



مهلت تحویل: دوشنبه ۷ تیرماه ۱۴۰۰

مقدمه:

کتابخانه Keras و TensorFlow

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 مشتق پذیر دلخواهی را با تنها استفاده از دو لایه مخفی از نورون‌ها ممکن می‌سازد. بنابراین امکان پیاده‌سازی و یادگیری توابع غیر خطی نیز با کمک آن‌ها ممکن است. با توجه به پیشرفت‌های اخیر این ساختار و کاربرد روز افزون آن در صنعت، فریم‌ورک‌های 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 مسئله قصد داریم با استفاده از شبکه‌های عصبی نژاد هر فرد را بر اساس تصویر چهره او تشخیص دهیم. توصیه می‌شود به خاطر حجم زیاد مجموعه داده این تمرین را با استفاده از google colab انجام دهید.

معرفی مجموعه داده

مجموعه داده [UTKFace](#) یک مجموعه داده چهره در مقیاس بزرگ با دامنه سن بالا (از 0 تا 116 سال است). این مجموعه داده شامل بیش از 20,000 تصویر چهره همراه با سن، جنسیت و نژاد مربوط به هر فرد است. این تصاویر تنوع زیادی در ژست، حالت چهره، نور، وضوح و غیره را پوشش می‌دهد. از این مجموعه داده به عنوان مثال برای تشخیص چهره، تخمین سن، پیشرفت / رگرسیون سن و ... استفاده می‌شود.



نکات پیاده‌سازی

در این پروژه باید به کمک رابط Keras یک شبکه Forward Feed ایجاد کنید که حداقل شامل دو لایه مخفی باشد (با در نظر گرفتن لایه‌های ورودی و خروجی (softmax) باید شبکه شما حداقل دارای 4 لایه باشد).

فاز اول: بررسی و پیش‌پردازش داده

- با استفاده از رابط Keras داده‌های را بخوانید. سپس تصاویر را grayscale کرده و به مقیاس 100 در 100 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 برجسته‌سازی One Hot Encoding انجام دهید و دلیل این کار را شرح دهید.

فاز دوم: طراحی شبکه عصبی

یک شبکه‌ی عصبی با شرایط زیر طراحی کنید:

1. بهینه‌ساز شما باید از نوع SGD باشد.
2. مقدار learning rate باید 0.01 باشد.
3. تعداد epoch ها باید 10 باشد.
4. اندازه batch_size باید 32 باشد.
5. تابع فعال‌سازی تمام لایه‌ها غیر از لایه آخر باید relu باشد.

نکته ۱: دقت کنید batch size در تمامی مراحل بعدی 32 است مگر آنکه به طور مستقیم ذکر شود که آن را تغییر دهید.

نکته ۲: پس از هر مرحله از تمرین (بعد از اتمام تمامی epoch ها) موارد زیر باید چاپ شود:

- نمودار مقدار loss در هر epoch
- نمودار مقدار accuracy در هر epoch

نکته ۳: پس از هر مرحله از تمرین (بعد از اتمام تمامی epoch ها) موارد زیر باید روی داده‌های تمرین و تست چاپ شود:

- معیارهای recall
- معیارهای precision
- معیارهای F1

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: طبقه بندی داده‌ها

در این بخش باید شبکه عصبی ایجاد شده را روی داده‌ها تمرین دهید.

در این بخش باید داده‌های را که از فایل‌ها دریافت می‌کنید 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 بدست آوردن یک شبکه عصبی مناسب می‌توانید تابع‌های فعال ساز دیگری مثل tanh و sigmoid و Leaky Relu را هم بررسی کنید ولی الزامی ندارد.

نکته: در فاز قبل با بسیاری از مسائل و پارامترهایی که خود توانایی ساخت و تغییر آن را داشتید آشنا شده‌اید و آن پارامترها در این فاز بررسی نمی‌گردند و فقط مسائلی که قادر به پیاده‌سازی آنها نبودید در ادامه به کمک Keras بررسی می‌شوند.

قسمت اول) تاثیر optimizer

در این بخش تاثیر optimizer ها را بررسی می‌کنیم.

- تحقیق کنید که momentum چیست و چرا استفاده از آن مفید است؟
- مدل را با momentum های 0.5 و 0.9 تمرین دهید و نتیجه را توضیح دهید.
- آیا همواره با افزایش momentum نتیجه بهبود میابد؟ 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 قبل مقایسه کنید. در قسمتهای بعدی از Adam استفاده کنید (از نرخ اولیه پیشفرض برای تمرین به کمک Adam استفاده کنید که مقدار آن 0.001 است).

قسمت دوم) تاثیر epoch

در این بخش تاثیر تعداد epoch ها را بررسی می‌کنیم.

- مدل را به تعداد 20 epoch تمرین دهید. برای سولات بعدی تعداد epoch ها را برابر 20 در نظر بگیرید.
- چرا معمولا شبکه‌های عصبی را در چندین epoch تمرین می‌دهند؟ آیا در همه مسائل نیاز به آن است که شبکه عصبی در چندین epoch تمرین نماید؟
- آیا همواره استفاده از تعداد epoch های بیشتر برای تمرین مفید است؟ اگر جواب شما منفی است، راه حل‌های مقابله با اتفاق نامطلوبی که روی میدهد را بیان کنید.

