Why Lift so Heavy?
Slimming Large Language Models by Cutting Off the Layers

Shuzhou Yuan[*1], **Ercong Nie**[*2,3], **Bolei Ma**[2,3], **Michael Färber**[1]
[1]Karlsruhe Institute of Technology, Germany
[2]LMU Munich, Germany
[3]Munich Center for Machine Learning (MCML), Germany
shuzhou.yuan@kit.edu, nie@cis.lmu.de, bolei.ma@lmu.de

Abstract

Large Language Models (LLMs) possess outstanding capabilities in addressing various natural language processing (NLP) tasks. However, the sheer size of these models poses challenges in terms of storage, training and inference due to the inclusion of billions of parameters through layer stacking. While traditional approaches such as model pruning or distillation offer ways for reducing model size, they often come at the expense of performance retention. In our investigation, we systematically explore the approach of reducing the number of layers in LLMs. Surprisingly, we observe that even with fewer layers, LLMs maintain similar or better performance levels, particularly in prompt-based fine-tuning for text classification tasks. Remarkably, in certain cases, models with a single layer outperform their fully layered counterparts. These findings offer valuable insights for future work aimed at mitigating the size constraints of LLMs while preserving their performance, thereby opening avenues for significantly more efficient use of LLMs.

1 Introduction

Stacking decoder layers enables Large Language Models (LLMs) to generate deep representations of input text, but also enlarges the model size of LLMs. Beginning with the vanilla Transformers using 6 decoder layers (Vaswani et al., 2017), language models have been rapidly growing to include more layers. Examples are Llama2 7B with 32 layers (Touvron et al., 2023), OPT with 24 layers (Zhang et al., 2022), and GPT2-XL with even 48 decoder layers (Radford et al., 2019). While LLMs with multiple layers have achieved state-of-the-art performance in Natural Language Processing benchmarks (Wang et al., 2018), adding just one layer could introduce millions of additional parameters to the LLM.

[*] Equal contribution.

Figure 1: Our research demonstrates that retaining only a single layer in LLMs, by simply cutting off additional layers, sustains equivalent performance compared to utilizing the full complement of layers.

Researchers aiming to reduce the training cost of LLMs have explored various strategies, including freezing the parameters of LLMs and incorporating additional trainable modules. These approaches encompass techniques such as Adapter tuning (Houlsby et al., 2019), Prompt tuning (Lester et al., 2021), and Prefix tuning (Li and Liang, 2021). Additionally, several studies have investigated the disparities in representations across different layers, highlighting the significance of updating the last few layers for language models (Kovaleva et al., 2019; Liu et al., 2019; Lee et al., 2019). While these methods demonstrate promising performance improvements and offer substantial savings in training costs, they do not address the issue of the cumbersome size of LLMs. Consequently, significant resources are still required for the storage and inference of these models.

To address the challenges associated with the large storage demand of LLMs and to explore the impact of the number of layers in LLMs, we propose a novel approach of reducing the number of decoder layers in LLMs by removing specific layers directly. In contrast to previous methods that involve freezing the parameters of LLMs while retaining the complete architecture, our approach involves simply trimming down the layers and re-

taining only the remaining layers in the model. Through experimentation across various classification tasks employing different styles, we surprisingly observe that a model with just one layer maintains performance comparable to that of the full model. By pruning the layers of the model, we effectively remove 93% of the parameters in GPT2-XL (Radford et al., 2019) and 88% of the parameters in OPT (Zhang et al., 2022).

Demonstrating a significant yet efficient finding, our work contributes to the field of NLP in the following aspects:

i) We systematically investigate the influence of the number of layers in LLMs on their performance and discover that reducing the number of layers does not lead to a decrease in the model's performance for classification tasks.

ii) Our findings provide valuable insights for future research aimed at reducing the size of models, thereby offering a potential solution to the significant challenges of huge computational budgets in LLM research.

