

بسم الله الرحمن الرحيم



دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

استفاده از پردازش لبه به منظور بهبود پیاده‌سازی یادگیری فدرال در شبکه‌های اینترنت اشیاء

گزارش پروژه کارشناسی مهندسی کامپیوتر

امیررضا حسینی

استاد راهنما

دکتر امیر خورسندی

شهریور ماه ۱۴۰۲

تقدیم به
پدر و مادر عزیزم

فهرست مطالب

عنوان	صفحه
فهرست مطالب	پنج
فهرست تصاویر	هفت
چکیده	۱
فصل اول: مقدمه	۲
۱-۱ موضوع پژوهش و مسئله	۳
۲-۱ پیشینه تحقیق	۴
۳-۱ اهداف و دستاوردهای تحقیق	۵
۴-۱ ساختار گزارش	۵
فصل دوم: پیش زمینه و مروری بر مطالعات انجام شده	۶
۱-۲ نگاه نزدیک تر به الگوریتم های یادگیری	۷
۱-۱-۲ یادگیری متمرکز	۷
۲-۱-۲ یادگیری غیرمتمرکز	۸
۳-۱-۲ یادگیری توزیع شده	۸
۴-۱-۲ یادگیری فدرال	۹
۵-۱-۲ الگوریتم FedAVG	۱۰
۲-۲ شبکه عصبی کانولوشنی	۱۱
۳-۲ پردازش لبه	۱۳
۱-۳-۲ معماری های متنوع پردازش لبه	۱۵
۴-۲ یادگیری فدرال در اینترنت اشیا	۱۵
۵-۲ جمع بندی	۱۶
فصل سوم: چارچوب کاری پروژه	۱۷
۱-۳ مجموعه داده تعریف شده در مسئله	۱۷
۲-۳ مدل استفاده شده برای حل مسئله	۱۸
۳-۳ محیط توسعه استفاده شده	۱۹
۴-۳ پروتکل های انتقال داده در شبکه	۲۱

۲۲ ۵-۳ محیط شبیه‌سازی پیاده شده

۲۵ فصل چهارم: نتیجه‌گیری و پیشنهادها

۲۵ ۱-۴ معرفی معیارهای مختلف

۲۷ ۲-۴ نتایج به‌دست آمده

۲۹ ۳-۴ پیشنهادها برای آینده

۳۲ پیوست اول: پروتکل SAFA

۳۵ مراجع

۳۸ چکیده انگلیسی

فهرست تصاویر

۴	۱-۱ معماری یک شبکه یادگیری فدرال
۷	۱-۲ یادگیری متمرکز
۸	۲-۲ یادگیری غیر متمرکز
۹	۳-۲ یادگیری توزیع شده
۹	۴-۲ شمایی از یادگیری فدرال در اینترنت اشیا
۱۲	۵-۲ نمونه ی بخش بندی معنایی در ماشین های خودران
۱۳	۶-۲ معماری یک شبکه کانولوشنی
۱۴	۷-۲ دیاگرام یک شبکه دارای پردازش لبه
۱۵	۸-۲ انواع قرار گیری المان های اصلی در شبکه اینترنت اشیا
۱۶	۹-۲ کاربردهای یادگیری فدرال در اینترنت اشیا
۱۸	۱-۳ نمونه ای از تصاویر دیتاست CIFAR۱۰
۱۹	۲-۳ ساختار شبکه عصبی کانولوشن استفاده شده در مسئله ۱۰ کلاسه
۲۰	۳-۳ نمونه ای از پیاده سازی یادگیری فدرال روی دستگاه های تعبیه شده
۲۱	۴-۳ استفاده از nginx به عنوان پروکسی سرور
۲۲	۵-۳ محیط شبیه سازی شده در محیط نرم افزار GNS3
۲۳	۶-۳ نمونه ای از اجرای فرایند یادگیری و تست توسط یک شبکه ساده دارای دو کارگر
۲۴	۷-۳ نمونه ای از کنسول یک کارگر در حال اجرای یادگیری فدرال
۲۶	۱-۴ ماتریس درهم ریختگی برای مسئله با دو دسته
۲۷	۲-۴ نمودار دقت به دست آمده در هر دور از آموزش سراسری
۲۸	۳-۴ نمودار خطا در هر دور از آموزش سراسری
۳۳	آ-۱ سناریوای از اجرای پروتکل SAFA در ۳ دور

چکیده

امروزه، کاربرد هوش مصنوعی به سرعت در حوزه های مختلف گسترش پیدا کرده است. با توجه به نیازهای متفاوت در کاربردهای مجزا، تا کنون رویکردها و روش های گوناگونی برای تحقق زیرساخت های مربوط به فناوری هوش مصنوعی ارائه شده اند. یکی از این موارد، رویکرد یادگیری ماشینی برای حفظ حریم خصوصی است که به عنوان یادگیری فدرال شناخته می شود. با توجه به مقررات سختگیرانه حریم خصوصی در حوزه های مختلف از جمله پزشکی، صنایع و غیره، یادگیری فدرال امکان توسعه برنامه هایی را فراهم می کند که با استفاده از داده های حساس کاربر، به تجزیه و تحلیل اطلاعات و ارائه خدمات مبتنی بر هوش مصنوعی و یادگیری ماشین می پردازند. این موضوع می تواند زمینه ساز یک تحول اساسی در ارائه خدمات در امور مالی، مراقبت های بهداشتی، صنایع حمل و نقل و به صورت کلی اینترنت اشیا باشد. البته نکته حائز اهمیت این است که یادگیری فدرال ممکن است بر روی تعداد بسیار زیادی از دستگاه های نهایی ناهمگون و به صورت توزیع شده اجرا گردد که بسیاری از این دستگاه ها توان پردازشی و دسترسی به منابع انرژی محدود دارند. این یک چالش اساسی بر سر راه اجرای موثر، پایدار و مقیاس پذیر یادگیری خواهد بود.

در این پروژه، با توجه به ظرفیت پردازش لبه، ساختاری پیشنهاد می شود تا بتواند بستر مناسبی برای توسعه یادگیری فدرال در شبکه هایی با کاربرد در اینترنت اشیا را فراهم سازد. به صورت کلی در پردازش لبه، از ظرفیت های ذخیره سازی، ارتباطی و محاسباتی سرورهای لبه که نزدیک به دستگاه های پایانی مستقر شده اند، برای کاهش تاخیر سیستم استفاده می شود. به بیان دیگر با انجام محاسبات اولیه و پیش پردازش ها در نزدیکی محل تولید داده ها، میزان ترافیک ارسالی به سرور اصلی و در نتیجه تاخیر دسترسی به آن کاهش می یابد. لذا این رویکرد می تواند در بهبود عملکرد یادگیری فدرال نیز موثر باشد و این موضوع اصلی این پروژه را تشکیل می دهد.

واژه های کلیدی: ۱- یادگیری فدرال ۲- پردازش لبه ۳- اینترنت اشیا ۴- یادگیری ماشین.

فصل اول

مقدمه

در سال‌های اخیر اینترنت اشیا^۱ به سرعت در حال توسعه است و قابلیت‌های حساس و محاسبه فراگیر را برای اتصال طیف گسترده‌ای از وسایل به یکدیگر بر پایه اینترنت فراهم می‌کند [۱]. برای رسیدن به اهداف متنوع در رابطه با داده‌های تولید شده توسط دستگاه‌های اینترنت اشیا فراگیر، تکنیک‌های هوش مصنوعی^۲ مانند یادگیری عمیق^۳ به طور گسترده‌ای برای آموزش مدل‌های داده برای فعال کردن برنامه‌های اینترنت اشیا هوشمند مانند بهداشت هوشمند، حمل و نقل هوشمند و شهر هوشمند استفاده شده است [۲]. به طور سنتی، عملکردهای هوش مصنوعی در یک سرور ابری یا یک مرکز داده برای یادگیری و جدا کردن داده قرار می‌گیرد که این مدل امروزه با توجه به انفجار داده اینترنت اشیا، محدودیت‌های بحرانی را به وجود می‌آورد [۳]. با چنین رشد عظیم داده اینترنت اشیا در لبه شبکه، انتقال داده‌های عظیم تولید شده اینترنت اشیا به سرورهای از راه دور به دلیل منابع شبکه مورد نیاز و تأخیر ناشی از آن غیرقابل قبول است و ممکن است الزامات کاربردهایی که به زمان حساس هستند به خوبی برآورده نشوند. از طرف دیگر نیز استفاده از سرورهای شخص ثالث برای فرآیند آموزش هوش مصنوعی نگرانی‌های حفظ حریم خصوصی را به همراه می‌آورد [۴].

¹ Internet Of Things

² Artificial Intelligence

³ Deep Learning

۱-۱ موضوع پژوهش و مسئله

یادگیری فدرال^۱، برای پیاده‌سازی سیستم‌های اینترنت اشیاء هوشمند و حفظ حریم خصوصی در اوایل سال ۲۰۱۷ توسط محققان شرکت گوگل پیشنهاد شد [۵]. به طور فنی، یادگیری فدرال یک روش همکاری توزیع شده هوش مصنوعی است که اجازه می‌دهد با هماهنگ کردن چندین دستگاه با یک سرور مرکزی بدون به اشتراک گذاری مجموعه داده‌های واقعی، آموزش مدل روی داده‌ها را انجام داد. به عنوان مثال، چندین دستگاه اینترنت اشیاء می‌توانند به عنوان کارگران^۲ عمل کنند تا با یک تجمیع‌کننده^۳ (به عنوان مثال، یک سرور) برای انجام آموزش شبکه عصبی در شبکه‌های اینترنت اشیاء هوشمند ارتباط برقرار کنند. به طور خاص، تجمیع‌کننده ابتدا با یک مدل سراسری^۴ با پارامترهای یادگیری را شروع می‌کند. هر کارگر مدل فعلی را از تجمیع‌کننده دریافت می‌کند. سپس به کمک مجموعه داده محلی خود و با استفاده از روش‌های مناسب مانند نزول گرادیان تصادفی^۵، مبادرت به به‌روز رسانی مدل درونی خود می‌کند. سپس، تجمیع‌کننده در پایان هر دور تمام به‌روز رسانی‌های محلی را ترکیب کرده و یک مدل جهانی جدید و بهبود یافته را ساختاردهی می‌نماید. با استفاده از قدرت محاسبات توزیع شده کارگران، تجمیع‌کننده می‌تواند کیفیت آموزش را افزایش داده و در عین حال نشت اطلاعات حریم خصوصی مصرف‌کننده را به حداقل رساند. در دور بعد دوباره، کارگران محلی به‌روز رسانی جهانی را از تجمیع‌کننده دریافت کرده و به‌روز رسانی محلی بعدی خود را محاسبه می‌کنند تا زمانی که آموزش جهانی کامل شود.

یادگیری فدرال، با توجه به مفهوم عملیاتی نوآورانه خود می‌تواند مزایای مختلف مهمی را برای برنامه‌های اینترنت اشیاء ارائه دهد: اولاً در یادگیری فدرال، داده‌های خام برای آموزش در تجمیع‌کننده نیاز نیست؛ بنابراین، نشت اطلاعات حساس کاربر به طرف سوم خارجی به حداقل می‌رسد و درجه‌ای نسبتاً ایده‌آل از حفظ حریم خصوصی داده فراهم می‌شود. با توجه به قوانین حفاظت از حفظ حریم خصوصی داده که به طور فزاینده‌ای سخت‌گیرانه است، حفظ ویژگی حفاظت از حریم خصوصی یادگیری فدرال را یک راهکار ایده‌آل برای ساخت سیستم‌های اینترنت اشیاء هوشمند و امن می‌کند. [۳] ثانیاً از آنجا که نیاز به انتقال داده‌های اینترنت اشیاء به سرور وجود ندارد، استفاده از یادگیری فدرال کمک می‌کند تأخیر ارتباطاتی ناشی از بارگذاری داده‌ها کاهش یابد. همچنین سبب صرفه‌جویی از منابع شبکه می‌شود. با استفاده از منابع محاسباتی زیاد و مجموعه داده‌های متنوع از شبکه دستگاه‌های اینترنت اشیاء، یادگیری فدرال قادر است سرعت همگرایی کل فرآیند آموزش را افزایش دهد.

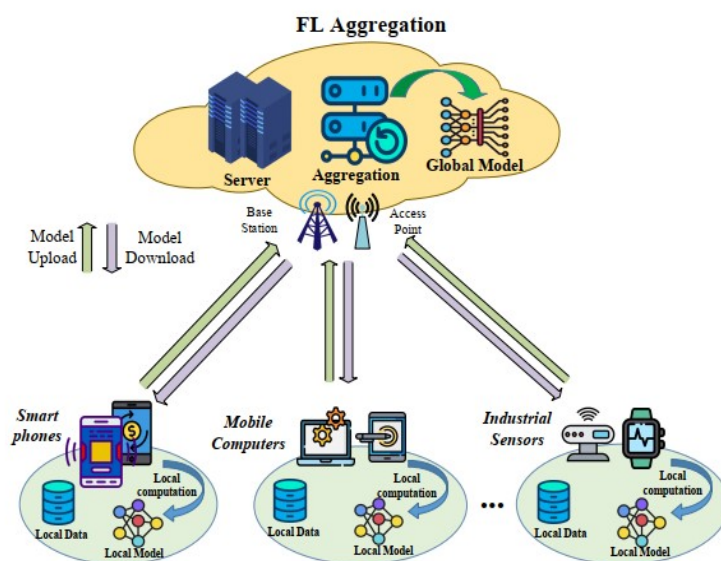
¹ Federated Learning

² Workers

³ Aggregator

⁴ Global Model

⁵ Stochastic Gradient Descent



شکل ۱-۱: معماری یک شبکه یادگیری فدرال [۳]

و نرخ‌های دقت یادگیری بهتری را به دست آورد که ممکن است با استفاده از روش‌های هوش مصنوعی متمرکز با داده‌های ناکافی و قابلیت‌های محاسباتی محدود به دست نیاید. یادگیری فدرال همچنین قابلیت توسعه شبکه‌های هوشمند را به دلیل طبیعت یادگیری توزیع شده خود بهبود می‌بخشد [۳]. شکل ۱-۱ یک شبکه یادگیری فدرال را نشان می‌دهد که با استفاده از رد و بدل کردن داده مربوط به مدل کار می‌کند.

