مقاله پژوهشی

مجله ماشین‌بینایی و پردازش‌تصویر
سال نهم، شماره اول، بهار ۱۴۰۱

ترکیب روش منظم‌سازی تُنُک و آسیب مغزی بهینه در کوچک‌سازی یک مدل یادگیری عمیق

محمود امین‌طوسی

چکیده

یکی از چالش‌های شبکه‌های عصبی پیچشی، به عنوان ابزار اصلی یادگیری عمیق، حجم زیاد برخی از مدل‌های مربوطه است. یک شبکه‌ی عصبی پیچشی به مثابه مدلی از مغز، متشکل از میلیون‌ها اتصال است. کاهش حجم این مدل‌ها از طریق حذف (هرس) اتصالات اضافی مدل انجام می‌شود که همانند یک آسیب مغزی است. دو روش منظم‌سازی تُنُک و آسیب مغزی بهینه از جمله مشهورترین شیوه‌های هرس مدل هستند. در این نوشتار با ترکیب این دو شیوه نتایج بهتری در کاهش حجم مدل حاصل شده است. ابتدا با استفاده از روش انتقال یادگیری، یک مدل بزرگ شبکه‌های عصبی پیچشی برای شناسایی طبقات هدف، آموزش داده شد؛ سپس با روش‌های منظم‌سازی تُنُک و آسیب مغزی بهینه، اتصالات اضافی آن هرس شدند. نتایج آزمایشات نشان داده است که در بیشتر مجموعه دادگان مورد بررسی، اعمال شیوه‌ی ترکیبی منظم‌سازی تُنُک و آسیب مغزی بهینه نسبت به اعمال هر یک از آنها به صورت جداگانه کاراتر است. برای یکی از مجموعه دادگان مورد بررسی، با روش ترکیبی پیشنهادی تعداد اتصالات مدل ۷۶ درصد کاهش داده شد، بدون آنکه کارایی آن کاهش یابد. این کاهش حجم مدل، زمان پردازشی را به یک سوم تقلیل داده است. کاهش حجم مدل می‌تواند امکان استفاده از آن در مرورگرها و سخت‌افزارهای ضعیف‌تر و همه‌گیرتر را تسهیل سازد.
(کد برنامه: https://github.com/mamintoosi/Reg-OBD-for-VGG-Pruning)

کلیدواژه‌ها

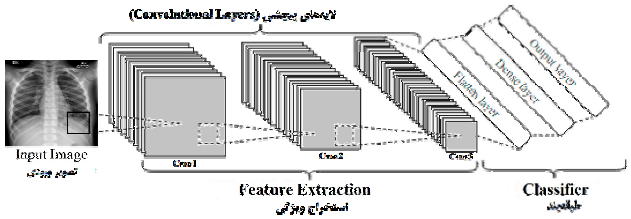
شبکه‌های عصبی پیچشی، هرس شبکه، یادگیری عمیق، بهینه‌سازی تُنُک، منظم‌سازی تُنُک

۱ مقدمه

در سالیان اخیر کاربردهای یادگیری عمیق در حوزه‌های مختلف و مخصوصاً پردازش تصویر رو به گسترش بوده است [۱-۴]. شبکه‌های عصبی پیچشی به عنوان ابزار اصلی یادگیری عمیق در حوزه‌ی پردازش تصویر، متضمن چندین لایه از انواع مختلفند که هر دسته از آنها عملکرد مشخصی دارند (شکل ۱). برخی از شبکه‌های پیچشی مشتمل بر میلیون‌ها اتصال می‌باشند که کاهش تعداد نورون‌ها و اتصالات آنها، با حفظ کارایی شبکه، می‌تواند در حجم حافظه و زمان پردازشی موردنیاز مؤثر باشد. حذف اتصالات و نورون‌های یک شبکه‌ی عصبی مصنوعی تداعی‌گر «آسیب مغزی» در بیماران مبتلا به این ضایعه است. آسیب مغزی یکی از مهم‌ترین علل شایع ناتوانی به ویژه مشکلات شناختی در جهان محسوب می‌شود که تحقیقات بسیاری را به خود معطوف نموده است [۵].

این مقاله در اردیبهشت ۱۴۰۰ دریافت گردید؛ در تیرماه همان سال بازنگری و سپس پذیرفته شد.
محمود امین‌طوسی، گروه علوم کامپیوتر، دانشکده ریاضی و علوم کامپیوتر، دانشگاه حکیم سبزواری
رایانامه: m.amintoosi@hsu.ac.ir

DOR: 20.1001.1.23831197.1401.9.1.3.4



شکل ۱- شمای کلی یک شبکه‌ی عصبی پیچشی. تصویر مورد بررسی به شبکه داده شده و در لایه‌ی آخر، طبقه‌بندی تصویر ورودی انجام می‌شود. در این شبکه‌ی فرضی، ۳ لایه‌ی پیچشی مربوط به استخراج ویژگی هستند. بین هر دو لایه هزاران اتصال وجود دارد که نمایش داده نشده است (برگرفته از [۶] با ویرایش مختصر).

آسیب تروماتیک مغزی، به علت کاهش اتصالات عصبی، با عوارض مختلف فیزیکی، عاطفی و هیجانی، یادگیری و شناختی همراه است[۷, ۸]. شواهدی وجود دارد که مغز پس از یک آسیب مغزی، سازماندهی مجدد[۱] می‌شود [9] و بهبود عملکرد مبتلایان با مداخله گزارش شده است [۵]. بر همین اساس شیوه‌هایی برای کاهش حجم شبکه‌های عصبی پیچشی ارائه شده است [۱۰, ۱۱]. از دیدگاه عصب‌شناسی شناختی، معماری یک سیستم شناختی آسیب‌دیده، همانند یک سیستم سالم است که یک یا چند جزء آن آسیب دیده یا حذف شده‌اند [12]. نتایج تحقیقات، مؤید انعطاف شبکه عصبی مغز و سازماندهی مجدد پویای شبکه عصبی پس از یک آسیب مغزی است[9]. فرآیند سازماندهی مجدد عملکردهای پایه[۲] [۱۳] پس از یک آسیب مغزی، مشابه الگوریتم پس‌انتشار خطا در شبکه‌های عصبی مصنوعی است [14]. بر همین اساس، شبکه‌ی عصبی پیچشی که بخش‌هایی از آن حذف شده‌اند، مورد آموزش مجدد قرار می‌گیرد تا عملکرد قبلی خود را بازیابد. مسئله‌ی اصلی که موجب شده روش مورد استفاده، «آسیب مغزی» نامیده شود[۱۰]، هرس شبکه است که به مثابه آسیب شبکه‌ی عصبی مغز می‌باشد؛ و چون اتصالاتی باید انتخاب و حذف شوند که ساختار باقیمانده، کارایی بهینه‌ای داشته باشد، به آن «آسیب مغزی بهینه» (OBD[۳]) اطلاق می‌شود. از دیگر روش‌های مورد استفاده در کاهش حجم مدل‌های یادگیری عمیق، شیوه‌های مبتنی بر منظم‌سازی[۴] [۱۵] هستند. در ادامه این شیوه‌ها بیان خواهند شد. هدف اصلی در این نوشتار بررسی امکان ترکیب این دو شیوه برای کاهش یک مدل یادگیری عمیق، در عین حفظ کارایی می‌باشد.

برای بررسی شیوه‌ی ترکیبی پیشنهادی، به جز مجموعه دادگان عمومی، از مجموعه‌ای از تصاویر پرتونگاری رایانه‌ای (رادیولوژی) افراد در شناسایی کووید۱۹ استفاده شده است. شیوه‌ی کار و نتایج آزمایشات ابتدا بر روی مجموعه دادگان کووید۱۹ و سپس بر روی مجموعه دادگان عمومی بیان خواهد شد. چارچوب کلی کار به این ترتیب است که ابتدا با روش انتقال یادگیری یک مدل از قبل آموزش دیده (بر روی تصاویر طبیعی) برای شناسایی مورد آموزش مجدد قرار گرفته و سپس تاثیر هر یک از دو روش هرس شبکه و ترکیب آنها در کارایی و کاهش حجم شبکه مورد بررسی قرار می‌گیرد. هدف، کاهش تعداد اتصالات این مدل، با ترکیب روش‌های فوق‌الذکر، به نحوی است که مدل حاصله کارایی اولیه خود را داشته باشد. در این نوشتار به دنبال بهترین مدل یا بیشترین میزان فشردگی یک مدل نیستیم، منظور اصلی آن است که نشان داده شود با روش ترکیبی پیشنهادی یک مدل حجیم مانند مدل VGG16[۵] در عین حفظ کارایی می‌تواند بسیار کوچک‌تر شود.

در ادامه ابتدا مرور مختصری بر شبکه‌های عصبی پیچشی خواهیم داشت؛ سپس موضوع شناسایی کووید۱۹ با این شبکه‌ها مرور خواهدشد. پس از آن شیوه‌های کاهش حجم مدل‌های یادگیری عمیق را خواهیم دید. روش ترکیبی پیشنهادی، نتایج آزمایشات و جمع‌بندی، دیگر بخش‌های ادامه‌ی این نوشتار را تشکیل می‌دهند.

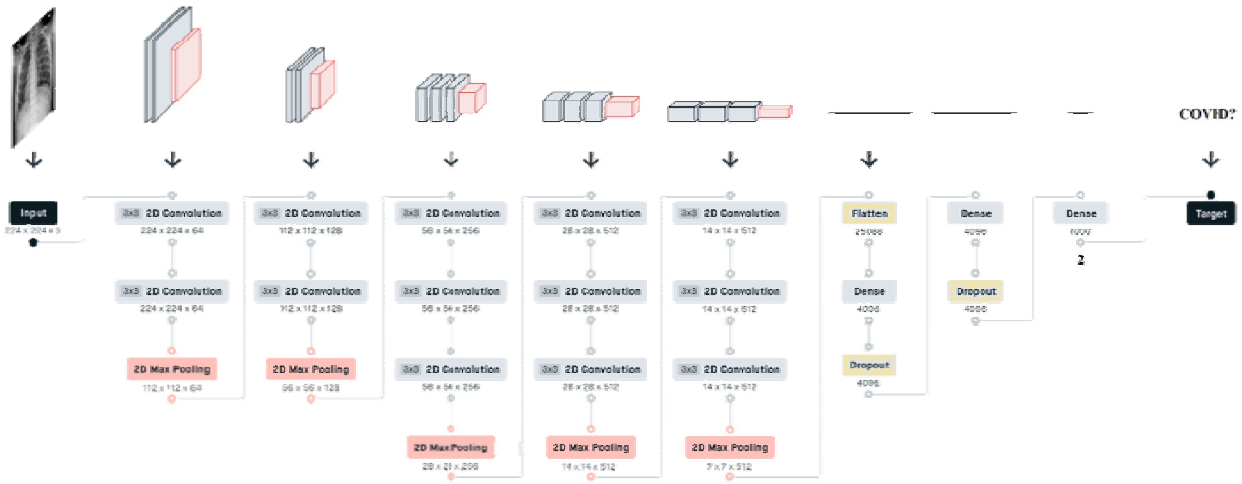
۱-۱ شبکه‌های عصبی پیچشی

افزایش قدرت محاسباتی سخت‌افزارها و ابداعات نرم‌افزاری سالیان اخیر در حوزه‌ی شبکه‌های عصبی موجب توجه مجدد به این شبکه‌ها و مخصوصاً شبکه‌های عصبی پیچشی شده است. برخلاف شبکه‌های سنتی چند لایه‌ی پرسپترونی که هر نود در یک لایه به تمام نودها (نورون‌های) لایه‌های قبل و بعد از خود متصل است، در شبکه‌های پیچشی هر نورون در یک لایه از یک گروه محلی از نورون‌های لایه‌ی قبل از خود تاثیر می‌پذیرد. چارچوب کلی یک شبکه‌ی عصبی پیچشی در شکل (۱) نشان داده شده است. لایه‌های پیچشی وظیفه‌ی استخراج ویژگی‌ها[۶] را به عهده دارند و لایه‌های تمام متصل[۷] کار طبقه‌بندی ویژگی‌های استخراج شده را انجام می‌دهند. ویژگی‌های استخراج شده، به صورت یک بردار درآمده (مسطح‌سازی[۸]) و به طبقه‌بند داده می‌شود. در یک مسئله‌ی طبقه‌بندی تصاویر، پس از آموزش شبکه، تصویر جدید به ورودی شبکه داده می‌شود، لایه‌ی خروجی، مشخص خواهد کرد که تصویر به کدام دسته از طبقات آموزش داده شده تعلق دارد.

