به نام خدا

دانشگاه تهران

دانشکدگان علوم و فناوری های میان رشته ای

Machine Learning

تمرین سوم

نام و نام خانوادگی	امید مرادی
شماره دانشجویی	830402071
تاریخ ارسال گزارش	13 دی

سوال اول

Vision Transformers (ViTs)

Vision Transformers (ViTs) معماری‌های شبکه عصبی هستند که برای پردازش تصاویر با استفاده از اصول مدل‌های ترانسفورماتور طراحی شده‌اند که در اصل برای پردازش زبان طبیعی (NLP) توسعه یافته‌اند.

ترانسفورماتورهای بینایی کاربردهای گسترده ای در کارهای رایج تشخیص تصویر مانند تشخیص اشیا، تقسیم بندی تصویر، طبقه بندی تصویر و تشخیص عمل دارند. علاوه بر این، ViTها در مدل‌سازی تولیدی و وظایف چند مدلی، از جمله زمینه‌سازی بصری، پاسخ‌گویی به سوال بصری، و استدلال بصری استفاده می‌شوند.

در ViTs، تصاویر به صورت توالی نمایش داده می‌شوند و برچسب‌های کلاس برای تصویر پیش‌بینی می‌شوند که به مدل‌ها اجازه می‌دهد تا ساختار تصویر را به طور مستقل یاد بگیرند. تصاویر ورودی به‌عنوان دنباله‌ای از وصله‌ها در نظر گرفته می‌شوند که در آن هر وصله با به هم پیوستن کانال‌های همه پیکسل‌ها در یک وصله و سپس نمایش خطی آن به بعد ورودی مورد نظر، به یک بردار واحد تبدیل می‌شود.

ساختار ViT:

1. **Image Tokenization**
2. **Linear Embedding**
3. **Positional Embeddings**
4. **Transformer Encoder**
5. **Classification Token**
6. **Output Layer**

ویژگی های اصلی:

ورودی مبتنی بر پچ:

ViT ها وصله های تصویر را به جای کل تصویر پردازش می کنند و به مدل این امکان را می دهند که اندازه های متغیر تصویر را مدیریت کند و به طور مستقل روی مناطق فضایی تمرکز کند.

مکانیسم توجه:

ماژول MHSA به مدل اجازه می دهد تا وابستگی های دوربرد را بیاموزد و بر روی قسمت های خاصی از تصویر که مربوط به یک کار است تمرکز کند.

مقیاس پذیری:

ViTها با مجموعه داده‌ها و مدل‌های بزرگ (به عنوان مثال، ترانسفورماتورهای عمیق‌تر و گسترده‌تر) به خوبی مقیاس می‌شوند، زیرا مکانیسم توجه به طور مؤثر اطلاعات جهانی را مدیریت می‌کند.

پیش آموزش و تنظیم دقیق:

ViTها زمانی بهترین عملکرد را دارند که روی مجموعه داده های مقیاس بزرگ (مانند ImageNet-21k، JFT-300M) از قبل آموزش داده شوند و سپس روی مجموعه داده های کوچکتر و مختص کار تنظیم شوند.

تعصبات استقرایی کمتر:

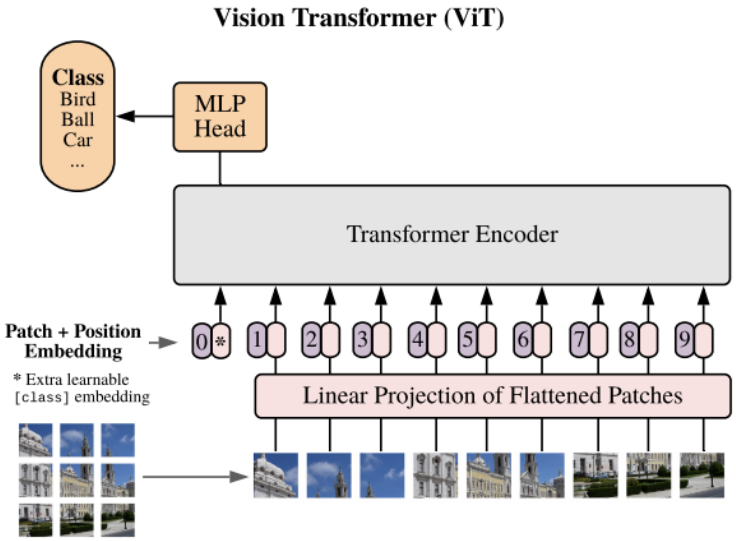
برخلاف شبکه‌های عصبی کانولوشنال (CNN)، ViTها به میدان‌های دریافتی محلی یا اشتراک وزن متکی نیستند. این باعث می‌شود آنها انعطاف‌پذیرتر باشند، اما در عین حال تشنه اطلاعات هستند.

