Proof of Learning (PoLe): Empowering Machine Learning with Consensus Building on Blockchains (Demo)

Yixiao Lan[1], Yuan Liu[1], Boyang Li [2], Chunyan Miao[2]
[1]Northeastern University, Shenyang, China
[2] Nanyang Technological University, Singapore
liuyuan@swc.neu.edu.cn

Abstract

The consensus algorithm is the core component of a blockchain system, which determines the efficiency, security, and scalability of the blockchain network. The representative consensus algorithm is the proof of work (PoW) proposed in Bitcoin, where the consensus process consumes large amount of compute in solving meaningless Hash puzzel. Meanwhile, the deep learning (DL) has brought unprecedented performance gains at heavy computate cost. In this demo, we channels the otherwise wasted computational power to the practical purpose of training neural network models, through the proposed proof of learning (PoLe) consensus algorithm. In PoLe, the training/testing data are released to the entire blockchain network (BCN) and the consensus nodes train NN models on the data, which serves as the proof of learning. When the consensus on the BCN considers a NN model to be valid, a new block is appended to the blockchain. Through our system, we investigate the potential of enpowering machine learning with consensus building on blockchains.

Introduction

A blockchain network (BCN) is a decentralized distributed system where participants are not necessary to trustworthy but able to collectively maintain a consistent database or ledger (Tapscott and Euchner 2019). The core component of a BCN is the consensus algorithm. The pioneering BCN is Bitcoin and its consensus algorithm is proof of work (PoW) (Nakamoto 2008). The PoW has been applied in various applications scenerios, such as financial services, and alternative cryptocurrency. However, a major drawback of a PoW based BCN is that the system consumes massive amount of compute and energy. Most of the energy is spent on calculating the nonce of hash functions, which serves no real purposes other than being difficult to compute.

On the other hand, machine learning, especially deep-learning, has been widely applied in many fields, such as business and manufacturing (Khan et al. 2020; Lin et al. 2020; Gai et al. 2020). Deep learning uses labeled data to train deep learning models in the way of iteratively updating network weights. A well structured deep learning model can fit any function, which makes deep-learning present strong productivity in solving specific problems. However, the training of deep learning model usually consumes a lot of computing resources. With the extensive application of deep learning, the demand of computing power will be more and more. Meanwhile, significant performance gain in deep learning has been derived from scaling up the network and training data (Mahajan et al. 2018; Radford et al. 2019; Devlin et al. 2019; Yalniz et al. 2019) and automatic design of neural network (NN) architectures (So et al. 2019; Real et al. 2019). These trends have created an ever-growing demand for computational power.

In this demo, we present a system which directs the computation and energy spent on blockchain consensus to the practical function of training machine learning models. We build the system based on our proposed proof of learning (PoLe) consensus algorithm where an asymmetric puzzle is designed to generate tampering prevention blocks (Lan, Liu, and Li 2020). The proposed PoLe-based BCN provides a platform on which users may commission a neural network model that meets their requirements. The BCN contains two types of participants, data nodes which announce tasks with a reward, and consensus nodes or miners which work to solve the announced tasks. After a data node announces a task, consensus nodes may accept it and seek a model that meets the announced minimum training accuracy. After receiving a valid solution which meets the training accuracy criterion, the data node releases the test set. The consensus nodes then collectively select the solution with the highest generalization performance and distribute the task reward accordingly. Thus, a PoLe-based blockchain can serve as a decentralized database and a machine-learning platform simultaneously.

Related Work

Blockchain technology is a distributed paradigm to boost construction of the Internet of value (Tapscott and Euchner 2019), which has gained tremendous momentum in the past decade. Blockchain enables distributed parties who do not fully trust each other to maintain a shared and consistent ledger. Blockchain networks are characterized by unique properties e.g. decentralization, transparency, tampering-resistance and programmability (Dinh et al. 2018).

