MSBD5014A Final Report
LEGO: Efficiently Generate Optimized LLMs through Low-Rank Decomposition and Assembly

Pingwei Sun
Stu Id. 20983818
BDT program, HKUST
psunah@connect.ust.hk

Abstract

As Large Language Models (LLMs) become widely applied across various domains, the associated deployment costs have gained increased attention. Previous research has demonstrated significant sparsity within models based on the transformer architecture, presenting opportunities for efficient model compression. In this paper, we have surveyed current LLM compression methods and delved deeper into the sparsity aspect of LLMs. Specifically, preliminary experiments are conducted to validate the low-rank characteristics within LLMs. Leveraging this property, we explore using matrix decomposition methods previously proven effective on BERT-like models for compressing LLMs. Our experiments identified certain limitations like computational resources and a novel compression approach based on low-rank approximation is proposed to achieve smaller, domain-specific LLMs effectively.

1 Introduction

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

Therefore, many methods for parameter-efficient fine-tuning (PEFT) and model compression have been proposed. Approaches such as P-tuning (Liu et al., 2023), LoRA (Hu et al., 2022) are employed to adapt the general-purpose LLMs to downstream tasks. Typically, the common practice involves updating existing model parameters or adding new parameters without optimizing the model size for deployment.

In the area of model compression, a large number of techniques previously applied to medium-sized pre-trained language models (PLMs) are now being ported to LLMs. For example, methods such as pruning and quantization are being used. Quantization is the most widely applied one due to its convenience and minimal performance degradation. However, quantization methods with less than 8 bits often do not reach their full potential due to the dependence on framework and hardware support.

In contrast, low-rank approximation is a more generalized solution. Singular Value Decomposition (SVD), a matrix decomposition algorithm, has been explored on BERT-like pre-trained language models (PLMs) and has yielded effective compression methods such as FWSVD (Hsu et al., 2022). However, the method typically aims for specific downstream tasks, requiring backpropagation or refine-tuning, which significantly burdens obtaining smaller, generally applicable LLMs.

In our experiments, FWSVD has proved effective for 50% compression on LLMs, but improvements are still necessary. Based on the generally low-rank features observed in the LLM, we propose a novel weighting strategy for SVD named **LEGO**. It uses the **L**ow-rank approximation to **E**ffectively **G**enerate an **O**ptimized LLM, decomposing a large model into a base model with general-purpose language modeling capabilities and specific modules customized for a range of downstream tasks. With this approach, a domain-specific LLM can be accomplished through component assembly rather than adding extra parameters as decorations or conducting updates in place.

2 Related Work

2.1 Effective Compressing Techniques

The current compressing 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's 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 function. 3) Quantization mainly aims at storage and memory efficiency. It is also widely used in industry because of its convenience, especially the post-training quantization strategy making PLMs available on edge devices after simple compression of weights but at the cost of a decrease in model performance. 4) Low-rank Decomposition facilitates the techniques for compression through low-rank approximation, decomposing one high-dimensional weight matrix into two matrices with low ranks. Though the parameter increase is necessary for qualitative changes in pre-trained models (Frankle and Carbin, 2019), there is also a consensus of redundancy in the model weights.

The ALBERT (Lan et al., 2020) leverages it to compress the embedding layer. Then FWSVD (Hsu et al., 2022) and TFWSVD (Hua et al., 2022) are proposed to extend it to the feed-forward layers of the transformer block.

2.2 Compression of LLM

Due to the impressive performance of large models and their high deployment costs, the methods mentioned above have also been transposed onto LLMs (Zhu et al., 2023). We investigate some popular approaches and compare their pros and cons in Figure 1.

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 LLMPruner (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, being the most practical and currently top-performing method approaches like GPTQ (Frantar et al., 2023), AWQ (Lin et al., 2023), have been widely adopted in numerous open-source projects for efficient inference and deployment of models. It is also compatible with other techniques. However, when compression rates are high, this method is constrained by the computing system and may not fully realize its acceleration potential.

In contrast, research on low-rank decomposition for LLM compression seems to be in its early stages, and there are a few remarkable works.

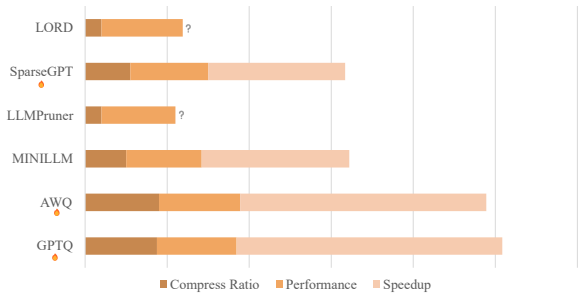

Figure 1: A rough comparison of compression methods, with accurate compression ratios. The benchmarks used for evaluating across different methods are not entirely consistent. Some methods do not report speedup in their papers, and stay closed source.

