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)
- 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.

بخش نظری (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.

نرمال‌سازی بچ در شبکه‌های کاملاً متصل روی هر نورون به‌طور مستقل اعمال می‌شود، محاسبه‌ی میانگین و واریانس برای هر ویژگی در طول مینی‌بچ انجام می‌شود و به همگرایی سریع‌تر مدل و پایداری آموزش کمک می‌کند. در شبکه‌های پیچشی نرمال‌سازی بچ روی کانال‌های هر نقشه‌ی ویژگی انجام می‌شود و میانگین و واریانس در هر کانال و در تمامی پیکسل‌های آن محاسبه می‌شود. در مرحله‌ی آموزش، برای نرمال‌سازی بچ از میانگین و واریانس محاسبه‌شده مینی‌بچ فعلی استفاده می‌شود که به کاهش تغییرات داخلی توزیع داده‌ها کمک می‌کند و سرعت یادگیری را افزایش می‌دهد. برای نرمال‌سازی بچ در مرحله‌ی آزمایش، از میانگین و واریانس میانگین‌گیری‌شده در طول آموزش استفاده می‌شود که باعث جلوگیری از تغییرات ناگهانی در خروجی مدل می‌شود.

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؟

Covariate shift به معنای تغییر در توزیع ویژگی‌های ورودی بین داده‌های آموزش و آزمایش است، در حالی که توزیع خروجی نسبت به ورودی ثابت می‌ماند. این تغییر می‌تواند باعث شود که مدل در هنگام آموزش به الگوهای خاصی وابسته شود که در داده‌های آزمایشی وجود ندارند، که منجر به کاهش دقت و تعمیم‌پذیری مدل می‌شود. نرمال‌سازی بچ با تنظیم میانگین و واریانس ویژگی‌ها در طول آموزش، تغییرات توزیع داده را کاهش داده و به پایداری و همگرایی بهتر مدل کمک می‌کند اما در مرحله‌ی آزمایش، به‌دلیل نبود مینی‌بچ‌ها، نرمال‌سازی بچ از میانگین و واریانس متحرک که در طول آموزش ذخیره شده‌اند، استفاده می‌کند. این تفاوت می‌تواند باعث ایجاد عدم تطابق بین توزیع فعال‌سازی‌هایی که مدل در زمان آموزش می‌بیند و آنچه در زمان آزمایش مشاهده می‌شود، گردد. اگر میانگین و واریانس محاسبه‌شده در طول آموزش دچار تغییرات زیادی باشند، این اختلاف ممکن است موجب ناپایداری عملکرد مدل در داده‌های آزمایشی شود.

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؟

حذف بایاس (b) از لایه‌ی کانولوشن باعث ایجاد اختلال در کارکرد شبکه نمی‌شود، زیرا نرمال‌سازی بچ شامل یک shift است که اثر بایاس را جبران می‌کند. به همین دلیل، در صورت استفاده از BatchNorm افزودن بایاس به لایه کانولوشن غیرضروری است. اگر پس از آموزش شبکه، در مرحله‌ی تست تمام وزن‌های کانولوشن را در یک عدد ثابت مانند 10 ضرب کنیم، خروجی خام کانولوشن نیز به همان نسبت تغییر می‌کند. اما از آنجایی که BatchNorm در زمان تست از میانگین و واریانس ثابت‌شده (محاسبه‌شده در زمان آموزش) استفاده می‌کند، این تغییر در وزن‌ها باعث ناهماهنگی می‌شود. به‌طور خاص، مقدار x در فرمول نرمال‌سازی ناگهان مقیاس‌بندی شده، در حالی که μ و σ^2 بدون تغییر باقی مانده‌اند. این اختلاف موجب تغییر توزیع خروجی و معمولاً افت عملکرد شبکه می‌شود. برای حفظ عملکرد، باید علاوه بر ضرب وزن‌ها، پارامترهای γ و β و حتی μ و σ^2 را نیز متناسب تغییر داد. اما اگر تنها وزن‌ها را تغییر دهیم، شبکه دیگر با توزیع داده‌های آموزش‌دیده‌شده همخوانی نخواهد داشت. در صورتی که تمام ورودی‌های شبکه (مثلاً تصاویر) در یک مقدار ثابت ضرب یا تقسیم شوند، خروجی اولیه کانولوشن به همان نسبت تغییر می‌کند. اما در مرحله‌ی BatchNorm، به دلیل ثابت بودن میانگین و واریانس ذخیره‌شده، این تغییر مقیاس ممکن است باعث عدم تطابق با مقادیر آموزش‌دیده‌شده و اختلال در عملکرد شبکه شود. اگر این تغییر مقیاس کوچک باشد، تأثیر آن ممکن است ناچیز باشد، اما تغییرات شدید می‌تواند باعث اختلال در پردازش ویژگی‌ها و کاهش دقت مدل شود.

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.

