Quantization Techniques for Efficient Large Language Model Inference

TinyML Asia

Jungwook Choi

Hanyang University, South Korea

choij@hanyang.ac.kr

2023. 11. 16 @ Sky 31 Convention

About the Speaker

Google Scholar https://scholar.google.com/citations?user=3qCaIbUAAAAJ&hl=ko&oi=ao

Jungwook Choi
FOLLOW
Hanyang University
Verified email at hanyang.ac.kr

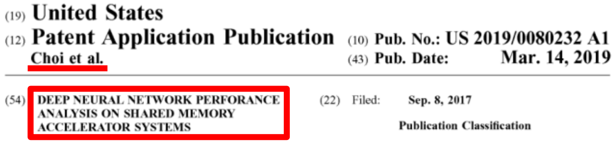
B.S.(2010), M.S.(2010) at SNU (South Korea)
Ph.D.(2015) at UIUC (US)

HW & SW research for efficient AI

TITLE	CITED BY	YEAR
Pact: Parameterized clipping activation for quantized neural networks J Choi, Z Wang, S Venkataramani, PIJ Chuang, V Srinivasan, ... arXiv preprint arXiv:1805.06085	803	2018
Training deep neural networks with 8-bit floating point numbers N Wang, J Choi, D Brand, CY Chen, K Gopalakrishnan Advances in neural information processing systems 31	483	2018
Hybrid 8-bit floating point (HFP8) training and inference for deep neural networks X Sun, J Choi, CY Chen, N Wang, S Venkataramani, VV Srinivasan, X Cui, ... Advances in neural information processing systems 32	179	2019
Accurate and Efficient 2-bit Quantized Neural Networks J Choi, S Venkataramani, V Srinivasan, K Gopalakrishnan, Z Wang, ... The Conference on Systems and Machine Learning (SysML)	175	2019
Adacomp: Adaptive residual gradient compression for data-parallel distributed training CY Chen, J Choi, D Brand, A Agrawal, W Zhang, K Gopalakrishnan Proceedings of the AAAI conference on artificial intelligence 32 (1)	172	2018
A scalable multi-TeraOPS deep learning processor core for AI trainina and inference B Fleischer, S Shukla, M Ziegler, J Silberman, J Oh, V Srinivasan, J Choi, ... 2018 IEEE symposium on VLSI circuits, 35-36	138	2018
Approximate computing: Challenges and opportunities A Agrawal, J Choi, K Gopalakrishnan, S Gupta, R Nair, J Oh, DA Prener, ... 2016 IEEE International Conference on Rebooting Computing (ICRC), 1-8	108	2016

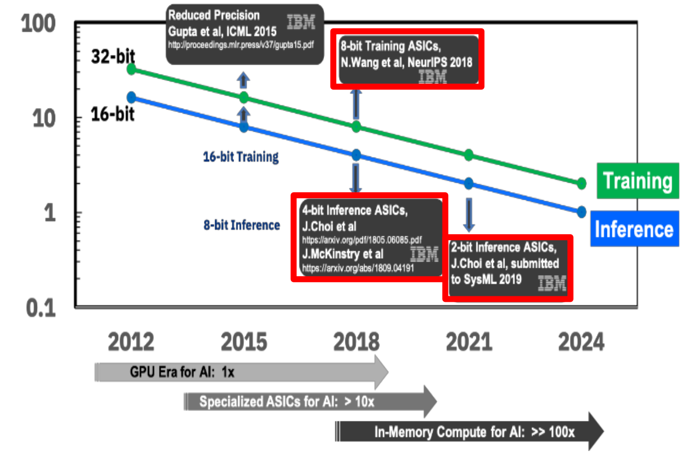
Research Highlights

- **Leading AI Accelerator Design at IBM Research (2015~2019)**
  - Develop **DNN accelerator chip** and its **SW stack** (→ IBM Telum)
  - Invent **8-bit DNN training** and **ultra-low bit DNN inference** methods



(19) United States
(12) Patent Application Publication
Choi et al.
(10) Pub. No.: US 2019/0080232 A1
(43) Pub. Date: Mar. 14, 2019
(54) DEEP NEURAL NETWORK PERFORANCE ANALYSIS ON SHARED MEMORY ACCELERATOR SYSTEMS
(22) Filed: Sep. 8, 2017
Publication Classification

IBM Research is Leading in Reduced Precision Scaling



- **Winner of AI Grand Challenge 2020 by Ministry of Science and ICT – Model Optimization Track ($600K Funding)**

홈 > 사회 > 교육

최정욱 한양대 교수팀, AI 그랜드챌린지 '경량화' 우승

(서울=뉴스1) 정지형 기자 | 2021-01-11 17:52 송고 news1 뉴스

Breakthrough in Deep Learning with Transformer Models

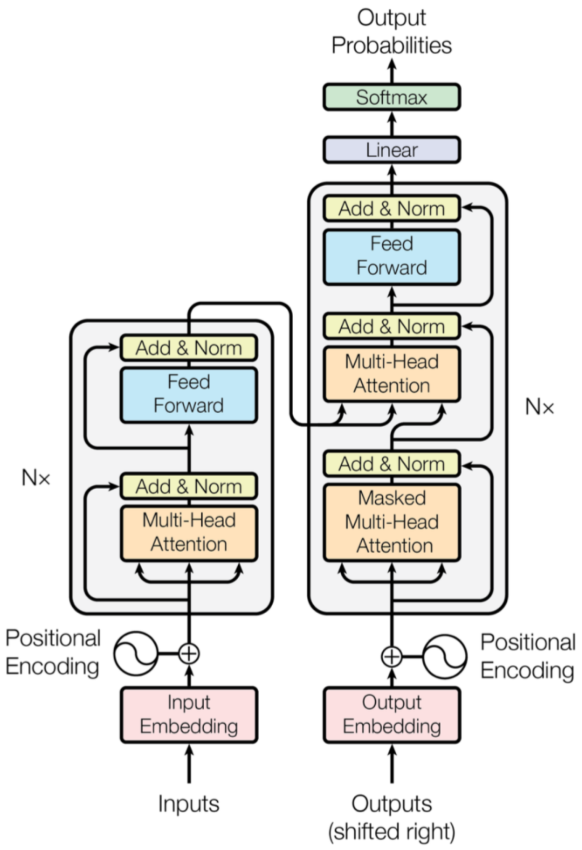


Figure 1: The Transformer - model architecture.

Translation

"The reason Boeing are doing this is to cram more seats in to make their plane more competitive with our products," said Kevin Keniston, head of passenger comfort at Europe's Airbus.

→ "La raison pour laquelle Boeing fait cela est de créer plus de sièges pour rendre son avion plus compétitif avec nos produits", a déclaré Kevin Keniston, chef du confort des passagers chez Airbus.