LoRD (Kaushal et al., 2023) decomposes from the low-rank output features to obtain a Code LLM, achieving a compression ratio of 20% with less than 1% increase in the metric of perplexity (PPL).

Some recent studies (Lin et al., 2023; Zhao et al., 2023) have found that while the weight matrix itself may not be low-ranked, certain sensitive parameters exist within it. Preserving these parameters intact is crucial for post-compression performance. Interestingly, the distribution of these sensitive weights tends to be low-ranked, either concentrated along rows or columns, which favors compression methods based on low-rank decomposition.

3 Effectively Generate Optimized LLM by Low-rank Approximation

The technique of low-rank approximation can be implemented in the following form:

$$Y = XW + b \approx (XL_1)L_2 + b \tag{1}$$

Where W is the original parameter matrix in shape $N \times M$ with rank r and can be decomposed to USV^T. The matrix S is a diagonal matrix with r singular values.

We can obtain the compressed structure by selecting k ranks and assigning L_1 as $(U\sqrt{S})_{[:,:k]}$ and L_2 as $(\sqrt{S}V^T)_{[:k,:]}$, achieving the compression ratio of $1 - k(M+N)/MN$. To achieve a high compress ratio with a small error the original matrix W is expected to be low-ranked (most singular values are 0) or possesses a relatively concentrated set of singular values. Therefore we analyze different types of matrices in the LLM to assess the feasibility of the low-rank approximation method and to select a suitable design.

3.1 Features of Low-Rank in LLMs

Weight Matrix. The analysis is applied to the weight matrices of LLaMA2-7B, and the results are illustrated in Figure 2. For comparison, the outcomes of the BERT model are included.

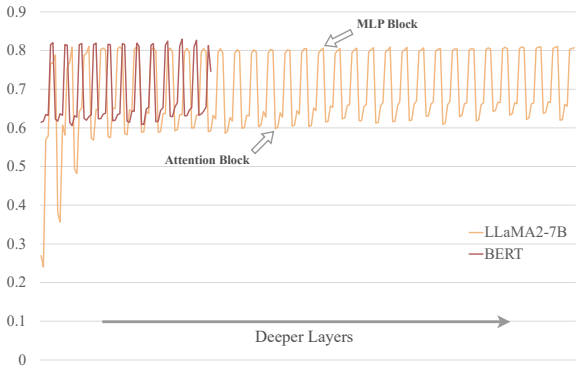

Figure 2: The percentage of singular values contributing to ninety percent of the total sum of singular values.

It is evident that the weight matrices of LLM and PLM are both dense, taking more than half of singular values to achieve 90% of the sum, and the distribution of singular values has no obvious concentration phenomenon. This means that directly compressing the weight matrix using SVD will produce large errors. Furthermore, we also observe that the matrices in the attention block i.e. $W_{Q,K,V}$ appear to exhibit a more concentrated singular value distribution than those in the MLP block.

Output Feature. Works like Yu and Wu, 2023 have verified that the feature map after the activation function is low-ranked in PLMs and Transformers in the computer vision field. Therefore, we profile the activation values of the LLM, and detailed results are presented in Appendix A. In a contemporaneous work, LoRD adopts the method of Atomic Feature Mimicking (AFM) (Yu and Wu, 2023) to achieve compression of large-scale code models through low-rank approximation, thus avoiding the computational intensity of full-parameter backpropagation.

Domain Specific Parameters. Although there are few low-rank features in the parameter matrix that support the use of SVD, it is common to observe them during domain-specific LLM fine-tuning, which is also a key reason why fine-tuning methods such as LoRA are effective.

3.2 Compress by Low-Rank Approximation

The previous analysis suggests that achieving low-error compression models through vanilla SVD is not feasible. Therefore, the key challenge lies in assessing the fine-graded importance of items within the matrix and then representing the crucial ones by matrices with a limited number of ranks.

FWSVD (Hsu et al., 2022), as the current state-of-the-art attempt in this direction, faces certain challenges when attempting to be applied to LLMs.

- Collecting fisher information (gradient) for all parameters on downstream datasets is computationally expansive.
- Since LLMs are well pretrained, the important parameters may be miss-weighted for their relatively stable gradients.

A recent work (Zhao et al., 2023), which fine-tunes LLMs on multiple languages has also proved that parameters that remain stable during fine-tuning may be crucial for the performance, they call them Core Linguistic Regions.

