Communication-Efficient Federated Learning via Optimal Client Sampling

Mónica Ribero [1] Haris Vikalo [1]

Abstract

Federated learning is a private and efficient framework for learning models in settings where data is distributed across many clients. Due to interactive nature of the training process, frequent communication of large amounts of information is required between the clients and the central server which aggregates local models. We propose a novel, simple and efficient way of updating the central model in communication-constrained settings by determining the optimal client sampling policy. In particular, modeling the progression of clients' weights by an Ornstein-Uhlenbeck process allows us to derive the optimal sampling strategy for selecting a subset of clients with significant weight updates. The central server then collects local models from only the selected clients and subsequently aggregates them. We propose two client sampling strategies and test them on two federated learning benchmark tests, namely, a classification task on EMNIST and a realistic language modeling task using the Stackoverflow dataset. The results show that the proposed framework provides significant reduction in communication while maintaining competitive or achieving superior performance compared to baseline. Our methods introduce a new line of communication strategies orthogonal to the existing user-local methods such as quantization or sparsification, thus complementing rather than aiming to replace them.

1. Introduction

Federated learning provides a private and efficient way of learning machine learning models when the data is distributed across many clients (McMahan et al., 2017; Kairouz et al., 2019). In `FedAvg`, the baseline FL procedure proposed in (McMahan et al., 2017), a server distributes an initial model to clients who independently update the model using their local training data; these updates are aggregated by the server which broadcasts a new global model to the clients and selects a subset of them to start a new round of local training; the procedure is repeated until convergence. Since clients communicate only their models to the server, federated learning offers data security that can be further strengthened using privacy mechanisms including those that provide differential privacy guarantees (Abadi et al., 2016; Wu et al., 2017; McMahan et al., 2018).

A major open challenge in federated learning is to reduce communication rates, considering the number of clients N can be in the order of millions, and a machine learning model can have several million parameters, leading to a communication bottleneck in real applications. One approach is to reducing each client communication budget by compressing the model (Tang et al., 2019; Konečnỳ et al., 2016; Suresh et al., 2017; Konečnỳ & Richtárik, 2018; Alistarh et al., 2017; Horvath et al., 2019; Caldas et al., 2018). Though, many clients overall updates could be dispensable. Yet, in current algorithms, the communication budget remains dependent on the fixed number of clients participating at each round. We propose a novel orthogonal communication reduction approach that consists in transmitting only relevant clients' updates.

Concretely, we model each user weights vector as an Ornstein - Uhlenbeck processes (OU) stochastic process during Stochastic Gradient Descent (SGD) and formulate the decision of sending or not the updates to the server as the one of sampling or not the process to maximize the quality of estimation, and minimize the number of samples. The optimal strategy is a simple threshold on the update's norm. We test our method on the EMNIST dataset and the Stackoverflow dataset and show communication reductions without sacrificing accuracy.

2. Formulation

OU process definition An Ornstein-Uhlenbeck processes (OU) is a stationary Gauss-Markov processes that, over time, tends to drift towards a mean value. Formally it can

[1] Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, Texas. Correspondence to: Mónica Ribero <mribero@utexas.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020. Copyright 2020 by the author(s).

be described by the stochastic differential equation

$$d\theta_t = \lambda(\mu - \theta_t)dt + \sigma dW_t \tag{1}$$

where W_t is a standard Wiener process.

SGD as an OU process Consider the loss function $\mathcal{L}(\theta; X) = \sum_{i=1}^{N} \ell_i(\theta)$, where X is a dataset with N samples and $\ell_i(\theta)$ is the loss of point $x_i \in X$ for $i = 1, \ldots, N$. In gradient descent, $\mathcal{L}$ is minimized by evaluating in each iteration an approximation of the gradient using a mini-batch $\mathcal{S} \subseteq X$ of the data. In particular,

$$\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} g_i(\theta).$$

The following observations and assumptions are commonly encountered in literature (see, e.g., (Mandt et al., 2016)).

