ICML
International Conference
On Machine Learning

BiE: Bi-Exponent Block Floating-Point for Large Language Models Quantization

Lancheng Zou[1], Wenqian Zhao[1], Shuo Yin[1], Chen Bai[1],
Qi Sun[2], Bei Yu[1]

[1] The Chinese University of Hong Kong
[2] Zhejiang University

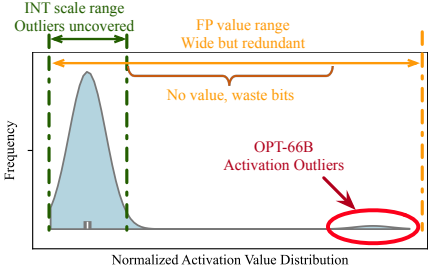
June 29, 2024

Outline

Introduction

Outlier Activations

Scaling up LLM beyond 6.7B parameters, systematic outliers with large magnitude will emerge in activations[1], leading to large quantization errors and accuracy degradation.



Value distribution of input activations in one `Linear` Layer of OPT-66B model

[1] **Tim Dettmers et al. (2022). "Llm. int8 (): 8-bit matrix multiplication for transformers at scale".** In: *arXiv preprint arXiv:2208.07339*.

Existing LLM Quantization Methods

- Integer quantization is the mainstream
  - LLM.int8()[2]: mixed precision, hard to implement efficiently on hardware
  - SmoothQuant[3]: Int8, great effort needed to mitigate the outliers
- Block floating-point is a promising choice for LLM Quantization[4]
  - combines the precision of floating-point with the efficiency of integer
  - achieves negligible loss under W6A6 without any complex transformation

Towards lower-bit LLM quantization, a novel data format is needed to better fit in the data distribution of LLM

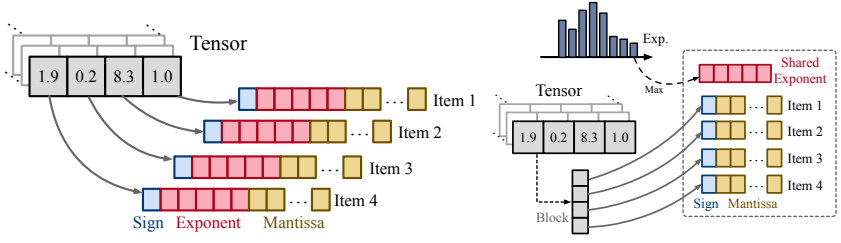
[2] **Tim Dettmers et al. (2022). "Llm. int8 (): 8-bit matrix multiplication for transformers at scale".** In: *arXiv preprint arXiv:2208.07339*.

[3] **Guangxuan Xiao et al. (2023). "Smoothquant: Accurate and efficient post-training quantization for large language models".** In: *Proc. ICML*. PMLR, pp. 38087–38099.

[4] **Cheng Zhang et al. (2023). "Revisiting Block-based Quantisation: What is Important for Sub-8-bit LLM Inference?"** In: *Proc. EMNLP*, pp. 9988–10006.

Motivation

BFP is a special case of floating-point where numbers within a block share a common exponent.
The data format of BFP includes three parts: shared exponent, private block mantissa and sign bit.



(a) FP16 (b) BFP

When using rounding to nearest scheme, the quantization error of block floating-point has zero mean and variance σ^2 which is defined as[5][6]

$$\sigma^2 = \frac{2^{-2L_m}}{12} \sum_{i=1}^{N_\gamma} p_{\gamma_i} 2^{2\gamma_i}, \tag{1}$$

where L_m is the bit length of the private block mantissa, p_{γ_i} is the probability mass function (PMF) of the shared exponents γ_i $(i = 1, 2, ...N_\gamma)$. $N_\gamma = 2^{L_E}$ is the number of available shared exponent levels, where L_E is the bit length of shared exponent. To reduce the quantization error, we can

- increase the bit length of private block mantissa L_m
- decrease the p_{γ_i} of relatively large shared exponent γ_i

[5] **Kari Kalliojarvi and Jaakko Astola (1996). "Roundoff errors in block-floating-point systems".** In: *IEEE TSP* 44.4, pp. 783–790.

