Fred Ehrsam Follow
Previously co-founder @Coinbase, trader @GoldmanSachs, computer science @DukeU.
Mar 13 · 9 min read

Blockchain-based Machine Learning Marketplaces

Machine learning models trained on data from blockchain-based marketplaces have the potential to create the world's most powerful artificial intelligences. They combine two potent primitives: private machine learning, which allows for training to be done on sensitive private data without revealing it, and blockchain-based incentives, which allow these systems to attract the best data and models to make them smarter. The result is open marketplaces where anyone can sell their data *and* keep their data private, while developers can use incentives to attract the best data for their algorithms to them.

Constructing these systems is challenging and the requisite building blocks are still being created, but simple initial versions look like they are starting to become possible. I believe these marketplaces will transition us out of the current era of Web 2.0 data monopolies into a Web 3.0 era of open competition for data and algorithms, where both are directly monetized.

Origin

The base of this idea came in 2015 from talking with Richard of Numerai. Numerai is a hedge fund that sends encrypted market data to any data scientist who wants to compete to model the stock market. Numerai combines the best model submissions into a "metamodel", trades that metamodel, and pays data scientists whose models perform well.

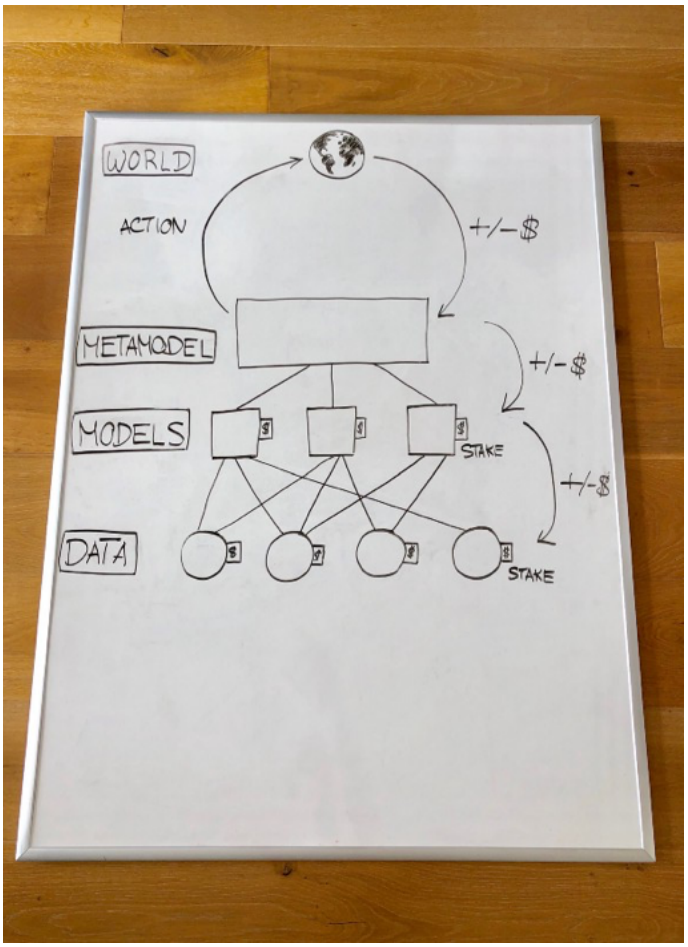
Having data scientists compete seemed like a powerful idea. So it got me thinking: can you create a fully decentralized version of this system that could be generalized to any problem? I believe the answer is yes.

Construction

As an example, let's try creating a fully decentralized system for trading cryptocurrencies on decentralized exchanges. This is one of many potential constructions:

Data Data providers stake data and make it available to modelers.

Model building Modelers choose what data to use and create models. Training is done using a secure computation method which allows models to be trained without revealing the underlying data. Models are staked as well.



Metamodel building A metamodel is created based on an algorithm that takes into account the staking of each model.

Creating a metamodel is optional—you can imagine models that are used without being combined into a metamodel.

