Matryoshka Quantization

Google DeepMind

2025. 04. 21

Efficient ML

Seung-taek Woo, Chiwoong Lee, Byeongho Yu

Preliminary

Quantization?

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

Preliminary

Quantization?

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

*** 97.7** →

INT8

127	-49	-117
-39	20	108
10	-10	68

Preliminary

Quantization?

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

*** 97.7** →

INT8

127	-49	-117
-39	20	108
10	-10	68

/ 97.7 →

FP16-Dequantization

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

Preliminary

Quantization?

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

*** 97.7**

INT8

127	-49	-117
-39	20	108
10	-10	68

/ 97.7

FP16-Dequantization

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

*** 5.4**

INT4

7	-3	-6
-2	1	6
1	-1	4

/ 5.4

FP16-Dequantization

1.3	-0.6	-1.1
-0.4	0.2	1.1
0.2	-0.2	0.7

: Quantization Error

POSTECH

Preliminary

Quantization?

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

*** 97.7** →

INT8

127	-49	-117
-39	20	108
10	-10	68

/ 97.7 →

FP16-Dequantization

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

*** 5.4** →

INT4

7	-3	-6
-2	1	6
1	-1	4

/ 5.4 →

FP16-Dequantization

1.3	-0.6	-1.1
-0.4	0.2	1.1
0.2	-0.2	0.7

*** 0.8** →

INT2

1	0	-1
0	0	1
0	0	1

/ 0.8 →

FP16-Dequantization

1.3	0	-1.3
0	0	1.3
0	0	1.3

(yellow): Quantization Error

Introduction

LLM Service

ChatGPT

Claude

Gemini

Copilot

Foundation LLM

Llama

deepseek

Qwen

Gemma

Introduction

LLM Service

ChatGPT
Claude
Gemini
Copilot

100B ~1T

Foundation LLM

Llama
deepseek
Qwen
Gemma

1B~ 700B

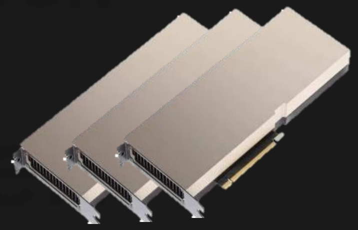
Introduction

LLM Service

Foundation LLM

100B(FP16) ≈ 200GB

Three A100-80Gs are needed for <u>inference only.</u>



Introduction

Problem: Need many GPUs.

Quantization is the solution!

Llama-2-70b-hf —FP16→ 140GB ⟶ 2 x A100-80G(160GB)

POSTECH

Introduction

Problem: Need many GPUs.

<u>Quantization</u> is the solution!

Llama-2-70b-hf —FP16→ 140GB → 2 x A100-80G(160GB)

140GB ↓ INT8

70GB → 1 x A100-80G(80GB)

Introduction

Problem: Need many GPUs.

Quantization is the solution!

Llama-2-70b-hf —FP16→ 140GB → 2 x A100-80G(160GB)

↓ INT8

70GB → 1 x A100-80G(80GB)

↓ INT4

35GB → 1 x A6000(48GB)

Introduction

Problem: Need many GPUs.

Quantization is the solution!

Llama-2-70b-hf —FP16→ 140GB → 2 x A100-80G(160GB)

↓ INT8

70GB → 1 x A100-80G(80GB)

↓ INT4

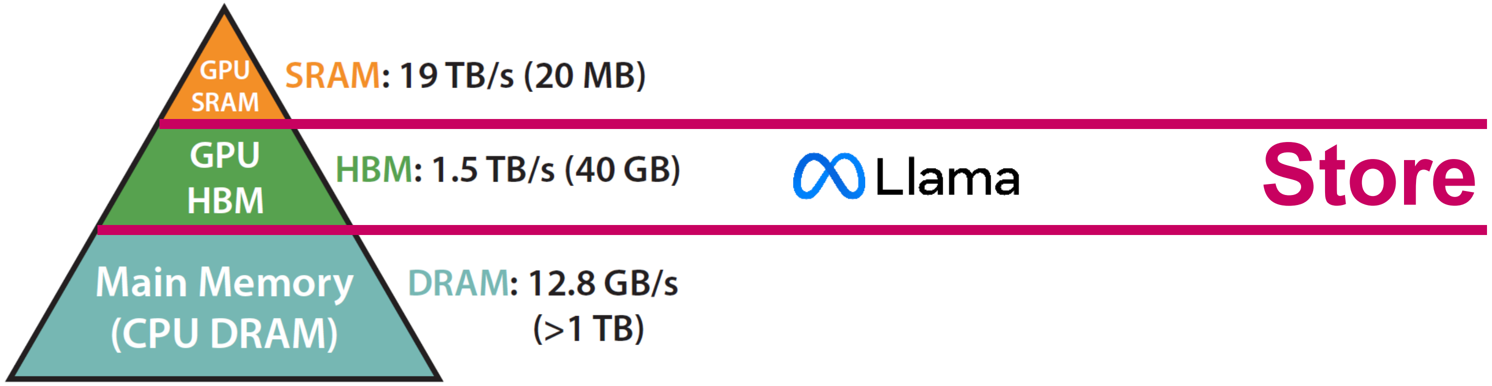
35GB → 1 x A6000(48GB)

↓ INT2

17.5GB → 1 x RTX3090(24GB)

Introduction

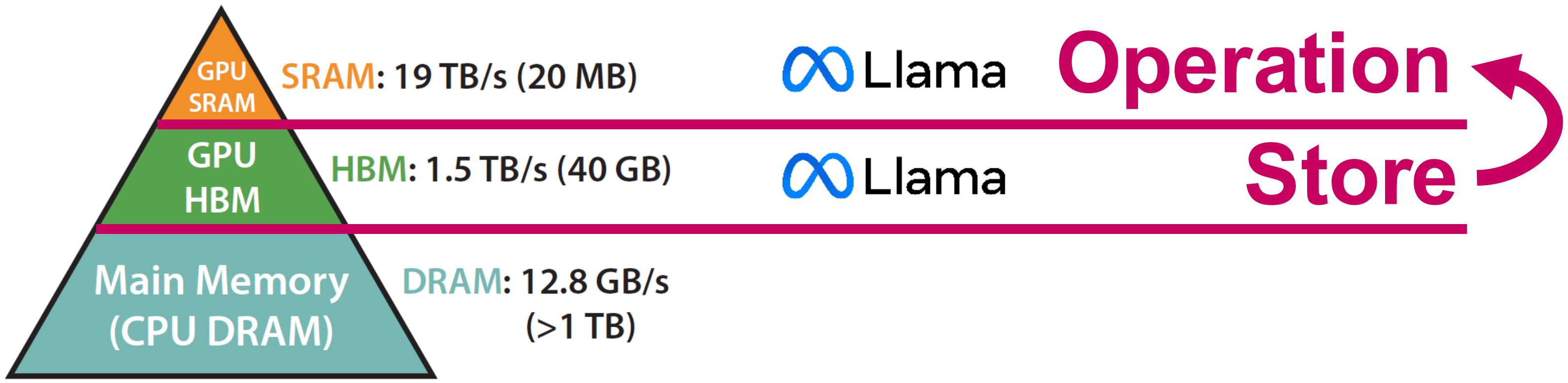
GPU Memory Hierarchy



Memory Hierarchy with
Bandwidth & Memory Size

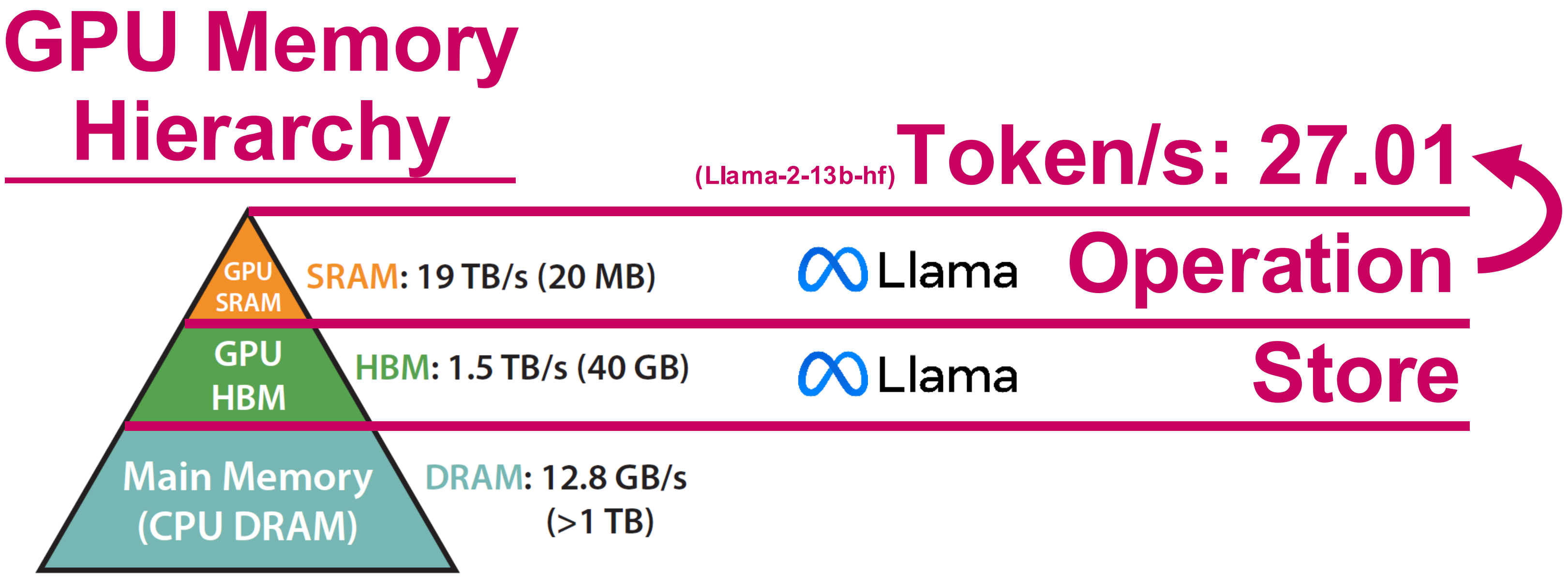
Introduction

GPU Memory Hierarchy



Memory Hierarchy with Bandwidth & Memory Size

Introduction



Memory Hierarchy with Bandwidth & Memory Size

Introduction

GPU Memory

13B(FP16) ≈ 26GB

Decoding latency is dominated by Memory Bound.

Bandwidth & Memory Size

Introduction

Problem: Low Speed.

Quantization is the solution!

Llama-2-13b-hf $\xrightarrow{\text{FP16}}$ 27.01 Token/s

POSTECH

Introduction

Problem: Low Speed.

Quantization is the solution!

Llama-2-13b-hf —FP16→ 27.01 Token/s

↓ INT4

43.94 Token/s

Introduction

Problem: Low Speed.

Quantization is the solution!

Llama-2-13b-hf —FP16→ 27.01 Token/s

↓ INT4

43.94 Token/s

↓ CUDA Graph Capture & Replay

322.17 Token/s

Introduction

However, current quantization methods[1, 2, 3]…

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

Need to optimize independently to target precision.

[1] Lee, Changhun, et al. "Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. No. 12. 2024.
[2] Lin, Ji, et al. "Awq: Activation-aware weight quantization for on-device llm compression and acceleration." *Proceedings of Machine Learning and Systems* 6 (2024): 87-100.
[3] Frantar, Elias, et al. "Gptq: Accurate post-training quantization for generative pre-trained transformers." *arXiv preprint arXiv:2210.17323* (2022).

Introduction

However, current quantization methods[1, 2, 3]…

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

INT8

127	-49	-117
-39	20	108
10	-10	68

[1] Lee, Changhun, et al. "Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. No. 12. 2024.
[2] Lin, Ji, et al. "Awq: Activation-aware weight quantization for on-device llm compression and acceleration." *Proceedings of Machine Learning and Systems* 6 (2024): 87-100.
[3] Frantar, Elias, et al. "Gptq: Accurate post-training quantization for generative pre-trained transformers." *arXiv preprint arXiv:2210.17323* (2022).

