Open Decentralized AI (ODA)

Introduction

Currently, the AI landscape is dominated by a few early players who control vast amounts of data and computing resources. This centralization not only stifles innovation but also places immense power in the hands of a select few. By decentralizing AI, we can democratize access to AI technologies, ensuring a more equitable distribution of benefits and fostering a vibrant community of contributors. Inspired by the transformative impact of open-source protocols like PGP/GPG, Bitcoin, Linux, and BitTorrent, I propose a new vision for AI development that's open distributed without a central party for: Storage, Inference, or Training.

Following are two proposed ways to do this (1) *ODA as an Extension of the Bitcoin Protocol* and (2) *ODA as a Layer 3 on the Lightning Network.*

ODA as an Extension of the Bitcoin Protocol

Overview

The Open Decentralized AI (ODA) model aims to leverage the Bitcoin protocol's decentralized, peer-to-peer architecture to create a similarly structured AI development network. This network would facilitate the distributed storage, inference, and training of AI models, ensuring no single entity controls the AI development landscape.

Key Components

Network:

The network consists of full nodes and lite-nodes: *Nodes (Miners)* - These nodes run the ODA protocol, create new blocks, verify transactions, and disperse model data and weights across the network; *Lite-Nodes (Users)* - These nodes primarily receive and disperse information but do not participate in block creation.

Storage and Data Handling:

Data, including models and weights, can be federated across the network but we need to ensure redundancy and accessibility even if nodes leave the network. The implementation would use the established distributed storage system methodology/literature with mechanisms like proof-of-storage or proof-of-space-time to ensure the data integrity of the network without central control. (I would argue distributed storage is a somewhat well understood topic - although uncertainty of trust might be somewhat novel). There are a few projects that attempt to provide decentralized storage: FileCoin, IONet, IPFS, etc.

Inference and Validation:

Inference involves passing inputs to the AI model and receiving outputs, an easy way to think about this is as data added to the Transaction, that will be run on a model to create an output that is validated across the network (similar to the transaction validation that already happens when validating a block in the Bitcoin network protocol). The validation of inferences would be integrated into the Bitcoin consensus protocol to maintain trust and security, but some immediate problems come to mind: Either large minimum requirements for all validators or small validators would be forced to leave out big transactions. Proof-of-Work depends: (1) validating if all transactions in a block are correct and (2) generating a nonce that satisfies the difficulty constraint on the block, because validating transactions are quick and usually take the same amount of time the miner is incentivized to grab any transactions that have a high fee and create a block with them. In our ODA Network some transactions might take longer to verify than others so there is another incentive to grab transactions that are quick to verify, and this might lead to problems with *Selfish Mining.* Here are a few solutions, but these haven't been fully thought through: (1) there is no problem, actually fee mechanism will handle it, (2) difficulty adjustment that changes based on how "difficult" validating the inference in the transaction probably was, (3) hope future things that can make validation much easier like zkML and opML, which allows the verification (the output is given by running the input on the function) to be less costly and performed by only the first the inferencer - this is done by generating a proof during inference time that could be validated later by the other validator nodes

Training and Model Updates:

We define training as: adjusting a model's weights to improve performance on the objective over the data. This can also be treated as a transaction within the blockchain that can be validated by running the model over the data and measuring the objective performance yourself. Changes that significantly alter the model's architecture, objective, or data would constitute a new model, this is to maintain the integrity of the validation process - all the validator is doing is using the public information about the model and the new weights to see if it does better on the objective than the old weights (Note we don't need to know how the training was done or ensure that the trainer actually did a certain type of compute, we just need to make sure the new weights are objectively better than the old on the predefined dataset). When explaining this to a friend a few questions were asked:

Q: Who has access to modify the model? A: All models and weights are shared in a peer-to-peer way. If you improve the model, you have the ability to host it so others can grab it which they must use in their validation.
Q: What to do with the risk of overfitting? A: Changing the objective, by adding complexity terms, adding data, adding noise to the data. Your ability to overfit is based on the amount of data, or how prone your objective is to overfitting. Although, if this is changed it is considered a new model.
Q: How are weight changes handled throughout the nodes? Both in the validity of the model and the consensus throughout the nodes (along proof-of-space-time)? A: Validators job is to guarantee that the change is an actual improvement on the objective over the data. In order to

check whether a block is valid they need the new weights and this incentivizes them to download them. Only once this step is complete the node gets the improved model's weights and uses them for all transactions using this model for all subsequent blocks. Note, objective is a function that takes in the architecture, weights, and the data and brings back a score, just the weights can be changed and nothing else (not architecture nor data). Changing anything else is by definition a new model.

