Journal of Soft Computing and Information Technology (JSCIT)
Babol Noshirvani University of Technology, Babol, Iran
Journal Homepage: jscit.nit.ac.ir
Volume 13, Number 2, Summer 2024, pp. 25-36
Received: 03/04/2023, Revised: 08/27/2024, Accepted: 11/12/2024

Application of Approximate Computing Methods in Convolutional Neural Networks to Improve Accuracy and Performance

Mohammadrzea Rafieinejad[1], Mohammadreza Binesh Marvasti[2*] and Seyyed Amir Asghari[3]
1- Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran.
2*- Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran.
3- Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran.
[1]mohamad.rafiei@khu.ac.ir, [2*]marvasti@khu.ac.ir, and [3]asghari@khu.ac.ir

Corresponding author's address: Mohammadreza Binesh Marvasti, Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran.

Abstract- Artificial neural networks are one of the most influential and important branches of science in the past decades, which have been used in a wide range of applications such as medical and pharmaceutical services, voice and speech recognition, computer vision, natural language processing, video and image processing, etc. . In order for neural networks to have the most efficiency, the algorithms used in them must have deep networks with many layers, which requires very high computing power both in the network training stage and in the inference stage. In recent years, approximate computing has been one of the most promising approaches to reduce energy consumption in applications that can tolerate accuracy reduction. In the proposed method, specific to the first layer of the convolution neural network, in order to compress and reduce the computations, each pixel is given a label using the K-Means clustering algorithm on the input images, and based on that, the convolution process is performed using the average of the clustered pixels, which it will reduce the number of multiplication operations by at least 90%. Then, in order to further compression, using one of the neural network pruning methods based on the L1-norm criterion, different amounts of pruning are applied to all the layers of that network. According to the results of this research, the criteria of accuracy, efficiency, and parameter size of convolutional neural networks have been improved using the proposed method on 2 separate convolutional neural networks and on 4 different datasets.

Keywords- Neural Networks, Approximate Computing, Energy Consumption, Computer Vision, Machine Learning

مجله علمی رایانش نرم و فناوری اطلاعات
دانشگاه صنعتی نوشیروانی بابل
صفحه مجله: jscit.nit.ac.ir
جلد ۱۳، شماره ۲، تابستان ۱۴۰۳، صفحه ۲۵-۳۶
دریافت: ۱۴۰۱/۱۲/۱۳، بازنگری: ۱۴۰۳/۰۶/۰۶، پذیرش: ۱۴۰۳/۰۸/۲۲

کاربردِ محاسبات تقریبی در شبکه‌های عصبی کانولوشنی به منظور بهبود دقت و کارایی

محمدرضا رفیعی نژاد[۱]، محمدرضا بینش مروستی*[۲]، سید امیر اصغری[۳]
۱- گروه مهندسی برق و کامپیوتر، دانشگاه خوارزمی، تهران، ایران.
*۲- گروه مهندسی برق و کامپیوتر، دانشگاه خوارزمی، تهران، ایران.
۳- گروه مهندسی برق و کامپیوتر، دانشگاه خوارزمی، تهران، ایران.
[1]mohamad.rafiei@khu.ac.ir, [2*]marvasti@khu.ac.ir, [3]asghari@khu.ac.ir

* نشانی نویسنده مسئول: محمدرضا بینش مروستی، تهران، خیابان انقلاب، خیابان شهید مفتح، دانشگاه خوارزمی، گروه مهندسی برق و کامپیوتر.

چکیده- شبکه‌های عصبی مصنوعی، یکی از تاثیرگذارترین و مهم‌ترین شاخه‌های علوم در دهه‌های گذشته هستند که در برنامه‌های کاربردیِ وسیعی مانند برنامه‌های خدمات درمانی و دارویی، تشخیص صدا و گفتار، بینایی کامپیوتر، پردازش زبان طبیعی، پردازش ویدئو و تصویر و ... به‌کار گرفته شده‌اند. برای اینکه شبکه‌های عصبی، بیشترین بازده را داشته باشند، الگوریتم‌های مورد استفاده در آن‌ها بایستی شبکه‌هایی عمیق با لایه‌های فراوان داشته باشند، که این امر نیاز به قدرت محاسباتیِ بسیار بالا چه در مرحله‌ی آموزش شبکه و چه در مرحله‌ی استنتاج دارد. در سال‌های اخیر محاسبات تقریبی، یکی از امیدبخش‌ترین رویکردها برای کاهش مصرف انرژی در برنامه‌هایی که قابلیت تحمل درجه‌ای از کاهش دقت را دارند، بوده است. در روش پیشنهادی، مختص لایه اول شبکه‌عصبی کانولوشن، به منظور فشرده‌سازی و کاهش محاسبات، با استفاده از الگوریتم خوشه‌بندیِ K-Means بر روی تصاویر ورودی، به هر پیکسل برچسبی داده می‌شود و براساس آن عمل کانولوشن با استفاده از میانگین پیکسل‌های هم‌خوشه انجام می‌گیرد که موجب کاهشِ دست کم ۹۰ درصد در تعداد عملیات ضرب خواهد شد. سپس به منظور فشرده‌سازیِ بیشتر، با استفاده از یکی از روش‌های هرس شبکه‌های عصبی بر مبنای معیار L_1-norm، مقدار مختلفی از هرس بر روی تمام لایه‌های آن شبکه اِعمال می‌شود. طبق نتایج این پژوهش، معیارهای دقت، کارایی و اندازه پارامترهای شبکه‌های عصبی کانولوشن با استفاده از روش پیشنهادی برروی ۲ شبکه عصبی کانولوشنی مجزا و بر روی ۴ مجموعه داده مختلف، بهبود یافته است.

واژه‌های کلیدی: شبکه‌های عصبی، محاسبات تقریبی، مصرف انرژی، بینایی کامپیوتر، یادگیری ماشین.

۱- مقدمه

محاسبات تقریبی در تمامی مسائل پیچیده‌ای که بتوان از راه‌حلی غیردقیق به نتیجه‌ای قابل قبول رسید، کاربرد دارد [۱]. امروزه شرکت‌های تولید فناوری کامپیوتری به سمت بهینه‌سازی مصرف انرژی در محصولات خود رفته‌اند، اما در مجموع، سرعت مصرف انرژی بسیار بالاتر از این حرکت می‌باشد. برای مثال حجم بالای داده‌های در دسترس مثلا در کلان داده و محاسباتی که باید بر روی آن‌ها انجام شود و همچنین تولید حسگرهای مختلف در محیط‌های گوناگون که با محیط واقعی در تعامل هستند و از آن داده دریافت می‌کنند. این داده‌ها در حجم انبوه تولید شده و معمولا افزونگی‌های بسیاری نیز دارند [۲]. لذا نقش محاسبات تقریبی به‌منظور کاهش بار محاسباتی و حذف این افزونگی‌ها و همچنین مصرف انرژی لازم به‌نظر می‌رسد.

۱-۱- نیاز به محاسبات تقریبی

در جهان امروز، با حجم انبوهی از داده‌هایی روبرو هستیم که از دنیای واقع گرفته شده‌اند. این داده‌ها از منابعی مانند حسگرها (و نه منابع سنتی) بدست می‌آیند و اغلب دارای خطا و افزونگی و اعوجاج هستند. برنامه‌های کاربردی که از این نوع داده‌ها استفاده می‌کنند مانند برنامه‌های تشخیص اشیاء یا پردازش محتوای چندرسانه‌ای معمولاً به نتایجی کاملاً دقیق نیاز ندارند، لذا اگر خطاهایی تا آستانه‌ی قابل قبولی اتفاق بیفتند، نتیجه همچنان معتبر و صحیح خواهد بود. برای مثال در پردازش ویدئو از میان میلیون‌ها فریم اگر یکی از آن‌ها دچار خطا شود، تغییر محسوسی در پخش آن ویدئو اتفاق نخواهد افتاد. به‌عنوان مثالی دیگر اگر دو طبقه‌بندی کننده در یک شبکه‌ی عصبی کانولوشن، تصاویر یک مجموعه داده را به درستی طبقه‌بندی کنند، بسیار دشوار خواهد بود که بگوییم کدام یک بهتر این کار را انجام داده است و برای ما تنها نتیجه‌ی درست، قابل اکتفا خواهد بود. لذا محاسبات تقریبی باید در برنامه‌هایی مورد استفاده قرار گیرد که به نوعی نسبت به بروز خطا، تحمل‌پذیر باشند [۲].

از طرف دیگر، اطمینان از عدم وجود خطا در تراشه‌های نانومتری امروزی بسیار هزینه‌بر و مصرف کننده‌ی انرژی می‌باشد، چراکه باید افزونگی‌ها و خطاها در تمامی سطوح سلسله مراتبیِ مدارهای مجتمع متراکم، بررسی شده و در صورت لزوم تصحیح شوند. با استفاده از محاسبات تقریبی در مراحل تعیین مشخصات و پیاده‌سازی برنامه‌های کاربردیِ تحمل‌پذیر خطا، محاسبات تقریبی مقدار قابل قبولی از خطا را وارد پروسه‌ی تولید می‌کند که نتایج شگفت‌آوری در بهینه‌سازی مصرف انرژی به‌دنبال خواهد داشت [۲].

۱-۲- روش‌های محاسبات تقریبی

روش‌های محاسبات تقریبی طیف گسترده‌ای از حوزه‌ها را در بر می‌گیرند. روش‌های محاسبات تقریبی می‌توانند از پایین‌ترین سطح سخت‌افزاری یعنی طراحی ترانزیستورها تا بالاترین سطوح نرم‌افزاری، مانند پیاده‌سازی نرم‌افزار لحاظ شوند. این روش‌ها تاثیر مستقیمی بر زمان اجرا، حافظه‌ی موردنیاز، میزان دسترسی به حافظه، یکپارچگی سخت‌افزاری و خروجی دارند. برای تعداد بسیار زیادی از سیستم‌ها (یا برنامه‌های کاربردی)، دقت و صحت پارامترهایی حیاتی نمی‌باشند، لذا این سیستم‌ها می‌توانند از مزایای محاسبات تقریبی بهره‌مند شوند [۳].