2 Related Work

Pruning, a prevalent technique for model compression method, seeks to discard non-essential parameters within language models. This approach ranges from removing individual weights (unstructured pruning) to excising higher-granularity structures (structured pruning), aiming to enhance efficiency without compromising performance (LeCun et al., 1989; Han et al., 2016; Sanh et al., 2020; Xu et al., 2021; Xia et al., 2022; Wang et al., 2023). Fan et al. (2019) pioneered the concept of layer-wise pruning, a method that involves randomly dropping out some layers of the model in the training to enhance the model's robustness towards layer reduction at inference time. Building on this foundation, Peer et al. (2022) and Sajjad et al. (2023) aimed to identify and remove an optimal subset of layers directly from the pretrained models for use in downstream tasks. Recent studies have explored the application of pruning techniques to LLMs (Xia et al., 2023; Jha et al., 2023; Frantar and Alistarh, 2023; Chen et al., 2023; Ashkboos et al., 2024). Unlike previous work, we investigate the effect of layer-wise pruning on LLMs with prompt-based fine-tuning quantitatively.

3 Experiment

3.1 Layer Pruning

We systematically adopt a top-layer removing strategy by only remaining the first n layers and drop all other layers. Specifically, for GPT2-XL, we keep the models with 1, 2, 12, and 24 layers, while for OPT, we consider models with 1, 2, 6, and 12 layers.[1]

3.2 Problem Statement

In this work, we formulate text classification task as prompt-based language modeling task. With only a few examples provided as input, our objective is to determine the class of a given text. Let M denote a language model, and let $\mathcal{L}$ represent a set of label words. For instance, in a binary classification task, we define a prompt as follows:

$$X = t_1, l_1, t_2, l_2, t_i \quad (1)$$

Here, t_1, l_1 and t_2, l_2 denote the examples and their label words, t_i represents the text to be classified. Utilizing the prompt X, the language model M aims to predict the next token l_i, which is then assigned as the label word for t_i. M is initialized with pretrained parameters ϕ, and updated by minimizing the cross-entropy loss.

3.3 Dataset

We conduct text classification tasks using four widely utilized datasets spanning various domains: **AGNews:** AG's news topic classification dataset, utilized for topic classification with 4 labels (Zhang et al., 2015). **EmoC:** EmoContext, facilitating 4-label emotion classification (Chatterjee et al., 2019). **SST-2:** Stanford Sentiment Treebank Binary, employed for sentiment analysis (Socher et al., 2013). **TREC:** Text REtrieval Conference Question Classification (TREC), designed for question type classification, encompassing 6 types (Li and Roth, 2002; Hovy et al., 2001).

3.4 Setting

To create prompts, we select one demonstration per class and append the sample to be predicted at the end of the prompt. These demonstrations are drawn from the original training set of the datasets and are excluded from subsequent training. For training, we randomly select 200 training samples

[1]The complete GPT-2 XL model consists of 48 layers, and the OPT-1.3b model comprises 24 layers.

n	Param	AGNews	EmoC	SST-2	TREC	Average	n	Param	AGNews	EmoC	SST-2	TREC	Average
			GPT2-XL							OPT			
48	1.6B	85.34	73.70	68.97	80.16	77.04	24	1.3B	76.10	74.80	64.31	76.80	73.00
24	819M	**86.66**	75.44	72.41	72.41	76.73	12	711M	77.52	**78.46**	69.13	80.44	76.39
12	450M	86.42	75.34	73.76	84.72	80.06	6	409M	77.48	77.22	**69.34**	81.72	76.44
2	143M	85.80	75.78	**74.22**	**85.12**	**80.23**	2	297M	80.60	74.66	70.92	81.88	77.01
1	112M	85.30	**76.76**	72.89	83.16	79.53	1	157M	**81.32**	77.42	69.17	**82.12**	**77.51**

Table 1: Prompt-based fine-tuning with different numbers of layers using language modeling head. n denotes the number of layers.

n	Param	AGNews	EmoC	SST-2	TREC	Average	n	Param	AGNews	EmoC	SST-2	TREC	Average
			GPT2-XL							OPT			
48	1.6B	85.48	73.34	67.13	81.04	76.75	24	1.3B	73.96	76.72	64.08	75.92	72.67
24	819M	85.94	71.94	71.40	82.68	77.99	12	711M	76.02	75.86	68.60	80.72	75.30
12	450M	**86.06**	73.82	72.18	83.28	78.83	6	409M	78.56	**77.56**	**71.15**	80.52	**76.95**
2	143M	85.28	**75.52**	**73.19**	**84.12**	**79.53**	2	297M	77.92	72.76	69.38	80.80	75.22
1	112M	85.50	75.28	73.12	83.40	79.33	1	157M	**79.88**	72.38	71.10	**82.56**	76.48

Table 2: Prompt-based fine-tuning with different numbers of layers using classification head. n denotes the number of layers.

from the original training set to ensure adequate model training. Additionally, we allocate 1000 samples from the original training set for validation purposes, while 1000 samples from the original test set are reserved for evaluation. Model selection is based on the accuracy achieved on the validation set, with the best-performing model then evaluated on the test set.