Economic Incentives:

Miners are incentivized through block rewards and transaction fees, similar to Bitcoin.
Early participants in the network might receive greater incentives, mirroring the early days of Bitcoin mining.

Implementation Considerations

- Minimal Changes to Bitcoin: The proposal seeks to make only necessary extensions to the existing Bitcoin protocol to accommodate AI-specific transactions.
    - Additions include:
        - Add input and output data to transactions, and validate the inferences and training updates by running the model yourself:
            - Requires (hashed?) model inputs & outputs (& model hash?) within the transaction
        - Add distinction for special transactions:
            - Inferring from a model
                - normal transaction just validate inference as well
            - Updating weights of a model
                - normal transaction just validate model has improved on objective over data
            - Adding a model -
                - Same as updating weights of a model, but just have to add model information (architecture, objective, and data) and initial weights to peer-to-peer storage.
        - Potentially add proof of storage
- Model Management: The protocol would need mechanisms to ensure all participants have the latest, validated versions of AI models and data (i.e. really good peer-to-peer network and file sharing).
- Scalability and Security: Ensuring the network can handle large-scale AI applications without compromising on speed or security is crucial (especially when not all validators can process all transactions).

Conclusion

The vision for an Open Decentralized AI is ambitious but achievable and this is supposed to be a blueprint. By taking inspiration from existing decentralized systems, we can create a resilient, inclusive, and innovative AI ecosystem.

ODA as a Layer 3 on the Lightning Network

Overview

The proposal for "ODA as a Layer 3 on the Lightning Network" aims to offer a potentially better way to implement the same protocol as a Layer 3 on top of Bitcoin and Lightning Protocol, this comes with the enhanced scalability and efficiency of deploying the protocol as such. This approach seeks to solve key issues related to secure and distributed storage, inference, and training of AI models.

Key Components

Network:

The network architecture for the ODA as a Layer 3 implementation on the Lightning Network is designed to handle AI operations—specifically inference and training—efficiently and securely. It operates through a series of independent channels and a dedicated side-chain, which work in conjunction to facilitate AI service provision and training verification. Here’s a detailed explanation of each component and how they interact within the network:

Storage: The primary objective for storage in the ODA Layer 3 model is to have models and data federated across the network, ensuring they are mostly always accessible. The method proposed is a Bittorrent-based peer-to-peer file sharing system, which allows for efficient distribution and redundancy. When models and data are needed, nodes on the network request the data via the peer-to-peer network and verify the integrity through hash checks. This method, while potentially resource-intensive, ensures that the data is available and valid across the network.

Inference: For the inference process, the objective is to establish a system where users pay AI Service Providers for inputs on inference tasks. This would be facilitated by the Lightning Network, which allows for the rapid, scalable transmission of additional input and output data necessary for AI computations. The incentive model here revolves around the creation of payment channels between users and AI Service Providers. Users would continue payments as long as they receive satisfactory responses. If a service provider delivers a poor response, the user can terminate the channel and reclaim any locked funds. This setup promotes a high standard of service, as AI Service Providers with poor performance will lose clients and reputation.

Training: In terms of training AI models, the proposal suggests that miners should be rewarded for utilizing computational resources to enhance models towards their defined objectives. This process requires consensus on a side-chain, where blocks are created based on a miner's ability to improve a model. Such a mechanism ensures that only beneficial modifications to models are validated and added to the blockchain. This blockchain would specifically track and store data on who has contributed to training the models, ensuring transparency and accountability.

Independent Payment Channels for Inference - At the core of the network are the independent channels, similar to those in the Lightning Network, which facilitate direct interactions between AI Service Providers and users. These channels are set up for specific AI inference tasks, where users request AI services (like image recognition, data analysis, etc.) and AI Service Providers deliver results in real-time. Each channel operates with a mechanism where users lock up funds in a micropayment channel with an AI Service Provider. As services are rendered, payments are incrementally made from the user to the service provider. This setup incentivizes the AI Service Providers to consistently deliver high-quality results, as their compensation is directly tied to their performance.

Service Continuity and Quality: If an AI Service Provider delivers unsatisfactory results, the user can terminate the channel and withdraw from the contract, reclaiming any unused funds. This direct relationship ensures that AI Service Providers maintain a high standard of service to retain business and reputation in the network.