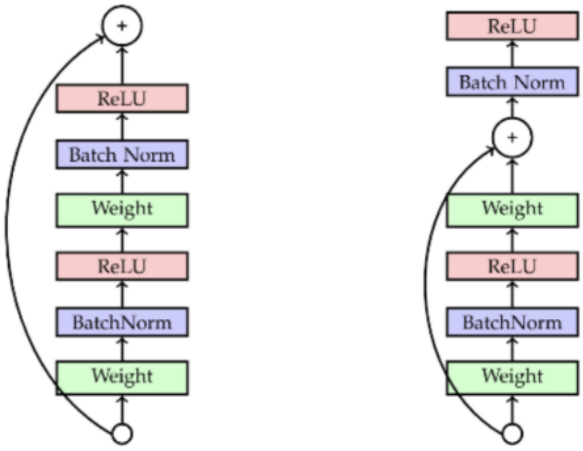
در نرمال‌سازی بچ، میانگین و واریانس از روی هر بچ (mini-batch) استخراج می‌شود که این مقادیر تنها تخمین‌هایی از پارامترهای واقعی توزیع داده‌ها هستند. به همین دلیل، هنگام محاسبه گرادیان‌ها، نویز ناشی از تغییرات این دو آماره در هر بچ به وجود می‌آید. این نویز به عنوان یک ویژگی ضمنی عمل می‌کند که باعث می‌شود مدل در مقابل بیش‌برازش مقاوم‌تر شود، زیرا هر بار که یک بچ متفاوت استفاده می‌شود، تغییراتی در گرادیان بوجود می‌آید و این امر به نوعی عملکرد مدل را به چندین شبکه مستقل که هر کدام داده‌های یک بچ را یاد می‌گیرند، تقسیم می‌کند. از سوی دیگر، dropout به صورت صریح با حذف تصادفی تعدادی از نورون‌ها در لایه‌های شبکه، به عنوان یک روش منظم‌سازی (regularization) عمل می‌کند. در dropout شبکه مجبور می‌شود وابستگی‌های زیاد به یک سری نورون خاص را کاهش دهد و درنتیجه مانع زیاد شدن وزن یال‌های متصل به آن‌ها شود (چون که در هر تکرار، بخش‌هایی از شبکه غیرفعال می‌شوند). این مکانیسم به مدل اجازه می‌دهد تا ویژگی‌های توزیعی بیشتری یاد بگیرد و از بیش‌برازش جلوگیری

می‌کند. تعبیر دیگری که می‌توان داشت این است که dropout با ایجاد یک ensemble از مدل‌های کوچک، عملکرد نهایی شبکه را بهبود می‌بخشد.

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

استفاده پشت سر هم از BatchNorm و dropout در بسیاری از معماری‌های مدرن مورد بحث قرار گرفته است. از یک سو نرمال‌سازی دسته‌ای با نرمال‌سازی آماره‌های میانگین و واریانس هر بچ، به کاهش تغییرات داخلی (internal covariate shift) و بهبود همگرایی شبکه کمک می‌کند. از سوی دیگر، dropout به‌طور صریح با حذف تصادفی نورون‌ها، به کاهش وابستگی‌های بیش از حد بین آن‌ها و جلوگیری از بیش‌برازش کمک می‌کند. اما استفاده همزمان از این دو روش ممکن است چالش‌هایی ایجاد کند: اگر dropout پس از BatchNorm اعمال شود، حذف تصادفی برخی از نورون‌ها می‌تواند آماره‌های محاسبه‌شده در هنگام آموزش را ناپایدار کند. به عبارت دیگر، dropout نویز اضافی‌ای به توزیع‌های خروجی وارد می‌کند که می‌تواند تأثیر نرمال‌سازی مثبت BatchNorm را تضعیف کند. از آنجا که BatchNorm به خودی خود تأثیر منظم‌کننده‌ای دارد (به واسطه کاهش نوسانات ناشی از تغییرات هر بچ)، در بسیاری از موارد استفاده از dropout پس از آن به عنوان منظم‌کننده ممکن است لازم نباشد و حتی به ضرر عملکرد شبکه تمام شود.

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:



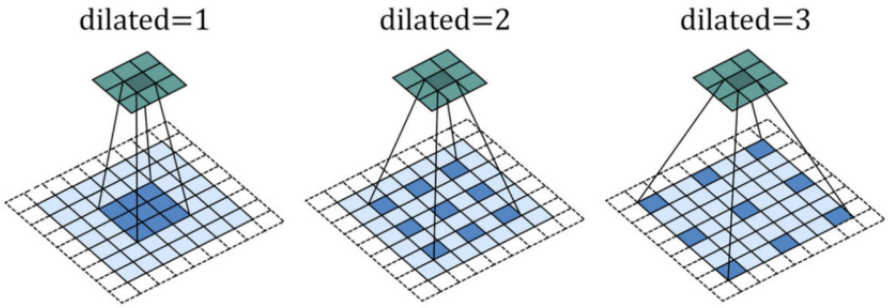
با در نظر گرفتن تأثیر نرمال‌سازی دسته‌ای بر انتشار گرادیان، پایداری بهینه‌سازی، سازگاری بین آموزش و تست، تغییر واریانس، یادگیری نمایش‌های عمیق، تحلیل کنید که چگونه مکان BN می‌تواند دینامیک آموزش شبکه را تحت تأثیر قرار دهد. (توجه کنید که منظور از Weight می تواند لایه تماما متصل و یا پیچشی باشد.)

در شبکه‌های باقی‌مانده، اتصالات میان‌بُر (Skip Connections) نقش اساسی در بهبود جریان گرادیان و جلوگیری از ناپدید شدن آن ایفا می‌کنند. این اتصالات، خروجی لایه‌های قبلی را مستقیماً به لایه‌های جلوتر منتقل می‌کنند تا در یادگیری لایه‌های بسیار عمیق اختلال کمتری ایجاد شود و مدل بتواند پارامترهای بیشتری را بدون افت عملکرد آموزش دهد. در ساختار سمت راست، ابتدا اطلاعات مسیر باقی‌مانده و مسیر اصلی با هم ترکیب شده و سپس نرمال‌سازی انجام می‌شود. این طراحی باعث می‌شود که خروجی مسیر اصلی بدون نرمال‌سازی با ورودی جمع شود که انتشار گرادیان را با مشکل مواجه می‌کند اما به دلیل نرمال‌سازی بعد از جمع شدن دو مسیر می‌تواند نرخ یادگیری بیشتری را بدون مشکلاتی نظیر انفجار گرادیان بپذیرد اما به هر حال ممکن است پایداری گرادیان کمتر شود و یادگیری در لایه‌های عمیق‌تر دشوارتر گردد. در ساختار دوم، قبل از ترکیب مسیر باقی‌مانده و مسیر اصلی، نرمال‌سازی انجام شده که باعث ثبات بیشتر در توزیع فعال‌سازی‌ها می‌شود و به همین دلیل، این روش انتشار گرادیان را یکنواخت‌تر کرده و پایداری آموزش را

افزایش می‌دهد. اگرچه مشکل این روش این است که خروجی‌ای که با ورودی جمع می‌شود همواره ۰ یا مثبت خواهد بود (به دلیل وجود فعال‌ساز Relu) اما با هم نسبت به روش سمت راست به دلیل نرمال‌سازی خروجی پیش از جمع شدن با ورودی، انتشار گرادیان بهتر و واریانس کمتری دارد. این امر همچنین سازگاری بین آموزش و تست را افزایش می‌دهد، زیرا مقدار مسیر باقی‌مانده در طول آموزش دچار نوسان نمی‌شود. در واقع از آنجا که نرمال‌سازی قبل از جمع انجام شده، مقدار مسیر باقی‌مانده بدون تغییر در ترکیب تأثیر می‌گذارد. به طور کلی بهتر از فعال‌ساز Relu بعد از جمع و نرمال‌سازی قبل از آن انجام شود.

