

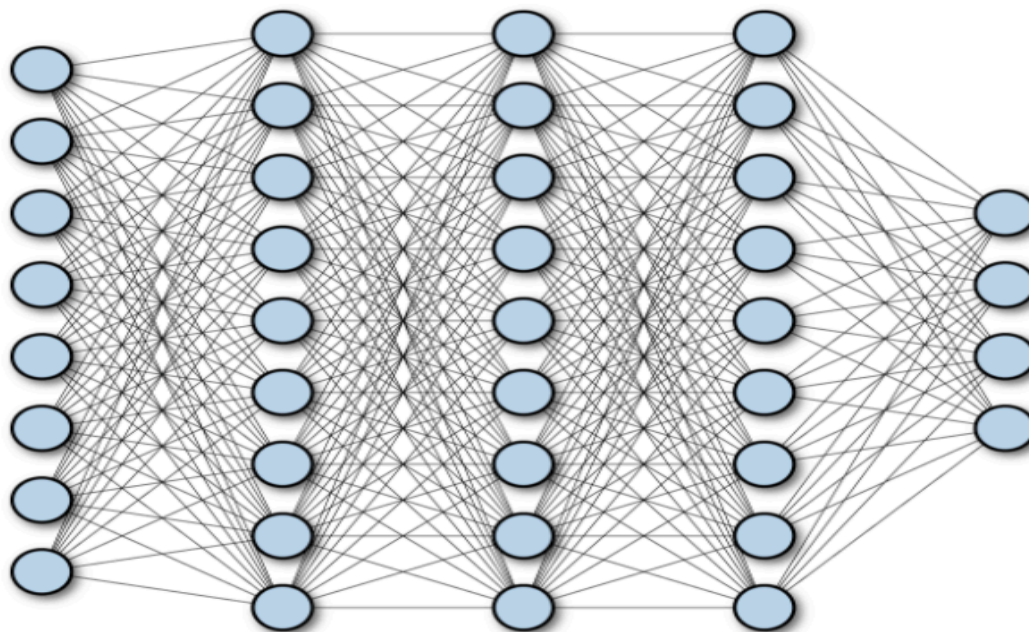


مهلت تحویل: دوشنبه ۱۲ خرداد ماه ۱۴۰۴، ساعت ۲۳:۵۹

مقدمه

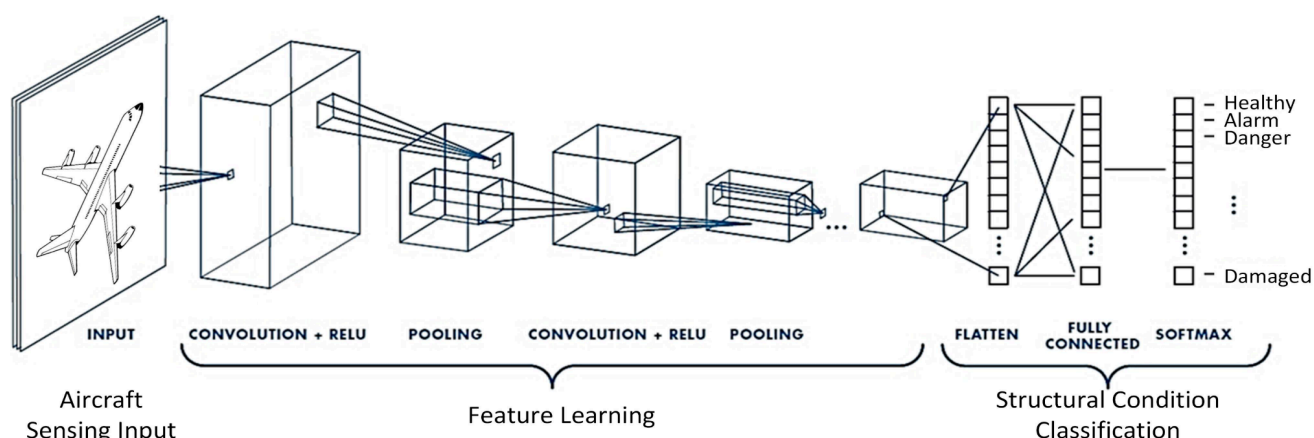
شبکه‌های عصبی Fully Connected

در شبکه‌های عصبی Fully Connected که در درس نیز با آن آشنا شدید، هر تصویر ابتدا مسطح شده و به صورت بردار به عنوان ورودی شبکه داده می‌شود. هر درایه این بردار (معادل با یک پیکسل تصویر) یک ویژگی برای آن محسوب می‌شود. شبکه قرار است بر اساس این ویژگی‌ها و با ساختن ترکیبات غیرخطی از آن‌ها، وزن اتصالات بین لایه‌هایش را طوری تنظیم کند، که خروجی آن ضمن داشتن کمترین خطا، کلاس تصویر ورودی متناظر را به درستی پیش‌بینی کند. شکل زیر ساختار کل این نوع از شبکه‌ها را نشان می‌دهد:



