دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

گزارش درس شبکه‌های عصبی و یادگیری عمیق

شبکه کانولوشنی نامتراکم

نگارش

مینو دولت‌آبادی

استاد درس

دکتر رضا صفابخش

بهمن ۱۴۰۱

چکیده

علی‌رغم اینکه شبکه‌های عصبی کانولوشنی عمیق عملکرد پیشرفته‌ای را در بسیاری از وظایف بینایی رایانه نشان می‌دهند، ماهیت غنی از پارامتر و محاسبات آن‌ها به طور قابل‌توجهی مانع استفاده کارآمد از این شبکه‌ها در دستگاه‌های با پهنای باند و توان محدود می‌شود. برای این منظور، در سال‌های اخیر علاقه زیادی در زمینه به حداقل رساندن حافظه و هزینه‌های محاسباتی استنتاج شبکه‌های عصبی مشاهده شده است. یکی از روش های مقابله با این مسائل استفاده از شبکه‌های عصبی کانولوشنی نامتراکم است. استفاده از این شبکه ها کاملاً کارآمد است زیرا به کمک آن‌ها می‌توان افزونگی در پارامترها را کاهش داد و از طرفی نیازی به اسکن همه پیکسل‌ها یا وکس‌های فضایی نیست و فقط کانولوشن برای عناصر غیر صفر محاسبه می‌شود. لذا در این گزارش به بررسی شبکه‌های عصبی کانولوشنی نامتراکم و تعمیم حاصل از آن‌ها پرداخته می‌شود.

واژه‌های کلیدی:

شبکه عصبی کانولوشنی، مینکوفسکی، کوانتیزاسیون، شبکه عصبی نامتراکم، تشخیص اشیاء سه‌بعدی، شبکه‌های عصبی باقی‌ماندگی

فهرست مطالب

صفحه

فهرست اشکال

صفحه

فهرست جداول

صفحه

فهرست اختصارات

عنوان اختصاری	عنوان کامل
کانا	کانولوشنی نامتراکم
کاسابنا	کانولوشنی ساختار یافته به‌هم‌پیوسته نامتراکم
کافنا	کانولوشنی فوق نامتراکم
کافنا-نت	کانولوشنی فوق نامتراکم-نت
کاناس	کانولوشنی نامتراکم سریع
کاناکا	کانولوشنی نامتراکم کانونی

فصل اول

مقدمه

مقدمه

امروزه شبکه‌های عصبی کانولوشنی عمیق [1]با اندازه مدل کوچک، هزینه محاسبات پایین و در عین‌حال دقت بالا به ویژه در دستگاه‌های تلفن همراه بسیار مورد توجه قرار گرفته‌اند. جهت دستیابی به چنین شبکه‌هایی تلاش‌هایی صورت گرفته است:

- فشرده‌سازی شبکه: فشرده‌سازی مدل از پیش آموزش‌دیده شده با تجزیه ماتریس هسته کانولوشنی یا حذف اتصالات یا کانال‌ها برای حذف افزونگی[2]
- طراحی معماری: طراحی هسته‌های کوچک، هسته‌های نامتراکم یا استفاده از محصول هسته‌هایی که اهمیت بیشتری دارند، جهت نزدیک شدن به یک تک هسته و آموزش شبکه‌ها از ابتدا.

تمرکز این کار مطالعاتی در طراحی معماری با استفاده از محصول هسته‌هایی که اهمیت بیشتری دارند برای ترکیب یک هسته است. برای تحقق این امر دو روند اصلی وجود دارد :

- ضرب هسته‌های با رتبه پایین(ماتریس) برای تقریب یک هسته با رتبه بالا، به عنوان مثال ماژول های تنگنا[3] [۱].
- ضرب ماتریس‌های نامتراکم، که اخیراً تلاش های تحقیقاتی را به خود جلب کرده است [۲] [۳].

الگوریتم‌های اخیراً توسعه یافته، مانند کانولوشن گروهی درهم[4] [۲] و Xception [۳]، یک هسته متراکم را با استفاده از حاصل ضرب دو هسته ساختاریافته - نامتراکم می‌سازند. مشاهده شده است که یکی از این دو هسته را می‌توان بیشتر تقریب زد. با انگیزه مشاهده اینکه کانولوشن‌های موجود در یک کانولوشن گروهی در IGC را می‌توان به همان روش بیشتر تجزیه کرد، ابتدادر فصل دوم کاسابنا(IGC-V2) معرفی می‌شود[۴]. سپس در بخش سوم به معرفی یک استراتژی کوانتیزه‌سازی جدید پرداخته می‌شود[۵]. در بخش چهارم و پنجم به ترتیب دو تعمیم از این شبکه‌ها با نام‌های کافنا و کاناس

[1] Deep convolutional neural networks

[2] redundancy

[3] bottleneck

[4] interleaved group convolution

معرفی می‌شوند[۶] [۷]. در بخش ششم توضیح داده می‌شود که چگونه این نوع از شبکه‌ها در تشخیص اشیاء سه‌بعدی مفید هستند[۸] و در نهایت در بخش آخر نوع جدیدی از این شبکه‌ها به نام شبکه‌های عصبی کانولوشنی مینکوفسکی معرفی و بررسی می‌شوند[۹].

فصل دوم

شبکه‌های عصبی کانولوشنی ساختار یافته به‌هم‌پیوسته نامتراکم [5]

[5] Interleaved Structured Sparse Convolutional Neural Networks

مقدمه

در این بخش قرار است مسئله طراحی معماری شبکه‌های عصبی کانولوشنی با توجه به حذف افزونگی در هسته‌های کانولوشنی مورد بررسی قرار گیرد. در این بخش کاسابنا(IGC-V2) (شکل ۱-۲) معرفی می‌شود. این شبکه، کانولوشن‌های گروهی که از دو هسته نامتراکم ساختاریافته تشکیل شده است را به محصول هسته‌های نامتراکم ساختار یافته‌تر تعمیم می‌دهد و افزونگی را بیشتر حذف می‌کند.

۲-۱-معرفی

عملیات در یک لایه کانولوشن در شبکه‌های عصبی کانولوشنی به یک عملیات ضرب بردار-ماتریسی در هر مکان متکی است:

$$\mathbf{y} = \mathbf{W}\mathbf{x} \qquad (۱-۱)$$

در این‌جا ورودی $\mathbf{x}$ متناظر با یک وصله در اطراف مکان کانال‌های ورودی و یک بردار SC_i -بعدی است، به‌طوری‌که S برابر با اندازه هسته (برای مثال 3×3) و C_i برابر با تعداد کانال‌های ورودی است . خروجی y نیز یک بردار C_o -بعدی است و همچنین متناظر با تعداد کانال‌های خروجی است . $\mathbf{w}$ از هسته‌های C_o تشکیل می‌شود و هر ردیف متناظر با هسته کانولوشن است. در این مقاله جهت افزایش شفافیت فرض می‌شود: $C_o = C_i = C$.

بلوک W_i^g و کانولوشن ۱×۱ در Xception متراکم هستند و می‌تواند با ضرب ماتریس‌های نامتراکم تشکیل شود. در نتیجه جهت بیشتر از بین بردن افزونگی و صرفه‌جویی در ذخیره‌سازی و زمان چنین فرایندی را می‌توان تعداد بارهای بیشتری تکرار کرد.

$$\begin{aligned}\mathbf{y} &= \mathbf{P}_L\mathbf{W}_L\mathbf{P}_{L-1}\mathbf{W}_{L-1}\dots\mathbf{P}_1\mathbf{W}_1\mathbf{x} \\ &= \left(\prod_{l=L}^{1}\mathbf{P}_l\mathbf{W}_l\right)\mathbf{x}.\end{aligned} \qquad (۲-۱)$$

در معادله مذکور $\mathbf{P}_l\mathbf{W}_l$ یک ماتریس نامتراکم است. $\mathbf{P}_l$ یک ماتریس جایگشتی[6] است، و نقش آن مرتب کردن مجدد کانال‌ها به گونه‌ای است که $\mathbf{W}_l$ یک ماتریس بلوکی نامتراکم باشد، و متناظر با کانولوشن گروه lام است، که در آن تعداد کانال‌ها در همه شاخه‌ها یکسان و برابر K_l هستند.

۲-۲-ساخت یک ماتریس هسته متراکم تشکیل‌شده

در ابتدا شرط زیر که تعمیم یافته کانولوشن‌های گروهی به هم پیوسته است به عنوان یک قاعده برای ساخت کانولوشن‌های گروه l به‌گونه‌ای که ماتریس هسته کانولوشن تشکیل‌شده متراکم باشد، مطرح می‌شود.

شرط تعمیم‌یافته: $\forall \mathrm{m}$ ، $(\mathbf{W}_L \prod_{l=L-1}^{m} \mathbf{P}_l\mathbf{W}_l)$ و $(\mathbf{W}_{L-1} \prod_{l=L-2}^{m} \mathbf{P}_l\mathbf{W}_l)$ متناظر با دو گروه کانولوشنی هستند. دو کانولوشن گروهی فوق را مکمل یکدیگر گویند اگر کانال‌هایی که در یک شاخه در یک کانولوشن گروهی قرار دارند در شاخه‌های مختلف نهفته باشند و از همه شاخه‌های کانولوشن گروه دیگر آمده باشند.

اکنون سوالی که مطرح می‌شود این است که چه زمانی مقدار پارامترها کوچکترین است؟

می‌توان گفت که تعداد پارامترهای کانولوشن گروه L که درمعادله (۱-۲) ارائه شده است و در شرط تعمیم‌یافته صدق می‌کند، پاسخ این سوال است.

تعداد پارامترها در کانولوشن گروه lام CK_l برای یک کانولوشن 1×1 و $CSK_{\ l}$ برای کانولوشن فضایی مانند $s = 3 \times 3$ است. جهت استفاده از پارامترهای کمتر تنها یک کانولوشن فضایی گروهی وجود دارد و بقیه 1×1 هستند. کانولوشن فضایی در هر کانولوشن گروهی نهفته است و بدون تأثیر بر تجزیه و تحلیل، فرض می‌شود که در کانولوشن گروه اول قرار دارد. بنابراین، تعداد کل پارامترهای Q، تعداد کوچکتری از پارامترها که در ماتریس‌های جایگشت نادیده گرفته شده‌اند، برابر است با:

$$Q = C\sum_{l=2}^{L} K_l + CSK_1 \qquad (۳-۲)$$

[6] permutation matrix

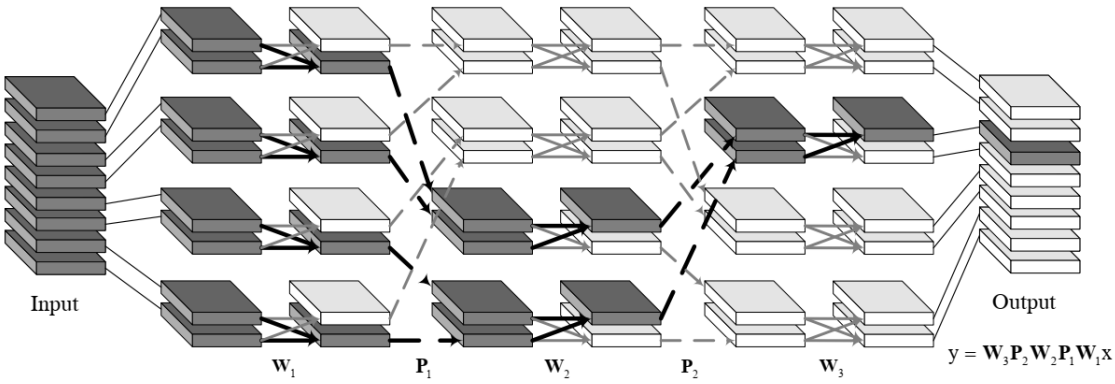
با توجه به نابرابری جنسن[۷]:

$$Q = C\sum_{l=2}^{L} K_l + CSK_1 \geqslant CL\left(SK_1\prod_{l=2}^{L} K_l\right)^{\frac{1}{L}} = CL(SC)^{\frac{1}{L}} \quad (۴-۲)$$

در این‌جا، برابری از خط دوم تا خط ســوم به دلیل معادله (۲-۲) برقرار اســت. برابری در خط دوم برقرار است، یعنی $Q = CL(SC)^{\frac{1}{L}}$، زمانی که شرط تعادل زیر برآورده شود:

$$SK_1 = K_2 = \cdots = K_L \left(= (SC)^{\frac{1}{L}}\right) \quad (۵-۲)$$

علاوه بر این، برای انتخاب L که کمترین مقدار پارامتر را به دســت می‌دهد، یک تحلیل تقریبی با در نظر گرفتن مشتق Q با توجه بهL ارائه می‌شود.