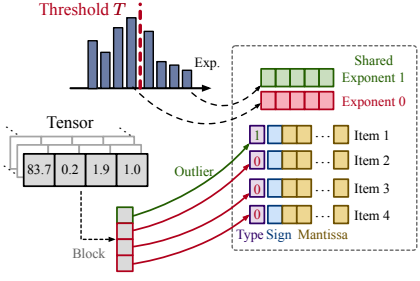
[6] **Zhourui Song, Zhenyu Liu, and Dongsheng Wang (2018). "Computation error analysis of block floating point arithmetic oriented convolution neural network accelerator design".** In: *Proc. AAAI*, vol. 32. 1.

Methods

Bi-exponent block floating-point (BiE) is the proposed numerical representation.

Different from vanilla BFP, it contains two shared exponents, e_n for normal part, and e_o for outlier part.

Threshold T is used to distinguish the normal and outlier parts, 1-bit type bit t_i indicates the component belongs to outlier part or normal part.



Data format of BiE

Given a block $\mathbf{X}$ with N elements in FP16, we can obtain its BiE representation $\mathbf{X}'$ as

$$t_i = \begin{cases} 0 & |x_i| \leq T, \\ 1 & |x_i| > T, \end{cases} \tag{2}$$

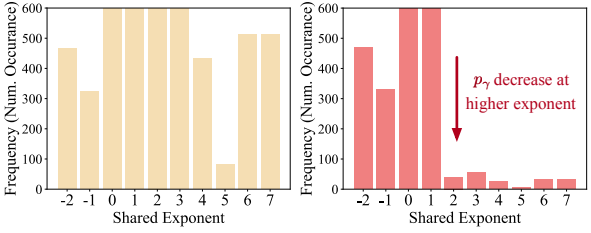
$$e_n = \max_{|x_i| \leq T} e_i, e_o = \max_{|x_i| > T} e_i, \tag{3}$$

$$m'_i = m_i >> (e_n \cdot (1 - t_i) + e_o \cdot t_i - e_i), \tag{4}$$

$$X'_i = 2^{e_n \cdot (1-t_i) + e_o \cdot t_i} \cdot (-1)^{s_i} \cdot m'_i, \tag{5}$$

Why It Works

The rationale is that bi-exponent can isolate the effect of a large shared exponent on other small-exponent values in the block to reduce the quantization error



(a) BFP (b) BiE

Threshold determination is rather critical before the BiE quantization flow.

We build an offline threshold searching strategy based on Bayesian Optimization. The search space is defined as:

$$\begin{aligned} \Omega &= \{T_1, T_2, ..., T_N\}\ ,\ N = N_a + N_w; \\ T_i &\in [P_{lo}(\mathbf{X}), P_{hi}(\mathbf{X})]. \end{aligned} \tag{6}$$

P_{lo} and P_{hi} are the lower-bound and higher-bound of the percentile. $\mathbf{X}$ is the activations or weights in calibrations.

It is essential to determine the threshold value within the central region of the magnitude distribution rather than selecting a large or small value to fully utilize the superiority of bi-exponent.

Hardware Implementation

BiE systolic array consists of three parts: FP to BiE converter (encoder), processing elements (PE) for BiE, and decoder (recover to FP).

Compared with fixed point quantization, the encoder and decoder do not involve costly floating-point multiplication.

