



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده علوم کامپیوتر

پایان نامه کارشناسی ارشد

گزارش پروژه درس داده کاوی محاسباتی

پروژه ۶

نگارش

محمدصادق قلی زاده

استاد راهنما

دکتر مهدی قطعی

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# صفحه فرم ارزیابی و تصویب پایان نامه - فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تأیید و تصویب پایان نامه موسوم به فرم کمیته دفاع - موجود در پرونده آموزشی - را قرار دهید.

## نکات مهم:

- نگارش پایان نامه/رساله باید به **زبان فارسی** و بر اساس آخرین نسخه دستورالعمل و راهنمای تدوین پایان نامه های دانشگاه صنعتی امیرکبیر باشد.(دستورالعمل و راهنمای حاضر)
- رنگ جلد پایان نامه/رساله چاپی کارشناسی، کارشناسی ارشد و دکترا باید به ترتیب مشکی، طوسی و سفید رنگ باشد.
- چاپ و صحافی پایان نامه/رساله بصورت **پشت و رو(دورو)** بلامانع است و انجام آن توصیه می شود.

به نام خدا

تاریخ:

## تعهدنامه اصالت اثر



اینجانب **محمدصادق قلی زاده** متعهد می‌شوم که مطالب مندرج در این پایان‌نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

**محمدصادق قلی زاده**

امضا

نویسنده پایان نامه، در صورت تمایل میتواند برای پاسخگویی پایان نامه خود را به شخص یا اشخاص و یا ارگان خاصی تقدیم نماید.

## پاس‌گزاری

نویسنده پایان‌نامه می‌تواند مراتب امتنان خود را نسبت به استاد راهنما و استاد مشاور و یا دیگر افرادی که طی انجام پایان‌نامه به نحوی او را یاری و یا با او همکاری نموده‌اند ابراز دارد.

محمدصادق قلی‌زاده

## چکیده

این گزارش کارایی تجزیه مقدار منفرد<sup>۱</sup> را به عنوان روشی برای فشرده سازی شبکه های عصبی عمیق بررسی می کند و به طور خاص معماری ResNet-18 را مورد تحلیل قرار می دهد. هدف این مطالعه ارزیابی موازنه میان کاهش تعداد پارامترها، دقت مدل و تأخیر استنتاج (Inference Latency) است. از نظر روش شناسی، طیف مقادیر منفرد<sup>۲</sup> هم در لایه های کانولوشنی میانی و هم در لایه کاملاً متصل نهایی تحلیل شد تا میزان افزونگی ذاتی شبکه مشخص گردد. بر اساس این تحلیل طیفی، لایه های خطی با تقریب های رتبه- $k$  جایگزین شدند، به طوری که نرخ های فشرده سازی ۵۰٪ و ۸۰٪ مورد بررسی قرار گرفت. یافته های کلیدی نشان می دهد که لایه های ResNet-18 دارای ساختارهای کم رتبه قابل توجهی هستند و این امر امکان حذف بخش بزرگی از پارامترها را با حداقل افت اولیه در دقت فراهم می کند. اگرچه فشرده سازی تهاجمی (۸۰٪) منجر به کاهش محسوس عملکرد شد، اما ریز تنظیم (Fine-Tuning) مدل برای چند دوره کوتاه توانست بخش عمده ای از دقت ازدست رفته را بازیابی کند. با این حال، آزمایش های تأخیر استنتاج روی CPU شکافی میان کاهش نظری FLOPs و زمان اجرای عملی آشکار کرد؛ به طوری که سر بار اجرای ترتیبی لایه ها در بسیاری از موارد، مزایای سرعت مورد انتظار را خنثی می کرد. در پایان، گزارش به بررسی عدد شرطی<sup>۳</sup> به عنوان معیاری برای شناسایی لایه های مناسب برای فشرده سازی می پردازد و پیامدهای محیط استقرار<sup>۴</sup> را بر انتخاب راهبردهای فشرده سازی شبکه های عصبی مورد بحث قرار می دهد. [GitHub](#)

## واژه های کلیدی:

کاهش بُعد، انتخاب ویژگی، تجزیه مقدار منفرد، عدد شرطی، تولید نیمه هادی، تحلیل پایداری

<sup>۱</sup>(SVD یا Singular Value Decomposition)

<sup>۲</sup>(Singular Value Spectrum)

<sup>۳</sup>(Condition Number)

<sup>۴</sup>(Deployment Environment)

