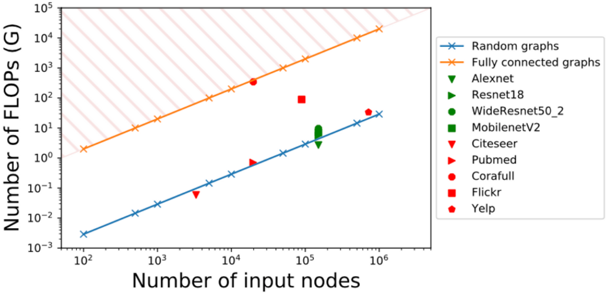
Quantized Training for Graph Neural Networks

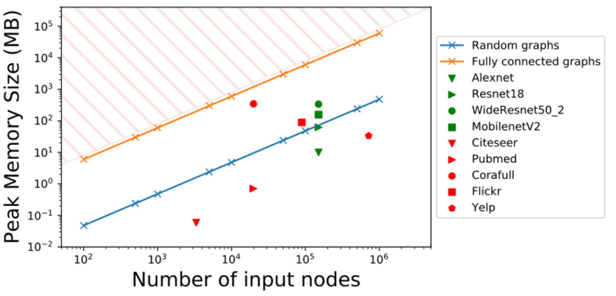
Issa Bqain

Supervisors: Dr Aaron Zhao and Dr Daniel Lo
Second Marker: Professor Christos-Savvas Bouganis

Motivation

Training Requirements



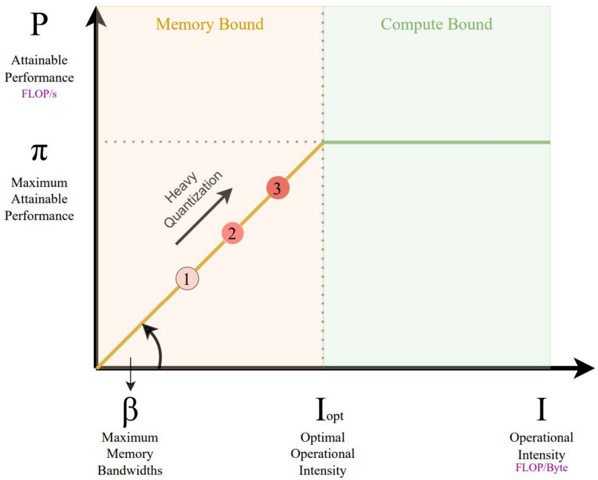


Motivation

Datasets

Dataset	#Graphs	#Nodes	#Edges	#Features	#Labels
Reddit	1	232,965	114,615,892	602	41
Yelp	1	716,847	13,954,819	300	100
Amazon Products	1	1,569,960	264,339,468	200	107
QM9	130,831	$\approx$18.0	$\approx$37.3	11	19
ZINC	249,456	$\approx$23.2	$\approx$49.8	1	1
PCBA	437,929	$\approx$26.0	$\approx$56.2	9	128

Motivation
Roofline Model



Yang et al. 2023

$$\text{Operational Intensity} = \frac{\text{No. of operations}}{\text{Memory Traffic}}$$

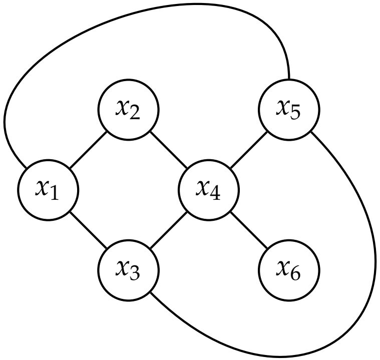
OBJECTIVES

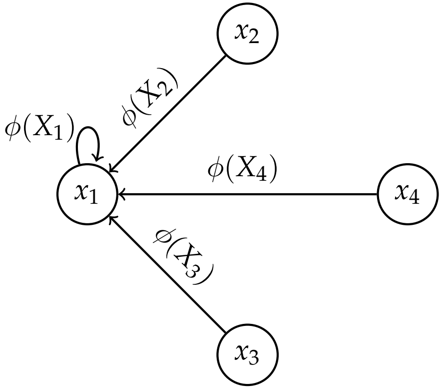
- **Quantizable Components**
  - Identify different quantizable components in GNN training and investigate the effect of various bit width quantizations on model accuracy
- **Quantization**
  - Fixed-Point Quantization
  - Microsoft Floating Point
- **Dynamic Quantization**
  - Investigate and Evaluate the effects of static and active dynamic quantization techniques on model accuracy