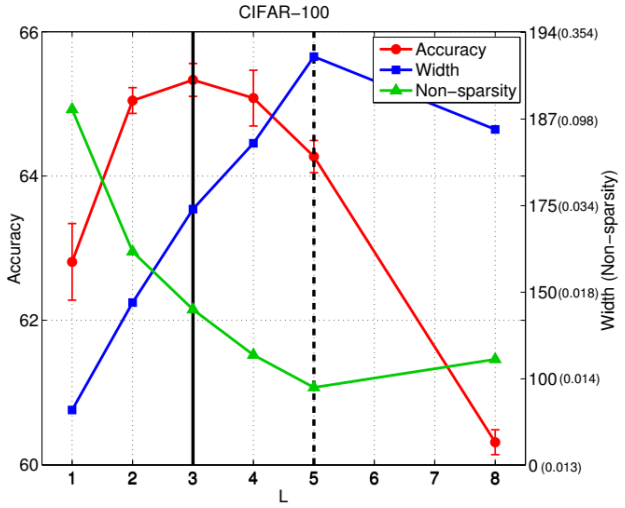


شکل ۲-۱-IGC - V2: شبکه عصبی کانولوشنی ساختار یافته به‌هم‌پیوسته نامتراکم.

[7] Jensen’s inequality

۲-۳-نتایج:

در نمودار زیر نشان داده می‌شود که چگونه تعداد لایه‌های L بر عملکرد CIFAR - 100 تأثیر می‌گذارد. حداکثر دقت در مقداری L از به‌دست می‌آید که در آن عرض و درجه عدم پراکندگی به تعادل می‌رسند.

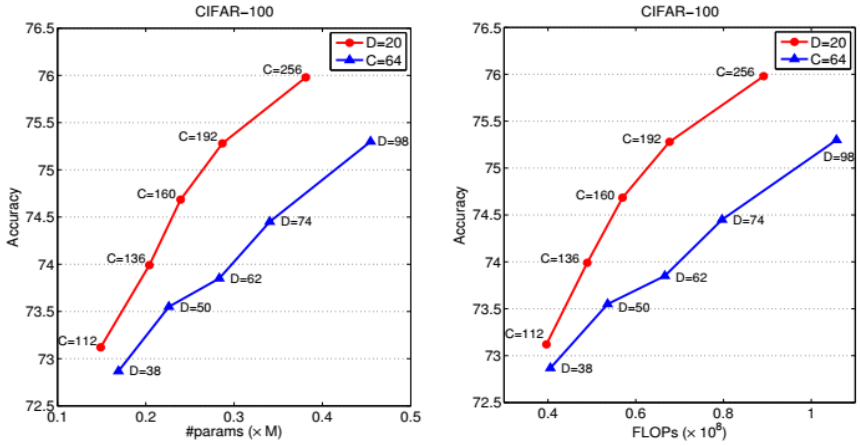


شکل ۲-۰- تاثیر تعداد لایه‌های L.

نمودار زیر چگونگی تغییر عملکرد را زمانی که شبکه ما عمیق‌تر یا گسترده‌تر می‌شود، نشان میدهد. در این مقاله از ساختار شبکه IGC-V2 * (Cx) در جدول ۲-۱ استفاده می‌شود و آزمایش‌ها با اعماق مختلف (D) و با عرض (C) انجام میشوند. با توجه به نمودار هم عریض‌تر شدن و هم عمیق تر شدن کارایی را افزایش می‌دهد و اثر عریض‌تر شدن بیشتر از عمیق‌تر شدن است.

جدول ۲-۱: معماری استفاده شده شبکه‌هایی که در آزمایش‌ها .

Output size	Xception (Cx)	IGC-V1 (Cx)	IGC-V2 (Cx)	IGC-V2* (Cx)
32×32	$(3 \times 3, x)$	$(3 \times 3, x)$	$(3 \times 3, x)$	$(3 \times 3, 64)$
32×32	$\begin{bmatrix} x \times (3 \times 3, 1) \\ (1 \times 1, x) \end{bmatrix} \times B$	$\begin{bmatrix} \frac{x}{2} \times (3 \times 3, 2) \\ 2 \times (1 \times 1, \frac{x}{2}) \end{bmatrix} \times B$	$\begin{bmatrix} x \times (3 \times 3, 1) \\ L-1, x, (1 \times 1, K_{s_1}) \end{bmatrix} \times B$	$\begin{bmatrix} x \times (3 \times 3, 1) \\ L^*-1, x, (1 \times 1, K) \end{bmatrix} \times B$
16×16	$\begin{bmatrix} 2x \times (3 \times 3, 1) \\ (1 \times 1, 2x) \end{bmatrix} \times B$	$\begin{bmatrix} x \times (3 \times 3, 2) \\ 2 \times (1 \times 1, x) \end{bmatrix} \times B$	$\begin{bmatrix} 2x \times (3 \times 3, 1) \\ L-1, 2x, (1 \times 1, K_{s_2}) \end{bmatrix} \times B$	$\begin{bmatrix} 2x \times (3 \times 3, 1) \\ L^*-1, 2x, (1 \times 1, K) \end{bmatrix} \times B$
8×8	$\begin{bmatrix} 4x \times (3 \times 3, 1) \\ (1 \times 1, 4x) \end{bmatrix} \times B$	$\begin{bmatrix} 2x \times (3 \times 3, 2) \\ 2 \times (1 \times 1, 2x) \end{bmatrix} \times B$	$\begin{bmatrix} 4x \times (3 \times 3, 1) \\ L-1, 4x, (1 \times 1, K_{s_3}) \end{bmatrix} \times B$	$\begin{bmatrix} 4x \times (3 \times 3, 1) \\ L^*-1, 4x, (1 \times 1, K) \end{bmatrix} \times B$
1×1	average pool, fc, softmax			
Depth	$3B + 2$			



شکل ۳-۰- عملکرد شبکه بر اثر تغییر عمق و عرض.

فصل سوم

گسسته‌سازی متمرکز برای شبکه‌های عصبی کانولوشنی نامتراکم[8]

[8] Focused Quantization for Sparse CNNs

مقدمه

در این بخش، به ویژگی‌های آماری شبکه‌های عصبی کانولوشنی پراکنده پرداخته می‌شود و کوانتیزاسیون متمرکز را ارائه می‌دهد، یک استراتژی کوانتیزه‌سازی جدید مبتنی بر توان دو مقادیر، که از توزیع وزن پس از هرس دقیق استفاده می‌کند. روش پیشنهادی به صورت پویا موثرترین نمایش عددی را برای وزن‌ها در لایه‌هایی با پراکندگی‌های مختلف کشف می‌کند که به طور قابل‌توجهی اندازه مدل را کاهش می‌دهد.

۳-۱- روش

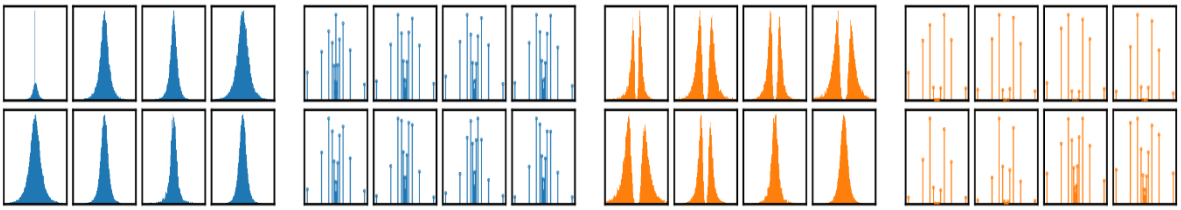
۳-۱-۱-مقدمات: انتقال کوانتیزاسیون[۹]

انتقال کوانتیزاسیون یک طرح کوانتیزه‌سازی است که مقادیر وزن را به توان‌های دو یا صفر محدود می‌کند. یک مقدار قابل نمایش برای انتقال کوانتیزاسیون $(\mathrm{k}+2)$ –بیتی به صورت زیر است:

$$v = s \cdot 2^{e-b} \qquad (۱-۳)$$

که در آن $\mathrm{s} = \{-1.0.1\}$ نشان‌دهنده صفر یا علامت مقدار است، e یک عدد صحیح محدود به بازه $[0.2^k - 1]$ و b بایاس است، یک مقدار لایه‌ای ثابت که میزان بزرگی مقدار کوانتیزه را مدیریت می‌کند. در این مقاله از $\hat{\theta} = \mathrm{Q}_{n.b}^{shift}[\theta]$ برای نشان دادن انتقال کوانتیزاسیون n بیتی با بایاس b از مقدار وزن θ به نزدیکترین مقدار قابل نمایش $\hat{\theta}$ استفاده شده است. همان‌طور که در شکل ۳-۱ نشان داده شده است، انتقال کوانتیزاسیون در لایه‌های نامتراکم موجب استفاده جزئی از محدوده مقادیر قابل مشاهده می‌شود، یعنی توزیع حاصل پس از کوانتیزه‌سازی $\mathrm{q}_{n.b}^{shift}(\theta)$ تقریبی ضعیف از توزیع وزن لایه اصلی $p(\theta|\mathcal{D})$ است، که در آن $\mathcal{D}$ مجموعه داده آموزشی است.

[9] Shift quantization



شکل ۳-۱- توزیع وزن ۸ لایه اول *ResNet-18* در *ImageNet* الف) توزیع وزن لایه‌ها ، (ج) توزیع وزن لایه‌ها (بدون احتساب صفرها) برای یک نوع نامتراکم. (ب) و (د) به ترتیب توزیع وزن نمودارهای سمت چپ انتقال کوانتیزاسیون ۵ بیتی.

۳-۱-۲-طراحی تابع کوانتیزاسیون متمرکز

به طور شهودی مطلوب است که کوانتیزاسیون بر روی مناطق با احتمال بالا در توزیع وزن در لایه‌های نامتراکم متمرکز شود. در اثر انجام این کار، ما می‌توانیم توزیع وزن‌های کولنتیزه شده را با مقادیر اصلی مطابقت دهیم ,و در نتیجه همزمان خطاهای گرد کردن کوچک‌تر می‌شوند. کوانتیزاسیون غیرمتمرکز $\mathrm{Q}[\theta]$ به طور خاص برای این منظور طراحی شده است و به صورت لایه‌ای اعمال می‌شود. با فرض اینکه $\theta \in \boldsymbol{\theta}$ مقدار وزن یک لایه کانولوشن باشد، می‌توان $\mathrm{Q}[\theta]$ را به صورت زیر تعریف کرد:

$$\mathrm{Q}[\theta] = z_\theta \alpha \sum_{c \in C} \delta_{c.\mathrm{m}_\theta} \mathrm{Q}_c^{\mathrm{rec}}[\theta]. \text{ where } \mathrm{Q}_c^{\mathrm{rec}}[\theta] = \mathrm{Q}_{\mathrm{n.b}}^{\mathrm{shift}}\left[\frac{\theta - \mu_c}{\sigma_c}\right]\sigma_c + \mu_c \qquad (۲-۳)$$

در این‌جا z_θ یک ثابت از پیش تعیین شده $\{0.1\}$ دودویی مقدار است که برای نشان دادن اینکه θ هرس شده است یا خیر، و برای تنظیم وزن‌های هرس شده روی 0 استفاده می‌شود. مجموعه مولفه‌های $c \in C$ مکان‌هایی را جهت تمرکز برای اعمال کوانتیزه‌سازی تعیین می‌کند که هر کدام با میانگین μ_c و انحراف معیار σ_c مشخص شده‌اند. دلتای کرونکر[۱۰] $\delta_{\mathrm{c.m}_\theta}$ زمانی که $c = m_\theta$ برابر با یک یا در غیر

[10] Kronecker delta

این صورت صفر ارزیابی می شود. به عبارت دیگر، ثابت $m_\theta \epsilon C$ تعیین می‌کند که کدام جزء در C برای کوانتیزه کردن θ استفاده شود. در نهایت، $Q_{rec}^{c}[\theta]$ به صورت محلی مولفه c را با انتقال کوانتیزاسیون، کوانتیزه می‌کند. علاوه بر این، یک عامل مقیاس‌پذیری قابل یادگیری از نظر لایه به نام α معرفی شده‌است که مقدار ۱ را می‌گیرد، و به طور تجربی دقت را بهبود می‌بخشد. بنابراین، با تنظیم μ_c و σ_c هر جزءc و یافتن تخصیص مناسب وزن‌ها به مؤلفه‌ها، توزیع وزن کوانتیزه‌شده $q_\phi(\theta)$ می‌تواند بسیار نزدیک با توزیع اصلی باشد، جایی که ϕ به عنوان مخفف برای نشان دادن ابرپارامترهای مربوطه مانند μ_c و σ_c استفاده می‌شود. در بخش زیر توضیح داده خواهد شد که چگونه می‌توان این ابرپارامترها را به طور موثر بهینه کرد.

