FedCD: Improving Computational Efficiency in non-IID Federated Learning

Kavya Kopparapu
Harvard University
Cambridge, USA
kavyakopparapu@college.harvard.edu

Jazz Zhao
Harvard University
Cambridge, USA
jzhao@college.harvard.edu

Eric Lin
Harvard University
Cambridge, USA
eric_lin@college.harvard.edu

***Abstract*—Over the past few years, federated learning has become an integral part of our lives by enabling many daily used as well as high stakes applications. However, real-world data rarely is identically and independently distributed (iid) across edge devices. In addition, the high communication cost between edge devices and the central server and the low storage capacity for the edge devices prevent it from being implemented more broadly. We develop an approach, FedCD, to both increase performance and decrease communication costs for federated learning on non-iid data and experimentally verify the soundness of our approach, which outperforms baselines.**

***Index Terms*—federated learning, non-iid, compression**

I. Introduction

Machine learning methods have already proven themselves to be accurate predictors in many areas of our lives, ranging from day-to-day applications such as digital assistants powered by Natural Language processing software [1] to high stakes domains such as recommendations for certain treatments in healthcare [2].

However, machine learning methods achieve highest success when they have large amounts of data to train on. Yet in many important domains such as healthcare, data is subject to strict privacy constraints preventing direct access of local data. For example, the guidelines put forward by the EU's General Data Protection Regulation (GDPR) [3] requires actors that other persons' personal data to be clear about their intentions to do so and have a lawful basis (oftentimes consent), plus they are subject to additional requirements when sharing data internationally.

Federated learning is one way in which this issue is addressed. In particular, it differs from standard machine learning approaches as it allows multiple edge devices to learn a shared global model without having to reveal their data to the central learner. Instead, each device trains the global model locally on its own data and sends an update to the central learner, which is averaged with other devices' updates to preserve each device's privacy. This allows central models to train on extremely privacy-sensitive data, such as data about individual persons' health [4].

However, the gold standard federated learning approach, FedAvg, and other more recently-developed approaches for data-conscious, perform poorly when data is not independent and identically (iid) distributed across devices, leading to significant oscillations between rounds of training before convergence as devices send conflicting updates based on their own data distribution. By cloning the global model into multiple smaller more specialized global models, we hope to both reduce the time to convergence without negatively impacting accuracy, and increase compression for each model.

Harvard CS 242: Computing at Scale. Submitted on May 11, 2020.

II. Problem to solve

Oftentimes devices fit in the mold of one of many archetypes, where an archetype describes a specific group of non-iid data. Previously proposed learning schemes compromise on a model fine-tuned to perform well on a specific archetype in favor of a single model that works well for all archetypes. We propose Federated Cloning and Deletion, or FedCD, a learning scheme that results in tailored models for each archetype while allowing federated learning from other edge devices through iterative cloning of models at specified milestones, adaptive updating of a highly-ranked subset of global models, and deletion of poor-performance models. We argue that by using the proposed efficient learning scheme, we can decrease the size of the model to be trained and the number of times edge devices communicate with the central learner.

III. Background, motivation, prior work

A. Background and motivation

In many real-world scenarios for learning, data may not be shared outside of an individual device or outside a circle of trusted devices. Edge devices often see bottlenecks in communication and compression, as on-device space constraints model size, and communication between the global model and the local devices can be costly, as the global model has to wait for a critical number of devices to push updates.

This in turn opens up many opportunities for improvements in compression and model efficiency, since currently, it is typical to see accuracy oscillations for non-iid data: Given a single global model, devices from different archetypes send potentially conflicting updates to the same model.

Our goal is to avoid this by using multiple more specialized models.

B. Prior Work

Reference [5] is a review about federated learning work that was published last year, which gives a good survey on previous work on compression and efficiency in federated learning.