Consensus algorithm is a method to achieve data consistency among multiple distributed nodes. The most classical and commonly used consensus algorithm is proof of work

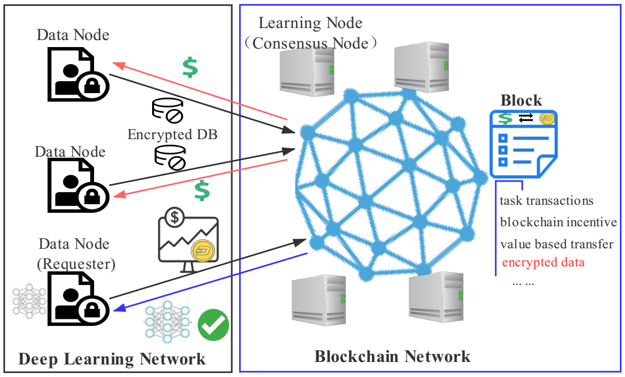


Figure 1: The Overview of the Proposed System

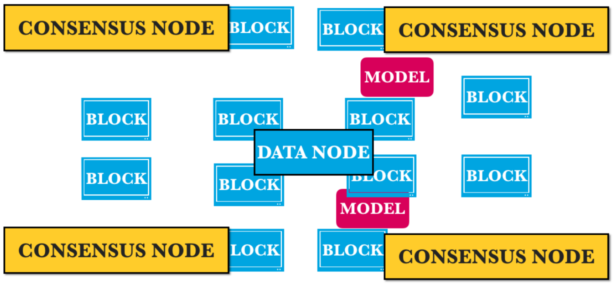


Figure 2: The Communications among Nodes

(PoW) proposed in Bitcoin white paper (Nakamoto 2008). In Bitcoin network, PoW algorithm lets each node repeatedly generate nonces when "mining" a valid block, until the hash value calculated by the nonce and content of the current and previous blocks is less than a target value, which is further used as the hash value of the current block. The blockchain network always considers the longest chain to be effective. When a node wants to change the content of a previous block and makes its modification effective, it needs to recalculate the hash value of all blocks after that block and makes the length of the modified chain longer than the unmodified chain. The computational power required by this process is prohibitively high, which guarantees data security and data consistency of blockchain networks.

Motivated to reduce the compute demand of PoW, proof of stake (PoS) (King and Nadal 2012) and distributed proof of stake (DPoS) (Larimer 2014) have been proposed. Other alternative consensus mechanisms such as credit-based PoW (Huang et al. 2019), proof of reputation (PoR) (Qianwei Zhuang 2019) and proof of negotiation (PoN) (Feng et al. 2020), DBFT (Liu et al. 2020) have been also proposed in the literature. PoR studies a two-chain architecture to construct the reputation of nodes in a separate chain and the next block generator is determined by the reputation chain (Qianwei Zhuang 2019). In PoN, the trustworthiness of miners are evaluated and a random-honest miner is selected based on negotiation rules. The PoN investigates parallel multi-block creation method to achieve high efficiency than traditional consensus mechanisms in one-by-one block creation (Feng et al. 2020). To date, PoW remains the most popular and widely accepted choice (Gervais et al. 2016).

System Architecture

The proposed system is a decentralized peer-to-peer network composed by two types of entities: data nodes and consensus nodes, as shown Figure 1. Communications between nodes are presented in Figure 2. A data node is a user who commissions machine learning tasks via the blockchain. The consensus nodes are the suppliers of the computational power to the system; they compete to train a model that meets the requirements as specified by the data node. The winner consensus node receives the reward specified by the data node. Besides rewards from data nodes, the blockchain also functions as a decentralized data storage such as those used by cryptocurrencies.

The roles of the data provider nodes and the consensus nodes are asymmetric. Data nodes provide the monetary incentives to the consensus nodes and are assumed to be responsible users who will not abuse the mechanism. In comparison, consensus nodes can join and leave the system at any time, and are assumed to be will to cheat when possible.

The test set should remain off the blockchain in order to prevent malicious consensus nodes from using the test set for training. Therefore, the data node only broadcasts the test set after it starts to receive trained models. The data node can decide to wait for a number of solutions to arrive before releasing the test set. After the test set is released, no solutions from consensus node will be accepted.

We design an encryption mechanism to prevent malicious nodes from starting training before other nodes. We follow inner-product functional encryption (Abdalla et al. 2015) and propose a method for the data node to upload data after encryption and the consensus nodes to access the inner product between user data and model parameters. A secure mapping layer (SML) is designed between encrypted data and the target neural network model.