عملکرد بالا:

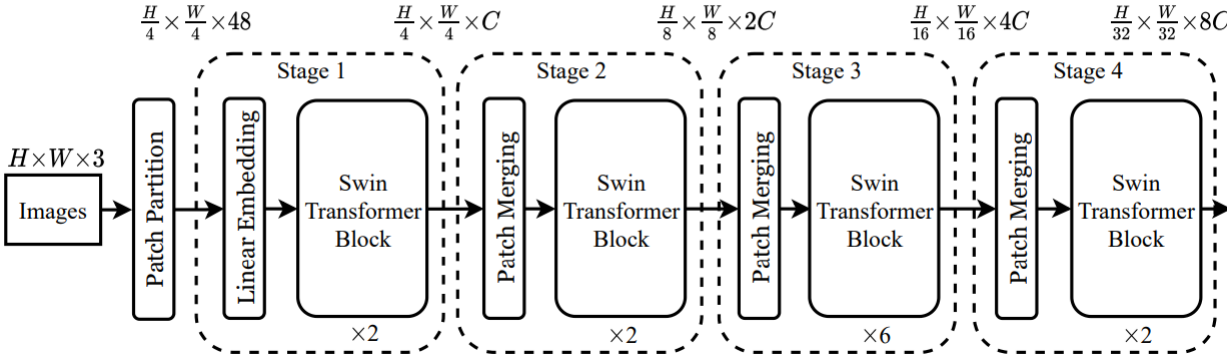
هنگامی که ViT ها به اندازه کافی از قبل آموزش دیده باشند، در بسیاری از وظایف بینایی کامپیوتری، از جمله طبقه بندی، تشخیص اشیا، و تقسیم بندی، به عملکردی پیشرفته دست می یابند.

انعطاف پذیری در بین روش ها:

معماری ترانسفورماتور را می توان با سایر مدالیته ها مانند متن تطبیق داد و ViTs را به یک مدل متحد کننده برای یادگیری چندوجهی تبدیل کرد.



Swin Transformers



این نمودار معماری ترانسفورماتور Swin را نشان می دهد. با تصویری به ابعاد H×W×3 شروع می شود که به تکه های ۴×۴ غیر همپوشانی (Patch Partition) تقسیم می شود. سپس این تکه‌ها به صورت خطی در بردارهای مشخصه بعد C جاسازی می‌شوند و در نتیجه نمایشی میانی با اندازه H/4×W/4×C ایجاد می‌شود.

معماری به چهار مرحله تقسیم می شود:

۱. مرحله ۱: بلوک های ترانسفورماتور Swin Patch های تعبیه شده را بدون تغییر اندازه فضایی آنها (H/4×W/4) پردازش می کنند، اما نمایش ویژگی را افزایش می دهند.
۲. مرحله ۲: ادغام وصله ابعاد فضایی را به H/8×W/8 کاهش می‌دهد، ابعاد ویژگی را دو برابر می‌کند و به ۲ Cمی‌رساند و به دنبال آن بلوک‌های ترانسفورماتور Swin بیشتر می‌شود.
۳. مرحله ۳: یکی دیگر از مراحل ادغام وصله، ابعاد فضایی را به H/16×W/16 کاهش می‌دهد، ابعاد ویژگی را به ۴ Cافزایش می‌دهد و بلوک‌ها این نمایش را پردازش می‌کنند.
۴. مرحله ۴: ادغام نهایی ابعاد فضایی را به H/32×W/32 با ابعاد ویژگی ۸ Cکاهش می‌دهد و به دنبال آن بلوک‌های ترانسفورماتور.Swin

معماری به تدریج وضوح فضایی را کاهش می دهد در حالی که ابعاد ویژگی را برای استخراج کارآمد ویژگی افزایش می دهد.

ویژگی های شبکه:

۱. نمایش سلسله مراتبی: شبکه تصاویر را در چهار مرحله پردازش می کند و به تدریج ابعاد فضایی را کاهش می دهد (از H/4×W/4 به H/32×W/32) در حالی که ابعاد ویژگی را

افزایش می دهد) از C به ۸ .(Cاین رویکرد سلسله مراتبی به طور موثر ویژگی های سطح پایین و بالا را به تصویر می کشد.

۲. ورودی مبتنی بر وصله: تصویر ورودی به وصله های غیر همپوشانی (۴×۴) تقسیم می شود که به صورت خطی در بردارهای ویژگی جاسازی شده اند. این به ترانسفورماتور اجازه می دهد تا تصاویر را به صورت توالی، مشابه مدل های پردازش زبان طبیعی، مدیریت کند.

۳. توجه مبتنی بر پنجره: هر بلوک ترانسفورماتور Swin توجه خود را در پنجره های محلی محاسبه می کند و پیچیدگی محاسباتی را در مقایسه با توجه جهانی کاهش می دهد. این باعث می شود شبکه برای تصاویر با وضوح بالا کارآمد باشد.

