



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

استاد درس: دکتر فاطمه شاکری

طراح پروژه: امید سقط چیان

زمستان ۱۴۰۲

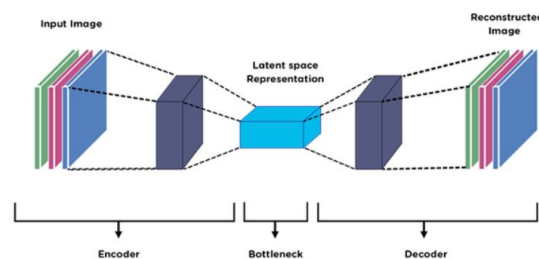
رمزگذار-رمزگشا با Pytorch

درس مباحثی در علوم کامپیوتر



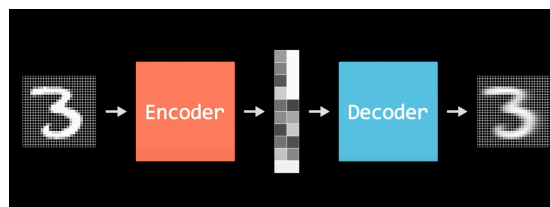
توضیحات اولیه

در دنیای یادگیری ماشین، ساختارهای encoder-decoder نقش مهمی را ایفا می‌کنند. این ساختارها اغلب برای تبدیل داده‌ها از یک فرمت به فرمت دیگر مورد استفاده قرار می‌گیرند، مانند ترجمه ماشینی که متنی از یک زبان به زبان دیگر تبدیل می‌کند. در یک ساختار encoder-decoder، بخش encoder وظیفه دارد تا داده‌های ورودی را دریافت و آن‌ها را به یک دنباله از ویژگی‌های فشرده (compressed features) تبدیل کند. سپس، decoder این نمایش میانی را می‌گیرد و سعی در بازسازی داده‌های اولیه دارد. به عکس زیر توجه کنید:



البته عکس‌هایی که ما با آن‌ها در این پروژه کار می‌کنیم سیاه سفید هستند و تنها یک کانال رنگی دارند

پروژه ما بر روی پیاده‌سازی این ساختار برای تصاویر MNIST، که شامل دست‌نویس‌های اعداد از ۰ تا ۹ است، تمرکز دارد. هدف از این پروژه، آموزش یک مدل encoder-decoder است تا بتواند تصاویر را به Latent Space Representation تبدیل کرده و در مرحله بعدی تصاویر را به درستی بازسازی کند. به مثال زیر توجه کنید:



ما مایل هستیم بجای اینکه عدد سه را در یک Tensor به ابعاد $28 * 28 * 1$ ذخیره کنیم، حافظه کمتری را مصرف کنیم در حالیکه که اطلاعات عدد ۳ را همچنان داشته باشیم. این وظیفه encoder است. در قسمت بعدی decoder با توجه به نمایش داده ما در Latent Space سعی در بازنمایی تصویر اصلی دارد که همان ۳ بود. حالا ما می‌خواهیم شبکه‌ای را بتوانیم آموزش دهیم که به درستی عملیات Encode و سپس Decode را انجام دهد.



نحوه خواندن داده ها

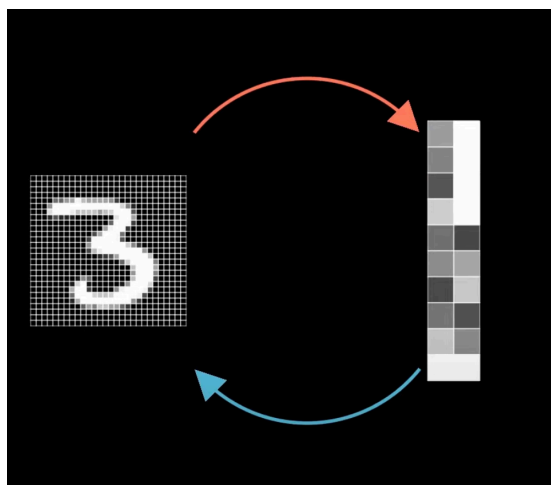
برای این قسمت باید از datasets خود pytorch استفاده کنید. دیتای مربوط به اعداد دست نویس را میتوانید از طریق آن دانلود کنید. همچنین برای قسمت آموزش شبکه انتظار میرود از dataloader pytorch استفاده کنید که داده ها را در هر مرحله از یادگیری به صورت رندوم بدون جایگذاری به شما بدهد. batch size را نیز برابر با ۳۲ در نظر بگیرید.

