Bounty Based Federated Learning

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

***Abstract*—The goal of the project is to address two of the major problems in the machine learning field. One is the privacy and availability of the data and the other is compute power required for the training of models.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 study the advantages of FML over standard machine learning, explore the feasibility of rewarding the users participating in the learning process of the model using block chain by building a proof of concept(PoC) and propose a sustainable and scalable architecture from the findings of the PoC.**

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

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 by introducing the FML method.

II. Related Work

There remains the concern of handing out our data, a useful resource to organizational training models for free of cost. This paper(BBFL) introduces a brief comparative study of federated learning over standard machine learning and studies the feasibility of rewarding the devices which participate in the training process and contribute their training parameters for improvement of the model by building a prototype. Devices are rewarded using the block chain network.

There are some of the earlier theoretical researches in this directions and following are some of them. Kunal et al[14] study clearly shows that federated learning reduces the training time and can be done in a iterative process by utilising the distributed computing resources. They key observation is that the accuracy difference between the federated and standard ML is low but there was a significant difference in time taken to train the models where the time taken by the central model is much more than the federated learning as it uses the compute resources of many devices and as the number of training rounds increases the accuracy increase in the federated learning model is high compared to the standard ML model.

BlockFL[4] 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[5] 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[6] 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 block chain 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

FedCoin[7] proposes the use of block chain network for computation of the data valuation. In the BBFL we'll discuss in details about the algorithms that can be used and the payment systems in detail.

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

- Implementing a Proof of Concept of the proposed system where users are rewarded using the block chain.

- Proposing an enhanced architecture from the learnings of Proof of Concept(PoC) where the data evaluation can be done in a sustainable way by reducing the computational cost.

III. Design and Architecture

As shown in the figure 1, In the system architecture, there are two modules: 1) Federated learning network. 2) Block chain network. In the FL network there will be a parameter server which initiates the training process by utilising the random devices from the available devices by using the data and compute power of the devices. After completing one round of the training it aggregates the weights, updates the model and then initiates the other round of training process. When training process is completed it rewards all the devices participated in the process equally. To reward the devices, parameter server receives the unique ids of the each device and sends a request to the flask server in the block chain network with the details about the devices and the appropriate amount to reward for each device. Flask server maintains the mapping of the device id with the appropriate block chain address of the particular device in the block chain network. Flask server upon receiving the request using the web3 package in python signs the transaction using the private key of the sender and sends a request to the block chain network to process the transaction.

Block chain network upon receiving the request creates a block and adds it to the block chain and completes the transaction. Devices can view their transactions, balance and rewarded amounts by using the wallets provided by the network. In the final step, devices fetch the updated model from the central server and updates their local models.

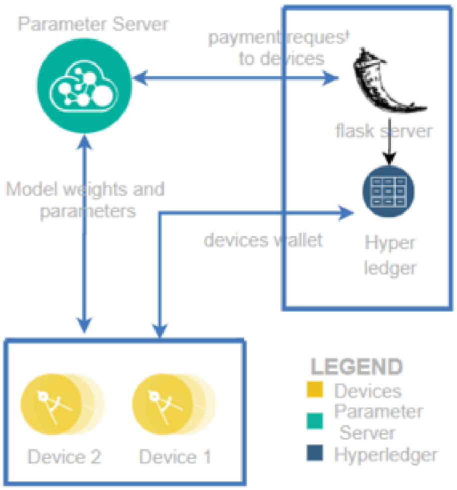


Figure 1. Architecture diagram

IV. Experiments and Implementation

The PySyft open source framework whose architecture is shown in the Figure 2, supports the encrypted, privacy preserving machine learning. It is used to implement the federated learning part of the architecture. It is built on top of the PyTorch framework. The block chain part of the architecture uses Ethereum framework and Ganache private block chain network for the payment related transactions. Apart from that all the modules are written in python and uses frameworks and packages like web3 and flask.

In the experiment as the first step, data is split across the multiple devices before the training starts. Using the PySyft framework, central server creates a PyTorch hook and distributes it to the clients. When the round is initiated by the central server, RPC calls are made to the clients and using the hook required operations are performed on the clients. Weights from the available clients are aggregated and a update is made to the central model when the round is finished. Details of the client ids participated in the training process is collected and a request is made to the flask server to reward the clients. Appropriate payment is made to the clients using the block chain network.

Data set used for the experiments consists approximately 20,000 surnames from 18 languages of origin and after the training, language of origin is predicted given a name.Recurrent Neural Network of the following architecture is deployed on the devices. 5 layer model with 128 neurons. 1 output layer to predict the language of origin. Negative log likelihood loss (NLL loss) function is used as the loss function.

Following is the overview of the various library components in the architecture.