Therefore, we propose a novel criterion for assessing the importance of items within a matrix. This criterion integrates LoRA matrices obtained from fine-tuning the model across multiple domains. Parameters with minimal changes in values during fine-tuning will be assigned greater weights during the following SVD process. Details are shown in Algorithm 1[1].

Algorithm 1: LoRA Weighted SVD

Input: Weight matrix $A \in R^{mn}$ in LLM

1 $Finetune\ on\ domains\ for\ LoRA\ l_i$
2 $Init\ zero\ matrices\ W$
3 **for** $l = l_1, \ldots, l_i$ **do**
4 $\quad extract\ W_l^1\ and\ W_l^2\ from\ l$
5 $\quad W+ = \left|W_l^1 * W_l^2\right|$
6 **end**
7 $W_{col} = \sqrt{mean(W, dim = 1)}$
8 $W_{row} = \sqrt{mean(W, dim = 0)}$
9 $USV^T = Algorithm_{svd}(W_{col} * A * W_{row})$
10 $L_1 = (U\sqrt{S})_{[:,:k]} \quad L_2 = (\sqrt{S}V^T)_{[:k,:]}$
11 **return** $L_1/W_{col},\ L_2/W_{row}$

[1]Calculation in steps 7-9 is reported to work better than multiplying row-consistent W with A in Guo et al., 2023.

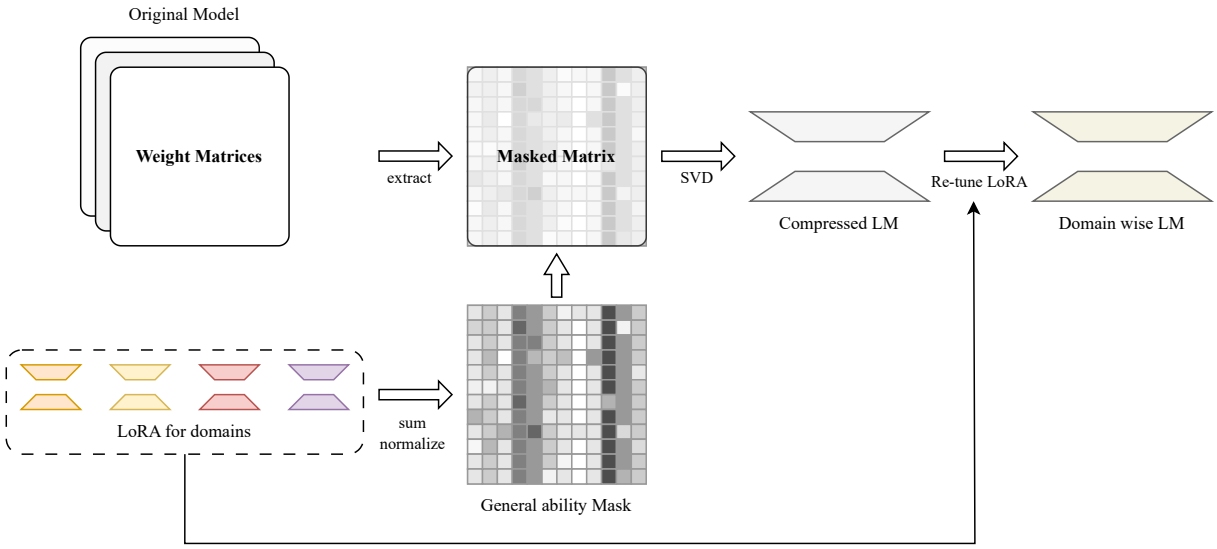

Figure 3: Illustration of the LEGO workflow. It is applied to all parameter matrices in LLMs, excluding the embedding layers.

3.3 Rebuild Domain Specific LLMs

Since we have the base model and domain-specific LoRA modules, obtaining a compressed model for a specific domain involves combining the base with LoRA and further adjusting the LoRA modules to compensate for errors introduced during the low-rank approximation process. Figure 3 illustrates the whole compressing flow.

4 Experiment

Due to time constraints and my previous leadership roles in both MSBD5018 and MSBD6000N group projects, only preliminary experiment results have been obtained. These results will serve as the baseline for subsequent experiments.

The coding of LEGO is finished, while the benchmark and test code are in progress. The following experiments will start in 3 weeks.

4.1 Preliminary Experiments.

Metric In the evaluation of the base model, we focus only on its language modeling capabilities. Therefore, in our preliminary experiments, we choose PPL as the evaluation metric and utilize the Wikitext-2 (Merity et al., 2016) dataset.

