ICML-25

BoA: Attention-aware Post-training Quantization without Backpropagation

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

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

Samsung Research

Introduction – Post-training Quantization

- **Post-training Quantization (PTQ)**
  - 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. → Compression is required.
  - Quantization is a promising solution and an essential step for deploying LLMs on resource-constrained 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.

Introduction – PTQ for LLMs

- **Backpropagation-free** Quantization
  - Key idea: iteratively quantize weights and update remaining weights based on the Hessian to compensate for the quantization error (e.g., GPTQ)
  - (+) do NOT rely on gradient-based optimization → fast!
  - (-) ignore inter-layer interactions, limiting the low-bit quantization performance
- **Transformation-based** Quantization
  - Key idea: transform a model into an equivalent quantization-robust form via smoothing (e.g., SmoothQuant, OmniQuant), rotation (e.g., QuaRot, SpinQuant), or permutation (e.g., DuQuant)
  - (+) can consider inter-layer interactions when optimizing quantization parameters (scales and zeros), smoothing factors, or rotation matrices
  - (-) rely on gradient-based optimization
  - (-) rely on the straight-through estimator (STE), which is unstable for low-bit quantization
  - (-) use the naïve nearest rounding when assigning quantized weights

Proposed Method (BoA)

- **Goal**: propose a backpropagation-free weight quantization method that can consider inter-layer dependencies
- **Primary Contributions**
  - Development of **attention-aware** Hessians that capture inter-layer interactions within the attention module
  - Integration of several relaxation techniques to mitigate the additional memory and computational overhead incurred by attention-aware Hessians
  - Evaluation of BoA through extensive experiments
    - Evaluation on various language models (OPT, LLaMA, LLaMA2, LLaMA3)
    - Synergy Verification with existing transformation-based methods

Contribution 1 – Attention-aware Hessians

- **Key Idea**: use the **attention reconstruction error** (instead of the layer-wise reconstruction error) when deriving the Hessian
- **Proposed Attention-aware Hessians**
  - Conventional GPTQ's Hessian relies solely on the layer input.

    → cannot consider the influence of other layers
  - Proposed Hessians involve the terms related to other layers (e.g., $\mathbf{K}_h^T\mathbf{K}_h$ for query) as well as the term related to the layer input.

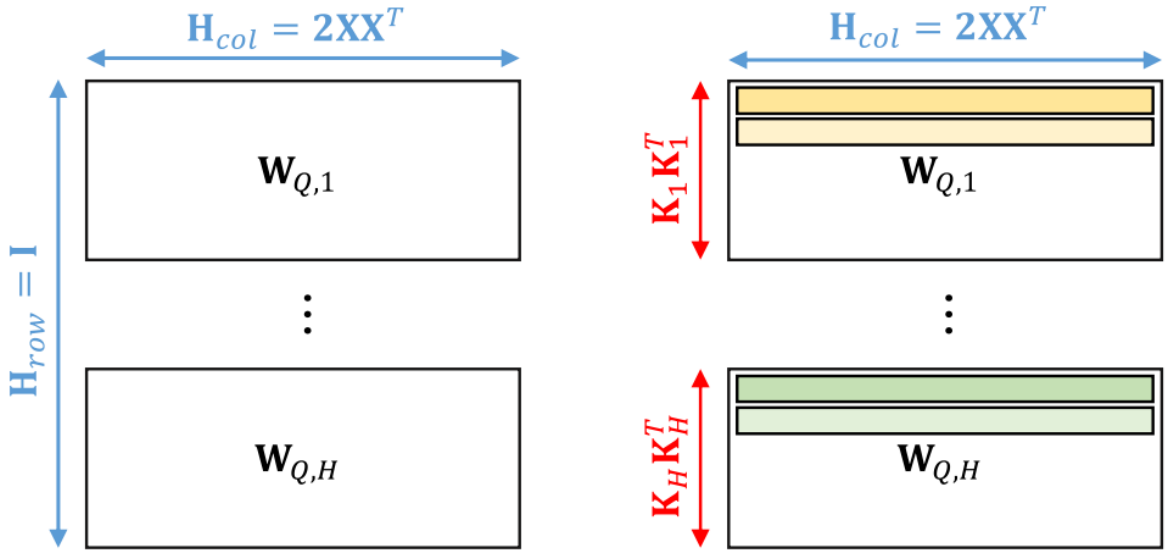
Table 1. Approximated Hessians in GPTQ and the proposed BoA

