Bayesian SignSGD Optimizer for Federated Learning

Paulo Abelha Ferreira **Pablo Nascimento da Silva** **Vinicius Gottin**

Roberto Stelling **Tiago Calmon**

Dell Technologies, Research Office, Rio de Janeiro, RJ, Brazil
{paulo.ferreira, pablo.dasilva, vinicius.gottin, roberto.stelling, tiago.calmon}@dell.com

Abstract

Federated Learning is a distributed Machine Learning framework aimed at training a global model by sharing edge nodes' locally trained models instead of their datasets. This presents three major challenges: communication between edge nodes and the central node; heterogeneity of edge nodes (e.g. availability, computing, datasets); and security. In this paper we focus on the communication challenge, which is two-fold: decreasing the number of communication rounds; and compressing the information sent back and forth between edge nodes and the central node. Particularly, we are interested in cases where strict constraints over the allowed network traffic of gradients may apply – e.g. frequent training of predictive models for globally distributed devices. The recent success of 1-bit compressor (e.g. majority voting SIGNSGD) is promising; however, such high-compression methods are known to have slow (or problematic) convergence. We propose a Bayesian framework, named BB-SIGNSGD, encompassing 1-bit compressors for a principled and flexible choice of how much information to carry from previous communication rounds during central aggregation. We prove that majority voting SIGNSGD is a special case of our framework when particular choices are taken within it. We present results from extensive experiments in five different datasets. We show that, compared to majority voting SIGNSGD, other choices within BB-SIGNSGD support higher learning rates to achieve faster convergence, competitive even with uncompressed communication.

1 Introduction

The main components of modern machine learning applications are data and computational power [23]. More specifically, deep neural network models are known to be both data-intensive and computationally hungry. However, in some applications collecting vast amounts of data from model training may raise security and privacy concerns (e.g., mobile phones, healthcare, smart cars, storage systems, etc.). Federated Learning (FL) has emerged as a framework for addressing these concerns by performing the model training on the user device, leveraging the device's computational power, and providing strong data privacy guarantees [13, 19].

The goal of Federated Learning is to train a centralized global model while the training data remains distributed on a large number of client nodes [13]. In this context, we assume that the central node can be any machine with reasonable computational power. Training a model on a Federated Learning setting is usually done as follows. First, the central node shares an initial model (a deep neural network) with all the distributed edge nodes or workers (henceforth, these terms are used interchangeably). Next, the workers train their models using their own data (without sharing it with other workers). Then, the central node receives the updated workers' models and aggregates them

1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL 2021), Virtual Meeting.

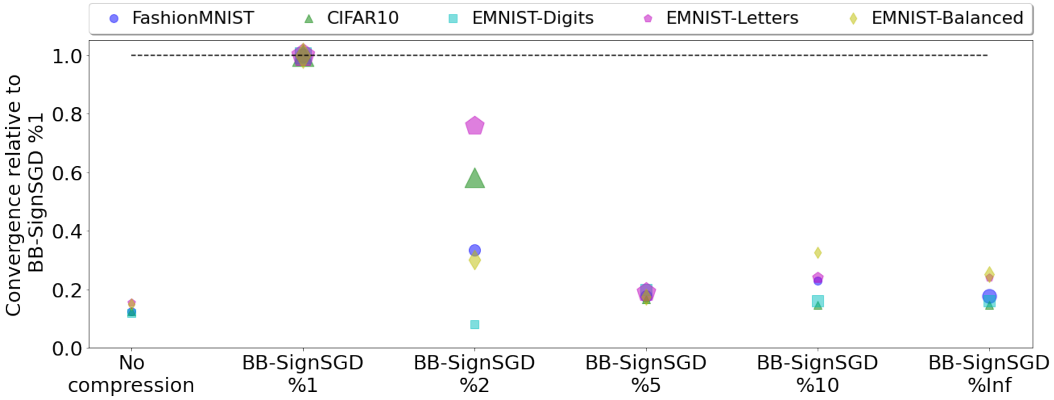

Figure 1: Converge (cycles) ratio relative to SignSGD (i.e. BB-SignSGD %1), measured as how fast the method achieved an accuracy at least 99% that of the maximum accuracy achieved by BB-SignSGD %1 (0.25 implies a method 4x faster). Size of marker indicates variance in experiments. Central learning rate $\eta_c = 5\text{e}{-3}$.

into a single central model. The central node communicates the new model to the workers, and the process repeats for multiple communication rounds until it convergences or reaches a required metric.