قسمت سوم) تاثیر loss function

در این بخش تاثیر تعداد تابع Loss های مختلف را بررسی می کنیم.

- شبکه عصبی را با استفاده از تابع Loss مشهور به MSE 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

در این بخش روش های regularization در شبکه های عصبی را بررسی می کنیم.

- با استفاده از رابط keras برای هر لایه regularization L2 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 0.1 قرار دهید و نتیجه را مشاهده نمایید و توضیح دهید.

پس از بدست آوردن بهترین طراحی شبکه خود، ۱۰ عکس که به درستی پیش بینی کرده اید و ۱۰ عکس که به اشتباه پیش بینی کرده اید را نمایش دهید.

فاز چهارم: ترسیم داده های با بعد کاهش یافته (امتیازی)

دو کاربرد عملی جالب encoder، عبارت است از data denoising و کاهش ابعاد برای ترسیم داده ها است.

در این بخش قصد داریم به کمک یک شبکه ی عصبی Feed Forward، بعد داده های ورودی را از 10000 به 2 کاهش دهیم و داده های با بعد کاهش یافته را در فضای دو بعدی ترسیم کنیم. برای این کار، شبکه ای با معماری زیر پیاده سازی کنید:

- لایه آخر شبکه (لایه خروجی) 5 نورون داشته باشد (به تعداد کلاسها).

- لایه یکی مانده به آخر شبکه 2 نورون داشته باشد (جهت ترسیم خروجی آن در فضای دوبعدی).

- لایه نخست شبکه 10000 نورون داشته باشد (به تعداد فضای ویژگی داده های ورودی).

- تعداد نورون های سایر لایه های شبکه از سمت ورودی به سمت خروجی کاهشی باشد.

بخش نخست شبکه فوق (شامل تمامی لایه ها بجز لایه آخر) را اصطلاحاً (Encoder) و وظیفه ی آن کاهش بعد ویژگی داده های ورودی است. در نهایت، داده های کاهش یافته در لایه آخر طبقه بندی می شوند.

قصد داریم تا داده های کاهش یافته ی خروجی لایه ی یکی مانده به آخر شبکه فوق را در فضای دو بعدی ترسیم نماییم. برای این کار، مراحل زیر را دنبال کنید:

- لازم است تا در ابتدا خروجی لایه‌ی یکی مانده به آخر کلاس شبکه ذخیره شود.
- در گام دوم شبکه‌ای با معماری مطرح شده طراحی کنید و آموزش دهید و با پارامترهای موردنظرتان آموزش دهید تا زمانی که به دقت مناسبی برسید.
- نهایتاً خروجی لایه‌ی یکی مانده به آخر شبکه‌ی آموزش داده شده را به ازای داده‌های آموزش و تست ترسیم کنید. نقاط مربوط به هر کلاس را با یک رنگ بخصوص نمایش دهید.
- در نهایت، با توجه به نمودارهای بدست آمده، نتیجه را تفسیر کنید و عملکرد شبکه را شرح دهید. با توجه به تصویر و دقت بدست آمده از شبکه، به نظر شما شبکه‌تان در تفکیک کدام دسته از داده‌ها از هم دچار مشکل می شود و کدام داده‌ها را به راحتی از هم تفکیک می کند؟

منابع

- برای یادگیری شروع کار با Keras می‌توانید از این [لینک](#) کمک بگیرید.
- برای آموزش شیوه استفاده از Google Colab می‌توانید از این [لینک](#) استفاده نمایید.
- در این [لینک](#) درباره‌ی Optimizer Adam مطالعه کنید.
- در این [لینک](#) درباره‌ی روش‌های مختلف regularization مطالعه کنید.
- در این [لینک](#) درباره‌ی ترسیم داده‌های با بعد کاهش یافته مطالعه کنید.

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-<#SID>.zip` تحویل دهید. محتویات پوشه باید شامل فایل `jupyter-notebook`، خروجی `html` و فایل های مورد نیاز برای اجرای آن باشد. توضیح و نمایش خروجی های خواسته شده بخشی از نمره این تمرین را تشکیل می دهد. از نمایش درست خروجی های مورد نیاز در فایل `html` مطمئن شوید.
 - توجه داشته باشید که علاوه بر ارسال فایل های پروژه، این پروژه به صورت حضوری نیز تحویل گرفته خواهد شد. بنابراین تمام بخش های پروژه باید قابلیت اجرای مجدد در زمان تحویل حضوری را داشته باشند. همچنین در صورت عدم حضور در تحویل حضوری نمره ای دریافت نخواهید کرد.
 - در صورتی که سوالی در مورد پروژه داشتید بهتر است در فروم درس مطرح کنید تا بقیه از آن استفاده کنند؛ در غیر این صورت توسط ایمیل با طراحان در ارتباط باشید.
 - استفاده از `jupyter notebook` در این پروژه الزامی است. همچنین پیشنهاد می شود با توجه به حجم بالای مجموعه داده و همچنین حتما از `Google Colab` استفاده نمایید.
 - هدف از تمرین، یادگیری شماسست. لطفا تمرین را خودتان انجام دهید.
- موفق باشید.