علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۲، ۱۳۹۴
صفحات ۶۰-۶۹
مقاله عادی

ارائه یک شبکه روی تراشه با کارآیی بالا و توان مصرفی کم برای شبکه‌های عصبی

نسرین اکبری بیتا دبیری مهدی مدرسی

دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران

چکیده

پیاده‌سازی سخت‌افزاری شبکه‌های عصبی به دلیل سفارشی‌سازی ساختار سخت‌افزار و حذف سربار نرم‌افزار سهم به‌سزایی در بهینه‌سازی توان و تاخیر انجام محاسبات عصبی دارد. نظر به اهمیت ارتباطات بین نورون‌ها در کارایی کلی شبکه‌های عصبی، در این مقاله یک هم‌بندی نوین شبکه روی تراشه جهت مدیریت ترافیک شبکه‌های عصبی ارائه شده است. این هم‌بندی، که براساس هم‌بندی معروف dragonfly ساخته شده است، برای انجام ترافیک چندپخشی و کاهش اتصالات بهینه گشته است. این هم‌بندی یک نمونه از هم‌بندی‌های سلسله مراتبی است و گره‌ها ابتدا در قالب گروه‌هایی تقیسم شده و در داخل هر گروه، از یک گذرگاه مشترک برای ارتباط آن‌ها استفاده می‌شود. سپس یک هم‌بندی سطح بالاتر گروه‌ها را به یکدیگر متصل می‌سازد. مشخصه اصلی هم‌بندی ارائه شده قطر کم و توانایی مناسب در انجام همه‌پخشی است. در این شبکه با انجام زمان‌بندی ارتباطات در زمان طراحی، از پیچیدگی مسیریاب‌ها کم شده که این امر زمینه‌ساز کاهش بیشتر توان و تاخیر شبکه می‌شود. این مقاله هم‌بندی پیشنهادی را با چند هم‌بندی پیشین مقایسه می‌کند که نتایج، نشان‌دهنده‌ی کاهش چشم‌گیر توان مصرفی و زمان تأخیر ارسال بسته‌ها و نیز افزایش گذردهی کلی شبکه تحت ترافیک چندپخشیِ شبکه‌های عصبی است.

کلمات کلیدی: شبکه روی تراشه، شبکه عصبی، راهگزینی مدار، کم‌توان، هم‌بندیdragonfly.

۱- مقدمه

پیاده‌سازی سخت‌افزاری شبکه‌های عصبی[1] یکی از موضوعاتی است که در سال‌های اخیر مورد توجه فعالان صنعتی و دانشگاهی مهندسی کامپیوتر قرار گرفته است. یک دلیل این امر نیاز روزافزون به عملکرد هوشمند در طیف وسیعی از سیستم‌های کامپیوتری، از حسگرها و سیستم‌های نهفته کوچک گرفته تا سرویس‌دهنده‌های بزرگ داده می‌باشد که شبکه‌های عصبی مصنوعی بهترین راه برای پیاده‌سازی الگوریتم‌های هوشمند شناخته می‌شوند.

اجرای یک شبکه عصبی بزرگ به‌صورت بی‌درنگ و با نرخ ورودی زیاد مستلزم انجام حجم زیادی از محاسبات سنگین اعشاری بوده که نیازمند به کارگیری یک ریزپردازنده و یا پردازنده‌ی گرافیکی قدرتمند است. از طرف دیگر، استفاده از پردازنده‌های قوی توان مصرفی و هزینه‌ی سیستم را به شدت بالا می‌برد که یک سامانه نهفته[2] مانند بینایی ربات جهت تشخیص چهره، چه از لحاظ هزینه‌ی تمام شده و چه از لحاظ توان مصرفی نمی‌تواند چنین هزینه‌ای را متحمل شود؛ بنابراین، استفاده از شتاب‌دهنده‌های سخت‌افزاری[3] که با پیاده‌سازی موازی شبکه عصبی بر روی سخت‌افزار (مثلا بر روی یک FPGA) و حذف سربار اجرای نرم‌افزار مدت زمان اجرا و توان مصرفی را کاهش می‌دهند، یکی از راه‌های مفید برای اجرای مناسب شبکه‌های عصبی است.

برای پاسخ به این نیاز، در سمت صنعت، شرکت‌های بزرگ طراحی و ساخت تراشه و سیستم‌های کامپیوتری مانند IBM، Intel، ARM، Xilinx، nVidia، Google، و بسیار شرکت‌های بزرگ و کوچک دیگر، اقدام به ساخت تراشه‌های خاص شبکه‌های عصبی کرده‌اند و یا هسته‌های پردازشی بر مبنای شبکه‌های عصبی را در محصولات خود مجتمع ساخته‌اند [۱][۲][۳][۴].

در سمت دانشگاه نیز گروه‌های معماری کامپیوتری بسیار در حال کار بر روی زمینه پیاده‌سازی سخت‌افزاری شبکه‌های عصبی هستند که این جریان در تعداد روز افزون مقالات در این زمینه در کنفرانس‌ها و ژورنال‌های معتبر رشته کامپیوتر منعکس می‌گردد [۵].

علاوه بر کاربرد در سیستم‌های هوشمند، در برخی پژوهش‌های قبلی نشان داده شده است که می‌توان از شبکه‌های عصبی برای پیاده‌سازی سریع توابع با بار

پردازشی سنگین استفاده کرد و یک تابع پیچیده محاسباتی را با مجموعه‌ای از عملیات ضرب و جمع پیاده‌سازی نمود. از این رو، استفاده از یک سخت‌افزار خاص منظوره برای محاسبات عصبی در پردازنده‌های همه منظوره و پردازنده‌های گرافیکی [۶] پیشنهاد شده است.

بیشتر سخت‌افزاره‌ای شتاب‌دهنده شبکه‌های عصبی به صورت یک تراشه بسیار هسته‌ای[۴] ساخته می‌شوند تا بتوانند از موازات ذاتی این مدل محاسباتی برای افزایش سرعت و برون‌دهی[۵] بهره ببرند. از آنجا که تعداد اتصالات بین نورون‌ها بسیار زیاد است (نورون‌های هر لایه، به صورت کامل به نورون‌های لایه‌ی بعد داده ارسال می‌نمایند) شبکه روی تراشه[۶] که بین هسته‌های پردازشی ایجاد می‌گردد، از تاثیر زیادی بر کارایی کل سیستم برخوردار است [۷].

به بیان دیگر، همان‌گونه که در بخش بعد گفته خواهد شد، هر دو عملیات ساده ضرب و جمع اعشاری که در یک واحد پردازشی شبکه عصبی انجام می‌شود نیازمند دریافت یک داده از واحدهای دیگر است که این امر نشان‌دهنده نسبت بالای ارتباطات به محاسبات در این مدل پردازشی است. این حجم بالای ارتباطات نسبت توان مصرفی و تاخیر شبکه روی تراشه را به کل توان و تاخیر سیستم افزایش می‌دهد و لذا بهینه‌سازی شبکه روی تراشه در این سیستم‌ها از اهمیت حیاتی برخوردار است.

در بسیاری از پژوهش‌های پیشین، از شبکه‌های روی تراشه راه‌گزینی بسته[۷] با هم‌بندی‌های[۸] معمولی مانند درخت[۹][۸]، توری[۱۰] [۹]، و Clos [۱۰]، به عنوان زیرساخت ارتباطی نورون‌ها استفاده شده است.

اما بسیاری از شبکه‌های عصبی که در سیستم‌های نهفته استفاده می‌شوند که این سیستم‌ها با یک نرخ ثابت ورودی‌ها را به شبکه عصبی اعمال می‌کنند. برای مثال، یک بخش هوشمند تشخیص چهره در یک سامانه نهفته، ورودی دوربین را با یک نرخ ثابت دریافت کرده و جهت تشخیص پارامترهای مورد نظر به بخش شبکه عصبی ارسال می‌دارد [۱۱]. از آنجا که در شبکه عصبی هم تمام نورون‌های که عملیات پردازشی یکسان (اما با عملوندهای متفاوت) را بر روی داده‌ها انجام می‌دهند، به شرط دریافت همزمان ورودی، خروجی خود را در یک زمان تولید می‌کنند.

این نظم در زمان‌بندی، در کنار نظم ذاتی در الگوی توزیع مکانی ترافیک در شبکه‌های عصبی (که نورون‌های هر لایه خروجی خود را برای تمام نورون‌های لایه بعد ارسال می‌نمایند) می‌تواند برای ساده‌سازی شبکه روی تراشه و در نتیجه کاهش مساحت، تاخیر، و توان مصرفی آن استفاده گردد.

در این مقاله، ما از این ویژگی نظم و پیش‌بینی‌پذیری در ترافیک تولیدی شبکه‌های عصبی استفاده کرده و یک شبکه روی تراشه ساده با زمانبندی ایستا[۱۱] جهت ایجاد ارتباط بین نورون‌ها ارائه می‌دهیم.