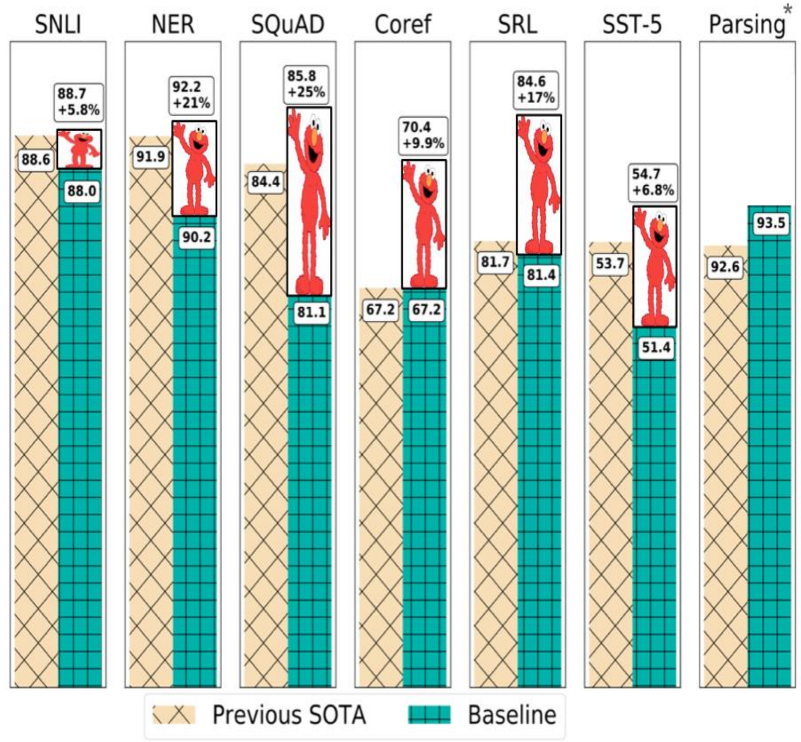
When asked about this, an official of the American administration replied: "The United States is not conducting electronic surveillance aimed at offices of the World Bank and IMF in Washington."

→ Interrogé à ce sujet, un fonctionnaire de l'administration américaine a répondu: "Les États-Unis n'effectuent pas de surveillance électronique à l'intention des bureaux de la Banque mondiale et du FMI à Washington"

Image Generation

Stable Diffusion Public Release — Stability AI

Natural Language Understanding



*Kitaev and Klein, ACL 2018 (see also Joshi et al., ACL 2018)

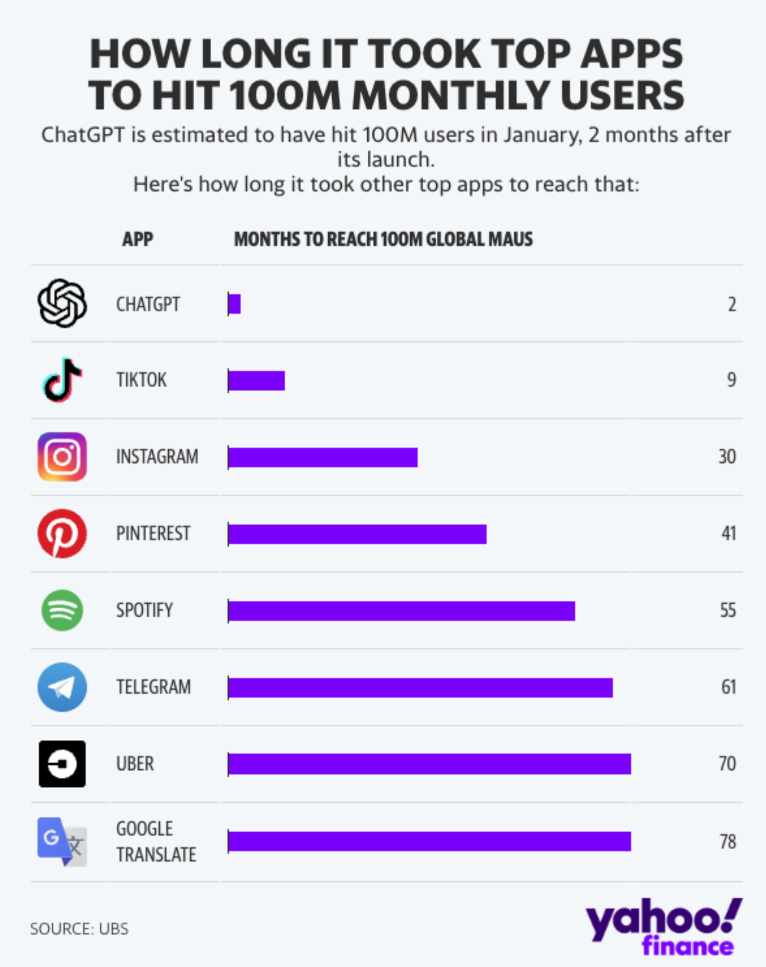
ChatGPT ("A Game Changer")

- Generative pre-trained transformer (GPT) tuned for language generation

Ask ChatGPT: Why LLM works so well?

why large language model is capable of answering questions in general? Please state top reasons in short sentences

1. Large language models are trained on massive amounts of data from various sources, which allows them to understand and generalize on the nuances of language and relationships between different concepts.
2. Large language models use transformer architecture, which allows them to process the entire input sequence at once and consider the context of the entire sentence, enabling them to understand the meaning of the question and generate an appropriate response.
3. Pre-training on a massive amount of text data enables large language models to understand the relationship between words, phrases and sentences, which allows them to generate human-like responses.



