بسم الله الرحمن الرحیم

دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

یادگیری عمیق

نیم‌سال دوم ۰۴-۰۳

مدرس: مهدیه سلیمانی

تمرین دوم | ددلاین تمرین : ۱۵ فروردین

- برای ارسال هر تمرین تا ساعت ۲۳:۵۹ روز ددلاین فرصت دارید. مهلت تاخیر (مجاز و غیر مجاز) برای این تمرین، ۷ روز است (یعنی حداکثر تاریخ ارسال تمرین ۲۲ فروردین است)
- مجموع نمرات تمرین نظری برابر ۱۱۰ می باشد که ۱۰ نمره آن جنبه اختیاری دارد و گرفتن نمره ۱۰۰ دریافت نمره کامل بخش نظری کفایت میکند و البته اگر نمره ای بالاتر از ۱۰۰ کسب کنید همان نمره ۱۰۰ برای شما در نظر گرفته میشود
- در هر کدام از سوالات، اگر از منابع خارجی استفاده کرده‌اید باید آن را ذکر کنید. در صورت همفکری با افراد دیگر هم باید نام ایشان را در سوال مورد نظر ذکر نمایید.
- پاسخ تمرین باید ماحصل دانسته‌های خود شما باشد. در صورت رعایت این موضوع، استفاده از ابزارهای هوش مصنوعی با ذکر نحوه و مصداق استفاده بلامانع است.
- پاسخ ارسالی واضح و خوانا باشد. در غیر این صورت ممکن است منجر به از دست دادن نمره شود.
- پاسخ ارسالی باید توسط خود شما نوشته شده باشد. به اسکرین‌شات از منابع یا پاسخ افراد دیگر نمره‌ای تعلق نمی‌گیرد.
- در صورتی که بخشی از سوال‌ها را جای دیگری آپلود کرده و لینک آن را قرار داده باشید، حتما باید تاریخ آپلود مشخص و قابل اتکا باشد.
- محل بارگذاری سوالات نظری و عملی در هر تمرین مجزا خواهد بود. به منظور بارگذاری بایستی تمارین تئوری در یک فایل pdf با نام HW2_[First-Name]_[Last-Name]_[Student-Id].pdf و تمارین عملی نیز در یک فایل مجزای زیپ با نام HW2_[First-Name]_[Last-Name]_[Student-Id].zip بارگذاری شوند.
- در صورت وجود هرگونه ابهام یا مشکل، در کوئرای درس آن مشکل را بیان کنید و از پیغام دادن مستقیم به دستیاران آموزشی خودداری کنید.
- طراحان این تمرین : علی رحیمی اکبر-امیرحسین ایزدی-امیرحسین علمدار-محمد مهدی واحدی-مهرداد مهابادی

بخش نظری (۱۰+۱۰۰ نمره)

پرسش ۱. Batch Normalization (۲۰ نمره)

۱. نحوه انجام نرمال‌سازی بچ در شبکه های تماما متصل و شبکه های پیچشی را با یکدیگر مقایسه کنید. همچنین نحوه اعمال نرمال‌سازی بچ در مرحله آموزش و آزمایش را نیز با یکدیگر مقایسه کنید.

۲. به صورت خلاصه Covariate shift را توضیح دهید و توضیح دهید چرا در نرمالسازی بچ، Covariate shift مابین داده های آموزش و آزمایش منجر به ناپایدار شدن در نتایج مدل برای دادگان آزمایش می شود؟

پرسش ۱. Batch Normalization (۲۰ نمره)

۱. نحوه انجام نرمال‌سازی بچ در شبکه های تماما متصل و شبکه های پیچشی را با یکدیگر مقایسه کنید. همچنین نحوه اعمال نرمال‌سازی بچ در مرحله آموزش و آزمایش را نیز با یکدیگر مقایسه کنید.

توی شبکه های fully connected، فرآیند نرمالسازی بچ به صورت عادی بعد از خروجی wx+b اتفاق می افتد ولی در حالت convolutional باید بعد از اعمال فیلتر ها بر روی ورودی و feature map تولید شده، بعد نهایی از این فرآیند برابر است با اعمال نرمالیزیشن بر روی عمق های ایجاد شده.

در شبکه‌های Fully Connected، Batch Normalization پس از محاسبه‌ی خروجی لایه (یعنی Wx+b) اعمال می‌شود و برای هر ویژگی (Feature) در کل نمونه‌های یک Batch، میانگین و واریانس محاسبه و نرمال‌سازی انجام می‌شود.

در شبکه‌های Convolutional نیز Batch Normalization بعد از خروجی لایه‌ی کانولوشن و قبل از تابع فعال‌سازی اعمال می‌شود، اما میانگین و واریانس برای هر کانال (Channel) به صورت مشترک روی تمام مکان‌های فضایی (Height و Width) و نمونه‌های Batch محاسبه می‌شود.

در مرحله‌ی آموزش (Training)، میانگین و واریانس از همان Batch جاری محاسبه شده و علاوه بر آن، میانگین و واریانس متحرک (Running Mean و Running Variance) نیز به‌روزرسانی می‌شوند. اما در مرحله‌ی آزمایش (Inference/Test)، دیگر از آمار Batch استفاده نمی‌شود و از همان مقادیر Running Mean و Running Variance که در زمان آموزش ذخیره شده‌اند استفاده می‌شود تا خروجی مدل مستقل از Batch ورودی باشد.

۲. به صورت خلاصه Covariate shift را توضیح دهید و توضیح دهید چرا در نرمالسازی بچ، Covariate shift مابین داده های آموزش و آزمایش منجر به ناپایدار شدن در نتایج مدل برای دادگان آزمایش می شود؟

در ابتدا یادآوری Covariate Shift:

توزیع ورودی‌های یک مدل تغییر کند، در حالی که رابطه‌ی بین ورودی و خروجی (Label) ثابت بماند.

Batch Normalization برای هر Batch، خروجی لایه را نرمال می‌کند تا میانگین و واریانس آن تقریباً ثابت بماند. بنابراین:

- توزیع ورودی لایه‌های بعدی کمتر تغییر می‌کند.
- آموزش پایدارتر و سریع‌تر می‌شود.

از این بابت اگر توزیع داده‌های Test با داده‌های Train متفاوت باشد، مدلی که روی توزیع Train یاد گرفته است ممکن است نتواند روی توزیع جدید به خوبی تعمیم دهد و دقت آن کاهش پیدا کند.

۳. شبکه CNN ای را در نظر بگیرید که از بلاک‌هایی به فرم زیر استفاده می‌کند:

$$(\text{ConvLayer}) \rightarrow (\text{BatchNorm}) \rightarrow (\text{Activation})$$

آیا حذف بایاس (b) از لایه کانولوشن در کارکرد این شبکه اختلال ایجاد می‌کند؟ چرا؟ همچنین فرض کنید شبکه را آموزش داده‌ایم؛ آیا ضرب کردن وزن ها در یک عدد مانند α در زمان آزمایش (Inference) ، عملکرد شبکه را تغییر می‌دهد؟ ضرب کردن این ضریب در تمام درایه‌های ورودی شبکه چطور؟

خیر، چون بنابر قوانین میانگین، وقتی عددی به تمامی پارامتر ها اضافه میشود، به میانگین هم همان مقدار اضافه می شود و توی نرمالیزیشن وقتی ما این مقدار را کم می کنیم، اثر اضافه شدن رو از بین می بریم.

بله. اگر در زمان آزمون وزن‌ها یا ورودی در عدد α ضرب شوند، خروجی لایه تغییر می‌کند، اما BatchNorm در مرحله آزمون از Running Mean و Running Variance ذخیره‌شده در زمان آموزش استفاده می‌کند و این تغییر را جبران نمی‌کند؛ بنابراین توزیع داده‌ها با توزیعی که شبکه روی آن آموزش دیده متفاوت شده و عملکرد مدل ممکن است کاهش یابد.

۴. نشانی دهید که نرمال‌سازی بچ، باعث ایجاد نویزی در برآورد مقادیر گرادیان‌ها در مرحله آموزش می‌شود که به‌طور ضمنی یک منظور ساز است. این اثر را با اثر منظم سازی dropout مقایسه کنید.

در مرحله آموزش، Batch Normalization میانگین و واریانس را از Mini-Batch جاری محاسبه می‌کند. از آنجا که هر Mini-Batch متفاوت است، خروجی BatchNorm و در نتیجه گرادیان‌های محاسبه‌شده نیز کمی تغییر می‌کنند. این تغییرات به صورت نویز تصادفی در فرآیند آموزش ظاهر شده و باعث کاهش Overfitting و ایجاد اثر منظم‌سازی (Regularization) می‌شود.

در مقابل، Dropout با حذف تصادفی تعدادی از نورون‌ها در هر مرحله آموزش، مسیر عبور اطلاعات و گرادیان را تغییر می‌دهد و بدین ترتیب از وابستگی بیش از حد نورون‌ها به یکدیگر جلوگیری می‌کند. بنابراین هر دو روش اثر منظم‌سازی دارند، اما منشأ نویز در BatchNorm تغییر آمار Mini-Batch و در Dropout حذف تصادفی نورون‌ها است.