این شبکه با انجام یک زمان‌بندی ایستا برای ارسال داده‌های هر نورون به سایر نورون‌ها، نوعی راه‌گزینی مدار[۱۲] را برای ارتباطات بین نورون‌ها فراهم می‌آورد. با انجام زمانبندی در زمان طراحی دیگر نیاز به وجود مسیریاب‌های[۱۳] هوشمند در شبکه روی تراشه نیست، زیرا زمان ارسال داده توسط هر نورون و نیز مسیری که برای تحویل دادن داده به تمام نورون‌هایی که به آن داده نیاز دارند از قبل تعیین شده و در جدول زمانبندی در تمام مسیریاب‌ها قرار گرفته است.

با حذف هوشمندی از مسیریاب‌ها، توان و مساحت آنها به اندازه قابل توجهی کاهش می‌یابد زیرا دیگر نیازی به مسیریابی، داوری، تخصیص کانال مجازی، کنترل جریان، و ذخیره‌سازی بسته‌ها در هر مسیریاب نمی‌باشد. حذف این فعالیت‌ها از مسیریاب‌های شبکه روی تراشه افزون بر کاهش توان و مساحت به کاهش تاخیر ارتباطات هم کمک می‌کند، زیرا انجام این مراحل مستلزم صرف زمان است و در شبکه‌های روی تراشه، هر یک از مراحل فوق در یک مدت کلاک انجام می‌شود [۱۲]. در معماری ارائه شده در این مقاله، مانند شبکه‌های راه‌گزینی مدار، هر داده پس از رسیدن به هر مسیریاب، در کلاک بعدی به صورت بی‌درنگ طبق جدول زمانبندی داخل مسیریاب به گره بعدی ارسال می‌شود و از این رو تاخیرکمی به بسته‌ها اعمال می‌کند.

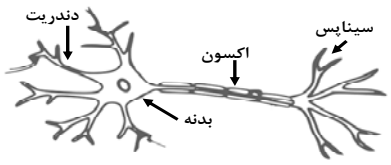
در این مقاله، شبکه روی تراشه پیشنهادی از هم‌بندیdragonfly استفاده می‌کند [۱۳]. Dragonfly یک هم‌بندی مدرن و با قطر[۱۴] کم است که در یکی از پیشرفته‌ترین نسل‌های ابر رایانه‌های شرکت Cray به کار رفته است [۱۴]. دلیل این انتخاب راحتی پیاده‌سازی آن در سطح مدار، قطر کم، و توانایی بالا در انجام ارتباطات چندپخشی[۱۵] است که آن را برای ترافیک شبکه‌های عصبی مناسب می‌کند.

در ادامه این مقاله، در بخش دوم به مروری بر مفاهیم پایه در شبکه‌های عصبی و ساختار آن‌ها خواهیم پرداخت. در بخش سوم، کارهای پیشین و مربوط به پیاده‌سازی سخت‌افزاری شبکه‌های عصبی مطرح شده است. بخش چهارم به معرفی و بررسی معماری شبکه روی تراشه پیشنهادی اختصاص دارد. در فصل بخش پنجم، نتایج حاصل از مقایسه طرح پیشنهادی با چند کار مطرح در این زمینه آمده است. در نهایت، فصل ششم به نتیجه‌گیری و جمع‌بندی در مورد معماری ارائه شده اختصاص دارد.

۲- پیش‌زمینه پژوهش

۲-۱- شبکه‌های عصبی زیستی

شبکه عصبی یک مدل محاسباتی بر مبنای یادگیری است که از سیستم عصبی موجودات زنده الهام گرفته شده است. نورون‌ها (یا سلول‌های عصبی) بخش اصلی مغز انسان را تشکیل داده و دارای یک شبکه‌ی پیچیده ارتباطی برای تبادل اطلاعات بین یکدیگر می‌باشند [۱۵]. شکل ۱ ساختار ساده شده یک نورون در سیستم عصبی انسان را نشان می‌دهد.



شکل ۱- ساختار کلی یک نورون در سیستم عصبی

هر نورون شامل سلول بدنه و دو نوع شاخه یعنی اکسون و دندریت برای دسترسی به خارج می‌باشد. هر نورون، سیگنال‌های نورون‌های دیگر را به شکل پالس الکتریکی از طریق دندریت (گیرنده) دریافت کرده و سیگنال‌های تولیدی خود را از طریق اکسون (فرستنده) به نورون‌های دیگر منتقل می‌کند. در این ساختار، در محلی به نام سیناپس، اکسون‌های هر نورون به دندریت‌های نورون دیگر متصل می‌شوند. در سیستم‌های زیستی، یادگیری با تنظیمات اتصالات سیناپسی که بین نورون‌ها قرار دارد به وجود می‌آید. به بیان دیگر، یادگیری با تغییر الگوی ارتباط بین نورون‌ها و شدت اثر آن‌ها بر یکدیگر ایجاد می‌گردد.

نکته جالب در مورد شبکه عصبی مغز انسان آن است که ارتباط بین نورون‌ها از طریق پالس‌هایی با فرکانس در حد چند صد هرتز انجام می‌شود که بارها از فرکانس کاری یک رایانه معمولی کمتر است. اما، با این حال، تصمیمات پیچیده‌ای مانند تشخیص چهره در مغز بسیار سریع و در حد چند میلی ثانیه انجام می‌گردد. دلیل این سرعت و کارآیی، پردازش موازی و توزیع‌شده در مقیاس بسیار بزرگ در مغز است: در مغز انسان بیش از ۱۰۰ میلیارد نورون وجود دارد که هریک از آن‌ها به طور متوسط با ۷ هزار نورون دیگر در ارتباط بوده و یک فرآیند پردازش را در کنار هم و با موازات فراوان به انجام می‌رسانند [۱۵].

۲-۲- شبکه‌های عصبی مصنوعی

انواع زیادی از مدل‌های محاسباتی تحت عنوان کلی شبکه‌های عصبی معرفی شده‌اند که هر کدام از بخشی از قابلیت‌های سیستم عصبی الهام گرفته و برای دسته‌ای از کاربردها قابل استفاده هستند. پر کاربردترین این مدل‌ها برای سیستم‌های نهفته هوشمند (مثلا در یک تشخیص‌دهنده تصویر) شبکه‌های عصبی پرسپترونی چند لایه (MLP)[۱۶] است که از یک لایه ورودی، n-1 لایه میانی (و یا پنهان) و یک لایه خروجی تشکیل شده است. بیشترین حالت پیاده‌سازی MLPها در حل مسائل کوچک دارای یک لایه ورودی، یک‌لایه میانی و یک لایه خروجی است. MLPها یکی از مهمترین رده‌های خانواده شبکه‌های عصبی پیشران[۱۷] هستند. در معماری پیشران، جریان داده فقط در یک جهت و از لایه ورودی به سمت لایه‌های پنهان و از آنجا به لایه خروجی حرکت می‌نماید.

به بیان دیگر، در این ساختار، ورودی نورون‌های لایه i خروجی نورون‌های لایه i-1 است (به‌غیر از اولین لایه پنهانی که به ورودی‌های شبکه عصبی متصل است). هر نورون در لایه‌های میانی و خروجی که در شکل ۲ نشان داده شده است، خروجی تمام نورون‌های لایه قبل را گرفته و مجموع حاصل ضرب ورودی‌ها در وزن مخصوص به هر ورودی را به صورت

$$y = \sum_{j=1}^{n} w_j x_j$$

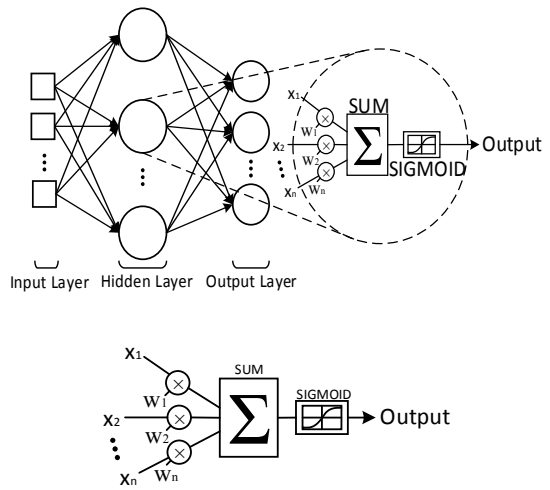
محاسبه می‌کند (ضرب نقطه‌ای آرایه وزن W و ورودی X در یکدیگر). در نهایت هم یک تابع فعال‌سازی[۱۸] (AF) جهت تولید نتیجه خروجی بر روی حاصل جمع نهایی (y) اعمال می‌شود. سپس جواب حاصل به تمام نورون‌های لایه بعد ارسال می‌گردد.

تابع سیگموئید[۱۹] یکی از پر استفاده‌ترین توابع فعال‌سازی در شبکه‌های عصبی مصنوعی بوده که تابعی هموار و اکیدا صعودی می‌باشد و به شکل زیر تعریف می‌شود (β پارامتر شیب است):

$$g(y)=1/(1+\exp(-\beta y))$$

در فرمول بالا، y ورودی تابع بوده که در واقع همان مجموع حاصل‌ضرب تمام ورودی‌ها در وزن‌های متناظرِ خود در یک نورون است [۱۶].

چون پیاده‌سازی این تابع مشکل است، معمولا در پیاده‌سازی سخت‌افزاری شبکه‌های عصبی، خروجی تابع سیگموید با یک جدول جستجو[۲۰] تقریب زده می‌شود.