۴. مکانیسم پنجره جابجا شده: با جابجایی پنجره ها در لایه های متناوب، شبکه تعاملات بین پنجره ای را فعال می کند و توانایی آن را برای مدل سازی وابستگی های جهانی افزایش می دهد.

۵. ادغام وصله: در هر انتقال مرحله، وصله‌های مجاور ادغام می‌شوند و اندازه فضایی را کاهش می‌دهند و بعد ویژگی را افزایش می‌دهند. این پردازش چند مقیاسی شبکه‌های عصبی کانولوشنال (CNN) را تقلید می‌کند.

۶. مقیاس پذیری: شبکه را می توان با تنظیم تعداد بلوک ها در هر مرحله، بعد ویژگی C و اندازه وصله های ورودی، مقیاس پذیر کرد و آن را با وظایف مختلف و بودجه های محاسباتی سازگار کرد.

۷. تطبیق پذیری: ساختار سلسله مراتبی و مکانیسم های توجه آن را برای کارهای بینایی متنوع، از جمله طبقه بندی تصویر، تشخیص اشیا و تقسیم بندی معنایی مناسب می کند.

۸. کارایی: با محدود کردن توجه به پنجره های کوچکتر و کاهش تدریجی اندازه فضایی، ترانسفورماتور Swin به پیچیدگی خطی نسبت به اندازه تصویر دست می یابد و از نظر محاسباتی کارآمد می شود.

EfficientNet

این تصویر معماری EfficientNet را نشان می‌دهد که یک تصویر ورودی با اندازه ۳×۲۲۴×۲۲۴ را در یک نقشه ویژگی پردازش می‌کند. از اجزای زیر تشکیل شده است:

۱. پیچیدگی اولیه (Conv 3x3): یک لایه کانولوشن استاندارد با هسته ۳×۳ تصویر ورودی را پردازش می کند و ویژگی های فضایی اساسی را استخراج می کند.

۲. بلوک‌های MBConv: اینها لایه‌های پیچیدگی گلوگاه معکوس موبایل (MBConv) با پیچیدگی‌های قابل تفکیک عمیق هستند که برای کاهش پیچیدگی محاسباتی طراحی شده‌اند.

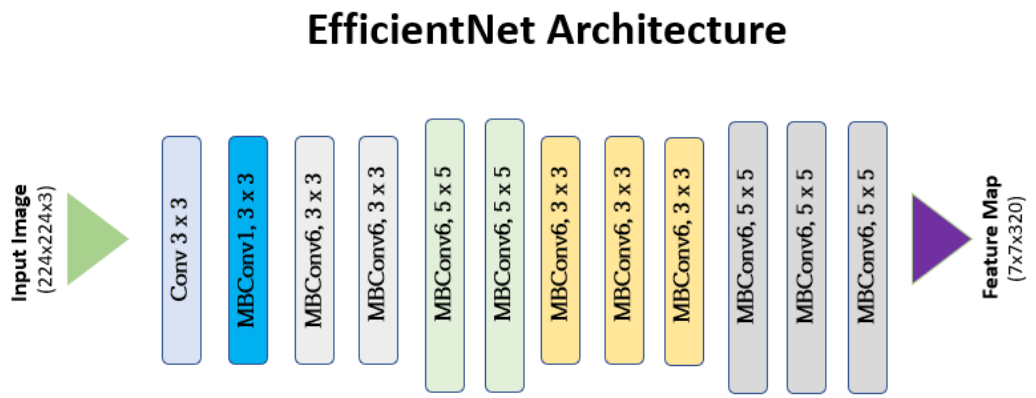
۱. MBConv1 با هسته ۳×۳ عملیات سبک وزن را در اوایل شبکه معرفی می کند.

۲. MBConv6 با هسته‌های ۳×۳ یا ۵×۵ عمق می‌افزاید و فضای ویژگی را با استفاده از ضریب گسترش ۶ برابری برای ابعاد ویژگی میانی افزایش می‌دهد.

۳. استخراج ویژگی های پیشرونده: بلوک های MBConv به طور متناوب بین هسته های ۳×۳ و ۵×۵ برای گرفتن اطلاعات متنی ریز و بزرگتر قرار می گیرند. وضوح فضایی به تدریج کاهش می یابد و کانال های ویژگی با عمیق شدن شبکه افزایش می یابد.

۴. فیچرمپ نهایی: خروجی یک نقشه ویژگی به ابعاد ۷×۷×۳۲۰ است که ویژگی های سطح بالای تصویر ورودی را خلاصه می کند.

این طراحی با مقیاس‌بندی سیستماتیک عمق، عرض و وضوح، کارایی و عملکرد مدل را متعادل می‌کند و آن را برای کارهای طبقه‌بندی بسیار موثر می‌سازد.