۲-۱ پیشنهاد تحقیق

پژوهشگران فعال در حوزه یادگیری فدرال در حال حاضر بیشتر بر دو چالش اصلی تمرکز دارند. اولاً، ناهمگونی داده در بین کارگران باعث کند شدن همگرایی مدل نسبت به یادگیری سنتی می‌شود. چالش دیگر هزینه ارتباطی در هر دو فرآیند بارگذاری و بارگیری مدل است که گلوگاه یادگیری توزیع شده است. به خصوص در مورد سناریوهایی با تعداد دستگاه‌های زیاد در اینترنت اشیاء، این چالش جدی‌تر است. تعداد زیادی از کارها برای مقابله با این دو چالش پیشنهاد شده‌اند که می‌توان آن‌ها را در دسته‌بندی الگوریتم‌های بهینه‌سازی و استراتژی‌های کارآمد ارتباطات طبقه‌بندی نمود [۴][۱۶][۱۷].

به عنوان مثال از الگوریتم‌های بهینه‌سازی می‌توان به انواع بهینه‌سازی‌های سراسری و محلی که سعی می‌کنند با استفاده از الگوریتم‌های مختلف، مانند گرادیان کاهشی، نمونه‌برداری تصادفی، گرادیان کاهشی تصادفی و غیره، به روزرسانی‌های مدل را در سرور مرکزی همگرا کنند. این روش‌ها برای کاهش هزینه ارتباطات و افزایش سرعت یادگیری مفید هستند. از طرف دیگر، می‌توان به استراتژی‌های کارآمدتر اشاره کرد که سعی می‌کنند با

استفاده از روش‌های مختلف، مانند هماهنگ‌سازی دوره‌ای، هماهنگ‌سازی تصادفی، هماهنگ‌سازی بر اساس شرط و غیره، زمان و شرایط ارسال به روزرسانی‌های مدل را تعیین کنند. این روش‌ها برای حل مشکلات ناشی از نامتقارن بودن داده‌ها و نامتعادل بودن توزیع داده‌ها بین گره‌های محلی مناسب هستند. پژوهش‌های مختلف در این زمینه عموماً بر روی یک یا ترکیبی از چند از این روش‌ها تمرکز دارند.

۱-۳ اهداف و دستاوردهای تحقیق

باتوجه به اهمیت بارز روش‌های نوین یادگیری ماشین در دنیای امروزه و به خصوص کاربرد آن در اینترنت اشیاء، در این پروژه، سعی شده تا با استفاده از ظرفیت پردازش لبه یک چارچوب یادگیری فدرال مناسب که در آن چالش ارتباطات و ناهمگونی داده‌ها تا حد خوبی مدیریت شده ارائه دهد. شبیه‌سازی این نظریه در چارچوب نرم‌افزارهای مطرح شبیه‌سازی از جمله GNS3^۱ و Flower و غیره انجام شده و نتایج نشان دهنده موثر بودن این راهکار است.

۱-۴ ساختار گزارش

در ادامه، فصل دوم مروری بر مفاهیم پایه مورد نیاز برای این پروژه را در بر می‌گیرد. فصل سوم به تعریف مسئله، معرفی و بررسی کامل مدل پیشنهادی برای حل مسئله، داده‌ها و چهارچوب کاری پروژه خواهد پرداخت. در نهایت، فصل چهارم ارزیابی و جمع‌بندی کلی پروژه را دربرمی‌گیرد.

^۱ Graphical Network Simulator-3

فصل دوم

پیش زمینه و مروری بر مطالعات انجام شده

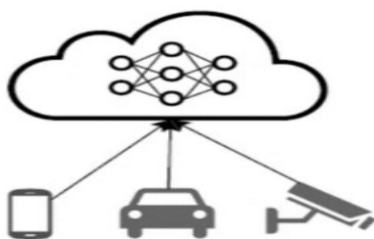
رشد چشمگیر فناوری به همراه سهولت دسترسی به اینترنت در سال‌های اخیر باعث شده که بیشتر دستگاه‌های اطراف خود را متصل به اینترنت ببینیم. این دنیای جدید که به دنیای اینترنت اشیاء معروف است شامل خانه‌های هوشمند، دستگاه‌های پوشیدنی، خودروهای خودران و تلفن‌های هوشمند و... است که همگی زندگی روزمره انسان را تغییر داده‌اند. استفاده از این سیستم‌ها همگی باعث تولید حجم قابل توجهی داده در طول روز می‌شود که شرکت‌های بزرگ فناوری از این داده‌ها بهره برده و با استفاده از آن‌ها اقدام به انواع سرویس‌دهی به کاربران خود می‌نمایند [۲]. شرکت‌های پیشرو برای تصمیم‌گیری‌های کلان مدیریتی و ارائه سرویس بهتر و با کیفیت‌تر به مشتریان، نیازمند استفاده از مدل‌های هوش مصنوعی برای ارتقاء کیفیت سیستم‌های هوشمند خود در جهت بهره‌برداری از این داده‌ها هستند. روش‌های متنوعی در رابطه با نحوه استفاده از این مدل‌ها وجود دارد. در ادامه روش‌های مختلفی از مطرح‌ترین روش‌ها توضیح داده می‌شود و نگاهی ریز بینانه‌تر به یادگیری فدرال و رابطه آن با اینترنت اشیاء انداخته می‌شود [۱۸].

۱-۲ نگاه نزدیک‌تر به الگوریتم‌های یادگیری

با توجه به رشد علم هوش مصنوعی و استفاده از روش‌های یادگیری ماشین، می‌توان از حجم بسیار زیاد داده تولید شده توسط گره‌های اینترنت اشیاء به نحو مطلوبی استفاده نمود و الگوریتم‌های مورد نظر، جهت رسیدن به اهداف مختلف را بر روی آن‌ها اجرا کرد. حال برای مدیریت و اجرای الگوریتم‌های یادگیری، روش‌های مختلفی وجود دارد که به توضیح هر یک از آن‌ها خواهیم پرداخت.

۱-۱-۲ یادگیری متمرکز

این روش که در اکثر سیستم‌های حال حاضر امروزی مورد استفاده قرار می‌گیرد به این نحو است که تمام گره‌ها اطلاعات موجود خود را به صورت کامل به سیستم سرویس‌دهنده ابری ارسال می‌نمایند و سرویس‌دهنده ابری در حالی که تمام داده‌ها را در اختیار دارد، اقدام به اجرای الگوریتم‌های مورد نظر می‌کند. در شکل ۱-۲ این روش به نمایش گذاشته شده است.



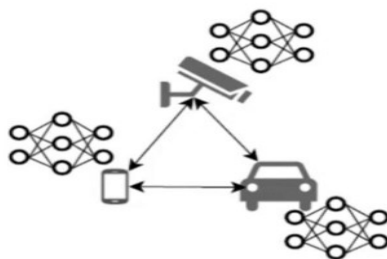
شکل ۱-۲: یادگیری متمرکز [۱۹]

سیستم‌های متمرکز تا پیش از این، اکثر نیازهای مربوطه را برطرف نموده‌اند ولی در دنیای امروزی و با توجه به زیاد شدن هر روزه دستگاه‌های متصل، موارد دیگری نیز مورد توجه واقع شده است. هزینه‌های ارتباطی ارسال حجم وسیع داده از یک سمت و نگرانی‌ها حول انتقال اطلاعات حساس و شخصی از سمت دیگر، توجه محققان را به سمت الگوریتم‌های غیر متمرکز و توزیع شده در یادگیری ماشین سوق داده است.

این روش دارای نقاط قوت و ضعف مختلفی است. از نقاط قوت این روش می‌توان به سرعت و دقت بالا، همگام‌سازی و یکنواختی داده‌ها و حفاظت و امنیت بالای داده‌ها اشاره کرد. از نقاط ضعف این روش می‌توان به وابستگی به هزینه و پایداری سرویس‌دهنده ابری، حساسیت به حجم و نوع داده‌های منتقل شده و چالش برانگیزی در برابر تغییرات محیط ذکر کرد. این روش در بسیاری از سرویس‌های آنلاین مورد استفاده قرار می‌گیرد.

۲-۱-۲ یادگیری غیرمتمرکز

در این روش هر گره به صورت مجزا اقدام به اجرای الگوریتم‌های مورد نظر می‌کند و در واقع پس از اجرای چند مرحله از کد، اطلاعات به‌روز شده را با گره‌های همسایه به اشتراک می‌گذارد. این کار به قدری ادامه پیدا می‌کند تا همگی به مقدار تعیین شده همگرا شوند. در شکل ۲-۲ این روش به نمایش گذاشته شده است.



شکل ۲-۲: یادگیری غیر متمرکز [۱۹]

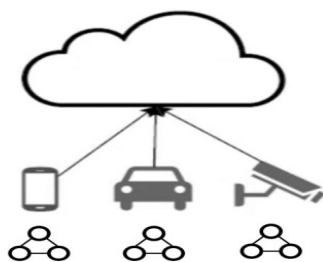
از مزایا و معایب این روش یادگیری این است که به گره‌ها اجازه می‌دهد که با توجه به شرایط و پارامترهای محلی خود، الگوریتم‌های خود را اجرا، تنظیم و بهینه‌سازی کنند. این روش همچنین باعث می‌شود که بار پردازشی و هزینه انتقال داده‌ها کاهش یابد و حریم خصوصی و حفاظت از داده‌ها حفظ شود. اما این روش نیز دارای چالش‌هایی است. از جمله سرعت و دقت پایین‌تر در پردازش داده‌ها، ناهمگام‌سازی و نامنظم بودن داده‌ها بین گره‌ها و نیاز به تضمین و تأیید صحت و کامل بودن داده‌ها. این روش در برخی از سیستم‌های آنلاین مانند بیت‌کوین^۱ و تورنت^۲ مورد استفاده قرار می‌گیرد.

۳-۱-۲ یادگیری توزیع شده

در این روش، مدیریت کل سیستم و تمام داده‌ها در اختیار یک هسته مرکزی قرار دارد ولی به دلیل نیاز به توان پردازشی بالا، این هسته بار پردازشی را بین گره‌های موجود تقسیم می‌کند. در ابتدای راه یادگیری توزیع شده، فرض بر این بوده است که تمام گره‌ها توان پردازشی یکسانی داشته و داده‌ها به میزان مساوی بین گره‌ها پخش خواهند شد. در شکل ۳-۲ این روش به نمایش گذاشته شده است.

^۱Bitcoin

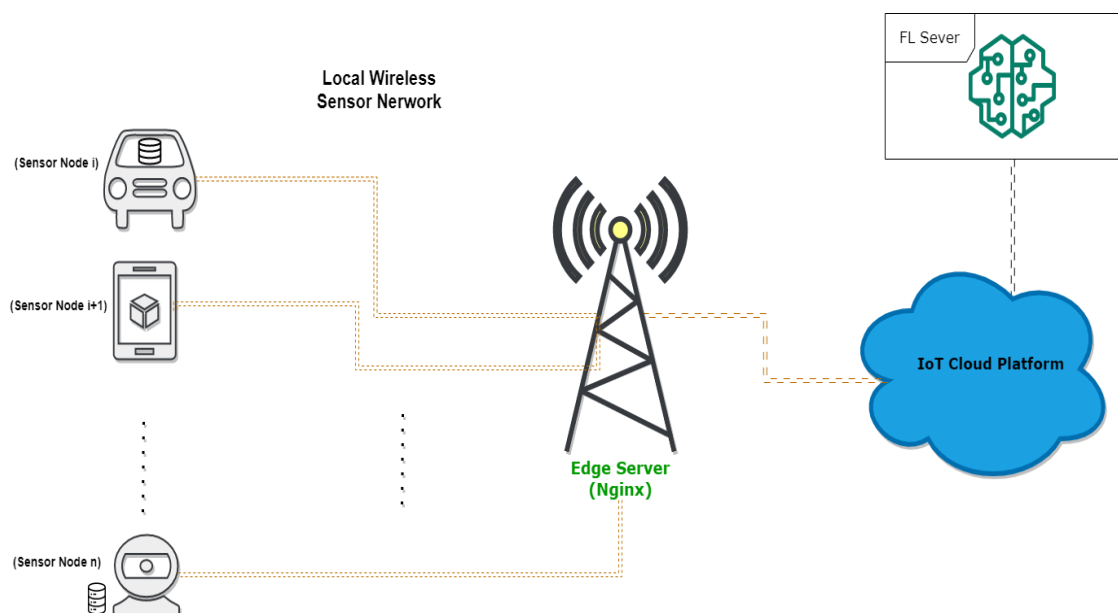
^۲Torrent



شکل ۲-۳: یادگیری توزیع شده [۱۹]

۴-۱-۲ یادگیری فدرال

همان‌طور که در فصل اول اشاره شد، در یادگیری فدرال بر خلاف روش‌های متمرکز یادگیری ماشین، تحلیل داده‌ها به دستگاه‌های لبه، گره و یا سرور گیرنده منتقل می‌شود. یادگیری فدرال راه‌حلی مطلوب برای مدل‌سازی داده‌ها در تعداد زیادی دستگاه کارگر است. در این چارچوب، به جای ارسال داده‌های خام، پارامترهای مدل‌های محلی در هر گام آموزش به سرور منتقل می‌شود. در شکل ۲-۴ این روش به نمایش گذاشته شده است. سرور در حقیقت نقش رهبری را ایفا می‌کند و با توجه به نوع داده‌ها، یا مدل شبکه عصبی ایجاد کرده و آن را به سمت کاربران ارسال می‌کند. حال کاربران با توجه به داده‌های خود شبکه را آموزش می‌دهند و بعد از چند بار تکرار، وزن‌های به‌روزرسانی شده را به سمت سرور بر می‌گردانند. همان‌طور که در شکل ۲-۴ مشاهده می‌شود، داده‌ها همگی در سمت کاربر قرار گرفته‌اند و به سمت سرور ارسال نمی‌شوند. عدم ارسال بخش اطلاعات گره‌ها در یادگیری فدرال، حفظ حریم شخصی کاربران را ارتقا می‌بخشد.



