NeurIPS-24

Towards Next-Level Post-Training Quantization of Hyper-Scale Transformers

Junhan Kim, Chungman Lee, Eulrang Cho, Kyungphil Park, Ho-young Kim, Joonyoung Kim, Yongkweon Jeon

{jun_one.kim, chungman.lee, dragwon.jeon}@samsung.com

Samsung Research

Introduction

- Motivation
  - With the explosive growth in model complexity, the performance of LLMs has been advancing.
  - The growth in scale has resulted in a corresponding increase in computational costs.

    → Efficient processing and compression of LLMs is required.
  - Quantization is a promising solution and indispensable procedure for facilitating the efficient deployment on devices that mainly support fixed-point arithmetic.
  - Considering the model complexity and required resources (e.g., training costs and available dataset), quantization-aware training (QAT) is not practical for compressing LLMs with billions of parameters.

    → Recent studies have focused more on PTQ.

Classic PTQ Methods

- Key idea
  - Instead of choosing the nearest quantized value, classic PTQ methods attempt to assign quantized values that minimize the loss degradation incurred by the quantization:

$$\min \mathrm{E}\left[\Delta \mathbf{w}^T \mathbf{H}^{(\boldsymbol{w})} \Delta \mathbf{w}\right]$$

  - Computing and storing the Hessian matrix $\mathbf{H}^{(\boldsymbol{w})}$ is infeasible.
    → Independence between different layers or blocks (e.g., Transformer block) has been assumed, relaxing the problem into the layer-wise or block-wise reconstruction problem:

$$\min \mathrm{E}\left[\left\| Q\left(\mathbf{W}^{(\ell)}\right)\mathbf{X} - \mathbf{W}^{(\ell)}\mathbf{X}\right\|_F^2\right] \quad \text{(layerwise recon.)}$$

$$\min \mathrm{E}\left[\left\| f\left(Q\left(\mathbf{W}^{(\ell)}\right), \mathbf{X}\right) - f\left(\mathbf{W}^{(\ell)}, \mathbf{X}\right)\right\|_F^2\right] \quad \text{(blockwise recon.)}$$

  - Approaches targeting block-wise reconstruction perform better due to the consideration of inter-layer dependencies inside the Transformer block.

