Electronic Communications of the EASST
Volume 080 (2021)

Conference on Networked Systems 2021 (NetSys 2021)

Federated User Clustering for non-IID Federated Learning

Lucas Pacheco, Denis Rosário, Eduardo Cerqueira, and Torsten Braun

4 pages

ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Federated User Clustering for non-IID Federated Learning

Lucas Pacheco[1], **Denis Rosário**[2], **Eduardo Cerqueira**[2], **and Torsten Braun**[1]

[1] University of Bern, Institute of Computer Science, Communications and Distributed Systems Group,
[2] Federal University of Pará, Institute of Technology

Abstract: Federated Learning (FL) is one of the leading learning paradigms for enabling a more significant presence of intelligent applications in networking considering highly distributed environments while preserving user privacy. However, FL has the significant shortcoming of requiring user data to be Independent Identically Distributed (IID) to make reliable predictions for a given group of users. We present a Neural Network-based Federated Clustering mechanism capable of clustering the local models trained by users of the network with no access to their raw data. We also present an alternative to the FedAvg aggregation algorithm used in traditional FL, which significantly increases the aggregated models' reliability in Mean Square Error by creating several training models over IID users.

Keywords: Federated Learning, Clustering, Deep Learning

1 Introduction

With the breakthrough of Artificial Intelligence (AI), we are witnessing a significant increase in AI-based applications, even in networking systems [XLL$^+$20]. The use of machine learning techniques for network management and optimization is a continuing trend that takes advantage of modern advances of both networking systems and AI techniques [KPT$^+$20]. Such trends, combined with advances in wireless communication technologies and the presence of computing capabilities at the edge of the network, have made it possible to perform distributed learning over connected users to forecast user mobility, bandwidth usage, channel conditions, and many more critical network conditions.While the majority of the events that impact network functioning are user-generated, such as user requests and mobility, collecting user data in order to train machine learning models at the network level raises privacy and scalability concerns, as training massive amounts of user data in a centralized manner may pose a limitation for the learning process [XLL$^+$20].

The distributed nature of computing power in modern networks (*i.e.,* in edge servers, end devices, and remote datacenters) enables the rise of Federated Learning (FL) as a more reliable paradigm for learning. FL is a promising solution to providing reliable learning among many users' data while preserving users' privacy, as the users in FL are clients performing the training process. FL is a distributed machine learning paradigm based on each participating user in the network performing the training process over their data on a predefined architecture distributed by an aggregation entity. In this context, the training process is followed by an aggregation step, in which each user sends its trained weights, which are averaged at a central server, thus yielding a global Neural Network (NN) model. NN models aggregated in FL converge equivalently to ones trained in a centralized manner while maintaining user privacy and scaling more efficiently due to the distributed nature of the processing [WHL$^+$20].

However, such aggregated models cannot be reliably used by network operators due to limitations on data distributions' homogeneity across users. FL has limited capabilities to converge in the presence of such non-IID (Independent Identically Distributed) data. Data from different users can vary greatly and be based on many different random distributions, which may have different representations in the trained weights NNs. Thus, in the aggregation phase of FL, it is necessary to group users with similar statistical features such that different learned featured do not cancel each other when averaged. User clustering can help to make the distribution of the aggregated models uniform. However, clustering requires knowledge of the user's raw data, which is not available in a federated environment [ZLL+18]. In this paper, we present NSIM, a federated clustering technique for NN aggregation in federated environments capable of grouping together IID users with no knowledge of the underlying data.

2 Related Work

Zhao et al. [ZLL+18] show how the accuracy of FL models can decrease when trained with non-IID data and how to create subsets of the training data with more similar models can mitigate this problem. However, this requires the aggregator to know data characteristics from the users that are not typically available in an FL environment. Zhang et al. [ZSGS09] also tackle the problem of training non-IID data in FL from the point of view of independence measures based on predefined kernels, which have been widely used in machine learning to assess the independencies between datasets and samples. Several of these methods are available and can be used to establish a similarity measure between trained NN models. However, they cannot necessarily adapt to arbitrary architectures and data representations. Kornblith et al. [KNLH19] investigate the methods used to assess the similarity of trained NNs based on the datasets used for training. The authors compare and investigate the efficiency of Centered Kernel Alignment (CKA) and Cannonical Correlation Analisys (CCA) methods for weight comparisons in NNs, and conclude that CCA is more appropriate as a NN similarity measure and that the layers closer to the output carry unique information about the user data that can be used for similarity comparisons.

3 Neural-based Federated User SIMilarity & Clustering

3.1 User Similarity

Given that two users n and m possess data $\{D_n, D_m\}$ with similar features, they converge to similar locally trained models. Thus, we can compare the final NN generated by each user and assess how similar their training data is. Several similarity measures have been introduced in the literature, such as the Longest Common Subsequence (LCSS). However, most similarity search techniques require the presence of raw user data. For instance, the LCSS metric consists of finding the longest common sequence between two trajectories in terms of the data points constituting such trajectories within a certain radius.

We propose a Neural-based Federated User SIMilarity estimator for FL environments (NSIM), which can take as input a given NN architecture and finds how each layer and its weights can learn features from the training datasets, and also estimates the similarity between the training datasets given the trained NN models. We denote the architectures given to users as A_u, and the user training data as D_u. To compute such an estimator, we take as input the format of the training user datasets for inputs and outputs (*e.g.,* in the case of mobility data, we can have a sequence of past geographical coordinates as inputs and a pair of geographical coordinates as output).