شکل ۲-۴: شمایی از یادگیری فدرال در اینترنت اشیا

معمولاً برای تخمین و سنجش عملکرد روش‌های یادگیری ماشینی از مدل‌های شبکه عصبی استفاده می‌شود که کاربرد وسیعی در اکثر زمینه‌ها مانند پردازش تصویر، دسته بندی، پیش‌بینی و غیره دارند. امروزه با آمدن روش‌هایی آموزش و توسعه این روش را از همیشه ساده‌تر شده است. علی‌رغم وجود مدل‌های گوناگون به عنوان مدل سراسری در یادگیری فدرال، اکثر استفاده‌های آن معطوف به مدل شبکه عصبی و به طور خاص تر شبکه‌های عصبی عمیق^۱ در پردازش تصویر به عنوان یکی از الزامات اساسی در اینترنت اشیاء می‌باشد. البته باید قبول کرد که در دسترس بودن و سادگی کار با آن‌ها نیز از دیگر عوامل روی آوردن به آن‌ها می‌باشد. در ادامه به یکی از مطرح‌ترین استفاده از این مدل‌های دسته بندی کننده می‌پردازیم.

۲-۱-۵ الگوریتم FedAVG

از مطرح‌ترین و ساده‌ترین روش‌های ادغام کردن پارامترهای ارسالی (به عنوان مثال وزن‌ها و بایاس‌های شبکه عصبی) توسط تجميع کننده الگوریتم میان‌گیری از آن‌ها با وزن یکسان بین تمامی کارگران می‌باشد. این الگوریتم با نام Federated Averaging برای اولین بار در [۵] مطرح شده است. در نسخه‌های اولیه یادگیری فدرال، این الگوریتم به صورت متمرکز درون تجميع کننده در انتهای هر دور از یادگیری اجرا می‌شود تا در نهایت تمام کارگران با وزن یکسان در فرآیند یادگیری سهمیه باشند. در ادامه شبه‌کد دو تکه اصلی مورد نیاز از اجرای این الگوریتم در تجميع کننده و کارگران آورده شده است.

از مهم‌ترین ایراداتی که می‌توان به این نوع الگوریتم گرفت این است که به علت ناهمگونی داده‌ها بین کارگران نباید در میان‌گیری وزن یکسانی به آنها داد. به همین دلیل طی سال‌های اخیر مدل‌های پیشرفته‌تری از این استراتژی معرفی شده است [۲۳].

الگوریتم ۱ بروزرسانی مشترک: اجرا در هر کارگر

۱: ورودی: بردار وزن مدل w ، اندازه دسته‌های کوچک محلی B	
۲: خروجی: بردار وزن مدل به‌روزشده w'	
۳: function بروزرسانی_مشترک(w, B)	
۴: تقسیم P_k به دسته‌هایی به اندازه B : (P_k, B) تقسیم به \leftarrow بچها	
۵: for $i = 1$ تا E do	
۶: do در for	
۷: $w \leftarrow w - \eta \cdot \nabla f(w,)$	
۸: return w	

▷ عبورهای آموزشی محلی

▷ بروزرسانی وزن‌های مدل

¹ Deep Neural Network

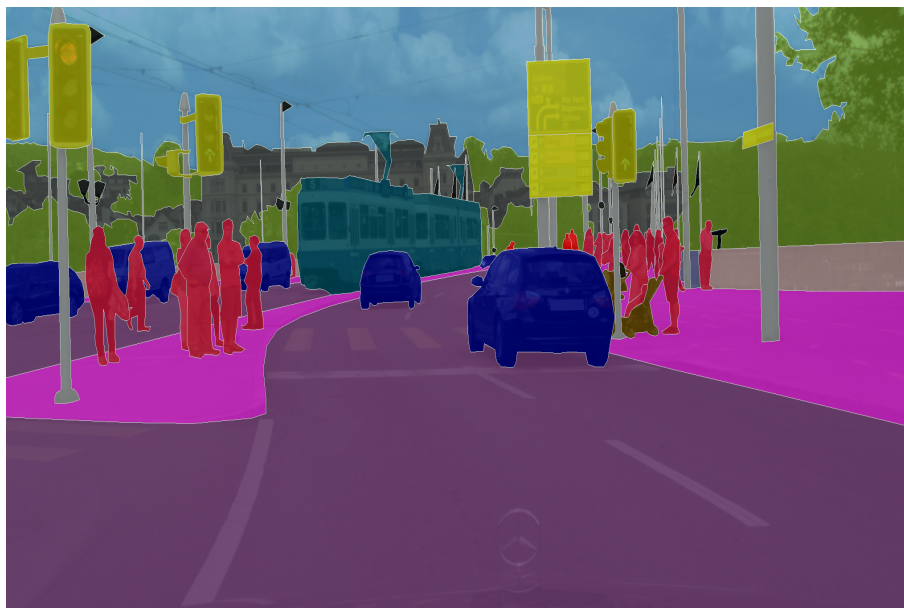
الگوریتم ۲ میانگین‌گیری مشترک: اجرا در سرور

۱:	ورودی: نرخ یادگیری جهانی η ، تعداد کارگرها در هر دور C ، تعداد عبور آموزشی محلی E ، اندازه مینی‌بچ محلی B
۲:	خروجی: بردار وزن مدل میانگین w_{avg}
۳:	function میانگین‌گیری-مشترک(η, C, E, B)
۴:	مقداردهی اولیه بردار وزن مدل: $w \leftarrow w_0$
۵:	for $t = 1$ تا ∞ do
۶:	انتخاب به صورت تصادفی C کارگر: (C) انتخاب کارگرها \leftarrow
۷:	for k در C do
۸:	انجام بروزرسانی مشترک برای کارگر k : (w, B) بروزرسانی-مشترک $w_k \leftarrow$
۹:	بروزرسانی بردار وزن مدل برای کارگر k : (w, w_k) تجمع وزن‌ها $w \leftarrow$
۱۰:	محاسبه میانگین وزن‌های تمام کارگرها: (w, C) محاسبه میانگین وزن‌ها $w_{avg} \leftarrow$
۱۱:	return w_{avg}

۲-۲ شبکه عصبی کانولوشنی

مدل کردن پردازش‌هایی که انسان قادر به انجام آن بر روی تصاویر است (برای مثال تشخیص هویت با استفاده از حس بینایی) با یک برنامه کامپیوتری از دستاوردهای چندین ساله محققین حوزه تصویر بوده است. با معرفی بخش‌بندی معنایی^۱ و شبکه‌های عمیق تصور شد که این مشکلات با استفاده از این روش جدید قابل حل شدن هستند. ولی مشکل بزرگ در مسیر استفاده شبکه‌های عمیق برای پردازش تصاویر، هزینه محاسبات این روش برای استفاده بر روی حجم زیاد داده‌ای که یک تصویر را تشکیل می‌دهد است. در حالت متداول شبکه‌های عمیق برای محاسبه مقدار خروجی هر نورون، خروجی تمام نورون‌های لایه قبل استفاده می‌شود و شیوه استفاده از خروجی هر یک از نورون‌های قبل با یک پارامتر مشخص می‌شود. یک تصویر نمونه با تراکم پیکسلی نسبتاً کم، برای مثال 256×256 را می‌توان به صورت یک بردار به طول ۶۵۵۳۶ در نظر گرفت. با توجه به نحوه پیچیدگی شبکه‌های عصبی عمیق می‌توان درک کرد که حرکت داده‌ای با این حجم در لایه‌های یک شبکه عمیق می‌تواند به چه اندازه از نظر محاسبات سخت و سنگین باشد.

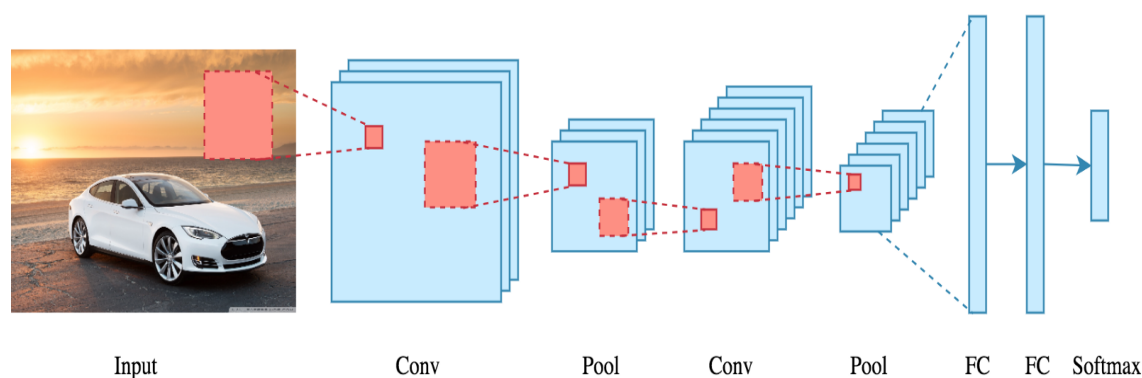
^۱ Semantic Segmentation



شکل ۲-۵: نمونه‌ی بخش بندی معنایی در ماشین‌های خودران [۲۰]

برای عبور از این مانع، شبکه‌ای جدید با محوریت پردازش داده‌های تصویری (به طور کلی تر پردازش سیگنال) طراحی شد و این الگوریتم جدید را شبکه‌های عصبی کانولوشن^۱ نام‌گذاری کردند. ایده اصلی این طرح جدید در این نکته بود که بهتر است برای محاسبه خروجی هر نورون از خروجی تمام نورون‌های لایه قبل استفاده نشود؛ بلکه کافی است فقط از همسایگان محدود از نورون‌های همسایه استفاده شود. در نتیجه هزینه محاسبات برای پردازش و مهم‌تر از آن تعداد پارامترهای مدل کاهش می‌یابد. هر لایه از این شبکه (در ساده‌ترین حالت) با یک کرنل تعریف می‌شود که مقادیر آن به عنوان پارامترهایی قابل تغییر فرض می‌شود که شبکه در طول فرآیند یادگیری آنها را تنظیم می‌کند. ورودی هر لایه تصویر (به عبارت دیگر نقشه ویژگی) و خروجی آن نیز به همین ترتیب است؛ ولی لزوماً رزولوشن این دو یکسان نیست. این کرنل بر روی تک‌تک پیکسل‌های تصویر اعمال می‌شود و تصویر خروجی تولید می‌شود. به این عملیات فیلتر کردن در شبکه عصبی کانولوشن می‌گویند. در شکل ۲-۶ معماری یک شبکه کانولوشن به عنوان نمونه آورده شده است.

^۱Convolutional Neural Networks



شکل ۲-۶: معماری یک شبکه کانولوشنی

این شبکه پس از معرفی توانست محبوبیت خاصی را بین محققان به دست آورد و همچنین مقدمه‌ای بر ساخت انواع پیشرفته‌تر از شبکه‌های عصبی برای عملیات مختلفی مانند تشخیص و شناسایی چهره^۱، بخش‌بندی معنایی تصویر، تشخیص اشیاء، دسته‌بندی و گروه‌بندی تصاویر و غیره بشود. [۲۱]

۳-۲ پردازش لبه

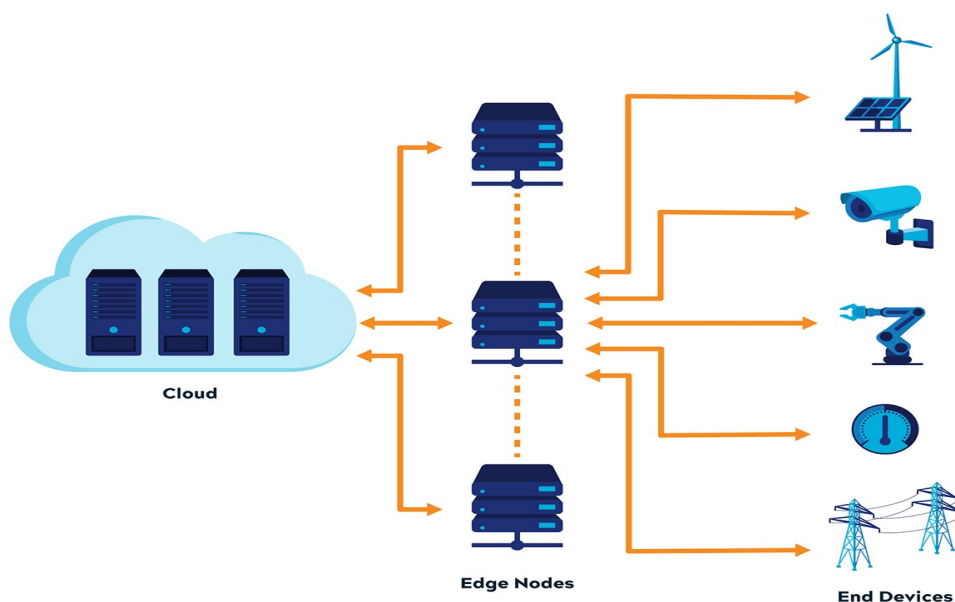
در روش‌های مبتنی بر الگوی ابری، داده‌ها باید از طریق اینترنت برای یک مرکز داده متمرکز ارسال شود تا در آنجا پردازش شوند و نتیجه برای منبع بازگردانده شود. این روش برای حجم محدود و مشخصی از داده‌ها عملکرد خوبی دارد، اما هنگامی که حجم عظیمی از داده‌ها قرار باشد برای مراکز داده ارسال شود، در آنجا پردازش شود و نتیجه برای منبع بازگردانده شود مناسب نیست. زیرا نیاز به پهنای باند زیاد وجود دارد که مقرون به صرفه نیست. همچنین مشکلات تاخیر و قطعی‌های غیرقابل پیش‌بینی شبکه، همگی می‌توانند باعث اختلال در عملکرد ارسال، دریافت و پردازش داده‌ها شوند. پژوهشگران برای حل این مشکل معماری پردازش لبه را ارائه کرده‌اند. پردازش لبه یک معماری توزیع شده است که در آن داده‌های کاربر در لبه شبکه و تا حد امکان نزدیک به دستگاه‌های انتهایی پردازش می‌شود. آمارها نشان می‌دهند که ساختارهای مبتنی بر پردازش لبه در حال تغییر الگوی پردازش اطلاعات هستند و این احتمال وجود دارد که در آینده تغییرات مهمی در حوزه یادگیری به خصوص یادگیری‌های توزیع شده به وجود آورند [۱۶].