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.



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.

در حالت پایه با اندازه فیلتر k_1 و نرخ اتساع d_1 گستره دید شبکه در هر بعد به اندازه $R_1 = (k_1-1)d_1+1$ خواهد بود. اگر k_1 برابر یک باشد این فرمول برابر یک می‌شود که به معنای گستره دید یک واحدی است. یعنی به ازای هر پیکسل در ورودی، یک واحد دید خواهیم داشت. حال اگر در لایه بعدی فیلتری به سایز k_2 داشته باشم و نرخ اتساع متناظر آن، گستره دید ما به اندازه $(k_2-1)d_2s_1$ خواهد بود و در مجموع (با احتساب لایه قبل) داریم: $R_2 = R_1+(k_2-1)d_2s_1$ با جایگذاری R_1 در این فرمول خواهیم داشت: $R_2 = 1 + (k_1 - 1)d_1 + (k_2 - 1)d_2s_1$. با ادامه این روند فرمول کلی به صورت زیر در خواهد آمد:

$$R_L = 1 + \sum_{\ell=1}^{L}((k_\ell - 1)d_\ell \prod_{i=1}^{\ell-1} s_i)$$

این عدد مربوط یک بعد است و مربع آن گستره دید در کل را به ما می‌دهد. حال اگر گستره دید نسبی را بخواهیم به فرمول زیر می‌رسیم:

$$R_{relative} = \frac{R_L \cdot R_L}{M.N}$$

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؟

می‌خواهیم تضمین کنیم که $R^{(L)}_{\text{relative}} > T$ در حالی که هزینه‌های محاسباتی (FLOPs) را به حداقل برسانیم. هزینه محاسباتی برای یک لایه CNN متناسب با این مقدار است:

$$\text{FLOPs}_\ell \propto \frac{k_\ell^2}{s_\ell^2}$$

برای به حداقل رساندن FLOPs در حالی که محدودیت گستره دید نسبی حفظ شود: تعادل بهینه به این صورت است:

$$\frac{\partial R^{(L)}_{\text{relative}}}{\partial d_\ell} / \frac{\partial \text{FLOPs}}{\partial d_\ell} = \frac{\partial R^{(L)}_{\text{relative}}}{\partial s_\ell} / \frac{\partial \text{FLOPs}}{\partial s_\ell}$$

این منجر به معادله زیر می‌شود:

$$(k_\ell - 1) \times \prod_{i=1}^{\ell-1} s_i = \frac{2k_\ell^2}{s_\ell^3}$$

بله، شرایط بهینه به L وابسته است زیرا ۱. با تعداد لایه‌های بیشتر، می‌توانیم رشد گستره دید را بین لایه‌های بیشتری توزیع کنیم، که این امر امکان استفاده از کرنل‌های کوچکتر یا نرخ‌های اتساع کمتر در هر لایه را فراهم می‌کند. ۲. گستره دید کلی تابعی از ترکیب تمام لایه‌ها است. ۳. کارایی محاسباتی بستگی به سرعت کاهش اندازه نقشه‌های ویژگی در لایه‌های مختلف دارد. ۴. با تعداد لایه‌های کمتر، نیاز به اتساع شدیدتر یا کرنل‌های بزرگتر برای دستیابی به همان گستره دید داریم. در عمل، شبکه‌های عمیق‌تر می‌توانند گستره‌های دید بزرگتری را با کارایی بیشتری از طریق ترکیب چندین عملیات کوچکتر به دست آورند، به جای استفاده از کرنل‌های بسیار بزرگ یا نرخ‌های اتساع بسیار زیاد در یک شبکه کم‌عمق.

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 تشخیص اشیاء، یکی از مشکلات رایج در مراحل استخراج ویژگی‌ها، عدم دقت در هم‌ترازی ناحیه مورد نظر (ROI Alignment) است. در روش‌های قدیمی‌تر مانند ROI Pooling مختصات ناحیه‌ها به مقادیر صحیح گرد می‌شد که این کار باعث ناهماهنگی بین ناحیه واقعی و ویژگی‌های استخراج‌شده می‌شد. اما در روش ROI Align این مشکل با حذف عملیات گردکردن و استفاده از درون‌یابی خطی (Linear Interpolation) برای محاسبه دقیق مقادیر ویژگی‌ها در مختصات اعشاری حل شده است. در این روش، مختصات ناحیه به بخش‌های مساوی تقسیم می‌شود و برای هر بخش، به‌جای استفاده از نزدیک‌ترین پیکسل، مقدار ویژگی‌ها بر اساس وزن‌دهی فاصله‌ای از چهار پیکسل اطراف محاسبه می‌گردد. این کار باعث هم‌ترازی دقیق‌تر و بهبود عملکرد مدل در وظایفی مانند تشخیص دقیق اشیاء می‌شود.

روابط مربوط به درون‌یابی خطی:

$$
\begin{aligned}
x_1 &= \lfloor x \rfloor, \quad x_2 = \lceil x \rceil \\
y_1 &= \lfloor y \rfloor, \quad y_2 = \lceil y \rceil \\
dx &= x - x_1 \\
dy &= y - y_1
\end{aligned}
$$

$$
\begin{aligned}
f(x,y) = (1-dx)(1-dy) \cdot f(x_1,y_1) + \\
dx(1-dy) \cdot f(x_2,y_1) + \\
(1-dx)dy \cdot f(x_1,y_2) + \\
dx \cdot dy \cdot f(x_2,y_2)
\end{aligned}
$$

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.

