Poster Abstract: ElastiCL: Elastic Quantization for Communication Efficient Collaborative Learning in IoT

Bharath Sudharsan
Data Science Institute,
NUI Galway, Ireland
b.sudharsan1@nuigalway.ie

Dhruv Sheth
Edge Impulse, San Jose,
California, USA
dhruvsheth.linkit@gmail.com

Shailesh Arya
Nirma University,
Ahmedabad, Gujarat, India,
aryashailesh12@gmail.com

Federica Rollo
University of Modena and
Reggio Emilia, Italy
federica.rollo@unimore.it

Piyush Yadav
Data Science Institute,
NUI Galway, Ireland
p.yadav1@nuigalway.ie

Pankesh Patel
University of South
Carolina, Columbia, USA
ppankesh@mailbox.sc.edu

John G. Breslin
Data Science Institute,
NUI Galway, Ireland
john.breslin@nuigalway.ie

Muhammad Intizar Ali
School of Electronic
Engineering, DCU, Ireland
ali.intizar@dcu.ie

ABSTRACT

Transmitting updates of high-dimensional neural network (NN) models between client IoT devices and the central aggregating server has always been a bottleneck in collaborative learning - especially in uncertain real-world IoT networks where congestion, latency, bandwidth issues are common. In this scenario, gradient quantization is an effective way to reduce bits count when transmitting each model update, but with a trade-off of having an elevated error floor due to higher variance of the stochastic gradients. In this paper, we propose ElastiCL, an elastic quantization strategy that achieves transmission efficiency plus a low error floor by dynamically altering the number of quantization levels during training on distributed IoT devices. Experiments on training ResNet-18, Vanilla CNN shows that ElastiCL can converge in much fewer transmitted bits than fixed quantization level setups, with little or no compromise on training and test accuracy.

CCS CONCEPTS

• **Computing methodologies → Distributed algorithms**.

KEYWORDS

Collaborative Learning, IoT Devices, Quantization Levels.

ACM Reference Format:
Bharath Sudharsan, Dhruv Sheth, Shailesh Arya, Federica Rollo, Piyush Yadav, Pankesh Patel, John G. Breslin, and Muhammad Intizar Ali. 2021. Poster Abstract: ElastiCL: Elastic Quantization for Communication Efficient Collaborative Learning in IoT. In *The 19th ACM Conference on Embedded Networked Sensor Systems (SenSys'21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3485730.3492885

Table 1: Summary of notations used during ElastiCL design.

Symbol	Description
s	Number of quantization levels
$Q(.)$	Stochastic uniform quantizer [5] with s quantization levels
η	Learning rate
L, σ^2, f^*	Parameters
$\mathbf{w}$	Parameter vector of the global model
d	Dimension of $\mathbf{w}$
f	Empirical risk that need to be minimized
τ	Iterations of the SGD method
B	Total bits transmitted by a client IoT device to server
n	Number of clients (IoT devices)
C_s	Number of bits transmitted by a client to server per round
k	Index of the communication round

1 INTRODUCTION

Due to privacy concerns, the distributed clients perform on-device learning and only share model updates with a central server. In such settings, using distributed versions of stochastic gradient descent (SGD) has gained attention due to its higher scalability characteristics. A popular example can be using the data-parallel schemes such as QSGD [1], Buckwild [3], TernGrad [18], SignSGD [2] in a setting with n distributed devices that split a large dataset among themselves. Here, each client keeps a private copy of the model parameters while having access to the global function's stochastic gradients that need to be minimized. Then, each device, at each training round, privately computes its stochastic gradient using the local data it sees. This learned information is then broadcasted/synchronized to other training-involved devices, using which aggregation is performed at each device to obtain the updated model parameters.

In such decentralized scenarios, to tackle IoT networks created bottlenecks [6, 10], the load on the transmission channel is reduced by limiting the clients-server communication frequency. However, this optimization is not enough for high-dimensional NN models with large size model updates [4]. During the model update step by clients, compression methods have also been investigated to reduce transmission packet size [16]. However, such compression methods usually add to the error floor of the training objective as they increase the variance of the updates. Thus, one needs to carefully choose the number of quantization levels to strike the best error-communication trade-off.

