OneBit: Towards Extremely Low-bit Large Language Models[2]

King-Siong Si

Institute of Multimedia Knowledge Fusion and Engineering,
Xi'an JiaoTong University

sjsinx@stu.xjtu.edu.cn

March 22, 2024

1. Background
2. Methodology
3. Results
4. Discussion
5. References

Background

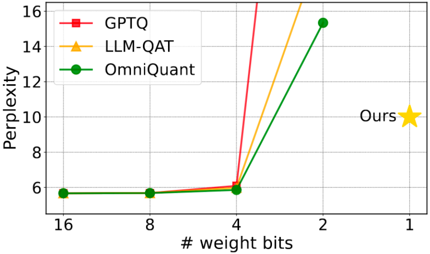


Figure 1: The perplexity (lower scores mean better performance) of existing widely-used low-bit quantization methods on LLaMA-7B.[1]

[1] From Figure 1 in [2].

Background

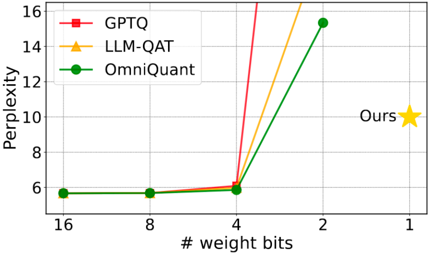


Figure 1: The perplexity (lower scores mean better performance) of existing widely-used low-bit quantization methods on LLaMA-7B.[1]

- Existing methods decline when compressing model weights to 1 bit, struggling to maintain effectiveness.

[1] From Figure 1 in [2].

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.
- Quantization in DL

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.
- Quantization in DL
    - Weight Quantization

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.
- Quantization in DL
  - Weight Quantization
  - Input Quantization

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.
- Quantization in DL
  - Weight Quantization
  - Input Quantization
  - Activation Quantization

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.
- Quantization in DL
  - Weight Quantization
  - Input Quantization
  - Activation Quantization
  - Using integer-only arithmetic or not

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.
- Quantization in DL
  - Weight Quantization
  - Input Quantization
  - Activation Quantization
  - Using integer-only arithmetic or not
- Uniform Quantization

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.
- Quantization in DL
  - Weight Quantization
  - Input Quantization
  - Activation Quantization
  - Using integer-only arithmetic or not
- Uniform Quantization

Quantization Function

$$q = \lfloor r/S \rceil + Z. \tag{1}$$

Key Concepts in Quantization

- Quantization: Quantization is the process of mapping continuous infinite values to a smaller set of discrete finite values.
- Quantization in DL
  - Weight Quantization
  - Input Quantization
  - Activation Quantization
  - Using integer-only arithmetic or not
- Uniform Quantization

Quantization Function

$$q = \lfloor r/S \rceil + Z. \tag{1}$$

- PTQ(Post Training Quantization)
- QAT(Quantization Aware Training)