یافتن ضریب مقیاس بین تصویر اصلی و نقشه ویژگی

$$Scaling\ Coefficient = \frac{32}{10} = 3.2$$

نگاشت مختصات از تصویر اصلی به نقشه ویژگی

$$
\begin{aligned}
x_{\text{feature}} &= \frac{4}{3.2} = 1.25 \\
y_{\text{feature}} &= \frac{8}{3.2} = 2.5
\end{aligned}
$$

اعمال درون‌یابی دوخطی برای محاسبه مقدار

از آنجا که مختصات اعشاری $(1.25, 2.5)$ داریم، باید طبق فرمول نوشته شده در قسمت قبل از چهار مختصات صحیح

نزدیک در نقشه ویژگی درون‌یابی کنیم:

$$
\begin{aligned}
x_1 &= \lfloor 1.25 \rfloor = 1, \quad x_2 = \lceil 1.25 \rceil = 2 \\
y_1 &= \lfloor 2.5 \rfloor = 2, \quad y_2 = \lceil 2.5 \rceil = 3 \\
dx &= 1.25 - 1 = 0.25 \\
dy &= 2.5 - 2 = 0.5
\end{aligned}
$$

$f(1,2) =$ مقدار پیکسل در موقعیت (۱،۲) نقشه ویژگی
$f(2,2) =$ مقدار پیکسل در موقعیت (۲،۲) نقشه ویژگی
$f(1,3) =$ مقدار پیکسل در موقعیت (۱،۳) نقشه ویژگی
$f(2,3) =$ مقدار پیکسل در موقعیت (۲،۳) نقشه ویژگی

$$
\begin{aligned}
f(1.25, 2.5) &= (1 - 0.25)(1 - 0.5) \cdot f(1,2) + 0.25(1 - 0.5) \cdot f(2,2) \\
&\quad + (1 - 0.25) \cdot 0.5 \cdot f(1,3) + 0.25 \cdot 0.5 \cdot f(2,3) \\
&= 0.375 f(1,2) + 0.125 f(2,2) + 0.375 f(1,3) + 0.125 f(2,3)
\end{aligned}
$$

بنابراین، مقدار پیکسل در مختصات $(4, 8)$ در تصویر اصلی با میانگین وزنی چهار نقطه نزدیک در نقشه ویژگی با وزن‌های محاسبه شده در بالا متناظر است.

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.

$$
w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \quad \mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
$$

$$
Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & 0 & w_3 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} 0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ 0 \end{bmatrix} = \begin{bmatrix} w_2 x_1 + w_3 x_2 \\ w_1 x_1 + w_2 x_2 + w_3 x_3 \\ w_1 x_2 + w_2 x_3 + w_3 x_4 \\ w_1 x_3 + w_2 x_4 \end{bmatrix}
$$

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.

$$\frac{\partial L}{\partial \mathbf{Y}} = \begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \frac{\partial L}{\partial y_2} \\ \frac{\partial L}{\partial y_3} \\ \frac{\partial L}{\partial y_4} \end{bmatrix}, \frac{\partial L}{\partial \mathbf{x}} = \frac{\partial L}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \\ \frac{\partial L}{\partial x_3} \\ \frac{\partial L}{\partial x_4} \end{bmatrix}$$

$$\begin{aligned}
\frac{\partial L}{\partial x_1} &= \frac{\partial L}{\partial y_1} w_2 + \frac{\partial L}{\partial y_2} w_1 \\
\frac{\partial L}{\partial x_2} &= \frac{\partial L}{\partial y_1} w_3 + \frac{\partial L}{\partial y_2} w_2 + \frac{\partial L}{\partial y_3} w_1 \\
\frac{\partial L}{\partial x_3} &= \frac{\partial L}{\partial y_2} w_3 + \frac{\partial L}{\partial y_3} w_2 + \frac{\partial L}{\partial y_4} w_1 \\
\frac{\partial L}{\partial x_4} &= \frac{\partial L}{\partial y_3} w_3 + \frac{\partial L}{\partial y_4} w_2
\end{aligned}$$

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؟

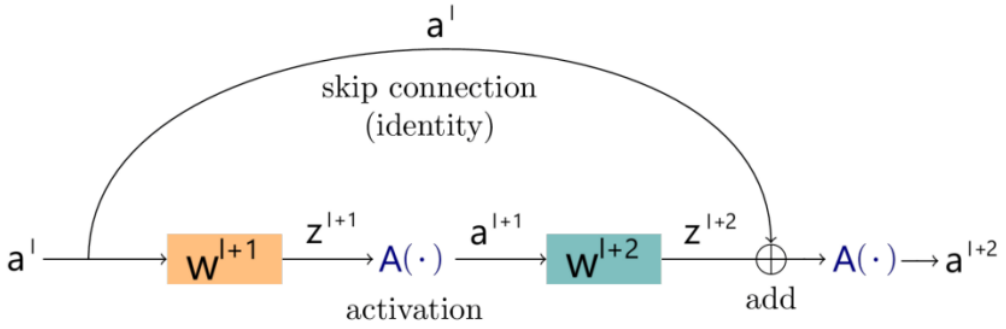
$$\begin{bmatrix} w_2 \\ w_3 \\ 0 \\ 0 \end{bmatrix} \cdot \frac{\partial L}{\partial y_1} + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ 0 \end{bmatrix} \frac{\partial L}{\partial y_2} + \begin{bmatrix} 0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} \frac{\partial L}{\partial y_3} + \begin{bmatrix} 0 \\ 0 \\ w_1 \\ w_2 \end{bmatrix} \frac{\partial L}{\partial y_4}$$

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؟

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} = \begin{bmatrix} w_1 & 0 & 0 & 0 \\ w_2 & w_1 & 0 & 0 \\ w_3 & w_2 & w_1 & 0 \\ 0 & w_3 & w_2 & w_3 \\ 0 & 0 & w_3 & w_2 \\ 0 & 0 & 0 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} w_1 x_1 \\ w_2 x_1 + w_1 x_2 \\ w_3 x_1 + w_2 x_2 + w_1 x_3 \\ w_3 x_2 + w_2 x_3 + w_1 x_4 \\ w_3 x_3 + w_2 x_4 \\ w_3 x_4 \end{bmatrix}$$

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.