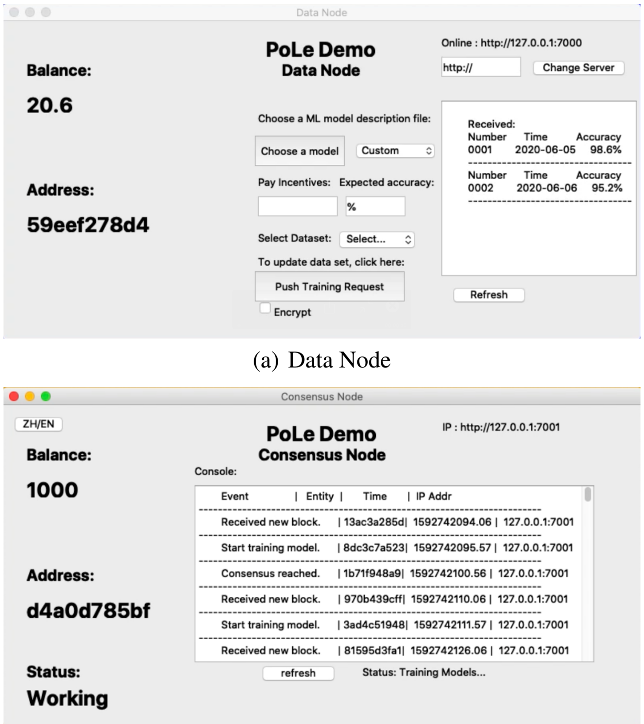
The behavior of the consensus nodes follow the Proof-of-Learning (PoLe) consensus algorithm.

Proof of Learning Consensus

The process of PoLe includes the following 4 steps: model training, new block broadcasting, model verification, new block confirmation,

Step 1: model training. A consensus node selects the task with highest value from the task list in the last block in its blockchain, and begins training the task. The value of a task is determined by the average reward in a unit of time. Since the task list is consistent, the highest value task should be the same for all the miners. Since the generation of SML is related with the current block hash, the malicious nodes are unable to start mining in advance.

Step 2: block broadcasting. When training has produced a model that meets the training accuracy specified by task requesters, the miner broadcasts a new block declaring its success.



Balance: 20.6

Address: 59eef278d4

Online : http://127.0.0.1:7000

http://

Choose a ML model description file:

Pay Incentives: Expected accuracy: %

Select Dataset:

To update data set, click here:

```
Received:
Number     Time        Accuracy
0001       2020-06-05  98.6%
--------------------------------
Number     Time        Accuracy
0002       2020-06-06  95.2%
--------------------------------
```

(a) Data Node

Balance: 1000

Address: d4a0d785bf

Status: Working

IP : http://127.0.0.1:7001

Console:

```
Event               | Entity |      Time       | IP Addr
-----------------------------------------------------------------------
Received new block.  | 13ac3a285d| 1592742094.06 | 127.0.0.1:7001
-----------------------------------------------------------------------
Start training model.  | 8dc3c7a523| 1592742095.57 | 127.0.0.1:7001
-----------------------------------------------------------------------
Consensus reached.  | 1b71f948a9| 1592742100.56 | 127.0.0.1:7001
-----------------------------------------------------------------------
Received new block.  | 970b439cff| 1592742110.06 | 127.0.0.1:7001
-----------------------------------------------------------------------
Start training model.  | 3ad4c51948| 1592742111.57 | 127.0.0.1:7001
-----------------------------------------------------------------------
Received new block.  | 81595d3fa1| 1592742126.06 | 127.0.0.1:7001
```

Status: Training Models...

(b) Consensus Node

Figure 3: The UI for Data Nodes and Consensus Nodes

Step 3: model verification. After consensus nodes receive a series of blocks and end up receiving test data, they first compare the test accuracy of these blocks and sort them in a descending order of the test accuracy. Then, miners evaluate the validity of each block by verifying the test accuracy of the trained model contained in the new block.

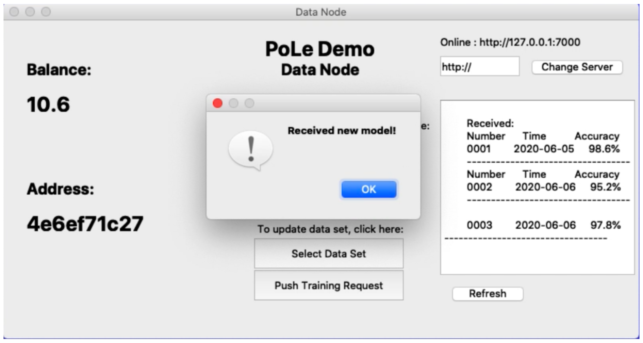
Step 4: block confirmation. A block firstly passes the verification process, then the block become the winning block to be accepted by other nodes.

