Incentivizing Participation to Distributed Neural Network Training

Spyridon Nikolaidis[1[0000-0001-6450-1005]] and Ioannis Refanidis[1[0000-0003-4697-4751]]

[1] University of Macedonia, Thessaloniki 54636, GREECE
{sp.nikola, yrefanid}@uom.edu.gr

Abstract. During the last years a vast number of online sensors continuously generate data that can be utilized to create novel deep learning applications. Training very large models requires enormous processing power; thus, the evident way to follow is to lease the power of a corporate data center. But the diffusion of Artificial Intelligence to an always increasing number of human activities, constantly attracts new researchers who wish to train and test their models. Our work on LEARNAE is a proposal for a purely distributed neural network training, based on a peer-to-peer and permissionless architecture. LEARNAE allows individual researchers to join forces, in order to collaboratively train a model. The process utilizes modern Distributed Ledger Technology and it is fully democratized, prioritizing decentralization, fault tolerance and privacy. In this paper we add another piece to the puzzle: A method for incentivizing peers to participate to the training swarm, even if they don't have any interest in the produced neural network. This is achieved by embedding a reward subsystem to LEARNAE; thus, peers who contribute to teamwork can receive a proportional digital payment.

Keywords: Decentralized Neural Network Training, Distributed Ledger Technology, Smart Contracts

1 Introduction

1.1 Synopsis of architecture

In Artificial Neural Network (ANN) training, many proposals claim that they are not centralized, but they support many different levels of decentralization. Ranging from low to high degree of decentralization, literature contains implementations with a parameter server, a cluster of parameter servers, peers with elevated role and, finally, pure peer-to-peer topologies. Our proposal is based on the last scheme, where all participating nodes have the same privileges and none of them is essential to the training process.

LEARNAE takes advantage of novel Distributed Ledger Technology (DLT), using it as data diffusion mechanism. The management algorithm that runs on peers is platform-agnostic; current implementation adopts two novel technologies: (a) IPFS [4], a fully decentralized filesystem and (b) IOTA [5], a decentralized infrastructure for the upcoming world of Internet of Things (IoT). Such choices ensure high resilience, since all information is propagated using gossip protocols, leaving no single point of failure.

LEARNAE implements data parallelism [6][7], according to which each worker keeps the entire model locally and trains it using available data. After being processed on workers, the produced models need to be merged, which is achieved by parameter averaging. During this stage, all parameters of the local model are averaged with the corresponding parameters of a selected remote model [8]. This introduces additional stochasticity, improving the overall accuracy. In use cases where privacy is not a priority, nodes can also exchange training data, using the same decentralized mechanism.

The collaborative training procedure is designed to work well with commodity hardware. No kind of synchronization is needed, and all data remain on the network for the peers to use them at their own pace. Furthermore, there is no indirect leakage, since the broadcasted models are not produced only from the private data of each peer, but they are increasingly affected by the weights of remote models created by neighbors, making reverse engineering practically impossible.

1.2 Our previous work

In [1] we proposed the fundamentals of a novel architecture (LEARNAE) that utilizes different types of Distributed Ledger Technology, to create an ecosystem for decentralized ANN training. The proposal assumed loosely connected peer-to-peer topologies, with unreliable connections and unpredictable downtimes. [1] defined the specifications of four different roles that nodes could select from, depending on processing power and data availability. The proposal was tested by emulating a training swarm of 10 peers on a single machine, using virtualization techniques.

In [2] we tested the proposed algorithm under real-life conditions. LEARNAE was deployed to a local network of 15 personal computers with commodity hardware and networking. The experiments studied the progress of the averaging process and resulted to tangible gains in terms of model accuracy.

In [3] we expanded deployment to a group of 20 Virtual Private Servers (VPS) and evaluated the resilience achieved through data duplication. For this purpose, a new subsystem was implemented, which emulated network disruptions and peer downtime. LEARNAE managed to withstand critical disconnections with no performance degradation to the produced model. [3] also introduced a novel way to embed low-energy IoT sensors, without compromising the overall decentralized philosophy.