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

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.

ابتدا حاصل skip connection را با متغیر k^{l+2} بازنویسی می‌کنیم: $k^{l+2} = z^{l+2} + a^z \quad \Rightarrow \quad a^{l+2} = A(k^{l+2})$

$$\frac{\partial a^{l+2}}{\partial a^l} = \frac{\partial a^{l+2}}{\partial k^{l+2}} \cdot \frac{\partial k^{l+2}}{\partial a^l} = A'(z^{l+2} + a^l) \cdot [1 + \frac{\partial z^{l+2}}{\partial a^l}] = A'(z^{l+2} + a^l) \cdot [1 + w^{l+2} \cdot A'(z^{l+1}) \cdot w^{l+1}]$$

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 $W^i < 1 - \epsilon$)

اتصالات میان‌بر (Skip Connections) با ایجاد مسیری مستقیم برای عبور گرادیان در فرایند پس‌انتشار، مانع ناپدیدشدن گرادیان می‌شوند. در شبکه‌های عمیق، ضرب مکرر مقادیر کوچک (مانند وزن‌ها یا مشتقات توابع فعال‌سازی) می‌تواند باعث شود که گرادیان کوچک و یادگیری مختل شود. افزودن یک اتصال میان‌بر که اغلب شامل یک نگاشت همانی است، باعث می‌شود که گرادیان بتواند بدون عبور از چندین تبدیل غیرخطی، مقدار خود را حفظ کند و در نتیجه به‌روزرسانی وزن‌ها پایدار باقی بماند. این را می‌توان در عبارتی که در قسمت قبل بدست آمد نیز مشاهده کرد، زیرا در حاصل گرادیان یک +۱ اضافه شده که مانع صفر شدن گرادیان حتی در صورت صفر شدن بخش باقی مانده می‌شود.

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).

فرض کنید که یک ورودی با ابعاد $D_f \times D_f \times M$ داریم که D_f اندازه تصویر و M تعداد کانال‌های ورودی است. فیلترهای مورد استفاده دارای ابعاد $D_k \times D_k \times M$ هستند. اگر تعداد این فیلترها N باشد، خروجی دارای ابعاد $D_p \times D_p \times N$ خواهد بود. میزان عملیات محاسباتی در کانولوشن عادی به صورت زیر محاسبه می‌شود:

$$\text{محاسبات کل} = N \times D_p^2 \times D_k^2 \times M \quad (۱)$$

در ورش معرفی‌شده در موبایل‌نت، کانولوشن عمقی قابل تفکیک شامل دو مرحله است:

- **کانولوشن عمقی**:در این مرحله، به جای اعمال یک فیلتر سه‌بعدی روی تمام کانال‌های ورودی، هر کانال ورودی به‌طور جداگانه با یک فیلتر مستقل پردازش می‌شود. این کار باعث کاهش چشمگیر محاسبات می‌شود. تعداد فیلترها برابر M و اندازه آنها $D_k \times D_k \times 1$ است. میزان محاسبات این مرحله برابر است با:

$$\text{محاسبات کانولوشن عمقی} = M \times D_p^2 \times D_k^2 \quad (۲)$$

- **کانولوشن نقطه‌ای**: در این مرحله، یک لایه کانولوشنی ۱*۱ اعمال می‌شود که وظیفه‌ی ترکیب اطلاعات بین کانال‌ها را بر عهده دارد. برای این منظور از فیلترهای $1 \times 1 \times M$ برای ترکیب ویژگی‌های استخراج‌شده از مرحله قبل استفاده می‌شود. میزان محاسبات این مرحله برابر است با:

$$\text{محاسبات کانولوشن نقطه‌ای} = M \times D_p^2 \times N \quad (۳)$$

بنابراین، کل میزان عملیات محاسباتی در کانولوشن تفکیکی عمقی برابر است با:

$$\text{محاسبات کل} = M \times D_p^2 \times (D_k^2 + N) \quad (۴)$$

واضحا نسبت محاسباتی این دو روش هم به صورت زیر است:

$$\frac{M \times D_p^2 \times (D_k^2 + N)}{N \times D_p^2 \times D_k^2 \times M} = \frac{1}{N} + \frac{1}{D_k^2} \quad (۵)$$

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.

فرض کنید یک فیلتر کانولوشن به صورت یک ماتریس $W \in \mathbb{R}^{k \times k}$ داشته باشیم (یعنی دارای ابعاد $k \times k$ است). با استفاده از تجزیه مقادیر تکین (SVD) می‌توانیم ماتریس W را به سه ماتریس U، Σ و V^T تجزیه کنیم:

$$W = U \Sigma V^T = \sum_{i=1}^{r} \sigma_i \, u_i \, v_i^T,$$

که در آن:

- r رتبه ماتریس W است.
- σ_i مقادیر تکین هستند.
- u_i و v_i بردارهای تکین می‌باشند.

