



مدرس: دکتر فدایی و دکتر یعقوب‌زاده

طراح: حسام اسداله‌زاده، شایان شبیهی، آرش محمدپور

مهلت تحویل: پنج‌شنبه ۲۲ دی ۱۴۰۱، ساعت ۲۳:۵۹

مقدمه

شبکه عصبی: Feed Forward

تعریف مسئله

معرفی مجموعه داده

بخش اول: بررسی و پیش‌پردازش داده

بخش دوم: تکمیل بخش‌های ناقص شبکه عصبی

بخش سوم: طبقه‌بندی داده‌ها

قسمت اول) آموزش شبکه

قسمت دوم) وزن‌دهی شبکه

قسمت سوم) تاثیر learning rate

قسمت چهارم) تاثیر activation function

قسمت پنجم) تاثیر batch size (امتیازی)

بخش چهارم) استفاده از کتابخانه Tensorflow و رابط Keras

کتابخانه Keras و TensorFlow

قسمت اول) آماده کردن داده و طراحی شبکه‌ی عصبی

قسمت دوم) بررسی تاثیر تغییرات مختلف و سوالات:

تاثیر optimizer (امتیازی):

تاثیر تعداد epoch:

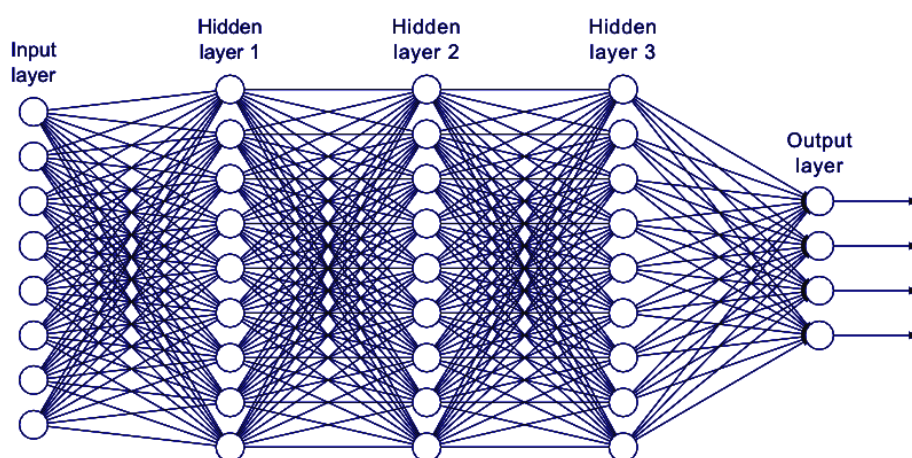
تاثیر Loss Function:

تاثیر regularization (امتیازی):

منابع

شبکه عصبی Feed Forward:

شبکه‌های عصبی یکی از قدرتمندترین ساختارهای یادگیری ماشین هستند که در سال‌های اخیر با توجه به افزایش قدرت محاسباتی پردازنده‌ها، کاربردهای بسیاری در حوزه‌های مختلف علمی و صنعتی پیدا کرده‌اند. 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 و طبقه‌بندی‌های غیرخطی با کمک آن‌ها ممکن است. در پروژه پنجم به پیاده‌سازی شبکه‌های عصبی Feed Forward جهت طبقه‌بندی تصاویر می‌پردازیم.



در شبکه‌های عصبی Feed Forward که در درس نیز با آن آشنا شدید، هر تصویر ابتدا مسطح¹ شده و به صورت بردار به عنوان ورودی شبکه داده می‌شود. هر درایه این بردار (معادل با یک پیکسل تصویر) یک ویژگی برای آن محسوب می‌شود. شبکه قرار است بر اساس این ویژگی‌ها و با ساختن ترکیبات غیرخطی از آن‌ها، وزن اتصالات بین لایه‌ها و پارامترهایش را طوری تنظیم کند، که خروجی آن ضمن داشتن کمترین خطا، کلاس تصویر ورودی متناظر را به درستی پیش‌بینی کند.

