Zero-Order Optimization for LLM Fine-Tuning

Grigoriy Evseev

consultant: Veprikov Andrey

expert: Aleksandr Beznosikov

MIPT

2025

Goal

1. Propose two novel zero-order optimization algorithms for fine-tuning Large Language Models.
2. Reduce memory consumption by eliminating the need for gradient backpropagation.
3. Demonstrate that the proposed methods outperform existing Zero-Order and First-Order baselines on the LLM fine-tuning task.

Zero-Order PEFT

Large Language Models require significant memory for training.
Backpropagation in first-order methods (SGD, Adam) causes high memory overhead.
Zeroth-Order methods only need forward passes, reducing memory usage.

Table: Memory consumption comparison for **full finetuning** evaluated on OPT-13B model with the MultiRC dataset.

Optimizer	Empirical Mem.
FO-SGD	97 GB
FO-Adam	239 GB
ZO-SGD	51 GB
ZO-Adam	151 GB

We propose a new Zero-Order optimization methods for LLM fine-tuning.
Compare them with existing methods like ZO-SGD and ZO-Adam.

Related Work

1. Revisiting Zeroth-Order Optimization for Memory-Efficient LLM Fine-Tuning (2024) *(arXiv)*
2. Fine-Tuning Language Models with Just Forward Passes (2024) *(arXiv)*
3. Simultaneous Computation and Memory Efficient Zeroth-Order Optimizer (2024) *(arXiv)*
4. Muon is Scalable for LLM Training (2025) *(arXiv)*
5. Old Optimizer, New Norm: An Anthology (2024) *(arXiv)*

Zero-order setup

Given a scalar-valued function $f(x)$ where $x \in \mathbb{R}^d$, the gradient estimation referred to as $\widehat{\nabla} f(x)$ is expressed using difference

$$\widehat{\nabla} f(x) = \frac{1}{q} \sum_{i=1}^{q} \left[\frac{f(x + \mu u_i) - f(x - \mu u_i)}{2\mu} u_i \right],$$

where u_i is a random normal vector, q is the number of function queries, and $\mu > 0$ is a smoothing parameter.

Zero-Order Sign Jaguar

Core idea: step 1

Generate $i \sim \mathcal{U}[1, n]$

$$
\begin{aligned}
z_+ &\leftarrow X^t \\
(z_+)_i &\leftarrow X^t_i + \tau \cdot \mathbf{1}^d \\
z_- &\leftarrow X^t \\
(z_-)_i &\leftarrow X^t_i - \tau \cdot \mathbf{1}^d
\end{aligned}
$$

Core idea: step 2

$$
\begin{aligned}
\widehat{\nabla} f^{t+1} &\leftarrow \widehat{\nabla} f^t \\
(\widehat{\nabla} f^{t+1})_i &\leftarrow \operatorname{sign}(f(z_+) - f(z_-)) \cdot \tau \cdot \mathbf{1}^d
\end{aligned}
$$

Zero-Order Muon

Core Idea

$$\mathbf{M}_t = \mu \mathbf{M}_{t-1} + \widehat{\nabla} \mathcal{L}_t(\mathbf{W}_{t-1})$$
$$\mathbf{O}_t = \text{Newton-Schulz}(\mathbf{M}_t)$$
$$\mathbf{W}_t = \mathbf{W}_{t-1} - \eta_t \mathbf{O}_t$$

Newton-Schulz update

$$\mathbf{X}_k = a\mathbf{X}_{k-1} + b(\mathbf{X}_{k-1}\mathbf{X}_{k-1}^{\mathrm{T}})\mathbf{X}_{k-1} + c(\mathbf{X}_{k-1}\mathbf{X}_{k-1}^{\mathrm{T}})^2\mathbf{X}_{k-1}$$

Usually, $a = 3.4445, b = 4.7750, c = 2.0315.$

Experiment setup

Main model: **OPT**

Task for LLM fine-tuning: **Binary Classification** (Stanford Sentiment Treebank v2, SST2)

Fine-tuning scheme: **Low-Rank Adaptation, LoRA** (Imposes low-rank weight perturbations)

Experiment results

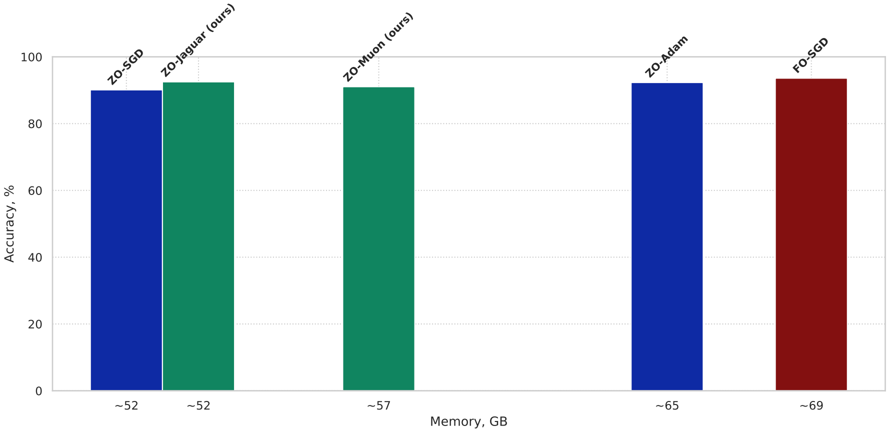

Figure: Results of OPT-13B on the tasks SST2 fine-tuned using ZO/FO optimizers in LoRA setting.

Conclusion

1. We proposed two novel zero-order optimizers for parameter-efficient LLM fine-tuning.
2. Our methods significantly reduce memory consumption by avoiding backpropagation.
3. Experiments on OPT-13B with LoRA demonstrate that both methods outperform existing zeroth- and first-order baselines in accuracy and efficiency.
4. Zero-order optimization shows strong potential as a practical tool for scalable and low-resource LLM fine-tuning.