We propose ElastiCL, a strategy to dynamically alter the number of quantization levels used to represent a model update and achieve a low error floor as well as communication/transmission efficiency. The key idea of ElastiCL is to bound the convergence of training error in terms of the number of bits transmitted, unlike traditional

SenSys'21, November 15–17, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9097-2/21/11.
https://doi.org/10.1145/3485730.3492885

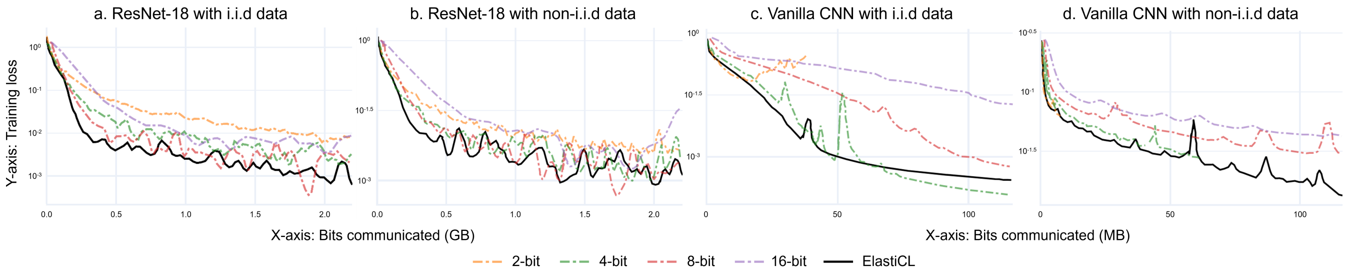

Figure 1: ElastiCL on ResNet-18 uses fewer bits to reach a lower loss - a loss of 0.036 in 0.297 GB, while the 2-bit method uses 1.78 GB. For non-i.i.d data distribution, ElastiCL on Vanilla CNN achieves the lowest error floor of 0.027 compared to others.

approaches which bound error with respect to number of training rounds. We use this convergence analysis to adapt the number of quantization levels during training based on the current training loss. ElastiCL can be considered orthogonal to other proposed methods of adaptive compression such as varying the spacing between quantization levels and reusing outdated gradients. An adaptive method for tuning the number of local updates or the communication frequency exists [17] - ElastiCL tunes the number of bits transmitted per round. Table 1 shows all the notations used during ElastiCL design in the next section.

2 PROPOSED DESIGN

The motivation behind elastically altering the number of quantization levels s during training is, smaller s (coarser quantization) results in poor convergence of training loss vs training rounds - but reduces the number of bits communicated per round C_s. Also, smaller s enables performing more training rounds for the same number of bits communicated, leading to a faster initial drop in training loss. So, to reach a lower error floor, one of the ElastiCL design principles is to start with a small s, followed by elastic increments as training progresses. When considering Q(.) with s quantization levels, if the η satisfies $1 - \eta L\left(1 + \frac{d\tau}{s^2 n}\right) - 2\eta^2 L^2 \tau(\tau - 1) \geq 0$, then the error upper bound in terms of B:

$$\frac{C_s}{B\tau} \sum_{k=0}^{(B/C_s)-1} \sum_{t=0}^{\tau-1} \mathbb{E}\left[\|f\left(\overline{\mathbf{w}}_{k,t}\right)\|_2^2\right] \leq A_1 \log_2(4s) + \frac{A_2}{s^2} + A_3 \quad (1)$$

Here, $\overline{\mathbf{w}}_{k,t} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{w}_{k,t}^{(i)}$ is the averaged model across training involved client devices at each step, and

$$A_1 = \frac{2\left(f\left(\mathbf{w}_0\right) - f^*\right) d}{\eta B\tau}, \quad A_2 = \frac{\eta L d\sigma^2}{n}$$
$$A_3 = \frac{\eta^2\sigma^2(\tau-1)L^2(n+1)}{n} + \frac{\eta L\sigma^2}{n} + A_1\frac{d+32}{d}$$

and f^* is the minimum value of our objective, $\mathbf{w}_0$ is a random initialization point. This error bound Eqn (1), for different values of s, can be used to analyse the trade-off between coarse and aggressive quantization. As s is decreased, the value of $A_1 \log_2(4s)$ decreases but it also adds to the variance of our quantized updates which increases A_2/s^2. ElastiCL when deployed on IoT devices, elastically changes s in the employed stochastic uniform quantizer to minimize error upper bound at every value B. To do this, the entire training process is discretized into uniform transmission intervals where, in each interval, B_0 bits are transmitted. Below we present how to find optimal s for each interval.