شکل ۲- یک شبکه عصبی مصنوعی سه لایه (MLP) به همراه ساختار داخلی نورون

این بدان معنا است که به جای پیاده‌سازی مستقیم تابع، برخی مقادیر ورودی و خروجی تابع در یک جدول ذخیره شده و در زمان اجرا نزدیک‌ترین مقدار موجود در جدول به خروجی‌های مورد نظر یافته شده و جواب متناظر با آن به عنوان خروجی نورون در نظر گرفته می‌شود. به دلیل ماهیت تقریبی[۲۱] شبکه‌های عصبی، کاهش دقت پیاده‌سازی جدولی، به شرط وجود مقادیر کافی در جدول، در بسیار از مواقع باعث کاهش دقت محسوسی نخواهد شد.

در شبکه‌های عصبی مصنوعی وزن‌های هر نورون در فرآیندی به نام یادگیری یا آموزش[۲۲] تعیین می‌شود. توانایی یادگیری از اصلی‌ترین ویژگی‌های شبکه عصبی است. فرآیند یادگیری در حوزه شبکه‌های عصبی مصنوعی را می‌توان به شکل تنظیم وزن‌های شبکه برای حل یک مسئله خاص نگاه کرد، به گونه‌ای که شبکه بتواند به‌طور کارآمد آن مسئله بخصوص را با معماری ارائه شده در شکل ۲ حل کند.

الگوریتم‌های زیادی برای یادگیری شبکه‌های عصبی مصنوعی وجود دارد که سه مدل اصلی آن‌ها عبارتند از یادگیری نظارتی، یادگیری بدون نظارت و یادگیری ترکیبی. یادگیری نظارتی[۲۳]، که معمول‌ترین روش یادگیری بوده و در این مقاله استفاده شده است، بر پایه ارائه مجموعه بزرگی از ورودی‌ها و جواب صحیح متناظر با هر ورودی به شبکه‌های عصبی است. در طی این فرآیند، ورودی‌ها یکی یکی به شبکه داده شده و وزن تمام نورون‌ها طوری تعیین و تنظیم می‌شوند که شبکه بتواند نزدیک‌ترین جواب را به جواب صحیح تولید کند. پس از تکرار اعمال ورودی‌ها به مقدار کافی، شبکه قادر است جوابی مناسب به ورودی‌هایی که براساس آن‌ها آموزش دیده است و همچنی نورودی‌های جدید پیدا نماید.

هدف این مقاله پیاده‌سازی شبکه‌های عصبی بر روی شبکه بر روی تراشه است و توضیح کامل الگوریتم‌های یادگیری فراتر از بحث‌های مورد توجه ما به شمار می‌رود. مانند بیشتر پیاده‌سازی‌های سخت‌افزاری شبکه‌های عصبی، فرض ما بر این است که فاز آموزش شبکه و تعیین بردار وزن‌های هر کدام از نورون‌ها به‌صورت نابرخط[۲۴] و از قبل انجام شده است. یکی از قوی‌ترین ابزار موجود برای انجام فرآیند یادگیری و تنظیم وزن‌ها نرم‌افزار MATLAB است. در بیشتر سیستم‌ها، ابتدا آموزش شبکه عصبی توسط MATLAB انجام می‌شود و سپس وزن‌های به دست آمده جهت تولید خروجی نورون‌ها در زمان اجرا مورد استفاده قرار می‌گیرند.

یکی از ویژگی‌های مهم شبکه‌های عصبی خاصیت تقریبی بودن آنهاست. این به آن معنا است که این مدل محاسباتی قادر به یافتن جواب یک مسئله با دقت کامل نیست بلکه جواب را با درصدی از خطا محاسبه می‌کند. هر چه کیفیت و در برخی مسایل اندازه و عمق شبکه عصبی افزایش پیدا کند، درصد خطا در خروجی کمتر خواهد شد و تقریب نزدیکتری به جواب اصلی مسئله انجام می‌گیرد. اما این درصد به صفر نمی‌رسد و بنابراین این مدل محاسباتی برای کاربردهای نیازمند به جواب دقیق (مثلا در کاربردهای بانکداری و یا محاسبات دقیق علمی) مناسب نیست. با این وجود طیف گسترده‌ای از کاربردهای روزمره‌ی سیستم‌های کامپیوتری مانند کاربردهای پردازش سیگنال، چندرسانه‌ای، و تشخیص و تحلیل الگو نیاز به جواب با دقت کامل ندارند و درصدی از خطا در آن‌ها قابل قبول و در برخی موارد (مانند کاربردهای چندرسانه‌ای) نامحسوس است.

علاوه بر مدل شبکه عصبی MLP که یک خانواده از رده شبکه‌های عصبی مصنوعی است، شبکه‌های اسپایکی[۲۵] نیز یکی دیگر از رده‌های مهم شبکه‌های عصبی به شمار می‌روند [۱۷].

شبکه‌های اسپایکی مدل دقیق‌تری از شبکه‌های عصبی بیولوژیکی هستند. آنچه شبکه‌های عصبی مصنوعی (که در این بخش معرفی شدند) از سیستم عصبی زیستی الهام گرفته‌اند، نحوه به هم پیوستن نورون‌ها و مقادیر رد و بدل شده بین آنها بوده، اما زمان رسیدن ورودی برای آنها اهمیتی ندارد. ولی در دنیای واقعی، نورون‌های مغز با یکدیگر از طریق ارسال پالس‌های کوچکی ارتباط برقرار می‌کنند و هر نورون در ورودی خود قطاری از پالس‌ها را گرفته و در خروجی خود نیز

قطاری از پالس‌ها را با فاصله زمانی مشخص برای نورون بعدی ارسال می‌کند. در شبکه‌های پالسی، با استفاده از مدلی که در آن پارامتر زمان نیز وارد شده است و اطلاعات با تغییر فاصله زمانی و فرکانس قطار پالس خروجی کد شده است، عملکرد شبکه بسیار بیشتر از قبل به عملکرد مغز نزدیک می‌شود. از این رو، از این مدل برای تحقیقات مرتبط با مدل‌سازی و شبیه‌سازی مغز استفاده می‌شود.

تمرکز ما در این مقاله بر شبکه‌های عصبی مصنوعی است، اما معماری ارائه شده می‌تواند برای پیاده‌سازی شبکه‌های اسپایکی نیز به کار گرفته شود.

۲-۳- کاربردهای شبکه‌های عصبی مصنوعی

امروزه گستردگی کاربرد شبکه‌های عصبی به تمام زمینه‌های علمی و صنعتی که نیازی به تشخیص الگو، تحلیل، تصمیم‌گیری، تخمین، و پیش‌بینی دارند رسیده است. به طور خاص برخی از کاربردهای شبکه‌های عصبی در ادامه آمده است.

شناسایی و طبقه‌بندی الگو[۲۶]: هدف از این مسئله تشخیص این مطلب است که تعیین شود یک ورودی متعلق به کدام الگو و یا رده‌های موجود است. به عنوان مثال می‌توان از تشخیص صدای یک فرد از روی امواج صوتی، تشخیص چهره با استفاده از تطبیق دادن تصویر شخص با ورودی‌هایی که از قبل به سیستم داده شده‌اند، و یا شناسایی اثر انگشت نام برد. هم‌چنین از دیگر کاربردهای آن می‌توان به طبقه‌بندی امواج مغز و قلب غیره اشاره کرد.

خوشه‌بندی[۲۷] و دسته‌بندی: در خوشه‌بندی (که از آن به‌عنوان طبقه‌بندی الگو بدون ناظر نیز یاد می‌شود)، هیچ طبقه‌بندی از قبل مشخص شده‌ای وجود ندارد. یک الگوریتم خوشه‌بندی، شباهت بین الگوها را کشف کرده و آن‌ها را در یک خوشه قرار می‌دهد.

برازش توابع[۲۸]: شبکه عصبی قادر است تا تنها با برخورد با تعداد محدودی نمونه، یک قانون کلی از آن را به دست آورده و نتایج این آموخته‌ها را به سایر نمونه‌ها تعمیم دهد. فرض کنید n الگوی آموزشی (به شکل زوج مرتب ورودی- خروجی به‌ازای ورودی) از یک تابع ناشناخته و مجهول به فرم $\{(x1,y1), (x2,y2), \ldots, (xn,yn)\}$ داده شده باشد. تقریب زدن توابع یعنی پیدا کردن یک تخمین مناسب از تابع ناشناخته‌ی داده شده به شکلی که اگر ورودی دیگری غیر از این مجموعه به آن بدهیم، این تابع تقریبی جوابی نزدیک به تابع ناشناخته را به ما بدهد. دقت این تابع تقریبی می‌تواند با توجه به ورودی‌های آموزشی از قبل داده شده به آن و هم‌چنین پیچیدگی خود تابع، کم یا زیاد شود. شبکه عصبی یکی از قوی‌ترین ابزارها برای تقریب زدن توابع است. علاوه بر سیستم‌های هوشمند، از این خاصیت شبکه عصبی برای پیاده‌سازی کم‌هزینه‌تر توابع سنگین با بار پردازشی زیاد استفاده می‌شود.