Using the metamodel A smart contract takes the metamodel and trades programmatically through decentralized exchange mechanisms on-chain.

Distributing gains/losses After some time period passes, trading produces a profit or loss. This profit or loss is divided up amongst contributors to the metamodel based on how much smarter they made it. Models which contributed negatively have some or all of their staked funds taken. Models then turn around and perform similar distributions/stake slashing to their data providers.

Verifiable computation Computation for each step is either performed centralized but verifiable and challengeable using a verification game like Truebit or decentralized using secure multiparty computation.

Hosting Data and models are either hosted on IPFS or with nodes in a secure multiparty computation network, as on-chain storage would be too expensive.

What makes this system powerful?

Incentives to attract the best data globally Incentives to attract data are the most potent part of the system as data tends to be the limiting factor for most machine learning. In the same way Bitcoin created an emergent system with the most compute power in the world through open incentives, a properly engineered incentive structure for data would cause the best data in the world for your application to come to you. And it's nearly impossible to shut down a system where data is coming from thousands or millions of sources.

Competition between algorithms Creates open competition between models/algorithms in places where it previously didn't exist. Picture a decentralized Facebook with thousands of competing newsfeed algorithms.

Transparency in rewards Data and model providers can see they are getting the fair value of what they've submitted since all computation is verifiable, making them far more likely to participate.

Automation Taking action on-chain and generating value directly in tokens creates an automated and trustless closed loop.

Network effects Multi-sided network effects from users, data providers, and data scientists make the system self-reinforcing. The better it performs, the more capital it attracts, which means more potential payouts, which attracts more data providers and data scientists, who make the system smarter, which in turn attracts more capital, and back around again.

Privacy

In addition to the points above, a major feature is privacy. It allows 1) people to submit data that otherwise would be too private to share and 2) prevents the economic value of the data and models from leaking. If left unencrypted in the open, the data and models will be copied for free and used by others who have not contributed any work (the "free rider" problem).

A partial solution to the free rider problem is to privately sell data. Even if buyers choose to resell or release the data, its value decays with time. However, this approach restricts us to short duration use cases and still creates typical privacy concerns. As a result, the more complicated but powerful approach is to use a form of secure computation.

Secure computation

Secure computation methods allow models to train on data without revealing the data itself. There are 3 main forms of secure computation being used and researched today: homomorphic encryption (HE), secure multi-party computation (MPC), and zero knowledge proofs (ZKPs). Multiparty computation is most commonly used for private machine learning at the moment, as homomorphic encryption tends to be too slow and it's not obvious how to apply ZKPs to machine learning. Secure computation methods are on the bleeding edge of computer science research. They are often orders of magnitude slower than regular computation and represent the main bottleneck to the system, but have been improving in recent years.

The Ultimate Recommender System

To illustrate the potential of private machine learning, imagine an app called "The Ultimate Recommender System". It watches everything you do on your devices: your browsing history, everything you do in your apps, the pictures on your phone, location data, spending history, wearable sensors, text messages, cameras in your home, the camera on your future AR glasses. It then gives you recommendations: the next web site you should visit, article to read, song to listen to, or product to buy.

