انتخاب کارای رتبه مناسب برای تجزیه تانسور در شبکه‌های عصبی عمیق

ایمان رحمانی اصل[۱]، مرتضی محجل کفشدوز[۲]، عبدالرضا رسولی کناری[۳]

۱- دانشجوی کارشناسی ارشد imanrahmanics@gmail.com

۲- استادیار دانشگاه صنعتی قم mohajjel@qut.ac.ir

۳-استادیار دانشگاه صنعتی قم rasooli@qut.ac.ir

خلاصه

امروزه شبکه‌های عصبی عمیق(Deep Neural Networks) پیشرفت‌ها وموفقیت‌های بزرگی در زمینه هوش مصنوعی به دست آورده‌اند. با بزرگ شدن این شبکه‌ها و پیچیده‌تر شدن آن‌ها، پیاده سازی این شبکه‌ها بر روی دستگاه‌هایی با منابع محدود مانند موبایل سخت‌تر شده است. به منظور حل این مشکل روش‌هایی برای فشرده‌سازی و شتابدهی این شبکه‌ها ارائه گردیده است. یکی از این روش‌ها تجزیه تانسور می باشد. در این مقاله از یکی از روش‌های تجزیه تانسور به نام تاکر سلسله مراتبی استفاده شده‌است. برای پیاده‌سازی تاکرسلسله‌مراتبی رتبه‌های مختلف به کار برده شده‌است. یکی از چالش‌های اصلی در این موضوع پیدا کردن رتبه مناسب برای تاکرسلسله مراتبی است. با روشی خاص که در مقاله ارائه شده می‌توان رتبه مناسب برای تجزیه را پیدا کرد. به وسیله‌ی این روش تعداد وزن‌ها به صورت قابل توجهی از جمله ۴۲۱ برابر فشرده می‌شود. در حالی که دقت این شبکه به صورت ناچیز در حدود ۳درصد تغییر می‌کند و حتی گاهی بهتر می‌شود.

کلمات کلیدی: فشرده‌سازی شبکه‌های عصبی عمیق، تجزیه تانسور، تجزیه تاکر سلسله مراتبی، انتخاب رتبه

۱. مقدمه

شبکه‌های عصبی عمیق در بسیاری از مسائل هوش مصنوعی مانند بینایی ماشین، پردازش زبان طبیعی(NLP) و ... به موفقیت خوبی دست پیدا کرده‌اند. دلیل این موضوع یادگیری ویژگی‌های سطح بالا از داده‌های حجیم است. همچنین این پیشرفت‌ها به خاطر مجموعه داده‌های حاضر و الگوریتم‌های توسعه پیداکرده و CPUها و GPUهای قوی می‌باشد. از طرف دیگر حجم داده‌ها و مقیاس شبکه‌های عصبی عمیق(DNNs) خیلی زیاد است. همچنین هزینه‌ی سخت‌افزار و زمان پردازش هم زیاد هست. علل مذکور سبب پیچیده‌شدن پیاده سازی DNNها روی دستگاه‌هایی با محدودیت منابع مصرفی مانند سیستم‌های نهفته و موبایل‌ها می‌شود. برای رفع این چالش روش‌ها و رویکردهای مختلفی برای فشرده‌سازی و شتابدهی شبکه‌های عصبی ارائه شده است.

این روش‌ها به دو دسته کلی سخت افزاری و نرم افزاری تقسیم می شوند که در بخش نرم افزاری به چهار دسته کلی فشرده‌سازی مدل، گسسته‌سازی، هرس کردن و تجزیه تانسور تعریف می‌شوند[1]، تا هزینه‌ی مصرفی و زمان اجرا را کم

کنند. در بین این روش‌ها تجزیه تانسور به خاطر تئوری ریاضیات قوی و پیاده سازی مختصر دارای مزیت می‌باشد[3]. در بخش تجزیه تانسور روش تاکر و CP (CANDECOMP/PARAFAC) از قدیمی ترین روش‌ها هستند. مشکل این روش‌های قدیمی بزرگ شدن ابعاد تانسورهای تجزیه (Curse of dimensionality) است. از روش‌های دیگر می‌توان به تانسور سلسله ای (Tensor Train) که توسط Oseledets در سال ۲۰۱۱ معرفی شد[2]، اشاره کرد. در این روش، تانسورهای وزن به صورت تانسورهای سه بعدی تبدیل می‌شوند تا بدین صورت فضای ذخیره‌سازی کمتر شود. این روش برای تجزیه لایه‌های کانولوشن مناسب تر از انواع دیگر تجزیه‌ها می‌باشد[3]. یک نوع دیگر از تجزیه‌ها تانسورحلقه‌ای(Tensor Ring) است[4]، که مدل توسعه یافته تانسورسلسله‌ای می‌باشد. روش دیگر تاکر سلسله مراتبی است که در ادامه توضیح داده شده است. کار اصلی این مقاله استفاده از تاکرسلسله مراتبی است. روش تاکر سلسله مراتبی مشکل بزرگ شدن ابعاد تانسورهای تجزیه را ندارد و همچنین برای تجزیه لایه‌های به هم متصل (Fully connected) مناسب تر از روش‌های تجزیه دیگر است[3]. همه‌ی این روش‌ها به دنبال کاهش افزونگی بین وزن‌ها در شبکه‌های عصبی مختلف هستند. در روش‌های تجزیه از رتبه‌های(Rank) مختلف برای تجزیه لایه‌ها استفاده می‌شود. وابسته به مقدار این رتبه‌ها مقدار کاهش پارامتر ها و فشرده سازی لایه‌ها و شبکه متفاوت خواهدبود. در واقع پیداکردن رتبه مناسب برای فشرده‌سازی خود یک چالش هست. در ضمن علاوه بر پیدا کردن رتبه، باید پارامترهای دیگر هم مانند دقت(Accuracy) شبکه را هم بررسی کرد تا دچار افت محسوس نشود. در واقع یک چالش این است که باید رتبه مناسب را پیدا کرد و هم مراقب بود مقدار دقت خیلی کاهش پیدا نکند.

