زمستان ۱۴۰۰، دوره ۶، شماره ۴، پیاپی ۲۳
DOR :20.1001.1.2538161.1400.6.4.2.6

دریافت مقاله: ۱۴۰۰/۰۲/۱۳
پذیرش مقاله: ۱۴۰۰/۰۷/۲۰

نوع مقاله: پژوهشی

بهینه‌سازی برون‌سپاری محاسباتی شبکه‌های عصبی عمیق برای تشخیص فعالیت انسانی

پریسا سعادتی
دانشجوی کارشناسی ارشد گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بجنورد، بجنورد، ایران
پست الکترونیکی: parisa.saadati@stu.ub.ac.ir

حمید فدیشه‌ای*
استادیار گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه بجنورد، بجنورد، ایران
پست الکترونیکی: fadishei@ub.ac.ir

چکیده

با توجه به محدودیت‌های منابع محاسباتی موجود در دستگاه‌های تلفن همراه، این دستگاه‌ها در اجرای برخی وظایف روزمره با چالش مواجه هستند. یک راه‌حل برای این مشکل، برون‌سپاری است که در آن، دستگاه پردازش خود را برای اجرا روی ابر محاسباتی ارسال می‌کند. مقاله حاضر با تمرکز بر روی کاربرد تشخیص فعالیت انسانی، روش‌هایی برای کاهش حجم داده‌های ارسالی وظایف به ابر، با تکیه بر نقاط مطلوب در معامله بین دقت استنتاج یادگیری و هزینه ارتباطات برون‌سپاری ارائه می‌کند. سه روش پیشنهادی کاهش تعداد نمونه‌های داده، کاهش دقت اعشار نمونه‌های داده و فشرده‌سازی نمونه‌های داده ارائه شده است که در روش اول نمونه‌های داده قبل از ارسال به صورت یکی در میان یا بیشتر حذف شده و در سمت ابر با تخمین درون‌یابی مجددا بازیابی می‌شوند. در روش کاهش دقت اعشار، نمونه‌های داده قبل از ارسال با یک نگاشت به عدد صحیح با تعداد بیت محدود تبدیل و در سمت ابر با نگاشت معکوس به صورت تخمینی بازیابی می‌شوند. در روش فشرده‌سازی، نمونه‌های داده قبل از ارسال، با یک الگوریتم فشرده‌سازی سبک دلتا به یکی از دو روش با اتلاف یا بدون اتلاف فشرده می‌شوند. نتایج عملی نشان می‌دهد اگر چه روش‌های کاهش تعداد نمونه و کاهش دقت اعشار سبب کاهش حجم داده ارسالی بدون تأثیر قابل توجه بر روی دقت تشخیص می‌شوند، روش کاهش دقت اعشار به دلیل میزان کاهش بیشتر حجم داده نسبت به روش کاهش تعداد نمونه‌ها برتری دارد. ضمناً روش فشرده‌سازی دلتا به اندازه دو روش دیگر موثر نمی‌باشد.

واژه‌های کلیدی: برون‌سپاری محاسباتی، شبکه‌های عصبی عمیق، تشخیص فعالیت انسانی

مقدمه

در سال‌های اخیر شاهد افزایش چشمگیر تحقیقات

* نویسندهٔ مسئول

و توسعه شبکه‌های تلفن همراه بوده‌ایم. با پیشرفت در پایانه‌های تلفن همراه و از آنجا که تلفن‌های هوشمند محبوبیت زیادی پیدا کردند، برنامه‌های جدید تلفن همراه مانند تشخیص چهره، پردازش تصویر، بازی‌های تعاملی و واقعیت افزوده توجه زیادی را به خود جلب کرده و انتظار از دستگاه‌های تلفن همراه برای اجرای برنامه‌های سنگین‌تر در حال افزایش است [۱]. اکنون کاربران تلفن همراه استفاده‌های روزمره فراوانی مانند جستجو در میان آهنگ‌ها، انجام بازی‌های ویدیویی، ضبط، ویرایش و بارگذاری فیلم، تجزیه و تحلیل مجموعه عکس‌های خود، فهرست‌بندی محتوا و مدیریت امور مالی دارند [۲]. با وجود نقش پررنگ این دستگاه‌ها در زندگی افراد، اجرای برنامه‌های پیچیده در دستگاه‌های تلفن همراه به دلیل محدودیت منابع آن‌ها از قبیل ظرفیت حافظه، سرعت پردازش گرافیکی و قدرت باتری چالش برانگیز است [۳].

دستگاه‌های تلفن همراه دارای قدرت محاسباتی نسبتاً ضعیف، محدودیت باتری و منابع سخت افزاری می‌باشند. همچنین برنامه‌های کاربردی دستگاه‌های تلفن همراه معمولاً نیاز به محاسبات فشرده و مصرف انرژی بالایی دارند. با توجه به محدودیت‌های منابع محاسباتی موجود در دستگاه‌های تلفن همراه، ممکن است این دستگاه‌ها نتوانند برنامه کاربردی را به‌طور موثر اجرا کنند [٤]. امروزه به‌خاطر افزونی کاربردهایی با حجم بالای پردازش، نیاز به محیط و منابع قدرتمندی که بتواند این محاسبات سنگین را به عهده بگیرد، می‌باشد. یک راه حل برای این مشکل به نام برون‌سپاری وجود دارد که در آن دستگاه‌های کم قدرت مانند تلفن همراه، اشیاء اینترنتی و... پردازش خود را بر روی ابر محاسباتی ارسال کرده و به کارسازهای ابری می‌سپارند [۵]. برون‌سپاری می‌تواند موجب صرفه‌جویی در مصرف انرژی و بهبود عملکرد شود و همین‌طور می‌تواند قابلیت محاسبات سیستم‌های تلفن همراه را تقویت کند [۲].

بسیاری از کاربردهای محاسباتی ذکر شده از یادگیری ماشین سود می‌برند که بنا به تعریف، قابلیت بهبود کارکرد یک برنامه بدون دخالت برنامه‌نویس، بلکه از طریق تجربه می‌باشد [٦]. یادگیری عمیق نوع خاص و پرکاربرد یادگیری ماشین می‌باشد که اخیراً مورد توجه محققان قرار گرفته است و قابلیت خود را در تشخیص با دقت بالا در کاربردهایی مثل پردازش تصویر، تشخیص فعالیت انسانی، پردازش گفتار و... نشان داده است [۷]. برای یادگیری عمیق معمولاً از شبکه‌های عصبی عمیق استفاده می‌شود که یک نمودار جریان داده هستند و از تعدادی لایه تشکیل شده‌اند، هر کدام از لایه‌ها عملکرد خود را بر روی داده‌های ورودی انجام می‌دهند و داده‌های خروجی را به لایه بعدی منتقل می‌کنند [۸]. شبکه‌های با حافظه دور و نزدیک (Long Short Term Memory) نوع خاصی از شبکه‌های عصبی عمیق هستند که توانایی یادگیری وابستگی‌های کوتاه مدت و بلند مدت را دارند. این شبکه‌ها برای اولین بار در سال ۱۹۹۷ توسط Hochreiter و Schmidhuber معرفی شدند [۹]. این شبکه‌ها به دلیل ساختار خاص و دارا بودن حافظه، در مورد داده‌هایی مثل مسئله تحقیق حاضر که وابستگی زمانی دارند معمولاً دقت بهتری نسبت به سایر انواع شبکه‌های عصبی نشان می‌دهند [۱۰].