۳-۱-۳- بهینه‌سازی کوانتیزاسیون متمرکز $Q[\theta]$

ابرپارامترهای μ_c وσ_c در کوانتیزه‌سازی مجدد متمرکز می‌توانند با اعمال فرآیند دو مرحله ای زیر که به صورت لایه ای است بهینه شوند، ابتدا مناطق با احتمال بالا را شناسایی می‌کند (نخستین بلوک در شکل ۳-۲)، سپس آن‌ها را به صورت محلی با انتقال کوانتیزاسیون (بلوک دوم و سوم در شکل ۳-۲) کوانتیزه می‌کند. در ابتدا متوجه می‌شویم که توزیع وزن به طور کلی شبیه ترکیبی از توزیع‌های گوسی است. لذا یافتن یک مدل آمیخته گوسی [11] $q_\phi^{mix}(\theta)$ که توزیع اصلی $p(\theta|\mathcal{D})$ را جهت بهینه‌سازی دقیق هدف بالا تقریب می‌زند، موثرتر است:

$$q_\phi^{mix}(\theta) = \sum_{c \in C} \lambda_c f\left(\theta \mid \mu_c . \sigma_c\right) \qquad (۳-۳)$$

به‌طوری‌که $f(\theta \mid \mu_c . \sigma_c)$ تابع چگالی احتمال توزیع گاوسی $N\ (\mu_c . \sigma_c)$ می‌باشد، λ_c غیرمنفی است و وزن ترکیبی مولفه cام و $\Sigma_{c \epsilon C}\, \lambda_c = 1$ را تعیین می‌کند. در این مرحله بایستی ابرپارامترهای μ_c ،σ_c و λ_c موجود در ϕ را یافت به‌طوری‌که $q_\phi^{mix}(\theta)$ را به حداکثر می‌رساند با توجه به اینکه $\theta \sim p(\theta|\mathcal{D})$. این مسئله به عنوان تخمین حداکثر درستنمایی [12] (MLE) شناخته می‌شود، و MLE را می‌توان به طور

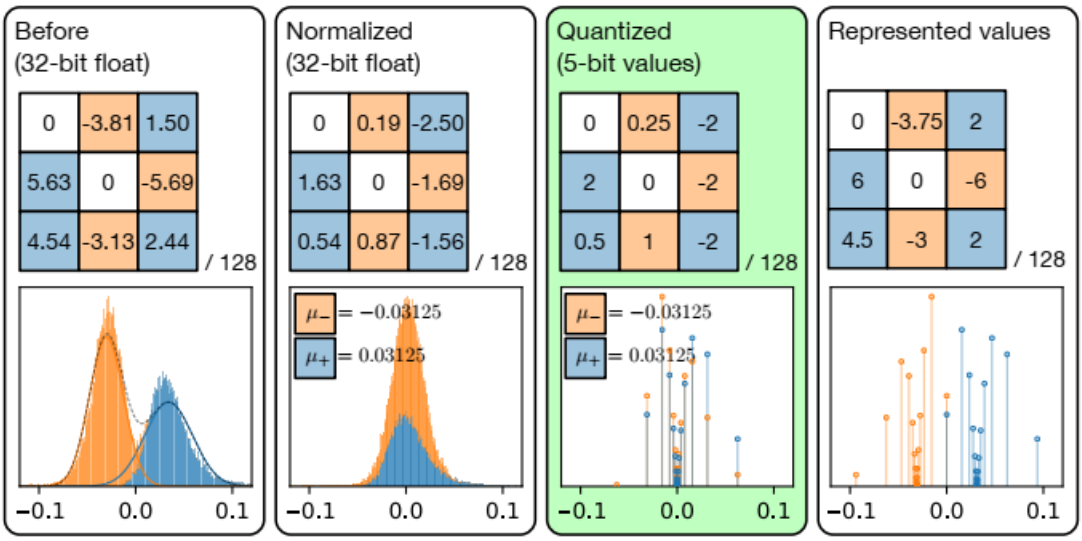
[11] gaussian mixture model

[12] maximum likelihood estimate

موثر توسط الگوریتم حداکثرسازی امید ریاضی[۱۳] (EM) محاسبه کرد. در عمل، استفاده از دو مولفه گاوسی، $C = \{+.-\}$ برای شناسایی مناطق با احتمال بالا در توزیع وزن کافی است. جهت همگرایی سریعتر EM، μ_- ، σ_- و μ_+ ، σ_+ به ترتیب با میانگین و انحراف معیار مقادیر منفی و مثبت در وزن لایه λ_- و λ_+ با $\frac{1}{2}$ مقداردهی اولیه می‌شوند.

سپس m_θ از مدل آمیخته تولید می‌شود، به‌طوری‌که به صورت جداگانه مولفه مورد استفاده برای هر وزن را انتخاب می‌کند. برای این کار، m_θ برای هر θ با نمونه‌برداری از یک توزیع طبقه‌ای ارزیابی می‌شود که در آن احتمال اختصاص یک مولفه c به m_θ، یعنی $p(m\theta = c)$، $\lambda_c f(\theta \mid \mu_c . \sigma_c) \Big/ q_\phi^{mix}(\theta)$ است.

در نهایت، ثابت b روی یک مقدار توان دو تنظیم می‌شود، برای اطمینان از اینکه $q_{n.b}^{shift}[\cdot]$ اجازه می‌دهد تا حداکثر نسبت $\frac{1}{2^n+1}$ از مقادیر سرریز شود و آنها را به حداکثر مقدار قابل نمایش تقریب می‌زند. در عمل، این انتخاب اکتشافی نتیجه بهتری از سطوح کوانتیزاسیون ارائه شده توسط انتقال کوانتیزاسیون نسبت به عدم اجازه سرریزها می‌دهد. پس از تعیین تمام ابرپارامترهای مربوطه با روشی که در بالا توضیح داده شد، $\hat{\theta} = Q[\theta]$ را می‌توان برای کوانتیزه کردن وزن لایه‌ها ارزیابی کرد.

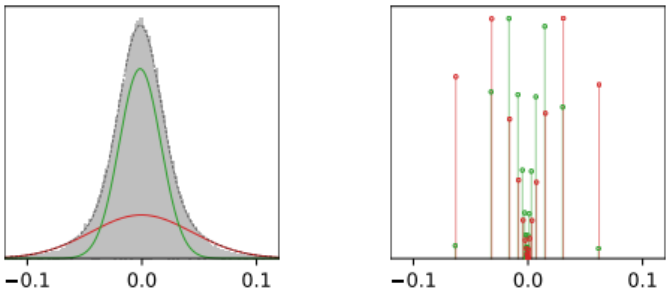


شکل ۳-۲- فرآیند گام به گام کوانتیزه‌سازی مجدد وزن‌های هرس نشده در block3f/conv1

[13] expectation-maximization

۳-۱-۴- انتخاب کوانتیزاسیون مناسب

توزیع وزن لایه‌های نامتراکم ممکن است همواره دارای چندین ناحیه با احتمال بالا نباشد. به عنوان مثال، برازش یک مدل آمیخته از دو مولفه گاوسی بر روی لایه در شکل ۳-۳، اجزای با همپوشانی زیادی را به دست می‌دهد. بنابراین این که از کدام مؤلفه برای کوانتیزه کردن یک مقدار وزن خاص استفاده می‌شود، تأثیر چندانی ندارد. تحت این سناریو، می‌توان به سادگی از انتقال کوانتیزه‌سازی n بیتی $[\cdot]$ $Q_{\mathrm{n.b}}^{shift}$ به جای یک n بیت $Q[\cdot]$ استفاده کرد که در داخل خود از یک $(\mathrm{n}-1)$ بیت علامت‌گذاری شده انتقال استفاده می‌کند.



شکل۳-۳: توزیع وزن لایه block22/conv1 در ResNet-18 نامتراکم آموزش دیده در ImageNet.

برای تصمیم گیری در مورد استفاده از انتقال یا کوانتیزاسیون مجدد، لازم است یک متریک برای مقایسه شباهت بین جفت مولفه‌ها معرفی شود. در حالی که واگرایی KL معیاری را برای تشابه ارائه می‌دهد، اما غیر متقارن است لذا برای این منظور نامناسب است. بنابراین در این مقاله پیشنهاد داده شده است که ابتدا توزیع مخلوط را نرمال کنید، سپس از متریک وسرترین[۱۴] ۲ بین دو مؤلفه گاوسی پس از نرمال‌سازی به عنوان یک معیار تصمیم‌گیری استفاده نمایید، این جداسازیوسرتین نامیده می‌شود.

$$\mathcal{W}(c_1.c_2)=\frac{1}{\sigma^2}\left(\left(\mu_{c_1}-\mu_{c_2}\right)^2+\left(\sigma_{c_1}-\sigma_{c_2}\right)^2\right) \qquad (۴-۳)$$

که در آن μ_c و σ_c به ترتیب میانگین و انحراف معیار مولفه $\mathrm{c}\in\{\mathrm{c}_1.\mathrm{c}_2\}$ هستند. σ^2 نیز نشان‌دهنده واریانس کل توزیع وزن است. FQ می‌تواند به‌طور تطبیقی برای استفاده از کوانتیزه‌سازی مجدد برای

[14] Wasserstein

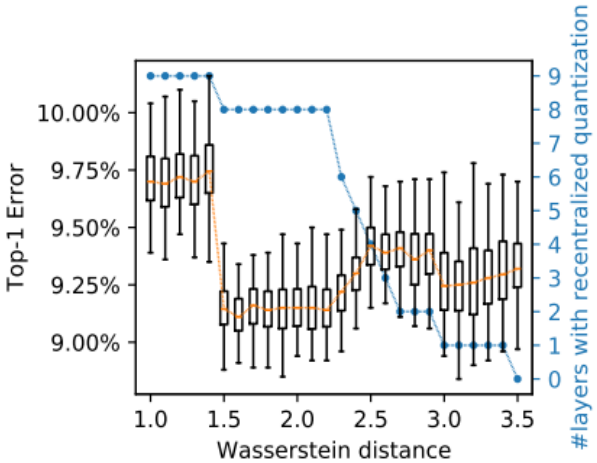
همه لایه‌های نامتراکم انتخاب کند، به جز زمانی که $\mathcal{W}(c_1.c_2) < w_{sep}$ ، و به جای آن از انتقال کوانتیزاسیون استفاده می‌شود.

۳-۲-۵: بهینه‌سازی مدل

برای بهینه‌سازی مدل نامتراکم کوانتیزه‌شده، فرآیند کوانتیزاسیون‌سازی که در بالا توضیح داده شد در آموزش پارامترهای مدل مبتنی بر گرادیان ادغام می‌شود. در ابتدا، ابرپارامترهای μ_c ،σ_c و λ_c برای هر لایه محاسبه می‌شوند، و ماسک انتخاب مولفه m_θ برای هر وزن θ با روش بخش ۳-۲-۳ ایجاد می‌شود. سپس مدل به‌دست‌آمده در جایی که حرکت رو به جلو از وزن‌های کوانتیزه $\hat{\theta} = Q[\theta]$ استفاده می‌کند، تنظیم می‌شود، و حرکت رو به عقب پارامترهای وزن ممیز شناور θ را با در نظر گرفتن کوانتیزاسیون به‌عنوان یک تابع هویت به‌روزرسانی می‌کند. در طول فرآیند تنظیم دقیق، ابرپارامترهای استفاده شده توسط $Q[\theta]$ با استفاده از توزیع وزن فعلی در هر k دوره بروز می‌شوند.

۳-۲-نتایج

تأثیر مقادیر مختلف آستانه بر فاصله وسترین در نمودار زیر قابل مشاهده است. هرچه آستانه بزرگتر باشد، تعداد لایه‌هایی که از کوانتیزه‌سازی مجدد به جای انتقال کوانتیزه‌سازی استفاده می‌کنند، کمتر می‌شود.



شکل ۳-۴: تأثیر مقادیر مختلف آستانه بر فاصله وسترین

دقت، پراکندگی و CR های فشـــرده‌ســـازی متمرکز در مدل‌های ImageNet. مدل‌های پایه قبل از فشـرده‌سازی مدل‌های متراکم هسـتند و از وزن‌های ممیز شـناور ۳۲ بیتی اسـتفاده می‌کنند و ۵ بیت و ۷ بیت نشـان‌دهنده تعداد بیت‌های اسـتفاده شـده توسـط وزن‌های جداگانه مدل‌های کوانتیزه شـده قبل از رمزگذاری هافمن است.