Message Passing Propagation

1. Message
2. Aggregate
3. Update

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$





Message

- $\mathcal{N}_i = \{j \ : \ e_{ij} \in E\}$
- $\phi(x_j) = W_j x_j + b$

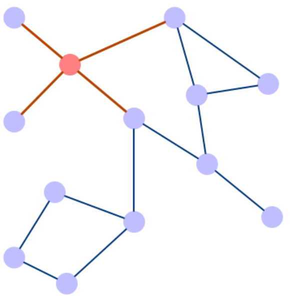
Aggregate

- $m_i = \gamma(\phi(x_i)) \forall j \in \mathcal{N}_i$

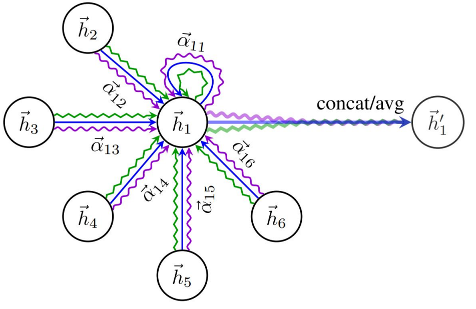
Update

- $\overline{x}_i = \lambda(\psi(\phi(x_i) + m_i))$

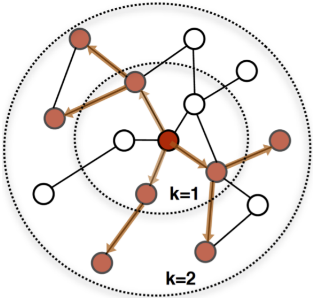
Graph Convolutional Layer (GCN)



Graph Attention Network (GAT)



Graph Sample and Aggregate (GraphSAGE)



Mapping a floating-point number to fixed-point $x_{fp} \in [\alpha_{fp}, \beta_{fp}] \implies x_i \in [\alpha_i, \beta_i]$

$$\boxed{x_i = \text{clamp}(\text{round}\big(\frac{1}{c}x_{fp} - d\big), \alpha_i, \beta_i)}$$

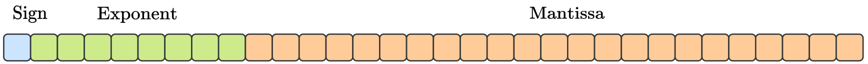
$$c = \frac{\beta_{fp} - \alpha_{fp}}{\beta_i - \alpha_i}$$
$$d = \frac{\alpha_{fp}\beta_i - \beta_{fp}\alpha_i}{\beta_{fp} - \alpha_{fp}}$$

$$\text{clamp}(m, n, p) = \begin{cases} n & \text{if } m < n \\ m & \text{if } n \leq m \leq p \\ p & \text{if } m > p \end{cases}$$

Quantization

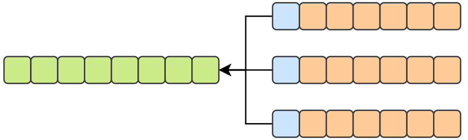
Microsoft Floating Point (MSFP)

IEEE-754 32-bit Floating Point



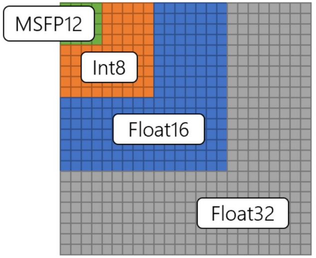
$$[(-1)^{s_0}\, 2^{e_0}\, m_0, (-1)^{s_1}\, 2^{e_1}\, m_1,\ \ldots.\ , (-1)^{s_n}\, 2^{e_n}\, m_n]$$

Microsoft Floating Point (MSFP)



$$2^{e_{\text{shared}}}[(-1)^{s_0}\, m_0, (-1)^{s_1}\, m_1,\ \ldots.\ , (-1)^{s_n}\, m_n]$$

Symmetric Quantization
Potential Density Gains



Arithmetic density relative to FP32, Rouhani et al. 2020.