ما برای فشرده‌سازی لایه‌ها و شتابدهی شبکه از روش تجزیه تاکر سلسله مراتبی برای هرلایه استفاده کردیم. علاوه براین برای انتخاب رتبه از روش انشعاب گونه (branch)[5]، در هر لایه استفاده می‌کنیم. در واقع ما علاوه بر تجزیه به دنبال انتخاب رتبه مناسب برای هر تجزیه در هر لایه هستیم. به اضافه باید مراقب بود در حین تجزیه مقدار دقت شبکه خیلی کاهش پیدا نکند و همچنین تعداد وزن‌ها به طور قابل توجهی کاهش پیدا کند. بعد از انتخاب تجزیه مناسب با رتبه مرتبطش برای زمان استنتاج شبکه به جای استفاده از لایه اولیه از لایه تجزیه شده با رتبه مناسبش استفاده می‌شود. به طور خلاصه کارما به صورت مراحل زیر است:

- تجزیه هر لایه با روش تاکر سلسله مراتبی به طوری که هر لایه از چند تاکر سلسله مراتبی با رتبه‌های متفاوت استفاده می‌کند.
- استفاده از روش انشعاب گونه برای انتخاب رتبه مناسب از بین رتبه‌های موجود
- جایگزین کردن تجزیه با رتبه منتخب برای فشرده‌سازی شبکه

درادامه در بخش ۲ راجع به کارهای مرتبط توضیح داده‌شده است. راجع به تجزیه به روش سلسله‌مراتبی و کاراصلی مقاله در بخش ۳ بحث شده است. در بخش ۴ نتایج و آزمایش‌ها بیان گردیده است. ودر آخر نتیجه‌گیری در بخش ۵ صورت گرفته است.

۲. کارهای مرتبط

در سال‌های اخیر کارهای زیادی در زمینه شتابدهی شبکه‌های عصبی و فشرده سازی آن‌ها انجام شده‌است. این کارها در ۴ دسته گسسته‌سازی، فشرده‌سازی مدل، هرس کردن و تجزیه تانسور قرار می‌گیرند[1]. تمرکز ما در زمینه تجزیه تانسور می‌باشد. از قبیل کارهای صورت گرفته در تجزیه تانسور می‌توان به تجزیه تانسور سلسله‌ای، تجزیه تاکر سلسله‌مراتبی ، تجزیه تانسور حلقه‌ای و ... اشاره کرد. برای آشنایی و بررسی کامل کارهای صورت گرفته در تجزیه تانسور به مقاله Deng2020 [1]، مراجعه نمایید. Wang [6]، در سال ۲۰۲۰ یک روش با استفاده از تجزیه‌ی تانسورسلسله

ای در هسته‌های کانولوشن برای شبکه‌های کانولوشن سه بعدی ارائه داد. Wu [3]، در سال ۲۰۲۰ یک روش ترکیبی برای تجزیه پیشنهاد داد. در کار او از تاکر سلسله مراتبی برای تجزیه‌ی ماتریس‌های وزن و از تانسورسلسله‌ای برای تجزیه هسته‌های کانولوشن استفاده گردیده است چراکه در این کار ادعا شده است که تاکر سلسله مراتبی برای قسمت به هم پیوسته و تانسور سلسله‌ای برای کانولوشن مناسب تر هستند. Diao و همکارانش [5]، در سال ۲۰۲۱ از روش هرس کردن برای فشرده‌سازی شبکه استفاده کردند. آن‌ها برای هر لایه از روش انشعاب گونه برای انتخاب روش مناسب بهره بردند. Yin [7]، در سال ۲۰۲۱ مقاله‌ای ارائه داد که درآن از روش تاکرسلسله مراتبی برای تجزیه شبکه‌های بازگشتی(RNN) استفاده کرد. Wang [8] درسال ۲۰۱۸ روش تانسورحلقه‌ای را برای شبکه های کانولوشن و به هم پیوسته(Fully connected) به کار برد. در سال ۲۰۲۰ با استفاده از یادگیری تقویتی روشی برای انتخاب رتبه‌های مناسب برای تانسورحلقه‌ای توسط Cheng [9]، ارائه گردید. Su [10]، روشی برای تجزیه LSTMها با استفاده از تانسورترین ارائه داد. در سال ۲۰۲۱، Li [11]، یک روش اکتشافی برای انتخاب رتبه مناسب برای تانسور حلقه‌ای ارائه داد. در روش او از الگوریتم ژنتیک استفاده شده است. یک روش برای محاسبه تانسور سلسله‌ای بر روی Tensor Flow توسط Novikov[12] ارائه گردیده که می تواند مقدار تجزیه را محاسبه کند. Lin [13] توانست با روش تقریب رتبه-کم (low-rank) بر روی شبکه‌های کانولوشن و تجزیه فیلترهای کانولوشن افزونگی بین کانال‌ها را کم کند. Lee[14] به بررسی عملگرهای تانسوری در انواع تجزیه پرداخت. Ahmadi [15] برای محاسبه تاکر یک الگوریتم تصادفی معرفی کرد که ابتدا از یک روش رتبه-کم مانند تاکر استفاده می‌کند . سپس الگوریتم تصادفی را بر روی آن اعمال می‌کند. Jiang [16] یک فریمورک محاسباتی برای CPU و FPGA را با تجزیه تاکر با ابعاد کم معرفی می‌کند. روش تانسور حلقه‌ای توسط Zhao[4] نخستین بار در سال ۲۰۱۶ معرفی گردیده است. Denton [17] یک ساختارخطی برای مجاسبه تجزیه رتبه-کم بر روی شبکه‌های کانولوشن ارائه داده است که می‌تواند در زمان تست، شبکه را کاهش دهد. Chunhua [18] یک طرح براساس تانسور سلسله‌ای معرفی می‌کند که علاوه بر کاهش تعداد وزن‌ها در شبکه عصبی عمیق، تعداد دسترسی به حافظه را هم کم می‌کند.

