دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

شبکه‌های عصبی دودویی

نگارش:

محمدرضا جفائی

استاد درس:

دکتر صفابخش

بهار ۱۴۰۱

چکیده

یادگیری عمیق (DL[1]) اخیراً توسعه سیستم‌های هوشمند را تغییر داده است و به طور گسترده در بسیاری از برنامه‌های کاربردی زندگی واقعی مورد استفاده قرار می‌گیرد. علیرغم مزایا و پتانسیل‌های مختلف این شبکه‌ها، نیازمندی این شبکه‌ها به منابع محاسباتی و ذخیره‌سازی بالا امکان پیاده‌سازی این روش‌ها را در دستگاه‌های مختلف محاسباتی با منابع محدود غیر-ممکن کرده است.

شبکه‌های عصبی دودویی (ش.ع.ب[2]) که تا حد زیادی در ذخیره‌سازی و محاسبات صرفه جویی می‌کنند، به عنوان یک تکنیک امیدوارکننده برای استقرار مدل‌های عمیق در دستگاه‌های با منابع محدود عمل می‌کنند. با این حال، دودویی شدن ناگزیر باعث از دست رفتن اطلاعات می‌شود و حتی بدتر از آن، ناپیوستگی آن همگرایی شبکه عمیق را با مشکل مواجه می‌کند. برای پرداختن به این مسائل، الگوریتم‌های مختلفی پیشنهاد شده‌اند و در سال‌های اخیر به پیشرفت‌های رضایت‌بخشی دست یافته‌اند.

در این گزارش ما در ابتدا به بررسی شبکه‌های دودویی، شیوه آموزش و مشکلات آن‌ها می‌پردازیم و پس از آن چندین روش بهبود شبکه‌های دودویی را بررسی می‌کنیم و در نهایت عملکرد روش‌های معرفی شده را با یکدیگر مقایسه می‌کنیم.

واژه‌های کلیدی:

شبکه عصبی دودویی، شبکه عصبی عمیق، هوش مصنوعی، دودویی سازی

[1] Deep Learning

[2] Binary Neural Networks

فهرست مطالب

فهرست اشکال

فهرست جداول

۱. مقدمه

با توسعه مداوم یادگیری عمیق[۱] ، شبکه‌های عصبی عمیق در زمینه‌های مختلف مانند بینایی کامپیوتر، پردازش زبان طبیعی و تشخیص گفتار پیشرفت چشمگیری داشته‌اند. ثابت شده است که شبکه‌های عصبی پیچشی(CNN[۳]) در زمینه‌های طبقه بندی تصویر [۲، ۳]، تشخیص اشیا [۴، ۵] و سایر فعالیت‌های مشابه قابل اعتماد هستند. بنابراین از این شبکه‌ها به طور گسترده در کاربردهای مختلف استفاده شده است.

به دلیل ساختار عمیق با تعداد زیادی لایه و میلیون‌ها پارامتر در شبکه‌های عصبی پیچشی عمیق، این شبکه‌ها ظرفیت یادگیری بالایی دارند و بنابراین معمولاً به عملکرد رضایت بخشی دست می‌یابند. به عنوان مثال، شبکه VGG-۱۶ [۶] حاوی حدود صد و چهل میلیون پارامتر ممیز شناور ۳۲ بیتی است و می‌تواند به دقت ۹۲.۷٪ برای طبقه بندی تصویر در مجموعه داده ImageNet[۴۳] دست یابد و کل شبکه نیاز به اشغال بیش از پانصد مگابایت فضای ذخیره سازی و انجام ۱.۶×۱۰^{۱۰} عملیات ممیز شناور دارد. این واقعیت باعث می‌شود که شبکه‌های عصبی پیچشی عمیق به شدت به سخت‌افزار با کارایی بالا مانند GPU[۴] متکی باشند، در حالی که در برنامه‌های کاربردی در واقعیت، معمولاً فقط دستگاه‌هایی (به عنوان مثال، تلفن‌های همراه و دستگاه‌های تعبیه‌شده) با منابع محاسباتی محدود در دسترس هستند[۷]. به عنوان مثال، دستگاه‌های تعبیه‌شده مبتنی بر FPGA[۵] معمولاً تنها چند هزار واحد محاسباتی دارند که با میلیون‌ها عملیات ممیز شناور در مدل‌های عمیق معمولی اختلاف زیادی دارد و تضاد شدیدی بین مدل پیچیده و منابع محاسباتی محدود وجود دارد. اگرچه در حال حاضر، تعداد زیادی سخت افزار اختصاصی برای یادگیری عمیق [۸، ۹] ساخته شده‌اند، که عملیات برداری کارآمدی را برای پیاده‌سازی پیچشی سریع در استنتاج رو به جلو ارائه می‌دهند، محاسبات و ذخیره سازی سنگین همچنان به ناچار کاربردها را محدود می‌کند.

بسیاری از مطالعات قبلی ثابت کرده‌اند که معمولاً افزونگی زیادی در ساختار شبکه‌های عمیق وجود دارد[۱۰، ۱۱]. به عنوان مثال، با کنار گذاشتن وزن‌های اضافی، می‌توان عملکرد ResNet-۵۰ را حفظ کرد و در عین حال بیش از ۷۵ درصد پارامترها و ۵۰ درصد زمان محاسباتی را ذخیره کرد[۱۲]. رویکردهای فشرده سازی شبکه‌های عمیق را می‌توان به پنج دسته هرس پارامتر[۱۳، ۱۴]، کمی‌سازی پارامتر[۱۵، ۱۶، ۱۷، ۱۸]، پارامترهای رتبه پایین[۱۹]، فاکتورسازی با پارامترهای رتبه پایین[۲۰]، فیلترهای پیچشی منتقل شده/فشرده شده[۲۱] و تقطیر دانش[۲۲] طبقه بندی کرد. هرس و کمی‌سازی پارامتر عمدتاً بر حذف افزونگی در پارامترهای مدل به ترتیب با حذف موارد اضافی یا فشرده سازی فضای پارامتر (به عنوان مثال، از وزن‌های ممیز شناور به عدد صحیح) تمرکز می‌کنند. فاکتورسازی با پارامترهای رتبه پایین از تکنیک‌های تجزیه ماتریس برای تخمین پارامترهای اطلاعاتی استفاده می‌کند. رویکردهای مبتنی بر فیلتر پیچشی فشرده به فیلترهای پیچشی با دقت طراحی شده برای کاهش پیچیدگی ذخیره سازی و محاسبات متکی هستند. روش‌های تقطیر دانش نیز سعی می‌کنند مدل فشرده‌تری را برای بازتولید خروجی یک شبکه بزرگ‌تر تولید کنند.

در میان تکنیک‌های فشرده‌سازی موجود، روش مبتنی بر کوانتیزاسیون به عنوان یک راه‌حل سریع بدین صورت عمل میکنه که با نمایش وزن شبکه با دقت بسیار پایینتر از مدل اصلی، مدل‌های بسیار فشرده را در مقایسه با وزن‌های ممیز

---

[۳] Convolutional Neural Network

[۴] Graphics Processing Unit

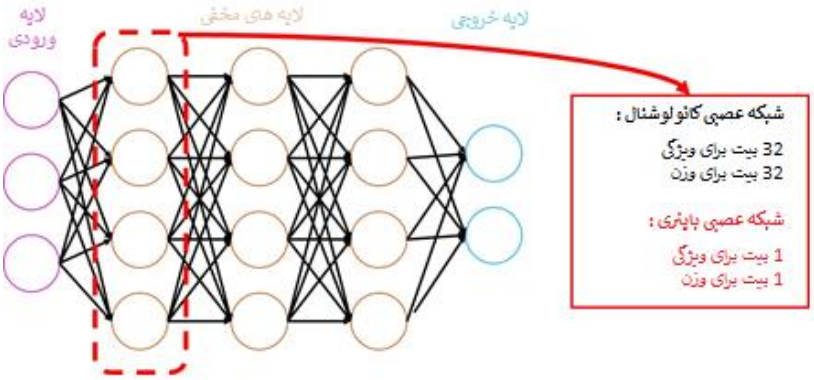
[۵] Field Programmable Gate Arrays

شناور به دست می‌آورد. در این روش، افراطی‌ترین کوانتیزه‌سازی دوتایی‌سازی است، دوتایی سازی یک کوانتیزاسیون ۱ بیتی است که در آن داده‌ها فقط می‌توانند دو مقدار ممکن داشته باشند، یعنی -۱(۰) یا +۱. برای فشرده‌سازی شبکه، هم وزن و هم تابع فعال‌سازی را می‌توان با ۱ بیت بدون نیاز به حافظه زیاد نشان داد .علاوه بر این، با دودویی کردن، عملیات ضرب ماتریس سنگین را می‌توان با عملیات XNOR بیتی و عملیات شمارش بیت جایگزین کرد. بنابراین، در مقایسه با سایر روش‌های فشرده‌سازی، شبکه‌های عصبی دودویی از تعدادی ویژگی سازگار با سخت‌افزار از جمله صرفه‌جویی در حافظه، بهره‌وری انرژی و شتاب قابل توجه برخوردار هستند. کار پیشگامانه ای مانند ش.ع.ب [۲۳] و XNOR-Net [۲۴] اثربخشی دودویی سازی، یعنی تا ۳۲ برابر صرفه جویی در حافظه و ۵۸ برابر سرعت در واحدهای پردازش را ثابت کرده است که توسط XNOR-Net برای لایه پیچشی یک بیتی به دست آمده است. به دلیل مزایای باورنکردنی ش.ع.ب با پارامترهای کمتر و سرعت استنتاج سریعتر می‌توان این شبکه‌ها را به راحتی در دستگاه‌های با منابع محدود مانند دستگاه‌های پوشیدنی اعمال و جاسازی کرد.. در سال‌های اخیر، با افزایش شبکه‌های سبک و کاربردی، توجه محققان بیشتری به ش.ع.ب معطوف شده و ش.ع.ب به یکی از موضوعات تحقیقاتی محبوب تبدیل شده است.

۲. شبکه‌های عصبی دودویی

۲-۱. ساختار شبکه‌های عصبی دودویی

ش.ع.ب نوعی شبکه عصبی است که توابع فعال‌سازی و وزن‌ها در تمام لایه‌های پنهان (به جز لایه‌های ورودی و خروجی) مقادیر ۱ بیتی هستند. می‌توان گفت ش.ع.ب یک مورد بسیار فشرده از شبکه پیچشی است. زیرا ش.ع.ب و شبکه پیچشی به جز فعال‌سازی‌ها و وزن‌های متفاوت ساختارهای یکسانی دارند. به فرآیند فشرده سازی مقادیر ۳۲ بیتی به ۱ بیتی، دودویی کردن می‌گوییم. با دودویی‌سازی نه تنها می‌توان ذخیره‌سازی مدل سنگین را ممکن کرد، بلکه هزینه‌های محاسباتی ماتریس را با استفاده از عملیات XNOR و شمارش بیت کاهش می‌دهد.



