

بسمه تعالی

تمرین سری سوم درس یادگیری عمیق دکتر محمد تشنه لب

لطفا در حل تمرین به موارد زیر توجه داشته باشید:

- تمرین را خودتان انجام بدهید و به کاری که ارائه می کنید، تسلط کامل داشته باشید.
- پیاده سازی تمرین را با زبان برنامه نویسی پایتون انجام دهید. بهتر است از فرمت notebook (ipynb) استفاده کنید تا نتایج اجرا در کنار کد قابل مشاهده باشد. محیط google colab برای این کار مناسب است.
- به طور کلی، استفاده از کتابخانه های آماده مجاز نیست.
- فایل ارسالی باید به صورت فشرده باشد و شامل یک فایل گزارش به صورت PDF یک فولدر شامل تمامی کدهای پیاده سازی شده و دیتاست های لازم، گزارش باید کامل و در عین حال مختصر و مفید باشد. گزارش باید شامل روابط ریاضیاتی پیاده سازی شده، پارامترهای شبکه مثل (تعداد ایپاک، نرخ یادگیری و ...)، پاسخ لازم سوالات پرسیده شده، ساختار شبکه (هرگونه جزئیات پیاده سازی) و نتایج اجرای کد (به همراه مقایسه نتایج با هم در صورت نیاز) باشد.
- نام فایل ارسالی باید به صورت HW3_Name_StudentNumber باشد. همچنین اسم (اسامی) و شماره دانشجویی خود را در ابتدای گزارش و در ابتدای کد به صورت کامنت بنویسید.
- نمره هر تمرین از ۳ بخش تشکیل می شود: گزارش، کد، ارائه. پس از تحویل هر تمرین، تمامی اعضای گروه باید در تمامی بخش های انجام شده مشارکت داشته باشند. تمرین را می توان به صورت گروه های ۲ نفره انجام داد. فایل نهایی را تنها یک نفر به نمایندگی در سامانه آپلود کند. دانشجویان دکتری باید به صورت تک نفره کار کنند.
- تمرین به صورت متنوع و گلچین آماده شده است تا با چالش های جدید و حتی مطرح نشده در کلاس روبرو شوید. هدف، افزایش قدرت تحلیل و حل مسئله شماست. در صورت مشاهده کدها و گزارش های مشابه، از هر دو گروه نمره کسر می گردد

بخش اول: سوالات تشریحی

سوال اول: تا الان متوجه شده اید که طراحی شبکه عصبی چند لایه پرسپترون برای دسته بندی چه چالش هایی داشته و با تکیه بر سعی و خطا می توان به یک مدل مناسب دست پیدا کرد. حال در شبکه های CNN علاوه بر طراحی دسته بند، بایستی قسمت استخراج ویژگی نیز طراحی شود. به نظر شما روند طراحی همزمان دسته بند و استخراج ویژگی بایستی چگونه انجام پذیرد تا بتوان به یک مدل CNN مناسب دست یافت؟

سوال دوم: در یک مسئله طبقه بندی تصویر، خروجی آخرین لایه کانولوشن یک شبکه CNN دارای ابعاد $8 \times 8 \times 256$ است. دو طراح شبکه، دو راهکار متفاوت برای بخش انتهایی شبکه پیشنهاد می کنند:

طراح اول: خروجی را پس از تخت سازی (Flatten) به یک لایه Fully Connected با ۱۰ خروجی متصل می کند .

طراح دوم: ابتدا Global Average Pooling (GAP) استفاده کرده و سپس خروجی را به یک لایه Fully Connected با ۱۰ خروجی متصل می کند.

به سؤالات زیر پاسخ دهید:

الف) Global Average Pooling را توضیح دهید و نشان دهید چگونه خروجی $8 \times 8 \times 256$ را به یک بردار ویژگی تبدیل می کند .

ب) ابعاد خروجی حاصل از GAP را مشخص کنید .

پ) تعداد پارامترهای لایه Fully Connected در هر دو روش را محاسبه و مقایسه کنید (بایاس را در نظر نگیرید) .

