مقاله پژوهشی

مدل‌سازی اندازه کاشی بهینه برای افزایش استفاده مجدد از داده‌ها در شبکه‌های عصبی کانولوشنی

سوفیا صیدی و مصطفی ارسالی صالحی نسب

چکیده: شبکه‌های عصبی مصنوعی نوعی از مدل‌های محاسباتی هستند که نحوه عملکرد آن‌ها، از شبکه‌های عصبی بیولوژیکی در مغز انسان، الهام گرفته شده است. شبکه‌های عصبی کانولوشنی، نمونه‌ای از این شبکه‌ها هستند که در کاربردهایی مانند کلاسه‌بندی تصویر، تشخیص اشیا، پردازش زبان طبیعی و بهداشت و درمان استفاده می‌شود.

با بزرگ‌تر شدن شبکه عصبی، تعداد پارامترها و حرکت داده بیشتر شده و نیاز به حافظه خارجی نیز، بیشتر می‌شود که همین امر باعث افزایش انرژی مصرفی می‌شود. یکی از راهکارهای اصلی برای کاهش انرژی مصرفی و مراجعات به حافظه خارجی، استفاده حداکثری از داده در هر یک از سطوح حافظه است. استفاده مجدد از داده می‌تواند در سه سطح که در ادامه بیان شده است، انجام شود. ۱- سطح مسیرداده و واحدهای پردازشی ۲- سطح حلقه و زمانبندهای محاسباتی ۳- سطح بین لایه‌ای و شبکه. کاشی‌بندی یکی از تکنیک‌هایی است که برای استفاده مجدد داده در سطح زمانبندها استفاده می‌شود. در این مقاله تعداد استفاده مجدد از داده‌ها را، به صورت یک فرمول ریاضی دقیق مدل می کنیم. سپس در قالب یک مساله بهینه‌سازی، پارامترهای بهینه را باهدف بیشترین استفاده مجدد از داده، برای هر پیکربندی از شبکه، به دست می آوریم. هم‌چنین رابطه بین پارامترهای ساختاری شبکه مانند اندازه کرنل و گام را با اندازه کاشی بررسی می کنیم که باتوجه به بررسی انجام شده، اندازه کاشی بهینه در ۷۰٪ لایه‌های شبکه، از ٤ برابر اندازه کرنل، کوچکتر است.

کلیدواژه: **شبکه‌های عصبی کانولوشنی، انرژی مصرفی، حافظه خارجی، استفاده مجدد از داده، کاشی‌بندی.**

۱- مقدمه

هوش مصنوعی تکنیکی است که به یک کامپیوتر یا ماشین امکان تقلید رفتار، عملکرد یا اعمال انسان را می‌دهد [۱]. حوزه‌های مختلفی مانند یادگیری عمیق و یادگیری ماشین، زیرمجموعه هوش مصنوعی هستند. در حالت کلی یادگیری عمیق زیرمجموعه‌ای از یادگیری ماشین و یادگیری ماشین نیز، زیرمجموعه‌ای از هوش مصنوعی است [۲]. یادگیری ماشین و یادگیری عمیق به عنوان ابزاری قوی، در تشخیص صدا، پردازش زبان طبیعی، تشخیص‌های پزشکی و بسیاری از حوزه‌های دیگر کاربرد دارند. در هر دو تکنولوژی، یادگیری از داده با استفاده از الگوریتم‌ها انجام می‌شود. با این تفاوت که یادگیری ماشین از نظریه‌های آماری برای یادگیری از داده استفاده می‌کند ولی یادگیری عمیق از شبکه‌های عصبی برای یادگیری، استفاده می‌کند.

شبکه‌های عصبی کانولوشنی[1] (CNN) یکی از مدل‌های یادگیری عمیق هستند که معمولا در پردازش داده‌های ساختاریافته، مانند تصویر، استفاده می‌شوند. به دلیل دقت بالایی که دارند از این شبکه‌ها در کاربردهای وسیع و حوزه‌های مختلفی استفاده می‌شود. کاربردهای پردازش تصویر، پردازش گفتار، کلاسه‌بندی تصویر در حوزه‌های صنعت، بهداشت و درمان نمونه‌ای از این حوزه‌ها هستند [۳].

شبکه‌های CNN ساختار لایه به لایه دارند و حدود ۹۰٪ از محاسبات آن‌ها، کانولوشن است [۴]. هرچه تعداد لایه‌های این شبکه‌ها بیشتر شود و پارامترهای آن‌ها، افزایش یابد، دقت آن‌ها بیشتر می‌شود. از طرفی برای اینکه بتوان از این شبکه‌های در کاربردهای بیشتر و پیچیده‌تر استفاده کرد، نیاز است که دقت آن‌ها بیشتر شود.

هرچه شبکه عمیق‌تر و تعداد لایه‌های آن بیشتر شود، به تبع آن تعداد پارامترهای یادگیری نیز بیشتر می‌شود. افزایش پارامترهای شبکه باعث افزایش نیاز به حافظه می‌گردد و درنتیجه ممکن است که حافظه داخلی برای ذخیره داده‌ها کافی نباشد و در نتیجه نیاز به حافظه خارجی افزایش می‌یابد. از طرفی افزایش اندازه حافظه خارجی موجب افزایش حرکت داده بین واحدهای محاسباتی و حافظه می‌شود. تحقیقات نشان داده است که انرژی مصرفی و زمان دسترسی به حافظه خارجی DRAM به ترتیب ۲۰۰ و ۱۰ برابر حافظه داخلی است. بنابراین کاهش استفاده از حافظه خارجیDRAM و انتقالات به آن و حرکت داده، می‌تواند بهبود انرژی مصرفی و کارایی را در پیش داشته باشد [۵].

استفاده مجدد و حداکثری از داده در کلیه سطوح سلسله مراتب حافظه، یکی از راهکارهای مهم برای کاهش انرژی مصرفی و استفاده از حافظه خارجی است. در شبکه CNN انواع داده مانند وزن‌ها، ورودی، داده‌های موقت بین لایه‌ای و داده‌های خروجی، وجود دارد [۶]. باتوجه به اینکه در کدام سطح از حافظه از داده استفاده مجدد شود و هم‌چنین باتوجه به انتخاب نوع داده برای استفاده مجدد، ممکن است حالت‌های مختلفی ایجاد شود که تحقیقات مختلفی در همین راستا انجام شده است.

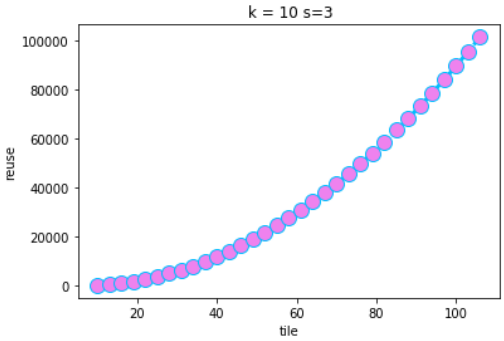
در یک گروه از تحقیقات برای استفاده مجدد داده در سطح واحدهای پردازشی، گروهی از وزن‌ها و داده‌های میانی تولید شده، در رجیسترهای مربوط به واحدهای پردازشی، ذخیره می‌شوند و از آن‌ها نهایت استفاده می‌شود [۷] تا [۹]. با استفاده از این روش، به دلیل اینکه داده‌های پرتکرار در حافظه رجیستر ذخیره شده‌اند، نیاز به حافظه خارجی و هم‌چنین حرکت داده، کاهش می‌یابد.

این مقاله در تاریخ ۵ آذر ماه ۱۴۰۳ دریافت و در تاریخ ۱۶ شهریور ماه ۱۴۰۴ بازنگری شد.

سوفیا صیدی، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران، (email: sakinehseydi@ut.ac.ir).

مصطفی ارسالی صالحی نسب (نویسنده مسئول)، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران، (email:mersali@ut.ac.ir).

1. Convolutional Neural Network



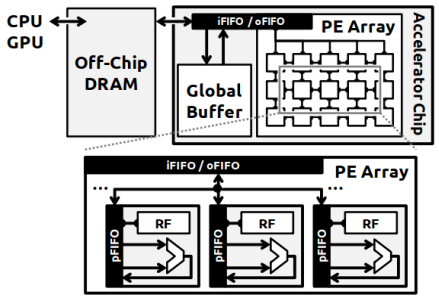
شکل ۱: نمودار تعداد استفاده مجدد از داده ورودی بر حسب اندازه کاشی.