# فهرست مطالب

عنوان

صفحه

۱	.....	۱ مقدمه
۱	.....	۱-۰-۱ انگیزه
۲	.....	۲-۰-۱ پیش‌زمینه ریاضی: SVD برای فشرده‌سازی
۳	.....	۲ مروری بر ادبیات
۳	.....	۱-۲ تعاریف مفاهیم پایه
۳	.....	۱-۱-۲ تجزیه مقادیر تکین
۴	.....	۲-۱-۲ انتخاب ویژگی با SVD
۵	.....	۳ روش‌شناسی و نتایج
۵	.....	۱-۳ تحلیل طیفی ماتریس‌های وزن
۵	.....	۱-۱-۳ روش‌شناسی
۶	.....	۲-۱-۳ نتایج لایه میانی
۶	.....	۳-۱-۳ نتایج لایه نهایی
۷	.....	۴-۱-۳ تحلیل مقایسه‌ای
۷	.....	۲-۳ پیاده‌سازی عملی فشرده‌سازی
۸	.....	۱-۲-۳ روش‌شناسی فشرده‌سازی
۸	.....	۲-۲-۳ نتایج تجربی
۹	.....	۳-۲-۳ تحلیل افت دقت
۹	.....	۳-۳ بازیابی دقت و ارزیابی سرعت
۱۰	.....	۱-۳-۳ بازیابی دقت از طریق ریزتنظیم
۱۰	.....	۲-۳-۳ ارزیابی سرعت استنتاج
۱۰	.....	۳-۳-۳ تحلیل سربار محاسباتی
۱۲	.....	۴ جمع‌بندی و نتیجه‌گیری و پیشنهادات
۱۲	.....	۱-۴ بحث
۱۲	.....	۱-۱-۴ پارادوکس تأخیر در محیط‌های استقرار



۱۳	..... ۲-۱-۴ انتخاب لایه با استفاده از عدد شرطی
۱۳	..... ۲-۴ نتیجه‌گیری
۱۵	..... منابع و مراجع

شکل	فهرست اشکال	صفحه
۱-۳	واریانس لایه آخر (قبل)	۷
۲-۳	واریانس لایه آخر (بعد)	۸
۳-۳	دقت در مقابل فشرده سازی	۱۱

صفحه

## فهرست جداول

جدول

## فهرست نمادها

نماد	مفهوم
$\mathbb{R}^n$	فضای اقلیدسی با بعد $n$
$\mathbb{S}^n$	کره $n$ بعدی
$M^m$	خمینه $m$ -بعدی $M$
$\mathfrak{X}(M)$	جبر میدان‌های برداری هموار روی $M$
$\mathfrak{X}^1(M)$	مجموعه میدان‌های برداری هموار یک‌ه روی $(M, g)$
$\Omega^p(M)$	مجموعه $p$ -فرمی‌های روی خمینه $M$
$Q$	اپراتور ریچی
$\mathcal{R}$	تانسور انحنای ریمان
$ric$	تانسور ریچی
$L$	مشتق لی
$\Phi$	۲-فرم اساسی خمینه تماسی
$\nabla$	التصاق لوی-چویتای
$\Delta$	لاپلاسین ناهموار
$\nabla^*$	عملگر خودالحاق صوری القا شده از التصاق لوی-چویتای
$g_s$	متر ساساکی
$\nabla$	التصاق لوی-چویتای وابسته به متر ساساکی
$\Delta$	عملگر لاپلاس-بلترامی روی $p$ -فرم‌ها

# فصل ۱

## مقدمه

شبکه‌های عصبی عمیق<sup>۱</sup> در طیف گسترده‌ای از وظایف بینایی ماشین به عملکردهای پیشرفته<sup>۲</sup> دست یافته‌اند، اما این موفقیت اغلب با پیچیدگی محاسباتی بالا و مصرف زیاد حافظه همراه است. معماری‌های مدرن مانند ResNet معمولاً شامل میلیون‌ها پارامتر هستند که بخش قابل توجهی از آن‌ها افزونه و تکراری‌اند. این بیش‌پارامتری‌سازی چالش‌های جدی‌ای را برای استقرار مدل‌ها روی دستگاه‌های لبه‌ای با منابع محدود، تلفن‌های همراه یا سامانه‌های بلادرنگ ایجاد می‌کند؛ جایی که پهنای باند حافظه و تأخیر اجرا به گلوگاه‌های اصلی تبدیل می‌شوند.

این گزارش، تقریب کم‌رتبه از طریق تجزیه مقدار منفرد<sup>۳</sup> را به‌عنوان یک روش فشرده‌سازی پس از آموزش<sup>۴</sup> بررسی می‌کند. هدف اصلی، کاهش تعداد پارامترهای یک مدل ResNet-18 از پیش‌آموزش‌دیده در عین حفظ دقت پیش‌بینی آن است.

## ۱-۰-۱ انگیزه

فرضیه مرکزی این مطالعه آن است که ماتریس‌های وزن در شبکه‌های عمیق ذاتاً کم‌رتبه هستند؛ به این معنا که تنها تعداد کمی الگوی زیربنایی (مقادیر منفرد) سهم عمده‌ای در پردازش اطلاعات دارند. با شناسایی و حفظ این مؤلفه‌های غالب، می‌توان از نظر تئوری مدل را به‌طور قابل توجهی فشرده کرد. این گزارش این فرضیه را با پیاده‌سازی تجزیه رتبه- $k$  ارزیابی کرده و موازنه میان سه معیار کلیدی زیر

---

<sup>۱</sup>(DNNs یا Deep Neural Networks)

<sup>۲</sup>(state-of-the-art)