شکل ۱: مقایسه شبکه عصبی دودویی و شبکه پیچشی

رستگاری و همکاران[۲۴]، گزارش کردند که ش.ع.ب نسبت به شبکه پیچشی ۳۲ بیتی می‌تواند ۳۲ برابر حافظه ذخیره کمتر و ۵۸ برابر عملیات پیچشی سریعتر داشته باشد. در شبکه پیچشی، اکثر هزینه‌های محاسباتی صرف ضرب ماتریس در عملیات پیچشی می‌شود. عملیات پیچیدگی اصلی بدون بایاس را می‌توان به صورت زیر بیان کرد:

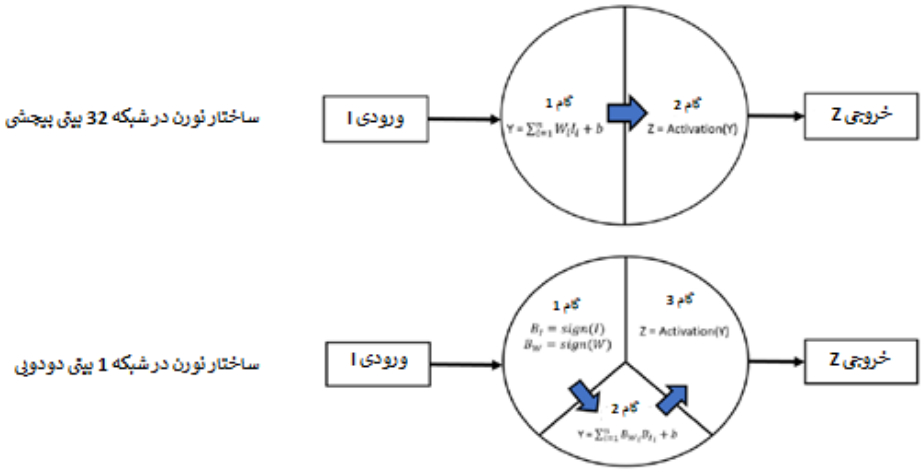
$$Z = I * W \qquad (۱)$$

جایی که I و W به ترتیب نشان دهنده ورودی و وزن هستند، Z خروجی عملیات پیچشی با ضرب ماتریس است. چنین عملیات ضربی، شامل تعداد زیادی عملیات ممیز شناور، از جمله ضرب ممیز شناور و جمع ممیز شناور است که دلیل عملکرد با سرعت پایین در استنتاج شبکه عصبی است.

یک شبکه عصبی از دو فرآیند آموزشی تشکیل شده است که شامل انتشار به جلو و انتشار به عقب است. انتشار رو به جلو فرآیند حرکت از لایه ورودی به لایه خروجی در شکل ۱ است که به استنتاج مدل نیز اشاره دارد. انتشار به عقب فرآیند حرکت از لایه خروجی به لایه ورودی در شکل ۱ است که روند تنظیم دقیق وزن‌های مدل را نشان می‌دهد. در بخش بعدی نحوه عملکرد ش.ع.ب در انتشار به جلو و انتشار به عقب بحث می‌شود.

۲-۲. قوانین یادگیری در شبکه‌های عصبی دودویی

در انشار رو به جلو متفاوت از شبکه پیچشی که از وزن و ورودی ۳۲ بیتی برای محاسبات استفاده می‌کند، سلول عصبی در مسیر رو به جلو ش.ع.ب، یک مرحله دودویی‌سازی به ورودی‌ها و وزن‌ها قبل از عملیات پیچشی به سلول اضافه می‌کند. هدف مرحله دودویی کردن، نمایش ورودی‌ها و وزن‌های ممیز شناور با استفاده از ۱ بیت است. شکل ۲ تفاوت در مراحل محاسباتی درون یک سلول عصبی را در امتداد مسیر رو به جلو بین ش.ع.ب ساده و شبکه پیچشی۳۲ بیتی را نشان می‌دهد.



شکل ۲: ساختار سلول عصبی در ش.ع.ب و شبکه پیچشی

در ش.ع.ب بیشتر از تابع علامت برای دودویی‌سازی استفاده می‌شود که به صورت زیر تعریف می‌شود.

$$\text{Sign(x)} = \begin{cases} +1 & \text{اگر } x \geq 0 \\ -1 & \text{درغیراینصورت} \end{cases} \quad (۲)$$

بعد از دودویی‌سازی وزن‌ها و ورودی‌ها به صورت زیر در می‌آیند.

$$I \approx Sign(I) = B_I \quad (۳)$$

$$W \approx Sign(W) = B_W \quad (۴)$$

که B_I و B_W به ترتیب ورودی دودویی شده و وزن دودویی شده هستند.

در شبکه‌های عصبی ، هسته پیچشی معمولاً به دو بخش دامنه و جهت تقسیم می‌شود، در حالی که نقشه‌های ویژگی فقط در جهت محاسبه کارآمد هستند. زیرا هنگام دودویی کردن وزن‌ها مقدار دامنه A توسط تابع علامت حذف می‌شود. بنابراین برای بهبود شبکه‌های دودویی می‌توان مقدار A را با $\hat{A}$ یک عدد ثابت است جایگزین نمود. بنابراین روش‌های دودویی‌سازی موجود را می‌توان در یک چارچوب یکپارچه فرمول‌بندی کرد که در این فرمول D جهت را نمایش میدهد و A دامنه را مشخص می‌کنند و به صورت زیر نشان می‌دهیم:

$$\hat{W} = D \odot \hat{A} \quad (۵)$$

که در آن $\odot$ ضرب عنصری بین ماتریس‌ها است. می‌توان خروجی پیچشی دودویی را به صورت زیر محاسبه کرد:

$$Z = (B_I \otimes B_W) \odot \hat{A} \quad (۶)$$

که در آن $\circledast$ نشان دهنده عملیات XNOR بیتی و عملیات شمارش بیت هستند.

بر همین اساس برای کاهش خطای کوانتیزاسیون در دودویی کردن یک شبکه عصبی عمیق، روش XNOR-Net [۲۴] دو عامل مقیاس‌بندی را به ترتیب برای وزن‌ها و ورودی‌*ها* معرفی می‌کند. در این گزارش برای سادگی، این دو فاکتور مقیاس‌بندی را به عنوان یک پارامتر ساده می‌کنیم که با α نمایش میدهیم. سپس، عملیات پیچشی دودویی را می‌توان به صورت فرموله کرد.

$$Z = (B_I \otimes B_W) \odot \alpha \quad (۷)$$

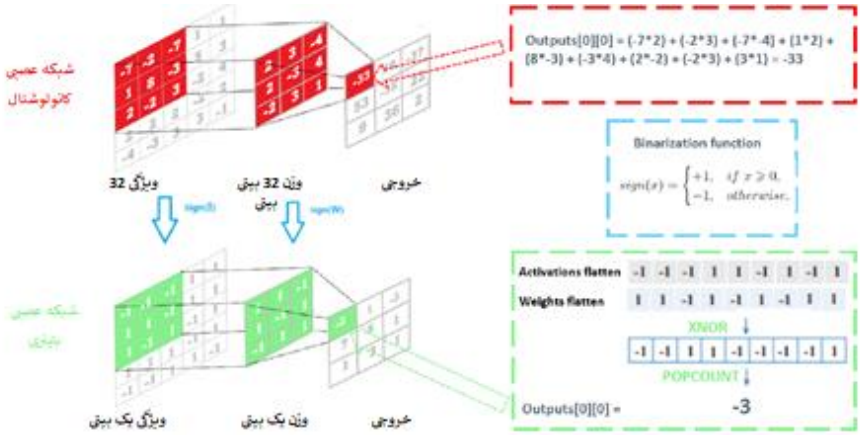
بنابراین خطای کوانتیزلسیون در ش.ع.ب به صورت زیر درمی‌آید

$$QE = \int_{-\infty}^{+\infty} f(w)\,(w - \alpha\, sign(w))^2\, dw \quad (۸)$$

که $f(w)$ تابع چگالی احتمال w است.

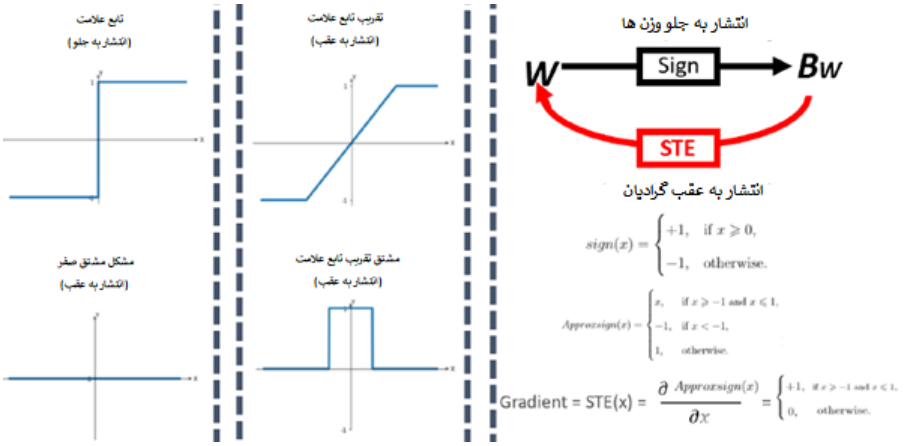
باتوجه به اینکه B_I و B_W تنها مقادیر ۱+ و ۱- دارند، می‌توانیم از XNOR بیتی و شمارش بیت برای جایگزینی محاسبه ضرب ماتریس استفاده کنیم.

شکل ۳ نمونه‌هایی از فرآیندهای محاسباتی پیچشی را در ش.ع.ب و شبکه پیچشی۳۲ بیتی ارائه می‌کند. با این کار با محاسباتی ش.ع.ب بسیار کاهش می‌یابد.



شکل ۳ : مقایسه محاسبات در شبکه پیچشی و ش.ع.ب

با استفاده از توضیحات بالا انتشار رو به جلو در ش.ع.ب به پایان میرسد و خروجی شبکه در لایه خروجی بدست می‌آید. حال توبت به انتشار خطا به عقب و بروزرسانی وزن‌ها می‌باشد. از آنجا که مشتق تابع علامت صفر است وزن‌های دودویی را نمی‌توان با روش نزول گرادیان بر اساس الگوریتم انتشار به عقب یاد گرفت. برای حل این مشکل، در شبکه‌های عصبی دودویی شده از تکنیکی به نام برآوردگر مستقیم (STE[۶]) استفاده می‌شود[۲۵] تا شبکه وزن‌های دودویی را در انتشار به عقب بیاموزد.



شکل ۴: انتشار خطا به عقب با استفاده از روش برآوردگر مستقیم

شکل ۴ فرآیند یادگیری وزن‌های دوتایی را در ش.ع.ب توضیح می‌دهد. در طول مراحل آموزش ش.ع.ب، وزن واقعی هر لایه با استفاده از برآوردگر مستقیم نگه داشته و به روز می‌شوند. پس از آموزش، وزن‌های دودویی ذخیره میشوند و وزن‌های واقعی دور ریخته می‌شوند. گرادیان تقریبی از فرمول زیر بدست می‌آید.

$$\frac{\partial \mathcal{L}}{\partial w^r} = \frac{\partial \mathcal{L}}{\partial w^b} \cdot \frac{\partial w^b}{\partial w^r} \approx \frac{\partial \mathcal{L}}{\partial w^b} \qquad (۹)$$

که در تابع بالا $\mathcal{L}$ تابع هزینه می‌باشد.