طبق قانون گرایش انرژی کومی، ((انرژی مورد نیاز برای محاسبات درصورت ثابت نگه داشتن بار محاسباتی، هر یک و نیم سال، با ضریبی از ۲ کاهش خواهد یافت)) [۴]، اما قانون کومی بوسیله‌ی اصل لاندوئر محدود شده است [۶،۵]، این اصل می‌گوید که حد پایینی برای حداقل میزان مصرف انرژی محاسبات وجود دارد که تا سال ۲۰۴۰ قانون کومی را تحت تاثیر قرار خواهد داد. بنابراین اگر تا سال ۲۰۴۰ فکری برای کاهش مصرف انرژی محاسبات نشود، وارد بحران خواهیم شد. اهمیت روش‌های محاسبات تقریبی در اینجا کاملاً روشن می‌شود [۷].

۱-۳- کابردهای محاسبات تقریبی

محاسبات وارد دوره‌ی طلائی تقریب شده است، دوره‌ای که سخت افزار و نرم افزار و حتی استنتاج از روی تقریب، تخمین زده می‌شوند. برای مثال برنامه‌های مسیریابی مکان مقصد را براساس داده‌هایی که از حسگرهای سخت افزارِ تعیین محلِ خودرو دریافت می‌کنند، تخمین می‌زنند. برنامه‌های تشخیص گفتار، سیگنال‌های آنالوگ را به محتمل‌ترین جمله، ترجمه می‌کنند و موتورهای جست و جو عبارت پرس و جو را به اطلاعات مرتبط تبدیل می‌کنند. تمامی برنامه‌های کاربردی ذکر شده به الگوریتم‌های حجیم و پیچیده‌ای نیاز دارند تا بتوانند پاسخ‌هایی قابل قبول، در مدت زمانی معقول و به تناسب حجم تقاضا به کاربران ارائه دهند. الگوریتم‌های مورد اشاره باید از لحاظ مصرف انرژی نیز کارآمد و بهینه باشند. راه‌کار مورد نیاز برای تامین موارد ذکر شده، اغلب، محاسبات تقریبی است [۲]. شناسائی بخش‌های مقاوم در برابر خطا، جزء دیگری از کاربرد تقریب در برنامه‌ها می‌باشد. حتی در برنامه‌های مقاوم در برابر خطا، هنوز قسمت‌های حساس به خطا نیز وجود دارند که بکارگیری محاسبات تقریبی در آن قسمت‌ها می‌تواند منجر به خطاهای مرگبار شود. بنابراین گام مهم دیگری که باید برای استفاده از محاسبات تقریبی برداشته شود، تعیین قسمت‌های مقاوم در برابر خطا در یک برنامه‌ی کاربردی است. یکی از روش‌های شناسائی تحمل‌پذیری، استفاده از نشانه‌گذاری در برنامه‌ها و دیگری بکارگیری انواع داده‌های تقریبی اختصاصی برای بیان اینکه تقریب در کجا امکان‌پذیر است و چگونه ممکن است بر نتایج تاثیر گذارد، می‌باشد [۸, ۹].

۱-۴- برنامه‌های کاربردی و حوزه‌های کاربردِ محاسبات تقریبی

در [۱۰] پردازنده‌ی تقریبی پیشرو در محاسبات تقریبی معرفی شده است که می‌تواند در دو حالت کار کند. یکی با دقت کامل و دیگری با دقت تقریبی (کاهش یافته، اما سریع‌تر). این پردازنده شامل یک ثَبات کنترلی تقریبی و یک پرچم حالت است. در [۱۱] قابلیت‌های محاسبات تقریبی برروی دستگاه‌های الکترونیکی قابل حمل بررسی شده است. دستگاه‌های قابل حمل، انرژی مصرفی خود را از باتری تامین می‌کنند و دارای محدودیت‌های مرتبط با باتری هستند و

طول عمر باتری نقش مهمی در این سیستم‌ها دارند. لذا اگر انرژی مصرفی محاسباتی به طریقی قابل کمینه شدن باشد، مزیت بسیار مهمی خواهد بود. بنابراین کاربرد محاسبات تقریبی در دستگاه‌های قابل حمل می‌تواند موضوع جالبی برای توسعه و تحقیقات در آینده باشد [۷].

۱-۵- کاربرد محاسبات تقریبی در شبکه‌های عصبی

در [۱۲] یک شبکه‌ی عصبی جدید از ترکیب یک تقریب‌ساز و یک شبکه‌ی پیش‌بینی کننده برای ایجاد یک نسخه‌ی تقریبی از یک شبکه‌ی عصبی سنتی ساخته شده است. این شبکه توانسته است زمان آموزش را به طور قابل توجهی کاهش دهد. یک شبکه‌ی عصبی می‌تواند رفتار یک تابع استاندارد را نسبت به ورودی‌های مختلف از طریق یادگیری ماشین بیاموزد. در یک سیستم پیچیده، می‌توان از شبکه‌های عصبی برای پیاده‌سازی سخت‌افزاری-نرم‌افزاریِ توابع تقریبی استفاده نمود. کدهای سنتی محاسبات تقریبی را می‌توان به شبکه‌های عصبی معادل آن منتقل کرد که دقت خروجی پایین‌تر اما کارائی زمان اجرای بالاتری دارند [۱۳،۳].

۲- مطالعات مرتبط

متعلق به یک نقشه‌ی ویژگی خروجی هرس می‌شوند. روش اشتراک‌گذاری وزن‌ها: این روش در سطح ساختاری یک شبکه‌ی عصبی کانولوشن پیاده‌سازی می‌شود. اشتراک‌گذاری وزن‌ها بار محاسباتی را کاهش نمی‌دهد، اما حافظه‌ی مورد نیاز برای ذخیره‌سازی این وزن‌ها کاهش می‌یابد.

تکنیک‌های محاسبات تقریبیِ سخت افزاری به منظور ساده‌سازی ساختار سخت افزاری برای پیاده‌سازی شبکه‌های عصبی کانولوشن و شتاب بخشیدن به کارائی آن‌ها به کار می‌روند. عملیات‌های ضرب و جمع در شبکه‌های عصبی کانولوشن به مقدار زیادی محاسبه و منابع حافظه نیاز دارند. بنابراین جمع‌کننده‌ها و ضرب‌کننده‌های تقریبی برای انجام عملیات‌های ضرب و جمع طراحی می‌شوند تا مصرف انرژی را کاهش داده و کارائی را بهبود بخشند. البته روش‌های محاسبات تقریبی در سطح حافظه نیز برای بهبود مصرف توان و انرژی در مدار، به کار می‌روند [۱].

الف) تقریب‌های کارکردی در ماژول‌های محاسباتی: ماژول‌های محاسباتی در یک شبکه‌ی عصبی کانولوشن، همان جمع‌کننده و ضرب‌کننده‌ها برای انجام عملیات‌های ضرب و جمع هستند. همان‌طور که می‌دانیم عملیات ضرب، عموما مصرف توان بالایی دارد و به واحدهای سخت‌افزاری زیادی نیاز دارد. بنابراین با طراحی ضرب کننده‌ها و جمع‌کننده‌های تقریبی می‌توانیم گذردهی و بهینه‌سازی مصرف انرژی را افزایش دهیم [۱]. در [۱۶] از جمع کننده، ضرب کننده و تقسیم کننده‌هایی برای بهبود کارائی در برنامه‌های کاربردیِ تحمل پذیر اِشکال استفاده شده است.

ب) تقریب‌های کارکردی در حافظه‌ها: در سطح سخت افزاری و برای شتاب بخشیدن به فرآیندهای پردازشیِ شبکه‌های عصبی کانولوشن، حافظه‌ها با محدودیت‌هایی نظیر پهنای باند، تاخیر در دسترسی، تراکم و توان مصرفی روبرو هستند. با محاسبات تقریبی، حافظه‌ها نیز می‌توانند به شکل تقریبی کار کنند تا از بهینه‌سازیِ مورد نیاز در مصرف انرژی بهره‌مند شوند.

ج) کاربرد حافظه‌های غیر فرار: این حافظه‌ها به منظور افزایش بهینه‌سازی مصرف توان و انرژی در یک سیستم به کار می‌روند. پلتفرم‌های سخت‌افزاری که برای پیاده‌سازی شتاب‌دهنده‌های شبکه‌های عصبی کانولوشن استفاده می‌شوند، با چالش‌های جابجایی داده در حافظه مواجه هستند [۱]. در [۱۷] روش‌های تقریبی برای فائق آمدن بر این چالش‌ها در سیستم‌ها معرفی شده‌اند. در شبکه‌های عصبی کانولوشن عملیات حسابی تقریبی گزینه‌ی مناسب‌تری برای ساده‌سازی طراحی شتاب‌دهنده‌ها است چراکه پیچیدگی طراحی را کاهش می‌دهد. داده‌ای که در شبکه‌های عصبی مورد استفاده قرار می‌گیرد، اگر با دقت بالا ذخیره شود، به مقدار زیادی حافظه برای ذخیره‌ی پارامترها و نتایج محاسبات میانی، نیاز دارد. عملیات حسابی با استفاده از نمایش ممیز شناور محاسبات را پیچیده می‌کند و پیچیدگی طراحیِ واحدهای سخت افزاری مانند جمع کننده‌ها و ضرب کننده‌ها را افزایش می‌دهد.

الف) ممیز ثابت: برای دستیابی به کارائی بالاتر و کاهش استفاده از منابع و همچنین حافظه، نمایش ممیز شناور می‌تواند با عملیات حسابیِ ممیز ثابت جایگزین شود. در ممیز ثابت، نمایش داده در عملیات حسابی به شکل عدد صحیح می‌باشد که به راحتی می‌تواند برروی پلتفرم‌های شتاب‌دهنده‌ها پیاده‌سازی شود. در [۱۸] یک چارچوب یادگیریِ روی تراشه به منظور کارائی بالا و هزینه‌ی بهینه‌ی معماری که از نمایش ممیز ثابت ۱۶-۸ بیتی استفاده می‌کند، معرفی شد.