In this paper we propose a way of incentivizing peers to participate to collaborative training. So far, a LEARNAE swarm consisted of nodes that had interest in the outcome and joined forces to achieve better model accuracy. Now a novel reward mechanism is added; peers can participate to a session and profit from their constructive averaging. This is achieved by embedding a gateway that can communicate with blockchains to both publish and acquire data. The design is platform agnostic and can work with any blockchain that has the ability to execute code. Having a credit system in place, peers have the ability to reward helpful neighbors, by sending digital micro-payments to them each time they contribute to a successful averaging

The rest of this paper is structured as follows: Section 2 presents the underlying Distributed Ledger Technology, Section 3 presents the architecture of our proposal,

Section 4 presents the experimental results and, finally, Section 5 concludes the paper and poses future research directions.

2 Distributed Ledger Technology

In our previous work, LEARNAE utilized two DLT projects: IPFS, for data diffusion, and IOTA, for IoT communications. In this paper we add the Ethereum Network [9], which is used (a) for "Proof of Identification" via its "Smart Contract" infrastructure, and (b) as a Payment platform for peer incentivization.

The underlying algorithm is platform-agnostic and is able to use any blockchain ecosystem that offers the needed functionality. Ethereum is selected here because of its wide adoption. To overcome monetary cost, for the conducted experiments we didn't use the main network (MainNet), but a testing network (TestNet) of Ethereum, called Goerli.

2.1 Blockchain fundamentals

Blockchain [10][11] is the first and most used type of DLT, comprised by a connected list of "blocks". Although this definition resembles to many other traditional data structures, blockchain introduced a plethora of novel concepts. Nowadays there are many different types, each one with its own special features. This brief presentation will cover the most common characteristics one can find in almost every implementation.

A blockchain is a distributed ledger; snapshots of its database are stored to multiple nodes. There is no central entity that is needed for coordination and all peers contribute to maintain its integrity. The consensus is achieved via well-established distributed algorithms [12], whereas each time there is a conflict, the majority decides which version of truth will be accepted. A new block is published periodically, containing info about changes that have to be applied to the blockchain. To maintain a strong coherence, every block contains a cryptographic hash of the previous one. In this way, malicious block altering would be easily tracked, since it would cause inconsistency. An important feature of blockchains is their immutability, meaning that users can only add info and cannot alter any of the previous data. So, the only way to update blockchain's status is to append the needed changes.

Almost all blockchains support a native token [13], which is a digital asset that can be used in transactions. Every participant has a unique pair of a private key and a public address. Blockchain consensus mechanism and asymmetric cryptography methods ensure that all token balances are protected and there can be no double spending.

