BlockML: A Useful Proof of Work System based on Machine Learning tasks

Middleware 2019 Doctoral Symposium

Andrea Merlina
University of Oslo
andremer@ifi.uio.no

Abstract

The computation required by permissionless blockchains to securely moderate the block proposal rate is wasteful both in terms of energy and computation. We present BlockML, an alternative system that replaces Bitcoin's crypto puzzle with the training of Machine Learning (ML) models, providing a useful side product to the consensus protocol of permissionless blockchains.

CCS Concepts • **Computing methodologies** → **Neural networks**; Cooperation and coordination;

Keywords blockchain, machine learning, useful proof of work

ACM Reference Format:
Andrea Merlina. 2019. BlockML: A Useful Proof of Work System based on Machine Learning tasks: Middleware 2019 Doctoral Symposium. In *Middleware '19: 20th International Middleware Conference Doctoral Symposium (Middleware '19), December 9–13, 2019, Davis, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3366624.3368156

1 Introduction

Permissionless blockchains allow an unauthenticated participant to freely join and leave the system while, at the same time, guaranteeing system security. The security is based on the Proof of Work algorithm, which limits the proposal rate, thus preventing Denial of Service attacks, and allows unauthenticated nodes to propose blocks by proving they have performed the necessary computation. Combining Proof of Work (PoW) with the longest chain selection rule constitutes the Nakamoto consensus [7]. Unfortunately, PoW involves a considerable waste of both energy and computation. Indeed, no meaningful use is made out of the Bitcoin's crypto puzzle solutions besides securing Bitcoin itself. At the time of writing, Bitcoin's crypto puzzle solutions are systematically produced every ten minutes on average by thousands of nodes scattered around the globe while consuming as much energy as Colombia and constituting 0.32% of the world energy consumption [4]. We propose a system that replaces Bitcoin's crypto puzzle with the training of Machine Learning (ML) models, thus securing the blockchain while producing a useful artifact out of the consensus protocol.

Furthermore, our solution promotes competition among the solvers of ML tasks and rewards them based on the accuracy of their models. The system promotes the use of efficient solving methods and incentivizes the development of new and effective ML techniques. In the rest of our submission, we describe our approach and the challenges we are facing.

Middleware '19, December 9–13, 2019, Davis, CA, USA

ACM ISBN 978-1-4503-7039-4/19/12…$15.00
https://doi.org/10.1145/3366624.3368156

2 Related work

The work most similar to ours is Coin.AI [2], in which the authors propose a useful work based on a brute force of the neural architecture search space of a Deep Neural Network (DNN). The search is performed until the DNN achieves the required threshold accuracy. The procedure for generating an architecture setup maps an input to the architecture, where the mapping is presented using context-free grammar, without providing examples of mapping functions. Moreover, establishing a target threshold for ML tasks a priori is a challenging undertaking because the obtainable accuracy is strictly data-dependent. In Proof of Deep Learning [5], the work to secure the blockchain is the supervised training of ML models. This solution makes strong assumptions on the honesty of the data provider. Additionally, validation is not efficient since full nodes need to execute the training process again. Finally, the authors of Proof of Learning [3] propose a system that selects a committee using a variation of Algorand [6], where the probability to be selected is proportional to the disk space occupied by trained machine learning models. Building upon Algorand, this solution inherits its open problem about the random selection of the committee. For instance, the committee should not be allowed to rank their own submitted proposals, situation possibly worsened by collusion attacks.

3 Background

3.1 Machine Learning

We focus on supervised Machine Learning, although the proposed approach can be extended to other kinds of ML tasks. In supervised learning, a data set of input-output pairs is provided to the trainer, whose goal is to infer a mapping function that can accurately predict pairs that have not been used during training. The mapping function is constructed as a sequence of linear (matrix) operations and non-linear transformations, represented by a Neural Network. By construction, the process is divided into two logically distinct phases. In the first phase, the model is trained and in the second phase, it is evaluated according to some metric.

3.2 PoW properties

An alternative to Proof of Work must satisfy several properties to be an effective block proposal moderation algorithm. We present the most relevant and non-trivial properties:

1. **Hardness**. Finding a solution requires actual work to be performed.
2. **Adjustability**. The expected amount of work to find a valid solution is tunable through system parameters.
3. **Efficiency**. The solution has to be efficient to verify.
4. **Sensitivity**. The work must be tied to a specific block and set of transactions. If any value in the block changes, nodes must be able to detect it and invalidate the solution.
5. **Creator free**. Finding a solution for a puzzle does not provide any advantage in solving any other puzzle.

4 Challenges

Replacing Proof of Work with the training of ML tasks presents challenges at different levels. A randomly generated Machine Learning task cannot, by construction, be useful. A useful PoW thus requires the introduction of an entity whose role is to contribute to the generation of a task and which is interested in the task solution. Inserting additional roles in a blockchain system modifies the consolidated incentive mechanism of Bitcoin-style permissionless blockchains. A task-based on ML involves data, possibly of considerable size, which affects communication latency and storage requirements. Additionally, the consolidated two phases process of training followed by evaluation needs to be kept separated also in the proposed system. Having a second phase introduces the risk of the supplier not releasing the data when required to do so, causing the first phase computation to be wasted. Moreover, a system based on alternating training and evaluation phases needs to progress in time windows (epochs). To establish a common notion of time in an asynchronous system is not trivial. Finally, in each epoch every miner must solve a task based on the same input data, to make the model comparison meaningful.