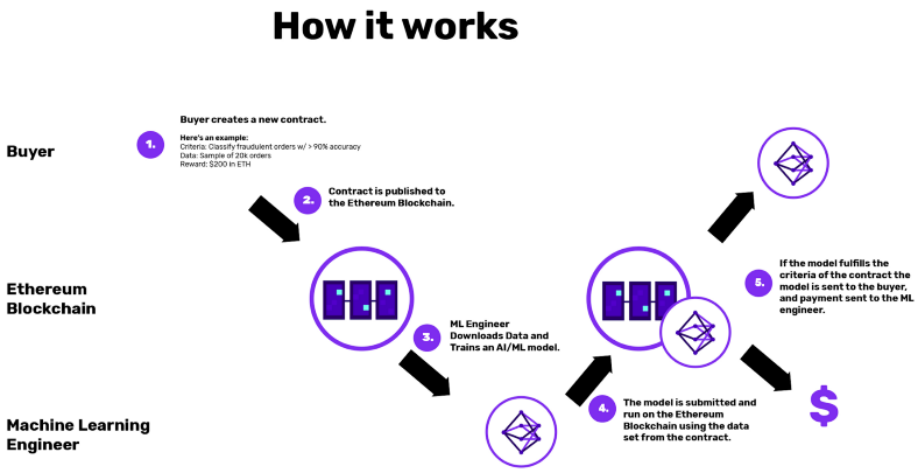
This recommender system would be extremely potent. More than any of the existing data silos of Google, Facebook, or others could ever be because it has a maximally longitudinal view of you and it can **learn from data that otherwise would be too private to consider sharing**. Similar to the prior cryptocurrency trading system example, it would work by allowing a marketplace of models focused on different areas (ex: web site recommendations, music) to compete for access to your encrypted data and recommend things to you, and perhaps even pay you for contributing your data or your attention to the recommendations generated.

Google's federated learning and Apple's differential privacy are one step in this private machine learning direction, but still require trust, don't allow users to directly examine their security, and keep data siloed.

Current approaches

It's very early. Few groups have anything working and most are trying to bite off one piece at a time.

A simple construction from Algorithmia Research places a bounty on a model that is accurate above a certain backtesting threshold:



Simple construction creating a bounty on a machine learning model by Algorithmia Research

Numerai currently takes things three steps further: it uses encrypted data (although not fully homomorphically), it combines crowdsourced

models into a metamodel, and it rewards models based on future performance (in this case, one week of stock trading) rather than backtesting through a native Ethereum token called Numeraire. Data scientists must stake Numeraire as skin in the game, incentivizing performance on what will happen (future performance), not what has happened (backtested performance). However, it currently centrally distributes data, limiting what feels like the most important ingredient.

No one has created a successful blockchain-based marketplace for data yet. The Ocean is an early attempt to outline one.

Still others are starting by building secure compute networks. Openmined is creating a multiparty compute network for training machine learning models on top of Unity that can run on any device, including game consoles (similar to Folding at Home), then expanding to secure MPC. Enigma has a similar tact.

A fascinating end state would be **mutually owned metamodels which give data providers and model creators ownership proportional to how much smarter they've made them**. The models would be tokenized, could pay dividends over time, and potentially even be governed by those who trained them. A sort of mutually owned hive mind. The original Openmined video is the closest construction to this I have seen so far.

What approaches are likely to work first?

I won't claim to know what precise construction is best, but I have some ideas.

One thesis I use to evaluate blockchain ideas is: on a spectrum of physically native to digitally native to blockchain native, the more blockchain native, the better. The less blockchain native, the more trusted third parties are introduced, increasing complexity and reducing ease of use as a building block with other systems.

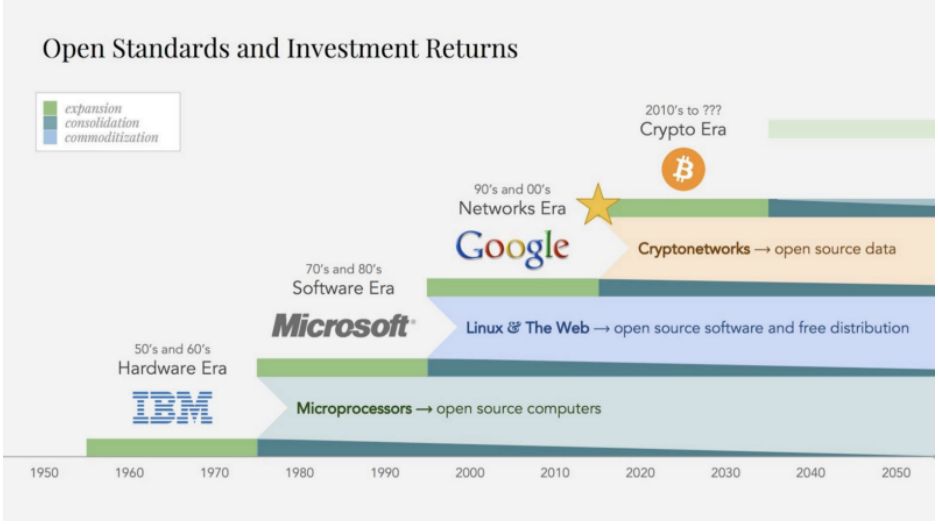
Here, I think that means a system is more likely to work if the value created is quantifiable—ideally directly in the form of money, and better yet, tokens. This allows for a clean, closed-loop system. Compare the

