Differentiable DARE-TIES for NeurIPS 2024 LLM Merging Competition

Toshiyuki Nishimoto Yoichi Hirose Yuya Kudo Nozomu Yoshinari
Rio Akizuki Kento Uchida Shinichi Shirakawa
Yokohama National University
nishimoto-toshiyuki-gf@ynu.jp hirose-youichi-kc@ynu.jp
kudo-yuya-dr@ynu.jp yoshinari-nozomu-ry@ynu.jp
akizuki-rio-pd@ynu.jp uchida-kento-fz@ynu.ac.jp
shirakawa-shinichi-bg@ynu.ac.jp

Abstract

With the increasing training cost of large language models, model merging is attracting attention. This report describes our effort in the area during the NeurIPS 2024 LLM Merging Competition. We developed *differentiable DARE-TIES*, which optimizes the merging parameters in a differentiable manner. Whereas existing methods rely on black-box optimization algorithms, our method utilizes gradient descent and is expected to optimize high-dimensional merging parameters more efficiently. We conducted experiments to examine the potential of our approach.

1 Introduction

Large language models (LLMs) have significantly advanced the field of natural language processing, enabling breakthroughs in tasks ranging from machine translation to conversational agents. As these models have been growing in scale and complexity, techniques for efficiently merging and optimizing them have become crucial. One such technique gaining attention is evolutionary model merge [1], which leverages evolutionary algorithms to optimize merging hyperparameters for combining multiple LLMs into a more powerful model.

The covariance matrix adaptation evolution strategy (CMA-ES) [2] is a popular evolutionary algorithm used in this context, which is also known as a well-performed continuous black-box optimization method. By conducting extensive evaluations—often numbering in the thousands—CMA-ES explores the parameter space to find well-optimized configurations for model merging. For instance, 1,000 evaluations with CMA-ES yielded the high-quality merged model in [1]. However, this approach has a drawback: the computational time required for convergence can be high due to the computational demands of evolutionary processes.

To address this challenge, we propose a novel method that transforms the design parameters for optimization into differentiable forms, enabling the use of gradient-based optimization techniques. By making the parameters differentiable, we can apply efficient gradient descent methods to optimize the merging process directly. This approach significantly reduces the convergence time compared to conventional evolutionary algorithms.

In our experiments, we observed that the proposed method could reduce the computational time by about ten times compared to the evolutionary model merge method with a typical setting.[1] Despite the efficiency of our approach, the performance of the merged models was promising as

[1] While our proposed method can reduce the computational time for optimization compared to evolutionary algorithm-based merging parameter optimization, the GPU memory usage of our method is large because all the tensors of the base and source models are loaded.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

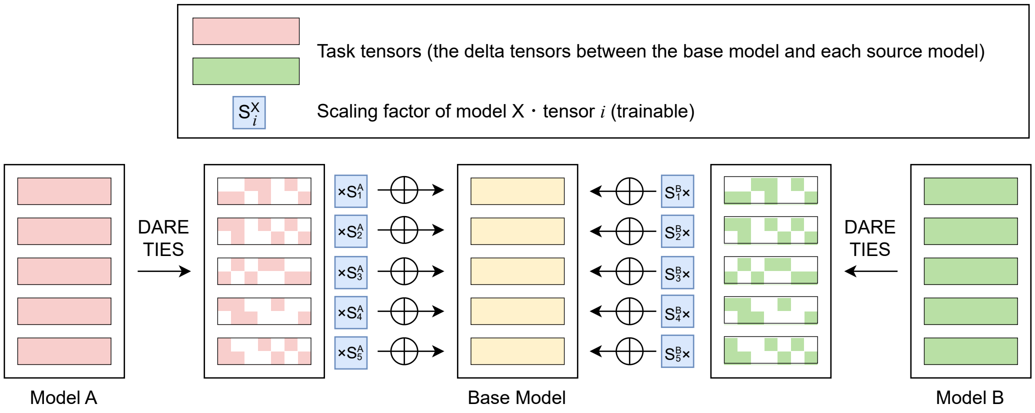


Figure 1: Overview of our Differentiable DARE-TIES method.

exemplified by our method achieving a fourth-place in the LLM-Merging competition. We submitted the model merging code for the competition using the merging configuration parameters obtained by the proposed method.