در حالت عادی محاسبات شبکه‌های CNN به صورت لایه به لایه انجام می‌شود، در همین جهت داده‌های میانی که تولید می‌شوند، تا زمان اتمام محاسبات یک لایه، باید در یک حافظه ذخیره شوند. به دلیل اینکه حجم داده‌های میانی تولید شده، زیاد است، ممکن است در حافظه داخلی که سریع هستند، جا نشوند و از حافظه خارجی برای ذخیره آن‌ها استفاده شود. در همین جهت، در گروه دیگری از تحقیقات [۱۰] و [۱۱]، با تغییر ترتیب داده‌های ورودی و ادغام کردن محاسبات چند لایه باهم، حداکثر استفاده مجدد از داده‌های یک پنجره را در طول چند لایه داشته‌اند و به دلیل اینکه محاسبات در آن‌ها به صورت به ترتیب و لایه به لایه، انجام نمی‌شود، نیاز به حافظه برای داده‌های میانی تولید شده بین لایه‌ها را کاهش داده‌اند.

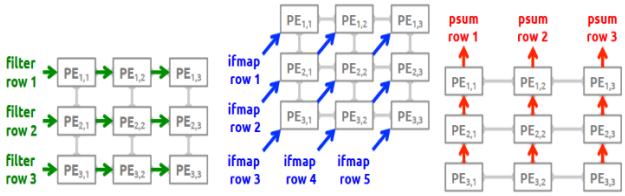
باتوجه به اینکه پایه محاسبات لایه‌های CNN، کانولوشن است و محاسبات هرلایه از ۷ حلقه تو در تو، تشکیل شده است. در نتیجه محاسبات آن، تکرارشونده است و با توجه به ترتیب حلقه‌ها ممکن است، محاسبات و ترتیب داده‌ها تغییر کنند که در گروهی از تحقیقات با تمرکز بر این موضوع، زمانبند و ترتیب مناسب حلقه برای محاسبات یک لایه از CNN را به‌دست آورده‌اند [۱۲]. در [۱۳]، با کنترل جریان داده، چارچوبی ارائه شده است که زمانبندی بهینه برای محاسبات را ارائه می‌کند. در [۱۴] محاسبات الگوریتم CNN با هدف بیشترین استفاده مجدد داده، زمابندی می‌شود که در آن تعداد دسترسی‌های حافظه خارجی به صورت فرمول ریاضی مدل شده است. ولی در آن مدل از اندازه کاشی[1] بهینه و ساختار مناسب شبکه صحبتی نشده است و همچنین فرمول برای استفاده مجدد از داده باتوجه به پیکربندی شبکه، پیشنهاد نکرده‌اند.

در [۱۵] و [۱۶]، اشاره شده است که از داده‌های ورودی به اندازه حاصل‌ضرب ابعاد کرنل وزن، استفاده مجدد می‌شود و از آن به عنوان استفاده مجدد کانولوشنی یاد کرده‌اند. در حالیکه اندازه دقیق استفاده مجدد از داده در مرزهای تصویر عددی کمتر است. همچنین به باتوجه به نتایج شبیه سازی انجام شده در گروه پژوهشی نویسندگان مقاله حاضر که در نمودار شکل ۱ نشان داده شده، اندازه کاشی می‌تواند در تعداد استفاده مجدد از داده تاثیر داشته باشد. در این شکل به عنوان نمونه برای یک لایه CNN با اندازه کرنل ۱۰ و گام برابر ۳، برای اندازه کاشی‌های مختلف، نتایج گزارش شده است. در همین جهت، در این مقاله، تعداد دقیق استفاده مجدد از داده ورودی با توجه به ساختار شبکه و پارامترهای کاشی‌بندی به صورت ریاضی، مدل شده است و همچنین در ادامه با استفاده از فرمول ریاضی در مساله بهینه‌سازی، ساختار بهینه شبکه و پارامترهای کاشی‌بندی به دست می‌آید.

1. Tile



شکل ۲: سلسله مراتب حافظه در ساختار شتابدهنده CNN [۷].



شکل ۳: مسیر داده در Eyeriss [۷].

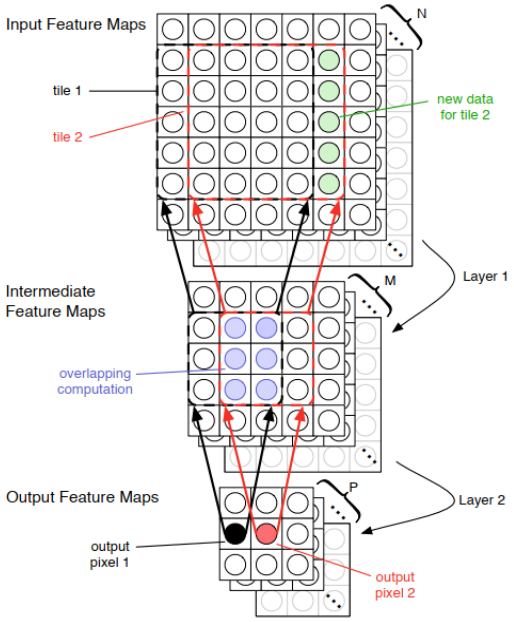
۲- کارهای پیشین

در شکل ۲، ساختار یک شتابدهنده سخت‌افزاری به صورت بلوکی آورده شده است. در حالت کلی و با توجه به شکل ۲، سه سطح از حافظه به صورت حافظه خارج تراشه (off-chip DRAM)، حافظه داخلی (Global buffer) SRAM و رجیستر فایل واحدهای پردازشی(RF)، موجود است و می‌توان گفت که مدل حافظه شتابدهنده‌های سخت‌افزاری شبکه‌های CNN به صورت سلسله مراتبی است.

برای کاهش انرژی مصرفی و استفاده مجدد از داده، باتوجه به اینکه در کدام یک از سطوح حافظه از داده مجدد استفاده می‌شود و نوع داده‌ی استفاده شده، حالت‌های مختلف ایجاد می‌شود. در تحقیقات انجام شده در [۱۷] تا [۱۹] فیلترهای وزن و یا خروجی‌های میانی تولید شده در رجیسترها به صورت ثابت باقی مانده و از آن‌ها به صورت حداکثر استفاده می‌شود. به طوری که این داده‌ها تا زمان انجام شدن کل عملیاتی که نیاز به این فیلترها دارند، در رجیسترها باقی می مانند و با استفاده مجدد از داده‌های پرتکرار در سطح رجیستر، نیاز به دسترسی به حافظه و حرکت داده را کاهش داده‌اند.

در [۱۲] مسیر داده‌ی Row Stationary پیشنهاد شده است که با استفاده از آن می‌توان استفاده مجدد داده برای انواع داده ورودی، وزن و خروجی‌های میانی را توامان داشت. با توجه به شکل ۳ در این مسیرداده، وزن‌ها از قبل در حافظه‌های واحد پردازشی (filter row) بارگیری و ذخیره می‌شوند و ورودی‌ها نیز به صورت مورب وارد می شوند (ifmap row) و داده‌های خروجی نیز (psum row) در حافظه‌های تعبیه‌شده درون تراشه ذخیره می‌شوند. بنابراین در این مسیر داده از داده‌های ورودی و وزن‌ها استفاده مجدد شده و همچنین استفاده مجدد داده را در سطح رجیستر و حافظه داخلی اعمال می‌کنند.

در [۱۳] برای کاهش حرکت داده و افزایش استفاده مجدد داده، داده-های میانی تولید شده، در پایین‌ترین و سریع‌ترین سطح حافظه، یعنی در رجیسترها ذخیره می‌شوند تا هر چندبار که به آن‌ها نیاز است، استفاده شوند. همچنین داده‌های ورودی نیز در ابتدا به صورت بلوک‌های سطح SRAM پارتیشن می‌شوند، بدین ترتیب از داده‌های ورودی در سطح حافظه میانی استفاده مجدد می‌شود.