ویژگی ها

EfficientNet برای عملکرد بالا با حداقل هزینه محاسباتی طراحی شده است. از مقیاس بندی ترکیبی برای تعادل سیستماتیک عمق (تعداد لایه ها)، عرض (تعداد کانال ها) و

وضوح (اندازه تصویر ورودی) برای مقیاس بندی کارآمد مدل استفاده می کند. این معماری دارای بلوک‌های MBConv است که از پیچیدگی‌های قابل تفکیک عمیق برای کاهش محاسبات و حفظ دقت استفاده می‌کنند. این هسته های ۳×۳ و ۵×۵ را برای استخراج ویژگی های متنوع ترکیب می کند و به تدریج ابعاد فضایی را کاهش می دهد در حالی که کانال های ویژگی را برای نمایش سلسله مراتبی ویژگی افزایش می دهد. EfficientNet با پارامترهای کمتر و FLOP کمتر در مقایسه با سایر شبکه ها به نتایج پیشرفته ای در طبقه بندی تصاویر دست می یابد.

سوال دوم

روش ارائه‌شده در مقاله برای تخمین عدم قطعیت تصمیم‌گیری شبکه‌های عصبی، از تزریق دراپ‌آوت در زمان تست به‌عنوان روشی برای کمّی‌سازی پس‌پرداز عدم قطعیت استفاده می‌کند. برخلاف دراپ‌آوت تعبیه‌شده سنتی که لایه‌های دراپ‌آوت در هر دو مرحله آموزش و تست فعال هستند، این روش لایه‌های دراپ‌آوت را به شبکه‌ای که قبلاً آموزش دیده است اضافه کرده و آن‌ها را تنها در مرحله تست فعال می‌کند. این امکان را فراهم می‌آورد تا بدون نیاز به آموزش شبکه با استفاده از دراپ‌آوت، عدم قطعیت تخمین زده شود و از بار محاسباتی ناشی از آموزش مجدد جلوگیری شود.

در این روش، از مونت‌کارلو دراپ‌آوت در زمان تست استفاده می‌شود که طی آن چندین پیش‌گذر تصادفی در شبکه انجام می‌شود. این فرآیند به تخمین توزیع پیش‌بینی‌های شبکه و عدم قطعیت مربوطه کمک می‌کند. میانگین و واریانس خروجی‌های تصادفی، به ترتیب پیش‌بینی و عدم قطعیت را نشان می‌دهند. برای بهبود کیفیت این تخمین‌ها، نرخ دراپ‌آوت بهینه از طریق کمینه‌سازی خطای لگاریتمی منفی (NLL) روی یک مجموعه اعتبارسنجی تنظیم می‌شود. علاوه بر این، یک ضریب مقیاس‌بندی معرفی می‌شود تا عدم قطعیت‌ها بر اساس رابطه بین خطای پیش‌بینی و عدم قطعیت تخمینی تنظیم شوند. این مقیاس‌بندی اطمینان حاصل می‌کند که تخمین‌های عدم قطعیت قوی و کالیبره‌شده باقی می‌مانند.

این روش از این جهت حائز اهمیت است که می‌توان آن را بر روی هر شبکه‌ای که قبلاً آموزش دیده اعمال کرد و نیاز به دراپ‌آوت در مرحله آموزش را از بین برد و بدین ترتیب تلاش محاسباتی را به طور قابل توجهی کاهش داد. نتایج تجربی در مسائل رگرسیون و وظایف چالش‌برانگیزی مانند شمارش جمعیت نشان می‌دهد که این روش به‌طور مؤثری عدم قطعیت را کمّی‌سازی کرده و در عین حال دقت پیش‌بینی رقابتی

را حفظ می‌کند. با مقیاس‌بندی مناسب مقادیر عدم قطعیت، این روش جایگزینی عملی و کارآمد برای دراپ‌آوت تعبیه‌شده در زمینه کمّی‌سازی عدم قطعیت ارائه می‌دهد.