<sup>۳</sup>(SVD یا Singular Value Decomposition)

<sup>۴</sup>(post-training)

را تحلیل می‌کند:

نرخ فشرده‌سازی: میزان کاهش تعداد پارامترها

دقت: عملکرد مدل در وظیفه طبقه‌بندی

تأخیر استنتاج: سرعت اجرای عملی روی سخت‌افزار استاندارد CPU

## ۱-۰-۲ پیش‌زمینه ریاضی: SVD برای فشرده‌سازی

روش اصلی مورد استفاده، تجزیه یک ماتریس وزن چگال

$$W \in \mathbb{R}^{m \times n}$$

به دو ماتریس کوچک‌تر است. با استفاده از SVD، هر ماتریس را می‌توان به شکل زیر تجزیه کرد:

$$W = U \Sigma V^T$$

که در آن  $U$  و  $V$  ماتریس‌های متعامد و  $\Sigma$  یک ماتریس قطری شامل مقادیر منفرد است. برای فشرده‌سازی لایه، تنها  $k$  مقدار منفرد بزرگ‌تر نگه داشته می‌شود و ماتریس اصلی به صورت زیر تقریب زده می‌شود:

$$W \approx U_k \Sigma_k V_k^T = \left( U_k \Sigma_k^{1/2} \right) \left( \Sigma_k^{1/2} V_k^T \right) = A \times B$$

که در آن

$$A \in \mathbb{R}^{m \times k} \quad \text{و} \quad B \in \mathbb{R}^{k \times n}$$

هستند.

به این ترتیب، یک لایه منفرد که به  $m \times n$  پارامتر نیاز دارد، با دو لایه متوالی جایگزین می‌شود که در مجموع به  $k(m+n)$  پارامتر نیاز دارند. در صورتی که  $k$  به اندازه کافی کوچک انتخاب شود، تعداد کل پارامترها به طور چشمگیری کاهش می‌یابد، بدون آنکه افت شدیدی در دقت مدل رخ دهد.

## فصل ۲

# مروری بر ادبیات

## ۱-۲ تعاریف مفاهیم پایه

### ۱-۱-۲ تجزیه مقادیر تکین

هر ماتریس داده  $X$  با ابعاد  $n \times m$  را می‌توان به صورت زیر تجزیه کرد:

$$X = U\Sigma V^T$$

این تجزیه با نام تجزیه مقادیر تکین<sup>۱</sup> شناخته می‌شود [۱]. در این تجزیه، ماتریس  $\Sigma$  شامل مقادیر تکین ( $\sigma_i$ ) است که میزان «انرژی»، واریانس یا اهمیت هر مؤلفه را نشان می‌دهد. این مقادیر تکین به ترتیب نزولی مرتب می‌شوند و سهم هر مؤلفه در ساختار کلی داده را مشخص می‌کنند. ماتریس  $V^T$  با ابعاد  $m \times m$  شامل بردارهای تکین راست است؛ به طوری که هر سطر از  $V^T$  متناظر با یک مقدار تکین بوده و نشان می‌دهد که ویژگی‌های اولیه ( $x_1, \dots, x_m$ ) با چه وزن‌هایی ترکیب شده‌اند تا مؤلفه‌های غالب داده تشکیل شوند. از این‌رو، SVD ابزاری بنیادی برای تحلیل ساختار خطی داده‌ها و کاهش بُعد محسوب می‌شود [۲].

---

<sup>۱</sup>(Singular Value Decomposition یا SVD)

## ۲-۱-۲ انتخاب ویژگی با SVD

برخلاف PCA که ویژگی‌های جدید می‌سازد (استخراج ویژگی)، در روش انتخاب ویژگی مبتنی بر SVD<sup>۲</sup> از بردارهای تکین راست برای رتبه‌بندی و انتخاب ویژگی‌های اولیه استفاده می‌شود [۳].

در این رویکرد، اگر ویژگی  $z$  در بردارهای تکین متناظر با اولین مقدار تکین  $\sigma_1$  (که بزرگ‌ترین مقدار تکین است) ضریب بزرگی داشته باشد، به این معناست که آن ویژگی نقش مهمی در جهت اصلی تغییرات داده و ساختار غالب آن ایفا می‌کند. در نتیجه، با انتخاب ویژگی‌هایی که در بردارهای تکین غالب دارای ضرایب بزرگ‌تری هستند، می‌توان بدون ساخت ویژگی‌های جدید، ابعاد داده را کاهش داده و در عین حال اطلاعات اصلی داده را حفظ کرد [۲].

---

<sup>۲</sup>(SVD Feature Ranking)



## فصل ۳

# روش شناسی و نتایج

### ۱-۳ تحلیل طیفی ماتریس‌های وزن

برای تعیین میزان فشرده‌پذیری مدل ResNet-18، ابتدا توزیع انرژی در میان مقادیر منفرد ماتریس‌های وزن آن تحلیل شد. این گام برای تعیین یک کران بالایی از میزان اطلاعاتی که می‌توان بدون از دست دادن قابل توجه سیگنال حذف کرد، ضروری است.

