Work in progress.

Autonet

Economy-as-a-Service for Deep Learning Applications

Andrei Taranu, Leonhard Horstmeyer

September 2021

1 Introduction

Every interaction between economic agents can be determined by the reasoning behind it. Some manner of intellectual performance is explicitly specified as premise for most business agreements and implicit in all acts of payment. Our monetary system is built on intelligence. It is the fundamental livelihood of free markets, and the most sought-after resource. While human intelligence is still the most qualitative, its subjective nature and temporal inconsistency are historical causes of financial instability. Digital and quantum technology provides another version of intelligence which, while less refined, is highly quantifiable and growing exponentially. The number of human functions being automated is steadily increasing, with deep learning applications able to tackle ever more abstract tasks. It therefore makes sense that an exchange token intrinsically tied to machine intelligence will benefit from added stability and acceptance.

2 What is Autonet

Autonet is:

1) an economy for AI services.

2) a technical and logical framework for crowd-sourcing the development of large-scale AI models.

3) a system of incentives that allows trustless collaboration between project stakeholders on both sides of the commercial relationship.

Machine Learning applications are different from traditional software in that they require *training*. The development process can be divided into the following steps: a) The developer inputs the logic for a certain task [1]. b) The model is exposed to examples in order for it to create its own set of rules by which it performs the task. The accuracy of the model is proportional to the amount of examples analyzed during the training phase, meaning that larger data sets will generally result in more efficient applications.

[1]eg: recognize if an image has dogs in it

The physical architecture used to train ML models is the same as the one used in validating cryptocurrency transactions. Over the last decade, blockchain mining has

produced the largest parallel-processing network, dwarfing all centralized data farms. The solution is to incentivise miners to switch from Bitcoin (and altcoins) to a token which accounts for the training of ML models. Autonet is a market powered by such a token. There are 4 types of stakeholders: Developers, Investors, Miners and Customers. The economic interactions between these are all regulated through Smart Contracts, and value is accounted for using the dedicated ATN token (c.f. Section ??).
End-to-end cycle of operation:
A developer writes a model and publishes it to the blockchain, along with a description of what that model does. Investors can decide to back that model and pay for its training. Miners are awarded tokens (subdivisions) for the cycles they dedicate towards training that model Once the model is mature, it is made available for commercial use. The revenue it generates is allocated back to the investors and the developers according to their respective shares.

3 Why Autonet?

to allow anyone and everyone to prosper from an otherwise centralized business vertical
to allow for a smooth and peaceful automation of responsibilities for a number of professional categories within the next decade. In the absence of an established framework that makes use of distributed equilibrium to reallocate the earnings of AI, over half of the current workforce runs the risk of becoming economically irrelevant. There are two potential and historically plausible sets of enabling factors:

1. The providers of AI services are incentivised to consolidate in order to maximize the performance of their models and thus control the entire offer side of the market, while being still more affordable for individual economic agents than actual human labor, forcing a rapid accentuation of wealth disparity.

2. The authorities in charge of dispensing universal income to the professional categories affected by automation, being themselves centralized entities, are subject to both intentional and accidental faults that are beyond the reasonable acceptance standards of dignified people. That means people are unlikely to feel comfortable with a situation in which their livelihood is dependent on a handout by central authorities.

Besides the above socio-economic arguments, the proposed project also aims to mitigate the risks associated with centralized research and development of leading-edge AI models. As distinct from traditional software, logic of which is developed entirely by programmers, machine learning models create their own operating algorithms. The largest models of this sort are comprised of billions of parameters that are automatically assigned through exposure to individual examples of paired inputs and outputs pertaining to their targeted utility. As a result, it is all but impossible for any team/organization of IT professionals to debug the governing logic of such cybernetic agents. The need for implementing global safety protocols in AI research has been repeatedly pointed out by subject-matter thought leaders, in formal addresses such as Open Letter on Artificial Intelligence. [?]

The open nature of Autonet's distributed marketplace allows the public to scrutinize individual subroutines of AI agents and effectively employs collective intelligence towards ensuring predictable model behavior. In addition to crowd-sourcing the audit and fine-tuning of their logic, Autonet agents are also provisioned for safety by virtue of having their actions confined to specific mandates enforced through Smart Contracts. In this sense, Autonet is extending the core value proposition of Blockchain technology to ensure trust in autonomous entities.

We are coining the term Economy-as-a-Service, to describe Autonet's operating model. In-keeping with the "as-a-service" principle, customers are utilizing the product in a self-serve style. However, unlike conventional digital products, the entire range of stakeholder interactions, complete with contractual arrangements is fully automated. This means once the platform is deployed on a particular Turing-complete Blockchain (such as Ethereum), it does not require maintenance or administrative support. Thus, its logical architecture does not require any type of privileged access to the platform and any value that is being collected from services rendered will get allocated on-chain according to the hard-coded Smart-Contract logic. This logic only allows capital to exit the platform under two circumstances:

1. As dividends paid to Model Shareholders. The shareholders are the developers of the un trained models along with the investors that funded its training. Dividends are paid according to their contribution.
2. As DAO grants. Integration initiatives are awarded regular funds based on the votes they raise from token holders.