شبکه‌های عصبی پیچشی

شبکه‌های عصبی پیچشی (Convolutional Neural Networks) نوعی مدل یادگیری عمیق هستند که به طور گسترده و به خصوص برای پردازش تصویر و ویدیو استفاده می‌شوند (شکل ۱). این مدل‌ها برای تجزیه و تحلیل و شناسایی موثر الگوهای بصری با استفاده از مفهوم Convolution طراحی شده‌اند. CNN ها معمولاً از چندین لایه شامل لایه‌های پیچشی^۱، لایه‌های ادغام^۲ و لایه‌های کاملاً متصل^۳ تشکیل می‌شوند. لایه‌های کانولوشن، فیلترهایی را روی تصویر ورودی اعمال می‌کنند و ویژگی‌ها را در مکان‌های مختلف تصویر یا ویدیو استخراج می‌کنند. لایه‌های ادغام ابعاد فضایی ویژگی‌ها را کاهش می‌دهند و لایه‌های کاملاً متصل بر اساس ویژگی‌های استخراج شده، عملیات طبقه‌بندی و یا رگرسیون مورد نظر را انجام می‌دهند. این معماری سلسله مراتبی، CNN ها را قادر می‌سازد تا الگوهای بصری پیچیده را به طور خودکار یاد بگیرند. این قابلیت مدل‌های پیچشی، آن‌ها را در کارهایی مانند طبقه‌بندی تصویر، تشخیص اشیا^۴ و بخش‌بندی تصویر^۵ بسیار پر قدرت می‌سازد. تصویر زیر ساختار کلی یک شبکه عصبی پیچشی را نشان می‌دهد:



¹ Convolutional Layers

² Pooling Layers

³ Fully Connected Layers

⁴ Object Detection

⁵ Image Segmentation

تعریف مسئله

در این تمرین، شما ابتدا با استفاده از یک شبکه عصبی fully connected و سپس با استفاده از یک شبکه عصبی پیچشی به طبقه‌بندی تصاویر با استفاده از کتابخانه **PyTorch** می‌پردازید و در نهایت نتایج حاصل از این دو شبکه را با یکدیگر مقایسه می‌کنید. پیشنهاد می‌شود قبل از انجام پروژه، نوتبوک مربوط به آشنایی با **PyTorch** را مطالعه کنید تا آشنایی لازم با این کتابخانه را بدست آورید. همچنین برای انجام پروژه از نوتبوک نیمه آماده ای که در اختیارتان قرار گرفته است استفاده کنید.

برای راحتی استفاده از کتابخانه‌ها و تسریع فرایند آموزش، توصیه می‌شود که از سرویس **Google Colab** استفاده کنید. توجه کنید که آموزش شبکه‌های عمیق (به خصوص شبکه‌های CNN) بر روی GPU بسیار سریع‌تر از CPU می‌باشد. در نتیجه توصیه می‌شود که در صورت استفاده از سرویس **Google Colab** نوع **Runtime Type** را بر روی GPU قرار دهید. دقت کنید که مدت زمان استفاده از GPU به ازای هر اکانت محدود می‌باشد، در نتیجه فقط در هنگام آموزش مدل از GPU استفاده کنید تا مدت زمان اختصاص یافته برای شما به اتمام نرسد.