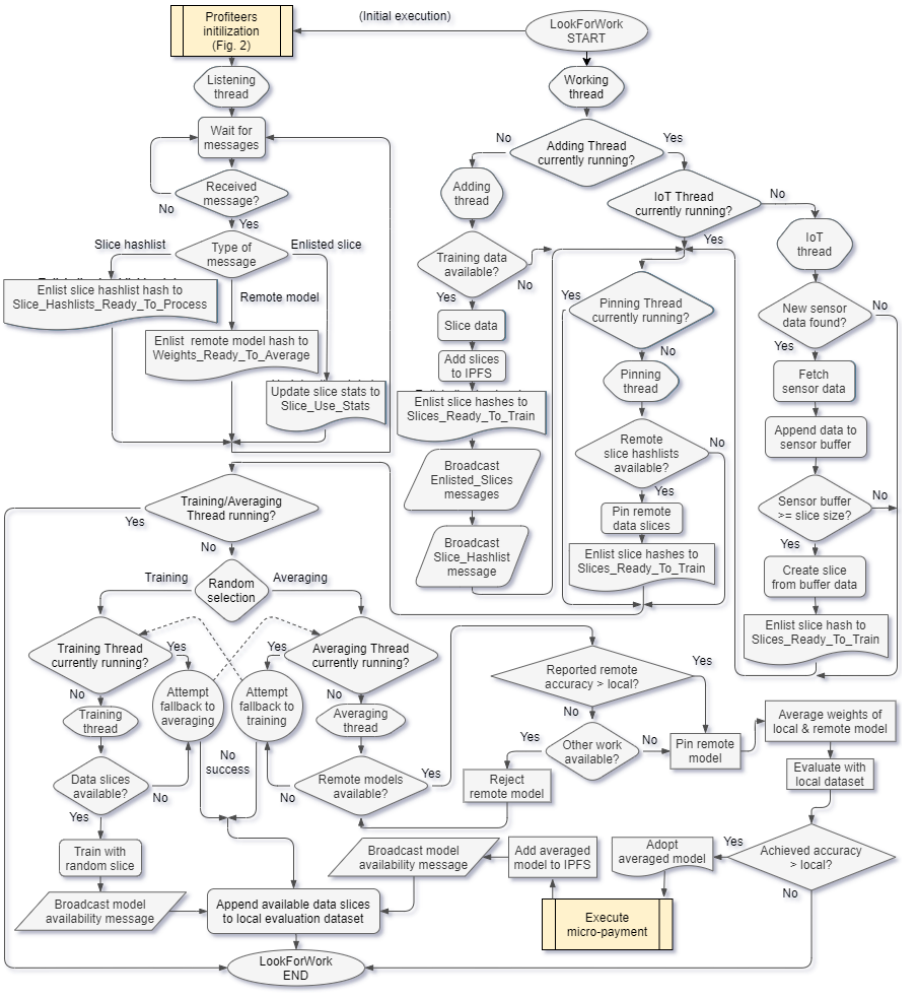

Fig. 1. The major parts of a node's workflow (incentivization in yellow color)

2.2 Smart Contracts

The first generation blockchains had a single goal: To secure token balances in a fully distributed and permissionless way. The Ethereum Network, as a second generation blockchain, introduced additional functionality. Its node software includes a virtual machine (Ethereum Virtual Machine – EVM) [14] that can execute pieces of code called "Smart Contracts" [15][16]. After code execution, the results are evaluated by the network's consensus mechanism. Smart Contracts have also the ability to store limited data and perform token transactions. The code of a published Smart Contract is available for everyone to review [17].

3 Proposed architecture

So far, peers joined a LEARNAE swarm to obtain improved models. This paper proposes a way to incentivize peers, by offering profit for their constructive averaging.

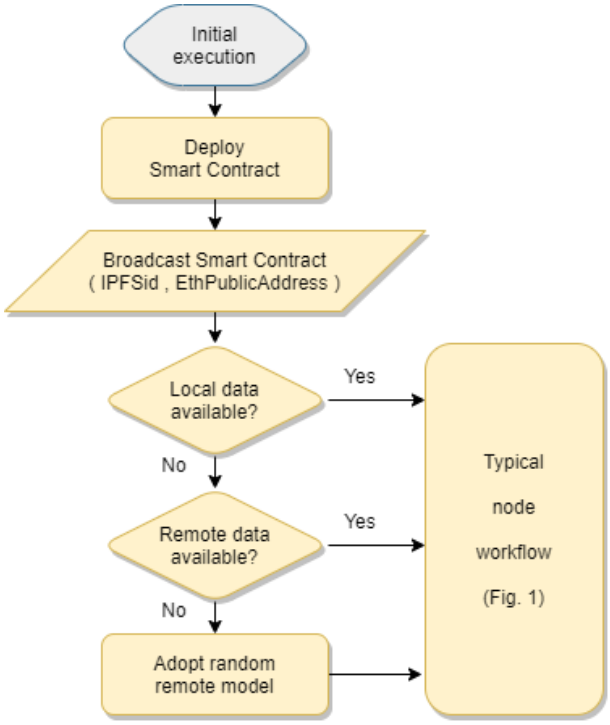

Fig. 2. Additional workflow section

Our previous work [1][2][3] contains a detailed presentation of a node's workflow. Fig. 1 shows how the new parts are embedded into the existing workflow, while Fig. 2 demonstrates the additions that have to be taken, in order to implement the incentivization subsystem. At first execution, a peer that wishes to receive payments, must deploy LEARNAE's Smart Contract to Ethereum Network (details in section 3.2). Next, it has to inform its neighbors by broadcasting a new type of metadata message, which contains its IPFS identification and the Ethereum public address of its Smart Contract.

If the LEARNAE session is in privacy mode, no training data will be transmitted, so peers that joined just for the reward will most likely have no data of their own. In these cases, in order to properly initialize their local model, peers will randomly adopt a remote model that was made available to the network. After that, they will fall back to typical node workflow.

3.1 Incentivization algorithm

Every time a peer improves its local model by averaging with a remote one, its incentivization algorithm sends an Eth micro-payment to the creator of the remote model. The amount of this payment is defined by a Reward Function, which takes into consideration the amount of improvement that occurred to local model accuracy. For the conducted experiments, in order to evaluate the whole procedure, we used a simple proportional formula:

$$Payment = RewardFactor * (AchievedAccuracy - CurrentAccuracy) \quad (1)$$

In general, any Reward Function can be used, linear or not. Eventually, rule of Supply and Demand will lead the participants to the appropriate reward level. This first implementation assumes that the peers who participate and have interest in the produced model are benevolent and they will reward their helpful neighbors, in order to maintain a connection of trust, since they benefit from it. We acknowledge that this approach can be enhanced to anticipate bad actors who will enjoy the contribution of others without rewarding them, thus research to this direction, under the scope of Game Theory, will be a part of our future work.