جدول ۳-۱: دقت، پراکندگی و CR های فشرده‌سازی متمرکز در مدل‌های ImageNet

Model	Top-1	Δ	Top-5	Δ	Sparsity	Size (MB)	CR (×)
ResNet-18	68.94	—	88.67	—	0.00	46.76	—
Pruned	69.24	0.30	89.05	0.38	74.86	8.31	5.69
5 bits	68.36	-0.58	88.45	-0.22	74.86	2.86	16.33
7 bits	68.57	-0.37	88.53	-0.14	74.86	2.94	15.92
ResNet-50	75.58	—	92.83	—	0.00	93.82	—
Pruned	75.10	-0.48	92.58	-0.25	82.70	11.76	7.98
5 bits	74.86	-0.72	92.59	-0.24	82.70	5.19	18.08
7 bits	74.99	-0.59	92.59	-0.24	82.70	5.22	17.98
MobileNet-V1	70.77	—	89.48	—	0.00	16.84	—
Pruned	70.03	-0.74	89.13	-0.35	33.80	6.89	2.44
7 bits	69.13	-1.64	88.61	-0.87	33.80	2.13	7.90
MobileNet-V2	71.65	—	90.44	—	0.00	13.88	—
Pruned	71.24	-0.41	90.31	-0.13	31.74	5.64	2.46
7 bits	70.05	-1.60	89.55	-0.89	31.74	1.71	8.14

فصل چهارم

شبکه‌های عصبی کانولوشنی فوق نامتراکم[۱۵]

[15] Super Sparse Convolutional Neural Networks

مقدمه

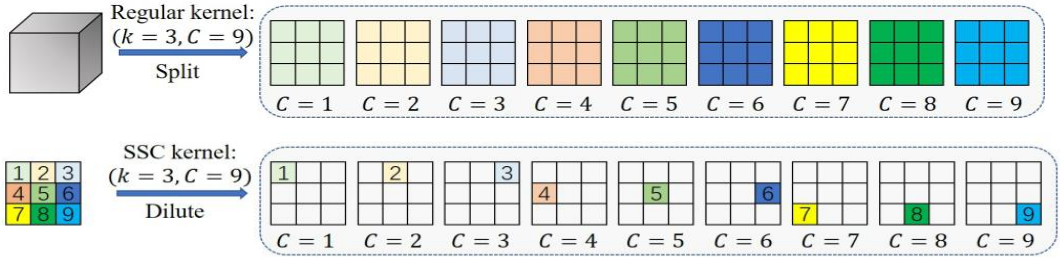
برای ساخت شبکه‌های موبایل کوچک، بدون از دست دادن عملکرد و رسیدگی به مسائل بیش از حد برازش ناشی از مجموعه داده‌های آموزشی، در این بخش یک هسته کافنا پیشنهاد می‌شود و شبکه متناظر آن کافنا-نت نامیده می‌شود.

۴-۱-هسته کافنا

در شکل ۴-۱، یک هسته کافنا را می‌توان به عنوان تضعیف کردن یک هسته فضایی دو بعدی (به نام هسته اصلی) به یک هسته سه بعدی مشاهده کرد. به این ترتیب، مکان‌های غیرصفر فضایی با هسته اصلی یکسان نگه داشته می‌شوند که می‌تواند ویژگی‌های هندسی عمومی را حفظ کند. و فرآیند تضعیف کردن می‌تواند تفاوت‌های ویژگی‌ها را در طول کانال حفظ کند. فرض کنید هسته کانولوشن معمولی ۴ بعدی $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{k\times k\times C\times D}$ باشد به طوری‌که $\mathrm{k}\times\mathrm{k}$ اندازه هسته فضایی را نشان می دهد. C و D به ترتیب به تعداد کانال‌های ورودی و خروجی اشاره دارد. هسته کافنا با $\boldsymbol{S} \in \mathbb{R}^{k\times k\times C\times D}$ نشان داده می‌شود. لازم ذکر است که $\boldsymbol{S}$ را می‌توان از تعمیم هسته اصلی $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{k\times k\times C\times D}$ به‌دست آورد. قابل توجه است که در یک هسته کافنا، $\mathrm{C}=\mathrm{k}\times\mathrm{k}$. بنابراین، تعریف فرمول کافنا می‌تواند بدین صورت باشد:

$$\boldsymbol{s}_{i,j}^{x.y}=\begin{cases}\boldsymbol{\mathcal{W}}_{j}^{x.y}. & i=x\times k+y\\ 0. & \text{otherwise}\end{cases} \qquad (۱-۴)$$

که x و y نشان‌دهنده موقعیت مکانی هستند و $x.y\epsilon\{0.1.2.k-1\}$. علاوه بر این $i\epsilon\{0.1.2.C-1\}$ و $j\epsilon\{0.1.2.D-1\}$.



شکل ۴-۱: مقایسه هسته کانولوشن معمولی و کافنا.

۴-۲-پیاده‌سازی عملگر کافنا

با توجه به یک تنسور[16] ورودی $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{w\times h\times C}$ که توسط هسته کافنا $\boldsymbol{\mathcal{T}}$ انجام می‌شود، تنسور خروجی مربوطه $\boldsymbol{\mathcal{O}} \in \mathbb{R}^{w\times h\times C}$ را می توان به صورت زیر به دست آورد:

$$\boldsymbol{\mathcal{O}}(x.y.q) = \sum_{p=0}^{C-1}\sum_{i=0}^{k-1}\sum_{j=0}^{k-1} \boldsymbol{\mathcal{T}}(i.j.p.q) \\ \boldsymbol{\mathcal{I}}(x+i-\delta_1.y+j-\delta_2.p) \qquad (۲-۴)$$

به طوری که $\delta_1 = \delta_2 = \left\lfloor {}^{k}/_{2} \right\rfloor$ و $q\epsilon\{0.1.2.D-1\}$. با توجه به معادله (۴-۱)، فرآیند محاسباتی عملیات کافنا را می‌توان به صورت زیر ساده کرد:

$$\boldsymbol{\mathcal{O}}(x.y.q) = \sum_{p=0}^{C-1} \boldsymbol{\mathcal{T}}\left(i_p.j_p.p.q\right) \\ \boldsymbol{\mathcal{I}}\left(x+i_p-\delta_1.y+j_p-\delta_2.p\right) \qquad (۳-۴)$$

به طوری که $\left(i_p.j_p\right)$ تنها مختصات مکانی یک پارامتر غیر صفر در کانال p ام است و $i_p \times \mathrm{k} + j_p = \mathrm{p}$. از معادله (۴-۳)، واضح است که یک مقدار خروجی در موقعیت مکانی خاص را می‌توان از طریق جمع C بار عملیات ضرب محاسبه کرد. علاوه بر این، از آنجایی که هر صفحه در یک هسته کافنا دارای یک پارامتر است، می‌توان آن را به یک هسته "نقطه‌ای" تبدیل کرد. هم‌چنین، به منظور حفظ روابط موقعیت مکانی در فرآیند محاسبات، از یک هسته انتقال $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{k\times k\times C\times D}$ استفاده می‌کنیم که بر روی نقشه‌های ویژگی برای گرفتن ویژگی‌ها عمل می‌کند. با توجه به موقعیت‌های مکانی غیر صفر هسته‌های کافنا عملیات انتقال در شکل ۴-۲ نشان داده شده است. تعریف این هسته انتقال به صورت زیر نشان داده شده است:

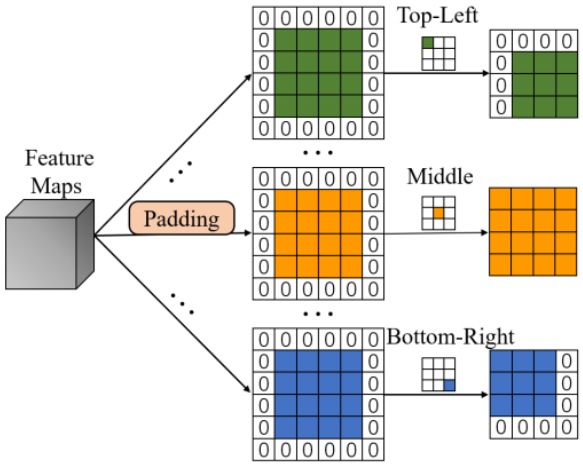
[16] tensor

$$\boldsymbol{\mathcal{P}}_{\boldsymbol{i.j}}^{\boldsymbol{x.y}} = \begin{cases} 1. & \boldsymbol{S}_{\boldsymbol{i.j}}^{\boldsymbol{x}} \neq 0 \\ 0. & \text{otherwise} \end{cases} \quad (۴-۴)$$

از معادله (۴-۳) و (۴-۴)، عملیات کافنا را می‌توان به صورت زیر فرموله کرد:

$$\boldsymbol{\mathcal{O}}(x.y.q) = \sum_{p=0}^{C-1} \boldsymbol{\mathcal{T}}(p.q)\boldsymbol{\mathcal{P}}\left(i_p.j_p.p.q\right) \boldsymbol{\mathcal{I}}\left(x + i_p - \delta_1.y + j_p - \delta_2.p\right) \quad (۵-۴)$$

از معادله (۴-۵)، عملیات کافنا را می‌توان با عملیات کانولوشنی انتقال و نقطه‌ای پیاده‌سازی کرد، که می‌تواند پارامترها و فلاپ‌های محاسباتی بسیار بیشتری را ذخیره کند.



شکل ۴-۲: فرآیند عملیاتی انتقال.

۴-۳-ماژول پایه کافنا و کافنا-نت[17]

در ابتدا یک لایه کانولوشن «نقطه‌ای» جهت نگاشت نقشه‌های ویژگی ورودی به یک فضای ابعادی مورد نیاز استفاده می‌شود، که همچنین می‌تواند ویژگی‌های خروجی ماژول قبلی را از گروه‌های مختلف ترکیب کند. کانال‌های نقشه ویژگی پیش‌بینی‌شده به‌دست‌آمده M هستند. سپس، دو لایه کافنا با یک عملیات

[17] SSC-Nets

"کانال مختلط"[18] در میان آن‌ها مجهز می‌شوند. قبل از هر عملیات کافنا، یک لایه کانولوشنال «گروهی نقطه‌ای» دیگر استفاده می‌شود تا هر ورودی کافنا را وادار کند تا در حد امکان ویژگی‌های هندسی مشابهی داشته باشد. در نهایت، نگاشت هویت نیز در ماژول کافنا اتخاذ شده است. ساختار جزئی ماژول پایه کافنا-نت در جدول ۴-۱ نشان داده شده است.

جدول ۴-۱:ساختار شبکه

Stage	Operation	Groups	Channels/Group
Conv1	1×1	1	M
Conv2	1×1	G	C
1^{st} SSC	*shift* 1×1	G G	C C
Shuffle	*shuffle*	C	G
Conv3	1×1	C	G
2^{nd} SSC	*shift* 1×1 1×1	G G C	C C G

۴-۴-مقایسه دو شبکه عصبی تلفن همراه پیشرفته

در نهایت، به منظور بررسی عملکرد کافنا-نت، کافنا-نت‌های مختلف با مدل مدرن تلفن همراه معرفی شده در مقاله ۱ مقایسه می‌شوند. درCIFAR ، همان‌طور که در جدول ۴-۲ نشان داده شده است که کافنا-نت می تواند عملکرد بسیار بهتری در CIFAR-10 ایجاد کند. در مورد CIFAR 100 ، از آن‌جایی‌که کافنا به طور کامل توسط کانولوشن‌های نقطه‌ای در عمل پیاده‌سازی می‌شود، کافنا-نت همواره بهترین دقت را به دست نمی‌آورد، اما می‌تواند الزامات دقت عمومی را برآورده کند.

جدول ۴-۲: مقایسه دو شبکه عصبی تلفن همراه پیشرفته.

روش	پارامترها(M)	CIFAR-10	CIFAR 100

[18] shuffle-channel

IGCV2∗-C416 (Xie et al. 2018)	0.7	94.51	77.05
SSC-Net-6-9	1.2	95.14	75.99
SSC-Net-3-18	2.1	94.55	77.13
SSC-Net-4-18	2.8	94.95	77.67

فصل پنجم

شبکه کانولوشنی نامتراکم سریع[19]

[19] Fast Sparse ConvNets

مقدمه

تنک‌شـــدگی وزن معمولاً منجر به مدل‌های کوچک‌تر و کارآمدتر از نظر محاســـباتی (از نظر تعداد عملیات‌های ممیز شـــناور) می‌شـــود، اما اغلب به عنوان ابزاری عملی برای تســـریع مدل‌ها نادیده گرفته می‌شود، زیرا این تصور نادرست وجود دارد که عملیات نامتراکم نمی‌تواند به اندازه کافی سریع باشد تا به ســـرعت‌های واقعی در طول اســـتنتاج دســـت یابد. برای پرداختن به این تصـــور اشـــتباه، در این مقاله، هسته‌های سریعی برای ضرب ماتریس نامتراکم - متراکم[20] (SpMM) معرفی می‌شود که به طور خاص شتاب شبکه‌های عصبی نامتراکم را هدف قرار می‌دهد.

