



مدرس: دکتر فدایی و دکتر یعقوبزاده

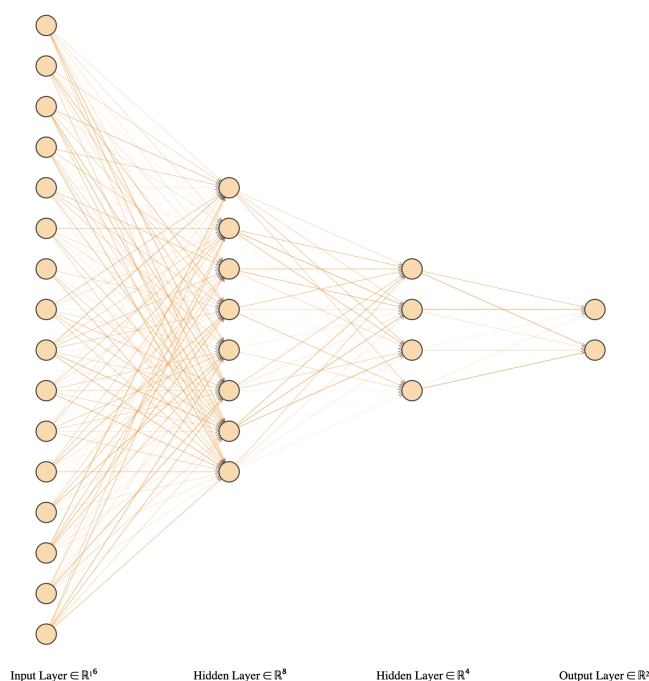
طراح: آرش هاتفی - حمید خدادادی - سینا نگارنده

مهلت تحویل: پنجشنبه 27 خرداد 1400

مقدمه

شبکه عصبی Feed Forward:

در فاز اول پروژه پنجم به پیاده سازی شبکه های عصبی Feed Forward جهت طبقه بندی تصاویر می پردازیم.



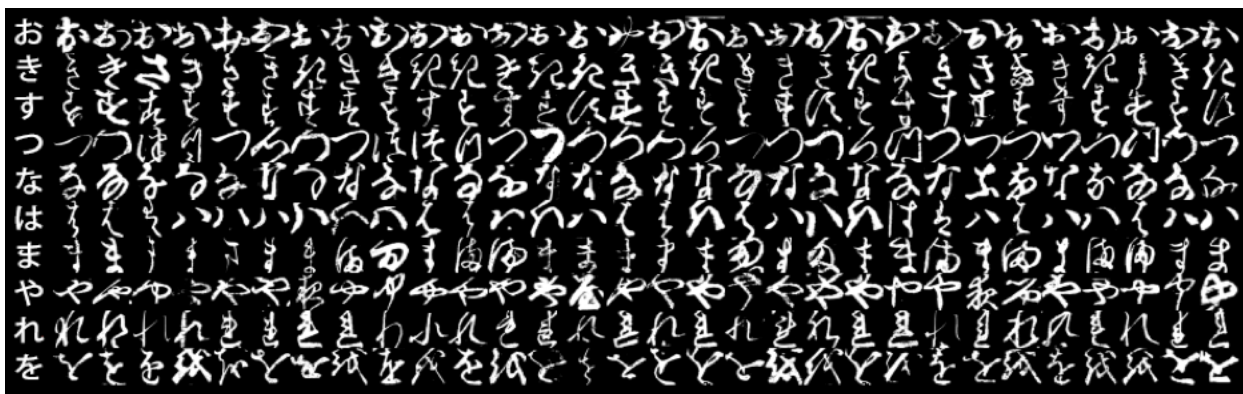
در شبکه های عصبی Feed Forward که در درس نیز با آن آشنا شدید، هر تصویر ابتدا مسطح شده و به صورت بردار به عنوان ورودی شبکه داده می شود. هر درایه این بردار (معادل با یک پیکسل تصویر) یک ویژگی برای آن محسوب می شود. شبکه قرار است بر اساس این ویژگی ها و با ساختن ترکیبات غیرخطی از آنها، وزن اتصالات بین لایه هایش را طوری تنظیم کند، که خروجی آن ضمن داشتن کمترین خطا، کلاس تصویر ورودی متناظر را به درستی پیش بینی 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 می‌گیرید 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 پارامترها را در فرایند یادگیری بررسی خواهید کرد.