۳. رویکرداصلی و روش پیشنهادی

در این مقاله برای نمایش تانسور بالای ۳ بعد از نماد اسکریپتی با حروف بزرگ مانند $\mathcal{A}$، $\mathcal{B}$ و ... و برای ماتریس از حروف بزرگ انگلیسی به صورت ساده مانندA، B و ... و برای بردار از حروف ساده کوچک مانند a ،b و... استفاده می‌کنیم.

۳-۱- تجزیه تاکر سلسله‌مراتبی

فرم پایه:

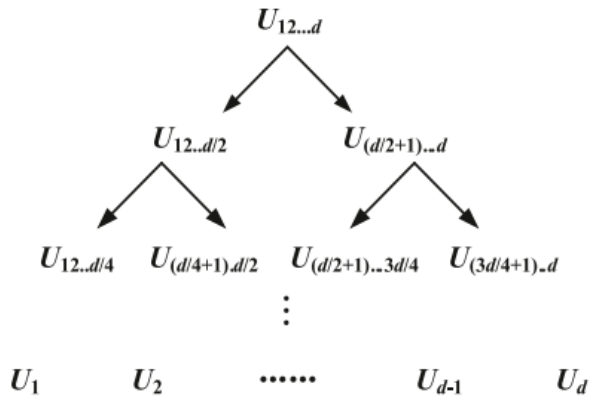
قالب تاکر سلسله‌مراتبی[3] توسط Hackbusch و Kuhn در سال ۲۰۰۹ معرفی گردید و بر اساس آن تجزیه تاکر سلسله‌مراتبی توسط Grasedyck در سال ۲۰۱۰ ارائه گردید. برای یک تانسور $\mathcal{A} \in R^{n_1 \times n_2 \times \ldots \times n_d}$ می‌توان ابعاد را به دو مجموعه تقسیم کرد مانند $t = \{t_1, t_2, \ldots, t_k\}$ و $s = \{s_1, s_2, \ldots, s_{d-k}\}$، بدین‌گونه یک شکل ماتریسی (matricization) از $\mathcal{A}$ مانند $A^{(t)} \in R^{n_{t_1} n_{t_2} \ldots n_{t_k} \times n_{s_1} n_{s_2} \ldots n_{s_{d-k}}}$ تولید می‌کنیم. به طور مشابه، مجموعه t می‌تولند به دو مجموعه t_l و t_v برای تولید دو ماتریس $A^{(t_l)}$ و $A^{(t_v)}$ تبدیل شود. اگر ما ماتریس‌های پایه ستونی مورد نظر هر یک از ماتریس‌های مذکور را به عنوان U_{t_v} و U_{t_l} و U_t تعریف کنیم، ما می‌توانیم داشته باشیم:

$$U_t = (U_{t_l} \otimes U_{t_v})B_t \quad (۱)$$

که $U_t \in R^{n_{t_1} n_{t_2} \dots n_{t_k} \times r_t}$ و $U_{t_l} \in R^{n_{t_{l_1}} n_{t_{l_2}} \dots n_{t_{l_i}} \times r_{t_l}}$ و $U_{t_v} \in R^{n_{t_{v_1}} n_{t_{v_2}} \dots n_{t_{vk-i}} \times r_{t_v}}$ ماتریس‌های کاهشی (truncated matrices) هستند و $B_t \in R^{r_{t_l} r_{t_v} \times r_t}$ ماتریس انتقال نامیده می‌شود. همه r_t، r_{t_l} و r_{t_v} به عنوان رتبه هستند. علاوه بر این، عملگر $\otimes$ ضرب کرونکر را نشان می‌دهد. به صورت بازگشتی با استفاده از معادله تا جایی که دیگر نتوان ماتریس‌های کاهشی را بیشتر تجزیه کرد می‌توان جلو رفت و بدین صورت ساختار تاکرسلسله‌مراتبی ساخته خواهدشد. این روند برای $\mathcal{A}$ به صورت زیر نمایش داده می‌شود.