۵-۱-پراکندگی شبکه‌ها

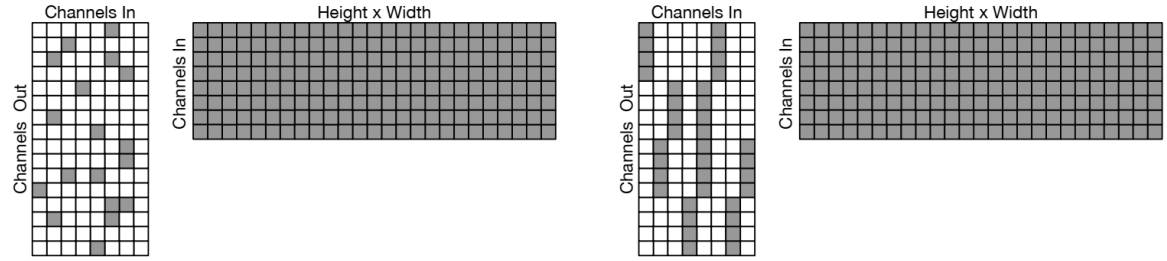
در این مقاله روی مجموعه داده ImageNet با تقویت استاندارد آموزش می‌دهیم و دقت‌های بالای ۱ را در مجموعه اعتبارســـنجی نمونه ارائه شـــده ۵۰ کیلویی گزارش می‌کنیم. برای پراکندگی شـــبکه ها از تکنیک هرس تدریجی [۴۸] استفاده می کنیم.

ما اولین پیچیدگی کامل را در ابتدای هر سه شبکه هرس نمی کنیم. سهم کلی آن در شمارش پارامترها، تعداد FLOP و زمان اجرا اندک اســـت و نیازی به معرفی یک عملیات پراکنده جدید ندارد. در عوض، ما یک هسته کانولوشنال متراکم را پیاده سازی می کنیم که تصویر را در طرح استاندارد HWC به عنوان ورودی می گیرد و طرح CHW مصرف شده توسط عملیات پراکنده در بقیه را خروجی می دهد

[20] Sparse Matrix-Dense Matrix

۵-۲-پیاده‌سازی هسته

نمودارکانولوشن 1×1 به عنوان SpMM در شکل ۵-۱ مشاهده می‌شود. طرح ارائه شده در این مقاله مستلزم این است که تنسورهای فعال‌سازی در قالب CHW ذخیره شوند، برخلاف کتابخانه‌های استنتاج سیار متراکم که خروجی‌های آن‌ها به صورت HWC هستند



شکل ۵-۱: کانولوشن نامتراکم 1×1 به عنوان SpMM . سمت چپ: پراکندگی بدون ساختار (یا اندازه بلوک 1). سمت راست: اندازه بلوک کانال خروجی 4.

سه بینش کلیدی وجود دارد که موجب عملکرد بالای هسته‌ها می‌شود:

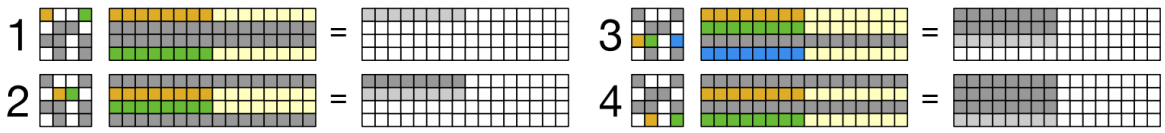
۱. در حالی که ماتریس وزن نامتراکم است، ماتریس فعال‌سازی متراکم باشد. این بدان معنی است که می‌توان بارها بردار را از ماتریس فعال‌سازی اجرا کرد و چندین موقعیت مکانی را به طور همزمان پردازش نمود.

۲. با پردازش ماتریس به ترتیب درست، می‌توان مقادیری را که به صورت تصادفی در حافظه پنهان L1 به آن‌ها دسترسی پیدا می‌کنند، نگه داشت، که دسترسی تصادفی از آن‌ها سریع و در زمان ثابت است.

۳. هنگامی که تعداد کانال‌های ورودی به اندازه کافی کم باشد، واکشی اولیه از فعال سازی‌ها می‌تواند از دست دادن حافظه پنهان را کاهش دهد.

۵-۳-بررسی مفهومی

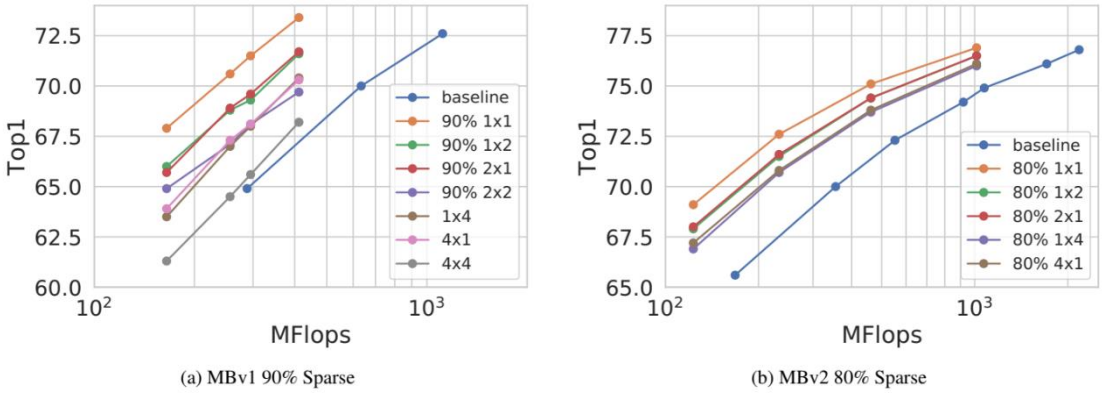
شکل ۵-۲ الگوهای خواندن و نوشتن حافظه در چند مرحله از هسته را نشان می‌دهد. شکل، 8 عنصر را نشان می‌دهد که همزمان برای تجسم پردازش می‌شوند، اما وجود ۱۶ عنصر برای یک اجرای واقعی



شکل ۵-۲: خواندن و نوشتن الگوریتم توسط تجسم حافظه.

طبیعی‌تر اســت زیرا مربوط به یک خط از حافظه اســت. حلقه بیرونی روی ســتون‌ها و حلقه داخلی روی ردیف‌ها قرار دارد. این مســئله اجازه می‌دهد تا هر نوار از ۱۶ موقعیت مکانی در فعال ســازی‌ها در حافظه پنهان L1 باقی بماند تا زمانی که دیگر مورد نیاز نباشد. در شکل ۵-۲ مراحل ۱ و ۲ حافظه نهان را آماده می‌کند، در حالی که مراحل بعدی ۳ و ۴ تمام مقادیر سمت راست را از حافظه نهان L1 بارگیری می‌کند.

علاوه بر بردارســازی در بعد HW، بهره‌گیری از مقادیر اندک ســاختار، در ماتریس وزن می‌تواند با افزایش استفاده مجدد از داده‌ها پس از بارگذاری مقادیر ، افزایش عملکرد قابل توجهی را ارائه دهد. محدود کردن الگوی پراکندگی به طوری‌که چندین کانال خروجی یا ورودی همه یک الگوی صــفر / غیر صــفر را به اشتراک می‌گذارند "بلوک‌ها" را در ماتریس وزن ایجاد می‌کند (سمت راست شکل ۵-۲). بلوک‌ها در بعد کانال خروجی امکان استفاده مجدد از داده‌ها را بیشتر از بلوک‌های بعد کانال ورودی می‌دهند. آزمایش‌ها (شــکل ۵-۳) نشــان می‌دهند که هر کدام از گزینه‌ها تأثیر یکســانی بر دقت دارد، بنابراین در این مقاله، مسدود کردن کانال خروجی با اندازه‌های ۲ و ۴ اجرا می‌شود. نامگذاری برای هسته ها این‌گونه است که عرض برداری مکانی آن‌ها به دنبال اندازه بلوک کانال خروجی ارائه داده می‌شود - 2 × 16 به معنای 16 پیکسل و پردازش 2 کانال خروجی در حلقه داخلی می‌باشد.



(a) MBv1 90% Sparse

(b) MBv2 80% Sparse

شکل ۵-۳: تأثیر اندازه بلوک بر صحت top-1.

فصل ششم

شبکه‌های کانولوشنی نامتراکم کانونی برای تشخیص اشیاء سه بعدی[۲۱]

[21] Focal Sparse Convolutional Networks for 3D Object Detection

مقدمه

داده‌های نامتراکم سه‌بعدی غیریکنواخت، مانند ابرهای نقطه‌ای[۲۲] یا وکسل‌ها در موقعیت‌های فضایی مختلف، به روش‌های متفاوتی به وظیفه تشخیص اشیاء سه‌بعدی کمک می‌کنند. در این مقاله، دو ماژول جدید برای افزایش قابلیت شبکه‌های نامتراکم معرفی می‌شود، که هر دو مبتنی بر یادگیری ویژگی نامتراکم با پیش‌بینی اهمیت موقعیت هستند. در این مقاله برای نخستین بار، نشان داده شده است که پراکندگی فضایی قابل یادگیری در کانولوشن نامتراکم برای تشخیص اشیا سه‌بعدی پیچیده ضروری است.

۶-۱-معرفی

یک چالش کلیدی در تشخیص اشیاء سه‌بعدی، یادگیری بازنمایی‌های مؤثر از داده‌های هندسی سه‌بعدی بدون ساختار و نامتراکم مانند ابرهای نقطه ای است.

صرف نظر از کانولوشن نامتراکم عادی یا زیرخمینه[۲۳]، موقعیت‌های خروجی $\mathrm{P_{out}}$ در تمام $p \in \mathrm{P_{in}}$ ثابت می‌باشد که نامطلوب است. در مقابل، در این مقاله تعیین تطبیقی اندازه‌های نامتراکم یا میدان پذیرنده به روشی دقیق انجام می‌شود. موقعیت های خروجی آزادسازی[۲۴] می‌شوند تا $\mathrm{P_{out}}$ به صورت پویا توسط ویژگی‌های نامتراکم تعیین شود. این فرآیند پیشنهادی در شکل ۶-۱ نشان داده شده است. در فرمول، موقعیت‌های خروجی $\mathrm{P_{out}}$ به اجتماعی از همه موقعیت‌های مهم با ناحیه منبسط و سایر موقعیت‌های بی‌اهمیت تعمیم می‌یابند. نواحی منبسط تغییر شکل پذیر و پویا نسبت به موقعیت های ورودی هستند. معادله زیر حاصل می‌شود:

$$P_{\mathrm{out}} = \left(\bigcup_{p \in P_{\mathrm{im}}} P(p.K^{d}_{\mathrm{im}}(p)) \right) \cup P_{\mathrm{in/im}} \qquad (۱-۶)$$

[22] point clouds

[23] submanifold

[24] relax

این فرآیند به سه مرحله تقسیم می‌شود: (الف) پیش‌بینی اهمیت مکعبی، (ب) انتخاب ورودی مهم، و (ج)تولید شکل خروجی پویا

- پیش‌بینی اهمیت مکعبی:

یک نقشه اهمیت مکعبی I_p شامل اهمیت ویژگی‌های نامزد خروجی در اطراف ویژگی ورودی در موقعیت p است. هر نقشه اهمیت مکعبی همان شکل K^{d} را با وزن هسته کانولوشن پردازشی اصلی دارد، به عنوان مثال، $K^3 = 3 \times 3 \times 3$ با اندازه هسته ۳. این نقشه توسط یک کانولوشن نامتراکم زیر خمینه اضافی با تابع سیگموئید پیش‌بینی می‌شود. مراحل آخر به نقشه‌های اهمیت مکعبی پیش بینی شده بستگی دارد.