ج) توضیح دهید چرا استفاده از GAP می تواند احتمال بیش برآزش را کاهش دهد .

د) فرض کنید در تصاویر آزمون، شیء مورد نظر نسبت به تصاویر آموزشی کمی جابه جا شده است. توضیح دهید چرا استفاده از GAP می تواند نسبت به این جابه جایی مقاوم تر از روش Flatten + Fully Connected باشد .

ه) به نظر شما در چه شرایطی استفاده از GAP ممکن است باعث از دست رفتن بخشی از اطلاعات مفید شود؟

سوال سوم: اتصال باقیماندگی (Residual Connection) در CNN ها برای رفع چه مشکلی پیشنهاد شد؟ با ذکر دلیل بیان کنید در چه شرایطی طول اتصال باقیماندگی بهتر است بلند باشد و در چه شرایطی کوتاه؟

سوال چهارم: هدف اصلی Pooling چیست؟ تفاوت Max Pooling و Average Pooling بیان کنید و بگویید چرا Max Pooling در بسیاری از کاربردهای بینایی محبوبتر است؟

بخش دوم: مقایسه نتایج پیاده سازی CNN با استفاده از کتابخانه و بدون کتابخانه

در این بخش ابتدا با استفاده از کتابخانه‌های PyTorch یا TensorFlow یک شبکه عصبی پیچشی طراحی و پیاده‌سازی کنید. شبکه باید حداقل شامل دو لایه کانولوشن و دو لایه تمام‌متصل باشد. انتخاب تعداد دقیق لایه‌ها و همچنین ابرپارامترهای شبکه مانند نرخ یادگیری، تابع فعال‌ساز، اندازه دسته‌ها، بهینه‌ساز، اندازه فیلترها و سایر تنظیمات بر عهده شماست.

دیتاست‌های مورد استفاده شامل Fashion-MNIST و CIFAR-10 و flower17 هستند. پس از آموزش مدل، لازم است مقادیر دقت، مقدار تابع هزینه و نمودارهای مورد نیاز گزارش شوند.

در مرحله بعد، دقیقاً همین شبکه عصبی را از ابتدا و بدون استفاده از کتابخانه‌های آماده یادگیری عمیق پیاده‌سازی کنید. در این بخش باید تمامی روابط ریاضی، عملگرهای کانولوشن، توابع فعال‌ساز، و مراحل پس‌انتشار خطا (Backpropagation) را شخصاً طراحی و پیاده‌سازی نمایید.

الف) نتایج حاصل از پیاده‌سازی با استفاده از کتابخانه و بدون استفاده از کتابخانه را به طور کامل با یکدیگر مقایسه کنید. در صورتی که با وجود یکسان بودن تمامی شرایط، اختلافی در نتایج مشاهده شد، علت‌های احتمالی این اختلاف را تحلیل و توضیح دهید.

ب) استخراج و تشخیص لبه‌ها در تصاویر یکی از مراحل مهم در پردازش تصویر است. در شبکه‌های عصبی پیچشی نیز نقشه‌های ویژگی (Feature Maps) حاصل از لایه اول کانولوشن معمولاً الگوهایی مشابه لبه‌ها را یاد می‌گیرند. با استفاده از مصورسازی خروجی‌های لایه کانولوشن اول برای یک تصویر دلخواه از مدل آموزش‌دیده، این موضوع را نشان دهید و تحلیل کنید.

نکات:

در هر دو آزمایش (با کتابخانه و بدون کتابخانه)، تمامی شرایط شامل تعداد لایه‌ها و مقادیر ابرپارامترها باید یکسان باشند.

در هر دو پیاده‌سازی استفاده از Batch Normalization و Dropout الزامی است.

در هر دو آزمایش باید منحنی تابع هزینه در طول آموزش رسم شود.

همچنین ماتریس درهم‌ریختگی (Confusion Matrix) برای داده‌های آموزش و آزمون ارائه گردد.

معیارهای Accuracy و F1-score برای هر دو مجموعه داده گزارش شوند.