Introduction

However, current quantization methods[1, 2, 3]…

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

INT8

127	-49	-117
-39	20	108
10	-10	68

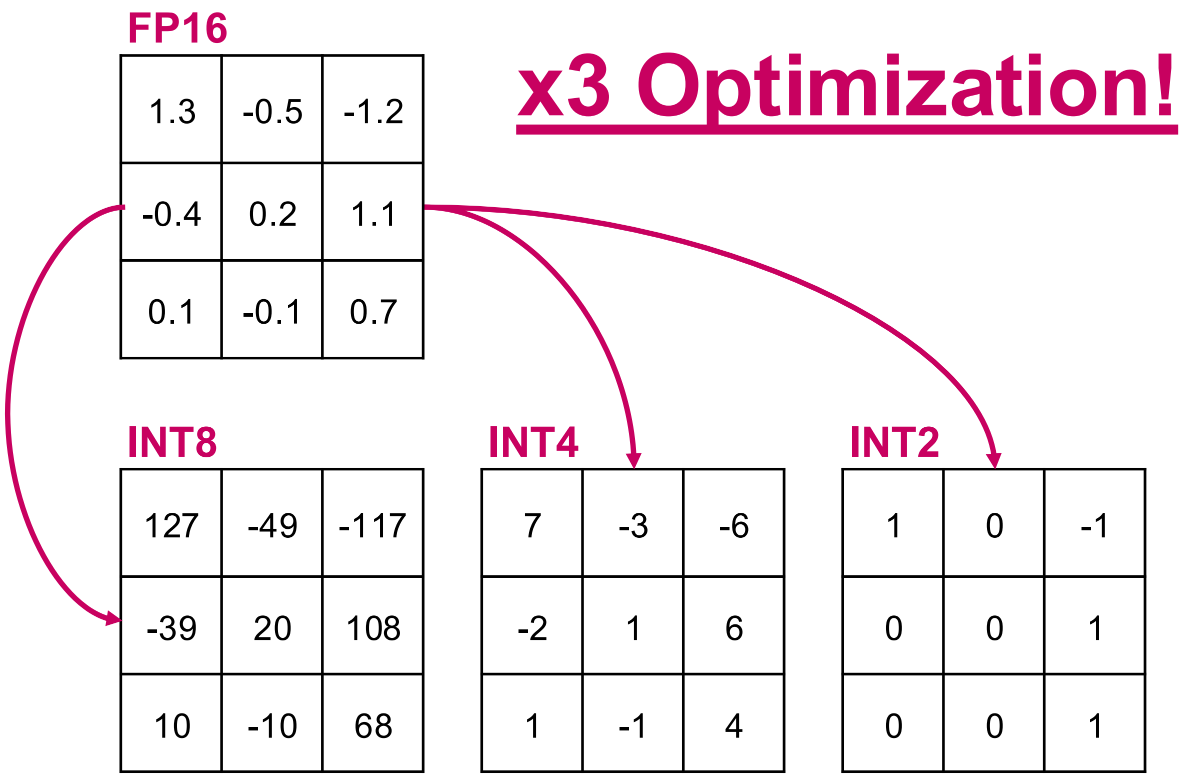
INT4

7	-3	-6
-2	1	6
1	-1	4

[1] Lee, Changhun, et al. "Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. No. 12. 2024.
[2] Lin, Ji, et al. "Awq: Activation-aware weight quantization for on-device llm compression and acceleration." *Proceedings of Machine Learning and Systems* 6 (2024): 87-100.
[3] Frantar, Elias, et al. "Gptq: Accurate post-training quantization for generative pre-trained transformers." *arXiv preprint arXiv:2210.17323* (2022).

Introduction

However, current quantization methods[1, 2, 3]…



[1] Lee, Changhun, et al. "Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. No. 12. 2024.
[2] Lin, Ji, et al. "Awq: Activation-aware weight quantization for on-device llm compression and acceleration." *Proceedings of Machine Learning and Systems* 6 (2024): 87-100.
[3] Frantar, Elias, et al. "Gptq: Accurate post-training quantization for generative pre-trained transformers." *arXiv preprint arXiv:2210.17323* (2022).

Introduction

Question: Can I extract multiple low-precision from a single optimization?

FP16

1.3	-0.5	-1.2
-0.4	0.2	1.1
0.1	-0.1	0.7

x1 Optimization!

INT8

127	-49	-117
-39	20	108
10	-10	68

INT4

7	-3	-6
-2	1	6
1	-1	4

INT2

1	0	-1
0	0	1
0	0	1

Introduction

Matryoshka

Introduction

Matryoshka

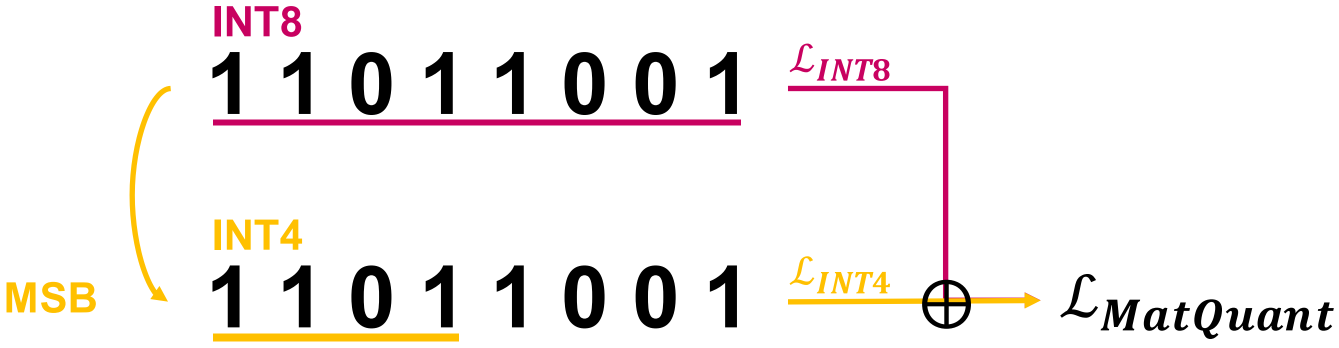
INT8

1 1 0 1 1 0 0 1

$\mathcal{L}_{INT8}$ → $\mathcal{L}_{MatQuant}$

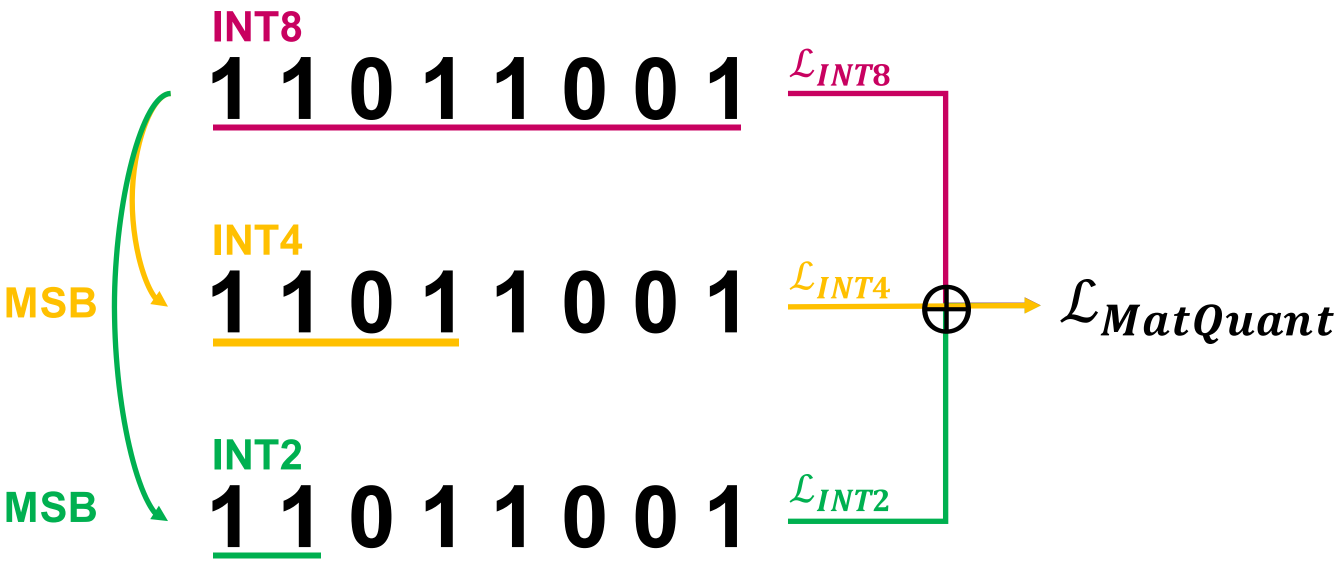
Introduction

Matryoshka



Introduction

Matryoshka



Introduction

Jointly optimize the loss for each precision level.

Preliminaries

Quantization Aware Training (QAT)

- Quantization Aware Training (QAT) learns a c-bit quantized model by minimizing end-to-end cross-entropy loss via gradient descent.
- It uses quantized weights during the forward pass and applies a **Straight-Through Estimator (STE)** to backpropagate gradients through the non-differentiable quantization operation.

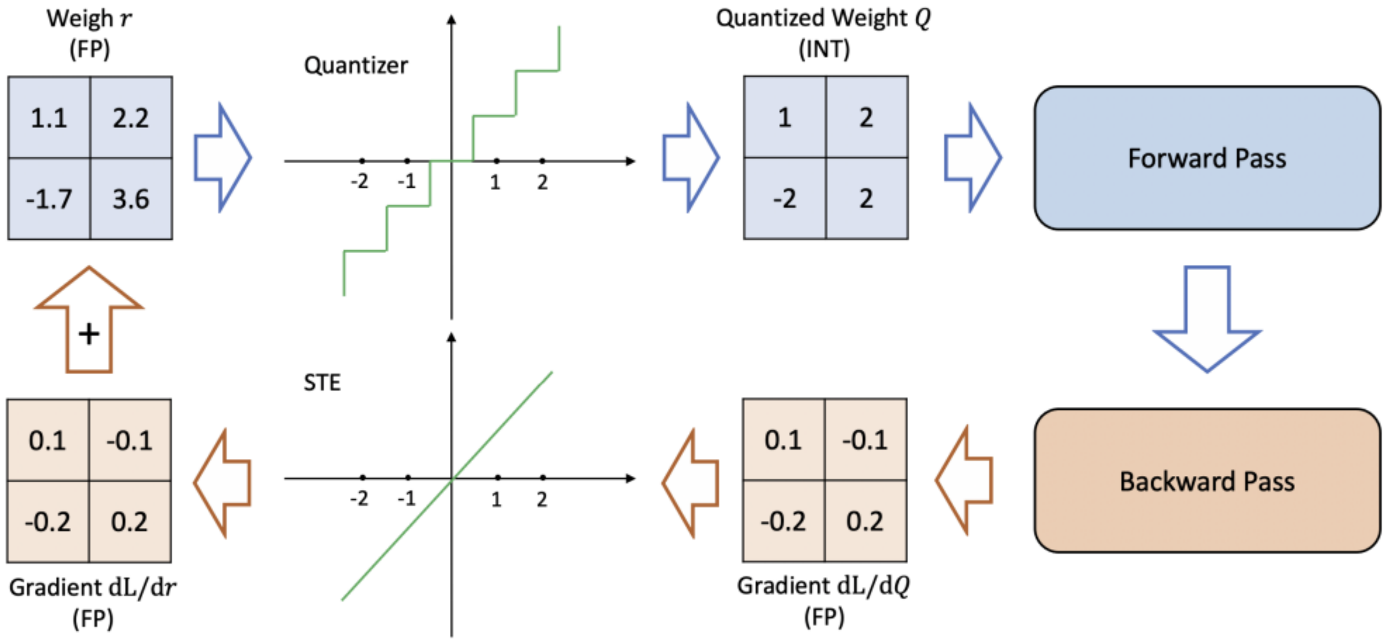


Figure 5: *Illustration of Quantization-Aware Training procedure, including the use of Straight Through Estimator (STE).*

