UDC 004.852

SimplexLoRA: Expand Important Adapters

G. R. Davydenko[1], I. V. Shalygin[1], and A. S. Veprikov[1]

[1]Moscow Institute of Physics and Technology

May 12, 2025

Language models have become central to many AI applications. Effective fine-tuning is essential to adapt these models to specific tasks. Traditional methods like Low-Rank Adaptation (LoRA) [1] add fixed-rank adapters to all layers, often resulting in memory inefficiency due to non-optimal layer selection. We propose **SimplexLoRA**, a novel fine-tuning framework that adaptively scales adapter ranks using simplex-constrained weighting, optimizing both memory usage and performance.

Fine-tuning large language models with LoRA involves optimizing the following objective:

$$\min_{A_i, B_i} \mathcal{L}(W_1 + A_1 B_1, \dots, W_n + A_n B_n),$$

where the adapter matrices $A_i \in \mathbb{R}^{n \times r}$ and $B_i \in \mathbb{R}^{r \times m}$ (with $r \ll n, m$), pretrained weights $W_i \in \mathbb{R}^{n \times m}$, and loss function $\mathcal{L}$ are given.

However, applying equal-rank adapters uniformly across all layers is inefficient and lacks flexibility. Our approach introduces a more efficient method for determining adapter ranks, allowing the model to learn them dynamically.

SimplexLoRA introduces a new trainable parameter — a weight vector $\omega = \{\omega_i\}_{i=1}^n$ — into the optimization problem, which becomes:

$$\min_{A_i, B_i} \mathcal{L}(W_1 + \omega_1 A_1 B_1, \dots, W_n + \omega_n A_n B_n).$$

We further assume that ω lies on the simplex, resulting in the final formulation:

$$\min_{\omega \in \Delta^{n-1},\ A_i, B_i} \mathcal{L}(W_1 + \omega_1 A_1 B_1, \dots, W_n + \omega_n A_n B_n).$$

This reformulation introduces a trainable weighting mechanism while maintaining the core LoRA structure.

We design the rank selection algorithm as follows: during initial fine-tuning steps, we apply gradient descent with projection onto the simplex for the weight vector. After a few iterations, we fix the ranks based on the learned weights and continue training in the standard LoRA manner with fixed weights and ranks.

The rank r_i at iteration k is computed as:

$$r_i^{(k)} = \left\lfloor \omega_i^{(k)} \cdot r^{(0)} \right\rfloor, \quad \text{where } r^{(0)} \text{ is the initial rank.}$$

Since changing a matrix's rank alters its structure, we use a combination of SVD and QR decompositions to preserve the product $A_i B_i$ when updating ranks.

We implemented SimplexLoRA within the PEFT library and developed a custom optimizer. We conducted extensive experiments, including evaluations on DeBERTa and LLaMA using the GLUE benchmark. The results demonstrate that SimplexLoRA achieves promising performance and improved efficiency.

References

[1] 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. *arXiv preprint arXiv:2106.09685*, 2021. `https://arxiv.org/abs/2106.09685`