دانشگاه صنعتي اميركبير
(پلی‌تکنیک تهران)
دانشکده ریاضی و علوم کامپیوتر

پروژه پنجم درس داده کاوی محاسباتی
علوم کامپیوتر، هوش مصنوعی و محاسبات نرم

فشرده سازی مدل های عمیق با تجزیه رتبه پایین و ارتقای کیفیت صنعتی

نگارش
هومن ذوالفقاری

استاد راهنما
دکتر مهدی قطعی

دی 1404

به نام خدا

تاریخ:

تعهدنامه اصالت اثر

اینجانب هومن ذوالفقاری متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتي اميركبير بوده و به دستاوردهاي ديگران كه در اين پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذكر گرديده است. اين پايان نامه قبلاً براي احراز هيچ مدرك هم‌سطح يا بالاتر ارائه نگرديده است.

در صورت اثبات تخلف در هر زمان، مدرك تحصيلي صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پيگيري قانوني خواهد داشت.

كليه نتايج و حقوق حاصل از اين پايان نامه متعلق به دانشگاه صنعتي اميركبير مي‌باشد. هرگونه استفاده از نتايج علمي و عملي، واگذاري اطلاعات به ديگران يا چاپ و تكثير، نسخه‌برداري، ترجمه و اقتباس از اين پايان نامه بدون موافقت كتبي دانشگاه صنعتي اميركبير ممنوع است.
نقل مطالب با ذكر مآخذ بلامانع است.

در صفحه تعهدنامه اصالت اثر، در قسمت بالا سمت چپ، تاریخ دفاع خود را جایگزین تاریخ نوشته شده کنید.

همچنین در صفحه تعهدنامه اصالت اثر، در خط اول، نام و نام خانوادگی خود را به صورت کامل با نام و نام خانوادگی نمونه، جایگزین کنید. در انتهای متن تعهد، در قسمت امضا نیز باید نام و نام خانوادگی کامل خود را وارد نماید.

هومن ذوالفقاری

امضا

چکیده

مدل‌های یادگیری عمیق علی‌رغم دقت بالا، معمولاً دارای تعداد زیادی پارامتر و هزینه محاسباتی قابل توجهی هستند که استفاده از آن‌ها را در محیط‌های عملیاتی و صنعتی محدود می‌کند. در این پروژه، از تکنیک‌های پیشرفته جبر خطی، به‌ویژه تجزیه مقادیر منفرد(SVD) ، برای فشرده‌سازی لایه‌های یک شبکه عصبی عمیق استفاده شده است. بدون تغییر در معماری کلی مدل، لایه‌های حجیم به تقریب‌های رتبه پایین تبدیل شده‌اند. نتایج نشان می‌دهد که با کاهش قابل توجه تعداد پارامترها، افت دقت ناچیز بوده و با استفاده از تنظیم دقیق(Fine-tuning) ، حتی بهبود دقت نیز حاصل می‌شود. همچنین، تأثیر فشرده‌سازی بر سرعت استنتاج در محیط CPU بررسی شده است.

صفحه

فهرست مطالب

Contents

فصل اول
مقدمه

مقدمه

لینک کد در گیتهاب:

https://github.com/hoomanzolfaghari84/Data-Mining-Course.git

با گسترش کاربردهای صنعتی یادگیری عمیق، نیاز به مدل‌هایی با مصرف حافظه کمتر و سرعت استنتاج بالاتر بیش از پیش احساس می‌شود. بسیاری از شبکه‌های عصبی عمیق دارای افزونگی پارامتری هستند و وزن‌های آن‌ها را می‌توان با تقریب‌های کم‌رتبه نمایش داد. تجزیه مقادیر منفرد (SVD)یکی از ابزارهای قدرتمند جبر خطی است که امکان تحلیل طیفی وزن‌ها و فشرده‌سازی مؤثر لایه‌های خطی را فراهم می‌کند.

