[Paper] [Code]

Towards Green AI in Fine-tuning Large Language Models via Adaptive Backpropagation

Kai Huang[1], Hanyun Yin[2], Heng Huang[3], Wei Gao[1]

University of Pittsburgh[1], University of Science and Technology of China[2], University of Maryland College Park[3]

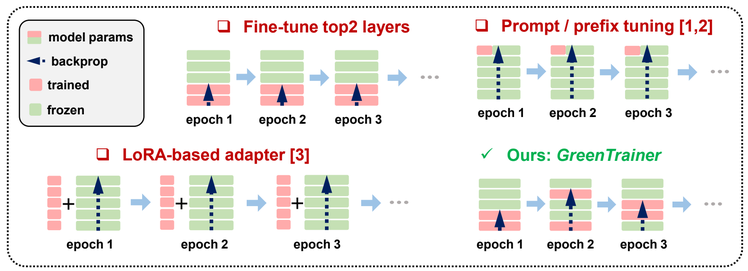
Overview

❖ **Our Goal:** Computationally efficient LLM fine-tuning towards Green AI

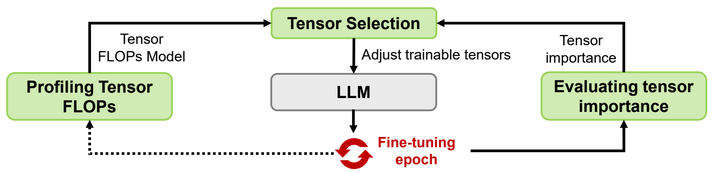
Many released LLMs — Intensive fine-tuning — Massive carbon emission

❖ **Address key limitation of existing work:** They lack accurate modeling for the backprop cost and cannot effectively reduce such cost

✓ *Adaptive backprop according to the FLOPs constraint*



❖ **GreenTrainer at high-level:** selectively fine-tuning model tensors to achieve desired training FLOPs reduction and retain accuracy.



Problem Formulation

❖ **GreenTrainer maximizes the training loss reduction while achieving the desired FLOPs reduction,** as a constrained optimization problem:

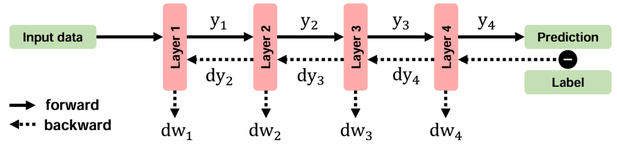
$$\max \Delta_{loss}(\boldsymbol{m}) \quad s.t. T_{\text{selective}}(\boldsymbol{m}) \leq \rho T_{\text{full}}$$

where $\boldsymbol{m}$ is a binary vector to be solved for tensor selection. $\boldsymbol{m}$ parameterizes both the loss reduction (Δ_{loss}) and per-batch FLOPs of training ($T_{\text{selective}}$), and $T_{\text{selective}}$ is constrained within a user-specified ratio (ρ) of the FLOPs of full fine-tuning (T_{full}).

Profiling Tensor FLOPs

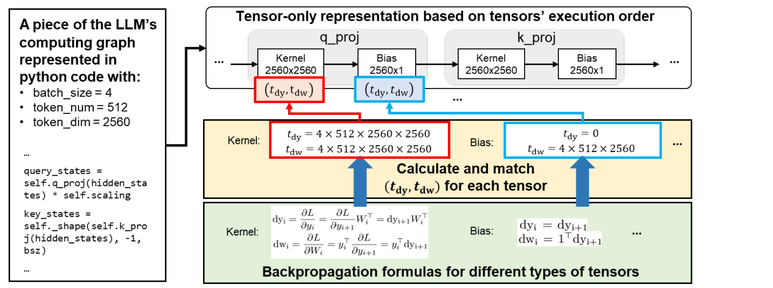
❖ **FLOPs modeling for backprop:** Backprop FLOPs in training can be decomposed into two parts using the chain rule. For example, when training a 4-layer dense NN without bias, each layer computes:

- dy_i as the loss's gradient w.r.t the activation y_i
- dw_i as the loss's gradient w.r.t the weight w_i



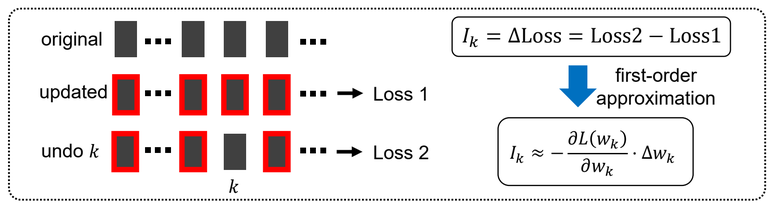
Even if a layer is not selected in fine-tuning, it still needs to compute and pass activation gradients to downstream layers. Based on this rationale, we can construct FLOPs models for LLM substructures. We adopt tensor-level selection to balance efficiency and granularity.