شکل ۴: مثالی از ادغام دو لایه [۱۱].

در دسته‌ی دیگری از تحقیقات [۱۰]، [۱۱] و [۲۰]، با ادغام چندلایه با هم و در نظر گرفتن آن‌ها مانند یک لایه، از پنجره‌ای از داده ورودی که در حافظه میانی است، به طور کامل استفاده مجدد می‌شود که بدین ترتیب حرکت داده و نیاز به دسترسی به حافظه کاهش می یابد. در حالت عادی، محاسبات لایه‌های شبکه CNN در شتابدهنده‌ها به ترتیب انجام می‌شود، یعنی ابتدا محاسبات یک لایه تمام می شود و سپس محاسبات لایه بعدی شروع می‌شود. برای همین داده‌های میانی تولیدشده و مورد نیاز در محاسبات لایه بعدی، در یک حافظه میانی ذخیره می شوند و معمولا حجم این داده‌ها زیاد است و به حافظه خارجی برای ذخیره آن‌ها نیاز است. در شکل ۴ مثالی از ادغام برای محاسبات دولایه نشان داده شده است. برای انجام محاسبات در ادغام لایه‌ها، ابتدا پنجره‌هایی از هر لایه که در تولید هر پیکسل از لایه آخر، نقش داشته‌اند، مشخص می‌شود. همین موضوع در شکل ۴ نیز مشاهده می‌شود که برای پیکسل مشکی و قرمز از آخرین لایه از ادغام، پنجره‌هایی که در تولید آن‌ها نقش داشته‌اند، در هر لایه مشخص می‌شود که به آن اصطلاحاً هرم گفته می‌شود. بعد از اینکه در اولین لایه پنجره‌ها مشخص شدند، با تغییر ترتیب پنجره‌ها در لایه اول، محاسبات یک پیکسل از خروجی تا لایه آخر انجام می‌شود و دیگر نیاز به حافظه برای ذخیره داده‌های میانی نیست.

در [۱۴]، سعی شده است که در کلیه سطوح حافظه، استفاده مجدد از داده حداکثر شود. برای این منظور تعداد دسترسی‌های به حافظه را به صورت یک فرمول ریاضی مدل می کنند و سپس با استفاده از فرمول ریاضی، ترتیبی از حلقه‌ها و محاسبات را استخراج می‌کنند که تعداد دسترسی‌ها به حافظه حداقل شود. همانگونه که در شکل۵ نشان داده شده است، عملیات یک لایه از شبکه CNN از حلقه‌های تودرتو تشکیل شده است. در [۱۴]، در ابتدا با توجه به تاثیر هر یک از حلقه‌ها در استفاده مجدد داده، حلقه‌ها تقسیم‌بندی می شوند و سپس الگوهایی از استفاده مجدد از داده را باتوجه به ترتیب حلقه‌ها استخراج می کنند و در نهایت با در نظر گرفتن ترتیب‌های مختلفی از حلقه‌ها و نحوه انجام محاسبات ، تعداد دسترسی‌ها به حافظه را به دست می‌آورند.

اما مقاله [۱۴]، فرمول دقیق ریاضی برحسب اندازه کاشی یا بلوک، پیشنهاد نکرده است و همچنین ارتباط بین استفاده مجدد از داده و پارامترهای پیکربندی یک لایه از شبکه نیز، مطرح نشده است.

```
Algorithm loops
f: for (int f = 0; f < F; f ++)
  k: for (int k = 0; k < K; k ++)
    c: for (int c = 0; c < C; c ++)
      x: for (int x = 0; x < X; x ++)
        y: for (int y = 0; y < Y; y ++)
          m: for (int m = 0; m < M; m ++)
            n: for (int n = 0; n < N; n ++)
```

شکل ۵: محاسبات یک لایه از CNN با حلقه‌های تودرتو [۱۴].

در [۱۶]، اندازه کاشی را محدود به سطح حلقه‌های لایه درنظر گرفته‌اند و سپس از داده‌های درون کاشی نهایت استفاده را کرده‌اند. اما در مدل پیشنهادی ما، اندازه کاشی فقط محدود به سطح حلقه‌ها نیست و می‌تواند هر مقداری داشته باشد، از طرفی ما با نگهداری داده‌ها درکاشی برای حلقه‌های بعدی، استفاده مجدد بیشتری از داده‌ها داریم که با افزایش آن تعداد دسترسی به حافظه خارجی کاهش می‌یابد.

۳- انگیزه و مفاهیم اولیه

در این بخش، توضیحاتی در رابطه با انگیزه پژوهش و همچنین مفاهیم اولیه، شرح داده شده است.

۳-۱ انگیزه و مفاهیم اولیه

برای افزایش دقت شبکه‌های CNN و استفاده از آن‌ها در کاربردهای پیچیده، نیاز است که تعداد لایه‌های آن‌ها بیشتر شده و شبکه عمیق‌تر شود. زمانی که تعداد لایه‌ها بیشتر می‌شود، حجم پارامترها و محاسبات نیز افزایش می‌یابد در نتیجه حرکت داده و نیاز به حافظه نیز بیشتر می‌شود. معمولا حجم حافظه داخلی محدود است و برای ذخیره داده‌های خروجی و میانی تولید شده نیاز به حافظه خارجی است. زمان دسترسی و انرژی مصرفی به حافظه خارجی بسیار بیشتر از حافظه داخلی است و روشی که بتواند مراجعات به حافظه خارجی را کاهش دهد، می‌تواند تاثیر زیادی در بهبود انرژی مصرفی داشته باشد.

یکی از راهکارهای اصلی برای کاهش انرژی مصرفی، استفاده مجدد از داده در حافظه‌های داخلی و کم‌توان است. کاشی‌بندی، تکنیکی است که با استفاده از آن می‌توان با بهره‌گیری از اصل محلیت، استفاده مجدد از داده در سطح حلقه و زمانبندی را داشت که در این مقاله، یک فرمول ریاضی دقیق برای تعداد استفاده مجدد داده بر حسب اندازه کاشی و پارامترهای شبکه پیشنهاد شده است. نوآوری‌های مقاله ما، به شرح زیر است:

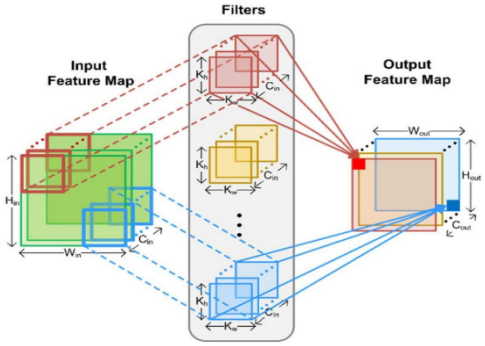
- ارائه یک فرمول ریاضی دقیق برای مدل کردن تعداد استفاده مجدد از داده بر حسب اندازه کاشی و پارامترهای ساختاری یک لایه از شبکه.
- استفاده از فرمول پیشنهادی برای یافتن اندازه کاشی بهینه با هدف افزایش بیشترین تعداد استفاده مجدد از داده.
- محدود کردن سایز کاشی‌های مجاز و کاهش فضای حالت.

۳-۲ مفاهیم اولیه

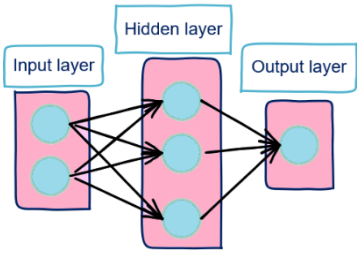
در این قسمت توضیحاتی در رابطه با مفاهیم لازم و اولیه آورده شده است.

۳-۲-۱ شبکه‌های عصبی CNN

نمونه‌ای از شبکه‌های عصبی مهم هستند که به وفور در کاربردهایی که ورودی آن‌ها ساختاریافته است، استفاده می‌شوند. از جمله کاربردهای



شکل ۶: ساختار یک لایه کانولوشنی [۲۱].