$$\begin{aligned} \mathcal{A} \to U_{12\dots d} &= (U_{12\dots\frac{d}{2}} \otimes U_{(\frac{d}{2}+1)\dots d})B_{12\dots d} \quad (۲) \\ &= [(\left(U_{12\dots\frac{d}{4}}\right. \\ &\otimes \left. U_{(\frac{d}{4}+1)\dots\frac{d}{2}}\right) B_{12\dots\frac{d}{2}}) \otimes (\left(U_{\frac{d}{2}+1\dots\frac{3d}{4}} \otimes U_{\frac{3d}{4}+1\dots d}\right) B_{\frac{d}{2}+1\dots d})]B_{12\dots d} \end{aligned}$$

که $U_{12\dots d} \in R^{n_1 n_2 \dots n_d \times 1}$ از تانسور $\mathcal{A}$ تغییر شکل داده می‌شود. این نوع نمایش فرم پایه‌ی تاکر سلسله مراتبی را نشان می‌دهد و می‌تواند به عنوان یک درخت ابعاد که از d ماتریس کاهشی و d-1 ماتریس انتقال تشکیل شده، کشیده شود که در شکل ۱ نشان داده شده است.



شکل ۱- ساختار سلسله مراتبی تاکر سلسله مراتبی[3]

فرم نرمال:

با استفاده از یک قانون عمومی $AB \otimes CD = (A \otimes C)(B \otimes D)$ می‌توان یک قالب تاکر سلسله‌مراتبی مختصرتری از معادله ۲ بدست آورد که در زیر نشان داده شده است:

$$U_{12\dots d} = (U_1 \otimes U_2 \otimes \dots \otimes U_d)(B_{12} \otimes B_{34} \otimes B_{34} \otimes \dots B_{(d-1)d}) \dots B_{12\dots d} \quad (۳)$$

به این فرم، فرم نرمال تاکر سلسله مراتبی گفته می‌شود. مهمترین ویژگی این فرم این است که محاسبات به صورت سطح به سطح از دید درخت ابعاد مرتب شده‌اند.

فرم قراردادی:

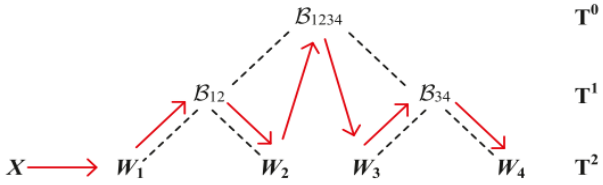
به طور جزئی اگر ما $B_t \in R^{r_{t_l} r_{t_v} \times r_t}$ در معادله ۱ را به عنوان یک تانسور ۳ بعدی $\mathcal{B}_t \in R^{r_{t_l} \times r_{t_v} \times r_t}$ تغییر شکل دهیم ما می توانیم داشته باشیم:

$$U_t = U_{t_l} \times^1 B_t \times^1 U_{t_v}^T \quad (۴)$$

که $\times^1$ ضرب قراردادی مود۱ گفته می‌شود. سپس با استفاده بازگشتی از فرمول بالا و استفاده از فرمول ۲ و ۳ خواهیم داشت:

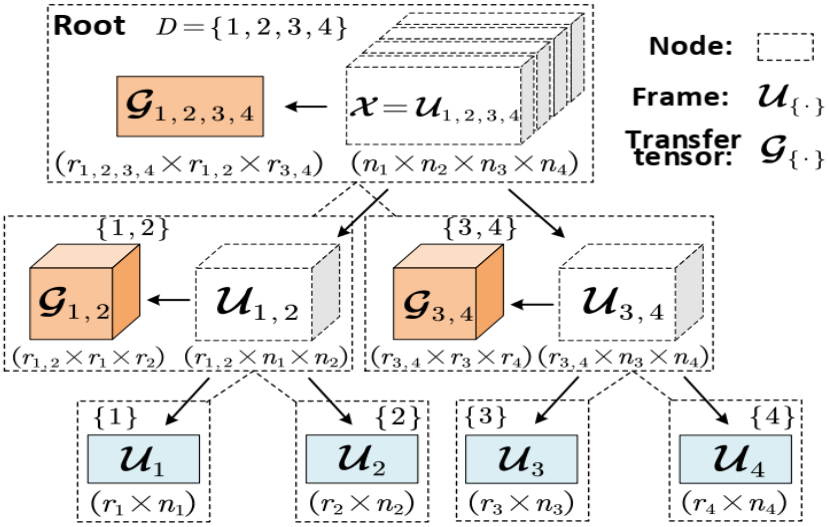
$$U_{12\dots d} = (\dots((U_1 \times^1 B_{12} \times^1 U_2) \times^1 B_{1234} \times^1 (U_3 \times^1 B_{34} \times^1 U_4)) \times^1 \dots \times^1 (U_{(d-1)} \times^1 B_{(d-1)d} \times^1 U_d) \dots) \quad (۵)$$

به فرمول ۵ فرم قراردادی تاکرسلسله مراتبی می‌گویند. در این فرم از ضرب کرونکر که هزینه زیادی دارد، پرهیز شده است. بنابراین روند محاسبات موردنظر هزینه محاسبات و ذخیره سازی زیادی مصرف نمی‌کنند. برای مثال، فرم قراردادی تاکر سلسله مراتبی می‌تواند اجازه دهد یک تانسور ورودی به همراه ماتریس‌های انتقال و کاهشی یک به یک محاسبه شود که در شکل ۲ نشان داده شده است.