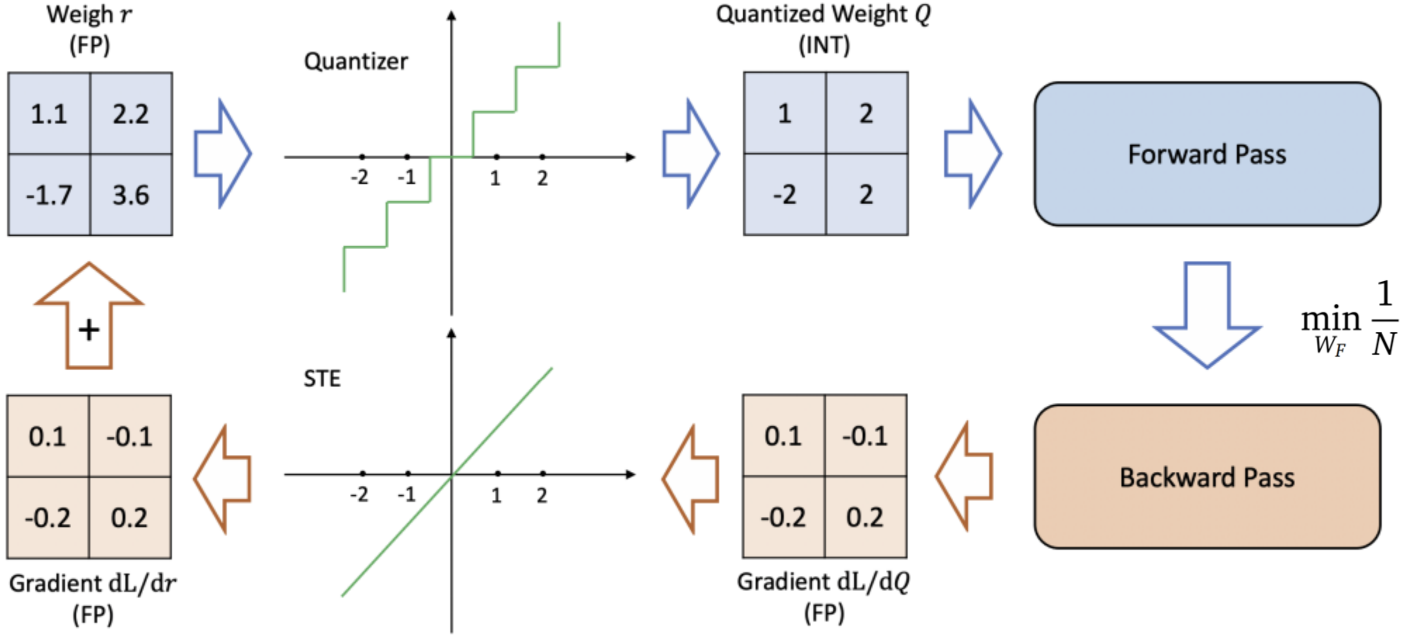
Preliminaries

Quantization Aware Training (QAT)

- Quantization Aware Training (QAT) … nizing end-to-end cross-entropy loss via gradient descent.
- It uses quantized weights during th … hrough Estimator (STE) to backpropagate gradients through the non-differentiable quantization …

$$Q_{\mathrm{MM}}(w, c) = \mathrm{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\max(w) - \min(w)}{2^c - 1}, \quad z = -\frac{\min(w)}{\alpha}$$



$$\min_{W_F} \frac{1}{N} \sum_{i \in [N]} \mathcal{L}_{\mathrm{CE}}\left(F(x_i; Q_{\mathrm{MM}}(W_F, c)), y_i\right)$$

predict actual

Figure 5: *Illustration of Quantization-Aware Training procedure, including the use of Straight Through Estimator (STE).*

Preliminaries

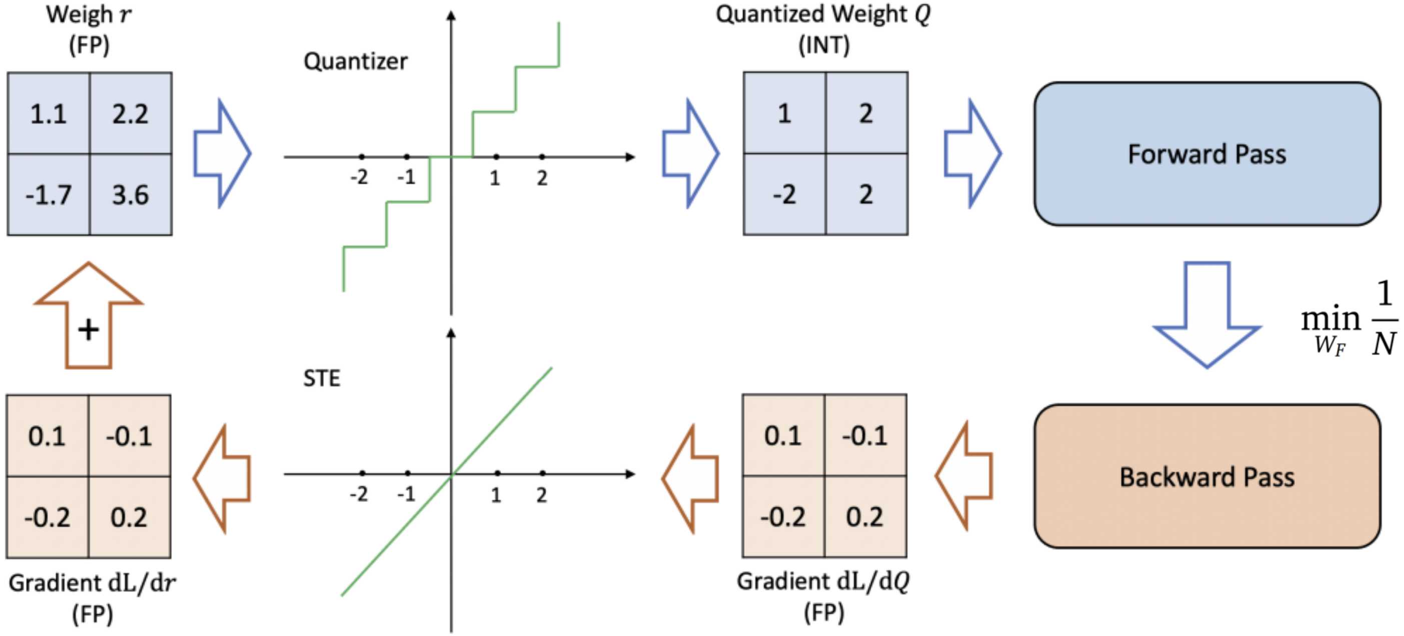
Quantization Aware Training (QAT)

- Quantization Aware Training (QAT … nizing end-to-end cross-entropy loss via gradient descent.
- It uses quantized weights during th … hrough Estimator (STE) to backpropagate gradients through the non-differentiable quantization …

Problem ?
= Not Differentiable

$$Q_{\mathrm{MM}}(w, c) = \mathrm{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\max(w) - \min(w)}{2^c - 1}, \quad z = -\frac{\min(w)}{\alpha}$$



$$\min_{W_F} \frac{1}{N} \sum_{i \in [N]} \mathcal{L}_{\mathrm{CE}}\left(F(x_i; Q_{\mathrm{MM}}(W_F, c)), y_i\right)$$

predict — actual

Figure 5: *Illustration of Quantization-Aware Training procedure, including the use of Straight Through Estimator (STE).*

Preliminaries

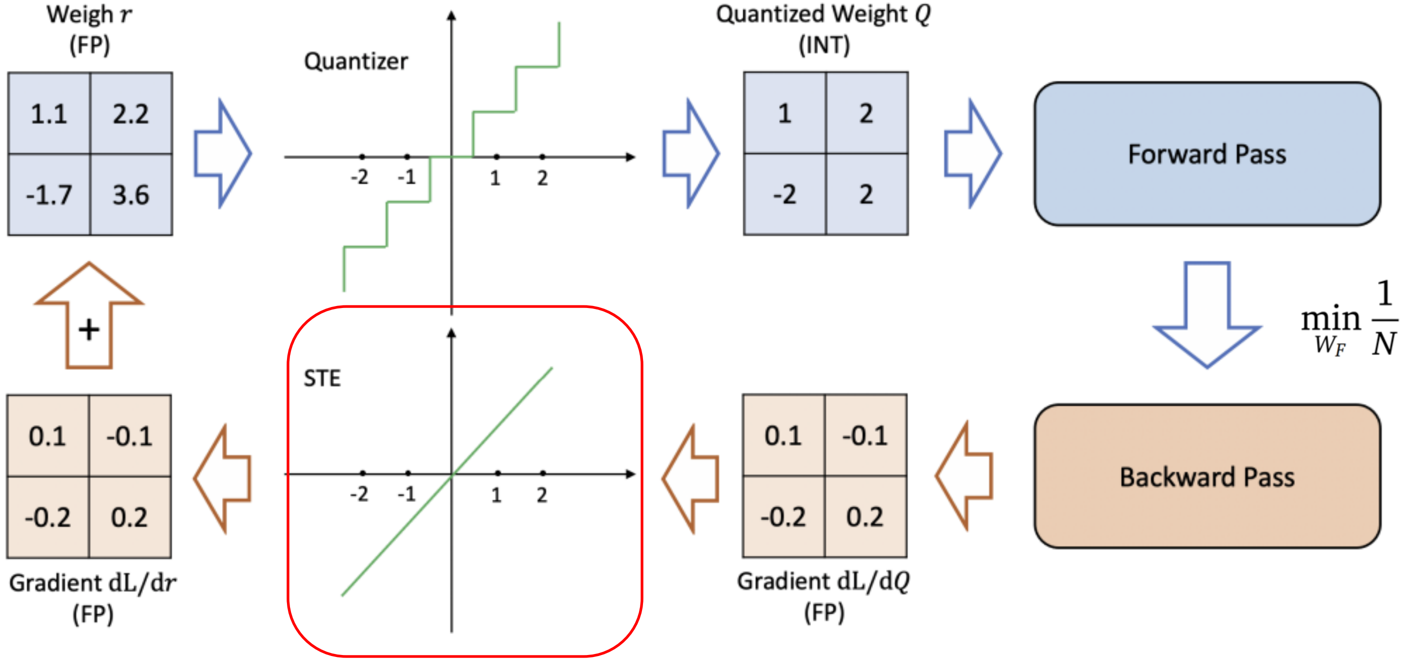
Quantization Aware Training (QAT)

- Quantization Aware Training (QAT ... nizing end-to-end cross-entropy loss via gradient descent.
- It uses quantized weights during th ... **hrough Estimator (STE)** to backpropagate gradients through the non-differentiable quantization ...

Problem ?
= Not Differentiable

$$Q_{MM}(w, c) = \text{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\max(w) - \min(w)}{2^c - 1}, \quad z = -\frac{\min(w)}{\alpha}$$



$$\min_{W_F} \frac{1}{N} \sum_{i \in [N]} \mathcal{L}_{CE}\left(F(x_i; Q_{MM}(W_F, c)), y_i\right)$$

predict actual

Figure 5: *Illustration of Quanti... procedure, including the use of Straight Through Estimator (STE).*

$$\frac{d\mathcal{L}}{dr} = \frac{d\mathcal{L}}{dQ} \cdot \frac{dQ}{dr} \approx \frac{d\mathcal{L}}{dQ}$$

POSTECH

Preliminaries

OmniQuant (ICLR2024, Spotlight)

- Unlike QAT, OmniQuant does not update the model parameters.
- Instead, it learns additional scaling and shifting parameters through gradient descent over layer-wise L2 error reconstruction.

$$Q_{\mathrm{MM}}(w, c) = \mathrm{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\max(w) - \min(w)}{2^c - 1}, \quad z = -\frac{\min(w)}{\alpha}$$

QAT

$$Q_{\mathrm{Omni}}(w, c) = \mathrm{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\gamma \cdot \max(w) - \beta \cdot \min(w)}{2^c - 1}, \quad z = -\frac{\beta \cdot \min(w)}{\alpha}$$

OmniQuant

POSTECH

Preliminaries

OmniQuant (ICLR2024, Spotlight)

- Unlike QAT, OmniQuant does not update the model parameters.
- Instead, it learns additional scaling and shifting parameters through gradient descent over layer-wise L2 error reconstruction.