2 Proposed Method

We introduce a differentiable model merging method based on DARE-TIES, as illustrated in Figure 1. In settings with a base model and multiple fine-tuned versions of it, which we refer to as source models, we first extract task tensors [3], representing the delta tensors between the base model and each source model. These tensors are then sparsified, following a process similar to DARE-TIES: a portion of each task tensor is randomly dropped with a ratio of p, and tensor values with the minority signs within each tensor are also removed. Next, each task tensor is scaled by learnable scaling factors and added to the corresponding tensor in the base model. While DARE-TIES typically applies scaling at the model or layer level [1], our method applies it at the tensor level for more granular model merging. We treat the tensors obtained by `named_parameters()` in PyTorch as the set of tensors. After merging, we optimize the cross entropy loss with respect to the learnable scaling factors using gradient-based methods, while keeping all other model parameters fixed.

2.1 Fine-Tuning Dataset Preparation

After optimizing the scaling factors, we fine-tuned the merged model on a small custom dataset containing 921 samples to align its output format. We created this dataset by randomly selecting three samples for each task in BIG-Bench [4] and FLAN Collection T0 sub-mixture [5]. This process was intended to reduce instances of generated text that did not match the competition's output format. In our experiments, fine-tuning the merged models required only a few minutes.

3 Experiment

We compared our merging method, which utilizes optimized scaling factors, against a version that employs randomly sampled scaling factors from a uniform distribution between 0 and 1.

3.1 Experimental Setup

Dataset We used tinyBenchmarks [6] to optimize merging parameters, which is a small dataset of 600 questions and answers. For the multiple-choice task, we used the correct answer statement as the answer.

Models We used Llama 3 8B [7] as the base model, and also used suzume-llama-3-8B-multilingual-orpo-borda-top75 [8], MAmmoTH2-8B-Plus [9] and Llama-3-Refueled [10] as source models. These

Table 1: The final leaderboard score of our methods.

Method	Private final leaderboard score
Ours	0.41
Ours w/o Fine-Tuning	0.40

models show high performance on the Open LLM Leaderboard 2 Average, Math Hard, and Big Bench Hard, respectively. Llama 3 8B has 291 tensors, so the total number of scaling factors is 873.

Evaluation We evaluated the performance of the merged models on the private final leaderboard of the LLM Merging Competition on Kaggle. The chat template for llama 3 was applied to the original questions in the test dataset during the evaluation process. For the multiple choice questions, we gave the model choices as a system prompt, and then selected the choice that maximized the rougeLsum f-measure with the model output as the answer.

Optimization We used Adam to optimize the scaling factors. In Adam, we set the learning rate to 0.1, β_1 to 0.9 and β_2 to 0.999, the batch size to 50, the number of epochs to 50. We set the initial scaling factors to 1. We cropped the scaling factors between 0 and 1 while optimizing. We set all density parameters for DARE to 0.5. The chat template is used in the optimization process.

Fine-Tuning We fine-tuned the optimized merged model using low-rank adaptation [11] with a linearly decaying learning rate from 10^{-5} to 0, a rank r of 8, a scaling factor α of 16, a dropout rate of 0.1, a batch size of 4, and AdamW as the optimizer, over one epoch.

3.2 Results

Table 1 shows final leaderboard score of LLM Merging Competition on Kaggle. Our method achieved a score of 0.41, placing fourth in the competition, where scores ranged from 0.17 to 0.46 in the competition. The result suggests that the differentiable approach to model merging is a promising strategy for achieving greater efficiency. Additionally, Figure 2 illustrates the distribution of optimized scaling factors for three source models. The scaling factors mostly gather around the extremes (zero and one) instead of middle values. This pattern suggests that the merging mainly depends on a few important tensors from each source model rather than combining all of them equally. This selective focus might mean that certain models provide unique and highly useful tensors, which are crucial for the final performance, while others are essentially filtered out by scaling factors near zero.