In contrast, [6] describe federated distillation, a framework in which every device is a student that learns a model from the mean result of the other devices. To decrease communication, a vector of normalized output is globally stored for every input label (averaged from the average output of each device) which acts as the teacher. Otherwise, data would either have to be shared or there would be a significant increase of information transmission for every model batch update.

Reference [7] is a seminal paper on gradient compression which includes quantization as well as other methods. They discuss sketched updates, compressing the entire model with quantization, and structured elimination of parts of the model via weight pruning.

Lastly, [8] propose gradient-based compression for sending updates between each edge device and the central server.

IV. Project goals and evaluation metric

Our goal is to produce a federated learning system with some n learners that have non-iid data from a subset of the global distribution, and to improve model compression by applying quantization more effectively.

To evaluate the efficiency of our approach, we will compare its performance to that of performance to several benchmarks on CIFAR-10. We are particularly interested in quantifying the tradeoff between aggregate compression and accuracy, in measuring the communication costs between the local learners and devices, and in assessing the time to convergence for each approach.

V. Proposed approach, novelty and secret weapon

The FedCD algorithms begins with a global model that all devices update to, as in the normal FedAvg algorithm. Then, every x rounds, we will clone and compress all existing models.

In each training round, every participating device sends a weighted update (plus randomization) to its top n models, and discards (no longer ranks and sends weight updates to) any models that consistently predict badly on that device's data. These decisions are made off of a ranking that is updated based on validation accuracy.

Ideally, we would see that the edge devices would form groups updating the same global model aligned to their archetypes.

By spawning copies of the model (with added random noise in the ranking to encourage some exploration in the models that are updated) we can learn the archetypes of the other edge devices and update weights based on the device's archetype, i.e. each model specializes to fit is devices' distribution without compromising privacy, thereby effectively addressing the problems that non-iid data pose to federated learning. Furthermore, compression via quantization allows for multiple smaller models on-device, and fewer rounds to convergence will result in reduced communication cost.

In particular, we modify the weight update function as follows. Let N be the number of devices. Let w_m denote the weight vector for model m. Then we have

$$w_m = \frac{\sum_{i=1}^{N} w_m^{(i)} c_m^{(i)}}{c^{(i)}} \tag{1}$$

where $c_m^{(i)} \geq 0$ denotes the ranking that device i assigns model m, where a larger ranking denotes a preferred model. As part of our experiments, we investigated multiple ways of generating $c_m^{(i)}$ based on the accuracy a that model m has on device i's data to increase differentiability between archetypes. We defined the ranking at round r as

$$c_m^{(i)}[r] = \frac{\sum_{k=r-4}^{r-1} c_m^{(i)}[k]}{3} \tag{2}$$

Empirically, we found that using an average of the 3 most recent rounds' validation accuracy resulted in the highest performance while being robust to oscillation.

Furthermore, to avoid each device having to store all models, we delete all models m for which the following holds

$$\max(c^{(i)}) - c_m^{(i)} \geq \sigma(c^{(i)}) \tag{3}$$

where $\max(c^{(i)}$ denotes the ranking that device i assigns to is most preferred model, and $\sigma(c^{(i)})$ denotes the standard deviation over the model rankings by device i. Note that using a standard deviation based deletion criterion ensures that any device will maintain at least two models if there are at least two global models.

VI. Intellectual points

Current work addressing non-iid federated larning either uses shared global datasets or accepts significantly increased communication costs (i.e. using a peer-to-peer learning scheme). In contrast, from cloning the global model, we can apply compression techniques such as quantization to ensure the cloned models fit within the on-device capacity requirements of the edge devices. In particular, FedCD has the ability to share weights between models while maintaining specialization, and our experiments with tuning model parameters show a way to speed up device specialization without global models having access to device data and compromising privacy.

VII. Work performed

We first developed a testing environment framework for simulating edge devices and a global model. In particular, we created a Device class containing device archetype, bias, and models, which supports both cloning and deleting models as well as ranking-based weight updates.