Idea in This Paper

- Bit-Width: 1

Idea in This Paper

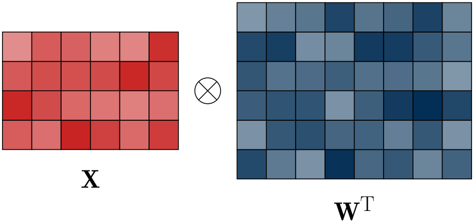
- Bit-Width: 1
- Weight Quantization

Idea in This Paper

- Bit-Width: 1
- Weight Quantization
- QAT

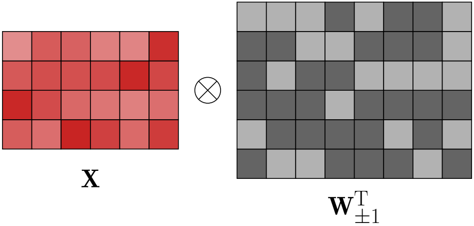
Idea in This Paper

- Bit-Width: 1
- Weight Quantization
- QAT



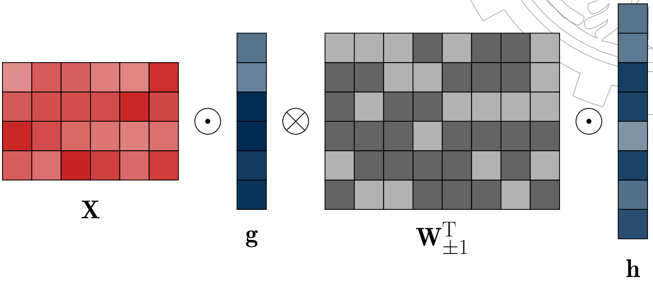
Idea in This Paper

- Bit-Width: 1
- Weight Quantization
- QAT



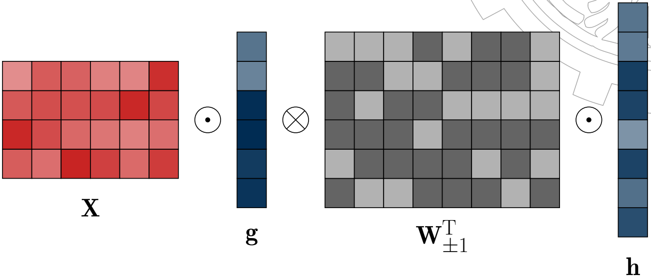
Idea in This Paper

- Bit-Width: 1
- Weight Quantization
- QAT



Idea in This Paper

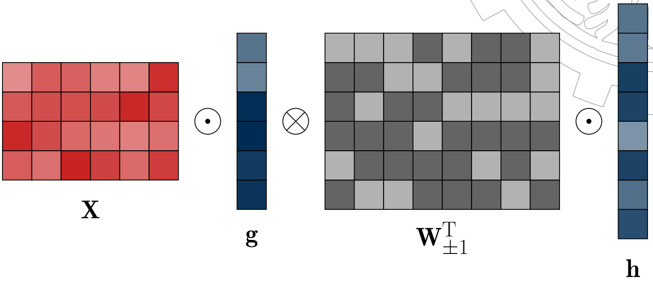
- Bit-Width: 1
- Weight Quantization
- QAT



- Why value vectors?

Idea in This Paper

- Bit-Width: 1
- Weight Quantization
- QAT



- Why value vectors?
- *XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks*[1]

God Knows?

- CNN: $\langle \mathcal{I}, \mathcal{W}, * \rangle$

God Knows?

- CNN: $\langle \mathcal{I}, \mathcal{W}, * \rangle$
- Find a new binary filter $\mathbf{B} \in \{+1, -1\}^{c \times w \times h}$ and a scale $\alpha \in \mathbb{R}^+$ such that $\mathbf{W} \approx \alpha \mathbf{B}$, then

$$\mathbf{I} * \mathbf{W} \approx \mathbf{I} * (\alpha \mathbf{B}) \tag{2}$$

$$= (\mathbf{I} \oplus \mathbf{B})\alpha. \tag{3}$$

God Knows?

- CNN: $\langle \mathcal{I}, \mathcal{W}, * \rangle$
- Find a new binary filter $\mathbf{B} \in \{+1, -1\}^{c \times w \times h}$ and a scale $\alpha \in \mathbb{R}^{+}$ such that $\mathbf{W} \approx \alpha \mathbf{B}$, then

$$\mathbf{I} * \mathbf{W} \approx \mathbf{I} * (\alpha \mathbf{B}) \tag{2}$$
$$= (\mathbf{I} \oplus \mathbf{B})\alpha. \tag{3}$$

- In this paper, Linear layers are designed as

God Knows?

- CNN: $\langle \mathcal{I}, \mathcal{W}, * \rangle$
- Find a new binary filter $\mathbf{B} \in \{+1, -1\}^{c \times w \times h}$ and a scale $\alpha \in \mathbb{R}^+$ such that $\mathbf{W} \approx \alpha \mathbf{B}$, then

$$\mathbf{I} * \mathbf{W} \approx \mathbf{I} * (\alpha \mathbf{B}) \tag{2}$$
$$= (\mathbf{I} \oplus \mathbf{B})\alpha. \tag{3}$$

- In this paper, Linear layers are designed as

Linear Layer

$$\mathbf{W}_{\pm 1} = \mathrm{Sign}(\mathbf{W}), \tag{4}$$
$$\mathbf{Y} = \mathrm{LayerNorm}\left(\left[(\mathbf{X} \odot \mathbf{g})\mathbf{W}_{\pm 1}^{\mathrm{T}}\right] \odot \mathbf{h}\right). \tag{5}$$

That Is It?!

- Q: How to get $\mathbf{W}_{\pm 1}$?

That Is It?!

- Q: How to get $\mathbf{W}_{\pm 1}$?
- A: Negative/Positive $\mapsto -1/1$

That Is It?!

- Q: How to get $\mathbf{W}_{\pm 1}$?
- A: Negative/Positive $\mapsto -1/1$
- Q: How to get $\mathbf{g}$ and $\mathbf{h}$?

That Is It?!

