DEEPSHIFT-Q: EFFICIENT COMPRESSION METHOD FOR FEDERATED LEARNING WITH MINIMAL ACCURACY LOSS

Tien T. Nguyen
Institut national de la recherche scientifique (INRS)
vincent.ngtt@mail.com

Anderson Avila
Institut national de la recherche scientifique (INRS)
anderson.avila@inrs.ca

ABSTRACT

In response to the escalating growth of data and the heightened need for privacy-preserving machine learning, distributed machine learning paradigms, specifically Federated Learning (FL), have emerged. FL allows users to benefit from shared models trained on their local devices without compromising personal data privacy. However, the considerable communication overhead in transmitting model updates between edge devices and the central server poses a significant challenge, especially with large neural network models and constrained network bandwidth. To mitigate this challenge, various compression methods, such as gradient compression, model broadcast compression, and local computation compression, have been proposed. This paper presents findings on employing DeepShift-Q as a compression method and compare it with other compression techniques, focusing on their trade-off between communication cost and training accuracy in Federated Learning, particularly for Non-IID (Non-Independently and Identically Distributed) data. Experimental results using the DeepShift-Q method demonstrate its efficiency in enhancing communication while respecting computational constraints on edge devices. The study contributes insights to the ongoing discourse on addressing communication challenges in Federated Learning.

Keywords Federated Learning · Compression Methods · Communication Cost · DeepShift-Q · Non-IID Data · Training Accuracy · Data Transfer

1 Introduction

In recent years, the explosive growth of data and the increasing demand for privacy-preserving machine learning have prompted the development of novel paradigms in the field of distributed machine learning. The terminology Federated Learning (FL) was introduced by researchers from Google in 2016 for a decentralized approach in training data distributed on edge devices. This learning technique enable users taking advantages of shared models trained on their local devise without storing their personal data at a central server[16]. Instead, it leverages the power of edge computing by allowing local computation and updating of model parameters on individual devices. These updated parameters are then securely aggregated, resulting in a global model that reflects insights from the entire decentralized network, all without the need to share raw data across devices. While FL presents a compelling solution for privacy and data security, it is not without its challenges. One of the primary obstacles that arise in FL settings is the considerable communication overhead involved in transmitting model updates between clients' devices and the central server. This communication overhead can be particularly burdensome when dealing with large neural network models or scenarios with constrained network bandwidth[11]. As the number of participating devices grows, so does the communication cost, making FL an impractical solution in resource-constrained environments.

To address the challenges of communication traffic in FL, there has been a number of proposed compression methods. The compression objectives can be named as gradient compression, model broadcast compression and local computation compression[10]. To the best of our knowledge, a study of DeepShift-Q's compression performance on non-IID data and comparison between these three compression methods have not been conducted for a complete review on the trade-off between communication cost and training accuracy in Federated Learning. In this paper, we study how much the trade-off between communication cost and training accuracy is when employing different compression methods and efficiency of DeepShift-Q for Non-IID data in the Federated Learning settings.

2 Related work

There are numbers of literature which aim to the objective of gradient compression. Konecny et al proposed structured updates and sketched updates techniques to reduce uplink communication cost[11]. Suresh et al studied the Distributed Mean Estimation for a given communication costs[22]. The quantized stochastic gradient descent (QSGD) method helps smoothly trade off communication bandwith and convergence time[2]. The DeepShift methods was introduced by Elhoushi et al to reduce computational demand and power requirements of the convolution neural networks (CNNs) by substituting conventional multiplication-based convolution and linear operations with bitwise-shift-based convolution and linear operations, respectively[8].

One of approaches which target model broadcast compression is the Qsparse-local-SGD algorithm. This method was introduced by Basu et al in purpose of reducing number of bits transmitted to reach target training accuracy. The Qsparse-local-SGD algorithm melds aggressive sparsity, quantization, and local computation while incorporating error correction[3]. FedPAQ is another technique to target this compression objective. With FedPAQ technique, a fraction of devices participate in each training round by which model updates are done locally and quantized periodically averaged before being transmitted to the central server[18].

To target local computation compression, Federated Droupout method enables users efficiently train a small subset of the model locally before sending the model updates to global server[5]. In addition, Caldas et al. employed lossy compression on the global model to reduce downlink communication[5]. To speed up computation, FedBoost was introduced by Hamer et al. by using federated ensemble method[9].