We maintain consistent hyperparameters across models, regardless of the number of layers. Specifically, we employ AdamW (Loshchilov and Hutter, 2019) as the optimizer with a learning rate of 5e-5. All models are trained for 50 epochs, with early stopping implemented after 15 epochs. Due to resource constraints, we set the batch size to 1. Five different random seeds [2] are utilized for the experiments, and the average accuracy across these seeds is reported as the final results.

3.5 Main Results

The results are presented in Table 1. For the GPT2-XL model, intriguingly, the model's performance demonstrates minimal variance across varying layer configurations. When fine-tuned with the complete set of layers, the average accuracy reaches 77.04%. Even with a reduction to 24 layers, the accuracy only experiences a slight decrease to 76.73%. Notably, a reduction to 2 layers yields the highest accuracy at 80.23%, surpassing the full-layer model. Remarkably, a one-layer model outperforms its full-layer counterpart, indicating the potential for significant parameter reduction from 48 to 1, saving nearly 1.5 billion parameters.

Similarly, OPT exhibits a consistent trend of performance improvement with decreasing layer count. While the 24-layer configuration achieves an average accuracy of 73%, performance consistently improves as layers are reduced. OPT with 12 layers achieves an accuracy of 76.39%, with 6 layers, 76.44%, with 2 layers, 77.01%, and with a single layer, the highest accuracy of 77.51% is attained. The reduction of model layers from 24 to 1 results in a saving of 1.14 billion parameters.

3.6 Replacement of Classification Head

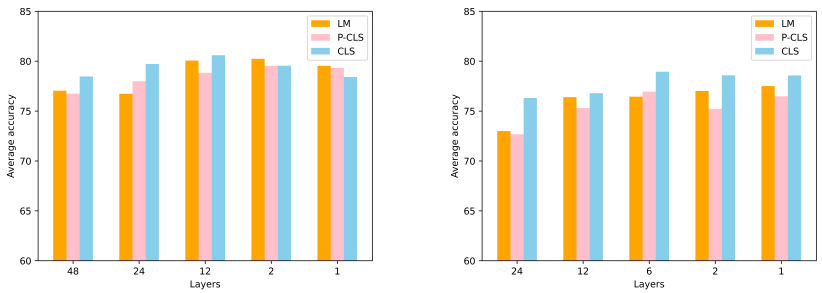
Given that the classification tasks are framed as language modeling endeavors, an inquiry arises regarding whether the consistent performance across varied layer configurations stems from the semantic understanding facilitated by the language modeling head. To explore this hypothesis, we substitute the language modeling head with a classification head, maintaining the experimental setting outlined in §3.4.

The outcomes of employing a classification head for prompt-based fine-tuning are delineated in Table 2. In comparison to the language modeling head, the performance of the classification head exhibits a slight decrease under identical layer counts. Nonetheless, the average accuracy of the classification head maintains consistent performance with marginal fluctuations. For GPT2-XL, employing the full complement of layers yields an average accuracy of 76.75%. Remarkably, the model achieves

[2]Random seeds: [0, 42, 421, 520, 1218]

n	Param	AGNews	EmoC	SST-2	TREC	Average	n	Param	AGNews	EmoC	SST-2	TREC	Average
			GPT2-XL							OPT			
48	1.6B	86.62	72.60	72.96	81.64	78.46	24	1.3B	78.94	76.02	68.35	81.92	76.31
24	819M	86.02	75.42	73.05	84.40	79.72	12	711M	83.30	75.12	67.02	81.72	76.79
12	450M	**86.96**	**76.58**	**73.76**	**85.04**	**80.59**	6	409M	83.82	**78.22**	70.85	82.88	**78.94**
2	143M	86.64	74.32	72.94	84.28	79.54	2	297M	**84.38**	76.54	**70.94**	82.44	78.58
1	112M	85.56	73.66	73.30	81.12	78.41	1	157M	81.86	77.80	70.87	**83.76**	78.57

Table 3: Fine-tuning with different numbers of layers using classification head. n denotes the number of layers.



