌دهد. از نمایش درست خروجی‌های مورد نیاز در فایل `html` مطمئن شوید.
- توجه داشته باشید که علاوه بر ارسال فایل‌های پروژه، این پروژه به صورت حضوری نیز تحویل گرفته خواهد شد. بنابراین تمام بخش‌های پروژه باید قابلیت اجرای مجدد در زمان تحویل حضوری را داشته باشند. همچنین در صورت عدم حضور در تحویل حضوری نمره‌ای دریافت نخواهید کرد.
- استفاده از `jupyter-notebook` در این پروژه الزامی است. همچنین پیشنهاد می‌شود با توجه به حجم بالای مجموعه داده و حجم زیاد محاسبات از `Google Colab` استفاده نمایید.
- هدف از تمرین، یادگیری شماسست. لطفاً تمرین را خودتان انجام دهید.
- در صورت وجود هرگونه ابهام درباره پروژه، بهتر است سوال خود را در فروم درس مطرح کنید تا بقیه از آن استفاده کنند، در غیر این صورت می‌توانید با طراحان پروژه در ارتباط باشید.

موفق باشید