GradPruner

Gradient-Guided Layer Pruning Enabling Efficient Fine-Tuning and Inference for LLMs

Anda Cheng

Ant Group

蚂蚁集团
ANT GROUP

Table of Contents

01 Introduction & Motivation

02 Methodology: GradPruner

03 Experimental Results

04 Conclusion & Future Work

蚂蚁集团
ANT GROUP

LLM Fine-tuning: Time-Consuming and Costly

High Performance, High Cost

Fine-tuning LLMs achieves great performance but is computationally expensive. For example, training a medical LLM can take up to 221 hours.

Memory vs. Inference

Methods like LoRA effectively reduce training memory footprint but do not improve inference efficiency, creating a gap between training and deployment.

Pruning Overhead

Structured pruning boosts inference speed but often requires additional training, knowledge distillation, or architecture search, adding significant overhead.

Key Challenges Addressed

Parameter Importance Measurement

Measuring parameter importance for specific downstream data and models without increasing memory or training time.

Structure Preservation

Maximizing parameter pruning while preserving the original model structure as much as possible.

Flexibility

Seamlessly supporting both full fine-tuning and LoRA fine-tuning workflows.

Our Solution: GradPruner

Core Idea: Utilize gradient information from the early stages of fine-tuning to guide layer pruning, enabling efficient fine-tuning and inference simultaneously.

01. Parameter Evaluation

Compute the Initial Gradient Information Accumulation Matrix (IGIA-Matrix) using a small number of LoRA fine-tuning steps to assess importance.

02. Layer Pruning

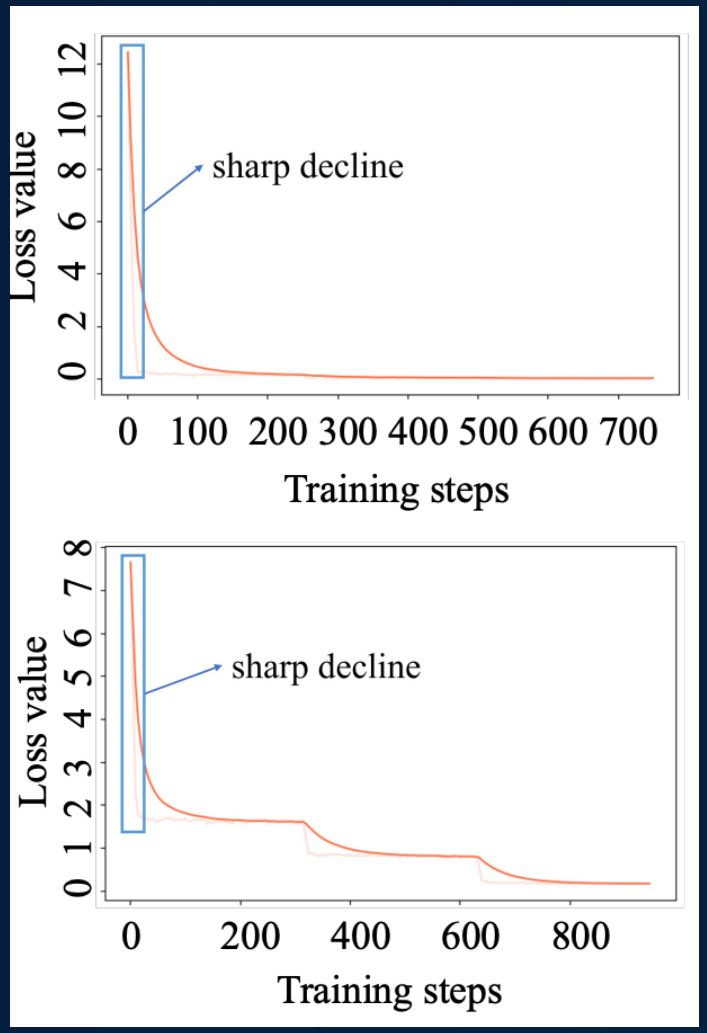
Based on the IGIA-Matrix, evaluate layer importance scores and selectively prune the less critical layers from the model.

03. Layer Merging

Merge the pruned layers with the remaining ones to further increase the overall pruning ratio while maintaining model accuracy.

蚂蚁集团
ANT GROUP

Motivation: Rapid Loss Drop in Early Fine-tuning



Key Observation

Loss function drops sharply within the initial 1% of training steps during fine-tuning.