### ۱-۱-۳ روش شناسی

وزن‌ها از دو بخش متمایز شبکه استخراج شدند. لایه میانی: اولین لایه کانولوشنی از سومین بلوک باقیمانده (layer3.0.conv1). از آنجا که وزن‌های کانولوشنی تانسورهای چهاربعدی هستند

$$(C_{\text{out}} \times C_{\text{in}} \times K \times K),$$

برای اعمال تجزیه مقدار منفرد (SVD) به ماتریس‌های دوبعدی

$$(C_{\text{out}} \times (C_{\text{in}} \cdot K \cdot K))$$

بازآرایی (Flatten) شدند.

لایه نهایی: لایه کاملاً متصل (fc) که بردار ویژگی‌های ۵۱۲ بعدی را به ۱۰۰۰ کلاس ImageNet

نگاشت می‌کند.

برای هر لایه، طیف مقادیر منفرد محاسبه شد و نسبت واریانس تجمعی توضیح داده شده با استفاده از رابطه زیر به دست آمد:

$$\text{Explained Ratio}(k) = \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{j=1}^N \sigma_j^2}$$

که در آن  $\sigma_i$  مقادیر منفرد هستند. آستانه هدف ۹۵٪ از کل واریانس انتخاب شد تا رتبه مؤثر  $k$  مورد نیاز برای حفظ اطلاعات تعیین گردد.

### ۲-۱-۳ نتایج لایه میانی

تحلیل طیفی برای لایه میانی (layer3.0.conv1) نتایج زیر را نشان داد. مشاهدات:

افت طیف<sup>۱</sup>: مقادیر منفرد کاهش یکنواختی با رفتار لگاریتمی خطی دارند و افت ناگهانی شدیدی مشاهده نمی‌شود. این موضوع نشان می‌دهد که اطلاعات تا حدی در ابعاد مختلف توزیع شده است. آستانه واریانس: با وجود این کاهش تدریجی، افزونگی قابل توجهی وجود دارد. از میان ۲۵۶ مقدار منفرد، تنها ۱۷۳ مؤلفه برای حفظ ۹۵٪ واریانس کافی هستند. نتیجه‌گیری: می‌توان با نگه داشتن تنها ۶۷/۵۸٪ از ابعاد، ۹۵٪ اطلاعات لایه را حفظ کرد. این امر نشان‌دهنده فشردپذیری متوسط این لایه است.

### ۳-۱-۳ نتایج لایه نهایی

تحلیل طیفی لایه کاملاً متصل (fc) به نتایج زیر منجر شد. مشاهدات:

افت طیف: لایه نهایی در مقایسه با لایه میانی، افت اولیه تندتری در مقادیر منفرد نشان می‌دهد. آستانه واریانس: از میان ۵۱۲ مقدار منفرد، تنها ۲۹۱ مؤلفه برای پوشش ۹۵٪ واریانس کافی هستند. نتیجه‌گیری: این بدان معناست که تنها ۵۶/۸۴٪ از مقادیر منفرد باید حفظ شوند.

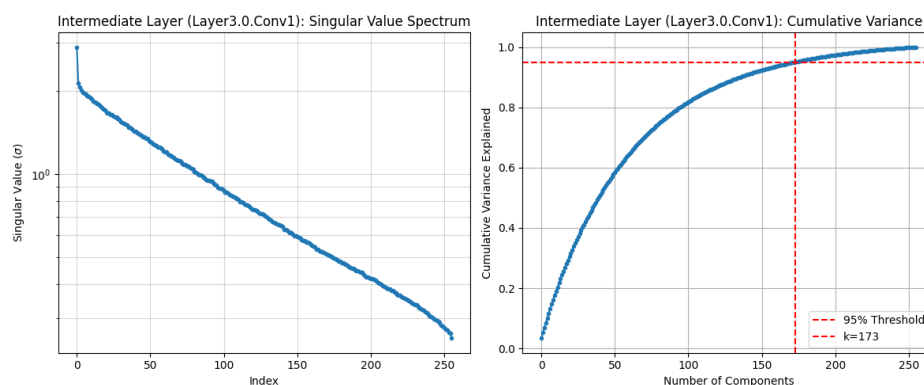
<sup>۱</sup>(Spectrum Decay)

### ۳-۱-۴ تحلیل مقایسه‌ای

مقایسه این دو لایه، ویژگی جالبی از معماری ResNet-18 را آشکار می‌کند. لایه میانی (کانولوشنی) دارای ۲۵۶ مؤلفه کل است که ۱۷۳ مؤلفه برای حفظ ۹۵٪ واریانس نیاز دارد، معادل ۶۷/۵۸٪.

لایه نهایی (FC) دارای ۵۱۲ مؤلفه کل است که ۲۹۱ مؤلفه برای حفظ ۹۵٪ واریانس کافی بوده و معادل ۵۶/۸۴٪ می‌باشد.