Side-Chain for Training Verification

Parallel to the inference channels, a side-chain operates specifically to manage and verify AI training processes. This side-chain is crucial for maintaining a decentralized ledger that records contributions made by various participants (trainers and block verifiers) in training AI models.

Training as Transactions: When miners/trainers improve an AI model by adjusting its weights, these adjustments are submitted to the side-chain as transactions. The side-chain operates under a consensus mechanism that validates these improvements on the model's objective over the public dataset.

Incentives and Rewards: To incentivize the computational effort spent on training, participants are rewarded with transaction fees generated from the inference transactions on the lightning-style network. This creates a sustainable model where the effort to train and improve AI models is directly rewarded by the economic activities generated through inference services. Specifically, the publickeys of the block-creator and the last trainer of the model being used are paid a non-zero amount decided by the Payment Chanel for every transaction.

Security and Integrity: The side-chain also plays a critical role in securing the network by ensuring that only verified and effective training modifications are recorded and recognized. This verification process protects the network from malicious activities or ineffective training efforts that could compromise the performance of deployed AI models. More research needs to be done to understand the attack landscape of this side-chain.

Network Symbiosis and Security

The dual structure of independent channels for inference and a dedicated side-chain for training creates a symbiotic environment within the network. Inference transactions provide a steady flow of fees that fund the training efforts, while the side-chain ensures that the quality and effectiveness of AI models continue to evolve and improve over time. This interdependence secures the network’s economic and operational viability, ensuring that both AI Service Providers and trainers are motivated to maintain high standards of service and contributions.

The incentive for both users and AI Service Providers in this model includes transaction fees paid to the last block producer and the last model trainer. This fee structure encourages ongoing participation and investment in the network’s health and growth. If users and service providers do not support the side-chain with these fees, the network's value might decrease, leading to a reduction in training activities.

Conclusion

By leveraging the principles of the Lightning Network for real-time, low-cost transactions and combining them with a robust side-chain for training verification, the ODA as a Layer 3 network offers a blueprint for a somewhat simpler protocol to achieve open decentralized AI.

Raw Notes - 20240426;

ODA as an extension of the Bitcoin Protocol

- Bitcoin Conference w/ Researchers doing stuff
- He likes how their community is shaped
    - Similar to how bitcoin works
    - contributions are Peer to peer
    - Proposes solutions as PRs and merge the PR when enough cred for it
        - Up to the miners to adopt the change
        - Similar to moving to the next update
    - Learning from the bitcoin community (no one has complete control)
- Wants to move this kinds of idea to AI
    - Data, weights, models don't get released in a similar way
    - Currently, world is dominated by the early players
        - Doesn't like it
        - People must be subscribed to this
        - Early players get the data to improve their models
    - Doesn't want the AI community to stay centralized
- Not completely centralized
    - Researchers able to contribute to advancements
    - Open Source Projects
    - Not as secure and accessible as it needs to be to bring people in
        - People/Market:
            - Bitcoin: Anyone with a wallet can use
                - Wallet: Public Key & Private Key
                - Miners: Anyone with a compute & storage
            - AI:
                - Developers:
                    - Ability to make a new model
                        - New Model: Unique in architecture, objective, or data.
- Open Decentralized AI
    - **Objective**: Create a secure network that allows for distributed storage, inference, and training.
        - Network: Connection of nodes and lite-nodes
            - Node: Miners; Run the ODA protocol, which creates blocks and verifies transactions and disperses information to other nodes
            - Lite-Nodes: Users; Only disperse and receive information to nodes
        - Storage: Federated across the network
            - Both models and data is held within the network
            - If nodes leave the network, models & data are preserved

- Implemented via Distributed Storage System (well understood)
  - New: Trustless (proof of storage/proof of space-time)
  - FileCoin, IONet, IPFS currently implements this
- Inference: Pass an input to the model and receiving an output
  - Needs to have a connection between lite-node and node
    - Done via scattering inputs via the network
  - Add validating the inference to the bitcoin consensus protocol
    - Needs to be handled this way also to maintain the trust aspect bitcoin holds
    - Implemented via the bitcoin consensus protocol
    - This is probably most secure thing to do
    - Problems:
      - Either large minimum requirements for validator or small validators leaving out big transactions
      - Proof-of-Work depends on checking a few things that include:
        - (1) validating if all transactions in a block is correct
        - (2) generating a nonce that satisfies the difficulty constraint on the block - where checking this is as easy as hashing
        - Because validating transactions are quick and usually take the same amount of time the miner is incentivized to grab any transactions that have a high fee and create a block with them
        - In our ODA Network some transactions might take longer to verify than others so there is another incentive to grab transactions that are quick to verify
        - Solution:
          - Make this cancel out with the difficulty proof but how???
            - Fee mechanism?
            - Actually fee mechanism will handle it
    - Future things that can make it easier:
      - zkML and opML
        - Which allows the verification (the output is given by running the input on the function) to be less costly and performed by the the inferencer - this is done by generating a