Implication

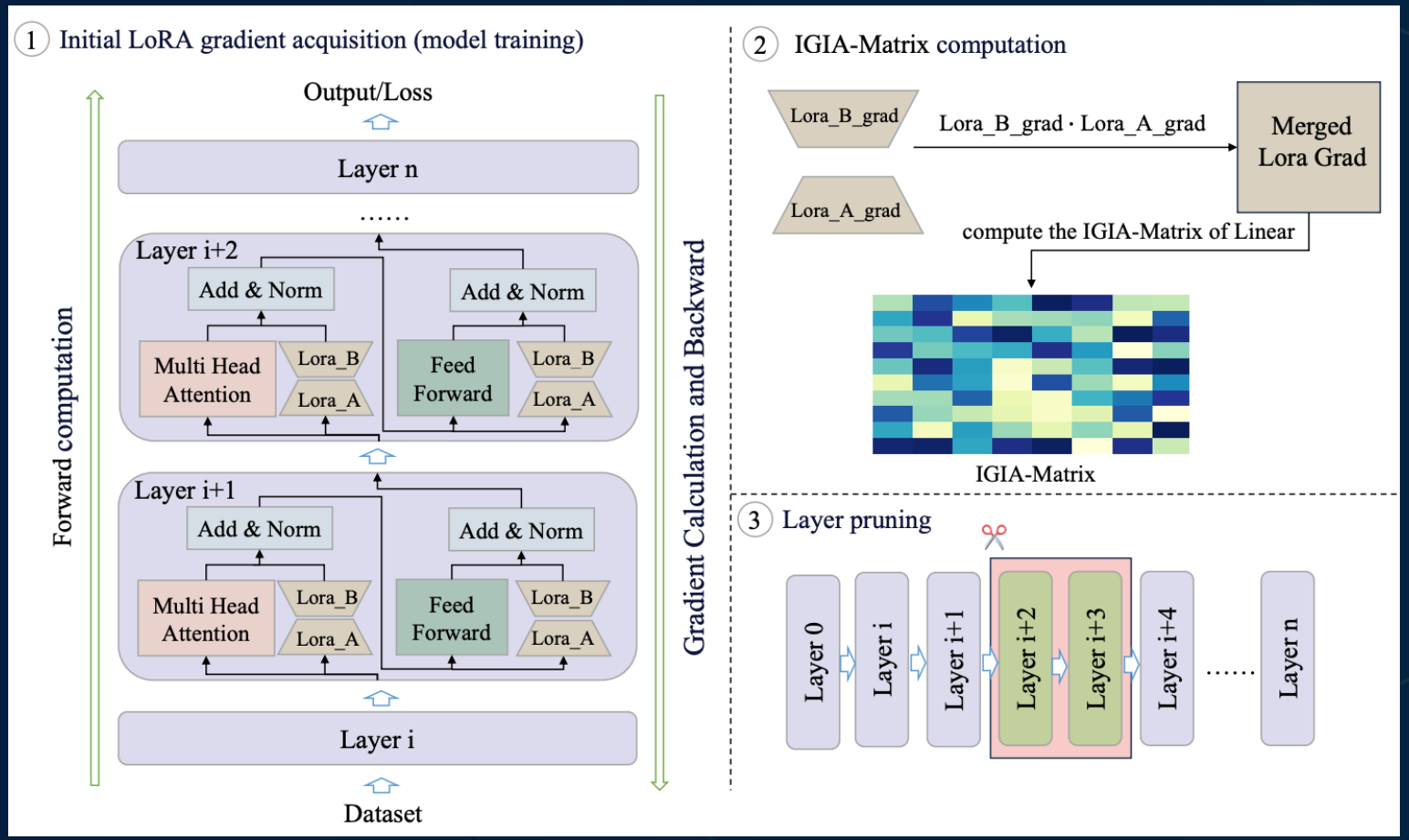
The model quickly learns key task knowledge, making parameter importance evident early on.

Conclusion

Gradient information from early steps can accurately identify important parameters.

蚂蚁集团
ANT GROUP

GradPruner: Parameter Importance & Layer Pruning



Step 1: Obtain Gradients

Perform a small number of LoRA fine-tuning steps to collect gradients of the LoRA weights.

Step 2: Compute IGIA-Matrix

Simulate original model gradients using LoRA gradients and compute the IGIA-Matrix to aggregate gradient information.

Step 3: Prune Layers

Sum IGIA values to get layer importance scores, then prune layers with the lowest scores.

