بسم الله الرحمن الرحیم

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

تمرین درس هوش مصنوعی در سیستمهای نهفته – آبان 1403

تمرین دوم

اهداف

هدف این تمرین، آشنایی با شبکه های عصبی پیچشی (CNN) و یادگیری اصول و مفاهیم فشرده‌سازی شبکه‌ها، مانند هرس وزنی و کوانتیزاسیون، است. این تمرین به دانشجویان کمک می‌کند تا با تکنیک‌های کاهش حجم مدل‌های عصبی آشنا شوند و درک کنند چگونه می‌توان از این روش‌ها برای بهبود کارایی مدل‌ها در محیط‌های با منابع محدود مانند دستگاه‌های تعبیه‌شده و موبایل استفاده کرد. همچنین، با پیاده‌سازی و آزمایش روش‌های فشرده‌سازی، دانشجویان می‌توانند تأثیر این تکنیک‌ها بر دقت مدل و مصرف منابع را تحلیل و مقایسه کنند.

۱- مقدمه

برای دستیابی به هدف تعیین شده، از میان طیف گستردهای از انواع شبکه های عصبی موجود، یک مثال ساده و کاربردی از شبکه ی CNN درنظر گرفته ایم. در ادامه توضیح مختصری درباره این دسته از شبکه ها داده می شود و سپس در بخش های بعدی نیازمندی های لازم برای انجام تمرین ذکر می شود.

۱-۱- شبکه CNN

شبکه های عصبی پیچشی (CNN) ها یکی از انواع مدل های یادگیری عمیق هستند که بیشتر در پردازش تصاویر و ویدئوها به کار می روند. این شبکه ها از ساختاری متشکل از لایه های پیچشی[۱]، لایه های تجمعی [۲]و لایه های کاملاً متصل [۳]بهره می برند. لایه های پیچشی وظیفه استخراج ویژگی های مهم از تصاویر را بر عهده دارند، که این امر از طریق فیلترهایی که روی تصویر حرکت می کنند و ویژگی های محلی مانند لبه ها، گوشه ها یا بافت ها را شناسایی میکنند، انجام می شود. لایه های تجمعی به کاهش ابعاد و پیچیدگی تصاویر کمک میکنند، در حالی که اطلاعات مهم را حفظ می کنند. در نهایت، لایه های کاملاً متصل وظیفه دارند تا از این ویژگی های استخراج شده برای انجام وظایفی مانند طبقه بندی یا تشخیص اشیاء استفاده کنند. شبکه های عصبی پیچشی به دلیل کارایی بالا در شناسایی الگوها و توانایی های خود در یادگیری ویژگی های پیچیده، در بسیاری از کاربردهای پردازش تصویر و بینایی کامپیوتری پیشرو هستند.