In practice, updating the central model involves frequently sending from the workers each model update, which generates bandwidth issues for very large models. In some settings, there may be strong limitations in the amount of communication allowed per edge device. There is a growing body of research on gradient compression and quantization to reduce communication costs without negatively impacting the convergence rate or accuracy [3, 15, 16]. One of the strongest methods for gradient compression is the federated version of SignSGD, with majority voting[1] [4, 5]. This method allows sending 1-bit per gradient component, a 32x gain as compared to a standard 32-bit floating-point representation.

In this work, we propose an aggregation step based on a Bayesian beta-Bernoulli model. The proposed Bayesian framework BB-SignSGD encompasses 1-bit compressors for a principled and flexible choice of how to incorporate information from previous communication rounds during central aggregation. We provide a proof that SignSGD is a special case of our framework when particular choices are taken within it (in Appendix A [2]). We present results from extensive experiments in five different datasets, providing empirical evidence on how variations in a single choice within our framework leads to faster convergence as compared to SignSGD, competitive even with uncompressed communication.

We validate our framework on the following computer vision datasets: CIFAR-10 [17], EMNIST [9] (*Digits*, *Letters* and *Balanced* versions) and FashionMNIST [27]. The datasets are split, i.i.d., to form the central and edge nodes' data. In terms of convergence rate, variations of BB-SignSGD consistently outperform or are leveled with SignSGD, with no harm to the model's accuracy and sometimes reaching equivalent convergence to uncompressed communication.

Figure 1 shows, for different datasets, the relative convergence ratio of all methods as relative to SignSGD. We have the no compression method, where full 32-bit precision is used during communication, and then variations of BB-SignSGD, where BB-SignSGD %1 is equivalent to SignSGD. We can see how variation BB-SignSGD %5, for instance, outperforms SignSGD on all datasets, achieving convergence to SignSGD's accuracy around 5x faster and being close to the convergence on uncompressed communication. Figure 1 is related to Table 4 in Appendix B [3].

[1] From here on we will refer to majority voting SignSGD simply as SignSGD

[2] Appendices might be separated from this file, but should be available along with it.

[3] Appendices might be separated from this file, but should be available along with it.

2 Related Work

Federated Learning Federated Learning is a growing field and most relevant to the work we present here are fronts such as robustness, convergence and compression for communication costs [13]. The standard aggregation method is FEDAVG [19], albeit with many works proposing different improvements on it [14, 18, 28], mostly focusing on heterogeneous settings.

There have also been work on global model update, such as [20] the authors present an analysis of the federated version of adaptive optimizers (Adagrad, Adam, among others), focusing on the trade-offs between communication efficiency and client data heterogeneity. Another example of improved update rules at the central node for the global model is [10]. Other interesting improvements include [25], where the authors propose a method that performs layer-wise matching.

Bayesian Federated Learning The work in [2] formulates Federated Learning as a posterior inference problem, with their aggregation method FEDPA, a generalization of FEDAVG. This is similar to our insight of treating aggregation as a posterior inference in a Bayesian way; however, FEDPA is not focused on high compression scenarios and requires computation of posterior at the edge nodes.

Other types of Bayesian perspective include, for instance [7], where the authors present a Bayesian model ensemble method combined with knowledge distillation in a teacher-student setting to arrive at a global model. Other works in model aggregation, include [29], where the authors propose a Bayesian non-parametric approach to model matching, with further improvements done in FEDMA [25] through iterative matching per layer. Like FEDMA, our method has also only a linear dependence on layer depth - in our case, only in terms of storage (i.e. beta prior parameters – see Section 4) and not communication. Finally, FEDBE [7] aims to improve aggregation robustness through Bayesian model ensemble. Similar to our work, FEDBE also strives for robustness in aggregation, but has no particular focus in high compression scenarios.

