Towards artificial neural network and blockchain conjunction: life-long deep learning system

Pavel Krolevets[1]

[1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, China*
pavelkrolevets@sjtu.edu.cn

December 16, 2019

Abstract

Blockchain and deep learning are two promising technologies flourishing in recent years. They come from different domains and barely have practical connections to each other. Moreover, its hard to find applications combining these technologies because of their different goals.

In this paper, we mediate between these two different technology domains, explore feasibility and new opportunities of implementation of artificial neural networks in blockchain state transition system (smart contracts). We propose a novel distributed incremental deep learning approach inspired by the neuroscience research findings in the brain dual-memory system and decreasing plasticity of neurons for short/long-term memory of an artificial neural network on the blockchain. Our approach is an attempt to solve two big problems: trust and long-term memory in the machine learning model. In the end, we define directions for the future research.

1 Introduction

Nowadays, capabilities of blockchain technology achieved different level of performance and security as it was with the first projects appeared along with cryptocurrency protocols. State transition virtual machines (smart contracts) implemented on new blockchain protocols made possible more complex applications than pure financial functions.

Blockchain technology is changing the landscape of financial industry by providing new possibilities of automation, security, and trust. Success of the first cryptocurrency Bitcoin proved that a long term cryptographical dream can materialize and be a mean of decentralized cryptocurrency. A successor, Ethereum platform went further and provided a new type of decentralized computation and storage - blockchain based virtual computational machine which can be used to execute the code in the decentralized manner [1, 2, 4].

The execution environment run using *quasi*-Turing complete language Solidity. The system state is altered given a series of bytecode instructions and a small tuple of environmental data. This is specified through a formal model of a virtual state machine, known as the Ethereum Virtual Machine (EVM). The state transition happens when new blocks are created by miners in the Prof-of-work consensus settings. It takes approximately 7-14 sec to create a new block, change system state, and execute the code [11]

On the other hand, due to Moors Law and dramatically increased computational performance of personal computers, machine learning filed got new possibilities, especially new performance of artificial neural networks [7]. The basic unit on an artificial neural network is a perpetration which approximates the model of a neuron in biology. New computational performance gave a possibility to combine this basic units to layers and create a really deep neural networks. Backpropagation and gradient descent are the main approaches used to optimize (train) a deep neural network [10, 3, 9], i.e. find percerpetrons connection weights which gives the minimum of loss between training samples and new samples.

Finding a minimum of loss function requires a lot of backpropogation cycles, that made this approach difficult for implementation on blockchain state transition and execution environment. However blockchain transactions speeds is increasing dramatically because of sharding and other scalability techniques. Eventually, blockchain interoperability finalizes the Web3 framework where machine learning can have a new opportunities like decentralized autonomous machine learning models (DAML).

Another important problem in machine learning where blockchain can be used is incremental learning and long term memory of artificial neural networks. Lifelong learning represents a long-standing challenge neural network systems. This is due to the tendency of learning models to catastrophically forget existing knowledge when learning from novel observations. Existing model of artificial deep neural networks suffer from catastrophic forgetting because of their high plasticity.

Approaches for life-learning systems can be divided into: methods regularizing to prevent catastrophic forgetting with previously learned tasks while retrain the whole network while, methods that selectively train the network and expand it if necessary to represent new tasks, and methods that model complementary learning systems for memory consolidation, e.g. by using memory replay to consolidate internal representations [?]

In this paper, we aim to explore conjunction of incremental neuro-science based deep learning model and decentralized settings of blockchain to build a trusted incremental artificial neaural network, which long term memory preserved on blockchain. Moreover, we will propose improvements for neural network architecture and optimization, and give recommendation for future research.

The combination of blockchain and deep learning can help to solve aforementioned problems and open new possibilities for creation of truly artificial intelligence existed in distributed immutable environment. In this paper we are making an attempt to creating a building block for future research and encourage another deep learning researchers to explore this area.

Problem Formulation: Here, P-participants collaboratively train supervised deep learning model to classify K_n classes at time t. They competently minimize loss function $J(\theta) = E_{(x,y)\sim p_{data}} L(f(\mathbf{x}, \theta), y)$ where L is the per-example loss function, $f(\mathbf{x}, \theta)$ is the predicted output when the input is $\mathbf{x}$, and p_{data} is the empirical distribution. In the supervised learning case, y is the target output. The loss collaboratively verified during training. Updated current model stored on blockchain has the minimum loss after training.