عملیات محاسبات شبکه‌های عصبی عمیق شامل دو مرحلهٔ آموزش و استنتاج می‌باشد که مسئله برون‌سپاری می‌تواند در هر دو مرحله مذکور مطرح شود. در مرحلهٔ آموزش با استفاده از داده‌های ورودی از قبل برچسب خورده، پارامترهای شبکه‌های عصبی عمیق (نظیر وزن یال‌ها) تعیین می‌شود، تا شبکه عصبی عمیق بتواند در مرحلهٔ کاربرد بر روی داده‌هایی که تا به حال مشاهده نشده است، استنتاج انجام دهد. پردازش هر لایه می‌تواند به‌عنوان یک عملیات بردار در نظر گرفته شود که پارامترهای آن به‌طور تکراری در حالی که شبکه‌های عصبی عمیق با داده‌های برچسب خورده آموزش می‌یابند، به‌روز می‌شوند. با توجه به این‌که کاربردهای عملی گوشی‌های هوشمند در مرحله استنتاج نقش پررنگ‌تری دارد، اکثر

محققین بر روی برون‌ســپاری محاسبات مرحله استنتاج تمرکز دارند. از طرف دیگــر، مرحلهٔ آموزش به دلیل نیاز گسترده به منابع محاسباتی به‌طور معمول در کارسازهای قدرتمنــد انجــام می‌شــود [۳]. دلیل دیگــر کم‌رنگ بودن برون‌سپاری مرحلهٔ آموزش این است که پس از انجام این مرحله، پارامترهای هر لایه ثابت هستند، بنابراین شبکه‌های عصبی عمیق مادامی که داده‌های آموزش تغییر نیافته‌اند، همــان پارامترها را برای اســتنتاج روی داده‌های ورودی بــه کار می‌برند. برخی از محققان بر روی برون‌ســپاری همه محاســبات شــبکه‌های عصبی عمیق یا بخش‌هایی از آن تمرکز کــرده و راه‌حل‌هایی برای این کار ارائه داده‌اند، تا بر موانعی از قبیل محدودیت‌های اســتفاده از باتری در سمت تلفن همراه و منابع محدود محاسباتی آن فایق آیند. برون‌ســپاری شــبکه‌های عصبی عمیق معمولاً با معامله همراه است و در ازای صرفه‌جویی در زمان اجرا و انرژی مصرفی، هزینه‌ای در کاهش دقت استنتاج پرداخت می‌گردد [۱۱]، این معامله تمرکز مقاله حاضر است و در این تحقیق بر روی اولین نقطه ورودی شبکه‌های عصبی عمیق یعنی داده‌های ارسال شده به ابر در مرحلهٔ استنتاج تمرکز شده است تا حجم آن‌ها به طرق مختلفی کاهش یابد.

کاربردی که در این تحقیق به آن پرداخته شــده است، تشخیص فعالیت‌های انسانی می‌باشد. تشخیص فعالیت‌های انسانی برای ارائه خدمات در دنیای اینترنت اشیاء ضروری اســت. با توجه به همه‌گیر بودن، قابلیت ســنجش و قدرت پردازش، تلفن‌های هوشــمند مدرن به دســتگاه‌های مورد توجه برای تشــخیص فعالیت انسانی تبدیل شده‌اند. با این حال، ظرفیت محــدود باتری و منابع تلفن هوشــمند مانع بهره‌برداری از چنین قدرت ســنجش و پردازش می‌شــود [۱۲]. تشخیص فعالیت‌های انسانی امکان می‌دهد بسیاری از فعالیت‌هــای بدنی را که یک کاربر تلفن هوشــمند انجام می‌دهد (نظیر پیــاده‌روی، دویدن و...) براســاس حرکات کاربر تشــخیص داده شــود. در این تحقیــق از داده‌های جمع‌آوری‌شده از حسگرهای شتاب‌ســنج، مجموعه داده WISDM استفاده شده است [۱۳]. این مجموعه داده حاوی بیش از دو میلیون نمونه شــامل پنج فعالیــت پیاده‌روی، دویدن، فعالیت پله‌ها، نشســتن و ایســتادن می‌باشد. این تحقیق، تلاشی برای بهینه‌سازی ارتباطات در برون‌سپاری مسئله تشخیص فعالیت انسانی با گوشی هوشمند از طریق بررســی معامله بین حجم داده ارســالی و دقت تشخیص است. نوآوری‌های این مقاله به شرح زیر می‌باشد:

- مطالعه روش‌های مختلف کاهش حجم داده‌های ارسالی به ابر برای برون‌سپاری استنتاج شبکه‌های عصبی عمیق.
- پیشــنهاد ســه روش کاهش تعداد نمونه‌ها، کاهش دقت اعشار و فشرده‌ســازی نمونه‌های داده برای بهینه‌سازی برون‌سپاری مذکور.
- بررسی و تحلیل روش‌های مختلف کاهش حجم داده‌های ارســالی برای یافتن نقاط مطلــوب در معامله بین دقت و هزینه ارتباطات.

۲- کارهای گذشته

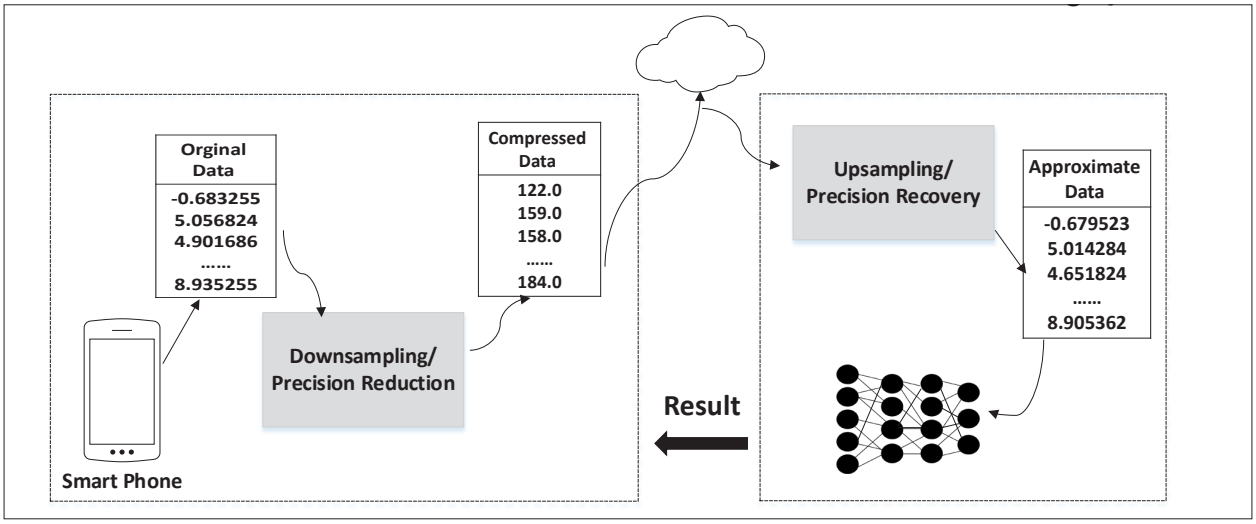
از آنجا که در ســال‌های اخیــر برنامه‌های تلفن همراه به‌طور فزاینــده‌ای مورد توجه همه جانبه قــرار گرفته‌اند و عملکــرد قدرتمندی را در دســتگاه‌های تلفن همراه ارائه کرده‌انــد و با توجــه به این‌که دســتگاه‌های تلفن همراه با چالش‌های محاســبات محدود و کم‌قدرت روبرو هســتند [۱٤]، توســعه‌دهندگان معمولاً تمایل دارند فرآیند محاسبه را به ابر منتقــل کنند. تحقیقات زیادی برای بهینه‌ســازی برون‌سپاری شبکه‌های عصبی عمیق انجام شده است. یک رویکرد کلی برای برون‌ســپاری شبکه‌های عصبی عمیق، بخش‌بندی می‌باشــد که در آن گراف شــبکه‌های عصبی عمیق به چند بخش تقســیم می‌شود و در هنگام اجرا و با در نظر گرفتن وابســتگی‌ها، فرایند محاسبه بر روی ابر و یا به صورت محلی انجام می‌شــود. در پژوهش Huang و همکارانش [٤]، برای اســتفاده بهینــه از منابع موجود، بار محاســباتی بین دستگاه، لبه و ابر توزیع شده است که این امــر می‌تواند تأخیر ارتباطی را به‌طور قابل توجهی کاهش