The system then generates several synthetic data conforming to the given schema and trains NN models with architecture A_u using the given data sequences. The system builds one model for each synthetic user-created, each model denoted here as M_{synth}. We compare the generated user data sequences pairwise with the LCSS algorithm to obtain a ground truth similarity to train the similarity estimator. NSIM consists of a NN which takes as inputs the weights from other thrained NNs. The NSIM model has trained to input the last layer of the M_{synth} models trained. Note that we input two models at a time, as it is a pairwise comparison. The output of NSIM is the LCSS value obtained for the pair of models input, and after a round of training over ground-truth LCSS values, NSIM can learn how similarity representations are encoded in a given architecture A_u. We obtain the architecture for the NSIM similarity estimation through a grid-search step executed at the beginning of the process.

We test the proposed similarity estimator's efficiency in an LSTM-based NN for trajectory prediction. Mobility data is based on the real-world mobility dataset of Monaco [CH18]. The user mobility prediction architecture consists of four LSTM cells, three hidden layers with 20 neurons, each with a dropout rate of 0.2, and an output layer with two neurons. All dense layers use a Leaky ReLU activation and random weights initializations. NSIM has been trained with synthetic mobility data generated with a Random Walk model for 100 users and the respective LCSS scores computed for such users. As shown in Figures 2 and 3, experimental results find that even traditional kernel alignment techniques, such as CCA and CKA, which output a distance measure between two weights matrixes, cannot correlate the outputs of the last hidden layer of the user models with the actual LCSS values obtained for given pairs of users, as seen by the low Pearson correlation scores achieved. On the other hand, the NSIM model achieves a Pearson correlation of 0.96 for the predicted LCSS similarity and the correct values based on the weights of the last hidden layer of the model. Furthermore, the score is achieved with significantly less computing cost than directly comparing user datasets with the LCSS algorithm.

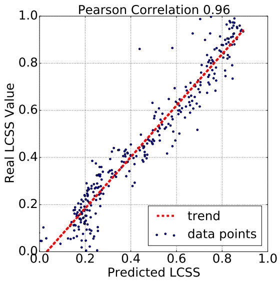

Figure 1: NSIM

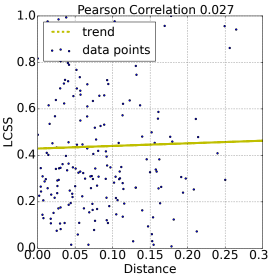

Figure 2: CCA

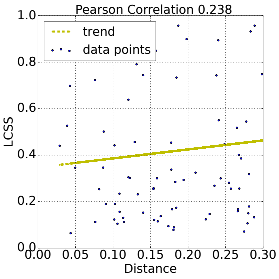

Figure 3: CKA

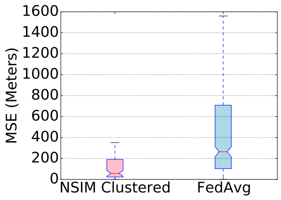

Figure 4: Accuracy of NSIM-clustered FL and FedAvg

3.2 User Clustering

Given an NSIM estimator trained for the architecture of a group of N participant users, each participant user performs a local training in the traditional FL paradigm. However, we perform similarity comparisons among all participating users and obtain a similarity measure for all users. We assume using an existing clustering algorithm based on the computed similarities to form C clusters of users. Since we do not know the number of clusters present beforehand, we chose the Density-based Spatial Clustering Of Applications With Noise (DBSCAN) algorithm, which labels each user in the network as belonging to one cluster.

After user clusters have been formed, the aggregator entity can perform the FedAvg operation on subsets of users defined by the clusters found by DBSCAN. Therefore, the final amount of NN models corresponds to the number of clusters formed. Figure 4 compares the error distribution on a validation set of the Monaco mobility trace in which each user trains a local model and models are aggregated based on NSIM clustering results or based on traditional FedAvg.

4 Conclusions

FL provides efficient learning models for connected users so that no raw user data must be sent, which can be intercepted or eavesdropped by malicious parties in the network. However, this approach falls into training a single model for possibly non-IID users, which compromises the final accuracy and applicability of the trained model for IA-based applications. We present a similarity estimator mechanism for arbitrary user data and architectures, which can detect similarity in user datasets to a high degree of confidence based on users' locally trained models. We also present an alternative to the traditional FedAvg algorithm, in which we first perform a federated clustering of users and aggregate models based on the clusters found. Experimental results show the similarity estimator's efficiency for federated clutering aggregation, significantly improving the aggregated models' predictions.

Bibliography

[CH18] L. Codeca, J. Härri. Monaco SUMO Traffic (MoST) Scenario: A 3D Mobility Scenario for Cooperative ITS. In *SUMO 2018, SUMO User Conference, Simulating Autonomous and Intermodal Transport Systems, May 14-16, 2018, Berlin, Germany*. Berlin, GERMANY, 05 2018.

[KNLH19] S. Kornblith, M. Norouzi, H. Lee, G. Hinton. Similarity of neural network representations revisited. *36th International Conference on Machine Learning, ICML 2019* 2019-June:6156–6175, 2019.

[KPT^{+}20] L. U. Khan, S. R. Pandey, N. H. Tran, W. Saad, Z. Han, M. N. Nguyen, C. S. Hong. Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism. *IEEE Communications Magazine* 58(10):88–93, 2020.

[WHL^{+}20] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, X. Chen. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Communications Surveys and Tutorials* 22(2):869–904, 2020.

[XLL^{+}20] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, P. Hui. Edge Intelligence: Architectures, Challenges, and Applications. *arXiv e-prints*, pp. arXiv–2003, 2020.

[ZLL^{+}18] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra. Federated Learning with Non-IID Data. *arXiv preprint arXiv:1806.00582*, 2018.

[ZSGS09] X. Zhang, L. Song, A. Gretton, A. J. Smola. Kernel measures of independence for non-iid data. In *Advances in neural information processing systems*. Pp. 1937–1944. 2009.