A. CIFAR Dataset

For testing our approach, we utilized subsets of the CIFAR-10 dataset [9], a standard for federated learning. Our testing environment had 40k training images, 10k validation images, and 10k testing images, in which each device has its own biased training/validation/testing set that is consistent with its archetype. We exclusively use a device's validation set to determine its ranking for a given model. Lastly, we evaluate the best performing model for each device against different versions of an unbiased and a biased testing dataset.

With this setup we were particularly interested in investigating the trade-off between aggregate compression and time to convergence.

B. Experimental Setup

Our experimental setup specified thee following characteristics for each edge device:

- Archetypes: Defined as a set of labels
- Bias: the fraction of a devices' local dataset that consists of the archetype labels
- Ranking: a normalized weighting of models on the device to update to (with a minimum of 2 active devices)

In addition, we had several experimental levers to design experiments:

- The number of archetypes versus the number of edge devices
- The complexity of thee archetypes (single label versus multiple labels)
- Milestones at which to clone the models
- Criteria to delete a global model from an edge devices' ranking
- Each edge devices' ranking of global models to update (variations on the random noise function, "stickiness" of the ranking from round to round)

VIII. Results and Discussion

A. Proof of Concept

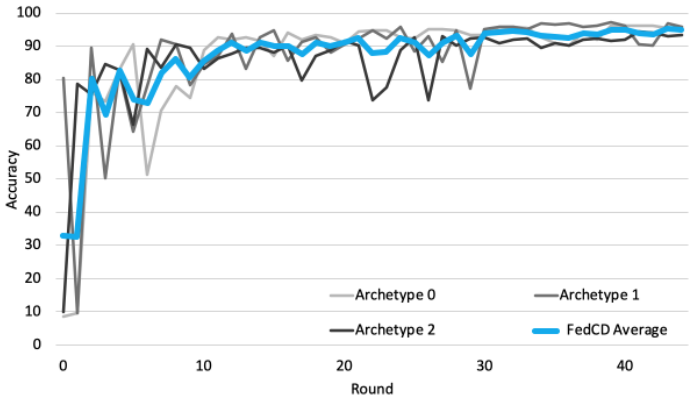

Fig. 1. Performance of FedCD on experiments with 3 archetypes.

Figure 1 shows FedCD on a proof of concept exeriment with 9 devices of 3 different simple archetypes and bias set to $\sim Unif(0.75, 0.85)$. We see that oscillations in test accuracy between devices are quickly mitigated by round 10 and convergence is reached by round 30. This, along with confirmation that cloning of models helped devices split off with respect to their archetypes, indicates that FedCD successfully dynamically split devices and helped arrive at faster convergence.

B. Effects of Quantization

During the cloning process of our FedCD algorithm, we dynamically split devices and let them self-sort into groups of similar archetypes. This allows each cloned model to specialize for a smaller subset of archetypes. As such, each model can be quantized to a smaller size without losing accuracy. We show this here with the same experimental parameters as Figure 1.

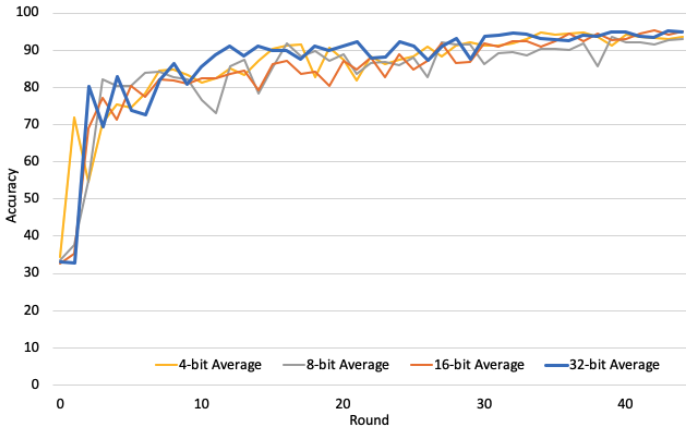

Fig. 2. Performance of different quantization levels on experiments with 3 archetypes.

