ICASSP24

Activation Compression of Graph Neural Networks using Block-Wise Quantization with Improved Variance Minimization

Sebastian Eliassen (she@di.ku.dk) &
Raghavendra Selvan (raghav@di.ku.dk)

UNIVERSITY OF COPENHAGEN

Motivation

- Graph Neural Networks (GNNs) have seen widespread use within many Machine Learning (ML) applications

Motivation

- Graph Neural Networks (GNNs) have seen widespread use within many Machine Learning (ML) applications
- GNNs do suffer from poor memory scaling w.r.t. the amount of nodes

Motivation

- Graph Neural Networks (GNNs) have seen widespread use within many Machine Learning (ML) applications
- GNNs do suffer from poor memory scaling w.r.t. the amount of nodes
- EXACT (Liu et al. 2022) addresses this through extreme activation compression

Motivation

- Graph Neural Networks (GNNs) have seen widespread use within many Machine Learning (ML) applications
- GNNs do suffer from poor memory scaling w.r.t. the amount of nodes
- EXACT (Liu et al. 2022) addresses this through extreme activation compression
- We build upon this work with two key contributions

Motivation

- Graph Neural Networks (GNNs) have seen widespread use within many Machine Learning (ML) applications
- GNNs do suffer from poor memory scaling w.r.t. the amount of nodes
- EXACT (Liu et al. 2022) addresses this through extreme activation compression
- We build upon this work with two key contributions
  1. Block-wise quantization of GNNs

Motivation

- Graph Neural Networks (GNNs) have seen widespread use within many Machine Learning (ML) applications
- GNNs do suffer from poor memory scaling w.r.t. the amount of nodes
- EXACT (Liu et al. 2022) addresses this through extreme activation compression
- We build upon this work with two key contributions
    1. Block-wise quantization of GNNs
    2. Variance minimization due to activation compression

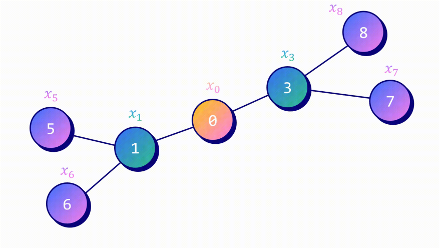
Overview

A Quick Introduction to GNNs

- Graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ with N nodes
  - $\mathbf{X} \in \mathbb{R}^{N \times F}$: Dense node feature matrix with F-dimensional features
  - $\mathbf{A} \in \{0, 1\}^{N \times N}$: Sparse adjacency matrix
  - $\mathbf{A}_{i,j} = 1$ if an edge exists between nodes i and j, otherwise $\mathbf{A}_{i,j} = 0$

A Quick Introduction to GNNs

- Graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ with N nodes
  - $\mathbf{X} \in \mathbb{R}^{N \times F}$: Dense node feature matrix with F-dimensional features
  - $\mathbf{A} \in \{0, 1\}^{N \times N}$: Sparse adjacency matrix
  - $\mathbf{A}_{i,j} = 1$ if an edge exists between nodes i and j, otherwise $\mathbf{A}_{i,j} = 0$



A Quick Introduction to GNNs

- Graph $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ with N nodes
  - $\mathbf{X} \in \mathbb{R}^{N \times F}$: Dense node feature matrix with F-dimensional features
  - $\mathbf{A} \in \{0, 1\}^{N \times N}$: Sparse adjacency matrix
  - $\mathbf{A}_{i,j} = 1$ if an edge exists between nodes i and j, otherwise $\mathbf{A}_{i,j} = 0$

- GNN Layer Update
  - $\mathbf{H}^{(\ell+1)} = \sigma\left(\mathbf{A}\,\mathbf{H}^{(\ell)}\,\mathbf{\Theta}^{(\ell)}\right)$
  - Initial node representations: $\mathbf{H}^{(0)} := \mathbf{X}$
  - Weights: $\mathbf{\Theta}^{(\ell)} \in \mathbb{R}^{D \times D}$ at layer ℓ
  - Non-linearity: $\sigma(\cdot)$

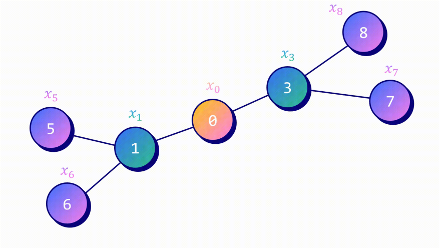


Figure: Animation of message-passing.