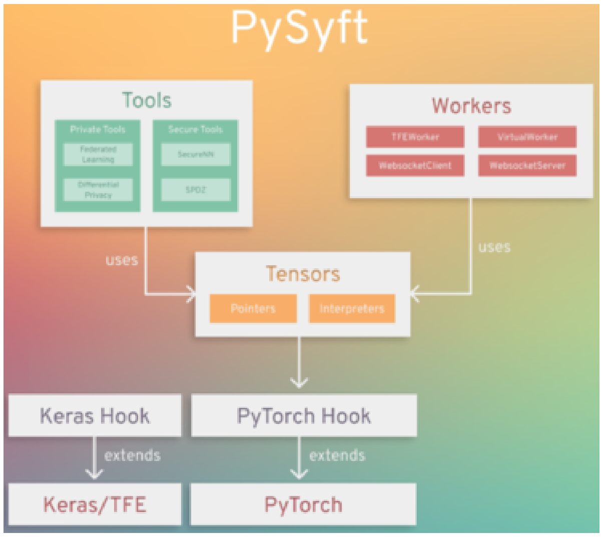


Figure 2. Loss function value in devices.

Following federated averaging algorithm is extended from the research of Han Yu and et al [7]. Federated Learning scenario with federated averaging algorithm: In a regular Federated learning scenario, we take $\mathcal{F}_{\rangle}(w) = loss(xi, yi; wt)$ as the prediction loss on a sample (xi, yi) with model weight parameters w at the t-th round. The weight parameters w^t is a d-dimensional vector. We assume that there are K clients, and each client has a Dataset locally, D_k with $n_k = |Dk|$.

The overall dataset is $D = \{D_1, D_2, ..., D_k\}$ with $n = |D| =$. The objective function that should be optimized is:

$$\min_{w \in \mathcal{R}^d} \mathcal{F}(w) \quad \text{where} \quad \mathcal{F}(w) = \frac{1}{n} \sum_{k=1}^{K} \sum_{i \in \mathcal{D}_k} \mathcal{F}_i(w)$$

Optimisation problems of the above nature are solved using the Stochaistic Gradient Descent (SGD) based methods. After computing the average gradient on the local data of the client, k. Each clent updates its local model and the central server aggregates the local model as the global FL model.

$$w_{t+1} \leftarrow \mathcal{A}\left(\{w_{t+1}^{k} | k = 1, \ldots, K\}\right)$$

where $\mathcal{A}$ is an aggregate function.

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

Algorithm 1 Federated Averaging

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

We can see from the graph in the Figure 3 that 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.

V. Results from the observation of POC

Some of the problems observed in the above design and architecture is that we are rewarding all the devices without validating and calculating the value of the data contributed by the users and rewarding all the users with the equal amount of money. This will not be a sustainable approach and will not be fair to the devices which are contributing the more valuable data. To solve this problem and as a add on to the existing research I propose the following design where the shapley values which are usually used in the cooperative game theory to calculate the contributions of users in a collaborative environment to be used to calculate the value of contributions made by the users.

The usage of shapley values for the data valuation is a compute heavy process and the complexity of computing grows exponentially as the number of devices increases. Following is the brief explanation of how the shapley values are calculated.

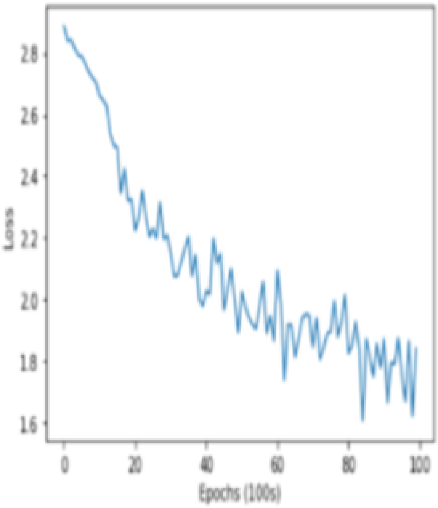
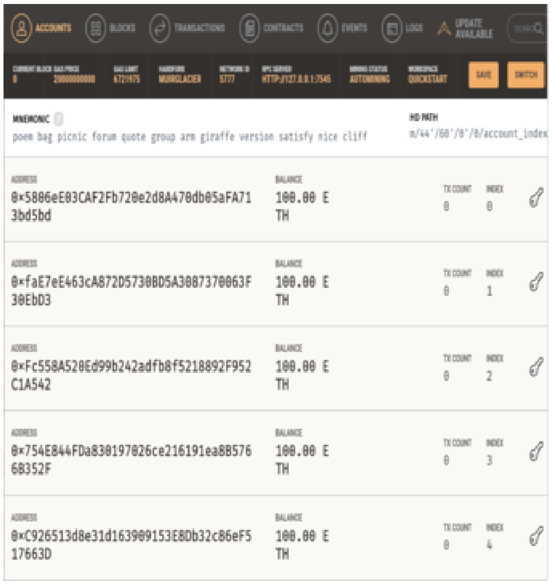