هدف این پروژه، بررسی عملی امکان فشرده‌سازی یک مدل از پیش آموزش‌دیده بدون تغییر معماری کلی آن، و ارزیابی اثر این فشرده‌سازی بر دقت، تعداد پارامترها و سرعت استنتاج است.

فصل دوم
تحلیل طیفی وزن های مدل

تحلیل طیفی وزن های مدل

مدل و داده

در این پروژه، یک شبکه عصبی چندلایه (MLP) آموزش‌دیده بر روی دیتاست **Fashion-MNIST** مورد استفاده قرار گرفت. مدل شامل 4 لایه خطی اصلی است که یکی از لایه‌های میانی (512→512) و لایه نهایی (10←512) برای تحلیل انتخاب شدند.

تحلیل مقادیر منفرد

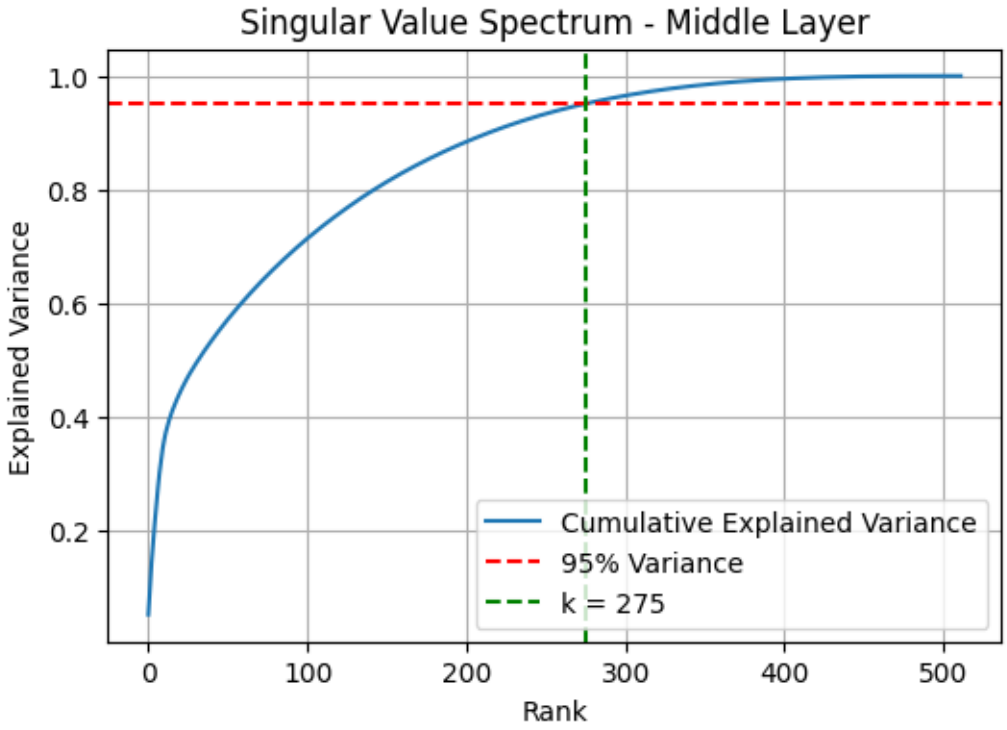
با اعمال تجزیه SVD روی ماتریس وزن‌ها، طیف مقادیر منفرد و واریانس تجمعی آن‌ها محاسبه شد.