سوال سوم

الف) لایه‌های پیچشی (Convolutional Layers) و لایه‌های تماما متصل (Fully Connected Layers)در شبکه‌های عصبی پیچشی (CNN) تفاوت‌های کلیدی دارند. لایه‌های پیچشی تنها با یک بخش کوچک از ورودی ارتباط دارند و وزن‌ها به صورت اشتراکی بین تمام بخش‌های تصویر اعمال می‌شوند، در حالی که در لایه‌های تماما متصل، هر نورون به تمام نورون‌های لایه قبلی متصل است که باعث افزایش تعداد پارامترها می‌شود. لایه‌های پیچشی اطلاعات مکانی داده‌ها را حفظ کرده و برای استخراج ویژگی‌های محلی مانند لبه‌ها و بافت‌ها استفاده می‌شوند، در حالی که لایه‌های تماما متصل ارتباط مکانی را از بین می‌برند و بیشتر برای ترکیب ویژگی‌های استخراج‌شده و انجام تصمیم‌گیری نهایی، مانند طبقه‌بندی، مناسب‌اند. همچنین لایه‌های پیچشی به دلیل استفاده از کرنل‌های کوچک و وزن‌های اشتراکی تعداد پارامترهای کمتری دارند، اما در لایه‌های تماما متصل به دلیل اتصال کامل، پارامترهای بیشتری وجود دارد. لایه‌های پیچشی معمولا در مراحل اولیه شبکه برای یادگیری ویژگی‌های سطح پایین به کار می‌روند، در حالی که لایه‌های تماما متصل در انتهای شبکه برای یادگیری روابط پیچیده‌تر و تصمیم‌گیری نهایی استفاده می‌شوند.

ب) فیلترها یا هسته‌ها در لایه‌های پیچشی (Convolutional Layers) با اعمال عملیات کانولوشن بر روی ورودی، به استخراج ویژگی‌های مختلف تصویر کمک می‌کنند. هر فیلتر به یک بخش کوچک از تصویر (receptive field)اعمال می‌شود و با جابجایی در سراسر تصویر، الگوهای خاصی مانند لبه‌ها، گوشه‌ها، بافت‌ها یا ویژگی‌های سطح بالاتر را شناسایی می‌کند. وزن‌های فیلترها در طول فرآیند آموزش به گونه‌ای به‌روزرسانی می‌شوند که فیلترها بتوانند ویژگی‌های خاص و مهم تصویر را یاد بگیرند. در لایه‌های اولیه شبکه، فیلترها معمولا ویژگی‌های ساده‌تر مانند لبه‌ها یا رنگ‌ها را شناسایی می‌کنند، در حالی که در لایه‌های عمیق‌تر، این فیلترها قادر به شناسایی ویژگی‌های پیچیده‌تر مانند اشکال، اجزا و ساختارهای کلی تصویر می‌شوند. بنابراین، فیلترها نقش کلیدی در تفکیک اطلاعات مفید از تصویر و آماده‌سازی آن برای پردازش و طبقه‌بندی ایفا می‌کنند.

ج) یکی از معایب و محدودیت‌های اصلی شبکه‌های عصبی پیچشی (CNN) در وظایف شناسایی اشیا (Object Detection)این است که این شبکه‌ها به طور ذاتی برای طبقه‌بندی طراحی شده‌اند و مکان اشیای مختلف را در تصویر مشخص نمی‌کنند. در مدل‌های اولیه مانندR-CNN ، مشکل اصلی سرعت پایین در

پردازش تصاویر بود، زیرا این مدل برای هر ناحیه پیشنهادی (Region Proposal) یک‌بار شبکه عصبی را اجرا می‌کرد. این باعث افزایش زمان پردازش به ویژه برای تصاویر با تعداد زیادی ناحیه پیشنهادی می‌شد. علاوه بر این، فرآیند استخراج ویژگی‌ها و طبقه‌بندی ناحیه‌ها به صورت جداگانه انجام می‌شد که کارایی کلی مدل را کاهش می‌داد.

Fast R-CNN این مشکل را با اجرای عملیات کانولوشن برای کل تصویر به جای هر ناحیه پیشنهادی حل کرد و ویژگی‌های مشترک بین ناحیه‌های پیشنهادی را استخراج کرد، اما همچنان فرآیند تولید ناحیه‌های پیشنهادی (Region Proposals) خارج از شبکه و با استفاده از روش‌های سنتی مانند Selective Search انجام می‌شد که محدودیت سرعت را ایجاد می‌کرد.

این مشکلات باعث توسعه مدل‌هایی مانند Faster R-CNN شد که با معرفی شبکه‌ای به نام RPN (Region Proposal Network) فرآیند تولید ناحیه‌های پیشنهادی را به صورت یکپارچه در داخل شبکه انجام داد. بنابراین، محدودیت‌های سرعت و جداسازی مراحل پردازش درCNN ها دلیل اصلی توسعه این مدل‌های پیشرفته‌تر بوده است.

