Fine-Tuning Language Models with Just Forward Passes

Source: https://arxiv.org/abs/2305.17333

Qitong Zhao

Introduction

- Fine-tuning pre-trained models (LMs) has been the dominant methodology for solving many language tasks, adapting to specialized domains, or incorporating human instructions and preferences.
- However, we are facing a problem that when LMs are scaled up, computing gradients for backpropagation requires a huge amounts of memory. It could reach up to 12 times more memory than inference. The reason is that it needs store activations during the forward pass, gradients during the backward pass, and also store gradient history, in the case of Adam.

ZO-SGD

- A classical zeroth-order optimization method (ZO-SGD) uses only differences of loss values to estimate the gradients
- ZO methods have but not to directly optimize large-scale models.

Our work

- Memory-efficient zeroth-order optimizer (MeZO): It adapts the classical ZO-SGD algorithm and reduces its memory consumption to the same inference.
- Generally, we apply MeZO to fine-tune large LMs and show that MeZO can successfully optimize LMs with billions of parameters. More specifically…

More Specifically…

- In MeZO, we adapt the ZO-SGD algorithm and a number of variants to operate in-place on arbitrarily large models with almost no memory overhead.
- We conduct comprehensive experiments across model types (masked LM and autoregressive LM), model scales (from 350M to 66B), and downstream tasks (classification, multiple-choice, and generation**). MeZO consistently demonstrates superiority over zero-shot, ICL, and linear probing**. Moreover, with RoBERTa-large, **MeZO achieves performance close to standard fine-tuning within 5% gap; with OPT-13B, MeZO outperforms or performs comparably to fine-tuning on 7 out of 11 tasks, despite requiring roughly 12× less memory**
- We demonstrate MeZO's compatibility with full-parameter tuning and PEFT.
- Further exploration showcases that MeZO can optimize non-differentiable objectives such as accuracy or F1 score, while still requiring only the same memory as inference.
- Our theory suggests that adequate pre-training ensures the per-step optimization rate and global convergence rate of MeZO depend on a certain condition number of the landscape instead of numbers of parameters. This result is in sharp contrast to existing ZO lower suggesting that the convergence rate can slow proportionally to the number of parameters.

MeZO Pseudo-code

Require: parameters $\boldsymbol{\theta} \in \mathbb{R}^d$, loss $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$, step budget T, perturbation scale ϵ, batch size B learning rate schedule $\{\eta_t\}$

for $t = 1, ..., T$ **do**
  Sample batch $\mathcal{B} \subset \mathcal{D}$ and random seed s
  $\boldsymbol{\theta} \leftarrow \texttt{PerturbParameters}(\boldsymbol{\theta}, \epsilon, s)$
  $\ell_+ \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})$
  $\boldsymbol{\theta} \leftarrow \texttt{PerturbParameters}(\boldsymbol{\theta}, -2\epsilon, s)$
  $\ell_- \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})$
  $\boldsymbol{\theta} \leftarrow \texttt{PerturbParameters}(\boldsymbol{\theta}, \epsilon, s)$ ▷ Reset parameters before descent
  $\texttt{projected\_grad} \leftarrow (\ell_+ - \ell_-)/(2\epsilon)$
  Reset random number generator with seed s ▷ For sampling z
  for $\theta_i \in \boldsymbol{\theta}$ **do**
    $z \sim \mathcal{N}(0, 1)$
    $\theta_i \leftarrow \theta_i - \eta_t * \texttt{projected\_grad} * z$
  end
end

Subroutine $\texttt{PerturbParameters}(\boldsymbol{\theta}, \epsilon, s)$
  Reset random number generator with seed s ▷ For sampling z
  for $\theta_i \in \boldsymbol{\theta}$ **do**
    $z \sim \mathcal{N}(0, 1)$
    $\theta_i \leftarrow \theta_i + \epsilon z$ ▷ Modify parameters in place
  end
  return $\boldsymbol{\theta}$

SPSA and ZO-SGD

Consider a labelled dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i \in [|\mathcal{D}|]}$ and a minibatch $\mathcal{B} \subset \mathcal{D}$ of size B, we let $\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})$ denote the loss on the minibatch. We introduce a classical ZO gradient estimate in this setting.