蚂蚁集团 ANT GROUP

GradPruner : Core Formulas for IGIA-Matrix

Simulate Gradient for W

$$\nabla_W L(x,y)_i \overset{\text{sim}}{=} \nabla_{W_B} L(x,y)_i \cdot \nabla_{W_B} L(x,y)_i$$

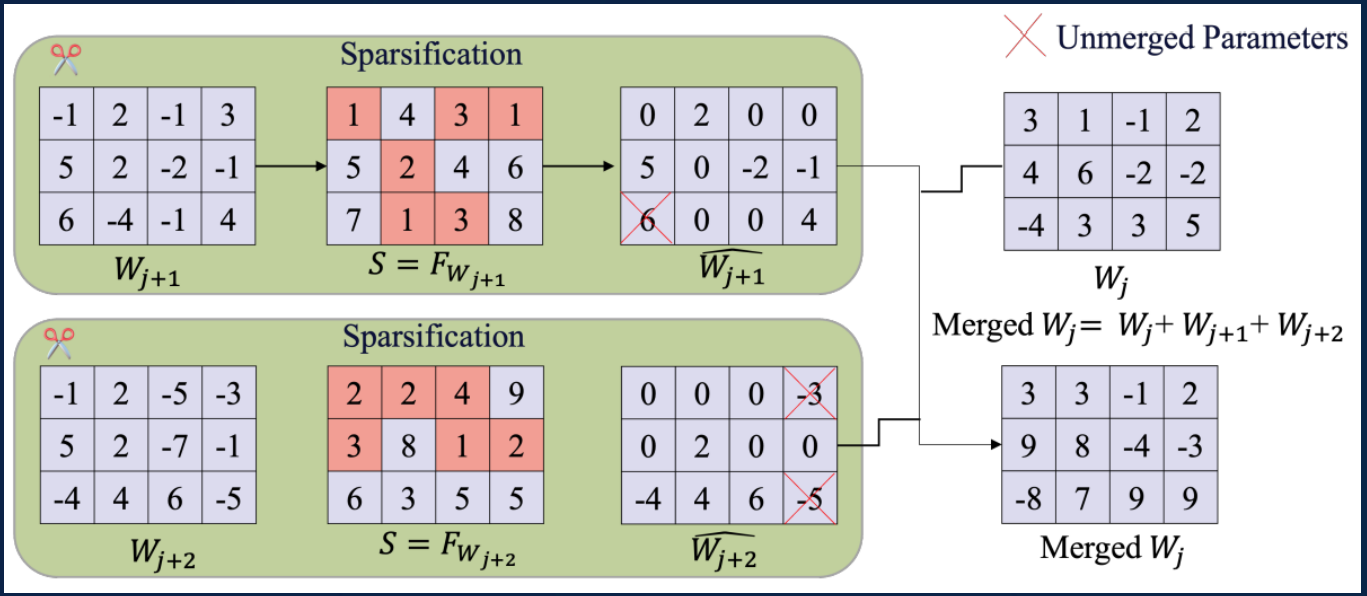
Compute IGIA-Matrix

$$F_W = \frac{1}{t} \sum_{i=1}^{t} (\nabla_W L(x,y)_i)^2$$

Calculate Layer Importance

$$Layer_j = \sum_{k=1}^{M} \sum_{l=1}^{H} F_{W_{kl}}$$

GradPruner : Layer Merging



$$M\ (W_j)_{kl} = \begin{cases} (W_j)_{kl} & if\ (\gamma_j)_{kl} != (\gamma_{j+1})_{kl} != (\gamma_{j+n})_{kl} \\ (W_j)_{kl} + (\hat{W_{j+1}})_{kl} & if\ (\gamma_j)_{kl} == (\gamma_{j+1})_{kl} != (\gamma_{j+n})_{kl} \\ (W_j)_{kl} + (\hat{W_{j+n}})_{kl} & if\ (\gamma_j)_{kl} != (\gamma_{j+1})_{kl} == (\gamma_{j+n})_{kl} \end{cases}$$

Objective

To further increase the pruning ratio without significant accuracy loss by intelligently merging layers.

Step 1: Sparsification

Sparsify pruned layers using the IGIA-Matrix as a criterion, retaining only the most critical parameters.

Step 2: Symbol-Based Merging

Merge sparsified layers with preceding ones. Crucially, only merge elements with the same sign to minimize destructive interference.

蚂蚁集团 ANT GROUP

Experimental Setup

Models

- Llama3.1-8B
- Mistral-7B-v0.3

Datasets (8 Tasks)

Medical Domain

PubMedQA, MedMCQA

Financial Domain

BillSum, FinGPT

General Reasoning

HellaSwag, WinoGrande, ARC, PIQA

Baselines

SOTA Pruning Methods

- APT, SAT, LLMPruner
- LaCo, MINITRON

Direct Fine-tuning

- Llama3.2-3B (Smaller Model)

Overall Performance: 40% Pruning with <1% Accuracy Drop

Key Result: High Efficiency

Achieves 40% parameter reduction with only a 0.99% drop in task accuracy across all datasets.