3.2 Distributed proof of identification

For the incentivization to work, peers have to be able to prove their identity. This would be extremely easy by using a certificate authority, but in LEARNAE's case every aspect owes to be fully decentralized. For this reason, we use the Ethereum's Smart Contract infrastructure, in order to provide a Distributed Proof of Identity (DPoID) mechanism.

When joining a LEARNAE session, each peer has to deploy a specific Smart Contract to Ethereum blockchain. This contract contains two data fields, "PoID" (Proof of Identification) and "Timestamp". The contract's constructor method is automatically executed upon creation. The creator-peer passes one parameter (PoID) to it, which is comprised by its IPFSid and the public Ethereum address of its digital wallet. The constructor internally assigns current date and time to Timestamp field (Fig. 3).

After deploying the Smart Contract, the peer broadcasts the contract's address to its neighbors. All the other nodes execute this Smart Contract's $getPoid()$ and $getTimestamp()$ on the blockchain, to acquire the needed data. Every node maintains a local directory, which contains the IPFSid, the public Ethereum wallet address and the timestamp of every Smart Contract broadcasted to the network.

For example: Peer A improves its local model via averaging with a remote model sent from peer B. Peer A scans its local directory, finds the Ethereum public wallet address that corresponds to B's IPFSid, and sends a micro-payment to that address.

A malicious actor could attempt to highjack payments by broadcasting Smart Contracts that contain its own Ethereum public wallet address but with the IPFSid from another – more active – peer. This attempt would be recognized and ignored by its neighbors. That is because the same IPFSid would be linked with two different Ethereum public addresses. In such cases, the Smart Contract with the earliest timestamp is accepted, as any other is deemed fraudulent.

