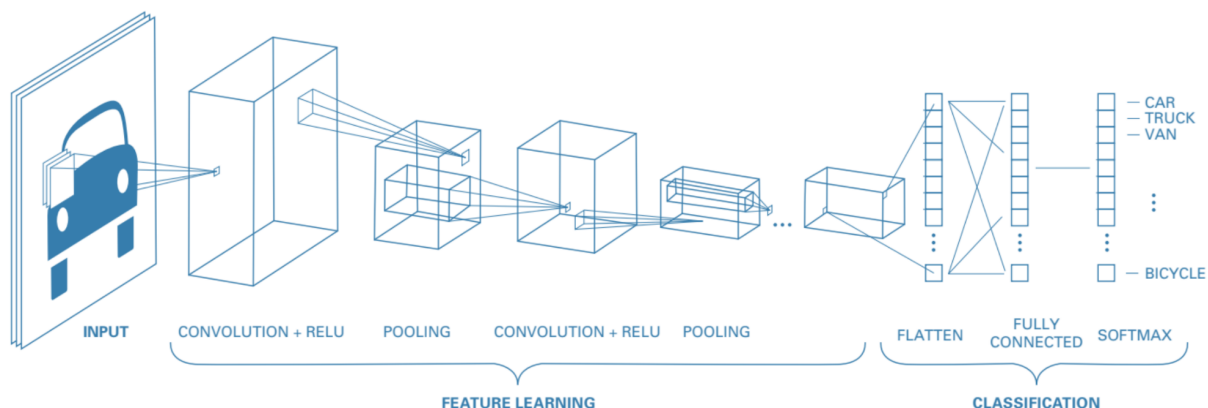




مقدمه

شبکه‌های عصبی کانولوشنی یا پیچشی (Convolutional Neural Networks) نوعی مدل یادگیری عمیق هستند که به طور گسترده و به خصوص برای پردازش تصویر و ویدیو استفاده می‌شوند. 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 کانولوشنی، آن‌ها را در کارهایی مانند طبقه‌بندی تصویر، تشخیص اشیا^۴ و بخش‌بندی تصویر^۵ بسیار پر قدرت می‌سازد.

CNN ها در حوزه‌های مختلف به موفقیت‌های بزرگی دست یافته‌اند و به پیشرفت در زمینه بینایی ماشین کمک چشمگیری کرده‌اند. تاریخچه این شبکه‌ها به سالهای ۱۹۸۰ و ۱۹۹۰ برمی‌گردد، زمانی که با وجود فراگیر شدن رایانه‌ها و روش‌های جدید پردازش تصویر مانند تبدیل فوریه و Small Discrete Fourier Transform و ... همچنان محدودیت‌های زیادی در این حوزه وجود داشت. اما با پیدایش شبکه‌های کانولوشنی، این مدل‌ها توانستند با نوآوری‌های خود در زمینه تشخیص الگو و تصویر، به پردازش تصویر به صورت خودکار و با کارایی بالا بپردازند.



