Clustered Federated Learning

Felix Sattler
Fraunhofer HHI
felix.sattler@hhi.fraunhofer.de

Klaus-Robert Müller
TU Berlin
klaus-robert.mueller@tu-berlin.de

Wojciech Samek
Fraunhofer HHI
wojciech.samek@hhi.fraunhofer.de

Abstract

Federated Learning (FL) is currently the most widely adopted framework for collaborative training of (deep) machine learning models under privacy constraints. Albeit it's popularity, it has been observed [11][3] that Federated Learning yields suboptimal results if the local clients' data distributions diverge. To address this issue, we present Clustered Federated Learning (CFL), a novel Federated Multi-Task Learning (FMTL) framework, which exploits geometric properties of the FL loss surface, to group the client population into clusters with jointly trainable data distributions. In contrast to existing FMTL approaches, CFL does not require any modifications to the FL communication protocol to be made and comes with mathematical guarantees on the clustering quality even for non-convex objectives. As clustering is only performed after Federated Learning has converged to a stationary point, CFL can be viewed as a post-processing method that will always achieve greater or equal performance than FL by allowing clients to arrive at more specialized models. We verify our theoretical analysis in experiments with deep neural networks.

1 Introduction

Federated Learning [8][4][2][1][6] is a distributed training framework, which allows multiple clients (typically mobile or IoT devices) to jointly train a single deep learning model on their combined data in a communication-efficient way, without requiring any of the participants to reveal their private training data to a centralized entity or to each other. In doing so, Federated Learning implicitly makes the assumption that it is possible for one single model to fit all client's data generating distributions φ_i at the same time. Given a model $f_\theta : \mathcal{X} \to \mathcal{Y}$ parametrized by $\theta \in \Theta$ and a loss function $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ we can formally state this assumption as follows:

Assumption 1. ***("Conventional Federated Learning"):*** *There exists a parameter configuration $\theta^* \in \Theta$, that (locally) minimizes the risk on all clients' data generating distributions at the same time*

$$R_i(\theta^*) \leq R_i(\theta) \; \forall \theta \in B_\varepsilon(\theta^*), i = 1, .., m \tag{1}$$

Hereby $R_i(\theta) = \int l(f_\theta(x), y) d\varphi_i(x, y)$ is the risk function associated with distribution φ_i.

It is easy to see that this assumption is not always satisfied. Concretely it is violated if either (a) clients have disagreeing conditional distributions $\varphi_i(y|x) \neq \varphi_j(y|x)$ or (b) the model f_θ is not expressive enough to fit all distributions at the same time. Simple counter examples for both cases are presented in Figure 1. In the following we will call two clients and their distributions φ_i and φ_j *congruent* (with respect to f and l) if they satisfy Assumption 1 and *incongruent* if they don't.

The goal in Federated Multi-Task Learning is to provide every client with a model that optimally fits it's local data distribution. If the clients' data distributions are incongruent the ordinary Federated Learning framework, in which all clients are treated equally and only one single global model is learned, is not capable of achieving this goal. We suggest to generalize the conventional Federated Learning Assumption, in order to incorporate FL problems where the clients' distributions adhere to a clustering structure:

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

Assumption 2. ***("Clustered Federated Learning"):*** *There exists a partitioning* $\mathcal{C} = \{c_1, .., c_k\}$, $\bigcup_{i=1}^{k} c_k = \{1, .., m\}$ *of the client population, such that every subset of clients* $c \in \mathcal{C}$ *satisfies the conventional Federated Learning Assumption.*

2 Clustered Federated Learning

In this paper, we address the question of how to solve distributed learning problems that satisfy Assumption 2. This will require us to first identify the correct partitioning $\mathcal{C}$, which at first glance seems like a daunting task, as under the Federated Learning paradigm the server has no access to the clients data, their data generating distributions or any meta information thereof. However, as we will see, there exists an explicit criterion based on which the clustering structure can be inferred.
To see this, let us first look at the following simplified Federated Learning setting with m clients, in which the data on every client was sampled from one of two data generating distributions φ_1, φ_2 such that w.l.o.g. $D_1, .., D_k \sim \varphi_1(x, y)$ and $D_{k+1}, .., D_m \sim \varphi_2(x, y)$. Every Client is associated with an empirical risk function $r_i(\theta) = \sum_{x \in D_i} l_\theta(f(x_i), y_i)$ which approximates the true risk arbitrarily well if the number of data points on every client is sufficiently large $r_i(\theta) \approx R_{I(i)}(\theta)$. For demonstration purposes let us first assume equality. Then the Federated Learning objective becomes