[۶] straight-through estimator

۳. بهبود شبکه‌های عصبی دودویی

اگرچه ش.ع.ب ساده دارای سرعت استنتاج سریعتر و وزن‌های تک بیتی است، دقت ش.ع.ب بسیار کمتر از دقت شبکه پیچشی است. دلیل آن از دست دادن شدید اطلاعات به دلیل دودویی شدن پارامترها، از جمله ورودی دودویی و وزن‌های دودویی است. دو دلیل اصلی برای کاهش عملکرد وجود دارد: (۱) خطای کوانتیزاسیون بزرگ در انتشار رو به جلو و (۲) عدم تطابق گرادیان در حین انتشار خطا به عقب این دو دلیل هستند. برای پرداختن به موضوع فوق، انواع راه حل-های بهینه‌سازی در سال‌های اخیر ارائه شده است. ما در این بخش به بررسی روش‌های مختلف بهبود شبکه‌های دودویی می‌پردازیم و چند مورد از روش‌های بهینه‌سازی را بررسی می‌کنیم.

۳-۱. روش‌های بهینه‌سازی شبکه‌های عصبی دودویی

برای مدل ش.ع.ب روش بهینه‌سازی و بهبود مختلفی وجود دارد. می‌توان این روش‌ها را به ۵ دسته تقسیم کرد. کمینه سازی خطای کوانتیزاسیون، بهبود تابع هزینه، تقریب گرادیان، تغییر ساختار توپولوژی شبکه و استراتژی و ترفندهای آموزشی روش‌های بهبود ش.ع.ب می‌باشند. همینطور روش کمینه سازی خطای کوانتیزاسیون را می‌توان به سه زیردسته تقسیم نمود. این سه زیردسته شامل ضریب مقیاس ، تابع کوانیزاسیون و توزیع وزن‌ها می‌باشند.

۳-۲. بهینه‌سازی مبتنی بر ضریب مقیاس

هسته‌های پیچشی معمولاً به دو بخش دامنه و جهت تقسیم می‌شود اما نقشه‌های ویژگی فقط در جهت محاسبه هسته ش.ع.ب کارآمد هستند. در برخی از کارهای پیشین [۲۴، ۳۴] تلاش‌های بسیاری در تعیین $\hat{A}$ شده است تا شکاف بین ش.ع.ب و شبکه پیچشی کاهش یابد. در بخش‌های بعدی براساس [۳۵] روشی تطبیقی جهت محاسبه $\hat{A}$ معرفی می‌شود.

۳-۲-۱. روش دامنه تطبیقی داده

روش‌های موجود در محاسبه دامنه تطبیقی داده‌ها برای تقریب بهتر نقشه‌های ویژگی با دقت کامل، شکست می‌خورند، که دلیل اصلی شکاف عملکرد ش.ع.ب و همتای با دقت کامل آن‌ها را توضیح می‌دهد. بدون در نظر گرفتن دامنه داده ورودی، یک شکاف اجتناب ناپذیر بین خروجی دو شبکه وجود دارد، زیرا $\hat{A}$ ثابت برای ورودی‌های مختلف بی ربط است. برای پرداختن به این موضوع، یک ایده شهودی این است که اجازه دهیم $\hat{A}$ به یک تابع $\hat{A}$ (X) با داده ورودی به عنوان ورودی تبدیل شود. در ش.ع.ب، ما از $B_I \otimes B_W$ برای جایگزینی X استفاده می‌کنیم، زیرا $B_I \otimes B_W$ حاوی اطلاعات هر دو داده‌های ورودی و وزن‌ها است که ظرفیت بازنمایی بهتری خواهد داشت. از آنجایی که دامنه A ثابت نیست اما با داده‌های ورودی تطبیق دارد، به این روش شبکه عصبی دودویی داده‌تطبیقی (DA_BNN[۷]) می‌گویند و داریم:

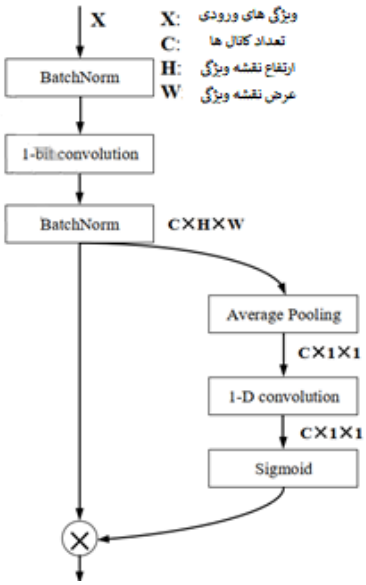
$$Z = (B_I \otimes B_W) \odot \hat{A}(B_I \otimes B_W) \quad (۱۰)$$

[۷] Data-adaptive Binary Neural Networks

که در آن $\hat{A}$ (.) به ورودی مربوط می‌شود و بار محاسباتی ش.ع.ب را اضافه می‌کند. برای رفع این مشکل، روش‌های مبتنی بر توجه [۳۶] استفاده شده است و یک ماژول سبک وزن را برای پیاده‌سازی $\hat{A}$ (.) معرفی شده است. در دو بخش بعدی ماژولی را با در نظر گرفتن هر دو سطح کانال و فضایی معرفی می‌کنیم.

۲-۲-۳. دامنه کانال سازگار با داده

برای سادگی در این دو بخش $(B_I \otimes B_W)$ را با $\widehat{M}$ نشان می‌دهیم. برای محاسبه دامنه کانال $\widehat{A}_c(\widehat{M})$ ، نقشه‌های ویژگی را از دو منظر، درون کانال‌ها و بین کانال‌ها، مشابه مکانیسم توجه در نظر می‌گیریم.



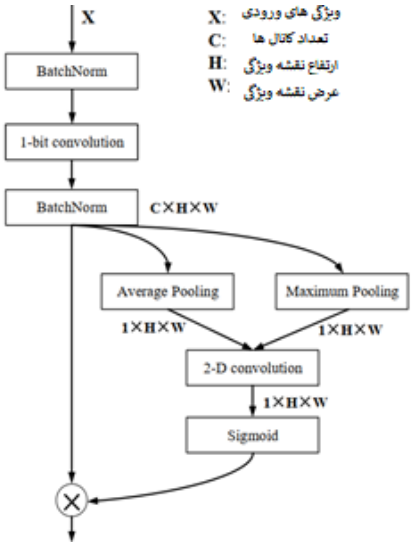
شکل ۵: محاسبات دامنه کانال

برای کاهش دو بعد دیگر و استخراج ویژگی‌های درون کانال‌ها، لایه Global Average Pooling را استفاده می‌کنیم. در مقایسه با لایه پیچشی، این لایه هیچ پارامتر اضافی و محاسباتی اضافه نمی‌کند. با در نظر گرفتن تعامل متقابل کانال، یک لایه پیچشی یک بعدی برای ترکیب اطلاعات هر کانال با همسایگانش اعمال می‌شود. با این حال، از آنجایی که پارامترهای پیچشی با ارزش واقعی اغلب نزدیک به صفر هستند و به راحتی تحت تأثیر کاهش وزن قرار می‌گیرند، دوتایی شدن پارامترها همیشه به معنای تقویت در مقایسه با پیچشی ارزش واقعی است. نتیجه پیچشی دودویی معمولاً در مقایسه با پیچیدگی ارزش واقعی بسیار بزرگتر است. بنابراین، دامنه $\widehat{A}_c(\widehat{M})$ باید یک مقدار کوچک باشد، که با یک تابع سیگموئید که دامنه را به (۰، ۱) ترسیم می‌کند، این مشکل حل می‌شود. علاوه بر این، تابع سیگموئید برای تضمین این که تابع ما صرفاً اطلاعات دامنه را یاد می‌گیرد، نه جهت را نیز استفاده می‌شود. با انجام این کار، تابع دامنه کانال را به صورت زیر نشان می‌دهیم:

$$\hat{A}_c(\widehat{M}) = \sigma(k_c * AvgPool(\widehat{M})) \quad (۱۱)$$

جایی که σ نشان دهنده تابع سیگموئید است و k_c نشان دهنده هسته پیچشی یک بعدی است.

۳-۲-۳. دامنه فضایی سازگار با داده

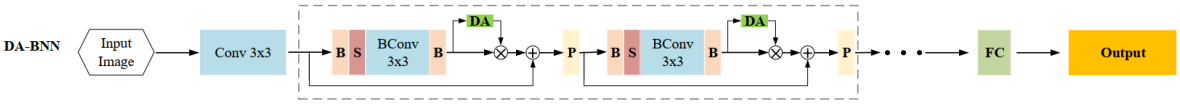


شکل ۶: محاسبات دامنه فضایی

مشابه محاسبه شبکه توجه در بخش قبلی، از لایه‌های ادغام و پیچشی برای محاسبه دامنه فضایی استفاده می‌کنیم. در شکل ۶، ساختار مربوطه را نشان می‌دهیم. تفاوت این دو شبکه در این است که ما از میانگین و حداکثر ادغام با هم استفاده می‌کنیم و سپس از یک پیچیدگی ۳×۳ به جای پیچیدگی ۱ بعدی برای داده‌های مکانی استفاده می‌کنیم. ما دامنه فضایی $\hat{A}_S(\hat{M})$ را به صورت زیر محاسبه می‌کنیم:

$$\hat{A}_S(\hat{M}) = \sigma(k_s * [AvgPool(\hat{M}); MaxPool(\hat{M})]) \quad (۱۲)$$

با این حال، هنگامی که ویژگی‌ها در بلوک بعدی دودویی می‌شوند، اطلاعات دامنه حذف می‌شوند و فقط اطلاعات جهت حفظ می‌شوند. برای حفظ اطلاعات دامنه، ویژگی‌ها را با استفاده از یک نرمال‌ساز اضافی که قبل از دوتایی شدن نقشه ویژگی اضافه شده است، دوباره توزیع می‌کنیم. با انجام این کار، با بهبود توزیع ویژگی، دامنه تا حدی به جهت در شکل تبدیل می‌شود. بنابراین شبکه نهایی به شکل زیر درمی‌آید.



شکل ۷: شبکه دامنه تطبیقی

با استفاده از تابع معرفی شده در بالا با افزایش ناچیز محاسبات $\hat{A}$ به یک تابع $\hat{A}$ (X) با داده ورودی به عنوان ورودی تبدیل می‌شود و با این کار شکاف بین ش.ع.ب و شبکه عصبی پیچشی کاهش می‌یابد که دربخش نتایج بهبود ایجاد شده را بررسی می‌کنیم.