Superiority: Best Baseline

Consistently outperforms all baseline pruning methods in terms of accuracy retention at the same ratio.

Comparison: Beats Smaller Model

Pruned Llama3.1-8B outperforms the directly fine-tuned Llama3.2-3B on downstream tasks.

Table 1: Main Experimental Results

Method	Params	Accuracy	Drop
Baseline (Full)	100%	89.2%	-
Llama3.2-3B	37.5%	85.4%	3.8%
GradPruner (Ours)	**60%**	**88.21%**	**0.99%**

蚂蚁集团 ANT GROUP

Comparison with SOTA Pruning Methods

Quantitative Advantage

Shows significant accuracy gains (up to ~5%) over LLMPruner, LaCo, and MINITRON.

Outperforms APT/SAT

Consistently better than efficient fine-tuning methods (APT/SAT) in both FFT and LoRA settings.

Robustness Across Models

Improvement is consistent across Llama3.1-8B, Mistral-7B and different fine-tuning paradigms.

Performance Comparison (Table 1)

Method	Llama3.1-8B	Mistral-7B	Avg. Gain
LLMPruner	62.1	60.5	-2.4%
APT/SAT	63.5	61.8	-1.1%
GradPruner (Ours)	**65.8**	**64.2**	**+4.8%**

蚂蚁集团 ANT GROUP

Efficiency Gains in Training and Inference

Training Efficiency Optimization

Training Time

↓ 36%

Training Memory

↓ 34%

Significant reduction compared to dense models, achieving faster convergence without sacrificing accuracy.

Inference Efficiency Optimization

Inference Time

↓ 38%

Inference Memory

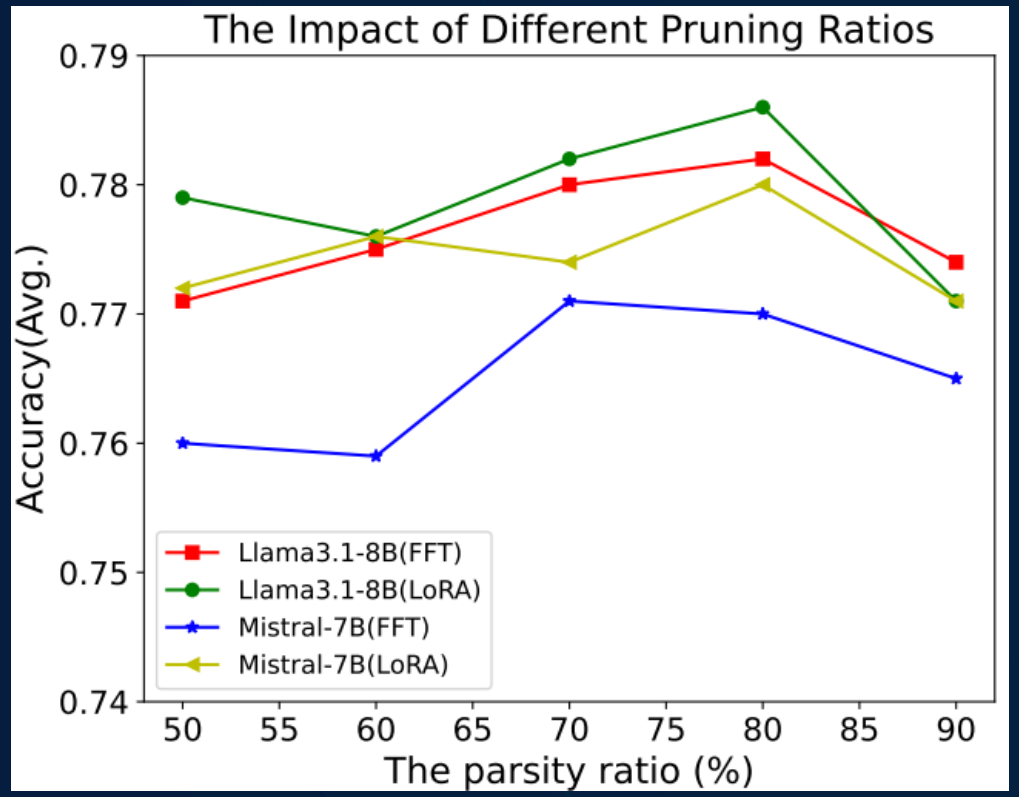
↓ 39%

Highly efficient deployment with reduced latency and memory footprint.