	FP32	FP16	M16	M15	M14	M13	M12	M11	INT8	INT4
Memory Density	1.0x	2.0x	3.8x	4.3x	4.9x	5.8x	7.1x	9.1x	4.0x	8.0x
Arithmetic Density	1.0x	1.8x	8.8x	10.8x	13.9x	18.3x	31.9x	50.9x	7.7x	20.9x

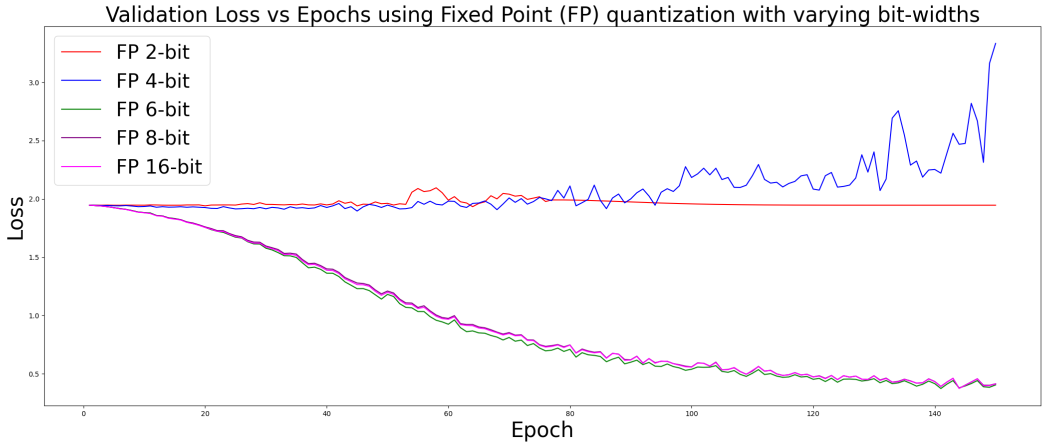
Dynamic Quantization

Static Dynamic Quantization (**SDQ**)

- **SDQ[**x, y, z, v**]** will train using bit-width x for the first 25%, y for the next 25% and so forth.

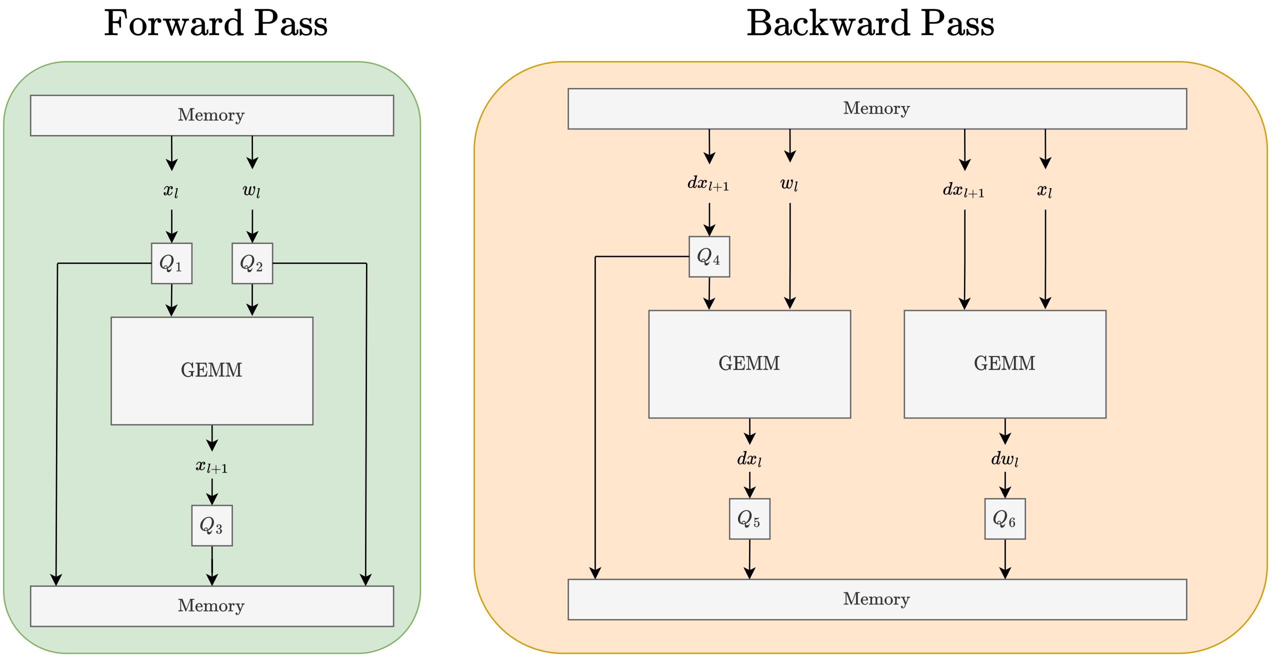
Active Dynamic Quantization (**ADQ**)

