Lightweight Yet Powerful: Transforming Large AI Models with Model Compression Techniques for On-Device AI - Focus on Sparse Matrix Representation and Quantization

Gi-Ho Park[1], Chiwon Han[1], Sungyuny Bae[1], Jaeung Lee[2], Keunho Byeon[2], Juhee Choi[3], Jin Tae Kwak[2], Hyesoon Kim[4]

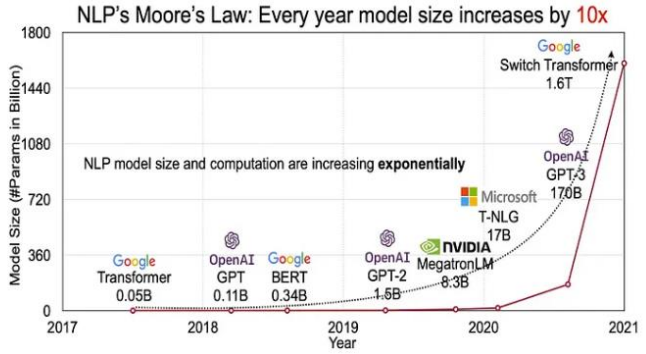
[1]Sejong University, [2]Korea University,
[3]SangMyung University, [4]Georgia Institute of Technology

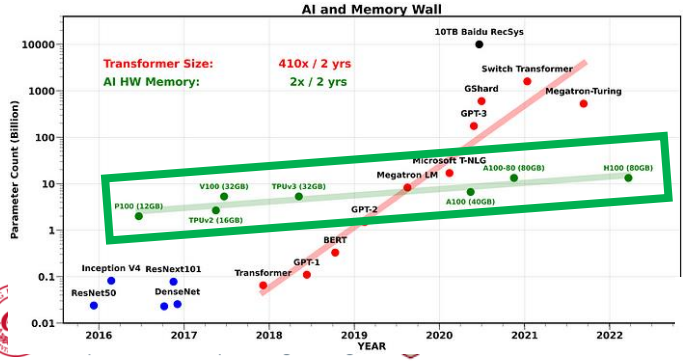
Outline

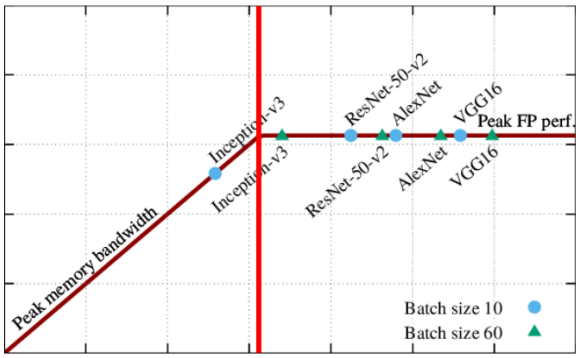
- Large AI Model and On-Device AI
- Model Compression Techniques
- Pruning
- Quantization
- Issues of Model Compression Techniques
- Sparse Matrix Representation, Merged Pruning/Quantization for Large AI Models and On-Device AI Accelerators
- Summary

AI Model Size Growth and Memory Bottleneck

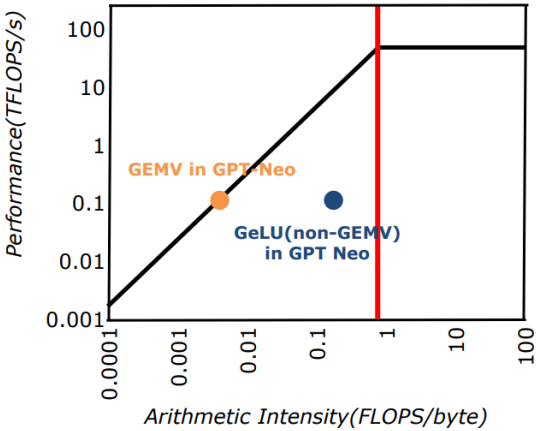
- **Increase in the Usage of Large Language Models (LLMs) and Transformer-based Models → Substantial increase in Model Size (number of parameters), Computational Demands, and Required Memory Usage**
- **Growth in Model Size Beyond the Level of Improvements in Computational Performance and Memory Bandwidth of Processing Units → Growing Demand for Model Lightweighting Techniques**
- **LLMs, unlike traditional CNN models, exhibit memory (bandwidth)-bound characteristics → Increasing Importance of Model Compression Techniques**







