Do You Want to Fine-tune a 175B LLM with a 24GB GPU?

King Abdullah University of Science and Technology

PRADA Lab

Liangyu Wang, Jie Ren, Hang Xu, Junxiao Wang, David E. Keyes, Di Wang

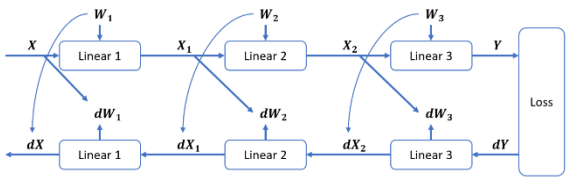
ZO (Zeroth-order Optimizer) & Motivation

Given a loss function $f(\cdot)$ and a model x with parameters in d dimensions, the gradient $\hat{\nabla} f(x)$ can be estimated by:

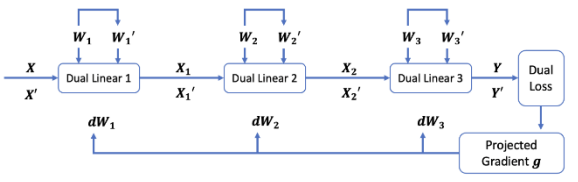
$$\hat{\nabla} f(x) = gz \in \mathbb{R}^d,$$

$$g = \frac{f(x+\epsilon z) - f(x-\epsilon z)}{2\epsilon} \in \mathbb{R}^1,$$

where z is a random direction vector drawn from the standard Gaussian Distribution $N(0, I)$, and ε is a small perturbation step size.