نوع خاص این مسئله پیش‌بینی و تخمین است که اگر مقدار یک پدیده در n واحد زمانی اخیر داده شده باشد، مقدار نمونه بعدی که در زمان آینده یعنی n+1 خواهد آمد را پیش‌بینی کند.

بهینه‌سازی[۲۹]: طیف گسترده‌ای از مسائل موجود در علوم مهندسی، پزشکی و اقتصاد را می‌توان در قالب مسئله بهینه‌سازی مطرح کرد که سعی در بهینه کردن یک تابع هزینه/خطا دارند. شبکه‌های عصبی یکی از ابزارهای مفید برای حل این مسائل به شمار می‌روند.

۲-۴- شبکه‌های عصبی سخت‌افزاری در صنعت

یکی از شرکت‌هایی پیش‌رو در فناوری که در تلاش برای طراحی تراشه‌ای مانند مغز انسان می‌باشد، شرکت IBM است. این شرکت در قالب پروژه SyNAPSE تراشه‌هایی را تولید کرده است که تماماً شبکه عصبی بوده و برای کاربردهای نیازمند به یادگیری و هوش مصنوعی استفاده می‌شوند [۱]. به عنوان نمونه، تراشه TrueNorth یکی از محصولات این پروژه است. این تراشه ۴۰۹۶ هسته پردازشی را در قالب یک شبکه روی تراشه جای داده است. هر هسته پردازشی می‌تواند تا ۲۵۶ نورون را در خود جای دهد و بنابراین کل تراشه توانایی اجرایی کمی لیون نورون را دارد. این تراشه خاص منظوره فقط شامل شبکه عصبی است و به دلیل حذف سربار نرم‌افزار و با وجود آن‌که از بیش از ۵ میلیارد ترانزیستور ساخته شده است دارای توان مصرفی کمتر از یک وات است [۱].

شرکت گوگل نیز در اوایل سال ۲۰۱۶ از ساخت یک پردازنده خاص منظوره براساس شبکه‌های عصبی خبر داد [۲]. این پردازنده که TPU نام دارد در کنار پردازنده‌های اصلی در مراکز داده قرار گرفته و انجام پردازش‌های مربوط به هوش مصنوعی و تحلیل داده‌ها که توسط شبکه‌های عصبی به خوبی قابل پیاده‌سازی است را بر عهده دارد. همچنین شرکت مایکروسافت از تراشه‌های قابل پیکربندی مجدد (FPGA) برای پیاده‌سازی پردازنده‌های هوشمند بر پایه شبکه عصبی استفاده می‌کند. طبق گزارش‌های این شرکت، از این پردازنده‌ها در عملیات‌های مختلف تحلیل داده و نیز جستجوی هوشمند در موتور جستجوی Bing استفاده می‌شود [۳].

شرکت nVidia نیز که پیشگام طراحی و ساخت پردازنده‌های گرافیکی است قابلیت‌های سخت‌افزاری اجرای سریع شبکه‌های عصبی را در جدیدترین نسل پردازنده‌های خود، یعنی پردازنده‌های گرافیکی GP100 قرار داده است. طبق گزارش این شرکت، پردازنده جدید می‌تواند شبکه‌های عصبی را ۱۷ بار سریع‌تر از پردازنده نسل قبلی خود اجرا نماید [۴].

Zeroth یک پلاتفرم برای اجرای شبکه‌های عصبی در دستگاه‌های قابل حمل هوشمند (مثلا تبلت‌ها) بوده که شامل یک سخت‌افزار تخصصی اجرای شبکه عصبی و کتابخانه‌های نرم‌افزاری مربوطه است [۱۹]. این پلاتفرم محصول شرکت Qualcomm است که یکی از پیشروترین سازندگان پردازنده برای تلفن‌های هوشمند و تبلت‌ها به شمار می‌رود. این شرکت Zeroth را در آخرین نسل پردازنده‌های موبایل خود (Snapdargon 8x) و برای تشخیص چهره و نیز مدیریت هوشمند مصرف باتری دستگاه قرار داده است.

۳- کارهای پیشین

تاکنون، کارهای پژوهشی و صنعتی بسیار زیادی بر روی پیاده‌سازی سخت‌افزاری شبکه‌های عصبی هم به صورت دیجیتال و هم آنالوگ انجام شده است.

در این میان هدف بیشتر کارهای انجام شده، علاوه بر افزایش سرعت اجرای شبکه‌های عصبی، مدیریت ترافیک بین نورون‌ها، کاهش توان مصرفی، کاهش نیاز به پهنای باند حافظه بوده است.

در زمینه توان مصرفی، بیشتر کارهای موجود بر دو مشخصه مهم شبکه‌های عصبی تمرکز کرده‌اند: (۱) تحمل‌پذیری در برابر کاهش دقت و (۲) حجم زیاد محاسبات تکراری.

در [۲۰] وجود مقدار قابل توجهی عملیات ریاضی تکراری در شبکه‌های عصبی گزارش شده است که می‌توان آنها را جهت ذخیره انرژی حذف کرد. با توجه به این مشاهده، یک شتاب‌دهنده شبکه عصبی با نام CORN پیشنهاد شده است که از با استفاده مجدد از محاسبات گذشته به نورون‌ها اجازه می‌دهد تا نتایج محاسبات تکراری خود را با یکدیگر به اشتراک بگذارند. این استفاده مجدد از محاسبات به طور میانگین ۲۶٪ از انرژی شبکه‌های عصبی را در مقایسه با برخی طراحی‌های مدرن کم توان کاهش می‌دهد.

استفاده از ویژگی ذاتی شبکه‌های عصبی در تحمل‌پذیری در برابر کاهش دقت یکی دیگر از ابزارهای مهم برای کاهش توان مصرفی آن‌ها است. به طور کلی

روش‌های موجود در این دسته شامل روش‌های کاهش پهنای بیتی (دقت) اعداد در محاسبات عصبی [۲۱] و یا استفاده از واحدهای محاسبات تقریبی است [۲۲]. در برخی کارها نشان داده شده است که حتی با نصف کردن دقت بیتی اعداد (از اعداد ممیز ثابت ۱۶ بیتی به ۸ بیتی)، خطای خروجی شبکه عصبی برای بسیار از کاربردها قابل قبول است. با این وجود، بسیار از روش‌ها به دنبال کاهش هوشمندانه دقت بیتی هستند. مثلا، در [۲۱]، تاثیر کاهش دقت تک تک نورون‌ها در نتیجه خروجی شبکه عصبی اندازه‌گیری شده و پهنای بیتی هر نورون براساس حساسیت خروجی نسبت به آن تعیین می‌گردد.

پهنای باند حافظه یکی دیگر از ملاحظات اساسی در طراحی سخت‌افزاری شبکه‌های عصبی است [۲۳]. دلیل این امر آن است که یک شبکه عصبی تعداد زیادی داده ورودی و وزن را از حافظه واکشی می‌کنند که نیازمند پهنای باند زیادی از حافظه است. به طور خاص، یک شبکه کانولوشنی بزرگ نوعی که برای پردازش تصویر ساخته می‌شود (مانند پردازش‌هایی که در عینک هوشمند گوگل انجام می‌شود) نیاز به چند مگابایت حافظه جهت نگهداری وزن‌ها و ۳۰ تا ۶۰۰ هزار عملیات به ازای هر پیکسل از یک تصویر دارد [۲۳] که فراهم آوردن این پهنای باند مسئله‌ای چالش برانگیز برای یک سیستم نهفته است.

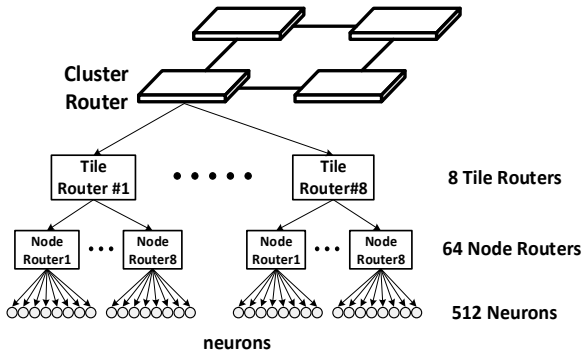
روش ارائه شده در [۲۴] یکی از شاخص‌ترین تلاش‌ها برای کاهش نیاز به پهنای باند در شبکه‌های عصبی هستند. در این مقالات یک شتاب‌دهنده برای شبکه‌های عصبی کانولوشنی در مقیاس بزرگ، با تأکید بر تأثیری که حافظه در کارایی و توان مصرفی طراحی شتاب‌دهنده‌ها دارد طراحی شده است. با بهینه‌سازی الگوی دسترسی به حافظه، این شتاب‌دهنده که با فنّاوری ۶۵ نانومتری پیاده‌سازی شده است قابلیت اجرای ۴۵۲ میلیون عملیات ممیز شناور در یک مساحت کوچک ۳ میلیمتر مربعی با مصرف انرژی ۴۸۵ میلی وات را دارد. سپس نشان داده شده است که روش ارائه شده نسبت به یک پردازنده مدرن برداری ۱۲۸ بیتی که با فرکانس ۲ گیگاهرتز کار می‌کند حدود ۱۱۷ برابر سریع‌تر بوده و ۲۱ برابر انرژی مصرفی کمتری دارد.