Implement details The dense model and the full-rank SVD model are tested to align with other works, indicating that the implementation of the decomposition is reasonable. We conduct experiments for the model with and without 8-rank LoRA fine-tuning on the training set at three different compression rates. Results are shown in Table 1.

5 Analysis

From the results of the SVD method, it is clear that the performance of the compressed model does not show a significant correlation with the retention ranking. This confirms that ordinary SVD is not suitable for low-rank approximation.

In contrast, weighted SVD demonstrates potential. At a compression ratio of 50%, the PPL of the fine-tuned model decreases significantly, indicating that the weighted SVD retains more domain-specific patterns within the matrix. However, there is still a gap in language modeling capabilities compared to LLMs.

As analyzed in § 3.2, this difference may be due to the critical parameters of small gradients not being well preserved during SVD, thereby compromising the model's capabilities.

6 Conclusion and Future Work

LEGO. The low-rank approximation has the advantage of generality and practicality as a compression method. However, there is limited research applying it to compressing LLMs. Methods shown to be effective for medium-sized PLMs fail to transfer to LLMs without incurring losses. In light of this, we propose a solution called LEGO, which decomposes the model into a base model and certain domain-specific LoRA modules. Experiments are underway to validate its performance.

Comp ratio	LLaMA2-1.3B				LLaMA2-7B	
	SVD		FWSVD		SVD	
Dense	8.13		-		5.47	
Full rank	8.12		-		5.47	
	w/o ft	w/ ft	w/o ft	w/ ft	w/o ft	w/ ft
50%	342357.34	4500.30	640.38	61.02	49486.63	1203.74
80%	69208.00	2519.09	57971.49	770.69	28181.95	1433.54
90%	69378.82	20029.31	95004.77	1504.05	130991.09	1527.52

Table 1: SVD and FWSVD baseline: values of PPL on Wikitext-2 with sequence length of 2048.

Workload.

- Analysis of matrices in the LLM.
- Propose LEGO for compression of LLM with low-rank approximation.
- Reproduce FWSVD (official code not released) and conduct preliminary experiments.
- Implementation of LEGO and to be evaluated.

Future work. The core functions of the LEGO have been implemented. This will be followed by a series of experiments after setting up a multi-GPU training environment with DeepSpeed and preparing several domains to be tested.

Plans include performance evaluation of the LEGO method in Language Modeling and other domains. We will provide further comparisons of current compression methods in terms of speedup and model performance and will explore the combination of LEGO and quantization methods.

References

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers.

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

Han Guo, Philip Greengard, Eric P. Xing, and Yoon Kim. 2023. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning.

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.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. 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*, abs/2212.10670.

Ayush Kaushal, Tejas Vaidhya, and Irina Rish. 2023. Lord: Low rank decomposition of monolingual code llms for one-shot compression.

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.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, Chuang Gan, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

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

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 preprint*, abs/1908.08962.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks.

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

Hao Yu and Jianxin Wu. 2023. Compressing transformers: Features are low-rank, but weights are not! In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 11007–11015. AAAI Press.

Jun Zhao, Zhihao Zhang, Yide Ma, Qi Zhang, Tao Gui, Luhui Gao, and Xuanjing Huang. 2023. Unveiling a core linguistic region in large language models.

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

A Study on Output Features

Yu and Wu, 2023 points out the phenomenon of low-rank features in transformer-based models, indicating that the feature maps obtained after activation functions exhibit low-rank properties.

We conduct profiling of intermediate results during LLM inference and validate the phenomenon. Results from selected layers are shown in Figure 4 and 5. The model undergoes forward propagation on the Wikitext dataset, and three regions of the hidden state are extracted and visualized along with their statistical values.

Initial tokens. In the dimension of sequence length (along the vertical axis), the hidden states of tokens exhibit relatively similar distributions, except for the initial few tokens. It is called attention sink (Xiao et al., 2023), which has been utilized to enhance the capability of LLM in generating long-form textual content.

Low-rank approximation based on output features. The AFM constructs a symmetric matrix using the output features and applies eigenvalue decomposition. The calculation can be expressed in the following form:

$$\mathbb{E}\left[yy^T\right] - \mathbb{E}[y]\mathbb{E}[y]^T = \hat{Q}\hat{\Lambda}\hat{Q}^T \tag{2}$$

$\hat{Q}$ and $\hat{\Lambda}$ stands for eigenvectors and eigenvalues respectively. Hence the $\hat{Q}_{:,:r}\hat{Q}^T_{:,:r} \approx I$, the linear function compressed to r-rank can be rewritten as the following one:

$$Y \approx \hat{Q}_{:,:r}\hat{Q}^T_{:,:r}WX + \hat{Q}_{:,:r}\hat{Q}^T_{:,:r}b \tag{3}$$

