Bounty Based Federated Learning

Prashanth Thipparthi
Department of Computer Science
University of Colorado at Boulder
prashanth.thipparthi@colorado.edu

***Abstract*—Federated Machine Learning (FML) enables multiple parties to collaborate on building models while protecting data privacy for the participants. A measure of the contribution for each party in FML enables fair credits allocation. In this paper we explore the usage of shapley values method as a measure and rewarding users by utilising block chain.**

***Index Terms*—federated learning, shapley values, block chain, distributed machine learning**

I. Introduction

In the current scenario, firms collect the data from the users and to have insights about the data they use standard machine learning approaches which require centralizing the training data on one machine or in a data center and utilizing robust cloud infrastructures for processing this data and training machine learning models.In this case, users give away their personal data and also they don't get rewarded for the same. One way to mitigate the above issues and also retrieve the insights from the data is Federated Learning[1]. Federated Learning is a distributed machine learning approach which enables model training on a large corpus of decentralised data and also offers users enhanced privacy. Federated learning was initially proposed in this paper[1]. The paper proposes to mitigate the issue of ownership and privacy of proprietary machine learning data.

There remains the concern of handing out our data, a useful resource to organizational training models, for free of cost. This paper(BBFL) proposes rewarding the devices which participate in the training process and also contribute their training parameters for improvement of the model. Devices are rewarded using the block chain network. There are some earlier researches in this directions and following are some of them.

BlockFL[] overcomes the single point of failure problem and extends the range of its federation to untrustworthy devices in a public network. It has a validation process of the local training results. Moreover, by providing rewards proportional to the training sample sizes, BlockFL promotes the federation of more devices with a larger number of training samples. The payment to devices is left to the miner to pay "out-of-pocket", which is not a scalable solution.

Deepchain[] presents a distributed, secure, and fair deep learning framework named DeepChain to solve these problems. DeepChain provides a value-driven incentive mechanism based on blockchain to force the participants to behave correctly. The blockwise-BA consensus protocol proposed relies on cryptographically selecting a worker to create a block which is validated by a committee; this method relies on choosing an honest committee, and for the random algorithm to be negligibly close to perfectly random, both issues which may not be true in practice.

Ethereum blockchain[] proposed an Ethereum blockchain implementation of machine learning to reward users for producing trained models for organizers. Given an organizer's published dataset and evaluation function, users compete to produce the first or the best training model that maximizes this evaluation function. One large problem that arises with this system is that all model evaluations are done on the blockchain which yields large gas costs; many users must each pay gas for their models to be evaluated, however only one or two users are paid out. Users needing to pay large gas prices in addition to effort and time into building and training a machine learning model for submission without a guarantee of repayment does not yield a sustainable system

The main contributions of this paper can be summarized as follows:

- Merging Federated Learning with blockchain to ensure both data privacy and security, and thus motivate more user contributions.
- Simple implementation with Python and Hyperledger Fabric to verify the feasibility of the system, with plans to implement a better measure for reward at a later date

II. Design and Architecture

In the system architecture, there will be a central parameter server which chooses the random devices from the available devices and starts the training process by using the data and compute power of the devices. After completing the training process it rewards all the devices equally. In the next submission, I want to use shapley values or other suitable algorithm after experimentation for rewarding the devices.

The rewarding process happens when the central parameter server requests the block chain framework to reward the appropriate points to the particular device.

In the training process central parmater server connects to the devices using the websockets and transfers the initial model to all the devices and when a particular device is selected for the training process, it sends a request to device to start the training process and after sometime it requests the weights from all the devices participated in the training and aggregates the weights received from all the devices and updates the central parameter server model. Then clients pull the updated model from the central server and updates the local model.