۳-۳. بهینه‌سازی مبتنی بر تابع کوانتیزاسیون

در [۲۸] بررسی شد که وزن‌های نهفته، نقش مهمی در دودویی‌سازی شبکه‌های عصبی عمیق ایفا می‌کنند. هنگام بررسی وزن‌های با ارزش واقعی یک شبکه عصبی عمیق معین متوجه می‌شویم که وزن‌هایی که در دو انتهای توزیع قرار

می‌گیرند، به سختی در طول آموزش ش.ع.ب به‌روزرسانی می‌شوند که به آن‌ها وزن مرده گفته می‌شود و متوجه می‌شویم که آن‌ها به بهینه‌سازی آسیب می‌رسانند و همگرایی آموزشی ش.ع.ب را کاهش می‌دهند. برای حل این مشکل، یک واحد گیره اصلاح شده (ReCU[۸]) معرفی می‌شود که هدف آن احیای وزن‌های مرده با حرکت آن‌ها به سمت اوج توزیع به منظور افزایش احتمال به روز رسانی این وزن‌ها است. می‌توان نشان داد که خطای کوانتیزاسیون پس از اعمال واحد گیره اصلاح شده یک تابع محدب است و می‌توان نقطه بهینه سراسری را برای این مسئله بدست آورد. در بخش‌های بعدی به بررسی این روش میپردازیم.

۱-۳-۳. واحد گیره اصلاح شده (ReCU)

برای حل مشکل بالا، واحد گیره اصلاح شده پیشنهاد شده است که هدف آن انتقال وزن‌های مرده به سمت اوج توزیع برای افزایش احتمال تغییر علائم آن‌ها است. برای هر وزن با ارزش واقعی واحد گیره اصلاح شده به صورت زیر بدست می‌آید.

$$ReCU\ (w) = Max\ (\ Min\ (\ w, Q_{(\tau)}\ )\ , Q_{(1-\tau)}\ ) \qquad (۱۳)$$

که در فرمول بالا $Q_{(\tau)}$ نشان دهنده چندک τ و $Q_{(1-\tau)}$ نشان دهنده چندک $1-\tau$ وزن‌ها است. در صورتی که مقدار $0.5 < \tau \leq 1$ باشد واحد گیره اصلاح شده در صورتی که مقدار w از $Q_{(1-\tau)}$ کوچکتر باشد مقدار آنرا با $Q_{(1-\tau)}$ جایگزین می‌کند و در صورتی که مقدار w از $Q_{(\tau)}$ بزرگتر باشد آنرا با $Q_{(\tau)}$ جایگزین می‌کند. به این ترتیب وزن‌های مرده احیا می‌شوند. ثابت می‌شود که وزن‌ها پس از اعمال واحد گیره اصلاح شده می‌توانند خطای کوانتیزاسیون کوچک‌تری به دست آورند. کارهای قبلی [۲۹، ۳۰] نشان داده‌اند که وزن‌های نهفته تقریباً از توزیع لاپلاس با میانگین صفر پیروی می‌کنند، به عنوان مثال، $w \sim La(0, b)$، که به معنای $Q_{(\tau)} + Q_{(1-\tau)} = 0$ است. بنابراین ، داریم

$$\int_{-\infty}^{Q_{(\tau)}} \frac{1}{2b} \exp(-\frac{|w|}{b})\ dw = \tau \qquad (۱۴)$$

بنابراین می‌توان نتیجه گرفت که

$$Q_{(\tau)} = -b \ln(2-2\tau) \qquad (۱۵)$$

با این حال، تعیین مقدار دقیق b دشوار است. بجای استفاده از مقدار دقیق b، می‌توانیم تقریب آن را از طریق تخمین حداکثر درست‌نمایی(MLE[۹]) به‌دست آوریم که به صورت زیر فرموله می‌شود.

$$\tilde{b} = Mean(|W|) \qquad (۱۶)$$

بنابراین $Q_{(\tau)}$ تابعی از τ است. پس از اعمال واحد گیره اصلاح شده به w، تابع چگالی احتمال تعمیم یافته w را می‌توان به صورت زیر نوشت.

$$f(w) = \begin{cases} \frac{1}{2b} \exp\left(\frac{-|w|}{b}\right), & \text{اگر } |w| < Q_{(\tau)} \\ 1-\tau, & \text{اگر } |w| = Q_{(\tau)} \\ 0, & \text{درغیراینصورت} \end{cases} \qquad (۱۷)$$

[۸] Rectified Clamp Unit

[۹] Maximum likelihood estimation

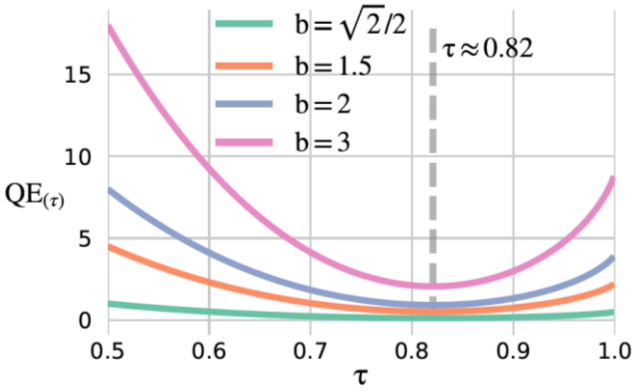
برای به دست آوردن خطای کوانتیزاسیون، ابتدا ضریب مقیاس را محاسبه می‌کنیم.

$$\begin{aligned}\alpha &= \mathbb{E}(|\mathrm{ReCU}\,(\mathcal{W})|)\\ &= \int_{-Q_{(\tau)}}^{Q_{(\tau)}} |w| f(w)\mathrm{d}w + \sum_{|w|=Q_{(\tau)}} |w| f(w)\\ &= \int_{0}^{Q_{(\tau)}} \frac{w}{b}\exp\left(-\frac{w}{b}\right)\mathrm{d}w + 2Q_{(\tau)}(1-\tau)\\ &= b - \left(Q_{(\tau)}+b\right)\exp\left(-\frac{Q_{(\tau)}}{b}\right) + 2Q_{(\tau)}(1-\tau)\end{aligned} \qquad (۱۸)$$

با جایگذاری مقدار تخمین زده شده b در فرمول بالا می‌توان دریافت که α تابعی از τ است. همانطور که میدانیم وزن‌ها از توزیع لاپلاس پیروی می‌کنند و $Q_{(\tau)} + Q_{(1-\tau)} = 0$ بنابراین با جایگذاری مقادیر بدست آمده در فرمول ۶ فرمول زیر بدست می‌آید.

$$\begin{aligned}\mathrm{QE}(\tau) &= \int_{-\infty}^{+\infty} f(w)(w-\alpha\mathrm{sign}\,(w))^2\mathrm{d}w\\ &= \int_{-Q_{(\tau)}}^{Q_{(\tau)}} f(w)(w-\alpha\mathrm{sign}\,(w))^2\mathrm{d}w\\ &\quad + \sum_{|w|=Q_{(\tau)}} f(w)(w-\alpha\mathrm{sign}\,(w))^2\\ &= \int_{0}^{Q_{(\tau)}} \frac{1}{b}\exp\left(-\frac{w}{b}\right)(w-\alpha)^2\mathrm{d}w\\ &\quad + \sum_{|w|=Q_{(\tau)}} (1-\tau)(w-\alpha\mathrm{sign}\,(w))^2\\ &= (\alpha-b)^2\left(1+\exp\left(-\frac{Q_{(\tau)}}{b}\right)\right) + b^2\\ &\quad - \left(\left(b+Q_{(\tau)}\right)^2 - 2\alpha Q_{(\tau)}\right)\exp\left(-\frac{Q_{(\tau)}}{b}\right)\\ &\quad +2(1-\tau)\left(Q_{(\tau)}-\alpha\right)^2.\end{aligned} \qquad (۱۹)$$

طبق معادله بالا ما دو مشاهده داریم: اولین مشاهده این است که همانطور که در شکل زیر نشان داده شده است، $QE_{(\tau)}$ وقتی $0.5 \leq \tau \leq 1$ باشد یک تابع محدب است و به ازای $\tau \approx 0.82$ به نقطه کمینه سراسری می‌رسد.



شکل ۸: نمودار خطای کوانتیزاسیون پس از اعمال واحد گیره اصلاح شده

دومین مشاهده این است که هنگامی که $\tau = 1$ است معادله به خطای کوانتیزاسیون نرمال تبدیل می‌شود. با این حال، ما نمی‌توانیم برای دنبال کردن کمترین خطای کوانتیزاسیون $\tau \approx 0.82$ قرار دهیم. در بخش بعدی، آنتروپی اطلاعات را تجزیه و تحلیل می‌کنیم و تضاد بین حداقل کردن خطای کوانتیزاسیون و به حداکثر رساندن آنتروپی اطلاعات را آشکار می‌کنیم. به طور کلی، نابرابری زیر را داریم.

$$QE(\tau) \leq QE(1) = QE, \qquad 0.82 \leq \tau \leq 1 \tag{۲۰}$$

یعنی در بازه ذکر شده واحد گیره اصلاح شده، خطای کوانتیزاسیون کوچکتری نسبت به مسئله اصلی ارائه می‌دهد.

۲-۳-۳. آنتروپی اطلاعات وزن‌ها

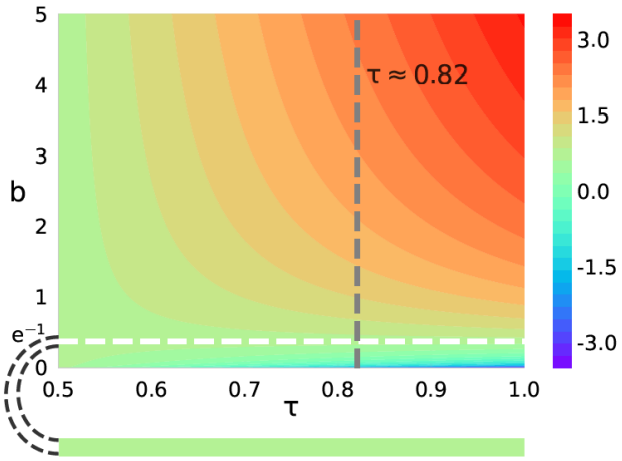
آنتروپی اطلاعات یک متغیر تصادفی است که سطح متوسط عدم قطعیت در نتایج احتمالی متغیر را نشان می‌دهد. آنتروپی اطلاعات به عنوان یک معیار کمی برای اندازه‌گیری تنوع وزن در ش.ع.ب استفاده می‌شود [۳۱، ۳۲]. معمولاً هرچه تنوع وزن‌ها بیشتر باشد، عملکرد ش.ع.ب بهتر است. با توجه به تابع چگالی احتمال $p(x)$ در حوزه X، آنتروپی اطلاعات به صورت تعریف می‌شود.

$$H(p) = \mathbb{E}\big(-\ln\big(p(x)\big)\big) = -\int_X p(x)\ln\big(p(x)\big)dx \tag{۲۱}$$

بنابراین آنتروپی اطلاعات W پس از اعمال واحد گیره اصلاح شده را می‌توان به صورت زیر محاسبه نمود.

$$\begin{aligned} H(f) &= -\int_{-Q(\tau)}^{Q(\tau)} f(w)\ln\left(f(w)\right)\mathrm{d}w \\ &\quad -\sum_{|w|=Q(\tau)} f(w)\ln\left(f(w)\right) \\ &= -\int_0^{Q(\tau)} \frac{1}{b}\exp\left(-\frac{w}{b}\right)\ln\left(-\frac{1}{2b}\exp\left(-\frac{w}{b}\right)\right)\mathrm{d}w \\ &\quad -2(1-\tau)\ln(1-\tau) \\ &= 2(\ln b + 1)\tau + \ln\frac{2}{b} - 1 \end{aligned} \tag{۲۲}$$

برای تحلیل فرمول نهایی بدست آمده ما سه حالت را در نظر میگیریم: حالت اول $b = e^{-1}$: در این حالت آنتروپی اطلاعات فرمول بالا برابر $\ln 2$ می‌شود و دیگر وابسته به τ نمی‌باشد. حالت دوم $b < e^{-1}$: دراین حالت آنتروپی اطلاعات یک تابع کاهشی یکنواخت از τ می‌شود. حالت سوم $b > e^{-1}$: دراین حالت آنتروپی اطلاعات یک تابع افزایشی یکنواخت از τ می‌شود.