شکل ۲-فرم قراردادی تاکرسلسله مراتبی[3]

برای ضرب ورودی در تاکر سلسله مراتبی باید از پیمایش پیش ترتیب درخت تجزیه تاکرسلسله مراتبی استفاده کرد.



شکل ۳-نحوه ضرب ورودی در درخت تجزیه تاکر[7]

شکل ۳ یک تانسور ۴ بعدی را نشان می‌دهد که به صورت سلسله مراتبی تجزیه شده‌است. $\mathcal{G}$ها ماتریس و تانسورهای انتقال را نشان می‌دهد. در واقع ابتدا تانسور ۴ بعدی به دو تانسور۳ بعدی تجزیه شده و سپس هر تانسور ۳بعدی به دو ماتریس تبدیل شده است. مقدار r ها همان رنک‌ها می‌باشد.

۲-۳-روش پیشنهادی

دراین روش ابتدا هرلایه به صورت انشعابی به دو شاخه تقسیم می‌شود. یک انشعاب لایه‌ی ساده شبکه‌ی اصلی بدون هیچ تجزیه‌ای و انشعاب دیگر تجزیه لایه به وسیله تاکرسلسله مراتبی به همراه رتبهi می‌باشد. سپس از بین دوگزینه، با استفاده از آموزش(train) انشعاب، تصمیم گرفته می‌شود یک و فقط یک انشعاب یک شود و دیگری صفر و آن انشعابی که یک شده به عنوان خروجی لایه در نظر گرفته می‌شود و برلایه‌ی موردنظر اعمال می‌شود. نکته‌ی مهم در اینجا این است که انتخاب بین صفرویک به واسطه تابع علامت زیر به کاربرده می‌شود.

$$TG(\omega)=\begin{cases}1 & \omega>0\\ 0 & \omega\le 0\end{cases} \qquad (۶)$$

انشعاب برای یادگیری شبکه و استفاده از گرادیان کاهشی در حین آموزش باید پیوسته باشد. (به منظور مشتق گرفتن) اما از آنجایی که تابع علامت ما صفر ویک را انتخاب می‌کند در نقطه ای مشتق‌پذیر نیست. به همین خاطر از یک تابع پیوسته برای شبیه‌سازی این تابع استفاده می‌شود. این تابع پیوسته به صورت زیر است[5]:

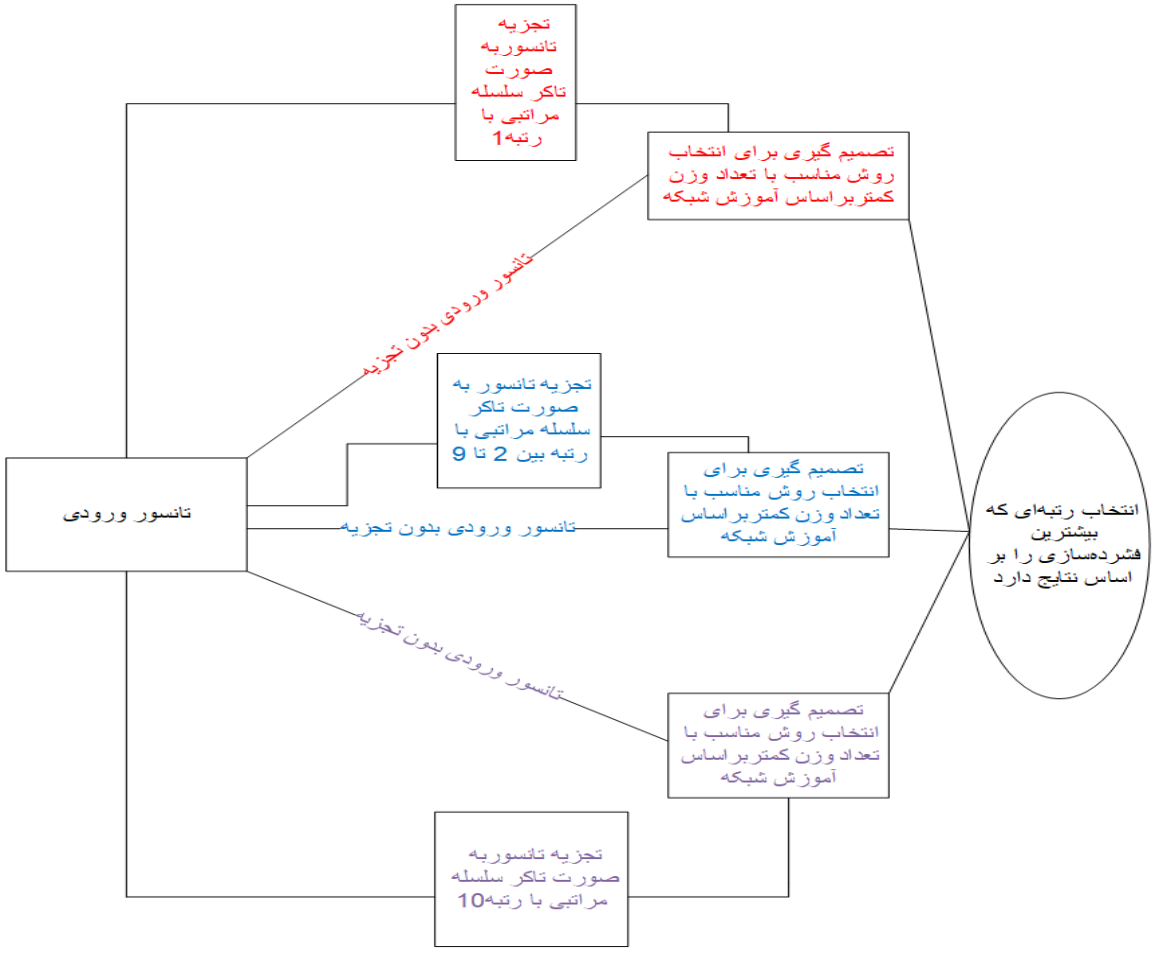
$$TG(\omega)\approx A(\omega)=k.(\tanh(t.\omega)+1) \qquad (۷)$$