QAT

$$Q_{\text{MM}}(w, c) = \text{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\max(w) - \min(w)}{2^c - 1}, \quad z = -\frac{\min(w)}{\alpha}$$

$$XW + b \longrightarrow X \cdot Q_{\text{MM}}(W) + b$$

OmniQuant

$$Q_{\text{Omni}}(w, c) = \text{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\gamma \cdot \max(w) - \beta \cdot \min(w)}{2^c - 1}, \quad z = -\frac{\beta \cdot \min(w)}{\alpha}$$

Smoothing Factor

$$XW + b \longrightarrow ((X - \delta) \oslash s) \cdot Q_{\text{Omni}}(W \odot s) + b + \delta \cdot W$$

Shifting Factor

$X \in \mathbb{R}^{n \times d}$
$W \in \mathbb{R}^{d \times d_o}$
$b \in \mathbb{R}^{d_o}$
$\delta \in \mathbb{R}^{d}$
$s \in \mathbb{R}^{d}$

QAT

OmniQuant

Preliminaries

OmniQuant (ICLR2024, Spotlight)

- Unlike QAT, OmniQuant does not update the model parameters.
- Instead, it learns additional scaling and shifting parameters through gradient descent over layer-wise L2 error reconstruction.

QAT

$$Q_{\mathrm{MM}}(w,c) = \mathrm{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\max(w) - \min(w)}{2^c - 1}, \quad z = -\frac{\min(w)}{\alpha}$$

$$XW + b \longrightarrow X \cdot Q_{\mathrm{MM}}(W) + b$$

Cross Entropy Loss

$$\min_{W_F} \frac{1}{N} \sum_{i \in [N]} \mathcal{L}_{\mathrm{CE}}\left(F(x_i; Q_{\mathrm{MM}}(W_F, c)), y_i\right)$$

OmniQuant

$$Q_{\mathrm{Omni}}(w,c) = \mathrm{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil, 0, 2^c - 1\right)$$

$$\alpha = \frac{\gamma \cdot \max(w) - \beta \cdot \min(w)}{2^c - 1}, \quad z = -\frac{\beta \cdot \min(w)}{\alpha}$$

Smoothing Factor

$$XW + b \longrightarrow ((X - \delta) \oslash s) \cdot Q_{\mathrm{Omni}}(W \odot s) + b + \delta \cdot W$$

Shifting Factor

Layer-Wise L2 Error

$$\min_{\gamma,\beta,\delta,s} ||F_l(W_F^l), X_l) - F_l(Q_{\mathrm{Omni}}(W_F^l), X_l)||_2^2$$

$X \in \mathbb{R}^{n \times d}$

$W \in \mathbb{R}^{d \times d_o}$

$b \in \mathbb{R}^{d_o}$

$\delta \in \mathbb{R}^{d}$

$s \in \mathbb{R}^{d}$

QAT

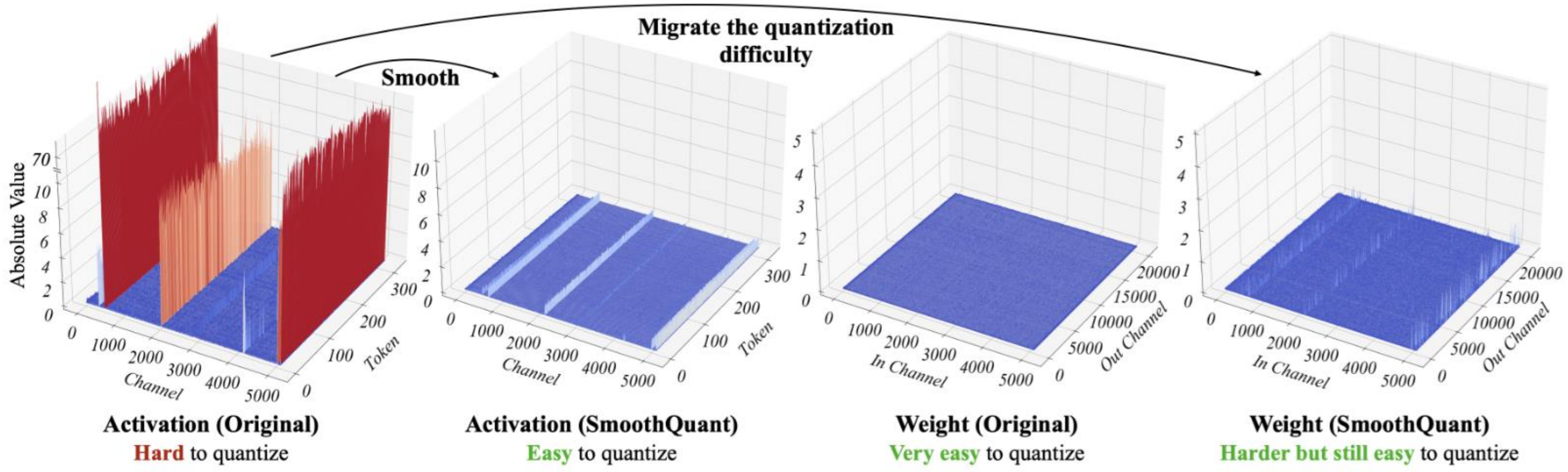
OmniQuant

Preliminaries

Smoothing Factor ; s

$$XW+b \rightarrow ((X-\delta) \oslash s)\cdot Q_{\mathrm{Omni}}(W \odot s)+b+\delta \cdot W$$

- The smoothing factor redistributes the quantization difficulty caused by activation outliers to the weights.
- The smoothing factor enables a mathematically equivalent transformation. $\mathbf{Y} = (\mathbf{X}\mathrm{diag}(\mathbf{s})^{-1}) \cdot (\mathrm{diag}(\mathbf{s})\mathbf{W}) = \hat{\mathbf{X}}\hat{\mathbf{W}}$



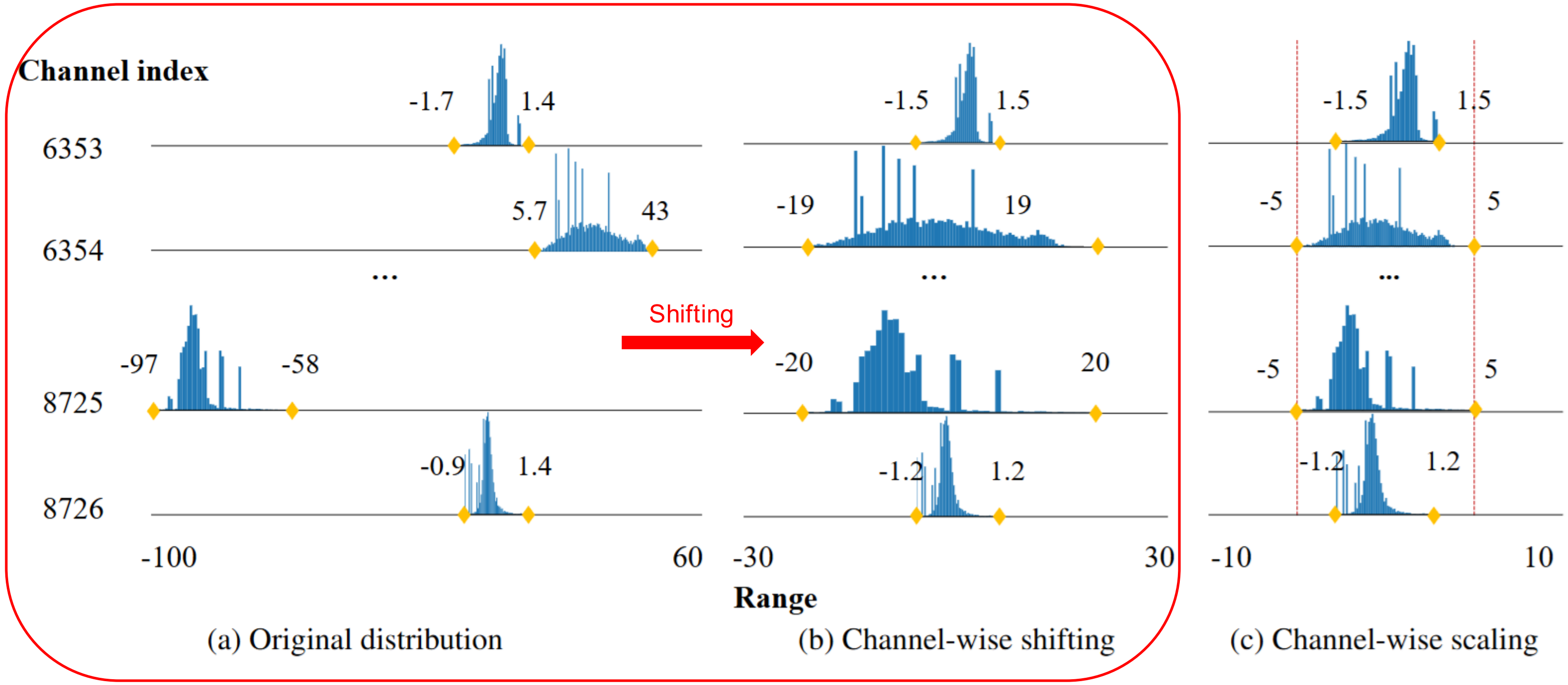
SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models

Preliminaries

Shifting Factor ; δ

$$XW+b \rightarrow ((X-\delta) \oslash s)\cdot Q_{\mathrm{Omni}}(W \odot s)+b+\delta \cdot W$$

- The shifting factor aligns channel centers to remove asymmetric outliers, making the distribution easier to quantize.
- The shifting factor enables a mathematically equivalent transformation.



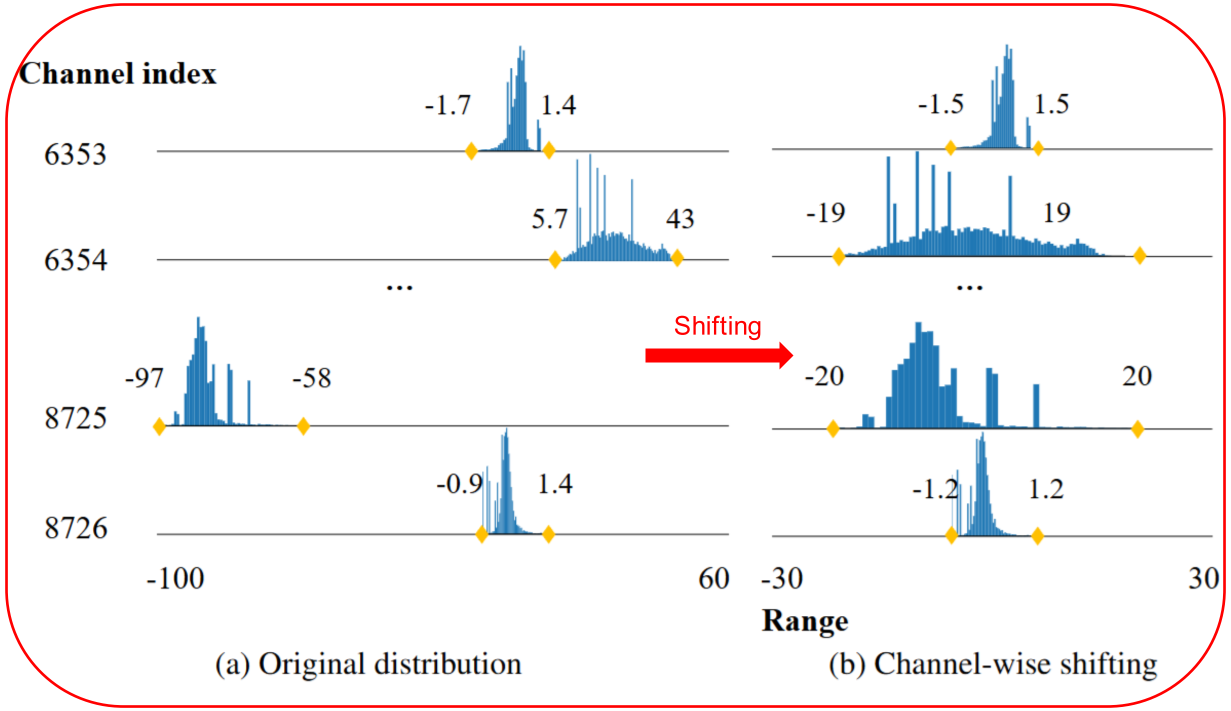
Outlier Suppression+ : Accurate quantization of large language models by equivalent and optimal shifting and scaling