شکل ۹: آنتروپی اطلاعات برای W

در بخش‌های قبلی مقدار b با میانگین مقادیر مطلق W تخمین زده شد. در آزمایشات به طور تجربی مشاهده شده است که در مواردی که $b \leq e^{-1}$ است آنتروپی اطلاعات برای فعال کردن عملکرد خوب بسیار کوچک است. بنابراین، یک b بزرگتر باید برای غلبه بر این مشکل استفاده شود. با این حال، در عمل، وزن‌های W به‌تدریج در طول آموزش شبکه به دلیل منظم‌سازی پراکنده می‌شوند و آنتروپی اطلاعات را غیرقابل کنترل می‌کند، که به ناچار تنوع را از دست می‌دهد. بنابراین، لازم است b را در یک مقدار نسبتاً بالا به روشی قابل کنترل برای حفظ آنتروپی اطلاعات حفظ کنیم. کار قبلی [۳۳] آنتروپی اطلاعات را با متمرکز کردن و استاندارد کردن وزن‌ها در هر انتشار رو به جلو به شرح زیر به حداکثر می‌رساند:

$$W' = \frac{W - \mathbb{E}(W)}{\sigma(W)} \quad (۲۳)$$

که در آن $\sigma(\ )$ نشان دهنده انحراف معیار است. با این حال، با آزمون و خطا می‌توان دریافت که این استانداردسازی است، که به بهبود عملکرد کمک می‌کند نه به مرکز بردن داده‌ها. بنابراین می‌توان معادله بالا را به سادگی تعمیم داد به معادله زیر:

$$W' = \frac{W}{K} \quad (۲۴)$$

که در آن K یک مقدار ثابت بزرگتر از صفر است. براساس فرمول بالا مقدار تخمینی b از فرمول زیر بدست می‌آید.

$$b' = Mean(|W'|) = \frac{b}{K} \quad (۲۵)$$

می‌توان مشاهده نمود که بخاطر توزیع لاپلاس وزن‌ها $\sigma(W) = \sqrt{2}b$ می‌باشد. بنابراین می‌توان $K = \sigma(W)$ قرار داد و بنابراین مقدار b تخمینی از فرمول زیر بدست می‌آید.

$$b' = \frac{b}{\sqrt{2}b} = \frac{\sqrt{2}}{2} > e^{-1} \quad (۲۶)$$

که آنتروپی اطلاعات را افزایش می‌دهد و توضیح می‌دهد که چرا تقسیم W بر انحراف استاندارد می‌تواند منجر به عملکرد بهتر در هنگام آموزش ش.ع.ب شود [۴۱]. با این وجود، مطابق شکل ۹، آنتروپی اطلاعات را می‌توان با b بزرگتر افزایش داد. بنابراین، از معادله زیر می‌توان وزن‌های جدید را بدست آورد:

$$W' = \frac{W}{\frac{\sigma(W)}{(\sqrt{2}b^*)}} \quad (۲۷)$$

که به راحتی می‌توان دریافت که با این جایگذاری مقدار $b^* = b'$ برابر است نوآوری پشت این تجزیه و تحلیل در این است که استانداردسازی ما با تنظیم دستی b^* بر اساس فرض $b^* > e^{-1}$ ، آنتروپی اطلاعات کنترل نشده را به یک آنتروپی قابل تنظیم تبدیل می‌کند، و بنابراین بهره اطلاعات معادله را تعمیم می‌دهد.

بنابراین، با استاندارد کردن وزن‌ها قبل از اعمال واحد گیره اصلاح شده، آنتروپی اطلاعات را می‌توان در یادگیری یک ش.ع.ب افزایش داد. با این وجود، افزایش اطلاعات از بزرگنمایی b هنوز بسیار محدود است. در مقابل، افزایش τ منجر به کسب اطلاعات بیشتر، با افزایش غیرمنتظره در خطای کوانتیزاسیون زمانی که ۰.۸۲ < τ می‌شود. بنابراین، یک تناقض ذاتی بین به حداقل رساندن خطای کوانتیزاسیون و به حداکثر رساندن آنتروپی اطلاعات در ش.ع.ب وجود دارد و باید با آزمون خطا یک مقدار متعادل بر اساس مسئله بدست آورد.

علیرغم عملکرد خوب شبکه هنگام استفاده از مقدار ثابت τ ، متوجه می‌شویم که واحد گیره اصلاح شده واریانس عملکرد را زمانی که $0.85 \leq \tau \leq 0.94$ است افزایش می‌دهد در حالی که آن را زمانی که $0.94 \leq \tau \leq 1$ ثابت نگه می‌دارد. برای حل این مشکل، یک زمان‌بندی نمایی را برای تطبیق τ در طول آموزش شبکه پیشنهاد می‌شود. انگیزه در این است که τ باید با مقداری در بازه [۰.۸۵، ۰.۹۴] شروع شود تا دقت خوبی را دنبال کند، و سپس به تدریج به بازه [۰.۹۶، ۱.۰۰] برای تثبیت واریانس عملکرد برود. بر این اساس، با توجه به τ_s اولیه و یک آستانه پایان τ_e ، τ_iدر دوره آموزش i ام به صورت زیر محاسبه می‌شود که در آن I تعداد کل دوره‌های آموزش شبکه می‌باشد.

$$\tau_i = \frac{\tau_e - \tau_s}{e-1} e^{i/I} + \frac{e \cdot \tau_s - \tau_e}{e-1} \quad (۲۸)$$

۴-۳. بهینه‌سازی مبتنی بر تابع هزینه:

همانطور که در بخش دوم بحث شد روش مبتنی بر گرادیان آموزش ش.ع.ب، از برآوردگر مستقیم برای مقابله با مشکل مشتق نداشتن تابع علامت در آموزش دودویی‌سازی استفاده می‌کنند و تقریباً در تمام کارهای بعدی به طور گسترده مورد استفاده قرار گرفته است. صرف نظر از تفاوت آن‌ها، یک متغیر کمکی با ارزش واقعی W، که به عنوان وزن نهفته نیز شناخته می‌شود، معمولاً برای کمک به آموزش متغیر دودویی در چارچوب مبتنی بر برآوردگر مستقیم استفاده می‌شود. از طریق تجزیه و تحلیل کل فرآیند تمرین، ما دو نقش برای وزن نهفته W را در ش.ع.ب داریم: (۱) در طول انتشار رو به جلو، W در بدست آوردن متغیر وزن دودویی و ضریب مقیاس‌بندی استفاده می‌شود.(۲) در طول انتشار به عقب، W مقدار خود را با گرادیان تقریبی به دست آمده از STE به روز می‌کند و برای تکرار بعدی انتشار به جلو آماده می‌شود.

از آنجایی که B_W همتای دودویی W با ارزش واقعی است، شهودی است که دقت یک مدل با W باید بیشتر از B باشد. با این حال، ارزیابی واقعی نشان می‌دهد که اینطور نیست. علت این پدیده را می‌توان به بایاس در آمارهای حفظ شده توسط نرمال سازی دسته‌ای (BN[۱۰]) در ش.ع.ب نسبت داد. برای کاهش از دست دادن اطلاعات دودویی‌سازی، یک لایه نرمال

[۱۰] Batch Normalization

سازی دسته‌ای معمولاً قبل از علامت تابع استفاده می‌شود، که میانگین ارزش ویژگی‌ها را به صفر می‌رساند. بنابراین اگر از W به جای B_W برای انجام پیچش استفاده کنیم اما همچنان از آمار به ارث رسیده از نرمال سازی دسته‌ای قدیمی استفاده کنیم، دقت کاهش می‌یابد.

این پدیده کاهش دقت را می‌توان به سادگی با محاسبه مجدد آمار لایه نرمال سازی دسته‌ای کاهش داد. علاوه براین با انگیزه این واقعیت که گرادیان تابع علامت در پس انتشار با HardTanh() تقریب می‌شود، پیشنهاد می‌شود هنگام پیش استنتاج با W ، فعال سازی دودویی را با HardTanh() جایگزین شود. بنابراین W می‌تواند دقت ارزیابی را برای مقایسه با B_W، که با تحلیل فوق مطابقت دارد، بازیابی کند. اگرچه هنوز فاصله کمی بین ارزیابی با B و W وجود دارد.

از تجزیه و تحلیل بالا، ما می‌دانیم که وزن نهفته W به طور مستقیم با نقشه‌های ویژگی در چارچوب آموزش ش.ع.ب، پیچشی را انجام نمی‌دهد، و از قابلیت آن به عنوان یک استخراج کننده ویژگی با ارزش واقعی به طور درست استفاده نمی‌شود. در بخش بعدی روشی ارائه می‌شود که از W با ارزش واقعی برای بهبود آموزش دودویی استفاده شود و W را به نمودار محاسباتی اضافه کند، که می‌تواند جزئیات بیشتری را نسبت به ویژگی‌های دودویی استخراج شده توسط B_W معرفی کند.

۱-۴-۳. استنباط با وزن‌های نهفته

در بخش قبلی بررسی کردیم که محاسبه مجدد آمار نرمال سازی دسته‌ای و جایگزینی تابع علامت با HardTanh تاثیر قابل توجهی بر عملکرد W دارد. بجای محاسبه مجدد آمارگان بعد از آموزش ما در طول آموزش از دو مجموعه μ ، σ^2 برای W و B_W استفاده می‌کنیم تا آمار لایه‌ای را از طریق معادلات زیر ثبت کنیم.

$$\mu = \frac{1}{N}\sum_{i=1}^{n} F^i \qquad (۲۹)$$

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{n} (F^i - \mu)^2 \qquad (۳۰)$$

با داشتن دو مجموعه μ ، σ^2 برای W و B_W خروجی‌ها را می‌توان زیر بدست آورد. ما از Y و $\hat{Y}$ برای نشان دادن ویژگی-هایی که به ترتیب توسط B و W بدست می‌آیند استفاده می‌کنیم. برای شبکه‌ای با L لایه، ویژگی Y_l، $\tilde{Y}_l$ لایه l با استفاده معادله ای به شرح زیر محاسبه می‌شود:

$$\begin{cases} Y_l = \Psi\left(BN_{\mathrm{B}}\left(\alpha B \circledast \operatorname{sign}\left(Y_{l-1}\right)\right)\right) \\ \tilde{Y}_l = \Psi\left(BN_{\mathrm{W}}\left(W * \mathit{h}\mathrm{ard}\ \tanh\left(\tilde{Y}_{l-1}\right)\right)\right) \end{cases} \qquad (۳۱)$$

که در آن Ψ تابع فعال سازی غیرخطی را نشان می‌دهد. BN_{W}و BN_{B} ضرایب وابسته قابل یادگیری یکسانی دارند اما آمار در حال اجرا مربوطه خود را به صورت جداگانه حفظ می‌کنند.