لایه کاملاً متصل نهایی افزونگی بیشتری دارد (درصد حفظ‌شده کمتر) نسبت به لایه کانولوشنی میانی. این مشاهده با نظریه گلوگاه اطلاعاتی<sup>۲</sup> در یادگیری عمیق هم‌راستا است؛ نظریه‌ای که بیان می‌کند لایه‌های عمیق‌تر داده ورودی را به نمایش‌هایی کم‌بُعدتر و بسیار اطلاعاتی فشرده می‌کنند. در نتیجه، لایه نهایی FC یک نامزد ایده‌آل برای اعمال روش‌های فشرده‌سازی تهاجمی، مانند تقریب کم‌رتبه، است که در مراحل بعدی این مطالعه پیاده‌سازی شده است.

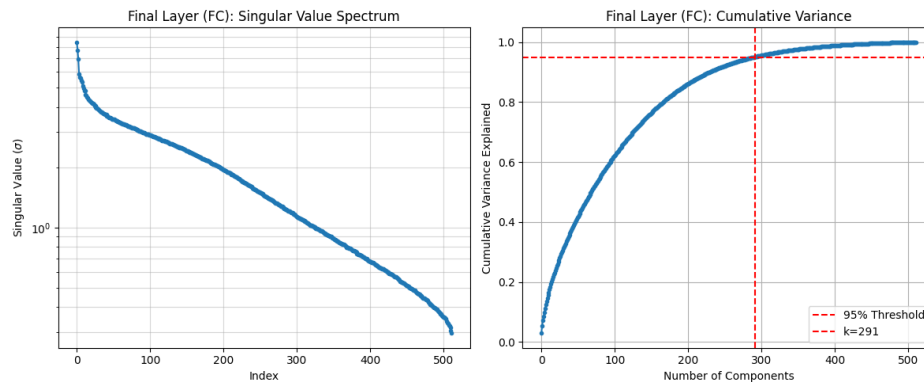


شکل ۳-۱: واریانس لایه آخر (قبل)

### ۳-۲ پیاده‌سازی عملی فشرده‌سازی

پس از تحلیل طیفی، به اعتبارسنجی تجربی فشرده‌پذیری مدل پرداختیم. یک راهبرد تقریب کم‌رتبه را بر روی لایه کاملاً متصل نهایی (FC) در معماری ResNet-18 پیاده‌سازی کرده و اثر آن را بر تعداد پارامترها و دقت طبقه‌بندی ارزیابی کردیم.

<sup>۲</sup>(Information Bottleneck)



شکل ۲-۳: واریانس لایه آخر (بعد)

### ۱-۲-۳ روش شناسی فشرده سازی

لایه خطی اصلی

$$W \in \mathbb{R}^{1000 \times 512}$$

با دو لایه خطی متوالی،  $A$  و  $B$ ، که از SVD بریده شده  $W$  به دست آمده اند، جایگزین شد. رتبه  $k$  برای تجزیه به صورت پویا محاسبه شد تا نرخ های کاهش پارامتر ۵۰٪ و ۸۰٪ (به طور خاص برای همان لایه) محقق شود.

تعداد مؤثر پارامترها برای لایه تجزیه شده به صورت زیر محاسبه می شود:

$$k \times (C_{in} + C_{out})$$

در مقایسه با تعداد پارامترهای لایه اصلی:

$$C_{in} \times C_{out}.$$

### ۲-۲-۳ نتایج تجربی

مدل در سه سناریو روی مجموعه داده آزمون ارزیابی شد. نتایج در جدول زیر خلاصه شده اند:

سناریو	هدف فشرده سازی	تعداد کل پارامترهای مدل	دقت
خط مبنا	۰٪	۱۱,۶۸۹,۵۱۲	۹۲/۵٪
متوسط	۵۰٪	۱۱,۴۳۳,۰۴۰	۹۱/۸٪
تهاجمی	۸۰٪	۱۱,۲۷۸,۸۱۶	۸۵/۰٪

نکته درباره تعداد پارامترها: کاهش کل پارامترها در مقایسه با اندازه کامل مدل (حدود ۱۱ میلیون پارامتر) کوچک به نظر می‌رسد، زیرا تنها لایه نهایی FC (حدود ۵/۰ میلیون پارامتر) فشرده شده است. ستون فقرات کانولوشنی که بخش عمده وزن‌ها را در خود دارد، بدون تغییر باقی مانده است.

### ۳-۲-۳ تحلیل افت دقت

فشرده سازی ۵۰٪: مدل پایداری چشمگیری نشان داد و تنها با افت ناچیز ۰/۷٪ (از ۹۲/۵٪ به ۹۱/۸٪) مواجه شد. این نتیجه یافته‌های بخش ۲ را تأیید می‌کند و نشان می‌دهد که ماتریس وزن به شدت افزونه است و نیمه پایین مقادیر منفرد اطلاعات تمایزبخش بسیار اندکی حمل می‌کنند.