Preliminaries

Shifting Factor ; δ

$$XW+b \rightarrow ((X-\delta)\oslash s)\cdot Q_{\text{Omni}}(W\odot s)+b+\delta\cdot W$$

- The shifting factor aligns channel centers to remove asymmetric outliers, making the distribution **easier to quantize**.
- The shifting factor enables a **mathematically equivalent** transformation.



$$\alpha = \frac{\max(w)-\min(w)}{2^c-1}$$

Assuming c = 8 (bit)

(Before shifting)

$$\alpha = \frac{43-(-97)}{255} = 0.549$$

(After shifting)

$$\alpha = \frac{20-(-20)}{255} = 0.157$$

Outlier Suppression+ : Accurate quantization of large language models by equivalent and optimal shifting and scaling

Method

MatQuant

- If we want to extract a ***r*-bit** model from a ***c*-bit** model ($0 < r < c$), we can just **slice out** the r most significant bits (MSBs) – using a right shift, followed by a left shift of the same order.

$$q^c = Q(w,c) = \text{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil,\ 0,\ 2^c - 1\right)$$

$$S(q^c, r) = \text{clamp}\left(\left\lfloor \frac{q^c}{2^{c-r}} \right\rceil, 0, 2^r - 1\right) * 2^{c-r}$$

- Example
  - c=8, r=4 (8bit → 4bit)
  - q8=234

$$\underbrace{\left\lfloor 234/16 \right\rceil}_{=\lfloor 14.625 \rceil = 14} \xrightarrow{\text{clamp}} 14 \xrightarrow{\times 16} 224$$

1 1 1 0 1 0 1 0 → 1 1 1 0 → 1 1 1 0 0 0 0 0

INT8 INT4

Method

MatQuant

- If we want to extract a r**-bit** model from a c**-bit** model ($0 < r < c$), we can just **slice out** the r most significant bits (MSBs) – using a right shift, followed by a left shift of the same order.

$$q^c = Q(w, c) = \mathrm{clamp}\left(\left\lfloor \frac{w}{\alpha} + z \right\rceil,\ 0,\ 2^c - 1\right)$$

$$S(q^c, r) = \mathrm{clamp}\left(\left\lfloor \frac{q^c}{2^{c-r}} \right\rceil, 0, 2^r - 1\right) * 2^{c-r}$$

- MatQuant's overall objective (Weight Quantization on FFN)

$$\min_P \frac{1}{N} \sum_{i \in [N]} \sum_{r \in R} \lambda_r \cdot \mathcal{L}\left(F(S(Q(\theta, c), r), x'_i), y'_i\right)$$

R = {8,4,2}

λ_r = Loss reweighing factor for bit-width r

Experiment Setting

MatQuant working with two popular **learning based quantization methods**:

1. **OmniQuant**
2. **QAT**

Models & Target Bit precisions

- Gemma-2 2B, 9B / Mistral 7B models.
- Default target quantization precisions **: int8, int4, int2**
  \+ the interpolative nature of MatQuant through evaluations on **int6 and int3**

Training

OmniQuant

- 128 examples with a sequence length of 2048 from the **C4 dataset** train using a batch size of 4
- train for a total of 10M tokens for all models except the int2 baseline,
  where we train the model for 20M tokens

QAT

- sample a fixed set of 100M tokens from the **C4 dataset** ,
  and train all our models using a batch size of 16 and a sequence length of 8192 for a single epoch

Evaluation Datasets

Calculating Perplexity with C4's test set

Downstream evaluations with zero-shot accuracy

- ARC-c, ARC-e.
- BoolQ
- HellaSwag
- PIQA
- Winogrande

Q. What is PPL ?

A. Perplexity (PPL) is a metric that measures **how well a language model predicts a sequence**. **lower PPL** values indicate **better performance.**

MatQuant with OmniQuant

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	OmniQuant	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	68.25	2.552	74.59	2.418	73.77	2.110
	MatQuant	68.02	2.570	74.05	2.438	73.65	2.125
int4	Sliced int8	62.87	2.730	72.26	2.480	38.51	4.681
	Baseline	67.03	2.598	74.33	2.451	73.62	2.136
	MatQuant	66.58	2.618	73.83	2.491	73.06	2.153
int2	Sliced int8	39.78	17.030	38.11	15.226	37.29	11.579
	Baseline	51.33	3.835	60.24	3.292	59.74	3.931
	MatQuant	**52.37**	**3.800**	**63.35**	**3.187**	**62.75**	**3.153**

MatQuant with OmniQuant

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	OmniQuant	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	68.25	2.552	74.59	2.418	73.77	2.110
	MatQuant	68.02	2.570	74.05	2.438	73.65	2.125
							81
int4	Baseline	67.03	2.598	74.33	2.451	73.62	2.136
	MatQuant	66.58	2.618	73.83	2.491	73.06	2.153
int2	Sliced int8	39.78	17.030	38.11	15.226	37.29	11.579
	Baseline	51.33	3.835	60.24	3.292	59.74	3.931
	MatQuant	**52.37**	**3.800**	**63.35**	**3.187**	**62.75**	**3.153**

Baseline (OmniQuant) is better, but MatQuant shows comparable performance

MatQuant with OmniQuant

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	OmniQuant	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	68.25	2.552	74.59	2.418	73.77	2.110
	MatQuant	68.02	2.570	74.05	2.438	73.65	2.125
int4	Sliced int8	62.87	2.730	72.26	2.480	38.51	4.681
	Baseline	67.03	2.598	74.33	2.451	73.62	2.136
	MatQuant	66.58	2.618	73.83	2.491	73.06	2.153
int2	Sliced int8	39.78	17.030	38.11	15.226	37.29	11.579
	Baseline	51.33	3.835	60.24	3.292	59.74	3.931
	MatQuant	**52.37**	**3.800**	**63.35**	**3.187**	**62.75**	**3.153**

In int2, MatQuant shows more accurate performance

MatQuant with OmniQuant

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	OmniQuant	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	68.25	2.552	74.59	2.418	73.77	2.110
int4	Sliced int8	62.87	2.730	72.26	2.480	38.51	4.681
	Baseline	67.03	2.598	74.33	2.451	73.62	2.136
	MatQuant	66.58	2.618	73.83	2.491	73.06	2.153
int2	Sliced int8	39.78	17.030	38.11	15.226	37.29	11.579
	Baseline	51.33	3.835	60.24	3.292	59.74	3.931
	MatQuant	**52.37**	**3.800**	**63.35**	**3.187**	**62.75**	**3.153**

Naïve bit slicing shows significant drop in accuracy

MatQuant with OmniQuant

Sliced Interpolation.

- Beyond the target quantization granularities (int8, int4, and int2), MatQuant allows for bit-width interpolation to bit-widths not optimized during training

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	OmniQuant	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
int6	Sliced int8	67.72	2.497	74.64	2.353	73.00	2.071
	Baseline	68.06	2.554	74.23	2.420	74.10	2.112
	MatQuant	67.52	2.574	73.92	2.440	73.63	2.127
int3	Sliced int8	41.35	6.024	54.18	3.977	39.21	10.792
	Baseline	64.37	2.727	73.23	2.549	71.68	2.211
	MatQuant	64.47	2.618	72.87	2.607	71.16	2.238

MatQuant with QAT

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	67.82	2.458	74.17	2.29	73.48	2.084
	MatQuant	67.44	2.449	74.52	2.262	72.58	2.104
int4	Sliced int8	67.13	2.483	73.36	2.276	71.76	2.18
	Baseline	67.03	2.512	73.26	2.324	72.13	2.105
	MatQuant	66.59	2.499	73.24	2.429	71.99	2.148
int2	Sliced int8	39.27	10.217	40.40	7.259	37.41	9.573
	Baseline	47.74	3.433	56.02	2.923	54.95	2.699
	MatQuant	**52.20**	**3.055**	**62.29**	**2.265**	**61.97**	**2.524**

MatQuant with QAT

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	67.82	2.458	74.17	2.29	73.48	2.084
	MatQuant	67.44	2.449	74.52	2.262	72.58	2.104
int4	Baseline	67.03	2.512	73.26	2.324	72.13	2.105
	MatQuant	66.59	2.499	73.24	2.429	71.99	2.148
int2	Sliced int8	39.27	10.217	40.40	7.259	37.41	9.573
	Baseline	47.74	3.433	56.02	2.923	54.95	2.699
	MatQuant	**52.20**	**3.055**	**62.29**	**2.265**	**61.97**	**2.524**

Baseline (QAT) is better, but MatQuant shows comparable performance

MatQuant with QAT

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	67.82	2.458	74.17	2.29	73.48	2.084
	MatQuant	67.44	2.449	74.52	2.262	72.58	2.104
int4	Sliced int8	67.13	2.483	73.36	2.276	71.76	2.18
	Baseline	67.03	2.512	73.26	2.324	72.13	2.105
	MatQuant	66.59	2.499	73.24	2.429	71.99	2.148
int2	Sliced int8	39.27	10.217	40.40	7.259	37.41	9.573
	Baseline	47.74	3.433	56.02	2.923	54.95	2.699
	MatQuant	**52.20**	**3.055**	**62.29**	**2.265**	**61.97**	**2.524**

In int2, MatQuant shows more accurate performance

POSTECH

MatQuant with QAT

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	67.82	2.458	74.17	2.29	73.48	2.084
int4	Sliced int8	67.13	2.483	73.36	2.276	71.76	2.18
	Baseline	67.03	2.512	73.26	2.324	72.13	2.105
	MatQuant	66.59	2.499	73.24	2.429	71.99	2.148
int2	Sliced int8	39.27	10.217	40.40	7.259	37.41	9.573
	Baseline	47.74	3.433	56.02	2.923	54.95	2.699
	MatQuant	**52.20**	**3.055**	**62.29**	**2.265**	**61.97**	**2.524**

Naïve bit slicing shows significant drop in accuracy

MatQuant with QAT

Sliced Interpolation.

- Models trained using MatQuant with QAT exhibit strong interpolative performance similar to that of MatQuant with OmniQuant.