Definition 1 (Simultaneous Perturbation Stochastic Approximation or SPSA [83]). *Given a model with parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ and a loss function $\mathcal{L}$, SPSA estimates the gradient on a minibatch $\mathcal{B}$ as*

$$\widehat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) = \frac{\mathcal{L}(\boldsymbol{\theta} + \epsilon \boldsymbol{z}; \mathcal{B}) - \mathcal{L}(\boldsymbol{\theta} - \epsilon \boldsymbol{z}; \mathcal{B})}{2\epsilon} \boldsymbol{z} \approx \boldsymbol{z}\boldsymbol{z}^\top \nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) \tag{1}$$

where $\boldsymbol{z} \in \mathbb{R}^d$ with $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}_d)$ and ϵ is the perturbation scale. *The n-SPSA gradient estimate averages $\widehat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})$ over n randomly sampled $\boldsymbol{z}$.*

Definition 2 (ZO-SGD). *ZO-SGD is an optimizer with learning rate η that updates parameters as $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \widehat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B}_t)$ where $\mathcal{B}_t$ is the minibatch at time t and $\widehat{\nabla}\mathcal{L}$ is the SPSA gradient estimate.*

Experiment 1 from RoBERTa-large

- **Process**: We conduct comprehensive experiments on both medium-sized masked LMs (RoBERTa-large, 350M) and large autoregressive LMs (OPT-13B, 30B, 66B) in few-shot and many-shot settings with prompts. We also explore both full-parameter tuning and PEFT including LoRA and prefix-tuning. We compare MeZO with zero-shot, in-context learning (ICL), linear-probing (LP), and fine-tuning with Adam (FT). MeZO uses substantially less memory than FT but requires significantly more training steps.
- **Goal**: We first show that MeZO improves substantially over zero-shot, ICL, and LP across model types, sizes, and task types. Moreover, MeZO performs comparably to FT over a number of tasks, while drastically reducing the memory cost by, for example, 12× on OPT-13B. Further experiments demonstrate that MeZO can optimize non-differentiable objectives, such as accuracy and F1 score. We compare the memory consumption of ICL, FT, LP, and MeZO in the following figures.

Results

Task	SST-2	RTE	CB	BoolQ	WSC	WIC	MultiRC	COPA	ReCoRD	SQuAD	DROP
Task type	classification							– multiple choice –		— generation —	
Zero-shot	58.8	59.6	46.4	59.0	38.5	55.0	46.9	80.0	81.2	46.2	14.6
ICL	87.0	62.1	57.1	66.9	39.4	50.5	53.1	87.0	**82.5**	75.9	29.6
LP	**93.4**	68.6	67.9	59.3	63.5	60.2	63.5	55.0	27.1	3.7	11.1
MeZO	91.4	66.1	67.9	67.6	63.5	**61.1**	60.1	**88.0**	81.7	**84.7**	30.9
MeZO (LoRA)	89.6	67.9	66.1	**73.8**	**64.4**	59.7	61.5	87.0	81.4	83.8	**31.4**
MeZO (prefix)	90.7	**70.8**	**69.6**	73.1	57.7	59.9	**63.7**	84.0	81.2	84.2	28.9
FT (12x memory)	92.0	70.8	83.9	77.1	63.5	70.1	71.1	79.0	74.1	84.9	31.3

Table 1: Experiments on OPT-13B (with 1,000 examples). ICL: in-context learning; LP: linear probing; FT: full fine-tuning with Adam. MeZO outperforms zero-shot, ICL, and LP across the board, and achieves comparable (within 1%) or better performance than FT on 7 out of 11 tasks.

Task	SST-2	RTE	BoolQ	WSC	WIC	SQuAD
30B zero-shot	56.7	52.0	39.1	38.5	50.2	46.5
30B ICL	81.9	66.8	66.2	56.7	51.3	78.0
30B MeZO/MeZO (prefix)	**90.6**	**72.6**	**73.5**	**63.5**	**59.1**	**85.2**
66B zero-shot	57.5	**67.2**	66.8	43.3	50.6	48.1
66B ICL	89.3	65.3	62.8	52.9	54.9	81.3
66B MeZO/MeZO (prefix)	**93.6**	66.4	**73.7**	**63.5**	**58.9**	**85.0**

Table 2: Experiments on OPT-30B and OPT-66B (with 1,000 examples). We report the best of MeZO and MeZO (prefix). See Appendix E.2 for more results. We see that on most tasks MeZO effectively optimizes up to 66B models and outperforms zero-shot and ICL.

Results Summary

- MeZO works significantly better than zero-shot, linear probing, and other memory-equivalent methods.
- With enough data, MeZO achieves comparable performance (up to 5% gap) to FT.
- MeZO works well on both full-parameter tuning and PEFT.

Experiment 2 from OPT Family

we extend MeZO to the OPT family, on a scale of 13B, 30B, and 66B. We select both SuperGLUE tasks4 (including classification and multiple-choice) and generation tasks. We randomly sample 1000, 500, and 1000 examples for training, validation, and test respectively for each dataset.