که $t=T_{\min}10^{\frac{i}{N}\times\log\left(\frac{T\max}{T_{\min}}\right)}$ ، $k=\frac{1}{2}\max\left(\frac{1}{t},1\right)$ و $T_{\min}=0.1$ و $T_{\max}=10$ و N تعداد دوره‌ها(epoch) در فرآیند یادگیری و i دوره فعلی را نشان می‌دهد.

با استفاده از تابع فوق می توان خود انشعاب را هم در آموزش دخالت داد. بعد از آموزش لایه وکل شبکه هر انشعاب مشخص شده و براساس آن شبکه‌ی جدید ساخته می‌شود. در این مقاله ما ابتدا برای هرلایه شبکه یک رتبه به منظور استفاده در تاکرسلسله مراتبی انتخاب می‌کنیم وسپس آن را با حالت بدون تجزیه مقایسه می‌کنیم(در واقع دو انشعاب داریم) بعد ازآن خود شبکه بعد از یادگیری تصمیم می‌گیرد کدام حالت مناسب تر است. و در آخر در مرحله استنتاج به جای این لایه‌ها ،لایه‌های آموزش داده شده در شبکه قرار می‌گیرد.

انتخاب رتبه مناسب یکی از دغدغه‌های این مقاله می‌باشد. ما برای انتخاب رتبه مناسب از بین اعداد یک تا ده رتبه‌ها را انتخاب کردیم. سپس برای هر کدام یک شبکه‌ی انشعابی درست کردیم. بعد از آموزش میزان کم شدن پارامترها و وزن‌ها به دست می‌آید. از بین گزینه‌ها آن رتبه‌ای انتخاب می‌شود که مقدار بیشتری شبکه را فشرده کرده باشد.

مراحل روش پیشنهادی در شکل ۴ به نمایش در آمده است.



شکل ۴-مراحل اجرای روش پیشنهادی

از دیگر نکات مهم این مقاله استفاده از یک تابع loss سفارشی(customize) شده‌است. ما در این‌جا علاوه بر استفاده از تابع‌های مرسوم یک تابع دیگر از مقدار پارامترها را هم اضافه کردیم تا بدین وسیله میزان فشرده‌سازی شبکه را اندازه بگیریم و در آموزش شبکه این معیار را هم در نظر گرفته باشیم.

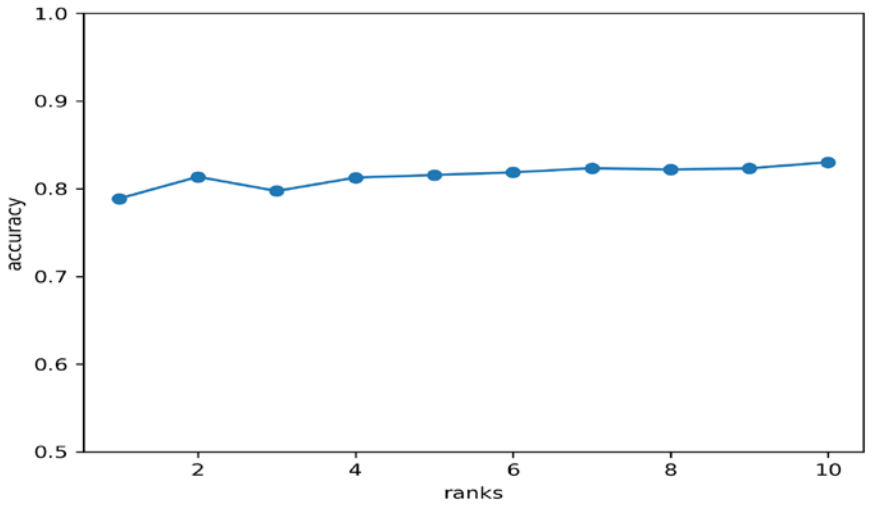
۴. آزمایش‌ها

در این مقاله ما از مجموعه داده Fashion-Mnist استفاده کردیم. این مجموعه داده شامل ۷۰۰۰۰ عکس ۲۸ در ۲۸ پیکسل سیاه وسفید است. این داده ۱۰ الگو دارد که به ازای هر ۷۰۰۰ عکس یک الگو موجود است. ما در این تحقیق از ۶۵۰۰۰ عکس برای داده آموزشی و از ۵۰۰۰ عکس برای تست استفاده کردیم.