شکل ۱. معماری کلی یک CNN

^۱ Convolutional Layers

^۲ Pooling Layers

^۳ Fully Connected Layers

^۴ Object Detection

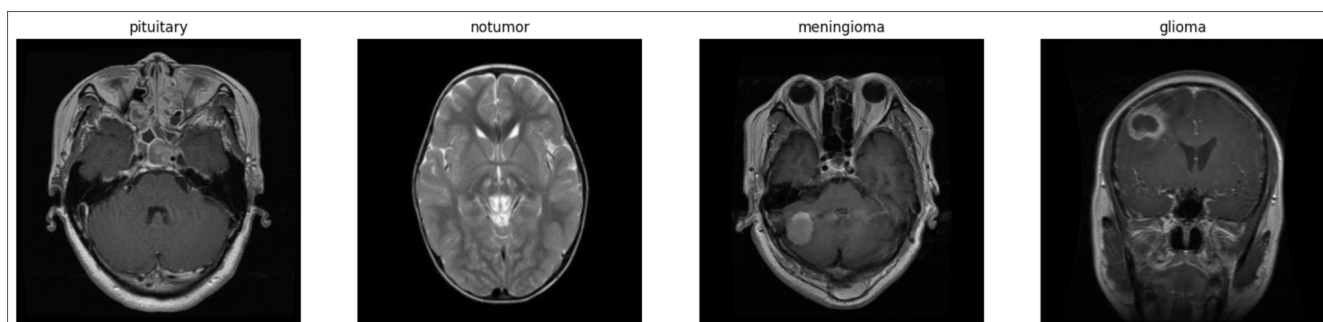
^۵ Image Segmentation

تعریف مسئله

در این تمرین، شما به پیاده‌سازی یک شبکه کانولوشنی برای طبقه‌بندی تصاویر با استفاده از فریم‌ورک **PyTorch** می‌پردازید. برای راحتی استفاده از کتابخانه‌ها و تسریع فرایند آموزش، توصیه می‌شود که از سرویس‌های مثل **Google Colab** و یا **Kaggle** 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 و در صورت استفاده از سرویس **Kaggle** هم از **Accelerator** های آن استفاده کنید. دقت کنید که مدت زمان استفاده از GPU در هر دوی این سرویس‌ها 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 و به مشکل نخورید.

آشنایی با مجموعه داده

مجموعه داده استفاده شده در این تمرین، مجموعه تصاویر مربوط به تصاویر MRI از مغز برای تشخیص تومورهای مغزی مختلف است. این دیتاست شامل عکس‌های مختلف متعلق به ۴ کلاس **notumor**، **glioma**، **meningioma** و **pituitary** است. توجه کنید که تعداد داده‌های موجود در هر کلاس با هم برابر نیستند. دیتاست مربوطه را می‌توانید از این **لینک** دریافت و استفاده کنید.



شکل ۲. نمونه تصاویر متعلق به کلاس‌های مختلف دیتاست

پیش پردازش داده

- ابتدا یک کلاس دیتاست شخصی‌سازی شده که از کلاس **Dataset** پایتورچ استفاده می‌کند بنویسید.
- تمام تصاویر موجود در دیتاست را با استفاده از ابزارهای موجود به ابعاد 512×512 تغییر اندازه دهید.
- درباره تاثیر عملیات **Normalization** مقدار پیکسل‌های عکس‌ها تحقیق کرده و مزایا و معایب آن را در گزارش بیاورید. همچنین در صورت نتیجه گرفتن مفید بودن آن برای این پروژه، **Normalization** را روی عکس‌های دیتاست اعمال کنید.
- دیتاست موجود را به دو بخش **train** و **test** با نسبت مناسب تقسیم کنید و دلیل انتخاب این نسبت را شرح دهید.
- تعداد داده‌های موجود برای هر کلاس را در مجموعه داده **train** و **test** را با استفاده از نمودار نمایش دهید.
- یک نمونه عکس از هر کلاس دیتاست را (مطابق شکل ۲) نمایش دهید.

طبقه‌بندی تصاویر با استفاده از شبکه CNN

در این بخش از پروژه می‌خواهیم شبکه خود را ساخته و سپس آن را آموزش دهیم. همانطور که توضیح داده شد، شبکه‌های کانولوشنی، با استفاده از لایه‌های کانولوشنی، لایه‌های ادغام و لایه‌های کاملاً متصل، قادرند اطلاعات مربوط به الگوها و ویژگی‌های مختلف در تصویر را استخراج و استفاده کنند. لایه کانولوشنی با استفاده از عملیات کانولوشن، فیلترها را بر روی تصویر اعمال کرده و نمایش جدیدی از تصویر را ایجاد میکند که شامل ویژگی‌های محلی است. این لایه‌ها به صورت مکرر در سراسر شبکه استفاده می‌شوند تا ویژگی‌های سطح بالاتر را استخراج کنند.

لایه‌های ادغام به منظور کاهش ابعاد تصویر و حذف اطلاعات بی‌اهمیت، استفاده می‌شوند. این لایه‌ها با استفاده از معیارهایی مانند حداکثرگیری^۶ یا میانگین‌گیری^۷، به کاهش ابعاد ویژگی‌های استخراج شده می‌پردازند.