Challenges of LLM: Large Computational Cost

Exponential size increase of Hyper-scale AI models

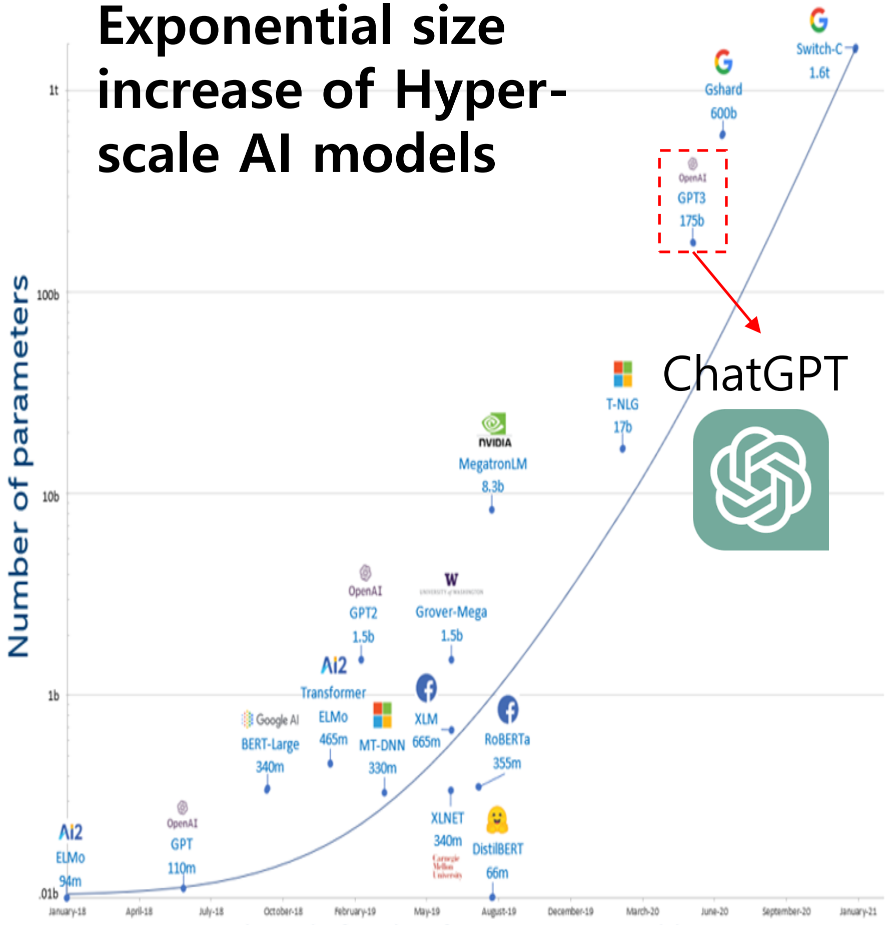
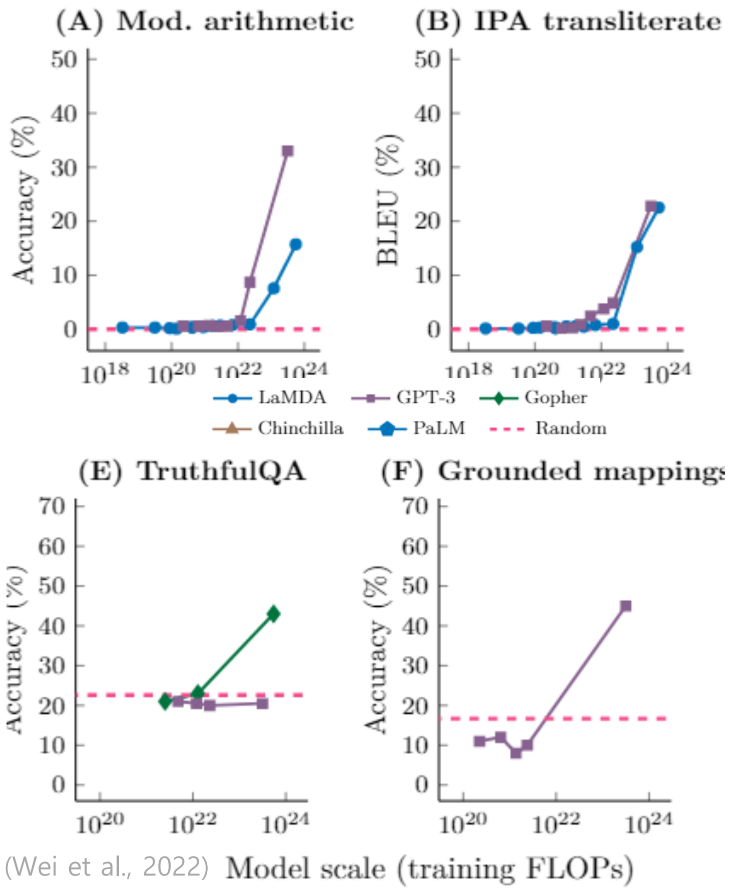


Figure 1: Exponential growth of number of parameters in DL models

https://towardsdatascience.com/the-rise-of-cognitive-ai-a29d2b724ccc

Large computation required for emergent abilities

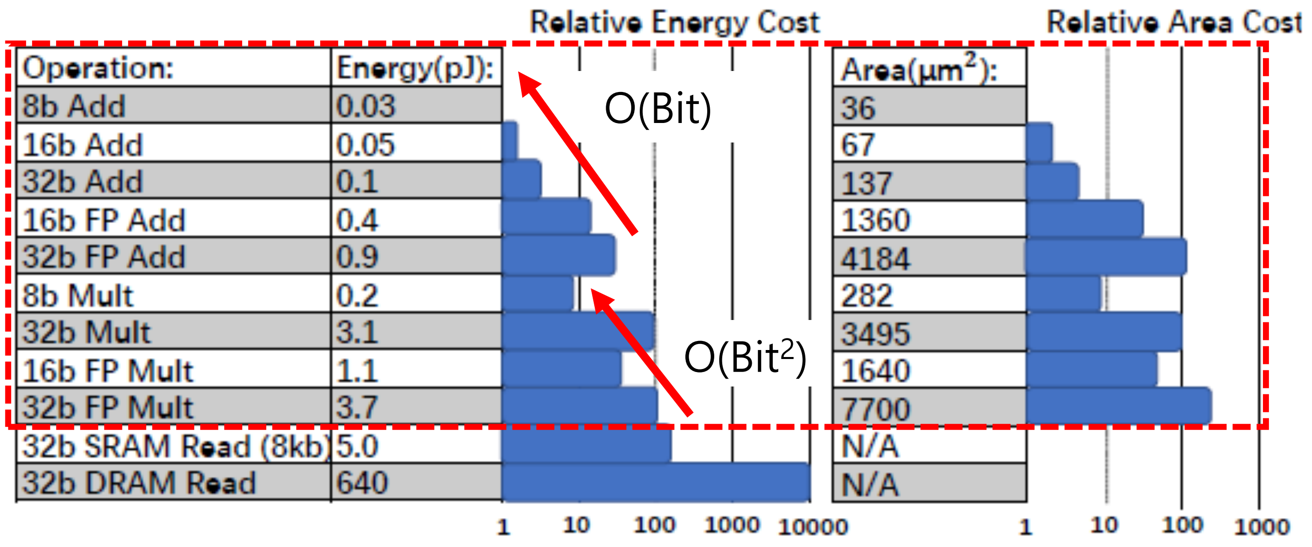


(Wei et al., 2022)