۱-۲- مدل طبقه بندی

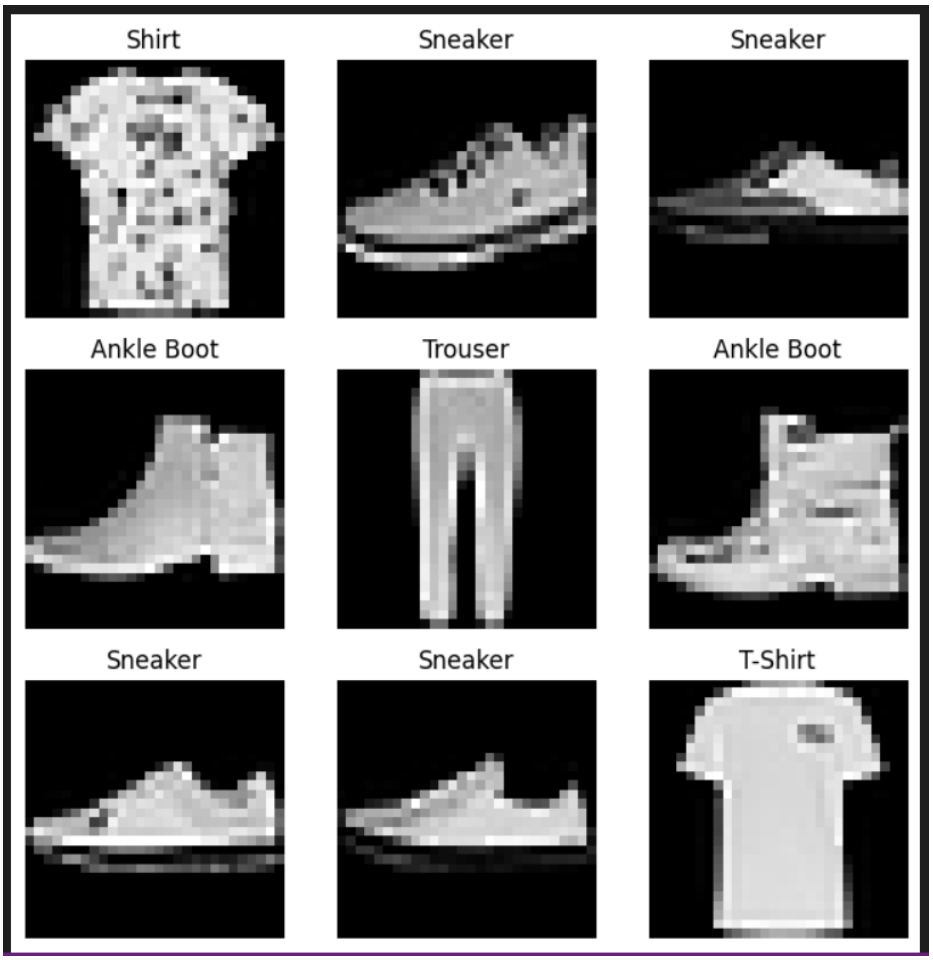
یکی از کاربردهای بسیار مهم و رایج شبکه های CNN ، طبقه بندی یا برچسب گذاری ورودی است. این مدلها ابتدا ورودی مورد نظر را دریافت می کنند ، سپس پردازش و تحلیل این داده ها را با استفاده از لایه های میانی انجام می دهند و در نهایت کلاس مرتبط با ورودی داده شده را در خروجی نشان می دهند .

---

۱ Convolutional Layers
۲ Pooling Layers
۳ Fully Connected Layers

۱-۳- دیتاست

در این تمرین ما از یک شبکه عصبی CNN برای طبقه بندی مجموعه داده هایFashionMNIST استفاده می کنیم. در شکل ۱ نمونه هایی از داده های FashionMNIST نشان داده شده است .



شکل ۱- بخشی از مجموعه داده FashionMNIST

۱-٤- فشرده سازی

تاکنون با شبکه‌های CNN و یکی از کاربردهای آن آشنا شدید. اگرچه این مدل‌ها کوچک هستند و به راحتی با دقت بیتی بالا بر روی پلتفرم‌هایی با منابع محدود قابل پیاده‌سازی هستند، ولی در دنیای واقعی شبکه‌های بزرگی وجود دارند که پیاده‌سازی آن‌ها با دقت بیتی بالا، به دلیل حجم بسیار زیاد پارامترهای موجود، که نیازمند تبادلات حافظه‌ای زیاد و محاسبات سنگین ناشی از آن هستند، بر روی چنین پلتفرم‌هایی امکان‌پذیر نیست یا هزینه بسیار زیادی را تحمیل می‌کند.

یکی از راهکارهای ساده و موثر برای کاهش حجم این مدل‌ها، هرس کردن آن‌هاست. در هرس [۴]،پارامترهایی که تاثیر کمی در دقت شبکه دارند یا در نتایج خروجی کمتر موثر هستند، حذف یا صفر می‌شوند. این کار با هدف کاهش حجم مدل و مصرف حافظه انجام می‌شود و می‌تواند به کاهش زمان پردازش و کارایی بیشتر کمک کند.