از کل داده‌ها، ۷۰ درصد برای آموزش، ۱۰ درصد برای اعتبارسنجی و ۲۰ درصد برای آزمون استفاده شود.

شکل گرافیکی شبکه نهایی شده برای هر دیتاست را رسم کنید.

نوشتن تمامی روابط پیشرو و پسرو الزامی است.

بخش سوم: استفاده از معماری Inception

در این تمرین، هدف طراحی، پیاده‌سازی و ارزیابی یک معماری کانولوشنی جهت طبقه‌بندی تصاویر مجموعه داده Linnaeus 5 (نسخه ۱۲۸×۱۲۸) است. این مجموعه داده را می‌توانید از این [لینک](#) دریافت نمایید. دقت کنید که مجموعه داده‌ی آموزشی و آزمون تفکیک شده است. همچنین برای پایش وضعیت همگرایی و جلوگیری از بیش‌برازش، تخصیص ۱۰٪ از داده‌های آموزشی به عنوان مجموعه اعتبارسنجی الزامی است.

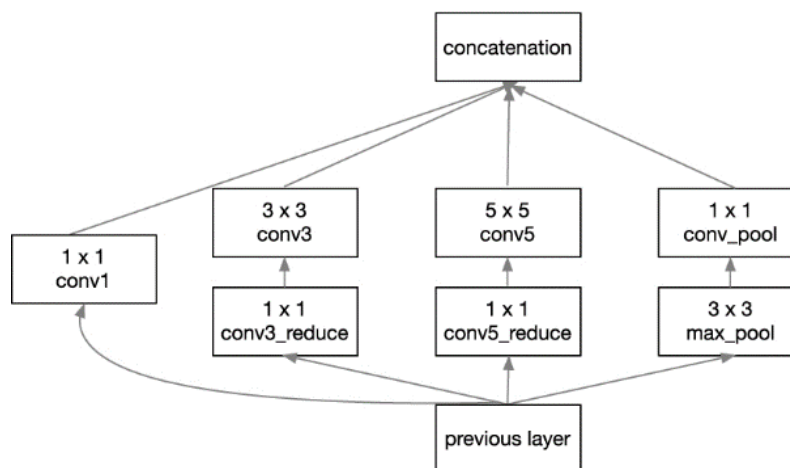
الف) معماری CNN مناسب خود را طراحی کنید. در انتخاب ابرپارامترها، نرخ یادگیری، توابع فعال‌ساز و استراتژی‌های منظم‌سازی دقت کافی به خرج دهید. سپس معماری خود را با افزودن اتصالات میان‌بر (Residual Connections) ارتقا دهید. تاثیر این تغییر را بر پویایی گرادیان، نرخ همگرایی و معیارهای نهایی عملکرد (نظیر Accuracy و F1-score) تحلیل و گزارش نمایید.

ب) در طراحی معماری شبکه‌های عصبی کانولوشنی انتخاب سایز فیلتر (3×3 ، 5×5 یا 7×7) یکی از چالش‌های بنیادین در فرآیند استخراج ویژگی است. فیلترهای کوچک‌تر، میدان دید محدودی دارند که منجر به تمرکز بر ویژگی‌های محلی و جزئیات ریز بافت تصویر می‌شود؛ در حالی که فیلترهای بزرگ‌تر، با گسترش میدان دید، قابلیت ادراک الگوهای کلان‌تر و ویژگی‌های سراسری را فراهم می‌کنند.

با توجه به ماهیت تصاویر در دنیای واقعی و تفاوت در ابعاد و مقیاس سوژه‌ها استفاده از یک فیلتر با ابعاد ثابت، محدودیت‌های جدی در استخراج ویژگی‌های بهینه ایجاد می‌کند. برای تحلیل دقیق‌تر، به نمونه‌های ارائه‌شده از تصاویر گربه توجه کنید:



در تصاویری که سوژه بخش عمده‌ای از قاب تصویر را اشغال کرده است (مانند تصویر سمت راست)، فیلترهای بزرگ‌تر با استخراج الگوهای ساختاری و کلی، بازنمایی مؤثرتری از شیء ایجاد می‌کنند. در مقابل، برای تصاویری که سوژه دارای ابعاد کوچکی است (مانند تصویر سمت چپ)، فیلترهای کوچک‌تر با تمرکز بر ویژگی‌های جزئی و محلی، عملکرد مطلوب‌تری در تشخیص الگوهای متمایزکننده ارائه می‌دهند. این تضاد در نیاز به مقیاس‌های مختلف برای استخراج ویژگی، ضرورت استفاده از معماری‌هایی نظیر Inception را تبیین می‌کند که با به‌کارگیری فیلترهایی با ابعاد متنوع در یک لایه، امکان استخراج همزمان ویژگی‌های چندمقیاسی را فراهم می‌سازند.



بهترین مدلی که در قسمت قبل به دست آورده اید را انتخاب کنید و تمامی لایه های کانولوشنی با اندازه فیلتر ثابت آن را با ساختار فوق جایگزین کنید و مدل خود را مجدد آموزش داده و نتایج را با قسمت قبل مقایسه کنید. استفاده از کتابخانه مجاز است و نوشتن تمامی روابط پیشرو و پسرو الزامی است.

بخش چهارم: یادگیری انتقالی

آموزش شبکه های عصبی عمیق همواره با چالش های هزینه های محاسباتی بالا و نیاز به حجم داده های زیاد همراه است. یادگیری انتقالی به عنوان یک پارادایم کلیدی، با بهره گیری از دانش اندوخته شده در مدل های پیش آموزش دیده راهکاری کارآمد برای حل این محدودیت ها است. در این رویکرد، بخش استخراج ویژگی مدل پیش آموزش دیده حفظ شده و دسته بند (Classifier) اصلی حذف می شود؛ سپس یک دسته بند جدید جایگزین می گردد. فرآیند آموزش معمولاً طی دو مرحله انجام می شود:

۱. فاز انجماد (Freezing): وزن های لایه های استخراج ویژگی ثابت نگه داشته شده و تنها دسته بند جدید آموزش می بیند.

۲. فاز تنظیم دقیق (Fine-tuning): لایه ها به صورت تدریجی (از لایه های نهایی به سمت لایه های ابتدایی) آزاد شده و کل مدل با نرخ یادگیری پایین به روزرسانی می شود.

الف) استدلال فنی خود را در خصوص ضرورتِ غیرقابل آموزش بودن وزن های لایه های استخراج ویژگی در گام اول یادگیری انتقالی بیان کنید. چرا بلافاصله پس از اتصال دسته بند جدید، کل شبکه را به روزرسانی نمی کنیم؟

ب) از مدل حاصل در بخش سوم قسمت ب برای انتقال یادگیری استفاده کرده و مجموعه تصاویر flower را دسته بندی کنید. دیتاست را از لینک زیر دانلود کنید. این تصاویر دارای ابعاد مختلفی می باشند لذا نیاز است تصاویر را هم سایز و متناسب با ورودی شبکه کنید.

https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz

به نظر شما پیچیدگی دسته بند جدید چه تاثیری در روند و گام های انتقال یادگیری دارد و بایستی با چه مکانیزمی معماری دسته بند جدید طراحی شود؟