- Encoder: two comparator trees are used for shared exponents and type bits determination
- PE: obtains a partial sum of aligned mantissa products, and the max shared exponents summation e_m
- Decoder: includes a leading-zero detector and a shifter to achieve normalization

Given two blocks $\mathbf{X}_1$ and $\mathbf{X}_2$ with N elements in BiE format, the dot product of them can be formulated as

$$e_m = e_{o_1} + e_{o_2} \tag{7}$$

$$2^{e_m} \sum_{0}^{N-1} (-1)^{s_{1,i} \oplus s_{2,i}} (m_{1,i} \cdot m_{2,i} >> (e_m - e_{1,i} - e_{2,i})) \tag{8}$$

where e_m is the max exponent combination of the two blocks which is determined by the summation of the two outlier shared exponents e_{o_1} and e_{o_2}, $e_{1,i}$ and $e_{2,i}$ represent the shared exponent used for the ith element of $\mathbf{X}_1$ and $\mathbf{X}_2$.

In BiE arithmetic, it mainly involves **INT multiplication**, **INT addition** and **shifting**, which are all efficient for hardware.

Experiments

PTQ results on OPT

Table: BiE's performance for OPT-30B and OPT-66B on other zero-shot tasks including multiple choice, commonsense reasoning, etc.. We **highlight** our 4-bit BiE results that can achieve nearly lossless quantization performance.

Model	Method	Config	LAMBADA	Arc_easy	PIQA	COPA	QNLI	SST2	**Average↑**
OPT-30B	FP16	/	71.45%	70.03%	77.64%	82.00%	51.78%	66.51%	69.90%
	SmoothQuant	W8A8	71.71%	69.61%	77.75%	84.00%	52.52%	66.63%	70.37%
	SmoothQuant	W6A6	0.54%	41.29%	57.94%	66.00%	51.44%	58.26%	45.91%
	BFP	W4A4	61.25%	68.18%	76.28%	85.00%	54.09%	66.51%	68.55%
	BiE (Ours)	W4A4	**70.11**%	**69.15**%	**77.26**%	**85.00**%	**52.15**%	**67.66**%	**70.22**%
	BFP	W3A3	7.45%	56.40%	66.43%	77.00%	50.92%	56.42%	52.44%
	BiE (Ours)	W3A3	65.11%	68.14%	75.84%	81.00%	52.19%	61.12%	67.23%
OPT-66B	FP16	/	73.96%	71.12%	78.73%	86.00%	52.19%	68.58%	71.76%
	SmoothQuant	W8A8	73.26%	71.21%	78.35%	86.00%	51.84%	63.19%	70.64%
	SmoothQuant	W6A6	0.00%	25.29%	53.32%	55.00%	50.67%	51.26%	39.26%
	BFP	W4A4	63.83%	68.86%	76.66%	86.00%	52.10%	64.68%	68.69%
	BiE (Ours)	W4A4	**72.50**%	**70.58**%	**77.26**%	**85.00**%	**51.95**%	**70.07**%	**71.23**%
	BFP	W3A3	3.22%	36.49%	55.44%	61.00%	51.09%	52.98%	43.37%
	BiE (Ours)	W3A3	11.97%	38.34%	56.04%	60.00%	49.68%	53.33%	44.89%

Table: Comparison with different methods and different quantization configurations for OPT-models on Wikitext2 (**Perplexity**↓). We **highlight** our 4-bit BiE results which are comparable with SmoothQuant W8A8.

Method	Config	6.7B	13B	30B	66B
FP16	/	10.64	9.91	9.33	9.12
SmoothQuant	W8A8	11.33	12.79	9.35	9.62
SmoothQuant	W6A6	13.16	13.75	82.54	3383.21
BFP	W4A4	11.22	11.15	9.90	14.16
BiE (Ours)	W4A4	**10.93**	**10.39**	**9.37**	**9.82**
BFP	W3A3	14.61	13.85	13.83	137.72
BiE (Ours)	W3A3	12.10	11.13	10.01	32.41

Conclusion

- BiE can be naturally adapted to the numerical distribution characteristics of the LLMs and achieve negligible loss in 4-bit activations and weights quantization.
- BiE can balance precision and hardware efficiency.
- BiE is not limited to LLM quantization, it can be used in any model with any distribution.

THANK YOU!