The Memory Bottleneck of GNNs

- Memory usage of activations
  - During the forward-pass all intermediate results $\left(\mathbf{H}^{(\ell)}\mathbf{\Theta}^{(\ell)}\right) \in \mathbb{R}^{N \times D}$ and node embedding matrices $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times D}$ are stored in memory.
  - Results in $\mathcal{O}\left(LND\right)$ space complexity, with L being the number of layers.
  - For this reason we focus on compressing activation maps.

Random projection

- Projection of the activations into a lower-dimensional space
- $\mathbf{H}^{(\ell)}_{\text{proj}} = \text{RP}(\mathbf{H}^{(\ell)}) = \mathbf{H}^{(\ell)}\mathbf{R}$ where $\mathbf{R} \in \mathbb{R}^{D \times R}$ is the normalized Rademacher matrix with $R < D$ (Achlioptas 2001).
- $\mathbf{R}$ has the following property: $\mathbb{E}[\mathbf{H}^{(\ell)}\mathbf{R}\mathbf{R}^\top] = \mathbb{E}[\mathbf{H}^{(\ell)}\mathbf{I}] = \mathbb{E}[\mathbf{H}^{(\ell)}]$

Random projection

- Projection of the activations into a lower-dimensional space
- $\mathbf{H}^{(\ell)}_{\text{proj}} = \text{RP}(\mathbf{H}^{(\ell)}) = \mathbf{H}^{(\ell)}\mathbf{R}$ where $\mathbf{R} \in \mathbb{R}^{D \times R}$ is the normalized Rademacher matrix with $R < D$ (Achlioptas 2001).
- $\mathbf{R}$ has the following property: $\mathbb{E}[\mathbf{H}^{(\ell)}\mathbf{R}\mathbf{R}^\top] = \mathbb{E}[\mathbf{H}^{(\ell)}\mathbf{I}] = \mathbb{E}[\mathbf{H}^{(\ell)}]$
- For this reason, R defines the projected dimensionality.

Stochastic Rounding

- Maps activations from `FLOAT32` to lower precision integers

Stochastic Rounding

- Maps activations from `FLOAT32` to lower precision integers
- The quantization, using b bits, consists of mapping your activations $\mathbf{h} \in \mathbb{R}^D$ into $B = 2^b - 1$ buckets and then rounding them to an integer. Specifically:

Stochastic Rounding

- Maps activations from `FLOAT32` to lower precision integers
- The quantization, using b bits, consists of mapping your activations $\mathbf{h} \in \mathbb{R}^D$ into $B = 2^b - 1$ buckets and then rounding them to an integer. Specifically:
  1. A shift and scale into $[0, B]$:
     $\bar{\mathbf{h}} = (\mathbf{h} - \min(\mathbf{h}))\frac{B}{\max(\mathbf{h}) - \min(\mathbf{h})}$

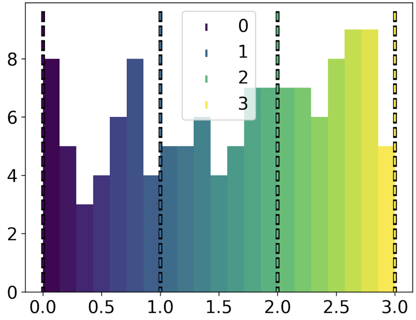


Figure: Example histogram of some $\bar{\mathbf{h}}$ with $b = 2$. Colors denote what integer a value most likely stochastically rounds to.

Stochastic Rounding

- Maps activations from FLOAT32 to lower precision integers
- The quantization, using b bits, consists of mapping your activations $\mathbf{h} \in \mathbb{R}^D$ into $B = 2^b - 1$ buckets and then rounding them to an integer. Specifically:
  1. A shift and scale into $[0, B]$:
     $$\bar{\mathbf{h}} = (\mathbf{h} - \min(\mathbf{h}))\frac{B}{\max(\mathbf{h}) - \min(\mathbf{h})}$$
  2. A stochastic rounding (SR) operation denoted by $\lfloor \cdot \rceil$:
     $$\mathbf{h}_{\mathtt{INT}} = \mathrm{Quant}(\mathbf{h}) = \lfloor \bar{\mathbf{h}} \rceil$$

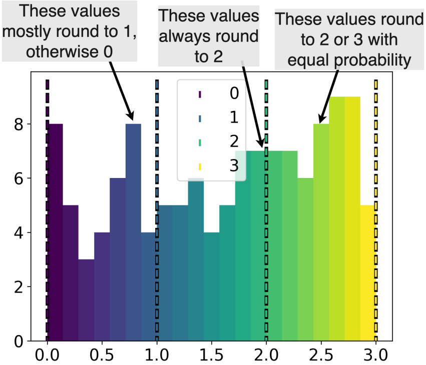


Figure: Example histogram of some $\bar{\mathbf{h}}$ with $b = 2$. Colors denote what integer a value most likely stochastically rounds to.