SignSGD for Federated Learning Gradient quantization in general has been proposed in Federated Learning to mitigate communication costs. Most of the work so far has focused on unbiased methods, such as PQASGD [11], D-PSGD [22], QSGD [3], TERNGRAD [26] and ATOMO [24]. In contrast to QSGD [3] and CPSGD [1], our method yields 1-bit updates, allowing for 1-bit communication from central to edge nodes.

In terms of 1-bit compression, one of the first proposals was [21], with a rich theoretical and practical ideas being developed thereafter. Theoretical and empirical studies have supported the idea that sign-based compression, despite its biased nature, can converge well, in homogeneous settings [4, 5, 6]. There have also been works studying the heterogeneous settings, with [8] adding noise to improve convergence, and [12] proposes a parameter estimation approach to improve convergence with a stochastic version of SIGNSGD that enables one to put theoretical bounds convergence.

3 Background: Federated Learning Formulation

We start by considering a commonplace Federated Learning scenario where we have a central node coordinating a number of workers (edge nodes), each with their private dataset. The workers jointly optimize a global model without sharing their private data. This optimization occurs through a number of cycles (communication rounds), each composed of two steps: edge-side training and central aggregation. The process starts by all workers sharing the same initial model and, at each new cycle, the central node syncs back to the workers an updated model. A cycle involves four main concepts: edge-side training; gradient compression (at the edge and central node); central aggregation; and edge-side model update.

Edge-side training Each worker starts with an initial global model $w_i = \bar{w}$, which it optimizes on batches of its private dataset performing SGD for a given number of steps (SGD is used as an example, but the edge nodes could perform other types of optimization). Denote by w_i and B_i the model and the private dataset at worker i respectively, then we can express edge-side training as performing SGD for K steps as

$$w_i \leftarrow w_i - \eta_w \nabla \ell(w_i, b), d \sim B_i, \text{for } k = 1, 2, ..., K \tag{1}$$

where η_w is the learning step at the edge nodes, ℓ is a loss function and b is a sample (batch) from a worker's private dataset. After performing SGD, each worker arrives at a trained model w_i.

Gradient compression After training, each worker compresses its gradients prior to sending them to the central node, where they will be decompressed, aggregated, compressed and sent back to the edge nodes. In this work we consider "pseudo-gradients" $g_i \in \mathbb{R}^{\mathbb{D}}$, with $g_i = w_i - \bar{w}$ defined as the difference of each worker's updated model to the initially shared model. We define a gradient compressor as a pair of functions $(c(\cdot), d(\cdot))$ for compression and decompression, respectively. Each worker compresses its pseudo-gradient g_i before sending them back to the central node as $g_i^c \leftarrow c(g_i)$. Please note that g_i^c will still be a D-dimensional vector (albeit possibly in a different space than the uncompressed g_i). Performing no compression simply means we have $c(\cdot) = d(\cdot)$ being the identity function.

Central aggregation After receiving the compressed pseudo-gradients g_i^c from each worker i, the central node first decompresses them as $g_i^d \leftarrow d(g_i^c)$ and then performs an aggregation step to arrive at a unique global gradient vector $\bar{g}$, which is then re-compressed to be sent back to each worker as $\bar{g}^c$, A standard approach for aggregation is FEDAVG, where we take the mean of the decompressed gradients. The complete aggregation and compression step is given by

$$\bar{g}^c \leftarrow c\left(\frac{1}{M}\sum_{i=1}^{M} d(g_i^c)\right). \tag{2}$$

Edge-side model update After each worker receives the global compressed aggregated gradients $\bar{g}^c$, it decompresses them prior to performing an update

$$w_i \leftarrow \bar{w} - \eta_c d\left(\bar{g}^c\right). \tag{3}$$

Please note that the update is performed using a common central learning rate η_c, which is not necessarily the same as the edge-side learning rate η_w used for model training (equation 1).