CNN roofline model



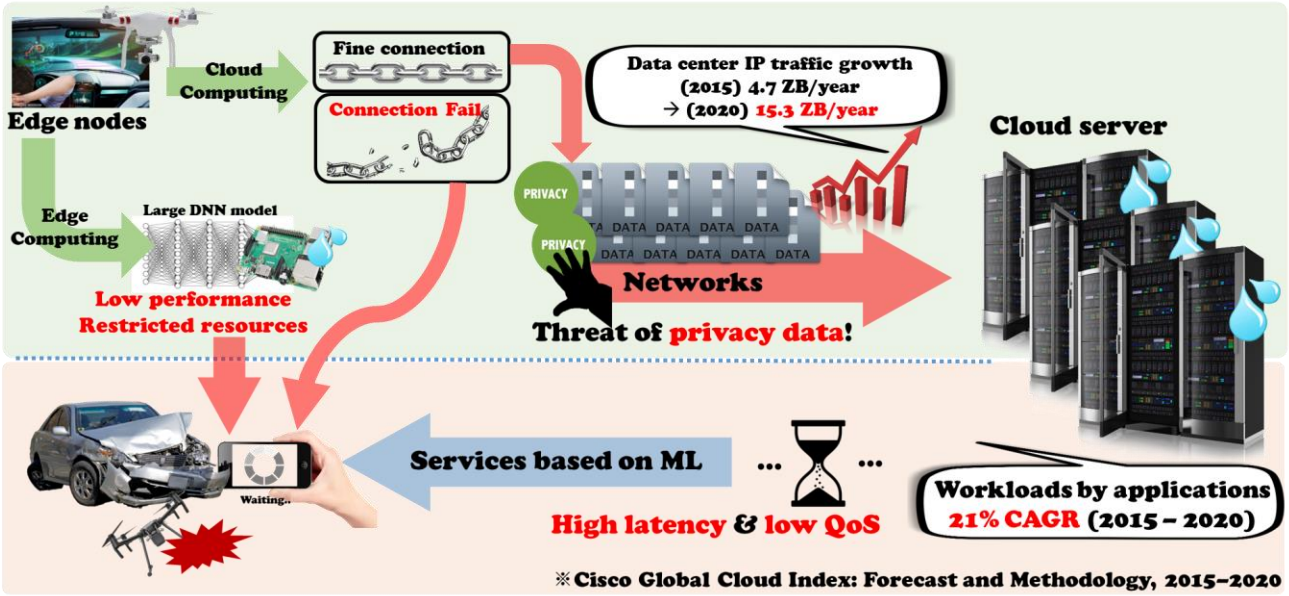
LLM (GPT) roofline model

Figure from https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

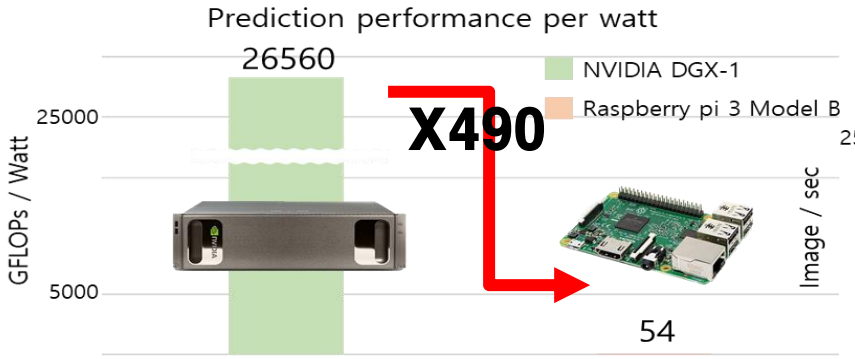
Cloud based / On-Device AI system

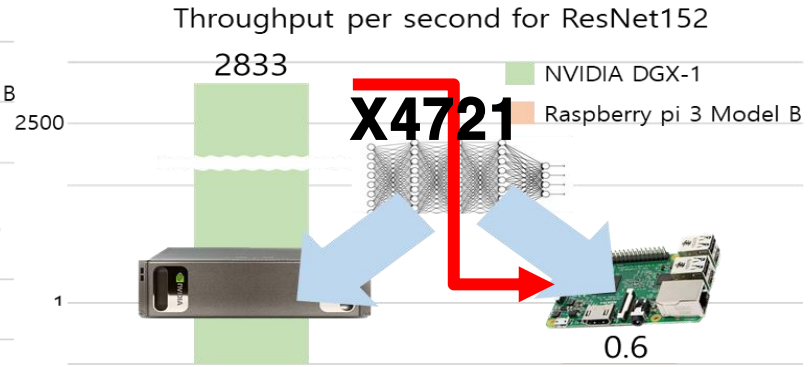
- Increase in the Number of Information Gathering Sensors and Devices → Rapid Growth in Data Volume
- Rising Demand for Stability and Accuracy in Intelligent Services → Increased Complexity of AI Models
- Increasing Privacy and Security Concerns → Demand for Local Processing of Collected Data

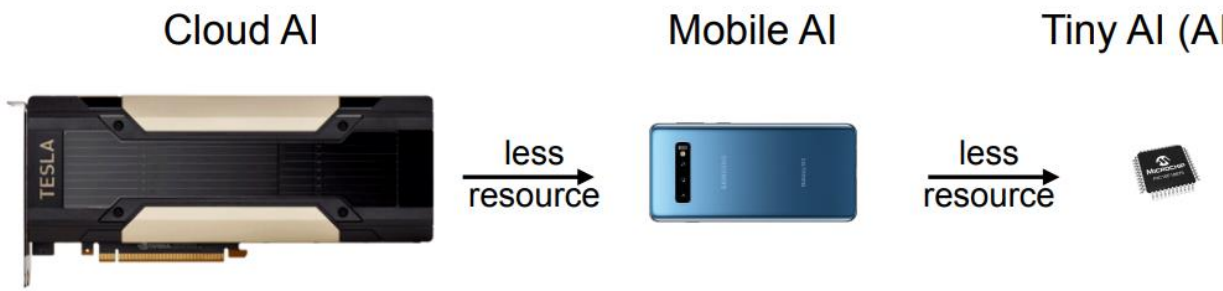
→ The increasing necessity for lightweight AI models and On-device AI accelerators to support services directly on edge devices



Edge devices with lower performance and memory capacity compared to cloud-based AI

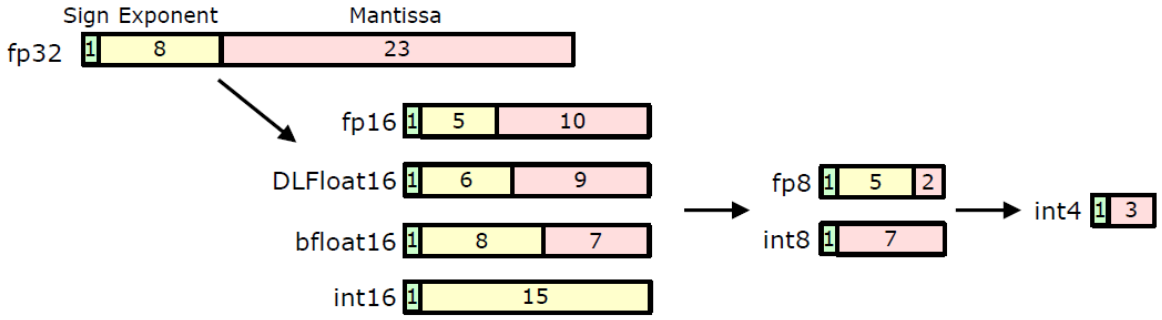




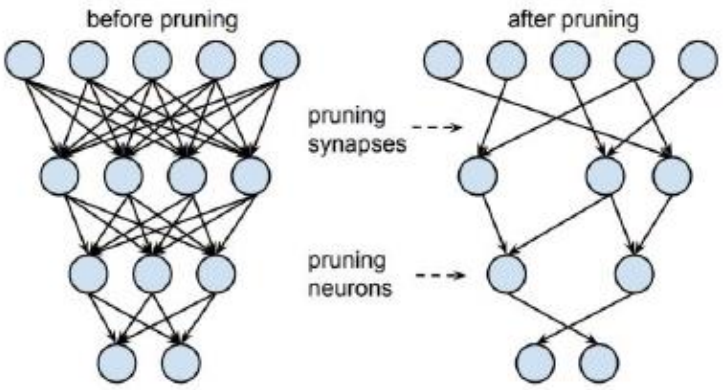


ICLR 2020: AutoML for TinyML with Once-for-All Network

Model Compression for Lightweight AI Models: Quantization, Pruning, Knowledge Distillation