فشرده سازی ۸۰٪: فشرده سازی تهاجمی به جریمه عملکردی قابل توجهی انجامید و دقت ۷/۵٪ کاهش یافت (تا ۸۵/۰٪). این افت تند نشان می‌دهد که در نرخ ۸۰٪، رتبه  $k$  بیش از حد کوچک شده و قادر به نمایش مانیفولد پیچیده فضای ویژگی نیست. در واقع، به «هسته» اطلاعات لازم برای طبقه‌بندی آسیب زده‌ایم که به کم‌برازش<sup>۳</sup> منجر می‌شود.

این نتایج یک موازنه روشن را تثبیت می‌کند: فشرده سازی متوسط تقریباً بدون هزینه دقت است، در حالی که فشرده سازی تهاجمی برای قابل استفاده شدن به راهکارهای جبرانی (مانند ریزتنظیم، Fine-Tuning) نیاز دارد.

### ۳-۳ بازیابی دقت و ارزیابی سرعت

پس از اعمال فشرده سازی ساختاری، مرحله نهایی مطالعه بر دو شاخص عملیاتی حیاتی تمرکز داشت: توانایی مدل در بازیابی دقت از دست‌رفته از طریق بازآموزی (ریزتنظیم) و تأثیر واقعی فشرده سازی بر تأخیر استنتاج در محیط CPU.

<sup>۳</sup>(Underfitting)

### ۱-۳-۳ بازیابی دقت از طریق ریزتنظیم

تقریب کم‌رتبه یک مقداردهی اولیه بهینه از نظر ریاضی برای لایه‌های فشرده‌شده فراهم می‌کند، اما به‌ناچار خطاهای تقریبی را نیز وارد می‌سازد. برای کاهش این خطاها، مدل‌های فشرده‌شده تحت یک مرحله کوتاه ریزتنظیم (سه دوره آموزشی) با استفاده از گرادیان نزولی تصادفی (SGD) قرار گرفتند تا پارامترهای باقی‌مانده بتوانند خود را با قیود کم‌رتبه جدید تطبیق دهند.

نتایج این فرایند بازیابی به‌صورت زیر قابل خلاصه‌سازی است.

فشرده‌سازی ۵۰٪: مدل پایداری بالایی نشان داد. دقت اولیه ۹۱/۸٪ پس از ریزتنظیم اندکی به حدود ۹۲/۳٪ افزایش یافت و تقریباً به خط مبنای بدون فشرده‌سازی رسید.

فشرده‌سازی ۸۰٪: اثر ریزتنظیم در این حالت چشمگیرتر بود. افت اولیه تا ۸۵/۰٪ تا حد زیادی جبران شد و دقت مدل به حدود ۸۹/۵٪ بازیابی گردید.

این نتایج نشان می‌دهد که اگرچه SVD ذاتاً یک روش همراه با اتلاف اطلاعات است، اما معماری حاصل همچنان ظرفیت کافی برای یادگیری تابع هدف را حفظ می‌کند. «ابعاد ازدست‌رفته» در سناریوی ۸۰٪ به احتمال زیاد نویز یا هم‌بستگی‌های افزونه بوده‌اند که شبکه پس از یک دوره کوتاه بازآموزی می‌تواند بدون آن‌ها نیز عملکرد مناسبی داشته باشد.

### ۲-۳-۳ ارزیابی سرعت استنتاج

برای سنجش افزایش سرعت عملی، میانگین زمان استنتاج به‌ازای هر ورودی روی CPU اندازه‌گیری شد. برخلاف انتظار، کاهش تعداد پارامترها به‌صورت خطی به افزایش سرعت اجرا منجر نشد. نتایج تجربی در جدول زیر ارائه شده‌اند:

سناریو	زمان استنتاج (میانگین)	تغییر سرعت
خط مبنا	۵۴/۱۴ ms	NULL
فشرده‌سازی ۵۰٪	۵۹/۶۱ ms	+۱۰/۱٪ (کندتر)
فشرده‌سازی ۸۰٪	۵۳/۸۱ ms	-۰/۶٪ (تقریباً خنثی)