In Federated Learnng, non-independent and identically distributed (non-IID) data is a typical feature of training machine learning models. The reason for non-IID data is that data located on clients' edge devices tend to be biased depending on clients' geographical places, demographics, personal habits, or even regulations of government where clients reside in. Indeed, in machine learning, the prevalent assumption that training and testing data adhere to identical distributions is frequently breached in practical real-world applications and scenarios [17]. There are typical manners in which data commonly diverge from being distributed identically such as covariate shift which pertains to alterations in the distribution of the input variables [19]; prior probability shift which occurs when there is an assumption that a causal model in the form of P(x|y)P(y) is valid, but the distribution P(y) undergoes changes between training and test situations [20]; concept drift which is defined as changes in the target concepts induced by a shifting context [27]; and concept shift which is closely connected to concept drift, arising when a model trained on data sampled from one distribution must be applied to data drawn from a different distribution [26]. Additionally, Independence data violations occur whenever the distribution Q undergoes changes during training, with a prominent example being observed in cross-device Federated Learning [10]. In Federated Learning settings, mobile devices engage in the process under specific conditions, such as idleness, charging, and availability of free wireless connection. These conditions introduce various sources of cyclic bias. Even when focusing solely on devices within a particular time zone, there is a potential for demographic bias [7]. Bonawitz et al. showed that the count of devices eligible for training tends to be significantly higher during nighttime at a given location [4].

One of proposed approaches for non-IID data in Federated Learning settings is dataset distillation. The approach involves maintaining a constant model while endeavoring to condense the knowledge acquired from a sizable training dataset into a smaller one. The goal is to generate a new, significantly smaller synthetic dataset that exhibits nearly the same performance as the original dataset [25]. Another technique to handle non-IID data is data augmentation which aims to enhance the variety of training data through random transformations or knowledge transfer and can serve as a strategy to alleviate localized data imbalance challenges in Federated Learning [23]. In essence, algorithmic solutions to the non-IID problem in Federated Learning primarily encompass meta-learning, multitask learning, and lifelong learning [15].An algorithm based approach of meta-learning for non-IID data is the local fine-tuning, considered one of the most traditional and effective methods for personalization, which involves adjusting the local models using local data after receiving the global model from the server [24]. FedAvg is a popular algorithm of this approach.

In this study, our contribution is to examine training accuracy and communication cost (via uploading bandwidth of edge device's model updates) of the employment of DeepShift-Q method in the Federated Learning settings.

3 Experiments

To examine performance of the DeepShift-Q method in improving communication efficiency while facilitating computational resources limit of edge devices in the Federated

Learning settings, we train the Multilayer Perception (MLP) model and Convolutional Neural Networks (CNNs) model with MNIST and CIFAR10 dataset. The MNIST dataset, which comprises 60,000 images for training and 10,000 for testing purposes, contains grayscale images measuring 28 by 28 pixels depicting numbers from zero to nine[13]. The CIFAR-10 dataset, comprising 50,000 training images and 10,000 testing images all in color and sized at 32 by 32 pixels, includes images of 10 different species of animals and vehicles that are commonly recognized[12].

In our experiment, we employ two neural networks: Multilayer Perception and Convolutional Neural Networks. MLP neural networks are constructed with layers comprising interconnected units. In these fully connected networks explored in this study, every node within a layer links to every node in the subsequent layers. These MLP structures typically encompass at least three layers: an initial input layer, one or more intermediary hidden layers, and a final output layer[6].Convolutional Neural Networks (CNNs) are a type of deep learning models inspired by biological systems. These networks streamline the process by combining the three stages into a singular neural network, trained from the raw pixel values to generate classifier outputs in an end-to-end fashion[13]. CNNs are composed of numerous layers of artificial neural networks that currently outperform traditional methods in tasks like pattern recognition, as well as detecting and identifying images and objects[1].

In Federated Learning, a significant hurdle is encountered with non-IID data, which complicates the convergence of global models and decreases accuracy. To set up the experiments, we distribute the datasets such that: every client receives an equal portion of data, calculated by dividing the number of training images in each dataset by a fixed total of clients which is set to 100 for all subsequent experiments. Additionally, we perform pathological non-IID partitions for both MNIST and CIFAR-10 datasets, dividing them into 200 shards to amplify this challenging scenario.