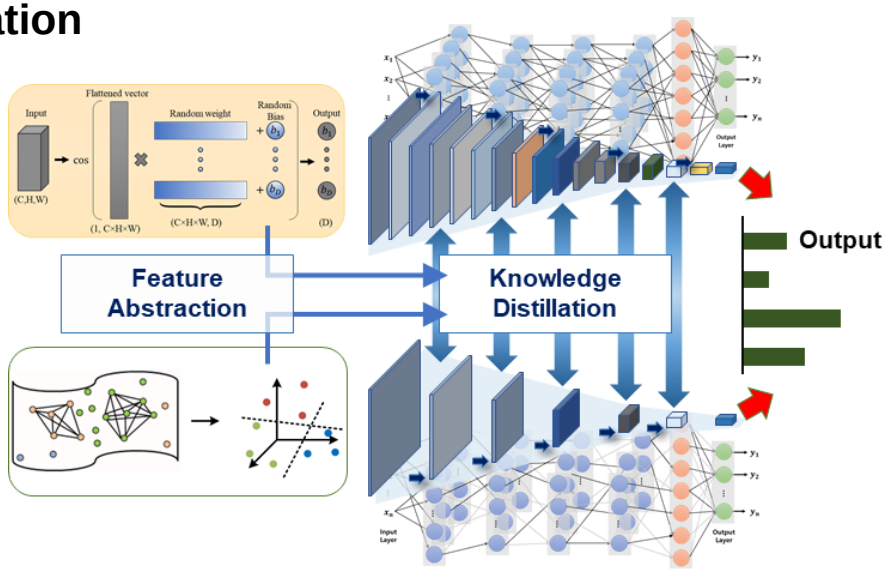


(a) Quantization



Han, NeurIPS '15

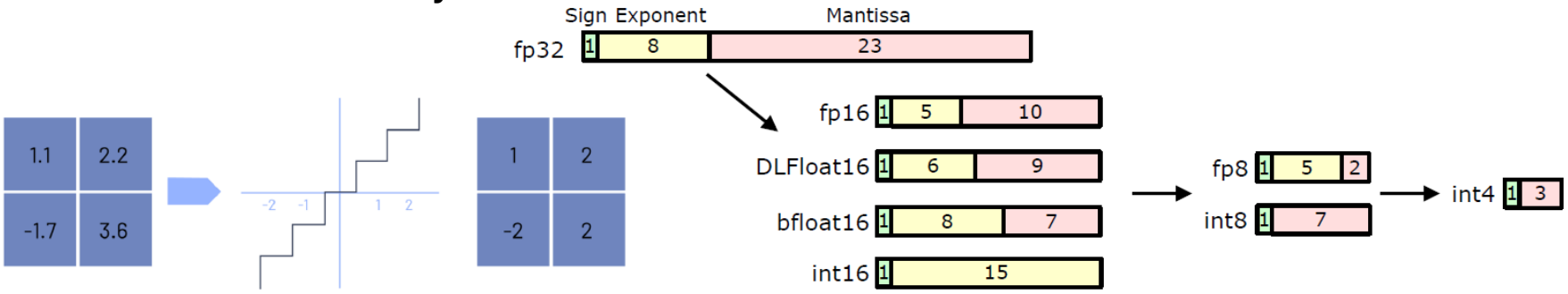
(b) Pruning



(c) Knowledge Distillation

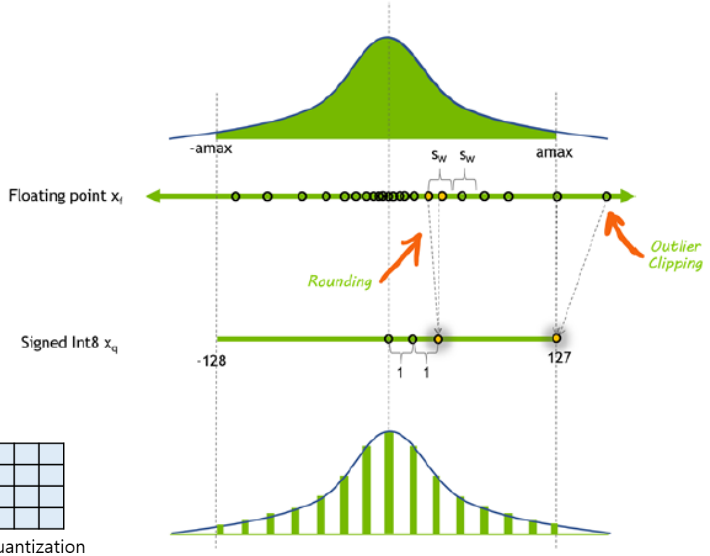
Quantization

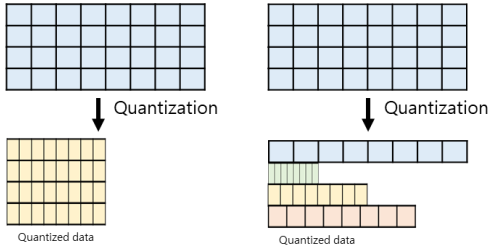
- **Deep learning models, especially large language models (LLMs), tend to be probabilistic and over-parameterized → allowing for the utilization of approximation such as lower precision data formats (e.g., FP64, 32 → FP16, FP8, INT8, INT4, etc.).**
- **Quantization has various advantages.**
  - **High-throughput by ease of computation and utilization of SIMD (Single Instruction Multiple Data) operation**
  - **Reduced memory traffic due to decreased bit demand per element.**
  - **Decreased on-chip storage requirements.**
  - **Reduced energy consumption resulting from data movement.**
- **Efficiency vs. Accuracy tradeoff**
  - **The trade-off between improved efficiency due to approximation and decreased model accuracy.**



Quantization design choices and issues

- Range (floating point, integer)
- Mapping
- Scaling granularity
- Post-training quantization (PTQ) vs quantization-aware training (QAT)
- PTQ for LLM: without retraining
- Accuracy drop with PTQ
- Outlier in LLM



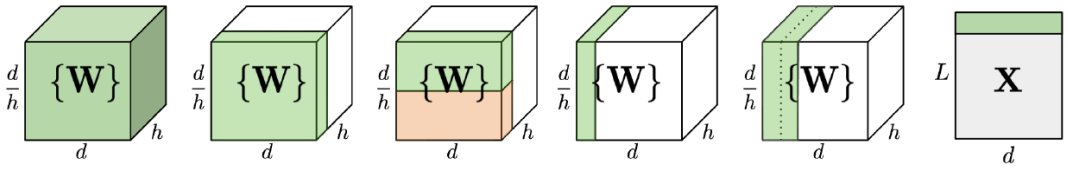


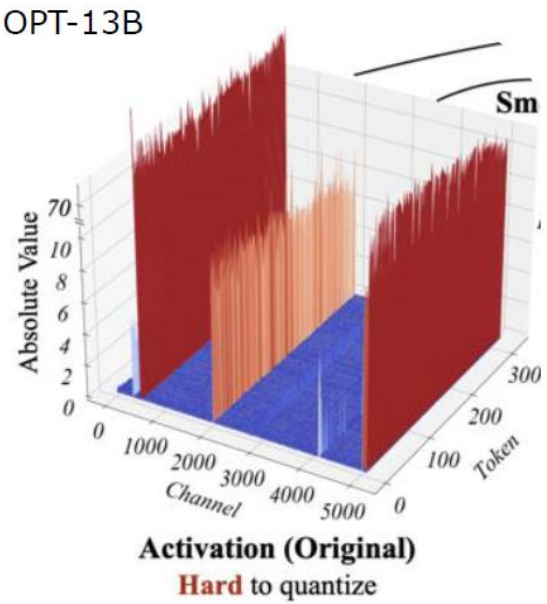
ISSCC 2024 - Forum 2.7: <Quantizing LLMs for Efficient Inference at the Edge>

ISSCC 2024 Short Course Marian Verhelst

NVIDIA white paper : Achieving FP32 Accuracy for INT8 Inference Using Quantization Aware Training with NVIDIA TensorRT

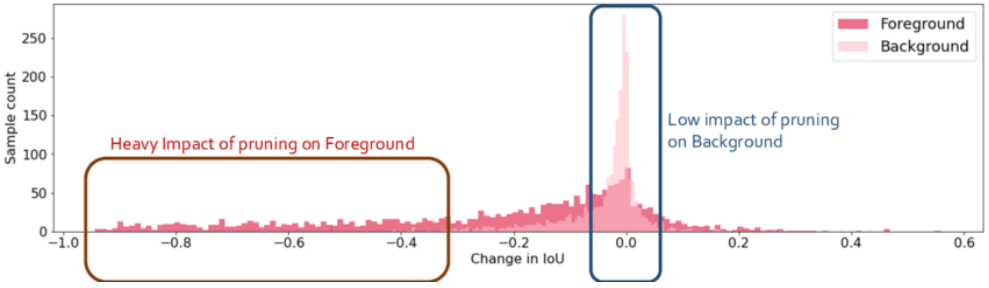
Accuracy trade off	--	++
Memory footprint	++	--
Compute complexity	++	--
Accuracy per Memory size	-	+