Observation 1: The central limit theorem implies that $\frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} g_i(\theta) \to \mathcal{N}(g(\theta), B(\theta)B(\theta)^T)$, where $g(\theta)$ denotes the full gradient and $B(\theta)B(\theta)^T$ is the corresponding covariance matrix.

Assumption 1: When θ approaches a stationary value, $B(\theta) = B$ is constant (Mandt et al., 2016).

Assumption 2: The iterates θ_t lie in a region where the loss can be approximated by a quadratic form $\mathcal{L}(\theta) = \frac{1}{2}\theta^T A \theta$ (readily justified in the case of smooth loss functions), and the process reaches a quasi-stationary distribution around a local minimum.

Predicated on the above, the discrete process

$$\Delta\theta = \theta_{t+1} - \theta_t \approx -\eta g(\theta) - \sqrt{\frac{\eta}{N}} B \mathcal{N}(0, \eta I)$$

can be interpreted as obtained by discretizing the OU process

$$d\theta_t = -g(\theta)dt + \sqrt{\frac{\eta}{N}} B dW_t = -A\theta_t dt + \sqrt{\frac{\eta}{N}} B dW_t.$$

2.1. Thresholding: optimal sampling of OU processes

Rate-constrained sampling of stochastic processes has been widely studied in literature, primarily in the context of communications and control (Imer & Basar, 2005; Bommannavar & Başar, 2008; Nayyar et al., 2013; Rabi et al., 2012; Nar & Başar, 2014; Sun et al., 2017; Ornee & Sun, 2019; Guo & Kostina, 2020). It has been proved analytically and numerically that thresholding strategies outperform deterministic sampling (Nar & Başar, 2014; Sun et al., 2017; Ornee & Sun, 2019) for estimating stochastic processes. Specifically, the following thresholding strategy outperforms deterministic sampling; the threshold is derived from a frequency constraint:

$$\tau = \inf\{t \geq 0 : |\theta_t - \mathbb{E}[\theta_t|\theta_0]| > \gamma\}$$

To establish a connection to federated learning, we recall the arguments from the previous subsection and note that in each round t of an FL procedure, client i ”observes” a partial sample path of an OU process (i.e., the progression of its weights during local training); the sample path starts from a point (i.e., model weights) broadcasted by the server at the beginning of a training round. Invoking the above sampling optimality results, we propose to schedule transmission of updates if the norm of the difference between a locally updated and the previously broadcasted model exceeds a judiciously selected threshold.

2.2. Selecting the threshold

To summarize, we propose a scheme where client i is selected if the norm of its weights update $\Delta_t^i := \theta_{t+1}^i - \theta_t$ exceeds threshold γ, i.e., if $\|\Delta_t^i\|_2 > \gamma$; ideally, thresholding reduces communication without incurring significant accuracy loss compared to baseline. In this section we explore two thresholding strategies listed below.

1. Fixed threshold (**FT**): The server provides a fixed threshold γ to all clients before training starts. During training, and before sending local updates, each client tests $\|\Delta_t^i\|_2 > \gamma$. The clients for which this holds true send their updates; the others communicate only the size of their local data set (to be used in computation of weights in the model aggregation step).

2. Adaptive threshold (**AT**): At each iteration t, all clients report to the server $\|\Delta_t^i\|_2$ (just one float number per client); the server in turn computes the empirical mean μ_t and variance σ^2 of the received norms, and sends back to the clients $\gamma_t = \mu_t - \sigma_t$ to use as the threshold.