۵. بررسی کنید که آیا استفاده پشت سر هم بلوک نرمال‌سازی بچ و dropout عموما می‌تواند مفید باشد؟ چرا؟

معمولاً استفاده‌ی پشت سر هم از BatchNorm و Dropout خیلی مفید نیست، چون BatchNorm خودش تا حدی خاصیت regularization دارد و Dropout نیز با خاموش کردن تصادفی نورون‌ها توزیع فعال‌سازی را تغییر می‌دهد. بنابراین Dropout می‌تواند آمار mini-batch مورد استفاده در BatchNorm را ناپایدار کند (بخاطر دستکاری تعداد نورون های فعال و خاموش) و آموزش را سخت‌تر یا کندتر سازد.

۳. شبکه CNN ای را در نظر بگیرید که از بلاک‌هایی به فرم زیر استفاده می‌کند:

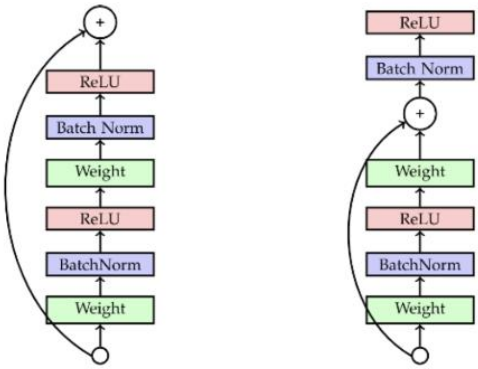
$$(\text{ConvLayer}) \rightarrow (\text{BatchNorm}) \rightarrow (\text{Activation})$$

آیا حذف بایاس (b) از لایه کانولوشن در کارکرد این شبکه اختلال ایجاد می‌کند؟ چرا؟ همچنین فرض کنید شبکه را آموزش داده‌ایم؛ آیا ضرب کردن وزن ها در یک عدد مانند α در زمان آزمایش (Inference) ، عملکرد شبکه را تغییر می‌دهد؟ ضرب کردن این ضریب در تمام درایه‌های ورودی شبکه چطور؟

۴. نشانی دهید که نرمال‌سازی بچ، باعث ایجاد نویزی در برآورد مقادیر گرادیان‌ها در مرحله آموزش می‌شود که به‌طور ضمنی یک منظور ساز است. این اثر را با اثر منظم سازی dropout مقایسه کنید.

۵. بررسی کنید که آیا استفاده پشت سر هم بلوک نرمال‌سازی بچ و dropout عموما می‌تواند مفید باشد؟ چرا؟

۶. شبکه‌های باقی‌مانده (ResNet) نقش مهمی در بهبود یادگیری عمیق داشته‌اند و امکان آموزش مدل‌های بسیار عمیق را فراهم کرده‌اند. با این حال، مکان قرارگیری نرمال‌سازی بچ (BN) نسبت به اتصالات میان‌بر تأثیر قابل توجهی بر پایداری آموزش، تعمیم‌پذیری و رفتار مدل در مرحله تست دارد. دو طراحی متفاوت برای بلوک باقی‌مانده را در نظر بگیرید که به بلوک باقی‌مانده با پیش‌فعال‌سازی و پس‌فعال‌سازی معروف است:

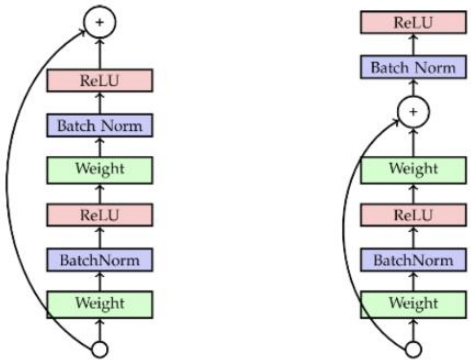


با در نظر گرفتن تأثیر نرمال‌سازی دسته‌ای بر انتشار گرادیان، پایداری بهینه‌سازی، سازگاری بین آموزش و تست، تغییر واریانس، یادگیری نمایش‌های عمیق، تحلیل کنید که چگونه مکان BN می‌تواند دینامیک آموزش شبکه را تحت تأثیر قرار دهد. (توجه کنید که منظور از Weight می تواند لایه تماما متصل و یا پیچشی باشد.)

پرسش ۲. Dilated Convolution (۱۵ نمره)

در شبکه های پیچشی به صورت متداول از لایه های کانولوشن ساده استفاده می شود که با آن آشنا هستید. نوع دیگری از لایه ها که می توان از آنان ددر شبکه های پیچشی استفاده نمود، لایه کانولوشن گسترش یافته یا متسع است. در شکل ۵ تصویر شهودی از فیلتر کانولوشن گسترش یافته ارائه شده است، این فیلتر ها میان خانه هایی که فیلتر با استفاده از اطلاعات آن‌ها لایه بعد را محاسبه می‌کنند فاصله می‌اندازند یا به بیانی دیگر در زمان اعمال فیلتر و انجام عملیات ضرب کانولوشن، بر روی ورودی با گام (dilated) بزرگتری حرکت می کنیم، توجه کنید طول گام مفهومی متفاوت نسبت به طول گام (stride) در لایه های شبکه کانولوشن دارد.

۶. شبکه‌های باقی‌مانده (ResNet) نقش مهمی در بهبود یادگیری عمیق داشته‌اند و امکان آموزش مدل‌های بسیار عمیق را فراهم کرده‌اند. با این حال، مکان قرارگیری نرمال‌سازی بچ (BN) نسبت به اتصالات میان‌بر تأثیر قابل توجهی بر پایداری آموزش، تعمیم‌پذیری و رفتار مدل در مرحله تست دارد. دو طراحی متفاوت برای بلوک باقی‌مانده را در نظر بگیرید که به بلوک باقی‌مانده با پیش‌فعال‌سازی و پس‌فعال‌سازی معروف است:



با در نظر گرفتن تأثیر نرمال‌سازی دسته‌ای بر انتشار گرادیان، پایداری بهینه‌سازی، سازگاری بین آموزش و تست، تغییر واریانس، یادگیری نمایش‌های عمیق، تحلیل کنید که چگونه مکان BN می‌تواند دینامیک آموزش شبکه را تحت تأثیر قرار دهد. (توجه کنید که منظور از Weight می تواند لایه تماما متصل و یا پیچشی باشد.)

در شبکه‌های عمیق، residual connection برای تسهیل یادگیری و جلوگیری از ضعف گرادیان استفاده می‌شود، چون به گرادیان اجازه می‌دهد یک مسیر مستقیم از لایه‌های بالاتر به لایه‌های پایین‌تر داشته باشد. در بلوک post-activation، BN و ReLU (سمت چپ) بعد از convolutionها و قبل از addition قرار می‌گیرند، اما در طراحی pre-activation، BN و ReLU قبل از convolution و قبل از جمع residual اعمال می‌شوند. معمولاً pre-activation باعث عبور بهتر گرادیان، پایداری بیشتر آموزش، و عملکرد بهتر در شبکه‌های خیلی عمیق می‌شود، چون مسیر identity تمیزتر باقی می‌ماند و residual branch راحت‌تر یاد گرفته می‌شود. بنابراین مکان BN نسبت به جمع residual روی optimization و generalization تأثیر مهمی دارد.

پرسش ۲. Dilated Convolution (۱۵ نمره)

در شبکه های پیچشی به صورت متداول از لایه های کانولوشن ساده استفاده می شود که با آن آشنا هستید. نوع دیگری از لایه ها که می توان از آنان ددر شبکه های پیچشی استفاده نمود، لایه کانولوشن گسترش یافته یا متسع است. در شکل ۵ تصویر شهودی از فیلتر کانولوشن گسترش یافته ارائه شده است، این فیلتر ها میان خانه هایی که فیلتر با استفاده از اطلاعات آن‌ها لایه بعد را محاسبه می‌کنند فاصله می‌اندازند یا به بیانی دیگر در زمان اعمال فیلتر و انجام عملیات ضرب کانولوشن، بر روی ورودی با گام (dilated) بزرگتری حرکت می کنیم، توجه کنید طول گام مفهومی متفاوت نسبت به طول گام (stride) در لایه های شبکه کانولوشن دارد.