در شکل ۱ لایه‌هایی که استخراج ویژگی را انجام می‌دهند با رنگ خاکستری متمایز شده‌اند. همانند پیچش در پردازش سیگنال، در این شبکه‌ها هم برای هر عمل پیچش یک فیلتر (کرنل[۹]) مورد نیاز است. مربع‌های کوچک خط‌چین در لایه‌های Conv1, Conv2 بیانگر این فیلترها هستند. بین این لایه‌ها هزاران اتصال وجود دارد. یادگیری در این شبکه‌ها به معنی تنظیم درست وزن این اتصالات است که با روش‌های بهینه‌سازی انجام می‌شود.

[1] Reorganize
[2] Reorganization of Elementary Functions
[3] Optimal Brain Damage (OBD)
[4] Regularization

[5] Visual Geometry Group: https://www.robots.ox.ac.uk/~vgg/software/
[6] Feature Extraction
[7] Fully Connected or Dense
[8] Flatten
[9] Filter or Kernel



شکل ۲- معماری مدل VGG16 برای طبقه‌بندی دو کلاسه. این مدل دارای ۱۳ لایه‌ی پیچشی (Convolution) و ۳ لایه‌ی تمام متصل (Dense) است که با رنگ خاکستری نشان داده شده‌اند (برگرفته از سایت www.topbots.com با ویرایش مختصر).

در بخش طبقه‌بند نیز بین لایه‌ی تمام متصل و مجاورین آن هزاران اتصال وجود دارد. مدل‌های مختلف شبکه‌های پیچشی تعداد لایه‌ها و تعداد اتصالات متفاوتی دارند. مدل VGG16 که در شکل ۲ جزییات بیشتری از آن نمایش داده شده است، بیش از ۱۳۰ میلیون پارامتر آموزش‌پذیر (اتصال) دارد که به دنبال کاهش این تعداد زیاد پارامتر هستیم. به جز مدل VGG مدل‌های مشهور دیگری هم موجود هستند؛ تعداد لایه‌ها، نوع لایه‌ها، عمق هر لایه و نحوه‌ی اتصال لایه‌های مختلف به یکدیگر از جمله مواردی هستند که باعث تفاوت ساختار (معماری) شبکه‌ها می‌شوند. مدل VGG در دو فرم اصلی ۱۶ و ۱۹ لایه ارائه شده است[۱۶]. مدل‌های ResNet [۱۷]، AlexNet [18] و DarkNet [۱۹] از جمله دیگر مدل‌های مشهور این حوزه هستند. از نظر تعداد اتصالات، مدل VGG از حجیم‌ترین مدل‌ها محسوب می‌شود.

در این نوشتار برای نمایش نحوه و اثر کاهش تعداد اتصالات از مدل VGG16 استفاده خواهد شد که در شکل ۲ نمایش داده شده است. این مدل، شامل ۱۳ لایه‌ی پیچشی (Conv) و ۳ لایه‌ی کاملاً متصل (Dense) است. اولین بلاک از لایه‌های این شبکه (ستون اول بعد از تصویر ورودی در شکل ۲) شامل دو لایه‌ی پیچشی و یک لایه‌ی ادغام بیشینه[1] است. لایه‌های ادغام بیشینه اندازه ورودی را تغییر می‌دهند که ویژگی‌ها در مقیاس‌های مختلف دیده شوند. هر لایه‌ی پیچشی در سطر اول با مکعب مستطیل خاکستری و لایه‌های ادغام بیشینه با رنگ قرمز نمایانده شده‌اند. در زیر مکعب مستطیل‌ها، ویژگی‌های هر لایه در یک کادر ذکر شده است. به عنوان نمونه، اولین لایه‌ی پیچشی شامل ۶۴ فیلتر 3×3 است که نتیجه‌ی آنها 224×224 است. این ۶۴ مستطیل با اندازه‌ی 224×224 مترادف با یک لایه از لایه‌های خاکستری رنگ در شکل ۱ هستند.

جدول ۱- پارامترهای مدل نمایش داده شده در شکل ۲.

Layer (type)	Output Shape	Param #
2D Conv-1	[64, 224, 224]	1,792
2D Conv-2	[64, 224, 224]	36,928
2D Conv-3	[128, 112, 112]	73,856
2D Conv-4	[128, 112, 112]	147,584
2D Conv-5	[256, 56, 56]	295,168
2D Conv-6	[256, 56, 56]	590,080
2D Conv-7	[256, 56, 56]	590,080
2D Conv-8	[512, 28, 28]	1,180,160
2D Conv-9	[512, 28, 28]	2,359,808
2D Conv-10	[512, 28, 28]	2,359,808
2D Conv-11	[512, 14, 14]	2,359,808
2D Conv-12	[512, 14, 14]	2,359,808
2D Conv-13	[512, 14, 14]	2,359,808
Dense-1	[4096]	102,764,544
Dense-2	[4096]	16,781,312
Dense-3	[2]	8,194
	Total params:	134,268,738

تعداد پارامترهای مدل VGG16 در جدول ۱ آمده است. لایه‌ها مطابق شکل ۲ شماره‌گذاری شده‌اند. از آنجا که لایه‌های ادغام بیشینه دارای وزن نیستند در این جدول ذکر نشده‌اند. سیزده لایه‌ی پیچشی دوبعدی (2D Conv) و ۳ لایه‌ی تمام متصل (Dense) در مجموع دارای ۱۳۴,۲۶۸,۷۳۸ وزن اتصال هستند.

[1] Max Pooling

اولین لایه‌ی تمام متصل (Dense-1) بیشترین تعداد پارامتر را دارد. ورودی به این لایه، خروجی آخرین لایه‌ی ادغام بیشینه است (آخرین کادر قرمز در شکل ۲) که متشکل از ۵۱۲ تصویر ۷×۷ است. به این معنی که ورودی طبقه‌بند۲۵۰۸۸ = ۷ × ۷ × ۵۱۲ مؤلفه دارد. این تعداد ۰۸۸.۲۵، در تعداد نورون‌های اولین لایه‌ی تمام متصل (۴۰۹۶) که ضرب شود، تعداد اتصالات بین این دو لایه — که ۱۰۲,۷۶۴,۵۴۴ است — حاصل می‌شود. اگر تعداد فیلترهای لایه‌ی پیچشی قبل کمتر شود، تعداد اتصالات لایه‌ی طبقه‌بند نیز کمتر خواهد شد؛ که بعداً به این موضوع پرداخته خواهد شد.

مدل‌هایی مانند VGG روی میلیون‌ها تصویر طبیعی (عموماً تصاویر اشیاء و حیوانات) از حدود هزار دسته و با صرف صدها ساعت زمان پردازشی آموزش دیده‌اند. تعداد دسته‌هایی که مدل VGG16 روی آن آموزش دیده، هزار دسته می‌باشد که در آخرین لایه در شکل ۲ دیده می‌شود. به چنین مدل‌هایی، مدل از قبل آموزش دیده[1] گفته می‌شود. استفاده از یک مدل از قبل آموزش دیده برای دسته‌بندی موارد جدیدی که در هزار دسته‌ی اولیه نبوده‌اند، امری مرسوم در حوزه یادگیری عمیق است که از آن با عنوان «انتقال یادگیری[2]» نام برده می‌شود. این روش، نیاز به تعداد زیاد تصاویر آموزشی از دسته‌های جدید و بار پردازشی برای آموزش مجدد شبکه را بسیار کاهش می‌دهد. در بسیاری از روش‌های شناسایی که بر پایه یادگیری عمیق هستند و منجمله در حوزه‌ی شناسایی کووید۱۹ از این شیوه استفاده شده است[۲۰].

اگر قرار به استفاده از این روش در یک مسئله‌ی دو کلاسه — مثل تشخیص کووید۱۹ از غیر آن — باشد، در لایه‌ی آخر ۲ نورون نیاز خواهیم داشت؛ به همین دلیل در آخرین لایه‌ی شکل ۲، زیر عدد هزار، عدد 2 نوشته شده است.

۱-۲ شناسایی کووید۱۹

بیماری مسری ناشی از کرونا ویروس به یک اپیدمی جهانی تبدیل شده و کووید۱۹ (COVID-19) جدیدترین عضو این خانواده به سرعت در حال گسترش است. به دلیل عدم وجود داروهای درمانی اثبات شده، تشخیص زودهنگام این بیماری از اهمیت بالایی برخوردار است. تجارب کشورهای مختلف نشان داده است که بررسی تصاویر پرتونگاری رایانه‌ای (رادیولوژی) و برش‌نگاری رایانه‌ای (سی‌تی‌اسکن)[3] می‌تواند نقش مهمی در تشخیص این بیماری داشته باشد[۲۰-۲۳]. تصویربرداری قفسه‌ی سینه، ناهنجاری‌ها و بی‌قاعدگی‌هایی را در همه بیماران گزارش شده نشان داده است [۲۲]. وجود این ناهنجاری‌ها در تصاویر افراد مبتلا، پایه‌ی اصلی روش‌های هوش مصنوعی برای تفکیک تصاویر قفسه‌ی سینه‌ی افراد سالم و مبتلاست. در روش‌های مرسوم یادگیری ماشین، ویژگی‌های شاخصی از تصاویر که بیانگر تفاوت‌ها باشند باید استخراج شده و به طبقه‌بند داده شود، اما شیوه‌های جدید حوزه‌ی یادگیری عمیق می‌توانند با داشتن تعداد کافی از نمونه‌های آموزشی، ویژگی‌های موردنیاز را به صورت خودکار استخراج کنند. به همین دلیل بسیاری از تحقیقات شناسایی کرونا از روی تصاویر، مبتنی بر یادگیری عمیق هستند که برخی از آنها متضمن میلیون‌ها اتصال و لذا دارای حجمی زیاد می‌باشند.

می‌توان گفت همه‌ی مدل‌های مشهور شبکه‌های عصبی پیچشی در تشخیص کووید۱۹ از روی تصاویر قفسه سینه بکار گرفته شده‌اند. در [۲۴] از انتقال یادگیری و مدل ResNet-101 استفاده شده است. دقت[4] بدست آمده بر روی حدود هفت هزار تصویر رادیولوژی ۷۱,۹ درصد بوده است. منظور از دقت، تعداد نمونه‌هایی است که به درستی طبقه‌بندی شده‌اند، تقسیم بر تعداد کل نمونه‌های مورد بررسی. در [۲۵] ده معماری مختلف شبکه‌های عصبی پیچشی مورد مقایسه قرار گرفته‌اند؛ شبکه‌های AlexNet, VGG-16, VGG-19, SqueezeNet, GoogleNet, MobileNet-V2, ResNet-18, ResNet-50, ResNet-101 و Xception شبکه‌های پیچشی مورد مقایسه بوده‌اند. در این تحقیق دقت مدل VGG16، 83.3 گزارش شده است.