4 Conclusion

In this report, we proposed *differentiable DARE-TIES* that optimizes the scaling factors for model merging efficiently. Our method optimized the scaling factors by the gradient method. We found experimentally that *differentiable DARE-TIES* shows competitive results at very low computational cost. Although we only optimized the scaling factors, simultaneously optimizing other merging parameters, including non-differentiable ones, is an interesting direction.

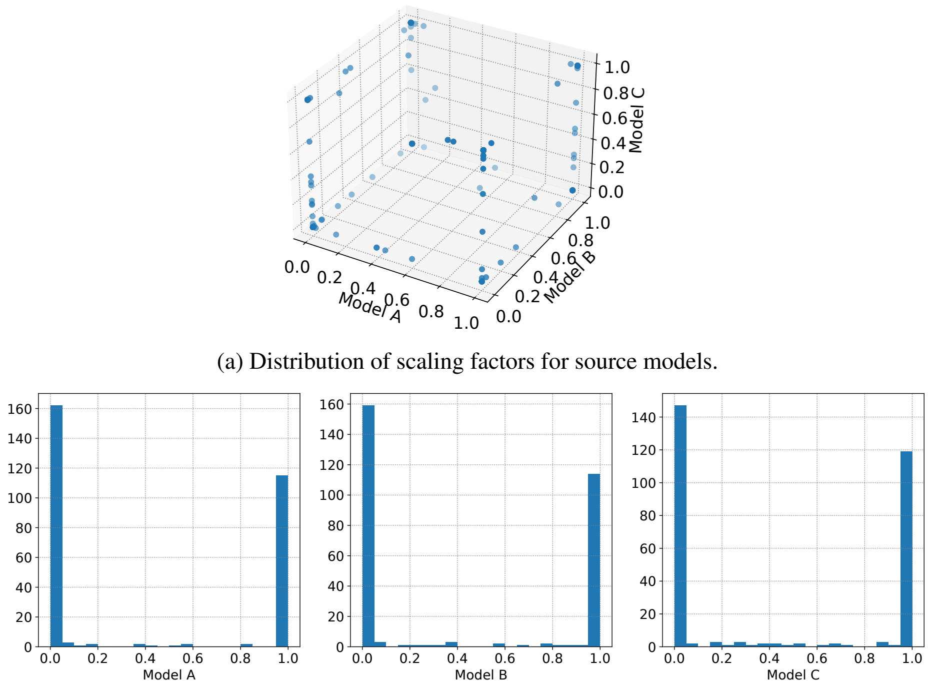
References

[1] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *arXiv preprint arXiv:2403.13187*, 2024.

[2] Nikolaus Hansen. *The CMA Evolution Strategy: A Comparing Review*, pages 75–102. Springer Berlin Heidelberg, 2006.

[3] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.

[4] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game:



(a) Distribution of scaling factors for source models.

(b) Histograms of the scaling factors separately for each source model.

Figure 2: Distribution of optimized scaling factors across source models. Model A indicates suzume-llama-3-8B-multilingual-orpo-borda-top75, model B is MAmmoTH2-8B-Plus, and model C is Llama-3-Refueled.

Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

[5] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning (ICML)*, 2023.

[6] Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating LLMs with fewer examples. In *International Conference on Machine Learning (ICML)*, 2024.

[7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and et al. Ahmad Al-Dahle. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[8] Peter Devine. Are you sure? Rank them again: Repeated ranking for better preference datasets. *arXiv preprint arXiv:2405.18952*, 2024.

[9] Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. MAmmoTH2: Scaling instructions from the web. *arXiv preprint arXiv:2405.03548*, 2024.

[10] Refuel Team. Announcing Refuel LLM-2, 2024. `https://www.refuel.ai/blog-posts/announcing-refuel-llm-2` (Accessed on 11 1, 2024).

[11] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.