Participants are able to increment the capacity of the model and learn new classes K_{n+i} at time $t+1$. The model capacity incrementally updated and new-classes successfully learned and collaboratively verified.

At time $t+1$ the model able successfully classify K_{n+i} classes.

Our contributions include:

1. life-learning deep learning model with changing plasticity and increasing capacity, which is supported by neuroscience research
2. use of blockchain for collaboratively increment the model memory and capacity (see Figure **??**)
3. dual memory approach supported by neuroscience to decrease blockchain computational costs

This paper is organized as follows. In Section II, we review related work in blockchain and deep learning domains. In Section III, we describe our novel approach for decentralized incremental deep learning approach In Section IV, we describe experimental setup. In Section V, we present the results and findings. In conclusion, we summarize the meaning of this paper and state challenges for the future.

2 Related work

We observe that most of research trying to combine blockchain and artificial neural networks lies in the area of federation of deep learning model training [5, 6] , privacy/security improvements, or blockchain data analysis, and cryptocurrency price prediction. In [6] authors propose an approach to train collaboratively and store a machine learning model on blockchain. They propose to use non-financial (like reputation) and financial initiatives to train a machine learning model in decentralized settings. As a blockchain state transition system authors chose Ethereum smart contract environment and experimented on supervised learning approach left unsuper-

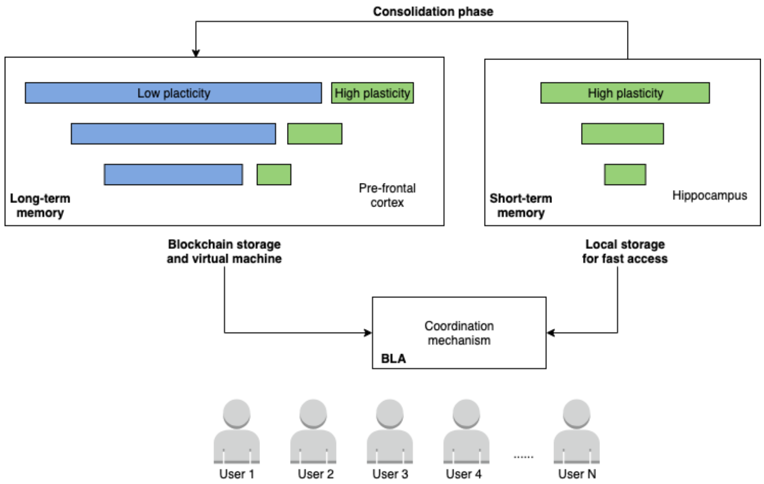

Figure 1: Decentralized life-learning system on blockchain

vised learning models for the future research. The algorithm consists of three steps:

1. Commitment phase: a provider deposits currency as bounty, defines a loss function, and separates training and testing data.
2. Participation phase: participants stake some currency and provide optimization data to update the model
3. Reward phase: each participant receives back staked currency plus bounty based on loss minimization.

In [8] authors using a brain model to create incremental long term memory for artificial neural networks. They base the model on the neuroscience research and propose a generative memory efficient model that does not store previous sample examples, like one of the best methods iCaRL. This is a brain-inspired dual-memory system in which new memories are consolidated from a network for recent memories inspired by the mammalian hippocampal complex to a network for long-term storage inspired by-medial prefrontal cortex. It is based on three DNNs which are mutually interconnected:

- DNN simulating hippocampal complex for recent memory.
- inspired by the medial prefrontal cortex (mPFC) for long-term storage.
- basolateral amygdala (BLA) that determines whether to use HC or mPFC

In [?] authors reviewed recent literature using EEG and fMRI studies on human brain activity to understand the memory retention and recall processes as well as the various brain regions associated with these processes. They found that most of the studies confirm that several regions of the brain, particularly amyglada, cerebral cortex and hippocampus are firmly connected with memory processes like retention and recall.

Based on the research the authors postulate that the correct memory information retrieval, working and episodic memory are significantly influenced by the cerebral cortex in the temporal, frontal and parietal lobes. Hippocampus is responsible processes of new memory creation and encoding, and for transferring of data from short-term to long-term memory. Moreover, long-term potentiation plays a vital role in new memory creation, and learning. Potency of the is based on the synaptic communication network power. Recall and retention of new data into the memory is related to strengthen synaptic connections.