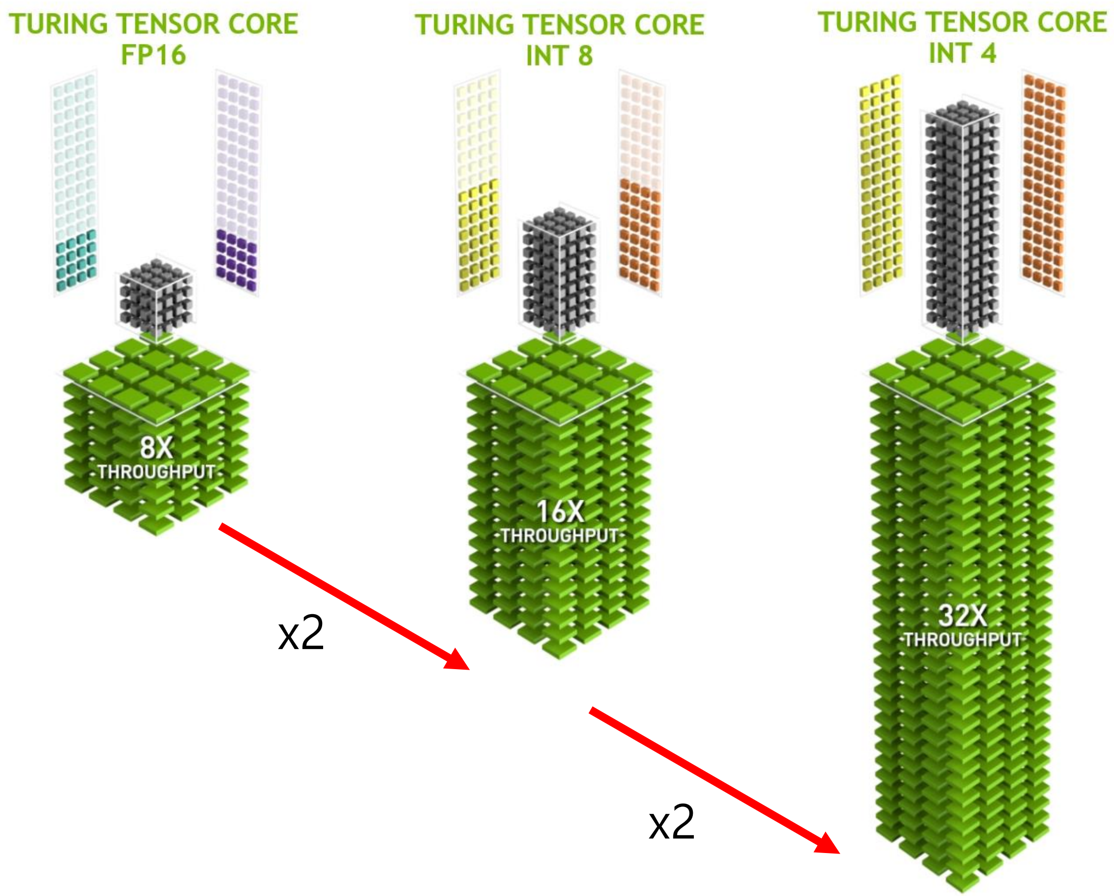
Operation cost + Environmental impact

Reduced-Precision Quantization to the Rescue

- Energy/Memory savings on addition and multiplication ➔ Potential for efficient inference
  - Productized in recent deep learning accelerators (Ex. NVIDIA Tensor Core)



Operation:	Energy(pJ):	Area(μm^2):
8b Add	0.03	36
16b Add	0.05	67
32b Add	0.1	137
16b FP Add	0.4	1360
32b FP Add	0.9	4184
8b Mult	0.2	282
32b Mult	3.1	3495
16b FP Mult	1.1	1640
32b FP Mult	3.7	7700
32b SRAM Read (8kb)	5.0	N/A
32b DRAM Read	640	N/A



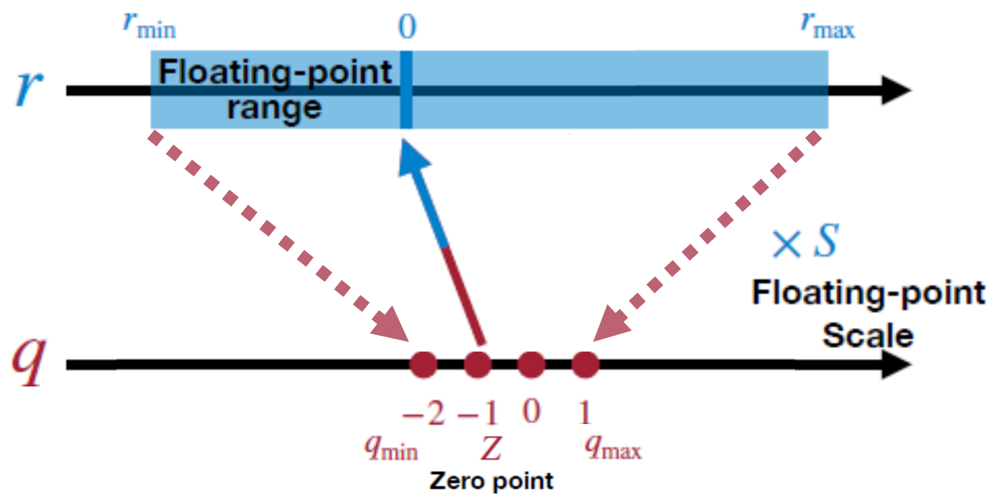
- Most computation of DNN: Multiply-Accumulate (MAC)
- Reduced-precision MAC simplifies compute unit
  - Addition: Savings proportional to bit reduction
  - Multiplication: Savings proportional to (bit reduction)^2

Courtesy of part of slides: NVIDIA, Horowitz (2014)

Reduced-Precision Quantization – Basic

- Represent real values with finite number of states encoded by reduced bit-precision

Asymmetric Linear Quantization



weights (32-bit float)					quantized weights (2-bit signed int)				zero point (2-bit signed int)	scale (32-bit float)

$$\begin{bmatrix} 2.09 & -0.98 & 1.48 & 0.09 \\ 0.05 & -0.14 & -1.08 & 2.12 \\ -0.91 & 1.92 & 0 & -1.03 \\ 1.87 & 0 & 1.53 & 1.49 \end{bmatrix} \Rightarrow \left(\begin{bmatrix} 1 & -2 & 0 & -1 \\ -1 & -1 & -2 & 1 \\ -2 & 1 & -1 & -2 \\ 1 & -1 & 0 & 0 \end{bmatrix} - -1 \right) \times 1.07$$

$$r = (q - Z) \times S$$

Floating-point | Integer | Integer | Floating-point

Integer:
- quantization parameter
- allow real number $r=0$ be exactly representable by a quantized integer Z

Floating-point:
- quantization parameter

Binary	Decimal
01	1
00	0
11	-1
10	-2

$$S = \frac{r_{max} - r_{min}}{q_{max} - q_{min}} = \frac{2.12 - (-1.08)}{1 - (-2)} = 1.07$$

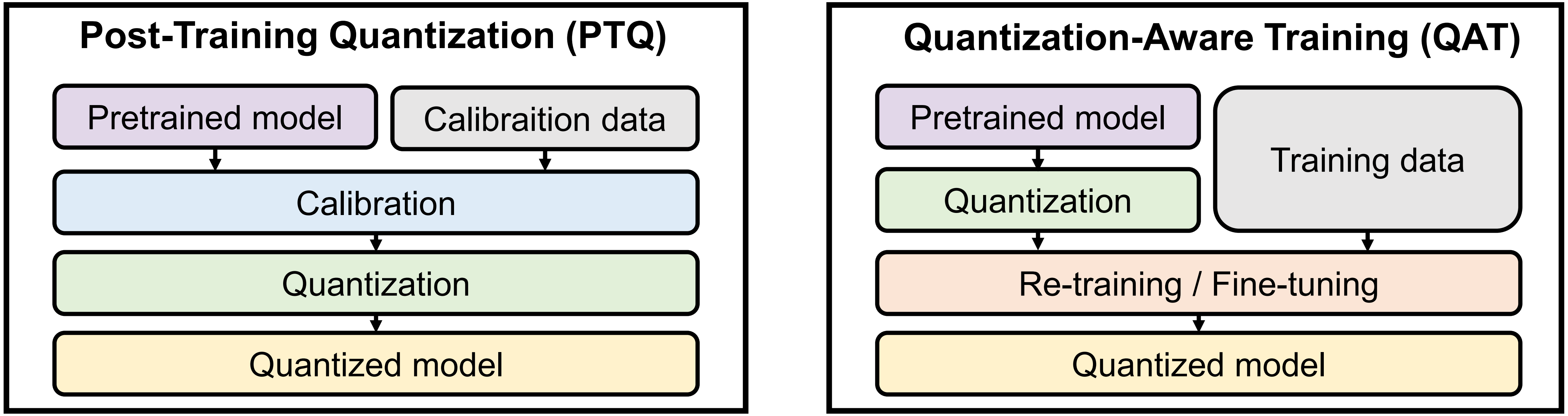
$$Z = q_{min} - \frac{r_{min}}{S} = \text{round}\left(-2 - \frac{-1.08}{1.07}\right) = -1$$

