

Morphware: A Decentralized Machine Learning System

Kenso Trabing
kenso@morphware.org

Abstract

Hardware accelerators help data scientists accelerate the development machine learning models, but the prohibitively high cost of these hardware accelerators is a barrier to entry for many data scientists. This work introduces a system that rewards owners of accelerators by auctioning off their idle computing power and subsequently facilitates the associated sub-routines, which train and test the machine learning models in a decentralized capacity, on behalf of the data scientists.

1 Introduction

The computational resources required to run state-of-the-art machine learning workloads is doubling approximately every three-and-a-half months.¹

The solution proposed, to address this scarcity, is a peer-to-peer network where practicing data scientists, machine learning engineers, and computer science students can pay video game players to train models on their behalf, in Morphware tokens; a new cryptocurrency.

1.1 Background Information

Machine learning models can be categorized as either supervised, semi- or unsupervised learning algorithms. The training of a supervised learning algorithm can be described as a search for the optimal combination of weights, to apply to a set of inputs; to predict a desirable output: and the computational complexity required to solve these types of problems is the impetus of this work.

The same hardware (i.e., graphics processing units) that is used to render video games can also accelerate the training of supervised learning algorithms like deep neural networks.

1.1.1 Machine Learning

Machine learning models can vary by a degree of supervision and parameterization.²

The purpose of training a supervised-parameterized model, like an artificial neural network, is to lower the error rate that spans the numerical distance between a prediction and an observation.

Training a machine learning model is preceded by pre-processing, and followed by testing.

Data scientists logically separate the data that is made available to machine learning models while they are training from the data that is made available to them while they are testing, so the model does not overfit the set of available data and perform comparatively worse on unseen data.

Training and testing data are usually selected from the same file or directory in pre-processing.

1.1.1.1 Deep Learning

Deep learning models are a subset of machine learning models and they are especially computationally intensive to train because of their interconnected layers of latent variables.

1.1.2 File Sharing

The machine learning model does not need access to the testing set of data when it is training.

1.1.2.1 Secret Sharing

Public-key encryption makes it possible to share a file only with an intended recipient.³

Asymmetric cryptography also enables blockchain technology like smart contract execution.

1.1.3 Sealed-bid Auctions

Information recorded to distributed ledgers is necessarily public, but a Boolean argument can be parameterized in a smart contract by a bidder, to designate whether or not a bid is authentic.

1.1.3.1 Second-price Auctions

Second-price, sealed-bid auctions guarantee bidders, with authentic bids, non-negative utility.⁴

2 Implementation

The development process of a machine learning model starts with exploratory data analysis, and is followed by an iterative cycle that vacillates between model selection and feature engineering.

The purpose of this work is to enable end users to iterate faster through this development process by creating access to a decentralized network of computers that can accelerate their workloads.

End users are paired with, and pay, worker nodes via a sealed-bid, second-price reverse auction. They pay worker nodes, to train their models, and validator nodes, to test the models trained by worker nodes; as well as the work of each other: with Morphware Tokens (an ERC-20 token).

2.1 The Network

The roles and responsibilities of members of the network include two autonomous peer-types.

2.1.1 Human Input Required

End Users

End users submit models to be trained by the workers and tested by the validators.

2.1.2 No Human Input Required

Worker Nodes

Workers are the nodes that earn tokens by training models submitted by the end users.

Validator Nodes

Validators are the nodes that earn tokens by testing models trained by the workers.

2.2 The Protocol

The end user submits the model, to be trained by the workers and tested by the validators, through the client; which communicates with the network through its back-end daemon.

Peer-assisted delivery enables the propagation of an algorithm and its corresponding dataset from an end user to a worker, or validator, but the initial solicitation of work by the end user is posted to the smart contract; along with relevant responses to the end user from workers or validators.

The initial solicitation of work by the end user includes the estimated runtime of the training period, along with magnet links to the algorithm; the training set; and the testing set of data.

A response to a solicitation of work from a worker includes a magnet link to the model that they trained, which is subsequently tested by a number of validators. The worker and validators are then rewarded tokens, if the model that was trained meets the required performance threshold.