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
int6	Sliced int8	67.72	2.497	74.64	2.353	73.00	2.071
	Baseline	68.06	2.554	74.23	2.420	74.10	2.112
	MatQuant	67.52	2.574	73.92	2.440	73.63	2.127
int3	Sliced int8	41.35	6.024	54.18	3.977	39.21	10.792
	Baseline	64.37	2.727	73.23	2.549	71.68	2.211
	MatQuant	64.47	2.618	72.87	2.607	71.16	2.238

Comparison OmniQuant vs QAT

- While OmniQuant only trains the auxiliary parameters needed for quantization, **QAT also updates the weight parameters.**

Data type	Method	Gemma-2 2B	
	OmniQuant	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	Baseline	68.25	2.552
	MatQuant	68.02	2.570
int4	Sliced int8	62.87	2.730
	Baseline	67.03	2.598
	MatQuant	66.58	2.618
int2	Sliced int8	39.78	17.030
	Baseline	51.33	3.835
	MatQuant	**52.37**	**3.800**
int6	Sliced int8	67.72	2.497
	Baseline	68.06	2.554
	MatQuant	67.52	2.574
int3	Sliced int8	41.35	6.024
	Baseline	64.37	2.727
	MatQuant	64.47	2.618

Data type	Method	Gemma-2 2B	
	QAT	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	Baseline	67.82	2.458
	MatQuant	67.44	2.449
int4	Sliced int8	67.13	2.483
	Baseline	67.03	2.512
	MatQuant	66.59	2.499
int2	Sliced int8	39.27	10.217
	Baseline	47.74	3.433
	MatQuant	**52.20**	**3.055**
int6	Sliced int8	67.53	2.401
	Baseline	67.75	2.460
	MatQuant	67.33	2.453
int3	Sliced int8	59.56	2.882
	Baseline	61.75	2.678
	MatQuant	60.76	2.734

Comparison OmniQuant vs QAT

- While OmniQuant only trains the auxiliary parameters needed for quantization, **QAT also updates the weight parameters.**

Data type	Method	Gemma-2 2B	
	OmniQuant	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	Baseline	68.25	2.552
	MatQuant	68.02	2.570
int4	Sliced int8	62.87	2.730
	Baseline	67.03	2.598
	MatQuant	66.58	2.618
int2	Sliced int8	39.78	17.030
	Baseline	51.33	3.835
	MatQuant	**52.37**	**3.800**
int6	Sliced int8	67.72	2.497
	Baseline	68.06	2.554
	MatQuant	67.52	2.574
int3	Sliced int8	41.35	6.024
	Baseline	64.37	2.727
	MatQuant	64.47	2.618

Data type	Method	Gemma-2 2B	
	QAT	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	Baseline	67.82	2.458
	MatQuant	67.44	2.449
int4	Sliced int8	67.13	2.483
	Baseline	67.03	2.512
	MatQuant	66.59	2.499
int2	Sliced int8	39.27	10.217
	Baseline	47.74	3.433
	MatQuant	**52.20**	**3.055**
int6	Sliced int8	67.53	2.401
	Baseline	67.75	2.460
	MatQuant	67.33	2.453
int3	Sliced int8	59.56	2.882
	Baseline	61.75	2.678
	MatQuant	60.76	2.734

QAT exhibits **lower ppl** than Omniquant

Comparison OmniQuant vs QAT

- While OmniQuant only trains the auxiliary parameters needed for quantization, **QAT also updates the weight parameters.**

OmniQuant exhibits **higher Task Accuracy** than QAT

Data type	Method	Gemma-2 2B	
	OmniQuant	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	Baseline	68.25	2.552
	MatQuant	68.02	2.570
int4	Sliced int8	62.87	2.730
	Baseline	67.03	2.598
	MatQuant	66.58	2.618
int2	Sliced int8	39.78	17.030
	Baseline	51.33	3.835
	MatQuant	**52.37**	**3.800**
int6	Sliced int8	67.72	2.497
	Baseline	68.06	2.554
	MatQuant	67.52	2.574
int3	Sliced int8	41.35	6.024
	Baseline	64.37	2.727
	MatQuant	64.47	2.618

Data type	Method	Gemma-2 2B	
	QAT	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	Baseline	67.82	2.458
	MatQuant	67.44	2.449
int4	Sliced int8	67.13	2.483
	Baseline	67.03	2.512
	MatQuant	66.59	2.499
int2	Sliced int8	39.27	10.217
	Baseline	47.74	3.433
	MatQuant	**52.20**	**3.055**
int6	Sliced int8	67.53	2.401
	Baseline	67.75	2.460
	MatQuant	67.33	2.453
int3	Sliced int8	59.56	2.882
	Baseline	61.75	2.678
	MatQuant	60.76	2.734

Comparison OmniQuant vs QAT

- While OmniQuant only trains the auxiliary parameters needed for quantization, **QAT also updates the weight parameters.**

OmniQuant exhibits **higher Task Accuracy** than QAT

Data type	Method	Gemma-2 2B	
	OmniQuant	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	Baseline	68.25	2.552
	MatQuant	68.02	2.570
int4	Sliced int8	62.87	2.730
	Baseline	67.03	2.598
	MatQuant	66.58	2.618
int2	Sliced int8	39.78	17.030
	Baseline	51.33	3.835
	MatQuant	**52.37**	**3.800**
int6	Sliced int8	67.72	2.497
	Baseline	68.06	2.554
	MatQuant	67.52	2.574
int3	Sliced int8	41.35	6.024
	Baseline	64.37	2.727
	MatQuant	64.47	2.618

Data type	Method	Gemma-2 2B	
	QAT	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	Baseline	67.82	2.458
	MatQuant	67.44	2.449
int4	Sliced int8	67.13	2.483
	Baseline	67.03	2.512
	MatQuant	66.59	2.499
int2	Sliced int8	39.27	10.217
	Baseline	47.74	3.433
	MatQuant	**52.20**	**3.055**
int6	Sliced int8	67.53	2.401
	Baseline	67.75	2.460
	MatQuant	67.33	2.453
int3	Sliced int8	59.56	2.882
	Baseline	61.75	2.678
	MatQuant	60.76	2.734

QAT exhibits **lower ppl** than Omniquant

Comparison OmniQuant vs QAT

- While OmniQuant only trains the auxiliary parameters needed for quantization, **QAT also updates the weight parameters.**

QAT ➔ overfitting to the C4 subset

Data type	Method	Gemma-2 2B	
	OmniQuant	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	[illegible]	[illegible]	[illegible]
	[illegible]	[illegible]	[illegible]
int4	Sliced int8	62.87	2.730
	Baseline	67.03	2.598
	MatQuant	66.58	2.618
int2	Sliced int8	39.78	17.030
	Baseline	51.33	3.835
	MatQuant	**52.37**	**3.800**
int6	Sliced int8	67.72	2.497
	Baseline	68.06	2.554
	MatQuant	67.52	2.574
int3	Sliced int8	41.35	6.024
	Baseline	64.37	2.727
	MatQuant	64.47	2.618

Data type	Method	Gemma-2 2B	
	QAT	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8	[illegible]	[illegible]	2.458
	[illegible]	[illegible]	2.449
int4	Sliced int8	67.13	2.483
	Baseline	67.03	2.512
	MatQuant	66.59	2.499
int2	Sliced int8	39.27	10.217
	Baseline	47.74	3.433
	MatQuant	**52.20**	**3.055**
int6	Sliced int8	67.53	2.401
	Baseline	67.75	2.460
	MatQuant	67.33	2.453
int3	Sliced int8	59.56	2.882
	Baseline	61.75	2.678
	MatQuant	60.76	2.734

OmniQuant exhibits **higher Task Accuracy** than QAT

QAT exhibits **lower ppl** than Omniquant

Comparison OmniQuant vs QAT

- While OmniQuant only trains the auxiliary parameters needed for quantization, **QAT also updates the weight parameters.**

OmniQuant exhibits **higher Task Accuracy** than QAT

Data type	Method	Gemma-2 2B	
	OmniQuant	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8			
int4			
int2			
	MatQuant	**52.37**	**3.800**
int6	Sliced int8	67.72	2.497
	Baseline	68.06	2.554
	MatQuant	67.52	2.574
int3	Sliced int8	41.35	6.024
	Baseline	64.37	2.727
	MatQuant	64.47	2.618

Data type	Method	Gemma-2 2B	
	QAT	Task Avg.	log pplx.
bfloat16		68.21	2.551
int8			2.458
			2.449
int4			2.483
			2.512
			2.499
int2			10.217
			3.433
	MatQuant	**52.20**	**3.055**
int6	Sliced int8	67.53	2.401
	Baseline	67.75	2.460
	MatQuant	67.33	2.453
int3	Sliced int8	59.56	2.882
	Baseline	61.75	2.678
	MatQuant	60.76	2.734

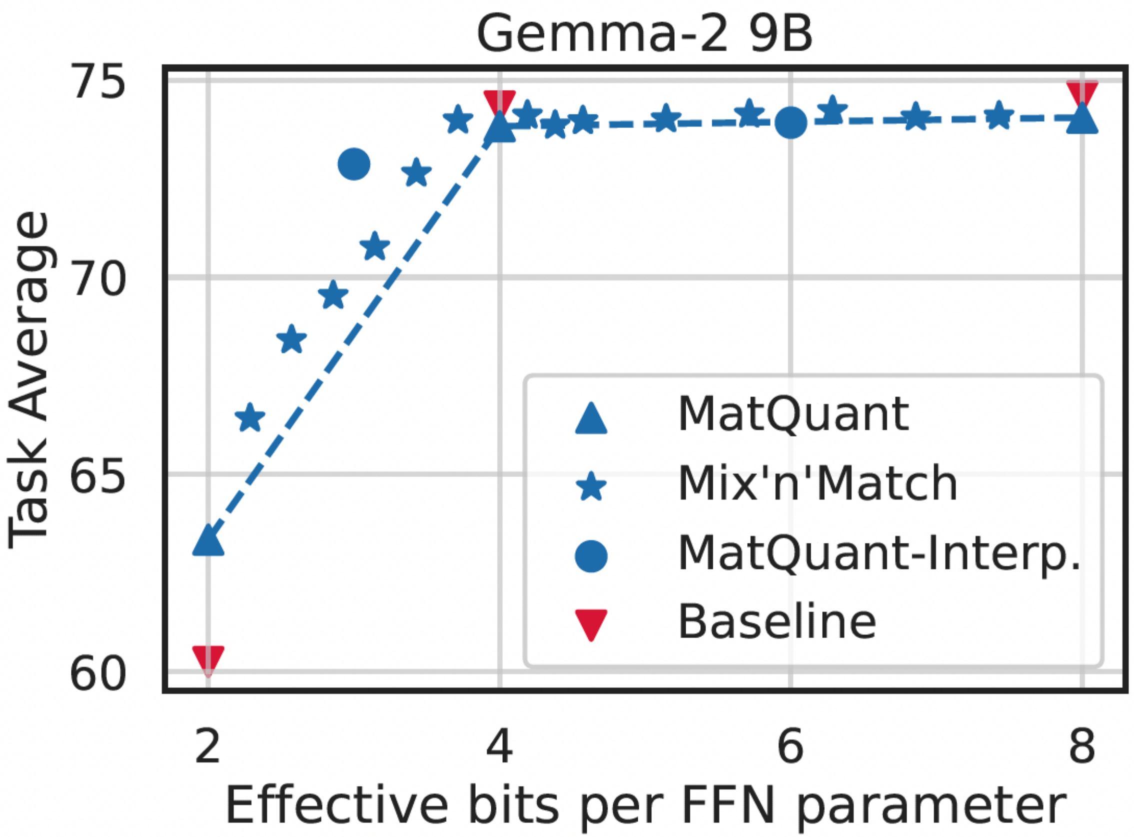
QAT ➔ overfitting to the C4 subset

1. **the need for high-quality data for QAT**
2. **Users are better off using resource-friendly methods like OmniQuant.**

QAT exhibits **lower ppl** than Omniquant

Additional: Layerwise Mix'n'Match

- Mix'n'Match provides a mechanism to **obtain a combinatorial number of strong models** by using layerwise different quantization granularities ,
  from the target bit-widths – i.e., int8, int4, and int2 across layers



Ablation studies: Weightings (λr) for MatQuant

$$\min_{P} \frac{1}{N} \sum_{i \in [N]} \sum_{r \in R} \boxed{\lambda_r} \cdot \mathcal{L}\left(F(S(Q(\theta, c), r), x_i'), y_i'\right)$$

Loss coefficient for each target bits (8,4,2 bits)

< overall objective of MatQuant>

Data type	Weightings	Gemma-2 2B	Gemma-2 9B	Mistral 7B
	8 4 2		Task Avg.	
int8	(0.1, 0.1, 1)	**68.02**	**74.05**	73.27
	(0.2, 0.2, 1)	67.91	73.91	73.44
	(0.3, 0.3, 1)	68.01	73.88	73.56
	(0.4, 0.4, 1)	67.95	73.84	**73.65**
int4	(0.1, 0.1, 1)	66.58	73.83	72.76
	(0.2, 0.2, 1)	67.47	73.8	73.16
	(0.3, 0.3, 1)	66.97	73.25	73.47
	(0.4, 0.4, 1)	**67.48**	**74.32**	**73.66**
int2	(0.1, 0.1, 1)	**52.37**	63.35	63.25
	(0.2, 0.2, 1)	51.88	64.04	**63.99**
	(0.3, 0.3, 1)	51.05	**64.1**	63.6
	(0.4, 0.4, 1)	51.69	61.98	62.75