As we can see in Figure 2, there was no significant effect of quantization on the performance and only slight effect on the convergence time of the resulting models.

C. Hierarchical Archetypes

A scenario with several distinct archetypes, as in the previous experiment, is rare in the real world. Instead, it is more likely that there are "meta-archetypes" with sub-types within these more comprehensive types of edge devices. To test the applicability of FedCD, we constructed two sets of data (meta-archetypes with labels 0+1+2 and 3+4+5) with 6 archetypes represented by the labels, i.e. an edge device of archetype 1 only has access to training examples with labels 0, 1, and 2.

The experiment was run on 18 devices with bias $\sim Unif(0.6, 0.7)$.

From Figures 3 and 4, it's clear that in 45 rounds the FedAvg algorithm could not converge while thee FedCD algorithm has fully converged, showing that it can learn more complicated archetypes better than the baseline.

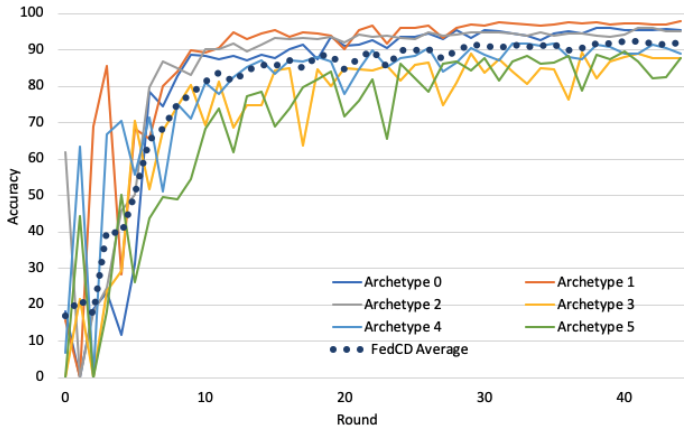

Fig. 3. Performance of the FedCD algorithm with 6 archetypes within 2 meta-archetypes.

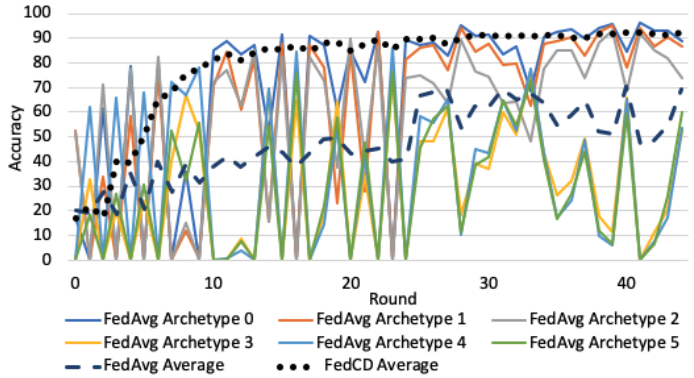

Fig. 4. Performance of the FedCD algorithm against FedAvg on an experiment with 6 archetypes within 2 meta-archetypes.

D. Communication Costs

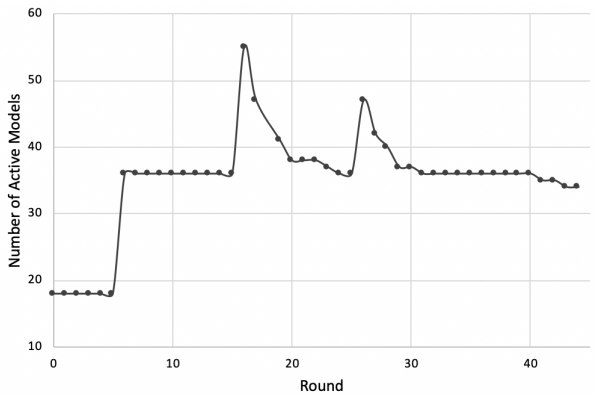

Fig. 5. Total number of models updated per round across 45 training rounds.