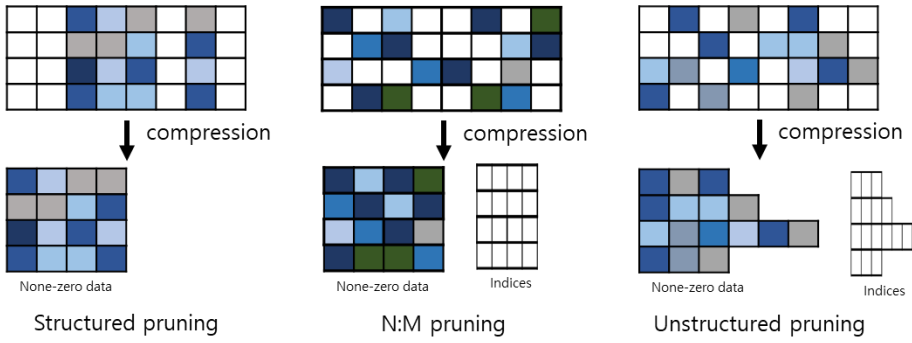


Pruning

- **Pruning, a lightweighting technique in deep learning, involves selecting less impactful values based on criteria like magnitude and zeroing them out. This process creates compressible sparse matrices, reducing memory usage and computational workload.**
- **Pruning techniques can be categorized based on the criteria used to select values to be zeroed out:**
  - **Unstructured pruning, Structured pruning, N:M pruning**
- **The selection method for zeroing out values entails a trade-off between accuracy and memory usage.**
  - **Structured and N:M pruning technique: select values to be zeroed out based on predefined rules or constraints. → Potentials for decreasing in accuracy.**
  - **Unstructured pruning: minimize accuracy degradation by zeroing out values without specific rules. → irregularity like variable data and index counts, reduced memory reference efficiency, decreased resource utilization, scheduling issues, and increased hardware complexity.**

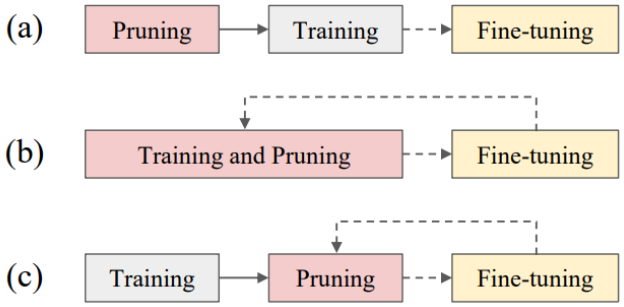


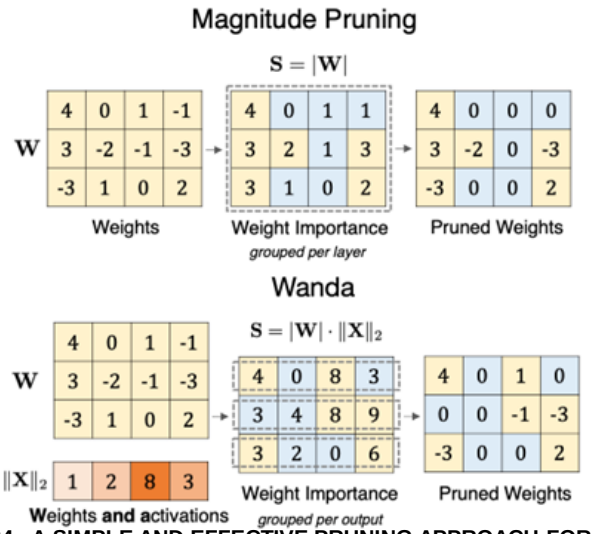
Attend Who is Weak: Pruning-assisted Medical Image Localization under Sophisticated and Implicit Imbalances

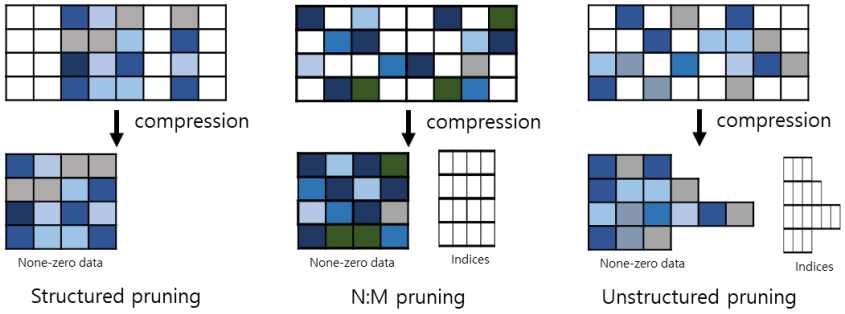


Pruning – design choices, issues

- **Pruning Timing**
  - **Pre-training pruning, during-training pruning, post-training pruning**
- **Pruning methods**
  - **magnitude-based, loss-based, regularization**
  - **Iterative, one-shot pruning**
  - **Structured pruning, unstructured pruning**
- **Indexing of pruned sparse matrices**
  - **CSR (Compressed Sparse Row)**
  - **RLE (Run Length Encoding)**





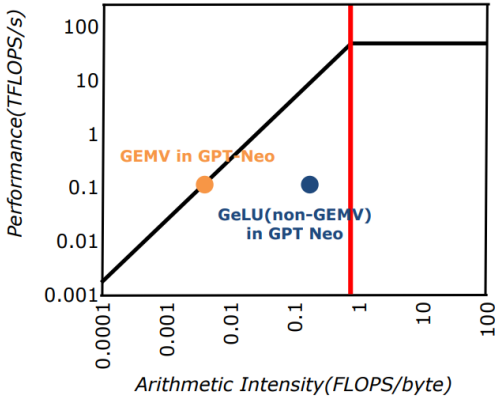


	Structured pruning	N:M pruning	Unstructured pruning
Accuracy trade off	--	+	++
Memory footprint	++	+	--
Compute complexity	++	+	--
Accuracy per Memory size	-	+	+

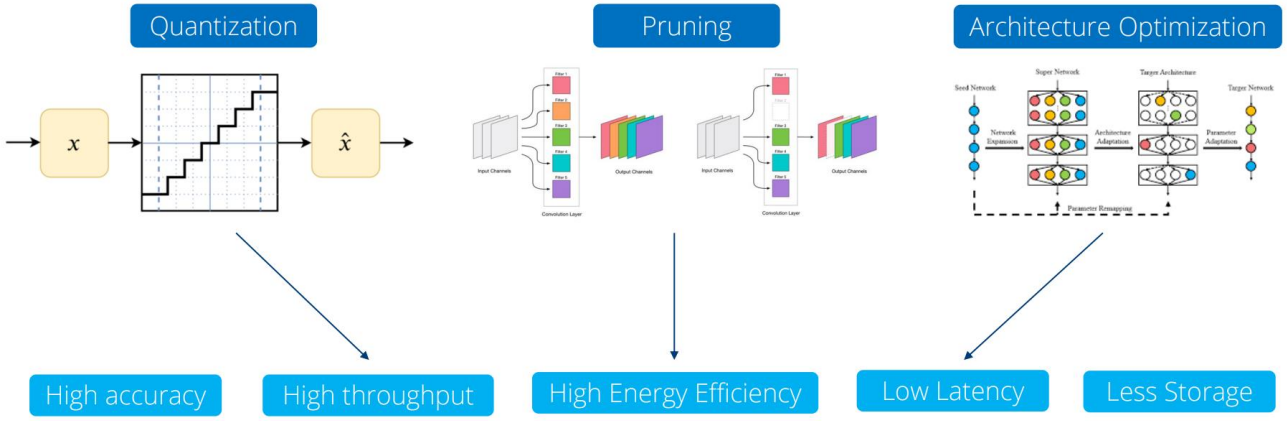
ICLR 2024 : A SIMPLE AND EFFECTIVE PRUNING APPROACH FOR LARGE LANGUAGE MODELS

The Necessity of Quantization/Pruning

- **In the case of LLMs, unlike traditional CNNs, they exhibit a memory-bandwidth-bound nature.**
- **Quantization and pruning techniques are crucial for lightweighting LLM models with such characteristics. These techniques aim to reduce model size, computational demands, and energy consumption for operations and memory references.**
- **Decreasing data precision through quantization and reducing the number of processed data through pruning can yield different effects in hardware design.**