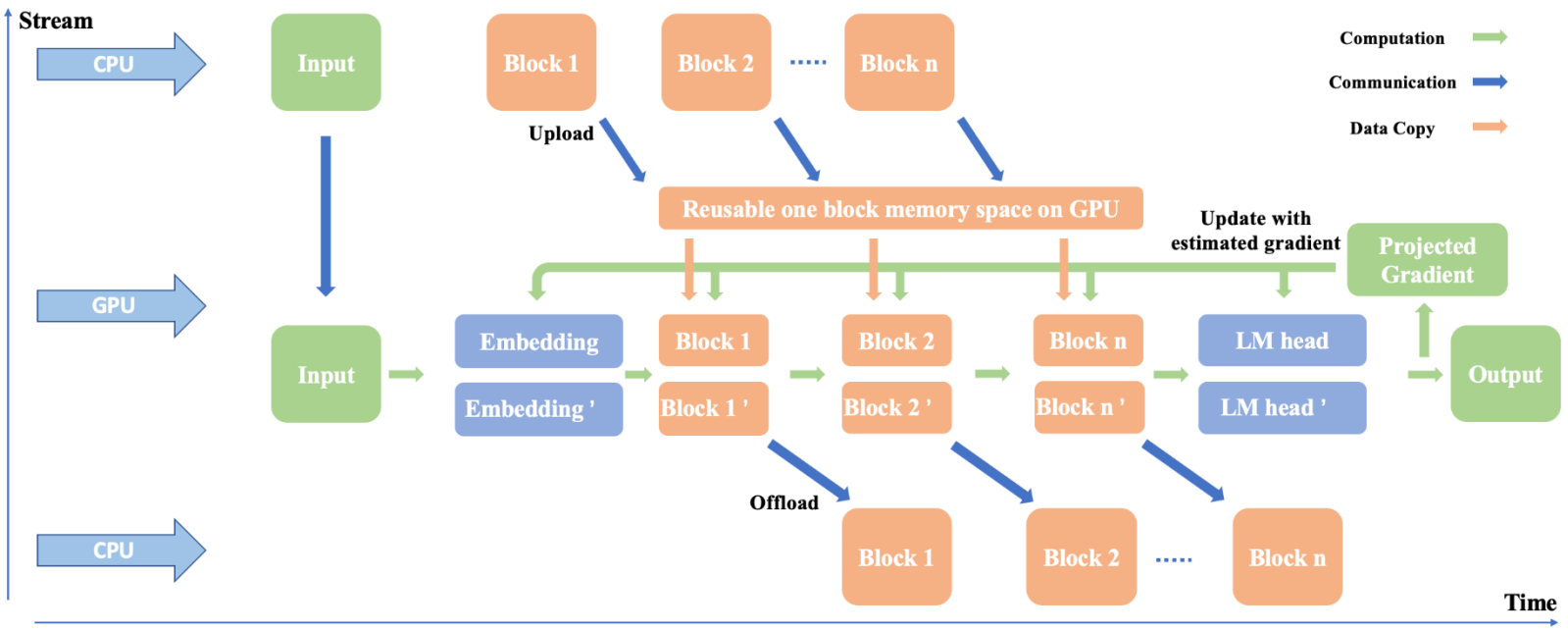


(a) Model using first-order optimizer with forward-backward passes workflow



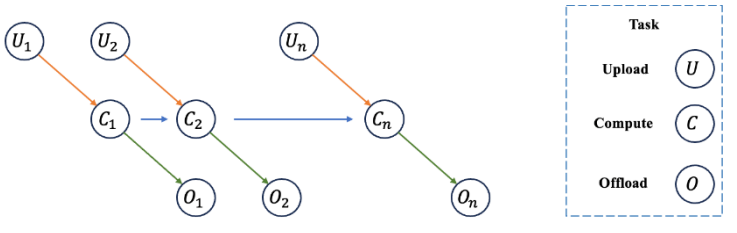
(b) Model using zero-order optimizer with only forward passes workflow

ZO-Offloading Framework



Dynamic Scheduler for ZO-Offloading

(a) ZO-Offloading without overlap



(b) ZO-Offloading with overlap

Algorithm 1 ZO-Offloading Dynamic Scheduler

Require: Transformer blocks $\{W_i\}_{i=1}^N$ with number of transformer blocks N, embedding parameters $Embedding$, and LM head $LMhead$.

Initialize a dynamic scheduler $S\{\cdot\}$ to control dual forward computation $C(\cdot)$, uploading $U(\cdot)$, and offloading $O(\cdot)$ operations.
Asynchronously launch $S\{U(W_1), C(Embedding)\}$.
for $i = 1$ to $N - 1$ **do**
  Synchronously wait until $U(W_i)$ finished.
  if $i = 1$ **then**
    Asynchronously launch $S\{U(W_{i+1}), C(W_i)\}$.
  else
    Synchronously wait until $C(W_{i-1})$ finished.
    Asynchronously launch $S\{U(W_{i+1}), C(W_i), O(W_{i-1})\}$.
  end if
end for
Synchronously wait until $U(W_N)$ and $C(W_{N-1})$ finished.
Asynchronously launch $S\{C(W_N), O(W_{N-1})\}$.
Synchronously wait until $C(W_N)$ finished.
Asynchronously launch $S\{C(LMhead), O(W_N)\}$.

Experiment Results Compared with MeZO[1]

Model	GPU Memory Usage (MB) ↓				Throughput (tokens/sec) ↑			
	MeZO(32)	ZO-Offload(32)	MeZO(16)	ZO-Offload(16)	MeZO(32)	ZO-Offload(32)	MeZO(16)	ZO-Offload(16)
OPT-125M	3091	2941(x0.95)	1801(x0.58)	**1661(x0.54)**	14889	13074(x0.89)	**31058(x2.09)**	**31058(x2.09)**
OPT-350M	4219	3393(x0.81)	2389(x0.57)	**1643(x0.39)**	5274	5099(x0.97)	**13508(x2.56)**	12284(x2.32)
OPT-1.3B	9117	4413(x0.48)	4887(x0.54)	**2651(x0.29)**	1954	1954(x1.00)	**6788(x3.47)**	**6788(x3.47)**
OPT-2.7B	15277	5261(x0.34)	7933(x0.52)	**3111(x0.20)**	1087	1087(x1.00)	**4227(x3.89)**	**4227(x3.89)**
OPT-6.7B	32083	8329(x0.26)	16311(x0.51)	**4539(x0.14)**	499	499(x1.00)	**2455(x4.92)**	**2455(x4.92)**
OPT-13B	58251	12113(x0.21)	29411(x0.50)	**6445(x0.11)**	270	270(x1.00)	**1406(x5.21)**	1340(x4.96)
OPT-30B	-	18879	63953	**10369**	-	122	**651**	597
OPT-66B	-	29937	-	**14143**	-	40	-	**273**
OPT-175B	-	49203	-	**24667**	-	14	-	**37**

[1] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. 2023. Fine-tuning language models with just forward passes. Advances in Neural Information Processing Systems 36 (2023), 53038– 53075.