ب) ممیز ثابت داینامیک: در یک شبکه‌ی عصبی کانولوشن، مقادیر مرتبط با وزن‌ها و پارامترها در لایه‌های مختلف، می‌توانند محدوده‌ی نمایش متفاوتی داشته باشند. نمایش ممیز ثابت قابلیت فراهم آوردنِ محدوده‌ی گسترده‌ای از مقادیر را برای لایه‌های گوناگون دارد، در غیراینصورت دقت کاهش خواهد یافت. بازه‌ی داینامیکی گسترده، نمایش ممیز ثابت را محدود می‌کند. بنابراین ممیز ثابت داینامیک زمانی استفاده می‌شود که اجازه‌ی استفاده از بازه‌ی بزرگی را برای نمایش پارامترهای توابع فعالسازیِ لایه‌های مختلفِ شبکه

داشته باشیم [۱].

پ) کاهش تعداد بیت‌های ذخیره سازی توان ۲: در این روش، وزن‌های شبکه‌ی عصبی کانولوشن به قالب توان ۲، کاهش داده می‌شوند. در این روش نمایش داده با تعداد بیت‌های کمتری صورت می‌گیرد و در نتیجه کاهش دقت کمتری نیز خواهیم داشت. این روش اجازه‌ی استفاده از انتقال‌دهنده‌ی بشکه‌ای به جای ضرب کننده‌ی معمولی را می‌دهد که باعث کاهش پیچیدگی طراحی می‌شود [۱].

ت) کاهش تعداد بیت‌های ذخیره سازی باینری: این روش از کاهش تعداد بیت‌های ذخیره سازی با نمایش وزن‌ها و پارامترهای شبکه به شکل تک بیتی، دسترسی به حافظه و پهنای باند حافظه را کاهش می‌دهد. شبکه‌های عصبیِ برآمده از این روش، شبکه‌های عصبی باینری نام دارند. در این روش، می‌توان از مالتی پلکسر به جای ضرب کننده‌های پیچیده استفاده کرد [۱]. دو نوع باینری سازی داریم: وزن‌ها و توابع فعالسازی باینری شده با دقت کامل و وزن‌ها و توابع باینری شده.

۳- روش پیشنهادی

تعداد بسیار بالای پارامترها در شبکه‌های عصبی چالش‌های متعددی بوجود آورده است. از جمله‌ی این چالش‌ها نیاز به پهنای باندِ حافظه بالا و فضای ذخیره سازی حجیم برای این پارامترها می‌باشد. این گلوگاه‌ها در مورد سیستم‌های نهفته و دستگاه‌های قابل حمل ملموس‌تر هستند و اهمیت بیشتری دارند. در سال‌های گذشته، روش‌های بسیاری برای کاهش پهنای باند و حجم شبکه، همچنین کاهش مصرف انرژی در شبکه‌های عصبی کانولوشن با استفاده از راه‌کارهای محاسبات تقریبی معرفی شده‌اند. این ایده‌ها، طیف وسیعی از راه‌کارهای محاسبات تقریبی از جمله سخت‌افزاری (تولید ضرب‌کننده‌های تقریبی مختص شبکه‌های کانولوشن برای کاهش مصرف انرژی)، نرم‌افزاری (هرس پارامترهای دارای افزونگی و غیرتاثیرگذار) و سخت‌افزاری-نرم‌افزاری (کاهش تعداد بیت نمایش داده‌ها، کوانتیزاسیون) را دربر می‌گیرند.

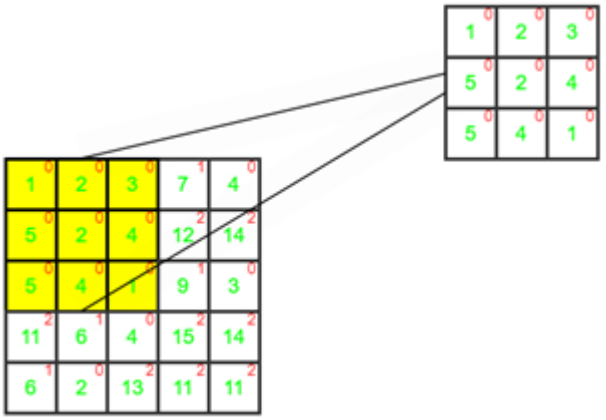
روش پیشنهادی، برای لایه اول یک شبکه‌ی عصبی کانولوشن به کار می‌رود و با بکارگیری الگوریتم خوشه‌بندی K-Means بر روی پیکسل‌های تصاویر آموزش و ارزیابی ، به هر پیکسل، متناسب با خوشه‌ای که در آن قرار دارد، یک برچسب می‌دهد. ابتدا خوشه‌بندی با الگوریتم K-Means (برای مثال با فرض $k = 3$)، انجام می‌شود. این خوشه‌بندی **پیش از فاز آموزش شبکه** صورت می‌گیرد. بنابراین تاثیری در زمان پردازشِ شبکه نخواهد داشت. در طراحی الگوریتم پیشنهادی، از این برچسب‌ها برای تعیین نوع کانولوشن بین کرنل و تصویر ورودی، یعنی کانولوشن معمولی یا کانولوشن تقریبی استفاده می‌شود. دو حالت اتفاق می‌افتد، این دو حالت را حالت معمول (حالتی که کانولوشن رایج انجام می‌شود) و حالت تقریبی (حالتی که کانولوشن تقریبی انجام می‌شود) نامگذاری می‌کنیم. لازم به ذکر است که روش پیشنهادی برای تمامی مجموعه داده‌ها، چه مجموعه داده‌هایی شامل تصاویر رنگی و چه مجموعه داده‌هایی شامل تصاویر سیاه و سفید، کارایی داشته و محدودیتی از این جهت ندارد. در ادامه، ابتدا این دوحالت به تفصیل شرح داده خواهند شد، سپس الگوریتم پیشنهادی مورد بررسی قرار خواهد گرفت.



شکل ۱: تصاویری از مجموعه داده CIFAR10 و خوشه‌بندی شده همان تصاویر با الگوریتم K-Means

۳-۱- حالت تقریبی در روش پیشنهادی

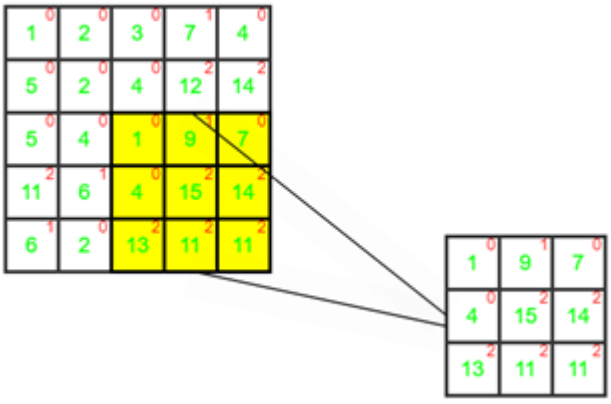
حالت تقریبی، حالتی است که تمامی ۹ پیکسل (با فرض ۳ × ۳ بودن کرنل) برچسب‌های یکسانی داشته باشند، در واقع تمامی ۹ پیکسل در یک خوشه قرار دارند. در این حالت به جای ضرب درایه به درایه و نظیر به نظیر که در کانولوشن معمولی به کار می‌رود، میانگین این ۹ پیکسل فقط یکبار در جمع وزن‌های فیلتر ضرب می‌شود. درنتیجه به جای ۹ ضرب که در کانولوشن معمولی انجام می‌شود، در کانولوشن تقریبی، از یک ضرب استفاده می‌کنیم که به معنای تعداد ۸ ضرب کاهش در عملیات محاسباتی می‌باشد. عملیات ضرب، بیشترین سهم مصرف انرژی را در عملیات محاسباتی دارد، بنابراین کاهش این تعداد ضرب، موجب کاهش محسوس مصرف انرژی خواهد شد. در شکل ۲ نحوه کانولوشن نشان داده شده است. همان‌گونه که در شکل ۲ قابل درک است، می‌توان دریافت که تمامی ۹ پیکسل سمت چپ در بالای جدول، برچسب‌های یکسانی دارند، بنابراین برای این پیکسل‌ها، از کانولوشن تقریبی به جای کانولوشن معمولی استفاده می‌کنیم.



شکل ۲: حالت کانولوشن تقریبی. جدول زرد رنگ، نشان‌دهنده پیکسل‌هایی می‌باشد که در یکی از لغزش‌های کرنل برروی تصویر ورودی، به منظور کانولوشن در فیلتر، اسکن شده‌اند و برچسب‌های یکسان دارند (مقادیر پیکسل‌ها فرضی هستند)

۳-۲- حالت معمول در روش پیشنهادی

حالت معمول، حالتی است که ۹ پیکسلِ قرارگرفته در ناحیه زرد رنگِ شکل ۳ (با فرض ۳ × ۳ بودن کرنل) برچسب یکسانی نداشته باشند. در این حالت، این ۹ پیکسل، در یک خوشه قرار ندارند. بنابراین برای این حالت، کانولوشن معمولی (مطابق شکل ۳)، یعنی ضرب درایه به درایه و درنهایت جمع آن‌ها استفاده شده است.