To summarize, N clients are selected at the beginning of round t. The server broadcasts model parameters θ_t, and the selected clients locally performs SGD with mini-batches of size B for E epochs. Then, following a communication threshold rule R (selected among those described in Sec 2.2) that evaluates if the local model update Δ_t^i exceeds the threshold, each client *locally* decides whether to communicate its updates or not, and transmits either the model updates θ_{t+1}^k or a negative-acknowledgement message, respectively. In both cases, the client sends its training data size n_i to enable weighing according to $w_i = \frac{n_i}{\sum_j n_j}$. Finally, following the optimal sampling strategy, the server estimates each client's parameters as

$$\hat{\theta}_{t+1}^k = \begin{cases} \theta_{t+1}^k, & \text{if client sent updates} \\ e^{-\lambda\Delta t}\theta_t + (1 - e^{-\lambda})\mu, & \text{otherwise.} \end{cases} \tag{2}$$

Parameters λ and μ are estimated via least-squares. Note that the server's computationally cheap alternative to estimation is to simply reuse the client's model from the previous round, i.e., to set $\hat{\theta}^i_{t+1} = \theta_t$; in our experiments, we observed that this alternative consistently provides high accuracy. Finally, the server computes a new model according to $\theta_{t+1} = \sum_{i=1}^{N} w_i \hat{\theta}^i_t$

3. Results and discussion

We test our methods on a classification task on EMNIST and a realistic language modeling task using the Stackoverflow dataset. The EMNIST dataset, a reprocessed version of the original NIST dataset where each image is linked to its original writer. The dataset counts with 3843 users. The Stackoverflow is a language modelling dataset with questions and answers collected from 342,477 unique users. We implemented client selection strategies defined in Sec 2.2 and use a grid search to determine the value of $\gamma = 0.5$ in FT. We compare them with the baseline `FedAvg` with no client subselection, and with a random sampling strategy that drops clients at random to match the communication rate of the best approach (FT or AT). We train a convolutional neural network (cnn) with two 5x5 convolution layers, with 32 and 64 channels respectively, and interleaved with 2x2 max-pooling, a dense layer with 512 neurons with ReLU activation and a 10 unit softmax output layer. For Stackoverflow we train train a recursive neural network that first embeds words into a 96-dimensional space, followed by an LSTM and finally a dense layer.

At each round, 50 clients are uniformly selected to update the model. Each client locally trains for $E = 20$ epochs on EMNIST and $E = 1$ epochs for Stackoverflow, using SGD. We train the each model for 100 rounds.

A simplified summary is in Table 1. On EMNIST, a fixed threshold is the best strategy, achieving 95 % accuracy using only 19 % of the communication. However, finding the threshold 0.5 requires tuning. Adaptive strategies overcome this difficulty and still provide comparable performance and communication savings. On Stackoverflow, both methods achieve comparable performance to the baseline with half of the communication. Fig 2 shows the cummulative communication spent by each method. In all cases, the random strategy decreases the communication but also has a considerable drop in the performance. Fig 1 shows the efficacy of our method where our methods achieve significantly higher accuracy per amount of communication.

Our approaches prove efficient and can be combined with compression approaches to obtain even lower communication rates. Determining a simple efficient optimal scheme for selecting this threshold remains an open question.

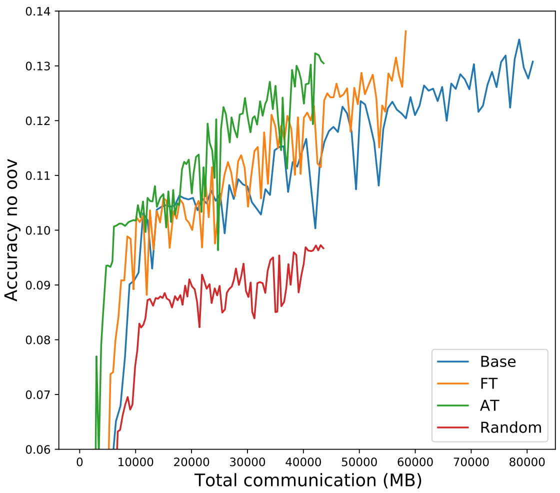

Figure 1. Performance for different client selection strategies.

