Modeling Robustness to Distribution Diversity Among Participants in Federated Learning

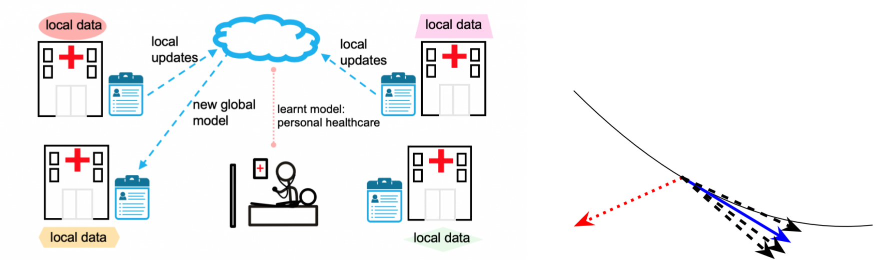
Tonmoay Deb

December 2021

1 Introduction and Background

Federated Learning (FL), proposed by [6] is a new paradigm in decentralized machine learning, where a group of participants who each possess local data jointly train a model. In this process, each participant uses their local data to compute a gradient given the model parameters. Then, it shares the gradient to a central server, keeping the local data private. The server takes gradients from each of the participants and aggregates them into a global gradient update. The globally updated model parameters are then shared to each of the participants for use in the next iteration. In this process, the data owners, i.e., local machines, never share sensitive information, yet all devices can train a sophisticated model collaboratively. This "privacy-preserving" technique has a number of practical impacts, from personal privacy to institutional. For example, scanner machines, e.g., X-ray, have sensitive hospital patient information, and thus there are often legal barriers preventing the open sharing of such data. However, if hospitals want to train a model that predicts lung damage, it may require a significant amount of training samples. Here, the federated learning approach can be helpful because multiple hospitals will share a gradient of X-ray images locally and update their models from the global updates without releasing any confidential information. Figure 1a illustrates the scenario of federated learning in the medical domain. In mobile devices, e.g., mobile phones, federated learning might come in handy to train language models without exposing any keystroke information to anyone. However, as there is no central verification of the data and edge devices' gradient, federated learning can be easily exposed to adversarial attacks. The attacks can appear in several forms, sometimes unintentionally.

Several works of follow-up research focused on making federated learning robust against adversarial attacks. The initial line of research in federated learning robustness assumed some hypothetical attack models. This report will assume that our training objective is to train a homogeneous model, where the local machines share weights for a global update. In this setting, the weight update is done by FedAVG or FedSGD [6]. These two approaches mainly focus on averaging the weights, which may affect the overall training weights even if a



(a) Federated Learning in Hospital Data (b) Malicious Gradient

Figure 1: An example of federated learning (left) setup with confidential data (image taken from [9]). However, any of the local devices may have a malicious gradient value (right image, marked as red) to manipulate the global update (image taken from [2]).

device sends anomaly weight to the global server. In Figure 1b, we can see that the red gradient is malicious compared to other gradients, and this will affect the whole training procedure if the anomaly can not be detected. [2] formulated the problem as "Byzantine Attack" to tackle this problem, where a fraction of the local devices can act maliciously. In this scenario, the malicious gradient push may be intentional as the infected device(s) know the internal update mechanism. They proposed Byzantine-resilient aggregation that can perform robustly as long as a majority of the participants are honest. This is just one canonical example of adversarial robustness in FL – many different adversarial models have been characterized [5], including e.g. Sybil-based poisoning [1, 3] attacks, wherein the adversary is capable of creating additional FL participants to skew the honest-malicious ratio. In this work, we propose a setting with a weaker adversarial model than we have observed in our literature search. Though the model supposes weaker adversaries, we believe that this model will fulfill a natural and useful niche in FL.

Our main focus is to address the scenario of FL participants who have diverse local distributions. We assume that a certain portion of devices generates data from a shifted distribution, and the gradient averaging contributes to misclassification. Some existing FL frameworks may be applicable to this problem such as domain adaptation [7], transfer learning [4], and representation learning [10]. To address robustness in distribution shift for federated learning, [8] proposed AUROR, a framework to find anomaly distribution of features in participating devices. Here, the global model has access to certain masked features of the local devices, which are used to cluster and find the anomaly distributions. However, we restrict privacy by setting the local devices to share only gradient to the global model in our problem (as we assume our structure is fully decentralized). Below we discuss more on our problem setup.