prior example of a cryptocurrency trading system to one that identifies tumors in X-rays. In the latter, you'd need to convince an insurance company that the X-ray model is valuable, negotiate how valuable, and then trust a small group of physically present people to verify the model's success/failure.

That's not to say more clearly positive sum for society uses which are digitally native won't emerge. Recommender systems like the one previously mentioned could be enormously useful. If attached to curation markets, they are another case where the model can take action programmatically on-chain and the system's reward is tokens (in this case from the curation market), again creating a clean closed loop. It seems obscure now, but I expect the realm of blockchain-native tasks to expand with time.

Implications

First, decentralized machine learning marketplaces can dismantle the data monopolies of the current tech giants. They standardize and commoditize the main source of value creation on the internet for the last 20 years: proprietary data networks and the strong network effects surrounding them. As a result, **value creation gets moved up the stack from data to algorithms.**



Standardization and commoditization cycles in tech, where we are nearing the end of the networks era of data monopolies. Chart from Placeholder.

Said another way, they **create a direct business model for AI**. Both feeding and training it.

Second, they create the most powerful AI systems in the world, attracting the best data and models to them through direct economic incentives. Their strength increases through multi-sided network effects. As the Web 2.0 era data network monopolies become commoditized, they seem like a good candidate for the next re-aggregation point. We are probably a few years out from this, but it seems directionally correct.

Third, as the recommender system example shows, **search gets inverted**. Instead of people searching for products, products search and compete for people (credit to Brad for this framing). Everyone might have personal curation markets, where recommender systems compete to place the most relevant content in their feed, and relevance is defined by the individual.

Fourth, they allow us to get the same benefits of the powerful machine-learning based services we are used to from companies like Google and Facebook without giving away our data.

Fifth, machine learning can advance more quickly, as any engineer can access an open marketplace for data, not just a small group of engineers in the large Web 2.0 companies.

Challenges

First and foremost, secure computation methods are currently very slow and machine learning is already computationally expensive. On the flip side, interest in secure computation methods has started picking and performance is increasing. I have seen novel approaches with significant performance improvements to HE, MPC, and ZKPs within the last 6 months.

Calculting the value a particular set of data or model provides to the metamodel is hard.

Cleaning and formatting crowdsourced data is challenging. We're likely to see some combination of tools, standardization, and small businesses

pop up to solve this.

Finally and ironically, the business model for creating the generalized construction of this sort of system is less clear than creating an individual instance it. This seems to be true of a lot of new crypto primitives, including curation markets.

Conclusion

The combination of private machine learning with blockchain incentives can create the strongest machine intelligences in a wide variety of applications. There are significant technical challenges which feel solvable over time. Their long term potential is enormous and a welcome shift away from the current grip large internet companies have on data. They are also a bit scary—they bootstrap themselves into existence, self-reinforce, consume private data, and become almost impossible to shut down, making me wonder if creating them is summoning a more powerful Moloch than ever before. In any case, they are another example of how cryptocurrencies will slowly, then suddenly make their way into every industry.

Thanks to Andrew Trask, Richard Craib, Trent McConaghy, Brad Burnham, Joel Monegro, Simon de la Rouviere, Gavin Uhma, Morten Dahl, Jonathan Libov, Matt Huang, Laura Behrens Wu, Naval Ravikant, and Daniel Gross for conversations which contributed to this post.