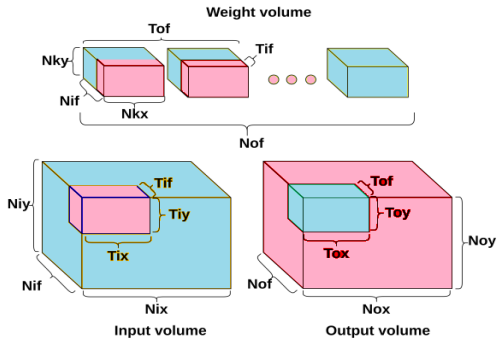


شکل ۷: سه لایه تماماً متصل.

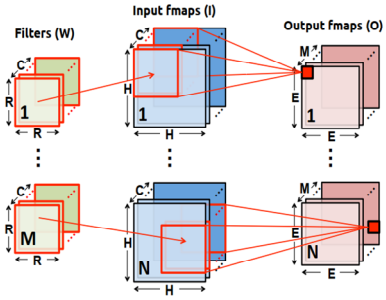
آن می‌توان، پردازش تصویر، پردازش زبان طبیعی، سنسورهای هوشمند و پزشکی را نام برد. شبکه‌های CNN ساختار لایه‌ای دارند و عملیات اصلی آن‌ها کانولوشن است.

در این شبکه‌ها ابتدا در لایه‌های کانولوشنی، ویژگی‌های مهم از ورودی مانند بافت و لبه استخراج می‌شود و سپس در لایه‌های تماما متصل کلاسه‌بندی بر اساس ویژگی‌ها انجام می‌شود. در ادامه توضیحاتی در رابطه با اجزای این شبکه آورده شده است.

- لایه‌های کانولوشنی: اجزا اصلی شبکه‌های CNN لایه‌های کانولوشنی هستند. باتوجه به شکل۶، در این لایه‌ها نقشه‌های ویژگی خروجی با کانوالو شدن فیلترهای وزن در تصویر ورودی ایجاد می‌شوند که در نهایت یکسری ویژگی‌ها از روی تصویر استخراج می‌شود. باید توجه داشت که هر یک از فیلترهای وزن که پارامترهای آن‌ها قابل یادگیری است، ویژگی خاصی از تصویر را استخراج می‌کنند و یک صفحه دو بعدی در خروجی ایجاد می‌کنند.
- لایه‌های جمع‌آوری[1]: یکی از لایه‌های اصلی است که هدف آن کاهش ابعاد نقشه ویژگی است. درواقع در این لایه‌ها با حفظ اطلاعات مهم از نقشه ویژگی، محاسبات و ابعاد کاهش می‌یابد. هم‌چنین با استفاده از این لایه‌ها می‌توان از پیش‌برازش[2] نیز جلوگیری نمود.
- لایه‌های تماماً متصل[3]: با توجه به شکل ۷، لایه‌هایی هستند که در آن‌ها، نورون‌های یک لایه به همه نورون‌های لایه بعدی و قبلی متصل هستند. در شبکه CNN، پس از اینکه با استفاده از لایه‌های کانولوشنی و جمع‌آوری، ویژگی‌های مهم استخراج شدند و توسط شبکه آموزش دیدند، با استفاده از لایه‌های تماماً متصل کلاسه بندی و تقسیم‌بندی می‌شوند.



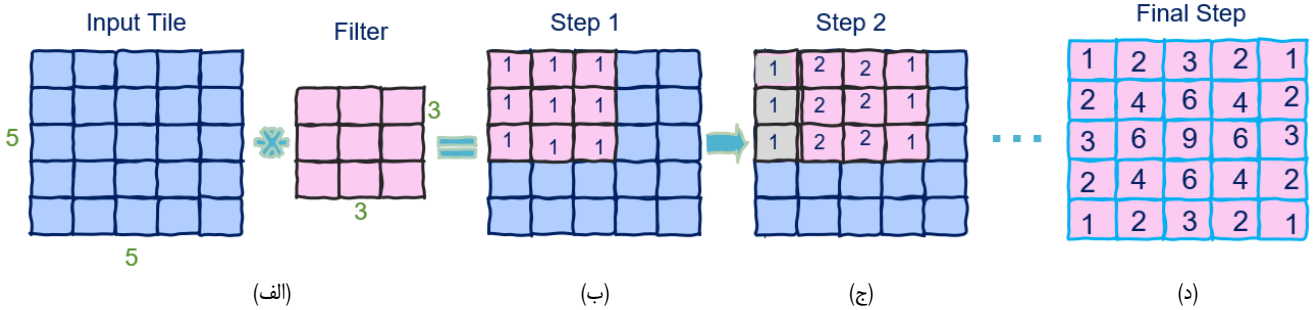
شکل ۸: شیوه کاشی‌بندی [۱۰].



شکل ۹: محاسبات یک لایه کانولوشنی [۷].

- حاشیه‌گذاری[4]: شیوه‌ای است که برای کنترل ابعاد نقشه‌های ویژگی استفاده می‌شود و به معنای اضافه کردن پیکسل‌های اضافی در اطراف مرزهای تصویر ورودی است و قبل از عملیات کانولوشن به تصویر اضافه می‌شود. با استفاده از این تکنیک می‌توان اطلاعات مهم و ابعاد خروجی را حفظ و کنترل نمود.
- کاشی‌بندی[5]: تکنیکی است که برای بهینه‌سازی استفاده از حافظه و کارایی استفاده می‌شود. با توجه به شکل ۸، در این تکنیک داده بزرگ و حجیم به بلوک‌های کوچکی پارتیشن می‌شود که به این بلوک‌ها کاشی گفته می‌شود و پردازش‌ها شکسته شده و به صورت کاشی به کاشی انجام می‌شود. با استفاده از این تکنیک، به دلیل اینکه داده‌هایی که نزدیک و همسایه هم هستند، بارگیری شده‌اند، می‌توان از اصل محلیت مکانی بهره برد. هم‌چنین داده‌ها تا زمانی که از آن‌ها نهایت استفاده شود، در کاشی باقی می‌مانند در نتیجه از اصل محلیت زمانی نیز می‌توان بهره برد.
- استفاده مجدد از داده[6]: ساختار شتابدهنده‌های شبکه‌های عصبی CNN معمولا از سه سطح کلی حافظه، رجیسترفایل، حافظه داخلی SRAM و حافظه خارجی DRAM تشکیل شده است. بهتر است که داده‌هایی که استفاده از آن‌ها پرتکرار است در حافظه‌های سریع‌تر و با انرژی مصرفی کمتر، ذخیره شوند و از آن‌ها به صورت حداکثری استفاده مجدد شود تا نیاز به حافظه خارجی کند و پرانرژی کاهش یابد. با توجه به اینکه در کدام یک از سطوح حافظه استفاده مجدد شود و اینکه از کدام نوع از داده، استفاده مجدد می‌شود، حالت‌های متفاوتی رخ می‌دهد.باتوجه به شکل ۵، محاسبات لایه‌ی کانولوشن به صورت حلقه‌های تو در تو است و در نتیجه به دلیل خاصیت تکراری بودن عملیات، از داده‌های وزن و وروی به صورت متناوب استفاده می‌شود.

1. Pooling
2. Overfitting
3. Fully Connected
4. Padding
5. Tiling
6. Data Reuse



شکل ۱۰: عملیات کانولوشن یک کاشی و تعداد استفاده از هر المان از کاشی.

- در شکل ۹، محاسبات یک لایه کانولوشنی آورده شده است. اگر ابعاد فیلتر وزن R×R در نظر گرفته شود، از هر کدام از پیکسل‌های ورودی تقریباً R×R مرتبه، استفاده می‌شود. هم‌چنین با توجه به اینکه برای تولید هر یک از پیکسل‌های خروجی، کل فیلترهای وزن تاثیر داشته‌اند، از هر فیلتر وزن به اندازه E×E مرتبه استفاده می‌شود. از طرفی از هر فیلتر وزن به اندازه N کانال ورودی، دو مرتبه استفاده می‌شود. هم‌چنین اگر M فیلتر در یک لایه باشد، به تعداد M بار از هر پیکسل ورودی، دوباره استفاده می‌شود.

٤- مدل ریاضی استفاده مجدد داده و بهینه‌سازی

در این بخش توضیحاتی در رابطه با مدل ریاضی آورده شده است سپس اطلاعاتی درباره بهینه‌سازی مساله برای داشتن حداکثر استفاده مجدد از داده آورده شده است.