آشنایی با مجموعه داده

مجموعه داده مورد استفاده در این تمرین، **CIFAR-10** است. **CIFAR-10** یکی از مجموعه داده‌های پرکاربرد در حوزه یادگیری ماشین و بینایی کامپیوتر است که برای آموزش و ارزیابی الگوریتم‌های طبقه‌بندی تصویر مورد استفاده قرار می‌گیرد. این مجموعه شامل 60,000 تصویر رنگی با ابعاد 32×32 پیکسل است که به 10 کلاس مختلف تقسیم‌بندی شده‌اند و هر کلاس در مجموع شامل 6000 تصویر است. این 10 کلاس عبارتند از:

- Airplane
- Automobile
- Bird
- Cat
- Deer
- Dog
- Frog
- Horse
- Ship
- Truck

این مجموعه داده به دو بخش تقسیم می‌شود:

- داده‌های آموزش: 50,000 تصویر برای آموزش مدل‌ها
- داده‌های تست: 10,000 تصویر برای ارزیابی عملکرد مدل‌ها

داده‌های train و test این مجموعه داده برای شما در نوتبوک قرار داده شده است و همچنین داده train به دو بخش validation و train شکسته شده است. حال باید این داده‌ها را به فرمت **DataLoader** تبدیل کنید تا آماده استفاده در شبکه عصبی باشند.

- چرا از همان فرمت Dataset استفاده نمی‌کنیم و آن را به فرمت DataLoader در می‌آوریم؟
DataLoader چه مزیتی نسبت به فرمت Dataset دارد؟
- برای آشنایی بهتر با مجموعه داده، از هر کلاس 5 تصویر را به صورت رندوم visualize کنید.

بخش اول: طبقه بندی با استفاده از شبکه عصبی Fully Connected

در این بخش، از یک شبکه عصبی fully connected برای طبقه‌بندی تصاویر استفاده می‌کنید. کد مربوط به ساختار این شبکه عصبی را که در قالب کلاس FullyConnectedNetwork آمده است، تکمیل کنید. این کلاس از nn.Module ارث‌بری می‌کند. در واقع تمام مدل‌های PyTorch باید از این کلاس پایه مشتق شوند تا بتوانند از قابلیت‌هایی مثل ثابت پارامترها، ذخیره و بارگذاری وزن‌ها و ... استفاده کنند. همانطور که توضیح داده شد، هر تصویر در CIFAR-10 دارای ابعاد 32×32 است و تعداد کانال‌های آن هم برابر 3 است (به دلیل RGB بودن تصاویر). همچنین تعداد کلاس‌های نهایی برابر 10 تا می‌باشد. سپس به سراغ تعریف معماری این شبکه عصبی بروید. در طراحی شبکه خود می‌توانید از dropout استفاده کنید. این روش یکی از روش‌های کارا برای کاهش بیش‌برازش⁶ در شبکه‌های عصبی است.

- درباره نحوه عملکرد dropout و تاثیر آن روی مدل تحقیق کنید.

توجه: به دلیل اینکه می‌خواهیم در نهایت عملکرد این مدل را با CNN مقایسه کنیم، این مقایسه باید در شرایط یکسانی انجام شود تا از صحت آن مطمئن شویم و به همین علت، تعداد پارامترهای قابل آموزش⁷ این دو مدل باید تقریباً با هم برابر و در بازه $33,500,000 \pm 500,000$ باشد. بنابراین سعی کنید شبکه خود را به گونه‌ای بسازید که تعداد پارامترهای آن در این بازه باشد.

⁶ Overfitting

⁷ Trainable Parameters

- حال بر اساس معماری توضیح داده شده برای این مدل، تعداد پارامتر های مدل را به صورت دستی محاسبه کرده و نحوه محاسبه را ذکر کنید و در نهایت آن را با نتیجه کد داده شده در نوتبوک مقایسه کنید.

حال یک instance از مدلی که ساخته شده ایجاد کنید و آن را به device ای (GPU یا CPU) که می خواهید مدل را روی آن آموزش دهید منتقل کنید.

سپس نیاز دارید که یک تابع هزینه⁸ یک الگوریتم بهینه سازی⁹ برای مدل خود انتخاب کنید.

- درباره تابع هزینه **CrossEntropy** تحقیق کنید و دلیل استفاده از آن را با توجه به ماهیت مسئله توضیح دهید.

- بهینه ساز های مختلفی برای وزن ها و پارامتر های شبکه های عصبی وجود دارد. یکی از معروف ترین بهینه ساز های موجود، بهینه ساز **Adam** است. درباره نحوه کارکرد این بهینه ساز تحقیق کنید. چه تفاوت های اصلی ای بین Adam و SGD که در درس با آن آشنا شدید وجود دارد؟

حال فرایند آموزش مدل را شروع کنید. در طی این فرایند، باید موارد زیر برای هر epoch نشان داده شوند:

- Number of Epoch
- Train Loss
- Train Accuracy
- Validation Loss
- Validation Accuracy

پس از اتمام فرایند آموزش، مقادیر Train Loss و Validation Loss را به ازای هر epoch ذخیره کنید تا در ادامه بتوانید نمودار های مربوطه را برای تحلیل مدل رسم کنید و مقایسه درستی از مدل ها داشته باشید.