د) در شبکه‌های عصبی باقی‌مانده(ResNet) ، مفهوم اتصالات باقی‌مانده (Residual Connections) به گونه‌ای است که ورودی هر لایه یا بلوک مستقیماً به خروجی آن اضافه می‌شود. این ساختار باعث می‌شود که به جای یادگیری مستقیم نگاشت اصلی، شبکه تنها تغییرات یا "باقی‌مانده" بین ورودی و خروجی را مدل‌سازی کند. این رویکرد با فرمول $y=F(x)+x$ بیان می‌شود، که در آن x ورودی، $F(x)$ تابع یادگیری شده توسط لایه، و yyyخروجی است. اتصالات باقی‌مانده با فراهم کردن مسیر مستقیم برای انتقال گرادیان، مشکلاتی مانند ناپدید شدن گرادیان (Vanishing Gradient) را کاهش می‌دهند و یادگیری در شبکه‌های عمیق را تسهیل می‌کنند. علاوه بر این، این اتصالات یادگیری را بهینه‌تر می‌کنند زیرا شبکه تنها نیاز به مدل‌سازی تغییرات جزئی دارد. این معماری به شبکه امکان می‌دهد که عمیق‌تر شود (حتی تا ۱۵۲ لایه) بدون کاهش دقت یا مشکلات بهینه‌سازی، و دقت مدل را به طور چشمگیری افزایش می‌دهد. اتصالات باقی‌مانده به عنوان یک نوآوری کلیدی، امکان‌پذیری آموزش شبکه‌های بسیار عمیق را فراهم کرده و عملکرد مدل‌های یادگیری عمیق را بهبود بخشیده‌اند.

ه) برای مقابله با مسئله داده‌های کم در شبکه‌های عمیق، می‌توان از روش‌های مختلف regularization استفاده کرد. در ورودی داده‌ها، روش‌هایی مانند **افزایش داده‌ها (Data Augmentation)** با ایجاد تغییرات جزئی روی داده‌ها (مانند چرخش، تغییر مقیاس، برش، وارونگی یا تغییر روشنایی) و **نرمال‌سازی (Normalization)**برای مقیاس‌بندی ویژگی‌ها، تنوع داده‌ها را افزایش داده و عملکرد شبکه را بهبود

می‌دهند. در ساختار شبکه، استفاده از **Dropout** برای غیرفعال کردن تصادفی نورون‌ها در هر مرحله، **نرمال‌سازی دسته‌ای (Batch Normalization)** برای تثبیت یادگیری، و **انتظام‌بخشی وزن‌ها (Weight Regularization)**با اعمال جریمه L1 یا L2 روی وزن‌ها، از پیچیدگی بیش از حد مدل جلوگیری می‌کند. در خروجی شبکه، روش **توقف زودهنگام (Early Stopping)** با متوقف کردن آموزش هنگام ثابت شدن عملکرد روی داده‌های اعتبارسنجی و استفاده از **روش‌های ترکیبی Ensemble) (Methods**برای ترکیب چندین مدل و افزایش دقت، از بیش‌برازش جلوگیری می‌کنند. این روش‌ها با توجه به نوع داده و مسئله می‌توانند تعمیم‌پذیری مدل را به طور مؤثری افزایش دهند.

سوال چهارم

۵)

تابع هزینه انتخابی

تابع هزینه‌ای که در این مسئله استفاده شده است، **sparse_categorical_crossentropy** می‌باشد. این تابع برای مسائل دسته‌بندی چندکلاسه مناسب است و با داده‌هایی که برچسب‌های آنها به صورت عددی (مانند ۰ تا ۹ برای CIFAR-10) ذخیره شده‌اند، به خوبی کار می‌کند. در این حالت نیازی به تبدیل برچسب‌ها به فرمت یک-داغ one-hot) نیست. این تابع خطای پیش‌بینی مدل را با استفاده از احتمال پیش‌بینی‌شده کاهش می‌دهد، به طوری که مدل به سمت پیش‌بینی دقیق‌تر حرکت کند. استفاده از این تابع باعث بهینه‌سازی عملکرد در مسائل طبقه‌بندی چندکلاسه می‌شود.

تعداد نورون‌ها و تابع فعال‌ساز لایه‌ی آخر

در لایه‌ی آخر مدل، **10نورون** قرار دارد که معادل تعداد کلاس‌های موجود در مجموعه داده CIFAR-10 است. این تعداد نورون‌ها تضمین می‌کند که مدل برای هر کلاس یک خروجی مجزا داشته باشد. همچنین از تابع فعال‌ساز **softmax** استفاده شده است. این تابع مقادیر خروجی هر نورون را به احتمال هر کلاس تبدیل می‌کند و خروجی‌ها را به یک توزیع احتمالی نرمالیزه شده (جمع احتمالات برابر ۱) تبدیل می‌کند. این ویژگی برای دسته‌بندی چندکلاسه ضروری است، زیرا با احتمال بالا، کلاس درست پیش‌بینی می‌شود. انتخاب این ترکیب باعث می‌شود مدل بتواند خروجی‌ای قابل تفسیر ارائه دهد.

