تمرین دوم

سوال اول: تحلیل گرادیان و پیاده‌سازی لایه کانولوشن سفارشی (تحت PyTorch)

در جزوه، فرآیند کانولوشن و پس‌انتشار (Backpropagation) در شبکه‌های CNN به تفصیل بررسی شده است. فرض کنید در یک پروژه پردازش تصویر پزشکی، نیاز به طراحی یک لایه کانولوشن جدید داریم که در آن، وزن‌های فیلتر $W \in \mathbb{R}^{K\times K}$ پیش از اعمال روی ورودی، به صورت درایه‌ـبه‌ـدرایه (Element-wise) در یک ماتریس ماسکِ قابل یادگیری $M \in \mathbb{R}^{K\times K}$ ضرب می‌شوند (یعنی فیلتر نهایی برابر با $W \odot M$ است).

۱. با فرض یک ورودی تک‌کاناله، استراید برابر ۱ و بدون پدینگ (Padding $= 0$)، روابط ریاضی دقیق برای گرادیان‌های $\frac{\partial L}{\partial W}$ و $\frac{\partial L}{\partial M}$ را استخراج کنید (L تابع خطای MSE است).

۲. این لایه را با استفاده از عملیات unfold در PyTorch پیاده‌سازی کنید (از nn.Conv2d به طور مستقیم استفاده نکنید). یک Tensor تصادفی ورودی به ابعاد (1, 1, 5, 5) و یک هدف به ابعاد (1, 1, 3, 3) بسازید. یک فرآیند آموزش (یک Forward و Backward با خطای MSE) روی این داده‌های سنتز شده اجرا کنید و نشان دهید که مقادیر گرادیان‌های به دست آمده برای W و M صحیح است و شبکه به درستی کار می‌کند.

سوال دوم: معماری Bottleneck در ResNet و مدیریت منابع محاسباتی

با توجه به مبحث ResNet و نقش بلوک‌های Bottleneck در کاهش بار محاسباتی (کاهش بعد با کانولوشن 1×1، انجام پردازش فضایی با 3×3 و افزایش مجدد بعد با 1×1):

۱. فرض کنید یک بلوک Bottleneck با ورودی به ابعاد (Batch=1, Channels=256, H=14, W=14) داریم. ساختار پایه شامل کانولوشن‌های 1×1 (با ۶۴ فیلتر)، 3×3 (با ۶۴ فیلتر) و 1×1 (با ۲۵۶ فیلتر) است. تعداد دقیق پارامترهای این بلوک (با فرض عدم استفاده از Bias و Batch Normalization) را به صورت تحلیلی محاسبه کنید.

۲. اگر برای اجرای این مدل روی یک سخت‌افزار لبه (Edge Device) با محدودیت شدید حافظه، نیاز باشد تعداد پارامترهای این بلوک کاهش یابد و تصمیم بگیریم تعداد فیلترهای میانی را از ۶۴ به ۳۲ کاهش دهیم، درصد دقیق کاهش پارامترهای کل بلوک را محاسبه کنید. این تغییر چه تاثیری بر میدان دریافت (Receptive Field) و ظرفیت استخراج ویژگی مدل خواهد داشت؟

۳. هر دو نسخه از این بلوک (۶۴ فیلتری و ۳۲ فیلتری) را در PyTorch با استفاده از nn.Conv2d پیاده‌سازی کرده و با ارسال یک تانسور ورودی تصادفی سنتز شده، درستی محاسبات تحلیلی خود را در کد (با چاپ خروجی متد numel() برای هر دو مدل و نمایش یکسان بودن Shape خروجی‌ها) اثبات کنید.

سوال سوم: معماری YOLO و ترکیب با معیار Soft Dice Score

در جزوه، سیستم YOLO به عنوان یک رگرسور یکپارچه و همچنین معیار Dice Score به عنوان یک تابع خطای پیشرفته معرفی شده‌اند. در YOLO کلاسیک از خطای مربعات (MSE) برای پیش‌بینی ابعاد Bounding Box استفاده می‌شود که در شرایطی که اشیاء مقیاس متفاوتی دارند، عملکرد ایده‌آلی ندارد.

۱. با الهام از مفهوم Soft Dice Score در جزوه، یک تابع خطای سفارشی ترکیبی برای YOLO پیشنهاد دهید که در آن برای بخش مکان‌یابی (Localization)، به جای خطای MSE از نسخه‌ای مشتق‌پذیر از Dice Loss (با

مدل‌سازی مساحت Bounding Boxها) استفاده شود. فرمول‌بندی ریاضی این Loss Function را به صورت دقیق بنویسید و توضیح دهید چرا برای اشیاء با اندازه‌های بسیار متفاوت، بهتر از MSE عمل می‌کند.

۲. تابع خطای پیشنهادی خود را در PyTorch پیاده‌سازی کنید.

۳. یک سناریوی تست با تانسورهای پیش‌بینی (Prediction) و هدف (Target) سنتز شده برای یک گرید بسیار کوچک ($S = 2$) و فرض یک جعبه به ازای هر سلول ($B = 1$) طراحی کنید. خروجی تابع خطای خود را برای دو حالت محاسبه و چاپ کنید: حالت الف) پیش‌بینی کاملاً با هدف منطبق است. حالت ب) پیش‌بینی هیچ هم‌پوشانی با هدف ندارد. نشان دهید که کد شما مقادیر منطقی و مورد انتظار را برمی‌گرداند.

سوال چهارم: تلفیق ویژگی‌های Denoising و Contractive Autoencoders

جزوه به دو نوع اتوانکودر تنظیم‌شده (Regularized Autoencoders) شامل DAE (حذف‌کننده نویز) و CAE (انقباضی) اشاره دارد. فرض کنید در حال طراحی یک سیستم تشخیص ناهنجاری (Anomaly Detection) برای سیگنال‌های زمانی حساس هستید که هم در برابر نویزهای ورودی مقاوم باشد و هم مشتقات فضای پنهان (Latent Space) نسبت به ورودی را به شدت کنترل کند تا تغییرات کوچک باعث جابجایی‌های نامعقول در فضای ویژگی نشود.

۱. تابع هدف جامع برای شبکه‌ای که به طور همزمان مکانیزم DAE و CAE را پیاده‌سازی می‌کند، فرموله کنید. نقش هر یک از ترم‌های این Loss Function را به صورت ریاضی تحلیل کنید.

۲. محاسبه ماتریس ژاکوبین (Jacobian) برای ترم CAE معمولاً از نظر محاسباتی با استفاده از Autograd بسیار سنگین است. یک معماری ساده یک‌لایه برای این اتوانکودر در PyTorch طراحی کنید که در آن، ترم جریمه نرم فروبنیوس (Frobenius Norm) ژاکوبین به صورت تحلیلی با استفاده از ویژگی‌های مشتق تابع فعال‌ساز Sigmoid با کارایی بالا محاسبه شود.

۳. با تولید یک مجموعه داده سنتز شده بسیار کوچک شامل ۱۶ نمونه (هر نمونه یک بردار ۲۰ بعدی شامل سیگنال سینوسی با نویز گاوسی)، یک حلقه آموزش (Training Loop) کوتاه برای ۵۰ Epoch بنویسید که این مدل را بهینه‌سازی کند. سپس یک نمونه کاملاً ناهنجار (تولید شده توسط نویز یکنواخت) به شبکه داده و خطای بازسازی (Reconstruction Error) آن را به عنوان امتیاز ناهنجاری (Anomaly Score) محاسبه و چاپ کنید تا نشان دهید شبکه متوجه تفاوت الگو می‌شود.