Federated SIGNSGD Here we consider the compressor for federated SignSGD as a pair of functions $(c(\cdot), d(\cdot))$, with $c : \mathbb{R}^{\mathbb{D}} \to \{0, 1\}^D$ and

$$c(g) \overset{def}{=} \frac{sign(g) + 1}{2}, \tag{4}$$

where we define $sign(0) = 1$ and $sign(\cdot)$ is applied element-wise on the components of g. The function $d : \{0, 1\}^D \to \{-1, 1\}^D$ is defined as

$$d(g^c) \overset{def}{=} 2g^c - 1. \tag{5}$$

SIGNSGD compresses each gradient from its (usual) float-point representation down to a single bit per gradient. This binary representation is then converted into a sign $\{-1, +1\}$ at the central node through the $d(\cdot)$ function. The central aggregation of these decompressed sign gradients is done through FEDAVG, $\frac{1}{M}\sum_{i=1}^{M} d(g^c)$, leading to a mean vector that does not necessarily belong to $\{-1, +1\}^D$. The aggregation can be considered a majority voting [5]: it is easy to see that the sign of the aggregated mean yields the same results as choosing gradients through majority voting across workers.

4 BB-SIGNSGD

We propose to view SIGNSGD through a Bayesian perspective. More specifically, we provide a reinterpretation of the SIGNSGD compressor for Federated Learning by assimilating it into a beta-Bernoulli probabilistic model. We name our proposed framework BB-SIGNSGD, where we make use of the same compressor functions, but perform a different aggregation step.

Bayesian beta-Bernoulli Interpretation The central node has a Bayesian beta-Bernoulli model, where the compressed gradients $g_i^c \in \{0,1\}^D$, one per each of the M workers, are interpreted to be M observations of a D-dimensional binary random vector $\mathsf{g} = [\mathsf{g}_1, \mathsf{g}_2, ..., \mathsf{g}_\mathsf{D}]$, with independent components g_j, each coming from a separate Bernoulli distribution

$$\mathsf{g}_\mathsf{j} \sim \text{Bern}\left(\mathsf{g}_\mathsf{j}; \theta_\mathbf{j}\right) \tag{6}$$

The θ parameter in the Bernoulli distribution represents the bias of each binary gradient component g_j, and each θ_j is drawn from a separate beta distribution

$$\theta_j \sim \text{beta}\left(\theta_j; \alpha_\mathbf{j}, \beta_\mathbf{j}\right). \tag{7}$$

We then have the Bayesian update for a beta posterior inference over each gradient bias parameter θ_i

$$\text{beta}\left(\theta_j\right) \propto \text{Bern}\left(\mathsf{g}_\mathsf{j}|\theta_\mathbf{j}\right) \text{beta}\left(\theta_j|\alpha_\mathbf{j}, \beta_\mathbf{j}\right) \tag{8}$$

Bayesian central aggregation The Bayesian update runs at every cycle, effectively updating each beta posterior parameter θ_j; using as prior, the posterior from the previous cycle. We also envision the possibility of not using the previous posterior as prior at any given cycle - we call this "resetting" the prior. We consider three decisions to be made at the central node in order to choose:

1. The prior values to use for α_j, β_j (at each cycle)
2. θ_j, from the updated beta posterior (e.g. sampling, mean, mode...)
3. g_j, from a Bernoulli distribution with the chosen θ_j (e.g. sampling, mean, mode...)

After making these three choices, we arrive at gradient components g_j. Please note that these are not necessarily binary values. For instance, if our third choice is the expected value of g_j, it is not necessarily true that $\mathbb{E}[\mathsf{g}_\mathsf{j}; \theta_j] \in \{0,1\}$, albeit it will be in the $[0,1]$ range. Therefore, we first apply $d(\cdot)$ to get each gradient component g_j in the $[-1,1]$ range, and then apply $c(\cdot)$ to compress it back to a binary value. Finally, please also note that in this beta-Bernoulli framework, there is no aggregation (such as FEDAVG) being applied across workers; the Bayesian model already yields a single gradient $\bar{g}_j$ per component. Formally, $\bar{g}_j^c = c(d(g_j))$.

We provide a naming scheme BB-SIGNSGD %* where we fix choices (2) and (3) to be: (2) θ_j as the mode of the beta posterior; and (3) $g_j = \mathbb{E}[\mathsf{g}_\mathsf{j}; \theta_j]$. We then vary the cycle at which we effectively "reset" the prior back to uniform. BB-SIGNSGD %5 means resetting the prior every 5 cycles; and BB-SIGNSGD %Inf means we never reset the prior.