We employ different compression methods, comprising topk sparsification and quantization, to compare with DeepShift-Q method. All these different compression methods are compared to model training without compression so that we can evaluate the variation of communication cost under the form of model update size among these methods.

We use classic algorithm FederatedAveraging (FedAvg) for optimization purpose in the Federated Learning settings. FedAvg was introduced by McMahan et al.(2017) and was showed that FedAvg algorithm achieves successful training of various model architectures, which include MLP and CNNS, with relatively minimal communication rounds[16]. The consistent results are expected to obtain for other optimization algorithm. Regarding DeepShift-Q and quatization methods, we decide to compress to represent model weight under 4 bits.

FedAvg procedure can be summarized as above. In the first step (1), during each communication round, global model weights w_t of t^{th} communication round are sent from central server to k^{th} edge device to obtain a next round locally updated model weights w_{t+1}^k, which are computed by learning rate η and this k^{th} edge device's average gradient g_k. In next step (2), the global model weights are updated for next communication round (t+1) by computing weighted average of all K edge devices' local updates, which are transmitted from all clients to central server.

$$w_{t+1}^k : = w_t - \eta g_k \tag{1}$$

$$w_{t+1} : = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{2}$$

Table 1: Uplink data transfer for training with MLP model after 240 communication rounds

	Dataset	
Compression strategy	MNIST	CIFAR10
Top-10	2.8GB	9.2GB
Quantization (4 bits)	11.6GB	38.3GB
DeepShift-Q (4 bits)	9.3GB	30.7GB

The focus of this study is examination the efficacy of DeepShift-Q as a compression strategy in FL settings with non-IID data. Elhoushi et al. (2021) presented new operators, LinearShift and ConvShift, which, during the forward pass, substitute multiplication with bitwise shift operations and sign flipping. In the DeepShift-Q method, during the forward and backward passes, the weight matrix W is quantized to $\widetilde{W_q}$ by rounding it to the nearest power of 2[8].

$$\begin{aligned} S &= \text{sign}(W) \\ \widetilde{S} &= S \end{aligned} \tag{3}$$

$$\begin{aligned} P &= \log_2 (abs(W)) \\ \widetilde{P} &= round(P) \end{aligned} \tag{4}$$

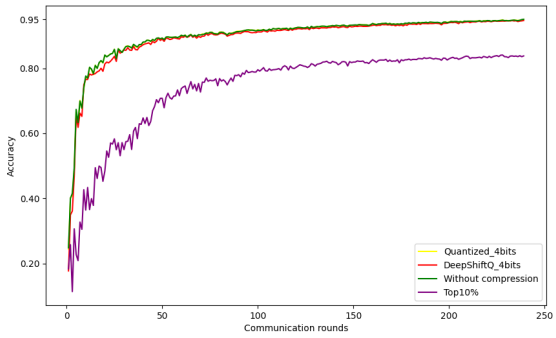
$$\widetilde{W_q} = flip(2^{\widetilde{P}}, \widetilde{S}) \tag{5}$$

To examine performance of DeepShift-Q method with non-IID data in FL, we compare training accuracy/loss and the volume of data transfer, which is measured by size of uplink local model updates, after certain number of communication rounds with two other popular compression methods: model updates quantization [11][2] and top-k sparsification method.[14][21]. We employ DeepShift-Q and quantization methhods with same 4 bits compression. For top-k method, we compress with top 10%Ẇe run experiments applying these compression strategies on training MLP and CNNs models with non-IID MNIST and CIFAR10 data.