Dequantization

- The quantized embeddings $\mathbf{h}_{\mathtt{INT}}$ are dequantized in the backward-pass.

Dequantization

- The quantized embeddings $\mathbf{h}_{\mathtt{INT}}$ are dequantized in the backward-pass.
- Dequantization linearly maps $\mathbf{h}_{\mathtt{INT}}$ back to $\mathbf{h}$'s range, by performing the inverse transformation.

Dequantization

- The quantized embeddings $\mathbf{h}_{\mathtt{INT}}$ are dequantized in the backward-pass.
- Dequantization linearly maps $\mathbf{h}_{\mathtt{INT}}$ back to $\mathbf{h}$'s range, by performing the inverse transformation.
- Equation: $\hat{\mathbf{h}} = \frac{\max(\mathbf{h}) - \min(\mathbf{h})}{B} \mathbf{h}_{\mathtt{INT}} + \min(\mathbf{h})$.

Dequantization

- The quantized embeddings $\mathbf{h}_{\texttt{INT}}$ are dequantized in the backward-pass.
- Dequantization linearly maps $\mathbf{h}_{\texttt{INT}}$ back to $\mathbf{h}$'s range, by performing the inverse transformation.
- Equation: $\hat{\mathbf{h}} = \frac{\max(\mathbf{h}) - \min(\mathbf{h})}{B} \mathbf{h}_{\texttt{INT}} + \min(\mathbf{h})$.
- Property: $\mathbb{E}[\hat{\mathbf{h}}] = \mathbf{h}$

Dequantization

- The quantized embeddings $\mathbf{h}_{\mathtt{INT}}$ are dequantized in the backward-pass.
- Dequantization linearly maps $\mathbf{h}_{\mathtt{INT}}$ back to $\mathbf{h}$'s range, by performing the inverse transformation.
- Equation: $\hat{\mathbf{h}} = \frac{\max(\mathbf{h}) - \min(\mathbf{h})}{B} \mathbf{h}_{\mathtt{INT}} + \min(\mathbf{h})$.
- Property: $\mathbb{E}[\hat{\mathbf{h}}] = \mathbf{h}$
- Stochastic rounding (SR) keeps $\hat{\mathbf{h}}$ unbiased, with rounding probability proportional to boundary proximity

Dequantization

- The quantized embeddings $\mathbf{h}_{\texttt{INT}}$ are dequantized in the backward-pass.
- Dequantization linearly maps $\mathbf{h}_{\texttt{INT}}$ back to $\mathbf{h}$'s range, by performing the inverse transformation.
- Equation: $\hat{\mathbf{h}} = \frac{\max(\mathbf{h})-\min(\mathbf{h})}{B}\mathbf{h}_{\texttt{INT}} + \min(\mathbf{h})$.
- Property: $\mathbb{E}[\hat{\mathbf{h}}] = \mathbf{h}$
- Stochastic rounding (SR) keeps $\hat{\mathbf{h}}$ unbiased, with rounding probability proportional to boundary proximity
- This also applies to non-integer rounding values.

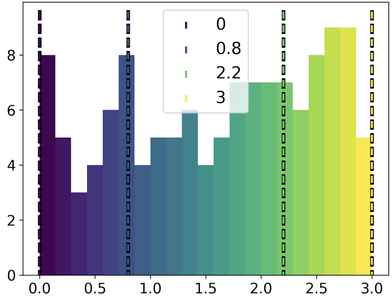


Figure: Same $\bar{\mathbf{h}}$ as before, but with non-uniform bin widths (quantization boundaries).

Overview

Block-wise quantization

- Taking inspiration from Chen et al. 2021; Dettmers et al. 2021, we group the input tensor such that G elements are quantized at a time.
- This is done with

$$\mathbf{H}^{(\ell)}_{\texttt{block}} \in \mathbb{R}^{\frac{N \cdot R}{G} \times G} := \text{reshape}\left(\mathbf{H}^{(\ell)}_{\texttt{proj}}, G\right),$$

  where reshape denotes the reshape function as known from packages like Numpy or Pytorch.
- Since each quantization operation is done row-wise, this increases concurrency.