- Q: How to get $\mathbf{W}_{\pm 1}$?
- A: Negative/Positive $\mapsto -1/1$
- Q: How to get $\mathbf{g}$ and $\mathbf{h}$?
- A: Low-Rank Approximation

That Is It?!

- Q: How to get $\mathbf{W}_{\pm 1}$?
- A: Negative/Positive $\mapsto -1/1$
- Q: How to get $\mathbf{g}$ and $\mathbf{h}$?
- A: Low-Rank Approximation
- SVID(Sign-Value-Independent Decomposition)

That Is It?!

- Q: How to get $\mathbf{W}_{\pm 1}$?
- A: Negative/Positive $\mapsto -1/1$
- Q: How to get $\mathbf{g}$ and $\mathbf{h}$?
- A: Low-Rank Approximation
- SVID(Sign-Value-Independent Decomposition)

SVID

$$\mathbf{W} = \mathbf{W}_{\pm 1} \odot \mathbf{W}_{\text{value}} \tag{6}$$

$$\approx \mathbf{W}_{\pm 1} \odot (\mathbf{a}\mathbf{b}^{\mathrm{T}}) \tag{7}$$

That Is It?!

- Q: How to get $\mathbf{W}_{\pm 1}$?
- A: Negative/Positive $\mapsto -1/1$
- Q: How to get $\mathbf{g}$ and $\mathbf{h}$?
- A: Low-Rank Approximation
- SVID(Sign-Value-Independent Decomposition)

SVID

$$\mathbf{W} = \mathbf{W}_{\pm 1} \odot \mathbf{W}_{\text{value}} \tag{6}$$

$$\approx \mathbf{W}_{\pm 1} \odot (\mathbf{a}\mathbf{b}^{\mathrm{T}}) \tag{7}$$

- Balance rank and precision.

That Is It?!

- $\mathbf{X}\left(\mathbf{W}_{\pm 1}^{\mathrm{T}} \odot (\mathbf{b}\mathbf{a}^{\mathrm{T}})\right) = \left[(\mathbf{X} \odot \mathbf{b}^{\mathrm{T}})\mathbf{W}_{\pm 1}^{\mathrm{T}}\right] \odot \mathbf{a}^{\mathrm{T}}$ with the same complexity. See Appendix 1 for details.

That Is It?!

- $\mathbf{X}\left(\mathbf{W}_{\pm 1}^{\mathrm{T}} \odot (\mathbf{b}\mathbf{a}^{\mathrm{T}})\right) = \left[(\mathbf{X} \odot \mathbf{b}^{\mathrm{T}})\mathbf{W}_{\pm 1}^{\mathrm{T}}\right] \odot \mathbf{a}^{\mathrm{T}}$ with the same complexity. See Appendix 1 for details.
- Why not use low-rank approximation directly?

That Is It?!

- $\mathbf{X}\left(\mathbf{W}_{\pm 1}^{\mathrm{T}} \odot (\mathbf{b}\mathbf{a}^{\mathrm{T}})\right) = \left[(\mathbf{X} \odot \mathbf{b}^{\mathrm{T}})\mathbf{W}_{\pm 1}^{\mathrm{T}}\right] \odot \mathbf{a}^{\mathrm{T}}$ with the same complexity. See Appendix 1 for details.
- Why not use low-rank approximation directly?
- Less error[2]

That Is It?!

- $\mathbf{X}\left(\mathbf{W}_{\pm 1}^{\mathrm{T}} \odot (\mathbf{b}\mathbf{a}^{\mathrm{T}})\right) = \left[(\mathbf{X} \odot \mathbf{b}^{\mathrm{T}})\mathbf{W}_{\pm 1}^{\mathrm{T}}\right] \odot \mathbf{a}^{\mathrm{T}}$ with the same complexity. See Appendix 1 for details.
- Why not use low-rank approximation directly?
- Less error[2]
- QAT: based on KD