Method	Layer	$\mathbf{H} = \mathbf{H}_{\text{col}} \otimes \mathbf{H}_{\text{row}}$
GPTQ	$\mathbf{W}_{\{Q,K,V\}}$	$2\mathbf{X}\mathbf{X}^T \otimes \mathbf{I}$
BoA	$\mathbf{W}_{Q,h}$	$2\mathbf{X}\mathbf{X}^T \otimes \mathbf{K}_h^T\mathbf{K}_h$
	$\mathbf{W}_{K,h}$	$2\mathbf{X}\mathbf{X}^T \otimes \mathbf{Q}_h^T\mathbf{Q}_h$
	$\mathbf{W}_{V,h}$	$2\mathbf{X}\mathbf{A}_h^T\mathbf{A}_h\mathbf{X}^T \otimes \mathbf{W}_{\text{out},h}^T\mathbf{W}_{\text{out},h}$
	$\mathbf{W}_{\text{out},h}$	$2\mathbf{X}_{\text{out},h}\mathbf{X}_{\text{out},h}^T \otimes \mathbf{I}$

Contribution 1 – Attention-aware Hessians

- **Proposed Attention-aware Hessians**
  - Conventional GPTQ's Hessian implies the independence between different rows.
  - Proposed Hessians model the **dependency between different rows**.

    → Quantization error of a certain row can be compensated by updating other rows.



(a) Hessians in the conventional GPTQ (left) and the proposed BoA (right)

Contribution 2 – Efficient Inverse Computation

- **Additional Computational Overhead of Attention-aware Hessians**
  - To update weights based on Hessian (after quantizing certain weights), the inverse Hessian and its Cholesky decomposition are needed.

$$\boldsymbol{\delta \mathbf{w}} = \frac{Q(w_i) - w_i}{[\mathbf{U}]_{i,i}} [\mathbf{U}]_{i,:} \quad \text{where } \mathbf{U} = \text{Chol}(\mathbf{H}^{-1})^T$$

  - For the proposed attention-aware Hessians, the corresponding computational complexity is $O(d_h^3 d^3)$, which is significantly larger than that of GPTQ ($O(d^3)$).
    - In our case, $\mathbf{H} = \mathbf{H}_{col} \otimes \mathbf{H}_{row}$ where $\mathbf{H}_{col}$ is a $d \times d$ matrix and $\mathbf{H}_{row}$ is a $d_h \times d_h$ matrix. → $\mathbf{H}$ is a $d_h d \times d_h d$ matrix.

Contribution 2
– Efficient Inverse Computation

- **Efficient Computation of the Inverse Hessian and Its Cholesky Decomposition**
  - Useful property of Kronecker product

$$\mathbf{H}^{-1} = (\mathbf{H}_{col} \otimes \mathbf{H}_{row})^{-1} = \mathbf{H}_{col}^{-1} \otimes \mathbf{H}_{row}^{-1}$$

$$\mathbf{H}^{-1} = \mathbf{U}_{col}^{T}\mathbf{U}_{col} \otimes \mathbf{U}_{row}^{T}\mathbf{U}_{row} = (\mathbf{U}_{col} \otimes \mathbf{U}_{row})(\mathbf{U}_{col} \otimes \mathbf{U}_{row})^{T}$$

  - Computing the inverse / Cholesky decomposition for each component $\mathbf{H}_{col}^{-1}$ and $\mathbf{H}_{row}^{-1}$.

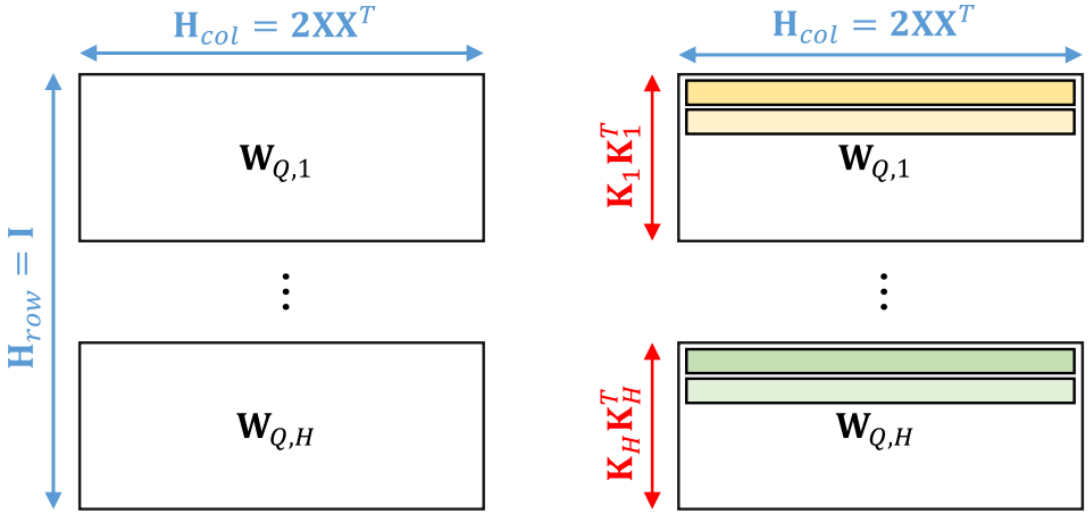
    →We do NOT need to compute for the full matrix.

