Towards Low-Rank Sparsity for Language Model Compression

Anonymous ACL submission

Abstract

In this paper, we first investigate two lines of methods for compressing large pre-trained language models (PLMs): weights pruning and low-rank factorization. We discover an exclusive low-rank sparsity pattern in models produced by a family of first-order weights pruning algorithms, which motivates us to unite the two approaches and achieve more effective model compression. We further propose two techniques: sparsity-aware SVD and mixed-rank fine-tuning, that improves the initialization and training of the compression procedure respectively. Experiments on natural language understanding and question answering tasks show that the proposed method achieves superior compression-performance trade-off compared to existing approaches.

1 Introduction

Large-scale Transformer-based (Vaswani et al., 2017) pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019) have demonstrated state-of-the-art performance across a variety of natural language processing tasks, including natural language understanding and natural language generation. These models are largely over-parametrized (Nakkiran et al., 2019) in that they usually contain hundreds of millions of parameters, rendering them computationally intensive and inefficient in terms of both memory and inference latency. Due to this disadvantage, the application of PLMs in low-resource scenarios are limited.

To alleviate this problem, model compression (Louizos et al., 2018; Ben Noach and Goldberg, 2020) has emerged as a timely research field. Among all compression techniques, *weights pruning* and *low-rank factorization* have received much attention because of their interesting properties. In weights pruning, model weights are systematically zeroed out by certain pruning criteria, e.g., learnable importance score. Though effective, the resultant unstructured sparse matrices require special linear algebra implementation (Yao et al., 2018) or hardware (Cao et al., 2019) to obtain a perceivable reduction in memory and computation, which is unfriendly to most practitioners. In contrast, low-rank factorization decomposes the original weight matrix into smaller sub-matrices, providing direct improvement in efficiency. However, it tends to underperform weights pruning given the same parameter budget. A natural question is: *how can we make the best use of these two methods to achieve both high accuracy and efficiency?* In an attempt to answer this question, we first conduct a preliminary study on common weights pruning and low-rank factorization methods for better understanding of their mechanisms. For weights pruning, we select algorithms with and without using gradient-level information when computing the importance score, denoted as *first-order* and *zero-order* weights pruning. For low-rank factorization, we select singular value decomposition (SVD) due to its widespread use. From our experiments, we make the following observations: (1) under a high compression ratio, low-rank factorization fails to retain satisfactory performance because the weight matrices of densely fine-tuned models are nearly *full-rank* (767 on average for BERT-base) and much task-specific information is lost after low-rank approximation;

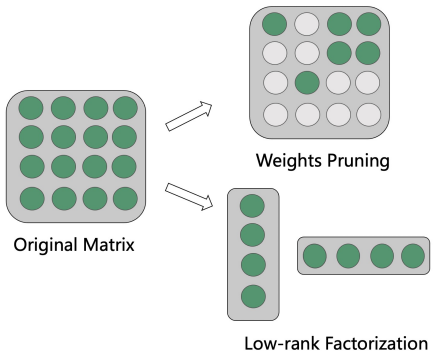

Figure 1: Illustration of weights pruning and low-rank factorization applied on a single weight matrix.

(2) only first-order weights pruning produces sparse weight matrices that are *low-rank*, showing that gradient information is essential for an accurate indicator of weights importance so as to discover the intrinsic low-rank structure.

The above two findings motivate us to explore the possibility of performing low-rank factorization on low-rank sparse models. Concretely, for a given PLM and training data, we first apply first-order weights pruning (Sanh et al., 2020) on it to obtain its low-rank sparse counterpart. Then, each sparse weight matrix is factorized into two smaller sub-matrices via SVD, which are further fine-tuned to recover full performance. The procedure is model architecture-agnostic and hence can be potentially applied to a broad set of existing PLMs.

Moreover, we noticed that the vanilla SVD is not designed for sparse matrices because it penalizes the approximation error of each parameter equally (Chen et al., 2018). Also, due to the reduced capacity, the joint fine-tuning of low-rank sub-matrices may converge to solutions with lower generalization performance. To address the first problem, we propose sparsity-aware SVD, a weighted variant of SVD that better reconstructs unpruned (hence more important) parameters. To address the second problem, we introduce mixed-rank fine-tuning, a regularized training scheme where the low-rank sub-matrices are randomly replaced with sparse matrix from which they are factorized.

We test on natural language understanding and question answering tasks. Experimental results show that the proposed **L**ow-rank **P**rune-**A**nd-**F**actorize (LPAF) approach outperforms existing approaches in terms of compression-performance trade-off. When augmented with sparsity-aware SVD and mixed-rank fine-tuning, the performance is further improved. We also conduct in-depth analysis to provide guidance regarding different resource budgets.

Our contributions are summarized as follows:

- Through a comprehensive preliminary study, we discover a high-rank phenomenon in models produced by fine-tuning/zero-order pruning and a low-rank phenomenon by first-order pruning, which highlights the possibility of a more efficient parametrization of low-rank sparse matrices using low-rank factorization.

- Based on our findings, we propose a novel model compression framework which combines *first-order pruning* and *low-rank factorization*. As further optimizations, we propose *sparsity-aware SVD* which prioritizes reconstruction of unpruned weights at initialization, and *mixed-rank fine-tuning* to compensate for the reduced capacity during training.

- Experiments on natural language understanding and question answering tasks show that our approach can achieve a 5.4x-12.3x reduction in model size and 4.6x-10.6x reduction in computation cost while retaining 98%-93% of the performance of the original BERT.

References

Matan Ben Noach and Yoav Goldberg. 2020. Compressing pre-trained language models by matrix decomposition. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 884–889, Suzhou, China. Association for Computational Linguistics.

Shijie Cao, Chen Zhang, Zhuliang Yao, Wencong Xiao, Lanshun Nie, Dechen Zhan, Yunxin Liu, Ming Wu, and Lintao Zhang. 2019. Efficient and effective sparse lstm on fpga with bank-balanced sparsity. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '19, page 63–72, New York, NY, USA. Association for Computing Machinery.

Patrick H. Chen, Si Si, Yang Li, Ciprian Chelba, and Cho-Jui Hsieh. 2018. Groupreduce: Block-wise low-rank approximation for neural language model shrinking. *CoRR*, abs/1806.06950.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Christos Louizos, Max Welling, and Diederik P Kingma. 2018. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2019. Deep double descent: Where bigger models and more data hurt. *CoRR*, abs/1912.02292.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *CoRR*, abs/2005.07683.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Zhuliang Yao, Shijie Cao, Wencong Xiao, Chen Zhang, and Lanshun Nie. 2018. Balanced sparsity for efficient DNN inference on GPU. *CoRR*, abs/1811.00206.