SIGNSGD There are three particular choices we make on the BB-SIGNSGD aggregation that leads us to SIGNSGD. These are (1) a uniform prior $\alpha_j = \beta_j = 1$ at every cycle; (2) θ_j as the mode of the beta posterior; and (3) $g_j = \mathbb{E}[\mathsf{g}_\mathsf{j}; \theta_j]$ as the expected value of the updated Bernoulli distribution. We name this instance of our method BB-SIGNSGD %1. We provide a proof of the equivalence of BB-SIGNSGD %1 and SIGNSGD in Appendix A [4].

5 Experiments

In this paper, we focus on exploring the first decision during central aggregation, i.e., the choice of prior parameters α_i, β_i for the beta distribution at each cycle. Particularly, we investigate how a "resetting" of the prior to uniform at different cycles affects the convergence rate. The suffix $\%n$ indicates that we are resetting the prior every n cycle(s); thus $\%Inf$ means that we are *never* resetting the prior. All BB-SIGNSGD variations are making the same second and third choices: (2) θ_j as the

[4] Appendices might be separated from this file, but should be available along with it.

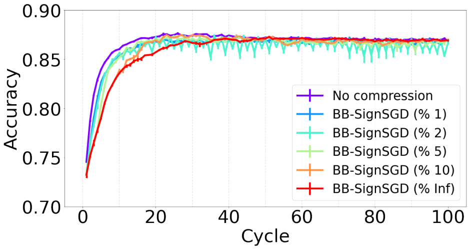

Figure 2: FashionMNIST, $\eta_c = 2\text{e}{-3}$. It presents stable learning, however convergence is twice as long.

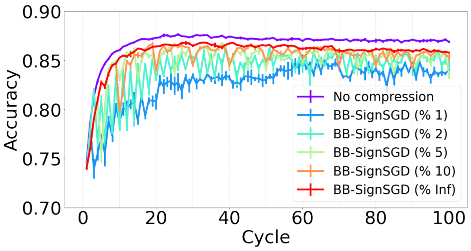

Figure 3: FashionMNIST, $\eta_c = 5\text{e}{-3}$. It shows that BB-SIGNSGD with various parameter choices overperformed SignSGD (blue line) in terms of convergence.

mode of the beta posterior; and (3) $g_j = \mathbb{E}[\text{g}_\text{j}; \theta_j]$ as the expected value of the updated Bernoulli distribution. Finally, we also investigate different values for the central learning rate η_c.

We validate our framework on the following computer vision datasets: CIFAR-10 [17], EMNIST [9] and Fashion-MNIST [27]. The datasets are split, i.i.d., to form the workers' training data (90%) and the central node validation data (10%). All results are reported relative to the central node's validation data. We perform experiments on our own developed simulator built with a gradient compression experimentation focus. All experiments were done with a single central node, 10 edge nodes and 100 max communication cycles. Training at the edge nodes was done with a batch size of 64 for 1 epoch using the Adam optimizer with default learning rate of $\eta_w = 1\text{e}{-3}$. Finally, we focus on convergence in this section since different methods achieved very similar maximum accuracy after 100 cycles - tables for accuracy across all datasets, methods and learning rates can be found in Appendix B [5].

Figures 2 and 3 show the accuracy evolution for different methods for the FashionMNIST dataset. Figure 2 shows the evolution for a central learning rate $\eta_c = 2\text{e}{-3}$; and Figure 3 for $\eta_c = 5\text{e}{-3}$. We can see how a smaller central learning rate stabilizes convergence for all methods; nonetheless, even with a larger learning rate of $\eta_c = 5\text{e}{-3}$, BB-SIGNSGD variations remain relatively stable and achieve faster convergence than BB-SIGNSGD %1. In Appendix B [6], we show convergence and accuracy tables as well as figures for all the different datasets and learning rate variations.