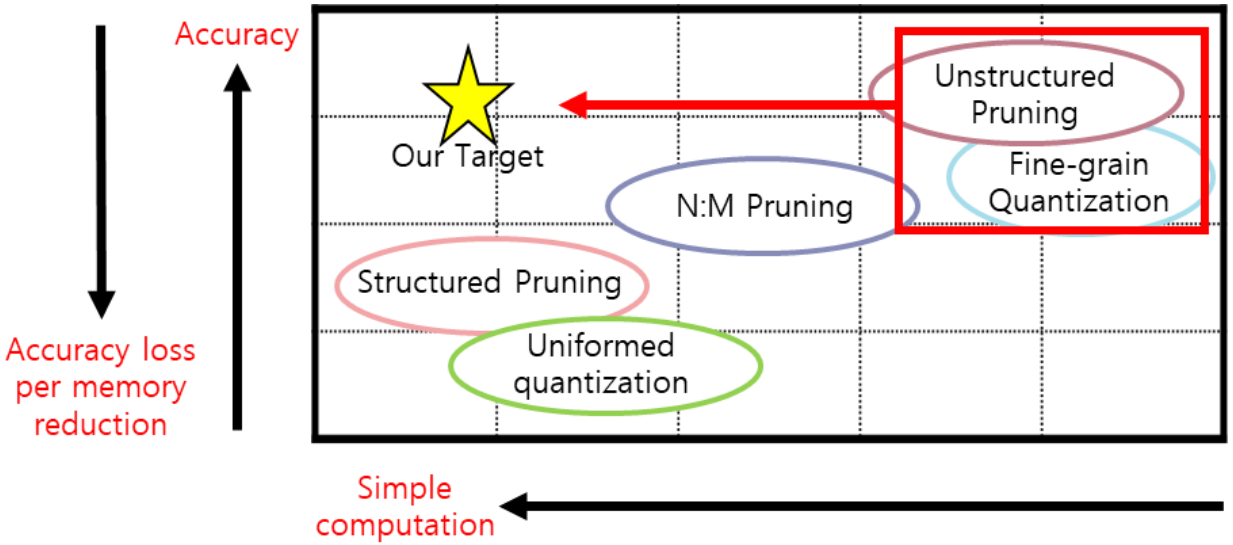
LLM (GPT) roofline model



ISSCC 2024 - Forum 2.7: <Quantizing LLMs for Efficient Inference at the Edge>

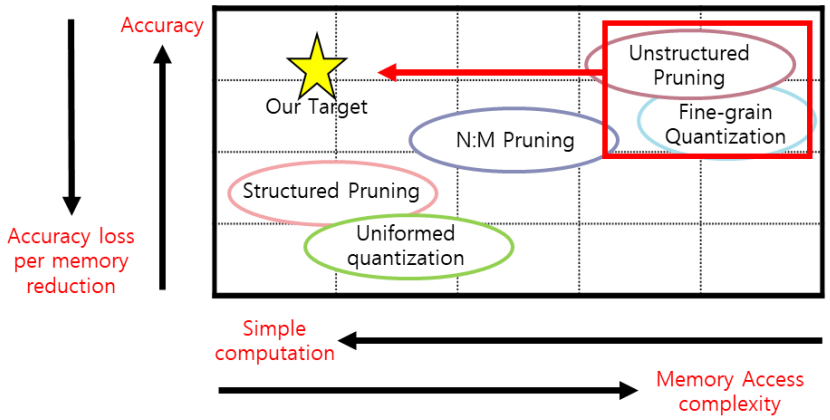
The Trade-offs in Quantization and Pruning.

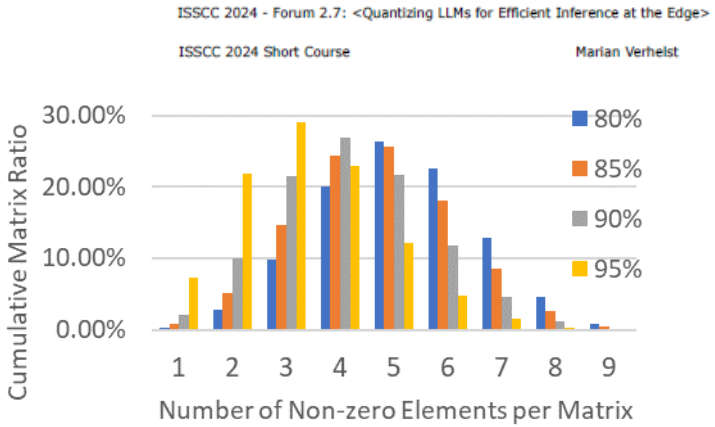
- **There are trade-offs such as model size, required computation, memory bandwidth, accuracy, and computation/memory access complexity.**
  - **Techniques like quantization and pruning offer advantages such as reducing model size, required computation, and memory bandwidth. However, they inherently lead to a loss of model accuracy.**
  - **To mitigate accuracy loss from quantization and pruning, more sophisticate techniques such as fine-grain quantization and unstructured pruning are utilized. However, these methods may lead to issues such as increased computational and memory access complexity, as well as hardware complexity.**
  - **In the case of LLMs, Post-Training Quantization (PTQ) is a practical method.**



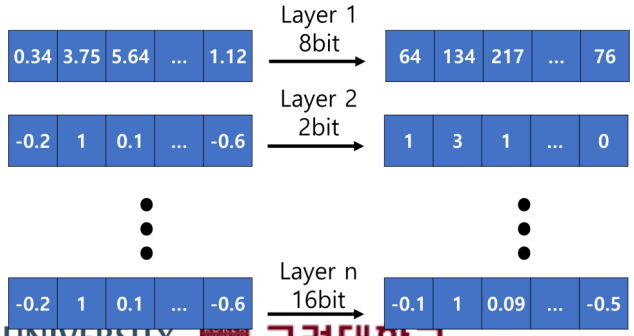
Challenges for Applying Fine-Grained Quantization and Unstructured Pruning: Irregularity

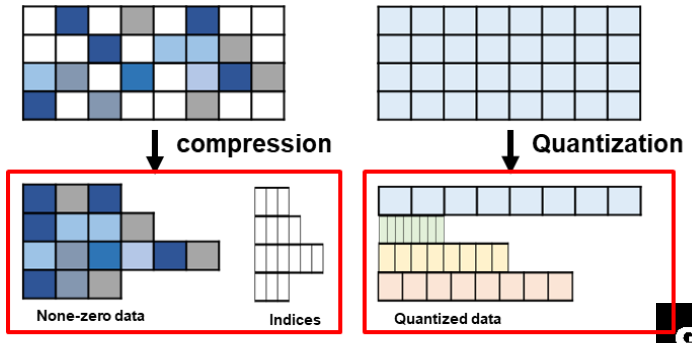
- **Performing fine-grain quantization and unstructured pruning:**
  - **Offers significant advantages in terms of accuracy.**
  - **However, the irregularity and complexity of data and index information due to the diversity of non-zero elements within the matrix, as well as the use of data with varying bit widths, result in increased irregularity and complexity in computations and memory references.**