Ablation studies: Weightings (λr) for MatQuant

$$\min_P \frac{1}{N} \sum_{i \in [N]} \sum_{r \in R} \lambda_r \cdot \mathcal{L}\left(F(S(Q(\theta, c), r), x_i'), y_i'\right)$$

Loss coefficient for each target bits (8,4,2 bits)

< overall objective of MatQuant>

Low coefficient for 8bit/4bit

→ Higher accuracy in int8/int4

→ Lower accuracy in int2

Data type	Weightings	Gemma-2 2B	Gemma-2 9B	Mistral 7B
	8 4 2	Task Avg.		
int8	(0.1, 0.1, 1)	**68.02**	**74.05**	73.27
	(0.2, 0.2, 1)	67.91	73.91	73.44
	(0.3, 0.3, 1)	68.01	73.88	73.56
	(0.4, 0.4, 1)	67.95	73.84	**73.65**
int4	(0.1, 0.1, 1)	66.58	73.83	72.76
	(0.2, 0.2, 1)	67.47	73.8	73.16
	(0.3, 0.3, 1)	66.97	73.25	73.47
	(0.4, 0.4, 1)	**67.48**	**74.32**	**73.66**
int2	(0.1, 0.1, 1)	**52.37**	63.35	63.25
	(0.2, 0.2, 1)	51.88	64.04	**63.99**
	(0.3, 0.3, 1)	51.05	**64.1**	63.6
	(0.4, 0.4, 1)	51.69	61.98	62.75

Ablation studies: Weightings (λr) for MatQuant

$$\min_{P} \frac{1}{N} \sum_{i \in [N]} \sum_{r \in R} \lambda_r \cdot \mathcal{L}\left(F(S(Q(\theta, c), r), x'_i), y'_i\right)$$

λ_r: Loss coefficient for each target bits (8,4,2 bits)

< overall objective of MatQuant>

Data type	Weightings	Gemma-2 2B	Gemma-2 9B	Mistral 7B
	8 4 2	Task Avg.		
int8	(0.1, 0.1, 1)	**68.02**	**74.05**	73.27
	(0.2, 0.2, 1)	67.91	73.91	73.44
	(0.3, 0.3, 1)	68.01	73.88	73.56
	(0.4, 0.4, 1)	67.95	73.84	**73.65**
int4	(0.1, 0.1, 1)	66.58	73.83	72.76
	(0.2, 0.2, 1)	67.47	73.8	73.16
	(0.3, 0.3, 1)	66.97	73.25	73.47
	(0.4, 0.4, 1)	**67.48**	**74.32**	**73.66**
int2	(0.1, 0.1, 1)	**52.37**	63.35	63.25
	(0.2, 0.2, 1)	51.88	64.04	**63.99**
	(0.3, 0.3, 1)	51.05	**64.1**	63.6
	(0.4, 0.4, 1)	51.69	61.98	62.75

High coefficient for 8bit/4bit

→ Higher accuracy in int2

→ Lower accuracy in int8/int4

POSTECH

Ablation studies: Single Precision (S.P.) MatQuant

- Eliminate other target bits loss (8bit & 4bit), except for 2bit loss

$$\min_P \frac{1}{N} \sum_{i \in [N]} \sum_{r \in R} \boxed{\lambda_r} \cdot \mathcal{L}\left(F(S(Q(\theta, c), r), x_i'), y_i'\right)$$

Loss a for each target bits (8,4,2 bits)

< overall objective of MatQuant>

λ_r : r is a target bit, λ_8, λ_4 : 0, λ_2 : 1

int2	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
Method	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
OmniQuant	51.33	3.835	60.24	3.292	59.74	3.931
S.P. MatQuant	**53.42**	**3.631**	**64.02**	**3.171**	**63.58**	**2.976**
MatQuant	52.37	3.800	63.35	3.187	62.75	3.153
QAT	47.74	3.433	56.02	2.923	54.95	2.699
S.P. MatQuant	52.08	**3.054**	**62.66**	**2.656**	61.48	**2.509**
MatQuant	**52.20**	3.055	62.29	2.660	**61.97**	2.524

Ablation studies: Co-distillation for MatQuant

- Outputs from a higher-precision model ➔ used for lower-precision nested model training. either in a standalone fashion or alongside the ground truth target (weighted equally).

Gemma-2 9B		OmniQuant		QAT	
Data type	Config.	Task Avg.	log pplx.	Task Avg.	log pplx.
int8	[8, 4, 2]	**74.05**	2.438	74.52	2.262
	[8, 4, 8 → 2]	72.76	2.473	74.75	2.242
	[8, 4, 2, 8 → 2]	73.99	**2.435**	**74.87**	**2.240**
	[8, 4, 2, 8 → 4; 2]	73.85	2.437	74.81	2.240
int4	[8, 4, 2]	**73.83**	2.491	73.24	2.295
	[8, 4, 8 → 2]	72.65	2.519	73.76	2.279
	[8, 4, 2, 8 → 2]	73.63	2.486	73.77	**2.276**
	[8, 4, 2, 8 → 4; 2]	73.55	**2.478**	**73.93**	2.277
int2	[8, 4, 2]	63.35	**3.187**	62.29	**2.660**
	[8, 4, 8 → 2]	62.64	3.289	62.31	2.670
	[8, 4, 2, 8 → 2]	62.91	3.138	**62.70**	2.673
	[8, 4, 2, 8 → 4; 2]	**64.32**	3.227	62.60	2.670

Ablation studies: Co-distillation for MatQuant

- Outputs from a higher-precision model ➔ used for lower-precision nested model training. either in a standalone fashion or alongside the ground truth target (weighted equally).

8→ 4: use int8 outputs for int4 training

8→2 : use int8 outputs for int2 training

Gemma-2 9B		OmniQuant		QAT	
Data type	Config.	Task Avg.	log pplx.	Task Avg.	log pplx.
	$[8, 4, 2]$	**74.05**	2.438	74.52	2.262
	$[8, 4, 8 \rightarrow 2]$	72.76	2.473	74.75	2.242
	$[8, 4, 2, 8 \rightarrow 2]$	73.99	**2.435**	**74.87**	**2.240**
	$[8, 4, 2, 8 \rightarrow 4; 2]$	73.85	2.437	74.81	2.240
int4	$[8, 4, 2]$	**73.83**	2.491	73.24	2.295
	$[8, 4, 8 \rightarrow 2]$	72.65	2.519	73.76	2.279
	$[8, 4, 2, 8 \rightarrow 2]$	73.63	2.486	73.77	**2.276**
	$[8, 4, 2, 8 \rightarrow 4; 2]$	73.55	**2.478**	**73.93**	2.277
int2	$[8, 4, 2]$	63.35	**3.187**	62.29	**2.660**
	$[8, 4, 8 \rightarrow 2]$	62.64	3.289	62.31	2.670
	$[8, 4, 2, 8 \rightarrow 2]$	62.91	3.138	**62.70**	2.673
	$[8, 4, 2, 8 \rightarrow 4; 2]$	**64.32**	3.227	62.60	2.670

Ablation studies: Co-distillation for MatQuant

- Outputs from a higher-precision model ➔ used for lower-precision nested model training. either in a standalone fashion or alongside the ground truth target (weighted equally).

Gemma-2 9B		OmniQuant		QAT	
Data type	Config.	Task Avg.	log pplx.	Task Avg.	log pplx.
int8	[8, 4, 2]	**74.05**	2.438	74.52	2.262
	[8, 4, 8 → 2]	72.76	2.473	74.75	2.242
	[8, 4, 2, 8 → 2]	73.99	**2.435**	**74.87**	**2.240**
	[8, 4, 2, 8 → 4; 2]	73.85	2.437	74.81	2.240
int4	[8, 4, 2]	**73.83**	2.491	73.24	2.295
	[8, 4, 8 → 2]	72.65	2.519	73.76	2.279
	[8, 4, 2, 8 → 2]	73.63	2.486	73.77	**2.276**
	[8, 4, 2, 8 → 4; 2]	73.55	**2.478**	**73.93**	2.277
int2	[8, 4, 2]	63.35	**3.187**	62.29	**2.660**
	[8, 4, 8 → 2]	62.64	3.289	62.31	2.670
	[8, 4, 2, 8 → 2]	62.91	3.138	**62.70**	2.673
	[8, 4, 2, 8 → 4; 2]	**64.32**	3.227	62.60	2.670

Ablation studies: FFN + ATTN Weight Quantization

- Using QAT, apply MatQuant to **FFN**, and also ATTN

Data type	Method	Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		74.38	2.418	73.99	2.110
int8	Baseline	74.61	2.353	73.73	2.091
	MatQuant	74.85	2.333	73.88	2.182
int4	Sliced int8	73.15	2.362	71.46	2.290
	Baseline	72.98	2.40	71.87	2.132
	MatQuant	74.01	2.396	71.44	2.441
int2	Sliced int8	38.97	23.467	35.06	10.640
	Baseline	-	-	-	-
	S.P. MatQuant	**45.69**	**3.780**	35.35	7.761
	MatQuant	44.19	3.826	**38.36**	**10.971**
int6	Sliced int8	74.49	2.290	73.61	2.104
	Baseline	74.65	2.357	73.72	2.093
	MatQuant	74.57	2.340	74.04	2.161
int3	Sliced int8	64.19	2.895	39.01	6.018
	Baseline	-	-	-	-
	S.P. MatQuant	67.68	2.520	67.59	2.335
	MatQuant	63.63	2.937	40.55	4.776

Ablation studies: FFN + ATTN Weight Quantization

- Using QAT, apply MatQuant to **FFN**, and also ATTN

< FFN MatQaunt >

Data type	Method	Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		74.38	2.418	73.99	2.110
int8	Baseline	74.17	2.29	73.48	2.084
	MatQuant	74.52	2.262	72.58	2.104
int4	Sliced int8	73.36	2.276	71.76	2.18
	Baseline	73.26	2.324	72.13	2.105
	MatQuant	73.24	2.429	71.99	2.148
int2	Sliced int8	40.40	7.259	37.41	9.573
	Baseline	56.02	2.923	54.95	2.699
	MatQuant	**62.29**	**2.265**	**61.97**	**2.524**
int6	Sliced int8	74.15	2.232	73.35	2.097
	Baseline	74.31	2.293	72.71	2.077
	MatQuant	74.30	2.265	72.59	2.106
int3	Sliced int8	68.70	2.512	64.33	2.493
	Baseline	69.9	2.43	68.82	2.197
	MatQuant	70.41	2.429	67.16	2.324

< ATTN + FFN MatQaunt >

Data type	Method	Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		74.38	2.418	73.99	2.110
int8	Baseline	74.61	2.353	73.73	2.091
	MatQuant	74.85	2.333	73.88	2.182
int4	Sliced int8	73.15	2.362	71.46	2.290
	Baseline	72.98	2.40	71.87	2.132
	MatQuant	74.01	2.396	71.44	2.441
int2	Sliced int8	38.97	23.467	35.06	10.640
	Baseline	-	-	-	-
	S.P. MatQuant	**45.69**	**3.780**	35.35	7.761
	MatQuant	44.19	3.826	**38.36**	**10.971**
int6	Sliced int8	74.49	2.290	73.61	2.104
	Baseline	74.65	2.357	73.72	2.093
	MatQuant	74.57	2.340	74.04	2.161
int3	Sliced int8	64.19	2.895	39.01	6.018
	Baseline	-	-	-	-
	S.P. MatQuant	67.68	2.520	67.59	2.335
	MatQuant	63.63	2.937	40.55	4.776