PTQ for LLMs

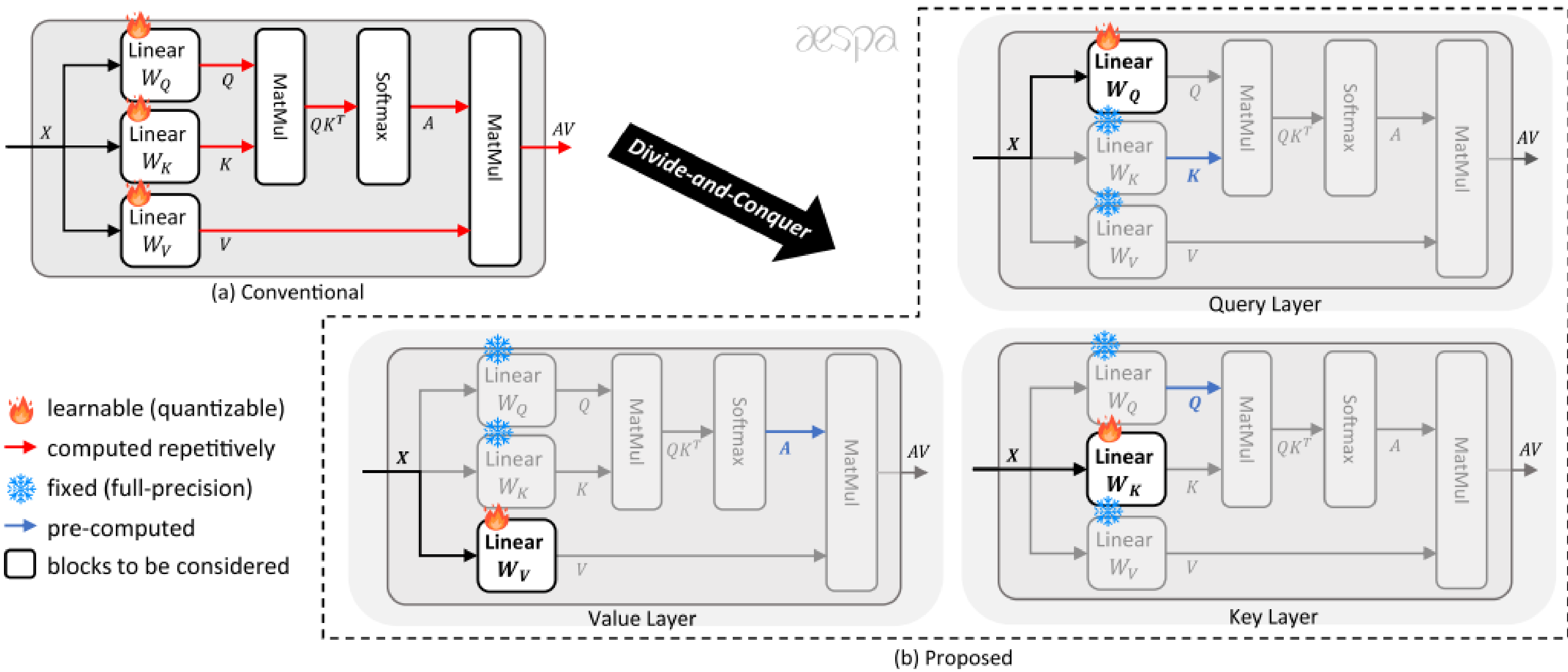
- Recent trends
  - While achieving competitive performance, classic PTQ methods require too much processing time (e.g., more than 20 GPU hours for 3B models).

    → **NOT** suitable for the real-world deployment of LLMs where models to be deployed are frequently updated.
  - For simplicity, recent methods either focus on layer-wise reconstruction (**NOT** block-wise reconstruction) or give up optimizing a weight-rounding policy:
    - GPTQ: weight-rounding optimization method targeting layer-wise reconstruction
    - AWQ, Z-Fold, OmniQuant, AffineQuant: quantization parameter (e.g., scale and zero-point) optimization methods that rely on a naïve nearest-rounding.

    → Limited low-bit quantization performance

Proposed Method

- Main goal
  - Optimize the weight-rounding policy efficiently, yet targeting block-wise reconstruction to consider inter-layer dependencies inside the attention module
- Key idea 1 – novel quantization strategy
  - Quantize each layer separately, yet targeting block-wise reconstruction



Proposed Method

- Key idea 2 – refined quantization objectives
  - Under the proposed quantization strategy, the block-wise reconstruction error can be simplified by factoring out common terms affected by full-precision layers.
  - e.g., quantization of value projection layer ($\mathbf{W}_V$)

(original) $$\min_{\Delta \boldsymbol{W}_Q, \Delta \boldsymbol{W}_K, \Delta \boldsymbol{W}_V} \mathbb{E}\left[\left\|\mathrm{SA}(\widehat{\boldsymbol{Q}}, \widehat{\boldsymbol{K}}, \widehat{\boldsymbol{V}}) - \mathrm{SA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\right\|_F^2\right]$$

(proposed) $$\min_{\Delta \boldsymbol{W}_Q, \Delta \boldsymbol{W}_K, \Delta \boldsymbol{W}_V} \mathbb{E}\left[\left\|\mathrm{SA}(\boldsymbol{Q}, \boldsymbol{K}, \widehat{\boldsymbol{V}}) - \mathrm{SA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\right\|_F^2\right]$$

$$= \mathbb{E}\left[\left\|\boldsymbol{A}\widehat{\boldsymbol{V}} - \boldsymbol{A}\boldsymbol{V}\right\|_F^2\right] = \mathbb{E}\left[\|\boldsymbol{A}\Delta\boldsymbol{V}\|_F^2\right]$$

$$= \boxed{\mathbb{E}\left[\|\Delta\boldsymbol{W}_V \boldsymbol{X} \boldsymbol{A}^T\|_F^2\right]}.$$