همانطور که می‌دانیم اگر دقت W بهبود یابد، دقت همتای دودویی آن نیز بهبود می‌یابد زیرا وزن دودویی با تقریب وزن از W بدست می‌آید. ساده‌ترین ایده، نظارت مستقیم بر W با کاهش آنتروپی متقابل در پیش‌بینی خروجی آخرین لایه $\tilde{Y}_L$ است. اما متوجه می‌شویم که در عمل این تابع نمی‌تواند همگرا شود.. در بخش بعدی سعی می‌کنیم این مشکل را بر طرف کنیم.

۲-۴-۳. تقریب آگاه از برچسب

در بخش قبلی وزن نهفته را به نمودار محاسباتی اضافه کردیم و دو ویژگی مختلف Y و $\hat{Y}$ بدست می‌آوریم. هر دو ویژگی با معماری شبکه یکسان استخراج می‌شوند، تفاوت این است که یکی توسط وزن با ارزش واقعی W و فعال‌سازی HardTanh انجام می‌شود، در حالی که دیگری توسط وزن دودویی انجام می‌شود و تابع فعال سازی آن تابع علامت می‌باشد.

در طول انتشار رو به جلو، جزئیات مختلف در Y_l و $\tilde{Y}_l$ توسط W و B_W در چندین لایه ساخته می‌شود و در ویژگی‌های لایه ماقبل آخر جمع می‌شوند. سپس آن‌ها برای طبقه بندی به طبقه بندی خطی وارد می‌شوند و حاوی اطلاعات سطح بالا هستند که توسط ش.ع.ب با ارزش واقعی استخراج شده است. حال ما از $\hat{Y}_{L-1}$ برای ارائه نظارت اضافی برای بهبود عملکرد ش.ع.ب استفاده می‌کنیم. برای سادگی، زیرنویس را حذف می‌کنیم و فقط از Y و $\hat{Y}$ برای نمایش ویژگی‌های لایه ماقبل آخر استفاده می‌کنیم. برای شروع، با توجه به دسته‌ای از تصاویر $\{X^i\}_{i=1\dots N}$، ابتدا تقریب نمایش ساده‌ای را انجام می‌دهیم، و Y را با کمینه کردن فرمول زیر به $\hat{Y}$ نزدیک می‌کنیم:

$$\sum_{i}^{N} \phi_{\hat{Y}_i, Y_i} = ||\hat{Y}^i - Y^i||_2^2 \qquad (۳۲)$$

در اینجا، ما از $\phi_{\hat{Y}_i, Y_i}$ برای نشان دادن فاصله ℓ_2 دو بردار Y_l و $\tilde{Y}_l$ استفاده می‌کنیم. به این ترتیب، نمایش‌های Y و $\hat{Y}$ از یک تصویر مجبور می‌شوند نزدیکتر شوند. با این حال، تقریب سطح نمونه ممکن است به طور کامل از اطلاعات دسته کدگذاری شده توسط نمایش‌ها بهره برداری نکند. به عبارت دیگر، تقریب سطح نمونه، بازنمایی‌های استخراج شده توسط ستون فقرات دودویی و ستون فقرات پنهان را برای هر تصویر به طور جداگانه تراز می‌کند، که از اطلاعات برچسب در خود نمونه استفاده نمی‌کند.

با در نظر گرفتن اطلاعات دسته کدگذاری شده توسط نمایش‌های ماقبل آخر بین نمونه‌های مختلف، ما نظارت بر برچسب را با کشیدن $\hat{Y}$ و Y با همان برچسب به یکدیگر بر اساس معادله ۳۲ بیشتر معرفی می‌کنیم. ما از تابع هزینه تقریب نمایش آگاه از برچسب ($\mathcal{L}_{rep}$)را به صورت زیر فرموله می‌کنیم.

$$\sum_{i}^{N} K \left[\phi_{\hat{Y}_i, Y_i} + \sum_{j \in I(i)} (\phi_{Y_i, Y_i} + \phi_{\hat{Y}_i, Y_j} + \phi_{Y_i, \hat{Y}_j}) \right] \qquad (۳۳)$$

که در معادله بالا $I(i)$ مجموعه داده‌هایی می‌باشند که دسته یکسانی دارند و ضریب K یک ضریب نرمال‌ساز می‌باشد.

در معادله بالا، مورد $\phi_{\hat{Y}_i, Y_i}$ از معادله ۳۲ گرفته شده است. برای تراز کردن نمایش‌ها. علاوه بر این، ϕ_{Y_i, Y_i} ، $\phi_{\hat{Y}_i, Y_j}$ و $\phi_{Y_i, \hat{Y}_j}$ نمایش‌های تراز شده را به صورت جفت به هم نزدیک می‌کنند، که به طور متقارن تغییرات ویژگی‌های درون کلاس را کاهش می‌دهند. به این ترتیب، ما به تقریب بازنمایی از سطح نمونه به سطح دسته دست می‌یابیم. توجه داشته باشید که $\phi_{\hat{Y}_i, \hat{Y}_j}$ در عبارت تابع هزینه گنجانده نشده است، زیرا ما شیب را از Y جدا می‌کنیم. تابع هزینه تقریبی نمایش آگاه از برچسب را می‌توان از طریق نزول گرادیان بهینه کرد و گرادیان با توجه به $\hat{Y}$ را می‌توان به صورت زیر محاسبه کرد:

$$\frac{\partial \mathcal{L}_{rep}}{\partial Y^i} = 2K\left[\left(Y^i - \tilde{Y}^i\right) + \sum_{j\in I(i)}\left(2Y^i - Y^j - \tilde{Y}^i\right)\right] \quad (۳۴)$$
$$= 2K\left[(2|I(i)| + 1)Y^i - \sum_{j\in\{I(i)+i\}}\tilde{Y}^j - \sum_{j\in I(i)} Y^j\right]$$

بنابراین تابع هزینه نهایی که ترکیب تابع هزینه معرفی شده و تابع هزینه متقابل آنتروپی استاندارد $\mathcal{L}_{ce}$ است به صورت زیر می‌شود.

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda\,\mathcal{L}_{rep} \quad (۲۵)$$

که λ یک عامل تعادل در $\mathcal{L}_{rep}$ است که میزان تقریب نمایش را منظم می‌کند. بنابراین این با استفاده از این تابع هزینه برای آموزش ش.ع.ب می‌توان خروجی را به خروجی شبکه با وزن اصلی نزدیک نمود.

۳-۵. بهینه‌سازی مبتنی بر ساختار توپولوژیکی شبکه

روش‌های تقریب تابع علامت موجود، همگی بر بزرگی گرادیان نقاط حساس تأکید دارند، اما جهت بهینه‌سازی را نادیده می‌گیرند. تابع علامت ممکن است جهت بهینه‌سازی ناپایداری را به دلیل ناپایداری نقاط حساس ارائه دهد. بدیهی است که وزن‌های نزدیک به صفر نامطمئن تر هستند و در نتیجه در حین دودویی سازی آسیب پذیر و ناپایدارتر هستند. یادگیری با جهت نامشخص باعث همگرایی و بی ثباتی کند برای ش.ع.ب می‌شود.

هلوگن و همکاران [۳۷] پیشنهاد می‌کنند که به طور مستقیم وزن‌های دودویی را با توجه به گرادیان بهینه‌سازی کنیم و از به روز رسانی وزن‌های کمکی با دقت کامل صرف نظر کنیم. با این حال، این رویکرد در هنگام تلاش برای تخمین گرادیان مورد نیاز برای تغییر علامت بی اثر است. در بخش زیر روشی توسط[۳۸] پیشنهاد شده است را بررسی می‌نماییم که در آن عدم قطعیت دودویی‌سازی را مدل کرده و جهت بهینه‌سازی بر اساس عدم قطعیت تعیین می‌شود و از عدم قطعیت می‌توان برای تعیین دوتایی کردن وزن‌ها استفاده کرد.

۳-۵-۱. عدم قطعیت در ش.ع.ب

به طور شهودی، علامت وزن‌های نزدیک به صفر ممکن است به طور مکرر در فرآیند آموزش تغییر کنند، که منجر به نامطمئن‌تر شدن وضعیت می‌شوند. در مقابل، حالتی که از وزن صفر فاصله دارد، پایدارتر و در نتیجه مطمئن تر است. به منظور تخمین کمی عدم قطعیت ش.ع.ب، می‌توان یک تابع جدید را برای تخمین عدم قطعیت دوتایی وزن با در نظر گرفتن ویژگی‌های زیر معرفی کرد. همانطور که میدانیم عدم قطعیت در ۰ حداکثر است و به تدریج با نزدیک شدن وزن‌ها به ۱-/۱+ کاهش می‌یابد. با استفاده از مقدار مستمر پیش‌بینی‌شده و هدف آن، عدم قطعیت را به صورت زیر مدل‌سازی می‌کنیم:

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left(-\frac{x^2}{2\,\sigma^2}\right) \quad (۳۶)$$

ما این تابع گاوسی را برای فرمول‌بندی عدم قطعیت ش.ع.ب اعمال می‌کنیم. دودویی شدن با عدد بالاتر به معنای اطمینان کمتر و پتانسیل بیشتر برای برگشت است. اگر چه معادله ۳۶ اندازه گیری عدم قطعیت را برای ش.ع.ب ایستا فراهم می‌کند. برای بهینه‌سازی بهتر، همچنین لازم است که نوسان عدم قطعیت در آموزش ش.ع.ب پویا در نظر گرفته شود. برای این منظور، ما به طور کلی عدم قطعیت را به صورت زیر برآورد می‌کنیم:

$$\hat{f}(x_t) = \begin{cases} f(x) & t \le m \\ 1 - \prod_{i=t-m}^{t} \left(1 - f(x_i)\right) & t > m \end{cases} \quad (۳۷)$$

جایی که t بیانگر t-امین تکرار است. به این ترتیب، ما به صورت پویا عدم قطعیت ش.ع.ب را در فرآیند آموزش محاسبه می‌کنیم.