Finding optimal s for each transmission interval. We propose selecting an s at any B (assuming $\mathbf{w}_0$ as the point of initialization) by setting the derivative of our error upper bound in Eqn (1) to zero. Doing so, we get a closed form solution of an optimal s as:

$$s^* = \sqrt{\frac{\eta^2 L\sigma^2\tau B \log_e(2)}{n\left(f\left(\mathbf{w}_0\right) - f^*\right)}} \quad (2)$$

At the beginning of the k-th transmission interval, clients are viewed as restarting learning at a new initialization point $\mathbf{w}_0 = \mathbf{w}_k$. Using Eqn (2) we see that the optimal s for transmitting the next B_0 bits can be given as:

$$s_k^* = \sqrt{\frac{\eta^2 L\sigma^2\tau B_0 \log_e(2)}{n\left(f\left(\mathbf{w}_k\right) - f^*\right)}}$$

As $f\left(\mathbf{w}_k\right)$ becomes smaller, the value of s_k^* increases supporting ElastiCL concept - elastically increasing s as training progresses. In practice, parameters L, σ^2 and f^* are unknown. Hence, to obtain a practically usable sequence for s_k^*, it is assumed $f^* = 0$ and divide s_k^* by s_0^* to obtain approximate adaptive rule in Eqn (3). Here, grid search can be used to find s_0^* value.

$$s_k^* \approx \sqrt{\frac{f\left(\mathbf{w}_0\right)}{f\left(\mathbf{w}_k\right)}}\, s_0^* \quad (3)$$

Model training rate. ElastiCL design so far assumed the existence of a fixed learning rate η. In practice, for better convergence, the learning rate needs to be gradually decreased during training. By extending Eqn (3), an adaptive/elastic sequence of s for a given sequence of learning rates is given in Eqn (4). Here, η_0 is the initial learning rate, and η_k is the learning rate in the k-th interval.

$$\text{ElastiCL} = s_k^* \approx \sqrt{\frac{\eta_k^2 f\left(\mathbf{w}_0\right)}{\eta_0^2 f\left(\mathbf{w}_k\right)}}\, s_0^* \quad (4)$$

3 EXPERIMENTATION

We tested ElastiCL by comparing its performance with fixed quantization schemes. Here, each learned parameter (model update) is represented using $b = \{2, 4, 8, 16\}$ bits, and a stochastic uniform quantizer is employed. The performance is measured by training Vanilla CNN and ResNet-18 for classification tasks using the CIFAR-10 dataset. The number of local updates is set to $\eta = 0.1$, $\tau = 10$, then let ElastiCL train on 10 devices for the CNN and 6 devices for the ResNet-18. To replicate the real-world heterogeneous IoT devices [11, 14], edge GPUs, AIoT boards [12, 13] were used as clients on which ElastiCL was deployed. The resource-constrained MCU devices can also be involved by employing IoT hardware-friendly training algorithm like Train++ [15], Glob2Train [8], Edge2Train [7], ML-MCU [9]. During training, the learning rate was reduced by a factor of 0.9 every 100 training rounds. The testing was performed by storing and using both independent and identically distributed (i.i.d), non-i.i.d distributions data on clients. In the non-i.i.d settings, for each dataset, sorting was performed according to the target class labels, then equally split among client devices.

The testing results are presented in Figure 1, which shows that ElastiCL can reach an error floor using much fewer bits. It can be observed that ElastiCL performs better for ResNet-18 (see Fig. 1. a-b) but showed slightly inferior performance than the 4-bit setting for Vanilla CNN (see Fig. 1. c). As observed in Eqn 4, a decreasing learning rate tries to reduce s_k^* while the drop in training loss does the opposite. In such scenarios, it is recommended to use a conservative learning rate sequence to improve ElastiCL performance/benefits. We report that ElastiCL achieved a test accuracy of 71.46% for ResNet-18, whereas 72.19% was achieved by the 16-bit quantization method. For Vanilla CNN, ElastiCL reached 83.76% test accuracy, while 83.15% was reached by the 16-bit method.

4 CONCLUSION

This paper presented ElastiCL, a communication efficient strategy to enable ethical intelligence extraction from large-scale data generated through a plethora of IoT sensors. ElastiCL is compatible and contributes to collaborative learning (techniques such as federated learning, split learning, distributed ensemble learning) by providing the ability to elastically alter the number of quantization levels during distributed training. Testing of ElastiCL on heterogeneous IoT hardware demonstrated its robustness to system variability, which is vital to scaling machine learning training on ubiquitous resource-limited client nodes in low bandwidth IoT networks.