در اصطلاح لبه شبکه به مکانی اشاره دارد که در آن داده‌ها تولید می‌شوند و تجهیزات محاسباتی در آنجا نصب شده‌اند. در محاسبات سنتی سازمانی، داده‌ها در سرورهای مرکزی ذخیره می‌شوند و از طریق شبکه محلی در اختیار کاربران قرار می‌گیرند. به عبارت دیگر، داده‌ها در زیرساخت‌های سازمانی ذخیره و پردازش شده و نتایج پردازش به کاربران ارسال می‌شود. این معماری بر اساس الگوی کلاینت و سرور است که بسیاری از برنامه‌های

^۱ Face Recognition

تجاری بر اساس آن عمل می‌کنند. با این حال، با پیدایش پردازش لبه، داده‌ها در نزدیک‌ترین مکان به منبع تولید آن‌ها پردازش شده و نتایج به کاربران ارسال می‌شود. این روش کمک می‌کند تا هزینه‌های ارتباطی کاهش یابد و حفظ حریم شخصی کاربران بهتر از قبل رعایت شود.

از آنجا که، تعداد دستگاه‌های متصل به اینترنت و حجم داده‌هایی که توسط دستگاه‌ها تولید می‌شوند و توسط کسب‌وکارها استفاده می‌شوند، فراتر از ظرفیت زیرساخت‌های مراکز داده سنتی است. یک مثال ساده در این زمینه، داده‌های تولیدشده در شبکه‌های اتومبیل‌های هوشمند است که به لحاظ تجاری و بازاریابی ارزش زیادی دارند و توسط خودرو تولید می‌شوند که عضو شبکه‌های اتومبیلی هستند. در سویی دیگر، داده‌های حساس به زمان وجود دارند که توسط تجهیزاتی مثل دوربین‌های نظارت تصویری ضبط می‌شوند و تصاویر از طریق اینترنت برای اپراتوری که مسئولیت نظارت بر دوربین‌ها را برعهده دارد ارسال می‌شود تا اگر مورد مشکوکی بود، اپراتور واکنش لازم را انجام دهد. در این روش نه تنها به پهنای باند زیادی برای ارسال داده‌ها نیاز است، بلکه باید اپراتور به سرعت به موارد مشکوک واکنش نشان دهد. حال اگر داده‌های تصویری به شکل محلی توسط الگوریتم‌های هوشمند تحلیل شده و موارد مشکوک در قالب یک پیام متنی ساده برای اپراتور ارسال شود، به میزان قابل توجهی در پهنای باند صرفه‌جویی انجام می‌شود و زمان پاسخ‌گویی به رخدادها نیز کاهش پیدا می‌کند. در این حالت، فشار مضاعف بر اینترنت یا شبکه‌های گسترده وارد نمی‌شود و با مشکل ازدحام^۱ و اختلال^۲ روبه‌رو نخواهد شد.



شکل ۲-۷: دیاگرام یک شبکه دارای پردازش لبه

همین مسئله باعث شده تا معماران شبکه‌های کامپیوتری به جای طراحی مراکز داده متمرکز، به سراغ طراحی‌های

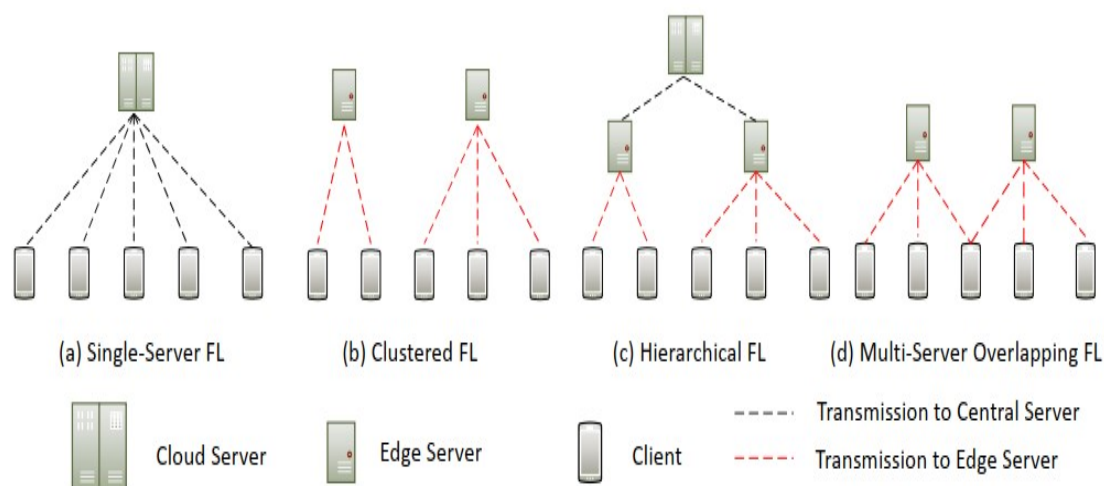
^۱ Congestion

^۲ Disruption

مبنتی بر الگوی پردازش لبه بروند. به طوری که منابع ذخیره سازی و محاسباتی از مرکز داده به مکانی انتقال داده شود که نزدیک به منبع تولیدکننده داده ها است. دانستن این موضوع جالب خالی از لطف نیست که پردازش لبه بر مبنای یک تئوری خیلی ساده شکل گرفته است، اگر نمی توانید داده ها را به مرکز داده نزدیک کنید [۱۷]، مرکز داده را به داده ها نزدیک کنید.

۲-۳-۱ معماری های متنوع پردازش لبه

همان طور که در شکل ۲-۸ آمده است، می توان معماری های متنوعی برای حضور سرور لبه در شبکه اینترنت اشیاء متصور شد. بسته به کاربرد و محل فیزیکی دستگاه ها می توان یکی از این معماری ها را برگزید. در این پروژه به دلیل ساده تر بودن و فراگیر بودن در کاربرد از معماری سلسله مراتبی استفاده شده است.

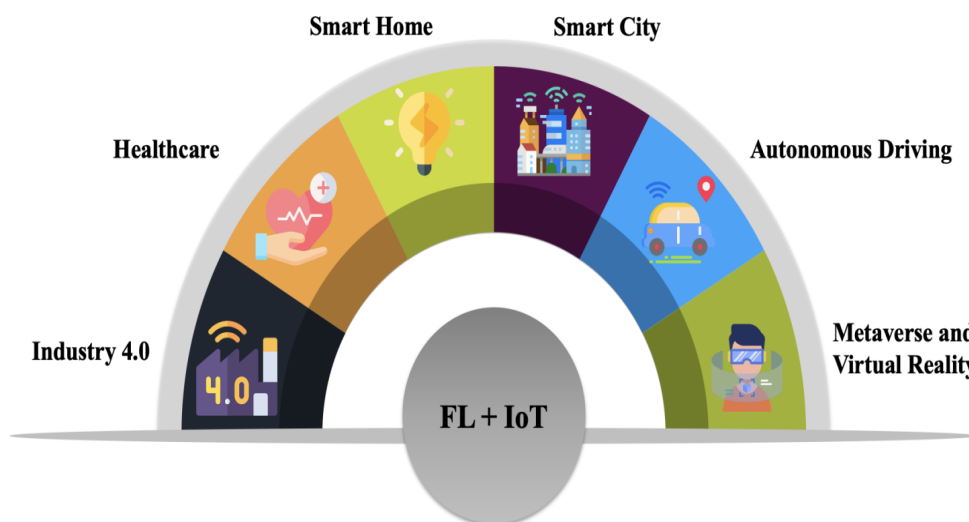


شکل ۲-۸: انواع قرار گیری المان های اصلی در شبکه اینترنت اشیاء [۳]

۲-۴ یادگیری فدرال در اینترنت اشیاء

شبکه های نوظهور اینترنت اشیاء همچون دستگاه های پوشیدنی، خودروهای خودران یا خانه های هوشمند شامل تعداد بسیار زیادی حسگر هستند. این حسگرها توانایی جمع آوری عکس العمل و سازگاری با داده ها برای کاربرد بی درنگ را برای دستگاه های اینترنت اشیاء فراهم می سازند. به عنوان مثال، خودروهای خودران به منظور عملکرد صحیح نیازمند یک مدل به روز از ترافیک شهری، اماکن و رفتار افراد پیاده هستند که این داده ها را از حسگرها دریافت می کند. ساخت مدلی سراسری در این حوزه به دلیل عدم تمایل افراد به در اختیار گذاشتن اطلاعات فردی، مانند اطلاعات مکانی و محدودیت در ارتباطات هر دستگاه بسیار مشکل است. از این رو، روش های یادگیری فدرال می توانند مدلی سازگار با تغییرات ایجاد نمایند و این سیستم ها را در عین حفظ حریم

شخصی به خوبی آموزش دهند. شکل ۲-۹ بیانگر تنها بخشی از حوزه‌هایی است که در آینده‌ای نه چندان دور از ترکیب این دو حوزه به وجود می‌آید.



شکل ۲-۹: کاربردهای یادگیری فدرال در اینترنت اشیا [۳]

۲-۵ جمع بندی

ایجاد بستری که تمامی موارد بالا در آن نقش پررنگی ایفا می‌کنند، می‌تواند محیط خوبی برای توسعه برنامه‌های مختلف با توجه به نیازهای کاربر باشد. این پروژه تنها بخشی از این نیازها را پوشش می‌دهد که کاربرد نسبتاً بالایی در دسته‌بندی تصاویر را ارائه می‌کند. در اینجا، سعی می‌شود با استفاده از ساده‌سازی‌های صورت گرفته، به تأکید بر پیاده‌سازی سیستمی که از یادگیری فدرال بهره می‌برد پرداخته شود. در نهایت با توجه دانش قبلی در مورد موارد ذکر شده در این فصل و بهره‌گیری از شبیه‌سازی‌های متنوع، به دسته‌بندی داده‌های مجموعه داده CIFAR-۱۰ با استفاده از یک شبکه عصبی کانولوشن روی یک بستر شبکه اینترنت اشیا که در آن از پردازش لبه استفاده شده خواهیم پرداخت. در فصل بعدی به جزئیات بیشتر این شبیه‌سازی پرداخته می‌شود.

فصل سوم

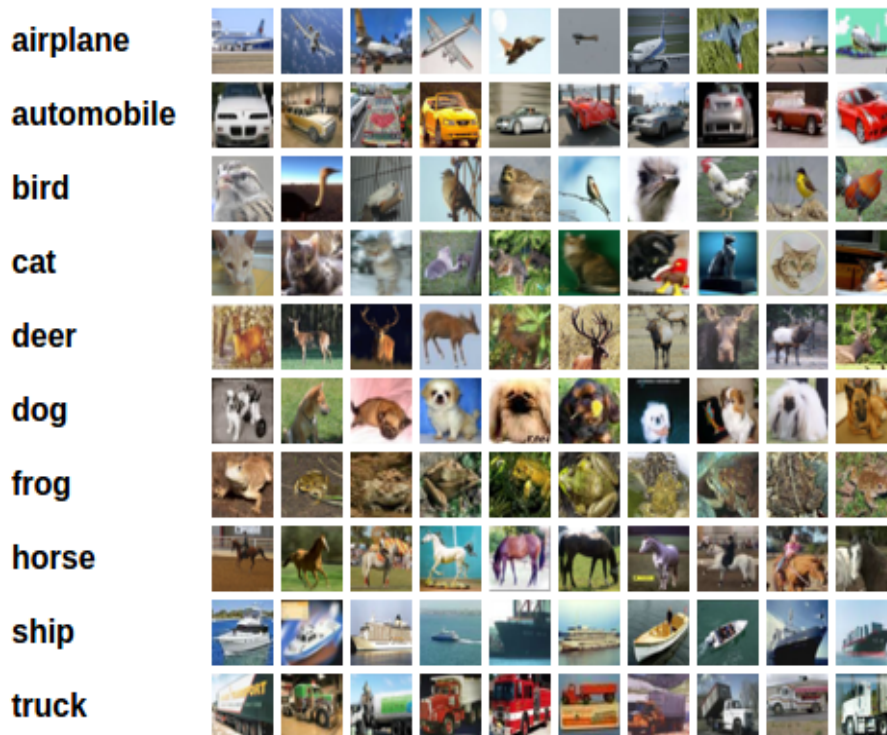
چارچوب کاری پروژه

این فصل به معرفی داده‌های مسئله و همچنین معرفی و بررسی کامل مدل پیشنهادی مبتنی بر معماری شبکه عصبی کانولوشن می‌پردازد. مضاف بر این، چارچوب^۱ استفاده شده و محیط شبیه‌سازی نیز مورد بحث قرار می‌گیرد.

۳-۱ مجموعه داده تعریف شده در مسئله

این بخش، به بررسی داده‌های مورد استفاده در آموزش شبکه‌های عصبی و مدل پیشنهادی برای حل مسئله می‌پردازد. این داده‌ها و مدل پیشنهادی برای حل مسئله، بر اساس مجموعه داده CIFAR-۱۰ و کتابخانه PyTorch تعریف شده‌اند. مجموعه داده CIFAR-۱۰ شامل ۶۰,۰۰۰ تصویر رنگی ۳۲ در ۳۲ پیکسل در ۱۰ کلاس مختلف با ۶۰۰۰ تصویر در هر کلاس است. این کلاس‌ها شامل هواپیما، اتومبیل، پرنده، گربه، گوزن، سگ، قورباغه، اسب، کشتی و کامیون هستند. از این تعداد تصاویر، ۵۰,۰۰۰ تصویر برای آموزش و ۱۰,۰۰۰ تصویر برای تست استفاده می‌شود. شکل ۳-۱ نمایی از این کلاس‌ها را نشان داده است.

^۱ Framework



شکل ۳-۱: نمونه‌ای از تصاویر دیتاست CIFAR۱۰

برای آموزش شبکه عصبی، از کتابخانه PyTorch استفاده می‌شود. PyTorch یک کتابخانه یادگیری عمیق با قابلیت‌های بالا برای آموزش شبکه‌های عصبی است. در این پروژه، یک شبکه عصبی کانولوشنی تعریف شده و با استفاده از داده‌های آموزش CIFAR-۱۰ آموزش دیده است. برای تعریف شبکه عصبی و آموزش آن، از تابع هزینه CrossEntropyLoss و الگوریتم بهینه‌سازی نزول گرادیان تصادفی^۱ با نرخ یادگیری 0.001 استفاده شده است. تابع هزینه CrossEntropyLoss یک تابع هزینه مناسب برای مسائل طبقه‌بندی چند کلاسه است.