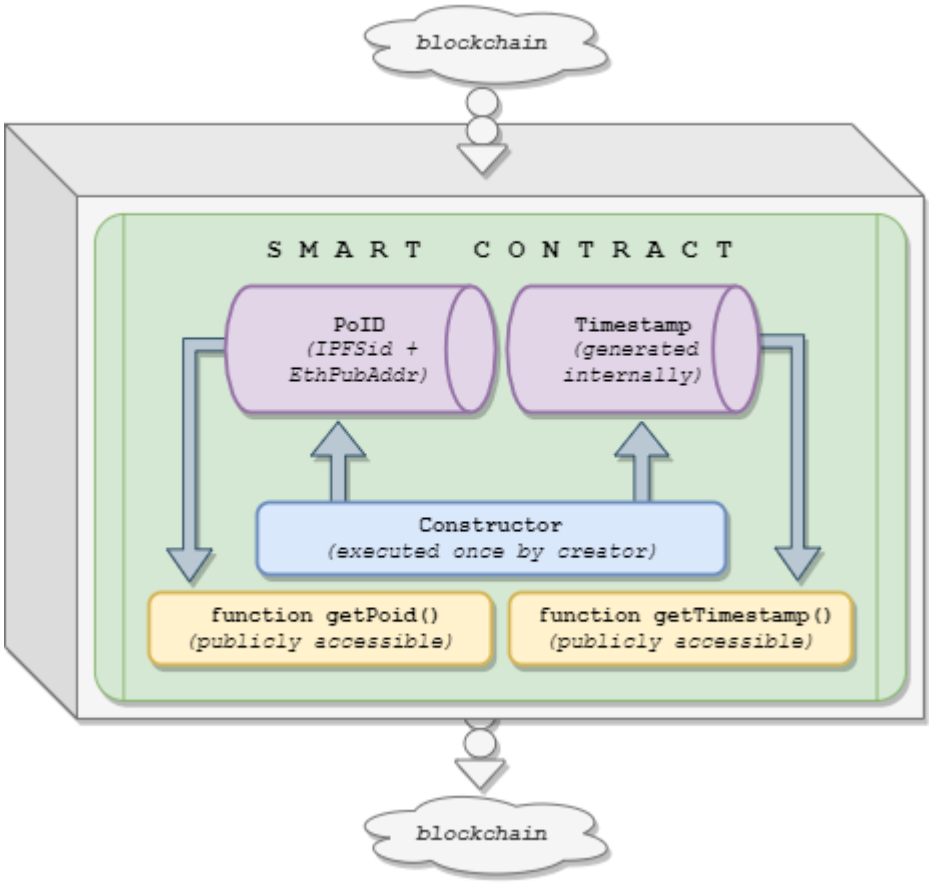

Fig. 3. Structure of DPoID Smart Contract

4 Conducted experiments

[1][2][3] showed that benefits – in terms of model accuracy – from collaborative training can be seen even in cases with a small number of participants. In this paper we focus on some preliminary metrics concerning the incentivization system.

The following graphs represent a LEARNAE session of 20 peers. Each of them had a local dataset of 10000 instances, which was sliced to 10 parts of 1000 instances, to facilitate averaging. Data privacy was enabled, meaning that no training data were shared among the nodes. The dataset used was HEPMASS[1].

[1] http://archive.ics.uci.edu/ml/datasets/hepmass

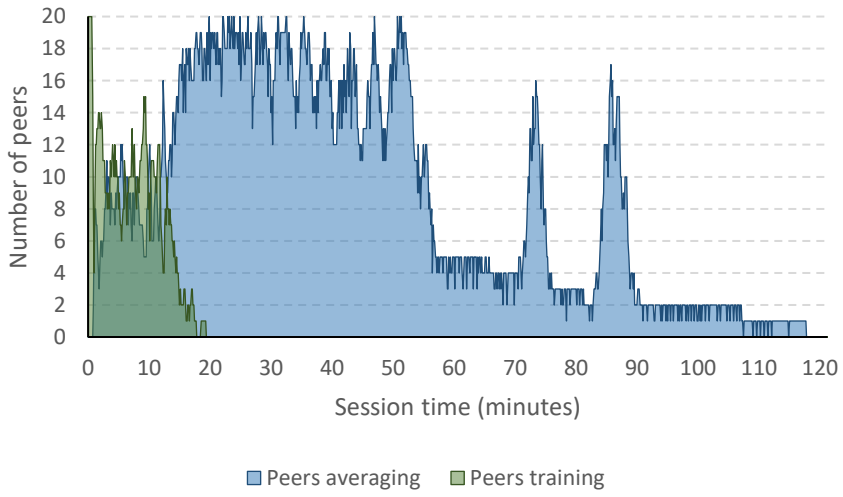

Fig. 4. Work type distribution

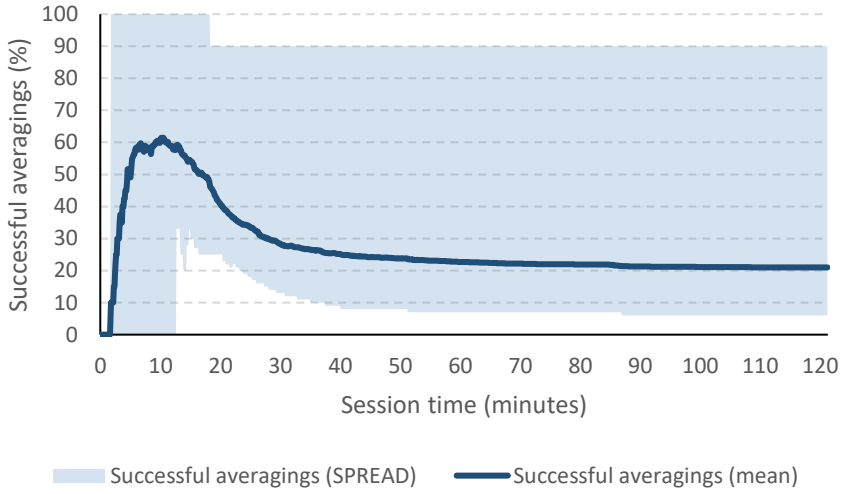

Fig. 5. Cumulative success rate of averaging process

As seen in Fig. 4, the initial phase is equally shared between training and averaging. When all available data are consumed, peers dedicate all of their time to averaging attempts. Finally, when the swarm approaches convergence, averaging cannot offer more model improvement and session concludes. As displayed in Fig. 5, at the beginning the percentage of successful averagings is high, and peers rapidly benefit from the models shared by their neighbors. Even at the end of the session, the mean percentage

of successful averagings is above 20%. Fig. 6 shows the progress of mean model accuracy. Detailed analysis of these graphs can be found in [1][2][3].