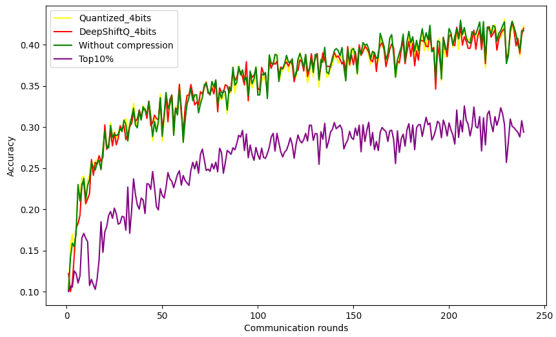
4 Results

Figure 1 and Figure 2 indicate difference in training accuracy and loss for each of three studied compression strategies from training MLP model with Non-IID MNIST and CIFAR10 datasets. Apparently, quantization method has highest accuracy and lowest loss, closely followed by DeepShift-Q method. The difference of training accuracy and loss between these two strategies is trivial. However, the size of uplink data transfer in case of DeepShift-Q is only around 80% (Table 1) of the size of data transfer from quantization method. Figure 3, Figure 4 and Table 2 show the same results with CNNs model.

We expect the similar results when employing these three compression methods with other datasets to train other machine learning models.



(a) Non-IID MNIST

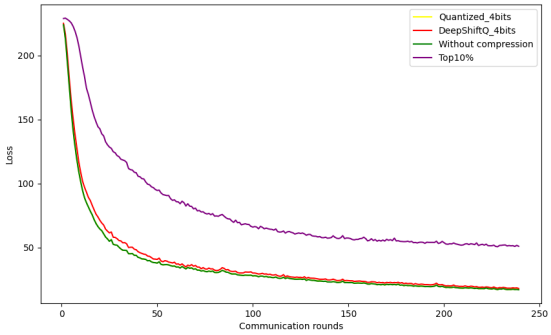


(b) Non-IID CIFAR10

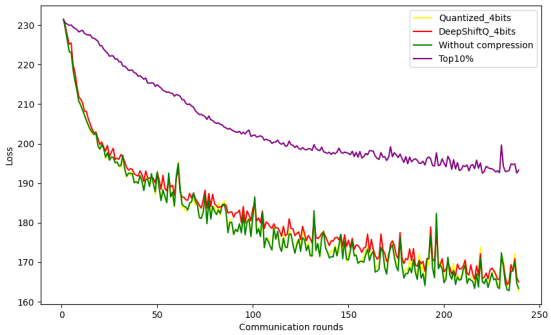
Figure 1: Training accuracy on MNIST and CIFAR10 datasets with MLP model

Table 2: Uplink data transfer for training with CNN model

Compression strategy	Dataset	
	MNIST (39 communication rounds)	CIFAR10 (366 communication rounds)
Top-10	1.33GB	1.34GB
Quantization (4 bits)	5.54GB	5.56GB
DeepShift-Q (4 bits)	4.43GB	4.43GB



(a) Non-IID MNIST



(b) Non-IID CIFAR10

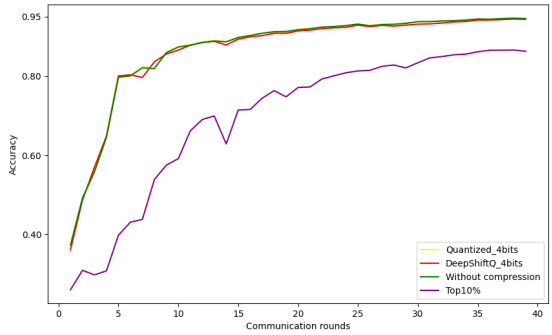
Figure 2: Training loss on MNIST and CIFAR10 datasets with MLP model

5 Conclusion

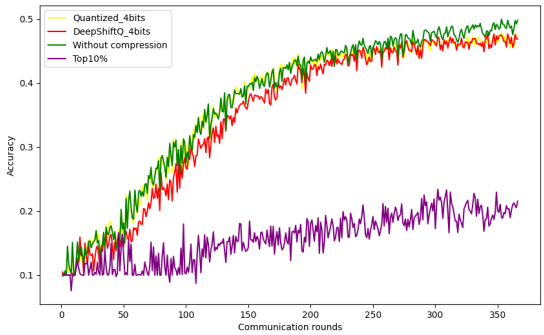
While size of data transfer with using DeepShift-Q compression method is four fifth of quantization method and the difference in training accuracy/loss between two methods is trivial, DeepShift-Q has a decent performance with non-IID data under FL settings. Due to bottlenecks of computational resource and limited network bandwidth, training machine learning models on edge device's local non-IID data to priority data privacy can be challenging. In order to facilitate communication hurdle in FL, many compression methods have been proposed. Our study contributes to this area by showing that DeepShift-Q can be utilized as a efficient compression method with minimal accuracy loss to improve communication efficiency in FL.