Courtesy of part of slides: MIT 6.5940 TinyML and Efficient Deep Learning Computing by Prof. H. Song

Quantization Techniques for Efficient LLM Inference

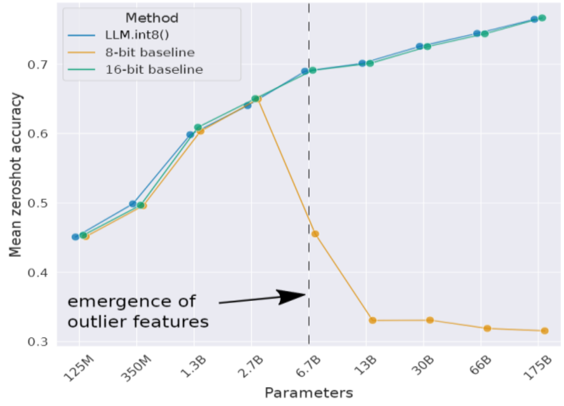
- Quantization represents real values with finite number of bits (in floating/fixed-point formats)

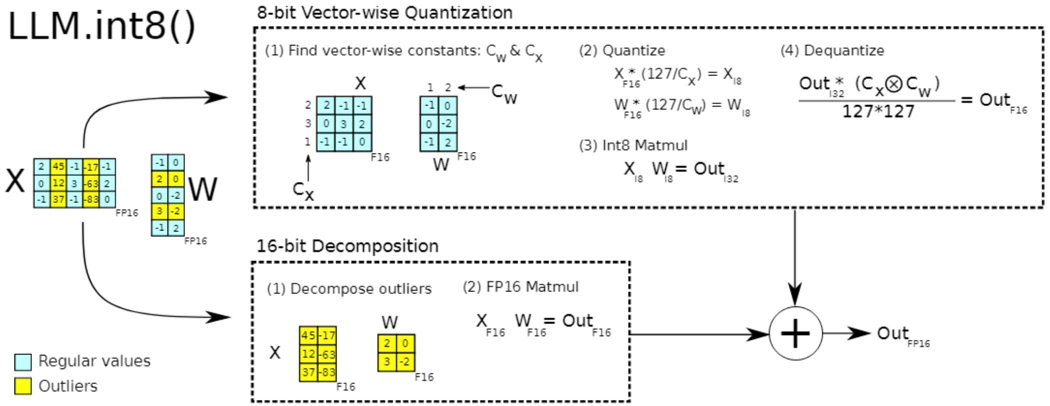
- Two types of DNN quantization: QAT and PTQ
  - Post-Training Quantization (**PTQ**) directly converts parameters and activations to reduced-precision
  - Quantization-Aware Training (**QAT**) exploits re-training to compensate quantization error



INT8 Activation & Weight PTQ for Efficient LLM Inference

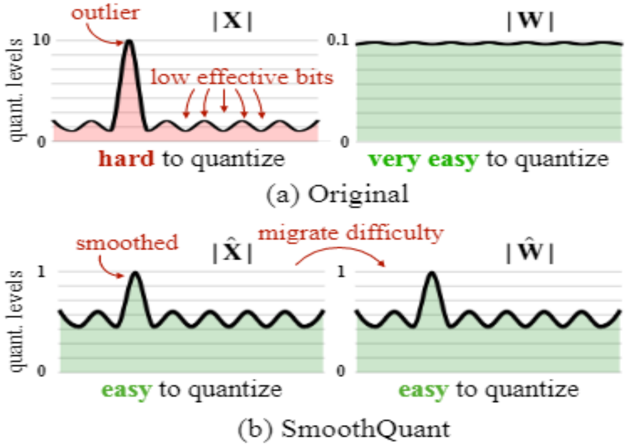
- **Challenges: How to handle large activation outliers observed in large LMs (>6.7B)**
  - **LLM.int8()**: Separate outlier columns (> α=6.0) and compute them in FP16
  - **SmoothQuant**: Scale weights (and descale activation) to flatten activation outliers



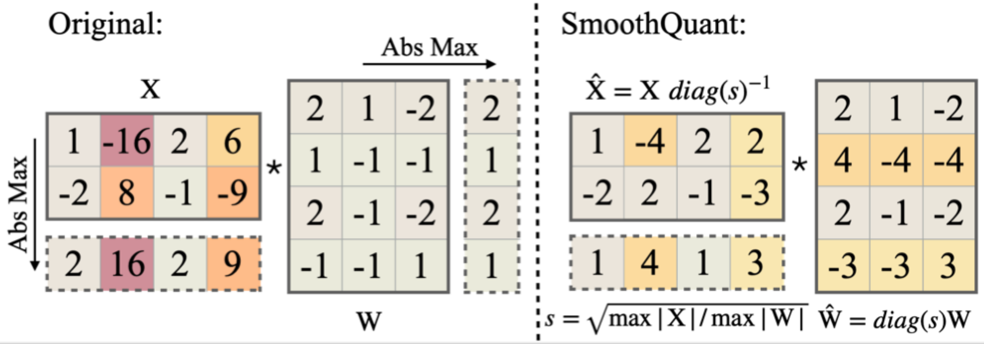


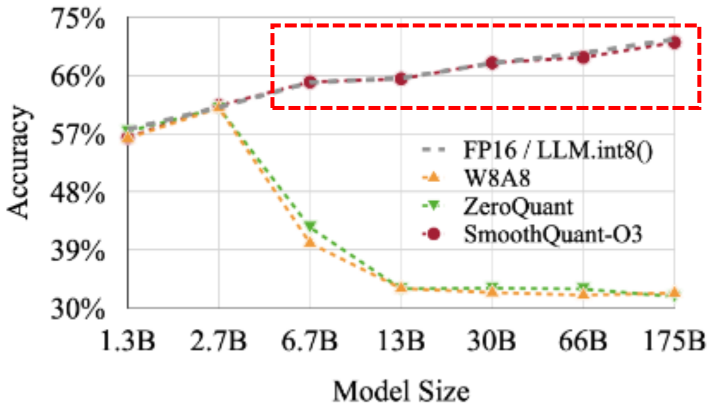
Parameters	125M	1.3B	2.7B	6.7B	13B
32-bit Float	25.65	15.91	14.43	13.30	12.45
Int8 absmax	87.76	16.55	15.11	14.59	19.08
Int8 zeropoint	56.66	16.24	14.76	13.49	13.94
Int8 absmax row-wise	30.93	17.08	15.24	14.13	16.49
Int8 absmax vector-wise	35.84	16.82	14.98	14.13	16.48
Int8 zeropoint vector-wise	25.72	15.94	14.36	13.38	13.47
Int8 absmax row-wise + decomposition	30.76	16.19	14.65	13.25	12.46
Absmax LLM.int8() (vector-wise + decomp)	25.83	15.93	14.44	**13.24**	**12.45**
Zeropoint LLM.int8() (vector-wise + decomp)	**25.69**	**15.92**	**14.43**	**13.24**	**12.45**