هر عبارت $\sigma_i u_i v_i^T$ یک تنسور رتبه یک محسوب می‌شود. به عبارت دیگر، فیلتر W را می‌توان به صورت مجموع r کانولوشن رتبه یک بیان کرد، به‌طوری که هر کدام با فیلتر $u_i v_i^T$ اعمال می‌شود. ارتباط کانولوشن عمقی با تجزیه مقادیر تکین را می‌توان این طور بیان کرد که اگر فیلتر اصلی W دارای رتبه $r = 1$ باشد، تجزیه SVD آن معادل اعمال یک کانولوشن عمقی به دنبال یک کانولوشن 1×1 خواهد بود. در حالت کلی هم تجزیه W به r تنسور رتبه یک معادل استفاده از r فیلتر عمقی و ترکیب آن‌ها با r کانولوشن 1×1 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؟

در کانولوشن استاندارد، هر فیلتر $K \times K$ بر روی هر کانال ورودی اعمال شده و سپس تمامی کانال‌های ورودی ترکیب می‌شوند تا یک خروجی تولید شود. تعداد عملیات ممیز شناور (FLOPs) برای یک لایه کانولوشنی استاندارد به صورت زیر است:

$$\text{FLOPs}_{\text{Standard}} = H \times W \times C_{\text{in}} \times C_{\text{out}} \times K^2$$

همانطور که گفتیم در کانولوشن عمقی قابل تفکیک، دو مرحله داریم:

$$\text{FLOPs}_{\text{Depthwise}} = H \times W \times C_{\text{in}} \times K^2$$

$$\text{FLOPs}_{\text{Pointwise}} = H \times W \times C_{\text{in}} \times C_{\text{out}}$$

بنابراین، کل هزینه محاسباتی در کانولوشن عمقی قابل تفکیک برابر است با:

$$\text{FLOPs}_{\text{Separable Depthwise}} = H \times W \times C_{\text{in}} \times K^2 + H \times W \times C_{\text{in}} \times C_{\text{out}}$$

نسبت کاهش هزینه محاسباتی در کانولوشن عمقی قابل تفکیک نسبت به کانولوشن استاندارد به صورت زیر است:

$$\text{Ratio Reduction} = \frac{\text{FLOPs}_{\text{Separable Depthwise}}}{\text{FLOPs}_{\text{Standard}}} = \frac{H \times W \times C_{\text{in}} \times K^2 + H \times W \times C_{\text{in}} \times C_{\text{out}}}{H \times W \times C_{\text{in}} \times C_{\text{out}} \times K^2}$$

با ساده‌سازی خواهیم داشت:

$$\text{Ratio Reduction} = \frac{1}{C_{\text{out}}} + \frac{1}{K^2}$$

کانولوشنهای عمقی قابل تفکیک با کاهش چشمگیر محاسبات، ظرفیت نمایشی مدل را نیز کاهش میدهند، زیرا پارامترهای کمتری برای یادگیری ویژگی‌های پیچیده وجود دارد. این کاهش پارامترها ممکن است منجر به افت عملکرد در مسائل پیچیده شود. بنابراین، یک تضاد (Trade-off) بین کارایی محاسباتی و ظرفیت نمایشی وجود دارد. برای جبران این مسئله، معمولاً از روشهایی مانند افزایش عمق شبکه یا ترکیب با لایه‌های دیگر استفاده میشود.

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.

بلوک‌های باقیمانده معمولی ResNet:

- ویژگی‌های ورودی ابتدا از طریق یک سری لایه کانولوشنی عبور می‌کنند
- سپس ورودی به خروجی این کانولوشن‌ها اضافه می‌شود (ایجاد "اتصال میان‌بر")
- الگوی معمول: ورودی ← کانولوشن ← کانولوشن ← افزودن ورودی ← خروجی
- تعداد کانال‌ها معمولاً در لایه‌های کانولوشنی ثابت می‌ماند یا افزایش می‌یابد

بلوک‌های باقیمانده معکوس (استفاده شده در MobileNetV2):

- ویژگی‌های ورودی ابتدا از یک لایه انبساط عبور می‌کنند که تعداد کانال‌ها را افزایش می‌دهد
- سپس یک کانولوشن عمقی تفکیک پذیر (depthwise) اعمال می‌شود
- در نهایت، یک لایه پروجکشن (کانولوشن‌های یک در یک) تعداد کانال‌ها را کاهش می‌دهد تا با ورودی مطابقت کند
- اتصال میان‌بر، ورودی را به خروجی نهایی (پس از پروجکشن) متصل می‌کند
- الگو: ورودی ← افزایش کانال‌ها ← کانولوشن عمقی ← کاهش کانال‌ها ← افزودن ورودی ← خروجی

تفاوت‌های کلیدی:

- بلوک‌های معکوس ابتدا فضای ویژگی را گسترش و سپس فشرده می‌کنند، در حالی که بلوک‌های معمولی ResNet ابعاد کانال را حفظ یا افزایش می‌دهند
- بلوک‌های معکوس از کانولوشن‌های عمقی تفکیک‌پذیر استفاده می‌کنند که از نظر محاسباتی کارآمدتر هستند
- اتصال باقیمانده در بلوک‌های معکوس از یک ساختار "گلوگاهی" عبور می‌کند، در حالی که در ResNet معمولی از ساختاری عبور می‌کند که ابعاد را حفظ یا گسترش می‌دهد

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 ادعا درست است. زیرا کانولوشن‌های عمقی تفکیک‌پذیر روی هر کانال به صورت جداگانه عمل می‌کنند و اطلاعات مکانی را استخراج می‌کنند. یعنی با استفاده از فیلترهای مکانی $k \times k$، ویژگی‌های محلی مانند لبه‌ها، بافت‌ها و الگوهای هندسی را شناسایی می‌کنند و ارتباطات بین پیکسل‌های مجاور را حفظ می‌کنند. این روش اصلا اطلاعات کانالی را ترکیب نمی‌کند (چون هر کانال c_i به طور مستقل پردازش می‌شود). گلوگاه‌های خطی اما از طریق لایه‌های کاملا متصل خطی یا کانولوشن 1×1 عمل می‌کنند روی بُعد کانال تمرکز دارند و ارتباطات بین کانال‌ها را مدل‌سازی می‌کنند.