  - The corresponding computational complexity is

$$O(d^3) + O\left(d_h^3\right) \approx O(d^3)$$

Contribution 3 – Simultaneous Head-wise Quantization

- **Additional Processing Time Incurred by Attention-aware Hessians**
  - Since the proposed attention-aware Hessians model the row-wise dependency, we can compensate for the quantization error of a certain row by updating other rows.
  - To do so, the rows must be quantized sequentially (NOT simultaneously).
    - e.g., The second row can be quantized after being updated to compensate for the quantization error of the first row.

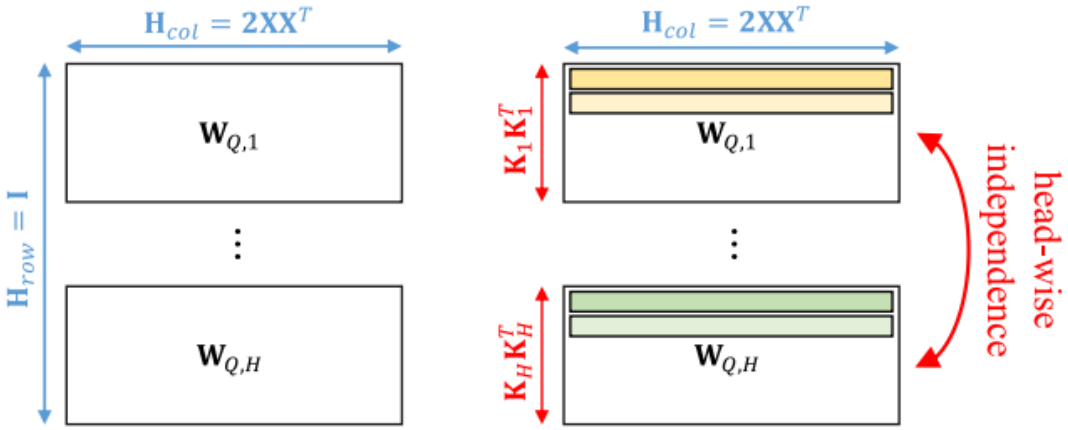


(a) Hessians in the conventional GPTQ (left) and the proposed BoA (right)

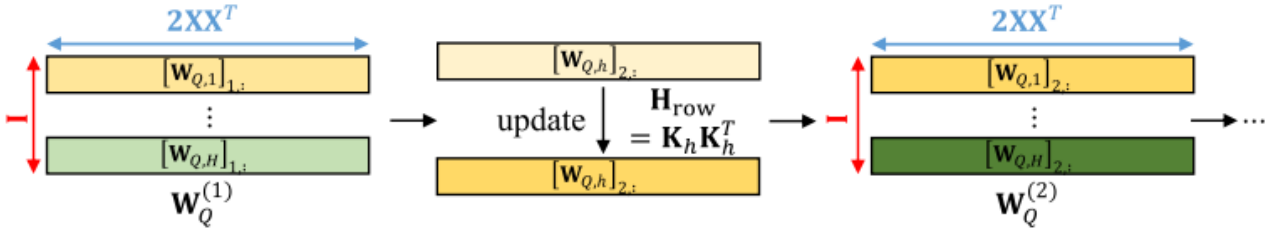
Contribution 3
– Simultaneous Head-wise Quantization

- **Simultaneous Quantization of Different Attention Heads**
  - We assume independence between different attention heads.

    → Rows belonging to different heads can be quantized simultaneously!