$s_l = 1$

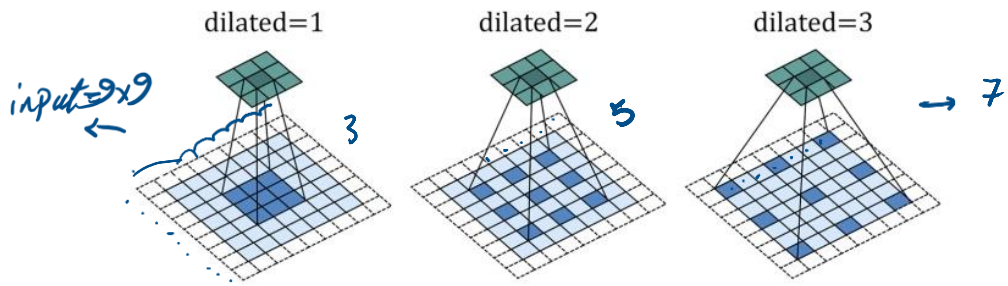
$k_l = 3 \qquad d_l = 1, 2, 3$

$k_l \qquad R^{(l)}$

$3 \rightarrow 7 - 5 - 3$

$5 \rightarrow 5 - 3 - 1$

$R^{(l)} = \frac{M - k_l - 2(d_l - 1)}{s_l} + 1$



شهودی از کانولوشن گسترش یافته با گام های متفاوت

همانطور که در شکل ۱ نیز مشخص است این روش، یک روش کم هزینه برای افزایش محدوده دید شبکه های پیچشی است. کانولوشن گسترش یافته بصورت فرم بسته ریاضی زیر تعریف می‌شود.

$$(K \star_D I)(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} K(m,n) I(i + Dm, j + Dn)$$

فرض کنید یک شبکه عصبی کانولوشنال با L لایه طراحی شده است که هر لایه شامل فیلترهای کانولوشن با پارامترهای زیر است:

- اندازه فیلتر (Kernel Size): $k_\ell \times k_\ell$ (مربعی)،
- نرخ اتساع (Dilation Rate): d_ℓ،
- گام (Stride): s_ℓ،
- بدون پدینگ (No Padding).

هدف ما بررسی تأثیر پارامترها بر **گستره دید نسبی** (Relative Receptive Field) است. گستره دید نسبی به صورت نسبت گستره دید در خروجی لایه Lـام به اندازه ورودی اصلی $(M \times N)$ تعریف می‌شود.

$$\boxed{R^{(L)} = \frac{M - k_l - 2(d_l - 1)}{s_l}}$$

۱. فرمول کلی گستره دید نسبی $R^{(L)}_{\text{relative}}$ را به صورت تابعی از k_ℓ، d_ℓ، و s_ℓ استخراج کنید.

۲. فرض کنید هدف ما این است که گستره دید نسبی بیشتر از یک حد آستانه (T) باشد، در حالی که هزینه‌های محاسباتی (FLOPs) کمترین مقدار ممکن باشد:

- معادله‌ای برای تعیین شرایط بهینه d_ℓ و s_ℓ بنویسید.
- آیا این شرایط بهینه به تعداد لایه‌ها (L) وابسته است؟ چرا؟

به صورت کلی یک trade off ای بین هزینه محاسبات با کیفیت بررسی مورد بحث قرار می گیرد، با افزایش تعداد s_l درسته که سایز feature map ها کوچکتر و هزینه محاسباتی کاهش یافته ولی از طرفی پیکسل های نادیده گرفته شده افزایش می یابند و همچنین با کوچکتر کردن

سایز فیلتر ها نیز با توجه به روابط خود کلاس درس که متوجه شدیم به صورت نمایی با سایز فیلتر ها در ارتباط هست بنابراین باید فیلتر های با سایز کم ولی تعداد بیشتر استفاده شود.

قطعاً هم به تعداد لایه ها وابسته خواهد بود، چرا که با افزایش تعداد لایه ها باید از فیلتر های جدید استفاده کرد و خب همین موضوع باعث افزایش موارد محاسباتی می شود.

پرسش ۳. ROI Alignment (۱۰ نمره)

۱. یکی از مسائل در برخی روش‌های تشخیص شی، ROI Alignment است. نحوه‌ی کار کرد درون‌یابی خطی و نزدیک‌ترین همسایه را توضیح دهید. برای درون‌یابی خطی روابط مربوطه را بنویسید.

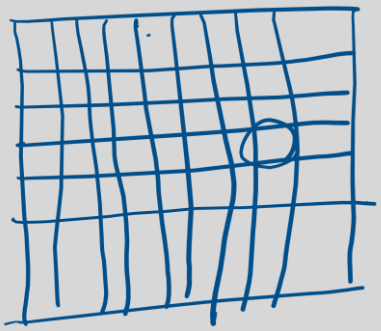
در روش‌های کلاسیک مانند ROI Pooling، برای استخراج ویژگی‌های یک ناحیه پیشنهادی (ROI) از نقشه ویژگی (Feature Map)، مجبور به گسسته‌سازی (Quantization) مختصات هستیم. به این صورت که مختصات floating-point مربوط به ROI (موقعیت شیء) به مقادیر صحیح (Integer) رُند می‌شوند. این رُند کردن (Rounding) باعث بروز «خطای عدم تراز» (Misalignment) بین فضای ورودی و فضای ویژگی می‌شود. این خطا در کارهای حساسی مثل Instance Segmentation، که نیازمند دقت پیکسلی است، منجر به افت شدید عملکرد می‌شود.

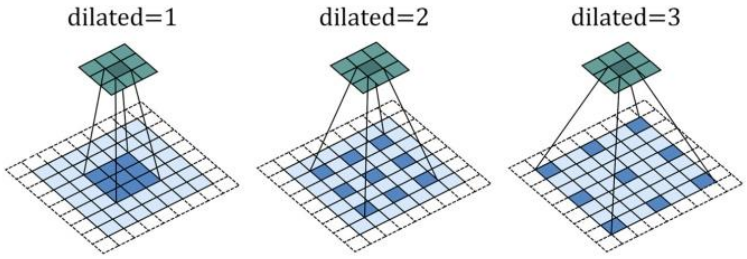
۲. یک عکس ۳۲ در ۳۲ را در نظر بگیرید، فرض کنید به یک activation map ۱۰ در ۱۰ تبدیل شده باشد. مقدار متناظر با نقطه‌ی x=۴ و y=۸ در عکس اولیه را در نقشه‌ی نهایی برحسب مقادیر پیکسل‌های نقشه محاسبه کنید.

فرض کنیم

$$\text{feature map} = \frac{N-F}{\text{stride}} + 1 \Rightarrow \frac{32-F}{\text{stride}} + 1 = 10 \Rightarrow \frac{27}{\text{stride}} = 9 \Rightarrow \text{stride} = 3$$

$i=1$ و $j=2$





شهودی از کانولوشن گسترش یافته با گام های متفاوت

همانطور که در شکل ۱ نیز مشخص است این روش، یک روش کم هزینه برای افزایش محدوده دید شبکه های پیچشی است. کانولوشن گسترش یافته بصورت فرم بسته ریاضی زیر تعریف می‌شود.

$$(K \star_D I)(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} K(m,n) I(i+Dm, j+Dn)$$

فرض کنید یک شبکه عصبی کانولوشنال با L لایه طراحی شده است که هر لایه شامل فیلترهای کانولوشن با پارامترهای زیر است:

- اندازه فیلتر (Kernel Size): $k_\ell \times k_\ell$ (مربعی)،
- نرخ اتساع (Dilation Rate): d_ℓ،
- گام (Stride): s_ℓ،
- بدون پدینگ (No Padding).

هدف ما بررسی تأثیر پارامترها بر **گستره دید نسبی** (Relative Receptive Field) است. گستره دید نسبی به صورت نسبت گستره دید در خروجی لایه Lـام به اندازه ورودی اصلی ($M \times N$) تعریف می‌شود.

۱. فرمول کلی گستره دید نسبی $R^{(L)}_{\text{relative}}$ را به صورت تابعی از k_ℓ، d_ℓ، و s_ℓ استخراج کنید.

۲. فرض کنید هدف ما این است که گستره دید نسبی بیشتر از یک حد آستانه (T) باشد، در حالی که هزینه‌های محاسباتی (FLOPs) کمترین مقدار ممکن باشد:

- معادله‌ای برای تعیین شرایط بهینه d_ℓ و s_ℓ بنویسید.
- آیا این شرایط بهینه به تعداد لایه‌ها (L) وابسته است؟ چرا؟

پرسش ۳. ROI Alignment (۱۰ نمره)

۱. یکی از مسائل در برخی روش‌های تشخیص شی، ROI Alignment است. نحوه‌ی کار کرد درون‌یابی خطی و نزدیک‌ترین همسایه را توضیح دهید. برای درون‌یابی خطی روابط مربوطه را بنویسید.

۲. یک عکس ۳۲ در ۳۲ را در نظر بگیرید، فرض کنید به یک activation map ۱۰ در ۱۰ تبدیل شده باشد. مقدار متناظر با نقطه‌ی x=۴ و y=۸ در عکس اولیه را در نقشه‌ی نهایی برحسب مقادیر پیکسل‌های نقشه محاسبه کنید.