[۴] Pruning

انواع هرس شامل هرس وزنی[۵]، که وزن‌های کوچک را حذف می‌کند، و هرس کانالی[۶]، که کانال‌های ناکارآمد را کاهش می‌دهد، هستند. در این روش‌ها سعی می‌شود که کمترین افت دقت[۷] یا افزایش خطا رخ دهد.
علاوه بر هرس، یکی دیگر از روش‌های بسیار پرکاربرد و کم‌خطا، استفاده از کوانتیزاسیون[۸] است. کوانتیزاسیون فرآیندی است که مقادیر پیوسته را به مجموعه محدودی از مقادیر گسسته یا سطوح تقریب می‌دهد یا گرد می‌کند. این فرآیند به‌طور معمول در پردازش سیگنال دیجیتال و فشرده‌سازی داده‌ها برای نمایش و ذخیره‌سازی داده‌ها با کارآیی بیشتر استفاده می‌شود. در مدل‌های شبکه عصبی، عموما پارامترهای وزن مدل را از اعداد ممیز شناور[۹] ۳۲ بیت به اعداد ممیز ثابت[۱۰] ۱۶ بیت یا کمتر گرد می‌کنند؛ به عنوان مثال، با تبدیل وزن‌ها از ۳۲ به ۸ بیت ممیز ثابت، حجم مدل تا چهار برابر کاهش می‌یابد، که مقدار قابل توجهی است.

۲- پیش نیازهای انجام تمرین

۱. آشنایی اولیه با پایتان و شبکه های عصبی پیچشی
۲. Pytorch Framework

برای اجرای کدها می توانید از colab استفاده کنید .

۳- مراحل انجام تمرین

در این تمرین هدف آشنایی با شبکه های CNN و فریمورک Pytorch جهت آموزش مدل در پایتان ، سپس استفاده از روش های فشرده سازی برای دستیابی به حداقل حجم ممکن و حفظ صحت مدل تا حد امکان و در نهایت ذخیره سازی بهترین وزن ها می باشد.

۳-۱- مرحله اول

یک فایل کد پایتان hw-2.ipynb مربوط به طراحی طبقه بند برای داده های FashionMNIST در اختیار شما قرار داده شده است .

موارد زیر را انجام دهید :

۱) پس از دریافت و آماده سازی دیتاست ، کد های سلول را اجرا کنید و نتایج آن را گزارش کنید .

۲) نمودار خطای داده های آموزش و اعتبار سنجی را رسم کرده و گزارش کنید .(مقدار دقت برروی داده های تست یا ارزیابی از ۹۱.۹۶ کمتر نباشد)

۳) با استفاده از report_classification در sklearn ،مقادیر precision ، recallو score-f1 را برای مدل نهایی گزارش و تحلیل کنید.

۴) مدل با بهترین دقت را ذخیره نمایید.

---

[۵] Weight Pruning
[۶] Channel Pruning
[۷] accuracy
[۸] Quantization
[۹] Floating-Point
[۱۰] Fixed-Point

۳-۲- مرحله دوم

در این قسمت میخواهیم به کمک تکنیک های فشرده سازی مانند هرس وزنی ، هرس کانالی، کوانتیزاسیون و ترکیب این روش ها به حداقل حجم مدل CNN ذخیره شده در مرحله قبل(ضمن حفظ صحت آن تا حدامکان) دستیابیم. در هرس وزنی، پارامترهایی که تاثیر کمتری در خروجی شبکه دارند، حذف یا صفر می‌شوند. به طور معمول، وزن‌هایی که مقدارشان نزدیک به صفر است، نقش کمتری در محاسبات شبکه ایفا می‌کنند و حذف آن‌ها می‌تواند حجم مدل را بدون افت زیاد در دقت کاهش دهد.

۱) در کد پایتان داده شده تابع هرس وزنی با عنوان "apply_pruning" در اختیار شما قرار گرفته شده است.شیوه عملکرد این تابع را توضیح دهید . هرس وزنی را برای مدل به طوری اجرا نمایید تا بیشترین درصد تُنکی[۱۱] را با حداکثر یک درصد افت دقت در داده های ارزیابی داشته باشید . مقدار تُنکی بدست آمده را اعلام نمایید.

۲) در هرس یک‌باره[۱۲] که در مرحله قبل انجام دادید، تمامی وزن‌های غیرضروری شبکه در یک مرحله شناسایی و حذف می‌شوند اما در هرس مرحله به مرحله[۱۳]، هرس وزن‌ها به تدریج و طی چندین مرحله انجام می‌شود. در این روش، ابتدا درصد کمی از وزن‌ها حذف می‌شوند و سپس مدل دوباره آموزش داده می‌شود تا به دقت مطلوب نزدیک شود. هرس وزنی را این بار به صورت مرحله به مرحله اجرا نمایید و بیشترین مقدار تنکی را با حداکثر یک درصد افت دقت در داده های ارزیابی اعلام نمایید.
همچنین تابعی تحت عنوان "check_sparsity"بنویسید که با دریافت مدل، درصد صفرهای تولید شده از هرس را در هر لایه از آن بررسی و اعلام نماید.