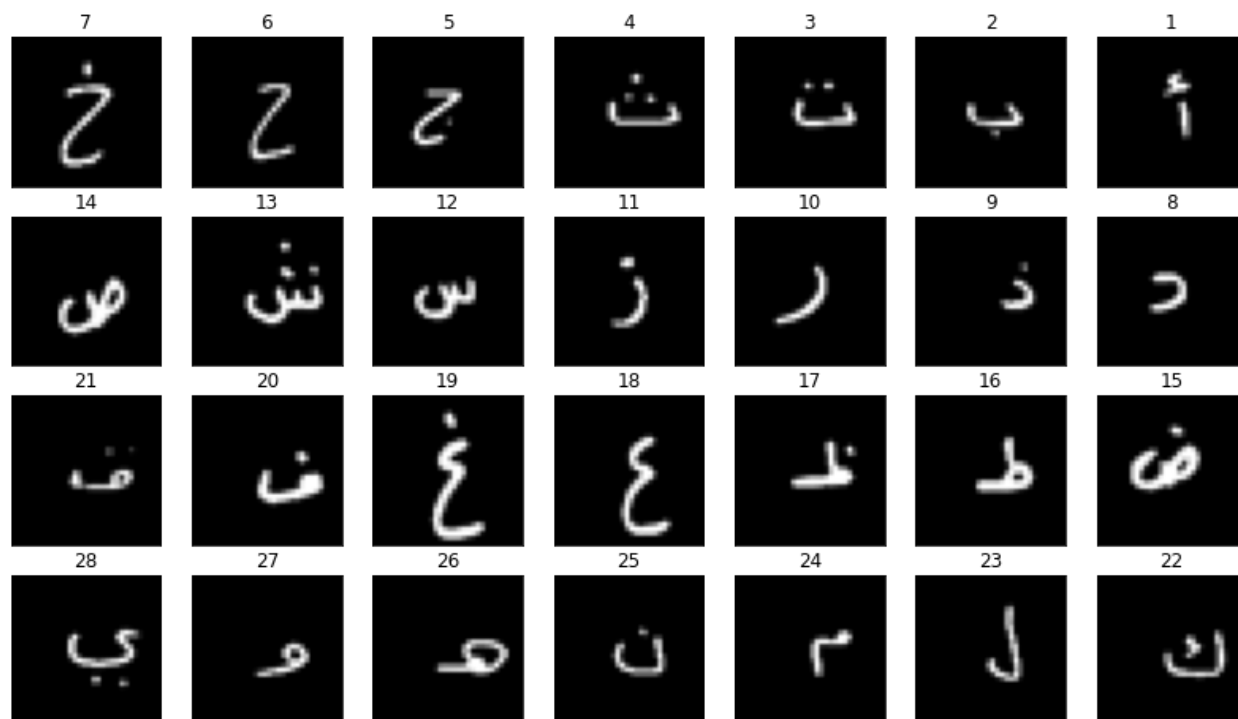
تعریف مسئله

در این تمرین، در بخش اول به پیاده‌سازی یک شبکه‌ی عصبی Feed Forward از پایه و با استفاده از کتابخانه NumPy می‌پردازید. جهت تسریع این فرایند، یک Notebook ناقص از پیاده‌سازی شبکه نیز در اختیار شما قرار می‌گیرد که لازم است از آن استفاده نمایید. در بخش دوم، به کمک کدهای بخش اول، چند شبکه‌ی عصبی را روی داده‌های داده شده آموزش خواهید داد و تاثیر برخی از عوامل را در فرایند یادگیری بررسی خواهید کرد. در بخش آخر نیز با کتابخانه‌های آماده برای پیاده‌سازی شبکه‌های عصبی آشنا خواهید شد. برای راحتی استفاده از کتابخانه‌ها و انجام محاسبات، می‌توانید از سرویس **Google Colab** استفاده کنید.

¹ Flatten

معرفی مجموعه داده

در بخش اول این تمرین شما با مجموعه داده الفبای دست‌نویس عربی (Arabic Handwritten Characters Dataset) کار خواهید کرد. مجموعه داده شامل تصاویری از ۲۸ حرف الف تا ی در زبان عربی است. در مجموع 16800 تصویر در این مجموعه داده موجود است. اطلاعات بیشتر راجع به مجموعه داده و تعداد تصاویر هر رقم در مجموعه داده را می‌توانید [اینجا](#) بخوانید.