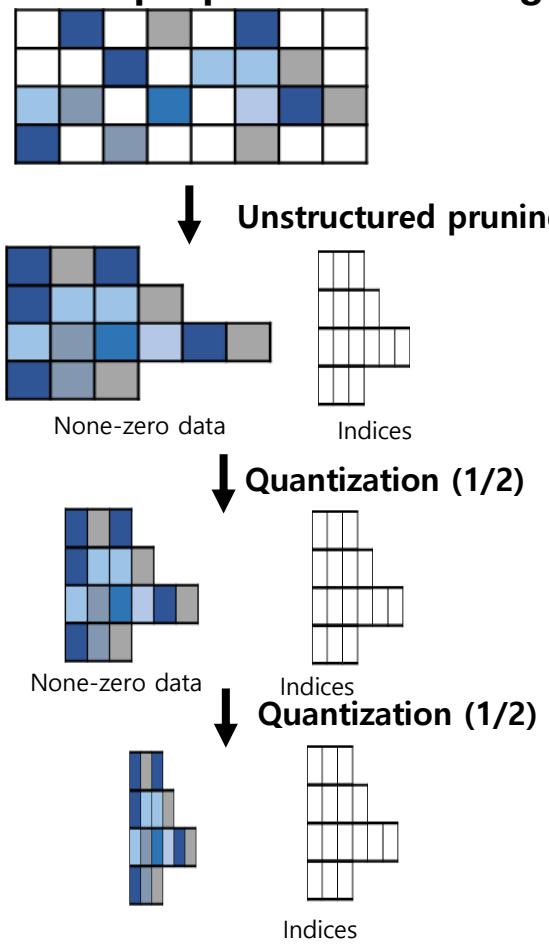
The distribution of non-zero elements as the pruning ratio changes (VGG16)

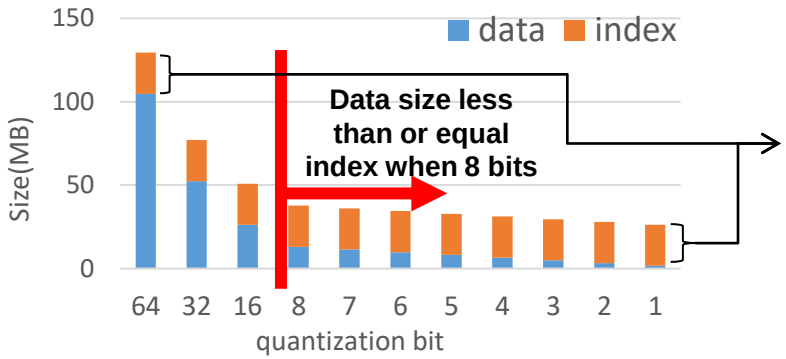




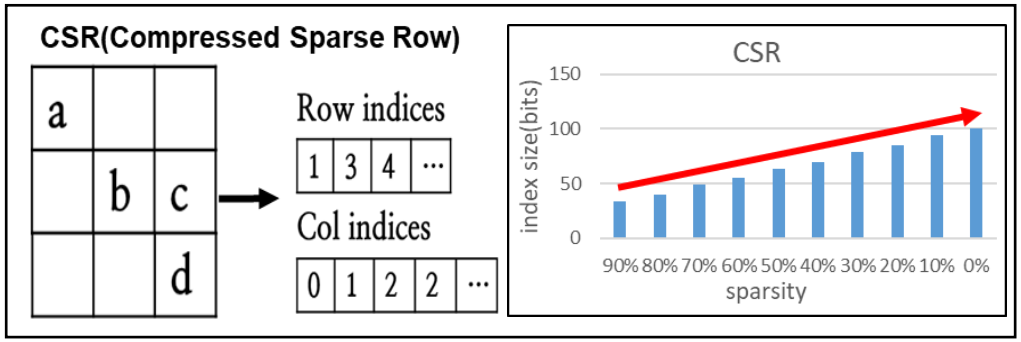
Irregularity and Increase of Index Size Storage Ratio with Aggressive Quantization and Sparsity.

- **As quantization progresses (FP32 → FP16 → FP8/INT8 → INT4), data size decreases while index size remains unchanged, leading to a relative increase in the proportion of storage capacity dedicated to storing index information.**



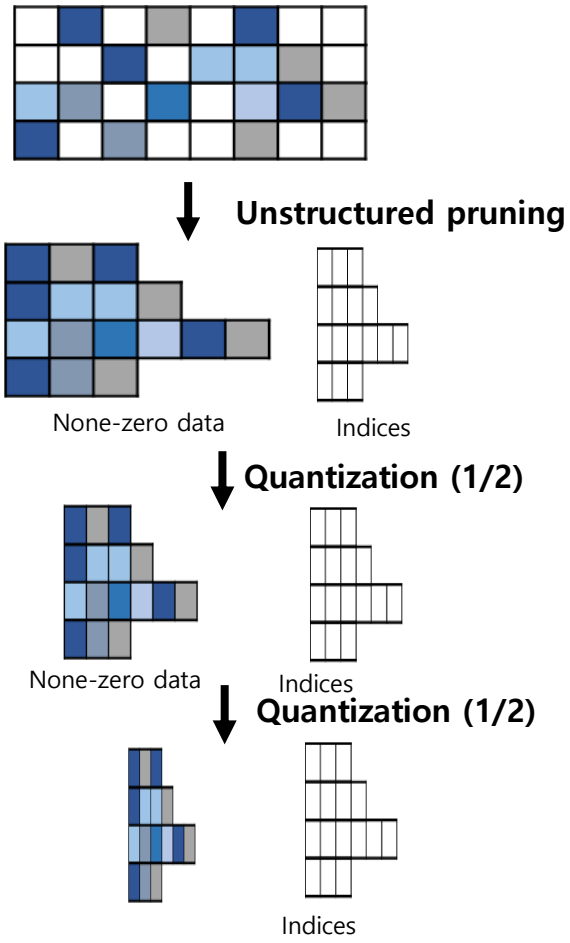


The data size decreases with the level of quantization, while the index size remains constant regardless of the degree of quantization.



Irregularity of Memory Access with Model Compression Techniques

- **The irregularity in both data and index sizes makes it difficult to efficient memory access.**



a	0	0
0	b	c
0	0	d

Array1

0	A	0
0	B	0
0	0	C

Array2

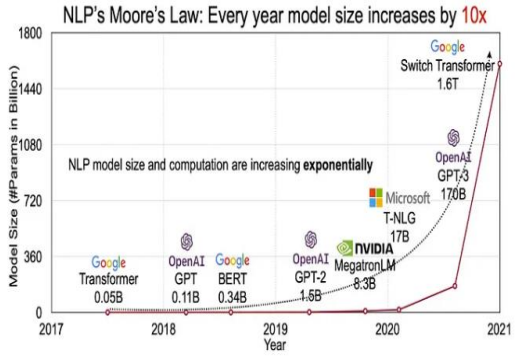
Memory

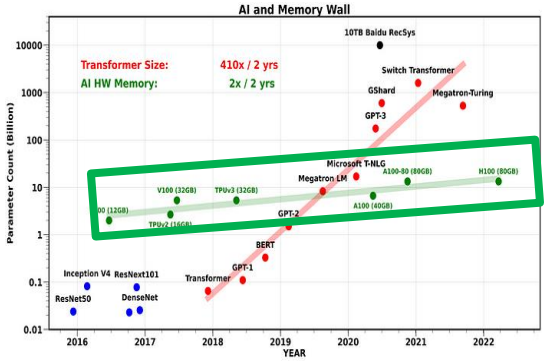
Row indices1	0	1	3	4
Col indices1	0	1	2	3
Row indices2	0	1	2	3
Col indices2	1	1	3	