در مرجع [۲۰] چهار مدل MobileNet ,VGG-19 ,VGG-16و InceptionResNetV2 بر روی تصاویر رادیولوژی مورد مقایسه قرار گرفته‌اند. مجموعه دادگان مورد استفاده شامل ۱۸۱ نمونه کووید۱۹ و ۳۶۴ نمونه سالم بوده است. دقت مدل VGG16، ۹۳,۶ گزارش شده است. در [۲۶] مدل‌های VGG16, VGG19, MobileNet V2, Inception V3, Xception, InceptionResNet V2, ResNet152 V2 و DenseNet201 و NASNetLargeبر روی سه دسته از تصاویر نرمال، کووید۱۹ و ذات‌الریه اعمال شده‌اند. مجموعه دادگان مورد استفاده ۳۳۶ تصویر رادیولوژی بوده است. مطابق نتایج گزارش شده، از بین ۹ مدل مورد بررسی، VGG16 با دقت ۹۶ درصد بالاترین کارایی را داشته است. در مرجع [۲۷] مدلهای VGG19، Mobile Net، Inception و Xception مورد مقایسه قرار گرفته‌اند. نتایج حاصله بر روی ۷۲۸ تصویر رادیولوژی، برتری مدل VGG19 با دقت 98.75 را نشان داده است.

گرچه که دقت‌های بسیار بالا گزارش شده است، اما به *دلایل متعدد از جمله تفاوت در مجموعه دادگان مورد استفاده و اختلاف در پارامترهای شبکه‌های عصبی پیچشی، انتخاب یک مدل به عنوان بهترین مدل، بر اساس مقایسه‌ی نتایج مقالات مختلف به آسانی میسر نیست*. هدف در این نوشتار انتخاب یک مدل بهینه نیست، هدف اصلی، کاهش تعداد اتصالات (پارامترهای) یک مدل حجیم از شبکه‌های عصبی پیچشی با ترکیب دو روش هرس شبکه‌های عصبی و بررسی تاثیر این کاهش در کارایی مدل است.

[1] Pre-Trained Model
[2] Transfer Learning
[3] Computerized Tomography (CT) Scan
[4] Accuracy

کاهش حجم مدل می‌تواند موجب کاهش زمان پردازش و کاهش حجم حافظه موردنیاز برنامه گردد. در نتیجه، برنامه مربوطه می‌تواند روی دستگاه‌هایی با توان پردازشی کمتر و حجم حافظه‌ی کم، مانند موبایل‌ها نیز اجرا شود. درخصوص کاهش حجم مدل‌های شبکه‌های عصبی پیچشی کارهای متعددی انجام شده است که در ادامه به برخی از آنها اشاره می‌شود.

۲ کاهش حجم شبکه‌های عصبی پیچشی

تحقیقات نشان داده است که بخش زیادی از اتصالات شبکه‌های عصبی پیچشی بزرگ بدون افت کارایی آن می‌توانند حذف شود [۲۸]. حذف اتصالات عموماً با صفر کردن وزن اتصالاتی که کمتر از آستانه خاصی هستند صورت می‌پذیرد [۲۹]. روش‌های مختلفی برای کاهش تعداد نورون‌ها یا تعداد اتصالات شبکه‌ها ارائه شده است. در [۳۰] اولین بار ایده دراپ‌اوت[1] توسط هینتون[2] مطرح شد؛ در این شیوه نورون‌هایی از شبکه (به همراه اتصالات مربوطه) به صورت تصادفی حذف می‌شوند. گرچه ایده‌ی اصلی ساده است اما در عمل کارایی بسیار خوبی داشته و عموم مدل‌های جدید متضمن چندین لایه دراپ‌اوت هستند. در [۳۱] ایده‌ی دراپ‌اوت هینتون به حذف تصادفی وزن‌ها تعمیم داده شده است. در [۳۲] کارایی این شیوه بر روی شبکه‌های پیچشی مورد بررسی قرار گرفته است.

شیوه‌ی کلی دیگری که در هرس وزن‌های شبکه‌ها استفاده می‌شود، منظم‌سازی[3] است. در [۳۳] این مسئله در قالب یک مسئله‌ی بهینه‌سازی تُنُک[4] فرموله و حل شده است. در [۱۵, ۳۴] از روش‌های منظم‌سازی برای حل مسئله استفاده شده است. نُرم وزن‌های اتصالات شبکه به عنوان یک جمله به تابع هدف مسئله افزوده شده است و روال بهینه‌سازی، همزمان با کمینه کردن تابع هدف به کاهش نُرم وزن‌ها نیز می‌پردازد. وزن‌های با ارزش کمتر از یک حد آستانه، هرس شده و حجم مدل کاهش می‌یابد.

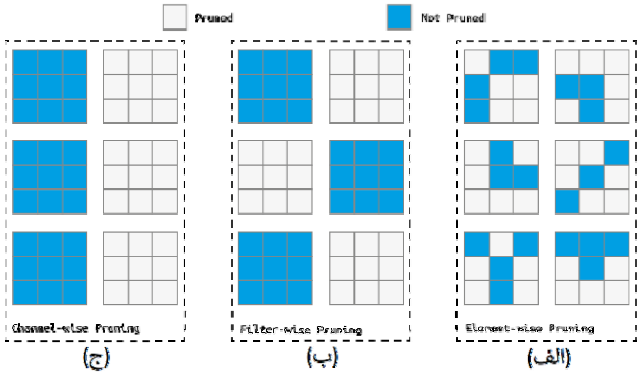
در [۱۱] یک شیوه‌ی مبتنی بر استفاده از بسط تیلور تابع هدف و مبتنی بر روش «آسیب مغزی بهینه» [۱۰] ارائه شده و برتری آن نسبت به چندین روش مرسوم در این حوزه نشان داده شده است. در این نوشتار از ترکیب این دو شیوه‌ی منظم‌سازی و آسیب مغزی بهینه برای کوچک‌سازی مدل VGG16 در کاربرد شناسایی کووید۱۹ استفاده خواهد شد. در ادامه، این دو شیوه به صورت مختصر مرور خواهند شد.

۲-۱ روش منظم‌سازی در کاهش اتصالات شبکه

فرض کنید بردار W بیانگر مجموعه پارامترها (وزن‌های) آموزش‌پذیر مدل شبکه‌ی عصبی پیچشی باشد؛ $J(W)$، تابع هزینه شبکه باشد. این تابع به عنوان مثال می‌تواند تعداد نمونه‌هایی باشد که توسط مدل به اشتباه طبقه‌بندی می‌شوند. تابع هزینه شبکه با قید منظم‌سازی تُنُک به صورت زیر نمایش داده می‌شود:

$$J(W) = loss(W|D) + \lambda \sum_{l=1}^{L} R(W^l) \quad (۱)$$

که در آن $loss(W|D)$ تابع هزینه‌ی مرسوم شبکه‌های عصبی پیچشی، D مجموعه دادگان آموزشی، λ ضریب اهمیت بخش منظم‌ساز، L تعداد لایه‌ها و $R(W^l)$ عبارت منظم‌ساز برای مجموعه وزن‌های لایه‌ی lام است.



شکل ۳- نمایش الگوهای مختلف هرس (برگرفته از [۳۵])

مشهورترین روش منظم‌سازی تُنُک، منظم‌سازی بر مبنای نرم L_2 است که به صورت $\|W^l\|_2^2$ تعریف می‌شود. روش معروف دیگر، استفاده از نرم یک به‌صورت $\|W^l\|_1$ است [۳۶] که با صرفنظر کردن از وزن‌های کم اهمیت، سعی در کاهش بیش‌برازش دارد. این دو شیوه می‌توانند بر روی همه‌ی وزن‌های شبکه یا بر روی گروه‌های خاصی از وزن‌های شبکه اعمال شوند. کارهای متعددی بر روی اعمال منظم‌سازی بر روی گروه‌های خاصی از وزن‌ها همچون فیلترها، ویژگی‌ها، لایه‌ها یا اتصالات ورودی و خروجی نورون‌ها انجام شده است. در شکل ۳ چند نمونه الگوی هرس نشان داده شده است. شکل ۳ (الف) هرس وزن‌ها بدون گروه‌بندی یا الگوی مشخص و شکل‌های (ب و ج) هرس در سطح فیلتر و یا کانال‌های خاصی را نشان می‌دهند.

گروه‌بندی‌های مختلف و نُرم‌های مختلف مورد استفاده، روش‌های متعدد منظم‌سازی را شکل می‌دهند که در قالب عبارت R رابطه‌ی (۱) به تابع هدف اضافه می‌شوند. به عنوان مثال در [۳۷] از تابع زیر برای منظم‌سازی استفاده شده که هدف آن صفر کردن همزمان وزن‌ها در گروه‌های خاص است:

$$R_{GL}(W^l) = \sum_{g\in G} \|W_g^l\|_2 = \sum_{g\in G} \sqrt{\sum_i w_{g,i}^{l\,2}} \quad (۲)$$

---

[1] Dropout

[2] جفری هینتون (Geoffrey Hinton) روانشناس شناختی، دانشمند علوم کامپیوتر و یکی از افرادی است که از آنها به عنوان پدران یادگیری عمیق یاد می‌شود.

[3] Regularization

[4] Sparse Optimization

که در آن G مجموعه‌ی گروه‌های وزنی، $g \in G$ یکی از گروه‌های وزنی، W_g^l بردار یا ماتریس مرتبط با گروه g (که زیربردار یا زیرماتریسی از W^l است) و $w_{g,i}^l$ وزن با اندیس i در گروه g است. در [۳۸] نشان داده شده است که با شیوه‌ی منظم‌سازی تُنُک گروهی می‌توان تعداد پارامترهای شبکه را کاهش داد و حتی دقت مدل را بالا برد.

۲-۲ روش آسیب مغزی بهینه در کاهش اتصالات

مجدداً فرض کنید بردار W بیانگر مجموعه پارامترها (وزنها یا اتصالات) مدل شبکه‌ی عصبی و $J(W)$ تابع هزینه شبکه باشد؛ در روش آسیب مغزی بهینه، هدف، یافتن مجموعه وزن‌های W'، به نحوی است که مقدار تابع هزینه با این وزنها ($J(W')$)، تفاوت زیادی نسبت به مقدار هزینه‌ی اولیه نداشته و ضمناً بیشتر عناصر W' برابر با صفر باشند. اگر حد بالای تعداد عناصر غیرصفر بردار وزن، B باشد، مسئله اصلی را می‌توان به صورت رابطه‌ی (۳) نوشت:

$$\min_{W'} |J(W') - J(W)| \quad s.t. \quad \|W'\|_0 \le B \qquad (۳)$$

نرم صفر یک بردار، تعداد عناصر غیرصفر آن‌را مشخص می‌کند. مدل VGG16، با ۱۳ لایه‌ی پیچشی و ۳ لایه‌ی تمام متصل بیش از ۱۳۰ میلیون وزن اتصالی دارد (جدول ۱). عموماً انتخاب وزن‌ها برای هرس به صورت تکی صورت نمی‌پذیرد، بلکه مثلاً یک فیلتر از یک لایه‌ی پیچشی برای حذف انتخاب می‌شود. مدل VGG16 دارای ۴۲۲۴ فیلتر در ۱۳ لایه‌ی پیچشی خود است ($2 \times 64 + 2 \times 128 + 3 \times 256 + 6 \times 512 = 4224$).

