Hierarchical Approach to Model Aggregation in Federated Learning

Aryan
Kshitij Jha

Introduction

In Federated Learning aggregating client models is the most crucial task, that is performed by a central server. For instance, the aggregation of of google's smart keyboard AI Model is done at Google's servers. This introduces centralization in the process of aggregation along with other issues like decreased contributing client's trust and single point of failure [5]. To reduce or to eliminate the single point of failure we propose a method of decentralised hierarchical aggregation, in which the process of aggregation is also outsourced to the client devices. This reduces computation at the centralised server as well. Further TCP like protocol is proposed to ensure reliability of data transfer between aggregating clients [3]. The decentralized aggregation also reduces the time complexity of the task by parallelizing it amongst several aggregator nodes.

Related Work

Blockchain can improve federated learning by providing a decentralized and transparent platform for data privacy, secure model updates, and consensus among participants. It enhances data integrity, accountability, and trust in the federated learning process, while also enabling secure data marketplaces and incentivization mechanisms for collaboration. Several contributions of integrating blockchain into FL has been done in order to solve the problems of centralization. On page 5 of the paper [5] by Juncen Zhu et al., the authors talk about how single point aggregation by a centralised server may invoke privacy concerns amongst clients participating in FL. Peer-to-Peer federated learning is also introduced in the same paper but the problems still persists in the P2P approach. The synchronization of all the clients performing their respective tasks still is done by a central server, instead we are proposing this to be done using a smart contract with pre-defined logic to ensure fairness during client selection. Privacy concerns still exists as clients are now sharing their models with fellow clients, however this can be addressed by introducing reputation system and performing client selection based off that. Furthermore the single point of failure in case of centralised aggregation as discussed in the paper [4] by Zexin Wang et al., can also be addressed by introducing blockchain Incentive Mechanism by Zexin Wang, Biwei Yan, and Anming Dong. It presents a realistic image of how the FL can benefit from incorporation of blockchain. This solves the problem of single point of failure as well as data privacy to an extent.

Motivation

Decentralization of the FL currently solves a lot of its existing problems. There is an FL model requester making a general request on the Blockchain [4]. The data owner matching the needs train the models locally and upload it to the Blockchain platform. These transactions are recorded on the chain, and the node that aggregates the model first and generates the transaction is rewarded according to the guidelines of smart contract. However, the contest for aggregation of all the models as well generating the transaction might be computationally inefficient as well as time consuming. Since all the computation done by the nodes who lost are wasted and the competitive nature of the aggregation leads to potential of cheating as well as detriment to the quality of the model. Aggregation done at a centralized system also makes the system vulnerable to various attacks and single point of failure. Therefore in terms of centralized aggregation we believe that there is a scope for improvements by filling in the gap by introducing hierarchical aggregation.

Research Objectives

Our main objective is to eliminate centralization in the process of aggregation reliably. Other peripheral objectives include increasing trust between client by selecting aggregators within themselves in a fair and unpredictable manner, so no adversarial network can be run to reverse engineer data from model parameters. This increased transparency would also serve as an incentive for more clients to participate in the process of Federated Learning as data providers. To test our proposed design we will compare it against baseline models for aggregation in FL like FedAvg, Multi-KRUM etc., and evaluate if our proposed design meets the thresholds of efficiency both in terms of time and resources while still giving good accuracy.

Methodology

We'd like to solve this problem by parallelizing the task of aggregation. There are two types of nodes in the system, one set of nodes are called trainers and as the name suggests, they will train the models assigned to them by the smart contract. The second type of nodes are called the aggregator nodes, these nodes are further divided into multiple levels, and are thus called level-1 aggregators, level-2 aggregators etc. Each of the aggregators nodes at Nth level will get models from (N-1)th level to aggregate and this will continue on until there are no more aggregators in the level above a certain level, i.e., model has reached the final level (top) after which it can be successfully merged with the global model. The nodes will be selected according to the reputation scores. We will use smart contract to maintain the credibility (or reputation scores) and continually update them upon each round of FL. These scores will help in determining the trainer nodes and aggregator node at each level of the hierarchical aggregation. We will compare our aggregating model against presently used methods of aggregation such as FedAvg, Multi-KRUM, Multi-Boolean etc. We will adjust parameters such as number of nodes at each level of hierarchical aggregation and number of models per aggregating node. For federated learning we are using a simple Multi-layer perceptron for classification, we are using tensorflow keras for implementing the model, and sklearn for data processing and model metrics such as accuracy. The data set we are using is called swarm behaviour [1] and we will also try to test our model against other data sets such as MNIST [2].

Calendar Schedule

- Week 1 – Literature Review, to find the gaps and be on top of current research.
- Week 2 – Defining the problem statement and preparing proposal.
- Week 3 - 4 – Submitting Proposal and start establishing the code base.
- Week 5 – Running Experiments on already available models and aggregation methods
- Week 6 - 8 – Implementing our proposed solution of Hierarchical Aggregation
- Week 9 – Running benchmarks and comparisons
- Week 10 – Start writing the paper
- Week 11 - 12 – Refactoring the code base, handover and submission

References

[1] Dr Shadi Abpeikar, A/Prof Kathryn Kasmarik, A/Prof Michael Barlow, and Md Mohiuddin Khan. Swarm behaviour data set, 2020.

[2] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[3] University of Southern California Information Sciences Institute. Transmission control protocol darpa internet program protocol specification. RFC 793, RFC Editor, 9 1981.

[4] Zexin Wang, Biwei Yan, and Anming Dong. Blockchain empowered federated learning for data sharing incentive mechanism. *Procedia Computer Science*, 202:348–353, 2022. International Conference on Identification, Information and Knowledge in the internet of Things, 2021.

[5] Juncen Zhu, Jiannong Cao, Divya Saxena, Shan Jiang, and Houda Ferradi. Blockchain-empowered federated learning: Challenges, solutions, and future directions. *ACM Comput. Surv.*, 55(11), feb 2023.