Tables 1 and 2 show the relative convergence of all methods to BB-SIGNSGD %1 (i.e. SIGNSGD). To calculate this, we first find the maximum accuracy achieved by BB-SIGNSGD %1 across all 100 cycles. Then, for any particular method's accuracy evolution, we report the cycle at which it achieves at least 99% of BB-SIGNSGD %1's maximum accuracy. Please note that, for the case of BB-SIGNSGD %1 itself, we report the cycle at which it achieves at least 99% of its own maximum accuracy. We report the mean and standard deviation for three experiments running for three different starting seeds.

In Table 1, we show the convergence for all methods and datasets, with the central learning rate set at $\eta_c = 2\text{e}{-3}$. We use this learning for a fair comparison, since this is the one with which BB-SIGNSGD %1 performed the best. Please note how BB-SIGNSGD %5 surpasses BB-SIGNSGD on all datasets but EMNIST-Digits, on which it achieves a comparable convergence.

Table 2 show convergence of all methods for a particular dataset FashionMNIST, but now varying the central learning rate. We can see from the results how BB-SIGNSGD %5 levels with or surpasses BB-SIGNSGD %1 with all learning rates. Particularly, we note how BB-SIGNSGD %5 achieves a lower convergence cycle (10 ± 1, with $\eta_c = 5\text{e}{-3}$) as compared to BB-SIGNSGD %1 using its best learning rate (14 ± 2, with $\eta_c = 2\text{e}{-3}$).

From both Tables 1 and 2, we can see how even for BB-SIGNSGD %1's best learning rate of $\eta_c = 2\text{e}{-3}$, we find other BB-SIGNSGD variations that performed equivalently or better (with special attention to BB-SIGNSGD %5). Furthermore, as we increase the central learning rate, BB-SIGNSGD %1 destabilizes its convergence, and other Bayesian variations maintain stability (again,

[5] Appendices might be separated from this file, but should be available along with it.

[6] Appendices might be separated from this file, but should be available along with it.

Table 1: Convergence relative to BB-SIGNSGD %1 (i.e. SIGNSGD), with central learning rate $\eta_c = 2\text{e}{-3}$. We report the cycle at which the given method achieves at least 99% of BB-SIGNSGD %1's maximum accuracy. Values are mean and standard deviation for three different starting seeds.

		BB-SIGNSGD				
Dataset	No compr.	%1	%2	%5	%10	%Inf
FashionMNIST	12 ± 1	14 ± 2	16 ± 2	$\mathbf{13 \pm 2}$	17 ± 2	24 ± 2
CIFAR10	11 ± 1	17 ± 2	$\mathbf{13 \pm 0}$	$\mathbf{13 \pm 1}$	14 ± 0	18 ± 2
EMNIST-Digits	4 ± 0	$\mathbf{5 \pm 1}$	$\mathbf{5 \pm 1}$	6 ± 0	9 ± 0	10 ± 1
EMNIST-Letters	11 ± 0	15 ± 2	15 ± 2	$\mathbf{13 \pm 1}$	15 ± 2	31 ± 4
EMNIST-Balanced	11 ± 1	17 ± 2	16 ± 2	$\mathbf{15 \pm 1}$	17 ± 1	36 ± 6

Table 2: Convergence relative to BB-SIGNSGD %1 (i.e. SIGNSGD), with varying central learning rate for FashionMNIST. We report the cycle at which the given method achieves at least 99% of BB-SIGNSGD %1's maximum accuracy. Values are mean and standard deviation for three different starting seeds.

		BB-SIGNSGD				
η_c	No compr.	%1	%2	%5	%10	%Inf
$1\text{e}{-3}$	13 ± 0	$\mathbf{22 \pm 2}$	23 ± 2	$\mathbf{22 \pm 2}$	24 ± 0	48 ± 1
$2\text{e}{-3}$	12 ± 1	14 ± 2	16 ± 2	$\mathbf{13 \pm 2}$	17 ± 2	24 ± 2
$5\text{e}{-3}$	7 ± 0	57 ± 2	19 ± 1	$\mathbf{10 \pm 1}$	13 ± 0	$\mathbf{10 \pm 2}$

with special attention to BB-SIGNSGD %5). This is true for other datasets, as shown in Appendix B [7].