محتویات پوشه مجموعه داده به صورت زیر است.

AI-CA5

- └─ TrainImgs.csv
- └─ TrainLabels.csv
- └─ TestImgs.csv
- └─ TestLabels.csv

ساختار این مجموعه داده به این صورت است که هر ستون، مقدار یک پیکسل از یک عکس 32x32 است و می‌توانید با reshape کردن هر سطر به این سایز، تصویر مربوطه را مشاهده کنید.

بخش اول: بررسی و پیش‌پردازش داده

- در مجموعه داده train، از هر کلاس به دلخواه یک تصویر انتخاب کرده و نمایش دهید. برای هر تصویر، برچسب آن را نیز به همراه تصویر نمایش دهید. (از کتابخانه matplotlib استفاده کنید)
- مقادیر هر پیکسل در چه محدوده‌ای قرار می‌گیرد؟

- تعداد داده‌های موجود و تعداد کلاس‌های موجود را چاپ کنید.
- ۱۶ کلاس اول (1 تا 16) از ۲۸ کلاس موجود را از مجموعه داده train و test خود انتخاب کنید. در ادامه با این ۱۶ کلاس کار خواهیم کرد.
- تعداد تصاویر هر دسته را برای مجموعه داده train و test محاسبه کنید و برای آن‌ها نمودار میله‌ای رسم کنید.
- مقدار داده‌ها را به گونه‌ای scale کنید که قبل از اطلاعات دادن به شبکه عصبی، مقدار هر pixel بین 0 تا 1 باشد. در صورت انجام ندادن این کار چه مشکلی ممکن است رخ دهد؟
- 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.

بخش دوم: تکمیل بخش‌های ناقص شبکه عصبی

یک فایل Notebook شامل کدهای ناقص مورد نیاز برای پیاده سازی شبکه عصبی Feed Forward آپلود شده و در این قسمت با تکمیل بخشهای مختلف این فایل، نهایتاً یک کلاس FeedForwardNN خواهید داشت که به کمک آن می‌توانید شبکه‌های عصبی Feed Forward با معماری‌های مختلف پیاده کنید و آموزش دهید. پارامترهای شبکه موردنظر از طریق روش SGD^2 در طی فرایند آموزش به‌روزرسانی خواهد شد. بخش‌های حذف شده از کد که لازم است آنها را کامل کنید، با #TODO مشخص شده‌اند.

بخش سوم: طبقه‌بندی داده‌ها

در این بخش به پیاده‌سازی و آموزش شبکه‌های عصبی Feed Forward با پارامترهای مختلف به کمک کلاس FeedForwardNN می‌پردازیم. در بخش Training Sample از Notebook ناقص، یک مثال از شیوه‌ی استفاده از کلاس FeedForwardNN جهت پیاده‌سازی شبکه‌های عصبی آورده شده است.

² Stochastic Gradient Descent

یک شبکه‌ی عصبی طراحی کنید و آن را با شرطهای زیر آموزش دهید:

1. مقدار learning rate باید 0.005 باشد.
2. تعداد epoch ها باید 100 باشد.
3. اندازه batch_size باید 32 باشد.
4. تابع فعال ساز ReLU در تمام لایه‌ها استفاده شود. (به جز لایه آخر که Identical است.)
5. از وزندهی اولیه uniform و یا normal استفاده کنید.

در تمام بخش‌های آینده، باید نتیجه خود را به صورت پیشفرض با شرطهای گفته شده بالا بدست آورید و فقط پارامترهای خواسته شده در هر مرحله را تغییر دهید.

قسمت اول) آموزش شبکه

- یک شبکه‌ی عصبی با شرطهای گفته شده را طراحی کنید و آموزش دهید. (می‌توانید از شبکه‌ای که در بخش Training Sample قرار گرفته استفاده کنید.)
- سعی کنید معماری شبکه (تعداد لایه‌ها و تعداد نرون‌ها در هر لایه) را طوری تغییر دهید که به دقت بهتری برسید.