همان‌گونه که گفته شد، به دلیل ترافیک زیاد، شبکه ارتباطی بین واحدهای پردازشی در یک سخت‌افزار چندهسته‌ای شتاب‌دهنده شبکه عصبی نقش زیادی در کارآیی کلی سیستم دارد. از این رو طراحی مناسب این بخش مورد توجه پژوهشگران زیادی قرار گرفته است.

از همبندی توری دو بعدی در بسیاری از پیاده‌سازی‌های قبلی شبکه‌های عصبی در مقیاس صنعتی [۹] و پژوهشی [۷] استفاده شده است. در [۱۰] از شبکه میان‌ارتباطی چند مرحله‌ای Clos جهت مدیریت مؤثرتر ارتباطات میان نورونی در داخل تراشه استفاده شده است. در این مقاله، نشان داده شده است که به دلیل تشابه میان الگو ترافیک شبکه‌های عصبی (ترافیک بر پایه چندپخشی چند سطحی) و ویژگی‌های همبندی‌های میان‌ارتباطی چند مرحله‌ای، این همبندی‌ها گزینه مناسبی برای استفاده در شبکه‌های عصبی هستند. شبکه Clos، یکی از مهم‌ترین کلاس‌های همبندی‌های میان ارتباطی چند مرحله‌ای است و نتایج ارزیابی‌های مقاله نشان می‌دهد که می‌تواند ترافیک چندپخشی شبکه‌های عصبی را بهتر از توپولوژی توری که در بسیاری از پیاده‌سازی‌های شبکه عصبی مورد استفاده قرار گرفته است مدیریت کند و میانگین تأخیر کمتری را تولید می‌کند.

در پروژه‌ی بزرگ اروپایی EMBRACE، یک معماریِ مبتنی بر شبکه‌های روی تراشه‌ی سلسله مراتبی[۳۰] (H-NoC) برای شبکه‌های عصبی ارائه شده است [۸]. در شبکه‌های میان‌ارتباطیِ سلسله مراتبی، ترافیک شبکه به دو بخش ترافیک محلی (درون هر ناحیه) و ترافیک سرتاسری (بین ناحیه‌ها) تقسیم می‌شود و از یک همبندیِ ترکیبی به نام درخت- توری، برای بهره‌وری بهتر الگوریتم‌های چندپخشی سود می‌برد. بلوک‌های اصلی تشکیل‌دهنده‌ی هر تراشه در این پروژه، Cluster، Tile و Neuron نام دارند که در واقع لایه‌های مختلف سلسله مراتب این شبکه هستند. هر واحد Neuron میتواند ۱۰ نورون از شبکه عصبی را در خود جای دهد. سپس، در هر واحد Tile ۱۰ عدد واحد Neuron و در هر واحد Cluster ۴ عدد واحد Tile قرار می‌گیرند. در مجموع ۴۰۰ نورون در هر واحد Cluster قرار دارد که در قالب شبکه سلسله مراتبی به هم متصل هستند. شکل ۳ معماری قرارگیری خوشه‌ها را در همبندی درختی EMBRACE نشان می‌دهد [۸].

در این سیستم، برای میزبانی از شبکه‌های عصبی بزرگتر، واحدهای Cluster با همبندی توری مدور به همدیگر متصل می‌شوند.



شکل ۳- همبندی سلسله مراتبی درختی در EMBRACE [۸]

۴- معماری ارائه شده

بیشترین پیاده‌سازی‌های موجود شبکه روی تراشه موجود برای شبکه‌های عصبی از همبندی پایه یا تغییریافته‌ی توری استفاده می‌کنند. همبندی یک شبکه توری شامل یک ماتریس چند بعدی از گره‌ها است که توسط یک ساختار ارتباطی منظم، که هر گره را به طور مستقیم به گره‌های قبلی و بعدی خود در هر بعد متصل می‌کند، تشکیل شده است. با افزایش اندازه شبکه، میانگین تاخیر شبکه توری به طور قابل توجهی افزایش پیدا می‌کند که ناشی از عدم‌وجود مسیر مستقیم بین گره‌هایی است که در ساختار ماتریسی از هم فاصله زیادی دارند. به علاوه، مدل ترافیک شبکه‌های عصبی به صورت چندپخشی است که همبندی توری برای این نوع ترافیک مناسب نمی‌باشد. بنابراین استفاده از همبندی‌هایی با قابلیت ذاتی چندپخشی می‌تواند گزینه مناسب‌تری برای پیاده‌سازی ارتباط داخلی درون شبکه‌های عصبی باشد.

در این بخش، یک معماری سفارشی موازی برای شبکه‌های عصبی پیشنهاد می‌کنیم که از هم‌بندی میان ارتباطی dragonfly برای ارتباط بین نورون‌ها استفاده می‌کند.

همبندی میان ارتباطی dragonfly یکی از هم‌بندی‌های جدید شبکه‌های میان ارتباطی است که در ابر رایانه‌های مدرن سری XC محصول شرکت Cray به کار رفته است [۱۴]. ویژگی مهم این همبندی فراهم آوردن قطر کم با برقراری هوشمندانه‌تر اتصالات[۳۱] بین گره‌ها نسبت به توری است.

این همبندی در شکل ۴ نشان داده شده است. در این همبندی، گره‌های شبکه در قالب چندین گروه دسته‌بندی می‌شوند که تمام مسیریاب‌های داخل هر گروه به وسیله یک شبکه کاملا پیوسته[۳۲] با یکدیگر در ارتباط هستند. درجه هر مسیریاب a+p+h است که دارای p درگاه[۳۳] برای اتصال به پردازنده، a درگاه برای اتصال به سایرمسیریاب‌های هم‌گروهی، h درگاه برای اتصال به سایر مسیریاب‌ها در گروه‌های دیگر است. تعداد اتصالات بین گروه‌ها بسته به تعداد گروه‌ها (که می‌تواند متغیر باشد) است: بین هر دو گروه باید دست کم یک اتصال وجود داشته باشد و در صورت کم بودن گروه‌ها می‌توان چند اتصال بین هر دو گروه داشت. در شکل ۴، یک شبکه با a=3، p=2 و h=2 نشان داده شده و ۵ گروه از پردازنده‌ها را به هم متصل کرده است.

در این شکل، از آنجا که هر گروه دارای ۴ مسیریاب و هر مسیریاب دارای ۲ اتصال به بیرون گروه است (مجموعاً ۸ اتصال)، هر گروه می‌تواند ۲ اتصال به هریک از ۴ گروه دیگر در این شبکه داشته باشد.

در این همبندی، مسیریابی از هر گره به گره دیگر در سه گام انجام می‌پذیرد. در گام اول، بسته به آن گره‌ای در گروه خود ارسال می‌شود که به گروه مقصد متصل است. نظر به همبندی کاملا پیوسته داخل گروه، این کار به صورت بیشینه نیاز به پیمایش یک گام دارد. اگر خود گره مبدا مستقیماً به گروه مقصد وصل باشد، در این مرحله نیاز به حرکت بسته نیست. سپس بسته، اتصال بین گروه‌ها را پیموده و به یک مسیریاب در گروه مقصد می‌رسد و از آنجا با بیشینه یک گام به مقصد اصلی تحویل داده می‌شود.

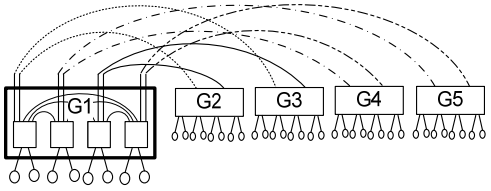
در این مقاله ما با تغییراتی این همبندی را برای پیاده‌سازی روی تراشه مناسب سازی کرده و یک مدل همه‌پخشی برای آن ارائه می‌کنیم.

اولین مشکل این همبندی برای پیاده‌سازی روی تراشه شبکه کاملا پیوسته درون هر گروه است که هم نیاز به سیم‌بندی زیاد و متقاطع دارد و هم تعداد درگاه‌ها و در نتیجه اندازه کراسبار و تعداد میانگیرها[۳۴] را افزایش می‌دهد.

برای رفع این مشکلات ما از یک گذرگاه مشترک[۳۵] برای اتصالات داخل گروه استفاده می‌کنیم. گذرگاه مشترک برای اتصال تعداد زیادی از عناصر به دلیل مقیاس ناپذیری توان و پهنای باند مناسب نیست. اما در این سیستم، ما با محدود کردن عناصر پردازشی داخل هر گروه، مانع از بروز مشکل مقیاس‌پذیری گذرگاه مشترک می‌شویم.

در این سیستم با فرض داشتن ساده‌ترین حالت سیستم، یعنی p=1 و h=1، هر مسیریاب دارای یک درگاه برای اتصال به پردازنده، یک درگاه گذرگاه مشترک برای اتصال به سایر مسیریاب‌های هم‌گروهی، و یک درگاه برای اتصال به سایر مسیریاب‌ها در گروه‌های دیگر است. این ساختار ساده، مساحت و نیز توان مصرفی شبکه روی تراشه تا حد زیادی کاهش می‌دهد. می‌توان با افزایش پارامترهای این همبندی، یعنی a، p، و h، مصالحه‌ای بین مساحت و برون‌دهی شبکه ایجاد کرد. ما در این مقاله فقط همبندی پایه با p=1، h=1، و a=1 را ارزیابی کرده و ارزیابی این مصالحه را به کارهای بعدی موکول می‌کنیم.