شکل ۳: حالت کانولوشن معمول. جدول زرد رنگ، نشان‌دهنده پیکسل‌هایی می‌باشد که در یکی از لغزش‌های کرنل برروی تصویر ورودی، به منظور کانولوشن در فیلتر، اسکن شده‌اند و در یک خوشه قرار ندارند (مقادیر پیکسل‌ها فرضی هستند)

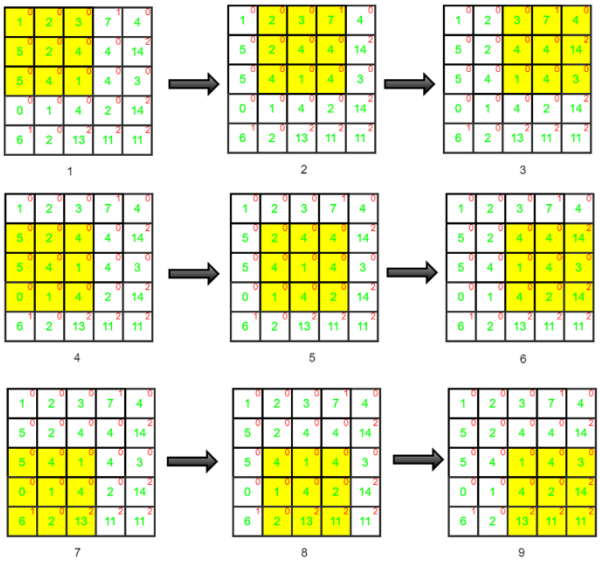
۳-۳- محاسبه نقشه ویژگی در روش پیشنهادی

اگر فیلترها را با نماد $F_{i,j}$ برای لایه i اُم و کانالِ (یا نورون) j اُم نشان دهیم، نقشه ویژگی خروجی، O_j طبق رابطه‌ی ۱ بدست خواهد آمد.

$$O_j = \sum_i (x_i * F_{i,j}) + B_j \qquad (۱)$$

رابطه ۱، محاسبه نقشه ویژگی، در حالت معمول را نشان می‌دهد.

در این رابطه، x_i نماد نقشه ویژگی لایه i اُم و B_j مقدار بایاس مربوط به کانال j اُم می‌باشد. در شکل ۴ حالت‌های مختلف لغزش کرنل بر روی تصویر ورودی و تشکیل حالت‌های تقریبی و معمولی نمایش داده شده است.



شکل ۴: حالت‌های مختلف لغزشِ کرنل برروی تصویر ورودی، حالت‌های ۱، ۴ و ۵ حالت‌های تقریبی و حالت‌های ۹، ۸، ۷، ۶ و ۳ حالت‌های معمول (استاندارد) هستند

برای لایه اول x_i همان تصویر ورودی است. خروجیِ هر حالت (ضرب درایه به درایه و در نهایت جمع اعداد، برای حالت معمول و ضرب میانگین در جمع وزن‌ها برای حالت تقریبی) یکی از درایه‌های نقشه ویژگی خروجی را می‌سازد. در شکل ۴ تمامی حالات لغزش کرنل بر روی تصویر نمایش داده شده است.

اگر مقدار میانگینِ پیکسل‌های حالت تقریبی را که باید درایه به درایه در کرنل ضرب شوند با نماد M، و همچنین وزن‌های یک فیلتر را با نماد w و اندازه طول (یا عرض) کرنل را با نماد k نشان دهیم، رابطه ساخت یک درایه از نقشه ویژگی خروجی، برای حالت تقریبی الگوریتم پیشنهادی برای کانال j اُم، به شکل زیر خواهد بود:

$$O_{approximate} = M \times \sum_{k=1}^{K_{max}} w_k \qquad (۲)$$

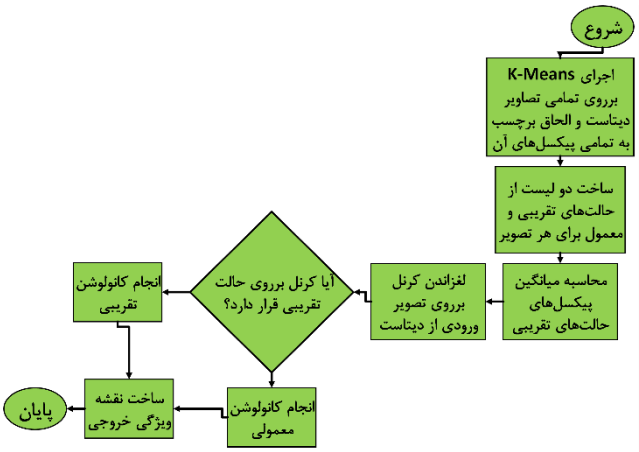
همان‌گونه که در رابطه ۲ مشخص است، واضح است که تنها یک ضرب برای حالت تقریبی وجود دارد. در این رابطه اندازه کرنل هرچه باشد، بزرگ یا کوچک، تاثیری در تعداد ضرب انجام شده نخواهد داشت و در هر صورت تنها یک ضرب خواهیم داشت.

۳-۴- الگوریتم روش پیشنهادی

این الگوریتم ابتدا تمامی تصاویر مجموعه داده (تصاویر آموزش و

تصاویر ارزیابی) را **قبل از فاز آموزش شبکه**، با استفاده از الگوریتم خوشه‌بندیِ K-Means خوشه‌بندی می‌کند و به هر پیکسل از هر تصویر، برچسبی الحاق می‌کند (هر پیکسل در آن تصویر مربوط به چه خوشه‌ای است)، سپس در حین اجرای فاز آموزش شبکه، حالت‌های تقریبی و معمول را برای هر تصویر که به شبکه داده می‌شود، تعیین می‌کند.

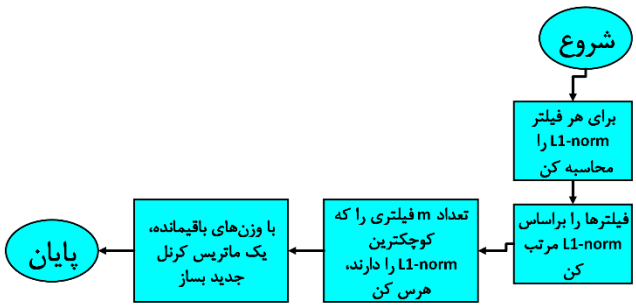
در ادامه، میانگین پیکسل‌های حالت‌های تقریبی را محاسبه می‌کند و برای حالت‌های معمول، کانولوشن معمولی و برای حالت‌های تقریبی کانولوشن تقریبی انجام می‌دهد و در نهایت نقشه ویژگی خروجی را بر اساس خروجی‌های حالت‌های تقریبی و حالت‌های معمولی می‌سازد. در شکل ۵ بلوک دیاگرام روش پیشنهادی نمایش داده شده است.



شکل ۵: فلوچارت روش پیشنهادی

۳-۵- هرس فیلترها با استفاده از معیار L_1-Norm در یک شبکه کانولوشنی

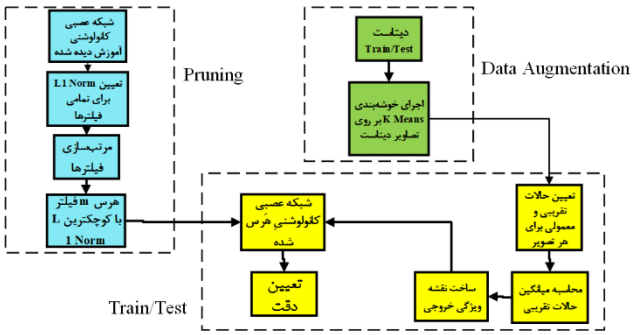
در [۱۹]، بر اساس جمع قدرمطلق وزن‌های هر فیلتر، یعنی $|\sum F_{i,j}|$ یا L1-Norm آن، اهمیت هر فیلتر ارزیابی می‌شود. هرچه این مقدار کمتر باشد، آن فیلتر کم اهمیت‌تر است، بنابراین می‌توانیم تعداد دلخواهی از فیلترهای کم اهمیت را هرس کنیم. یعنی ابتدا تمامی L1-Norm های تمامی لایه‌ها محاسبه شده و سپس مرتب‌سازی می‌شوند. درنهایت، تعداد m فیلتر که کوچکترین L1-Norm را داشته باشند (که در جدول‌های ۲، ۳، ۴ و ۵ به نام **میزان هرس** و به صورت **درصد** نگارش شده است. همچنین مقدار m به **صورت دلخواه**، فرضاً ۱۰ درصد از کوچکترین فیلترها، **توسط ما** تعیین می‌گردد) در یک گام و در تمامی لایه‌های شبکه هرس می‌شوند. الگوریتم هرس، طبق فلوچارت شکل ۶ توضیح داده شده است.



شکل ۶: فلوچارت (روند نما) روش تغییر یافته هرس [۲۶]

۳-۶- بلوک دیاگرام روش پیشنهادی و روش هرس بکار رفته بر روی یک شبکه عصبی کانولوشن

در روش بکار رفته در این ژورنال، تغییراتی در دو فاز جداگانه تصاویر مجموعه داده و شبکه عصبی کانولوشن انجام می‌شود. **در فاز نخست**، ابتدا تمامی تصاویر مجموعه داده (تصاویر آموزش و تصاویر ارزیابی) با استفاده از الگوریتم K-Means خوشه‌بندی می‌شوند. سپس، در حین آموزش شبکه، حالات تقریبی و حالات معمولی برای پیکسل‌های هر تصویر مشخص می‌شوند و میانگین پیکسل‌های حالات تقریبی محاسبه خواهند شد. در نهایت با استفاده از کانولوشن تقریبی برای حالات تقریبی و کانولوشن معمولی برای حالات معمولی، نقشه ویژگی خروجی ساخته خواهد شد. تمامی مراحل فاز اول، مختص لایه اول شبکه عصبی کانولوشن می‌باشد. **در فاز دوم**، پیش از آموزش شبکه، ابتدا یک شبکه عصبی کانولوشنیِ از قبل آموزش داده شده[۱] (مانند VGG16, Densenet121, Resnet110 و ...) انتخاب شده و طبق الگوریتم هرس، مقادیر L_1-Norm تمامی فیلترهای تک تک لایه‌های آن محاسبه می‌شوند. سپس این فیلترها مرتب سازی شده و تعداد m فیلتر (m به دلخواه انتخاب می‌شود) در تمامی لایه‌ها که کوچکترین مقادیر L_1-Norm را دارند، هرس می‌شوند. پس از آغازِ مرحله آموزش شبکه، نقشه ویژگی خروجیِ آماده شده از فاز نخست به شبکه عصبی کانولوشنیِ هرس شده داده می‌شود تا دقت شبکه اندازه‌گیری شود.