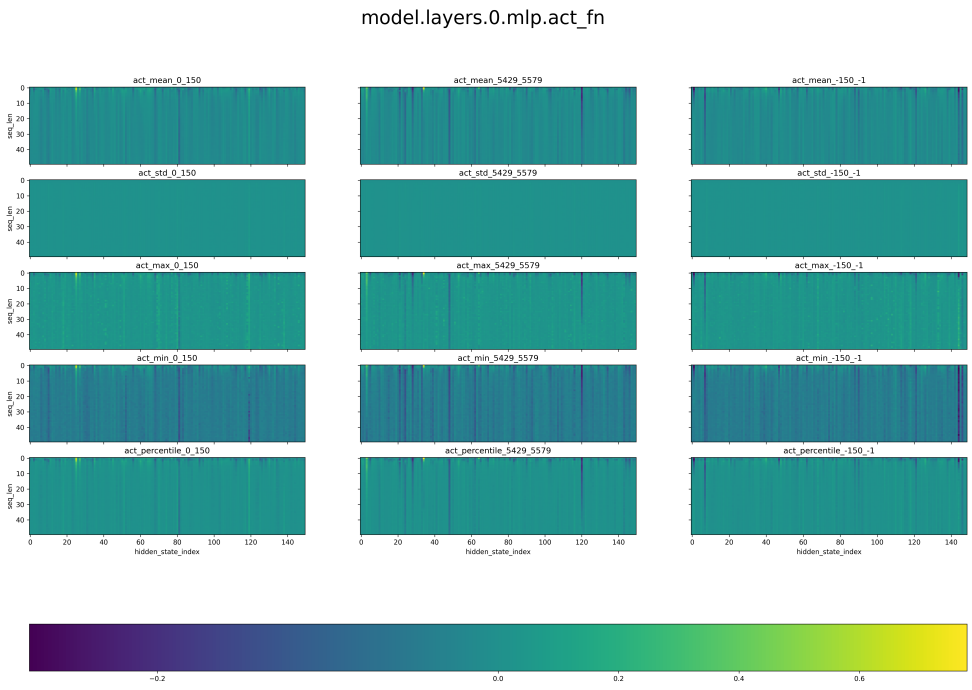

Figure 4: Profile of the feature map in the 1-st layer of LLaMA2-7B.

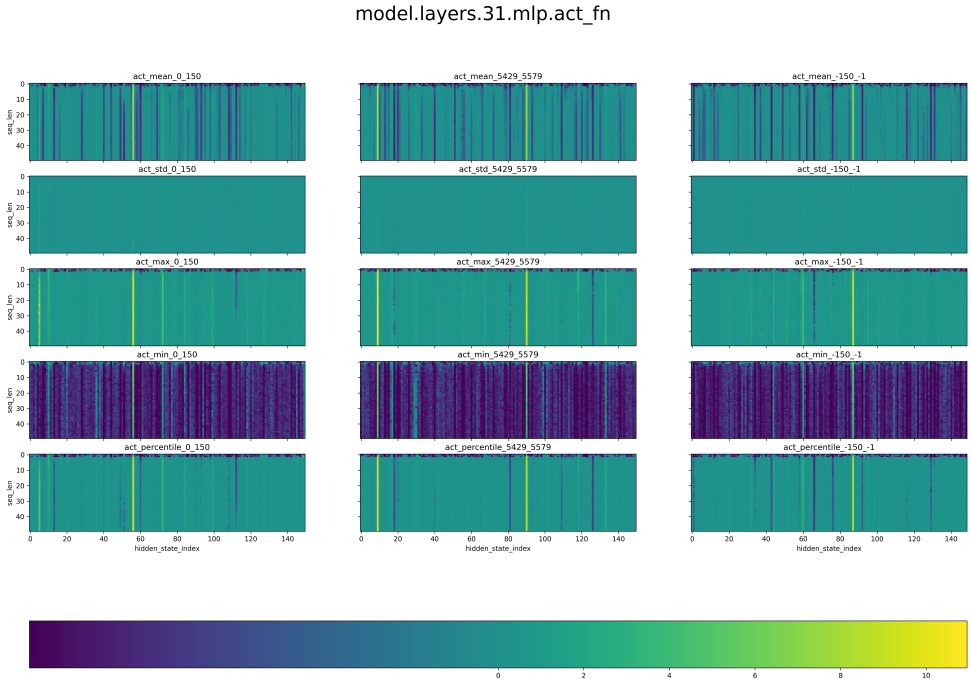

Figure 5: Profile of the feature map in the 32-nd layer of LLaMA2-7B.