نرخ یادگیری و بهینه‌ساز(optimizer)

بهینه‌سازی در این مدل با استفاده از الگوریتم **Adam** انجام می‌شود .Adam ترکیبی از روش‌های RMSPropو SGDاست که با تطبیق نرخ یادگیری به صورت پویا برای هر پارامتر، سرعت همگرایی مدل را افزایش می‌دهد و به ثبات بیشتری در آموزش دست می‌یابد. نرخ یادگیری پیش‌فرض **0.001**استفاده شده است، که در بسیاری از مسائل یادگیری عمیق عملکرد مناسبی دارد. این مقدار در صورت نیاز می‌تواند با استفاده از یک کاهش‌دهنده نرخ یادگیری به تدریج تنظیم شود تا مدل در انتهای آموزش بتواند به نتایج بهتری دست یابد Adam .به دلیل ویژگی‌های تطبیقی خود، انتخاب مناسبی برای مسائل پیچیده مانند طبقه‌بندی تصاویر در CIFAR-10 است.

(۱۰

عملکرد شبکه روی داده‌های باکیفیت (32x32) نسبت به داده‌های رزولوشن پایین (16x16) به‌طور قابل توجهی بهتر بوده است. در حالت داده‌های باکیفیت، دقت مدل (High-res on High-res) برابر با **0.1166** است. این مقدار در حالت ارزیابی داده‌های رزولوشن پایین روی مدل آموزش‌دیده با داده‌های باکیفیت (Low-res on High-res)به **0.0946**کاهش می‌یابد.

مدل آموزش‌دیده با داده‌های رزولوشن پایین روی داده‌های رزولوشن پایین (Low-res on Low-res) دقتی برابر با **0.1133**نشان داده است، که مشابه عملکرد داده‌های باکیفیت روی رزولوشن پایین (High-res on Low-res)است. این کاهش دقت ناشی از از بین رفتن اطلاعات جزئی و بافت تصویر در داده‌های رزولوشن پایین است، که برای تفکیک دقیق کلاس‌ها بسیار حیاتی است. همچنین، مقادیر Precision و F1-Score نیز در داده‌های باکیفیت بیشتر از رزولوشن پایین بوده‌اند؛ به‌طور مثال، مقدار Precision برای High-res on High-resبرابر **0.1154**است، در حالی که برای Low-res on High-res این مقدار به **0.0919**کاهش یافته است.

برای بهبود عملکرد مدل در شرایط رزولوشن پایین، تکنیک‌های داده‌افزایی مانند چرخش، نویزگذاری و تغییر روشنایی می‌توانند تنوع داده‌ها را افزایش داده و مدل را در برابر شرایط مختلف مقاوم کنند. استفاده از معماری‌های پیشرفته‌تری مانند ResNet یا EfficientNet که قابلیت یادگیری بهتر و استخراج ویژگی‌های عمیق‌تر را دارند، توصیه می‌شود. همچنین، بازسازی تصاویر رزولوشن پایین با استفاده از شبکه‌های سوپررزولوشن می‌تواند اطلاعات ازدست‌رفته را بازیابی کرده و دقت را بهبود بخشد.

به‌عنوان مثال، استفاده از سوپررزولوشن می‌تواند مقادیر دقت را از **0.0946**در حالت Low-res on High-res به مقادیری نزدیک‌تر به **0.1166**در حالت High-res on High-res ارتقا دهد. علاوه بر این، استفاده از مکانیزم‌های توجه مانند SE-Block می‌تواند اطلاعات مهم را حتی در رزولوشن پایین استخراج کند. تنظیم

دقیق نرخ یادگیری و افزایش تعداد اپوک‌ها نیز می‌تواند به بهبود تدریجی دقت کمک کند. در مجموع، تأثیر کاهش رزولوشن بر دقت پیش‌بینی قابل توجه بوده است، اما با روش‌های پیشنهادی می‌توان این مشکل را تا حد زیادی کاهش داد.

گزارش نهایی

شبکه طراحی‌شده برای مجموعه داده CIFAR-10 با هدف طبقه‌بندی تصاویر به ده کلاس مختلف مورد ارزیابی قرار گرفت. این گزارش شامل نحوه آماده‌سازی داده‌ها، معماری شبکه، تنظیمات هایپرپارامترها، نتایج، و تحلیل دقیق تأثیر رزولوشن داده‌ها بر عملکرد مدل است.

تقسیم‌بندی دیتاست و دلایل انتخاب درصدها

برای آماده‌سازی داده‌ها، مجموعه CIFAR-10 به سه بخش تقسیم شد: ۷۰ درصد برای آموزش، ۱۵ درصد برای اعتبارسنجی و ۱۵ درصد برای تست. این تقسیم‌بندی به مدل اجازه می‌دهد که با داده‌های کافی آموزش ببیند، در حین آموزش بر روی داده‌های اعتبارسنجی ارزیابی شود و در نهایت عملکرد آن روی داده‌های تست بررسی شود. ۷۰ درصد داده‌ها برای آموزش انتخاب شدند تا مدل بتواند الگوهای پیچیده‌تری از داده‌ها یاد بگیرد. ۱۵ درصد داده‌های اعتبارسنجی برای جلوگیری از بیش‌برازش به کار گرفته شد و داده‌های تست به منظور ارزیابی نهایی مدل استفاده شدند.