Although in the worst-case there is a theoretically exponential communication cost overhead due to the cloning of every model at the milestones, in practice if there are archetypes present in the data (as in our experiments) models were pushed towards favoring a single model and deleting other models that didn't fit their data well. This tendency to prefer a single model is aided by the randomness in the ranking function.

In Figure 5, we can see that the number of active models initially increases during the cloning phases (milestones at rounds 5, 15, and 25) and drops during the consolidation phase as each edge device begins to delete models they no longer update to. In the end, each of the 18 devices update exactly two active models as per the ranking function's requirements. This can be further decreased if needed.

E. Discussion

We have shown that using FedCD, we achieve significantly higher accuracy and faster convergence compared to FedAVG with at most twice the storage and communication requirements of FedAVG with non-iid data that fits a certain archetype.

Note that this is due to our deletion function requiring that each edge device must maintain two models at all times. However, in practice this could be further reduced to a single model after the models have already converged. Then communication cost and storage cost would be equal for edge devices as compared to having a single global model, and potentially even lower if we can use existing compression techniquest more aggressively without significant loss in accuracy due to specialization.

IX. Conclusion

A. Contribution and Assessment

We have introduced FedCD as a novel method for addressing non-iid federated learning without globally shared data or significant communication overhead. From several experiments of varying complexity, it is clear that compared to the baseline of FedAvg, FedCD performs significantly better than other approaches like FedCurv by converging in fewer rounds.

B. Future Work

From this project, we learned how to scale smaller experiments to larger datasets and more complex setups, and we also familiarized ourselves with ways to integrate methods for data compression with a data privacy-conscious context.

Future work could apply our approach to more datasets, e.g. MNIST. It could also replace or add on to our choice of compression (weight quantization) by e.g. investigating teacher-student or weight pruning based methods. Note that teacher-student networks in Federated Learning have primarily been used in the context of reducing communication overhead with non-iid data (Kim and Park, 2018).

Finally, our work can also be extended to possible solutions for malicious device attacks in Federated Learning. For instance, FedCD may be used to mitigate malicious devices by splitting them off into a separate model by themselves.

REFERENCES

[1] A. Kaplan and M. Haenlein, "Siri, siri, in my hand: Who's the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence," *Business Horizons*, vol. 62, no. 1, pp. 15–25, 2019.

[2] C. J. Sims, L. Meyn, R. Caruana, R. Rao, T. Mitchell, and M. Krohn, "Predicting cesarean delivery with decision tree models," *American Journal of Obstetrics and Gynecology*, vol. 183, no. 5, pp. 1198–1206, 2000.

[3] M. Krzysztofek, *GDPR : General Data Protection Regulation (EU) 2016/679 : post-reform personal data protection in the European Union*, ser. European monographs ; 107. Alphen aan den Rijn, The Netherlands: Wolters Kluwer, 2019.

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[5] P. Kairouz, H. Mcmahan, B. Avent, A. Bellet, M. Bennis, N. Arjun, K. Bonawitz, C. Zachary, G. Cormode, R. Cummings, R. D'Oliveira, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. Stich, Z. Sun, A. Suresh, F. Tramèr, J. Wang, X. Li, Z. Xu, Q. Yang, F. Yu, Y. Han, and Z. Sen, "Advances and open problems in federated learning," *arXiv.org*, 2019. [Online]. Available: http://search.proquest.com/docview/2325124335/

[6] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *arXiv.org*, 2018. [Online]. Available: http://search.proquest.com/docview/2139384574/

[7] J. Konečný, H. Mcmahan, F. Yu, P. Richtárik, A. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv.org*, 2017. [Online]. Available: http://search.proquest.com/docview/2076473432/

[8] K. Ahmed and P. Richtárik, "Gradient descent with compressed iterates," *arXiv.org*, 2020. [Online]. Available: http://search.proquest.com/docview/2289191385/

[9] The cifar-10 dataset. [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html