در این همبندی می‌توان با یک زمانبندی ایستا، همانگونه که در فصل‌های قبل به آن اشاره شد، بخش کنترلی مسیریاب‌ها را نیز بسیار ساده کرده و با یک مسیریاب ساده به برون‌دهی قابل قبول رسید.



شکل ۴- همبندی dragonfly

همبندی میان ارتباطی dragonfly به کار رفته در ابر رایانه Cray یک نمونه از شبکه‌های درجه بالا[۳۶] است. در همبندی‌های میان ارتباطی درجه بالا، مسیریاب‌ها دارای درگاه‌های بیشتری نسبت به شبکه‌هایی درجه پایین (مانند توری) هستند، اما پهنای بیتی درگاه‌ها کمتر است. درگاه‌های بیشتر در شبکه‌های درجه بالا، منجر به تولید قطر کمتر می‌شود و از این رو، این شبکه‌ها قابلیت مقیاس‌پذیری بهتری از نظر تاخیر زمانی دارند. اما از سوی دیگر، زمانی که پهنای بیتی اتصال کمتر از عرض بسته باشد تعداد فلیت‌های یک بسته و در نتیجه تاخیر ارسال بیشتر می‌شود. اما این مشکل در شبکه‌های عصبی که دارای بسته‌های ذاتی کوچکی هستند مشکلی محسوب نمی‌شود: بسته‌های تولید شده توسط شبکه عصبی حامل خروجی نورون‌ها است که یک عدد اعشاری است ۱۶ یا ۳۲ یا بیتی به همراه چند بیت کنترلی است که در برابر بسته‌های ۵۱۲ بیتی (شامل یک بلوک حافظه) در بسیاری از سیستم‌های چندپردازنده‌ای روی تراشه بسیار کوچک‌تر است. اما در پیاده‌سازی خاص ما از این همبندی، با جایگزینی بخشی از همبندی با یک گذرگاه مشترک، ضمن سود بردن از قطر کم، مشکل نیاز به درگاه‌های زیاد هم حل می‌شود.

۴-۱- نگاشت[۳۷] شبکه‌های عصبی روی dragonfly

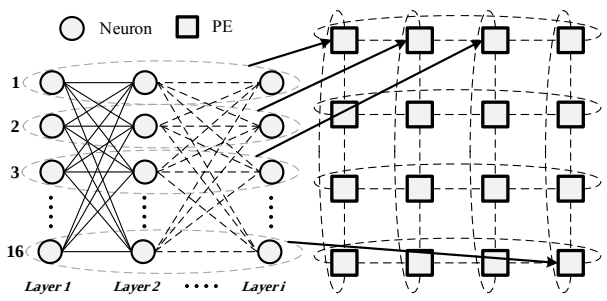
در این بخش، از آنجا که نحوه نگاشت نورون‌ها تاثیر مستقیمی بر زمان‌بندی شبکه دارد، ابتدا یک روش برای نگاشت نورون‌ها بر رویگره‌های هم‌بندی dragonfly ارائه کرده و سپس یک زمان‌بندی مناسب برای ارتباطات بین نورون‌ها ایجاد می‌کنیم.

از آنجایی که بسیار محتمل است که در یک شتاب‌دهنده‌ی سخت‌افزاری، تعداد نورون‌ها بسیار بیشتر از تعداد هسته‌های پردازشی باشد، لازم است تا با تکنیکی مناسب، نورون‌ها دسته‌بندی شده و هر دسته روی یک هسته‌ی پردازشی نگاشته شود. در بسیاری از کارهای قبلی، نورون‌های یک لایه را در یک گره یا خوشه‌ای از گره‌ها قرار می‌دادند. در این حالت، برخی از گره‌ها که فقط شامل نورون‌های لایه‌ی ورودی هستند، فقط فرستنده بوده و برخی دیگر که نورون‌های لایه‌های میانی را در بردارند، در یک فاز فرستنده (به لایه بعدی) و در فاز دیگر گیرنده (از لایه قبلی) خواهند بود. لذا بار ترافیکیِ شبکه به طور متعادل در تمام گره‌ها پخش نخواهد شد.

برای رفع این مشکل در معماری پیشنهادی، ما نورون‌هایی از لایه‌های مختلف را در یک دسته قرار می‌دهیم. با استفاده از این تکنیک، هر دسته شامل نورون‌هایی از تمام لایه‌های شبکه عصبی خواهد بود. بنابراین، اگر هر دسته روی یک هسته‌ی پردازشی نگاشت داده شود، آن هسته‌ی پردازشی در هر لحظه می‌بایست هم داده‌های تولید شده از نورون‌های خود را ارسال و هم داده‌های رسیده از دسته‌های دیگر (دیگر هسته‌های پردازشی) را دریافت کند، در نتیجه در یک زمان، هم فرستنده و هم‌گیرنده خواهد بود. در این مدل نگاشت، تمام گره‌ها یکسان بوده و کار یکسان و مشابه‌ای انجام می‌دهند.

همان‌طور که در شکل ۵ نمایش داده شده است، تکنیک دسته‌بندی بر روی یک شبکه‌ی چندلایه‌ی که هر لایه از آن ۱۶ نورون دارد، اعمال شده و دسته‌ها به طور منظم روی عناصر پردازشیِ یک شبکه روی تراشه با هم‌بندیِ توری نگاشت داده شده‌اند.

در این مدل نگاشت، لزومی بر برابری تعداد نورون‌های لایه‌ها وجود ندارد و اگر تعداد نورون‌های لایه‌های ورودی و خروجی کمتر از لایه‌های میانی باشد (که معمولا هست) می‌توان تعداد مختلفی از نورون‌های لایه در هر دسته داشت. همچنین می‌توان در صورت لزوم چند دسته را در یک گره شبکه روی تراشه جای داد.



شکل ۵- دسته‌بندی نورون‌های برای نگاشت بر روی شبکه روی تراشه

با این کار، بار کاریِ تمام هسته‌های پردازشی یکنواخت خواهد بود و در نتیجه ترافیک تولید شده در شبکه به صورت متعادل در سرتاسر شبکه توزیع خواهد شد. ارتباطات بین نورون‌ها در این مدل تبدیل به یک مدل ترافیکی ویژه می‌شود که در آن هر گره داده‌های خود را به همه گره‌های دیگر ارسال می‌کند و از تمام گره‌های دیگر داده دریافت می‌نماید.

۴-۲- زمان‌بندی ارتباطات شبکه‌های عصبی روی dragonfly

در این بخش، یک زمان‌بندی ایستای برای ارتباطات بین نورون‌ها در شبکه روی تراشه dragonfly طراحی می‌گردد.

در این روش با استفاده از زمان‌بندی ایستا تمام اجزای هوشمند مسیریاب‌ها، مانند واحدهای مسیریابی و داوری و کنترل جریان و نیز میان‌گیرها حذف شده و در نتیجه، توان، تاخیر، و مساحت مرتبط به آنها از مسیریاب حذف می‌گردد. در عوض، مسیری که بسته‌ها در هر مسیریاب طی می‌کنند از قبل تعیین شده و در قالب یک جدول اتصال ورودی- خروجی در مسیریاب‌ها قرار می‌گیرد. این جدول تعیین می‌کند که در هر سیکل، کدام ورودی به کدام خروجی متصل باشد تا ارتباطات در نظر گرفته شده برقرار گردد.

با اعمال روش دسته‌بندی و نگاشت نورون‌ها، هر عنصر پردازشی در هر لحظه باید خروجی نورون‌های خود را برای دیگر نورون‌ها ارسال کند و متقابلاً، بقیه‌ی عناصر پردازشیِ شبکه نیزخروجی خود را برای نورون‌های آن عنصر می‌فرستند.

این زمان‌بندی در چند سیکل انجام می‌شود و در پایان آن، تمام گره‌ها داده تمام گره‌های دیگر را دارند. در ادامه این روال را برای همه پخشی داده گره شماره ۱ هر گروه در یک شبکه dragonfly با پنج گروه چهار گره‌ای دنبال می‌کنیم.

- سیکل ۱: ابتدا گره شماره ۱ در هر گروه داده خود را از طریق گذرگاه مشترک به تمام گره‌های دیگر گروه ارسال می‌نماید. در پایان این سیکل تمام گره‌های هر گروه داده گره شماره ۱ را خواهند داشت.
- سیکل ۲: تمام گره‌های هر گروه از طریق اتصال خارجی خود (که به یک گروه دیگر وصل است) داده‌ی گره شماره ۱ گروه خود را برای گروه دیگری که به آن متصل هستند ارسال می‌نماید. برای مثال، اگر به گروه شماره ۱ توجه کنیم، گره‌های ۱ تا ۴ داده‌ی گره ۱ گروه خود را که در سیکل قبل دریافت کرده‌اند را به ترتیب برای گره مربوطه (گره شماره ۱، همان‌طور که در شکل ۴ مشاهده می‌شود) در گروه‌های ۱ تا ۴ ارسال می‌دارند. در پایان این سیکل، اگر مجددا به گروه شماره ۱ توجه کنیم، گره‌های ۱ تا ۴ این گروه، داده‌ی گره ۱گروه‌های به ترتیب ۱ تا ۴ را دارند.
- سیکل‌های ۳ تا ۶: گره‌های شماره ۱ تا ۴ در هر گروه داده دریافتی از گروه دیگر را از طریق گذرگاه مشترک به تمام گره‌های دیگر گروه خود ارسال می‌نماید. در پایان این سیکل تمام گره‌های هر گروه داده گره شماره ۱ تمام گروه‌های دیگر را خواهند داشت.