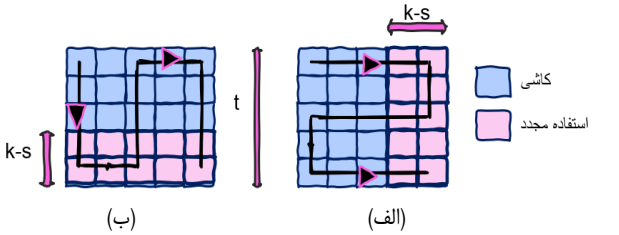
٤-١ مدل ریاضی پیشنهادی برای محاسبه تعداد استفاده مجدد از داده

در شکل ۱۰، یک لایه دوبعدی از شبکه CNN رسم شده است. در این لایه ابعاد کاشی ورودی ۵×۵، اندازه گام ۱ و اندازه فیلتر وزن برابر با ۳×۳ می‌باشد. برای اینکه مشخص شود که از پیکسل‌های داخل یک کاشی چند بار استفاده می‌شود، باید روال اجرای محاسبات لایه، دنبال شود و تعداد استفاده از المان‌ها محاسبه شود. در مرحله اول، کرنل وزن با یک پنجره ۳ در ۳ از ورودی کانوالو می‌شود و از هر کدام از المان‌های پنجره ورودی به اندازه یک بار استفاده می‌شود و در حالت (ج) که مرحله بعدیست، کرنل به اندازه گام روی تصویر ورودی شیفت داشته و پیکسل-هایی از ورودی که در مرحله قبل نیز از آن‌ها استفاده شد، دوباره استفاده می‌شوند. بدین ترتیب بعد از تمام شدن کل عملیات کانولوشن کرنل با کاشی، در حالت (د) مشخص شده است که تعداد استفاده از هریک از پیکسل‌های ورودی چقدر است. تعداد کل استفاده از پیکسل‌ها، در این مثال خاص، حاصل جمع اعداد نوشته شده در جایگاه هر پیکسل در (د) است.

در حالت کلی، اگر برای یک لایه اندازه کاشی برابر $t\times t$ باشد و اندازه کرنل وزن برابر $k\times k$ و گام برابر s باشد. تعداد پیکسل‌های خروجی که داده‌های درون کاشی تولید می‌شوند از (۱)، به دست می‌آید:

$$No_t = \frac{t-k}{s} - ۱ \qquad (۱)$$

برای هرکدام از پیکسل‌های خروجی تعداد $k\times k$ المان ورودی استفاده شده است، بنابراین تعداد کل استفاده از پیکسل‌های یک کاشی از (۲) به دست می‌آید:



شکل ۱۱: استفاده مجدد از داده‌های کاشی.

$$Use_T = No_t.No_t.k.k = \left(\frac{t-k}{s} - ۱\right)^۲ .k^۲ \qquad (۲)$$

در حالت کلی دو نوع از استفاده مجدد از داده با استفاده از کاشی‌بندی وجود دارد که در ادامه آورده شده است:

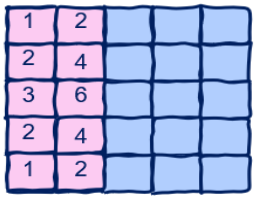
- استفاده از داده‌ها بعد از اولین بارگیری از حافظه : زمانی که داده‌ها از حافظه بارگیری می‌شوند و داخل کاشی قرار می‌گیرند، باتوجه به شکل ۱۰- ج، ممکن است که از هر پیکسل چندین بار استفاده شود. کل تعداد استفاده از داده‌های داخل کاشی از (۲)، به دست می‌آید. اما از طرفی به دلیل اینکه هدف اصلی محاسبه تعداد استفاده مجدد است، پس به اندازه $t\times t$ از تعداد استفاده کل، باید کم شود. بنابراین تعداد استفاده مجدد در این حالت از (۳)، به‌دست می‌آید.

$$Reuse_T = Use_T - t.t = \left(\frac{t-k}{s} - ۱\right)^۲ .k^۲ - t^۲ \qquad (۳)$$

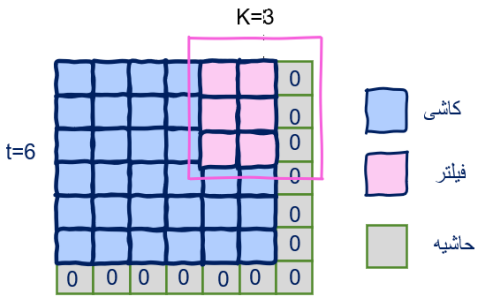
- استفاده از داده‌ها بعد از دومین بارگیری از حافظه : زمانی که که داده‌های سری جدید از حافظه خارجی بارگیری می‌شوند و در کاشی قرار می‌گیرند، از داده‌های قبلی که در کاشی بوده‌اند دوباره استفاده می‌شود. با توجه به شکل ۱۱، اگر جهت حرکت کرنل بر روی تنسور ورودی به صورت افقی باشد، مانند حالت الف، از $k-s$ ستون دوباره استفاده می‌شود و اگرجهت حرکت مانند حالت ب باشد، از $k-s$ ردیف، دوباره استفاده می‌شود.

نواحی صورتی در شکل ۱۱، داده‌هایی هستند که درون کاشی باقی مانده و دوباره استفاده می‌شوند و بقیه ناحیه کاشی با داده‌های جدید پر می‌شوند. باتوجه به شکل ۱۲، با در نظر گرفتن $k=۳, t=۵$ و $s=۱$ به عنوان نمونه، تعداد ۲ ستون از داده‌های کاشی، در کاشی باقی می‌مانند و بقیه ستون‌ها با داده‌های جدید پر می‌شوند. در تصویر مشخص است که از داده‌های این دو ستون، چندبار استفاده می‌شود.

برای اینکه تعداد دقیق استفاده مجدد از داده‌هایی که در کاشی باقی مانده‌اند، حساب شود، باید تعداد امکان استفاده از این داده‌ها محاسبه شود که این امکان با توجه به پارامترهای ساختاری شبکه به دست می‌آید.



شکل ۱۲: استفاده مجدد از داده‌های کاشی.



شکل ۱۳: حاشیه‌گذاری باتوجه به اندازه کاشی.

باتوجه به اینکه تعداد t ردیف (ستون) به طور کامل و تعداد k-s ستون (ردیف)، در کاشی باقی می‌ماند، باید تعداد استفاده ازداده‌های محدوده $t.(k-s)$ را محاسبه کرد. تعداد استفاده از داده‌ها در راستای ستون (ردیف) از (۴) به دست می‌آید.

$$Use_{partial} = k.(k-s) + k.(k-2s) + \ldots + k.(k-is) \quad (4)$$
$$\text{where } i = \frac{k}{s}$$

در (۴)، تعداد استفاده از داده‌های در راستای ردیف (ستون)، زمانی که کرنل با ابعاد $k \times k$ به تعداد i بار بر پنجره ورودی ضرب می‌شود. حال به دلیل اینکه در ابعاد دیگر کاشی، به اندازه کامل و t ردیف داده وجود دارد، با استفاده از (۱)، تعداد داده‌های خروجی تولیدشده با کاشی t، برابر No_t است و محاسبات به اندازه No_t بار تکرار می‌شود. در نتیجه تعداد داده‌ها نیز، No_t برابر $Use_{partial}$ می‌شود. در حالت کلی تعداد کل استفاده از داده‌هایی که در کاشی باقی می‌مانند و دوباره از آن‌ها استفاده می‌شود، از (۵)، به دست می‌آید.

$$Use_{reload} = No_t.k.[(k-s) + k.(k-2s) + \ldots + k.(k-is)] \quad (5)$$
$$\text{where } i = \frac{k}{s}$$

محاسبه تعداد کل استفاده مجدد از داده: در دو بخش قبلی، تعداد استفاده از داده را در حالت‌های بعد از یک بار بارگیری شدن داده در کاشی و بعد از اولین بار، محاسبه شد که تعداد کل استفاده مجدد از داده از (۳) و (۵) دست می‌آید که در (۶)، آورده شده است

$$Reuse_T = Reuse_{load} + Use_{reload} = \left(\frac{t-k}{s} - 1\right)^2 .k^2 - t^2 + No_t.k.[(k-s) + (k-2s) + \ldots + (k-2s)] \quad (6)$$

۴-۲ انتخاب کاشی‌های مجاز و الگوریتم پیشنهادی

برای کاهش فضای حالت برای بهینه‌سازی، کاشی‌هایی به عنوان مجاز انتخاب شده‌اند که برای محاسبات داده‌های آن‌ها و تولید خروجی به حاشیه‌گذاری در داده ورودی نیازی نباشد. در یک مثال عددی اگر

Function Optimum _Tile (k, s):

- Find valid tiles with no padding when:
  - $Tile \in valid\ tile \quad when\ (t-k)\%s=0$
  - $valid_{tiles} = \{t_1, t_2, \ldots, t_m\}$
- Find Data reuse for valid tiles with equation (6):
  - $\#data_{reuse} = \{reuse_{t1}, reuse_{t2}, \ldots, reuse_{tm}\}$
- Choose optimum tile when:
  - $t_{opt} = t_n \ \ when \ \ reuse_{t(n+1)} - reuse_{tn} \leq 10\% * reuse_{tn}$

شکل ۱۴: الگوریتم یافتن اندازه بهینه کاشی.