Results

Models	Methods	Perplexity(↓)		Zero-shot Accuracy(↑)						
		Wiki2	C4	Winogrande	Hellaswag	PIQA	BoolQ	ARC-e	ARC-c	Avg.
OPT-1.3B	FP16	14.63	14.72	59.67	53.73	72.42	57.68	50.80	29.69	54.00
	GPTQ	9.5e3	3.8e3	49.33	25.57	52.07	39.60	26.68	23.63	36.15
	LLM-QAT	4.9e3	2.1e3	49.72	25.72	50.05	37.83	25.76	**25.09**	35.70
	OmniQuant	42.43	55.64	**51.85**	33.39	60.94	56.45	38.76	23.38	44.13
	OneBit	**25.42**	**22.95**	51.14	**34.26**	**62.57**	**59.45**	**41.25**	24.06	**45.46**
OPT-2.7B	FP16	12.47	13.17	60.93	60.59	74.81	60.28	54.34	31.31	57.04
	GPTQ	8.7e3	3.9e3	49.88	26.47	49.84	39.88	25.76	**26.02**	36.31
	LLM-QAT	3.7e3	1.4e3	52.09	25.47	49.29	37.83	24.92	25.60	35.87
	OmniQuant	30.25	41.31	51.62	**38.21**	62.19	54.25	40.82	24.74	45.31
	OneBit	**21.86**	**20.76**	**51.67**	38.18	**63.87**	**54.28**	**43.39**	24.40	**45.97**
LLaMA-7B	FP16	5.68	7.08	66.85	72.99	77.37	73.21	52.53	41.38	64.06
	GPTQ	1.9e3	7.8e2	49.41	25.63	49.95	43.79	25.84	27.47	37.02
	LLM-QAT	7.1e2	3.0e2	51.78	24.76	50.87	37.83	26.26	25.51	36.17
	OmniQuant	15.34	26.21	52.96	43.68	62.79	58.69	41.54	29.35	48.17
	OneBit	**10.38**	**11.56**	**60.30**	**50.73**	**67.46**	**62.51**	**41.71**	**29.61**	**52.05**
LLaMA-13B	FP16	5.09	6.61	70.17	76.24	79.05	68.47	59.85	44.54	66.39
	GPTQ	3.2e3	9.9e2	50.67	25.27	50.00	42.39	26.14	27.39	36.98
	LLM-QAT	1.8e3	1.2e3	51.62	25.40	50.33	37.83	27.02	26.87	36.51
	OmniQuant	13.43	19.33	53.83	54.16	68.99	62.20	**45.50**	30.38	52.51
	OneBit	**9.18**	**10.25**	**62.90**	**56.78**	**70.67**	**64.16**	44.53	**32.00**	**55.17**

Figure 2: Main results of evaluation experiment.[1]

[1] From Table 1 in [2].

Results

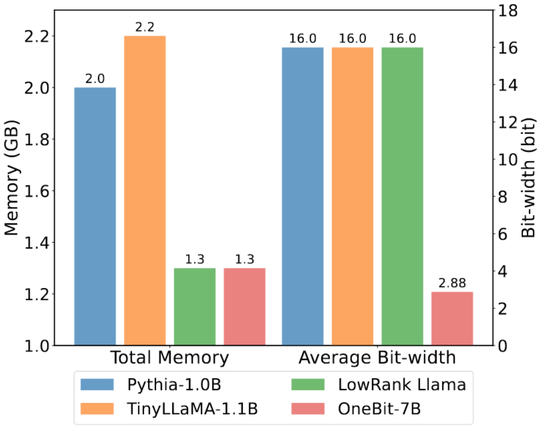


Figure 3: Memory footprint and bit-width. [1]

[1] From Figure 3(c) in [2].

Discussion

- The position of LayerNorm

Discussion

- The position of LayerNorm
- The details of KD

Discussion

- The position of LayerNorm
- The details of KD
- SVD vs. NMF

Discussion

- The position of LayerNorm
- The details of KD
- SVD vs. NMF
- Backward? Gradient?

[1] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi.
Xnor-net: Imagenet classification using binary convolutional neural networks.
In *European conference on computer vision*, pages 525–542. Springer, 2016.

[2] Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che.
Onebit: Towards extremely low-bit large language models.
arXiv preprint arXiv:2402.11295, 2024.

忠果敦精
恕毅篤勤
任力勵求
事行志學

THANKS!

Complexity of Two Types

Suppose that $\mathbf{X}$ is a n by m matrix, while $\mathbf{W}$ is a k by m matrix. Then, $\mathbf{a}$ and $\mathbf{b}$ are vectors of length k and m, respectively.

In the former method, when calculating $\left(\mathbf{X}\mathbf{W}_{\pm 1}^{\mathrm{T}} \odot (\mathbf{b}\mathbf{a}^{\mathrm{T}})\right)$, we need $(k \times m + k \times m + n \times m \times k)$ multiplications.

In the latter one, we need $(n \times m + n \times m \times k + n \times k)$ multiplications.

The difference between them is $2km - nm - nk = \frac{1}{2}\left((2k-n)(2m-n) - n^2\right)$. That is to say, a sufficient condition for the latter method to be better than the former one is $k > n \wedge m > n$.