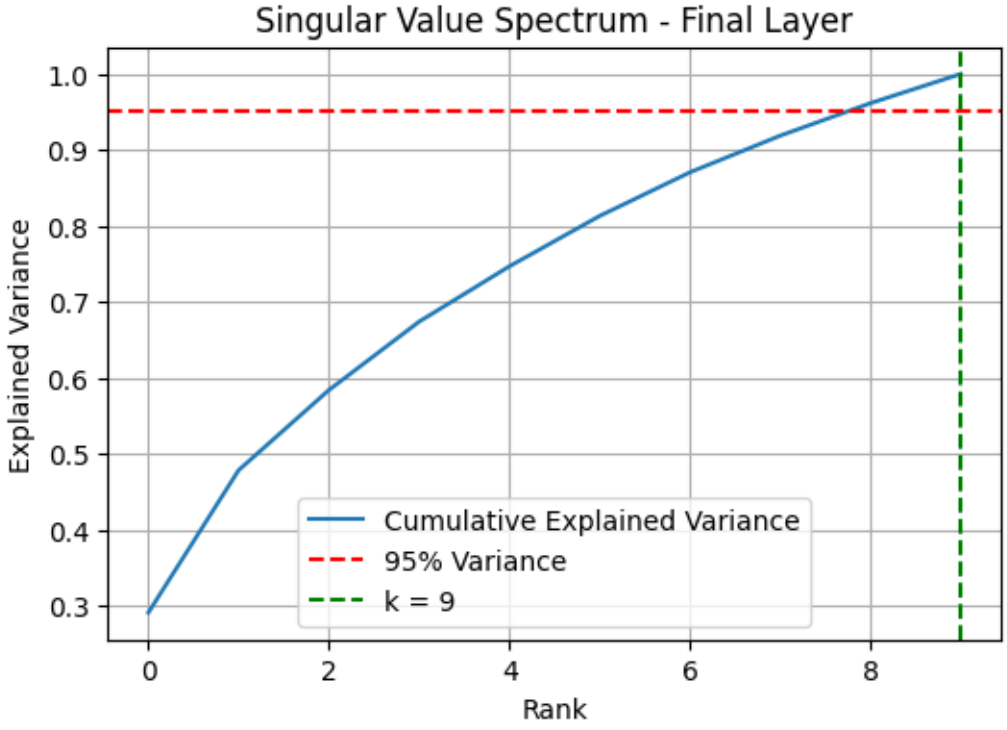
نتایج به‌دست‌آمده:

- **لایه میانی**:
  - رتبه لازم برای حفظ ۹۵٪ واریانس : **275**
  - درصد مقادیر منفرد استفاده‌شده: 53.71%
- **لایه نهایی**:
  - رتبه لازم برای حفظ ۹۵٪ واریانس : **9**
  - درصد مقادیر منفرد استفاده‌شده: 90.00%

تحلیل

این نتایج نشان می‌دهند که لایه‌های میانی دارای افزونگی ساختاری بالاتری هستند و گزینه مناسب‌تری برای فشرده‌سازی محسوب می‌شوند، در حالی که لایه نهایی به‌دلیل ابعاد کوچک‌تر، ظرفیت فشرده‌سازی محدودی دارد.


Singular Value Spectrum - Middle Layer
Explained Variance
Rank
Cumulative Explained Variance
95% Variance
k = 275
1.0
0.8
0.6
0.4
0.2
0
100
200
300
400
500


Singular Value Spectrum - Final Layer
Explained Variance
Rank
Cumulative Explained Variance
95% Variance
k = 9
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8

فصل سوم
پیاده سازی عملی فشرده سازی

پیاده سازی عملی فشرده سازی

روش فشرده‌سازی

لایه خطی انتخاب‌شده با استفاده از SVD به دو لایه متوالی با رتبه k جایگزین شد. این کار بدون تغییر عملکرد ریاضی کلی لایه و تنها با یک تقریب کم‌رتبه انجام گرفت.

سناریوهای آزمایشی

مدل در سه حالت بررسی شد:

- بدون فشرده‌سازی
- فشرده‌سازی با ٪۵۰ رتبه
- فشرده‌سازی با ٪۸۰ رتبه

دقت اولیه مدل

حالت مدل	دقت(%)
بدون فشرده‌سازی	86.92
فشرده‌سازی ٪۵۰	86.93
فشرده‌سازی ٪۸۰	86.98

نتایج نشان می‌دهد که حتی بدونFine-tuning ، فشرده‌سازی منجر به افت دقت قابل توجهی نشده است. (در اینجا بهبود یافته، گویا مدل اندکی اورفیت کرده است)

مقایسه تعداد پارامترها

حالت مدل	تعداد پارامتر
بدون فشرده‌سازی	932362
فشرده‌سازی ٪۵۰	932362
فشرده‌سازی ٪۸۰	774666

کاهش محسوس تعداد پارامترها، کارایی حافظه مدل را بهبود می‌دهد.