جدول ۱: اندازه بهینه کاشی بر اساس پارامترهای اندازه کرنل (k) و گام (S).

اندازه کرنل	اندازه گام	اندازه بهینه کاشی
۶	۱	۱۵
۶	۲	۲۶
۶	۵	۵۶
۱۳	۱	۲۱
۱۳	۸	۹۳
۵	۱	۱۴
۵	۲	۲۵

اندازه کاشی برابر $k=۳, t=۶$ و $s=۲$ انتخاب شود. با توجه به شکل ۱۳، در سومین تکرار کرنل بر روی تصویر ورودی، نیاز به افزودن یک ستون با مقدار صفربرای محاسبه خروجی است. که درواقع باعث افزایش حجم حافظه می‌شود ولی داده صحیح ندارد.

در همین جهت، برای جلوگیری از ایجاد حاشیه‌گذاری در اطراف تصویر و کم کردن حافظه و محاسبات، باید سایز کاشی طوری انتخاب شود که حاشیه‌گذاری ایجاد نشود. در واقع باید در (۱)، مقدار $t-k$ بر s بخش‌پذیر باشد تا حاشیه ایجاد نشود.

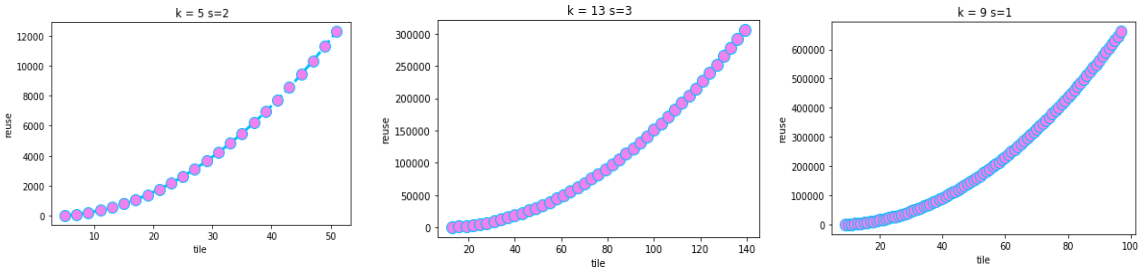
$$Tile \in valid_{tile} \text{ when } (t-k)\%s = ۰ \quad (7)$$

با استفاده از (۷)، از بین کاشی‌ها، کاشی‌های مجاز انتخاب می‌شوند و سپس با استفاده از الگوریتم شکل ۱۴، اندازه بهینه کاشی انتخاب می‌شود.

در الگوریتم پیشنهادی، در ابتدا سایزکاشی‌های مجاز به طوریکه نیاز به حاشیه‌گذاری نباشد، به دست می‌آید و سپس با استفاده از (۶) برای کاشی‌ها، تعداد استفاده مجدد داده حساب می‌شود و کاشی‌ای به عنوان کاشی بهینه انتخاب می‌شود که بعد از آن با زیاد شدن اندازه کاشی، تعداد افزایش استفاده از داده، کمتر از ۱۰٪ باشد.

باتوجه به الگوریتم و مدل ریاضی پیشنهادی، برای هر پیکربندی از لایه شبکه ، کاشی بهینه به دست می‌آید.

اندازه کاشی بهینه را بر اساس پارامترهای گام (s) و کرنل (k) به دست آوردیم که نتایج آن در جدول ۱، آورده شده است. باتوجه به نتایج، می‌توان استنتاج نمود که هرچه گام کوچکتر و نزدیک به یک باشد، اندازه کاشی بهینه نیز کوچکتر است، زیرا زمانی که اندازه گام نزدیک یک است، هم‌پوشانی بین پنجره‌هایی از ورودی که پردازش می‌شوند زیاد شده و استفاده از داده‌ها نیز، زیاد می‌شود. برای همین اندازه کاشی کوچک برای استفاده از داده‌ها کافی است. اما زمانی که اندازه گام بزرگتر شده و به اندازه کرنل نزدیک می‌شود، هم‌پوشانی‌ها کاهش داشته و درواقع تعداد استفاده از هر داده کاهش می‌یابد. درنتیجه برای افزایش تعداد استفاده مجدد از داده، باید اندازه کاشی بزرگتر شود تا بتواند هم‌پوشانی‌های بیشتری را در خود نگهداری کند.



(ج) (ب) (الف)

شکل ۱۵: نمودارتعداد استفاده مجدد از داده برحسب اندازه کاشی، (الف) برای لایه با اندازه کرنل ۹ و اندازه گام ۱، (ب) برای لایه با اندازه کرنل ۵ و اندازه گام ۲، (ج) برای لایه با اندازه کرنل ۱۳ واندازه گام ۳.

جدول ۲: اندازه بیشترین کرنل در شبکه‌های CNN مشهور.

شبکه	اندازه کرنل
AlexNet	۱۱ × ۱۱
VGGNet	۳ × ۳
ResNet	۷ × ۷
GoogleNet	۵ × ۵
MobileNet	۳ × ۳
Xception	۳ × ۳
NasNet	۷ × ۷
SqueezeNet	۳ × ۳

۴-۳ شبیه‌سازی و روش بهینه‌سازی

در قسمت ۴-۱، تعداد دقیق استفاده مجدد از داده بر حسب اندازه کاشی و پارامترهای شبکه، به صورت فرمول ریاضی مدل شد. برای ارزیابی و صحت‌سنجی فرمول، یک لایه از شبکه شبیه‌سازی و تعداد استفاده از پیکسل از کاشی، با شمارنده، شمارش شد که با اعدد حاصله از فرمول، تطابق داشت. همچنین خروجی لایه نیز صحیح بود و صحت‌سنجی انجام شد.

برای انتخاب اندازه کاشی بهینه، ابتدا با استفاده از (۷)، کاشی‌های مجاز به دست می‌آید و سپس برای هرکدام از کاشی‌ها تعداد استفاده مجدد داده از (۷) به دست می‌آید و کاشی به عنوان کاشی بهینه انتخاب می‌شود که بعد از آن با افزایش اندازه کاشی، تعداد استفاده مجدد از داده، کمتر از ۱۰٪ کاهش داشته باشد.

۵- نتایج

در این بخش، نتایج و شبیه‌سازی‌ها آورده شده است.

۵-۱ رابطه تعداد استفاده مجدد داده بر حسب اندازه کاشی

برای بررسی تاثیر اندازه کاشی بر تعداد استفاده مجدد از داده، برای پیکربندی‌های مختلف، با استفاده از (۶)، تعداد استفاده مجدد داده به دست آمد که نمونه‌ای از نمودار آن‌ها در شکل ۱۵، آورده شده است. در این نمودارها برای پیکربندی‌های مختلف از لایه‌ها با اندازه کرنل k، و گام s، که در عنوانشان نوشته شده، استفاده مجدد داده محاسبه شده است.

جدول ۳: میانگین نسبت اندازه کاشی بر اندازه کرنل در پیکربندی‌های مختلف.