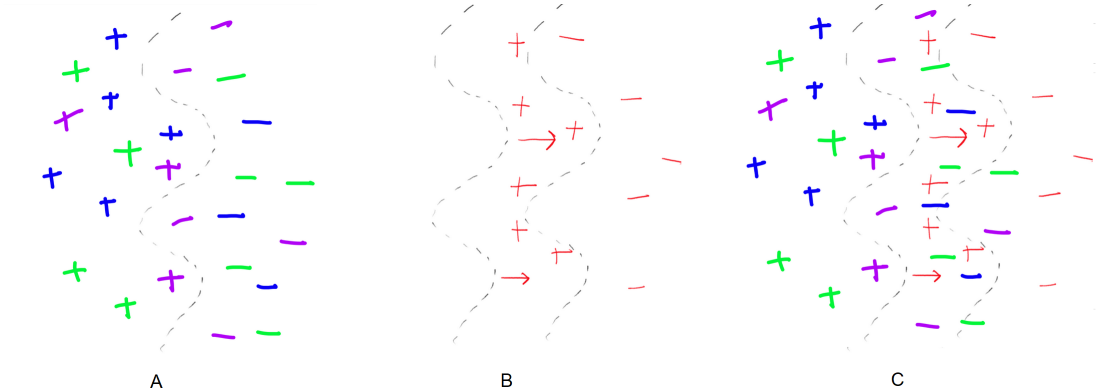


Figure 2: A cartoon of aggregated data among parties with distinct distributions participating in FL. **A.** The data of three parties sampling from roughly the same distribution. **B.** The data of a fourth party whose distribution is shifted to the right in comparison to the initial three. **C.** The data of all four parties superimposed. Note that the region between the two dotted lines contains a mixture of positive and negative labeled examples, potentially resulting in degraded model accuracy.

1.1 Motivation

As FL is particularly vulnerable to data poisoning attacks, the problem of mitigating threats from malicious adversarial FL participants has received substantial attention in previous work. Significantly less attention has been devoted to addressing a setting wherein *honest* participants are sampling data from related but distinct distributions. As mentioned, there exist tools such as federated transfer learning, or federated representation learning, that could potentially be employed to address this problem. However, to our knowledge these solutions are thus far totally informal – no explicit formulation of this setting, nor algorithms with formal guarantees of robustness against its specific challenges, are known to us.

To think through this setting, we provide a sketch of an example data set distributed among FL participants (Figure 2). Consider four hospitals that want to jointly train a model without sharing user data. Since they are hospitals, we can reasonably expect them to follow a semi-honest adversarial model (that is, they will execute the FL protocol faithfully, but we still need a protocol that prevents sharing of patient data). However, suppose that due to differences in machine calibration, differing opinions of experts working at the hospitals, different patient demographics etc., one of the hospitals has data that comes from a noticeably shifted distribution compared to the others (see the Red party in Figure 2). This diversity among data distributions could result in degraded performance from the trained model, *even though all parties are behaving honestly.* Further, the consequences of these differing local distributions of training data will be particularly severe for the Red party. This is because in the most basic forms of FL (e.g. averaging gradient updates), the jointly trained model converges to a solution that compromises between performance on all of its training data points. Thus the resulting model will likely be particularly unsuitable for a party whose data is sampled from a divergent distribution. In the present work, our main contribution is that we provide a model that captures this and related scenarios.

2 Model

We model the scenario of distribution diversity among participants in FL as follows. Suppose S is a set of n parties, and $\mathcal{A} \subset S$ is a set of corrupted parties with size k. Suppose each party $i \in [n]$ has $m_i \in \mathcal{N}$ which represents the number of data points they will contribute to the FL protocol. Suppose D is some uncorrupted distribution, and $\{D'_1, ..., D'_k\}$ is a set of corrupted distributions that can optionally be defined to be within a certain distance of the true distribution D. Then the sampling process for our model is defined as follows.

- For $P_i \in \mathcal{A}$:
  - For $j \in [1, m_i]$:
    - Sample $(x_{i,j}, y_{i,j}) \sim D'_i$, assign it to P_i
- For $P_i \in S \setminus \mathcal{A}$:
  - For $j \in [1, m_i]$:
    - Sample $(x_{i,j}, y_{i,j}) \sim D$, assign it to P_i