شکل ۷: مراحل مختلف روش پیشنهادی و روش هرس [۱۹] و آموزش شبکه عصبی هرس شده

۴- نتایج تجربی

به منظور مقایسه همه جانبه و دقیق روش پیشنهادی و اخذ نتایج معتبر و قابل اتکا، از ۴ مجموعه داده مختلف و ۲ شبکه عصبی کانولوشنی استفاده شده است.

اولین مجموعه داده [۲۰] CIFAR10 است. این مجموعه داده شامل ۶۰۰۰۰ تصویر رنگی با ابعاد ۳۲×۳۲ در ۱۰ کلاس مختلف می‌باشد. مجموعه داده آموزش آن شامل ۵۰۰۰۰ تصویر و مجموعه داده ارزیابی آن شامل ۱۰۰۰۰ تصویر است. دومین مجموعه داده [۲۱] SVHN می‌باشد. این مجموعه داده تصاویری از پلاک‌های واقعی خانه‌های مختلف است و شامل ۹۹۲۸۹ تصویر از اعداد دیجیتالی بین ۰ تا ۹ می‌باشد، بنابراین ۱۰ کلاس دارد. در این مجموعه داده، ۷۳۲۵۷ تصویر برای مجموعه داده آموزش و ۲۶۰۳۲ تصویر برای مجموعه داده ارزیابی اختصاص داده شده است. [۲۲] RAF-DB سومین مجموعه داده استفاده شده در این ژورنال می‌باشد. این مجموعه داده دارای ۱۲۲۷۱ تصویر برای آموزش و ۳۰۶۸ تصویر برای ارزیابی می‌باشد. تصاویر این مجموعه داده، دربرگیرنده چهره انسان‌ها با احساسات مختلف است و دارای ۷ کلاس می‌باشد. مجموعه داده چهارم [۲۲] Compound RAF-DB می‌باشد. این مجموعه داده ۳۱۶۲ تصویر برای آموزش و ۷۹۲ تصویر برای ارزیابی دارد که دربرگیرنده چهره انسان‌ها با ترکیبی از دو احساس مختلف است و دارای ۱۲ کلاس می‌باشد.

در این ژورنال از ۲ شبکه عصبی کانولوشنی استفاده شده است. یک شبکه از نوع Resnet [۲۴] می‌باشد که نسخه Resnet50 آن بکار رفته است. شبکه دوم، [۲۵] Densenet121 می‌باشد. معماری‌های این دو شبکه به صورت زیر تغییر یافته‌اند. در شبکه Resnet50، ابتدا به جای لایه اول کانولوشن آن، لایه روش پیشنهادی قرار داده شده است. سپس از یک سری لایه ترتیبی (متشکل از لایه‌های کانولوشن و نرمال‌سازی دسته‌ها و تابع فعالسازی Relu و همچنین لایه انتخاب بیشینه نقشه ویژگی) تا دو لایه مانده به انتهای شبکه استفاده شده است. در انتها نیز دو لایه **تماما متصل** قرار داده شده است. همچنین برای شبکه Densenet121، معماری آن به یک لایه ترتیبی و دو **لایه تماما** متصل تغییر یافته است. در این معماری نیز، روش پیشنهادی جایگزین لایه اول کانولوشن شده است. در نهایت، در تمام شبیه‌سازی‌ها از تنظیمات زیر استفاده شده است:

اندازه هر دسته داده: ۶۴، تعداد دورهای آموزش شبکه: ۱۵، نرخ یادگیری شبکه برای مجموعه داده‌های CIFAR10 و SVHN: ۰.۰۱، نرخ یادگیری شبکه برای مجموعه داده RAF-DB: 0.001، تابع بهینه‌ساز: Stochastic Gradient Descent (SGD)، تابع اتلاف: Cross Entropy Loss. همچنین در تمامی شبیه‌سازی‌ها برای تعیین حالات تقریبی و معمول (استاندارد)، علاوه بر تصاویر و برچسب‌ها، تصاویر خوشه‌بندی شده نیز به شبکه داده شده است که علت بیشتر بودن پارامترهای شبکه VGG16 با روش پیشنهادی می‌باشد. همچنین برای محاسبه تعداد عملیات ممیز شناور، این مقدار برای هر لایه در هر شبکه به شکل مجزا (با توجه به نوع لایه) برای هر لایه، محاسبه شده و سپس با هم جمع شده است.

۴-۱- نتایج شبیه‌سازی روش پیشنهادی برای مجموعه داده‌های مختلف بر روی شبکه Resnet50

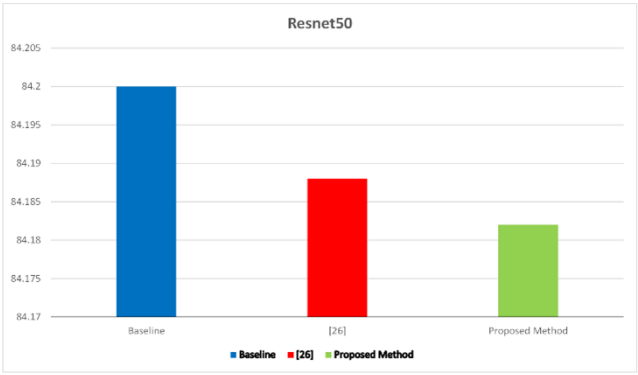
در این قسمت، نتایج حاصل از بکارگیری روش پیشنهادی بر روی شبکه Resnet50 در مجموعه داده‌های مختلف در جدول ۱ نمایش داده شده است. در شبکه Resnet50، ابتدا به جای لایه اول کانولوشن آن، لایه روش پیشنهادی قرار داده شده است. سپس از یک سری لایه ترتیبی[۲] (متشکل از لایه‌های کانولوشن و نرمال‌سازی دسته‌ها و تابع فعالسازی Relu و همچنین لایه انتخاب بیشینه نقشه ویژگی) تا دو لایه مانده به انتهای شبکه استفاده شده است. در انتها نیز دو لایه **تماما متصل** قرار داده شده است. در این جدول بهبود دقت، تعداد عملیات ممیز شناور و همچنین تعداد پارامترهای شبکه در روش پیشنهادی نسبت به مقاله [۲۶] و شبکه پایه، کاملاً مشهود است.

جدول ۱ نتایج حاصل از شبیه‌سازیِ مجموعه داده‌های CIFAR16، SVHN و RAF-DB و Compound RAF-DB بر روی شبکه Resnet50

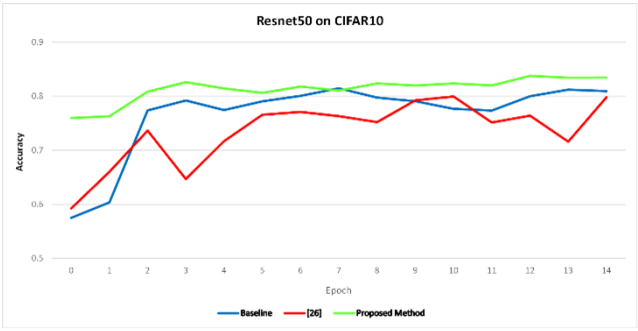
روش / مجموعه داده	شبکه پایه			مقاله [26]			روش پیشنهادی		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
CIFAR 10	81.46	84.21 MMac	23639744	79.95	84.1876 MMac	23633472	83.76	84.1826 MMac	23632064
SVHN	96.88	84.21 MMac	23639744	93.72	84.1876 MMac	23633472	95.03	84.1826 MMac	23632064
RAF-DB	74.80	84.21 MMac	23639744	73.20	84.1876 MMac	23633472	75.78	84.1826 MMac	23632064
Compound RAF-DB	48.23	84.21 MMac	23639744	56.78	84.1876 MMac	23633472	63.89	84.1826 MMac	23632064

در شکل ۸ نمودار میله‌ایِ مقایسه تعداد عملیات ممیز شناور برای شبکه Resnet50 برای مجموعه داده‌های مختلف نشان داده شده است. در این شکل، بهبود تعداد عملیات ممیز شناور در روش پیشنهادی (رنگ سبز)، نسبت به مقاله [۲۶] و شبکه پایه مشهود است.

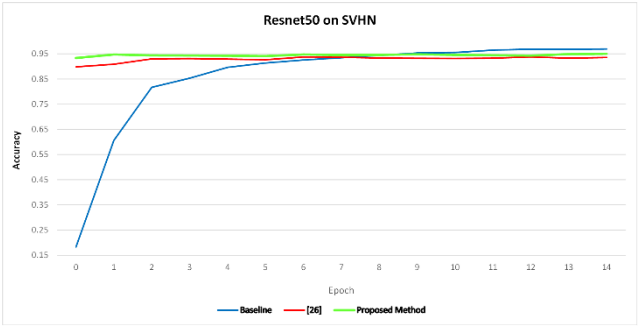
در شکل‌های ۹ و ۱۰ دیاگرام‌های دقت شبکه Resnet50 برای روش پیشنهادی بر روی مجموعه داده‌های CIFAR10 و SVHN نشان داده شده است. محور افقی با برچسب Epoch معرف تعداد دورهای آموزش و محور عمودی نشان‌دهنده دقت اندازه‌گیری شده برای مجموعه ارزیابی می‌باشد.