مسئله‌ی (۳) حتی با این تعداد هم یک مسئله‌ی بهینه‌سازی ترکیبیاتی محسوب می‌شود که بررسی همه حالات آن عملی نیست. در [۲, ۱۰, ۱۱] با استفاده از بسط تیلور تابع هزینه، میزان تغییر تابع هزینه (تابع هدف) در صورت صفر شدن برخی از مؤلفه‌های W، در قالب مشتق تابع هدف، ضرب‌در مؤلفه‌ی مربوطه برآورد شده است. اگر h مجموعه وزن‌های یک فیلتر خاص باشد، میزان تغییر تابع هزینه در صورت صفر شدن این وزنها را به صورت زیر داریم:

$$\Delta J(h) = |J(h=0) - J(h)| \qquad (۴)$$

بر اساس بسط تیلور تابع هدف حول $h = 0$ داریم:

$$J(h) = J(0) + J'(0)h + \frac{J''(0)}{2!}h^2 + \cdots$$
$$\rightarrow \quad J(h=0) \cong J(h) - \frac{\partial J}{\partial h}h$$
$$\rightarrow \quad |\Delta J(h)| \cong \left|\frac{\partial J}{\partial h}h\right| \qquad (۵)$$

مطابق رابطه‌ی (۵) تغییر تابع هدف، درصورت صفر شدن برخی از مؤلفه‌های بردار وزن، می‌تواند برحسب مشتق تابع هدف، ضربدر مؤلفه‌ی مربوطه تخمین زده شود. از این موضوع برای انتخاب اتصالاتی که اثر کمی در تغییر تابع هدف دارند استفاده می‌شود. اتصالات بر اساس رابطه فوق مرتب شده و وزن‌هایی که کمترین میزان تغییر تابع هدف را باعث می‌شوند، هرس می‌شوند.

با این ایده، در یک روند تکراری، در هر دور، تعدادی از وزن‌ها انتخاب و هرس می‌شوند. سپس شبکه مجدداً آموزش می‌بینند تا اتصالات باقیمانده برای جبران اثر حذف اتصالات هرس شده، مقداری بروزرسانی شوند و شبکه کارایی قبل از آسیب خود را بازیابد.

۳ روش پیشنهادی و نتایج روی کووید۱۹

در بخش‌های پیش برخی از روش‌های کاهش حجم پارامترهای یک شبکه بیان شدند. ساده‌ترین روش، شیوه‌ی دراپ‌اوت است؛ گرچه که به صورت گسترده‌ای مورد استفاده قرار می‌گیرد اما با این روش حجم شبکه در عمل قابل کاهش نیست؛ چرا که ساختارهای مورد استفاده، اجازه چنین کاری را نمی‌دهند. به عنوان مثال اگر به مانند شکل ۳، یک فیلتر با یک ماتریس ۳×۳ پیاده‌سازی شده باشد، امکان حذف عناصر آبی‌رنگ شکل ۳ (الف) و حفظ ساختار ماتریسی وجود ندارد.

در عموم پیاده‌سازی‌های انجام شده‌ی روش‌های هرس شبکه و منجمله در پیاده‌سازی[1] شیوه‌ی مطرح شده در [۱۵] فقط به صفر کردن وزن‌ها اکتفا می‌شود و حجم واقعی مدل کاهش نمی‌یابد.

در [۱۵]، *هفده روش مختلف منظم‌سازی مورد بررسی قرار گرفته است*. نتایج مرجع اخیر نشان داده است که از بین روش‌های متعدد مورد بررسی، شیوه‌ی زیر برای کاهش حجم مدل VGG16 بیشترین کارایی را داشته است (جدول III از مرجع [۱۵]):

$$R_{HSQ-GL_{1/2}}(W^l) = \sum_{j=1}^{ic_l}\left(\sum_{i=1}^{oc_l}\sqrt{\sum_{h=1}^{H_l}\sum_{w=1}^{W_l}|w_{i,j,h,w}^l|}\right)^2 \qquad (۶)$$

که در آن ic_l ،oc_l ابعاد W^l در کانال‌های ورودی و خروجی و H_l و W_l ابعاد کرنل مربوطه می‌باشد. به عنوان نمونه در لایه‌ی پیچشی سوم مدل VGG16 نمایش داده شده در جدول ۱، ابعاد کانالهای ورودی و خروجی به ترتیب ۶۴ و ۱۲۸ و اندازه‌ی کرنل ۳×۳ است. انتظاری که از این قید می‌رود آن است که نورون‌های ورودی هر لایه هرس شوند. در ادامه‌ی این نوشتار، از این شیوه‌ی منظم‌سازی و با نام HSQGL12[2] استفاده خواهد شد.

شیوه‌ی دیگری که در بخش‌های پیش بیان شد، روش آسیب مغزی بهینه (OBD) بود که اتصالات شبکه، بر اساس حاصلضرب مشتق تابع هدف در وزن اتصال رده‌بندی و هرس می‌شوند.

در این پژوهش تاثیر هر یک از این دو روش به صورت مجزا و ترکیبی در کاهش حجم مدل مورد بررسی قرار خواهد گرفت. وابسته به تقدم و تأخر استفاده از دو روش فوق‌الذکر، دو روش ترکیبی می‌توان داشت: در روش اول، ابتدا از روش منظم‌سازی و سپس روش آسیب بهینه‌ی مغزی استفاده می‌شود و در روش دوم، ابتدا روش آسیب بهینه‌ی مغزی و سپس روش منظم‌سازی اعمال

[1] https://github.com/K-Mitsuno/hierarchical-group-sparse-regularization
[2] Hierarchical Squared Group Lasso $L_{1/2}$ Regularization

خواهد شد. نتایج آزمایشات نشان داده است که در این کاربرد شیوه‌ی ترکیبی اول، کارایی بهتری دارد. پیاده‌سازی در بستر پای‌تورچ[1] و مبتنی بر پیاده‌سازی مؤلفین [۱۵] و یک پیاده‌سازی[2] از [۱۱] صورت پذیرفته است.

۳-۱ دادگان کووید۱۹

داده‌های مورد استفاده، مجموعه تصاویر مورد استفاده در [۲۳, ۳۹] هستند که در کاگل[3] در معرض استفاده عموم قرار داده شده است. مجموعه داده‌ها شامل ۳۶۱۶ تصویر کووید مثبت ۱۰۱۹۲ تصویر نرمال و دو دسته‌ی دیگر است که در این پژوهش فقط از دو گروه فوق استفاده شده است. زیرمجموعه‌ای از تصاویر فوق توسط ابزارهای پایتون[4] به دسته‌های آموزش، اعتبارسنجی و آزمون تقسیم شدند. از آنجا که هدف این نوشتار بررسی روش ترکیبی پیشنهادی برای کاهش حجم مدل است، نیازی به یک مجموعه داده‌ی بزرگ برای آموزش مدل نیست؛ تعدادی تصویر که بتوان یک نرخ شناسایی در حدود میانگین موارد گزارش شده را بدست آورده و روش پیشنهادی را مورد ارزیابی قرار داد، کفایت خواهد کرد. از این‌رو ۲۰۰۰ نمونه برای آموزش، ۲۰۰ نمونه برای اعتبارسنجی و ۲۰۰ نمونه برای آزمون درنظر گرفته شدند. با این زیرمجموعه از کل تصاویر، میزان دقت ۸۳ درصد حاصل شد که نزدیک به میانگین دقت‌های ۷۱ تا ۹۸ درصد ذکر شده در منابع مذکور در ۱-۲ است. زیرمجموعه‌ی موردنظر از گیت‌هاب نگارنده قابل دسترس است[5].

۳-۲ آموزش مدل برای شناسایی کووید۱۹

مدل از قبل آموزش دیده‌ی VGG16 در پای‌تورچ[6] با حجم ۵۲۸ مگابایت به عنوان مدل اولیه انتخاب شده است. ابتدا ساختار لایه‌ی آخری مدل که به صورت پیش‌فرض برای هزار کلاس درنظر گرفته شده بود، برای دو کلاس (COVID و Normal) اصلاح و انتقال یادگیری انجام شد. در شکل ۲ ساختار این مدلِ اصلاح شده و در جدول ۱ پارامترهای آن ذکر شده بود. برنامه بر روی سرورهای گوگل با کارت گرافیک Tesla T4-15GB اجرا شده است. برای بروزرسانی وزن‌ها از روش بهینه‌سازی Adam[۴۰] استفاده شد.

مدل اصلاح شده‌ی VGG16، چهل اِپُک[7] بر روی داده‌های آموزشی، آموزش داده شد و میزان دقت در هر دور بر روی داده‌های اعتبارسنجی برآورد گردید. بر اساس نتایج بدست آمده، به لحاظ آنکه مدل دچار بیش‌برازش نشود، ۲۵ اپک برای ایجاد مدل مناسب تشخیص داده شد. آموزش مدل اصلی بر روی دو هزار تصاویر آموزشی، ۶ دقیقه به طول انجامید. این مدل را VGG-COVID خواهیم نامید. در این مرحله از آموزش، فقط لایه‌های مربوط به طبقه‌بند (یعنی سه لایه‌ی آخر، Dense-1,2,3 از جدول ۱) بروزرسانی می‌شوند. مدل حاصل دارای میزان دقت بر روی داده‌های اعتبارسنجی برابر با 0.85 و بر روی داده‌های تست برابر با 0.83 بود. مدت زمان مورد نیاز برای بررسی ۲۰۰ تصویر آزمون، ۱۰۰ ثانیه بوده است. بر روی تصاویر ورودی فقط نرمال‌سازی و تغییر اندازه و برش انجام شده است که تصاویر ورودی همه ۲۲۴ در ۲۲۴ باشند. حجم مدل حاصله به دلیل کاهش تعداد دسته‌ها به دو گروه، مقداری از مدل اصلی کمتر شده و 524.5 مگابایت می‌باشد. همه برنامه‌های این نوشتار با قابلیت اجرای آنلاین، از گیت‌هاب نگارنده قابل دسترس هستند[8].

۳-۳ کاهش اتصالات لایه‌های پیچشی با روش منظم‌سازی

روش HSQGL12 که در مرجع [۱۵] از بین ۱۷ روش هرس با شیوه‌ی منظم‌سازی، بهترین گزینه برای مدل VGG بوده است بر روی مدل VGG-COVID آموزش داده شده در بخش قبل اعمال شد. به این منظور مدل فوق‌الذکر بر روی داده‌های آموزشی و با افزودن قید رابطه‌ی (۶) ۳۰ اپک آموزش داده شد. این مدل را VGG-HSQGL12 خواهیم نامید. در اینجا فقط به لایه‌های پیچشی یا کانولوشنی (لایه‌های ۱ تا ۱۳ از جدول ۱) اجازه آموزش داده می‌شود. کل تعداد اتصالات آموزش‌پذیر در این لایه‌ها ۴۶۴.۷۱۰.۱۴ می‌باشد. مدت زمان آموزش ۱۶ دقیقه و ۱۳ ثانیه بوده است. مدت زمان مورد نیاز برای بررسی ۲۰۰ تصویر آزمون، مشابه مدل قبلی و حدود ۹۱ ثانیه بوده است. در انتها، میزان دقت بر روی داده‌های اعتبارسنجی 0.865 و بر روی داده‌های تست 0.83 بوده است.

با اینکه دقت مدل جدید بر روی داده‌های آزمون کمتر از مدل اولیه نشده است اما همانگونه که پیشتر ذکر شد، در پیاده‌سازیِ مؤلفین مرجع [۱۵] حذف واقعی اتصالات و نورون‌ها صورت نمی‌پذیرد، لذا حجم آن به مانند مدل VGG-COVID، ۵۲۴,۵ مگابایت و مشخصات مدل حاصله، دقیقاً همانند مدل VGG-COVID ذکر شده در جدول ۱ خواهد بود. با فرض سطح آستانه 0.001 برای کم اهمیت شمردن اتصالات، این شبکه ۲,۴۴۰,۲۹۶ وزن نزدیک به صفر (که صفر درنظر می‌گیریم) دارد. این تعداد، کمتر از دو درصد تعداد کل وزن‌های مدل VGG-COVID است؛ به این معنی که حذف واقعی آنها نیز حجم مدل را خیلی تحت تاثیر قرار نخواهد داد.