اندازه گام	اندازه کرنل	نسبت میانگین
۱	۱،۲، ... ،۱۷	۲٫۳۷
۲	۲،۳، ... ،۱۷	۳٫۴۰
۳	۳،۴، ... ،۱۷	۴٫۳۰
۴	۴،۵، ... ،۱۷	۵٫۱۳
۵	۵،۶، ... ،۱۷	۵٫۸۰
۶	۶،۷، ... ،۱۷	۶٫۳۰
۷	۷،۸، ... ،۱۷	۶٫۹۰
	میانگین	۴٫۸۸

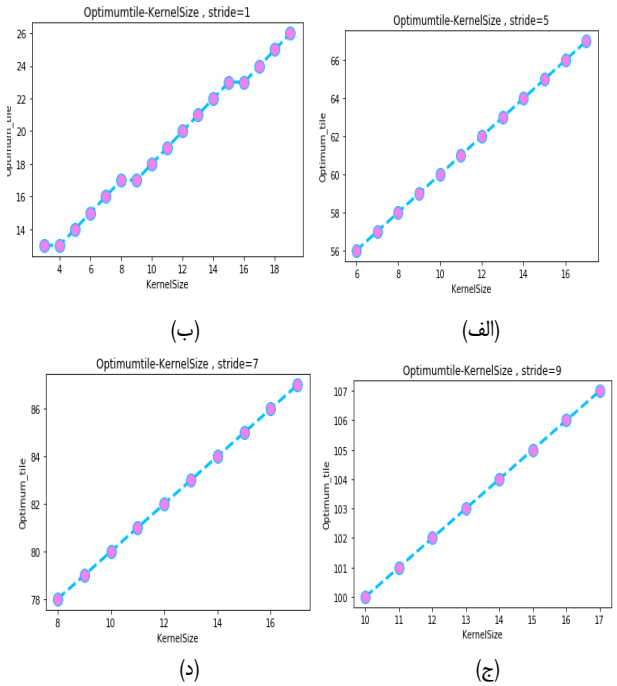
نقطه مشترک همه این نمودارها صعودی بودن تعداد استفاده مجدد داده با زیاد شدن اندازه کاشی، است. ازطرفی با افزایش اندازه کاشی، زمان کافی برای پردازش داده‌ها افزایش می‌یابد که ممکن است زمان اجرای کل را افزایش دهد. همچنین کنترل داده‌ها و پردازش آن‌ها سخت‌تر شده و ممکن است که عدم تعادل بین واحدهای پردازشی، ایجاد شود. سایز کاشی بزرگ‌تر نیاز به حافظه بزرگتر نیز دارد که معمولا محدودیت حافظه وجود دارد و زمان دسترسی به حافظه نیز افزایش می‌یابد. باتوجه به دلایل گفته شده، بهتر است که سایز کاشی بهینه طوری انتخاب شود که هم تعداد استفاده مجدد از داده عدد قابل قبولی باشد و هم اینکه اندازه کاشی، خیلی بزرگ نباشد. در همین جهت، الگوریتمی برای انتخاب اندازه کاشی بهینه، پیشنهاد کرده‌ایم که در قسمت بعدی آورده شده است.

۵-۲ تاثیر اندازه کرنل بر کاشی بهینه

در جدول ۲، میزان بیشترین اندازه کرنل در بین لایه‌های شبکه‌های مشهور CNN را گزارش شده است. با توجه به این جدول، مشخص است که بیشترین اندازه کرنل ۱۱×۱۱ است. در همین جهت برای بررسی تاثیر اندازه کرنل بر اندازه کاشی بهینه، برای حالت‌های مختلف از گام، اندازه کرنل را از ۳ تا ۱۷ در نظر گرفتیم.

اندازه کاشی بهینه بر اساس اندازه کرنل را باتوجه به الگوریتم پیشنهادی به دست آوردیم که در شکل ۱۶، نمودارهای مربوط به آن، نمایش داده شده است. نمودارها به ازای کرنل‌های مختلف و برای گام‌های ثابت رسم شده است.

در نمودارهای (الف) تا (د)، اندازه کاشی بهینه بر حسب کرنل‌های مختلف و گام ثابت ۱، ۵، ۷ و ۹، با استفاده از رابطه (۶) به دست آمده و



شکل ۱۶: نمودارهای اندازه کاشی بهینه برحسب کرنل‌های مختلف، (الف) گام ۱، (ب) گام ۵، (ج) گام ۷ ، و (د) گام ۹.

رسم شده است. در همه این نمودارها با افزایش مقدار کرنل، اندازه کاشی مجاز نیز افزایش می‌یابد. اما نکته قابل توجه این است که تقریبا کل نمودارها شیب یکسان و ثابتی دارند. به عبارت دیگر تقریباً نسبت $(OptimumTile/KernelSize)$ در هر نمودار عدد یکسانی است.

از همین رو جهت بررسی بیشتر، برای حالت‌های مختلف پیکربندی کرنل و گام، میانگین نسبت را محاسبه کردیم که نتیجه آن در جدول ۳، آورده شده است. در این جدول برای هریک از اندازه گام‌ها و کل اندازه کرنل‌های ممکن کوچکتر از ۱۷×۱۷، نسبت اندازه کاشی بهینه به کرنل را به دست آوردیم و میانگین این اعداد را حساب کردیم. برای مثال عدد ۲٫۳۷ برای گام یک، نشاندهنده این است که میانگین نسبت اندازه کاشی بهینه به اندازه کرنل برای پیکربندی‌هایی با گام یک و اندازه کرنل از ۲ تا ۱۷، برابر با ۲٫۳۷ شده است.

باتوجه به جدول ۳، نسبت اندازه کاشی بهینه بر حسب اندازه کرنل، نشان می‌دهد که اندازه کاشی بهینه، با افزایش گام، بزرگتر می‌شود. زیرا با بزرگتر شدن اندازه گام، هم‌پوشانی‌ها کاهش داشته و تعداد استفاده از هر داده کاهش می‌یابد. در نتیجه باید اندازه کاشی بزرگتر باشد تا بتواند، هم‌پوشانی‌های بیشتری را پوشش دهد و استفاده از داده افزایش یابد.

در ستون آخر نیز میانگین کل را برای نسبت‌ها گزارش کردیم که این عدد نیز نزدیک ۵ است و نشان می‌دهد که اندازه کاشی بهینه به طور میانگین ۵ برابر اندازه کرنل است تا تعداد استفاده مجدد از داده بهینه شود ودرواقع نیازی نیست که اندازه کاشی خیلی بزرگ باشد که در این صورت فقط بارحافظه دارد و تاثیری در تعداد استفاده مجدد از داده ندارد.

٦- مقایسه با روش قبلی

دراین قسمت برای لایه‌هایی از شبکه‌های LeNet و AlexNet که برای کلاسه‌بندی هستند، تعداد استفاده مجدد از داده مدل پیشنهادی‌مان و [۱۶] را، مقایسه کردیم. در جدول ، نتایج مربوطه را گزارش کرده‌ایم. برای لایه‌ها، اندازه کاشی باتوجه به محدودیت در مرجع [۱۶] و برای سطح حلقه‌ی یک بعد از خروجی، انتخاب شده است.

جدول ۴: تعداد استفاده مجدد از داده در لایه‌هایی از LeNet و AlexNet.

لایه	اندازه کاشی	[۱۶]	مدل پیشنهادی	درصد بهبود
LeNet.layer۱ $(Ni=۳۲, k=۵, s=۱)$	۵ ×۳۲	۷۰۰	۲۱۰۰	٪۶۷
LeNet.layer ۲ $(Ni=۱۴, k=۵, s=۱)$	۵ × ۱۴	۲۵۰	۷۵۰	٪۶۶
AlexNet.layer۱ $(Ni=۲۲۷, k=۱۱, s=۴)$	۱۱ ×۲۲۷	۶۶۵۵	۱۲۷۰۵	٪۴۷
AlexNet.layer ۱ $(Ni=۲۷, k=۵, s=۱)$	۵ ×۲۷	۵۷۵	۳۱۰۵	٪۸۱

با توجه به داده‌های جدول ۴، قابل مشاهده است که تعداد استفاده مجدد از داده با مدل پیشنهادی ما، به دلیل اینکه داده‌ها داخل کاشی نگهداری می‌شوند، بیشتر از ۵۰٪ افزایش داشته است و هرچه از داده بیشتر استفاده شود، بهره‌وری انرژی نیز، بیشتر است.