- انتخاب ورودی مهم:

$\mathrm{P_{im}}$ زیر مجموعه ای از $\mathrm{P_{in}}$ است. این شامل موقعیت های ویژگی های ورودی نسبتا مهم است. می‌توان $\mathrm{P_{im}}$ را به صورت زیر انتخاب کرد:

$$P_{\mathrm{im}} = \{p \mid I_0^p \geq \tau . p \in P_{\mathrm{in}}\} \qquad (۲-۶)$$

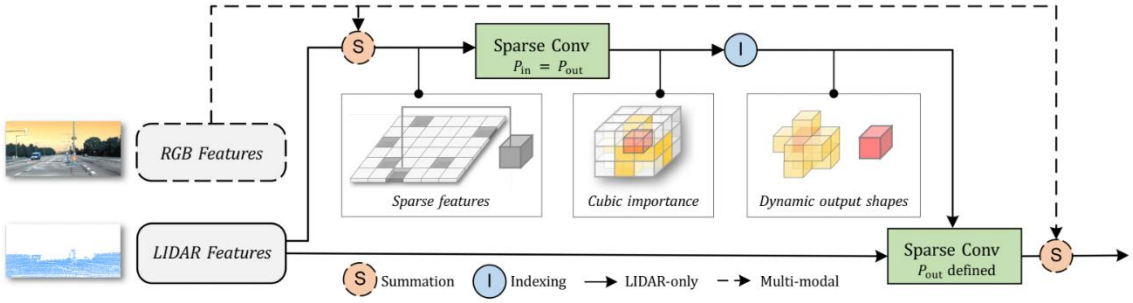
که در آن I_p^0 مرکز نقشه اهمیت مکعبی در موقعیت p است. و τ یک آستانه از پیش تعریف شده است. هنگامی که τ به ترتیب 0 یا ۱ باشد، فرمول ما به کانولوشن نامتراکم عادی یا زیرخمینه تبدیل می‌شود.

- تولید شکل خروجی پویا:

ویژگی‌های $\mathrm{P_{im}}$ به یک شکل پویا منبسط می‌شوند. خروجی اطراف p توسط شکل خروجی پویا $\mathrm{K_{im}^d}$ (p) تعیین می‌شود. توجه داشته باشید که اشکال خروجی قابل تغییر شکل در داخل اتساع اصلی بدون انحرافات هرس می‌شوند. مشابه با فرمول (۶-۲) به صورت زیر محاسبه می‌شود:

$$K_{\mathrm{im}}^d(p) = \{k \mid p + k \in P_{\mathrm{in}} . I_k^p \geq \tau . k \in K^d\}$$

برای ویژگی‌های بی‌اهمیت باقی‌ملنده، موقعیت‌های خروجی آن‌ها به‌عنوان ورودی، یعنی زیرخمینه ثلبت می‌شوند. حذف مستقیم آن‌ها یا استفاده از روشی کاملاً پویا بدون حفظ آن‌ها، روند آموزش را ناپایدار میکند.



شکل ۶-۱: چارچوب کانولوشن نامتراکم کانونی و گسترش چند وجهی آن.

۶-۲-مراحل اجرا

در تشخیص اشیاء سه بعدی، اشیاء پیش‌زمینه اطلاعات ارزشمندتری می‌باشند. بر اساس این موضوع، تابع زیان[۲۵] کانونی به عنوان یک تابع زیان هدف برای نظارت بر پیش‌بینی اهمیت اعمال در نظر گرفته می‌شود. اهداف عینی برای مراکز وکسل‌های ویژگی در جعبه‌های داده مرجع سه بعدی ساخته می‌شود. وزن زیان آن به عنوان 1 برای کلیه ماژول‌ها نگه داشته می‌شود.

نظارت اضافی از ضرب نقشه‌های اهمیت مکعبی پیش‌بینی شده در ویژگی‌های خروجی به عنوان توجه حاصل می‌شود. این امر باعث می‌شود که شاخه پیش‌بینی اهمیت به طور طبیعی متمایز شود. به‌طور تجربی نشان داده شده است که این شیوه توجه برای کلاس‌های کوچک، به عنوان مثال، عابر پیاده و دوچرخه‌سوار در دیتاست KITTI مفید است.

[25] loss

۶-۳-ادغام[۲۶] کاناکا

یک نسخه چند وجهی از کانولوشن نامتراکم کانونی، همان‌طور که در شکل (۶-۱) نشان داده شده‌است، در نظر گرفته می‌شود. در این بخش ویژگی‌های RGB از تصاویر استخراج می‌شوند و ویژگی‌های لایدار[۲۷] با آن‌ها تراز می‌شوند. ویژگی‌های استخراج‌شده با ویژگی‌های نامتراکم ورودی و خروجی مهم در کانولوشن نامتراکم کانونی ترکیب می‌شوند.

۶-۳-۱-استخراج ویژگی

ماژول ادغام سبک وزن است و شامل یک لایه conv-bn-relu و یک لایه max-pooling است. این عملیات وضوح تصویر ورودی را $\frac{1}{4}$ می‌کند. به دنبال آن ۳ لایه conv-bn-relu با اتصال باقی‌ماندگی[۲۸] وجود دارد. سپس شماره کانال کاهش می‌یابد تا با ویژگی‌های نامتراکم، با یک لایه از پرسپترون چند لایه سازگار باشد.

۶-۳-۲-تراز ویژگی

یکی از مشکلات رایج در حین ادغام، ناهماهنگی در نگاشت از فضای سه بعدی به فضای دو بعدی است. داده‌های ابر نقطه ای معمولاً با تبدیل و تقویت پردازش می‌شوند. تبدیلات شامل معکوس کردن، مقیاس مجدد، چرخش و انتقال است. تقویت معمولی نمونه‌برداری از داده مرجع وکپی کردن اشیاء از صحنه‌های دیگر است. برای تبدیل‌های معکوس، مختصات ویژگی‌های نامتراکم را با پارامترهای تبدیل ثبت شده معکوس می‌کنیم. برای نمونه‌برداری از داده مرجع، اشیاء دو بعدی مربوطه را روی تصاویر کپی می‌کنیم. به‌جای استفاده از یک مدل تقسیم‌بندی اضافی، برای ساده‌سازی، اشیاء مستقیماً در جعبه‌های مرزی برش داده می‌شود.

مراحل ادغام.

[26] Fusion

[27] LIDAR

[28] residual

ویژگی‌های RGB تراز شـده مسـتقیماً با ویژگی‌های نامتراکم ترکیب می‌شـوند، زیرا شـماره‌های کانال یکسـانی دارند. ویژگی‌های RGB تراز شـده دو بار در این ماژول با ویژگی‌های نامتراکم ترکیب می‌شـوند. ابتدا با ویژگی‌های ورودی برای پیش‌بینی اهمیت مکعب ترکیب می‌شود. سپس ویژگی‌های RGB فقط با ویژگی‌های نامتراکم خروجی مهم ترکیب می‌شـود. به طور کلی، لایه‌های چند حالته از نظر پارامترها و استراتژی‌های ادغام سبک وزن هستند. آن‌ها به طور مشترک با آشکارسازها آموزش می‌بینند. این روش یک راه‌حل کارآمد و اقتصادی برای ماژول ادغام در تشخیص اشیاء سه بعدی ارائه می‌دهد.

۶-۴-شبکه‌های کاناکا

کاناکا و گسـترش چندوجهی آن می‌توانند به راحتی جایگزین همتایان خود در شـبکه‌های مازه[۲۹] آشـکارسـازهای سـه بعدی شـوند. در طول آموزش، از تنظیمات اولیه یا نرخ یادگیری خاصـی برای ماژول های معرفی شـده اسـتفاده نمی‌شـود. شـاخه پیش‌بینی اهمیت به وسـیله پس انتشـار از طریق تابع ضـرب توجه و از تابع زیان هدف آموزش داده می‌شود. شبکه‌های ماز در آشکارسازهای شی سه بعدی معمولاً از یک لایه سـاقه و چهار مرحله تشـکیل شـده‌اند. هر مرحله، به جز مرحله اول، شـامل یک کانولوشـن منظم نامتراکم با زیرنمونه‌برداری و دو بلوک زیرخمینه اسـت. در مرحله اول، یک یا دو لایه کانولوشـن نامتراکم وجود دارد. به طور پیش فرض، هر کانولوشن نامتراکم با نرمال‌سازی دسته‌ای و فعال‌سازی ReLU دنبال می‌شود.

[29] backbone network

فصل هفتم شبکه‌های عصبی کانولوشنی فضایی-زمانی ۴ بعدی[30]: شبکه‌های عصبی کانولوشنی مینکوفسکی[31]

[30] 4D Spatio-Temporal ConvNets

[31] Minkowski Convolutional Neural Networks

مقدمه

در بسیاری از برنامه‌های روباتیک و VR/AR، ویدیوهای سه بعدی منابع ورودی به‌راحتی در دسترس هستند (توالی از تصاویر عمقی یا اسکن‌های لایدار). با این حال، در بسیاری از موارد، فیلم‌های سه بعدی فریم به فریم یا از طریق شبکه‌های دو بعدی یا الگوریتم‌های ادراک سه بعدی پردازش می‌شوند. در این بخش، شبکه‌های عصبی کانولوشنی ۴ بعدی برای ادراک فضایی-زمانی پیشنهاد می‌شود که می‌تواند مستقیماً چنین ویدئوهای سه بعدی را با استفاده از پیچش‌های با ابعاد بالا پردازش کند.

۷-۱- معرفی

ادراک فضایی-زمانی ۴ بعدی اساساً به ادراک سه بعدی به عنوان برشی از ۴ بعد در امتداد بعد زمانی یک اسکن سه بعدی نیاز دارد. با این حال، در درجه اول بایستی ادراک سه بعدی به ویژه تقسیم‌بندی سه بعدی با استفاده از شبکه‌های عصبی پوشش داده شود.

۷-۲-کانولوشن و تنسور نامتراکم

در داده‌های گفتار، متن یا تصویر سنتی، ویژگی‌ها به طور فشرده استخراج می‌شوند. با این حال، برای اسکن‌های سه بعدی، چنین نمایش متراکمی ناکارآمد است، زیرا بیشتر فضا خالی است. در عوض، می‌توان فضای غیر خالی را تحت عنوان مختصات آن و ویژگی مرتبط ذخیره کرد. این نمایش یک بسط N بعدی از یک ماتریس نامتراکم است. به طور خلاصه، می‌توان مجموعه‌ای از مختصات ۴ بعدی را به صورت $\mathrm{C} = \{(\mathrm{x_i}.\mathrm{y_i}.\mathrm{z_i}.\ \mathrm{t_i})\}_\mathrm{i}$ یا به $\mathcal{F} = \{\boldsymbol{f}_\mathrm{i}\}_\mathrm{i}$ و ویژگی های مرتبط C به عنوان یک ماتریس یا به عنوان ماتریس $\mathcal{F}$ نشان داد. لذا یک تنسور نامتراکم را می توان به صورت زیر نوشت:

$$C = \begin{bmatrix} x_1 & y_1 & z_1 & t_1 & b_1 \\ & & \vdots & & \\ x_N & y_N & z_N & t_N & b_N \end{bmatrix}, F = \begin{bmatrix} \mathbf{f}_1^T \\ \vdots \\ \mathbf{f}_N^T \end{bmatrix} \qquad (۱-۷)$$

که در آن $\mathrm{b_i}$ شاخص‌های دسته‌ای مختصات و $\mathrm{f_i}$ یک بردار است.

