Make Large Language Models More Efficient by Compressing the Low-rank Matrices

Pingwei Sun

BDT Program, Hong Kong University of Science and Technology

psunah@connect.ust.uk

Abstract

LLMs like ChatGPT exhibit impressive capabilities but face challenges due to their extensive parameter sizes. Although numerous experiments have been conducted to compress pretrained models, most of them are centered around BERT-like models. To explore efficient compression methods for LLMs, the project focuses on low-rank matrices in generative large language models. We will conduct a two-step experiment involving structural analysis and innovative compression techniques. This investigation aims to strike a balance between model size and performance. If time permits, optimizations for HPC will be considered.

1 Introduction

Large language models like ChatGPT have demonstrated remarkable capabilities across various domains. However, the extensive parameter size of these models has posed challenges in training and deployment.

Since pretrained models were introduced to the NLP field, the research on compressing has started to overcome its weakness of high computational cost. The current methods mainly fall into four directions (Xu and McAuley, 2022): **(1) Knowledge Distillation,** which is commonly applied in the industry, always takes more training steps (Sanh et al., 2019) and the student performance highly depends on hyper-parameters and initial distributions (Turc et al., 2019). **(2) Pruning** can maintain the original performance well by removing the redundant parameters, but the irregular structure produced by unstructured pruning may disable the hardware acceleration func. **(3) Quantization** mainly aims at storage and memory efficiency. It is also widely used in industry because of its convenience, especially the PTQ strategy making LLMs available on edge devices after simple compression of weights but at the cost of impact on accuracy performance. **(4) Low-Rank Factorization** focuses on the low-rank matrices in networks. Though the increase in parameters is necessary for qualitative changes in pretrained models (Frankle and Carbin, 2019), there is also a consensus of redundancy in the model weights (Sainath et al., 2013). It facilitates the techniques for compression through extracting intrinsic dimensions in low-rank matrices, potentially offering insights into the explainability of models to a certain extent (Zhang et al., 2023b).

Due to their impressive capabilities and demanding computational resource requirements, the above compressing methods have also been explored on LLMs (Zhu et al., 2023). In the field of knowledge distillation, works like MINILLM (Gu et al., 2023) have been conducted to transfer knowledge to small language models. Others are also exploring methods to preserve the emergent abilities during the process (Huang et al., 2022). In the aspect of pruning, Both SparseGPT (Frantar and Alistarh, 2023) and LLM-Pruner (Ma et al., 2023) have achieved impressive compression ratios and preserved the capabilities of LLMs through structured and unstructured pruning strategies, respectively. As for quantization, supports from frameworks (TensorRT, llama.cpp, MNN, etc.) has made it possible for 7B-models to run on smartphones.

In contrast, research on low-rank factorization for LLM compression seems to be in its early stages, and there are few remarkable works (Zhang et al., 2023a). In this project, our focus will be on the techniques of compressing low-rank matrices. We will analyze the matrices and combine the existing methods to propose a specific compression approach for LLMs, making a balance between model size and performance trade-offs.

Categoreis	Paper	Compression ratio	Energy retained (wo/wt re-FT)
Numerical	ALBERT	10%	-
	FWSVD	40% (transformer blocks only)	70% / 97%
	TFWSVD	40% (transformer blocks only)	80% / 99%
	Shapeshifter	80% - 90%	95%
	KnGPT2	33%	95% / 99%
	TensorGPT	33%	-
Combination	LPAF	84%	46% / 95%
	LoRAPrune	90%	90%

Table 1: Comparison of **Matrix Compression Techniques**.

2 Related Work

Low-Rank factorization. Singular value decomposition (SVD), as a commonly used method, has been applied to a series of works, such as ELMo (Winata et al., 2019) and ALBERT (Lan et al., 2020). Though being widespread, SVD tends to result in poor performance when the compression ratio increases. Recently, numerical optimization methods like Kronecker product representation in Shapeshifter (Panahi et al., 2021), KnGPT2 (Edalati et al., 2022) and Fisher-Weighted SVD (Hsu et al., 2022; Hua et al., 2022) have been proposed, but they are not time-efficient. Besides, the method of tensor-train has also been tested on embedding layers of LLMs (Xu et al., 2023) and further works are expected for the other structures.

Combined techniques. However, the weight matrices of transformers are not usually low-rank (Yu and Wu, 2023), which defeats the numerical factorization methods. To achieve high-level compression, researchers attempt to combine low-rank factorization and unstructured pruning into **LPAF** (Ren and Zhu, 2023). In LPAF, a 3-step framework is proposed for language model compression. By first unstructured pruning weights and then decomposing them into submatrices, the framework successfully overcomes the obvious degradation when the compression ratio is high in SVD, and gets rid of the dependence of unstructured pruning on specific hardware systems.