۳-۴ کاهش اتصالات لایه‌های پیچشی با روش آسیب مغزی بهینه

[1] PyTorch: https://pytorch.org/
[2] https://github.com/jacobgil/pytorch-pruning
[3] https://www.kaggle.com/tawsifurrahman/covid19-radiography-database
[4] split-folders · PyPI
[5] https://github.com/mamintoosi/Reg-OBD-for-VGG-Pruning/tree/main/data/COVID-Radiography
[6] https://download.pytorch.org/models/vgg16-397923af.pth
[7] اِپُک (Epoch): هر دور که همه‌ی داده‌های آموزشی در روال آموزشی مورد استفاده قرار می‌گیرند، یک اپک نام دارد.
[8] https://github.com/mamintoosi/Reg-OBD-for-VGG-Pruning

به صورت جداگانه، روش آسیب مغزی بهینه (OBD) [۱۱] بر روی مدل VGG-COVID آموزش داده شده‌ی قبلی با هدف کاهش حدود ۷۰ درصدی اتصالات اعمال شد. به این منظور مدل فوق‌الذکر بر روی داده‌های آموزشی ۲۵ اپک آموزش داده شد. این مدل را VGG-COVID نامیده و به عنوان مرجع مقایسه در نظر خواهیم گرفت. در اینجا نیز فقط به لایه‌های پیچشی (لایه‌های ۱ تا ۱۳ از جدول ۱) اجازه آموزش داده شد. مدت زمان آموزش ۱۶ دقیقه و پنجاه ثانیه بوده است. در انتها، میزان دقت بر روی داده‌های اعتبارسنجی 0.79 و بر روی داده‌های تست 0.73 بوده است.

جدول ۲- پارامترهای مدل OBD. فقط لایه‌های پیچشی (کانولوشنی) مدل VGG_COVID با روش [۱۱] هرس شده‌اند.

Layer (type)	Output Shape	Param #
Conv2d-1	[33, 224, 224]	924
Conv2d-2	[32, 224, 224]	9,536
Conv2d-3	[79, 112, 112]	22,831
Conv2d-4	[81, 112, 112]	57,672
Conv2d-5	[129, 56, 56]	94,170
Conv2d-6	[136, 56, 56]	158,032
Conv2d-7	[132, 56, 56]	161,700
Conv2d-8	[210, 28, 28]	249,690
Conv2d-9	[206, 28, 28]	389,546
Conv2d-10	[160, 28, 28]	296,800
Conv2d-11	[175, 14, 14]	252,175
Conv2d-12	[169, 14, 14]	266,344
Conv2d-13	[122, 14, 14]	185,684
Dense-1	[4096]	24,489,984
Dense-2	[4096]	16,781,312
Dense-3	[2]	8,194
	Total params:	43,424,594

تعداد ۲۵ اپک فوق‌الذکر که در قالب پنج مرحله‌ی هرس و هر مرحله، پنج اپک برای ترمیم شبکه، درنظر گرفته شده، به این دلیل بوده است که از نظر زمان اجرا، معادل زمان اجرای روش منظم‌سازی باشد. در واقع با چندین بار اجرای دو روش، تعداد اپک مناسب برای هر دو شیوه به نحوی انتخاب شده است که مدل‌ها دچار بیش‌برازش نشده و در عین حال زمان اجرای تقریباً یکسانی داشته باشند که عملکرد آنها قابل مقایسه باشد.

در روش «آسیب مغزی بهینه» در یک روال تکراری در هر دور ۵۱۲ فیلتر انتخاب و هرس شدند (در مجموع ۵×۵۱۲=۲۵۶۰ فیلتر). بعد از هر سری حذف ۵۱۲ فیلتر، شبکه پنج اپک آموزش مجدد داده شد تا شبکه با آسیب وارد شده به واسطه‌ی حذف اتصالات هرس شده، تطبیق پیدا کند.

تعداد اتصالات مدل حاصله از بیش از ۱۳۴ میلیون (جدول ۱) به حدود 43.5 میلیون کاهش پیدا کرد (جدول ۲) که معادل حدود ۳۲ درصد از تعداد پارامترهای مدل VGG-COVID می‌باشد.

حجم واقعی مدل جدید حدود ۱۷۰ مگابایت شد که این نیز حدود ۳۲ درصد حجم مدل VGG-COVID می‌باشد. به عبارت دیگر میزان کاهش حجم پارامترها و مدل حدود ۶۸ درصد بوده است. با این مدل هرس شده، مدت زمان مورد نیاز برای بررسی ۲۰۰ تصویر آزمون، ۳۱ ثانیه بوده است (0.16 ثانیه برای هر تصویر) که در مقایسه با بیش از ۹۰ ثانیه‌ی دو روش پیشین، حدود ۶۶ درصد کاهش زمان پردازش داشته است.

۳-۵ کاهش اتصالات با روش ترکیبی

به عنوان شیوه‌ی پیشنهادی، ترکیب دو روش فوق‌الذکر را در این بخش خواهیم داشت. نتایج آزمایشات انجام شده نشان داده است که ابتدا انجام شیوه‌ی منظم‌سازی و سپس اعمال روش آسیب مغزی بهینه بر روی مدل حاصله، نتایج بهتری نسبت به حالتی دارد که ترتیب جابجا شود؛ لذا در این بخش نتایج این شیوه ذکر خواهد شد. در شیوه‌ی ترکیبی پیشنهادی ابتدا بر روی مدل VGG-COVID، روش منظم‌سازی مذکور در بخش ۳-۳ اعمال می‌شود که نتیجه‌ی آن همان مدلی می‌شود که HSQGL12 نامیده شد. حال بر روی این مدل که دقت بهتری نسبت به مدل اولیه دارد، اما حجم آن با مدل VGG-COVID یکسان است، شیوه‌ی آسیب مغزی بهینه اعمال می‌شود. مدت زمان آموزش ۱۷ دقیقه و ۴۵ ثانیه بوده است. جدول ۳ مشخصات مدل حاصله را نشان می‌دهد.

جدول ۳- پارامترهای مدل ترکیبی HSQGL12-OBD. ابتدا با روش منظم‌سازی، وزن اتصالات کاهش یافته و سپس لایه‌های پیچشی با روش آسیب مغزی بهینه هرس شده‌اند.

Layer (type)	Output Shape	Param #
Conv2d-1	[36, 224, 224]	1,008
Conv2d-2	[36, 224, 224]	11,700
Conv2d-3	[90, 112, 112]	29,250
Conv2d-4	[78, 112, 112]	63,258
Conv2d-5	[166, 56, 56]	116,698
Conv2d-6	[128, 56, 56]	191,360
Conv2d-7	[149, 56, 56]	171,797
Conv2d-8	[212, 28, 28]	284,504
Conv2d-9	[211, 28, 28]	402,799
Conv2d-10	[179, 28, 28]	340,100
Conv2d-11	[178, 14, 14]	286,936
Conv2d-12	[137, 14, 14]	219,611
Conv2d-13	[64, 14, 14]	78,976
Dense-1	[4096]	12,849,152
Dense-2	[4096]	16,781,312
Dense-3	[2]	8,194
	Total params:	31,836,655

جدول ۴- خلاصه نتایج آزمایشات انجام شده. موارد با حروف پررنگ، نشان دهنده‌ی کارایی برتر هستند. مدل VGG-COVID به عنوان مدل پایه برای مقایسات درنظر گرفته شده است.

نام مدل توضیح مختصر	VGG-COVID مدل VGG آموزش دیده برای شناسایی کووید۱۹	HSQGL12 اعمال روش منظم‌سازی برای هرس اتصالات مدل VGG-COVID	OBD اعمال روش آسیب مغزی بهینه برای هرس اتصالات مدل VGG-COVID	HSQGL12-OBD **روش پیشنهادی: ترکیب دو روش هرس شبکه، ابتدا روش منظم‌سازی و سپس آسیب مغزی بهینه**
دقت روی داده‌های آزمون	0.83	0.83	0.73	0.83
تعداد اتصالات مدل	134,268,738	134,268,738	43,424,594	31,836,655
نسبت کاهش تعداد اتصالات به مدل پایه	-	۰	0.32	**0.24**
حجم فایل مدل (مگابایت)	524.5	524.5	169.6	**124.4**
نسبت حجم مدل به مدل پایه	-	۱	۰,۳۲	**۰,۲۴**
میزان کاهش حجم مدل	-	۰	۶۸٪	**۷۶٪**
زمان پردازش ۲۰۰ تصویر آزمون (ثانیه)	۱۰۰	۹۱	**۳۱**	۳۳
میانگین زمان پردازش برای هر تصویر	0.5 ثانیه	0.46 ثانیه	**0.16 ثانیه**	0.17 ثانیه

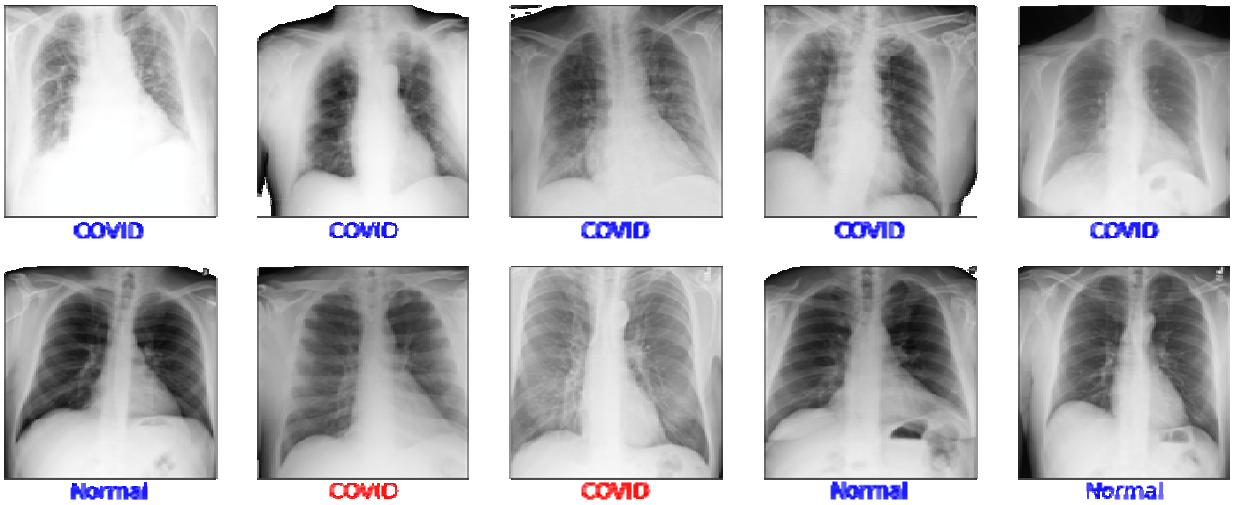
میزان دقت بر روی داده‌های اعتبارسنجی 0.89 و بر روی داده‌های تست 0.83 و زمان تست ۳۳ ثانیه بوده است (0.17 ثانیه برای هر تصویر). این مدل را HSQGL12-OBD خواهیم نامید. در اینجا نیز به مانند بخش ۴-۳، ۲۵۶۰ فیلتر هرس شدند اما همان‌گونه که در جدول ۳ ملاحظه می‌شود، تعداد کل پارامترهای مدل حاصله ۶۵۵.۸۳۶.۳۱ است که از تعداد پارامترهای مدل OBD (جدول ۲) با همین شیوه و همین تعداد فیلتر هرس شده، کمتر است. به چرایی موضوع در بخش بعد پرداخته خواهد شد. این تعداد اتصالات، حدود ۲۴ درصدِ اتصالات شبکه‌ی VGG-COVID هستند که به معنی کاهش ۷۶ درصدی تعداد پارامترهای شبکه‌ی اولیه می‌باشد.