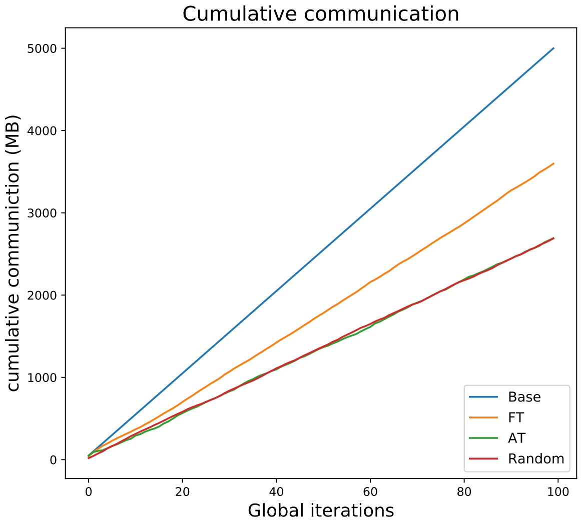

Figure 2. Communication for different client selection strategies.

Table 1. Results for EMNIST and Stackoverflow

	Accuracy		**Overall comm.**(GB)	
Dataset	EMNIST	Stack	EMNIST	Stack
Baseline	**97.5%**	**13.07 %**	**33.27**	81.0
FT	95.4%	13.63 %	**6.3** (19%)	58.3 (72%)
AT	**97.4%**	13.04 %	27.9 (83%)	**43.6** (54%)
Random	94.2 %	9.67 %	**6.3** (19%)	**43.6** (54 %)

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. pp. 308–318. ACM, 2016.

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pp. 1709–1720, 2017.

Bommannavar, P. A. and Başar, T. Optimal estimation over channels with limits on usage. *IFAC Proceedings Volumes*, 41(2):6632–6637, 2008.

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

Guo, N. and Kostina, V. Optimal causal rate-constrained sampling for a class of continuous markov processes. *arXiv preprint arXiv:2002.01581*, 2020.

Horvath, S., Ho, C.-Y., Horvath, L., Sahu, A. N., Canini, M., and Richtarik, P. Natural compression for distributed deep learning. *arXiv preprint arXiv:1905.10988*, 2019.

Imer, O. C. and Basar, T. Optimal estimation with limited measurements. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 1029–1034. IEEE, 2005.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

Konečnỳ, J. and Richtárik, P. Randomized distributed mean estimation: Accuracy vs. communication. *Frontiers in Applied Mathematics and Statistics*, 4:62, 2018.

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

Mandt, S., Hoffman, M., and Blei, D. A variational analysis of stochastic gradient algorithms. In *International conference on machine learning*, pp. 354–363, 2016.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J. (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

McMahan, H. B., Andrew, G., Erlingsson, U., Chien, S., Mironov, I., Papernot, N., and Kairouz, P. A general approach to adding differential privacy to iterative training procedures. *arXiv preprint arXiv:1812.06210*, 2018.

Nar, K. and Başar, T. Sampling multidimensional wiener processes. In *53rd IEEE Conference on Decision and Control*, pp. 3426–3431, Dec 2014. doi: 10.1109/CDC.2014.7039920.

Nayyar, A., Başar, T., Teneketzis, D., and Veeravalli, V. V. Optimal strategies for communication and remote estimation with an energy harvesting sensor. *IEEE Transactions on Automatic Control*, 58(9):2246–2260, 2013.

Ornee, T. Z. and Sun, Y. Sampling for remote estimation through queues: Age of information and beyond. *arXiv preprint arXiv:1902.03552*, 2019.

Rabi, M., Moustakides, G. V., and Baras, J. S. Adaptive sampling for linear state estimation. *SIAM Journal on Control and Optimization*, 50(2):672–702, 2012.

Sun, Y., Polyanskiy, Y., and Uysal-Biyikoglu, E. Remote estimation of the wiener process over a channel with random delay. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 321–325. IEEE, 2017.

Suresh, A. T., Yu, F. X., Kumar, S., and McMahan, H. B. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3329–3337. JMLR.org, 2017.

Tang, H., Lian, X., Zhang, T., and Liu, J. Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression. *arXiv preprint arXiv:1905.05957*, 2019.

Wu, X., Li, F., Kumar, A., Chaudhuri, K., Jha, S., and Naughton, J. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. pp. 1307–1322. ACM, 2017.