Ablation studies: FFN + ATTN Weight Quantization

- Using QAT, apply MatQuant to **FFN**, and also ATTN

< FFN MatQaunt >

Data type	Method	Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		74.38	2.418	73.99	2.110
int8	Baseline	74.17	2.29	73.48	2.084
	MatQuant	74.52	2.262	72.58	2.104
int4	Sliced int8	73.36	2.276	71.76	2.18
	Baseline	73.26	2.324	72.13	2.105
	MatQuant	73.24	2.429	71.99	2.148
int2	Sliced int8	40.40	7.259	37.41	9.573
	Baseline	56.02	2.923	54.95	2.699
	MatQuant	**62.29**	**2.265**	**61.97**	**2.524**
int6	Sliced int8	74.15	2.232	73.35	2.097
	Baseline	74.31	2.293	72.71	2.077
	MatQuant	74.30	2.265	72.59	2.106
int3	Sliced int8	68.70	2.512	64.33	2.493
	Baseline	69.9	2.43	68.82	2.197
	MatQuant	70.41	2.429	67.16	2.324

< ATTN + FFN MatQaunt >

Data type	Method	Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		74.38	2.418	73.99	2.110
int8	Baseline	74.61	2.353	73.73	2.091
	MatQuant	74.85	2.333	73.88	2.182
int4	Sliced int8	73.15	2.362	71.46	2.290
	Baseline	72.98	2.40	71.87	2.132
	MatQuant	74.01	2.396	71.44	2.441
int2	Sliced int8	38.97	23.467	35.06	10.640
	Baseline	-	-	-	-
	S.P. MatQuant	**45.69**	**3.780**	35.35	7.761
	MatQuant	44.19	3.826	**38.36**	**10.971**
int6	Sliced int8	74.49	2.290	73.61	2.104
	Baseline	74.65	2.357	73.72	2.093
	MatQuant	74.57	2.340	74.04	2.161
int3	Sliced int8	64.19	2.895	39.01	6.018
	Baseline	-	-	-	-
	S.P. MatQuant	67.68	2.520	67.59	2.335
	MatQuant	63.63	2.937	40.55	4.776

Very Poor Performance when Quantize ATTN & FFN Both!!

Additional Consideration

Deployment Considerations

- Current hardware accelerators → native support for int8 and int4 quantized models.
- Custom-implemented CUDA kernels → can support int2 and int3

Additional Consideration

Deployment Considerations

- Current hardware accelerators ➔ native support for int8 and int4 quantized models.
- Custom-implemented CUDA kernels ➔ can support int2 and int3

➔ To apply it to various settings as above, you need to be prepared for all kinds of configurations.

Additional Consideration

Deployment Considerations

- Current hardware accelerators → native support for int8 and int4 quantized models.
- Custom-implemented CUDA kernels → can support int2 and int3

→ To apply it to various settings as above, you need to be prepared for all kinds of configurations.

MatQuant can be a simple solution for deployment!

- MatQuant can generate a large number of models at inference time.
- Depending on the serving environment,
  we can choose between Mix'n'Match models and homogeneous sliced models.

Additional Consideration

Extension to Floating Point

- Extending MatQuant to floating-point representations, such as FP8 and FP4, presents significant challenges. → **Why?**

Additional Consideration

Extension to Floating Point

- Extending MatQuant to floating-point representations,
  such as FP8 and FP4, presents significant challenges. ➔ **Why?**

For example,

- slicing the first two or 4bits from 8bits is easy
- However, this would not be the case when slicing two exponent bits from FP8.

Additional Consideration

Extension to Floating Point

- Extending MatQuant to floating-point representations,
  such as FP8 and FP4, presents significant challenges. ➔ **Why?**

For example,

- slicing the first two or 4bits from 8bits is easy
- However, this would not be the case when slicing two exponent bits from FP8.

➔ **needs further research !**

Summary for MatQuant

Strength

1. Eliminates the need to perform quantization optimization multiple times for different bit precisions, as it can be handled with a single optimization.

Summary for MatQuant

Strength

1. Eliminates the need to perform quantization optimization multiple times for different bit precisions, as it can be handled with a single optimization.

2. For various deployment environments, the required bit precision can be allocated at the inference. In other words, specific optimization for each environment is not necessary.

Summary for MatQuant

Strength

1. Eliminates the need to perform quantization optimization multiple times for different bit precisions, as it can be handled with a single optimization.

2. For various deployment environments, the required bit precision can be allocated at the inference. In other words, specific optimization for each environment is not necessary.

3. Even with int8 and int4, it shows performance comparable to the baseline, and in particular, it demonstrates clear performance improvements over the baseline at int2.

Summary for MatQuant

Weakness

1. Poor Performance

- Most recent quantized models are deployed with 8-bit or 4-bit precision.
  ➔ because the performance degradation with 2-bit quant is too severe to justify the memory savings.
- However, MatQuant shows little to no performance improvement at int8 or int4, raising concerns about its practicality in real-world deployment scenarios.

Data type	Method	Gemma-2 2B		Gemma-2 9B		Mistral 7B	
	OmniQuant	Task Avg.	log pplx.	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		68.21	2.551	74.38	2.418	73.99	2.110
int8	Baseline	68.25	2.552	74.59	2.418	73.77	2.110
	MatQuant	68.02	2.570	74.05	2.438	73.65	2.125
int4	Sliced int8	62.87	2.730	72.26	2.480	38.51	4.681
	Baseline	67.03	2.598	74.33	2.451	73.62	2.136
	MatQuant	66.58	2.618	73.83	2.491	73.06	2.153
int2	Sliced int8	39.78	17.030	38.11	15.226	37.29	11.579
	Baseline	51.33	3.835	60.24	3.292	59.74	3.931
	MatQuant	**52.37**	**3.800**	**63.35**	**3.187**	**62.75**	**3.153**

< MatQuant with

Summary for MatQuant

Weakness

2. no justification for poor performance in ATTN/FFN Quant

- The paper merely states that applying QAT to both the attention and FFN modules leads to instability at extremely low bit settings.
- However, it does not provide any justification or further explanation for this observation.

< FFN MatQaunt >

Data type	Method	Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		74.38	2.418	73.99	2.110
int8	Baseline	74.17	2.29	73.48	2.084
	MatQuant	74.52	2.262	72.58	2.104
int4	Sliced int8	73.36	2.276	71.76	2.18
	Baseline	73.26	2.324	72.13	2.105
	MatQuant	73.24	2.429	71.99	2.148
int2	Sliced int8	40.40	7.259	37.41	9.573
	Baseline	56.02	2.923	54.95	2.699
	MatQuant	**62.29**	**2.265**	**61.97**	**2.524**
int6	Sliced int8	74.15	2.232	73.35	2.097
	Baseline	74.31	2.293	72.71	2.077
	MatQuant	74.30	2.265	72.59	2.106
int3	Sliced int8	68.70	2.512	64.33	2.493
	Baseline	69.9	2.43	68.82	2.197
	MatQuant	70.41	2.429	67.16	2.324

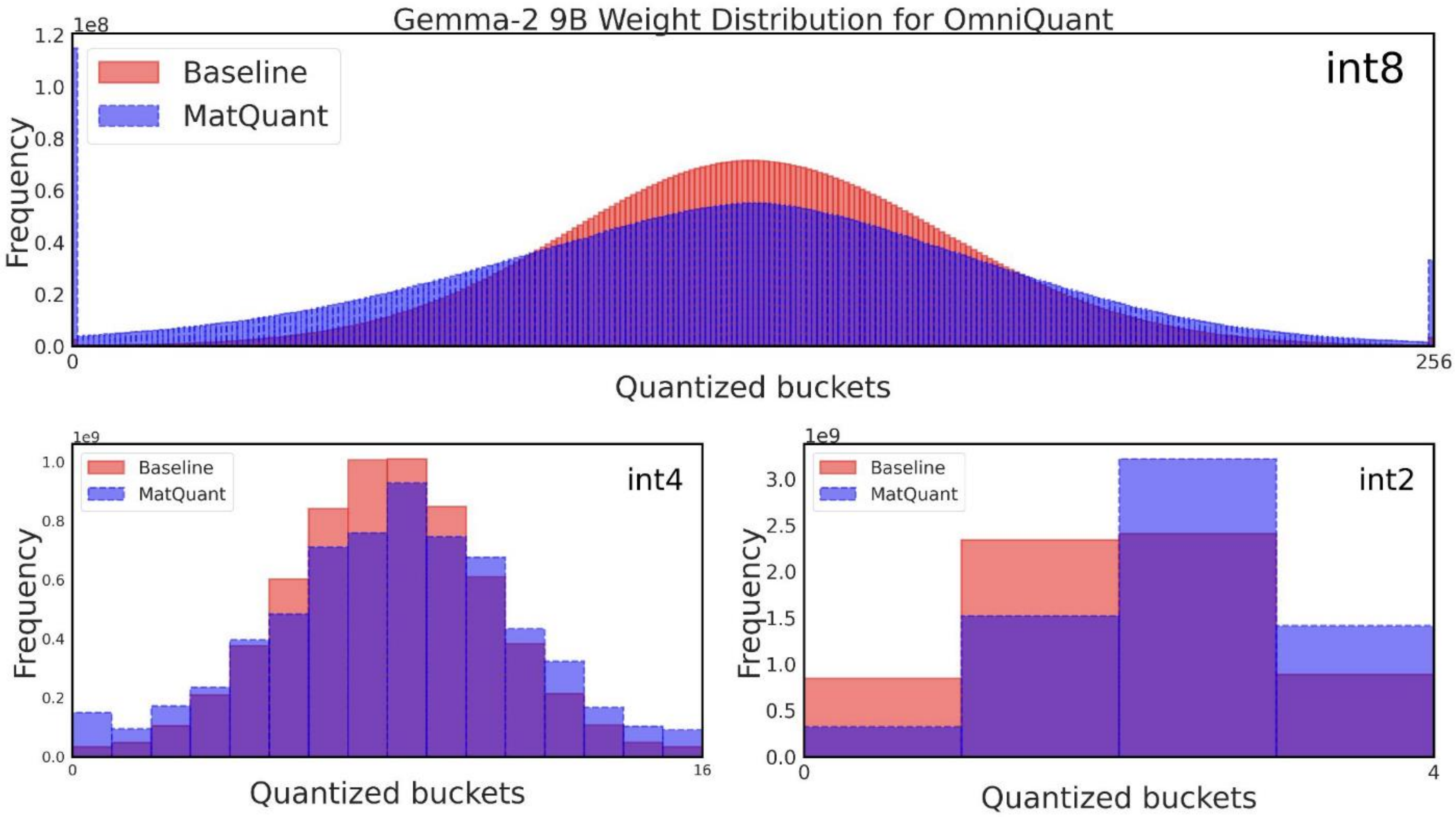
< ATTN + FFN MatQaunt >

Data type	Method	Gemma-2 9B		Mistral 7B	
	QAT	Task Avg.	log pplx.	Task Avg.	log pplx.
bfloat16		74.38	2.418	73.99	2.110
int8	Baseline	74.61	2.353	73.73	2.091
	MatQuant	74.85	2.333	73.88	2.182
int4	Sliced int8	73.15	2.362	71.46	2.290
	Baseline	72.98	2.40	71.87	2.132
	MatQuant	74.01	2.396	71.44	2.441
int2	Sliced int8	38.97	23.467	35.06	10.640
	Baseline	-	-	-	-
	S.P. MatQuant	**45.69**	**3.780**	35.35	7.761
	MatQuant	44.19	3.826	**38.36**	**10.971**
int6	Sliced int8	74.49	2.290	73.61	2.104
	Baseline	74.65	2.357	73.72	2.093
	MatQuant	74.57	2.340	74.04	2.161
int3	Sliced int8	64.19	2.895	39.01	6.018
	Baseline	-	-	-	-
	S.P. MatQuant	67.68	2.520	67.59	2.335
	MatQuant	63.63	2.937	40.55	4.776

Very Poor Performance when Quantize ATTN & FFN Both!!

Thank you.

Appendix


Gemma-2 9B Weight Distribution for OmniQuant
1e8
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
Baseline
MatQuant
int8
0
256
Quantized buckets
1e9
1.0
0.8
0.6
0.4
0.2
0.0
Frequency
Baseline
MatQuant
int4
0
16
Quantized buckets
1e9
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Frequency
Baseline
MatQuant
int2
0
4
Quantized buckets