۳-۵ خلاصه نتایج آزمایشات

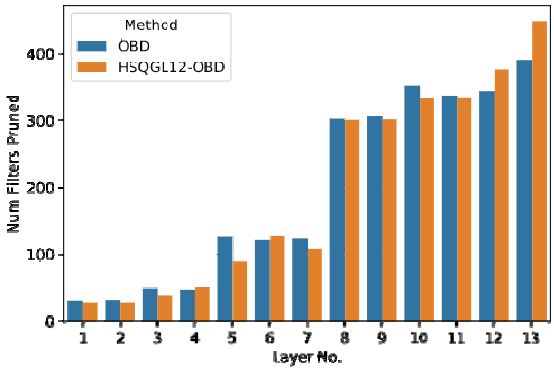
نتایج آزمایشات بر روی مدل VGG16، به عنوان یکی از مدل‌های مشهور شبکه‌های عصبی پیچشی و در کاربرد شناسایی کووید۱۹، نشان داد که با ترکیب روش‌های هرس شبکه می‌توان تعداد اتصالات را ۷۶ درصد کاهش داد، بدون آنکه کارایی مدل کاهش یابد. این کاهش حجم مدل، زمان پردازشی را به یک سوم تقلیل داده است. این کاهش حجم، بدون کاهش کارایی شبکه نشان می‌دهد بسیاری از ویژگی‌های استخراج شده توسط مدل VGG16 در شناسایی کووید۱۹ مثمرثمر نبوده و می‌توانند نادیده گرفته شوند.

جدول ۴ خلاصه نتایج آزمایشات را نشان می‌دهد. همان‌گونه که در دو ستون آخر این جدول مشاهده می‌شود، روش‌های مبتنی بر آسیب مغزی بهینه، در عین حفظ کارایی مدل، تعداد اتصالات بیشتری را هرس کرده و به مدل کوچک‌تری رسیده‌اند. با اغماض از یک‌صدم ثانیه زمان پردازشی بیشتر ستون آخر برای هر تصویر (روش ترکیبی پیشنهادی) نسبت به ستون سوم، می‌توان گفت روش پیشنهادی نسبت به سایر روش‌ها در عین حفظ کارایی اولیه‌ی مدل، زمان پردازشی و حجم مدل کمتری ارائه داده است. حجم مدل نتیجه، ۲۴ درصد حجم مدل اولیه شده است که به معنی کاهش ۷۶ درصدی حجم مدل اولیه است. قابل ذکر است که شیوه‌ی HSQGL12 در بین ۱۷ روش هرس شبکه در مرجع [۱۵] برای مدل VGG بیشترین کارایی را داشته است که روش ترکیبی پیشنهادی توانسته است نتایج آن‌را بهبود بخشد.

دو هر دو شیوه‌ی دو ستون آخر (یعنی OBD و HSQGL12-OBD) ۲۵۶۰ فیلتر هرس شده‌اند، اما ستون آخر، تعداد اتصالات کمتری نسبت به قبلی خود دارد. نمودار ۱ فراوانی تعداد فیلترهای حذف شده در هر یک از ۱۳ لایه‌ی پیچشی در دو روش فوق را نشان می‌دهد. با ملاحظه‌ی این نمودار مشخص می‌شود که روش پیشنهادی نسبت به روش OBD، در لایه‌های آخر فیلترهای بیشتری را هرس کرده است، در حالیکه روش OBD در لایه‌های اول فیلترهای بیشتری نسبت به روش پیشنهادی هرس کرده است؛ و از آنجا که عمق فیلترها (و در نتیجه تعداد پارامترها) در لایه‌های آخر مدل VGG نسبت به لایه‌های اول بیشتر است، لذا تعداد اتصالات هرس شده با شیوه‌ی HSQGL12-OBD نسبت به شیوه‌ی OBD بیشتر شده است. البته صرفنظر از تفاوت مختصر تعداد پارامترها، روش پیشنهادی کارایی بیشتری نسبت به OBD دارد.



شکل ۴- یک نمونه خروجی روش پیشنهادی بر روی ده تصویر آزمون. تصاویر سطر اول، همه کووید۱۹ و تصاویر سطر دوم همه نرمال هستند. برچسب زیر هر تصویر، خروجی مدل را مشخص می‌کند. اگر خروجی مدل با مقدار درست مطابقت داشته باشد، برچسب با رنگ آبی و در غیر اینصورت با رنگ قرمز نمایش داده شده است.



نمودار ۱- نمودار فراوانی تعداد فیلترهای هرس شده در هر یک از ۱۳ لایه‌ی پیچشی با روش OBD و روش HSQGL12-OBD.

یک نمونه خروجی روش پیشنهادی بر روی مجموعه دادگان کووید۱۹ در شکل ۴ آمده است. برنامه‌های پایتون مرتبط با مقاله به همراه داده‌ها از گیت‌هاب نگارنده قابل دسترس است[1].

۴ نتایج آزمایشات روی مجموعه دادگان عمومی

در این بخش نتایج ترکیب دو شیوه‌ی هرس شبکه بر روی چند مجموعه داده‌ی عمومی[2] MNIST، FashionMNIST و STL10 ذکر خواهد شد. مجموعه داده‌ی MNIST شامل ۷۰هزار نمونه ارقام انگلیسی دست‌نویس ۲۸×۲۸ پیکسل است. مجموعه داده‌ی FashionMNIST مشابه MNIST است با این تفاوت که تصاویر از ده گروه زیر هستند:

T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot

و مجموعه دادگان STL10 شامل ۱۳ هزار تصویر رنگی از ده گروه زیر است:

airplane, bird, car, cat, deer, dog, horse, monkey, ship, truck

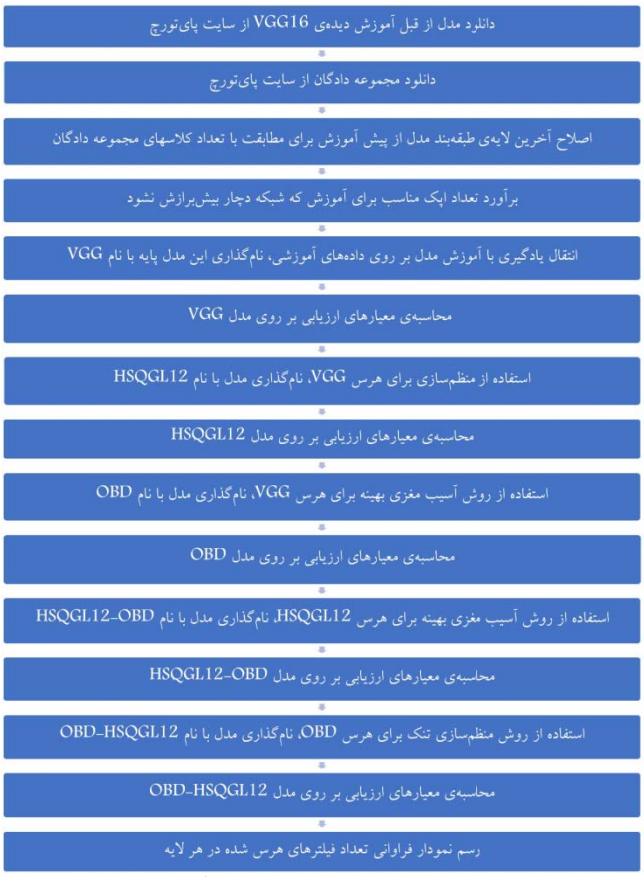
از آنجا که زمان آموزش و هرس شبکه با روش‌های مختلف روی کل مجموعه دادگان فوق زیاد است، از دو مجموعه‌ی اول ده درصد و از مجموعه دادگان سوم پنجاه درصد نمونه‌های آموزشی و آزمون به تصادف انتخاب شده‌اند. با زیرمجموعه‌های انتخاب شده، زمان موردنیاز برای اجرای برنامه‌های این نوشتار، بر روی سرورهای گوگل با کارت گرافیک Tesla T4-15GB، حدود ۳ ساعت بوده است.

نحوه‌ی کلی اجرا بر روی مجموعه دادگان مشابه هم است که در روندنمای ۱ نشان داده شده است. برنامه‌ی مربوط به هر مجموعه داده با نام main_DATASETNAME.ipynb از گیت‌هاب مرتبط با برنامه‌های این مقاله در دسترس است.

برای هر یک از مجموعه دادگان، ابتدا مدل از قبل آموزش دیده‌ی VGG16، ده اپک بر روی مجموعه‌ی آموزشی، آموزش داده شد. مدلی که کمترین خطا بر روی داده‌های اعتبارسنجی را داشته است، به عنوان مدل پایه‌ی VGG16 به منظور هرس انتخاب می‌گردد. برخلاف مدل آموزش دیده برای کووید۱۹ که آنرا VGG-COVID نامیدیم، در این بخش مدل اولیه‌ی حاصل از انتقال یادگیری را برای همه‌ی مجموعه دادگان VGG خواهیم نامید و منظور مدل VGG16 است که برای آن مجموعه دادگان خاص آموزش دیده است. به این ترتیب برنامه‌ها، مدل‌ها و نمودارها و نتایج همه با یک نام‌گذاری واحد خواهند بود.

با هر یک از روش‌های منظم‌سازی و آسیب مغزی بهینه و ترکیب آنها هرس مدل پایه انجام و با استفاده از بسته‌ی PyCM [۴۱] که توسط سپند حقیقی و همکاران ایشان آماده شده است، مدل نتیجه بر اساس معیارهای مختلف مورد ارزیابی قرار گرفت.

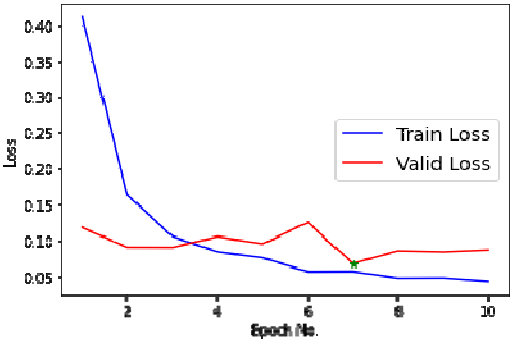
[1] https://github.com/mamintoosi/Reg-OBD-for-VGG-Pruning
[2] https://pytorch.org/vision/stable/datasets.html



روندنمای ۱- روال کلی اجرا بر روی مجموعه دادگان عمومی

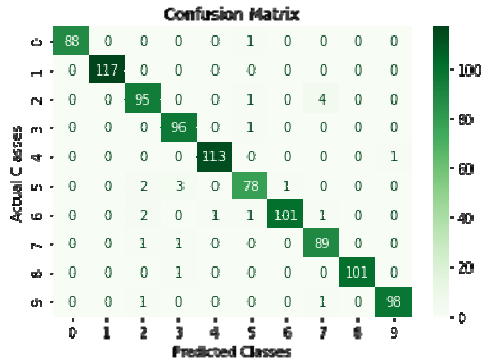
۴-۱ نتایج آزمایشات روی MNIST

پس از انجام سه مرحله‌ی اول مذکور در روندنمای ۱، نتیجه‌ی مرحله‌ی چهارم به صورت زیر بوده است:



شکل ۵: نمودارهای روند کاهش خطا داده‌های آموزشی و اعتبارسنجی بر روی داده‌های MNIST

مدل اپک ۷، به عنوان مدل بهینه برای هرس انتخاب شد. ماتریس درهم‌ریختگی مدل حاصل به صورت نمایش داده شده در شکل ۶ است.