۷- نتیجه‌گیری

شبکه‌های عصبی CNN، نمونه‌ای از شبکه‌های عصبی عمیق هستند که در زمینه‌های مختلفی مانند، پردازش تصویر، زبان طبیعی، پزشکی و بسیار موارد دیگر کاربرد دارند. با افزایش لایه‌ها و تعداد پارامترها برای بهبود دقت، حرکت داده در شبکه زیاد شده و نیاز به حافظه نیز برای ذخیره داده‌های میانی تولید شده، بیشتر می‌شود. در بیشتر موارد حافظه داخلی برای ذخیره کافی نیست و به حافظه خارجی DRAM نیاز است که انرژی مصرفی به شدت افزایش می‌یابد.

برای کاهش انرژی مصرفی و کاهش استفاده از حافظه خارجی، بهتر است که از داده‌ها در هر یک از سطوح حافظه، بیشترین استفاده شود. ما در این مقاله، تعداد استفاده مجدد از داده را بر حسب اندازه کاشی و پارامترهای لایه از شبکه به صورت ریاضی مدل کردیم که با استفاده از مدل ریاضی پیشنهادی می‌توان تعداد استفاده مجدد از داده را به طور دقیق به دست آورد. در ادامه با انتخاب کاشی‌های مجاز که با استفاده از آن‌ها نیاز به حاشیه‌گذاری نباشد، فضای حالت را کاهش دادیم و با استفاده از الگوریتم پیشنهادی خود، برای هر یک از پارامترهای یک لایه از شبکه، سایز کاشی بهینه برای داشتن بیشترین استفاده مجدد از داده، استخراج کردیم. همچنین رابطه اندازه کاشی بهینه بر اساس اندازه گام و اندازه کرنل را نیز به دست آوردیم که این رابطه نشان داد که اندازه کاشی بهینه به طور میانگین ۵ برابر اندازه کرنل در شبکه است.

مراجع

[1] S. Genovese, "Artificial intelligence: a guide for thinking humans," *ORDO*, vol. 71, no. 1, pp. 444-449, Apr. 2020.

[2] O. Campesato, Artificial Intelligence, Machine Learning, and Deep Learning, Mercury Learning and Information, 2020.

[3] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu, "Quantized CNN: A unified approach to accelerate and compress convolutional networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 10, pp. 4730-4743, Oct. 2018.

[4] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 12, pp. 6999-7019, Dec. 2022

[5] Y. Ma, Y. Cao, S. Vrudhula, and J. Seo, "Optimizing loop Operation and dataflow in FPGA acceleration of deep convolutional neural networks," in *Proc. of the 2017 ACM/SIGDA Int. Symp. on Field-*

Programmable Gate Arrays, pp. 45-54, Monterey, CA, USA, 22-24 Feb. 2017.

[6] P. Dhilleswararao, S. Boppu, M. S. Manikandan, and L. R. Cenkeramaddi, "Efficient hardware architectures for accelerating deep neural networks: survey," *IEEE Access*, vol. 10, pp. 131788-131828, 2022.

[7] Y. –H. Chen, J. Emerl, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," *ACM SIGARCH Comput. Archit. News*, vol. 44, no. 3, pp. 367-379, Jun. 2016.

[8] S. Zheng *et al.*, "Efficient scheduling of irregular network structures on CNN accelerators," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3408-3419, Nov. 2020.

[9] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," in *Proc. IEEE Int. Solid-State Circuits Conf.*, pp. 10-14, San Francisco, CA, USA, 9-13 Feb. 2014.

[10] E. Valpreda *et al.*, "HW-Flow-Fusion: inter-layer scheduling for convolutional neural network accelerators with dataflow architectures," *Electron.*, vol. 11, no. 18, Article ID: 2933, Sept. 2022.

[11] M. Alwani, H. Chen, M. Ferdman, and P. Milder, "Fused-layer CNN accelerators," in *Proc. of the Annual Int. Symp. on Microarchitecture*, 12 pp., Taipei, Taiwan, 15-19 Oct. 2016.

[12] J. Li, *et al.*, "SmartShuttle: Optimizing off-chip memory accesses for deep learning accelerators," in *Proc. 2018 Design, Automation and Test in Europe Conf. Exhib.*, pp. 343-348, Dresden, Germany, 19-23 Mar. 2018.

[13] Q. Nie and S. Malik, "MemFlow: Memory-driven data scheduling with datapath co-design in accelerators for large-scale inference applications," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 39, no. 9, pp. 1875–1888, Sept. 2020

[14] Q. Nie and S. Malik, "CNNFlow: Memory-driven data flow optimization for convolutional neural networks," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 28, no. 3, Article ID: 40, Feb. 2022.

[15] A. Parashar, *et al.*, "Timeloop: A systematic approach to DNN accelerator evaluation," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Software*, pp. 304-315, Madison, WI, USA, 2019.

[16] A. Stoutchinin, F. Conti, and L. Benini, Optimally Scheduling CNN Convolutions for Efficient Memory Access, arXiv Preprint, arXiv:1902.01492, Feb. 2019.

[17] L. Cavigelli, *et al.*, "Origami: A convolutional network accelerator," in *Proc. ACM 25th edition on Great Lakes Symp. on VLSI,* pp. 199-204, Pittsburgh, PA, USA, 20-22 May 2015.

[18] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proc. of the 32nd Int. Conf. on Machine Learning*, pp. 1737-1746, Lille, France, 6-11 Jul. 2015.

[19] C. Zhang, *et al.* "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. 2015 ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, pp. 161-170, Monterey, CA, USA, Feb. 2015.

[20] I. Dadras, S. Seydi, M. H. Ahmadilivani, J. Raik, and M. E. Salehi, "Fully-Fusible Convolutional Neural Networks for End-to-End Fused Architecture with FPGA Implementation," in *Proc. 2023 30th IEEE Int. Conf. Electron. Circuits Syst.*, 5 pp., Istanbul, Turkey, 4-7 Dec. 2023.

[21] B. Rokh, A. Azarpeyvand, and A. Khanteymoori, "A comprehensive survey on model quantization for deep neural networks in image classification," *ACM Trans. on Intelligent Systems and Technology*, vol. 14, no. 6, Article ID: 97, Dec. 2023.

سوفیا صیدی در سال ۱۳۹۵ مدرک کارشناسی مهندسی برق خود را از دانشگاه سراسری زنجان و در سال ۱۳۹۷ مدرک کارشناسی ارشد مهندسی برق خود را از همان دانشگاه دریافت کرد. از سال ۱۳۹۸ وارد دوره دکتری معماری سیستم‌های کامپیوتری در دانشگاه تهران شد. زمینه‌های علمی مورد علاقه وی شامل موضوعات طراحی مدارهای دیجیتال، بلوک‌های محاسبات تقریبی، بهبود حافظه در شتابدهنده‌های شبکه‌های عصبی، معماری کامپیوتر، پیاده‌سازی سخت‌افزاری شبکه‌های عصبی و مهندسی نرم‌افزار است.

مصطفی صالحی نسب مدرک کارشناسی مهندسی کامپیوتر را در سال ۱۳۸۰ از دانشگاه تهران و همچنین مدرک کارشناسی ارشد مهندسی کامپیوتر را در سال ۱۳۸۲ از دانشگاه صنعتی امیرکبیر دریافت کرد. ایشان در سال ۱۳۸۹ مدرک دکتری مهندسی کامپیوتر را از دانشگاه تهران دریافت کرد و در حال حاضر به عنوان استادیار در دانشکده مهندسی برق و کامپیوتر دانشگاه تهران مشغول به کار است. علایق تحقیقاتی او شامل معماری کامپیوتر، طراحی سیستم‌های نهفته، هم‌طراحی سخت‌افزار/ نرم‌افزار و پیاده‌سازی شتابدهنده‌های شبکه‌های عصبی عمیق است. وی سرپرستی آزمایشگاه پژوهشی سیستم‌های نهفته چندهسته‌ای را در دانشکده مهندسی برق و کامپیوتر دانشگاه تهران برعهده دارد.