۷-۳-کانولوشن نامتراکم تعمیم‌یافته:

کانولوشن نامتراکم تعمیم یافته نه تنها همه کانولوشن‌های نامتراکم بلکه کانولوشن‌های متراکم معمولی را نیز در بر می‌گیرد. فرض کنید $\mathrm{x}_{\mathrm{u}}^{\mathrm{in}} \in \mathbb{R}^{N^{in}}$ یک بردار ویژگی ورودی در یک فضای D بعدی در $\mathrm{u} \in \mathbb{R}^{D}$ (یک مختصات D بعدی باشد)، و وزن هسته کانولوشنی $\mathbf{W} \in \mathrm{R}^{\mathrm{K^D} \times \mathrm{N^{out}} \times \mathrm{N^{in}}}$ باشد. سپس وزن‌ها با ماتریس‌های $\mathrm{K^D}$ به اندازه $\mathrm{N^{out}} \times \mathrm{N^{in}}$ تحت عنوان $\mathrm{W_i}$ برای $|\{\mathrm{i}\}_{\mathrm{i}}| = \mathrm{K^D}$ به وزن‌های فضایی تقسیم می‌شوند. در نهایت، کانولوشن متراکم معمولی در بعد D به صورت زیر است:

$$\mathbf{x}_{\mathbf{u}}^{\mathrm{out}} = \sum_{\mathbf{i} \in \mathcal{V}^D(K)} W_{\mathbf{i}} \mathbf{x}_{\mathbf{u}+\mathbf{i}}^{\mathrm{in}} \text{ for } \mathbf{u} \in \mathbb{Z}^D \qquad (۲-۷)$$

به طوری که $\mathcal{V}^D(K)$ لیستی از انحرافات در ابرمکعب D بعدی است که در مرکز مبدا قرار دارد. به عنوان مثال، $\mathrm{V}^1(3) = \{-1.\ 0.\ 1\}$ کانولوشن نامتراکم تعمیم‌یافته در معادله (۳-۷) تعداد پارامترهای معادله (۲-۷) را کم می‌کند.

$$\mathbf{x}_{\mathbf{u}}^{\mathrm{out}} = \sum_{\mathbf{i} \in \mathcal{N}^D(\mathbf{u}, \mathcal{C}^{\mathrm{in}})} W_{\mathbf{i}} \mathbf{x}_{\mathbf{u}+\mathbf{i}}^{\mathrm{in}} \text{ for } \mathbf{u} \in \mathcal{C}^{\mathrm{out}} \qquad (۳-۷)$$

که در آن $\mathcal{N}^D$ مجموعه‌ای از انحرافات است که شکل یک هسته را تعریف می‌کند و $\mathcal{N}^D(\mathbf{u}, \mathcal{C}^{\mathrm{in}}) = \{\mathbf{i} | \mathbf{u} + \mathbf{i} \in \mathcal{C}^{\mathrm{in}}, \mathbf{i} \in \mathcal{N}^D\}$ به عنوان مجموعه‌ای از انحرافات از مرکز فعلی، u، که در $\mathcal{C}^{\mathrm{in}}$ وجود دارد، شناخته می‌شود. $\mathcal{C}^{\mathrm{in}}$ و $\mathcal{C}^{\mathrm{out}}$ مختصات ورودی و خروجی از پیش تعریف شده تنسورهای نامتراکم هستند. توجه داشته باشید که مختصات ورودی و خروجی لزوماً معادل نیستند. شکل هسته کانولوشن به طور دلخواه با $\mathcal{N}^D$ تعریف می‌شود. این تعمیم بسیاری از موارد خاص مانند هسته‌های کانولوشنی منبسط و هسته‌های ابرمکعبی را در بر می‌گیرد. یکی دیگر از موارد خاص این حالت است که $\mathcal{C}^{\mathrm{in}} = \mathcal{C}^{\mathrm{out}}$ و $\mathcal{N}^D = \mathcal{V}^D(K)$، در این حالت "کانولوشن زیرخمینه نامتراکم" است. اگر $\mathcal{C}^{\mathrm{in}} = \mathcal{C}^{\mathrm{out}} = \mathbb{Z}^D$ و $\mathcal{N}^D = \mathcal{V}^D(K)$، کانولوشن نامتراکم تعمیم‌یافته معادل کانولوشن متراکم است. برای کانولوشن‌های گام به گام، $\mathcal{C}^{\mathrm{in}} \neq \mathcal{C}^{\mathrm{out}}$.

۷-۴-مهندسی مینکوفسکی

۷-۴-۱-کوانتیزاسیون تنسور نامتراکم

اولین مرحله در شبکه عصبی کانولوشنی نامتراکم، پردازش داده‌ها برای تولید یک تنسور نامتراکم است که ورودی را به مختصات منحصر به فرد و ویژگی‌های مرتبط تبدیل می‌کند. در الگوریتم۱، تابع GPU برای این فرآیند فهرست می‌شود. به طور خاص، برای بخش‌بندی معنایی، یک برچسب برای هر جفت ویژگی مختصات ورودی ایجاد می‌کند. اگر بیش از یک برچسب معنایی مختلف در یک وکسل وجود داشته باشد، در طول آموزش با علامت‌گذاری آن با IGNORE_LABEL، این وکسل نادیده گرفته می‌شود. ابتدا، همه مختصات به کلیدهای هش تبدیل می‌شود و جفت برچسب هش-کلید منحصر به فرد برای حذف برخورد پیدا می‌شود. توجه داشته باشید که SortByKey، UniqueByKey، و ReduceByKey همگی توابع استاندارد کتابخانه Thrust هستند. تابع کاهش $<= f\left((l_x . i_x) . (l_y . i_y)\right)$

Algorithm 1 GPU Sparse Tensor Quantization

Inputs: coordinates $C_p \in \mathbb{R}^{N \times D}$, features $F_p \in \mathbb{R}^{N \times N_f}$,
target labels $\mathbf{l} \in \mathbb{Z}_+^N$, quantization step size v_l
$C'_p \leftarrow \text{floor}(C_p \,/\, v_l)$
$\mathbf{k} \leftarrow \text{hash}(C'_p)$, $\mathbf{i} \leftarrow \text{sequence(N)}$
$((\mathbf{i}', \mathbf{l}'), k') \leftarrow \text{SortByKey}((\mathbf{i}, \mathbf{l}), \text{key=}\mathbf{k})$
$(\mathbf{i}'', (\mathbf{k}'', \mathbf{l}'')) \leftarrow \text{UniqueByKey}(\mathbf{i}', \text{key=}(\mathbf{k}', \mathbf{l}'))$
$(\mathbf{l}''', \mathbf{i}''') \leftarrow \text{ReduceByKey}((\mathbf{l}'', \mathbf{i}''), \text{key=}\mathbf{k}'', \text{fn=}f)$
return $C'_p[\mathbf{i}''', :], F_p[\mathbf{i}''', :], \mathbf{l}'''$

(IGNORE_LABEL، i_x) جفت کلیدهای برچسب را می‌گیرد و برچسب نادیده گرفتن را برمی‌گرداند زیرا حداقل دو جفت کلید برچسب در یک کلید به معنای برخورد برچسب است. یک نسخه CPU نیز به طور مشابه کار می‌کند با این تفاوت که تمام کاهش‌ها و مرتب سازی‌ها به صورت سریال پردازش می‌شوند.

مرحله بعدی در خط لوله، تولید مختصات خروجی $\mathcal{C}^{\text{out}}$ با توجه به مختصات ورودی $\mathcal{C}^{\text{in}}$ (معادل ۷-۳) است. هنگامی که در شبکه‌های عصبی معمولی استفاده می‌شود، این فرآیند فقط به اندازه گام لایه کانولوشنی (یا ادغام)، مختصات ورودی، و اندازه گام تنسور نامتراکم ورودی (حداقل فاصله بین مختصات) نیاز دارد. علاوه بر این، همچنین از تنظیم پویا یک مختصات خروجی دلخواه $\mathcal{C}^{\text{out}}$ برای کانولوشن نامتراکم تعمیم‌یافته پشتیبانی می‌شود.

در مرحله بعد، برای ادغام ورودی‌ها با یک هسته، به یک نگاشت نیاز است تا مشخص شود کدام ورودی‌ها بر روی خروجی‌ها تأثیر می‌گذارند. این نگاشت نقشه‌های هسته نامیده می‌شود و آن‌ها به عنوان جفت لیست‌هایی از شاخص‌های ورودی و شاخص‌های خروجی، $\mathrm{M} = \{(\mathrm{I_i}. \mathrm{O_i})\}_i$ برای هر $i \in \mathcal{N}^D$ تعریف می‌شوند. در نهایت، با توجه به مختصات ورودی و خروجی، نقشه هسته، و وزن هسته $\mathrm{W_i}$، می‌توان کانولوشن نامتراکم تعمیم‌یافته را با تکرار در هر یک از انحرافات $i \in \mathcal{N}^D$ (الگوریتم ۲) محاسبه کرد.

Algorithm 2 Generalized Sparse Convolution

Require: Kernel weights $\mathbf{W}$, input features F^i, output feature placeholder F^o, convolution mapping $\mathbf{M}$,

1: $F^o \leftarrow \mathbf{0}$ // set to 0
2: **for all** $W_\mathbf{i}, (I_\mathbf{i}, O_\mathbf{i}) \in (\mathbf{W}, \mathbf{M})$ **do**
3: $\quad F_{\text{tmp}} \leftarrow W_\mathbf{i}[F^i_{I_\mathbf{i}[1]}, F^i_{I_\mathbf{i}[2]}, ..., F^i_{I_\mathbf{i}[n]}]$ // (cu)BLAS
4: $\quad F_{\text{tmp}} \leftarrow F_{\text{tmp}} + [F^o_{O_\mathbf{i}[1]}, F^o_{O_\mathbf{i}[2]}, ..., F^o_{O_\mathbf{i}[n]}]$
5: $\quad [F^o_{O_\mathbf{i}[1]}, F^o_{O_\mathbf{i}[2]}, ..., F^o_{O_\mathbf{i}[n]}] \leftarrow F_{\text{tmp}}$
6: **end for**

به طوریکه I[n] و O[n] به ترتیب عنصر n ام فهرست شاخص‌های I و O و F^i_n و F^o_n نیز به ترتیب بردارهای ورودی و خروجی n ام هستند. کانولوشن نامتراکم تعمیم یافته جابجا شده (دکانولوشن[۳۲]) به طور مشابه کار می‌کند با این تفاوت که نقش مختصات ورودی و خروجی معکوس است.

[32] deconvolution

۷-۴-۲-ادغام ماکسیمم[۳۳]

بر خلاف تنســـورهای متراکم، در تنســـورهای نامتراکم، تعداد ویژگی‌های ورودی در هر خروجی متفاوت است. بنابراین، این یک پیاده‌سازی غیر ضروری برای ادغام ایجاد می‌کند. فرض کنید I و O برداری باشند

Algorithm 3 GPU Sparse Tensor MaxPooling

Input: input feature F, output mapping $\mathbf{O}$
$(\mathbf{I}', \mathbf{O}') \leftarrow$ SortByKey($\mathbf{I}$, key=$\mathbf{O}$)
$\mathbf{S} \leftarrow$ Sequence(length($\mathbf{O}'$))
$\mathbf{S}', \mathbf{O}'' \leftarrow$ ReduceByKey($\mathbf{S}$, key=$\mathbf{O}'$, fn=f)
return MaxPoolKernel($\mathbf{S}'$, $\mathbf{I}'$, $\mathbf{O}''$, F)

که همه $\{\mathrm{I_i}\}_i$ و $\{\mathrm{O_i}\}_i$ را به ترتیب برای $i \in \mathcal{N}^D$ به هم الحاق کرده اســـت. ابتدا تعداد ورودی‌ها در هر مختصات خروجی و شاخص‌های آن ورودی ها را پیدا کنید. الگوریتم ۳ ویژگی‌های ورودی را که به همان مختصـات خروجی نگاشـت می‌شـوند، کاهش می‌دهد. Sequence(n) دنباله ای از اعداد صـحیح از 0 تا $n-1$ و تابع کاهش $f((k_1.\ v_1).(k_2.v_2)) = min(v_1.v_2)$ را تولید می‌کند که حداقل مقدار جفت کلید-ارزش را برمی‌گرداند. MaxPoolKernel یک هسته CUDA سفارشی است که ویژگی‌ها را با استفاده از 'S کاهش می‌دهد که شامل شاخص ابتدایی I و شاخص‌های خروجی مربوطه O است.

[33] Max Pooling

۷-۴-۳-ادغام مجموع، ادغام میانگین و ادغام سراسری

ادغام میانگین و ادغام سراسری، میانگین ویژگی‌های ورودی را برای هر مختصات خروجی محاسبه می‌کند. الگوریتم این کار بدین صورت است:

Algorithm 4 GPU Sparse Tensor AvgPooling

Input: mapping $\mathbf{M} = (\mathbf{I}, \mathbf{O})$, features F, one vector $\mathbf{1}$
S_M = coo2csr(row=$\mathbf{O}$, col=$\mathbf{I}$, val=$\mathbf{1}$)
F' = cusparse_csrmm(S_M, F)
N = cusparse_csrmv(S_M, $\mathbf{1}$)
return F'/N

۷-۵-توابع غیر فضایی

برای توابعی که نیازی به اطلاعات مکانی (مختصات) ندارند، می‌توان توابع را مستقیماً روی ویژگی‌های F اعمال کرد. برای مثال، غیر خطی‌ها به اطلاعات مکانی مانند ReLU نیاز ندارند.