معماری شبکه

معماری شبکه طراحی‌شده به صورت زیر است:

لایه	نوع	تعداد فیلتر / نورون	ابعاد فیلتر	تابع فعال‌ساز	جزئیات اضافی
ورودی	-	-	-	-	(32, 32, 3)
کانولوشن ۱	Conv2D	64	(3, 3)	ReLU	BatchNorm, Dropout
کانولوشن ۲	Conv2D	128	(3, 3)	ReLU	BatchNorm, Dropout
کانولوشن ۳	Conv2D	256	(3, 3)	ReLU	BatchNorm, Dropout
تخت‌سازی	Flatten	-	-	-	-
لایه کاملاً متصل	Dense	256	-	ReLU	BatchNorm, Dropout
خروجی	Dense	10	-	Softmax	-

این معماری شامل سه لایه کانولوشن برای استخراج ویژگی، یک لایه تخت‌سازی برای تبدیل داده‌های چندبعدی به یک بردار و دو لایه کاملاً متصل برای طبقه‌بندی است.

توضیح انتخاب هایپرپارامترها

برای این مدل، نرخ یادگیری اولیه ۰.۰۰۱ انتخاب شده است که مقدار پیش‌فرض برای بهینه‌ساز Adamاست. این نرخ یادگیری تطبیقی باعث می‌شود مدل با سرعت مناسب همگرا شود. بهینه‌ساز Adamانتخاب شد، زیرا ترکیبی از مزایای روش‌های SGD و RMSProp را دارد و می‌تواند نرخ یادگیری را برای هر پارامتر به صورت دینامیک تنظیم کند. تابع هزینه sparse_categorical_crossentropyبرای محاسبه خطا استفاده شد، زیرا این تابع برای مسائل طبقه‌بندی چندکلاسه بسیار مناسب است و با برچسب‌های عددی کار می‌کند.

نمودارهای Loss وAccuracy

نمودارهای زیر کاهش Loss و افزایش Accuracy را در طول آموزش نشان می‌دهند. این نمودارها مشخص می‌کنند که مدل در داده‌های باکیفیت عملکرد بهتری داشته است و داده‌های رزولوشن پایین باعث کاهش دقت و افزایش خطا شده‌اند. (نمودارها در فایل‌های همراه قرار دارند.)

نتایج و مقایسه آنها

نتایج ارزیابی مدل در چهار حالت مختلف به شرح زیر است:

حالت	Accuracy	Precision	F1-Score
High-res on High-res	0.116667	0.115470	0.112602
Low-res on High-res	0.094667	0.091958	0.085875
High-res on Low-res	0.113333	0.122447	0.107741
Low-res on Low-res	0.113333	0.122447	0.107741

این نتایج نشان می‌دهند که مدل در داده‌های باکیفیت عملکرد بهتری داشته و کاهش رزولوشن باعث کاهش دقت و کارایی مدل شده است.

تحلیل دقیق از تأثیر رزولوشن داده‌ها بر عملکرد مدل

کاهش رزولوشن داده‌ها باعث کاهش قابل توجه دقت مدل شده است. اطلاعات مهمی که در داده‌های باکیفیت موجود بودند، در داده‌های رزولوشن پایین از دست رفته‌اند و مدل نمی‌تواند ویژگی‌های کافی برای تفکیک کلاس‌ها استخراج کند. این امر به وضوح در ماتریس‌های آشفتگی مشاهده می‌شود، جایی که مدل در تشخیص صحیح برخی از کلاس‌ها دچار اشتباه شده است. همچنین، کاهش رزولوشن باعث افزایش نویز در ویژگی‌های استخراج‌شده شده و عملکرد مدل را کاهش داده است.

پیشنهادهایی برای بهبود عملکرد مدل

برای بهبود عملکرد مدل در شرایط رزولوشن پایین، چند پیشنهاد ارائه می‌شود. استفاده از تکنیک‌های داده‌افزایی مانند چرخش، نویزگذاری و برش می‌تواند به بهبود یادگیری کمک کند. بهره‌گیری از معماری‌های پیشرفته‌تر مانند ResNet یا MobileNet که توانایی یادگیری ویژگی‌های پیچیده‌تر را دارند، می‌تواند عملکرد مدل را افزایش دهد. بازسازی تصاویر رزولوشن پایین با استفاده از شبکه‌های سوپررزولوشن نیز می‌تواند اطلاعات ازدست‌رفته را بازیابی کند. در نهایت، استفاده از مکانیزم‌های توجه (Attention Mechanisms)و تنظیم دقیق هایپرپارامترها از جمله نرخ یادگیری و تعداد اپوک‌ها می‌تواند به بهبود نتایج منجر شود.

منابع