❖ **Profiling Tensors FLOPs of Multi-Head Attention Module (an example)**



Evaluating Tensor Importance

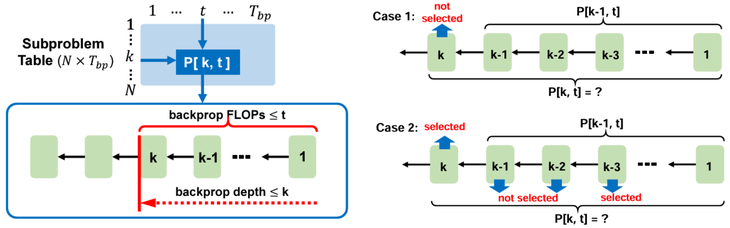
❖ **Tensor importance:** Defined as the training loss reduction by this tensor's update. It can be rigorously computed as how much training loss increases back if we undo the tensor's update.



Tensor Selection

❖ **Dynamic Programming (DP):** We define subproblems with downscaled backprop depth and FLOPs reduction objectives. The recursion relation is decided by discussing whether to select the new tensor in the next subproblem or not.

✓ Our DP algorithm is performed at runtime with negligible overhead.



Performance Evaluation

❑ **LLMs:** OPT, BLOOMZ, FLAN-T5 (2.7B~6.7B)

❑ **Datasets:** SciTLDR, DialogSum, PIQA, WebQuestions

❑ **Baselines:** Full fine-tuning (Full FT), Fine-tune top2 layers, Prefix Tuning [1], LoRA [3]

GreenTrainer (GT) can **save up to 40-60% training FLOPs** and wall-clock time without noticeable accuracy loss. With on-par training FLOPs budget, GT can **improve accuracy by up to 4%** compared to baselines.

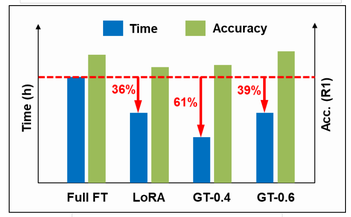
Method	Accuracy (%)	PFLOPs	Time (h)
LoRA	49.5	174.0	6.27
GT-0.5	59.2	130.5	4.69

OPT-2.7B on PIQA dataset

Method	Accuracy (%)	PFLOPs	Time (h)
LoRA	19.6	16.0	0.55
GT-0.5	28.7	12.0	0.50
GT-0.6	29.5	14.0	0.61

OPT-2.7B on WebQuestion dataset

GreenTrainer with different FLOPs objective (ρ)



OPT-2.7B on SciTLDR dataset

Method	SciTLDR		
	PFLOPs	Time (h)	R1/R2/RL
Full FT	41.8	0.92	32.9/14.9/27.1
LoRA	27.9 (33%↓)	0.59 (36%↓)	28.2/12.1/21.0
GT-0.36	14.9 (64%↓)	0.32 (65%↓)	4.1/1.7/3.6
GT-0.4	16.6 (60%↓)	0.36 (61%↓)	28.6/11.6/23.5
GT-0.5	20.8 (50%↓)	0.46 (50%↓)	30.5/13.1/25.2
GT-0.6	25.0 (40%↓)	0.56 (39%↓)	33.4/15.3/27.8
GT-0.7	29.2 (30%↓)	0.68 (26%↓)	33.1/15.2/27.6
GT-0.8	33.4 (20%↓)	0.77 (16%↓)	33.1/15.5/27.6

OPT-2.7B on SciTLDR dataset

References

[1] Li, Xiang Lisa, and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation." ACL 2021

[2] Lester, Brian, Rami Al-Rfou, et al.. "The power of scale for parameter-efficient prompt tuning." EMNLP 2021

[3] Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." ICLR 2022