۳) پس از انجام هر دو روش، نتایج (دقت نهایی و میزان تنکی هرلایه) را با هم مقایسه نموده و بررسی نمایید که کدام روش دقت بالاتری حفظ کرده ولی فشرده‌سازی بهتری داشته است.

۴) در کد پایتان داده شده در بخش " quantization utils " توابع لازم جهات اعمال کوانتیزاسیون بر روی مدل اورده شده است. تابع‌های linear_quantize و linear_dequantize به طور کامل پیاده سازی نشده اند. این توابع عمل کوانتیزاسیون و عکس آن جهت تقریب عدد اولیه را انجام می‌دهند. با توجه به کامنت‌های TODO قسمت‌های خواسته شده را تکمیل کنید.

۵) پس تکمیل و پیاده سازی مراحل قبل ، با استفاده از تابع "quantize_model " مدل را با ۸ بیت کوانتایز نمایید و آموزش دهید. نمودار دقت و خطا در طول آموزش را گزارش کنید. همچنین مقادیر precision ، recallو score-f1 را نیز اعلام نمایید.(مقدار بیت ها برای وزن و توابع فعال ساز[۱۴] را یکسان در نظر بگیرید)

۶)مرحله قبل را با ۶ ، ۴و ۲ بیت نیز تکرار نمایید و نتایج را گزارش و تحلیل کنید. بهترین فشرده سازی با کمترین افت دقت در کدام حالت حاصل میشود؟

---

۱۱ sparsity
۱۲ One-Shot Pruning
۱۳ Step-by-Step Pruning
۱۴ Activation Functions

۷) تابعی بنویسید که با دریافت اطلاعات یک مدل از ورودی تعداد نورون های هر لایه و تعداد بیت(حجم نهایی مدل را بر حسب KB، با توجه به ماتریس وزن هر لایه، محاسبه کند ، سپس حجم مدل اصلی و مدل کوانتایز شده را گزارش کنید. و میزان فشرده سازی ایجاد شده را بیان کنید .

۸) پس از رسیدن به بهترین نتایج از قسمت های قبل ، مدل نهایی را prune و کوانتایز کنید. (با کمترین میزان افت دقت) نتایج را گزارش کرده و در نهایت مدل را ذخیره کنید .

لازم است موارد زیر جهت تحویل تمرین و ارائه گزارش رعایت شوند:

- گزارش خود را در بخشهای مجزا شامل چکیده، نحوهی انجام کار، نتایج به دست آمده، تحلیل نتایج، نتیجه گیری و ضمائم بیاورید. فایل گزارش باید بر اساس فرمت قرار داده شده در سایت درس باشد .
- درصورت استفاده از تکنیکهای اضافه برای فشرده سازی، در گزارش توضیح دهید .
- فایل گزارش به صورت doc و PDF باشد. کد خود را نیز آپلود کنید .
- تمرین را فرمت YourName_StudentNo_EAI2.rar آپلود نمایید.
- گروه ها حتما دو نفره باشند .
- بارگذاری فایلهای گزارش توسط یکی از اعضای گروه کافی است .
- نمره از 100 محاسبه می شود و به ازای هر روز تاخیر در اپلود تمرین، به اندازه 2^x (x تعداد روز تاخیر) از نمره شما کسر میشود .
- در صورت مشاهده تشابه زیاد در کدها و گزارش، نمره 100- برای هر دو گروه اعمال خواهد شد .
- تمرین تحویل حضوری دارد که زمان آن بعدا اعلام خواهد شد.

طراحان تمرین :

پویا جمیل دهی(pouyajamildehi@gmail.com)

حانیه شادلو (haniehshadlo1999@gmail.com)

مهدی محمدی نسب (mahdimn2011@yahoo.com)