شکل ۸: مقایسه مقدار عملیات ممیز شناور برای شبکه پایه، مقاله [۲۶] و روش پیشنهادی برای مجموعه داده‌های CIFAR10، SVHN و RAF-DB و Compound RAF-DB



شکل ۹: مقایسه میزان دقت میان شبکه پایه، مقاله [۲۶] و روش پیشنهادی در شبکه Resnet50 برای مجموعه داده CIFAR10



شکل ۱۰: مقایسه میزان دقت میان شبکه پایه، مقاله [۲۶] و روش پیشنهادی در شبکه Resnet50 برای مجموعه داده SVHN

با بررسی شکل ۹، واضح است که دقت روش پیشنهادی از مقاله [۲۶] و شبکه پایه بیشتر می‌باشد. در شکل ۱۰ نیز دقت روش پیشنهادی از مقاله [۲۶] بهتر است.

۴-۲- نتایج شبیه‌سازی روش پیشنهادی و اِعمال روش هرس [۱۹] برای مجموعه داده‌های مختلف بر روی شبکه Resnet50

برای شبکه Resnet50 روش هرس مقاله [۱۹] برای لایه کانولوشن اِعمال شده است. با توجه به اَفت محسوس دقت پس از هرسِ لایه‌های نرمال‌سازیِ دسته[۳] از هرس کردن این لایه‌ها در شبکه Resnet50صرف نظر شده است. در روش هرس بکاررفته، ابتدا تمامی L1-Norm های تمامی لایه‌ها محاسبه شده و سپس مرتب‌سازی می‌شوند. درنهایت، تعداد m فیلتر که کوچکترین L1-Norm را داشته باشند (که در جدول‌های ۲، ۳، ۴ و ۵ به نام **میزان هرس** و به صورت **درصد** نگارش شده است، همچنین مقدار m به **صورت دلخواه**، فرضاً ۱۰ درصد از کوچکترین فیلترها، **توسط ما** تعیین می‌گردد) در **یک گام**[۴] و در تمامی لایه‌های شبکه هرس می‌شوند. در جدول‌های ۲، ۳، ۴ و ۵ میزان دقت، تعداد عملیات ممیز شناور و اندازه پارامترها پس از هر بار انجام هرس، به میزان معین، به ترتیب برای مجموعه داده‌های CIFAR10، SVHN و RAF-DB بیان شده است.

جدول ۲ مقایسه روش پیشنهادی به‌همراه روش هرس [۱۹] با مقاله [۲۶] و شبکه پایه با مقدار هرس‌های متفاوت برای شبکه Resnet50 و مجموعه داده CIFAR10

روش / نرخ هرس	شبکه پایه			مقاله [26]			روش پیشنهادی و روش هرس [19]		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
40%	72.1	50.6016 MMac	14205094	79.73	50.5868 MMac	14200925	82.04	50.5836 MMac	14200486
50%	76.73	42.1996 MMac	11846432	79.26	42.1864 MMac	11842747	81.95	42.1846 MMac	11842592
60%	75.98	33.7975 MMac	9487770	79.26	33.7860 MMac	9484550	83.08	33.7855 MMac	9484698
70%	76.54	25.3954 MMac	7129108	79.19	25.3855 MMac	7126339	82.34	25.3864 MMac	7126803
80%	64.91	16.9933 MMac	4770444	79.41	16.9851 MMac	4768138	83.03	16.9873 MMac	4768909
90%	58.07	8.5913 MMac	2411782	78.69	8.5849 MMac	2409993	82.39	8.5882 MMac	2411014

جدول ۳ مقایسه روش پیشنهادی به‌همراه روش هرس [۱۹] با مقاله [۲۸] و شبکه پایه با مقدار هرس‌های متفاوت برای شبکه Resnet50 و مجموعه داده CIFAR10

روش / نرخ هرس	شبکه پایه			مقاله [28]			روش پیشنهادی و روش هرس [19]		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
30%	83.48	50.6000 MMac	16563756	93.23	55.3660 MMac	18123895	83.00	50.5836 MMac	16558381

جدول ۴ مقایسه روش پیشنهادی به‌همراه روش هرس [۱۹] با مقاله [۲۶] و شبکه پایه با مقدار هرس‌های متفاوت برای شبکه Resnet50 و مجموعه داده SVHN

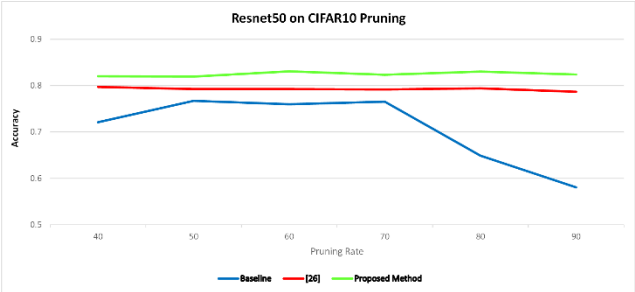
روش / نرخ هرس	شبکه پایه			مقاله [26]			روش پیشنهادی و روش هرس [19]		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
40%	91.57	50.6016 MMac	14205094	93.19	50.5868 MMac	14200925	75.83	50.5836 MMac	14200486
50%	93.31	42.1996 MMac	11846432	93.73	42.1864 MMac	11842747	78.97	42.1846 MMac	11842592
60%	93.06	33.7975 MMac	9487770	93.63	33.7860 MMac	9484550	95.16	33.7855 MMac	9484698
70%	92.07	25.3954 MMac	7129108	93.89	25.3855 MMac	7126339	95.01	25.3864 MMac	7126803
80%	91.67	16.9933 MMac	4770444	93.67	16.9851 MMac	4768138	94.84	16.9873 MMac	4768909
90%	89.9	8.5913 MMac	2411782	93.68	8.5849 MMac	2409993	94.76	8.5882 MMac	2411014

جدول ۵ مقایسه روش پیشنهادی به‌همراه روش هرس [۱۹] با مقاله [۲۶] و شبکه پایه با مقدار هرس‌های متفاوت برای شبکه Resnet50 و مجموعه داده RAF-DB

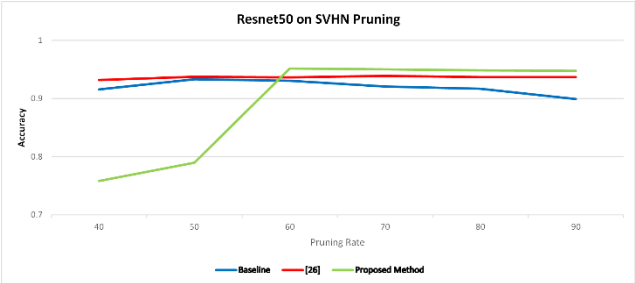
روش / نرخ هرس	شبکه پایه			مقاله [26]			روش پیشنهادی و روش هرس [19]		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
40%	78.38	50.6016 MMac	14205094	72.84	50.5868 MMac	14200925	84.21	50.5836 MMac	14200486
50%	78.00	42.1996 MMac	11846432	72.29	42.1864 MMac	11842747	83.63	42.1846 MMac	11842592
60%	79.96	33.7975 MMac	9487770	73.95	33.7860 MMac	9484550	83.27	33.7855 MMac	9484698
70%	79.59	25.3954 MMac	7129108	71.44	25.3855 MMac	7126339	82.62	25.3864 MMac	7126803
80%	79.13	16.9933 MMac	4770444	67.24	16.9851 MMac	4768138	80.81	16.9873 MMac	4768909
90%	79.98	8.5913 MMac	2411782	52.86	8.5849 MMac	2409993	81.26	8.5882 MMac	2411014

در جداول ۲، ۴ و ۵ پس از استفاده از روش پیشنهادی و روش هرس [۱۹]، تعداد عملیات ممیز شناور، اندازه پارامترها و همچنین دقت نسبت به مقاله [۲۶] و شبکه پایه بهبود یافته است. در جدول ۳ اندازه پارامترها و تعداد عملیات ممیز شناور برای روش پیشنهادی و روش هرس [۱۹] در تمامی حجم‌های مختلف هرس، بهتر شده است. همچنین میزان دقت بعد از هرسِ ۶۰٪ همواره نسبت به مقاله [۲۶] و شبکه پایه بهتر بوده است. در جدول ۹ تعداد عملیات ممیز شناور، اندازه پارامترها نسبت به مقاله [۲۸] بهبود یافته است.

در شکل‌های ۱۱ و ۱۲ میزان دقت به دست آمده بر اساس مقادیر مختلف هرس برای شبکه Resnet50 نشان داده شده است.



شکل ۱۱: دقت به دست آمده برای درصدهای گوناگونِ هرس در شبکه Resnet50 و مجموعه داده CIFAR10



شکل ۱۲: دقت به دست آمده برای درصدهای گوناگونِ هرس در شبکه Resnet50 و مجموعه داده SVHN

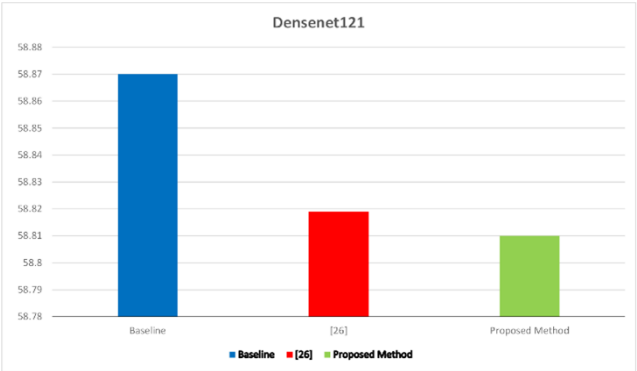
۴-۳- نتایج شبیه‌سازی روش پیشنهادی برای مجموعه داده‌های مختلف بر روی شبکه Densenet121

همان‌گونه در جدول ۶ مشاهده می‌شود، معیارهای دقت، تعداد عملیات ممیز شناور و اندازه پارامترها برای روش پیشنهادی و بر روی شبکه Densenet121 در تمامی مجموعه داده‌ها نسبت به مقاله [۲۶] و شبکه پایه بهتر می‌باشد. برای شبکه Densenet121، معماری آن به یک لایه ترتیبی و دو **لایه تماما** متصل تغییر یافته است. در این معماری نیز، روش پیشنهادی جایگزین لایه اول کانولوشن شده است.