شکل ۶- ماتریس درهم‌ریختگی مدل پایه‌ی VGG بر روی داده‌های MNIST، با دقت برابر با 0.98.

مقادیر سایر معیارهای ارزیابی در جدول انتهای این بخش به همراه سایر مدل‌ها نمایش داده خواهد شد. سپس روش‌های هرس شبکه‌ی منظم‌سازی و آسیب مغزی بهینه و ترکیب آنها بر روی این مدل پایه اعمال شده است که نتایج میانی و ماتریس درهم ریختگی مدل حاصل در برنامه‌ی مربوطه قابل مشاهده است که به لحاظ جلوگیری از طولانی شدن مطلب فقط خلاصه نتایج در جدول ۶ ذکر شده است. بهترین نتایج کسب شده در هر ستون با حروف پررنگ نمایش داده شده است (به جز در معیارهایی که همه مثل هم بوده‌اند).

جدول ۶- نتایج ارزیابی روش‌های مختلف هرس بر روی مجموعه دادگان MNIST

Sparsity		Size	Macro					
ZPR	PPR	MB	TPR	TNR	FPR	FNR	ACC	Method
۰/۰۱۳	-	۵۲۵	۰/۹۷	۱	۰	۰/۰۳	۰/۹۸	VGG
۰/۱۱۶	۰	۵۲۵	۰/۹۸	۱	۰	۰/۰۲	۰/۹۸	HSQGL۱۲
۰/۵۸۹	۰/۵۸۵	۲۱۸	۰/۹۸	۱	۰	۰/۰۲	۰/۹۸	OBD
۰/۳۸۱	۰/۳۷۵	۳۲۸	۰/۹۹	۱	۰	۰/۰۱	۰/۹۹	HSQGL۱۲-OBD
۰/۶۰۰	۰/۵۸۵	۲۱۸	۰/۹۸	۱	۰	۰/۰۲	۰/۹۸	OBD-HSQGL۱۲

منظور از ACC، FNR، FPR، TNR و TPR، به ترتیب دقت[1]، نرخ منفی کاذب[2]، نرخ مثبت کاذب[3]، نرخ منفی صادق[4] و نرخ مثبت صادق[5] است که به صورت زیر محاسبه می‌شوند:

$$FNR = FN/(FN+TP),\quad FPR = FP/(FP+TN)$$
$$TPR = TP/(TP+FN),\quad TPR = TP/(TP+FN)$$

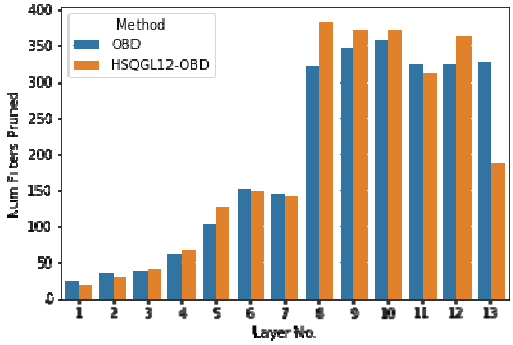
که البته حالت «کلان[6]» معیارهای فوق محاسبه شده است. منظور از کلان آن است که معیارهای فوق برای هر کلاس به صورت «یکی در برابر باقی[7]» محاسبه شده و سپس میانگین‌گیری انجام شده است.

ستون Size، مشخص کننده‌ی حجم فایل مدل بر حسب مگابایت (MB) است. برای میزان تُنُکی (Sparsity) دو معیار در دو ستون

[1] Accuracy
[2] False Negative Rate (FNR)
[3] False Positive Rate (FPR)
[4] True Negative Rate (TNR)
[5] True Positive Rate (TPR)
[6] Macro
[7] One-vs-All

آخر ذکر شده است: PPR[1] و ZPR[2]. منظور از PPR، نرخ پارامترهای (اتصالات) هرس شده است. یعنی تعداد اتصالاتی که واقعاً از مدل حذف شده‌اند به کل اتصالات مدل پایه. منظور از ZPR، نرخ پارامترهایی است که صفر یا حذف شده‌اند. اگر وزن یک پارامتر (اتصال) کمتر از یک‌دهزارم بوده است، آن وزن صفر تلقی شده است. همان‌گونه که مشاهده می‌شود، در هر معیار، حداقل یکی از دو شیوه‌ی ترکیب دو روش هرس، نتایجی بهتر یا مساوی با هر یک از روش‌های هرس به تنهایی داشته است.

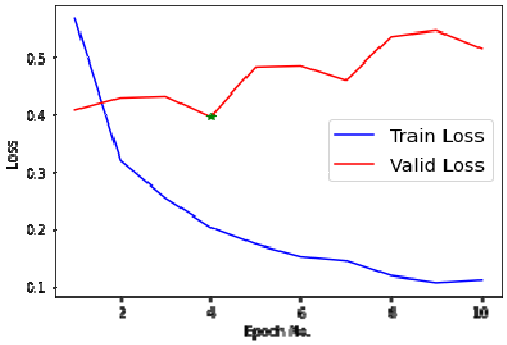
قبلاً در نمودار ۱، نمودار فراوانی تعداد فیلترهای هرس شده برای دو مدل OBD و HSQGL12-OBD برای کوویید۱۹ نمایش داده شد. در نمودار ۲، نمودار مشابه برای دو مدل مذکور بر روی داده‌های MNIST نشان داده شده است. از آنجا که فقط روش آسیب‌مغزی بهینه‌ی مورد استفاده (OBD) هرس واقعی اتصالات را انجام می‌دهد، فقط این دو مدل در مقایسه آمده‌اند. مطابق جدول ۶، حجم مدل OBD از HSQGL12 کمتر بوده است (۲۱۸ در برابر ۳۲۸ مگابایت)؛ نمودار ۲ مؤید این اختلاف حجم هست: اگر به جدول ۱ که تعداد فیلترها و تعداد پارامترهای مدل VGG16 را نشان می‌دهد توجه شود، آخرین لایه‌ی پیچشی، یعنی لایه‌ی شماره‌ی ۱۳ که قبل از لایه‌ی Dense1 قرار دارد، تاثیر به‌سزایی در تعداد اتصالات لایه‌ی طبقه‌بند دارد؛ و در لایه‌ی ۱۳، تعداد فیلترهای هرس شده‌ی روش OBD حدود نصف روش دیگر است که موجب کمتر بودن حجم کلی آن شده است.



نمودار ۲- نمودار فراوانی تعداد فیلترهای هرس شده در هر یک از ۱۳ لایه‌ی پیچشی با روش OBD و روش HSQGL12-OBD بر روی MNIST.

۴-۲ نتایج آزمایشات روی FashionMNIST

مراحل فوق‌الذکر بر روی مجموعه دادگان FashionMNIST نیز اجرا شد. شکل ۷ نمودارهای روند کاهش خطا داده‌های آموزشی و اعتبارسنجی بر روی داده‌های FashionMNIST را نشان می‌دهد.



شکل ۷: نمودارهای روند کاهش خطا داده‌های آموزشی و اعتبارسنجی بر روی داده‌های FashionMNIST

مدل تولیدی در اپک چهارم به عنوان مدل پایه (VGG) انتخاب و با روش‌های مختلف هرس گردید. ماتریس درهم‌ریختگی و سایر نتایج در برنامه‌ی مربوطه قابل مشاهده است که خلاصه‌ی آنها در جدول ۷ آمده است.

جدول ۷: نتایج ارزیابی روش‌های مختلف هرس بر روی مجموعه دادگان FashionMNIST

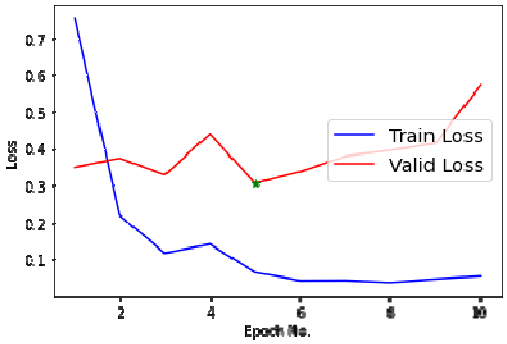
Sparsity		Size	Macro					
ZPR	PPR	MB	TPR	TNR	FPR	FNR	ACC	Method
0.013	-	525	0.88	0.99	0.01	0.12	0.89	VGG
0.115	0	525	0.88	0.99	0.01	0.12	0.89	HSQGL12
0.568	**0.563**	**229**	0.9	0.99	0.01	0.1	0.9	OBD
0.384	0.378	326	0.89	0.99	0.01	0.11	0.89	HSQGL12-OBD
0.577	**0.563**	**229**	**0.91**	0.99	0.01	**0.09**	**0.91**	OBD-HSQGL12

همان‌گونه که مشاهده می‌شود، شیوه‌ی ترکیبی آخر، نتایجی بهتر یا مساوی با هر یک از روش‌های هرس و روش اول ترکیبی داشته است. نمودارهای فراوانی تعداد فیلترهای هرس شده در فایل برنامه موجود است که از آنجا که حجم فایلها که در جدول ۷ ذکر شده، مرتبط با این نمودارها هست از درج آنها خودداری شده است.

۴-۳ نتایج آزمایشات روی STL10

شکل ۸ نمودارهای روند کاهش خطا داده‌های آموزشی و اعتبارسنجی بر روی داده‌های STL10 را نشان می‌دهد. مدل اپک پنجم به عنوان مدل پایه (VGG) انتخاب و با روش‌های مختلف هرس گردید. ماتریس درهم‌ریختگی و سایر نتایج در برنامه‌ی مربوطه قابل مشاهده است که خلاصه‌ی آنها در جدول ۸ آمده است.

[1] Pruned Parameters Rate
[2] Zeros Parameters Rate



شکل ۸: نمودارهای روند کاهش خطا داده‌های آموزشی و اعتبارسنجی بر روی داده‌های STL10

جدول ۸: نتایج ارزیابی روش‌های مختلف هرس بر روی مجموعه دادگان STL10

Sparsity		Size	Macro					
ZPR	PPR	MB	TPR	TNR	FPR	FNR	ACC	Method
0.013	-	525	**0.92**	**0.99**	**0.01**	**0.08**	**0.92**	VGG
0.105	0	525	0.83	0.98	0.02	0.17	0.84	HSQGL12
0.642	0.638	190	0.81	0.98	0.02	0.19	0.81	OBD
0.648	**0.644**	**187**	0.8	0.98	0.02	0.2	0.8	HSQGL12-OBD
0.648	0.638	190	0.82	0.98	0.02	0.18	0.82	OBD-HSQGL12

برخلاف مجموعه دادگان قبلی که مدل‌های حاصل از هرس، دقتی به خوبی مدل اولیه داشتند، در اینجا دقت مدل‌های هرس شده حدود ۱۰ درصد کمتر از مدل اولیه است. البته اگر معیار TPR مدنظر باشد، تقریباً با همان کارایی مدل‌های هرس شده، این معیار را برآورده کرده‌اند.

از نظر سه معیار آخر مرتبط با حجم و میزان تُنُکی مدل هرس شده، مدل HSQGL12-OBD بیشترین فشردگی را داشته است.