LLM.int8() (Dettemers et al., NeurIPS 2022)



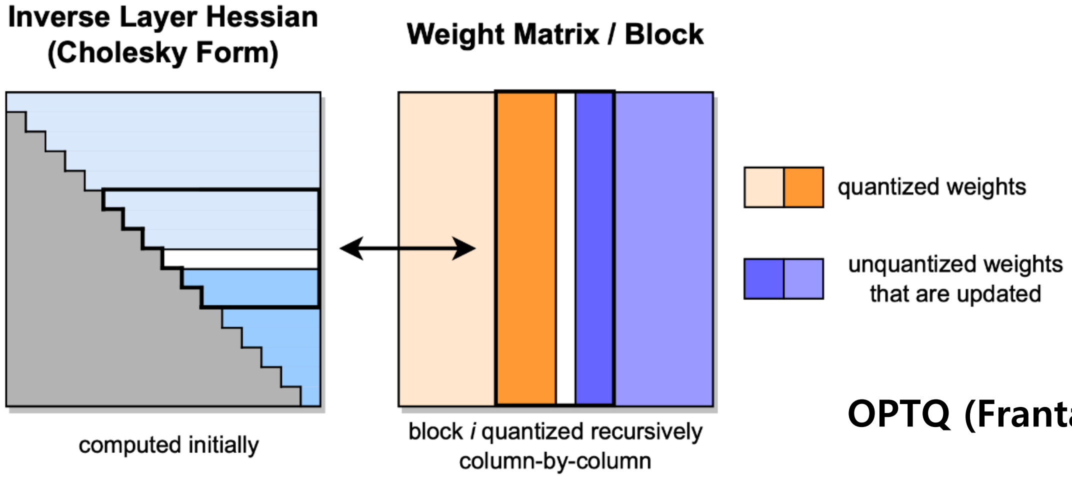
SmoothQuant (Xiao et al., ICML 2023)





Weight-Only PTQ for Memory-Efficient LLM Inference (1/2)

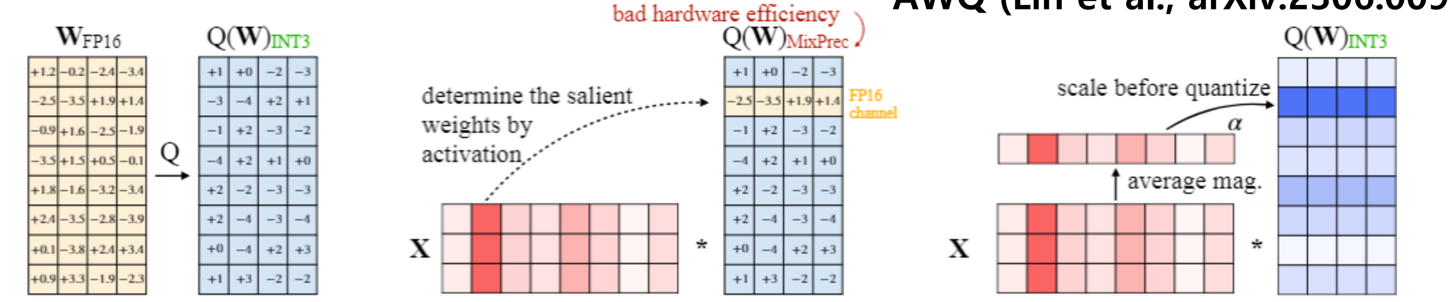
- **Manipulate activation/weight** to minimize impact of weight quantization
  - **OPTQ**: Update weights to minimize Hessian approximated quantization errors
  - **AWQ**: Scale salient weights (that are affected by activation most) to minimize output errors



OPT	Bits	125M	350M	1.3B	2.7B	6.7B	13B	30B	66B	175B
full	16	27.65	22.00	14.63	12.47	10.86	10.13	9.56	9.34	8.34
RTN	4	37.28	25.94	48.17	16.92	12.10	11.32	10.98	110	10.54
OPTQ	4	**31.12**	**24.24**	**15.47**	**12.87**	**11.39**	**10.31**	**9.63**	**9.55**	**8.37**
RTN	3	1.3e3	64.57	1.3e4	1.6e4	5.8e3	3.4e3	1.6e3	6.1e3	7.3e3
OPTQ	3	**53.85**	**33.79**	**20.97**	**16.88**	**14.86**	**11.61**	**10.27**	**14.16**	**8.68**

OPTQ (Frantar et al., ICLR 2023) Table 3: OPT perplexity results on WikiText2.

AWQ (Lin et al., arXiv:2306.00978)



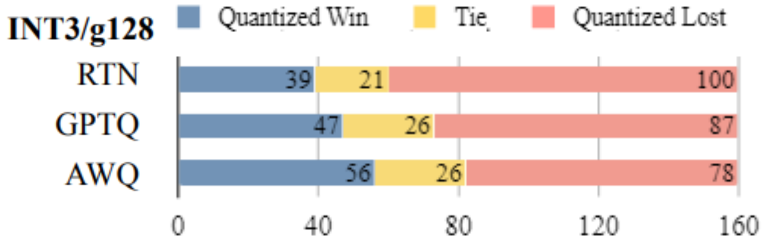
(a) RTN quantization (**PPL** 43.2) (b) Keep 1% salient weights in FP16 (**PPL** 13.0) (c) Scale the weights before quantization (**PPL** 13.0)

OPT / PPL↓		125M	1.3B	2.7B	6.7B	13B	30B	66B
FP16	-	31.95	16.41	14.32	12.29	11.5	10.67	10.09
INT3 g128	RTN	58.49	206.54	595.28	43.16	45.37	28.84	423.39
	GPTQ	41.93	**18.53**	15.79	13.13	**12.01**	**11.00**	11.48
	AWQ	**41.10**	**18.53**	**15.62**	**12.99**	12.03	11.03	**10.46**
INT4 g128	RTN	35.51	17.70	15.12	13.02	11.89	11.00	10.44
	GPTQ	34.23	16.92	14.69	12.51	**11.60**	**10.74**	10.24
	AWQ	**33.96**	**16.85**	**14.61**	**12.44**	**11.60**	10.75	**10.16**

Weight-Only PTQ for Memory-Efficient LLM Inference (2/2)