پرسش ۴. Convolution Gradient (۱۵ نمره)

۱. بردار یک بعدی $\vec{x}$ با چهار درایه را در نظر بگیرید. فرض کنید روی این بردار یک کانولوشن یک بعدی با سایز کرنل ۳ و Padding ۱ اعمال کنیم. عملیات انجام شده را به فرم ماتریسی بنویسید و خروجی را محاسبه کنید.

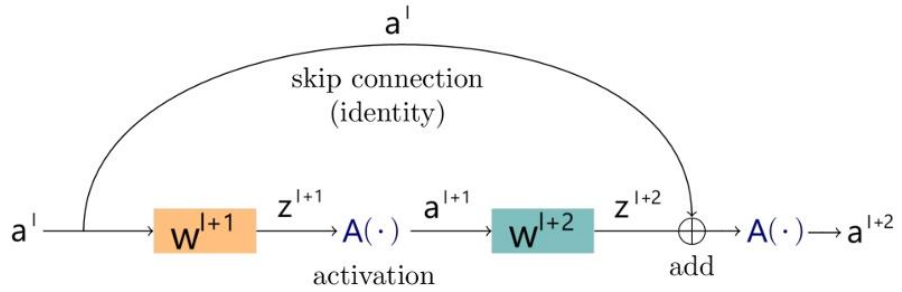
۲. حال فرض کنید یک تابع زیان روی خروجی این لایه اعمال شده و به زیان L رسیده‌ایم. مشتق L را نسبت به $\vec{x}$ با کمک قاعده زنجیره‌ای محاسبه کنید.

۳. به طور دقیق مشخص کنید باید چه عملیاتی روی مشتق جزئی تابع زیان نسبت به خروجی این لایه انجام دهیم تا مشتق جزئی زیان نسبت به بردار $\vec{x}$ بدست آید؟

۴. حال فرض کنید بردار $\vec{x}$ به طول چهار به شما داده شده است و می‌خواهید با کمک upsampling آن را به فضای $\mathbb{R}^6$ ببرید. برای اینکار از Transpose Convolution با padding صفر و stride یک استفاده می‌کنیم. اگر کرنل ما $\vec{w}$ به طول سه باشد عملیات را به فرم ماتریسی بنویسید. ماتریس حاصل را با ماتریس بخش ۱ مقایسه کنید. اگر padding یک باشد چه اتفاقی می‌افتد؟

پرسش ۵. محو شدن گرادیان (۱۵ نمره)

یکی از مشکلاتی که در الگوریتم های انتشار به عقب (back propagation) وجود دارد بحث محو شدن گرادیان (Gradient Vanishing) است. این موضوع مهم و قبل از اهمیت زیاد باعث عدم آموزش درست و کامل مدل در روند آموزش می‌شود. در این مسئله قصد داریم به بررسی این اتفاق بپردازیم.



۱. ابتدا مقدار $\frac{\partial a^l}{\partial a^{l+2}}$ را بدون در نظر گرفتن skip connection محاسبه کنید.

۲. حال یکی از راه حل ها که در بسیاری از مدل ها استفاده میشود در نظر گرفتن skip connection می‌باشد. این حالت چه کمکی به مدل می‌کند؟ $\frac{\partial a^l}{\partial a^{l+2}}$ با کمک قاعده زنجیره‌ای محاسبه کنید.

۳. با توجه به نتایج دو قسمت قبل بگویید که این الگوریتم به چه صورت می‌تواند مشکل محو شدن گرادیان را حل کند.(فرض کنید که داریم $W^i < 1 - \epsilon$)

پرسش ۶. MobileNet (۳۵ نمره)

معماری‌های MobileNet (شامل نسخه‌های V1، V2، و V3) از جمله شبکه‌های عصبی پیچشی سبک‌وزن هستند که به طور خاص برای اجرا بر روی دستگاه‌های کم‌مصرف مانند گوشی‌های هوشمند و سخت‌افزارهای لبه طراحی شده‌اند.

پرسش ۴. Convolution Gradient (۱۵ نمره)

۱. بردار یک بعدی $\vec{x}$ با چهار درایه را در نظر بگیرید. فرض کنید روی این بردار یک کانولوشن یک بعدی با سایز کرنل ۳ و Padding ۱ اعمال کنیم. عملیات انجام شده را به فرم ماتریسی بنویسید و خروجی را محاسبه کنید.

$$\begin{bmatrix} 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ 0 \end{bmatrix} * \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 + w_2 x_1 + w_3 x_2 \\ w_1 x_1 + w_2 x_2 + w_3 x_3 \\ w_1 x_2 + w_2 x_3 + w_3 x_4 \\ w_1 x_3 + w_2 x_4 + 0 \end{bmatrix}_{4\times 1}$$

صورت تمرین ⇐ ؟

(معکوس)

صورت ماتریسی ؟

$$\begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & w_3 \end{bmatrix}_{4\times 6} \times \overset{x}{\begin{bmatrix} 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ 0 \end{bmatrix}}_{6\times 1} = \begin{bmatrix} 0 + w_2 x_1 + w_3 x_2 \\ w_1 x_1 + w_2 x_2 + w_3 x_3 \\ w_1 x_2 + w_2 x_3 + w_3 x_4 \\ w_1 x_3 + w_2 x_4 + 0 \end{bmatrix}_{4\times 1}$$

۲. حال فرض کنید یک تابع زیان روی خروجی این لایه اعمال شده و به زیان L رسیده‌ایم. مشتق L را نسبت به $\vec{x}$ با کمک قاعده زنجیره‌ای محاسبه کنید.

$y = wx \rightarrow y' = w^T$

$$\frac{\partial L}{\partial x} = \frac{\partial y}{\partial x} \times \frac{\partial L}{\partial y} \xrightarrow[\text{ReLU}]{\text{با فرض}}$$

w^T ∇

بوده

Activation Function

$$\begin{bmatrix} w_1 & 0 & 0 & 0 \\ w_2 & w_1 & 0 & 0 \\ w_3 & w_2 & w_1 & 0 \\ 0 & w_3 & w_2 & w_1 \\ 0 & 0 & w_3 & w_2 \\ 0 & 0 & 0 & w_3 \end{bmatrix}_{6\times 4} \times \underbrace{\max(0, y)}_{\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}_{4\times 1}}_{4\times 1}$$

۳. به طور دقیق مشخص کنید باید چه عملیاتی روی مشتق جزئی تابع زیان نسبت به خروجی این لایه انجام دهیم تا مشتق جزئی زیان نسبت به بردار $\vec{x}$ بدست آید؟

بعد از ضرب دو ماتریس بالا ماتریسی با ابعاد 6x1 در می‌آید که هم بعد با X بوده که نشان دهنده عملیات تحلیلی است

۴. حال فرض کنید بردار $\vec{x}$ به طول چهار به شما داده شده است و می‌خواهید با کمک upsampling آن را به فضای $\mathbb{R}^6$ ببرید. برای اینکار از Transpose Convolution با padding صفر و stride یک استفاده می‌کنیم. اگر کرنل ما $\vec{w}$ به طول سه باشد عملیات را به فرم ماتریسی بنویسید. ماتریس حاصل را با ماتریس بخش ۱ مقایسه کنید. اگر padding یک باشد چه اتفاقی می‌افتد؟

$$\begin{bmatrix} w_1 & 0 & 0 & 0 \\ w_2 & w_1 & 0 & 0 \\ w_3 & w_2 & w_1 & 0 \\ 0 & w_3 & w_2 & w_1 \\ 0 & 0 & w_3 & w_2 \\ 0 & 0 & 0 & w_3 \end{bmatrix}_{6\times 4} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}_{4\times 1} = \begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix}_{6\times 1}$$

$\rightarrow w^T \rightarrow$ Transpose کرنل

ماتریس نمایی دوردیف صفر خواهد داشت

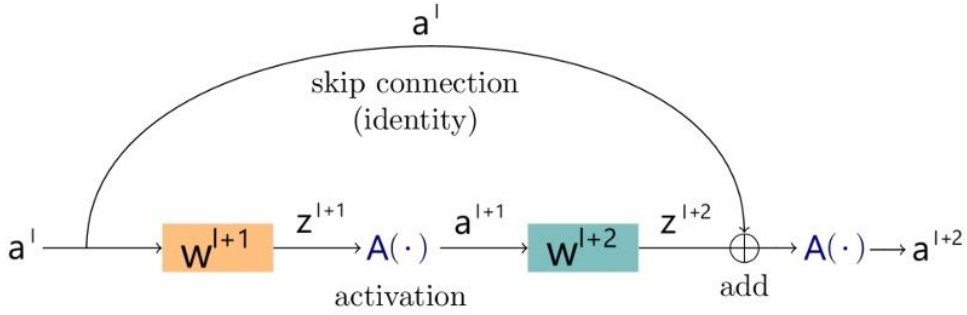
که اندیس اولشان آن می‌باشد، بنابراین

ماتریس $\vec{w}$ دو ردیف اول

و آخر تمام صفر خواهد داشت

پرسش ۵. محو شدن گرادیان (۱۵ نمره)

یکی از مشکلاتی که در الگوریتم های انتشار به عقب (back propagation) وجود دارد بحث محو شدن گرادیان (Gradient Vanishing) است. این موضوع مهم و قبل از اهمیت زیاد باعث عدم آموزش درست و کامل مدل در روند آموزش می‌شود. در این مسئله قصد داریم به بررسی این اتفاق بپردازیم.