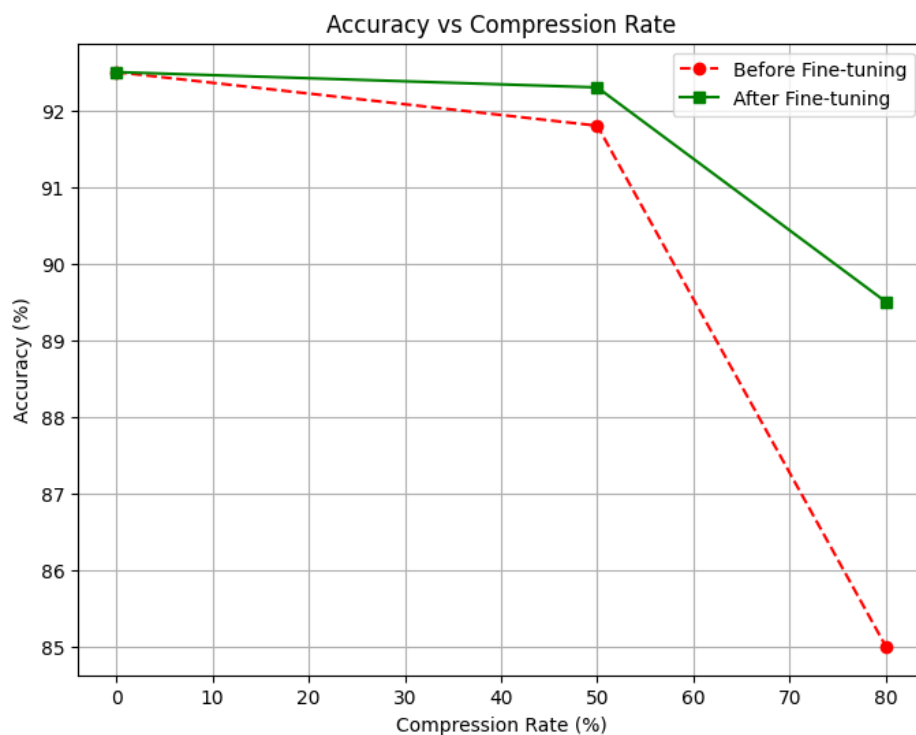
### ۳-۳-۳ تحلیل سربار محاسباتی

نتایج، یک پارادوکس رایج در فشرده‌سازی مدل را آشکار می‌کند که به «سربار دسته کوچک» (Small Batch Overhead) معروف است.

در فشردسازی ۵۰٪، کاهش تعداد عملیات ممیز شناور (FLOPs) برای جبران سربار معماری جدید کافی نبود. با جایگزینی یک لایه ( $W$ ) با دو لایه متوالی ( $A$  و  $B$ )، تعداد فراخوانی‌های کرنل و عملیات خواندن/نوشتن حافظه تقریباً دو برابر شد. در CPU، این سربار تأخیر بر صرفه‌جویی محاسباتی غالب شد و در نهایت زمان استنتاج افزایش یافت.

تنها در فشردسازی ۸۰٪، کاهش اندازه محاسبات (ابعاد ضرب ماتریسی) به اندازه‌ای شد که بتواند سربار لایه اضافی را جبران کند. با این حال، حتی در این حالت نیز مدل صرفاً هم‌سطح با خط مبنا عمل کرد (۵۳/۸۱٪ در برابر ۵۴/۱۴٪)، نه به‌طور معنادار سریع‌تر.

این نتایج نشان می‌دهد که فشردسازی مبتنی بر SVD روی یک لایه منفرد، بیشتر برای کاهش نیاز ذخیره‌سازی (Memory Footprint) مؤثر است تا بهبود تأخیر اجرا روی CPU، مگر آنکه نرخ فشردسازی بسیار بالا باشد یا این روش بر روی لایه‌های کانولوشنی محاسبات سنگین اعمال شود.



شکل ۳-۳: دقت در مقابل فشردسازی

## فصل ۴

# جمع‌بندی و نتیجه‌گیری و پیشنهادات

### ۴-۱ بحث

نتایج تجربی نشان‌دهنده رابطه‌ای پیچیده میان کاهش نظری مدل و عملکرد عملی در محیط‌های استقرار است. این بخش پیامدهای تحلیلی این یافته‌ها را بررسی می‌کند.

### ۴-۱-۱ پارادوکس تأخیر در محیط‌های استقرار

نتایج ما نشان داد که کاهش تعداد پارامترها لزوماً به استنتاج سریع‌تر منجر نمی‌شود. این ناهمخوانی را می‌توان به گلوگاه‌های خاص هر محیط سخت‌افزاری نسبت داد.

CPU (حساس به تأخیر): همان‌طور که در آزمایش‌ها مشاهده شد، CPU از سربار ترتیبی رنج می‌برد. جایگزینی یک عملیات بزرگ ماتریسی با دو عملیات کوچک‌تر، تعداد واكشی دستورالعمل‌ها و فراخوانی‌های کرنل را دو برابر می‌کند. برای اندازه‌های دسته کوچک، این سربار از کاهش عملیات حسابی (FLOPs) پیشی می‌گیرد و در نتیجه، زمان استنتاج افزایش می‌یابد.

GPU (مبتنی بر گذردهی): در محیط‌های به‌شدت موازی، شکستن ماتریس‌ها می‌تواند به پدیده «گرسنگی» منجر شود؛ جایی که هسته‌های GPU به‌طور کامل به‌کار گرفته نمی‌شوند، زیرا ماتریس‌های تجزیه‌شده آن‌قدر کوچک هستند که ظرفیت سخت‌افزار را اشباع کنند.

لبه/موبایل (محدود به پهنای باند حافظه): اگرچه در این مطالعه آزمایش نشد، تحلیل نظری نشان می‌دهد که این همان جایی است که فشردن‌سازی مبتنی بر SVD بیشترین مزیت را دارد. در دستگاه‌های موبایل، هزینه انرژی و زمان برای واكشی وزن‌ها از RAM اغلب از هزینه محاسبه بیشتر است. با کاهش



چشمگیر اندازه مدل (ذخیره‌سازی)، فشار روی پهنای باند حافظه کم می‌شود که احتمالاً به افزایش سرعت واقعی و صرفه‌جویی در باتری در استقرارهای لبه‌ای منجر خواهد شد.