6 Discussion and Conclusion

In this work we focus on scenarios with limited communication bandwidth, where gradient compressor quantization provides a feasible solution to the Federated Learning problem. We propose a novel Bayesian framework for 1-bit compression, named BB-SIGNSGD, where we treat central aggregation as an inference problem, with model weights from edge nodes taking the role of observations. More concretely, we model each gradient as a Bernoulli random variable, and perform posterior inference on a beta-Bernoulli model to arrive at aggregated gradients. Furthermore, we prove that SignSGD is a special case of BB-SIGNSGD when standard choices are taken within the framework: 1) a uniform prior reset at every cycle; 2) mode of the beta posterior; and 3) expected value of the gradient. Our framework incurs only a minor linear cost of storing beta prior parameters (two real numbers per model weight) - and no extra cost for BB-SIGNSGD %1, since the prior is reset at every cycle.

We validate our framework on five computer vision datasets. In our experiments we vary two aspects: number of cycles before resetting the beta prior; and the central learning rate. We find that BB-SignSGD with higher reset cycles outperforms SIGNSGD, achieving the same accuracy at much lower cycles, and in some cases being competitive with no gradient compression. The key insight is that delayed resetting of the prior within our framework allows for increased learning rate without disrupting model convergence and thus allowing for faster convergence.

References

[1] Naman Agarwal, Ananda Theertha Suresh, Felix Yu, Sanjiv Kumar, and H Brendan Mcmahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7575–7586, 2018.

[7] Appendices might be separated from this file, but should be available along with it.

[2] Maruan Al-Shedivat, Jennifer Gillenwater, Eric Xing, and Afshin Rostamizadeh. Federated learning via posterior averaging: A new perspective and practical algorithms. In *International Conference on Learning Representations (ICLR)*, 2021.

[3] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 30:1709–1720, 2017.

[4] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018.

[5] Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd with majority vote is communication efficient and byzantine fault tolerant. In *Seventh International Conference on Learning Representations (ICLR)*, 2019.

[6] David Carlson, Ya-Ping Hsieh, Edo Collins, Lawrence Carin, and Volkan Cevher. Stochastic spectral descent for discrete graphical models. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):296–311, 2015.

[7] Hong-You Chen and Wei-Lun Chao. Fedbe: Making bayesian model ensemble applicable to federated learning. In *Advances in Neural Information Processing Systems*, International Workshop on Scalability, Privacy, and Security in Federated Learning, 2020.

[8] Xiangyi Chen, Tiancong Chen, Haoran Sun, Zhiwei Steven Wu, and Mingyi Hong. Distributed training with heterogeneous data: Bridging median-and mean-based algorithms. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[9] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[10] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *Neurips Workshop on Federated Learning*, 2019.

[11] Peng Jiang and Gagan Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2530–2541, 2018.

[12] Richeng Jin, Yufan Huang, Xiaofan He, Tianfu Wu, and Huaiyu Dai. Stochastic-sign sgd for federated learning with theoretical guarantees. *arXiv preprint arXiv:2002.10940*, 2020.

[13] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[15] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[16] Jakub Konečnỳ and Peter Richtárik. Randomized distributed mean estimation: Accuracy vs. communication. *Frontiers in Applied Mathematics and Statistics*, 4:62, 2018.

[17] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

[18] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smithy. Feddane: A federated newton-type method. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 1227–1231. IEEE, 2019.

[19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[20] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *Ninth International Conference on Learning Representations, ICLR*, 2021.

[21] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth Annual Conference of the International Speech Communication Association*. Citeseer, 2014.

[22] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7663–7673, 2018.

[23] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

[24] Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, pages 9871–9882, 2018.

[25] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

[26] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. *NIPS'17: Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

[27] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[28] Xin Yao, Tianchi Huang, Rui-Xiao Zhang, Ruiyu Li, and Lifeng Sun. Federated learning with unbiased gradient aggregation and controllable meta updating. *Workshop on Federated Learning for Data Privacy and Confidentiality (FL - NeurIPS 2019, in Conjunction with NeurIPS 2019)*, 2019.

[29] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261. PMLR, 2019.