۱. ابتدا مقدار $\frac{\partial a^l}{\partial a^{l+2}}$ را بدون در نظر گرفتن skip connection محاسبه کنید.

① $z^{l+1} = w^{l+1} a^l$

② $a^{l+1} = A(z^{l+1})$

③ $z^{l+2} = w^{l+2}(a^{l+1})$

④ $a^{l+2} = A(z^{l+2})$

$$\frac{\partial a^{l+2}}{\partial a^l} = \frac{\partial a^{l+2}}{\partial z^{l+2}} \times \frac{\partial z^{l+2}}{\partial a^{l+1}} \times \frac{\partial a^{l+1}}{\partial z^{l+1}} \times \frac{\partial z^{l+1}}{\partial a^l} = A'(z^{l+2}) \cdot (w^{l+2})^T \times A'(z^{l+1}) \cdot (w^{l+1})^T$$

۲. حال یکی از راه حل ها که در بسیاری از مدل ها استفاده میشود در نظر گرفتن skip connection می‌باشد. این حالت چه کمکی به مدل می‌کند؟ $\frac{\partial a^l}{\partial a^{l+2}}$ با کمک قاعده زنجیره‌ای محاسبه کنید.

1. $z^{l+1} = w^{l+1} a'$
2. $a^{l+1} = A(z^{l+1})$
3. $z^{l+2} = w^{l+2}(a^{l+1})$
4. $a^{l+2} = A(z^{l+2} + a')$

$$= w^{l+2}\left(A\left(w^{l+1} a'\right)\right)$$

براساس رابطه ۲ و ۳

$$\frac{\partial a^{l+2}}{\partial a'} = \frac{\partial a^{l+2}}{\partial (z^{l+2} + a')} \Rightarrow \frac{\partial a^{l+2}}{\partial a'} = \frac{\partial\left(w^{l+2}\left(A\left(w^{l+1} a'\right)\right) + a'\right)}{\partial a'}$$

$$\Downarrow$$

$$\frac{\partial a^{l+2}}{\partial a'} = A'(z^{l+2})\, w^{l+2} A'(z^{l+1})\, w^{l+1} + I$$

۳. با توجه به نتایج دو قسمت قبل بگویید که این الگوریتم به چه صورت می‌تواند مشکل محو شدن گرادیان را حل کند.(فرض کنید که داریم $W^i < 1 - \epsilon$)

با اضافه شدن $\acute{a}$ به روابط، مقدار گرادیان برخلاف مسیری که در سؤال یک طی می کند و پیوسته در حال ضرب شدن در مقادیر کمتر از یک هست، مقدار خالص خود را عبور داده و باعث تشدید گرادیان می شود.

این مدل‌ها با کاهش تعداد محاسبات و پارامترها، بدون افت چشمگیر در دقت، توانسته‌اند به تعادلی میان کارایی و عملکرد دست یابند. از مهم‌ترین نوآوری‌های به‌کاررفته در این معماری‌ها می‌توان به کانولوشن‌های عمقی قابل تفکیک (Depthwise Separable Convolutions)، بلوک‌های باقیمانده معکوس (Inverted Residual Blocks) دارای گلوگاه‌های خطی (Linear Bottlenecks)، مکانیزم‌های فشرده‌سازی و تحریک (SE Blocks) و استفاده از جستجوی معماری عصبی (NAS) اشاره کرد. در این سوال به بررسی برخی از این موارد می‌پردازیم.(لازم به ذکر است برای حل این سوال پیشنهاد اکید میشود از سرچ در منابع مختلف برای mobilenet بهره ببرید)

۱) کانولوشن‌های عمقی قابل تفکیک و تئوری تقریب
کانولوشن‌های عمقی قابل تفکیک سنگ بنای شبکه‌های پیچشی سبک‌وزن هستند که باعث کاهش قابل توجه هزینه‌های محاسباتی می‌شوند و در عین حال ظرفیت نمایشی منطقی را حفظ می‌کنند.

۱-۱. کانولوشن عمقی قابل تفکیک را بطور کامل حین مقایسه با کانولوشن عادی، توضیح دهید(به صورت ریاضی).

۱-۲. عمل ریاضی کانولوشن استاندارد را درنظر بگیرید و آن را بصورت مجموع تنسورهای رتبه یک با استفاده از تجزیه مقادیر تکین بیان کنید. توضیح دهید این تجزیه چگونه به ساختار کانولوشن عمقی قابل تفکیک مرتبط است.

۱-۳. نسبت کاهش FLOPs در کانولوشن‌های عمقی قابل تفکیک نسبت به کانولوشن‌های استاندارد را برای اندازه ورودی $H \times W$، تعداد کانال‌های ورودی C_{in}، تعداد کانال‌های خروجی C_{out} و اندازه فیلتر K را بطور پارامتری محاسبه کنید. آیا بین ظرفیت نمایشی و هزینه‌های محاسباتی ناشی از این تقریب تضادی وجود دارد؟

۲) بلوک‌های باقیمانده معکوس و تحلیل جریان گرادیان
بلوک‌های باقیمانده معکوس، که در MobileNetV2 معرفی شدند، با ترکیب باقی‌مانده معکوس و گلوگاه‌های خطی پیشرفتی در شبکه‌های پیچشی سبک‌وزن ایجاد کردند و باعث بهبود ظرفیت نمایشی در حالی که هزینه‌های محاسباتی کاهش می‌یابد، شدند.

۲-۱. بلوک‌های باقیمانده معکوس را در حین مقایسه با بلوک‌های باقیمانده عادی (ResNet)، توضیح دهید.

۲-۲. آیا این ادعا که کانولوشن‌های عمقی قابل تفکیک اطلاعات مکانی را حفظ می‌کنند، در حالی که گلوگاه‌های خطی اطلاعات کانالی را حفظ می‌کنند، صحیح است؟

۲-۳. تحلیل کنید که چگونه ضریب انبساط t بر جریان گرادیان و پایداری بهینه‌سازی تأثیر می‌گذارد. ثابت کنید که افزایش t خطر ناپدید شدن گرادیان‌ها را کاهش می‌دهد، اما هزینه‌های محاسباتی را افزایش می‌دهد. مقدار بهینه t را که تعادلی بین جریان گرادیان و کارایی ایجاد می‌کند، بر چه اساسی می‌توان انتخاب کرد؟

۳) مکانیزم‌های فشرده‌سازی و تحریک و بازنگری ویژگی‌های کانالی
بلوک‌های فشرده‌سازی و تحریک، که در MobileNetV3 استفاده شدند، تنظیم تطبیقی پاسخ‌های ویژگی در سطح کانال افزایش می‌دهند. این بلوک‌ها با استفاده از یک مکانیزم توجه (attention) وزن‌های کانال‌ها را مطابق با اطلاعات جهانی موجود در نقشه‌های ویژگی ورودی خود تنظیم می‌کنند.

۳-۱. اعمالی که در یک بلوک فشرده‌سازی و تحریک در یک لایه رخ می‌دهد را به صورت ریاضی فرمول بندی کنید. فرض کنید نقش ویژگی ورودی این بلوک با ابعاد $C \times H \times W$ است.

۳-۲. ثابت کنید که بلوک‌های SE ظرفیت نمایشی را با بازنگری تطبیقی پاسخ‌های ویژگی‌های کانالی بهبود می‌دهند. از تئوری اطلاعات برای اندازه‌گیری اطلاعات متقابل بین ورودی و خروجی بلوک SE استفاده کنید. توضیح دهید که ضریب کاهش r چگونه بر تضاد بین هزینه‌های محاسباتی و عملکرد تأثیر می‌گذارد.

۳-۳. FLOPs اضافه شده توسط یک بلوک SE با $r = 16$ برای $C = 64$ را محاسبه کنید. این مقدار را با کل FLOPs یک کانولوشن عمقی قابل تفکیک با پارامترهای مشابه مقایسه کنید.