Value1	a	b	c	d
Value2	A	B	C	

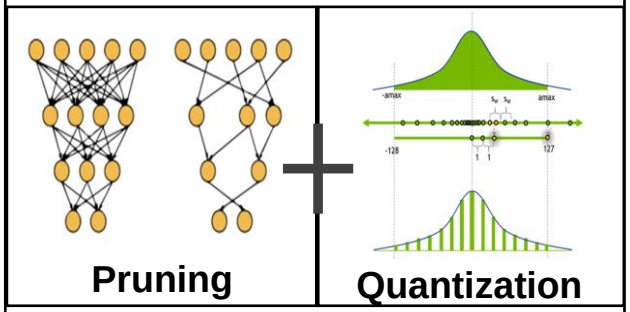
Demands for Model Compression Techniques & Design Issues.

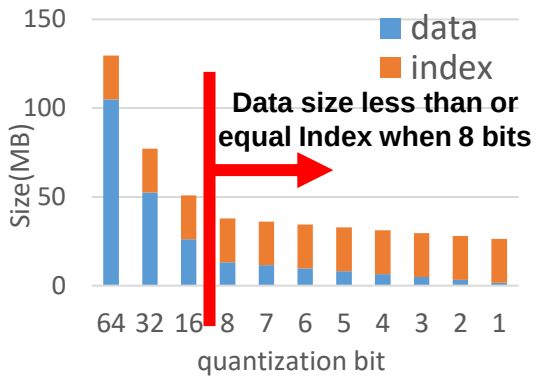
Issue 1: Explosive Growth of AI Model Size



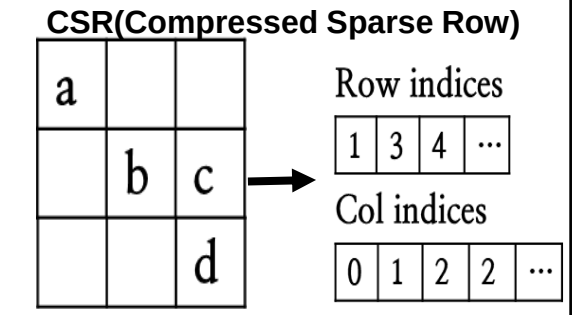


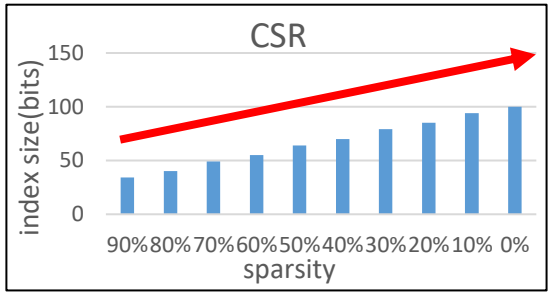
Issue 2: Increase in Index Data Overhead with Pruning and Aggressive Quantization.

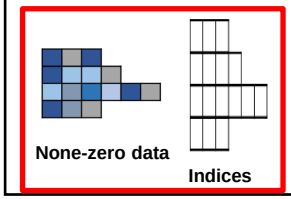


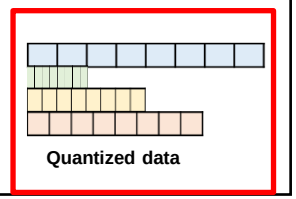


Issue 3: Irregularities of Memory Access in Index and Data Sizes with Conventional Pruning and Quantization.



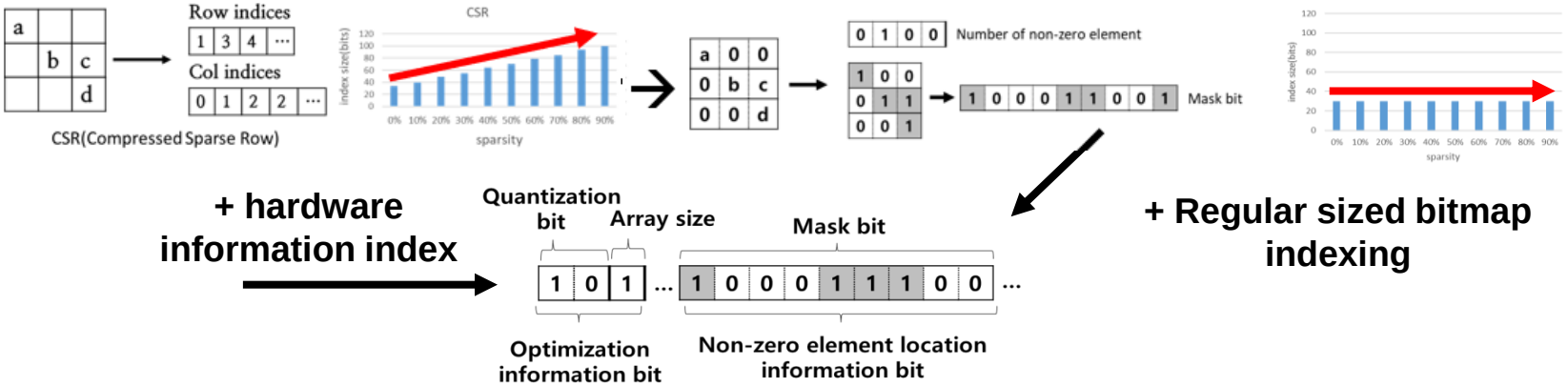






A Novel Sparse Matrix Indexing Scheme, Considering both Quantization and Pruning.

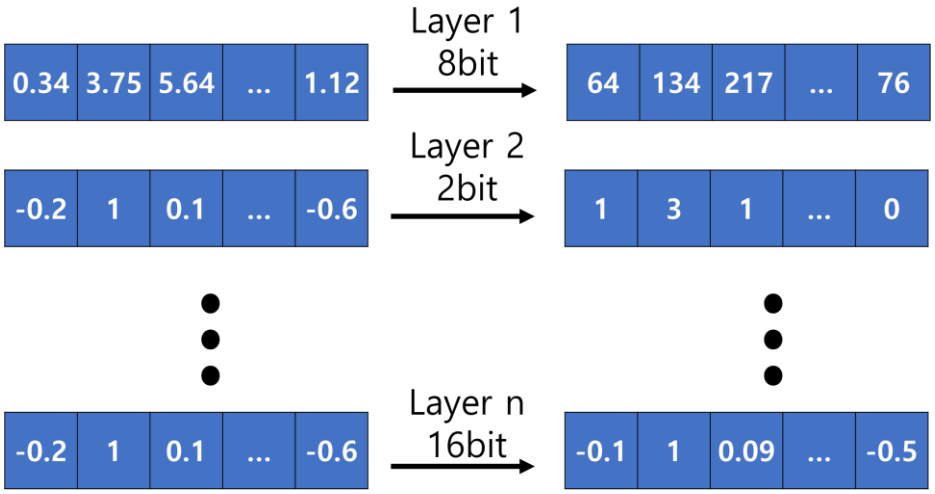
- **Optimization information bitmap indexing:**
  - **a novel sparse matrix indexing technique**
  - **the index size remains constant regardless of the number of non-zero elements in the matrix (in contrast to existing sparse matrix indexing techniques such as CSR, RLE, where the index size varies based on the number of non-zero elements in the matrix).**
  - **includes optimization information for operations and memory access, along with indexing information for non-zero elements and quantization data, to optimize computation and memory references**
  - **has smaller index size (cf. CSR)**