proof during inference time that could be validated later by the other validator nodes

    - You can use something like lightning for scalability
- Training: Adjusting the weights of an existing model to achieve a higher score of its objective over the data.
  - Anoops questions
    - Who has access to modify the model?
      - If you improve the model, you have the ability to modify it.
        - Risk of overfitting
          - Solution
            - Changing the objective?
            - Adding complexity term
            - Adding data
            - Adding noise to the data
          - Your ability to overfit is based on the amount of data, or how prone your objective is to overfitting
            - If this is changed, it is a new model
        - This way, it maintains its decentralized nature
    - How is this change handled throughout the nodes? Both in validity of the model and the consensus throughout the nodes (along proof-of-space-time)
      - Validators job is to guarantee that the change is an actual improvement on the objective over the data
        - Only once this step is complete does the node get updated with the improved model
        - Objective is a function that takes in the architecture, weights, and the data and brings back a score
    -
  - Just the weights can be changed and nothing else (not architecture nor data)
    - Changing anything else is by definition a new model
  - Problems:
    - Managing memory
      - Model's weights may be different across every user (depends on the miner's willingness to update)
        - Because of this, validation becomes tricky
        - Solution:
          - Treat the training as a transaction

- In order to validate the transaction, you need to be able to download the weights and hash it

- Needs to handle
  - Users/Wallet: People who uses the models to infer
  - Miners: People who maintain the network
  - Developers: People who update the weight and add models
    - Adding Model: Creating a new transaction with the architecture, initial weights, its objective, and its data.
- Wants to make minimum extensions to bitcoin network
- What is the incentive for miners?
  - For bitcoin, early days not worth it to be a miner due to low value of bitcoin
    -
  - For every block you make, you get block rewards and fees.
    - Block reward: minting a coin
    - Fees: How much people are willing to pay to add a block
  - Incentivized to join early since you’d mint more coins
  - Value of fees increase while value of block rewards decrease
- Minimum changes on bitcoin
  - Additions
    - Validation on training & inferences
      - Requires (hashed?) model inputs & outputs (& model hash?) within the transaction
    - Add distinction on transactions for:
      - Adding a model
      - Inferring from a model
      - Updating weights of a model
      - Normal bitcoin transactions
    - Maybe add proof of storage
      - Implementation on distributed systems may be added for storage of models
      - Enforce people to have the updated version of the network?
    - Keeping account of whether or not a computer is capable of running a model-related transaction (inferring, training, etc.)

ODA as a Bitcoin L3

- Need to solve (Securely and Distributed):
    - Storage
        - **Objective:** Models and Data federated across the network
            - Mostly always accessible
        - **Method:** Bittorrent-like peer-to-peer file sharing
            - When Models and Data are needed ask the peer-to-peer network for the download and check the hash (pretty expensive)
        - **Incentive:** To get block reward you need other validators to validate blocks with the existing as well as your new models and data
    - Inference
        - **Objective:** User pays AI Service Provider for input on inference
        - **Method:** Lightning Network with additional Input and Output data
        - **Incentive:** User and AI Service Provider create a channel, and user sends payment until AI Service Provider gives bad response which you can then leave and get back locked funds. AI Service Providers will have a reputation and bad providers will get less clients.
    - Training
        - **Objective:** Miners get rewarded for spending compute to bring models closer to their objective
            - This needs consensus
        - **Method:** Side-Chain with consensus to pay older block
            - Similar to the training done in *ODA as an extension of the Bitcoin Protocol* - blocks are made based on a miners ability to improve a model towards its objective, which other models can validate. This builds a blockchain which holds information about who spent every training the models the inference network are using
        - **Incentive:** When User and AI Service Provider create a channel and make transactions they will pay a fee to the last block producer and last model trainer. (If they don’t Side-Chain will lose value and people will stop training)