لایه‌های کاملاً متصل، در نهایت با استفاده از ویژگی‌های استخراج شده توسط لایه‌های کانولوشنی و ادغام، تصمیم‌گیری نهایی را برای طبقه‌بندی تصویر انجام می‌دهند. این لایه‌ها مشابه لایه‌های معمولی در شبکه‌های عصبی عمل می‌کنند و خروجی نهایی را تولید می‌کنند که شامل احتمال‌های مربوط به تعلق تصویر به هر کلاس است.

دقت کنید که ورودی شبکه عصبی در این فاز، تصاویر هستند که ساختار چند بعدی دارند. همچنین ویژگی‌های استخراج شده توسط لایه‌های کانولوشنی ما هم ساختار چند بعدی خواهند داشت ولی ورودی لایه‌های کاملاً متصل ما به صورت یک بردار هستند. در نتیجه نیاز است که بین این دو بخش از یک لایه Flatten استفاده کنید.

معماری شبکه CNN

در این بخش ابتدا باید کلاس مربوط به مدل خود را با معماری مد نظر خود پیاده‌سازی کنید. برای این کار می‌توانید از ماژول‌ها و لایه‌های آماده موجود در PyTorch مثل **Conv2d**، **MaxPool2d**، **Linear**، **ReLU** و **Sequential** استفاده کنید. دقت کنید که معماری استفاده شده، نحوه اتصال لایه‌ها به هم و همچنین پارامترهای استفاده شده برای هر لایه و کل شبکه در این بخش حائز اهمیت است و باید برای استفاده از آن‌ها دلیل منطقی داشته باشید. برای شروع کار و درک بهتر ساختار کلی یک شبکه CNN، می‌توانید از این **لینک** استفاده کرده و یا این **لینک** را مطالعه کنید و برای طراحی شبکه خود از آن ایده بگیرید. همچنین لازم است که توضیحات لازم درباره نحوه عملکرد و همچنین دلیل استفاده از آن‌ها به طور کامل در گزارش آورده شود. موارد خواسته شده در این بخش شامل **kernel size** و **stride** برای لایه‌های مختلف، **padding** استفاده شده در لایه‌های مختلف، تعداد فیلترهای لایه‌های convolutional، عمق شبکه convolutional و **fully connected**، توابع فعال‌سازی استفاده شده بین لایه‌ها و تعداد نورون‌های موجود در لایه‌های **fully connected** و دیگر موارد استفاده شده توسط شما در پیاده‌سازی مدلتان است. به طور مثال نیاز است که در گزارش خود، توابع فعال‌سازی مختلف و تفاوت‌های آن‌ها با هم و همچنین دلیل استفاده از هریک از آن‌ها در شبکه خود را ذکر کنید.

^۶ Max Pooling

^۷ Average Pooling

تابع هزینه

پس از تعریف مدل خود، نیاز است تا تابع هزینه⁸ مناسب برای محاسبه loss شبکه در حین آموزش و در نتیجه آموزش صحیح مدل را انتخاب کنید. برای این بخش هم می‌توانید از **ابزارهای آماده** ارائه شده توسط PyTorch استفاده کنید، اما نیاز است که توابع هزینه متفاوت، نحوه عملکرد آن‌ها، کاربرد هر یک از آن‌ها و همچنین دلیل انتخاب تابع هزینه نهایی خود را ذکر کنید.

بهینه‌ساز

برای آموزش مدل همچنین نیاز به یک بهینه‌ساز⁹ داریم. در درس با بهینه‌ساز SGD¹⁰ آشنا شده‌اید. این روش بهینه‌سازی با استفاده از روش گرادیان کاهشی به بهینه‌سازی وزن‌ها و پارامترهای شبکه عصبی می‌پردازد تا loss کلی کاهش یافته و دقت مدل افزایش پیدا کند. بهینه‌سازهای قدرتمندتر دیگری نیز وجود دارند که آن‌ها به طور آماده برای استفاده در PyTorch وجود دارند. درباره بهینه‌ساز **Adam** تحقیق کرده و به طور مختصر نحوه عملکرد و تفاوت‌های آن با SGD را شرح دهید. همچنین در صورت استفاده از هر بهینه‌ساز برای مدل، دلیل خود را مختصراً شرح دهید.