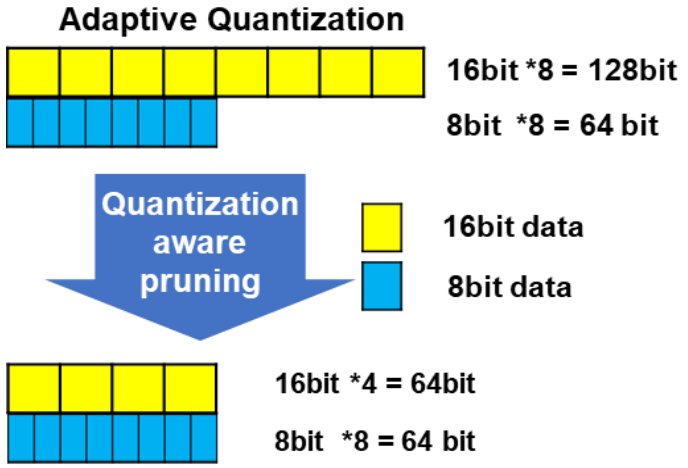
Compression representation technique utilizing bitmaps and containing performance-optimization information

Methods for Ensuring Regularity in Memory Access

- **Adaptive quantization, quantization aware pruning techniques**
  - **Proposal for an adaptive quantization technique that applies varying levels of quantization to layers, matrix/vector units, etc., considering outliers.**
  - **Quantization aware pruning technique that adjusts the level of pruning based on the degree of quantization to alleviate irregularities in memory and computation resulting from quantization and pruning.**



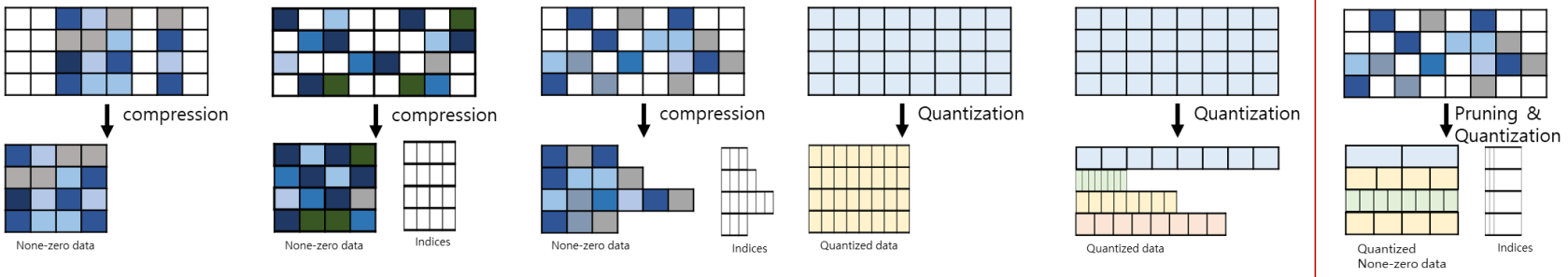
Adaptive Quantization



Quantization Aware Pruning

Adaptive Quantization, Quantization Aware Pruning

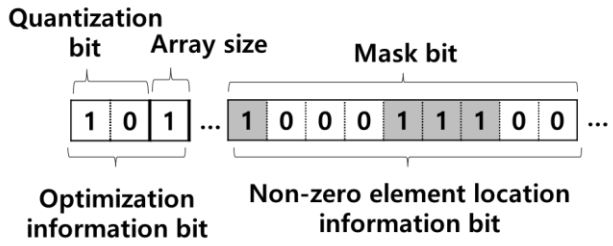
- **Adaptive quantization, quantization aware pruning techniques**
  - **Proposal for an adaptive quantization technique that applies varying levels of quantization to layers, matrix/vector units, etc., considering outliers.**
  - **Quantization aware pruning technique that adjusts the level of pruning based on the degree of quantization to alleviate irregularities in memory and computation resulting from quantization and pruning.**

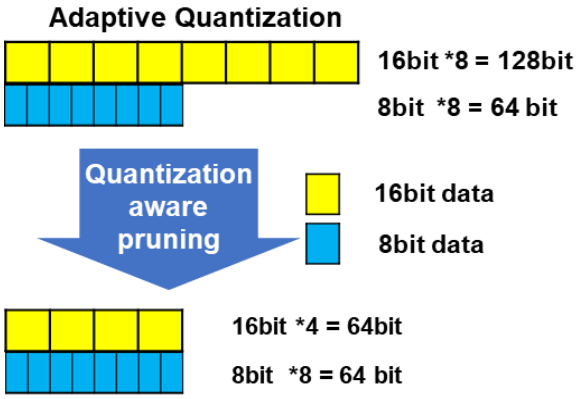


	Structured pruning	N:M pruning	Unstructured pruning	Uniformed quantization	Fine grain quantization	Our goal
Accuracy trade off	--	+	++	--	++	++
Memory footprint	++	+	--	++	--	+
Compute complexity	++	+	--	++	--	+
Accuracy per Memory size	-	+	+	-	+	++

Summary

- As AI models grow larger, the need for lightweighting techniques becomes essential especially for On-device AI.
- However, maintaining accuracy while lightweighting the model involves a trade-off that increases memory access and computational complexity.
- The research proposes a new sparse matrix indexing technique based on bitmap, where the index size remains constant regardless of the number of non-zero elements in the matrix and quantization-aware pruning to deliver the regularity for efficient memory access, higher utilization and less AI accelerator hardware complexity.





Hardware friendly sparse matrix representation and quantization aware pruning techniques

Acknowledgements

- **This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00167143)**

References

- Rangharajan Venkatesan, “Introduction to Machine Learning Applications and Hardware Aware Optimizations,” ISSCC 2024 Short Course Machine Learning Hardware: Considerations and Accelerator Approaches
- Mingjie Sun, et. al., “A SIMPLE AND EFFECTIVE PRUNING APPROACH FOR LARGE LANGUAGE MODELS,” ICLR 2024 :
- Bram Verhoef, “Quantization LLMs for Efficient Inference at the Edge,” ISSCC 2024 Forum on Energy-Efficient AI-Computing Systems for Large-Language Models
- Song Han, “AutoML for TinyML with Once-for-All Network,” ICLR 2020 NAS Workshop
- S. Han, et al., “Learning both weights and connections for efficient neural network” NeurIPS , 2015
- Marian Verhelst, “Architecture and design approaches to ML hardware
- acceleration: edge and mobile environments,” ISSCC 2024 Short Course Machine Learning Hardware: Considerations and Accelerator Approaches
- Zhang Y, et. al., “Learning best combination for efficient n: M sparsity,” Advances in Neural Information Processing Systems. 2022
- Ajay Jaiswal, et. al., “Attend Who is Weak: Pruning-assisted Medical Image Localization under Sophisticated and Implicit Imbalances,” 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)
- ISSCC 2024 Short Course Machine Learning Hardware: Considerations and Accelerator Approaches
- ISSCC 2024 Forum on Energy-Efficient AI-Computing Systems for Large-Language Models