دهـد. بخش‌بندی می‌تواند با در نظـر گرفتن خصوصیات پویای بارکاری و شرایط باری لبه در هنگام برون‌سپاری و زمان اجرا همراه باشـد که علاوه بر بهبود زمان تأخیر، بهره‌وری انرژی در سیستم‌هایی با منابع محدود را افزایش دهد [۳]. برای غلبه بر محدودیت‌های منابع در پژوهش [۱۵] روشـی برای بخش‌بندی شبکه‌های عصبی عمیق با هدف کاهش زمان محاسبه و افزایش استفاده از منابع در شبکه ارائه شده است.

در پژوهـش [۱۶] که بر یادگیـری عمیق برای اینترنت اشیاء در محیط محاسـباتی لبه تمرکز دارد، به دلیل توان پـردازش محدود گره‌هـای لبه، یک روش برون‌سـپاری محاسباتی جدید برای بهینه‌سازی عملکرد برنامه یادگیری عمیق ارائه شده که محافظت از حریم خصوصی کاربر در برون‌سپاری داده‌ها را نیز به همراه دارد. روش IONN [۱۷] مدل شبکه عصبی عمیق کارخواه را به چند قسمت تقسیم و سپس یک به یک در کارساز لبه بارگذاری می‌کند. کارساز با رسیدن هر قسمت از شبکه عصبی عمیق، مدل را به‌صورت تدریجـی ایجاد می‌کند و به کارخواه اجازه می‌دهد تا حتی قبـل از بارگذاری کل مدل، برون‌سـپاری جزئی از اجرای شـبکه عصبی عمیق را آغاز کنـد، این امر مصرف انرژی و بهره‌وری کوئری در برون‌سپاری محاسبات شبکه‌های عصبی عمیق از دسـتگاه تلفن همراه به کارساز را بهبود می‌بخشـد. همچنین بخش‌بندی بین ابر، لبه و دستگاه‌های انتهایی با استفاده از مقیاس‌پذیری شبکه‌های عصبی عمیق به لحاظ اندازه و مقطع جغرافیایی آن‌ها، هزینه‌های ارتباطی را کاهش می‌دهد و سبب بهبود دقت می‌شود [۱۸].

روش کلی دیگـر که در مقابل رویکرد بخش‌بندی قرار می‌گیرد، انتقال کل سـاختار شـبکه عصبی به ابر اسـت، به‌عنوان مثال پژوهش [۱۹] از رویکرد انتقال کل سـاختار شبکه عصبی از کارخواه به یک کارساز لبه استفاده کرده است. تحقیق مذکور در پیاده‌سازی روش پیشنهادی خود از بستر اجرای برنامه‌های وب بهره‌برداری می‌کند تا شبکه عصبی عمیق را در قالب وضعیت اجرای یک برنامه وب از کارخواه به کارساز لبه انتقال دهد. بعد از انجام محاسبات در لبـه، حالت اجرای جدید از کارسـاز لبه بـه کارخواه مهاجرت می‌کند تا کارخـواه بتواند اجرای برنامه را ادامه دهد. پژوهـش [۱۱] بین تفکیک‌پذیری و نـرخ قاب معامله می‌کند تا بتواند بین تجزیـه و تحلیل وضوح بالا با تأخیر بالا و وضوح پایین بـا تأخیر کم تصمیم بگیرد و کارکرد برون‌سـپاری در این است که حداقل نرخ قاب و دقت را با توجه به شرایط شبکه بهبود بخشد. رویکرد فشرده‌سازی یکی دیگر از روش‌های برون‌سـپاری شـبکه‌های عصبی عمیق اسـت. در پژوهش [۲۰] پـدرام و همکارانش با ارائه یک معماری یادگیری عمیق برای خدمات محاسـباتی ابر و دسـتگاه تلفن هوشـمند توانسـته‌اند اندازه ویژگی‌های مورد نیاز برای ارسـال به ابـر را کاهش دهند و تأخیر و مصرف انرژی شـبکه‌های عصبی عمیـق در برنامه‌های تلفـن همراه را بهبود ببخشـند. همینطور در پژوهش [۲۱] با طراحی یک چارچوب که نسبت به تغییرات شبکه مقاوم است، اندازه داده‌های منتقل شده در شبکه را کاهش داده و سبب بهبود ترافیک شبکه، توان عملیاتی و دقت شده‌اند. در تحقیق دیگری مدلی تحت عنوان JointDNN ارائه شده است [۲۲] کـه در آن علاوه بر این‌که با پـردازش برخی لایه‌ها بر روی دسـتگاه تلفن همراه و برخی بر روی کارساز ابر برون‌سـپاری شـبکه‌های عصبی عمیق را بهینه کرده‌اند، توانسته‌اند با استفاده از روش فشرده‌سازی با کاهش ابعاد ویژگی‌ها هزینه‌های ارتباطی و زمان لازم برای ارسال داده به ابر را کاهش دهند.

با تأمل در کارهای گذشـته می‌توان دریافت که بخش عظیمـی از تحقیقات بر روی بخش‌بندی معطوف هسـتند که ایـن رویکرد با چالش‌هایی مواجه اسـت. معمولاً مدل شبکه‌های عصبی عمیق آن‌قدر بزرگ است که حتی در نظر گرفتن یک قسـمت از مدل روی تلفن همراه سربار زیادی را بـه وجود می‌آورد. همچنین با توجـه به این‌که معمولاً مرحلهٔ آموزش روی ابر انجام می‌شـود، بهتر اسـت مدل شبکه‌های عصبی عمیق نیز روی ابر باشد. مدل‌های شبکه



شکل ۱: چارچوب روش پیشنهادی در این مقاله

عصبی عمیق در عمل معمولاً پویا هستند و اگر پارامترها و وزن‌های آن‌ها بخواهند مرتباً به روز شوند، فرستادن بخشی یا کل مدل بر روی تلفن همراه هزینه‌بر خواهد بود. ضمناً شبکه‌های عصبی عمیق آن‌قدر بزرگ هستند که ارسال کل آن به صورت برنامه وب به ابر با هزینه زیادی مواجه خواهد بود.