آموزش مدل

در این بخش نیاز است که یک **DataLoader** تعریف کرده و در فرایند آموزش از آن استفاده کنید. همچنین درباره پارامتر batch size و تاثیر آن در فرایند آموزش توضیح دهید. در پایان نیاز است که مدل خود را با استفاده از نتایج بخش‌های قبل آموزش دهید. همچنین توجه کنید که پس از اتمام فرایند آموزش باید نمودار loss و accuracy برحسب epoch را هم برای دادگان آموزش و هم برای دادگان ارزیابی در حین فرایند آموزش رسم کنید.

تاثیر روش‌های Regularization بر روی آموزش مدل

روش‌های Regularization در فرایند آموزش شبکه‌های عصبی، روش‌هایی هستند که برای جلوگیری از overfitting در حین آموزش مورد استفاده قرار می‌گیرند. دو روش بسیار متداول برای انجام این کار استفاده از Dropout و Batch Normalization هستند. ابتدا درباره نحوه عملکرد هر یک از این دو روش تحقیق کرده و در گزارش ذکر کنید. سپس هر دوی این روش‌ها را روی مدل اعمال کرده و تاثیر آن‌ها روی فرایند آموزش را نمایش داده و تحلیل کنید. همچنین نیاز است که نتایج مدل با اعمال این روش‌ها را با نتایج مدل بدون Regularization مقایسه کنید. توصیه می‌شود که از لایه Dropout بین لایه‌های بخش Fully Connected و از لایه Batch Normalization در میان لایه‌های convolutional استفاده کنید. دقت کنید که در این بخش نیز مثل بخش قبل نیاز است که دلیل استفاده از پارامترهای استفاده شده در لایه‌ها مثل batch size و یا احتمال drop کردن را توجیه نمایید.

⁸ Loss Function

⁹ Optimizer

¹⁰ Stochastic Gradient Descent

ارزیابی و تحلیل نتایج

پس از آموزش مدل نوبت به ارزیابی عملکرد مدل بر روی مجموعه داده ارزیابی می‌رسد. در این بخش نیاز است که مقادیر ۴ معیار Accuracy، Precision، Recall و F1 Score را (که در پروژه سوم با آن‌ها آشنا شده‌اید) به ازای هر کلاس و همچنین به طور کلی (در صورت نیاز برای میانگین‌گیری از میانگین Macro استفاده کنید) گزارش داده و تحلیل کنید.

پس از محاسبه معیارهای ذکر شده، Confusion Matrix مربوط به نتایج به دست آمده را هم رسم کرده و تحلیل کنید.

لینک‌های مفید

برای یادگیری شروع کار با PyTorch می‌توانید از این [لینک](#) کمک بگیرید.

برای آموزش شیوه استفاده از Google Colab می‌توانید از این [لینک](#) و یا این [لینک](#) استفاده نمایید.

برای آشنایی بیشتر با توابع هزینه مختلف و کارکرد آن‌ها می‌توانید این [لینک](#) را مطالعه کنید.

در این [لینک](#) می‌توانید درباره بهینه‌ساز Adam اطلاعات بیشتری کسب کنید.

برای آشنایی بیشتر با روش‌های جلوگیری از overfitting می‌توانید این [لینک](#) را مطالعه نمایید.

نکات پایانی

- توضیحات مربوط به هر بخش از پروژه را بطور خلاصه و در عین حال مفید در گزارش خود ذکر کنید. از ابزارهای تحلیل داده مانند نمودارها استفاده کنید.
- پس از مطالعه کامل و دقیق صورت پروژه، در صورت وجود هرگونه ابهام یا سوال با طراحان پروژه در ارتباط باشید.
- نتایج، گزارش و کدهای خود را در قالب یک فایل فشرده با فرمت AI_CA5_[stdNumber].zip در سامانه ایلرن بارگذاری کنید.
- محتویات پوشه باید شامل فایل jupyter-notebook، خروجی html و فایل‌های مورد نیاز برای اجرای آن باشد. از نمایش درست خروجی‌های مورد نیاز در فایل html مطمئن شوید.
- دقت کنید که نیازی به آپلود مجموعه داده‌ها در سامانه ایلرن نیست.
- هدف از تمرین، یادگیری شماست. لطفا تمرین را خودتان انجام دهید

موفق باشید