در شکل ۱۳ نمودار میله‌ای مقایسه تعداد عملیات ممیز شناور بر روی شبکه Densenet121 در مجموعه داده‌های مختلف نشان داده شده است.

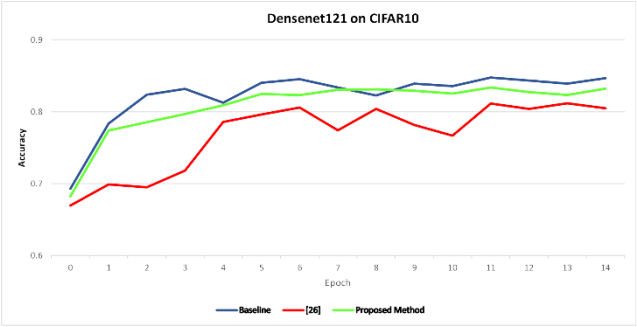
جدول ۶ نتایج حاصل از شبیه‌سازیِ مجموعه داده‌هایCIFAR10، SVHN، RAF-DB، وCompound RAF-DB برروی شبکهDensenet121

روش / مجموعه داده	شبکه پایه			مقاله [26]			روش پیشنهادی		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
CIFAR 10	84.78	58.87 MMac	7017984	81.18	58.8173 MMac	7011712	83.38	58.8055 MMac	7010304
SVHN	94.78	58.87 MMac	7017984	94.67	58.8173 MMac	7011712	95.47	58.8055 MMac	7010304
RAF-DB	74.73	58.87 MMac	7017984	69.88	58.8173 MMac	7011712	75.26	58.8055 MMac	7010304
Compound RAF-DB	64.37	58.87 MMac	7017984	73.94	58.8173 MMac	7011712	74.86	58.8055 MMac	7010304

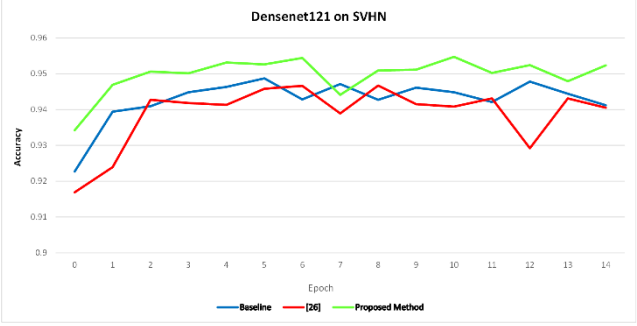


شکل ۱۳: نمودار میله‌ایِ مقایسه مقدارعملیات ممیز شناور برای شبکه پایه، مقاله[۲۶] و روش پیشنهادی برای مجموعه داده‌های CIFAR10،SVHNوRAF-DB و Compound RAF-DB برروی شبکه Densenet121

از شکل ۱۳ واضح است که تعداد عملیات ممیز شناور برای روش پیشنهادی از روش مقاله [۲۶] و شبکه پایه کمتر است. در ادامه در شکل‌های ۱۴ و ۱۵ دیاگرام مقایسه دقت روش پیشنهادی و روش مقاله [۲۶] و شبکه پایه بر روی شبکه Densenet121 و مجموعه داده‌های CIFAR10 و SVHN نمایش داده شده است.



شکل ۱۴: مقایسه میزان دقت میان شبکه پایه، مقاله [۲۶] وروش پیشنهادی در شبکه Densenet121 برای مجموعه داده‌CIFAR10



شکل ۱۵: مقایسه میزان دقت میان شبکه پایه، مقاله[۲۶] و روش پیشنهادی در شبکه Densenet121 برای مجموعه داده SVHN

در شکل ۱۴ میزان دقت روش پیشنهادی از مقاله [۲۶] بیشتر می‌باشد. همچنین در شکل ۱۵ مشاهده می‌شود که میزان دقت روش پیشنهادی از مقاله [۲۶] و شبکه پایه بیشتر می‌باشد.

۴-۴- نتایج شبیه‌سازی روش پیشنهادی و اِعمال روش هرس [۱۹] برای مجموعه داده‌های مختلف بر روی شبکه Densenet121

در این قسمت نتایج استفاده از روش پیشنهادی و روش هرس مقاله [۱۹] در مجموعه داده‌های CIFAR10 و SVHN و RAF-DB و Compound RAF-DBبا روش مقاله [۲۶] و شبکه پایه مقایسه می‌شود. برای انجام این شبیه‌سازی، لایه‌های کانولوشن و نرمال‌سازی دسته‌ها در شبکه Densenet121 هرس شده‌اند. نتایج ارزیابی نشان داد که این شبکه با جایگزینی روش پیشنهادی در لایه اول کانولوشن خود و سپس هرس آن، حتی به مقدار ۹۰٪ نیز اُفت محسوس دقت نخواهد داشت. در عین حال روش پیشنهادی دارای تعداد عملیات ممیز شناور کمتر و اندازه پارامترهای کمتر نسبت به روش مقاله [۲۶] و شبکه پایه می‌باشد. در ادامه در جدول‌های ۷، ۸ و ۹ نتایج مقایسه میزان دقت، تعداد عملیات ممیز شناور و اندازه پارامترها نسبت به میزان مختلف هرس، برای شبکه پایه، مقاله [۲۶] و روش پیشنهادی در مجموعه داده‌های مختلف برای شبکه Densenet121 نمایش داده شده است.

جدول ۷ مقایسه روش پیشنهادی به‌همراه روش هرس [۱۹] با مقاله [۲۶] و شبکه پایه با مقدار هرس‌های متفاوت برای شبکه Densenet121 و مجموعه داده‌CIFAR10

روش / نرخ هرس	شبکه پایه			مقاله [26]			روش پیشنهادی و روش هرس [19]		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
40%	55.27	35.4589 MMac	4227111	77.96	35.4261 MMac	4223211	82.19	35.4202 MMac	4222503
50%	55.95	29.6061 MMac	3529392	78.15	29.5777 MMac	3526020	82.60	29.5738 MMac	3525552
60%	53.20	23.7533 MMac	2831673	76.00	23.7289 MMac	2828772	81.78	23.7275 MMac	2828601
70%	49.92	17.9005 MMac	2133955	74.66	17.8802 MMac	2131530	80.89	17.8812 MMac	2131651
80%	63.56	12.0478 MMac	1436237	69.51	12.0311 MMac	1434250	80.70	12.0349 MMac	1434701
90%	80.50	6.1950 MMac	738519	69.51	6.1824 MMac	737023	79.05	6.1885 MMac	737751

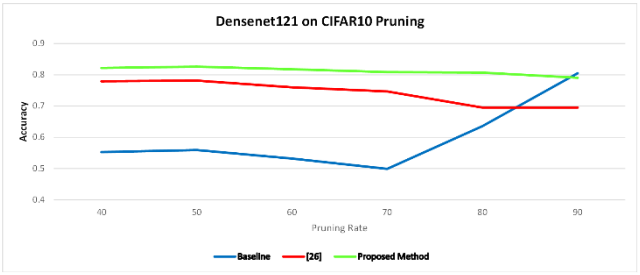
جدول ۸ مقایسه روش پیشنهادی به‌همراه روش هرس [۱۹] با مقاله [۲۶] و شبکه پایه با مقدار هرس‌های متفاوت برای شبکه Densenet121 و مجموعه دادهSVHN

روش / نرخ هرس	شبکه پایه			مقاله [26]			روش پیشنهادی و روش هرس [19]		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
40%	85.75	35.4589 MMac	4227111	93.61	35.4261 MMac	4223211	95.33	35.4202 MMac	4222503
50%	85.73	29.6061 MMac	3529392	93.31	29.5777 MMac	3526020	95.53	29.5738 MMac	3525552
60%	87.29	23.7533 MMac	2831673	93.34	23.7289 MMac	2828772	95.17	23.7275 MMac	2828601
70%	87.34	17.9005 MMac	2133955	93.54	17.8802 MMac	2131530	95.40	17.8812 MMac	2131651
80%	90.58	12.0478 MMac	1436237	93.38	12.0311 MMac	1434250	94.94	12.0349 MMac	1434701
90%	93.58	6.1950 MMac	738519	92.77	6.1824 MMac	737023	94.84	6.1885 MMac	737751

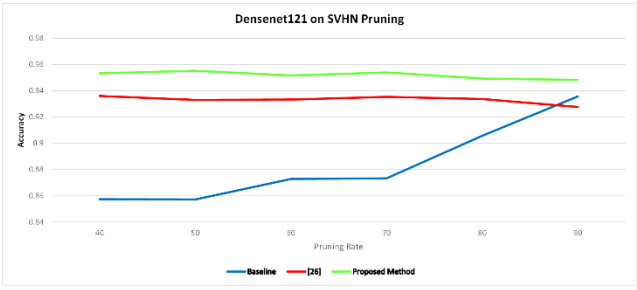
جدول ۹ مقایسه روش پیشنهادی به‌همراه روش هرس [۱۹] با مقاله [۲۶] و شبکه پایه با مقدار هرس‌های متفاوت برای شبکه Densenet121 و مجموعه دادهRAF-DB

روش / نرخ هرس	شبکه پایه			مقاله [26]			روش پیشنهادی و روش هرس [19]		
	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها	دقت	تعداد عملیات ممیزشناور	اندازه پارامترها
40%	78.29	35.4589 MMac	4227111	77.86	35.4261 MMac	4223211	80.71	35.4202 MMac	4222503
50%	77.18	29.6061 MMac	3529392	77.41	29.5777 MMac	3526020	79.58	29.5738 MMac	3525552
60%	76.89	23.7533 MMac	2831673	76.23	23.7289 MMac	2828772	79.14	23.7275 MMac	2828601
70%	76.59	17.9005 MMac	2133955	75.74	17.8802 MMac	2131530	79.01	17.8812 MMac	2131651
80%	70.56	12.0478 MMac	1436237	72.91	12.0311 MMac	1434250	77.39	12.0349 MMac	1434701
90%	70.89	6.1950 MMac	738519	68.44	6.1824 MMac	737023	73.92	6.1885 MMac	737751

در شکل‌های ۱۶ و ۱۷ ارزیابی معیار دقت بر مبنای درصدهای گوناگونِ هرسِ شبکه Densenet121 با مجموعه داده‌هایCIFAR10و SVHN نشان داده شده است.