(a) Hessians in the conventional GPTQ (left) and the proposed BoA (right)



(b) Quantization procedure in BoA

Experimental Results

- **Effectiveness of Simultaneous Quantization of Different Attention Heads**
  - Without the proposed simultaneous quantization, a significantly long processing time is required.
    - e.g., more than one day for 7B, nearly 6 days for 30B
  - By quantizing rows belonging to different heads simultaneously, processing time can be significantly reduced.
    - e.g., more than **40 times** reduction for 30B

Table 2. Processing time (hour) of BOA with and without simultaneous quantization of different heads

Simultaneous Quantization	LLaMA Model Size		
	7B	13B	30B
X	27.75	51.66	135.4
O	**0.961**	**1.553**	**3.295**

Experimental Results

- **Comparison with GPTQ**
  - The proposed BoA significantly surpasses GPTQ on all models in both perplexity and zero-shot accuracy performance.
    - e.g., **10%p improvement** on INT3 quantized LLaMA3.2-1B
    - e.g., **20%p improvement** on INT3 quantized LLaMA3.2-3B

Table 3. Weight-only quantization performance on LLaMA3 models without transformation

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA3.2-1B	FP16	Baseline	13.15	38.14	63.26	69.51	60.78	54.38	34.60	74.37	59.51	56.82
	INT2	RTN	6.3e4	26.96	25.59	41.53	26.05	0.01	26.40	51.52	50.59	31.08
		GPTQ	538.9	25.26	26.64	37.83	26.41	0.22	27.60	51.41	48.46	30.48
		BoA	**312.2**	25.09	26.85	40.06	27.17	1.42	27.00	51.96	51.07	**31.33**
	INT3	RTN	1.9e3	25.60	26.94	54.13	29.90	0.58	27.00	52.18	49.17	33.19
		GPTQ	112.0	24.06	39.48	53.85	31.07	13.85	27.80	60.07	49.33	37.44
		BoA	**26.43**	30.63	55.26	59.97	48.33	31.95	29.60	66.65	53.99	**47.05**
LLaMA3.2-3B	FP16	Baseline	11.04	46.16	67.80	78.62	70.44	62.15	36.00	75.52	67.40	63.01
	INT2	RTN	2.0e4	26.79	26.52	37.89	25.93	0.00	30.80	50.92	49.09	30.99
		GPTQ	98.19	24.83	27.78	52.32	33.83	4.38	28.60	52.23	51.14	34.39
		BoA	**54.64**	25.77	35.48	57.52	35.63	14.42	29.00	56.96	53.43	**38.53**
	INT3	RTN	882.6	26.37	27.86	45.87	37.04	1.44	26.00	53.92	48.46	33.37
		GPTQ	46.14	28.92	37.63	44.01	39.27	18.76	28.60	61.64	54.70	39.19
		BoA	**13.64**	42.32	66.12	77.52	64.46	54.18	35.20	72.69	62.51	**59.38**

Experimental Results

- **Integration with Transformation-based Methods**
  - For weight-activation quantization, we integrate the proposed BoA with the existing transform-based method (SpinQuant) to suppress activation outliers.
  - The outstanding weight-quantization performance of BoA leads to the state-of-the-art performance for the weight-activation quantization.
    - e.g., **10%p improvement** on LLaMA2-7B
    - e.g., **12.5%p improvement** on LLaMA2-13B

Table 7. Weight-activation quantization performance on transformed LLaMA2 models

Model	Precision	Method	Wiki2 PPL (↓)	Zero-shot Accuracy (↑)								
				Arc-c	Arc-e	BQ	HS	LAMB	OBQA	PIQA	WG	Average
LLaMA2-7B	FP16	Baseline	5.473	45.90	74.66	77.92	75.94	70.86	44.00	78.89	68.90	67.13
	W2A4KV4	SpinQuant-RTN	23.23	25.51	34.01	62.20	32.09	14.48	26.00	55.17	50.91	37.55
		SpinQuant-GPTQ	24.29	22.95	36.74	59.88	32.83	13.81	26.20	56.64	51.30	37.54
		BoA†	**11.80**	26.79	49.20	63.09	48.05	37.76	30.80	63.55	57.85	**47.14**
LLaMA2-13B	FP16	Baseline	4.885	49.06	77.65	80.49	79.38	73.41	45.80	80.69	72.22	69.84
	W2A4KV4	SpinQuant-RTN	11.71	26.96	40.61	63.12	44.54	29.64	29.20	60.88	53.67	43.58
		SpinQuant-GPTQ	15.54	23.55	41.29	62.17	35.76	16.50	29.80	59.52	51.54	40.02
		BoA†	**8.974**	31.74	55.98	64.80	56.22	49.18	34.40	68.44	59.27	**52.50**

Conclusion

- We proposed a novel **backpropagation-free** weight quantization method that can **consider inter-layer dependencies**.
- We developed **attention-aware** Hessians that capture inter-layer interactions within the attention module.
- To mitigate the additional computational overhead incurred by the proposed attention-aware Hessians, we incorporated several techniques.
- From extensive experiments, we validated the efficacy of the proposed BoA.
- Code will be available at

https://github.com/SamsungLabs/BoA