Encoder

شما در این قسمت باید از nn.Conv2d استفاده کنید که همان Convolution دو بعدی است. حداقل ۳ بار این تابع را به کار ببرید و بین هربار استفاده از تابع ReLU به عنوان فعالساز استفاده کنید. انتظار می رود شکل داده ورودی پس از دادن به شبکه Encoder از $28 * 28 * 1$ به $1 * 1 * 64$ تبدیل شود.

Decoder

شما در این قسمت باید از nn.ConvTranspose2d استفاده کنید که تقریباً میتوان گفت همان معکوس Convolution دو بعدی است. حداقل ۳ بار این تابع را به کار ببرید و بین هربار استفاده از تابع ReLU به عنوان فعالساز استفاده کنید. انتظار می رود شکل داده ورودی پس از دادن به شبکه Decoder از $1 * 1 * 64$ به $28 * 28 * 1$ تبدیل شود.



شکل ۱.۰: فلش بالایی همان عملیات Encode کردن و فلش پایینی عملیات Decode کردن میباشد



فرآیند یادگیری

انتظار می‌رود با توجه به توضیحات داده شده بتوانید تشخیص دهید خطا را باید به چه صورت تعریف کنید. همچنین می‌توانید از optimizer دلخواه خود استفاده کنید.

بصری سازی

قبل از اینکه شبکه شروع به یادگیری بکند برای ۱۰ نمونه تصویر نشان دهید تصویر اولیه و تصویر بازنمایی شده توسط شبکه به چه شکل است. در ادامه نشان دهید برای آن تصاویر پس از آنکه شبکه شروع به یادگیری کرده است، بازنمایی آن‌ها بهبود داشته است.

معیارهای ارزیابی - نکات ارسال

- از شما انتظار می‌رود تمامی قسمت‌ها را مطابق با توضیحات گفته شده پیاده سازی کنید. پیاده سازی که در چهارچوب گفته شده نباشد نمره ای نخواهد داشت.
- برای قسمت‌ها و اجزای مختلف پروژه خود به زبان فارسی داکيومنت بنویسید و عملکرد آن را کامل و دقیق شرح دهید.
- نمره شما از این پروژه پس از دادن ارائه در عددی بین ۰ تا ۱ بسته به میزان تسلط شما ضرب می‌شود.
- فایل ارسالی خود را به صورت فشرده و با فرمت نامگذاری project_3_idGroup.zip نامگذاری کنید. فایل فشرده شامل یک فایل ژوپیتِر نوت بوک^۱ و لینک گوگل کولب^۲ می‌باشد. تنها ارسال لینک گوگل کولب پذیرفته نیست.
- همچنین می‌توانید گزارش تمرین خود را به صورت یک فایل پی دی اف^۳ ارسال کنید. در اینصورت باید فایل کامل کد خود را همراه با فایل پی دی اف ارسال کنید و همچنین در توضیح هر بخش، عکسی از قطعه کدی که توضیح می‌دهید را در فایل پی دی اف قرار دهید. توضیحات فایل پی دی اف شما باید به زبان فارسی باشد.

مهلت تمرین

برای این پروژه تا ۲۱ اردیبهشت زمان دارید. در جلسه تدریسیاری به حل مشکلات شما خواهیم پرداخت.

¹Jupyter Notebook

²Google Colab

³PDF