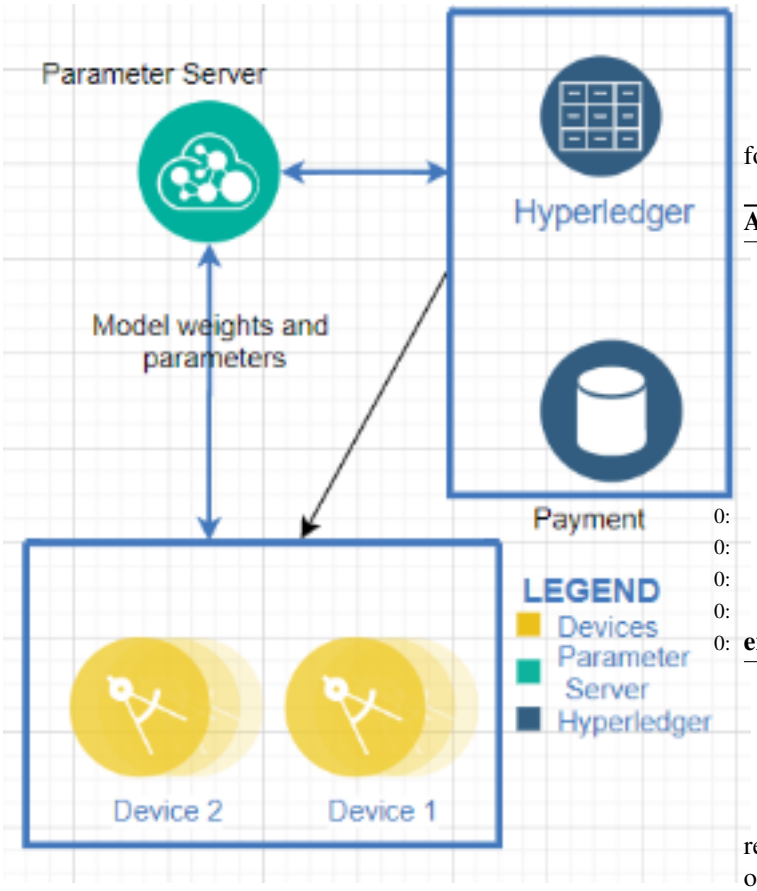

Figure 1. Architecture diagram

Central parameter server requests the block chain framework to reward the devices which participated in the training process and appropriately a block is added in the networl

III. Experiments and Implementation

In the current implementation, there are two google cl machines which are used as the devices and a another macl which is used as the central parameter server. Data set util for experiments consists approximately 20,000 surnames f 18 languages of origin, and predict to which languag name belongs based on the name's spelling. Recurrent Ne Networks are deployed on the devices for the same.

PySyft open source framework is used to implement federated learning on the experimental setup.As part of experiment devices will be opening their web sockets rel to the federated learning and central parameter server will s the rnn model to the devices. Following is the forward m defined in the pytorch

combined = torch.cat((input, hidden), 1)
hidden = i2h(combined)
output = i2o(combined)
output = softmax(output)

and following are the model training parameters

batchsize = 1
learningrate = 0.005
epochs = 10000
federateAfterNbatches = 15000
seed = 1

After every 'n' iterations federated averaging is done in the following manner.

Algorithm 1 Federated Averaging

if $iter\%args.federateAfterNbatches == 0$ **then**
  for All workerName, modelPointer in modelPointers.items() **do**
    modelsLocal[workerName] = modelPointer.copy().get()
  end for
  modelAvg = utils.federatedAvg(modelsLocal)
  for worker in workers$_virtual$ **do**
    modelCopiedAvg = modelAvg.copy()
    modelPtr = modelCopiedAvg.send(worker)
    modelPointers[worker.id] = modelPtr
  end for
end if=0

IV. Initial Results

We can see from the below graph the loss function value reduced on the central parameter server as we train the model on for more number of epochs. Experimental results as shown that predictions on the devices varies as the data is distributed randomly on to the devices.

Prediction results converge on the devices after certain number of epochs.

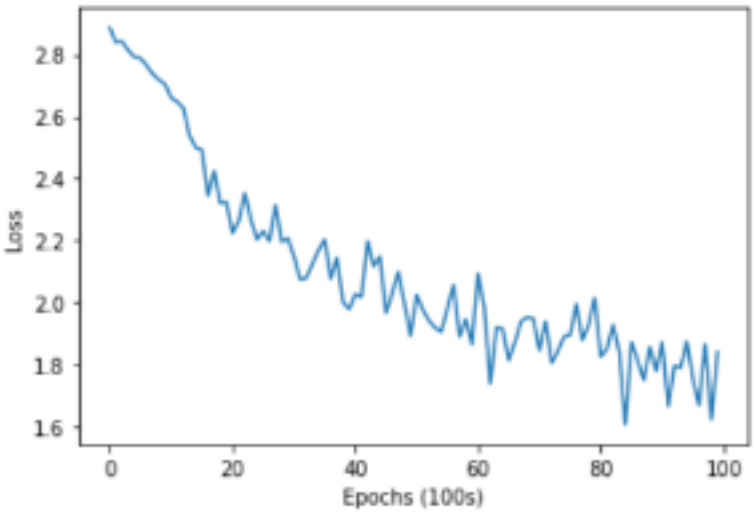

Figure 2. Loss function value in devices.

The code for the federated learning can be found in the follwing git repository. github.com/prashanth-thipparthi/distributed$_systems_project$

REFERENCES

[1] B. McMahan and D. Ramage, “Federated Learning: Collaborative machine learning without centralized training data”, Google Research Blog, 2017.

[2] J. Koneny, H. B. McMahan, F. X. Yu, P. Richtrik, A. T. Suresh, and D. Bacon.“Federated learning: Strategies for improving communication efficiency”, arXiv preprint arXiv:1610.05492, 2016.

[3] I. Martinez, S. Francis and A. S. Hafid, ”Record and Reward Federated Learning Contributions with Blockchain,” 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Guilin, China, 2019, pp. 50-57.

[4] H. Kim, J. Park, M. Bennis, and S.-L. Kim. “On-device federated learning via blockchain and its latency analysis”, arXiv preprint arXiv:1808.03949, 2018.

[5] J. Weng, J. Weng, J. Zhang, M. Li, Y. Zhang, and W. Luo, “Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive”, Cryptology ePrint Archive, Report 2018/679, 2018.

[6] B. Kurtulmus and K. Daniel. “Trustless machine learning contracts;evaluating and exchanging machine learning models on the ethereum blockchain”, arXivpreprint arXiv:1802.10185, 2018.