## ۴-۱-۲ انتخاب لایه با استفاده از عدد شرطی

برای پرهیز از رویکرد آزمون‌وخطا در فشرده‌سازی همه لایه‌ها، عدد شرطی<sup>۱</sup> یک معیار دقیق برای انتخاب لایه‌ها فراهم می‌کند. این کمیت که به‌صورت نسبت بزرگ‌ترین به کوچک‌ترین مقدار منفرد تعریف می‌شود،

$$\kappa = \frac{\sigma_{\max}}{\sigma_{\min}},$$

«پایداری» یک لایه را کمی‌سازی می‌کند.

عدد شرطی بالا (بدشروط): متناظر با طیفی با «دُم سنگین» (کاهش سریع) است، همان‌گونه که در لایه‌های ResNet-18 مشاهده شد. این وضعیت نشان‌دهنده افزونگی بالاست و لایه را به گزینه‌ای عالی برای فشرده‌سازی تهاجمی تبدیل می‌کند.

عدد شرطی پایین (خوب‌شرط): متناظر با طیفی نسبتاً تخت است. این بدان معناست که لایه عملاً «تمام‌رتبه» بوده و از همه ابعاد به‌طور مؤثر استفاده می‌کند. فشرده‌سازی چنین لایه‌هایی به افت دقت فوری و غیرقابل جبران منجر خواهد شد.

## ۴-۲ نتیجه‌گیری

این گزارش امکان‌پذیری استفاده از تجزیه مقدار منفرد<sup>۲</sup> برای فشرده‌سازی معماری ResNet-18 را بررسی کرد. تحلیل‌ها تأیید می‌کنند که شبکه‌های عصبی عمیق دارای افزونگی کم‌رتبه قابل توجهی هستند. ما توانستیم ۵۰٪ از پارامترهای لایه کاملاً متصل نهایی را با افت دقت ناچیز (حدود ۰/۷٪) حذف کنیم و بخش عمده دقت از دست‌رفته در فشرده‌سازی ۸۰٪ را از طریق ریزتنظیم کوتاه‌مدت بازیابی نماییم.

با این حال، این مطالعه هشدار جدی درباره سرعت استنتاج ارائه می‌دهد. نتیجه می‌گیریم که فشرده‌سازی مبتنی بر SVD در درجه اول یک تکنیک بهینه‌سازی حافظه است، نه لزوماً یک روش بهینه‌سازی تأخیر، به‌ویژه در استقرارهای مبتنی بر CPU. برای پیاده‌سازی‌های آینده، توصیه می‌شود

<sup>۱</sup>(Condition Number)

<sup>۲</sup>(SVD)

لایه‌های محدود به حافظه (مانند لایه FC نهایی) در اولویت فشرده‌سازی قرار گیرند و لایه‌های محدود به محاسبه (مانند کانولوشن‌های اولیه) دست‌نخورده باقی بمانند تا ویژگی‌های بهینه تأخیر حفظ شود.

## منابع و مراجع

- [1] Golub, Gene H. and Van Loan, Charles F. Matrix Computations. Johns Hopkins University Press, Baltimore, 4 ed. , 2013.
- [2] Hastie, Trevor, Tibshirani, Robert, and Friedman, Jerome. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, 2 ed. , 2009.
- [3] Jolliffe, Ian T. Principal Component Analysis. Springer, New York, 2 ed. , 2002.

# Abstract

This report investigates the efficacy of Singular Value Decomposition (SVD) as a technique for compressing deep neural networks, specifically analyzing the ResNet-18 architecture. The study aims to evaluate the trade-offs between parameter reduction, model accuracy, and inference latency. Methodologically, the singular value spectra of both intermediate convolutional layers and the final fully connected layer were analyzed to determine intrinsic redundancy. Based on this spectral analysis, linear layers were replaced with rank- $k$  approximations at compression rates of 50% and 80%. Key findings indicate that ResNet-18 layers exhibit significant low-rank structures, allowing for substantial parameter removal with minimal initial impact on accuracy. While aggressive compression (80%) resulted in a notable performance drop, fine-tuning the model for short epochs successfully recovered the majority of lost accuracy. However, inference latency tests on CPU revealed a discrepancy between theoretical FLOPs reduction and practical runtime, where the overhead of sequential layer execution often negated speed gains. The report concludes by discussing the utility of the condition number as a metric for selecting compressible layers and the implications of deployment environments on compression strategies.

## Key Words:

Dimensionality Reduction, Feature Selection, Singular Value Decomposition (SVD), Recursive Feature Elimination (RFE), Semiconductor Manufacturing, Stability Analysis



**Amirkabir University of Technology**  
**(Tehran Polytechnic)**

**Department of Computer Science**

**M. Sc. Thesis**

## **Fifth CDM Project**

**By**

**Mohammad Sadegh Gholizadeh**

**Supervisor**

**Dr. Mahdi Ghatei**