$$F(\theta) := \sum_{i=1}^{m} |D_i|/|D| r_i(\theta) = a_1 R_1(\theta) + a_2 R_2(\theta) \tag{2}$$

with $a_1 = \sum_{i=1}^{k} |D_i|/|D| > 0$ and $a_2 = \sum_{i=k+1}^{m} |D_i|/|D| > 0$. Under standard assumptions it has been shown [7] that the Federated Learning optimization protocol converges to a stationary point θ^* of the Federated Learning objective. In this point it holds that $\nabla F(\theta^*) = 0$ from which it follows that

$$\nabla R_1(\theta^*) = -\frac{a_2}{a_1} \nabla R_2(\theta^*) \tag{3}$$

Now we are in one of two situations. Either it holds that $\nabla R_1(\theta^*) = \nabla R_2(\theta^*) = 0$, in which case we have simultaneously minimized the risk of all clients. This means φ_1 and φ_2 are congruent and we have solved the distributed learning problem. Otherwise φ_1 and φ_2 are incongruent and the *cosine*

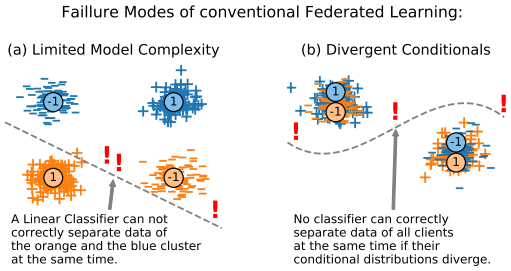

Figure 1: Two toy cases in which the Federated Learning Assumption is violated. Blue points belong to clients from the first cluster while orange points belong to clients from the second cluster. Left: Federated XOR-problem. An insufficiently complex model is not capable of fitting all clients' data distributions at the same time. Right: If different clients' conditional distributions diverge, *no model* can fit all distributions at the same time. In both cases the data on clients belonging to the same cluster can be easily separated.

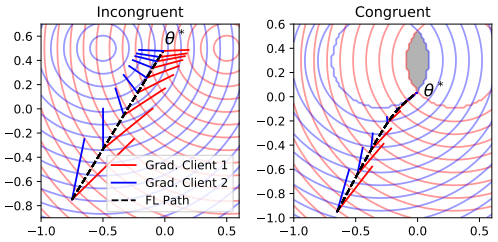

Figure 2: Displayed are the optimization paths of Federated Learning with two clients, applied to two different toy problems with incongruent (left) and congruent (right) risk functions. In the incongruent case Federated Learning converges to a stationary point of the FL objective where the gradients of the two clients are of positive norm and point into opposite directions. In the congruent case there exists an area (marked grey in the plot) where both risk functions are minimized. If Federated Learning converges to this area the norm of both client's gradient updates goes to zero. By inspecting the gradient norms the two cases can be distinguished.

similarity between the gradient updates of any two clients is given by

$$\alpha(\nabla r_i(\theta^*), \nabla r_j(\theta^*)) := \frac{\langle \nabla r_i(\theta^*), \nabla r_j(\theta^*) \rangle}{\|\nabla r_i(\theta^*)\| \|\nabla r_j(\theta^*)\|} = \frac{\langle \nabla R_{I(i)}(\theta^*), \nabla R_{I(j)}(\theta^*) \rangle}{\|\nabla R_{I(i)}(\theta^*)\| \|\nabla R_{I(j)}(\theta^*)\|} = \begin{cases} 1, & I(i) = I(j) \\ -1, & I(i) \neq I(j). \end{cases} \tag{4}$$

For a visual illustration of the result we refer to Figure 2. This insightful consideration tells us that, after Federated Learning has converged to a stationary solution, *we can distinguish clients based on their hidden data generating distribution by only inspecting the cosine similarity between their gradient updates*. The result can be readily generalized to more than two data generating distributions and empirical risk functions which deviate from the true risk, an extended manuscript is currently in preparation. Now we are in the position to improve Federated Learning for all clients by grouping the client population into clusters corresponding to their data generating distribution and training a separate model for each cluster.