(a) GPT2-XL (b) OPT

Figure 2: The average accuracy of different layers for three classification methods. LM denotes language modeling head, P-CLS denotes prompt-based fine-tuning with classification head, CLS denotes classification head without prompt.

its highest accuracy with only 2 layers, with even a single-layer configuration outperforming the full-layer model. A similar trend is observed for the OPT model, wherein a reduction in layers correlates with performance enhancement. While the full-layer configuration achieves an average accuracy of 72.67%, the single-layer model surpasses it with an accuracy of 76.48%.

3.7 Further Discussion: Classification

We delve deeper into the classification task employing the conventional classification paradigm, wherein text categorization relies solely on the input text without demonstrations. By removing demonstrations and providing the model with input examples, we apply a classification head to categorize the text, adhering to the experimental settings articulated in §3.4.

As depicted in Table 3, the average accuracy of the classification setting conforms to the trend observed in prompt-based fine-tuning. The reduction in layers does not exert a discernible impact on the model's performance. For GPT2-XL, the model consistently maintains an average accuracy of approximately 80%, irrespective of the number of layers. Optimal accuracy is attained by the model with 12 layers. Analogously, OPT exhibits similar performance regardless of layer count, with the single-layer configuration outperforming the full-layer model, both for GPT2-XL and OPT.

Furthermore, we visually represent the results of the three classification methods we have implemented in Figure 2. Notably, these methods yield similar outcomes, with layer count exerting negligible influence on model performance. Intriguingly, models with fewer layers often outperform those with full layers. These findings suggest that neither the language modeling head nor the classification head singularly dictates the consistent performance observed with decreasing layer count. Consequently, this discovery holds promise for optimizing resources in classification tasks by economizing training and storage resources.

4 Conclusion

This study delves into the impact of layer count on the performance of LLMs. Our investigation unveils that variations in the number of layers do not result in performance degradation for classification tasks employing LLMs. Remarkably, even with a reduction in the number of LLM layers from 48 to 1, both GPT2-XL and OPT consistently maintain or even enhance performance. This discovery offers valuable insights into reducing the training and storage costs associated with LLMs, presenting promising avenues for optimizing resource allocation in such models.

Limitations

This study exclusively focuses on examining the impact of the number of layers in LLMs on classification tasks, without extending the investigation to other NLP tasks. Furthermore, we compare the performance of various classification schemes but refrain from delving deeper into elucidating the underlying reasons behind the observed phenomenon where the models exhibit consistent performance regardless of the number of layers.

References

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Tianyi Chen, Tianyu Ding, Badal Yadav, Ilya Zharkov, and Luming Liang. 2023. Lorashear: Efficient large language model structured pruning and knowledge recovery. *arXiv preprint arXiv:2310.18356*.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*.

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.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *4th International Conference on Learning Representations, ICLR*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Ananya Harsh Jha, Tom Sherborne, Evan Pete Walsh, Dirk Groeneveld, Emma Strubell, and Iz Beltagy. 2023. How to train your (compressed) large language model.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. *Advances in neural information processing systems*, 2.

Jaejun Lee, Raphael Tang, and Jimmy Lin. 2019. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

David Peer, Sebastian Stabinger, Stefan Engl, and Antonio Rodríguez-Sánchez. 2022. Greedy-layer pruning: Speeding up transformer models for natural language processing. *Pattern Recognition Letters*, 157:76–82.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. On the effect of dropping layers of pre-trained transformer models. *Computer Speech and Language*, 77:101429.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Xinpeng Wang, Leonie Weissweiler, Hinrich Schütze, and Barbara Plank. 2023. How to distill your BERT: An empirical study on the impact of weight initialisation and distillation objectives. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1843–1852, Toronto, Canada. Association for Computational Linguistics.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ NeurIPS 2023)*.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1528, Dublin, Ireland. Association for Computational Linguistics.

Dongkuan Xu, Ian En-Hsu Yen, Jinxi Zhao, and Zhibin Xiao. 2021. Rethinking network pruning – under the pre-train and fine-tune paradigm. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2376–2382, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.