3 Decentralized incremental model for life-long learning

We aim to achieve following features using a combination of incremental learning model and blockchain decentralized storage:

1. Trusted and transparent machine learning model because its loss was verified during decentralized training
2. Continuousness incremental learning mechanism through mutations of the model on blockchain storage
3. Modular architecture to allow mutations of the model on the blockchain with addition of new smart contracts

Blockchain decentralization can give an ability to collectively train DNN model and preserve its in immutable state for the future use. Moreover, it can give benefits in achieving the state of the art performance of the DNN in public consensus settings and preserving it in theoretically unlimited temper resistant storage.

In mammalian brain memory system long-term memories are preserved by decreased plasticity of syntactical connections. In our approach we will preserve wights on the blockchain after training cycles and make them available for other participants. New neurons can be added at later stages and trained for new classes. Basically we decrease the neurons plasticity of the model and store it on the blockchain. Neural network expansion of-chain creates new high plasticity connections that can be added to the existing model after training (see Figure 2 on page 6). After new memory is trained it can be appended on-chain effectively increasing learned classes and overall capacity.

This model similar to the approach of *progressive neural networks* introduced by Rusu at. al. [?] where a pool of pre-trained models exists for each learned task (τ_n). When a new task τ_{n+1} is learned, a new neural network is created.

Another interesting feature of mammalian brain is that it consists of dual memory system for short-term (hippocampus, high plasticity) and long-term (pre-frontal cortex, low plasticity), which are coordinated by a third special part of the brain (basolateral amygdala). We propose a deep learning system based on this principle in combination with blockchain (see Figure 2 on page 6).

In our approach we propose to keep neural network model with already learned tasks on blockchain.

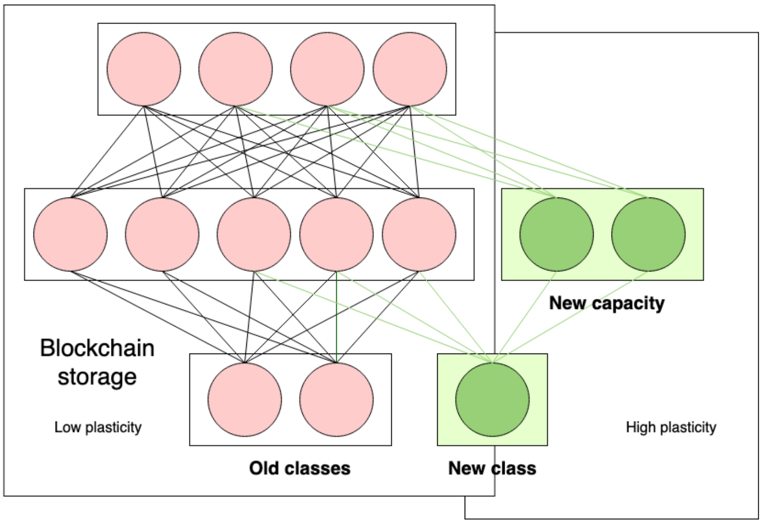

Figure 2: Neural network expansion

3.0.1 Incentive mechanism

Neural network stored on blockchain provides benefits like easier implemented initiative mechanism and trust. We propose to incentivise neural network users to increment the network by learned new classes off-chain. It allows for the DNN to grow slowly new information, as it mimics processes in the pre-frontal cortex where the long-term memories are stored. Participant and users of the neural network model will benefit from its availability. They are able to increase an accuracy of the whole network, or incrementally learn new tasks by appending existed pre-trained model. To keep the trust in the model it should be easily verified. Adversarial participants should be punished for the misbehavior.

Because a publicly trusted neural network model can be beneficial for many participants in decentralized settings, we propose non-financial incentive model based on reputation. For contributing to the model training and incremental learning a participant can receive publicly available reputation for decreasing the loss or teaching the model new classes.