4 Economy

Tokenomics and balance of incentives is coded in this repository using Smart Contracts: https://github.com/autonet-code/contracts

Following is a description of the stakeholder types and their relationships, covering deployment, production and cash flow. These are developer, investor, miner and customer.

User journeys:

Developers create the logic for the untrained model, consisting of the actual structure of the neural network. That means defining the number of layers, the loss and activation functions, and the node count for each layer. They also specify a data set source or data set acquisition strategy that relies on scraping the data from public sources such as Wikipedia. To ensure compatibility, they can just clone and customize the provided template project. After that, it gets listed on the market where it can be seen by potential investors.

Investors acquire shares in a project by paying for its training. Each project has a specific training goal that derives from the amount of compute it needs to consume before it can become commercially viable. If that's not reached before a set deadline, then the investors get their money back. But if it is, then the project is picked up by the miners.

Miners: One unfair advantage over other federated training systems is that mining can be done directly in the cross-platform client. Miners are nodes of TensorFlow clusters. They dynamically synchronize their jobs over TCP using the MapReduce logic present in Apache Spark. There is a hierarchical validation mechanism that we call proof of intelligence, by which the weights and biases that every node submits have to perform above a certain threshold on a validation set.

The final validation is performed distributedly in the project contract on a test data set pulled from IPFS. After that, the project is labeled as mature and goes on to serve customers.

Customers can try out the mature agents directly in the client and then they can hire them by acquiring a number of API calls that need to be exhausted before a specific block number is reached.

The investors already paid the miners, so when the mature agent starts earning it's time for the developers and the investors to get paid with the generated revenue.

5 Node Environment

The Node Runtime is designed to be platform-agnostic and capable of operating natively on both desktop and mobile devices. This flexibility allows for a more extensive and diverse network of nodes to participate in the Autonet ecosystem, thereby increasing the overall computational capacity and reliability of the system. To achieve this, the node runtime is built using a combination of platform-independent technologies and adaptive resource management strategies, ensuring optimal performance on various devices with different hardware configurations.

5.1 Kademlia

One of the key components of the Autonet node runtime is its utilization of the Kademlia distributed hash table (DHT) for establishing consensus between nodes. Kademlia is a decentralized, peer-to-peer protocol that allows nodes to locate and share data in an efficient and scalable manner. By leveraging Kademlia, Autonet nodes can quickly and reliably synchronize their states, coordinate their efforts in training and validating models, and maintain an up-to-date view of the overall network.

The integration of Kademlia within the Autonet node runtime works as follows:

Node Initialization: When a new Autonet node is launched, it generates a unique node ID and joins the Kademlia network. The node ID serves as the node's address within the DHT and is used for routing and data storage purposes.

Node Discovery: Upon joining the network, the new node begins the process of discovering other nodes in the Autonet ecosystem. This is achieved by sending lookup requests to its nearest neighbors, as determined by the XOR metric used by Kademlia. As the node receives responses, it updates its routing table with the discovered nodes and continues the discovery process iteratively until it has established a comprehensive view of the network.

Data Storage and Retrieval: Autonet nodes use the Kademlia DHT to store and retrieve data related to the training and validation of machine learning models. This includes model weights and biases, training datasets, validation metrics, and proof submissions. When a node needs to access a specific piece of data, it sends a request to the Kademlia network, which routes the request to the node responsible for storing that data. The data is then retrieved and returned to the requesting node, ensuring efficient and decentralized access to information.

Consensus Formation: As nodes in the Autonet ecosystem work on training and validating models, they periodically exchange updates with their peers using the Kademlia protocol. This allows nodes to maintain a consistent view of the network state and ensure that their efforts are aligned with the rest of the network. When a node submits a proof of intelligence, it propagates the proof through the Kademlia network, allowing other nodes to independently verify the submission and reach consensus on its validity.

Scalability: Kademlia's decentralized architecture provides inherent fault tolerance and scalability for the Autonet node runtime. As nodes join or leave the network, the Kademlia DHT automatically adapts to maintain its efficiency and reliability. This ensures that the Autonet ecosystem remains robust and capable of handling fluctuations in node participation and computational resources.

5.2 Apache Spark

By leveraging Spark's capabilities, Autonet can effectively distribute the tasks associated with training and validating machine learning models among the participating nodes, while also maintaining a high level of fault tolerance and resource optimization. Here's how Apache Spark can be integrated into the Autonet node runtime in a decentralized self-scrutinizing system:

Spark Cluster Setup: Each Autonet node participating in the network sets up a local Apache Spark instance, which can operate as either a standalone worker or part of a larger Spark cluster. Nodes can dynamically join or leave the Spark cluster based on their availability and resource constraints, ensuring a fluid and adaptable network of computational resources.