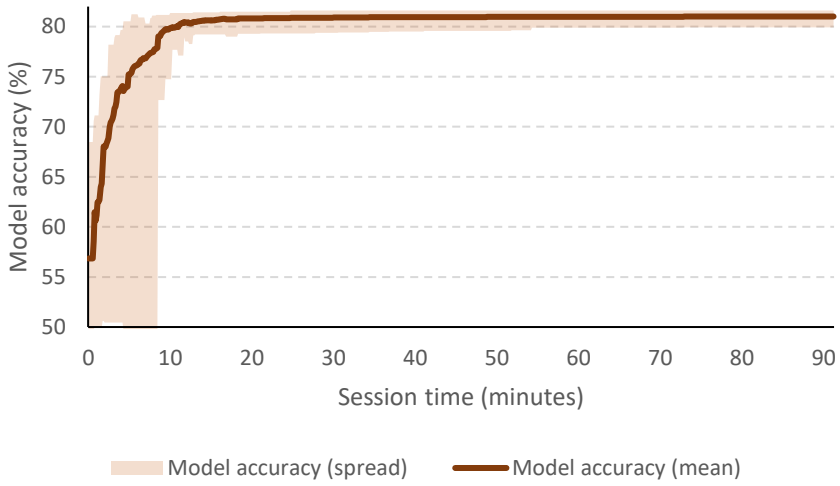

Fig. 6. Progress of model accuracy

Fig. 7 shows the number of Eth payments throughout the session. For the conducted experiments, peers sent micro-payments 3-16 times (with a mean value of 9.4).

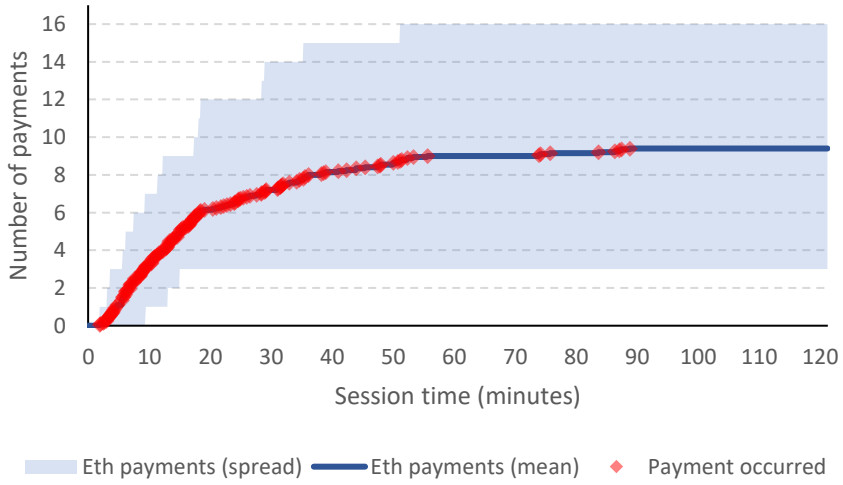

Fig. 7. Cumulative number of Eth payments

As shown in Fig. 8, during the collaborative training, peers sent 0.0005-0.0068 Eth to their neighbors (with a mean value of 0.003). It is important to outline that the conducted experiments serve as a proof of concept; the actual values of a real-life session

would be auto-regulated by supply-and-demand of processing power. So, every session could use pre-agreed fee values or, alternatively, nodes could adapt the fees dynamically to maintain the interest of profiteers. This feature will be studied in a future version of our coordinating algorithm.