We propose the following algorithm for initial collaborative training with non-financial initiative. Its similar to [6] with modifications suitable for incremental learning settings:

1. **Initial deployment phase**
   (a) Initiator specifies the initial parameters of the model, loss function $L(h, D)$, and threshold for the loss, bellow which the model can accept new classes
   (b) Initiator secretly divides a *test* dataset to Dn parts
   (c) Initiator uploads hashes of these parts to a smart
   (d) Smart contract randomly chooses Dk hashes
   (e) Initiator uploads Dk datasets to the smart contract

(f) If hashes doesn't match data, abort

2. **Training phase**

   (a) Participants query model parameters and initial model

   (b) Participants query test datasets Dk

   (c) Participant query the latest loss value and weights from the smart contract

   (d) Participant verify the loss $L(h, D)_t$ based on downloaded weights W_t, and k test datasets: if doesn't match *abort* and commit the vote on weights W_t

   (e) Participant train the model

   (f) If $L(h, D)_{t+1} < L(h, D)_t | Dk$, commit $L(h, D)_{t+1}$; SHA256(W_{t+1}); updated weights W_{t+1}

   (g) Every training epoch the lowest loss is chosen $\min(L(h, D)_{t+1})$ and SHA256(W_{t+1}) verified

3. **Reward phase**

   (a) For each participant assign a badge for participation to a reward smart contract

   (b) For the minimal committed loss $L(h, D)_{t+1}$ the participant is assigned special badge

The algorithm for incremental learning new classes is provided in Appendix A. Its main difference to the above algorithm is that initiator increment classes and neurons in the model. Other participant retrain the model with fixed weights from the previous model W_t to a new increased capacity weights W_{t+1}. This is effectively mimics the processes of neurogenesis in the brain when the plasticity of long-term memory decreases and new neurons forms new circuits in hippocampus [**?**].

4 Experimental setup

To evaluate the performance and cost of incremental deep learning on blockchain we conducted evaluation on a simple supervised feed forward model with softmax loss on MNIST dataset. We performed tests on multi-participant setup. Adversarial attacks on the model we left for the future research as our goal was to find the existing limits of blockchain and machine learning technologies for life-learning imitation. The proof-of-concept code can be found on the authors github repository

5 Experimental results

6 Conclusion

References

[1] Andreas M. Antonopoulos. *Mastering Bitcoin: Programming the Open Blockchain*. O'Reilly Media, 2017.

[2] Andreas M. Antonopoulos and Gavin Wood Ph. D. *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media, 2018.

[3] Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu.

Advances in optimizing recurrent networks. *CoRR*, abs/1212.0901, 2012.

[4] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform, 2014. Accessed: 2016-08-22.

[5] Xuhui Chen, Jinlong Ji, Changqing Luo, Weixian Liao, and Pan Li. When machine learning meets blockchain: A decentralized, privacy-preserving and secure design. pages 1178–1187, 12 2018.

[6] Justin D. Harris and Bo Waggoner. Decentralized & collaborative AI on blockchain. *CoRR*, abs/1907.07247, 2019.

[7] G.E. Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434, 2007.

[8] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *CoRR*, abs/1711.10563, 2017.

[9] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[10] D.E. Rumelhart, G.E. Hintont, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[11] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

Appendices

A Algorithm for incremental collaborative training

1. **Initial incremental phase**
   (a) Initiator checks if the model allows for incremental learning: $if false, abort$
   (b) Initiator secretly divides a *test* dataset to Dn parts
   (c) Initiator uploads hashes of these parts to a smart contract, which are appended to previous state
   (d) Smart contract randomly chooses Dk hashes
   (e) Initiator uploads Dk datasets to the smart contract
   (f) If hashes doesn't match data, abort
2. **Training phase**
   (a) Participant change model parameters P_t by (only) increment classes and/or neurons in the model weights W_t
   (b) Participant query the latest loss value and weights from the smart contract
   (c) Participant verify the loss $L(h, D)_t$ based on downloaded weights W_t, and k test datasets: if doesn't match *abort* and commit the vote on weights W_t
   (d) Participant train the model with incremented neurons, classes, and old fixed weights W_t
   (e) If $L(h, D)_{t+1} < L(h, D)_t | Dk$, commit $L(h, D)_{t+1}$; SHA256(W_{t+1}); updated weights W_{t+1}
   (f) Every training epoch the lowest loss is chosen $\min(L(h, D)_{t+1})$ and SHA256(W_{t+1}) verified
3. **Reward phase**
   (a) For each participant assign a badge for participation to a reward smart contract
   (b) For the minimal committed loss $L(h, D)_{t+1}$ the participant is assigned special badge