Results

Model	RoBERTa-large (350M)				OPT-13B
Task	**SST-2**	**SST-5**	**SNLI**	**TREC**	**SQuAD**
Zero-shot	79.0	35.5	50.2	32.0	46.2
Cross entropy (FT)	93.9	55.9	88.7	97.3	84.2
Cross entropy (MeZO)	93.3	53.2	83.0	94.3	84.7
Accuracy/F1 (MeZO)	92.7	48.9	82.7	68.6	78.5

Table 3: MeZO with non-differentiable objectives. For classification ($k = 512$), we use MeZO with full-parameter and optimize accuracy; for SQuAD (1,000 examples), we use MeZO (prefix) and F1.

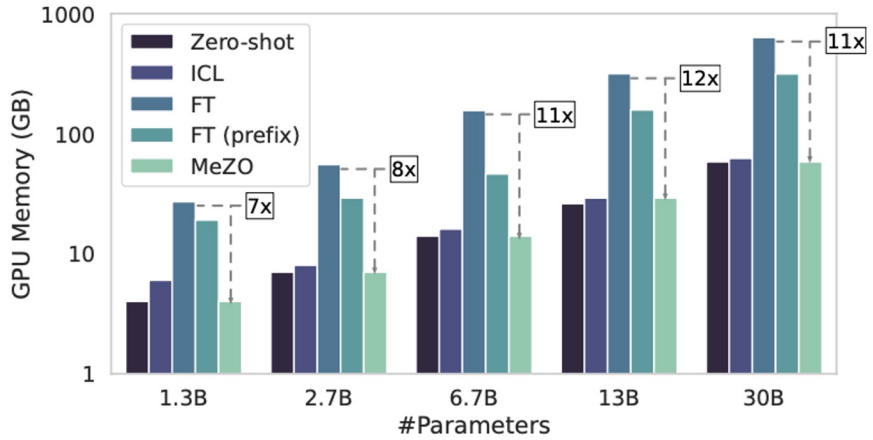

Figure 3: GPU memory consumption with different OPT models and tuning methods on MultiRC (400 tokens per example on average).

Hardware	Largest OPT that can fit		
	FT	**FT-prefix**	**MeZO**
1×A100 (80GB)	2.7B	6.7B	30B
2×A100 (160GB)	6.7B	13B	66B
4×A100 (320GB)	13B	30B	66B
8×A100 (640GB)	30B	66B	175B†

Figure 4: Largest OPT models that one can tune with specific hardwares and algorithms. † : projected results without actual testing.

Results Summary

- MeZO outperforms memory-equivalent methods and closely approaches fine-tuning results.
- MeZO exhibits strong performance across classification, multiple-choice, and generation tasks.
- MeZO Scales up to 66 billion parameter models: While directly fine-tuning models at such scales are extremely costly, MeZO can effectively optimize these models and outperform zero-shot and ICL.

Theory

Highlights that MeZO can optimize large LMs, although optimization should be catastrophically slow when training so many parameters.

- When the loss landscape exhibits favorable conditions, we can derive a convergence rate independent of the number of parameters.
- The loss decreases per step at a rate independent of the parameter dimension *d*, and that, under stronger conditions, the algorithm converges in time independent of *d*.

Some Definitions and Lemmas

Definition 3 (Unbiased Gradient Estimate)**.** *Any minibatch gradient estimate* $\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B})$ *is said to be unbiased if* $\mathbb{E}[\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B})] = \nabla \mathcal{L}(\boldsymbol{\theta})$.

Lemma 1 (Descent Lemma)**.** *Let* $\mathcal{L}(\boldsymbol{\theta})$ *be* ℓ*-smooth.*[6] *For any unbiased gradient estimate* $\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B})$,

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \leq -\eta \left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\right\|^2 + \frac{1}{2}\eta^2 \ell \cdot \mathbb{E}[\left\|\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B}_t)\right\|^2]. \tag{2}$$

The descent lemma highlights the importance of the gradient norm, which we derive for MeZO below.

Lemma 2. *Let* $\mathcal{B}$ *be a random minibatch of size* B*. Then, the gradient norm of MeZO is*

$$\mathbb{E}_x \left[\left\| \widehat{\nabla} \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) \right\|^2 \right] = \frac{d+n-1}{n} \mathbb{E} \left[\left\| \nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) \right\|^2 \right].$$

where n *is the number of* $\boldsymbol{z}$ *sampled in* n*-SPSA (Definition 1) and* d *is the number of parameters.*

Analysis

- When $n \ll d$, MeZO has a much larger gradient norm than SGD. Also, the descent lemma also shows that to guarantee loss decrease, one needs to choose the learning rate as

$$\eta \leq \frac{2\left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\right\|^2}{\ell \cdot \mathbb{E}[\|\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B})\|^2]} \quad \xRightarrow{\text{Lemma 2}} \quad \eta_{\mathrm{ZO}} = \frac{n}{d+n-1}\eta_{\mathrm{SGD}}$$

,where η_{ZO} and η_{SGD} are the maximum permissible learning rates for MeZO and SGD respectively.