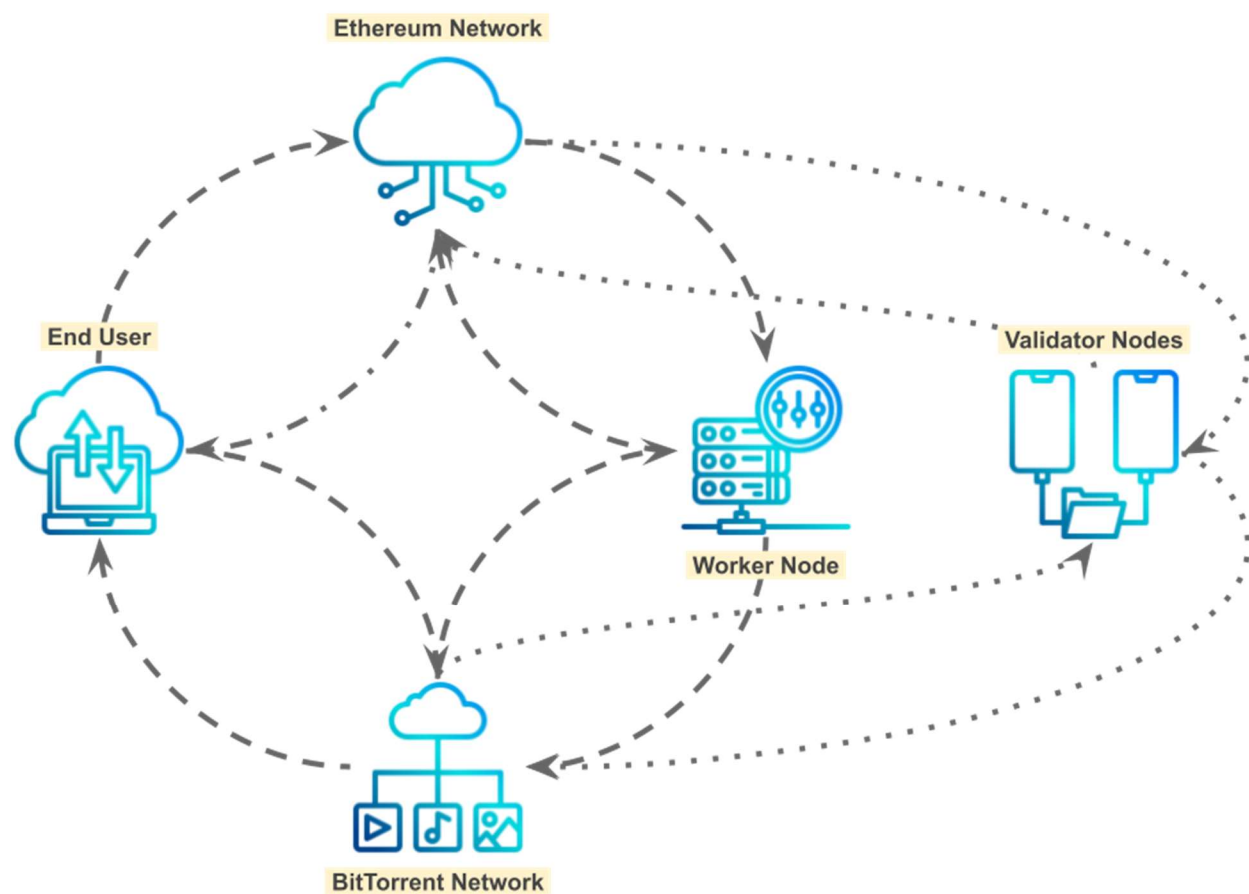


Figure 1 *System Architecture*

2.3 The Client

The client is the graphical user interface that end users and operators of workers or validators interact with. End users submit their jobs through it, in the form of Python files or Jupyter notebooks, and can track the progress of them, via a dashboard that plots their loss functions.

2.4 The Daemon

The daemon is responsible for seeding algorithms and their respective datasets, submitted by the end user through the client, and sending the initial solicitation of work to the smart contract.

It is also responsible for the training and testing of the models, by the workers and validators.

3 Discussion

The system introduced in this work could incentivize the design and deployment of circuits (e.g., weight-stationary systolic arrays), on reprogrammable fabrics, for machine learning algorithms that are either too new or have insufficient market demand for a chip foundry to manufacture.

3.1 Privacy Concerns

The end user has to option to encrypt each of their algorithms and datasets, with the public key of the worker node that wins the auction and; separately: the public keys of the validator nodes.

3.2 Security Considerations

The secure execution of arbitrary code on a remote computer is outside the scope of this work.

4 Conclusion

This work introduces a peer-to-peer network where end users can pay video game players to train machine learning models, on their behalf, in a new cryptocurrency: Morphware tokens.

5 Acknowledgements

Special thanks for Kent Trabing, Rebecca Sealfon, Darshan Raju, Vincent Chiodo, David Rentschler, and William Mantly, Jr. for providing their feedback.

6 Citations

1. Patterson, David. “A New Golden Age for Computer Architecture.” Artificial Intelligence Conference, September 7, 2018, Hilton Union Square, San Francisco, CA. Keynote Address.
2. Géron, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. 2nd ed., O’Reilly, 2019.
3. Schneier, Bruce, et al. *Cryptography Engineering*. Wiley, 2010.
4. Roughgarden, Tim. “Lecture #2: Mechanism Design Basics”, (CS364A: Algorithmic Game Theory, September 25, 2013).