شکل ۱۶: دقت به دست آمده برای درصدهای گوناگونِ هرس در شبکه Densenet121و مجموعه داده CIFAR10



شکل ۱۷: دقت به دست آمده برای درصدهای گوناگونِ هرس در شبکه Densenet121 و مجموعه داده SVHN

۵- نتیجه گیری

در سال‌های اخیر در زمینه طراحی و ساخت شتاب‌دهنده‌های تقریبی به منظور افزایش سرعت محاسبات در شبکه‌های عصبی کانولوشنی عمیق، تحقیقات بسیاری انجام گرفته است. برای مثال، در برخی از این تحقیقات، کرنل‌ها و توابع فعالسازی و مقادیر بایاس با محاسبه گام کوانتیزه‌سازی (که براساس مقدار بیشینه مطلق و کمینه کردن مقدار خطای انتشارِ کوانیتزه‌سازی بدست می‌آید)، کوانتیزه شده‌اند [۲۹]. البته پژوهش‌های اخیر نشان می‌دهند که حذف فیلترهایی با بیشترین افزونگی در لایه‌های کانولوشن بر حذف آن‌هایی که کمترین اهمیت را دارند، ارجحیت دارد [۳۰]. در برخی از پژوهش‌ها برای هرس فیلترها از مجموع قدرمطلق مقادیر وزن‌ها در هر سطر از کرنل برای هرس استفاده می‌کنند. بدین ترتیب که سطری با بیشترین مقدار را حفظ کرده و بقیه سطرها را حذف می‌کنند و به‌جای مقادیر آن‌ها عدد ۰ قرار می‌دهند [۳۰]. همچنین در پژوهشی دیگر از الگویی برای هرس استفاده می‌شود که در آن کل ماتریس فیلتر به بلاک‌های کوچکتر تقسیم می‌شود و سپس کانال‌های ورودی در ماتریس فیلتر را به یک سری گروه سطری تقسیم‌بندی می‌کنند و به همین ترتیب کانال‌های خروجی را نیز به گروه‌های ستونی می‌شکنند [۳۱].

این شتاب‌دهنده‌های خاص منظوره با بهره‌گیری از روش‌های محاسبات تقریبی، می‌توانند سرعت محاسبات را تا چندین برابر افزایش دهند. برای مثال کوانتیزاسیون با کاهش تعداد بیت‌های لازم برایذخیره‌سازی پارامترهای شبکه‌های عصبی (از Float 32 به Int 8) باعث کاهش محسوس حجم شبکه و همچنین افزایش سرعت محاسبات می‌شود.

با توجه به گسترش روزافزون برنامه‌های کاربردی مرتبط با یادگیری عمیق، طراحی و توسعه ایده‌هایی که حجم پارامترهای قابل ذخیره این شبکه‌ها را کاهش داده و همچنین با بهره‌مندی از محاسبات تقریبی و نادیده گرفتن محاسبات غیر تاثیرگذار، سرعت محاسبات را افزایش دهد، بیشاز پیش حس می‌شود. از این رو در سال‌های

آینده بیشتر می‌توان از تکنیک‌های نرم‌افزاری محاسبات تقریبی مانند هرس کردن و تکنیک‌های سخت افزاری محاسبات تقریبی مانند ضرب‌کننده‌های تقریبی (برای انجام عمل کانولوشن در شبکه‌های عصبی کانولوشنی) و همچنین تکنیک‌های سخت‌افزاری– نرم‌افزاری محاسبات تقریبی مانند کوانتیزاسیون بهره برد.

۶- مراجع

[1] N. Manikandan, M. Priyanka2, Sasikumar, R. Muthaiah, "Approximation Computing Techniques to Accelerate CNN Based Image Processing Applications – A Survey in Hardware/Software Perspective", International Journal of Advanced Trends in Computer Science and Engineering, Volume 9, No.3, 2020

[2] Q. Xu, T. Mytkowicz and N. Sung Kim, "Approximate Computing: A Survey", IEEE DESIGN & TEST, 2015

[3] G. Rodrigues, F. L. Kastensmidt, and A. Bosio, "Survey on Approximate Computing and Its Intrinsic Fault Tolerance", Electronics, 2020

[4] J. Koomey, S. Berard, M. Sanchez, H. Wong, "Implications of historical trends in the electrical efficiency of computing", IEEE Ann. Hist. Comput. 33(3), 46–54, 2011

[5] A. Be´rut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dillenschneider, E. Lutz, "Experimental verification of Landauer/'s principle linking information and thermodynamics", Nature 483(7388), 187–189, 2012

[6] H. Esmaeilzadeh et al., "Neural acceleration for general-purpose approximate programs", MICRO, 2012

[7] H. B. Barua, K. C. Mondal, "Approximate Computing: A Survey of Recent Trends—Bringing Greenness to Computing and Communication", The Institution of Engineers (India), 2019

[8] A. Sampson, W. Dietl, E. Fortuna, D. Gnanapragasam, L. Ceze, and D. Grossman. EnerJ: Approximate Data Types for Safe and General Low-Power Computation", In Proc. International Conference on Programming Language Design and Implementation (PLDI), pp. 164-174, 2011.

[9] J. Bornholt, T. Mytkowicz, and K. S. McKinley, "A first-order type for uncertain data", In Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems, USA, 2014

[10] G. G. Henry, T. Park, R. E. Hooker, "VIA Alliance Semiconductor Co Ltd", 2019

[11] V. Pejovic, "Towards Approximate Mobile Computing, Newsletter, GetMobile: Mobile Computing and Communications, vol. 22, no. 4, 2018

[12] Z. Peng, X. Chen, C. Xu, N. Jing, X. Liang, C. Lu, L. Jiang, "AXNet: ApproXimate computing using an end-to-end trainable neural network", in Proceedings of the International Conference on Computer-Aided Design, p. 11, ACM, 2018

[13] A. Mant, R.S. Yazdanbakhsh, A. Park, J. Thwaites, B. Esmaeilzadeh, H. Hassibi, A. Ceze, L. Burger, "General-purpose code acceleration with limited-precision analog computation", In Proceedings of the International Symposium on Computer Architecture, 2014

[14] E. Wang, J. J. Davis, P. Y. K. Cheung, and G. A. Constantinides, "LUTNet: Rethinking Inference in FPGA Soft Logic", 2019 IEEE 27th Annu. Int. Symp. Field-Programmable Cust. Comput. Mach., pp. 26–34, 2019

[15] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning bothWeights and Connections for Efficient Neural Networks," Advances in neural information processing systems, pp. 1135–1145, 2015

[16] H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. I. E. Han, "A Review, Classification, and Comparative Evaluation of Approximate arithmetic circuits", ACM Journal on Emerging Technologies in Computing Systems (JETC), vol. 13, no. 4, pp. 1–34, 2017

[17] M. Imani, M. Samragh, Y. Kim, and S. Gupta, "RAPIDNN: In-Memory Deep Neural Network Acceleration Framework", arxiv preprint arxiv, 2018

[18] D. Hammerstrom, "A VLSI architecture for high-performance, low-cost, on-chip learning", 1990 IJCNN International Joint Conference on Neural Networks, 1990

[19] H. Li, H. Samet, A. Kadav, H. P. Graf, I. Durdanovic, "Pruning Filters for Efficient Convnets", International Conference on Learning Representations (ICLR), 2017

[20] A. Krizhevsky, G. Hinton, "Learning Multiple Layers of Features from Tiny Images", Technical report, University of Toronto, https://www.cs.utoronto.ca/~kriz/ learning-features-2009-TR.pdf.2009, 2009

[21] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning", NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011

[22] S. Li, W. Leng, "Real World Expression Recognition: A Highly Imbalanced Detection Problem", 9th IAPR International Conference on Biometrics (ICB), 2016

[23] K. Simonyan, A. Zesserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", International Conference on Learning Representations (ICLR), 2015

[24] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778., 2016

[25] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261-2269, doi: 10.1109/CVPR.2017.243., 2017

[26] G. S. Chung, C. S. Won, "Filter Pruning by Image Channel Reduction in Pre-trained Nonvolutional Neural Networks", Multimedia Tools and Applications, Springer, 2020

[27] J. Wang, T. Jiang, Z. Cui, Z. Cao, "Filter Pruning with a Feature Map Entropy Importance Criterion for Convolution Neural Networks Compressing", Elsevier, 2021

[28] S. Yeom, P. Seegerer, S. Lapuschkin, A. Binder, S. Wiedemann, K. Muller, W. Samek, "Pruning by Explaining: A Novel Criterion for Deep Neural Network Pruning", Pattern Recognition, Elsevier, 2021

[29] X. Sui, Q. Lv, L. Zhi, B. Zhu, Y. Yang, Y. Zhang, Z. Tan, "A Hardware-Friendly High-Precision CNN Pruning Method and Its FPGA Implementation", Sensors, MDPI, 2023

[30] Z. Wang, C. Li, X. Wang, "Convolutional Neural Network Pruning with Structural Redundancy Reduction", Computer Vision and Pattern Recognition, IEEE, 2021

[31] M. Lin, Y. Zhang, Y. Li, B. Chen, F. Chao, M. Wang, S. Li, Y. Tian, R. Ji, "1xN Pattern for Pruning Convolutional Neural Networks", Pattern Analysis and Machine Intelligence, IEEE Transactions, 2022.

[1] Pre-Trained
[2] Sequential
[3] Batch Normalization
[4] One-Shot