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؟

افزایش ضریب انبساط t باعث کاهش خطر ناپدید شدن گرادیان‌ها می‌شود. این ادعا را می‌توان به شکل زیر اثبات کرد: اگر یک شبکه عصبی عمیق با توابع فعال‌سازی غیرخطی را در نظر بگیریم، مشتقات این توابع معمولاً بین 0 و 1 قرار دارند. هنگام انتشار پسرو گرادیان‌ها، این مشتقات در هم ضرب می‌شوند و طبق قاعده زنجیره‌ای، هر چه تعداد لایه‌ها بیشتر باشد، مقادیر کوچک‌تر بیشتر ضرب می‌شوند. فرض کنید $\frac{\partial L}{\partial x_i}$ گرادیان تابع خطا نسبت به ورودی x_i باشد و $\phi'(x)$ مشتق تابع فعال‌سازی باشد:

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y} \cdot \phi'(x_i) \cdot w_i$$

اگر ضریب انبساط t را به فضای ویژگی‌ها اضافه کنیم:

$$\frac{\partial L}{\partial x_i} = \frac{\partial L}{\partial y} \cdot \phi'(t \cdot x_i) \cdot w_i \cdot t$$

با افزایش t، ضریب t در انتهای معادله، اثر مقادیر کوچک $\phi'(x)$ را جبران می‌کند و از ناپدید شدن گرادیان‌ها جلوگیری می‌کند. افزایش t منجر به افزایش هزینه‌های محاسباتی می‌شود، زیرا:

- افزایش t به معنای افزایش ابعاد فضای ویژگی میانی است
- تعداد عملیات‌های محاسباتی با نرخ $\mathcal{O}(t)$ افزایش می‌یابد
- مصرف حافظه برای ذخیره‌سازی ویژگی‌های میانی افزایش می‌یابد

پیچیدگی محاسباتی یک لایه با ضریب انبساط t را می‌توان به صورت زیر محاسبه کرد:

$$C(t) = C_{\text{base}} \cdot t$$

که در آن C_{base} هزینه پایه محاسبات بدون انبساط است. برای انتخاب مقدار بهینه t که تعادل بین جریان گرادیان و کارایی محاسباتی ایجاد می‌کند، می‌توان معیارهای زیر را در نظر گرفت:

۱. **پیچیدگی مدل**: برای مدل‌های عمیق‌تر، t بزرگتر برای مقابله با ناپدید شدن گرادیان‌ها مناسب‌تر است

۲. **محدودیت‌های محاسباتی**: با توجه به منابع محاسباتی در دسترس (حافظه و قدرت پردازش)

۳. **اندازه مجموعه داده**: برای داده‌های بزرگتر، می‌توان t کوچکتر انتخاب کرد زیرا داده‌های بیشتر به تعمیم‌پذیری بهتر کمک می‌کنند

۴. **آزمایش تجربی**: تست مقادیر مختلف t روی مجموعه اعتبارسنجی و مقایسه دقت و سرعت همگرایی

یک رویکرد علمی برای انتخاب مقدار بهینه t این است که رابطه زیر را بهینه کنیم:

$$\text{Score}(t) = \alpha \cdot \text{Accuracy}(t) - \beta \cdot \text{ComputationalCost}(t)$$

که در آن α و β وزن‌های اهمیت دقت و هزینه محاسباتی هستند. این رابطه را می‌توان به صورت تابع پارامتری زیر نیز نمایش داد:

$$t_{\text{opt}} = \underset{t}{\text{argmax}} \ (\alpha \cdot \text{Accuracy}(t) - \beta \cdot t)$$

با مشتق‌گیری از تابع فوق نسبت به t و قرار دادن آن مساوی صفر، می‌توان نقطه بهینه را یافت:

$$\alpha \cdot \frac{d\text{Accuracy}(t)}{dt} - \beta = 0$$

$$\frac{d\text{Accuracy}(t)}{dt} = \frac{\beta}{\alpha}$$

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.

فرض کنید نقشه ویژگی ورودی با ابعاد $C \times H \times W$ به عنوان $\mathbf{X}$ داده شده است. اطلاعات Global هر کانال با **Pooling Average Global** محاسبه می‌شود:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j)$$

که در آن:

- z_c: مقدار فشرده‌شده برای کانال c
- $x_c(i,j)$: مقدار ویژگی در موقعیت مکانی (i,j) از کانال c

خروجی این مرحله بردار $\mathbf{z} \in \mathbb{R}^C$ است. بردار $\mathbf{z}$ به یک شبکه عصبی با دو لایه کاملاً متصل (FC) داده می‌شود:

$$\mathbf{s} = \sigma\left(\mathbf{W}_2 \cdot \delta\left(\mathbf{W}_1 \cdot \mathbf{z}\right)\right)$$

که در آن:

- $\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ و $\mathbf{W}_2 \in \mathbb{R}^{C \times \frac{C}{r}}$: ماتریس‌های وزن
- δ: تابع فعال‌سازی ReLU
- σ: تابع فعال‌سازی Sigmoid
- $\mathbf{s} \in [0,1]^C$: وزن‌های اهمیت کانال‌ها

وزن‌های $\mathbf{s}$ به نقشه ویژگی ورودی اعمال می‌شوند:

$$\hat{y}_c(i,j) = s_c \cdot x_c(i,j)$$

خروجی نهایی $\hat{\mathbf{Y}}$ یک تنسور با ابعاد $C \times H \times W$ است. این مکانیزم با رابطه زیر به‌طور خلاصه بیان می‌شود:

$$\hat{\mathbf{Y}} = \mathbf{s} \odot \mathbf{X}$$

که در آن $\odot$ ضرب elementwise (مولفه‌به‌مولفه) 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.