Algorithm: Clustered Federated Learning recursively bi-partitions the client population in a top-down way: Starting from an initial set of clients $c = \{1, .., m\}$ and a parameter initialization θ_0, CFL performs Federated Learning according to Algorithm 1, in order to obtain a stationary solution θ^* of the FL objective. After Federated Learning has converged, the stopping criterion

$$0 \leq \max_{i \in c} \|\nabla_\theta r_i(\theta^*)\| < \varepsilon_2 \tag{5}$$

is evaluated. If criterion (5) is satisfied, we know that all clients are sufficiently close to a stationary solution of their local risk and consequently CFL terminates, returning the FL solution θ^*. If on the other hand, criterion (5) is violated, this means that the clients are incongruent and the server computes the pairwise cosine similarities α between the clients' latest transmitted updates according to equation (4). Next, the server separates the clients into two clusters in such a way that the maximum similarity between clients from different clusters is minimized

$$c_1, c_2 \leftarrow \arg \min_{c_1 \cup c_2 = c} \left(\max_{i \in c_1, j \in c_2} \alpha_{i,j} \right). \tag{6}$$

CFL is then recursively re-applied to each of the two separate groups starting from the stationary solution θ^*. Splitting recursively continues on until none of the sub-clusters violate the stopping criterion anymore, at which point all groups of mutually congruent clients have been identified. The entire recursive procedure is presented in Algorithm 2.

3 Related Work

Federated Learning [8][4] is currently the dominant framework for distributed training of machine learning models under communication- and privacy constraints. Federated Learning assumes the clients to be congruent, i.e. that one central model can fit all client's distributions at the same time. Different authors have investigated the convergence properties of Federated Learning in congruent iid and non-iid scenarios: [10],[9] and [12] perform an empirical investigation, [7] prove convergence guarantees. Conventional Federated Learning [8][4] has been extensively investigated in congruent

Algorithm 1: FederatedLearning (FL)

Input: initial parameters θ, set of clients c
repeat
  for $i \in c$ ***in parallel*** **do**
    • $\theta_i \leftarrow \theta$
    • $\Delta\theta_i \leftarrow \text{SGD}_n(\theta_i, D_i) - \theta_i$
  end
  • $\theta \leftarrow \theta + \sum_{i \in c} \frac{|D_i|}{|D_c|} \Delta\theta_i$
until $\| \sum_{i \in c} \frac{|D_i|}{|D_c|} \Delta\theta_i \| < \varepsilon_1$
return θ

Algorithm 2: ClusteredFederatedLearning (CFL)

Input: initial parameters θ, set of clients c
• $\theta^* \leftarrow$ FederatedLearning(θ, c)
• $\alpha_{i,j} \leftarrow \frac{\langle \nabla r_i(\theta^*), \nabla r_j(\theta^*) \rangle}{\|\nabla r_i(\theta^*)\| \|\nabla r_j(\theta^*)\|}, i, j \in c$
if $\max_{i \in c} \|\nabla r_i(\theta^*)\| \geq \varepsilon_2$ **then**
  • $c_1, c_2 \leftarrow \arg\min_{c_1 \cup c_2 = c} (\max_{i \in c_1, j \in c_2} \alpha_{i,j})$
  • $\theta_i^*, i \in c_1 \leftarrow$ ClusteredFederatedLearning(θ^*, c_1)
  • $\theta_i^*, i \in c_2 \leftarrow$ ClusteredFederatedLearning(θ^*, c_2)
else
  • $\theta_i^* \leftarrow \theta^*, i \in c$
end
return $\theta_i^*, i \in c$