در این مقاله شبکه‌ی ما از ۵ لایه به هم پیوسته(FC) که در لایه یک از ۱۰۲۴ نورون، در لایه‌های بعدی به ترتیب از ۵۱۲، ۲۵۶، ۱۲۸ و ۱۰ نورون استفاده شده است. توابع فعالیتی (Activation Function) که ما استفاده کردیم عبارت اند از Relu و Softmax .

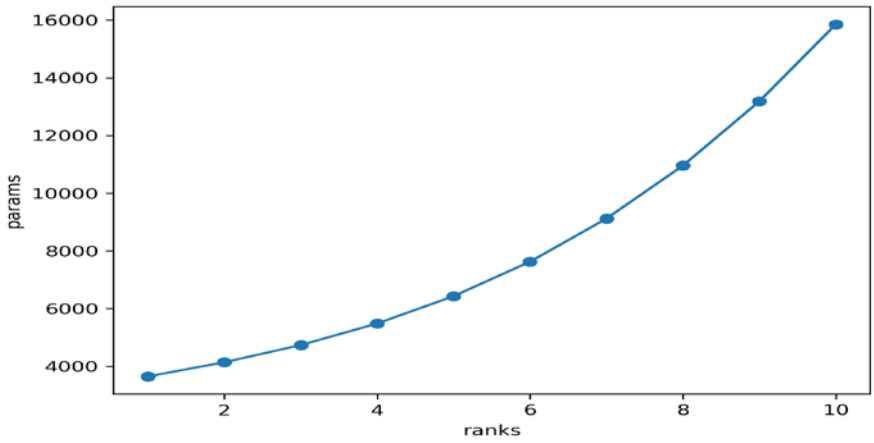
ما برای ارزیابی روش خود از ۵ معیار ارزیابی استفاده کردیم. مقدار رتبه تجزیه تاکر سلسله مراتبی، میزان دقت شبکه بر روی داده‌ها تست، تعداد وزن‌ها یا پارامترهای بدون فشرده‌سازی و با فشرده‌سازی و مقدار pcr که میزان فشرده سازی را نشان می‌دهد.

$$pcr = \frac{params_{original}}{params_{compressed}} \quad (۸)$$



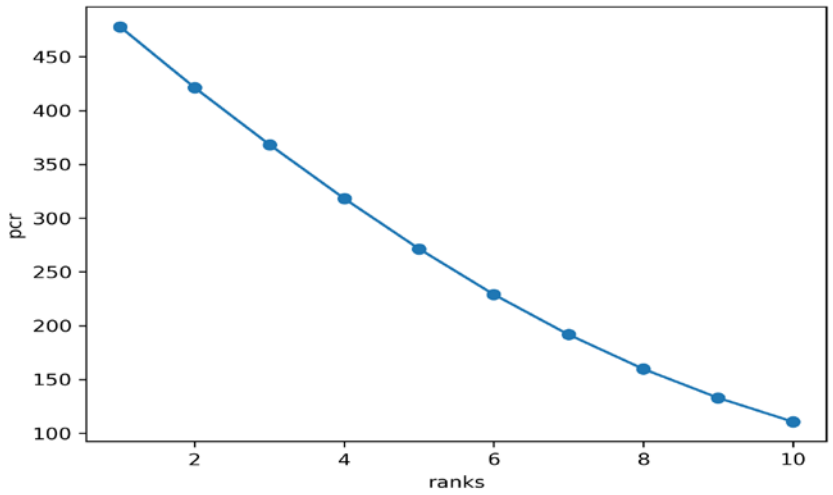
شکل ۵-مقدار دقت بر حسب مقدار رنک

شکل ۵ رابطه بین مقدار رتبه و دقت را نشان می‌دهد. نکته مهمی که به دست می‌آید این است که به ازای رتبه های مختلف مقدار دقت به طور ناچیزی تغییر می‌کند. در واقع مقدار رتبه در میزان دقت تاثیر کمی دارد.



شکل ۶-مقدار پارامترهای فشرده‌شده بر حسب رتبه

شکل ۶ رابطه بین رتبه و تعداد پارامترهای فشرده شده بعد از تجزیه را نشان می‌دهد. همان طور که مشاهده می‌شود با افزایش رتبه تعداد پارامترها افزایش می‌یابد. در واقع هرچه رتبه کمتر باشد پارامترهای کمتری در شبکه می‌ماند.



شکل ۷-مقدار pcr بر حسب مقدار رتبه

شکل ۷ ارتباط بین مقدار رتبه و مقدار pcr را نشان می‌دهد. نکته ای که از این شکل به دست می‌آید این است که میزان فشرده سازی در رتبه های کمتر بهتر از رتبه‌های بالاتر است.