برای اثبات بهبود ظرفیت نمایشی، از معیار اطلاعات متقابل (Mutual Information) استفاده می‌کنیم. اطلاعات متقابل بین ورودی $\mathbf{X}$ و خروجی $\tilde{\mathbf{X}}$ را می‌توان به صورت زیر نوشت:

$$I(\mathbf{X}; \tilde{\mathbf{X}}) = H(\tilde{\mathbf{X}}) - H(\tilde{\mathbf{X}}|\mathbf{X})$$

که در آن H تابع آنتروپی است. با توجه به این که $\tilde{\mathbf{X}}$ تابعی قطعی از $\mathbf{X}$ و پارامترهای مدل است، داریم:

$$H(\tilde{\mathbf{X}}|\mathbf{X}) = 0$$

بنابراین:

$$I(\mathbf{X}; \tilde{\mathbf{X}}) = H(\tilde{\mathbf{X}})$$

حال برای مقایسه ظرفیت نمایشی، دو حالت را در نظر می‌گیریم:

- حالت ۱: بدون بلوک SE که در آن $\tilde{\mathbf{X}} = \mathbf{X}$
- حالت ۲: با بلوک SE که در آن $\tilde{\mathbf{X}} = \mathbf{s} \odot \mathbf{X}$

برای بررسی تفاوت ظرفیت نمایشی، می‌توان نشان داد:

$$I(\mathbf{X}; \tilde{\mathbf{X}}_{SE}) - I(\mathbf{X}; \mathbf{X}) = H(\mathbf{s} \odot \mathbf{X}) - H(\mathbf{X})$$

با توجه به ماهیت تطبیقی بردار $\mathbf{s}$ که وابسته به محتوای کانال‌هاست، می‌توان نشان داد که این مقیاس‌گذاری باعث می‌شود که:

$$H(\mathbf{s} \odot \mathbf{X}) \geq H(\mathbf{X})$$

زیرا مکانیزم تطبیقی SE توزیع ویژگی‌ها را به سمت توزیعی با آنتروپی بالاتر سوق می‌دهد، که منجر به استفاده مؤثرتر از ظرفیت کانال‌ها می‌شود.
ضریب کاهش r در لایه میانی بلوک SE، به صورت مستقیم بر تضاد بین هزینه‌های محاسباتی و عملکرد تأثیر می‌گذارد:

$$\text{تعداد پارامترها} = C \cdot \frac{C}{r} + \frac{C}{r} \cdot C = \frac{2C^2}{r}$$

۱. **مقادیر کوچک r:**

- **مزایا:** مدل‌سازی قوی‌تر ارتباطات بین کانال‌ها، افزایش ظرفیت یادگیری
- **معایب:** افزایش تعداد پارامترها، افزایش زمان محاسبات و مصرف حافظه

۲. **مقادیر بزرگ r:**

- **مزایا:** کاهش تعداد پارامترها، کاهش هزینه‌های محاسباتی
- **معایب:** کاهش توانایی مدل‌سازی ارتباطات پیچیده بین کانال‌ها

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.

محاسبه‌ی FLOPs برای بلوک Squeeze-and-Excitation

$$\begin{aligned}
\text{Global Average Pooling:} \quad & FLOPs = H \times W \times C \\
\text{FC۱:} \quad & FLOPs = C \times \frac{C}{r} \\
\text{FC۲:} \quad & FLOPs = \frac{C}{r} \times C \\
\text{Channel-wise Multiplication:} \quad & FLOPs = H \times W \times C \\
\text{Total:} \quad & \boxed{2HWC + \frac{2C^2}{r} = 128HW + 512}
\end{aligned}$$

از محاسبات مربوط به لایه‌های فعال‌ساز به دلیل تاثیرگذاری ناچیز صرف نظر کردم.

محاسبه‌ی FLOPs برای کانولوشن عمقی قابل تفکیک

۱. کانولوشن عمقی (Depthwise) با فیلتر 3×3:

$$FLOPs_{\text{depthwise}} = H \times W \times C \times K_h \times K_w = H \times W \times 64 \times 9 = 576HW$$

۲. کانولوشن نقطه‌ای (Pointwise) با $C' = 64$:

$$FLOPs_{\text{pointwise}} = H \times W \times C \times C' = H \times W \times 64 \times 64 = 4096HW$$

۳. مجموع:

$$576HW + 4096HW = \boxed{4672HW}$$

نسبت این دو عدد به وضوح نشان می‌دهد مکانیزم‌های توجه مانند SE با وجود بهبود عملکرد، هزینه محاسباتی ناچیزی دارند.

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؟

جستجوی معماری عصبی (NAS) در MobileNetV۳ با هدف یافتن معماری بهینه‌شده برای دستگاه‌های موبایل انجام شد. فضای جستجوی NAS شامل مؤلفه‌های زیر بود:

- **انواع بلوک‌های ساختاری**: بلوک‌های residual inverted با لایه‌های گسترش (expansion) و فشرده‌سازی (squeeze-and-excitation).
- **پارامترهای لایه**: اندازه کرنل‌ها (۳x۳، ۵x۵)، تعداد فیلترها، نرخ گسترش (expansion ratio) و فعال‌سازها.
- **توپولوژی شبکه**: تعداد لایه‌ها، ترتیب بلوک‌ها و اتصالات بین آن‌ها.

مدل‌سازی فضای جستجو به عنوان گراف جهت‌دار بدون دور (DAG):

- هر **گره** در گراف نماینده یک عملیات (مانند کانولوشن، فشرده‌سازی، یا فعال‌سازی) است.
- **یال‌ها** جهت جریان داده و پارامترهایی مانند استراید (stride) یا تعداد فیلترها را مشخص می‌کنند.
- NAS با جستجو در این گراف، ترکیبی از گره‌ها و یال‌ها را برای رسیدن به دقت بالا و هزینه محاسباتی پایین انتخاب می‌کند.

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

این تکنیک شامل مراحل زیر است:

۱. **آموزش مدل پایه**: آموزش یک مدل نسبتاً بزرگ با دقت بالا.

۲. **بهینه‌سازی تدریجی**:

- **Pruning (هرس)**: حذف نورون‌ها یا اتصالات کم‌اهمیت بر اساس معیارهایی مانند مقدار وزن.
- **Quantization (کاهش دقت)**: تبدیل پارامترها به دقت‌های پایین‌تر (مانند ۸ـ بیتی).
- **جایگزینی بلوک‌های سنگین**: استفاده از بلوک‌های سبک‌تر مانند separable convolutions.

۳. **جستجوی مبتنی بر کارایی**: در نظر گرفتن هزینه محاسباتی و مصرف حافظه به عنوان قید در NAS