COALA: Numerically Stable and Efficient Framework for Context-Aware Low-Rank Approximation

Uliana Parkina[1] Maxim Rakhuba[1]

[1]HSE University

Introduction

To compress the weight matrix W of a large-language model using calibration data X, we consider the objective

$$\min_{\operatorname{rank}(W')\le r} \|(W - W')X\|_F, \tag{1}$$

where $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ with a low-rank W' that reproduces the same outputs for the current activations $X \in \mathbb{R}^{d_{\text{in}} \times \text{batch\_size}}$.

Our contribution

- We remove the numerical-stability issues of previous methods by introducing inversion-free formulas.
- We provide a more efficient solution based on QR factorization, which can be readily parallelized on GPUs.

WLA does not require matrix inversion!

Conventional route to solve problem

$$W' \;=\; \mathrm{SVD}_r(WS)\, S^{-1}, \qquad S \;=\; (X^\top X)^{1/2}.$$

- Needs X to have full row rank
- Unstable: evaluating $XX^\top$ leads to a loss of precision
- Expensive: must form $XX^\top$ and take its square root

Our route to solve problem

$$W' \;=\; U_r U_r^\top W, \qquad U_r \Sigma_r V_r^\top = \mathrm{SVD}_r(WX)$$

If $d_{\text{in}} \le$ batch_size, compute U_r by taking the SVD of $WR^\top$, where R is the upper-triangular factor from the QR decomposition of $X^\top$.

In exact arithmetic, these are identical – even if you can't see it!

Stability matters

Large approximation errors for other methods:

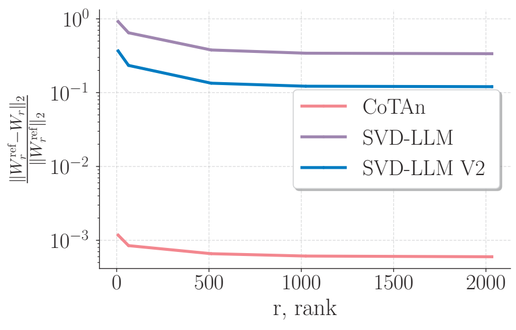


Figure 1. Relative error vs. rank for various methods.

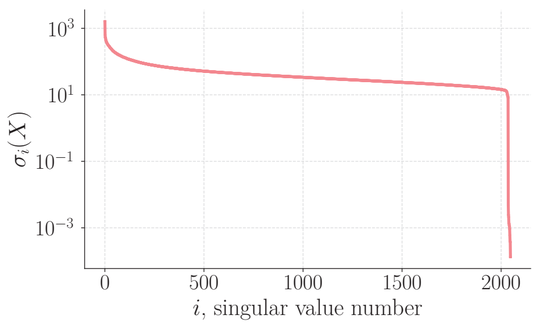


Figure 2. See what happens with $\sigma_i(X)$

Let's add regularization

$$\min_{\operatorname{rank}(W')\le r} \|WX - W'X\|_F^2 + \mu\|W' - W\|_F^2. \tag{2}$$

- Prevents overfitting on limited datasets
- Additionally boosts robustness
- Converges to a desired solution when $\mu \to 0$

What is the limit of W_μ as $\mu \to 0$?

Suppose that X has $\operatorname{rank}(X) = k \ge r$ and that $\sigma_r(WX) \ne \sigma_{r+1}(WX)$. Then, if the solution W_0 to the problem (1), and if W_μ denotes the solution to the regularized problem (2):

$$\|W_0 - W_\mu\|_F \le \frac{2\|W\|_2^2\|W\|_F\left(\dfrac{\sigma_1(X)}{\sigma_k(X)} + \max\left(1,\ \dfrac{\mu}{4\,\sigma_k^2(X)}\right)\right)}{\sigma_r^2(WX) - \sigma_{r+1}^2(WX)} \cdot \mu.$$

Efficiency

QR is still the quickest initializer for large-model compression, even with highly unbalanced matrices.