Comparison with State-of-the-Art Methods

GradPruner's efficiency gains are comparable to or better than other methods like LaCo, while maintaining superior accuracy. This demonstrates its robustness as a comprehensive pruning solution.

Ablation Study: Impact of Pruning Ratio



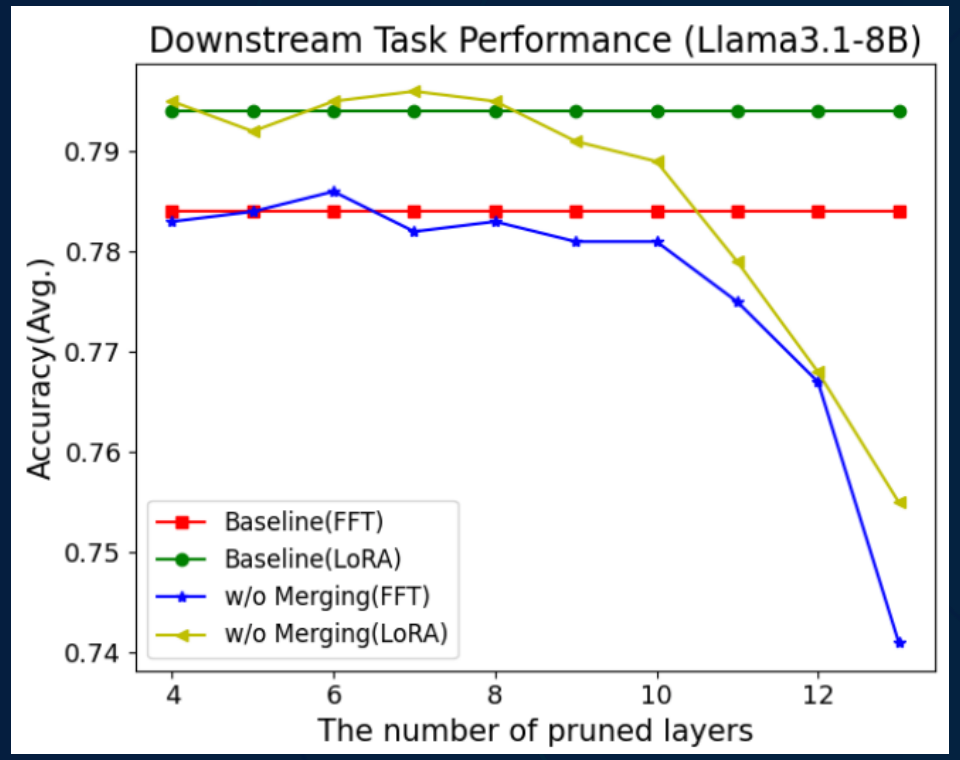
Performance Threshold Finding

Pruning up to 30% of layers has **negligible impact** on accuracy. Beyond this threshold, performance experiences a **sharp drop**.

Optimal Sparsity Configuration

An **80% sparsity rate** on pruned layers (before merging) is found to be optimal for balancing accuracy and pruning efficiency.

Ablation Study: Effectiveness of Layer Merging



Critical Component

Layer merging is crucial for pushing the pruning ratio beyond 30% without significant degradation.

Key Result: 40% Pruning

Enabled pruning of**3 additional layers**(total 40%) while maintaining downstream task accuracy.

Performance Comparison

Figure 5 highlights the stark contrast in accuracy drop between pruning alone and with merging.

蚂蚁集团
ANT GROUP

Conclusion

Novel Measurement

Proposed GradPruner, using initial gradients from LoRA fine-tuning to compute the IGIA-Matrix for evaluating parameter importance on downstream tasks.

Effective Merging

Introduced a sign-based layer merging strategy to maximize the pruning ratio while preserving the model's accuracy.

Strong Results

Achieves **40% parameter reduction** with only **0.99% accuracy drop**, outperforming SOTA baselines across diverse tasks.

蚂蚁集团
ANT GROUP

Future Work

Broader Applicability

Explore extending GradPruner to a wider range of model architectures (e.g., MoE, Vision-Language Models) and task types.

Dynamic Pruning

Investigate dynamic pruning strategies that adaptively adjust the pruning structure based on input content for greater efficiency.

Combination Techniques

Combine GradPruner with quantization, knowledge distillation, or other model compression techniques to further boost efficiency.

Thank You

andacheng.cad@gmail.com

https://github.com/secretflow/ACoLab/tree/main/PaperCode/GradPrune

蚂蚁集团
ANT GROUP