۳-۲ مدل استفاده شده برای حل مسئله

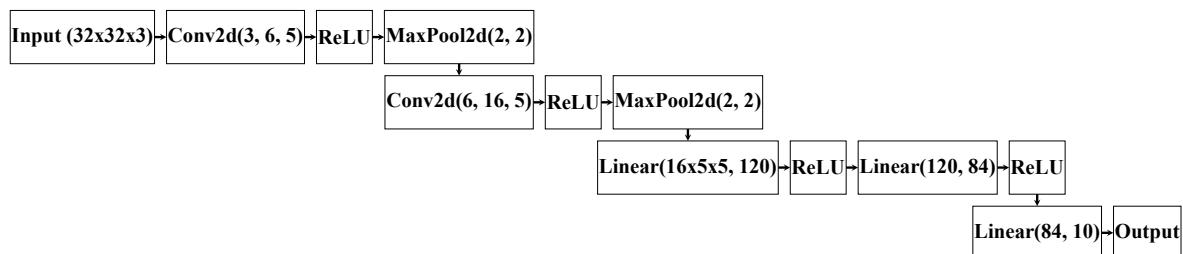
این مسئله از یک شبکه عصبی کانولوشن نسبتاً ساده بهره می‌گیرد. تابع فعال‌ساز^۲ استفاده شده در هر لایه آن ReLU است که باعث پایداری شبکه می‌شود و رفتار خطی از خودش نشان می‌دهد. شبکه دارای دو لایه کانولوشن و سه لایه تماماً متصل^۳ است. در هر لایه کانولوشن، از یک لایه MaxPool_{۲d} برای کاهش ابعاد ویژگی‌ها استفاده می‌شود. سپس در هر مرحله ویژگی‌های سطح بالاتر استخراج می‌شوند تا نوبت به لایه‌های

^۱Stochastic Gradient Decent

^۲Activation Function

^۳Fully Connected Layer

مخصوص به دسته‌بندی برسد. در نهایت بعد از استخراج ویژگی‌های منحصر به فرد هر تصویر و پیدا شدن مقدار احتمال حضور در هر یک از دسته‌ها، خروجی از طریق سه لایه تماماً متصل به یک بردار با ۱۰ عنصر تبدیل می‌شود که نشان‌دهنده احتمالات دسته‌بندی مختلف برای هر یک از کلاس‌های مجموعه داده است. نمودار مربوط به این شبکه در شکل ۳-۲ قابل مشاهده است.



شکل ۳-۲: ساختار شبکه عصبی کانولوشن استفاده شده در مسئله ۱۰ کلاسه

برای بهبود عملکرد شبکه، می‌توان از روش‌های مختلفی استفاده کرد. برای مثال، می‌توان تعداد فیلترها در لایه‌های کانولوشن را افزایش داد، یا از روش‌های منظم‌سازی مانند Dropout یا Batch Normalization استفاده کرد. همچنین، می‌توان تابع فعال‌ساز ReLU را با توابع فعال‌ساز دیگری مانند LeakyReLU یا Sigmoid جایگزین کرد. در نهایت، می‌توان شبکه را با داده‌های آموزش بسیار بیشتر و با استفاده از روش‌های بهینه‌سازی پیشرفته‌تر آموزش داد.

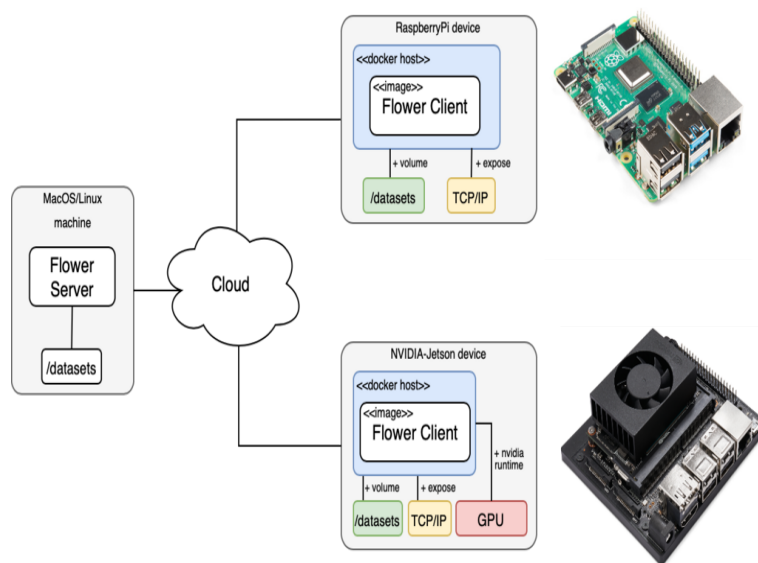
۳-۳ محیط توسعه استفاده شده

محیط‌های توسعه متعددی برای پیاده‌سازی سناریوهای مختلف تولید شده است. به عنوان مثال در تراز صنعتی، چارچوب FATE توسط محققین شرکت Facebook توسعه پیدا کرده است [۷]. نقطه قوت این محصول امنیت و ایمنی بالا و قابل اطمینان برای کاربردهای کلان صنعتی می‌باشد. محیط دیگری که برای کاربردهای عملی و سهولت در استفاده توسط مهندسين آلمانی توسعه یافته است، Flower نام دارد [۶].

FATE و Flower هر دو چارچوب‌های منبع‌باز^۱ هستند که به توسعه‌دهندگان اجازه می‌دهند سناریوهای گوناگون یادگیری فدرال را پیاده‌سازی کنند. FATE روی فراهم آوردن چارچوب محاسبات امن برای پشتیبانی از اکوسیستم هوش مصنوعی فدرال تمرکز دارد، در حالی که Flower در هدف فراهم آوردن رویکردی کاربرپسند به یادگیری فدرال که با هر چارچوب یادگیری ماشین و زبان برنامه نویسی سازگار است، قرار دارد. هر دوی این محیط‌ها ویژگی‌ها و قابلیت‌های منحصر به فرد خود را دارند، به طوری که توسعه‌دهندگان می‌توانند از بین این

^۱ Open Source

دو یکی را که بهترین گزینه برای نیازهای آن‌ها است، انتخاب کنند. در این پروژه به دلیل سهولت دسترسی و دسترسی در محیط‌های شبیه‌سازی مختلف و فراهم بودن مثال‌های متعدد از سناریوهای پیش‌فرض از این محیط به عنوان ابزاری برای پیاده‌سازی یادگیری فدرال در اینترنت اشیاء استفاده شده است. قابل ذکر است که این محیط نیازهای مربوط به پیاده‌سازی شرایط متنوع را به خوبی برطرف می‌سازد. از ویژگی‌های بارز این محیط می‌توان به سازگاری کامل با کتابخانه‌ها و محیط‌های دیگر توسعه یادگیری ماشین و هوش مصنوعی از جمله Tensorflow و Pytorch اشاره نمود.

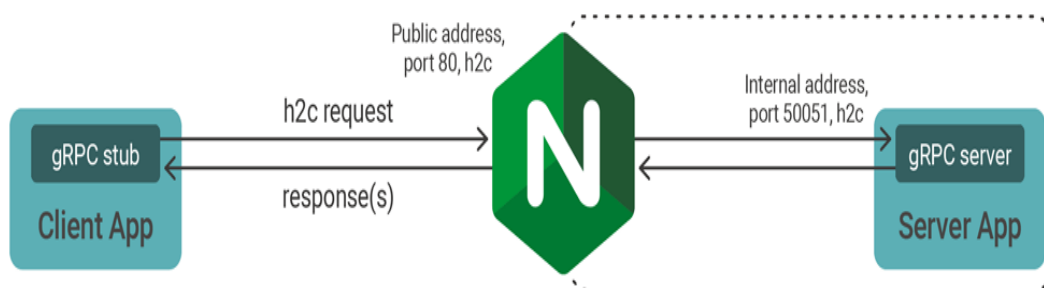


شکل ۳-۳: نمونه‌ای از پیاده‌سازی یادگیری فدرال روی دستگاه‌های تعبیه شده

برای پیاده‌سازی سرور لبه در دنیای واقعی نیز، مشابه سرور عمل می‌شود با این تفاوت که توان پردازشی می‌تواند تا حد امکان به اندازه‌ای که وقفه‌ای در مرحله یادگیری و آزمون صورت نگیرد، ساده‌تر در نظر گرفته شود. الزاماً فرض بر این است که سرور لبه و سرور تجمیع‌کننده هر دو همیشه در شبکه فعال هستند و دائماً در حال گوش دادن روی یک پورت خاص به کارگران درخواست‌دهنده هستند. از طرف دیگر، کارگران که دستگاه‌های انتهایی هستند، می‌توانند با توجه به ماهیت کم‌توان بودن برای مصرف بهینه انرژی، به خواب‌های عمیق یا کوتاه‌مدت بروند. در عین حال سرور لبه می‌تواند به جای آن‌ها پاسخگوی سرور تجمیع‌کننده باشد تا گره دستگاه کم‌توان مجدداً به شبکه اضافه شود و پارامترهای خود را ارسال کند.

۴-۳ پروتکل‌های انتقال داده در شبکه

تقریباً تمامی چارچوب‌های یادگیری فدرال از پروتکل امن gRPC^۱ استفاده می‌کنند. gRPC یک چارچوب متن‌باز و پر قدرت برای تماس‌های رویه‌ای از راه دور (RPC) است که توسط گوگل ایجاد شده است. این چارچوب قابلیت اجرا در هر محیطی را دارد و می‌تواند به طور کارآمد خدمات را درون و بین مراکز داده با پشتیبانی قابل‌تعویض برای توازن بار، ردیابی، بررسی سلامت و احراز هویت متصل کند. در این پروژه برای پردازش لبه از سرویس قدرتمند Nginx استفاده شده است تا بتواند خواسته‌های مربوط به Proxy سرور را در لبه شبکه پوشش دهد. از قابلیت‌های پیش‌فرض پلتفرم nginx پشتیبانی از این پروتکل است. همان‌طور که در شکل ۴-۳ نشان داده شده این پلتفرم می‌تواند به خوبی بسته‌های gRPC را انتقال بدهد. اما نکته حائز اهمیت آن است که ساختار بسته‌های gRPC باید در هر دو طرف کلاینت و سرور کاملاً مشخص شده باشد (اصطلاحاً protobuf). از آنجایی که nginx فاقد ساختار هر بسته است، تنها می‌تواند با استفاده از مشخصات داخل هدر فایل، اقدام به ارسال بسته کند. به عبارت دیگر، نمی‌توان به اطلاعات داخل بسته‌ها در وسط راه دسترسی پیدا کرد. البته از آنجایی که به صورت پیش‌فرض از این نوع بسته پشتیبانی می‌کند، قابلیت‌های خوبی در رابطه با مدیریت این نوع بسته قرار می‌دهد؛ مانند تعادل بار^۳ هنگام کار با چندین سرور به صورت همزمان و فشرده‌سازی^۴ بسته‌های رد و بدل شده در شبکه.



شکل ۴-۳: استفاده از nginx به عنوان پروکسی سرور

میدانیم که nginx به خودی خود یک server proxy می‌باشد که در نگاه بالاتر با داشتن یک سرور لبه از نوع gRPC می‌توانیم به مراتب عملیات بیشتری در شبکه انجام بدهیم و با داشتن دیتای ارسالی کلاینت و سرور در لبه، در جدیدی از قابلیت‌ها به روی مان باز خواهد شد.

^۱gRPC Remote Procedure Calls

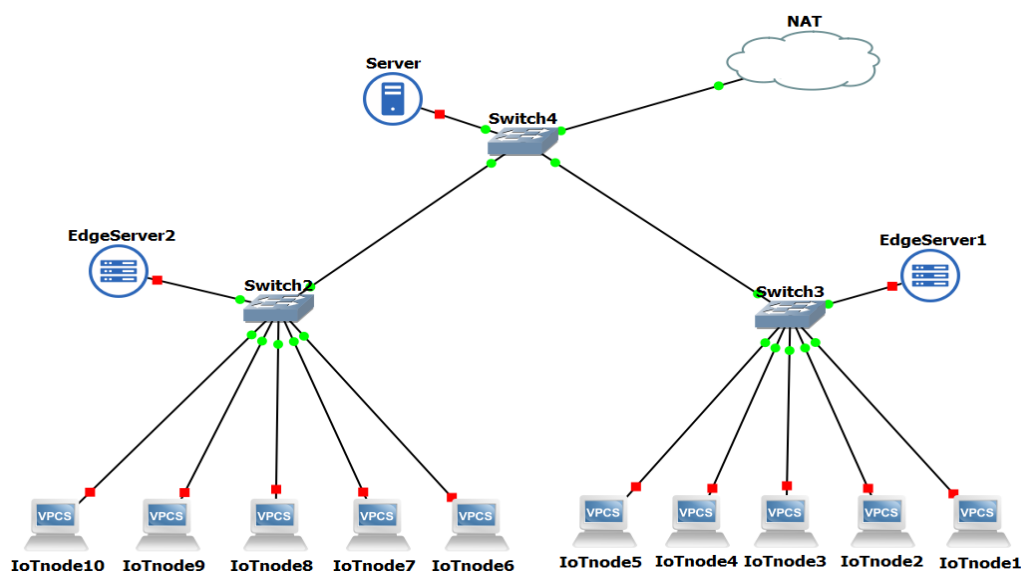
^۲Packet

^۳Load Balancing

^۴Compression

۵-۳ محیط شبیه‌سازی پیاده شده

برای پیاده‌سازی سناریو ذکر شده از محیط قدرتمند شبیه‌سازی GNS3 استفاده شده است. در این محیط از دستگاه‌هایی با توان پردازشی محدود به‌عنوان کارگر و یک سرور مرکزی به‌عنوان تجمیع‌کننده استفاده شده است. همچنین از دستگاه‌هایی با توان پردازشی کمتر از سرور اصلی تجمیع‌کننده اما بیشتر از کارگران به‌عنوان مرکز پردازشی لبه استفاده شده است. شبکه داخلی حاوی سرور اصلی، کارگران و سرور لبه به‌عنوان یک شبکه اینترنت اشیاء مستقل که می‌تواند از پروتکل‌های مطرح شبکه‌های سنسوری بی‌سیم مثل بلوتوث مش^۱ یا LORA^۲ استفاده کند، پشت یک NAT^۳ قرار دارد. تمامی این شبیه‌سازی‌ها بر روی یک لپ‌تاپ با ۱۶ گیگابایت رم و پردازنده ۸ هسته‌ای intel core i7 با فرکانس کاری ۳.۳ گیگاهرتز اجرا شده است. شکل ۵-۳ بیانگر تصویری از این شبیه‌سازی می‌باشد.