۳- روش پیشنهادی

هدف از این پژوهش، بهینه‌سازی برون‌سپاری محاسبات شبکه‌های عصبی عمیق از طریق کاهش حجم داده‌های ارسالی به ابر می‌باشد. چارچوب کلی روش پیشنهادی برای رسیدن به این هدف در شکل (۱) نشان داده شده است و شامل سه رویکرد مختلف است که در ادامه توضیح داده شده است.

کاهش تعداد نمونه‌های داده

اولین رویکرد پیشنهادی این مقاله برای بهینه‌سازی برون‌سپاری محاسبات شبکه‌های عصبی عمیق، کاهش تعداد نمونه‌های ارسالی به ابر می‌باشد. کاهش تعداد نمونه داده‌های ورودی همیشه حجم داده‌های ارسالی به ابر را کاهش می‌دهد و منجر به کاهش هزینه ارتباطات (زمان، قیمت و انرژی مصرفی) می‌شود. ضمناً در بعضی از موارد که داده‌های ورودی از حسگرها دریافت می‌شوند (نظیر مورد مطالعاتی تشخیص فعالیت انسانی که در ادامه توضیح داده شده است)، می‌تواند منجر به کاهش انرژی نمونه‌برداری داده‌های حسگر شود. دلیل این امر این است که انرژی مصرفی لایه حسگر در گوشی‌های هوشمند متناسب با نرخ نمونه‌برداری افزایش می‌یابد. در نتیجه هر چه تعداد نمونه‌هایی که در واحد زمان دریافت می‌شود کمتر باشد انرژی کمتری در سمت تلفن همراه مصرف می‌شود. البته کاهش نمونه‌ها، دقت استنتاج شبکه‌های عصبی عمیق را پایین خواهد آورد. هدف از این تحقیق مطالعه معامله بین دقت و هزینه‌های ارتباطی و پیدا کردن نقطه بهینه در این معامله است و باید دید آیا با بازگرداندن تقریبی نمونه‌های حذف شده امکان ترمیم دقت از دست رفته تا حدودی وجود دارد یا خیر. برای این منظور، نگارندگان داده‌ها را با حالت‌های مختلف حذف نمونه‌های میانی یعنی حذف نمونه داده به صورت حفظ یکی در میان (نگهداری یک نمونه داده از هر دو نمونه)، حفظ دو در میان (نگهداری یکی از هر سه نمونه داده) و سه در میان کاهش داده و سپس آن‌ها را با سیاست درون‌یابی به دو روش خطی و مکعبی بازیابی نموده‌اند.

درون‌یابی فرآیند شناخته شده‌ای برای تخمین مقادیر موجود بین نقاط مشخص داده است. برای پیدا کردن مقادیر در نقاط بین نقاط داده شده می‌توان با فرض خطی

بودن تغییرات، طبق رابطه (۱) میان‌گیری حسابی انجام داد:

$$y = y_a + (y_b - y_a)\frac{x-x_a}{x_b-x_a} \qquad (۱)$$

کــه در این رابطــه xa و xb دو نقطه موجــود و x نقطه مورد نظر برای یافتن مقدار درون‌یابی شده است. همچنین می‌توان با در نظر گرفتن تعداد نقاط بیشتر، درون‌یابی با چند جمله‌ای از درجات بالاتر انجام داد. در این پژوهش از دو نوع سیاست درون‌یابی خطی و مکعبی استفاده شده است [۲۳].

کاهش دقت اعشار نمونه‌های داده

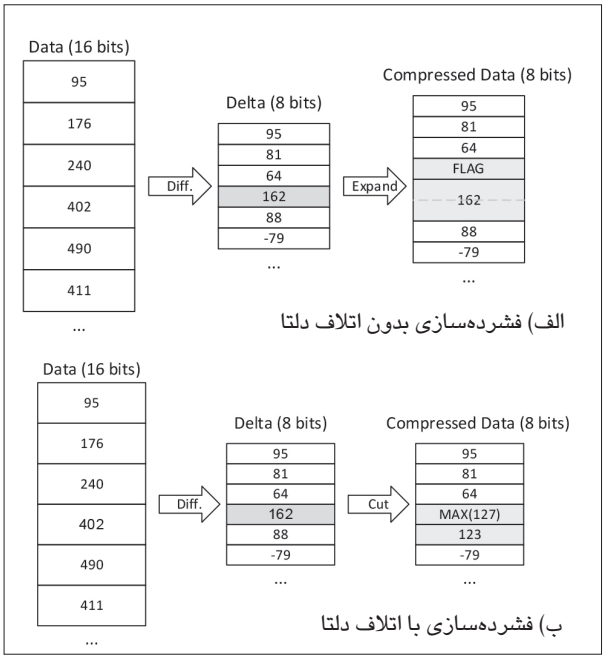
رویکرد پیشنهادی دوم در این مقاله برای بهینه‌سازی برون‌سپاری محاسبات شــبکه‌های عصبی عمیق، کاهش دقت اعشــار (Precision) نمونه داده‌های ارســالی به ابر می‌باشــد. برای بهینه‌سازی برون‌ســپاری می‌توان دقت اعشــار داده‌ها را کاهش داد تا با کم شــدن تعداد بیت‌های لازم بــرای هر نمونه داده، حجم داده‌های ارســالی به ابر کاهش یافته و در مصرف انرژی صرفه‌جویی شود. بدیهی است که در این روش معامله‌ای مانند روش قبل بین دقت و حجم داده ارســالی وجود دارد. در این روش با استفاده از رابطه نگاشت خطی (۲)، که در آن x متغیر اولیه و n تعداد بیت می‌باشد، داده‌های حســگرها را به جای نمایش ممیز شــناور می‌توان با یک عدد صحیح ۸ بیتــی، ۱۶ بیتی و... کد کرد. به‌عنوان مثال برای کدگذاری ۸ بیتی، کوچک‌ترین عدد خوانده شــده از حسگرها بر روی صفر و بزرگ‌ترین عدد بر روی ۲۵۵ نگاشت می‌شود و بقیه به تناسب در این فاصله قرار می‌گیرند.

$$f(x,n) = round\left(\frac{(x-\min(x))}{(\max(x)-\min(x))} \times (2^n - 1)\right) \qquad (۲)$$

پــس از دریافت داده‌هــای کاهش‌یافته در ســمت ابر می‌توان با اســتفاده از رابطه (۳) که معکوس نگاشت قبلی می‌باشــد، آن‌ها را به داده‌هایی نزدیک بــه داده‌های قبلی برگرداند:

$$g(\dot{x},n) = \min(x) + \frac{(\max(x)-\min(x))}{(2^n-1)} \times \dot{x} \qquad (۳)$$

در این رابطه x متغیر اولیه و $\dot{x}$ متغیر نگاشت شده بعد از کاهش دقت و n تعداد بیت مورد نظر برای اختصاص به هر نمونه داده است.