پرسش ۶. MobileNet (۳۵ نمره)

معماری‌های MobileNet (شامل نسخه‌های V1، V2، و V3) از جمله شبکه‌های عصبی پیچشی سبک‌وزن هستند که به طور خاص برای اجرا بر روی دستگاه‌های کم‌مصرف مانند گوشی‌های هوشمند و سخت‌افزارهای لبه طراحی شده‌اند. این مدل‌ها با کاهش تعداد محاسبات و پارامترها، بدون افت چشمگیر در دقت، توانسته‌اند به تعادلی میان کارایی و عملکرد دست یابند. از مهم‌ترین نوآوری‌های به‌کاررفته در این معماری‌ها می‌توان به کانولوشن‌های عمقی قابل تفکیک (Depthwise Separable Convolutions)، بلوک‌های باقیمانده معکوس (Inverted Residual Blocks) دارای گلوگاه‌های خطی (Linear Bottlenecks)، مکانیزم‌های فشرده‌سازی و تحریک (SE Blocks) و استفاده از جستجوی معماری عصبی (NAS) اشاره کرد. در این سوال به بررسی برخی از این موارد می‌پردازیم.(لازم به ذکر است برای حل این سوال پیشنهاد اکید میشود از سرچ در منابع مختلف برای mobilenet بهره ببرید)

۱) کانولوشن‌های عمقی قابل تفکیک و تئوری تقریب
کانولوشن‌های عمقی قابل تفکیک سنگ بنای شبکه‌های پیچشی سبک‌وزن هستند که باعث کاهش قابل توجه هزینه‌های محاسباتی می‌شوند و در عین حال ظرفیت نمایشی منطقی را حفظ می‌کنند.

۱-۱. کانولوشن عمقی قابل تفکیک را بطور کامل حین مقایسه با کانولوشن عادی، توضیح دهید(به صورت ریاضی).

در طی فرآیند کانولوشن عمقی قابل تفکیک، فرآیند فیلترینگ استاندارد به دو بخش تقسیم می شود، که عبارتست از:

- Depthwise convolution: که میاد و روی هر چنل ورودی یک فیلتر اعمال می کند و خب به صورت ریاضی یعنی C_{in} تا فیلتر $k \times k$ سایز داریم.
- Pointwise Convolution: به این شکل انجام گرفته می شود که برای ترتیب دادن به کانال ها، به از فیلتر سایز های 1×1 روی خروجی مرحله قبل اعمال می کنیم تا به حد مورد نظر برسد.

تعداد پارامتر ها در حالت استاندارد برابر: $C_{out} \times C_{in} \times K^2$ و در حالت تفکیکی: $C_{in} \times (K^2 + C_{out})$ می باشد.

۱-۲. عمل ریاضی کانولوشن استاندارد را درنظر بگیرید و آن را بصورت مجموع تنسورهای رتبه یک با استفاده از تجزیه مقادیر تکین بیان کنید. توضیح دهید این تجزیه چگونه به ساختار کانولوشن عمقی قابل تفکیک مرتبط است.

در تجزیه SVD (Singular Value Decomposition) برای تنسورها، ما تلاش می‌کنیم یک عملگر پیچیده را به صورت ترکیبی از عملگرهای مرتبه پایین‌تر بنویسیم. در حالت Depthwise، کانولوشن استاندارد را به دو عملگر رتبه-یک تقریب می‌زنیم:

- یک تنسور فضایی $(K \times K \times C_{in} \times 1)$
- یک تنسور کانالی (Channel wise) $(1 \times 1 \times C_{in} \times C_{out})$

این دقیقاً همان تجزیه رتبه-پایین (Low-rank approximation) است. با این کار، ما به جایِ داشتنِ یک ماتریس عظیمِ $K \times K \times C_{in} \times C_{out}$ (که اکثرِ اطلاعاتِ آن در واقعیتِ فیزیکیِ تصاویر همبستگی بالایی دارند و رتبه‌اش بالاست)، آن را به دو ماتریس بسیار کوچک‌تر فاکتور می‌گیریم که با ترکیبِ هم، خروجیِ مشابهی تولید می‌کنند.

۱-۳. نسبت کاهش FLOPs در کانولوشن‌های عمقی قابل تفکیک نسبت به کانولوشن‌های استاندارد را برای اندازه ورودی $H \times W$، تعداد کانال‌های ورودی C_{in}، تعداد کانال‌های خروجی C_{out} و اندازه فیلتر K را بطور پارامتری محاسبه کنید. آیا بین ظرفیت نمایشی و هزینه‌های محاسباتی ناشی از این تقریب تضادی وجود دارد؟

$$\frac{FLOP_{depthwise}}{FLOP_{standard}} = \frac{H \times W \times C_{in} \times (K^2 + C_{out})}{H \times W \times C_{in} \times C_{out} \times K^2} = \frac{1}{C_{out}} + \frac{1}{K^2}$$

خب قطعاً Trade off ای وجود داره وقتی به اندازه K^2 ما کاهش محاسبات داریم، یعنی داریم حالت یا محاسباتی را به صورت Naive یا قابل در نظر گرفته نشدن میگیریم، در این مورد ما داریم خصوصیات تصویر رو مستقل از تعداد چنل ها می گیریم که توی پروژه هایی مثل مسائل پزشکی و تصویر نگاری هایی که علاوه بر خصوصیات فضایی بر خصوصیات چنلی هم وابستگی وجود دارد.

۲) بلوک‌های باقیمانده معکوس و تحلیل جریان گرادیان

بلوک‌های باقیمانده معکوس، که در MobileNetV2 معرفی شدند، با ترکیب باقی‌مانده معکوس و گلوگاه‌های خطی پیشرفتی در شبکه‌های پیچشی سبک‌وزن ایجاد کردند و باعث بهبود ظرفیت نمایشی در حالی که هزینه‌های محاسباتی کاهش می‌یابد، شدند.

۲-۱. بلوک‌های باقیمانده معکوس را در حین مقایسه با بلوک‌های باقیمانده عادی (ResNet)، توضیح دهید.

توی حالت ResNet معمولی، ما میایم تعداد چنل هارو از زیاد به کمتر و بعد دوباره به بزرگ تبدیل (Transpose) می کنیم و Skip Connection می زنیم ولی توی ResNet معکوس، برعکس این کار انجام میشه، یک افزایش تعداد چنل در وسط داریم $Expansion: c \rightarrow tc$ و بعد کوچیکش می کنیم $Projection: tc \rightarrow c$ ، که خب این کار با یک فاکتور t روی تعداد چنل ورودی انجام میشود،

۲-۲. آیا این ادعا که کانولوشن‌های عمقی قابل تفکیک اطلاعات مکانی را حفظ می‌کنند، در حالی که گلوگاه‌های خطی اطلاعات کانالی را حفظ می‌کنند، صحیح است؟

بله چون همونطور که در بخش Depth Wise Convolutions مشاهده کردیم، معادلات جوری نوشته شدند که دو بعد عمق و ویژگی های مکانی (Spatial) رو مستقل از هم در نظر میگیرند و توی گلوگاه های خطی، چون داریم با یک ورودی که از Channel Wise مقدار کوچکتری شده با فاکتور t ، بنابراین اطلاعات به خودی خود کمتر میشوند، پس اگر از Activation Function هایی مثل ReLU استفاده کنیم که ممکن است باعث کاهش اطلاعات شوند (توی ReLU چون $\max(0, y)$ داریم پس منفیا صفر میشن) از Linear Actiavation Function استفاده میکنیم.

۲-۳. تحلیل کنید که چگونه ضریب انبساط t بر جریان گرادیان و پایداری بهینه‌سازی تأثیر می‌گذارد. ثابت کنید که افزایش t خطر ناپدید شدن گرادیان‌ها را کاهش می‌دهد، اما هزینه‌های محاسباتی را افزایش می‌دهد. مقدار بهینه t را که تعادلی بین جریان گرادیان و کارایی ایجاد می‌کند، بر چه اساسی می‌توان انتخاب کرد؟

خب اول باید یادآوری کنیم از ماهیت اینکه چرا اصلاً ما با فاکتور t به بیشتر سازی چنل های ورودی کردیم، که هدف افزایش ضرایب وزن دار ورودی و بررسی های عمقی مکانی بود، اما نباید هدف اصلی ساختار MobileNet که ایجاد معماری سبک برای انجام فرآیند های سنگین

بود فراموش شود، قطعاً وقتی با فاکتور های t بزرگ تری انجام می دهیم عملیات رو، از وزن های بیشتری هم استفاده می کنیم که درست است هر کدام تأثیر جداگانه ای از گرادیان رو به عهده می گیرند، اما نباید به ضرر اصل موضوع که هدف سبک بودن عملیات است تمام شود.

۳) مکانیزم‌های فشرده‌سازی و تحریک و بازنگری ویژگی‌های کانالی
بلوک‌های فشرده‌سازی و تحریک، که در MobileNetV3 استفاده شدند، تنظیم تطبیقی پاسخ‌های ویژگی در سطح کانال افزایش می‌دهند. این بلوک‌ها با استفاده از یک مکانیزم توجه (attention) وزن‌های کانال‌ها را مطابق با اطلاعات جهانی موجود در نقشه‌های ویژگی ورودی خود تنظیم می‌کنند.

۳-۱. اعمالی که در یک بلوک فشرده‌سازی و تحریک در یک لایه رخ می‌دهد را به صورت ریاضی فرمول بندی کنید. فرض کنید نقش ویژگی ورودی این بلوک با ابعاد $C \times H \times W$ است.