- Hence, we can see that without any further assumptions, MeZO can slow optimization by decreasing the largest
- MeZO reduces the loss decrease that can be obtained at each step and, as a consequence, slows convergence by a factor of *d* as well.

Continued…

Assumption 1 (Local r-effective rank)**.** *Let* $G(\boldsymbol{\theta}_t) = \max_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \|\nabla\mathcal{L}(\boldsymbol{\theta}_t;\{(\boldsymbol{x},\boldsymbol{y})\})\|$. *There exists a matrix* $\boldsymbol{H}(\boldsymbol{\theta}_t)$ *such that:*

1. *For all* $\boldsymbol{\theta}$ *such that* $\|\boldsymbol{\theta}-\boldsymbol{\theta}_t\| \le \eta d G(\boldsymbol{\theta}_t)$, *we have* $\nabla^2\mathcal{L}(\boldsymbol{\theta}) \preceq \boldsymbol{H}(\boldsymbol{\theta}_t)$.
2. *The effective rank of* $\boldsymbol{H}(\boldsymbol{\theta}_t)$, *i.e* $\mathrm{tr}(\boldsymbol{H}(\boldsymbol{\theta}_t))/\|\boldsymbol{H}(\boldsymbol{\theta}_t)\|_{op}$, *is at most* r.

Under this assumption, we show that the convergence rate of ZO-SGD does not depend on the number of parameters. Instead, the slowdown factor only depends on the effective rank of the Hessian.

Theorem 1 (Dimension-Free Rate)**.** *Assume the loss exhibits local* r*-effective rank (Assumption 1). If* $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_{ZO}\widehat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})$ *is a single step of ZO-SGD using the* n*-SPSA estimate with a minibatch of size* B, *then there exists a* $\gamma = \Theta(r/n)$ *such that the expected loss decrease can be bounded as*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \le -\eta_{ZO}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{2}\eta_{ZO}^2\ell\cdot\gamma\cdot\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\|^2] \tag{4}$$

By applying Equation (3), we can directly compare to the SGD descent lemma.

Corollary 1. *Choosing the learning rate* $\eta_{ZO} = \gamma^{-1}\cdot\eta_{SGD}$, *ZO-SGD obtains a loss decrease of*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \le \frac{1}{\gamma}\cdot\left[-\eta_{SGD}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{2}\eta_{SGD}^2\ell\cdot\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\|^2]\right]. \tag{5}$$

Here we see that comparing to SGD, the slowdown factor of ZO-SGD scales with the local effective rank r, which we argue is much smaller than the number of parameters d.

Global Convergence Analysis

- The global convergence rate also slows by a factor proportional to the local effective rank under stronger assumptions about the loss landscape.
- We assume that the landscape obeys the classical PL inequality: the gradient norm grows quadratically with the suboptimality of the iterate.

Global Convergence Analysis

Definition 4 (PL Inequality). *Let* $\mathcal{L}^* = \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$. *The loss* $\mathcal{L}$ *is* μ*-PL if, for all* $\boldsymbol{\theta}$, $\frac{1}{2} \|\nabla \mathcal{L}(\boldsymbol{\theta})\|^2 \geq \mu(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*)$.

Definition 5 (Gradient Covariance). *The SGD gradient estimate on a minibatch of size* B *has covariance* $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = B(\mathbb{E}\left[\nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) \nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})^\top\right] - \nabla \mathcal{L}(\boldsymbol{\theta}) \nabla \mathcal{L}(\boldsymbol{\theta})^\top)$.

Lemma 3 (Global Convergence of ZO-SGD). *Let* $\mathcal{L}(\boldsymbol{\theta})$ *be* μ*-PL and let there exist* α *such that* $\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta})) \leq \alpha(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*)$ *for all* $\boldsymbol{\theta}$. *Then after*

$$t = \mathcal{O}\left(\left(\frac{r}{n} + 1\right) \cdot \underbrace{\left(\frac{\ell}{\mu} + \frac{\ell\alpha}{\mu^2 B}\right) \log \frac{\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*}{\epsilon}}_{\text{SGD rate (Lemma 4)}}\right)$$

iterations of ZO-SGD we have $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] \leq \mathcal{L}^* + \epsilon$.

Related Work

- Zeroth-order optimization
- Memory-efficient backpropagation
- Gradient-free adaptation of large language models

Conclusion

- MeZO can effectively optimize large LMs across many tasks and scales.
- Experiments suggest that MeZO can optimize non-differentiable objectives, which backpropagation usually cannot do.
- Our theory illustrates why MeZO is not catastrophically slow when tuning billions of parameters. As a limitation, MeZO takes many steps in order to achieve strong performance.