When a new block is accepted as a winning block, the test data is then appended in the body of the block and the rewards of the task is transferred to the block owner. The task list in the new block is formed by subtracting the completed task from the original list in the previous block and adding the newly collected tasks. Consensus nodes will consider the transactions contained in the winning block to be valid and generate new blocks by attempting the next task from the task list of the winning block. Readers can access a detailed design of PoLe in (Lan, Liu, and Li 2020).

The Demonstration System

We have designed and implemented two end-user clients for data nodes and consensus nodes respectively. In our system, each user is identified with a unique address, which is generated in a similar mechanism as Bitcoin system. The account balance of the user, assigned IP address, and history action log are presented for users, as shown in Figure 3.

In the user interface for a data node in Figure 3(a), the node can specify a customrized model or choose a model from a predefined list. After setting up the model, the data node sets the training incentive payment and expected accuracy. The dataset can also been customized. Then the data node can push the training request to the blockchain consensus network. It should be noted that the data is optimal to be encrypted in the current version.



Balance: 10.6

Address: 4e6ef71c27

Online : http://127.0.0.1:7000

http://

Received new model!

e:

To update data set, click here:

```
Received:
Number     Time        Accuracy
0001       2020-06-05  98.6%
--------------------------------
Number     Time        Accuracy
0002       2020-06-06  95.2%
--------------------------------

0003       2020-06-06  97.8%
--------------------------------
```

Figure 4: Data Node Receiving Trained Model

Consensus nodes, by launching their client, can join the blockchain network at their will. The consensus nodes take the same task from the unfinished training requests in the previous block, and train the model according to the task specification. The real-time status of consensus nodes is also presented in their interface as shown in Figure 3(b).

The data nodes and consensus nodes form a peer-tp-peer network to exchange trained models and blocks, as shown in Figure 2. The new block generation follows the proposed PoLe, where the consensus node with the highest performance accuracy is rewarded with the task incentives through generating an incentive transaction with its generated new block. After the model has been trained well by the consensus nodes, the data node will be informed with the new model as in Figure 4. A full demo video is available in youtube[1] and youku[2].

Discussion

The designed demo system serves like a platform on which the data nodes post machine learning tasks by offering rewards and the consensus nodes compete for them.

The PoLe design encouranges the data node to accurately estimate the time it takes to complete the training and provide proportional reward. A data node may be tempted to intentionally overestimate the time limit in order to make consensus nodes to do more work for the same reward. However, under the current mechanism design, this will lead to low task priority. On the other hand, underestimating training time can lead to high task priority but cause training to terminate prematurely, yielding poor-performing models.

The presented demo system bears some resemblances to federated learning (Yang et al. 2019; Konecný et al. 2016) where various participants with different data collaborate in a secure and privacy-preserving manner to train one model.

[1] https://youtu.be/MOkLpWc2aMk

[2] https://v.youku.com/v_show/id_XNDcyNjU5MDk4NA

However, when compared with conventional settings of federated learning, the proposed system differs in important ways, including (1) collaboration versus competition among the participants, and (2) release verse assume the participants' participation intentions.

Conclusion

PoW-based Blockchain systems can effectively ensure data security at the cost of wasting huge computer resources. Meanwhile, rapid progress in deep learning has created an unsatisfied demand for computation power. We demo a system which channels otherwise wasted compute on blockchain to the practical benefits of training machine learning models. As neural network powered AI applications and blockchain networks continue to grow in the foreseeable future, we believe the PoLe consensus mechanism will contribute to meeting their demands for compute and reducing environmental impact from energy consumption.

Acknowledgments

This research is supported, in part, by the National Natural Science Foundation for Young Scientists of China under Grant No.61702090 and No. 61702084; and 111 Project (B16009); and the Fundamental Research Funds for the Central Universities (N2017006, N182410001); Nanyang Assistant Professorship (NAP).

References

Abdalla, M.; Bourse, F.; De Caro, A.; and Pointcheval, D. 2015. Simple Functional Encryption Schemes for Inner Products. In *IACR International Workshop on Public Key Cryptography*, 733–751.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of Computational Linguistics: Human Language Technologies*, 4171–4186.