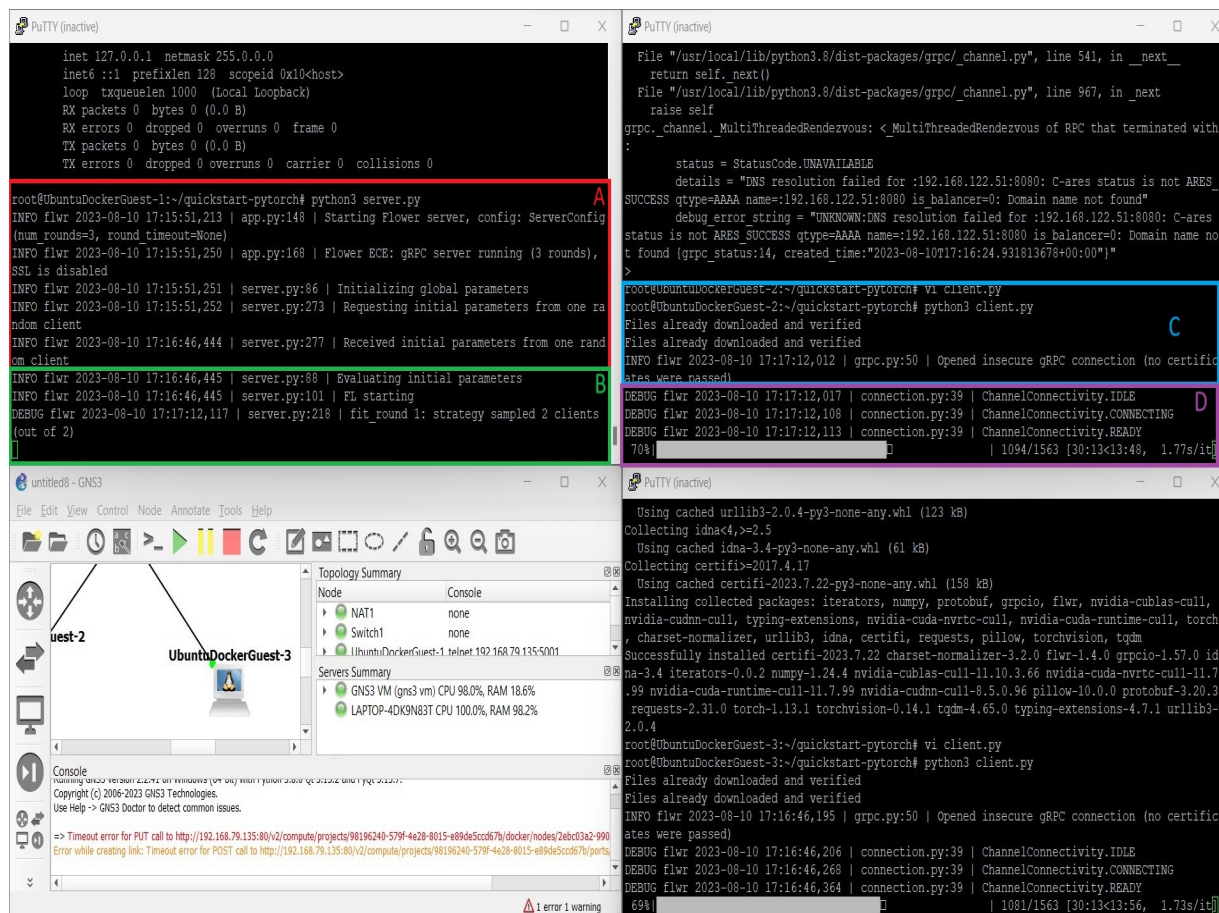


شکل ۵-۳: محیط شبیه‌سازی شده در محیط نرم‌افزار GNS3

^۱Bluetooth Mesh Sensor Networking

^۲Long Range Wide Area

^۳Network Address Translation



شکل ۳-۶: نمونه‌ای از اجرای فرایند یادگیری و تست توسط یک شبکه ساده دارای دو کارگر

باتوجه به شکل ۳-۶، قسمت‌های A و B مخصوص ترمینال سرور اصلی هستند و C و D نیز برای یکی از کارگران آورده شده‌اند. در ابتدا با اجرا شدن قسمت A، سرور شروع به گوش دادن روی پورت و آدرس داده شده (در اینجا آی‌پی آدرس سرور هاست در شبکه داخلی و روی یکی از پورت‌های آزاد معمولاً ۸۰۸۰) سپس منتظر می‌ماند تا تعدادی کارگر که حداقل آنها از قبل در کد برنامه تعریف شده به سرور وصل بشوند. سپس با رسیدن این تعداد به حداقل، فرآیند یادگیری آغاز می‌شود. وزن‌ها و بایاس‌های شبکه عصبی نیز در ابتدا در دور صفر از یکی از کارگران به صورت تصادفی دریافت می‌شوند (معمولاً اولین کارگری که وصل می‌شود). در کارگران نیز ابتدا با اجرا شدن کد برنامه در محیط سیستم عامل (در اینجا Ubuntu)، ابتدا اقدام به دریافت فایل‌های مجموعه داده می‌کند که در اینجا از داده‌های CIFAR-۱۰ استفاده شده است. پس از مبادرت به چک‌های اولیه از لحاظ معتبر بودن مجموعه داده و پیش‌نیازهای لازم، نوبت به ساخت کانال امن با استفاده از پروتکل GRPC می‌باشد. پس از رد و بدل کردن پیام‌های اولیه بر بستر HTTP2، کانالی منحصر به فرد بین این کارگر و سرور اصلی ایجاد شده و تا پایان یادگیری باز می‌ماند. سپس بعد از وصل شدن کارگرهای بعدی به همین منوال،

فرآیند در تمامی دستگاه‌ها شروع می‌شود.

```
INFO flwr 2023-09-05 12:22:14,289 | grpc.py:50 | Opened insecure gRPC connection (no certificates were passed)
DEBUG flwr 2023-09-05 12:22:14,291 | connection.py:39 | ChannelConnectivity.IDLE
DEBUG flwr 2023-09-05 12:22:14,292 | connection.py:39 | ChannelConnectivity.CONNECTING
DEBUG flwr 2023-09-05 12:22:14,292 | connection.py:39 | ChannelConnectivity.READY
100% | 10000/10000 [00:21<00:00, 457.67it/s] X
100% | 1563/1563 [01:04<00:00, 24.08it/s] Y
100% | 10000/10000 [00:23<00:00, 426.88it/s]
100% | 1563/1563 [01:14<00:00, 21.12it/s]
100% | 10000/10000 [00:26<00:00, 379.75it/s]
100% | 1563/1563 [01:06<00:00, 23.41it/s]
```

شکل ۳-۷: نمونه‌ای از کنسول یک کارگر در حال اجرای یادگیری فدرال

شکل ۳-۷ نمونه‌ای از این فرآیند را در هر دور در هر کارگر با جزئیات بیشتری نشان می‌دهد. همان‌طور که مشخص است، مجموعه داده CIFAR-۱۰ شامل ۵۰ هزار تصویر آموزش و ۱۰ هزار تصویر تست است که باید توجه داشتیم که همان‌طور که از روی شکل شبیه‌سازی پیدا است، قسمت X بیانگر مرحله تست روی ۱۰ هزار عکس و مرحله Y بیانگر آموزش روی ۵۰ هزار تصویر این مجموعه داده می‌باشد. نکته قابل توجه این است که با توجه به اینکه اندازه هر batch برابر با ۳۲ است، با تقسیم ۵۰ هزار تصویر به ۱۵۳۲ دسته ۳۲ تایی می‌رسیم تا در فرآیند آموزش از آن استفاده شود.

نکته دیگر مورد اهمیت این است که همان‌طور که توقع داریم، سرعت فرآیند آموزش به دلیل به‌روزرسانی شدن وزن‌ها در مدل از فرآیند تست سریع تست و آخرین المان از هر خط که به صورت تعداد عملیات بر ثانیه^۱ آمده است قرار دارد، با توجه به محدود بودن سخت‌افزاری در سیستم‌های اینترنت اشیا، همان‌طور که از شکل ۳-۶ واضح است، گاهی این سرعت تا ۱ عملیات بر ثانیه کاهش پیدا می‌کند که این خود نشانی از اهمیت آن به عنوان گلوگاه اصلی در یادگیری فدرال در شبکه‌هایی با منابع محدود است.

^۱ Iteration Per Second

فصل چهارم

نتیجه‌گیری و پیشنهادها

در این فصل، به ارزیابی مدل در برابر معیارهای مختلف پرداخته می‌شود و دقت به دست آمده توسط مدل روی داده‌های تست گزارش می‌شود. لازم به ذکر است همان‌طور که گفته شد، در هر دور از یادگیری فدرال، ۵۰۰۰۰ تصویر برای آموزش و ۱۰۰۰۰ تصویر برای ارزیابی در نظر گرفته شده است.

۴-۱ معرفی معیارهای مختلف

شاخص‌های ارزیابی به شرح زیر است:

- **TP^۱**: موارد مثبت که به درستی طبقه‌بندی شده‌اند.
- **TN^۲**: موارد منفی که به درستی طبقه‌بندی شده‌اند.
- **FN^۳**: موارد مثبت که به طور نادرست طبقه‌بندی شده‌اند.
- **FP^۴**: موارد منفی که به طور نادرست طبقه‌بندی شده‌اند.

^۱True Positive

^۲True Negative

^۳False Negative

^۴False Positive

البته باید توجه داشت که این تعاریف برای مسائل با دو دسته تعریف می‌شوند، اما برای مسائل چند دسته (مانند مسئله جاری) هم قابل تعمیم است. در [۸] به دو مورد از این روش‌ها پرداخته شده است. در این گونه مسائل فقط ابعاد ماتریس درهم‌ریختگی^۱ اضافه می‌شود و کلیت مدل تغییر نمی‌کند.

		مقادیر واقعی	
		مثبت	منفی
مقادیر پیش‌بینی شده	مثبت	درست مثبت TP	نادرست مثبت FP خطای نوع یک
	منفی	نادرست منفی FN خطای نوع دو	درست منفی TN

شکل ۴-۱: ماتریس درهم‌ریختگی برای مسئله با دو دسته

دقت: این معیار بیانگر دقت کل طبقه‌بندی است و بیانگر نرخ طبقه‌بندی صحیح است. همان‌طور که در رابطه ۴-۱ مشاهده می‌شود. تمامی مفاهیم فوق در این رابطه اثر گذارند.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (۴-۱)$$

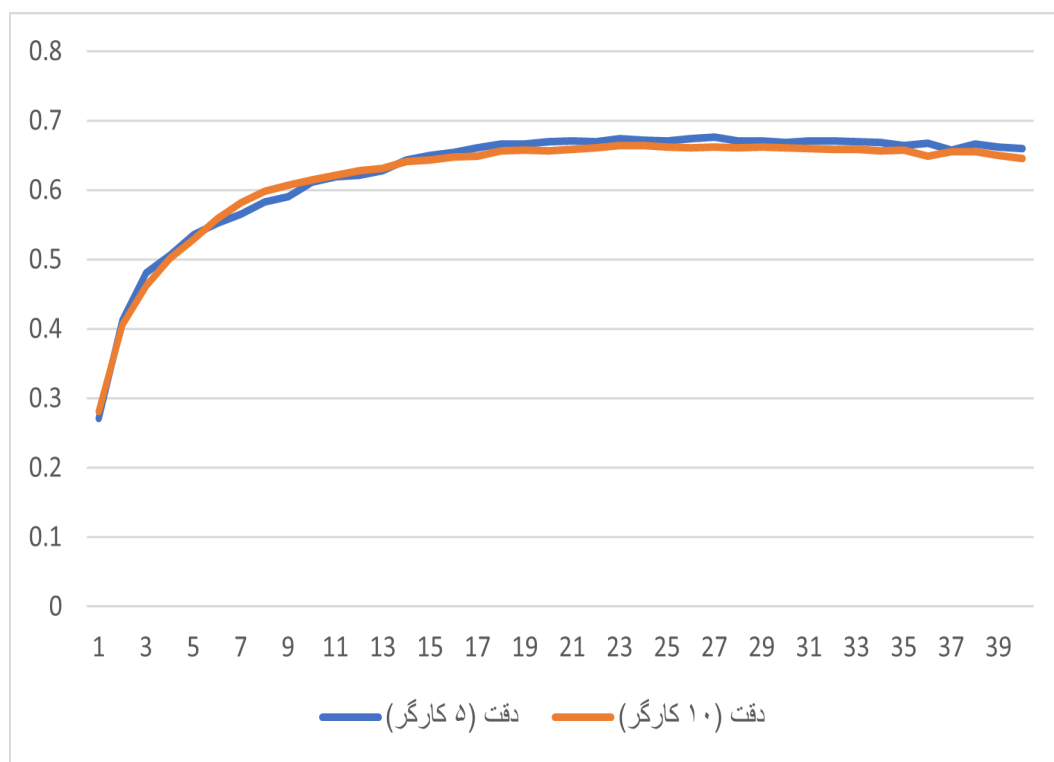
خطا: از این معیار در کنار معیارهای دیگر استفاده می‌شود و بیانگر مقدار فاصله از مقادیر واقعی هر نمونه برای پیش‌بینی توسط مدل است. در رابطه ۴-۲ نماد M بیانگر تعداد کلاس‌ها است و \log لگاریتم طبیعی است. $y_{o,c}$ یک نشانگر دودویی (۰ یا ۱) است اگر برچسب کلاس c طبقه‌بندی صحیح برای مشاهده o باشد (برچسب‌های از قبل دانسته) و $p_{o,c}$ احتمال پیش‌بینی شده‌ای است که مشاهده o از کلاس c باشد (برچسب‌های پیش‌بینی).