شکل ۲: روش پیشنهادی در فشرده‌سازی نمونه داده‌ها

فشرده‌سازی نمونه‌های داده

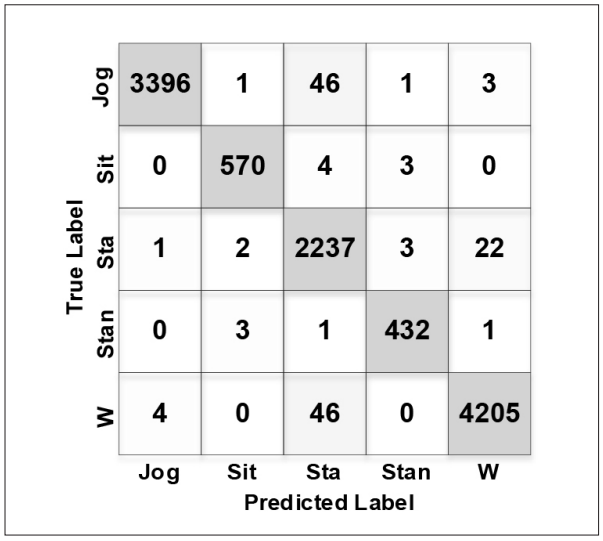
رویکــرد دیگــر مــورد مطالعــه در این مقالــه برای بهینه‌ســازی برون‌سپاری محاسبات شــبکه‌های عصبی عمیق، فشرده‌سازی نمونه‌های داده است. باید توجه داشت که اگر چه الگوریتم‌های فشرده‌ســازی مختلفی با ضریب فشرده‌ســازی بالا وجود دارند، باید روشــی با ســربار محاســباتی خیلی پایین انتخاب شــود. یکی از روش‌های ســبک برای فشرده‌ســازی جریــان داده‌هــا، روش دلتا می‌باشــد. این روش با فرض این‌که دامنه تغییر هر نمونه داده نســبت به مقــدار قبلی به حد کافی کوچک اســت، به جای ارســال نمونه‌ها، تفاضل آن‌هــا را با تعداد بیت کمتر ارسال می‌کند. بدیهی است که در صورت برقرار نبودن این فرض، الگوریتم مذکور قادر به فشرده‌سازی موثر داده‌ها نمی‌باشد. در این تحقیق از دو روش فشرده‌سازی با اتلاف و بدون اتلاف دلتا همانند شکل (۲) استفاده شده است.

در روش فشرده‌ســازی بدون اتلاف، فشرده‌سازی و غیر فشرده‌ســازی داده‌ها سبب هیچ گونه از دست رفتگی در آن‌ها نمی‌شــود. در این روش همان گونه که در شــکل (۲) الف مشاهده می‌شــود، با فرض این‌که نمونه داده‌ها n

بیــت (n = 32, 16, 8, 4)، دلتا m بیــت (m = 16, 8, 4, 2) و m < n است (به‌عنوان مثال: n = 32, m = 16)، به‌جای ارسال اعداد n بیتی به ابر، تفاضل آن‌ها در قالب m بیت ارســال می‌شــود. اگر تفاضل نمونه داده‌ها از m بیت (ظرفیت دلتا) بیشتر باشــد، اجباراً باید نمونه داده اصلی ارسال شود و برای تفکیک این حالت، قبل از نمونه داده‌اصلی یک علامت (داده ذخیره‌شــده) قرار داده شود. کوچک‌ترین عدد منفی بازه دلتا به‌عنوان علامت ذخیره اســتفاده شــده است. در نتیجه ســربار این علامت‌ها، با وجود فشرده‌سازی ممکن اســت حجم نهایی بعد از فشرده‌سازی از حجم اولیه داده بزرگ‌تر شود. این حالت زمانی اتفاق خواهد افتاد که دامنه تغییرات بین نمونه داده‌های پشــت ســر هم زیاد باشد و به‌طور مکرر نیاز به اســتفاده از علامت باشــد. در مقابل، روش فشرده‌ســازی با اتــلاف دلتا بــرآوردی تقریبی از داده‌های اصلی را نگه می‌دارد و بخشــی از داده اصلی را به ازای درجه فشرده‌ســازی از دســت می‌دهد. این روش همان‌طور که شکل (۲) ب نشان می‌دهد، هنگامی که تقاضل نمونه داده‌ها از m بیت بیشــتر باشــد (برای حالت خاص m = 8 در این شکل) حداکثر مقداری که در ۸ بیت می‌تواند نمایش یابد را ارســال نمــوده و از نمونه‌های بعدی برای اصلاح آن اســتفاده می‌کند. در نتیجه در فشرده‌سازی با اتلاف، افزایش حجم مشاهده نمی‌شود ولی به دلیل وجود اتــلاف، بهای کاهش حجم با از دســت رفتن دقت پرداخت می‌شود.

٤- پیاده سازی

تقریباً هر گوشی هوشمند مدرن دارای یک شتاب‌سنج ســه محوری است که شــتاب را در هر ســه بعد فضایی اندازه‌گیــری می‌کند. در این مقاله نگارنــدگان از داده‌های جمع‌آوری شــده از حسگرهای شتاب‌سنج استفاده کردند و یک شبکه عصبی LSTM، که با زبان پایتون در کتابخانه Tensorflow [٢٤] اجرا شــده است را به‌عنوان مدل مرجع آزمایش‌ها برای شناســایی فعالیت‌های انسانی (HAR) از داده‌های شتاب‌ســنج آموزش دادند. مدل شــبکه عصبی



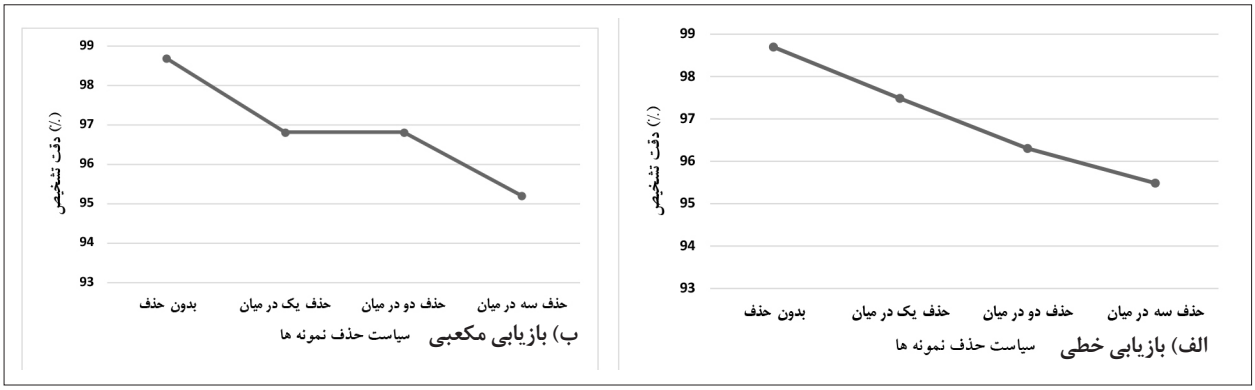
شکل ۳: دقت تشخیص به تفکیک فعالیت بدون حذف داده

مذکور شــامل دو لایه کاملاً متصل و دو لایه LSTM با ٦٤ واحد است.