در صورت سؤال تنها دو گام درخواست شده، در منابع و مراجع، سه گام ارائه شده:

۱. گام اول: عملیات فشرده سازی (Squeeze):

$$z_c = \underbrace{\frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} v_c(i,j)}_{average\ of\ the\ pixels}$$

z_c خروجی یک کانال که به صورت اسکالر هستش که نماینده کل اطلاعات اون کانال هست

$v_c(i,j)$ بیانگر پیکسل سطر i و ستون j هستش

۲. گام دوم: عملیات تحریک (Excitation):

اهمیت هر کانال (اسکالر های مرحله قبل) نسبت به بقیه:

$$s = \sigma(w_2 . \delta(w_1 . z))$$

Z بیانگر بردار خروجی از مرحله ی قبل به این مرحله

w_1 بیانگر ماتریس وزن‌های اولین لایه کاملاً متصل (FC). ابعاد آن $\frac{c}{r} \times C$ است. وظیفه آن کاهش ابعاد برای ایجاد گلوگاه (Bottleneck) و کاهش پیچیدگی محاسباتی است.

rr (Reduction Ratio): یک هایپرپارامتر (معمولاً ٤ در MobileNetV3) که نرخ فشرده‌سازی کانال‌ها را تعیین می‌کند.

δ (ReLU): تابع فعال‌ساز غیرخطی که برای مدل‌سازی روابط پیچیده بین کانال‌ها استفاده می‌شود.

w_2: ماتریس وزن‌های دومین لایه FC. ابعاد آن $C \times \frac{c}{r}$ است. وظیفه آن بازگرداندن ابعاد به تعداد کانال‌های اصلی (C) است.

σ: تابع فعالسازی سیگموئیدی که برای درجه بندی اهمیت خروجی از 0 (کاملاً بی اهمیت) تا ۱ (مهمترین) است.

s: بردار خروجی شامل C وزن توجه (Attention Weights) برای هر کانال.

۳. گام مقیاس‌دهی (Scale):

در این مرحله، وزن‌های یادگیری شده روی داده‌های اصلی اعمال می‌شوند.

$$\widetilde{X_c} = s_c \,.\, v_c$$

خروجی نهایی کانال c–ام. در واقع تمام مقادیر موجود در ماتریس v_c در عدد s_c ضرب می‌شوند.

۳-۲. ثابت کنید که بلوک‌های SE ظرفیت نمایشی را با بازنگری تطبیقی پاسخ‌های ویژگی‌های کانالی بهبود می‌دهند. از تئوری اطلاعات برای اندازه‌گیری اطلاعات متقابل بین ورودی و خروجی بلوک SE استفاده کنید. توضیح دهید که ضریب کاهش r چگونه بر تضاد بین هزینه‌های محاسباتی و عملکرد تأثیر می‌گذارد.

ظرفیت نمایشی یک شبکه را می‌توان توانایی آن در مدل‌سازی تابع نگاشت بین ورودی X و خروجی $\tilde{X}$ دانست. در یک لایه کانولوشن استاندارد، عملیات به صورت Y=W × X است که در آن وزن‌ها (W) برای تمام ورودی‌ها ثابت هستند. اما در بلوک SE، این عملیات به صورت مشروط (Conditional) در می‌آید:

$$\tilde{X} = diag(s(X)).X$$

که در آن $s(X)$ بردار توجه (Attention) است و خودش تابعی از X می‌باشد.

تحلیل اطلاعات متقابل (Mutual Information):

اگر $I(X;\tilde{X})$ را اطلاعات متقابل بین ورودی و خروجی در نظر بگیریم:

۱. در حالت کانولوشن ثابت: نگاشت کاملاً خطی و ایستا است. این یعنی فضای فرضیات (Hypothesis Space) محدود به ماتریس‌های وزن است.
۲. در حالت بلوک SE: ما یک وابستگی غیرخطی معرفی کرده‌ایم. با محاسبه بردار ss (که از طریق Squeeze به zz می‌رسد و سپس از طریق Excitation به ss تبدیل می‌شود)، شبکه در واقع یک فیلتر فشرده‌سازی اطلاعات ایجاد می‌کند.
۳. $I(X;\tilde{X})$ در بلوک SE نسبت به کانولوشن معمولی افزایش می‌یابد؛ چرا که بلوک SE اجازه می‌دهد «اطلاعات مربوط به زمینه کلی» (Global Context) در وزن‌دهی دخالت کند.
۴. در تئوری اطلاعات، این عمل مانند یک کانال با نویز متغیر است که در آن «گین» (Gain) هر کانال بر اساس اطلاعات ورودی تنظیم می‌شود. این کار باعث کاهش آنتروپی نویز (کانال‌های غیرمرتبط) و افزایش اطلاعاتِ مفید (Signal-to-Noise Ratio) در خروجی $\tilde{X}$ می‌شود.

۵. بنابراین، بلوک SE باعث می‌شود فضای خروجی $\tilde{X}$ بسیار غنی‌تر (Discriminative) از ورودی باشد، زیرا شبکه اکنون می‌تواند بر اساس «محتوا»، کانال‌های مهم را تقویت کند.

تعداد پارامترها در لایه‌های FC بلوک SE برابر است با:

$$Params \approx 2.\frac{C^2}{r}$$

با افزایش زیاد r ، باعث کاهش مقدار محاسبات می شود ولی شبکه ممکن است نتواند وابستگی های پیچیده بین کانال ها را یاد بگیرد، از طرفی اگر r را کوچک بگیریم گلوگاه شبکه بسیار بزرگ شده و بنابراین ظرفیت محساباتی افزایش پیدا می کند. نقطه بهینه (Sweet Spot)، طراحان MobileNetV3 با بررسی تجربی دریافتند که مقدار r=4 بهترین تعادل است. در این مقدار، شبکه فضای کافی برای حفظ وابستگی‌های کانالی (Mutual Information) را دارد، بدون اینکه هزینه محاسباتی (FLOPs) از حد استاندارد برای سخت‌افزارهای موبایل فراتر رود.

۳-۳. FLOPs اضافه شده توسط یک بلوک SE با $r = 16$ برای $C = 64$ را محاسبه کنید. این مقدار را با کل FLOPs یک کانولوشن عمقی قابل تفکیک با پارامترهای مشابه مقایسه کنید.

1) SE و Scale :

الف) بخش Squeeze : $\underbrace{H \times W}_{\text{Global sum}} \times C = 64\,HW$

ب) Excitation $\Rightarrow 4 \times C \times \frac{C}{r} = 4 \times 4^3 \times \frac{4^3}{4^2} = 4^5$

پ) Scale $\Rightarrow$ ضرب هر H و W در C

$64\,HW = C\,HW$

SE فرآیند کل $\Rightarrow 128\,HW + 4^5$

2) Depthwise : الف) filtering based on K kernel size : $H \times W \times C \times K^2$

ب) 1x1 kernel : $H \times W \times C^2 = H \times W \times 64^2$

C_{in} و C_{out}

$\Rightarrow$ sum : $64\,HW(64+K^2)$

از $C=64, k=3$ و $r=16$ و $H, W=28 \times 28$ $\rightarrow$

$$\frac{SE_{FLOPs}}{Depthwise_{flops}} = \frac{128\,HW + 4^5}{64\,HW(64+9)} = \frac{101376}{3662848} = 2.8\%$$

۴) جستجوی معماری عصبی (NAS) و تکنیک کوچک‌سازی پیش‌رونده
جستجوی معماری عصبی (NAS) برای بهینه‌سازی معماری MobileNetV3 استفاده شد و منجر به بهترین‌سطح عملکرد در دستگاه‌های موبایل شد.

۴-۱. این فرآیند را بطور کامل تحلیل کنید. فضای جستجو برای NAS در MobileNetV3 چه بوده است؟ از نظریه گراف برای مدل‌سازی فضای جستجو به عنوان یک گراف جهت‌دار بدون دور (DAG) چگونه می‌توان استفاده کرد؟

طراحی MobileNetV3 حاصل ترکیب دو تکنیک بهینه‌سازی مکمل است که هرکدام در دو مقیاس جداگانه ی کلی و جزئی عمل می‌کنند:

۱. مرحله اول (بهینه‌سازی کلان – MnasNet):

- از یک الگوریتم یادگیری تقویت‌شده (Reinforcement Learning) با عاملِ کنترل‌کنندهٔ RNN استفاده شد تا معماری‌های کلی (بلوک‌ها) را در سطح کلان شبکه جستجو کند. تابع پاداش چندهدفه (Multi-Objective Reward Function) که هدف هم‌زمان بیشینه‌سازی دقت (Accuracy) و کمینه‌سازی زمان تاخیر روی سخت‌افزار واقعی (Target Latency) را برعهده دارد:

$$Reward(\alpha) = ACC(\alpha) \times \left[\frac{Latency(\alpha)}{Tar}\right]^{\beta}$$

که در آن α یک معماری کاندید، Tar زمان تأخیر هدف، و β یک ضریب وزنی (معمولاً منفی و برای کنترل حساسیت زمان تاخیر) است.