Dinh, T. T. A.; Liu, R.; Zhang, M.; Chen, G.; Ooi, B. C.; and Wang, J. 2018. Untangling Blockchain: A Data Processing View of Blockchain Systems. *IEEE Transactions on Knowledge and Data Engineering* 30(7): 1366–1385.

Feng, J.; Zhao, X.; Chen, K.; Zhao, F.; and Zhang, G. 2020. Towards random-honest miners selection and multi-blocks creation: Proof-of-negotiation consensus mechanism in blockchain networks. *Future Generation Computer Systems* 105: 248–258.

Gai, K.; Guo, J.; Zhu, L.; and Yu, S. 2020. Blockchain Meets Cloud Computing: A Survey. *IEEE Commun. Surv. Tutorials* 22(3): 2009–2030.

Gervais, A.; Karame, G. O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; and Capkun, S. 2016. On the Security and Performance of Proof of work Blockchains. In *Proceedings of ACM SIGSAC Conference on CCS*, 3–16.

Huang, J.; Kong, L.; Chen, G.; Wu, M.-Y.; Liu, X.; and Zeng, P. 2019. Towards Secure Industrial IoT: Blockchain System with Credit-based Consensus mechanism. *IEEE Transactions on Industrial Informatics* 15(6): 3680–3689.

Khan, W. A.; Chung, S. H.; Awan, M. U.; and Wen, X. 2020. Machine learning facilitated business intelligence (Part I). *Industrial Management Data Systems* 120(1): 164–195.

King, S.; and Nadal, S. 2012. PPcoin: Peer-to-peer Cryptocurrency with Proof-of-stake. In *Proceedings of ACM SIGSAC Conference on CCS*, 1–27.

Konecný, J.; McMahan, H. B.; Ramage, D.; and Richtárik, P. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *CoRR* abs/1610.02527.

Lan, Y.; Liu, Y.; and Li, B. 2020. Proof of Learning (PoLe): Empowering Machine Learning with Consensus Building on Blockchains. *CoRR* abs/2007.15145.

Larimer, D. 2014. Delegated Proof-of-stake White Paper. http://8btc.com/doc-view-151.html, accessed on 2020 November 19th.

Lin, S.; Du, Y.; Ko, P.; Wu, T.; Ho, P.; Sivakumar, V.; and subbareddy, R. 2020. Fog Computing Based Hybrid Deep Learning Framework in effective inspection system for smart manufacturing. *Comp. Comm.* 160: 636–642.

Liu, Y.; Ai, Z.; Tian, M.; Guo, G.; and Jiang, L. 2020. DSBFT: A Delegation Based Scalable Byzantine False Tolerance Consensus Mechanism. In *Proceedings of 20th International Conference Algorithms and Architectures for Parallel Processing (ICA3PP)*, volume 12454, 426–440.

Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Bharambe, A.; and van der Maaten, L. 2018. Exploring the Limits of Weakly Supervised Pretraining. *CoRR* arXiv:1805.00932.

Nakamoto, S. 2008. Bitcoin: A Peer-to-peer Electronic Cash System. *White Paper* 1–8.

Qianwei Zhuang, Yuan Liu, L. C. Z. A. 2019. Proof of Reputation: A Reputation-based Consensus Protocol for Blockchain Based Systems. In *Proceedings of the International Electronics Communication Conference*, 131–138.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models Are Unsupervised Multitask Learners. *CoRR* arXiv:1912.12860.

Real, E.; Aggarwal, A.; Huang, Y.; and Le, Q. V. 2019. Regularized Evolution Forimage Classifier Architecture Search. In *Proceedings of AAAI*, 4780–4789.

So, D. R.; Liang, C.; ; and Le., Q. V. 2019. The Evolved Transformer. In *Proceedings of the 36th International Conference on Machine Learning*, 5877–5886.

Tapscott, D.; and Euchner, J. 2019. Blockchain and the Internet of Value. *Research Technology Management* 62(1): 12–18.

Yalniz, I. Z.; Jgou, H.; Chen, K.; Paluri, M.; and Mahajan, D. 2019. Billion-scale Semi-supervised Learning for Image Classification. *CoRR* arXiv:1905.00546.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology* 10(2).