توجه: همانطور که گفته شد، برای مقایسه این دو مدل باید تا حد امکان مواردی که بین دو مدل مشترک است را یکسان در نظر گرفت تا مقایسه این دو مدل تا حد امکان در شرایط یکسانی صورت گیرد. یکی از این موارد، تعداد epoch است. این تعداد را برای هر دو مدل خود **60** در نظر بگیرید.

حال با استفاده از مقادیر ذخیره شده، نمودار های Loss و Accuracy را بر اساس شماره epoch ها رسم کنید.

⁸ Loss Function

⁹ Optimization Algorithm

- نمودار ها را تحلیل کنید. آیا بیش‌برازش رخ داده است؟ چگونه می توان این موضوع را فهمید؟

در نهایت، مدل خود را بر روی داده test ارزیابی کنید. برای ارزیابی مدل، مقادیر train loss و test accuracy را محاسبه کنید.

بخش دوم: طبقه بندی با استفاده از شبکه عصبی پیچشی (CNN)

در این بخش، هدف ما حل همان مسئله‌ی طبقه‌بندی مجموعه‌داده‌ی CIFAR-10 با استفاده از یک شبکه‌ی عصبی پیچشی (Convolutional Neural Network یا CNN) است. سپس عملکرد این مدل را با شبکه‌ی عصبی Fully Connected که در بخش اول پیاده‌سازی کردید، مقایسه خواهیم کرد. هدف نهایی، درک بهتر تفاوت معماری و توانایی‌های یادگیری این دو نوع شبکه است.

1. آموزش و ارزیابی شبکه عصبی پیچشی: در ادامه‌ی مسیر قبلی، اکنون باید یک شبکه‌ی عصبی پیچشی طراحی و آموزش دهید تا بتوانید عملکرد آن را با مدل Fully Connected مقایسه کنید. برای این منظور، ابتدا ساختار شبکه را در قالب کلاسی با نام CNN پیاده‌سازی می‌کنید. این مدل باید به‌گونه‌ای طراحی شود که تعداد پارامترهای آموزش‌پذیر آن در حدود $33,500,000 \pm 500,000$ باشد تا شرایط مقایسه منصفانه با مدل قبلی فراهم شود.

پس از طراحی معماری شبکه، لازم است تعداد کل پارامترهای آن را به‌صورت دستی محاسبه کرده و نحوه‌ی محاسبه را به‌وضوح بیان کنید. سپس این مقدار را با عدد به‌دست‌آمده از اجرای کد در نوت‌بوک مقایسه کنید تا از صحت طراحی خود اطمینان حاصل نمایید.

در مرحله‌ی بعد، مشابه بخش قبل، باید تابع هزینه و الگوریتم بهینه‌سازی مناسبی انتخاب شود. پس از تعریف موارد فوق، مدل را به‌مدت 60 دوره‌ی آموزشی (epoch) آموزش دهید. در هر دوره، مقادیر خطا (loss) و دقت (accuracy) را برای داده‌های آموزش و اعتبارسنجی (validation) محاسبه، ذخیره و نمایش دهید.

- در مورد تفاوت عملکرد این دو مدل تفکر کنید. کدام یک عملکرد بهتری نشان داد؟ علت چیست؟
- در مرحله‌ی نهایی، از شما خواسته می‌شود 24 تصویر را که مدل به‌صورت نادرست طبقه‌بندی کرده است نمایش دهید. برای هر تصویر، باید برچسب پیش‌بینی‌شده و برچسب واقعی ذکر شود. در صورتی که تصاویر را پیش‌تر normalize کرده‌اید، پیش از نمایش باید آن‌ها را unnormalize کنید تا رنگ‌ها و ظاهر آن‌ها طبیعی و قابل تفسیر باشند.

2. جستجوی فضای ویژگی شبکه عصبی پیچشی:

پس از آموزش مدل، می‌توان خروجی لایه‌های میانی آن (مانند آخرین لایه کانولوشن یا لایه Fully Connected پیش از لایه خروجی) را به‌عنوان یک نمایش فشرده از تصویر در نظر گرفت. به این نمایش جدید، فضای ویژگی گفته می‌شود. در این بخش، هدف تحلیل و درک بهتر این فضا است. پیشنهاد می‌شود پیش از ادامه، در مورد فضای ویژگی در شبکه‌های CNN مطالعه کوتاهی انجام دهید.