$$\text{Loss} = - \sum_{c=1}^M y_{o,c} \log(p_{o,c}) \quad (۴-۲)$$

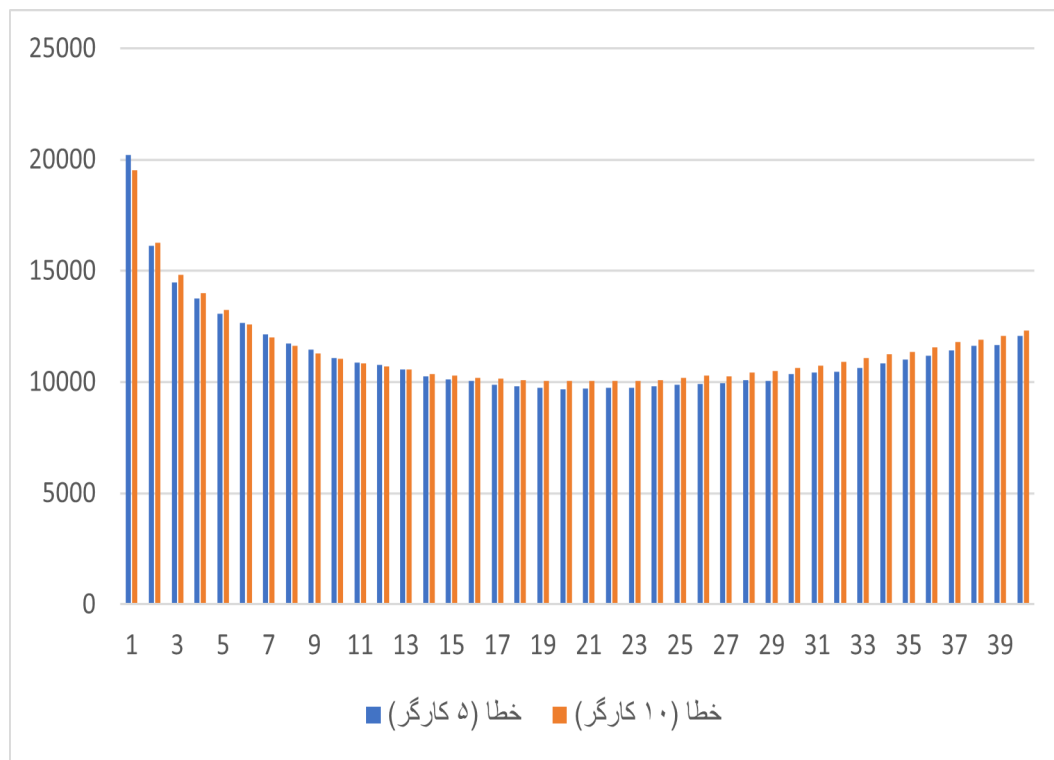
^۱Confusion Matrix

۲-۴ نتایج به دست آمده

در ادامه به تاثیر دو پارامتر تعداد کارگران و تعداد دوره‌های آموزشی بر هر یک از معیارهای دقت و خطای کل مدل شبکه عصبی فرآیند یادگیری فدرال پرداخته شده است. به عنوان نمونه در این شبیه‌سازی یک بار با ۱۰ کارگر و بار دیگر با ۵ کارگر اجرا شده است. لازم به ذکر است که نتایج آورده شده حالت محدودی از یک سناریو واقعی اینترنت اشیا است که میتواند گاهی تا هزاران دستگاه کم توان روی شبکه را نیز در بر بگیرد. در این آزمایش به دلیل کمبود منابع سخت افزاری و زمانی این تعداد را به اندازه یک شبکه داخلی (مثلاً خانه هوشمند) محدود شده است. طبق [۱۰] میانگین تعداد این دستگاه‌ها در کشور ژاپن در هر خانه برابر با ۳.۱۰ عدد است که باتوجه کمتر بودن این عدد در کشور ایران، یکبار با ۷ عدد و بار دیگر با ۱۰ عدد شبیه‌سازی شده است. در ادامه نمودارهای مرتبط با نتایج این شبیه‌سازی در تصاویر ۲-۴ و ۳-۴ برای شهود بیشتر و بهتر آورده شده است.



شکل ۲-۴: نمودار دقت به دست آمده در هر دور از آموزش سراسری



شکل ۴-۳: نمودار خطا در هر دور از آموزش سراسری

همان‌طور که از نمودارها مشخص است، فرآیند آموزش در ابتدا سرعت بالایی دارد و در ادامه به یک خط مماس با شیب نسبتاً کم همگرا می‌شود تا در نهایت به صفر میل کند. با توجه به اینکه این فرآیند دو بار و با تعداد کارگران مختلف انجام شده است، می‌توان دید که تعداد دور آموزشی رابطه مستقیم بیشتری با تعداد کارگران دارد. البته اگر از یک استراتژی دینامیک به جای FedAVG استفاده شود، توقع داریم که به دقت‌های بالاتری برسیم و حتی سرعت همگرایی نیز با شیب بیشتری طی شود.

نمودار دوم گویای این است که طی یک روند غیرخطی در دور ۱۹ام به نقطه کمینه از نظر خطای مدل دسته‌بندی کننده می‌رسیم و در ادامه این روند رو به افزایش است. این می‌تواند بیانگر این باشد که آموزش بیشتر باعث می‌شود شبکه عصبی استفاده شده به عنوان مدل، رو به بیش‌برازش^۱ شدن برود. همان‌طور که می‌دانیم ذات شبکه عصبی با آموزش زیاد به شدت مستعد بیش‌برازش شدن است. در [۱۱] روش‌هایی آورده شده تا با این گونه رفتار مقابله شود.

نکته حائز اهمیت در این شبیه‌سازی این است که زمان تقریبی این شبیه‌سازی با ۱۰ کارگر در ۴۰ دور حدوداً ۴۵ دقیقه می‌باشد که زمان نسبتاً کمی در مقایسه با ماهیت یادگیری فدرال دارد. طبیعتاً در پیاده‌سازی‌های واقعی در شبکه‌های اینترنت اشیاء این زمان نمایی افزایش پیدا می‌کند.

^۱ Overfitting

۴-۳ پیشنهادها برای آینده

همان‌طور که مشاهده شد، با قرار دادن یک لایه اضافه‌تر انتزاع^۱ تحت عنوان لایه پردازش لبه در شبکه‌های اینترنت اشیاء ای که از یادگیری فدرال برای پیش‌بردن اهداف مدیریتی و تصمیم‌گیری خود استفاده می‌کنند، می‌توان کنترل و انعطاف‌پذیری بیشتری به شبکه داد. همان‌طور که دیده شد، اضافه شدن این لایه به دلیل اینکه در دقت و کیفیت مدل استفاده شده داخل فرآیند یادگیری نسبت به عدم وجود آن حداقل پس‌رفتی به وجود نمی‌آورد و امید داریم که بهتر از قبل هم باشد، یک کار ارزشمند است. همچنین با اضافه شدن سرور لبه به سیستم یادگیری فدرال، دست برنامه‌نویسان نرم‌افزار نیز برای توسعه برنامه‌های کاربردی روی این شبکه‌ها به مراتب بازتر است.

اگرچه چندین راه حل تحقیقاتی برای کاهش چالش‌های اجرای یک سناریو خاص یادگیری فدرال در شبکه شامل دستگاه‌های پردازش لبه پیشنهاد شده است، اما هنوز چالش‌هایی وجود دارد که حل نشده‌اند. علاوه بر ابتکارات تحقیقاتی گزارش شده در قسمت‌های قبلی، ایده‌های نوین قابل بحثی وجود دارند که در زیر توضیح داده شده است:

- فرآیند آموزش یادگیری فدرال با پشتیبانی چند مدل: در فرآیند آموزش یادگیری فدرال، فرض می‌شود که کارگران شرکت‌کننده پارامترهای مدل متناظر خود را با توجه به یک مدل سراسری به‌روز می‌کنند. با این حال، ممکن است کارگران بخواهند چندین مدل را حتی در زمان بطلت خود آموزش دهند؛ بنابراین، جداسازی جمع‌آوری مدل سراسری از آموزش محلی به کارگران اجازه می‌دهد تا از الگوریتم‌های یادگیری مختلف استفاده کنند. به عنوان مثال، ممکن است لازم باشد هم‌زمان چندین مدل مختلف با استفاده از روش فدرال برای اهداف مختلف توسعه یابند؛ بنابراین، تکنولوژی‌های مناسب بایستی تجزیه و تحلیل و پیاده‌سازی شود.

- تأثیر کانال شبکه بی‌سیم: دستگاه‌های لبه اغلب از طریق کانال‌های بی‌سیم به سرویس‌دهنده‌های لبه یا ابری وصل شده‌اند؛ بنابراین، بررسی تأثیر الزامات شبکه، به ویژه ارتباطات بی‌سیم، در دقت آموزش مدل فدرال به عنوان یک روند تحقیقاتی آینده در نظر گرفته شده است. نویز، خسارت مسیر، ترافیک بالا و قطعی همه نقص‌هایی هستند که بایستی در سیستم‌های ارتباطات بی‌سیم در نظر گرفته شود. باید توجه داشت که در سناریوهای اینترنت اشیاء نیز قطع و وصل شدن دستگاه‌های انتهایی خود دلیلی محکم بر این تأثیرات مخرب می‌باشد.

- انتخاب پویای کارگران و جمع‌آوری تطبیقی مدل: جمع‌آوری مدل تطبیقی و انتخاب مشتری پویا برای تخصیص منابع با در نظر گرفتن رفتار ناهمگون داده، قدرت محاسباتی، اندازه داده، ظرفیت شبکه و قابلیت اطمینان لینک در بین کارهای تحقیقاتی آینده قرار دارد. بنابراین، بایستی به عنوان روندهای تحقیقاتی آینده،

¹ Abstraction

انتخاب پویای کارگران و جمع‌آوری تطبیقی مدل را بررسی کرد. به عنوان مثال، الگوریتم‌ها و پیچیدگی‌های مناسب برای درخواست‌های پویا کارگر و منابع ناهمگون در هر دو سمت کارگر و سرور اصلی دهنده. در چنین سناریوهایی، باید بتوان فرآیند آموزش یادگیری فدرال را مدیریت کرد. یکی از ایده‌های تعیین ضرایب و پارامترها برای انتخاب پویا کارگران و جمع‌آوری تطبیقی مدل، تخمین‌زدن تقریبی و تنظیم آن در سرور لبه می‌باشد. این محل که در مرکز شبکه اینترنت اشیاء قرار دارد، تخمین خوبی از رفتار کلی شبکه در اختیار الگوریتم‌های تنظیم‌کننده پارامترها قرار می‌دهد [۱۲].

- روش جدید یادگیری فدرال: اندازه مدل یادگیری فدرال بسیار بزرگ است تا بتواند در یک دستگاه لبه با منابع محدود جا شود. علاوه بر این، آموزش مدل یادگیری فدرال بسیار کند است و نمی‌تواند به همگرایی برسد و نیازهای تأخیر در برخی از برنامه‌های حساس به تأخیر را برآورده کند. روش جدید یادگیری فدرال مورد نیاز است تا به طور پویا اهداف را دستیابی کند؛ بنابراین، یادگیری فدرال پویا و تطبیقی بایستی برای دستگاه‌های لبه با منابع محدود تجزیه و تحلیل و پیاده سازی شود. به عنوان مثال میتوان با وزن دادن به گره‌ها در الگوریتم FedAVG و میانگین وزن دار گرفتن از آن‌ها در حین اجرای سیستم، به دقت و سرعت بالاتری دست پیدا کرد.
- پیاده‌سازی پروتکل‌های پیچیده به منظور بهبود کیفیت ارتباطات در فرآیند یادگیری: می‌توان از پروتکل‌های مختلف که بعضی سنگین و بعضی دیگر سبک هستند به منظور ایجاد یک پروسه یادگیری امن و مطلوب در بستر شبکه استفاده کرد. پیاده‌سازی این پروتکل‌ها می‌تواند معطوف به دستگاه‌های انتهایی نباشد و سرور لبه نیز در آن نقش مشخصی داشته باشد. یکی از این پروتکل‌ها که بدون در نظر گرفتن سرور لبه توسعه پیدا کرده پروتکل SAFA^۱ است که ساختاری وابسته به حالت^۲ ارائه می‌کند که باعث افزایش سرعت همگرایی می‌شود. در واقع با دسته‌بندی کارگران و نگه داشتن وضعیت اخیر آنها در پروسه یادگیری با دادن برچسب‌های مخصوصی به آنها و دسته‌بندی کردن آنان به هرکدام وظایف معینی می‌دهد. می‌توان با محدود کردن این پروتکل و اضافه کردن نقش سرور لبه به آن، به کار آن سرعت و دقت بیشتری بخشید [۱۳].

- امنیت بالاتر با استفاده از سرور لبه: از دانش قبلی می‌دانیم که یکی از بنیادی‌ترین دلایل به وجود آمدن یادگیری فدرال امنیت بالای آن در مقایسه با دیگر روش‌های یادگیری توزیع شده است. اما روش‌های جدیدی آمده که با داشتن دیتای مربوط به کارگران و سرور تجمیع‌کننده و شنود کردن این داده‌ها روی شبکه به روش‌های مرسوم مانند جعل آدرس^۳ و بررسی ترافیک شبکه^۴، می‌توان طی چند دور به مدل داخل سرور اصلی تا حد خوبی دست پیدا کرد [۱۴]. با اضافه کردن یک سرور لبه برای افزایش ایمنی، می‌توان از این راهکار جلوگیری کرد و

¹Semi-Asynchronous Protocol for Fast Federated Learning with Low Overhead

²Stateful

³Spoofing

⁴Sniffing

این روند را سخت‌تر و طولانی‌تر کرد.