(به دید اینکه با افزایش دقت، trade off ای بین تأخیر آن بوجود می آید بهش نگاه کن)

۲. مرحله دوم (بهینه‌سازی خرد – NetAdapt):

- پس از یافتن ساختار کلی توسط MnasNet، این الگوریتم به صورت لایه‌به‌لایه ظرفیت کانال‌ها و پهنای فیلترها را کوچک می‌کند. در هر گام، تعداد کانال‌های یک لایه کاهش می‌یابد و تاخیر شبکه روی سخت‌افزار بررسی می‌شود. لایه‌ای که با کمترین افت دقت بیشترین کاهش تاخیر را بدهد، تغییر شکل می‌یابد. این روند تا رسیدن به محدودیت تاخیر نهایی تکرار می‌شود.

فضای جستجو، مجموعه تمام معماری های ممکن است که الگوریتم NAS مجاز به انتخاب از میان آن‌هاست. برای هر بلوک در MobileNetV3، فضای تصمیم‌گیری شامل گزینه‌های زیر بود:

- نوع عملیات (Operator Type):
  - کانولوشن عمقیِ قابل‌تفکیک (DSC)
  - بلوک Inverted Residual (یادآوری MobeliNet2) همراه با یا بدون بلوک‌های Squeeze-and-Excitation (SE).
- اندازه کرنل کانولوشن عمقی ($K \times K$): انتخاب از بین فیلترهای 3×3 , 5×5 , 7×7.
- ضریب انبساط (t): مقادیر متغیر برای نرخ انبساط کانال میانی (مانند ۳، ٤، ٦).
- تعداد کانال خروجی (C): انتخاب ابعاد فیلترها در هر لایه.

- تابع فعال‌ساز (Activation Function): انتخاب بین ReLU و Hard-Swish (h-swish).
- عمق شبکه (NN): تعداد تکرار هر بلوک در مراحل مختلف شبکه.

در ریاضیات و نظریه گراف، یک شبکه عصبی پیش‌سو (Feed-forward) را می‌توان به صورت یک گراف جهت‌دار بدون دور (Directed Acyclic Graph - DAG) با نماد *G=(V,E)* مدل‌سازی کرد.

الف) تعریف مولفه‌های گراف:

- رئوس (*V*): هر راس $v_i \in V$ نشان‌دهنده یک تانسور ویژگی (Feature Tensor) یا حالت میانی در شبکه است. راس ورودی شبکه v_{in} و راس خروجی v_{out} نام دارد.
- یال‌ها (*E*): هر یال جهت‌دار $e_{ij} = (v_i , v_j)$ نشان‌دهنده یک عملیات محاسباتی (Operation) (مانند کانولوشن، پولینگ، یا تابع همانی) است که تانسور v_i را به تانسور v_j تبدیل می‌کند:

$$v_j = o^{(i,j)}(v_i)$$

(ب) فرمول‌بندی جریان اطلاعات در گراف:

برای هر گره j در گراف، مقدار تانسور از جمع اثر تمامی گره‌های قبلی متصل به آن محاسبه می‌شود:

$$v_j = \sum_{i<j} o^{(i,j)}(v_i)$$

ترتیب محاسبات از v_1 تا v_n بر اساس مرتب‌سازی توپولوژیک (Topological Sorting) گراف انجام می‌شود. عدم وجود دور (Acyclic) تضمین می‌کند که محاسبات دچار چرخه باطل و وابستگی متقابل نمی‌شوند.

ج) نحوه استفاده از DAG برای مدل‌سازی فضای جستجو (Super-DAG / Supernet):

در الگوریتم‌های مدرن NAS (مانند DARTS یا One-Shot NAS)، برای پیدا کردن معماری بهینه، یک ابرگراف (Supernet) طراحی می‌شود:

۱. تعریف یال‌های چندگانه (Candidate Operations):بین هر دو راس v_i و v_j، به‌جای یک عملیات واحد، مجموعه‌ای از عملیات کاندید *O* (مثل Conv3x3، Conv5x5، Identity، zero) قرار می‌گیرد.
۲. پلاستیسیته و بهینه‌سازی پیوسته:خروجی یال *(i,j)* به صورت یک ترکیب خطی وزندار از تمام عملیات کاندید فرمول‌بندی می‌شود:

$$\bar{o}^{(i,j)}(v_i) = \sum_{o \in O} \alpha_o^{(i,j)} . o(v_i)$$

که در آن $\alpha_o^{(i,j)}$ پارامترهای معماری (Architecture Parameters) هستند.

فرآیند هرس (Pruning) گراف:در پایان فرآیند جستجو، با تحلیل پارامترهای α، یال‌هایی که وزن ناچیزی دارند هرس می‌شوند و برای هر یال فقط عملیاتی که بیشترین وزن را دارد نگه داشته می‌شود. بدین ترتیب گراف نهایی که یک زیرگراف (Subgraph) بهینه از ابرگراف اولیه است استخراج می‌شود.

۴-۲. تکنیک کوچک‌سازی پیش‌رونده استفاده‌شده در NAS را توضیح دهید.

الگوریتم‌های NAS کلان (مانند MnasNet) معمولاً فضای جستجو را در سطح بلوک‌ها تحلیل می‌کنند و تخمین نامناسبی از سخت‌افزار دارند. با این حال، هنگامی که معماری کلی به دست آمد، برای بیشترین کاهش در زمان تأخیر (Latency) یا مصرف انرژی روی یک سخت‌افزار خاص، باید ابعاد کانال‌ها و فیلترهای هر لایه به صورت جزئی و دقیق تغییر و مورد بررسی قرار بگیرد.

روش‌های سنتی مانند هرس کردن (Pruning) معمولی اغلب بر اساس معیارهای ساده (مانند soft $L1$ وزن‌ها) کار می‌کنند که مستقیماً معیارهای سخت‌افزاری (مانند پهنای باند حافظه یا Latency واقعی) را بهینه‌سازی نمی‌کنند. NetAdapt این شکاف را با رویکردی هدایت‌شده با معیارهای سخت‌افزاری واقعی پر می‌کند.

الگوریتم آن:

فرآیند به صورت تکراری (Iterative) انجام می‌شود. در هر مرحله، هدف ما کاهش زمان تأخیر شبکه به اندازه یک گام مشخص (ΔR) با کمترین میزان افت دقت است.

گام ۱: تولید گزینه‌های کاندید برای هرس (Proposal Generation)

در شروع هر تکرار t، فرض کنید شبکه فعلی دارای L لایه کانولوشنی است. الگوریتم به نوبت برای هر لایه $l \in \{1,2,\ldots,L\}$ اقدام به تولید یک شبکه پیشنهادی (Proposal) می‌کند:

- برای لایه l، تعداد فیلترها (کانال‌های خروجی) را کاهش می‌دهیم تا زمانی که زمان تأخیر کل شبکه به اندازه دقیقاً ΔR کاهش یابد.
- این کار باعث تولید L شبکه کاندید می‌شود که در هر کدام، فقط یکی از لایه‌ها کوچک‌تر شده است اما همگی پاداش سخت‌افزاری یکسانی (کاهش تأخیر به میزان ΔR) به دست آورده‌اند.

به عبارتی: به جای اینکه شبکه را یک‌دفعه و کلی کوچک کنیم (که باعث افت شدید دقت می‌شود)، الگوریتم در هر مرحله فقط بخش خیلی کوچکی از شبکه را حذف یا کم‌حجم می‌کند.

گام ۲: آموزش کوتاه‌مدت (Short Fine-tuning)

برای اینکه ارزیابی دقیقی از توانایی هر شبکه کاندید داشته باشیم، هر یک از این L شبکه را برای تعداد Epoch های بسیار کم (مثلاً ۱ تا ۵ Epoch) روی داده‌های آموزش، Train کنیم تا وزن‌ها با ساختار جدید سازگار شوند و افت دقت اولیه جبران شود.

گام ۳: انتخاب بهترین کاندید (Selection)

از میان این L شبکه پیشنهاد شده، شبکه‌ای را انتخاب می‌کنیم که بیشترین دقت (یا کمترین افت دقت) را پس از آموزش کوتاه داشته باشد:

$$Net_t = arg\max_{l}(Accuracy\left(Net_{t-1,pruned_{at_l}}\right))$$

این شبکه به عنوان ساختار پایه برای تکرار بعدی (t+1) انتخاب می‌شود.

گام ٤: تکرار و آموزش نهایی (Iterate & Fine-tune)

این سه گام تا زمانی تکرار می‌شوند که زمان تأخیر کل شبکه به مقدار هدف (Target Latency) برسد. در نهایت، شبکه کوچک‌شده نهایی تحت یک فرآیند Fine-tuning طولانی‌مدت قرار می‌گیرد تا دقت آن بازیابی شده و به بیشینه خود برسد.