فصل چهارم
ارزیابی دقت و سرعت

ارزیابی دقت و سرعت

تنظیم دقیق(Fine-tuning)

مدل‌های فشرده‌شده به مدت ۳ ایپاک Fine-tune شدند. پس از تنظیم دقیق، دقت‌ها به شکل زیر بهبود یافت:

حالت مدل	دقت بعد از FT (%)
بدون فشرده‌سازی	86.91
فشرده‌سازی ۵۰٪	88.75
فشرده‌سازی ۸۰٪	88.66

این افزایش دقت نشان می‌دهد که فشرده‌سازی می‌تواند به‌عنوان نوعی منظم سازی (Regularization)عمل کند.

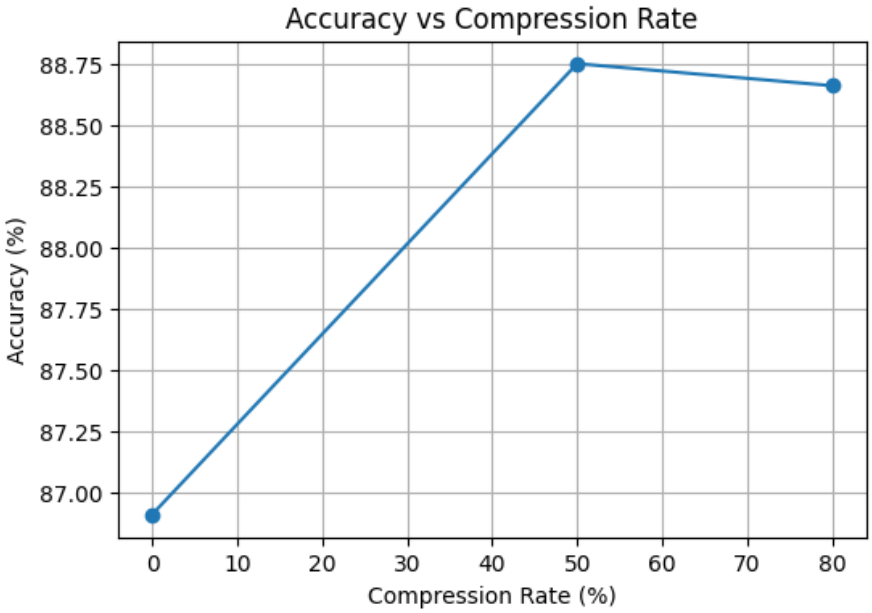
سرعت استنتاج روی CPU

حالت مدل	زمان استنتاج(ms)
بدون فشرده‌سازی	0.29
فشرده‌سازی ۵۰٪	0.30
فشرده‌سازی ۸۰٪	0.26

تفاوت زمان‌ها ناچیز است که نشان می‌دهد افزایش تعداد لایه‌ها اثر منفی قابل توجهی بر سرعت CPU در این مقیاس نداشته است.

نمودار دقت بر حسب نرخ فشرده‌سازی

نمودار رسم‌شده نشان می‌دهد که تا نرخ‌های بالای فشرده‌سازی، دقت مدل حفظ شده و حتی پس از Fine-tuning افزایش یافته است.



فصل پنجم
سوالات تحلیلی

سوالات تحلیلی

تأثیر تأخیر ناشی از افزایش تعداد لایه‌ها

در محیط‌هایی مانندGPU ، افزایش تعداد لایه‌های کوچک ممکن است به‌دلیل parallelism تأثیر کمی داشته باشد، اما در CPU و سیستم‌هایembedded ، تأخیر ناشی از فراخوانی لایه‌ها می‌تواند غالب شود. بنابراین فشرده‌سازی همیشه به افزایش سرعت منجر نمی‌شود و به معماری سخت‌افزار وابسته است.