آزمایش‌ها و نتایج

در ایــن بخش تلاش می‌شــود با توجــه به روش‌های پیشنهادی، نقطه بهینه در معامله بین دقت و هزینه ارتباطی مطالعه شــود. برای این منظور روش‌های پیشــنهادی این مقاله بر روی مجموعــه دادهٔ WISDM [۱۳] مورد ارزیابی قرار گرفته است. ابتدا دقت استنتاج شبکه عصبی عمیق برای داده‌های اصلی بدون روش‌های پیشــنهادی با اختصاص ۸۰٪ از داده‌ها به آموزش و مابقی آن به آزمایش، آزمایش شد. شکل (۳) جزئیات میزان دقت تشخیص به تفکیک فعالیت را نشان می‌دهد. در این شکل فعالیت‌ها شامل دویدن (Jog)، نشســتن (Sit)، بالا و پایین رفتن از پله‌ها (Sta)، ایســتادن (Stan) و قدم زدن (W) می‌باشد. دقت کلی تشخیص در این حالت (بدون صرفه‌جویی در حجم داده ارســالی) برابر با ۹۸/۷٪ می‌باشد.

در اولیــن آزمایــش، روش پیشــنهادی کاهش حجم داده‌های ارســالی بــه ابر از طریق کاهش تعــداد نمونه‌ها مورد ارزیابی قرار گرفت. بدین منظور، کسری از داده‌ها با حالت‌های مختلف حذف نمونه‌های میانی حذف شــده و با سیاست درون‌یابی به دو روش درون‌یابی خطی و مکعبی بازیابی شــد. هر دو گام یادگیری و اســتنتاج با داده‌های



شکل ۴: اثر حذف نمونه‌های داده قبل از ارسال به ابر بر روی دقت

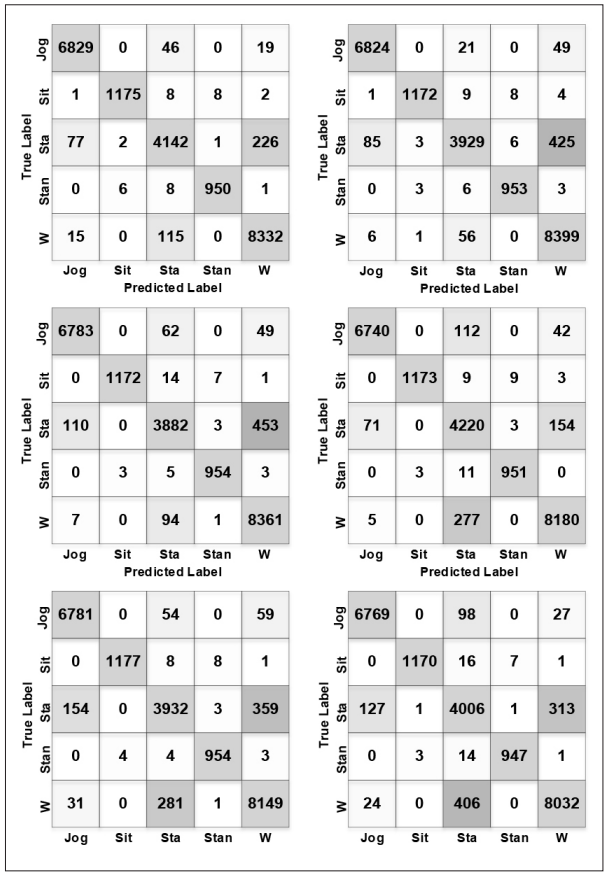
کاهش‌یافته انجام شـد. شکل (٤) اثر حذف نمونه‌های داده قبل از ارسال به ابر را بر روی دقت نشان می‌دهد که قسمت (الف) حذف یـک در میان، دو در میان و سـه در میان با بازیابی به روش خطی و قسمت (ب) برای بازیابی به روش مکعبی را نشان می‌دهد. همان‌طور که در شکل‌ها مشاهده می‌شـود به‌طـور کلی رویکـرد حذف نمونه‌هـا می‌تواند سودمند قلمداد شـود. به‌عنوان مثال حذف یک در میان با این‌که باعث کاهش چشمگیر ۵۰ درصدی حجم داده ارسالی می‌شـود تنها به میزان یک درصد در دقت پیش‌بینی‌ها اثر منفی دارد. ضمناً تفـاوت معناداری بین دو روش بازیابی خطی و مکعبی مشاهده نمی‌شود.

شکل (٥) اثر حذف نمونه‌های داده در دقت تشخیص به تفکیک فعالیت بـرای بازیابی خطی و مکعبی در حالت‌های مختلـف حذف را نشـان می‌دهد. همان‌طور که در شـکل مشـاهده می‌شـود عمده اثر منفی دقت تشخیص مربوط به فعالیت بالا و پایین رفتن از پله می‌باشـد. بنابراین دقت تشخیص به نوع فعالیت وابسته می‌باشد.

یکی از روش‌هـای دیگر برای کاهـش حجم داده‌های ارسـالی به ابر، کاهش دقت اعشار می‌باشد. قبل از ارسال به ابر، داده‌ها با تابع نگاشـت خطی به عدد صحیح n بیتی تبدیـل می‌شـود ($n = 1, 2, 4, 8, 16, 32$) و پس از دریافت داده‌ها در ابر محاسـباتی، با رابطـه معکوس به داده‌هایی نزدیک به داده‌های اولیه برگردانده می‌شـود. شکل (٦) اثر کاهـش دقت نمونه‌های داده قبل از ارسـال به ابر به روی دقت را برای حالت‌های مختلف نگاشت خطی نشان می‌دهد. همان‌گونه که انتظار می‌رود، کاهش دقت اعشـار با کاهش دقت تشـخیص همراه اسـت. اما این معامله دقت در ازای کاهش حجم سـودمند اسـت و به‌عنوان مثال در حالت ۸ بیتـی که به میزان ۸ برابر صرفه‌جویی در حجم ارسـالی (در مقایسه با دقت اعشـار ٦٤ بیتی ممیز شناور) حاصل می‌شـود، تنها در حدود یک درصد کاهش دقت تشخیص اتفاق افتاده اسـت. نکته جالب توجه این اسـت که حتی با کاهش بیشتر دقت اعشار و استفاده از دقت ٤ بیتی که منجر به کاهش ١٦ برابری حجم داده می‌شود، دقت تشخیص در همین محدوده باقی‌مانده است. البته کاهش دقت اعشار به میزان کمتر از ٤ بیت دقت تشـخیص را بـه حد غیر قابل قبولی پایین می‌آورد.

شکل (٧) جزئیات اثر کاهش دقت اعشار نمونه‌های داده را در دقت تشخیص به تفکیک فعالیت در حالت‌های مختلف نشـان می‌دهد. همان‌طور که در شـکل مشاهده می‌شود، مشـابه روش اول، بیشـترین اثر منفی کاهش دقت اعشار بر روی تشـخیص فعالیت بالا و پایین رفتن از پله‌ها اتفاق افتاده اسـت. همچنین کاهش بیش از اندازه دقت اعشـار سبب کاهش دقت تشـخیص اغلب فعالیت‌ها شده است که در آزمایش‌های ١ بیتی به وضوح قابل مشاهده است.