- **ADQ[x, y]** will train using x as the starting bit-width and y is the maximum bit-width.



Implementation

Quantized Linear Layer



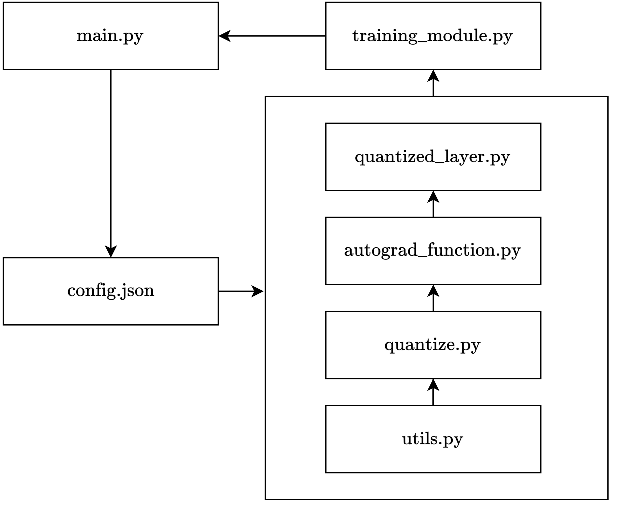
Emulated Quantization

Peak RAM (MB)

Model	CORA	PubMed	MUTAG
GCNConv - FP32	17.56	57.54	7.73
Fixed– 8bits	210.6	536.64	11.9
Fixed– 6bits	210.6	536.64	11.9
Fixed– 4bits	210.6	536.64	11.9
Fixed– 2bits	210.6	536.64	11.9
MSFP12 – 3m,8e,1s	169.808	438.31	10.11
MSFP13 – 4m,8e,1s	169.808	438.31	10.11
MSFP14 – 5m,8e,1s	169.808	438.31	10.11
MSFP15 – 6m,8e,1s	169.808	438.31	10.11
MSFP16 – 7m,8e,1s	169.808	438.31	10.11

Design
Benchmarking System

- In total, over 15000 tests were run including 6 datasets and over 40 quantization schemes.
- How can these results be run efficiently?



Conclusion

The results demonstrate it is possible to obtain accuracies within 1% of 32-bit floating point using various quantization schemes with significant potential gains in arithmetic and memory density

Quantization Scheme	Arithmetic Density	Memory Denisty
FP32	1.0x	1.0x
Fixed-Point	7.7x	4.0x
MSFP	18.3x	5.8x
Dynamic Quantization	26x	6.5x

References I

Rouhani, Bita et al. (Nov. 2020). *Pushing the Limits of Narrow Precision Inferencing at Cloud Scale with Microsoft Floating Point*. https://proceedings.neurips.cc/paper/2020/file/747e32ab0fea7fbd2ad9ec03daa3f840-Paper.pdf. ACM.

Yang, Guo et al. (2023). *Dynamic Stashing Quantization for Efficient Transformer Training*. arXiv: 2303.05295 [cs.LG].