پیوست اول

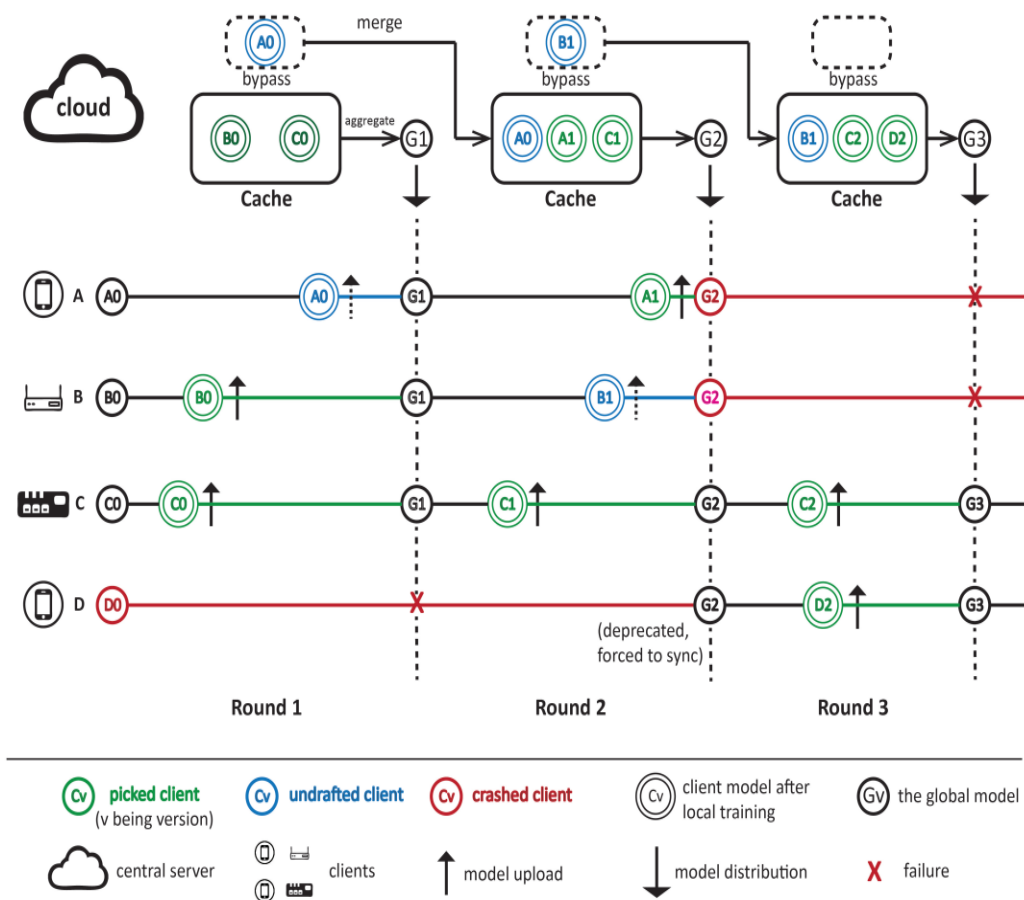
پروتکل SAFA

یکی از پروتکل‌هایی که قابلیت پیاده‌سازی برای فرآیند یادگیری فدرال دارد، پروتکل SAFA است. همان‌طور که در شکل آ-۱ مشهود است، این پروتکل با دسته‌بندی کارگران به ۳ دسته انتخاب شده^۱، تعیین نشده^۲ و قطع شده^۳ تقسیم می‌کند. این دسته بندی با توجه به نحوه برقراری ارتباط با کارگر در دوره‌های قبلی هست که در کش ذخیره می‌شود. پروتکل SAFA با استفاده از یک الگوریتم تصادفی، تعدادی از کارگران را به عنوان انتخاب شده انتخاب می‌کند و به آن‌ها دستور می‌دهد که مدل خود را به سرور ارسال کنند. سپس، سرور با دریافت مدل‌های ارسال شده، آن‌ها را با هم ترکیب می‌کند و مدل جدید را به کارگران انتخاب شده برمی‌گرداند. در صورتی که یک کارگر در یک دوره از ارسال یا دریافت مدل عقب بماند، به عنوان قطع شده در نظر گرفته می‌شود و در دوره‌های بعدی حذف می‌شود. همچنین، در صورتی که یک کارگر به طور مکرر در دوره‌های مختلف انتخاب نشود، به عنوان تعیین نشده شناخته می‌شود و احتمال انتخاب آن در دوره‌های آینده کاهش می‌یابد. پروتکل SAFA با استفاده از این روش، سعی دارد تا تعادل بین سرعت و دقت فرآیند یادگیری فدرال را حفظ کند [۱۳].

¹Picked Client

²Undrafted Client

³Crashed Client



شکل آ-۱: سناریوی اجرای پروتکل SAFA در دور ۳ در دور [۱۳]

با توجه به شکل، کش مدل‌های محلی آخرین کارگرهای انتخاب شده را که برای تجمیع استفاده می‌شوند، نگه می‌دارد. کارگرهایی که نتایج آن‌ها انتخاب نشده‌اند، کارگرهای رد شده هستند (با رنگ آبی)، مثلاً کارگر A در دور ۱ و کارگر B در دور ۲. به‌روزرسانی‌های این کارگرها در ساختار میان‌بر^۱ ذخیره می‌شوند تا از کار بیهوده در سمت محلی جلوگیری شود. کارگرهایی که به هر دلیلی (مثل خروج از برنامه یا قطع شدن اینترنت) نمی‌توانند آموزش محلی خود را تکمیل کنند، کارگرهای خراب هستند (با رنگ قرمز برجسته شده)، مثلاً کارگر D در دور ۱. کارگرهایی که به‌روزرسانی‌های آن‌ها انتخاب شده‌اند، کارگرهای انتخاب شده نامیده می‌شوند (با رنگ سبز)، مثلاً کارگر B و C در دور ۱.

توزیع مدل با تحمل تأخیر^۲: در این مرحله، سرور مدل خود را به کارگران مختلف ارسال می‌کند و از آن‌ها می‌خواهد که بر روی داده‌های خود به روزرسانی مدل را انجام دهند. سرور همچنین یک حداکثر زمان تأخیر را برای دریافت مدل‌های به روز شده تعیین می‌کند. این روش اجازه می‌دهد که کارگران با سرعت‌های مختلف و

^۱Bypass

^۲Model Distribution With Lag Tolerance

با تأخیرهای متفاوت در فرآیند یادگیری شرکت کنند.

اولین رسیدن برابر است با اولین ادغام^۱: در این مرحله، سرور با دریافت مدل‌های به روز شده از کارگران، آن‌ها را با هم ترکیب می‌کند. سرور برای ترکیب کردن مدل‌ها، از الگوریتم «اولین رسید، اولین ترکیب شد» استفاده می‌کند. این الگوریتم به این صورت است که سرور همواره یک نسخه از مدل را نگه دارد و هر گاه یک مدل جدید از یک کارگر دریافت کند، آن را با نسخه فعلی ترکیب کرده و نسخه جدید را ذخیره می‌کند.

تجميع تمایزدهنده^۲: در این مرحله، سرور با استفاده از یک الگوریتم تجميع تمایزدهنده، مقادیر وزن‌های مختلف در مدل را با هم مقایسه کرده و آن‌ها را به صورت چند جمله‌ای درجه دو نمایش می‌دهد. سپس، با استفاده از چند جمله‌ای حاصل، سطح خطای هر وزن را برآورد کرده و وزن‌های با خطای بالاتر را حذف یا کم کاربرد می‌کند. این روش باعث بهبود دقت و کارآمدی فرآیند یادگیری فدرال می‌شود.

باتوجه به کاربردهای متنوع این پروتکل و قابلیت انعطاف بالا خصوصاً زمانی که دستگاه‌های کارگر از سخت‌افزارهای ناهمگون بهره ببرند می‌تواند به شدت فرآیند یادگیری را تسریع کند.

^۱First Come First Merge

^۲Discriminative Aggregation

مراجع

- [1] E. Baccour et al., "Pervasive AI for IoT Applications: A Survey on Resource-Efficient Distributed Artificial Intelligence," in *IEEE Communications Surveys and Tutorials*, vol. 24, no. 4, pp. 2366-2418, Fourthquarter 2022, doi: 10.1109/COMST.2022.3200740.
- [2] A. A. Osuwa, E. B. Ekhonoragbon and L. T. Fat, "Application of artificial intelligence in Internet of Things," in *2017 9th International Conference on Computational Intelligence and Communication Networks (CICN)*, Girne, Northern Cyprus, 2017, pp. 169-173, doi: 10.1109/CICN.2017.8319379.
- [3] Dinh C. Nguyen, Ming Ding, Pubudu N. Pathirana, Aruna Seneviratne, "Federated Learning for Internet of Things: A Comprehensive Survey", *arXiv:2104.07914v1*, 2021.
- [4] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., and Cummings, R., "Advances and open problems in federated learning", *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1-210, 2021.
- [5] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Aguerre y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data" *AISTATS*, Google, Inc., 2017.
- [6] Beutel, Daniel J and Topal, Taner and Mathur, Akhil and Qiu, Xinchu and Fernandez-Marques, Javier and Gao, Yan and Sani, Lorenzo and Kwing, Hei Li and Parcollet, Titouan and Gus-

mão, Pedro PB de and Lane, Nicholas D, “Flower: A Friendly Federated Learning Research Framework” *arXiv preprint arXiv:2007.14390* , 2020.

- [7] Wang, Xingwei and Zhao, Hong and Zhu, Jiakeng, “GRPC: a communication cooperation mehanism in distributed systems”, *Association for Computing Machinery*, 1993.
- [8] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, “An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes,” in *Pattern Recognition*, vol. 44, no. 8, pp. 1761-1776, August 2011, Elsevier.
- [9] Liu, Yang and Fan, Tao and Chen, Tianjian and Xu, Qian and Yang, Qiang, “FATE: An Industrial Grade Platform for Collaborative Learning with Data Protection ”, *JMLR.org*, 2021.
- [10] J. Koetsier, “Smart Home: Apple Is The Fastest-Growing Connected Device Company,” *Forbes*, Aug 31, 2022
- [11] J. Brownlee, “Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions,” *Machine Learning Mastery*, December 13, 2018.
- [12] Portnoy, A., Tirosh, Y., Hendler, D., “Towards Federated Learning With Byzantine-Robust Client Weighting”, *arXiv:2004.04986v2*, 2021
- [13] Wentai Wu, Ligang He, Weiwei Lin, Rui Mao, Carsten Maple, Stephen Jarvis, “SAFA: a Semi-Asynchronous Protocol for Fast Federated Learning with Low Overhead”, *arXiv:1910.01355v4*, v. 4, 2021.
- [14] Geiping, J., Bauermeister, H., Dröge, H., Moeller, M., “Inverting Gradients – How easy is it to break privacy in federated learning?”, *arXiv:2003.14053v2*, 2020
- [15] Mahmod, Ahmad and Caliciuri, Giuseppe and Pace, Pasquale and Iera, Antonio, “Improving the Quality of Federated Learning Processes via Software Defined Networking”, *Association for Computing Machinery*, 2023.
- [16] Abreha, Haftay Gebreslasie, Mohammad Hayajneh, and Mohamed Adel Serhani, ”Federated Learning in Edge Computing: A Systematic Survey”, *Wireless Sensor Networks towards the Internet of Things*, Sensors 22, no. 2: 450 , 2022
- [17] Dhurgham Hassan Mahloul, Mohammed Hamzah Abed, ”A Comprehensive Survey on Federated Learning: Concept and Applications”, *Computer Vision and Pattern Recognition*, *arXiv:2201.09384* , 2022

- [18] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, Salman Avestimehr, “Federated Learning for Internet of Things: Applications, Challenges, and Opportunities”, *arXiv:2111.07494v4*, v. 4, 2022.
- [19] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, Junshan Zhang, “Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing”, *arXiv:1905.10083v1*, v. 1, 2019.
- [20] Andy Chen and Chaitanya Asawa, “Going beyond the bounding box with semantic segmentation”, *The Gradient*, 2018.
- [21] Zewen Li, Wenjie Yang, Shouheng Peng, Fan Liu, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects”, *arXiv:2004.02806v1*, v. 1, 2020.

- [۲۲] طلائی مهتاب ” توسعه‌ی الگوریتم و بررسی عملکرد یادگیری پراکنده با حریم شخصی تفاضلی تطبیقی“، *استقلال*، دانشگاه صنعتی اصفهان، شماره ۱۷۷۵۸، صص ۲۷-۳۴، ۱۴۰۱.
- [۲۳] بزرگ‌زاد علی، “بررسی روش‌های حل مشکل ناهمگونی آماری در یادگیری فدرال”، *سمینار کارشناسی ارشد هوش مصنوعی و رباتیک*، دانشگاه صنعتی اصفهان، ۱۴۰۱.

Edge Computing Empowers Federated Learning in Resource-Constrained IoT Networks

Amirreza Hosseini

amirreza.hosseini@ec.iut.ac.ir

September 2023

Department of Electrical and Computer Engineering
Isfahan University of Technology, Isfahan 84156-83111, Iran

Degree: B.Sc.

Language: Farsi

Supervisor: Dr. Amir Khorsandi (khorsandi@iut.ac.ir)

Abstract

Artificial Intelligence (AI) has rapidly expanded into various fields in recent years, becoming a fundamental technology in numerous applications. To meet the diverse needs of these applications, various approaches and methods have been proposed to achieve the infrastructure related to AI technology. One of these methods is the Machine Learning (ML) approach for privacy preservation, known as Federated Learning (FL). FL enables the development of applications that perform analysis and AI-based services on sensitive user data while complying with strict privacy regulations in various domains, including healthcare, finance, and transportation, particularly the Internet of Things (IoT). It is worth noting that FL can be executed on a large number of heterogeneous final devices, which are typically distributed and have limited processing power and energy resources. This presents a significant challenge that can hinder the effective, sustainable, and scalable execution of FL.

To address these challenges, recent research has proposed the use of edge computing in FL. Edge computing involves using the storage, communication, and computing capabilities of edge servers located near end devices to reduce system latency. Performing primary computations and preprocessing close to the data source can reduce the amount of traffic sent to the main server and decrease access delays. This approach can also be effective in improving the performance of FL, which is the main focus of this project.

Therefore, the objective of this project is to propose a structure that provides a suitable platform for developing FL in networks with edge computing capabilities. The proposed structure aims to address the challenges of executing FL on heterogeneous and resource-constrained devices by leveraging the capabilities of edge servers. The resulting system improves the performance and scalability of FL while complying with strict privacy regulations.

Key Words: Federated learning, Internet of Things, Edge computing, Machine Learning, Artificial Intelligence