۵ جمع‌بندی و نتیجه‌گیری

شبکه‌های عصبی پیچشی از ابزار اصلی حوزه‌ی یادگیری عمیق هستند که کاربردها و تحقیقات مرتبط بسیاری را به خود معطوف نموده است. روش انتقال یادگیری از جمله مرسوم‌ترین روش‌های بکار رفته در پژوهش‌های اخیر در این حوزه بوده است. در این روش، یک مدل از قبل آموزش‌دیده بر روی تصاویر آموزشی مربوطه، مورد آموزش مجدد قرار می‌گیرد. بیشتر مدل‌های شبکه‌های عصبی پیچشی که برای طبقه‌بندی تصاویر در انتقال یادگیری بکار برده می‌شوند، مانند VGG16، مدل‌های حجیمی هستند؛ کاهش حجم این مدل‌ها می‌تواند باعث شود که نرم‌افزار شناسایی، در سخت‌افزارهای با حجم حافظه کم و قدرت پردازشی پایین هم قابلیت اجرا داشته باشد. به علاوه در راستای حفظ حریم خصوصی افراد، حجم کم و قابل دانلود سریع چنین مدل‌هایی، امکان استفاده از آنها در بستر وب و اجرا روی دستگاه کاربر را فراهم می‌کند.

در این پژوهش ترکیب دو روش منظم‌سازی و آسیب مغزی بهینه برای کاهش تعداد اتصالات یک مدل از شبکه‌های عصبی پیچشی بکار گرفته شد. نتایج آزمایشات انجام شده بر روی چهار مجموعه دادگان متفاوت، نشان داد که در سه مجموعه داده، بدون افت کارایی شبکه، می‌توان بخش زیادی از اتصالات مدل پایه را با ترکیب این دو روش کم کرد. این کاهش حجم، زمان پردازشی مورد نیاز را نیز کاهش می‌دهد. البته برای یک مجموعه دادگان، کاهش حدود ۶۴ درصدی حجم مدل باعث کاهش حدود ده درصدی دقت مدل گردید که نشان دهنده‌ی آن است که شیوه‌ی ترکیبی وابسته به مجموعه دادگان مورد استفاده است.

در هر کاربردی، با دردست داشتن مدل آموزش دیده‌ی VGG، می‌توان مطابق این نوشتار تاثیر ترکیب دو روش هرس را بررسی و روش بهینه را برای عملیاتی کردن مدل حاصل برگزید. اگر کاهش حجم در قبال کاهش دقت احتمالی، برای کاربرد مدنظر قابل قبول بود، مدل هرس شده می‌تواند عملیاتی گردد.

کوچک‌تر شدن مدل هرس شده، با رعایت حداقل کارایی مدنظر، به این معنی است که برای کاربرد مربوطه، شبکه‌های عصبی پیچشی کوچک‌تر نیز کفایت خواهند کرد و احتمالاً بتوان به جای انتقال یادگیری بر روی مدلی حجیم، یک معماری کوچک‌تر شبکه‌های عصبی پیچشی، برگرفته از ساختار مدل هرس شده پی‌ریزی و از ابتدا به ساکن مورد آموزش قرار داد. در صورت استفاده از مدل‌های کوچک‌تر، حجم حافظه‌ی مورد نیاز برای مدل، زمان آموزش و زمان استنتاج توسط مدل حاصله بسیار کاهش پیدا کرده و می‌تواند در سخت‌افزارهای ضعیف و در بستر وب نیز مورد استفاده قرار گرفته و ابزار مربوطه در دسترس‌پذیرتر شود.

مراجع

[۱] جم پور، م. و م. جاویدی، "**یک معماری شبکه عصبی عمیق مشترک با ویژگی‌های صریح برای بازشناسی امضاء**". مجله ماشین بینایی و پردازش تصویر, ۲۰۲۱. ۷(۲):صص 69-57.

[۲] امین‌طوسی، م.، "**کاربرد بسط تیلور در کاهش حجم شبکه‌های عصبی پیچشی برای طبقه‌بندی نقاشی‌های سبک امپرسیونیسم و مینیاتور**". نشریه ریاضی و جامعه, ۱۳۹۹. ۵(۱): صص 16-1.

[۳] رمضانی، س. و ر. حسن زاده، "**تشخیص خرابی در قطعات فلزی از طریق تصاویر C-scan حاصل از حسگر AMR با استفاده از روش مبتنی بر یادگیری عمیق**". مجله ماشین بینایی و پردازش تصویر, ۲۰۲۱. ۷(۲): صص24-13 .

[۴] محمدی، م.، "**شناسایی شماره پلاک خودرو بر اساس یادگیری عمیق با نظارت ضعیف**". مجله ماشین بینایی و پردازش تصویر, ۲۰۲۱. ۷(۲): صص34-25 .

[۵] شریفی، ع.ا.، ح. زارع، و ج. حاتمی، "**تأثیر توانبخشی شناختی رایانه‌ای بر عملکرد حافظه‌ی فعال بیماران مبتلا به آسیب مغزی**

تروماتیک". تازه های علوم شناختی, ۱۳۹۴. ۱۷ (۴): صص 71-78.

[6] Kandel, I. and M. Castelli, *How Deeply to Fine-Tune a Convolutional Neural Network: A Case Study Using a Histopathology Dataset.* Applied Sciences, 2020. 10(10): p. 3359.

[7] Richard, N.M., et al., *External modulation of the sustained attention network in traumatic brain injury.* Neuropsychology, 2018. 32(5): p. 541-553.

[8] Soberg, H.L., et al., *Health-related quality of life 12 months after severe traumatic brain injury: a prospective nationwide cohort study.* J Rehabil Med, 2013. 45(8): p. 785-91.

[9] Fischer-Baum, S. and G. Campana, *Neuroplasticity and the logic of cognitive neuropsychology.* Cognitive neuropsychology, 2017. 34(7-8): p. 403-411.

[10] LeCun, Y., J. Denker, and S. Solla. *Optimal Brain Damage*. 1990. Morgan-Kaufmann.

[11] Molchanov, P., et al. *Pruning Convolutional Neural Networks for Resource Efficient Inference*. 2017. OpenReview.net.

[12] Frisch, S., *How cognitive neuroscience could be more biological—and what it might learn from clinical neuropsychology.* Frontiers in Human Neuroscience, 2014. 8.(۵۴۱)

[13] Mogensen, J. and H. Malá, *Post-traumatic functional recovery and reorganization in animal models: a theoretical and methodological challenge.* Scand J Psychol, 2009. 50(6): p. 561-73.

[14] Mogensen, J., *Reorganization of the injured brain: implications for studies of the neural substrate of cognition.* Frontiers in psychology, 2011. 2: p. 7-7.

[15] Mitsuno, K., J. Miyao, and T. Kurita, *Hierarchical Group Sparse Regularization for Deep Convolutional Neural Networks*, in *International Joint Conference on Neural Networks (IJCNN 2020)*. 2020: Glasgow (UK).

[16] Simonyan, K. and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015.

[17] He, K., et al. *Deep Residual Learning for Image Recognition*. 2016.

[18] Krizhevsky, A., I. Sutskever, and G.E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, in *Advances in Neural Information Processing Systems 25*, F. Pereira, et al., Editors. 2012, Curran Associates, Inc. p. 1097-1105.

[19] Redmon, J. and A. Farhadi, *YOLO9000: Better, Faster, Stronger.* CVPR, 2017.

[20] Mohammadi, R., et al., *Transfer Learning-Based Automatic Detection of Coronavirus Disease 2019 (COVID-19) from Chest X-ray Images.* Journal of Biomedical Physics and Engineering, 2020. 10(5): p. 559-568.

[21] Torkian, P., et al., *Common CT Findings of Novel Coronavirus Disease 2019 (COVID-19): A Case Series* Cureus, 2020. 12(3): p. e7434.

[22] Ghotbi, B., et al., *A Review of the Novel Corona Virus Disease (2019-nCoV).* Health Research Journal, 2020. 5(3): p. 180-187 (In Persian).

[23] Chowdhury, M.E.H., et al., *Can AI Help in Screening Viral and COVID-19 Pneumonia?* IEEE Access, 2020. 8: p. 132665-132676.

[24] Che Azemin, M.Z., et al., *COVID-19 Deep Learning Prediction Model Using Publicly Available Radiologist-Adjudicated Chest X-Ray Images as Training Data: Preliminary Findings.* International Journal of Biomedical Imaging, 2020. 2020: p. 8828855.

[25] Ardakani, A.A., et al., *Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks.* Computers in biology and medicine, 2020. **121**: p. 103795.

[26] Makris, A., I. Kontopoulos, and K. Tserpes. *COVID-19 Detection from Chest X-Ray Images Using Deep Learning and Convolutional Neural Networks*. 2020. New York, NY, USA: Association for Computing Machinery.

[27] Apostolopoulos, I.D. and T.A. Mpesiana, *Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks.* Physical and Engineering Sciences in Medicine, 2020. 43(2): p. 635-640.

[28] Han, S., et al., *Learning both weights and connections for efficient neural networks*, in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. 2015, MIT Press: Montreal, Canada. p. 1135—1143.

[29] Anwar, S., K. Hwang, and W. Sung, *Structured Pruning of Deep Convolutional Neural Networks.* J. Emerg. Technol. Comput. Syst., 2017. 13(3): p. Article 32.

[30] Hinton, G.E., et al., *Improving neural networks by preventing co-adaptation of feature detectors.* CoRR, 2012.

[31] Wan, L., et al. *Regularization of Neural Networks using DropConnect*. 2013. Atlanta, Georgia, USA: PMLR.

[32] Wu, H. and X. Gu, *Towards dropout training for convolutional neural networks.* Neural Networks, 2015. 71: p. 1-10.

[33] Liu, B., et al. *Sparse Convolutional Neural Networks*. 2015. IEEE Computer Society.

[34] Wen, W., et al., *Learning Structured Sparsity in Deep Neural Networks*, in *Advances in Neural Information Processing Systems 29*, D.D. Lee, et al., Editors. 2016, Curran Associates, Inc. p. 2074-2082.

[35] Frickenstein, A., et al. *DSC: Dense-Sparse Convolution for Vectorized Inference of Convolutional Neural Networks*. in *2019 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition Workshops (CVPR W). 2019.

[36] Tibshirani, R., *Regression Shrinkage and Selection via the Lasso.* Journal of the Royal Statistical Society. Series B (Methodological), 1996. 58(1): p. 267-288.

[37] Yuan, M. and Y. Lin, *Model selection and estimation in regression with grouped variables.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2006. 68(1): p. 49-67.

[38] Alvarez, J.M. and M. Salzmann, *Learning the number of neurons in deep networks*, in *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2016, Curran Associates Inc.: Barcelona, Spain. p. 2270–2278.

[39] Rahman, T., et al., *Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images.* Computers in Biology and Medicine, 2021. 132: p. 104319.

[40] Kingma, D.P. and J. Ba, *Adam: A Method for Stochastic Optimization*, in *3rd International Conference for Learning Representations*. 2015: San Diego.

[41] Haghighi, S., et al., *PyCM: Multiclass confusion matrix library in Python.* Journal of Open Source Software, 2018. **3**(25): p. 729.

محمود امین‌طوسی، عضو هیات علمی گروه علوم کامپیوتر دانشگاه حکیم سبزواری است. وی دوره‌های کارشناسی و کارشناسی ارشد خود را در دانشگاه فردوسی به اتمام رسانده و دوره دکترای خود را در رشته مهندسی کامپیوتر (گرایش هوش مصنوعی) در دانشگاه علم و صنعت ایران گذرانده است. علائق پژوهشی وی یادگیری عمیق، بینایی ماشین و بهینه‌سازی ترکیبی می‌باشد.