Figure 3. Loss function value in devices.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER	MINING STATUS	WORKSPACE
0	20000000000	6721975	MUIRGLACIER	5777	HTTP://127.0.0.1:7545	AUTOMINING	QUICKSTART

MNEMONIC
poem bag picnic forum quote group arm giraffe version satisfy nice cliff

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×5806eE03CAF2Fb720e2d8A470db05aFA713bd5bd	100.00 ETH	0	0
0×faE7eE463cA872D5730BD5A3087370063F30EbD3	100.00 ETH	0	1
0×Fc558A520Ed99b242adfb8f5218892F952C1A542	100.00 ETH	0	2
0×754E844FDa830197026ce216191ea8B5766B352F	100.00 ETH	0	3
0×C926513d8e31d163909153E8Db32c86eF517663D	100.00 ETH	0	4

Figure 4. Loss function value in devices.

After all the contributions are made by the users and the model is trained for the particular round, we calculate the increase in accuracy. In the next steps, we remove certain users data contribution, train the model and calculate the accuracy. In the similar way, we remove users data contributions in various combinations and finally arrive at the value of each user data and reward accordingly. In reality it uses complex formulae but to make it intuitive, I explained it in a straight forward approach. So, to reduce the computational complexity, I propose using the approximation algorithms which reduce the time complexity of the computation from the exponential to polynomial time.

In the architecture shown in Figure 5, block chain network of miners are utilized to compute the shapley values which are used for data evaluation instead of computing meaningless puzzles as in the traditional block chain. Following will be the payment scheme for the architecture. At first, the model requester who wants to start the training of the model will deposit M amount of coins with the central server. M coins are divided appropriately according to their contribution and paid to the FL clients, central server and the miners in the block chain network for computing the shapley values. Here the M amount should be fixed in a way that the total cost for computation should not exceed M amount of coins. Therefore, we can be sure that the user who wants to initiate the model will pay for the computation cost and all the stake holders in the system are rewarded according to their contribution and we can ensure that the entire system will be economically sustainable where all the stakeholders are charged or rewarded accordingly.

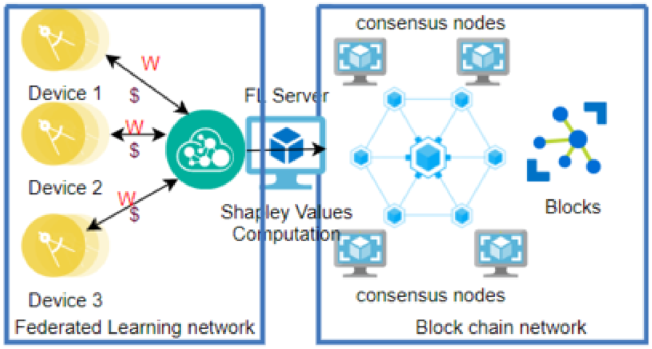


Figure 5. Loss function value in devices.

The presentation video for the project can be viewed in the you tube using the below link. YouTube

VI. Conclusion and Future work

From the observed results, we can conclude that federated learning provides a feasible solution to some of the problems in machine learning like the data privacy, availability and compute power and also we can reward the devices which provide the significant data points in improving the model. Project involved two parts: federated learning and the block chain and it was challenging and exciting to work on both the parts and complete the PoC to demonstrate the feasibility. As my future work, I want to work on the architecture proposed in the paper and contribute to the area of decentralised Artificial intelligence to the best of my ability.

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.

[7] Yuan Liu, Shuai Sun and Han Yu. ”FedCoin: A Peer-to-Peer Payment System for Federated Learning.” Feb 2020.

[8] R. C. Geyer, T. Klein, and M. Nabi, “Differentially Private Federated Learning: A Client Level Perspective”, arXiv:1712.07557, 2017

[9] EOSIO.”Eos.io”. technicalwhitePaperV2, https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md, cited May 2019

[10] https://github.com/OpenMined/PySyft

[11] https://github.com/coMindOrg/federated-averaging-tutorials

[12] [Bonawitz et al., 2019] Keith Bonawitz, Hubert Eichner,Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecn ´ y, Stefano ´Mazzocchi, H. Brendan McMahan, Timon Van Overveldt,David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: System design. CoRR,abs/1902.01046, 2019.

[13] [Khan et al., 2019] Latif U. Khan, Nguyen H. Tran,Shashi Raj Pandey, Walid Saad, Zhu Han, Minh N. H.Nguyen, and Choong Seon Hong. Federated learning for edge networks: Resource optimization and incentive mechanism. CoRR, abs/1911.05642, 2019.

[14] Kunal Chandiramani, Dhruv Garg, N Maheswari. Performance Analysis of Distributed and Federated Learning Models on Private Data. International Conference On Recent Trends In Advanced Computing