3 Method

We will explore how to compress LLMs at a high ratio without causing significant degradation by the following two steps.

Step 1: Structure Analysis. As the structural analysis of generative LLMs is still in its initial stage, in order to identify the bottlenecks, we need to sample the stacked decoders and analyze the characteristics of those inside matrices (such as rank, variances, sparsity, etc.). It is crucial for the design of subsequent approaches.

Step 2: Novel compression method. It is pointed out in LPAF that fine-tuned language models are full-rank, which causes information loss when doing SVD approximation directly. To achieve a trade-off between model size and performance, a compression strategy should be specified based on the analysis result in Step 1, and such a strategy may be combined with other techniques, like unstructured pruning, inspired by SparseGPT.

4 Experiment

4.1 Metrics

In order to accurately measure the capability of the above method and its performance at deployment, we evaluate it in terms of parameter amount, compression ratio, and inference time.

4.2 Benchmarks

MMLU (Hendrycks et al., 2021) and Common Sense QA (Talmor et al., 2019) datasets are currently recognized benchmarks that provide a comprehensive assessment of LLMs.

As a popular open-source model, LLaMA can serve as a preferable baseline. The latest version, LLaMA2, has shown competitive performance following GPT-4 in multiple evaluation tasks. However, due to constraints in computational resources, we will conduct experiments using the 7B-model.

5 Timeline

5.1 Week 1 to 2:

Revise research proposal. Prepare environment.

5.2 Week 3 to 5:

Reproduce the baseline and analysis of the results.

5.3 Week 6 to 10:

Build codes and design experiments.

5.4 Week 11 to 12:

Extra experiments (upon the previous results).

5.5 Week 13 to 14:

Write and revise the final report, submission before the examination week starting from 7th Dec.

6 Grading Criteria

Considering the requirements of MSBD5014 and this research-oriented topic, the specific grading criteria consist of the following components

- **A detailed proposal: 10%**
- **Group meetings and reports: 30%**
- **Final research report: 60%**

References

Ali Edalati, Marzieh Tahaei, Ahmad Rashid, Vahid Nia, James Clark, and Mehdi Rezagholizadeh. 2022. Kronecker decomposition for GPT compression. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 219–226, Dublin, Ireland. Association for Computational Linguistics.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *International Conference on Learning Representations,International Conference on Learning Representations*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. 2022. Language model compression with weighted low-rank factorization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Ting Hua, Yen-Chang Hsu, Felicity Wang, Qian Lou, Yilin Shen, and Hongxia Jin. 2022. Numerical optimizations for weighted low-rank estimation on language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1404–1416, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yukun Huang, Yanda Chen, Zhou Yu, and Kathleen McKeown. 2022. In-context learning distillation: Transferring few-shot learning ability of pre-trained language models. *arXiv preprint arXiv:2212.10670*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *arXiv preprint arXiv:2305.11627*.

Aliakbar Panahi, Seyran Saeedi, and Tom Arodz. 2021. Shapeshifter: a parameter-efficient transformer using factorized reshaped matrices. In *Advances in Neural Information Processing Systems*, volume 34, pages 1337–1350. Curran Associates, Inc.

Siyu Ren and Kenny Q. Zhu. 2023. Low-rank prune-and-factorize for language model compression.

Tara N. Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. 2013. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models.

arXiv: Computation and Language,arXiv: Computation and Language.

Genta Indra Winata, Andrea Madotto, Jamin Shin, Elham J Barezi, and Pascale Fung. 2019. On the effectiveness of low-rank matrix factorization for lstm model compression. *ArXiv preprint*, abs/1908.09982.

Canwen Xu and Julian McAuley. 2022. A survey on model compression and acceleration for pretrained language models.

Mingxue Xu, Yao Lei Xu, and Danilo P. Mandic. 2023. Tensorgpt: Efficient compression of the embedding layer in llms based on the tensor-train decomposition.

Hao Yu and Jianxin Wu. 2023. Compressing transformers: Features are low-rank, but weights are not!

Mingyang Zhang, Haozhen Haozhen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2023a. Pruning meets low-rank parameter-efficient fine-tuning.

Zhong Zhang, Bang Liu, and Junming Shao. 2023b. Fine-tuning happens in tiny subspaces: Exploring intrinsic task-specific subspaces of pre-trained language models.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models.