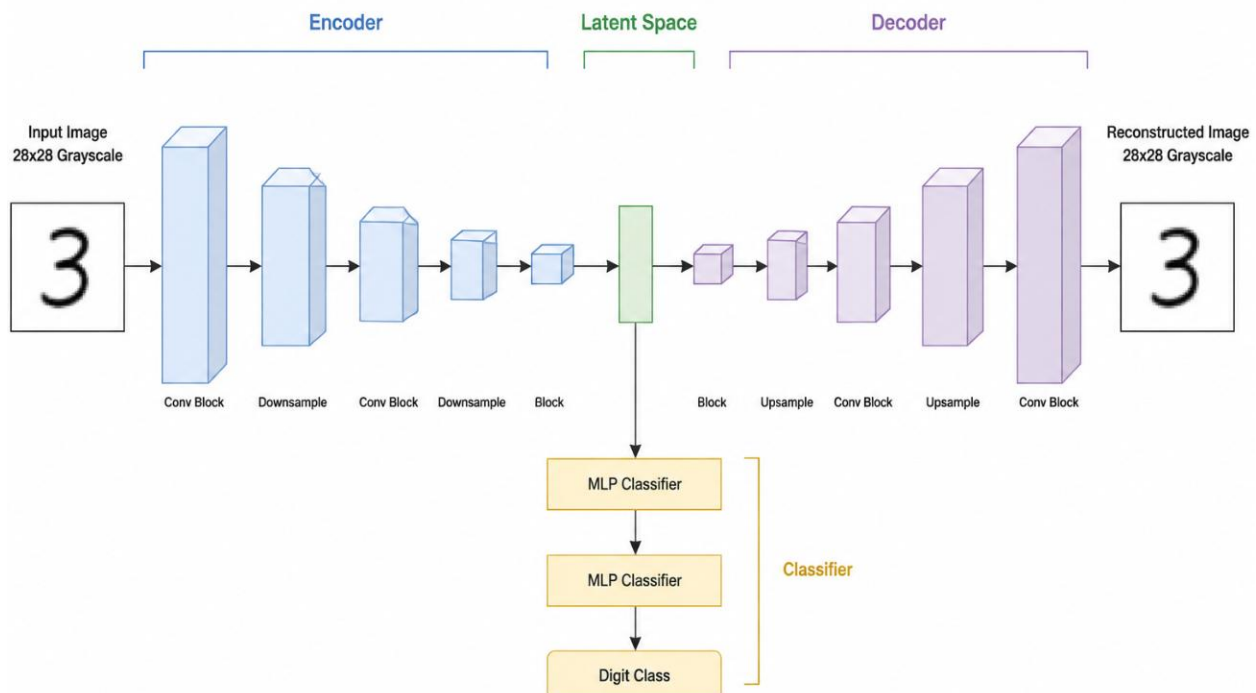
استفاده از کتابخانه مجاز است و نوشتن تمامی روابط پیشرو و پسرو الزامی است.

بخش پنجم: اتوانکدرهای پیچشی عمیق (Deep Convolutional Denoising Autoencoders)

مشابه سوال دوم بخش دوم تمرین اول، از مجموعه داده MNIST استفاده نمایید. فرآیند آغشته‌سازی تصاویر به نویز باید دقیقاً با همان پارامترها و شدتی انجام شود که در تمرین اول اعمال کردید تا امکان مقایسه عادلانه فراهم باشد. برخلاف تمرین قبل، از Flat کردن تصاویر خودداری کرده و داده‌ها را به صورت تانسورهای دوبعدی وارد شبکه کنید. یک Stacked Autoencoder مبتنی بر لایه‌های پیچشی طراحی کنید که بخش Encoder حداقل شامل ۳ لایه کانولوشن و بخش Decoder شامل لایه‌های Transposed Convolution جهت بازسازی معکوس تصویر و بخش Classifier نیز یک شبکه MLP دو لایه به انتهای آن جهت طبقه‌بندی باشد. ابتدا اتوانکدر را به صورت محلی آموزش دهید سپس آموزش را در دو مسیر مختلف ادامه دهید: مسیر اول: تنها وزن‌های کلاسیفایر آموزش ببینند و وزن‌های انکدر فریز شود.

مسیر دوم: در هنگام آموزش وزن‌های کلاسیفایر وزن‌های کانولوشن نیز مجدداً آموزش ببینند.

نتایج حاصل از این تمرین (معماری CNN) را با نتایج تمرین اول به دقت مقایسه کنید.



نکات:

- رعایت ثبات در ابرپارامترها (مانند اندازه Batch و تعداد Epoch) برای معتبر بودن مقایسه بین دو تمرین الزامی است.
- گزارش باید شامل جداول مقایسه‌ای دقیق برای Accuracy و Loss در هر دو مدل (قدیم و جدید) باشد.
- نوشتن روابط پیشرو و پسرو الزامی است.
- پیاده سازی این تمرین بدون استفاده از کتابخانه امتیاز مثبت دارد.