در رویکـرد فشرده‌سـازی نمونه‌هـای داده، به جای ارسـال اعداد n بیتی، تفاضل آن‌ها در قالب دلتای m بیتی ارسـال می‌شـود. البته قبـل از این کار، داده‌هـا از حالت نمایش اعشـاری به حالت صحیح n بیتی تبدیل می‌شوند. اثـر فشرده‌سـازی به هـر دو صورت بـدون اتلاف و با



شکل ۵: اثر حذف نمونه‌های داده بر دقت تشخیص به تفکیک فعالیت برای بازیابی خطی (سمت چپ) و مکعبی (سمت راست) با حذف یک در میان (بالا)، دو در میان (وسط) سه در میان (پایین).

اتلاف آزمایش شده اســت. معیار اندازه‌گیری مؤثر بودن فشرده‌سازی، نسبت فشرده‌سازی است که طبق رابطه (٤) تعریف می‌شود.

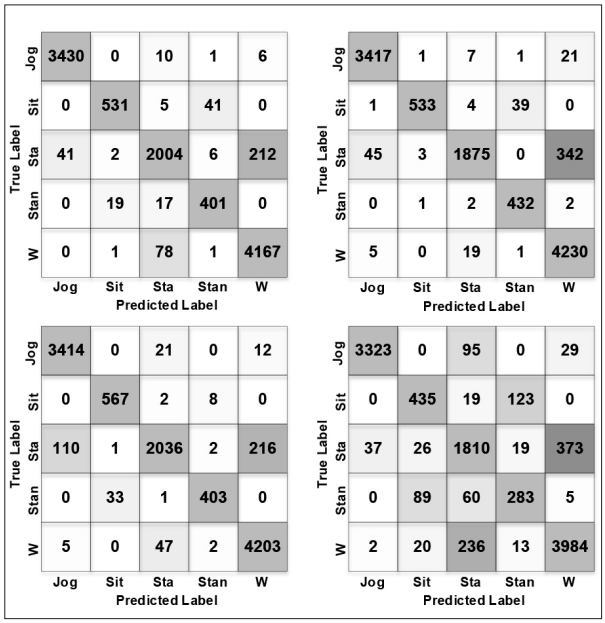
$$\text{نسبت فشرده سازی} = \frac{\text{اندازه داده غیر فشرده}}{\text{اندازه داده فشرده}} \qquad (٤)$$

اگرچه فشرده‌ســازی بدون اتلاف اثری بر روی دقت نــدارد، باید مؤثر بــودن آن از طریق اندازه‌گیری نســبت فشرده‌ســازی بررسی شود. از طرف دیگر، فشرده‌سازی با اتلاف پیشنهادی، اگرچه نســبت ثابتی از فشرده‌سازی ارائه می‌کند، باید اثر آن بر دقت تشخیص بررسی شود. در حالت بدون اتلاف ابتدا به جای ارسال داده‌های صحیح ۳۲ بیتی، تفاضل آن‌ها در قالب دلتای ۱٦، ۸، ٤ و ۲ بیتی ارسال می‌شود. در حالتی که تفاضل داده‌ها از ظرفیت دلتا بیشتر باشــد نمونه داده اصلی ارسال می‌شود و برای تفکیک این حالت قبل از داده اصلی یک علامت قرار داده می‌شود و با

شکل۶: اثر کاهش دقت اعشار نمونه‌های داده قبل از ارسال به ابر بر روی دقت تشخیص برای حالت‌های مختلف نگاشت خطی

وجود فشرده‌سازی ممکن است حجم نهایی از حجم اولیه بیشتر شــود. این مراحل را برای داده‌های ۱٦، ۸ و ٤ بیتی با دلتای ۸، ٤ و ۲ بیت آزمایش شــده و حجم داده‌ها مورد بررسی قرار گرفته است. همان‌طور که در شکل(۸) مشاهده می‌شود در اغلب حالت‌ها حجم نهایی بعد از فشرده‌سازی از حجم اولیه داده‌های اصلی بیشــتر شده است. در نتیجه، روش فشرده‌ســازی بدون اتلاف دلتا به جز در حالت داده ٤ بیتی با ظرفیت دلتا ۲ بیتی، روش مناســبی برای کاهش حجم داده‌های ارسالی به ابر نمی‌باشد.

روش فشرده‌سازی با اتلاف دلتا که برآوردی تقریبی از داده اصلی را نگه می‌دارد، همانند فشرده‌سازی با اتلاف به جای ارســال داده‌های صحیح ۳۲ بیتــی، تفاضل آن‌ها در قالب دلتای ۱٦، ۸، ٤ و ۲ بیتی ارســال می‌شــود با این تفاوت که در حالت بیشتر بودن ظرفیت دلتا حداکثر مقداری که در ظرفیت دلتا اســتفاده می‌شــود را قرار می‌دهد و از نمونه‌های بعدی برای اصلاح آن استفاده می‌کند. در نتیجه نسبت فشرده‌سازی همیشــه برابر با نسبت ایده‌آل اندازه داده به اندازه دلتا است، اما به دلیل از دست رفتن بخشی از داده اصلی، دقت تشخیص کمتر خواهد شد. در مواردی که اختلاف اندازه داده و اندازه دلتا قابل توجه باشــد، دقت به میزان فاحشی از دست خواهد رفت. از این رو آزمایش‌های این بخش با فشرده‌سازی داده‌های ۱٦، ۸ و ٤ بیت به ترتیب بــا دلتای ۸، ٤ و ۲ بیت انجام شــد کــه نتایج دقت حاصل در جدول (۱) قابل مشــاهده اســت. با توجه به این نتایج، فشرده سازی دلتا با اتلاف بهتر از حالت بدون اتلاف است.

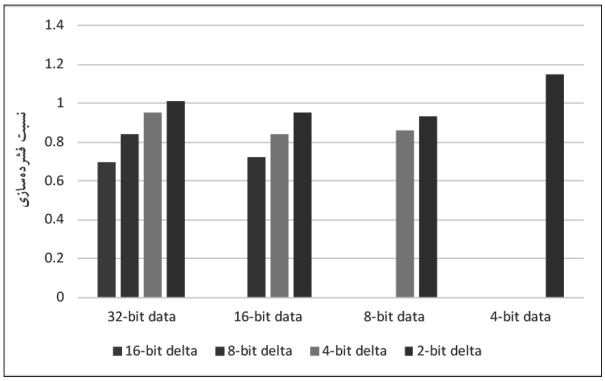


شکل ۷: اثر کاهش دقت اعشار نمونه‌ها بر دقت تشخیص به تفکیک فعالیت در نگاشت خطی ۳۲ بیتی (بالا سمت چپ)، ۸ بیتی (بالا سمت راست)، ۱۶ بیتی (پایین سمت چپ) و ۱ بیتی (پایین سمت راست)

به‌عنوان مثـــال در حالت داده ۵ بیتـــی در ازای دو درصد کاهش دقت، اندازه داده ارسالی به نصف کاهش یافته است. اما در هر حال روش فشرده‌سازی دلتا به اندازه دو روش قبل مؤثر نمی‌باشد.