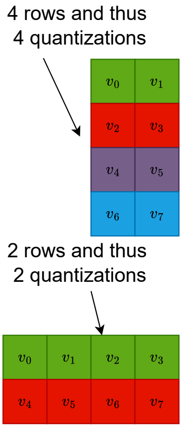


Figure: The matrix that has been reshaped to a lower row-count, also has fewer quantizations.

Results of block-wise quantization

Quant.	**G/R**	**Accuracy** ↑	**S** (e/s) ↑	**S Impr.** (%)	**M**(MB) ↓	**M Impr.** (%)

Results of block-wise quantization

Quant.	G/R	Accuracy ↑	S (e/s) ↑	S Impr. (%)	M(MB) ↓	M Impr. (%)
FP32	–	71.95 ± 0.16	13.07	-	786.22	-
INT2	1	71.16 ± 0.21	10.03	-	30.47	-
INT2	2	71.16 ± 0.34	10.23	+2.00	27.89	-8.47
	4	71.17 ± 0.22	10.46	+4.29	26.60	-12.70
	8	71.21 ± 0.39	10.54	+5.08	25.95	-14.83
	16	71.01 ± 0.19	10.55	+5.18	25.72	-15.59
	32	70.87 ± 0.29	10.54	+5.08	25.60	-15.98
	64	71.28 ± 0.25	10.54	+5.08	25.56	-16.11

Table: *G/R denotes the factor by which we increase the dimensionality via block-wise quantization. Standard deviations of test accuracy is computed over 10 runs*

Variance minimization

- While stochastic rounding (SR) is not biased, it does induce some variance.
- If we can minimize this variance, we can minimize the expected *quantization error*.
- Done by finding the quantization boundaries that minimize the variance.

Variance minimization

- While stochastic rounding (SR) is not biased, it does induce some variance.
- If we can minimize this variance, we can minimize the expected *quantization error*.
- Done by finding the quantization boundaries that minimize the variance.
- In order to do this we need three components:

Variance minimization

- While stochastic rounding (SR) is not biased, it does induce some variance.
- If we can minimize this variance, we can minimize the expected *quantization error*.
- Done by finding the quantization boundaries that minimize the variance.
- In order to do this we need three components:
    1. The distribution of activations (probability density function or pdf)

Variance minimization

- While stochastic rounding (SR) is not biased, it does induce some variance.
- If we can minimize this variance, we can minimize the expected *quantization error*.
- Done by finding the quantization boundaries that minimize the variance.
- In order to do this we need three components:
    1. The distribution of activations (probability density function or pdf)
    2. The variance induced as a function of the activation ($\text{Var}(\lfloor h \rceil)$)

Variance minimization

- While stochastic rounding (SR) is not biased, it does induce some variance.
- If we can minimize this variance, we can minimize the expected *quantization error*.
- Done by finding the quantization boundaries that minimize the variance.
- In order to do this we need three components:
  1. The distribution of activations (probability density function or pdf)
  2. The variance induced as a function of the activation (Var($\lfloor h \rceil$))
  3. Through integration, we can use (1) and (2) to calculate the expected variance, which we then minimize as a function of the boundaries.

Distribution of the activations

- SR is performed on the normalized activations $\overline{\mathbf{H}}^{(\ell)}_{\text{proj}}$, which are all of the activations transformed into the range $[0, B]$.

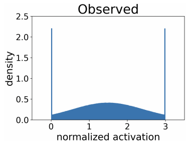


Figure: Histogram of observed and theorized $\overline{\mathbf{H}}^{(1)}_{\text{proj}}$ in a GNN model on the OGB-Arxiv data.

Distribution of the activations

- SR is performed on the normalized activations $\overline{\mathbf{H}}^{(\ell)}_{\text{proj}}$, which are all of the activations transformed into the range $[0, B]$.
- Two PDF's are hypothesized: $\mathcal{U}$ (EXACT)

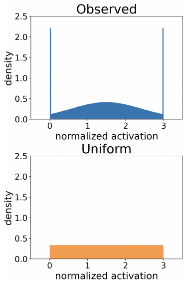


Figure: Histogram of observed and theorized $\overline{\mathbf{H}}^{(1)}_{\text{proj}}$ in a GNN model on the OGB-Arxiv data.