معرفی مجموعه داده

در این تمرین شما با یک مجموعه داده [KMnist](#) (کوتاه شده Kuzushiji-MNIST) شامل تصاویری در مقیاس 28 در 28 پیکسل به صورت grayscale و 75,000 تصویر از 20 کلاس یا کاراکتر مختلف، معرف یک هجا در خط هیراگانا (Hiragana) که کلمات با استفاده از آن‌ها ساخته می‌شوند (شبیه به یک الفبا)، کار خواهید کرد. فایل داده‌ها در صفحه درس قرار داده شده است.



محتویات پوشه مجموعه داده به صورت زیر است.

kmnist_dataset

- └─ test_data.csv
- └─ test_labels.csv
- └─ train_data.csv
- └─ train_labels.csv

مجموعه به دو قسمت train و test تقسیم شده که به تعداد 60,000 آن برای آموزش و 15,000 برای تست قرار گرفته اند. هر سطر از فایل‌های داده‌ها، به صورت یک بردار از یک تصویر مسطح به اندازه‌ی 784 (28 در 28) است. فایل‌های برچسب‌ها شامل برچسب داده‌هاست. هر برچسب می‌تواند با یکی از اعداد 0 تا 20 نمایش داده شود که به نمایندگی از یک هجا می‌باشد.

فاز اول: بررسی و پیش‌پردازش داده

- یک تصویر را در مجموعه داده train را به صورت رندوم بررسی کنید و نشان دهید (از کتابخانه matplotlib استفاده کنید)، مقادیر هر پیکسل در چه محدوده‌ای قرار می‌گیرد؟
- در مجموعه داده train، از هر کلاس به دلخواه یک تصویر انتخاب کرده و نمایش دهید. برای هر تصویر، نوع آن را نیز به همراه تصویر نمایش دهید.
- تعداد تصاویر هر دسته را برای مجموعه داده train و test محاسبه کنید و برای آن‌ها نمودار میله‌ای رسم کنید.
- مقدار داده‌ها را به گونه 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؟

فاز 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 Stochastic Gradient Descent در طی فرایند آموزش به‌روزرسانی خواهد شد. بخش‌های حذف شده از کد که لازم است آن‌ها را کامل کنید، با #TODO مشخص شده‌اند. معرفی کلاس‌ها به همراه methodهای آنها و توضیحات تکمیلی راجع به فایل Notebook در فایل pdf کنار آن قرار گرفته است.

فاز سوم: طبقه بندی داده ها

در این بخش به پیاده سازی 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 پیاده سازی شبکه‌های عصبی آورده شده است.

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

1. مقدار learning rate باید 0.001 باشد.
2. تعداد epoch ها باید 20 باشد.
3. اندازه batch_size باید 64 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.)

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 پیاده سازی نیست، تنها توضیح دهید.

قسمت سوم) تاثیر learning rate

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

- حال با کاهش و افزایش این پارامتر، شبکه را آموزش دهید و پس از یافتن مقدار بهینه برای شبکه خود، نتیجه را گزارش کنید.
 - همچنین رفتار شبکه را برای learning rate با مقدار بالاتر (10 برابر) و پایین‌تر (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 به دست 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 Tanh و Sigmoid عملکرد مناسبی برای این دست شبکه‌ها ندارند را بیان کنید.

- برتری Leaky Relu نسبت به Relu چیست؟

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

قسمت پنجم) تاثیر batch size

- عملکرد شبکه را به ازای با batch size مقدار 16 و 32 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 احتمالی batch size بسیار کوچک را شرح دهید.

برای تمام قسمت‌های بعد، از batch size بهینه‌ای که به دست آورده‌اید، استفاده کنید.

قسمت ششم) تاثیر epoch

تعداد epoch هایی که یک شبکه ی عصبی به آن اندازه آموزش داده می شود، در نتیجه نهایی موثر است.

- دلیل اینکه نیاز است تا شبکه‌های عصبی برای چند epoch آموزش داده شوند را توضیح دهید؟
- آموزش شبکه‌های عصبی معمولاً تا زمانی ادامه پیدا می کند که overfitting شروع شود. حال تعداد epoch هایی که شبکه را با آن آموزش می دهید را تغییر دهید، تا زمانی که overfit رخ دهد. در این حالت نمودار های accuracy و loss را بر روی داده‌های آموزش و تست رسم نمایید. با توجه به نمودار، نتایج را مقایسه و تفسیر کنید.

- آیا همواره افزایش تعداد epoch های بیشتر مفید است؟

- در صورتی که این روش همواره مفید نیست، چه راه حلی برای پیش‌گیری از اتفاق نامطلوبی که روی می‌دهد دارید؟

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 zip. <#SID>-AI-CA5 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.
- در صورت نیاز می توانید از سرویس Google Colab استفاده کنید.
- هدف از تمرین، یادگیری شماسست. لطفا تمرین را خودتان انجام دهید.

موفق باشید.