- Accuracy preserved by "good" quantization techniques

LLaMA-7B		MMLU (5-shot) ↑					Common Sense QA (0-shot) ↑				
		Hums.	STEM	Social	Other	Avg.	PIQA	Hella.	Wino.	ARC-e	Avg.
FP16	-	39.17%	32.32%	42.72%	42.56%	38.41%	78.35%	56.44%	67.09%	67.30%	67.30%
INT3 g128	RTN	31.37%	31.10%	36.04%	36.49%	33.43%	75.84%	53.10%	63.22%	66.04%	64.55%
	GPTQ	29.29%	29.04%	33.03%	31.65%	30.53%	70.89%	46.77%	60.93%	60.06%	59.66%
	GPTQ-R	33.98%	30.71%	37.78%	36.49%	34.26%	77.31%	53.81%	67.56%	63.72%	65.60%
	AWQ	35.15%	31.61%	39.27%	37.75%	**35.43%**	76.66%	53.63%	66.14%	65.70%	**65.53%**
INT4 g128	RTN	36.15%	33.03%	41.41%	41.21%	37.37%	77.86%	55.81%	65.59%	66.25%	66.38%
	GPTQ	35.55%	30.95%	39.29%	38.12%	35.39%	77.20%	53.98%	65.67%	61.62%	64.62%
	GPTQ-R	37.28%	31.36%	40.23%	40.77%	36.72%	78.45%	56.00%	66.85%	66.88%	67.05%
	AWQ	38.32%	32.00%	41.38%	42.07%	**37.71%**	78.07%	55.76%	65.82%	66.84%	**66.62%**



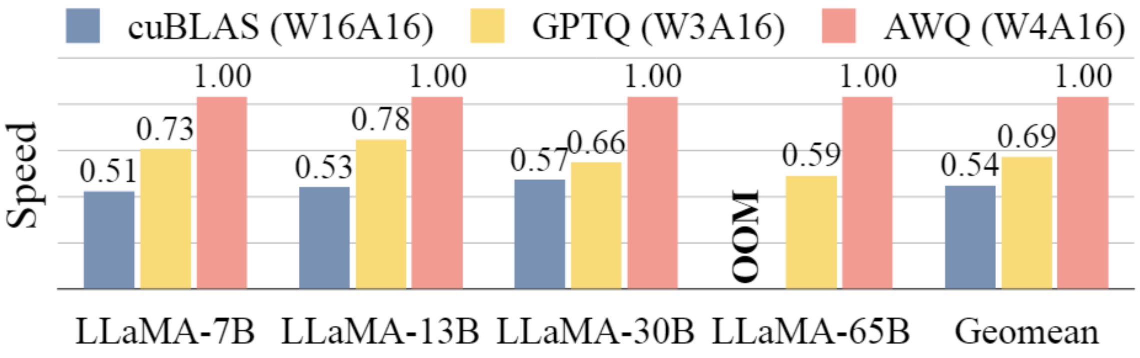
(b) Vicuna-13B

W4-RTN: A model airplane flying in the sky.

W4-AWQ: Two toy airplanes sit on a grass field.

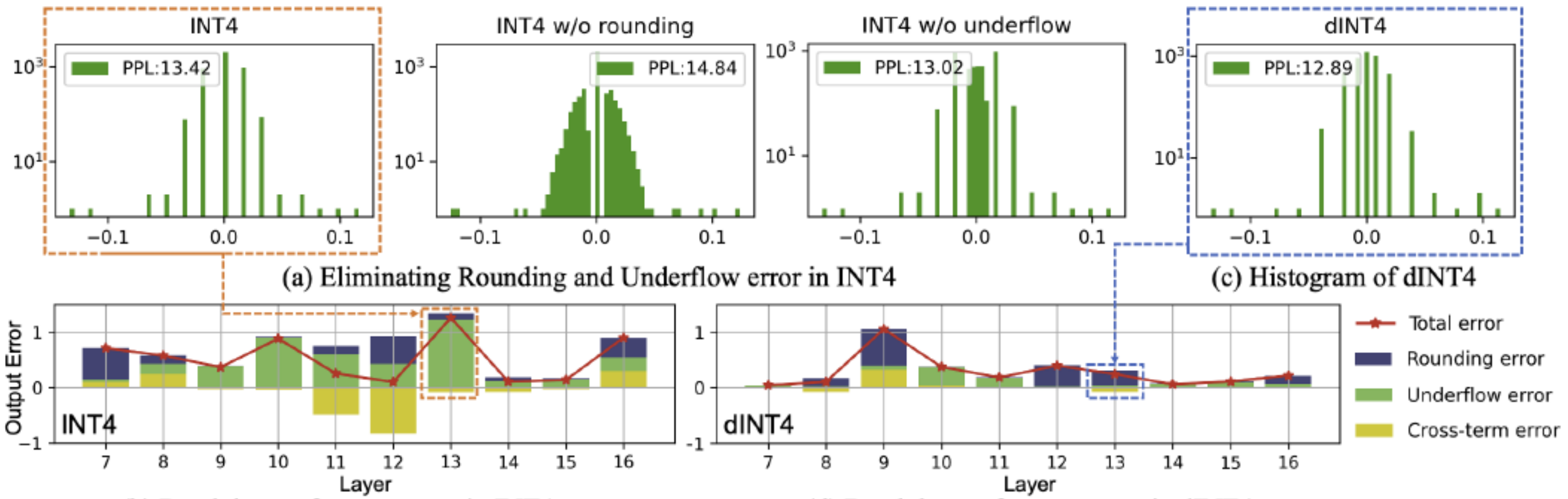
AWQ (Lin et al., arXiv:2306.00978)

- Performance improvements
  - Develop custom CUDA kernel
  - (Measured on A100 GPU)



PTQ for More Efficient LLM Inference (EMNLP 2023)

- Quantize weight (4-bit) AND activation (8-bit) together to enhance efficiency of LLMs
- Integer with denormal representation to overcome PTQ underflow for LLMs
  - INT4 weight leads to high quantization errors due to underflow
    - Small magnitude weights play important role as they are multiplied with activation
  - Suggest a new number format to represent small values using a denormal state (*dINT*)

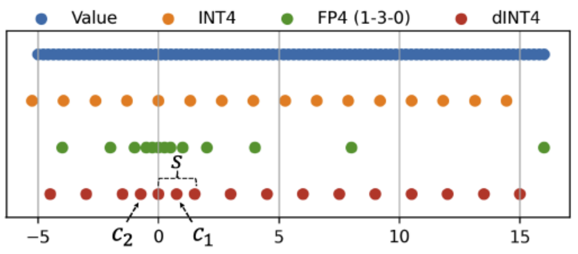


(a) Eliminating Rounding and Underflow error in INT4

(c) Histogram of dINT4

(b) Breakdown of ouput errors in INT4

(d) Breakdown of ouput errors in dINT4



$$\mathbf{E}[(\mathbf{WX} - (\mathbf{W} + \Delta_u + \Delta_r)\mathbf{X})^2]$$
$$= \mathbf{E}[(\Delta_u\mathbf{X})^2] + \mathbf{E}[(\Delta_r\mathbf{X})^2] + \mathbf{E}[2(\Delta_u\mathbf{X}\Delta_r\mathbf{X})]$$

Δ_u : Underflow error

Δ_r : Rounding error

Weight Scaling	OPTQ	A-bits	W/V-bits	LLaMa Family 7B	13B	30B
Baseline				35.20	47.15	58.50
-	-	INT8	INT4	28.05	40.82	48.40
	✓			27.05	42.95	53.30
SQ	✓			29.32	43.12	52.83
AQAS	✓			30.81	44.23	53.67
	✓		dINT4	31.00	44.73	55.50

Prior QATs for Transformer Decoder Models

- Successful W4-A4 quantization for **Encoder-only**/**Encoder-Decoder** models
- Noticeable accuracy degradation for **Decoder-only** models → Need improvements!