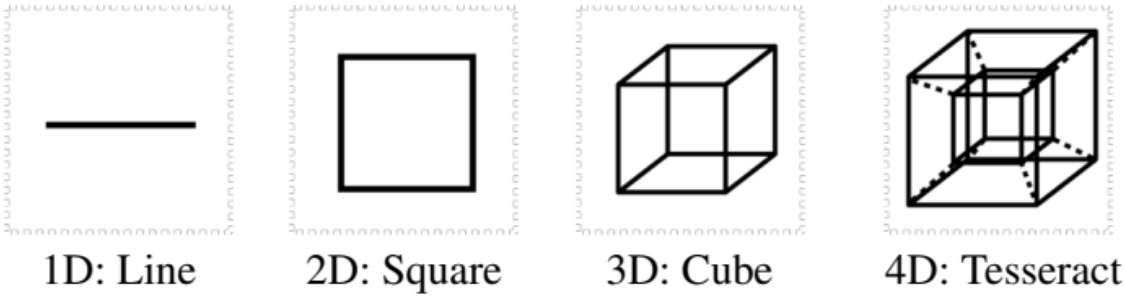
۷-۶- شبکه‌های عصبی کانولوشنی مینکوفسکی

از ویژگی خاصی از کانولوشن نامتراکم تعمیم‌یافته استفاده می‌شود و اشکال هسته غیرمتعارف پیشنهاد می‌گردد که با تعمیم بهتر، حافظه و محاسبات را ذخیره می‌کند. از طرفی جهت سازگاری مکانی-زمانی، یک میدان تصادفی شرطی با ابعاد بالا (در فضای ۷ بعدی فضا-زمان-رنگ) پیشنهاد می‌شود که می‌تواند ثبات را اعمال کند و شبکه پایه و میدان تصادفی شرطی را از مبدا به مقصد[۳۴] آموزش دهد.

[3434] end-to-end

۷-۷-هسته تسراکت[۳۵] و هسته هیبریدی[۳۶]

مساحت سطح داده‌های ۳ بعدی به صورت خطی نسبت به زمان و به صورت درجه دوم نسبت به وضوح فضـــایی افزایش می‌یابد. با این حال، اگر از یک ابرمکعب ۴ بعدی یا یک تســـراکت (شـــکل ۷-۱) برای هســـته‌های کانولوشـــن اســـتفاده شـــود، افزایش تصـــاعدی در تعداد پارامترها به احتمال زیاد منجر به پارامترسازی بیش از حد، برازش بیش از حد، و همچنین هزینه محاسباتی و مصرف حافظه بالا می‌شود. .



شکل ۷-۱: پیش بینی دوبعدی ابرمکعب در ابعاد مختلف.

لذا، یک هسته ترکیبی پیشنهاد می‌شود که از شکل هسته دلخواه کانولوشن نامتراکم تعمیم‌یافته، $\mathcal{N}^D$ استفاده می‌کند.

۷-۸-شبکه‌های باقی‌ماندگی مینکوفسکی[۳۷]

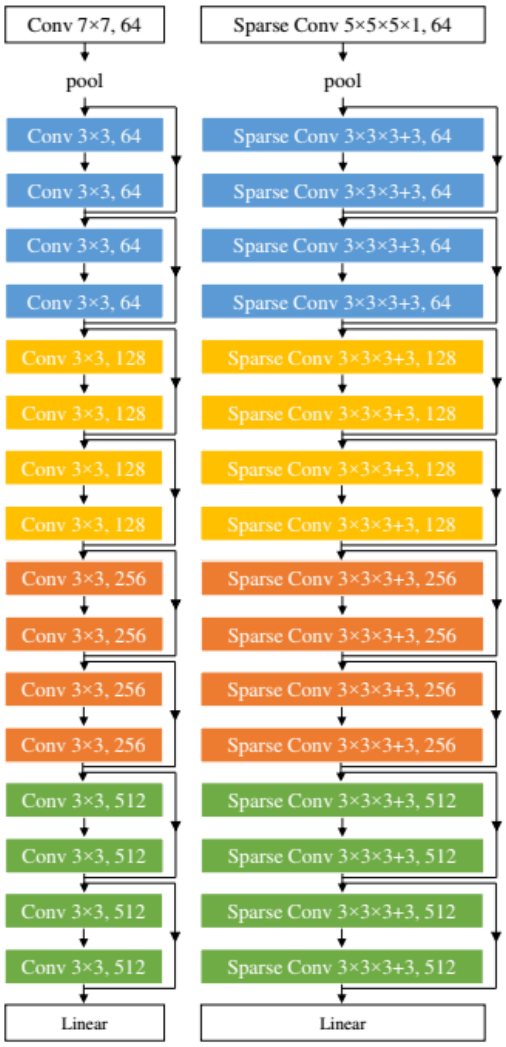
کانولوشن نامتراکم تعمیم‌یافته این اجازه را می‌دهد تا گام‌ها و اشکال هسته به طور دلخواه تعریف شوند.

لذا برای لایه اول، به جای کانولوشن 7×7 دوبعدی از کانولوشن نامتراکم تعمیم‌یافته $5 \times 5 \times 5 \times 1$ استفاده می‌شود. و برای مابقی، از طرح اصلی پیروی می‌شود و نسخه چهار بعدی ResNet18 در شکل ۷-۲ تجسم می‌شود.

[35] Tesseract

[36] Hybrid

[37] Residual Minkowski



شکل ۷-۲: معماری ResNet18(سمت چپ) و) MinkowskiNet18(راست).

۷-۹- CRF- سه‌گانه ثابت

پیش‌بینی‌های شبکه مینکوفسکی برای مراحل زمانی مختلف لزوماً در سراسر محور زمانی سازگار نیستند. برای شفاف‌تر کردن چنین سازگاری و بهبود پیش‌بینی‌ها، یک میدان تصادفی شرطی با یک هسته ثابت تعریف شده در یک فضای سه‌جانبه پیشنهاد می‌شود. فضای سه ضلعی از فضای ۳ بعدی، زمان ۱ بعدی و

فضای رنگی ۳ بعدی تشکیل شده است . این گسترش فضای دوطرفه در پردازش تصویر است. فضای رنگی اجازه می‌دهد تا نقاطی با رنگ‌های مختلف که از نظر فضایی مجاور هستند (مثلاً روی یک مرز) در فضای رنگی از هم دور باشند.

۷-۱۰-تعریف

فرض کنید یک گره[۳۸] CRF در فضای ۷-بعدی (فضا-زمان-کروما) x_i باشد. از اجزای بیرونی دوربین برای تبدیل مختصات فضایی یک گره x_i استفاده می‌شود تا در سیستم مختصات جهانی تعریف شود تا نقاط استاتیک حتی زمانی که ناظر حرکت می‌کند، در همان مختصات باقی بمانند.

برای هر گره x_i پتانسیل یگانه را به صورت $\varphi_u(x_i)$ و پتانسیلجفت به صورت $\varphi_p(x_i . x_j)$ تعریف می‌شود به‌طوری‌که x_j همسایه x_i یعنی $\mathcal{N}^7(x_i)$ است. فیلد تصادفی شرطی نهایی به صورت تعریف می‌شود:

$$P(\mathbf{X}) = \frac{1}{Z}\exp\sum_i \left(\phi_u(x_i) + \sum_{j\in\mathcal{N}^7(x_i)} \phi_p(x_i . x_j) \right) \quad (۴-۷)$$

که در آن Z تابع افراز است. X مجموعه‌ای از تمام گره‌ها است. و ϕ_p بایستی شرط ثلبت بودن $\phi_p(\mathbf{u}.\mathbf{v}) = \phi_p(\mathbf{u}+\tau_u.\mathbf{v}+\tau_v)$ را برای هر $\tau_u.\tau_v \in \mathbb{R}^D$ برآورده کند.

۷-۱۱-استنتاج متغیر

مسئله بهینه‌سازی $\arg\max_X P(X)$ غیرقابل حل است. بنابراین، از استنتاج تغییرات برای به حداقل رساندن واگرایی بین $P(X)$ بهینه و توزیع تقریبی $Q(X)$ استفاده می‌شود. به طور ویژه، از تقریب میدان

[38] node

میانگین $Q = \prod_i Q_i(x_i)$ به عنوان راه حل شکل بسته استفاده می‌شود. Q یک حداکثر محلی است اگر و فقط اگر:

$$Q_i(x_i) = \frac{1}{Z_i} \exp \mathop{\mathbf{E}}_{\mathbf{x}_{-i} \sim Q_{-i}} \left[\phi_u(x_i) + \sum_{j \in \mathcal{N}^7(x_i)} \phi_p(x_i, x_j) \right] \quad (۵-۷)$$

$\mathbf{x}_{-i}$ و Q_{-i} همه گره‌ها یا متغیرها را به جز متغیر i ام نشان می‌دهند. معادله نقطه ثابت نهایی معادله زیر است:

$$Q_i^+(x_i) = \frac{1}{Z_i} \exp \left\{ \phi_u(x_i) + \sum_{j \in \mathcal{N}^7(x_i)} \sum_{x_j} \phi_p(x_i, x_j) Q_j(x_j) \right\} \quad (۶-۷)$$

۷-۱۲- یادگیری با کانولوشن نامتراکم ۷ بعدی

جالب توجه اســـت که جمع وزنی $\phi_p(x_i, x_j) Q_j(x_j)$ در معادله(۷-۶) معادل یک کانولوشـــن نامتراکم تعمیم‌یافته در فضای ۷ بعدی است زیرا ϕ_p ثابت است و لبه‌ها را می‌توان با استفاده از $\mathcal{N}^7$ تعریف کرد. الگوریتم نهایی در الگوریتم ۵ آمده است.

Algorithm 5 Variational Inference of TS-CRF

Require: Input: Logit scores ϕ_u for all x_i; associated coordinate C_i, color F_i, time T_i

$Q^0(X) = \exp \phi_u(X)$, $C_{\text{crf}} = [C, F, T]$

for n from 1 to N **do**

$\tilde{Q}^n = \text{SparseConvolution}((C_{\text{crf}}, Q^{n-1}), \text{kernel}=\phi_p)$

$Q^n = \text{Softmax}(\phi_u + \tilde{Q}^n)$

end for

return Q^N

در نهایت، از ϕ_u به عنوان پیش‌بینی‌های یک شبکه ۴ بعدی مینکوفسکی استفاده می‌شود و هر دو ϕ_u و ϕ_p با استفاده از یک شبکه ۴ بعدی و یک شبکه ۷ بعدی مینکوفسکی با استفاده از معادله زیر آموزش داده می‌شود:

$$\frac{\partial L}{\partial \phi_p} = \sum_n^N \frac{\partial L}{\partial Q^{n+}} \frac{\partial Q^{n+}}{\partial \phi_p} \cdot \frac{\partial L}{\partial \phi_u} = \sum_n^N \frac{\partial L}{\partial Q^{n+}} \frac{\partial Q^{n+}}{\partial \phi_u} \quad (۷-۷)$$

۷-۱۳-نتایج

نتایج تقسیم‌بندی در مجموعه داده ۴ بعدی synthia در جدول ۷-۱ آورده شده است:

جدول ۷-۱: نتایج تقسیم‌بندی در مجموعه داده ۴ بعدی synthia

Method	mIOU	mAcc
3D MinkNet20	76.24	89.31
3D MinkNet20 + TA	77.03	89.20
4D Tesseract MinkNet20	75.34	89.27
4D MinkNet20	77.46	88.013
4D MinkNet20 + TS-CRF	78.30	90.23
4D MinkNet32 + TS-CRF	**78.67**	**90.51**

مقایسه الگوریتم‌های متفاوت نیز در جدول ۷-۲ آورده شده است:

جدول ۷-۲: مقایسه نتایج حاصل از الگوریتم‌های مختلف

Method	mIOU	mAcc
PointNet [23]	41.09	48.98
SparseUNet [9]	41.72	64.62
SegCloud [31]	48.92	57.35
TangentConv [30]	52.8	60.7
3D RNN [33]	53.4	71.3
PointCNN [16]	57.26	63.86
SuperpointGraph [15]	58.04	66.5
MinkowskiNet20	62.60	69.62
MinkowskiNet32	**65.35**	**71.71**

منابع و مراجع

[1] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[2] Zhang, Ting, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. "Interleaved group convolutions." In Proceedings of the IEEE international conference on computer vision, pp. 4373-4382. 2017.

[3] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251-1258. 2017.

[4] Xie, Guotian, Jingdong Wang, Ting Zhang, Jianhuang Lai, Richang Hong, and Guo-Jun Qi. "Interleaved structured sparse convolutional neural networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8847-8856. 2018.

[5] Zhao, Yiren, Xitong Gao, Daniel Bates, Robert Mullins, and Cheng-Zhong Xu. "Focused quantization for sparse cnns." Advances in Neural Information Processing Systems 32 (2019).

[6] Lu, Yao, Guangming Lu, Bob Zhang, Yuanrong Xu, and Jinxing Li. "Super sparse convolutional neural networks." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4440-4447. 2019.

[7] Elsen, Erich, Marat Dukhan, Trevor Gale, and Karen Simonyan. "Fast sparse convnets." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 14629-14638. 2020.

[8] Chen, Yukang, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. "Focal Sparse Convolutional Networks for 3D Object Detection." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5428-5437. 2022.

[9] Choy, Christopher, JunYoung Gwak, and Silvio Savarese. "4d spatio-temporal convnets: Minkowski convolutional neural networks." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3075-3084. 2019.