روال بالا، داده گره ۱ تمام گروه‌ها را در شش سیکل به تمام شبکه ارسال می‌دارد. با تکرار این روال برای چهار دفعه (در دفعه i، داده گره iام هر گروه)، می‌توان داده تمام گره‌ها را به تمام گره‌های دیگر رسانید.

۵- ارزیابی

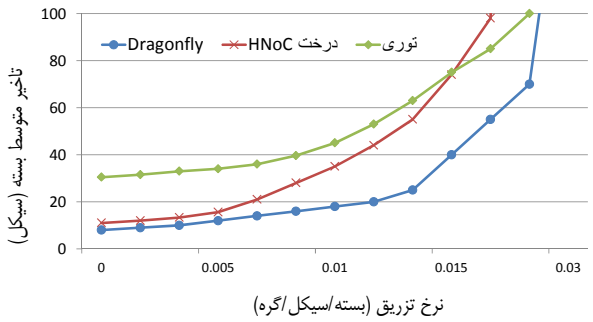
جهت ارزیابی معماری‌های ارائه شده، از چندین مجموعه داده‌ی معروف که مربوط به زمینه‌ی یادگیری ماشین است، به‌عنوان محک جهت ارزیابی استفاده می‌کنیم. برای مشخصات محک‌ها نیز ازکارهای قبلی [۱۰] [۲۵] بهره می‌گیریم. ساختار شبکه عصبی متناسب با هر محک را در جدول ۱ آورده شده است. لازم به ذکر است که شبکه به‌دست آمده برای هر برنامه جهت رسیدن به کمترین خطای خروجی با کوچک‌ترین تعداد نورون، پس از اجراها و جستجوهای مختلف در نظر گرفته می‌شود. برای ساخت و آموزش شبکه عصبی نیز، از نرم‌افزار MATLAB استفاده می‌کنیم.

جدول ۱- برنامه‌های محک و داده‌های ورودی

برنامه محک	ساختار لایه‌ها
Performance Modeling and Evaluation	128:256:128
Object Classification	3072:3000:10
Census Data Analysis	14:12:12:2
Hand Writing Digit Recognition	784:700:10

برای ارزیابی، علاوه بر همبندی dragonfly، همبندی درختی H-Noc در EMBRACE [۸] و همبندی توری را جهت انجام مقایسه در نظر می‌گیریم. برای هر سه شبکه، ۱۲۸ گره (مسیریاب) در نظر گرفته شده است. تمام درگاه‌های ورودی مسیریاب‌ها مجهز به میان‌گیرهای ۸ فیلیتی هستند. در شبکه‌های مورد مقایسه (توری و درخت) تمام مسیریاب‌ها بسته‌ها را بعد از انجام عمل ذخیره کردن در میانگیر، مسیریابی، و داوری در سه سیکل به گره بعد ارسال می‌کنند، اما در صورت بازنده شدن بسته‌ها در داوری، این زمان افزایش خواهد یافت.

در همبندی dragonfly، شبکه به ۱۶ گروه ۸ گرهی تقسیم می‌شود. هر مسیریاب یک درگاه به پردازنده، یک درگاه به گذرگاه مشترک (برای ارتباطات درون‌گروهی) و دو درگاه برای ارتباط با سایر گروه‌ها دارد. در همبندی درخت، ۸ واحد پردازشی به یک مسیریاب نورون، ۸ مسیریاب نورون به یک مسیریاب tile، و دو مسیریاب tile به وسیله یک مسیریاب خوشه متصل شده‌اند. در توری، ۴ گره به یک مسیریاب متصل هستند که یک توری ۴×۸ را تشکیل می‌دهند.



شکل ۶- میانگین تاخیر همه‌پخشی توری،dragonfly، و درخت

در آزمایش‌های انجام شده، ما فرض کرده‌ایم که گره‌های پردازشی عملیات ریاضی بر روی یک ورودی را به اندازه‌ای سریع انجام می‌دهند که تمام داده دریافتی به‌وسیله شبکه بدون تأخیر صف و در یک سیکل انجام می‌شود.

شکل ۶ کارایی همه‌پخشی سه شبکه با نشان دادن میانگین زمان تأخیر همه پخشی آن‌ها در ترافیک یکنواخت را مقایسه کرده است. تأخیر همه‌پخشی یک بسته از زمان تولید شدن تا زمانی که بسته در آخرین مقصد دریافت می‌شود است. همان‌گونه که شکل نشان می‌دهد، قطر کم شبکه dragonfly منجر به تأخیر کمتری برای همه‌پخشی در شبکه می‌شود. در این نمودار، بدترین عملکرد متعلق به شبکه توری است که بیشترین قطر را دارد.

پس از بررسی تحت ترافیک ساختگی[۳۸]، شبکه پیشنهادی را بر روی شبکه‌های عصبی واقعی لیست شده در جدول ۱ مورد ارزیابی قرار دادیم. ما تمامی محک‌ها را بر روی سه شبکه مورد بحث با ۱۲۸ گره پیاده‌سازی کردیم. در مورد برخی از شبکه‌های عصبی، لازم می‌شود که چندین گروه نورون را بر روی یک گره شبکه

نگاشت کرد که در این مورد از روش نگاشت اشاره شده در بخش قبل استفاده شده است.

جدول ۲ کارایی این سه هم‌بندی را بر حسب توان عملیاتی مورد مقایسه قرار داده است. ما توان عملیاتی را به عنوان نرخ ورودی ای از شبکه‌های عصبی (تعداد ورودی‌های شبکه عصبی در هر سیکل) تعریف می‌کنیم در آن نرخ، شبکه روی تراشه می‌تواند با تاخیر زیر ۱۰۰ سیکل در حال کار باشد. با افزایش نرخ ورودی از این مقدار، شبکه با افزایش تاخیر مواجه شده و به اشباع می‌رود (از کار می‌افتد). لذا معیار تعریف شده در این بخش برای توان عملیاتی، بیشینه توان شبکه روی تراشه برای پذیرفتن داده جدید است و معیاری مناسب برای مقایسه شبکه‌ها به شمار می‌رود.

برای مقایسه بهتر، اعداد جدول ۲ براساس نتایج به دست آمده از توری نرمال‌سازی شده است. همانطور که جدول ۲ نشان می‌دهد، شبکه ارائه شده باز هم به دلیل قطر کمتر و مسیریاب‌های سریع‌تر توان عملیاتی بیشتری نسبت به شبکه‌های مورد مقایسه ارائه می‌دهد. در این جدول نشان داده شده است که توان عملیاتی هم‌بندی ارائه شده به ترتیب بیش از ۸۹ و ۴۵ درصد بالاتر از توری و درخت است.

جدول ۲- مقایسه توان عملیاتی هم‌بندی‌های در نظرگرفته شده

برنامه محک	توان عملیاتی نرمال‌سازی شده (ورودی در هر سیکل)		
	dragonfly	درخت	توری
Performance Modeling	1.7	1.3	1
Object Classification	2.4	2.0	1
Census Data Analysis	1.4	1.1	1
Hand Writing Digit Recognition	1.9	1.5	1

ارزیابی توان مصرفی برای یکی از محک‌های نمونه (Object Classification) توان مصرفی ۸۱۲ میلی وات برای درخت، ۹۴۵ میلی وات برای توری، و ۵۸۰ میلی‌وات برای dragonfly را نشان می‌دهد. همان‌گونه که بحث شد، قطر کم و مسیریاب‌های ساده دلیل برتری dragonfly بر شبکه‌های دیگر از نظر مصرف توان است. اعداد توان فقط شامل توان ارتباطات است و توسط ابزار تخمین توانDSENT [۲۶] در فناوری ۴۵ نانومتر به‌دست آمده است.

۶- نتیجه‌گیری و کارهای آینده

در ساله‌ای اخیر حجم قابل‌توجهی از پژوهش بر روی پیاده‌سازی سخت‌افزاری شبکه‌های عصبی انجام شده است. دلیل عمده این توجه نیاز به وجود هوشمندی در بسیاری از سیستم‌های کامپیوتری، از یک سیستم نهفته کنترلی کوچک گرفته تا سرویس‌دهنده‌های بزرگ در مراکز داده، است. شبکه‌های عصبی یکی از کاراترین روش‌ها در پیاده‌سازی سیستم‌های دارای یادگیری و هوشمندی هستند. سرعت انجام محاسبات و توان مصرفی در بسیاری از این سیستم‌ها یک پارامتر محدود کننده است، بنابراین پیاده‌سازی سخت‌افزاری شبکه‌های عصبی با سفارشی‌سازی ساختار سخت‌افزار و حذف سربار نرم‌افزار سهم به‌سزایی در بهینه‌سازی توان و تاخیر انجام محاسبات در این مدل محاسباتی دارد. با توجه به اهمیت ارتباطات در پیاده‌سازی سخت‌افزاری شبکه‌های عصبی، در این مقاله یک شبکه روی تراشه با هم‌بندی dragonfly برای این سیستم‌ها ارائه کردیم. این شبکه دارای قابلیت مناسبی برای پیاده‌سازی همه‌پخشی است و با ارائه یک زمان‌بندی ایستا دارای مسیریاب‌هایی تا حد ممکن ساده است. نتایج ارزیابی نشان‌دهنده کاهش قابل توجه توان مصرفی و افزایش کارآیی نسبت به شبکه‌های روی تراشه ارائه شده قبلی است.

