



هدف این تمرین آشنایی با پیاده‌سازی لایه کانولوشن در شبکه‌های عصبی و پیاده‌سازی آن‌ها در کتابخانه پایتورچ است.

مهلت تحویل این تمرین پایان روز جمعه ۲۰ بهمن ۱۴۰۲ خواهد بود.

توجه: به دلیل طولانی بودن اجراها، تمرین دوم را بر روی [Google Colab](#) اجرا کنید. از منوی **Change Runtime Type**، گزینه **T4 GPU** را انتخاب نمایید.

۱- پیاده‌سازی

۱-۱: یک لایه Conv2d تشکیل دهید که بتواند عملیات کانولوشن با اندازه فیلتر $n * n$ (یک عدد فرد است) را با میزان padding مناسب، stride و تعداد فیلتر دلخواه بر روی تانسور ورودی اعمال کند. تصویر fruits که در پیوست تمرین است را بازگزاری کنید. این کار را میتوانیم با استفاده از کتابخانه OpenCV، Matplotlib و یا Pillow انجام دهید. سپس یک لایه کانولوشن با ۴ فیلتر تعریف کنید و نتیجه خروجی هر فیلتر را تک به تک با استفاده از کتابخانه Matplotlib نمایش دهید.

۱-۲: یک لایه MaxPooling را تشکیل دهید که با گرفتن اندازه پنجره و stride، عملیات Max Pooling را بر روی تانسور سه بعدی ورودی انجام دهد. سپس معماری زیر را پیاده‌سازی کرده و صرفاً عمل Forward را بر روی تصویر خوانده شده انجام دهید و خروجی را گزارش کنید. (نیازی به اعمال backward و بهینه‌سازی نیست.)

- یک لایه کانولوشنی با ورودی رنگی، ۴ فیلتر $3 * 3$ و پدینگ ۱ + batchNorm + تابع فعالسازی ReLU
- یک لایه کانولوشنی با ۸ فیلتر $3 * 3$ و پدینگ ۱ + تابع فعالسازی ReLU
- یک لایه Max Pooling با اندازه فیلتر $2 * 2$ و stride برابر ۲
- یک لایه کانولوشنی با ۱۶ فیلتر $3 * 3$ و پدینگ ۱ + batchNorm + تابع فعالسازی ReLU
- یک لایه کانولوشنی با ۱۶ فیلتر $3 * 3$ و پدینگ ۱ + تابع فعالسازی ReLU
- یک لایه Max Pooling با اندازه فیلتر $2 * 2$ و stride برابر ۲ و دراپ اوت با احتمال ۵ درصد
- یک لایه کانولوشنی با ۳۲ فیلتر $3 * 3$ و پدینگ ۱ + batchNorm + تابع فعالسازی ReLU
- یک لایه کانولوشنی با ۳۲ فیلتر $3 * 3$ و پدینگ ۱ + تابع فعالسازی ReLU
- یک لایه Max Pooling با اندازه فیلتر $2 * 2$ و stride برابر ۲
- دراپ اوت با احتمال ۱۰ درصد، یک لایه چگال با ۶۴ نورون و تابع فعالساز ReLU
- یک لایه چگال با ۳۲ نورون + تابع فعالساز ReLU + دراپ اوت با احتمال ۱۰ درصد
- لایه دسته‌بندی نهایی با ۲ نورون

۲- پایتورچ

۱-۲: مجموعه داده ارقام دست نویس را در نظر بگیرید. این مجموعه داده را با استفاده از کتابخانه torch vision فراخوانی کنید. دو شبکه کانولوشنی مناسبی طراحی کنید تا عمل دسته‌بندی را انجام دهند. در هر تکرار صحت (Accuracy) را بر روی مجموعه داده آزمایش بدست آورید. در پایان منحنی خطا-تکرار و دقت-تکرار همه‌ی حالات را در یک نمودار رسم کرده و این نمودار را تحلیل کنید. بهترین حالت کدام است؟ علت چیست؟ در تمرین گذشته ۲- بهترین نتیجه خود را آورده و صحت آن‌ها را با شبکه‌های کانولوشنی جدید مقایسه کنید.

۲-۲: در اکثر مسئله‌های دنیای واقعی، آموزش مدل به دلیل کمبود سخت‌افزار، کمبود زمان، تعداد کم داده دارای چالش است. یکی از بهترین راه حل‌های این چالش‌ها، استفاده از مدل‌های از پیش آموزش دیده و آموزش مجدد آن بر روی مجموعه داده جدید است که به آن Fine-Tuning گفته می‌شود. مدل از پیش آموزش دیده شده MobileNet V2 (Pre-Train) را بارگذاری کرده و آن را Fine-Tune کنید. نتیجه را با نتایج بخش قبل مقایسه کنید. مدل‌های از پیش آموزش دیده موجود برای دسته‌بندی در کتابخانه Pytorch و نحوه فراخوانی آنها را می‌توانید در این [لینک](#) مشاهده کنید.

مواردی که باید تحویل داده شود:

- فایل (یا فایل‌های) کد برنامه مورد استفاده برای پیاده‌سازی تمرین با فرمت ipynb. در یک پوشه به نام Code
- فایل گزارش با نام Doc.pdf

فایل‌های کد و گزارش را به صورت یک فایل فشرده در قالب zip و با نام NN_Name_Family_HW4 در سایت کوئرا بارگذاری کنید (به جای Name نام و به جای Family نام خانوادگی خود را قرار دهید).

مهلت تحویل این تمرین تا پایان روز جمعه 20 بهمن خواهد بود.

موفق باشید