(Wu, et al., ICML 2023)

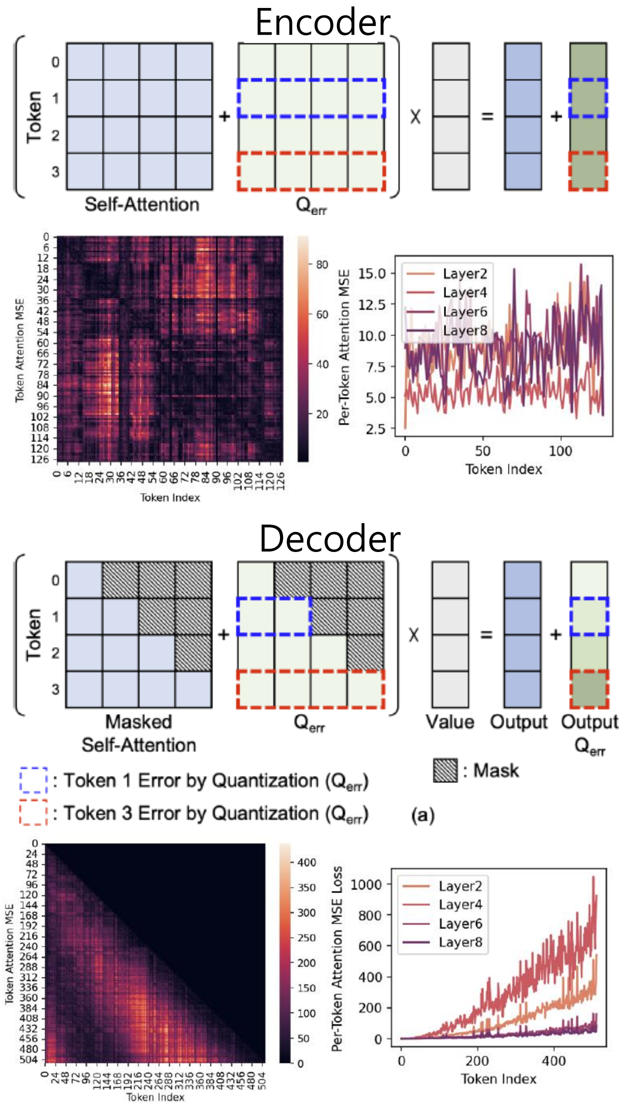
Table 1: The best quality for BERT/BART/GPT-type models (two sizes) over the validation datasets, respectively with metric Accuracy (Acc., higher is better), Rouge Lsum (RLsum, higher is better), and perplexity (PPL, lower is better).

Models	BERT-base (110M)		BART-base (140M)		GPT2-base (117M)		
Tasks	MNLI-m/mm	QQP	CNNDailyMail	XSUM	PTB	WIKI-2	WIKI-103
Metrics	Acc/Acc	F1/Acc	R1/R2/RLsum	R1/R2/RL	Perplexity	Perplexity	Perplexity
FP32 (teacher)	84.20/84.67	87.83/90.95	45.62/22.85/42.87	42.18/19.44/34.36	19.31	21.02	17.46
W4A4 (symmetric)	84.31/84.48	88.11/91.14	44.63/21.42/41.92	41.54/18.61/33.69	22.17	27.28	21.75
W4A4 (asymmetric)	84.29/84.65	88.17/91.19	44.83/21.67/42.08	41.53/18.56/33.62	21.72	25.99	21.54
Models	BERT-large (345M)		BART-large (406M)		GPT2-medium (355M)		
Tasks	MNLI-m/mm	QQP	CNNDailyMail	XSUM	PTB	WIKI-2	WIKI-103
Metrics	Acc/Acc	F1/Acc	R1/R2/RLsum	R1/R2/RL	Perplexity	Perplexity	Perplexity
FP32 (teacher)	86.65/85.91	88.08/91.07	44.82/21.67/41.80	45.42/22.37/37.29	15.92	15.92	12.75
W4A4 (symmetric)	86.25/86.20	88.30/91.17	45.12/21.73/42.31	44.39/21.28/36.33	17.69	19.51	14.57
W4A4 (asymmetric)	86.49/86.28	88.35/91.24	45.20/21.85/42.40	44.91/21.74/36.79	17.32	18.74	14.23

Encoder-only **Encoder-Decoder** **Decoder-only**

QAT for Sub-4-bit LLM Inference (NeurIPS 2023)

- Investigated root causes of accuracy degradation for quantized decoder models: **Error accumulation over tokens**
- Proposed a novel KD method with token-based loss scaling to alleviate error accumulation and improve accuracy of **fine-tuned quantized LLMs**
- Improved accuracy of x8 or x16 compressed LLMs



Precision	Quantization Method	Optimization Method	GPT-2 0.1B	GPT-2 0.3B	GPT-2 0.8B	GPT-2 1.5B	OPT 0.1B	OPT 1.3B	OPT 2.7B	OPT 6.7B
FP32 baseline			20.91	18.21	15.20	14.26	18.17	13.75	11.43	10.21
W4A32	PTQ	OPTQ [9]	22.41	19.35	17.26	15.86	19.75	14.30	11.82	11.73
	QAT	Logit [20]	20.98	18.54	16.79	15.42	17.60	**13.73**	11.82	11.20
		Logit+GT	21.51	18.58	15.49	14.89	19.63	15.03	12.58	11.78
		TSLD	**19.95**	**17.53**	**15.32**	**14.50**	**17.45**	13.90	**11.59**	**11.00**
W2A32	QAT	L2L+Logit [29]	23.79	21.21	-	-	20.47	-	-	-
		Logit [20]	22.84	19.87	16.46	15.27	18.86	14.80	12.26	11.33
		Logit+GT	23.80	20.20	17.77	16.52	21.62	16.41	13.20	12.41
		TSLD	**21.74**	**18.57**	**16.14**	**15.02**	**18.58**	**14.60**	**11.97**	**11.17**

Table 1: Perplexity of GPT-2 and OPT based on size and KD method

Summary

- Implementation of Large language models (LLMs) is cost-intensive, owing largely to their considerable demand for memory and computational power

- Efforts are increasing to advance quantization methods, aiming to optimize efficiency of LLM inference while maintaining model accuracy
  - Post training quantization (PTQ)
  - Quantization-aware training (QAT)

- Enhanced quantization techniques would unleash new opportunities for running large language models on edge devices

- References for our papers
  - Enhancing Computation Efficiency in Large Language Models through Weight and Activation Quantization (EMNLP 2023)
  - Token-Scaled Logit Distillation for Ternary Weight Generative Language Models (NeurIPS 2023)

Thank You!

Any Questions?

TINY ML ASIA
Seoul, South Korea
November 16, 2023

Copyright Notice