We note that this model can be seen as a relaxation of existing models of data corruption in centralized machine learning (e.g. simply sampling an ϵ-fraction of the data from a corrupted distribution). Thus some theoretical guarantees could likely be obtained from these simpler models. Further, the effects of the data corruption from our model could ostensibly be offset by algorithms designed for stronger forms of robustness (e.g. fully byzantine fault-tolerant FL). However, we argue that our model is important even still. Indeed, our setting provides unique advantages that could likely be leveraged to achieve more desirable results. The main advantages of our setting are (a): parties with data from divergent distributions are otherwise *not malicious*, and thus can be counted on to honestly execute local protocols which investigate and mitigate data corruption, and (b): each contributor of data has a *local distribution*, the structure of which can be leveraged by learning algorithms. We additionally note that while most corruption models are predicated on the goal of improving performance relative to the uncorrupted distribution, in our setting the performance of parties sampling from *corrupted distributions* is also important. For example, we would like to be able to leverage the results of the jointly trained model even for a hospital whose MRI machine has some technical malfunctions. By making assumptions about the distributions of corrupted parties' data, our setting makes it possible to design algorithms with this goal in mind.

2.1 A Byzantine Tolerant FL Algorithm for SGD

Many modern machine learning algorithms rely on Stochastic Gradient Descent (SGD), including neural networks, regression, matrix factorization, and support vector machines[2]. Therefore, it is useful to consider how we can make Federated Learning more resilient to different types of corruption and possibly malicious adversaries. Such a robust algorithm could produce more accurate classifiers in a Federated Learning situation where the participants would want to share their data indirectly with one another thereby leveraging all data without compromising privacy. It is particularly useful in a situation where the number of participants is relatively small and the demand for greater accuracy outweighs concerns for computational costs. In contrast, such an algorithm may be prohibitively expensive for use cases involving millions of participants, such as algorithms that make use of the data on edge devices in personal computing. We note that this type of algorithm is particularly suitable for application in the medical context that we are concerned with in this report, where the number of participants is limited, but a high premium is placed on the reliability of the final classifier and robustness of the learning algorithm. We also note that a fault-tolerant SGD algorithm is more suitable in Federated Learning because of its natural ability to leverage data from multiple participants indirectly during mini-batch updates without revealing each participant's data to the others.

The proposed algorithm, Krum[2], is designed to make the FL protocol tolerant against multiple Byzantine workers. Existing algorithms that ultimately rely on linear combinations of the gradient vector updates from the workers are shown to be unable to tolerate more than one Byzantine worker - a worker that arbitrarily corrupts his data, either maliciously or non-maliciously. For example, a least-square-distance based aggregation rule will be unable to select the vectors closest to the true distribution if a second Byzantine worker skew the distribution of the vectors so much that the other Byzantine worker has a smaller square-distance to other workers than the uncorrupted workers. Krum is tolerant against f faulty workers/participants among n total participants in the worst case, as long as $2f + 2 < n$, namely the corrupted workers constitute a minority of total workers. Another strong advantage of Krum is its time complexity ($O(n^2 \cdot d)$), where d is the dimension of the gradient which can be a very large number in neural network training.

3 Research Questions and Future Work

We sketched a simple framework that one might follow to achieve algorithms robust to the corruption in our setting. An outline proceeds as follows.

1. Perform a standard FL algorithm to train a model M

2. Using held out test data, parties evaluate the performance of M on their local distributions, and/or compute aggregate statistics of their data. They communicate these evaluations with the other parties. Using this information, the parties determine whether their distributions are divergent from the other participants.

3. Parties with divergent distributions perform a 'batch correction' protocol to align their data with the consensus distribution.

4. Parties perform a second pass of FL using the aligned data to train a model M'

    - Parties with divergent distributions can use the transformation from the previous step to feed data from their local distribution to M'

A potential area for future work would be to realize this sketch into a concrete algorithm. We suspect that federated transfer learning and/or federated representation learning are likely candidates for effectively realizing step 3. In particular, if the parties were able to learn a shared representation of their local data sources that better aligned all of the distributions, that would likely mitigate the source of corruption in our model.

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020.

[2] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 118–128, 2017.

[3] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018.

[4] Dashan Gao, Yang Liu, Anbu Huang, Ce Ju, Han Yu, and Qiang Yang. Privacy-preserving heterogeneous federated transfer learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2552–2559. IEEE, 2019.

[5] Lingjuan Lyu, Han Yu, Xingjun Ma, Lichao Sun, Jun Zhao, Qiang Yang, and Philip S Yu. Privacy and robustness in federated learning: Attacks and defenses. *arXiv preprint arXiv:2012.06337*, 2020.

[6] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

[7] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054*, 2019.

[8] Shiqi Shen, Shruti Tople, and Prateek Saxena. Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 508–519, 2016.

[9] Machine Learning Department University, Carnegie Mellon. Federated learning: Challenges, methods, and future directions.

[10] Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*, 2020.