The future extended version of this paper plans to include the following: Introduce ElastiCL assurance with proofs for its bounds on variance and communication bits; Report scalability performance of ElastiCL by distributed training by varying the count of involved devices; Report the solution quality provided by training using ElastiCL in terms of training loss, test accuracy, and variance; ElastiCL results comparison with popular schemes as Deep Gradient Compression, Atomo, TernGrad, QSGD, SignSGD, Buckwild.

ACKNOWLEDGEMENT

This publication has emanated from research supported in part by a research grant from Science Foundation Ireland (SFI) under Grant Number SFI/16/RC/3918 (Confirm) and also by a research grant from SFI under Grant Number SFI/12/RC/2289_P2 (Insight), with both grants co-funded by European Regional Development Fund.

REFERENCES

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Neural Information Processing Systems (NIPS).*

[2] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. 2018. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning (ICML).*

[3] Christopher M De Sa, Ce Zhang, Kunle Olukotun, and Christopher Ré. 2015. Taming the wild: A unified analysis of hogwild-style algorithms. In *Neural Information Processing Systems (NIPS).*

[4] Fartash Faghri, Ali Ramezani-Kebrya, et al. 2020. Adaptive Gradient Quantization for Data-Parallel SGD. In *Neural Information Processing Systems (NIPS).*

[5] Amirhossein Reisizadeh, Ramtin Pedarsani, et al. 2020. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics.*

[6] Federica Rollo, Bharath Sudharsan, Laura Po, and John G Breslin. 2021. Air Quality Sensor Network Data Acquisition, Cleaning, Visualization, and Analytics: A Real-world IoT Use Case. In *Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers.*

[7] Bharath Sudharsan, John G Breslin, and Muhammad Intizar Ali. 2020. Edge2train: A framework to train machine learning models (svms) on resource-constrained iot edge devices. In *10th International Conference on the Internet of Things (IoT).*

[8] Bharath Sudharsan, John G Breslin, and Muhammad Intizar Ali. 2021. Globe2Train: A Framework for Distributed ML Model Training using IoT Devices Across the Globe. In *18th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC).*

[9] Bharath Sudharsan, John G Breslin, and Muhammad Intizar Ali. 2021. ML-MCU: A Framework to Train ML Classifiers on MCU-based IoT Edge Devices. In *IEEE Internet of Things Journal.*

[10] Bharath Sudharsan, Pankesh Patel, John Breslin, Muhammad Intizar Ali, Karan Mitra, Schahram Dustdar, Omer Rana, Prem Prakash Jayaraman, and Rajiv Ranjan. 2021. Toward Distributed, Global, Deep Learning Using IoT Devices. In *IEEE Internet Computing.*

[11] Bharath Sudharsan, Pankesh Patel, John G Breslin, and Muhammad Intizar Ali. 2021. Enabling Machine Learning on the Edge using SRAM Conserving Efficient Neural Networks Execution Approach. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD).*

[12] Bharath Sudharsan, Pankesh Patel, Abdul Wahid, Muhammad Yahya, John G Breslin, and Muhammad Intizar Ali. 2021. Demo abstract: Porting and execution of anomalies detection models on embedded systems in iot. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation (IoTDI).*

[13] Bharath Sudharsan, Simone Salerno, Duc-Duy Nguyen, Muhammad Yahya, Abdul Wahid, Piyush Yadav, John G Breslin, and Muhammad Intizar Ali. 2021. TinyML benchmark: Executing fully connected neural networks on commodity microcontrollers. In *IEEE 7th World Forum on Internet of Things (WF-IoT).*

[14] Bharath Sudharsan, Piyush Yadav, John G Breslin, and Muhammad Intizar Ali. 2021. An SRAM Optimized Approach for Constant Memory Consumption and Ultra-fast Execution of ML Classifiers on TinyML Hardware. In *IEEE International Conference on Services Computing (SCC).*

[15] Bharath Sudharsan, Piyush Yadav, John G Breslin, and Muhammad Intizar Ali. 2021. Train++: An Incremental ML Model Training Algorithm to Create Self-Learning IoT Devices. In *18th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC).*

[16] Jun Sun, Zaiyue Yang, et al. 2019. Communication-efficient distributed learning via lazily aggregated quantized gradients. In *Neural Information Processing Systems (NIPS).*

[17] Jianyu Wang and Gauri Joshi. 2019. Adaptive Communication Strategies for Best Error-Runtime Trade-offs in Communication-Efficient Distributed SGD. In *arXiv.*

[18] 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 *Neural Information Processing Systems (NIPS).*