Proposed Method

- Key idea 3 – efficient loss computation based on pre-computations
  - Compute the value of loss functions based on certain pre-computed values
  - e.g., quantization of value projection layer ($\mathbf{W}_V$)

$$\mathbb{E}\left[\|\Delta \boldsymbol{W}_V \boldsymbol{X} \boldsymbol{A}^T\|_F^2\right] = \operatorname{tr}\left(\Delta \boldsymbol{W}_V \mathbb{E}\left[\boldsymbol{X} \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{X}^T\right] \Delta \boldsymbol{W}_V^T\right)$$

    - By computing $\mathrm{E}[\mathbf{X}\mathbf{A}^T\mathbf{A}\mathbf{X}^T]$ in advance and reusing it in the quantization process, we can avoid the overhead of computing $\mathrm{E}[\|\Delta\mathbf{W}_V\mathbf{X}\mathbf{A}^T\|_F^2]$ for every input $\mathbf{X}$.
    - Since $\mathrm{E}[\mathbf{X}\mathbf{A}^T\mathbf{A}\mathbf{X}^T]$ is pre-computed using all calibration data, we can compute the loss considering the entire calibration dataset without any memory issues.

      → Better estimate of the true gradient can be obtained, which could lead to a more consistent update and faster convergence.

Experimental Results

- Outstanding low-bit performance with reasonable processing time

Table 1: Performance (PPL ↓) of the proposed *aespa* and conventional block-wise PTQ methods.

(a) WikiText-2

Precision	Method	OPT				LLaMA			LLaMA2	
		125M	1.3B	2.7B	6.7B	7B	13B	30B	7B	13B
FP16	Baseline	27.65	14.63	12.47	10.86	5.677	5.091	4.101	5.472	4.884
INT3	BRECQ [18]	33.25	16.09	13.37	OOM	OOM	OOM	OOM	OOM	OOM
	OmniQuant [27]	39.14	17.59	14.87	12.87	6.716	5.798	4.963	6.798	5.751
	AffineQuant [20]	36.15	17.26	14.25	12.30	6.712	5.820	4.951	6.795	5.757
	aespa	**32.71**	**15.79**	**13.14**	**11.23**	**6.579**	**5.611**	**4.688**	**6.241**	**5.462**
INT2	BRECQ [18]	**60.38**	56.25	113.6	OOM	OOM	OOM	OOM	OOM	OOM
	OmniQuant [27]	NaN	399.6	1.6e3	4.9e3	18.18	NaN	10.15	35.40	20.19
	AffineQuant [20]	143.9	56.45	35.16	25.32	18.83	11.08	NaN	NaN	18.49
	aespa	71.18	**24.26**	**22.22**	**15.71**	**11.94**	**10.30**	**7.845**	**13.99**	**12.14**

Table 7: Cost of *aespa* and conventional methods (GFLOPS)

	125M	350M	1.3B	2.7B	6.7B	13B
$\mathcal{C}_{exist}$	6.7	7.5	11	15	34	41
$\mathcal{C}_{aespa}$	0.24	0.42	1.6	3.2	13	20

Table 14: Time and memory cost of *aespa* and existing methods

(a) INT2 quantization processing time

Method	OPT			
	125M	1.3B	2.7B	6.7B
BRECQ [18]	108.2 min	10.71 hr	19.15 hr	OOM
aespa	4.78 min	1.24 hr	2.83 hr	10.24 hr

Conclusion

- Propose a novel quantization method that optimizes the weight-rounding policy efficiently, yet targets block-wise reconstruction to consider inter-layer dependencies inside the attention module.
- Adopt a divide-and-conquer approach, simplifying the conventional quantization objective that requires repetitive compute-intensive attention operations.
- Propose a pre-computation-based efficient loss computation approach that facilitates 10 times faster quantization process.
- Code will be available at

https://github.com/SamsungLabs/aespa