محیط	تأثیر افزایش تعداد لایه‌ها	توضیح
CPUتک‌هسته‌ای	بسیار محسوس	هر عملیات محاسباتی پشت سر هم انجام می‌شود. افزایش لایه‌ها با زمان استنتاج (تقریبا خطی) رشد می‌کند
GPU	کمتر محسوس	پردازش موازی روی هزاران هسته انجام می‌شود، ولی همچنان لایه‌های بزرگ (مثل Conv با فیلترهای زیاد) باعث تاخیر می‌شوند.
تلفن همراه / Embedded		حتی چند لایه اضافی به دلیل محدودیت حافظه و توان پردازشی می‌تواند زمان استنتاج را به طور قابل توجهی شدید افزایش دهد.

کاهش رتبه یا فشرده‌سازی لایه‌ها می‌تواند تأخیر را کاهش دهد، چون تعداد پارامتر و FLOPs کم می‌شود.

نقش عدد وضعیت در انتخاب لایه‌های مستعد فشرده‌سازی

عدد وضعیت (Condition Number) معیاری از حساسیت ماتریس نسبت به اغتشاشات است. لایه‌هایی با عدد وضعیت بزرگ و افت سریع مقادیر منفرد، معمولاً افزونگی بالایی دارند و گزینه‌های مناسبی برای فشرده‌سازی کم‌رتبه محسوب می‌شوند.

عدد وضعیت (Condition Number) برای یک ماتریس W به صورت زیر تعریف می‌شود:

$$\kappa(W) = \frac{\sigma_{\max}}{\sigma_{\min}}$$

$\sigma_{\max}$و $\sigma_{\min}$ بزرگ‌ترین و کوچک‌ترین مقادیر منفرد هستند. عدد وضعیت نشان می‌دهد که **ماتریس تا چه حد «نزدیک به تنزل رتبه» است** و چقدر حساس به نویز/خطا است.

لایه با عدد وضعیت کوچک:

- مقادیر منفرد بزرگ و کوچک به صورت یکنواخت پخش شده‌اند.
- تقریب رتبه پایین باعث از دست رفتن اطلاعات مهم می‌شود ← کمتر مناسب فشرده‌سازی.

لایه با عدد وضعیت بزرگ:

- ماتریس دارای چند مقدار منفرد بزرگ و بقیه نزدیک صفر است.
- این لایه **مستعد فشرده‌سازی رتبه پایین** است، چون می‌توان بیشتر مقادیر کوچک را حذف کرد بدون کاهش دقت مدل.

فصل ششم
نتیجه گیری

نتیجه گیری

در این پروژه نشان داده شد که با استفاده از تجزیه رتبه پایین مبتنی بر SVD می‌توان مدل‌های یادگیری عمیق را بدون تغییر معماری کلی، به‌طور مؤثری فشرده‌سازی کرد. نتایج تجربی حاکی از کاهش تعداد پارامترها، حفظ یا حتی بهبود دقت پس ازFine-tuning ، و عدم افزایش محسوس زمان استنتاج هستند. این رویکرد، راهکاری عملی و صنعتی برای بهینه‌سازی مدل‌های عمیق در محیط‌های با محدودیت منابع محسوب می‌شود.

منابع و مراجع

[1] این پروژه نیاز کمی به منابع داشت و بیشتر از دانش خودم استفاده کردم

[2] https://docs.pytorch.org/

[3] https://en.wikipedia.org/wiki/Low-rank_approximation

[4] https://openaccess.thecvf.com/content_CVPR_2020/papers/Idelbayev_Low-Rank_Compression_of_Neural_Nets_Learning_the_Rank_of_Each_CVPR_2020_paper.pdf

[5] منابع اصلی درس

Amirkabir University of Technology
(Tehran Polytechnic)

… Department …

MSc or PhD Thesis

Title of Thesis

By
Name

Supervisor
Dr.

Advisor
Dr.

Month & Year