قسمت دوم) وزندهی شبکه

مقدار اولیه وزن‌ها در آموزش شبکه اهمیت دارد.

- اگر مقدار اولیه تمام وزن‌های شبکه برابر صفر قرار بود و شبکه را آموزش می‌دادید، نتیجه آن چه بود؟ خروجی را با حالت قبل (که در آن وزن‌ها به صورت مقدار تصادفی مقداردهی اولیه می‌شدند)، مقایسه کنید. (نیازی به پیاده‌سازی نیست)

قسمت سوم) تاثیر learning rate

یکی از پارامترهای مهم در آموزش دادن شبکه‌های عصبی، learning rate می‌باشد.

- رفتار شبکه را برای learning rate با مقدار بالاتر (مثلا 10 برابر) و پایین‌تر (مثلا 0.1 برابر) نسبت به حالت قبل را بدست آورید. نتیجه خود را با حالت قبل مقایسه کنید و توجیه کنید.

برای تمام قسمت‌های بعد، از learning rate بهینه‌ای که بدست آورده‌اید استفاده کنید.

قسمت چهارم) تاثیر activation function

- عملکرد شبکه‌ی طراحی شده در قسمت اول را به کمک Activation Function های زیر بسنجید و نتایج را مقایسه نمایید.

○ تابع فعال‌ساز Sigmoid

○ تابع فعال‌ساز Hyperbolic Tangent

○ تابع فعال‌ساز Leaky ReLU

- دلیل اینکه Sigmoid و Tanh عملکرد مناسبی برای این دست شبکه‌ها ندارند را بیان کنید.

- برتری Leaky Relu نسبت به Relu چیست؟

توجه: در ادامه، از activation function با بهترین نتیجه در لایه‌های شبکه استفاده نمایید.

قسمت پنجم) تاثیر batch size (امتیازی)

- عملکرد شبکه را به ازای batch size با مقادیر 16 و 256 بدست آورید. نتیجه خود را با حالت قبل مقایسه کنید و توجیه کنید.

- علت استفاده از batch در فرایند آموزش چیست؟ مزایا و معایب batch size بسیار کوچک و بسیار بزرگ را شرح دهید.

حال یک شبکه‌ی عصبی Sequential 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 precision, recall, F1 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). حال باید ساختار شبکه عصبی خود را به گونه‌ای تغییر دهید که بیشترین مقدار F1 برای نتیجه روی داده‌ی تست ایجاد شود (می‌توانید از ساختار بهینه‌ی بخش قبل استفاده کنید). برای بقیه بخش‌ها از همین ساختار بهینه استفاده نمایید. (برای به دست آوردن یک شبکه عصبی مناسب می‌توانید توابع فعال‌ساز دیگری مثل tanh و Leaky RELU و ... یا learning rate های مختلف را هم بررسی کنید)

قسمت دوم) بررسی تاثیر تغییرات مختلف و سوالات:

تاثیر optimizer (امتیازی):

- 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).

تاثیر تعداد epoch:

- 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 رخ می دهد را بیان کنید.
- (این بخش نیازی به پیاده سازی ندارد ولی می توانید برای پیدا کردن پاسخ سوالات از شبیه سازی کمک بگیرید)

تاثیر Loss Function:

- شبکه عصبی را با استفاده از تابع هزینه مشهور به MSE^3 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؟

تاثیر regularization (امتیازی):

- 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 dropout با مقدار 0.1 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 می توانید 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.

³ Mean squared error

نکات پایانی

- دقت کنید که هدف پروژه تحلیل نتایج و تاثیر عوامل مختلف 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.
- این تمرین شامل ۱۰ نمره امتیازی است؛ یعنی مجموع بارم بخش‌های این تمرین ۱۱۰ خواهد بود.
- 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 فایل‌های مورد نیاز برای اجرای آن باشد. تحلیل 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.
- در صورتی که سوالی در مورد پروژه داشتید بهتر است در فروم یا گروه درس مطرح کنید تا بقیه از آن استفاده کنند؛ در غیر این صورت با طراحان در ارتباط باشید.
- هدف از تمرین، یادگیری شماست. لطفاً تمرین را خودتان انجام دهید.

موفق باشید.