Distribution of the activations

- SR is performed on the normalized activations $\overline{\mathbf{H}}^{(\ell)}_{\text{proj}}$, which are all of the activations transformed into the range $[0, B]$.
- Two PDF's are hypothesized: $\mathcal{U}$ (EXACT) and $\mathcal{CN}$ (Ours).

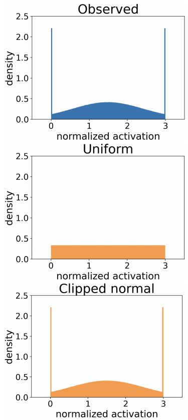


Figure: Histogram of observed and theorized $\overline{\mathbf{H}}^{(1)}_{\text{proj}}$ in a GNN model on the OGB-Arxiv data.

Distribution of the activations

- SR is performed on the normalized activations $\overline{\mathbf{H}}^{(\ell)}_{\text{proj}}$, which are all of the activations transformed into the range $[0, B]$.
- Two PDF's are hypothesized: $\mathcal{U}$ (EXACT) and $\mathcal{CN}$ (Ours).
- $\mathcal{CN}$ is the clipped normal distribution and is the result of clipping $\mathcal{N}$ such that the support lies in $[0, B]$.
- Empirically we have shown that we can define $\mathcal{CN}$ just from the dimensionality D, that is

$\mathcal{CN}_{[1/D]}$ is the pdf of y given,

$$y = \min(\max(0, X), B), \quad X \sim \mathcal{N}(\mu, \sigma),$$

where $\mu = B/2$ and $\sigma = -\mu/\Phi^{-1}(1/D)$.

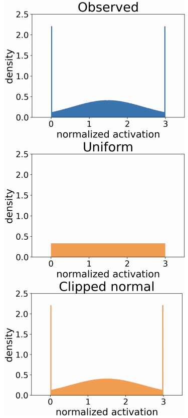


Figure: Histogram of observed and theorized $\overline{\mathbf{H}}^{(1)}_{\text{proj}}$ in a GNN model on the OGB-Arxiv data.

Distribution of SR variance

- Using Xia et al. 2020, we can estimate the variance induced by SR.
- This turns out to be

$$\mathrm{Var}(\lfloor h \rceil) = \sum_{i=1}^{i=B} \left(\delta_i(h - \alpha_{i-1}) - (h - \alpha_{i-1})^2\right),$$

where δ_i is the width of the bin containing h, and α_i is the starting position of the bin.

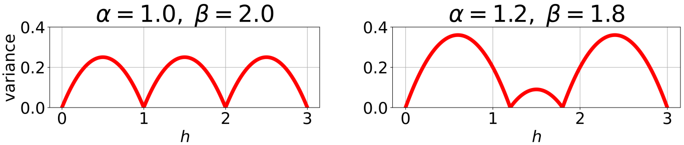


Figure: SR variance as a function of second (α) and third (β) boundary position.

Using the distributions to lessen variance induced by SR

- By combining the PDF of activations and the variance induced as a function of an activations ($\text{Var}(\lfloor h \rceil)$), we get:

$$\begin{aligned}\mathbb{E}[\text{Var}(\lfloor h \rceil)] = &\int_0^{\alpha} (\alpha \cdot h - h^2)\mathcal{CN}_{[1/D]}(h)\, dh \\ &+ \int_{\alpha}^{\beta} \left((\beta - \alpha)(h - \alpha) - (h - \alpha)^2\right)\mathcal{CN}_{[1/D]}(h)\, dh \\ &+ \int_{\beta}^{B} \left((B - \beta)(h - \beta) - (h - \beta)^2\right)\mathcal{CN}_{[1/D]}(h)\, dh\end{aligned}$$

- Using numerical integration we can minimize the above w.r.t. α and β (variance minimization), and cache the best boundaries for any D.