۵- نتیجه‌گیری

تحقیق حاضر به مطالعه امکان بهینه‌سازی برون‌سپاری محاســبات از طریــق کاهش حجــم داده ارســالی به ابر محاسباتی با تمرکز بر کاربرد تشخیص فعالیت انسانی با یادگیری عمیق اختصاص دارد. برای این منظور سه روش کاهش تعداد نمونه‌های داده، کاهش دقت اعشار نمونه‌های داده و فشرده‌سازی نمونه‌های داده پیشنهاد شده است. در روش اول، داده‌ها قبل از ارسال به صورت یکی در میان یا بیشتر حذف شده و در سمت ابر به روش تخمینی بازیابی می‌شوند. در روش دوم، نمونه داده‌ها قبل از ارسال با تابع نگاشت خطی به عدد صحیح با تعداد بیت کمتر تبدیل شده و در سمت ابر با رابطه معکوس نگاشت تخمین زده می‌شوند. در روش ســوم، داده‌ها با یک الگوریتم فشرده‌ســازی با سربار کم و به یکی از دو صورت فشرده‌سازی با اتلاف و



شکل ۸: اثر فشرده‌سازی بدون اتلاف دلتا بر حجم نمونه داده

جدول۱: تاثیر فشرده‌سازی با اتلاف نمونه داده ها در دقت

اندازه نمونه داده (بیت)	اندازه دلتا (بیت)	دقت بدون فشرده‌سازی	دقت با فشرده‌سازی	نسبت فشرده‌سازی
۸	۶	۹۶.۷	۹۴.۷	۰.۲۷
۴	۳	۹۷.۰	۹۵.۴	۰.۰۹
۴	۲	۹۷.۰	۹۵.۰	۰.۲۹

بدون اتلاف فشرده شده و در سمت ابر بازیابی می‌شوند. در دو روش پیشنهادی کاهش حجم تعداد نمونه‌ها و کاهش دقت اعشـــار نمونه‌های داده‌ها، کاهش حجم داده ارسالی تنها باعث کاهش اندکی در دقت تشخیص فعالیت می‌گردد. روش کاهش دقت اعشار به دلیل کاهش چشمگیرتر در حجم داده نسبت به روش اول برتری نشان می‌دهد. با استفاده از تنها ٤ بیت برای ارسال هر نمونه داده که باعث کاهش ١٦ برابری حجم داده‌ها می‌شود، دقت از دست رفته محدود به یک درصد می‌باشـــد. اگر چه روش فشرده‌سازی با اتلاف نتایج بهتری نســـبت به روش بدون اتلاف نشـــان می‌دهد، هیـــچ یک از این دو روش به کارآمدی روش کاهش دقت و کاهش تعداد نمونه‌های داده نیستند. به‌عنوان کار آینده این تحقیق، نیاز به بررسی اثر ترکیبی روش‌های کاهش حجم در معامله بین دقت تشخیص و هزینه ارتباطات خواهد بود. ضمنا شایســـتگی روش پیشنهادی در ســـایر کاربردهای مختلف نیازمند بررســـی می‌باشـــد. بررسی روش کاهش هزینه ارتباطی نظیر اســـتفاده از روش کاهش دقت اعشار نمونه‌ها با نگاشت‌های مختلف غیر خطی و یا بررسی سایر روش های فشرده‌ســـازی از دیگر کارهای پیش روی این تحقیق می‌باشد.

مراجع

1. Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton., "Speech recognition with deep recurrent neural networks", 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.
2. Eshratifar, Amir Erfan, and Massoud Pedram., "Energy and performance efficient computation offloading for deep neural networks in a mobile cloud computing environment", Proceedings of the 2018 on Great Lakes Symposium on VLSI. 2018.
3. Dey, Swarnava, Jayeeta Mondal, and Arijit Mukherjee., "Offloaded execution of deep learning inference at edge: Challenges and insights", 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2019.
4. Huang, Yutao, et al., "DeePar: A hybrid device-edge-cloud execution framework for mobile deep learning applications", IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2019.
5. Kemp, Roelof, et al., "Cuckoo: a computation offloading framework for smartphones", International Conference on Mobile Computing, Applications, and Services. Springer, Berlin, Heidelberg, 2010.
6. Mitchell, Tom M. "Machine learning", Burr Ridge, IL: McGraw Hill, 1997.
7. Karki, Aajna, et al., "Tango: A deep neural network benchmark suite for various accelerators", IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, 2019.
8. Guo, Kaiyuan, et al., "[DL] A survey of FPGA-based neural network inference accelerators", ACM Transactions on Reconfigurable Technology and Systems (TRETS), 2019.
9. Qin, Dongming, et al., "A novel combined prediction scheme based on CNN and LSTM for urban PM 2.5 concentration", IEEE Access 7: 20050-20059, 2019.
10. Hochreiter, Sepp, and Jürgen Schmidhuber., "Long short-term memory", Neural computation 9.8 : 1735-1780, 1997.
11. Ran, Xukan, et al., "Delivering deep learning to mobile devices via offloading", Proceedings of the Workshop on Virtual Reality and Augmented Reality Network, 2017.
12. Fadishei, H., "Energy-Efficient Human Activity Recognition on Smartphones: A Test-Cost Sensitive Approach", International Journal of Information and Communication Technology Research, 2018.
13. Weiss, Gary M., "WISDM Smartphone and Smartwatch Activity and Biometrics Dataset", UCI Machine Learning Repository: WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set, 2019.
14. Wang, Jianyu, et al., "Edge cloud offloading algorithms: Issues, methods, and perspectives", ACM Computing Surveys (CSUR), 2019.
15. Mohammed, Thaha, et al., "Distributed Inference Acceleration with Adaptive DNN Partitioning and Offloading", IEEE INFOCOM 2020-IEEE Conference on Computer Communications. IEEE, 2020.
16. Li, He, Kaoru Ota, and Mianxiong Dong., "Learning IoT in edge: Deep learning for the Internet of Things with edge computing", IEEE network, 2018.
17. Jeong, Hyuk-Jin, et al., "IONN: Incremental offloading of neural network computations from mobile devices to edge servers", Proceedings of the ACM Symposium on Cloud Computing, 2018.
18. Teerapittayanon, Surat, Bradley McDanel, and Hsiang-Tsung Kung., "Distributed deep neural networks over the cloud, the edge and end devices", IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2017.
19. Jeong, Hyuk-Jin, et al., "Computation offloading for machine learning web apps in the edge server environment", 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2018.
20. Eshratifar, Amir Erfan, Amirhossein Esmaili, and Massoud Pedram., "Bottlenet: A deep learning architecture for intelligent mobile cloud computing services", IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). IEEE, 2019.
21. Chung, Jae-Won, Jae-Yun Kim, and Soo-Mook Moon., "ShadowTutor: Distributed Partial Distillation for Mobile Video DNN Inference", arXiv preprint arXiv:2003.10735, 2020.
22. Eshratifar, Amir Erfan, Mohammad Saeed Abrishami, and Massoud Pedram., "JointDNN: an efficient training and inference engine for intelligent mobile cloud computing services", IEEE Transactions on Mobile Computing, 2019.
23. Rosloniec, S., "Fundamental numerical methods for electrical engineering", Vol. 18. Springer Science & Business Media, 2008.
24. https://www.tensorflow.org/.