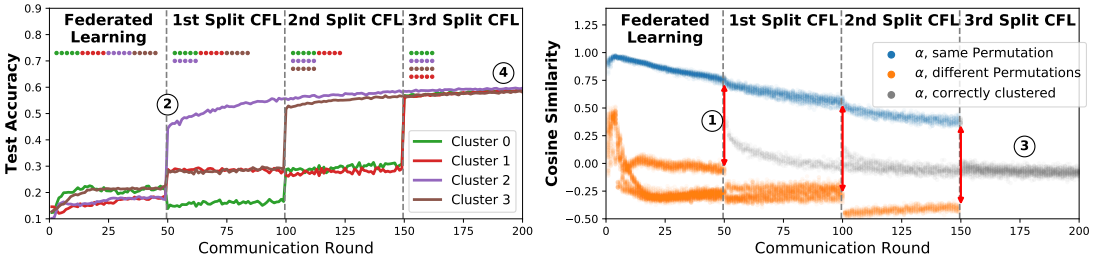

Figure 3: Clustered Federated Learning applied to the "permuted labels problem" on CIFAR with 20 clients and 4 different permutations. Left: Accuracy of the trained model(s) on their corresponding test sets. In communication rounds 50, 100 and 150 the client population is separated, by using the cosine similarity criterion (6), which leads to an immediate improvement in accuracy. Right: Pairwise cosine similarity between the weight-updates of different clients. If clients have the same data generating distribution their similarities are marked blue in the plot, otherwise orange. For all clients that have already been correctly clustered the pairwise cosine similarities are displayed in grey.

non-iid scenarios [9][12], but is unable to deal with the challenges of incongruent data distributions as argued in section 1. Existing Federated Multi-Task Learning approaches [11][3] allow clients to adapt their local models, but are only applicable to convex objective functions, incapable of distinguishing congruent from incongruent settings, and limited in their ability to scale to massive client populations. In contrast, CFL can be applied to arbitrary Federated Learning problems, incurs no computational overhead for the clients (and negligible overhead for the server) and is not restricted to a certain class of objective functions.

4 Experiments

The Cifar-10 dataset [5] contains 50000 32×32×3 training images in 10 categories. We split the training data randomly and evenly among 20 clients, which we group into 4 different clusters. All clients belonging to the same cluster apply the same random permutation $P_{c(i)}$ to their labels such that their modified training and test data is given by

$$\hat{D}_i = \{(x, P_{c(i)}(y)) | (x, y) \in D_i\}, \qquad \hat{D_i^{test}} = \{(x, P_{c(i)}(y)) | (x, y) \in D^{test}\}. \tag{7}$$

The clients then jointly train a 5-layer convolutional neural network on the modified data using CFL with 3 epochs of local training at a batch-size of 100. Figure 3 (left) shows the joint training progression: In the first 50 communication rounds, all clients train one single model together, following the conventional Federated Learning protocol. After these initial 50 rounds, training has converged to a stationary point of the Federated Learning objective and client test accuracies stagnate at around 20%. Conventional Federated Learning would be finalized at this point. At the same time, we observe (Figure 3, right) that a distinct gap in the pairwise cosine-similarities of the different clients has developed (①, red arrow), indicating an underlying clustering structure. In communication round 50 the client population is split up for the first time, which leads to an immediate 25% increase in validation accuracy for all clients belonging to the "purple" cluster which was separated out ②. Splitting is repeated in communication rounds 100 and 150 until all clusters have been separated and the pairwise cosine similarity between clients from the same cluster has dropped to close to zero (③, grey dots), which indicates that clustering is finalized. At this point the accuracy of all clients has more than doubled the one achieved by the Federated Learning solution and is now at close to 60% ④.

5 Conclusion

In this paper we presented Clustered Federated Learning, a Federated Multi-Task Learning approach that can improve *any* existing Federated Learning Framework by providing the participating clients with more specialized models. CFL comes with mathematical guarantees on the clustering quality, doesn't require any modifications to the FL communication protocol to be made and is able to distinguish situations in which a single model can be learned from the clients' data from those in

which this is not possible and only separates clients in the latter situation. Our experiments on (non-convex) deep neural networks show that CFL can achieve drastic improvements over the Federated Learning baseline in situations where the clients' data exhibits a clustering structure.

References

[1] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[2] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy preserving machine learning. *IACR Cryptology ePrint Archive*, 2017:281, 2017.

[3] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

[4] 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.

[5] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[6] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019.

[7] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[8] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[9] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[10] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Sparse binary compression: Towards distributed deep learning with minimal communication. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[11] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[12] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.