5 System design sketch

The BlockML epoch is defined as the time between Miners start competing to solve a task and the moment a winner is selected. Miners taking part in a competition train their model on the same task template and data (Section 5.2). The winner is selected as the best performing ML model on the test set according to the metric established by the Supplier (see Section 5.1). The model itself constitutes the proof that work has been done by the Miner. All the proposed models are evaluated and ranked by the validators. The block containing the winning model is appended to the blockchain, ending the current epoch and starting the following one. Since training time is limited, a solution deemed insufficiently inaccurate by the Supplier can be resubmitted as a future task. As the chain grows in time, the system populates a library of the best performing models for each epoch.

5.1 Participating entities

BlockML is a permissionless blockchain since participants are not authenticated and can freely and, in principle, even simultaneously adopt every role. Nonetheless, we identify three logically distinct roles: *Clients*, *Miners* and *Suppliers*. Suppliers pay a reward to Miners in exchange for a trained ML model, which is useful for the Supplier. No new coin is minted in the block and Miners earn only because of the reward provided by Suppliers, thus countering malicious entities that might try to win their own proposed task and profit out of it. Figure 1 shows the workflow.

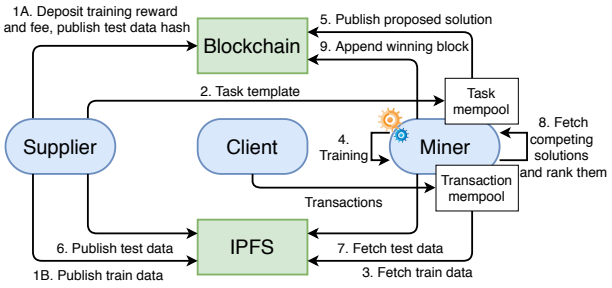

Figure 1. *BlockML workflow.*

The Supplier advertises a new task by depositing a reward for the winning Miner and uploading a secure hash of the test data along with the plain text of the training data on IPFS. Besides, a fee is put in escrow to mitigate the risk of the Supplier not releasing the test data. For each epoch, Miners work on the same data and task template as provided by the Supplier. Miners download the training data and start generating their model as described in Section 5.2. To take part in the competition and be ranked, Miners must submit their solutions before the Supplier of the current epoch releases the test set which is going to be used for the evaluation. Once it is available, Miners download it, along with the other proposed models and can establish the winner.

5.2 Model training

Preliminary to the training, every Miner builds a Merkle tree of transactions, similarly to Bitcoin. However, the tree is modified to contain additional information such as the digest of the training data, the previous block, and optional metadata. The modified Merkle Tree root and the desired complexity are input to a distributed algorithm executed by all the Miners. The algorithm outputs the configuration of the current epoch for the specific Miner that executed it. The configuration and the task template (provided by the Supplier) are combined to form the task definition which constitutes the problem solved by the Miners. Because of different transaction order, the configuration is different for every Miner and, as a consequence, the task definition as well. The Merkle tree root is used as a seed for a pseudo-random function F that encodes selected weights in multiple layers of the Neural Network. Those weights are kept constant during training i.e. we prevent the back-propagation algorithm from modifying them. This is done to tie the transactions to the work being done, as required by the Sensitivity property (Section 3.2). Miners are free to choose whichever Neural Network architecture they see fit for the task at hand, as long as they keep the weight selected by F constant. In each epoch, the new appended block modifies the following configuration and the task definition, preventing possible pre-training of models.

6 Conclusion and open problems

We have presented a Proof of Work system that replaces the wasteful computation with a useful task: the supervised training of Neural Networks. Trained models constitute the proof that work has been done and provide useful artifacts to Suppliers, property that lacks in current permissionless blockchains. The privacy of the data supplied for the task is out of the scope of current work. Suppliers must be aware that the data provided is freely accessible in plain text. Nonetheless, privacy-preserving ML [1] could be an interesting extension to the proposed system.

We identify a few open problems in our work, which we plan to address in the future:

- **Complexity control:** Fine-grained control over the complexity of the task since complexity is strictly related to provided data and the Neural Network architecture.
- **Transition between epochs:** Distributed nodes must be loosely synchronized on the epoch to act appropriately.
- **Complete analysis of incentives:** To design and analyze the incentive mechanism is challenging since it must consider border cases along with hard to quantify properties such as the usefulness of the task.

Acknowledgments

The author wants to thank Roman Vitenberg, Vinay Setty, Kaiwen Zhang and Robbert van Renesse for useful and fruitful discussions.

References

[1] [n. d.]. OpenMined. https://blog.openmined.org/

[2] Alejandro Baldominos and Yago Saez. 2019. Coin.AI: A Proof-of-Useful-Work Scheme for Blockchain-based Distributed Deep Learning. *Entropy* (2019). http://arxiv.org/abs/1903.09800

[3] Felipe Bravo-Marquez, Steve Reeves, and Martin Ugarte. 2019. Proof-of-Learning: a Blockchain Consensus Mechanism based on Machine Learning Competitions. In *IEEE International Conference on Decentralized Applications and Infrastructures*. https://www.researchgate.net/publication/330753314

[4] Cambridge Centre for Alternative Finance. [n. d.]. Cambridge Bitcoin Electricity Consumption Index (CBECI). https://cbeci.org/comparisons/

[5] Changhao Chenli, Boyang Li, Yiyu Shi, and Taeho Jung. 2019. Energy-recycling Blockchain with Proof-of-Deep-Learning. In *arXiv:1902.03912 [cs.CR]*. https://doi.org/10.1109/bloc.2019.8751419

[6] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. *ACM Symposium on Operating Systems Principles* (2017).

[7] Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. (2008). https://bitcoin.org/bitcoin.pdf