جدول ۱-بررسی پنج مولفه ارزیابی نسبت به یکدیگر

رتبه(rank)	دقت(accuracy)	پارامترهای اصلی	پارامترهای فشرده شده	pcr
R=1	77%	1.7M	3646	477.8
R=2	79%	1.7M	4138	421.16
R=3	81%	1.7M	4734	368.26
R=4	82%	1.7M	5482	318.15
R=5	81%	1.7M	6430	271.3
R=6	82%	1.7M	7626	228.98
R=7	83%	1.7M	9118	191.68
R=8	82%	1.7M	10954	159.72
R=9	83%	1.7M	13182	132.89
R=10	83%	1.7M	15850	110.69
بدون رتبه	82%	1.7M	ندارد	ندارد

باتوجه به جدول ۱ می توان مقدار پارامترها را به مقدار قابل توجهی کاهش داد بدون آنکه مقدار دقت تغییر قابل توجهی بکند و حتی در رتبه‌های بالا مقدار دقت بهتر هم شود. با استفاده از جدول بالا به دلیل فشرده‌سازی بیشتر نسبت به سایرین تجزیه با رتبه ۱ به عنوان تجزیه منتخب انتخاب می شود.

۵. نتیجه گیری

تجزیه تانسور یک روش خوب برای فشرده‌سازی شبکه‌های عصبی عمیق است. تاکرسلسله‌مراتبی یکی از روش‌های موثر تجزیه تانسور می‌باشد. ما با استفاده از این روش توانستیم علاوه بر فشرده‌سازی شبکه با تاکر سلسله مراتبی، رتبه‌ی مناسب برای این تجزیه را انتخاب کنیم به طوری که هم تعداد وزن‌ها به طور قابل توجهی کاهش پیدا کند و هم دقت به مقدار کمی تغییر کند.

روش ما همان طور که نتایج نشان داد، موثر واقع گردیده است. ما با این راه حل توانستیم شبکه را تا ۳۱۸ برابر فشرده کنیم بدون آنکه دقت کم شود.

ما درآینده می‌خواهیم از روش‌های دیگر تجزیه‌تانسور استفاده کنیم و همچنین یک روش خودکار برای انتخاب رتبه‌ی مناسب تجزیه ارائه دهیم. علاوه بر این ها می‌خواهیم این تجزیه‌ها را بر روی شبکه‌های کانولوشن هم مورد استفاده قرار دهیم.

مراجع

1. **Deng, Lei, et al. "Model compression and hardware acceleration for neural networks: A comprehensive survey."** ***Proceedings of the IEEE*** **108.4 (2020): 485-532.**
2. **Oseledets, Ivan V. "Tensor-train decomposition."** ***SIAM Journal on Scientific Computing*** **33.5 (2011): 2295-2317.**
3. **Wu, Bijiao, et al. "Hybrid tensor decomposition in neural network compression."** ***Neural Networks*** **132 (2020): 309-320.**
4. **Zhao, Qibin, et al. "Tensor ring decomposition."** ***arXiv prepriarXiv:1606.05535*** **(2016).**
5. **Diao, Huabin, et al. "Implementation of Lightweight Convolutional Neural Networks via Layer-Wise Differentiable Compression."** ***Sensors*** **21.10 (2021): 3464.**
6. **Wang, Dingheng, et al. "Compressing 3DCNNs based on tensor train decomposition."** ***Neural Networks*** **131 (2020): 215-230.**
7. **Yin, Miao, et al. "Towards Extremely Compact RNNs for Video Recognition with Fully Decomposed Hierarchical Tucker Structure."** ***Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*****. 2021.**
8. **Wang, Wenqi, et al. "Wide compression: Tensor ring nets."** ***Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*****. 2018.**

9. **Cheng, Zhiyu, et al. "A novel rank selection scheme in tensor ring decomposition based on reinforcement learning for deep neural networks." *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020.**
10. **Su, Jiahao, et al. "Convolutional tensor-train lstm for spatio-temporal learning." *arXiv preprint arXiv:2002.09131* (2020).**
11. **Li, Nannan, et al. "Heuristic rank selection with progressively searching tensor ring network." *Complex & Intelligent Systems* (2021): 1-15.**
12. **Novikov, Alexander, et al. "Tensor Train Decomposition on TensorFlow (T3F)." *J. Mach. Learn. Res.* 21.30 (2020): 1-7.**
13. **Lin, Shaohui, et al. "Espace: Accelerating convolutional neural networks via eliminating spatial and channel redundancy." *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.**
14. **Lee, Namgil, and Andrzej Cichocki. "Fundamental tensor operations for large-scale data analysis using tensor network formats." *Multidimensional Systems and Signal Processing* 29.3 (2018): 921-960.**
15. **Ahmadi-Asl, Salman, et al. "Randomized algorithms for computation of Tucker decomposition and higher order SVD (HOSVD)." *IEEE Access* 9 (2021): 28684-28706.**
16. **Jiang, Weiyun, et al. "Sparse Tucker Tensor Decomposition on a Hybrid FPGA-CPU Platform." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2020).**
17. **Denton, Emily L., et al. "Exploiting linear structure within convolutional networks for efficient evaluation." *Advances in neural information processing systems*. 2014.**
18. **Deng, Chunhua, et al. "TIE: Energy-efficient tensor train-based inference engine for deep neural network." *Proceedings of the 46th International Symposium on Computer Architecture*. 2019.**