References

[1] S. Albawi, O. Bayat, S. Al-Azawi, and O. N. Ucan. Social touch gesture recognition using convolutional neural network. *Computational Intelligence and Neuroscience*, 2018, 2017.



(a) Non-IID MNIST



(b) Non-IID CIFAR10

Figure 3: Training accuracy on MNIST and CIFAR10 datasets with CNNs model

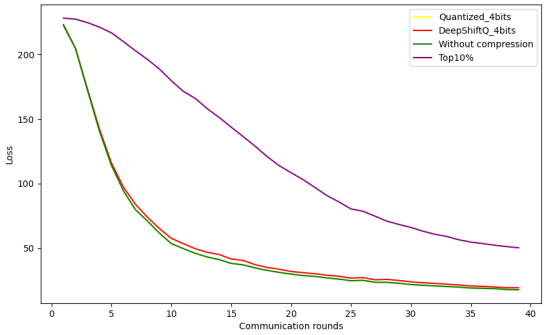
[2] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

[3] D. Basu, D. Data, C. Karakus, and S. Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. *Advances in Neural Information Processing Systems*, 32, 2019.

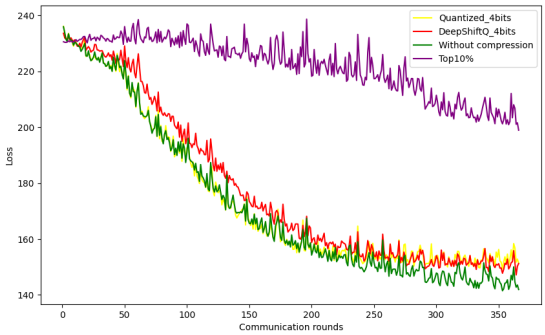
[4] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

[5] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[6] W. H. Delashmit, M. T. Manry, et al. Recent developments in multilayer perceptron neural networks. In *Proceedings of the seventh annual memphis area engineering and science conference, MAESC*, pages 1–15, 2005.



(a) Non-IID MNIST



(b) Non-IID CIFAR10

Figure 4: Training loss on MNIST and CIFAR10 datasets with CNNs model

[7] H. Eichner, T. Koren, B. McMahan, N. Srebro, and K. Talwar. Semi-cyclic stochastic gradient descent. In *International Conference on Machine Learning*, pages 1764–1773. PMLR, 2019.

[8] M. Elhoushi, Z. Chen, F. Shafiq, Y. H. Tian, and J. Y. Li. Deepshift: Towards multiplication-less neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2359–2368, 2021.

[9] J. Hamer, M. Mohri, and A. T. Suresh. Fedboost: A communication-efficient algorithm for federated learning. In *International Conference on Machine Learning*, pages 3973–3983. PMLR, 2020.

[10] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[11] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[12] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*, 2017.

[15] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin. A state-of-the-art survey on solving non-iid data in federated learning. *Future Generation Computer Systems*, 135:244–258, 2022.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[17] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012.

[18] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020.

[19] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[20] A. Storkey et al. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 30(3-28):6, 2009.

[21] P. Sun, W. Feng, R. Han, S. Yan, and Y. Wen. Optimizing network performance for distributed dnn training on gpu clusters: Imagenet/alexnet training in 1.5 minutes. *arXiv preprint arXiv:1902.06855*, 2019.

[22] A. T. Suresh, X. Y. Felix, S. Kumar, and H. B. McMahan. Distributed mean estimation with limited communication. In *International conference on machine learning*, pages 3329–3337. PMLR, 2017.

[23] M. A. Tanner and W. H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the American statistical Association*, 82(398):528–540, 1987.

[24] K. Wang, R. Mathews, C. Kiddon, H. Eichner, F. Beaufays, and D. Ramage. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

[25] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[26] G. I. Webb, L. K. Lee, B. Goethals, and F. Petitjean. Analyzing concept drift and shift from sample data. *Data Mining and Knowledge Discovery*, 32:1179–1199, 2018.

[27] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23:69–101, 1996.