۲-۵-۳. ش.ع.ب آگاه از عدم قطعیت

برای به حداقل رساندن عدم قطعیت ش.ع.ب، نویسنده یک تابع علامت قطعیت را پیشنهاد داده است. با توجه به تکرار t در آموزش، تابع علامت c را می‌توان به صورت زیر نشان داد:

$$\text{csign}\,(x_t) = \begin{cases} \text{sign}\,(x_t) & \hat{f}(x_t) \le max\left(\hat{f}(x_{t-1}), \Delta\right) \\ \text{csign}\,(x_{t-1}) & \text{otherwise} \end{cases} \quad (۳۸)$$

که در آن x_t وزن با دقت کامل است و $\hat{f}(x_t)$ را از معادله ۳۷ محاسبه می‌کنیم و در معادله بالا Δ یک آستانه برای عدم قطعیت است. برای راحتی، به جای یک مقدار مشخص برای Δ ، برای آن یک آستانه تطبیقی برای عدم قطعیت معرفی می‌کنیم. با جایگزین کردن تابع علامت با تابع علامت c، از نظر تئوری می‌توان ثابت کرد که عدم قطعیت دوتایی کاهش می‌یابد.

با جایگزینی تابع علامت c بجای تابع علامت همگرایی ش.ع.ب سریعتر می‌شود و دقت آن نیز افزایش می‌یابد.

۳-۶. بهینه‌سازی مبتنی بر استراتژی آموزشی شبکه

در این بخش، ما به بررسی روشی[۳۹] میپردازیم که برای دستکاری توزیع وزن به منظور اتخاذ یک توزیع دو وجهی پیشنهاد شده است. این دستکاری با ترکیبی منحصر به فرد از یک اصطلاح تنظیم دو وجهی و یک روش تمرینی جدید با تقلید تنظیم وزن در کنار تقطیر دانش (KD[۱۱]) انجام می‌شود. تقطیر دانش یک تکنیک برای انتقال دانش از یک مدل، به عنوان یک معلم، به مدل دیگر، به عنوان یک دانش آموز است [۴۰]. به طور کلی، یک شبکه عصبی عمیق بزرگ به عنوان معلم و یک مدل فشرده تر به عنوان دانش آموز در این آموزش حضور دارند. این توزیع قادر است خطای کوانتیزاسیون مدل را کاهش دهد.

۱-۶-۳. منظم کردن وزن

وزن‌ها و فعال‌سازی‌های شبکه عصبی عمیق معمولاً از توزیع گاوسی یا لاپلاس پیروی می‌کنند [۴]. برای ش.ع.ب، این توزیع بهینه نیست. نویسنده می‌خواهد توزیع را بدون آسیب رساندن به عملکرد تغییر دهد، با این کار می‌توان خطای کوانتیزاسیون را کاهش داد و در نتیجه عملکرد را افزایش داد. در این کار از کشیدگی به عنوان نماینده ای برای توزیع احتمال استفاده می‌شود. کشیدگی از فرمول زیر بدست می‌آید.

$$Kurtosis[x] = \left[\mathbb{E}\left(\frac{X-\mu}{\sigma}\right)^4\right] \quad (۳۹)$$

که در آن μ و σ گشتاور اول و دوم X می‌باشند. منظم‌سازی کورتوز برروی تابع هزینه به‌صورت زیر اعمال می‌شود:

[۱۱] Knowledge distillation

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_k \quad (۴۰)$$

که در آن $\mathcal{L}_{ce}$تابع هزینه هدف است. $\mathcal{L}_k$ عبارت کشیدگی و λ ضریب است. $\mathcal{L}_k$ به صورت تعریف شده است

$$\mathcal{L}_k = \frac{1}{L}\sum_{i=1}^{L} |Kurtosis[W_i] - K_T|^2 \quad (۴۱)$$

که در آن L تعداد لایه‌ها وK_T هدف قاعده سازی کشش است.در این کار تـأثیر گرادیـان $\mathcal{L}_k$، و مقـدارK_T بـر توزیـع وزن مدل بررسی می‌شود. برای انجام فرآیند بهینه‌سازی از ژاکوبین تابع هزینه استفاده می‌شـود و از ایـن بـه بعـد پارامترهـای توزیع را حذف می‌کنیم. همینطور میدانیم ژاکوبین$\mathcal{L}_{ce}$ بر توزیع وزن تأثیر نمی‌گذارد زیرا نسـبت بـه W خطـی اسـت. از آنجایی که می‌خواهیم توضیح دهیم که ژاکوبین چگونه بر تغییر توزیع وزن‌ها تأثیر می‌گذارد، بـر تحلیـل گرادیـان تمرکـز می‌کنیم.

$$\frac{\partial \mathcal{L}_K}{\partial W_i} = \frac{8}{\sigma} \cdot \underbrace{\frac{(W_i - \mu)^3}{\sigma^3}}_{\bar{\mu}_3} \cdot \underbrace{\left|\frac{(W_i - \mu)^4}{\sigma^4} - K_T\right|}_{\kappa} \quad (۴۲)$$

این عبارت گرادیان شامل یک ضریب ثابت است که بر توزیع تأثیر نمی‌گذارد و دو بخش دیگر دارد: (۱) ممان سوم کـه بـا $\bar{\mu}_3$ نشان داده شده است و (۲) فاصله کشش از K_Tکه با κ نشان داده می‌شود. $\bar{\mu}_3$، که یک توان فرد از داده‌هـا اسـت، بـه طور متقارن وزن‌ها را از مرکز تابع توزیع دور می‌کند. میانگین‌ها، وزن‌هایی با مقدار کوچک، دارای گرادیان‌هـایی بـا مقـدار مولفه چولگی حتی خواهند بود.

κ شکل انتهای توزیع جابجا شده را نگه می‌دارد .وقتی $K_T = 3$ ، توزیع گرادیـان را نرمـال نگـه مـی‌دارد، و توزیـع وزن نرمال می‌ماند. بررسی‌ها نشان می‌دهد که ش.ع.ب زمانی کارآمدتر هستند که توزیع وزن آنهـا دو وجهـی باشـد و بـه طـور متقارن از میانگین جابجا شود .گرادیان$\mathcal{L}_k$ زمانی یکنواخت است که $K_T = -1.2$ باشد. در ایـن حالـت وزن‌هـا بـه طـور یکنواخت از میانگین دور می‌شوند.

در عمل، این یک توزیع دو وجهی پس از اپک ایجاد می‌شود، اما خطای تعمیم پـذیری افـزایش مـی‌یابـد و بـه مـدل آسیب می‌رساند .یک مقدار میانی، مانند$K_T = 1$ ، ممکن است یک مقدار کشش کاملا بهینه نباشد، اما می‌تواند به آرامی به سمت توزیع مورد نظر W حرکت کند. با این کار هیچ آسیب قابل توجهی به تعمیم پذیری مدل نمی‌رسد.

می‌توان با تنظیمK_T مختلف برای لایه‌های مختلف به توزیع دووجهی دلخواه رسید، به شرطی که میـانگین همـه مقـادیر K_Tبرابر با مقدار مورد نظر باشد. ما در آموزش شبکه با دقت آنها را به عنوان مجموعه ای از فراپارامترها انتخاب می‌کنـیم، بنابراین توزیع مورد نظر در همه لایه‌ها وجود دارد. ما از توضیحات بالا به منظور آموزش شبکه معلمی استفاده می‌کنیم که برای دانش آموز که ش.ع.ب می‌باشد مناسب تر باشد. همینطور برای سادگی، در ش.ع.ب از یک مقـدار ثابـتK_T اسـتفاده می‌کند.

۲-۶-۳. تقلید توزیع وزن و مدل آگاه از توزیع دووجهی

استفاده از یک شبکه عصبی عمیق به عنوان راهنما می‌تواند به به حداقل رساندن خطای کوانتیزاسـیون کمـک کنـد . برای پرکردن این شکاف اطلاعاتی، نویسنده یک طرح آموزشی تقلید توزیع وزن معلم-دانش‌آموز (WDM[۱۲]) را ارائه می-

[۱۲] Weight Distribution Mimicking

کند. توزیع وزن معلم-دانش‌آموز از اطلاعات شبکه عصبی عمیق استفاده می‌کند تا با به حداقل رساندن فاصله بین توزیع-ها خطای کمتری را در ش.ع.ب بدست آورد .روش تقطیر دانش می‌تواند به صورت موازی کار کند. به همین دلیل تقطیر دانش اضافه شده است تا نتایج بهتری به دست می‌آید.

$$\mathcal{L}_{WDM} = \sum_{i=1}^{L} D_{KL}\left(\mathcal{D}(W_{i,T}) \,||\, \mathcal{D}(W_{i,S})\right) + \beta\, \mathcal{L}_{KD}(y_T, y_S) \quad (۴۳)$$

که در آن وزن‌ها در لایه I ام در شبکه معلم را با $W_{i,T}$ و در ش.ع.ب دانش آموز با $W_{i,S}$ نشان می‌دهیم. L تعداد لایه‌های پیچشی، $\mathcal{D}$ توزیع وزن‌ها، y_Tوy_S به ترتیب پیش‌بینی‌های معلم و دانش‌آموز هستند وD_{KL} به واگرایی KullbackLeibler اشاره دارد. روش پیشنهادی از یک معلم ثابت استفاده می‌کند، به این معنی که ما فقط شبکه دانش‌آموز را بهینه می‌کنیم و از معلم به عنوان الگوی رفتار مورد نظر خود استفاده می‌کنیم. شبکه‌های نمایش بیت کم در پیروی از راهنمایی‌های توزیع از منظم‌سازی کشیدگی مشکلاتی دارند و برای انجام این کار به آموزش طولانی‌تری نیاز دارند. استفاده از طرح توزیع وزن معلم-دانش‌آموز به کاهش زمان لازم برای تغییر توزیع وزن ش.ع.ب کمک می‌کند.

ش.ع.ب آگاه از توزیع دووجهی (BD_BNN[۱۳]) به روش معلم-دانش‌آموز و از طریق دو مرحله اصلی آموزش می‌بینند: (۱) ما یک شبکه عصبی عمیق و یک ش.ع.ب با معماری را آموزش می‌دهیم. تنظیم توزیع وزن، که در آن شبکه عصبی عمیق دارای مقادیر K_T متفاوت برای هر لایه است، همانطور که در بخش قبل توضیح داده شده است.

(۲) شبکه عصبی عمیق را به عنوان معلم و ش.ع.ب را به عنوان دانش آموز در طرح آموزشی توزیع وزن معلم-دانش‌آموز ، همانطور که در بخش بالا توضیح داده شد، وارد می‌کنیم، و از KD ساده برای آموزش استفاده می‌کنیم .تابع هزینه ش.ع.ب به صورت زیر است:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda\, \mathcal{L}_k + \alpha\, \mathcal{L}_{WDM} \quad (۴۴)$$

در فرآیند بهینه‌سازی ش.ع.ب آگاه از توزیع دووجهی، از تقریب گرادیان استفاده می‌شود. توزیع وزن ش.ع.ب آگاه از توزیع دووجهی در هر مرحله از آموزش تغییر می‌کند، بنابراین توزیع وزن نهایی برای فرآیند دودویی کردن بهینه است، بنابراین این شیوه آموزش منجر به کاهش خطای کوانتیزاسیون در طول آموزش ش.ع.ب می‌شود.

