Global QSGD: Practical Floatless Quantization for Distributed Learning with Theoretical Guarantees

Jihao Xin
KAUST

Marco Canini
KAUST

Peter Richtárik
KAUST

Samuel Horváth
MBZUAI

1 Introduction

The increase of deep learning models and dataset sizes make training time-consuming, while heavy communication, primarily due to gradients synchronization as a key bottleneck. Sapio et al. [7] show that communication can take up to 90% of a training iteration. A popular remedy is to reduce the size of communication data between nodes by applying gradient compression methods [1, 3, 6, 8, 9, 11]. Unfortunately, a majority of the proposed compressors are not natively compatible with the AllReduce collective communication primitive because of the change in data format and the need for custom reduction operations. To the best of our knowledge, the only compressors compatible with AllReduce are PowerSGD [10] and IntSGD [5, 7]. However, practical implementations of these methods are heuristic-based and do not come with rigorous theoretical guarantees. Concurrently, *C-Coll* [4] proposes error-bounded lossy compression with MPI collectives. We address this question: can we provide theoretical guarantees for gradient compression while retaining AllReduce compatibility for an efficient implementation?

2 Global Quantization

Our main contribution is a new compressor, *Global Quantization ($\mathcal{G}$-$\mathcal{Q}$)*, which quantizes 32-bit floats to smaller bitwidths utilizing the norm of the global gradient such that the quantized data is AllReduce-compatible. Assume gradient x, we define $\mathcal{G}$-$\mathcal{Q}$ as follows.

The *global quantization* operator with respect to the p norm and s levels

$$0 = l_s < l_{s-1} < l_{s-2} < \cdots < l_1 < l_0 = 1,$$

denoted $\mathcal{G}\text{-}\mathcal{Q}_l^{q,p}$, is defined as follows. Let $\mathbf{x} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{nd}$. Let $y_i \overset{\text{def}}{=} |x_i| / \|\mathbf{x}\|_{q,p} \in \mathbb{R}^d$ for all $i \in [n]$. Then

$$\mathcal{G}\text{-}\mathcal{Q}_l^{q,p}(\mathbf{x}) \overset{\text{def}}{=} \|\mathbf{x}\|_{q,p} \frac{1}{n} \sum_{i=1}^{n} \operatorname{sign}(x_i) \circ \xi_i(y_i)\,, \tag{1}$$

where $\xi_i(y_i)$ is an independent element-wise random rounding operator such that

$$(\xi_i(y_i))_j \overset{\text{def}}{=} \begin{cases} l_{u_i^j} & \text{with probability } \frac{(y_i)_j - l_{u_i^j+1}}{l_{u_i^j} - l_{u_i^j+1}} \\ l_{u_i^j+1} & \text{otherwise} \end{cases}, \tag{2}$$

for $j \in [d]$, where $u_i^j \in \{0, 1, 2, \dots, s\}$ is such that $l_{u_i^j} \le (y_i)_j \le l_{u_i^j+1}$.

We adopt two ways to cut the quantization intervals l:

- **Standard Dithering**, with linear levels, i.e., $l_i = {}^{s-i}/_{s}$.
- **Exponential Dithering [3]**, with exponential levels, i.e., $l_s = 0$ and $l_i = {}^{1}/_{2^{s-i}}$ for $i \in \{1, \dots, s\}$.[1]

Standard dithering is intrinsically linearly divided, and so can be aggregated directly using a fixed-point reduction operation in existing AllReduce implementations. We also propose a custom reduction operation for exponential dithering, which does not require dequantization. Thus, compressed data can be reduced through AllReduce primitive with the compression ratio up to $O(\sqrt{nd})$, where n is the number of computing nodes and d is the data size. On the other end, traditional quantization methods have to utilize AllGather whose compression ratio can only reach $O(\sqrt{d})$. We theoretically prove that global quantization is unbiased (i.e., error feedback is not required) and with bounded variance. We show that existing works with an unbiased compressor with bounded variance can seamlessly extend to global quantization with the same rate of convergence.

3 Evaluation

We implement $\mathcal{G}$-$\mathcal{Q}$ as a drop-in module for PyTorch DDP through a Python extension with CUDA offloads. We run experiments in a server with 4 A100 GPUs communicating via both NVLink and PCIe; we observe that our algorithm is beneficial even with the extremely fast NVLink. We also run large-scale validation in Google Cloud Platform (GCP) with 64 servers each equipped with 1 A100 GPU. Figure 1 shows that global quantization reaches 3.16× training speedup without loss of accuracy for the DeepLight model [2].

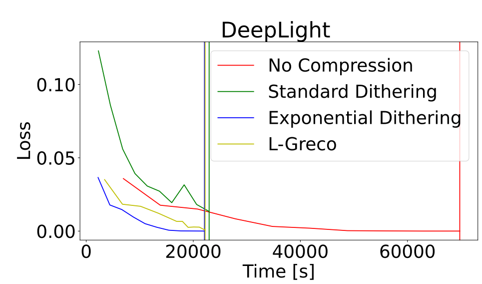

Figure 1. 10 epochs of DeepLight training behavior in GCP.

[1] We can work with any basis. We use base 2 for simplicity and the fact that this is naturally compatible with the binary representation of floats.

References

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-Efficient SGD via Gradient Quantization and Encoding. In *NeurIPS*.

[2] Wei Deng, Junwei Pan, Tian Zhou, Deguang Kong, Aaron Flores, and Guang Lin. 2021. DeepLight: Deep Lightweight Feature Interactions for Accelerating CTR Predictions in Ad Serving. In *WSDM*.

[3] Samuel Horváth, Chen-Yu Ho, Ľudovít Horváth, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. 2022. Natural Compression for Distributed Deep Learning. In *MSML*.

[4] Jiajun Huang, Sheng Di, Xiaodong Yu, Yujia Zhai, Jinyang Liu, Ken Raffenetti, Hui Zhou, Kai Zhao, Zizhong Chen, Franck Cappello, Yanfei Guo, and Rajeev Thakur. 2023. C-Coll: Introducing Error-bounded Lossy Compression into MPI Collectives. arXiv:2304.03890 [cs.DC]

[5] Konstantin Mishchenko, Bokun Wang, Dmitry Kovalev, and Peter Richtárik. 2021. IntSGD: Adaptive Floatless Compression of Stochastic Gradients. In *ICLR*.

[6] Ali Ramezani-Kebrya, Fartash Faghri, Ilya Markov, Vitalii Aksenov, Dan Alistarh, and Daniel M. Roy. 2021. NUQSGD: Provably Communication-efficient Data-parallel SGD via Nonuniform Quantization. *Journal of Machine Learning Research (JMLR)* 22, 114 (2021), 1–43.

[7] Amedeo Sapio, Marco Canini, Chen-Yu Ho, Jacob Nelson, Panos Kalnis, Changhoon Kim, Arvind Krishnamurthy, Masoud Moshref, Dan R. K. Ports, and Peter Richtárik. 2021. Scaling Distributed Machine Learning with In-Network Aggregation. In *NSDI*.

[8] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 2014. 1-Bit Stochastic Gradient Descent and Application to Data-Parallel Distributed Training of Speech DNNs. In *INTERSPEECH*.

[9] Nikko Strom. 2015. Scalable Distributed DNN Training using Commodity GPU Cloud Computing. In *INTERSPEECH*.

[10] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. 2019. PowerSGD: Practical Low-Rank Gradient Compression for Distributed Optimization. In *NeurIPS*.

[11] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2017. TernGrad: Ternary Gradients to Reduce Communication in Distributed Deep Learning. In *NeurIPS*.