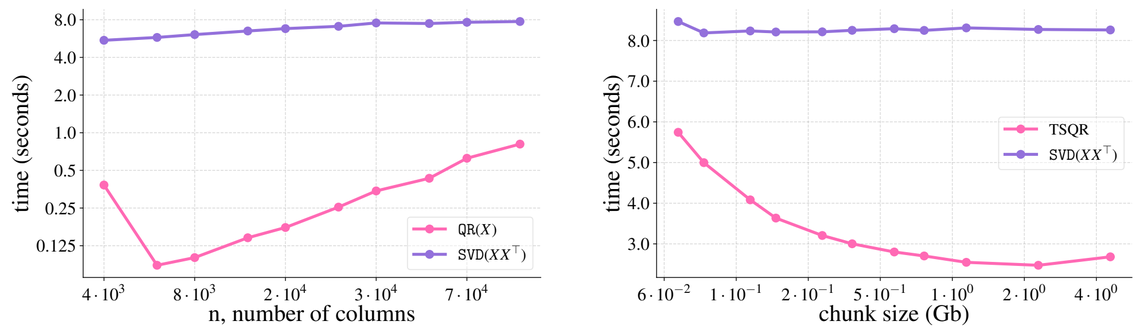
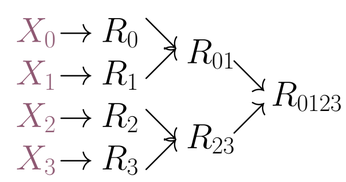


Figure 3. Runtimes for computing S: $SS^\top = XX^\top$ using two approaches. *Left:* Matrix $X \in \mathbb{R}^{4096\times n}$ for different n. *Right:* Matrix $X \in \mathbb{R}^{4096\times 3\cdot 10^5}$ split into chunks of different size. In this case, QR is computed using the TSQR method and the Gram matrix using $XX^\top = \sum_{i=1}^{p} X_i X_i^\top$.

TSQR: block-wise QR, accelerated on multiple GPUs for huge matrices.



Experiments

Table 1. Metric values of various compression methods. Experiments were conducted using the *Mistral-7B* model on the WikiText2 dataset and commonsense reasoning used for validation. The results for SliceGPT and FLAP.

Ratio	Method	MMLU	BoolQ	PIQA	WiNoG	HSweg	ARC-E	ARC-C	OBQA
0%	Mistral-7B	62.50	83.98	82.05	73.95	81.02	79.55	53.92	44.00
80%	FLAP	25.90	62.26	72.31	64.09	55.94	51.05	31.91	36.80
	SliceGPT	28.60	37.86	60.66	59.43	45.10	48.15	30.03	32.00
	SVD-LLM	41.80	68.29	73.39	68.43	61.75	71.34	40.53	36.60
	SoLA	**44.20**	66.09	73.67	68.75	63.32	69.99	39.76	39.20
	COALA	41.20	**78.07**	**77.04**	**68.82**	**65.06**	**72.13**	**43.43**	**40.20**
70%	FLAP	26.40	65.26	69.59	64.80	55.61	48.91	30.55	35.80
	SliceGPT	25.00	37.83	54.41	51.62	32.54	35.02	22.95	26.80
	SVD-LLM	28.20	64.62	64.91	64.17	47.36	58.25	30.72	34.20
	SoLA	**33.80**	62.57	68.39	**64.48**	53.00	60.90	32.76	37.60
	COALA	27.35	**63.82**	**70.40**	62.43	**51.02**	**63.63**	**35.49**	**36.00**

Table 2. Table reports the performance of LLaMA-3 1B-Instruct fine-tuned with rank-8 PEFT on the commonsense-reasoning set, using 24 examples for initialization. Under exact arithmetic, COALA ($\alpha = 2$) is identical to CorDA.

Method	BoolQ	PIQA	SIQA	HSweg	WiNoG	ARC-e	ARC-c	OBQA	Avg.
LoRA	**64.5**	**76.1**	71.5	82.4	53.8	76.8	58.5	68.2	75.0
PiSSA	**64.5**	76.0	71.5	**83.0**	52.0	**78.4**	**60.8**	**70.4**	75.4
CorDA	61.4	68.7	62.1	60.8	52.4	68.7	40.1	52.8	60.9
COALA $\alpha = 2$	64.4	75.9	72.6	82.7	54.3	78.2	59.5	68.0	75.4
COALA $\alpha = 1$	64.1	**76.1**	**72.8**	82.8	**56.0**	77.5	59.8	68.4	**75.5**

What about the PEFT?

The solution to the optimization problem

$$\min_{\operatorname{rank}(W')\le r} \operatorname{tr}\left((W - W')(XX^\top)^\alpha (W - W')^\top\right) \tag{3}$$

for an arbitrary $\alpha \in \mathbb{Z}_+$, is given by the formula:

$$W' = U_r U_r^T W, \quad \text{where} \quad U\Sigma V^\top = W(XX^T)^{\frac{\alpha}{2}}$$

This is equivalent to:

$\alpha = 0$

- Standard low-rank Approximation, Same as PiSSA (no weighting)

$\alpha = 1$

- Weighted Low-Rank Approximation
- Weights are applied during compression

$\alpha = 2$

- CorDA Method