سپس 5 نمونه از داده‌های تست که مدل آن‌ها را به درستی طبقه‌بندی کرده را به صورت تصادفی انتخاب کرده و فضای ویژگی آن را محاسبه کنید. اکنون می‌توانید برای هر داده نمونه‌گیری شده، 5 همسایه نزدیک آن را با استفاده از الگوریتم K Nearest Neighbor از فضای ویژگی مجموعه داده آموزش بیابید و نمایش دهید.

- بررسی کنید که چه شباهتی میان این نمونه‌ها و همسایگان آن‌ها وجود دارد. آیا این شباهت ظاهری یا معنایی است؟ چرا در فضای ویژگی چنین نزدیکی‌ای رخ داده است؟ چه ارتباطی بین داده‌های آموزش و تست در این فضا برقرار است؟

برای تحلیل بیشتر، با نمونه‌گیری از فضای ویژگی مجموعه داده آموزش به اندازه M ، یک ماتریس داده با ابعاد $M \times N$ (که N تعداد ابعاد فضای ویژگی است) بسازید. سپس با استفاده از الگوریتم t-SNE ابعاد آن را به 2 کاهش داده و داده‌ها را در فضای دوبعدی نمایش دهید.

- چه ارتباطی میان فضای ویژگی مدل و خوشه‌بندی داده‌ها مشاهده می‌کنید؟ علت چیست؟ در انتهای این بخش، قصد داریم نقشه‌های ویژگی (Feature Maps) را که حاصل پردازش تصویر توسط لایه‌های پیچشی هستند، بصری‌سازی کنیم. این نقشه‌ها نمایانگر بخش‌هایی از تصویر هستند که شبکه به آن‌ها توجه بیشتری داشته است.

- در مورد نقشه‌های ویژگی اطلاعات بیشتری کسب کنید. با استفاده از آن‌ها، چه اطلاعاتی را می‌توان بدست آورد؟

ابتدا یک تصویر تصادفی از مجموعه داده تست انتخاب می‌شود. سپس مدل را clip می‌کنیم. clip کردن مدل به این معناست که بخشی از شبکه عصبی (معمولاً چند لایه ابتدایی) را جدا کرده و فقط خروجی آن بخش را بررسی کنیم. این کار به ما اجازه می‌دهد تا نمایش میانی داده را در مراحل مختلف پردازش شبکه مشاهده کنیم و بفهمیم مدل در آن نقطه چه ویژگی‌هایی را استخراج کرده است. با این کار می‌توان عملکرد داخلی شبکه را به صورت مرحله‌به‌مرحله تحلیل کرد و لایه‌های مختلف را از نظر قدرت تشخیص ویژگی‌ها بررسی نمود. در نهایت نیز از شما خواسته می‌شود تا خروجی میانی مدل clip شده را به صورت بصری نمایش دهید.

نکات پایانی

- دقت کنید که کد شما باید به نحوی زده شده باشد که نتایج قابلیت بازتولید داشته باشند.
- درباره هر بخش از مراحل پروژه می‌بایست علل استفاده یا عدم استفاده از هر الگوریتم، مزایا و معایب، عملکرد، فرا پارامترها و وضعیت خروجی‌ها را بطور دقیق مطالعه کنید. از این موضوعات در زمان تحویل پرسیده خواهد شد.
- از نحوه عملکرد کدهایی که در اختیار شما قرار گرفته شده در حین تحویل پرسیده خواهد شد.
- پس از مطالعه کامل و دقیق صورت پروژه، در صورت وجود هرگونه ابهام یا سوال با طراحان پروژه در ارتباط باشید.
- نوت‌بوک کامل شده را با فرمت `AI_CA4_LASTNAME_STUDENTID.ipynb` نام گذاری کرده و در ای‌لرن بارگذاری کنید.
- توجه کنید این تمرین باید به صورت تک‌نفره انجام شود و پاسخ‌های ارائه شده باید نتیجه فعالیت فرد نویسنده باشد. در صورت مشاهده تقلب به همه افراد مشارکت‌کننده، نمره تمرین 100- و به استاد نیز گزارش می‌گردد. همچنین نوشته نشدن کدها توسط هوش مصنوعی نیز بررسی می‌شود!