۴. نتایج

۴-۱. مجموعه‌داده و جزئیات پیاده سازی

در این بخش، روش‌های بهینه‌سازی معرفی شده را بر روی دو مجموعه داده CIFAR-10 [۴۱] و ILSVRC-۲۰۱۲ ImageNet [۴۲] که به طور گسترده استفاده شده‌اند ارزیابی می‌کنیم. CIFAR یک مجموعه داده طبقه بندی تصاویر طبیعی در مقیاس کوچک با اندازه تصویر رنگی ۳۲در۳۲ پیکسل است. همینطور مجموعه آموزشی CIFAR10 از ۱۰۰۰۰ تصویر در ۱۰ کلاس تشکیل شده است. علاوه بر این، ILSVRC12 ImageNet مجموعه داده‌ای چالش برانگیز و متنوع‌تر است که شامل ۱.۲ میلیون تصویر آموزشی و ۵۰۰۰۰ تصویر اعتبارسنجی در ۱۰۰۰ کلاس است. مقیاس بزرگ و وضوح بالای آن، کار با آن را در مقایسه با CIFAR سخت‌تر می‌کند. ما برای مجموعه داده CIFAR10 از دو شبکه ResNet18

[۱۳] bi-modal distribution-aware BNN

و ResNet20 [۴۳] استفاده می‌کنیم و برای ارزیابی ImageNet از شبکه ResNet18 استفاده می‌کنیم. برای دودویی سازی، ما ویژگی‌ها و هسته‌ها را در لایه کانلوشن به جز لایه‌های اول و آخر ، دوتایی می‌کنیم و لایه اول و آخر را تغییر نمی‌دهیم.

۲-۴. نتایج

نتایج مربوط به استفاده از مجموعه داده CIFAR10 در جدول زیر ذکر شده است.

جدول ۱ : مقایسه روش‌های معرفی شده بر روی مجموعه داده CIFAR10

شبکه	روش	W/A	دقت در یک خروجی اول
ResNet-18	DA-BNN	-	-
	ReCU	**۱/۱**	**۹۲.۸٪**
	Label-aware-KD	۱/۱	۸۵.۱٪
	UaBNN	۱/۱	۸۲.۲٪
	BD-BNN	۱/۱	۹۲.۴۶٪
	FP	۳۲/۳۲	۹۴.۸٪
ResNet-20	DA-BNN	-	-
	ReCU	۱/۱	۸۷.۴٪
	Label-aware-KD	**۱/۱**	**۸۸.۶٪**
	UaBNN	-	-
	BD-BNN	۱/۱	۸۶.۵٪
	FP	۳۲/۳۲	۹۲.۱٪

در مجموعه داده CIFAR-10 بهترین عملکرد در شبکه ResNet-18 مربوط به واحد گیره اصلاح شده می‌باشد. اما در شبکه ResNet-20 روش تقریب نمایندگی آگاه از برچسب همراه با تقطیر دانش توانسته است بر واحد گیره اصلاح شده غلبه کند و جایگاه اول را بدست آورد. ستون‌های مشخص شده با – نشان دهنده آن است که نویسنده روش پیشنهادی خود را با شبکه و دیتاست مشخص شده ارزیابی ننموده است.

همینطور نتایج مربوط به استفاده از مجموعه داده ImageNet در جدول زیر ذکر شده است.

جدول ۲ :مقایسه روش‌های معرفی شده بر روی مجموعه دادهImageNet

شبکه	روش	W/A	دقت در یک خروجی اول	دقت در ۵ داده خروجی
ResNet-18	DA-BNN	۱/۱	۶۳.۱٪	۸۴.۳٪
	ReCU	**۱/۱**	**۶۶.۴٪**	**۸۶.۵٪**
	Label-aware-KD	۱/۱	۶۳.۸٪	۸۴.۹٪
	UaBNN	۱/۱	۶۰.۶٪	۸۲.۲٪
	BD-BNN	۱/۱	۶۳.۲۷٪	۸۴.۴۲٪
	FP	۳۲/۳۲	۶۹.۶٪	۸۹.۲٪

همانطور که در جدول بالا مشخص است روش واحد گیره اصلاح شده بهترین عملکرد را نسبت به سایر روش‌های معرفی شده در مجموعه داده ImageNet داشته است. علاوه بر آن فاصله دقت آن با شبکه عصبی عمیق عادی بسیار کاهش یافته است و این نشان از عملکرد بسیار خوب این روش در بهینه‌سازی ش.ع.ب دارد.

۵. بحث

همانطور که در بالا ذکر شد، از سال ۲۰۱۶، تکنیک‌های ش.ع.ب به دلیل توانایی آنها در استقرار مدل‌ها در دستگاه‌های با منابع محدود، توجه تحقیقاتی فزاینده‌ای را به خود جلب کرده‌اند. ش.ع.ب می‌تواند ذخیره سازی، پیچیدگی شبکه و مصرف انرژی را به میزان قابل توجهی کاهش دهد تا شبکه‌های عصبی را در دستگاه‌های تعبیه شده کارآمدتر کند. با این حال، دوتایی شدن به طور اجتناب ناپذیر باعث افت عملکرد قابل توجهی می‌شود. در این گزارش، ما ابتدا شبکه-های‌عصبی‌دودیی را معرفی و شیوه عملکرد و مزایا و معایب این شبکه‌ها را بررسی کردیم. سپس شیوه آموزش جلورو و عقب‌رو در این شبکه‌ها و تابع هزینه را بررسی نمودیم. پس از آن به بررسی علل دقت پایین این شبکه‌ها نسبت به شبکه عصبی پیچشی پرداختیم و در نهایت دسته‌بندی روش‌های مختلف بهبود شبکه‌های عصبی دودویی را بررسی کردیم و برای هر یک از این دسته‌ها روشی را برای بهینه‌سازی شبکه‌های عصبی دودویی توسط آن دسته بررسی کردیم.

منابع

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[2] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[3] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[4] Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., & Lin, D. (2019). Libra r-cnn: Towards balanced learning for object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 821-830).

[5] Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting masked faces in the wild with lle-cnns. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2682-2690).

[6] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[7] Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2018). Model compression and acceleration for deep neural networks: The principles, progress, and challenges. IEEE Signal Processing Magazine, 35(1), 126-136.

[8] Chen, Y. H., Yang, T. J., Emer, J., & Sze, V. (2019). Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 9(2), 292-308.

[9] Chen, Y. H., Krishna, T., Emer, J. S., & Sze, V. (2016). Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE journal of solid-state circuits, 52(1), 127-138.

[10] Cheng, Y., Yu, F. X., Feris, R. S., Kumar, S., Choudhary, A., & Chang, S. F. (2015). An exploration of parameter redundancy in deep networks with circulant projections. In Proceedings of the IEEE international conference on computer vision (pp. 2857-2865).

[11] Srinivas, S., & Babu, R. V. (2015). Data-free parameter pruning for deep neural networks. arXiv preprint arXiv:1507.06149.

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[13] Ge, S., Luo, Z., Zhao, S., Jin, X., & Zhang, X. Y. (2017, July). Compressing deep neural networks for efficient visual inference. In 2017 ieee international conference on multimedia and expo (icme) (pp. 667-672). IEEE.

[14] He, Y., Zhang, X., & Sun, J. (2017). Channel pruning for accelerating very deep neural networks. In Proceedings of the IEEE international conference on computer vision (pp. 1389-1397).

[15] Wu, J., Leng, C., Wang, Y., Hu, Q., & Cheng, J. (2016). Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4820-4828).

[16] Ge, S. (2018). Efficient deep learning in network compression and acceleration. In Digital Systems. London, UK: IntechOpen.

[17] Hu, Q., Li, G., Wang, P., Zhang, Y., & Cheng, J. (2018). Training binary weight networks via semi-binary decomposition. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 637-653).

[18] Xu, Z., Lin, M., Liu, J., Chen, J., Shao, L., Gao, Y., ... & Ji, R. (2021). Recu: Reviving the dead weights in binary neural networks. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 5198-5208).

[19] Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., & Lempitsky, V. (2014). Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv preprint arXiv:1412.6553.

[20] Lebedev, V., & Lempitsky, V. (2016). Fast convnets using group-wise brain damage. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2554-2564).

[21] Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European conference on computer vision (ECCV) (pp. 116-131).

[22] Xu, Z., Hsu, Y. C., & Huang, J. (2017). Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks. arXiv preprint arXiv:1709.00513.

[23] Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks. Advances in neural information processing systems, 29.

[24] Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016, October). Xnor-net: Imagenet classification using binary convolutional neural networks. In European conference on computer vision (pp. 525-542). Springer, Cham.
[25] Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830.
[26] Kim, M., & Smaragdis, P. (2016). Bitwise neural networks. arXiv preprint arXiv:1601.06071.
[27] Xu, Z., Lin, M., Liu, J., Chen, J., Shao, L., Gao, Y., ... & Ji, R. (2021). Recu: Reviving the dead weights in binary neural networks. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 5198-5208).
[28] Helwegen, K., Widdicombe, J., Geiger, L., Liu, Z., Cheng, K. T., & Nusselder, R. (2019). Latent weights do not exist: Rethinking binarized neural network optimization. Advances in neural information processing systems, 32.
[29] Zhong, K., Zhao, T., Ning, X., Zeng, S., Guo, K., Wang, Y., & Yang, H. (2020). Towards lower bit multiplication for convolutional neural network training. arXiv preprint arXiv:2006.02804, 3(4).
[30] Banner, R., Nahshan, Y., & Soudry, D. (2019). Post training 4-bit quantization of convolutional networks for rapid-deployment. Advances in Neural Information Processing Systems, 32.
[31] Raj, V., Nayak, N., & Kalyani, S. (2020). Understanding learning dynamics of binary neural networks via information bottleneck. arXiv preprint arXiv:2006.07522.
[32] Liu, C., Ding, W., Xia, X., Zhang, B., Gu, J., Liu, J., ... & Doermann, D. (2019). Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnns with circulant back propagation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2691-2699).
[33] Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., & Song, J. (2020). Forward and backward information retention for accurate binary neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2250-2259).
[34] Martinez, B., Yang, J., Bulat, A., & Tzimiropoulos, G. (2020). Training binary neural networks with real-to-binary convolutions. arXiv preprint arXiv:2003.11535.
[35] Zhao, J., Xu, S., Wang, R., Zhang, B., Guo, G., Doermann, D., & Sun, D. (2022). Data-adaptive binary neural networks for efficient object detection and recognition. Pattern Recognition Letters, 153, 239-245.
[36] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).
[37] Helwegen, K., Widdicombe, J., Geiger, L., Liu, Z., Cheng, K. T., & Nusselder, R. (2019). Latent weights do not exist: Rethinking binarized neural network optimization. Advances in neural information processing systems, 32.
[38] Zhao, J., Yang, L., Zhang, B., Guo, G., & Doermann, D. S. (2021, August). Uncertainty-aware Binary Neural Networks. In IJCAI (pp. 3441-3447).
[39] Rozen, T., Kimhi, M., Chmiel, B., Mendelson, A., & Baskin, C. (2022). Bimodal Distributed Binarized Neural Networks. arXiv preprint arXiv:2204.02004.
[40] Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2(7).
[41] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.
[42] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.
[43] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).