Task Partitioning: As new machine learning models are added to the Autonet ecosystem, the training and validation tasks associated with these models are partitioned into smaller, manageable chunks. These tasks can be represented as Spark jobs or tasks, which are then distributed across the Spark cluster, allowing multiple nodes to work on different parts of the model concurrently.

Data Storage and Processing: Autonet nodes use the Kademlia DHT for storing and retrieving model-related data, such as training datasets, model weights, and biases. Apache Spark can be configured to access this distributed data storage layer, enabling nodes to efficiently read and write data as needed during the model training and validation process.

Decentralized Model Training and Validation: With Spark, Autonet nodes can collaboratively train and validate machine learning models in a decentralized manner. Each node processes its assigned tasks, updating the model weights and biases based on the local data it has access to. Periodically, nodes share their updates with their peers using the Kademlia protocol, allowing the network to reach consensus on the state of the model and coordinate further training efforts.

Fault Tolerance and Resource Management: Apache Spark's built-in fault tolerance mechanisms and dynamic resource allocation features can be utilized to maintain the stability and efficiency of the Autonet node runtime. If a node fails or becomes unavailable during the model training process, Spark can automatically redistribute the affected tasks to other available nodes, ensuring minimal disruption to the overall progress. Additionally, Spark can adaptively manage the computational resources of each node, optimizing the workload distribution based on the current network conditions and node capacities.

By integrating Apache Spark into the Autonet node runtime, the system can effectively distribute the complex tasks associated with training and validating machine learning models across a decentralized network of nodes. This not only enhances the system's overall efficiency and scalability but also ensures a transparent and self-scrutinizing environment for the development and deployment of AI services.

5.3 Proof of Intelligence Operating Model:

The Proof of Intelligence (PoI) operating model is designed to ensure that the machine learning models developed within Autonet are efficiently trained and validated. PoI builds upon the concept of Proof of Work (PoW), where miners provide computational resources to validate transactions and maintain the blockchain. In PoI, miners dedicate their resources to train and validate machine learning models instead of solving complex mathematical problems.

The PoI model operates as follows:

Model Selection: Miners choose a model from the Autonet marketplace that has reached its funding goal and requires training.

Training Phase: Miners allocate their computational resources to train the selected model using the specified dataset. The training process includes updating the model's weights and biases based on the training examples.

Validation Phase: Once the model has been trained, it must be validated to ensure its performance meets a predetermined threshold. This process involves testing the model's performance on a validation dataset that is distinct from the training dataset. The performance is measured using a predefined metric, such as accuracy, F1-score, or Mean Squared Error (MSE).

Proof Submission: If the model's performance on the validation dataset surpasses the threshold, the miner submits the updated weights and biases as proof of their contribution to the model's training.

Proof Verification: Other miners in the network verify the submitted proof by testing the updated model on a separate test dataset. If the majority of the network agrees that the proof is valid, the miner is rewarded with ATN tokens for their contribution.

Model Deployment: Once the model has been validated and approved by the network, it is marked as "mature" and becomes available for commercial use.

6 Service Ontology

Individual agents can become building blocks for a more generalized intelligence by allowing models to serve as single nodes in other agents created within the platform. This process facilitates the creation of higher-level, more complex AI services by combining the capabilities of multiple, specialized models.

To achieve this, each model is assigned a unique weight and bias that represents its overall contribution to the composite agent. These values are dynamically updated based on the performance of the individual models during training and validation. This approach enables the development of modular AI systems that can be easily extended, adapted, or improved over time.

Service ontology in Autonet enables the following advantages:

Scalability: The modular approach allows the development of larger, more complex AI services by combining the strengths of multiple specialized models.

Adaptability: As new models and services are developed within the Autonet ecosystem, existing agents can easily incorporate these advancements to improve their overall capabilities.

Efficiency: By reusing and combining existing models, developers can save time and resources that would have been spent on developing similar functionality from scratch.

7 Outlook & Implementation.

Social-economic impact:

Autonet has the potential to democratize access to AI services and create a more inclusive and decentralized AI ecosystem. By enabling stakeholders to collaborate in the development, funding, training, and commercialization of AI models, Autonet can help prevent the concentration of AI capabilities in the hands of a few dominant entities. This decentralization can foster innovation, promote fair competition, and ensure a more equitable distribution of the benefits derived from AI advancements.

Recommendations:

Encourage participation: Incentivize developers, investors, miners, and customers to actively engage in the Autonet ecosystem to promote a diverse and thriving marketplace of AI services.

Develop educational resources: Provide resources and support to help stakeholders understand the Autonet platform, its underlying technology, and the potential benefits of participating in the ecosystem.

Foster collaboration: Encourage collaboration between stakeholders and across industries to drive innovation and the development of novel AI applications.

Monitor and address ethical concerns