Results of variance minimization

Dataset	Layer	R	$\mathcal{U}$	$\mathcal{CN}_{[1/D]}$	Reduction Factor $(\times)$	Var. Reduction (%)

Results of variance minimization

Dataset	Layer	R	$\mathcal{U}$	$\mathcal{CN}_{[1/D]}$	Reduction Factor ($\times$)	Var. Reduction (%)
Arxiv	layer 1	16	0.0495	0.0213	2.32	3.17
	layer 2	16	0.0446	0.0016	27.88	2.09
	layer 3	16	0.0451	0.0041	11.00	2.19
Flickr	layer 1	63	0.0674	0.0017	39.65	6.14
	layer 2	32	0.0504	0.0033	15.27	4.37

Table: *Jensen-Shannon divergence measure for Uniform and Clipped Normal distributions compared to the normalized activations* $\bar{\mathbf{h}}$ *at each layer of the GNN for Arxiv and Flickr datasets.*

Results of variance minimization

Dataset	Layer	R	$\mathcal{U}$	$\mathcal{CN}_{[1/D]}$	Reduction Factor ($\times$)	Var. Reduction (%)
Arxiv	layer 1	16	0.0495	0.0213	2.32	3.17
	layer 2	16	0.0446	0.0016	27.88	2.09
	layer 3	16	0.0451	0.0041	11.00	2.19
Flickr	layer 1	63	0.0674	0.0017	39.65	6.14
	layer 2	32	0.0504	0.0033	15.27	4.37

Table: *Jensen-Shannon divergence measure for Uniform and Clipped Normal distributions compared to the normalized activations* $\bar{\mathbf{h}}$ *at each layer of the GNN for Arxiv and Flickr datasets.*

Quant.	G/R	Accuracy $\uparrow$	S (e/s) $\uparrow$	S Impr. (%)	M(MB) $\downarrow$	M Impr. (%)
FP32	–	71.95 $\pm$ 0.16	13.07	-	786.22	-
INT2	1	71.16 $\pm$ 0.21	10.03	-	30.47	-
INT2+VM	1	71.20 $\pm$ 0.19	9.16	-8.67	30.47	0.00

Overview

Summary

- GNNs have seen a large increase in popularity withing the ML-field.

Summary

- GNNs have seen a large increase in popularity withing the ML-field.
- Unfortunately they can suffer from poor memory scaling.

Summary

- GNNs have seen a large increase in popularity withing the ML-field.
- Unfortunately they can suffer from poor memory scaling.
- EXACT (Liu et al. 2022) tries to alleviate this, via extreme activation compression

Summary

- GNNs have seen a large increase in popularity withing the ML-field.
- Unfortunately they can suffer from poor memory scaling.
- EXACT (Liu et al. 2022) tries to alleviate this, via extreme activation compression
- We try to show that you can improve this further, even in an already very compressed activation space.

Conclusion

- Significant memory reduction and slight runtime speedup achieved through block-wise quantization.

Conclusion

- Significant memory reduction and slight runtime speedup achieved through block-wise quantization.
- Non-uniform distribution of GNN activation maps demonstrated.

Conclusion

- Significant memory reduction and slight runtime speedup achieved through block-wise quantization.
- Non-uniform distribution of GNN activation maps demonstrated.
- Introduced variable and non-uniform bin widths in stochastic rounding to reduce quantization variance.

Conclusion

- Significant memory reduction and slight runtime speedup achieved through block-wise quantization.
- Non-uniform distribution of GNN activation maps demonstrated.
- Introduced variable and non-uniform bin widths in stochastic rounding to reduce quantization variance.
- Methods are model-agnostic: opportunities for applying these methods to other architectures and pre-trained networks.

Bibliography

- Achlioptas, Dimitris (2001). “Database-Friendly Random Projections”. In: *Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. PODS ’01. Santa Barbara, California, USA: Association for Computing Machinery, pp. 274–281. ISBN: 1581133618. DOI: 10.1145/375551.375608. URL: https://doi.org/10.1145/375551.375608.
- Chen, Jianfei et al. (2021). *ActNN: Reducing Training Memory Footprint via 2-Bit Activation Compressed Training*. arXiv: 2104.14129 [cs.LG].
- Dettmers, Tim et al. (2021). “8-bit Optimizers via Block-wise Quantization”. In: *CoRR* abs/2110.02861. arXiv: 2110.02861. URL: https://arxiv.org/abs/2110.02861.
- Liu, Zirui et al. (2022). “EXACT: Scalable Graph Neural Networks Training via Extreme Activation Compression”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=vkaMaq95_rX.
- Xia, Lu et al. (2020). *Improved stochastic rounding*. arXiv: 2006.00489 [math.NA].

Acknowledgements

- Partly funded by **European Union's Horizon Europe Research and Innovation programme** under grant agreements No. 101070284 and No. 101070408.
- Thanks to **STIBOFONDEN** for their generous support.
- Grateful for the collaboration at **SAINTS Lab**.
- Check out **carbontracker.info** for advancing CO2e reduction in ML.

Funded by the Horizon 2020 Framework Programme of the European Union