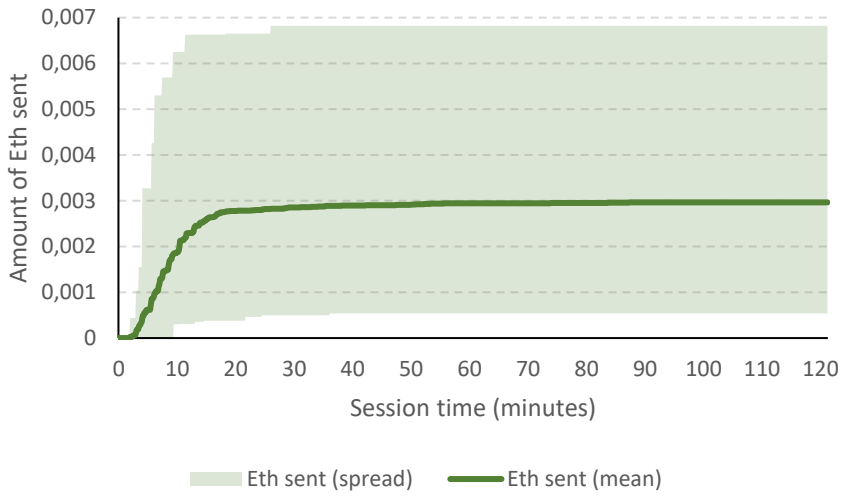

Fig. 8. Cumulative amount of Eth sent

5 Conclusions and future work

In this work we extended our previous research on Distributed Neural Network Training, by adding a subsystem that incentivizes peers to participate to a LEARNAE session. In this scenario, peers with no interest in the produced model can join the swarm, in order to benefit from their constructive averaging. We implement a method for a fully Distributed Proof of Identification and we conduct proof-of-concept experiments to evaluate the basic metrics.

There are still many issues with significant research interest. The incentivizing algorithm must be tested on: (a) Sessions without data privacy; in that way, peers that join just for profit would contribute their processing power to both training and averaging. (b) Sessions with two different groups, profiteers and data providers. Such experiments would allow a close study on the digital asset flow between these two groups. Another extensive research field is to find a fully distributed way to anticipate participants that refuse to pay rewards to their neighbors, taking into consideration concepts proposed by Game Theory. All of the above will be the subject of our future work.

References

1. Nikolaidis S, Refanidis I (2019) Learnae: Distributed and Resilient Deep Neural Network Training for Heterogeneous Peer to Peer Topologies. International Conference on Engineering Applications of Neural Networks 286-298. https://doi.org/10.1007/978-3-030-20257-6_24
2. Nikolaidis S, Refanidis I (2020) Privacy preserving distributed training of neural networks. Neural Comput & Applic. https://doi.org/10.1007/s00521-020-04880-0
3. Nikolaidis S, Refanidis I (2021) Using Distributed Ledger Technology to Democratize Neural Network Training. [Accepted for publication] Applied Intelligence Journal. Manuscript number #APIN-D-20-02984R1
4. Benet J (2014) IPFS - Content Addressed, Versioned, P2P File System. arXiv:1407.3561
5. Popov S, Saa O, Finardi P (2018) Equilibria in the Tangle. arXiv:1712.05385
6. Zhang X, Trmal J, Povey D, Khudanpur S (2014) Improving deep neural network acoustic models using generalized maxout networks in Acoustics, Speech and Signal Processing (ICASSP). IEEE International Conference
7. Miao Y, Zhang H, Metze F (2014) Distributed learning of multilingual dnn feature extractors using gpus
8. Dean J, Corrado GS, Monga R, Chen K, Devin M, Le QV, Mao M, Razato M, Senior A, Tucker P, Yang K, Ng AY (2012) Large Scale Distributed Deep Networks. Advances in Neural Information Processing Systems 1223-1231
9. Buterin V (2014) A next-generation smart contract and decentralized application platform. White Paper
10. Nofer M, Gomber P, Hinz O, et al (2017) Blockchain. Bus Inf Syst Eng 59, 183–187. https://doi.org/10.1007/s12599-017-0467-3
11. Pilkington M (2016) Blockchain technology: principles and applications. Research handbook on digital transformations. Edward Elgar Publishing
12. Mingxiao D, Ma X, Zhang Z, Wang X, and Chen Q (2017) A review on consensus algorithm of blockchain. IEEE international conference on systems, man, and cybernetics (SMC) 2567-2572
13. Phillip A, Chan JS, Peiris S (2018) A new look at cryptocurrencies. Economics Letters 163 6-9
14. Hildenbrandt E, Saxena M, Rodrigues N, Zhu X, Daian P, Guth D, Rosu G (2018) Kevm: A complete formal semantics of the ethereum virtual machine. IEEE 31st Computer Security Foundations Symposium (CSF) 204-217
15. Luu L, Chu D H, Olickel H, Saxena P, Hobor A (2016) Making smart contracts smarter. Proceedings of the 2016 ACM SIGSAC conference on computer and communications security 254-269
16. Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the internet of things. IEEE Access 4 2292-2303
17. Dannen C (2017) Introducing Ethereum and solidity (Vol. 318). Berkeley: Apress