یکی از راه‌ها برای ادامه این کار، یافتن یک چینش مناسب در سطح مدار برای این شبکه است تا بتوان آن را با مساحت قابل قبول در سطح تراشه پیاده‌سازی کرد. طرح جاری برای یک شتاب‌دهنده ساخته شده است که با توجه پیش‌بینی‌پذیری ارتباطات در این سیستم‌ها می‌توان با انجام زمان‌بندی ایستا از پیچیدگی مسیریاب‌ها کاست. اما استفاده از این هم‌بندی به عنوان یک زیرساخت کلی با قابلیت همه‌پخشی مناسب برای اجرای بدون زمان‌بندی شبکه‌های عصبی و نیز یافتن مدل‌هایی با کارآیی بالاتر برای نگاشت نورون‌ها بر روی گره‌های شبکه از دیگر راه‌ها برای ادامه پژوهش جاری است.

مراجع

[1] IBM SyNAPSE project, http://www.research.ibm.com, Jan. 2015.

[2] Tensor Processing Unit Architecture, https://cloudplatform.googleblog.com, Jan. 2015.

[3] Microsoft to Accelerate Bing Search with Neural Network, http://blog.microsoft.com, Jan. 2015.

[4] GP100 Pascal Whitepaper, http://www.nvidia.com, Jan. 2015.

[5] J. Hauswald, and et. al., "DjiNN and Tonic: DNN as a service and its implications for future warehouse scale computers," Proc. *International Symposium on Computer Architecture*, 2015.

[6] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Neural acceleration for general-purpose approximate programs," *Proc. International Symposium on Microarchitecture*, pp. 449–460, 2012.

[7] D. Vainbrand, and R. Ginosar, "Network-on-chip architectures for neural networks," *Proc. Network-on-chip Symposium*, 2010.

[8] S. Carrillo, and et. al., "Scalable hierarchical network-on-chip architecture for spiking neural network hardware implementations," *IEEE Transactions on Parallel and Distributed Systems*, vol. 45, no. 22, 2012.

[9] E. Painkras, and et. al., "SpiNNaker: A 1-W 18-Core System-on-chip for massively-parallel neural network simulation," *IEEE Journal of Solid-State Circuits*, pp. 1943-1953, 2013.

[10] A. Yasoubi, R. Hojabr, H. Takshi, M. Modarressi, and M. Daneshtalab, "CuPAN: high throughput on-chip interconnection for neural networks," *Proc. International Conference of Neural Information Processing*, 2015.

[11] D. Y. Kim, and et. al., "A neural network accelerator for mobile application processors," in *IEEE Transactions on Consumer Electronics*, vol. 61, no. 4, pp. 555-563, 2015.

[12] W. J. Dally, and B. Towles, *Principles and practices of interconnection networks*, Morgan-Kaufmann Publishers, 2004.

[13] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," *Proc. International Symposium on Computer Architecture*, Beijing, pp. 77-88, 2008.

[14] B. Alverson, "Cray high speed net working," *Proc. 20th Annual Symposium on High-Performance Interconnects (HOTI)*, 2012.

[15] S. Haykin, *Neural networks: A comprehensive foundation*, Upper Saddle River, NJ, USA: Prentice-Hall, 2008.

[16] A. K. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," *Journal of Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[17] W. Maass, and C. M. Bishop, *Pulsed neural networks*. MIT press, 2001.

[18] P. Merolla, and et. al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668-673, 2014.

[19] https://www.qualcomm.com/invention/cognitive-technologies/machine-learning, Jan. 2015.

[20] A. Yasoubi, R. Hojabr, and M. Modarressi, "Power-efficient accelerator design for neural networks using computation reuse," in *IEEE Computer Architecture Letters*, 2015.

[21] Q. Zhang, T. Wang, Y. Tian, F. Yuan, and Q. Xu, "ApproxANN: an approximate computing framework for artificial neural network," *Proc. Design, Automation & Test in Europe Conference*, 2015.

[22] S. Venkataramani, A. Ranjan, K. Roy, and A. Raghunathan, "Axnn: Energy-efficient neuromorphic systems using approximate computing," *Proc. International Symposium on Low Power Electronics and Design*, 2014.

[23] Y. Chen, and et. al., "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," *Proc. ISSCC*, pp. 262-263, 2016.

[24] T. Chen, Z. Du, N. Sun, J. Wang, C. Wu, Y. Chen, and O. Temam, "A high-throughput neural network accelerator," *IEEE Micro*, vol. 35, no. 3, pp. 24-32, 2015.

[25] A Firuzan, M. Modarressi, and M. Daneshtalab, "A reconfigurable network-on-chip for efficient implementation of neural networks," *Proc. International Symposium on Reconfigurable Communication-centric Systems-on-Chip*, 2015.

[26] C. Sun, and et. al., "DSENT: A tool connecting emerging photonics with electroniccs for opto-electronic networks-on-chip modeling," *Proc. Network-on-chip Symposium*, 2012.

نسرین اکبری دانشجوی کارشناسی‌ارشد رشته مهندسی کامپیوتر گرایش معماری سیستم‌های کامپیوتری از دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران است. ایشان دوره کارشناسی خود را در سال ۱۳۹۴ در رشته مهندسی کامپیوتر- سخت‌افزار از دانشگاه صنعتی خواجه نصیرالدین طوسی به اتمام رسانیده‌اند. شبکه‌های روی تراشه، مالتی‌مدیا بر روی شبکه، و بیوانفورماتیک از زمینه‌های تحقیقاتی مورد علاقه ایشان است.

آدرس پست الکترونیکی ایشان عبارت است از:

nasrin.akbari@ut.ac.ir

بیتا دبیری دانشجوی کارشناسی‌ارشد رشته مهندسی کامپیوتر گرایش معماری سیستم‌های کامپیوتری از دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران است. ایشان دوره کارشناسی خود را در سال ۱۳۹۳ در رشته مهندسی کامپیوتر-نرم‌افزار از دانشگاه شهید رجایی به اتمام رسانیده‌اند. شبکه‌های روی تراشه، شبکه‌های کامپیوتری و سیستم‌های نهفته بی‌درنگ از زمینه‌های تحقیقاتی مورد علاقه ایشان است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

bita.dabiri@ut.ac.ir

مهدی مدرسی مدرک کارشناسی خود را در سال ۱۳۸۲ از دانشگاه صنعتی امیرکبیر و مدرک‌های کارشناسی‌ارشد و دکترا را در سال‌های ۱۳۸۴ و ۱۳۸۹ در مهندسی کامپیوتر از دانشگاه صنعتی شریف دریافت کرده است. وی از سال ۱۳۹۱ عضو هیئت علمی گروه معماری سیستم‌های کامپیوتری در دانشکده مهندسی برق و کامپیوتر دانشگاه تهران است. وی همچنین به عنوان پژوهشگر در دانشگاه پلی‌تکنیک لوزان (EPFL) در سوییس (۲۰۰۹ تا ۲۰۱۰) و پژوهشگر غیرمقیم در پژوهشگاه دانش‌های بنیادی (IPM) (از ۱۳۸۵ تا کنون) فعالیت کرده است. زمینه‌های پژوهش مورد علاقه ایشان معماری کامپیوتر، شبکه روی تراشه، پردازش موازی، و سخت‌افزارهای یادگیر است که تاکنون بیش از ۷۰ مقاله درباره آنها منتشر کرده است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

modarressi@ut.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۴/۰۷/۰۹

تاریخ اصلاح: ۱۳۹۴/۰۹/۱۳

تاریخ قبول شدن: ۱۳۹۴/۰۹/۳۰

نویسنده مرتبط: دکتر مهدی مدرسی، دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران، تهران، ایران.

[1]Neural Network
[2]Embedded System
[3]Hardware Accelerator
[4]Many-Core Chip
[5]Throughput
[6]Network-on-Chip
[7]Packet-Switch

[8]Topology
[9]Tree
[10]Mesh
[11]Static Scheduling
[12]Circuit Switching
[13]Router
[14]Diameter
[15]Multicast
[16]Multi-Layer Perceptron
[17]Feed-Forward
[18]Activation Function
[19]Sigmoid
[20]Lookup Table
[21]Approximate
[22]Training
[23]Supervised Learning
[24]Offline
[25]Spiking Neural Networks
[26]Pattern Recognition and Classification
[27]Clustering
[28]Fitting
[29]Optimization
[30]Hierarchical
[31]Link
[32]Fully Connected
[33]Port
[34]Buffer
[35]Bus
[36]High-Radix
[37]Mapping
[38]Synthetic