Color Code:
Yellow = intro/concl/general
Red = problem
Green = sol'n
Blue = sources of interest
Purple = add'l/themes/notable

Gensyn Litepaper - Gensyn

docs.gensyn.ai/litepaper

Gensyn Litepaper

The hyperscale, cost-efficient compute protocol for the world's deep learning models

Background

1
AI compute demand > supply
• GPT3 175B
• DALL-E

The computational complexity of state-of-the-art Artificial Intelligence (AI) systems is doubling every 3 months, vastly outstripping compute supply. As a founding team--whether we've been publishing research into the evolution of deep neural architectures or building hurricane damage prediction models--we've experienced these limits first hand. GPT-3 175B, the largest GPT-3 model proposed by OpenAI in Brown et al. (2020) used a cluster of 1,000 NVIDIA Tesla V100 GPUs for training - roughly equivalent to 355 years of training on a single device. DALL-E from Ramesh et al. (2021), another Transformer model from OpenAI, has 12 billion parameters and was trained on over 400 million captioned images. OpenAI bore the cost of training DALL-E but controversially refused to open source the model, meaning that perhaps one of the most important state-of-the-art multimodal deep learning models remains inaccessible to all but a select few. The huge resource requirements for building these foundation models create significant barriers to access, and, without a method to pool resources whilst still capturing value, will likely cause stagnation in AI advancement. Many believe that these generalised models are the key to unlocking Artificial General Intelligence (AGI), making the current method of training in isolated, artificial silos seem absurd.

2
State of art ↓ open sourced

3
• Foundation models $$$
• Siloed training
• NO resource sharing

Current solutions which provide access to compute supply are either oligopolistic and expensive or simply unworkable given the complexity of compute required for large-scale AI. Meeting the ballooning demand requires a system which cost-efficiently leverages *all* available compute (as opposed to today's ~40% global processor utilisation). Compounding

4
5 = pool all compute avail. vs squo 40%

(5) Chip advances ↓ (5) ↑ demand

this problem right now is the fact that the compute supply itself is hamstrung by asymptotic advances in microprocessor performance - alongside supply chain and geopolitical chip shortages.

(6) Prob. = cost ↑ compute

We've spoken with more than 150 CTOs, machine learning (ML) researchers, and ML engineers who consistently describe the painful trade-off between purchasing their own hardware and sacrificing scalability, or renting scalable cloud resources for vastly increased costs. They recognise that cloud costs are typically inflated by provider profit margins and often wonder why on-demand, serverless-style compute doesn't exist for their ML work.

(7) Existing projects ↓ handle state dependent ML tasks

Voluntary grid computing services like SETI@Home, Folding@Home, and BOINC demonstrate that trustless, voluntarily-networked, latent compute can be used to solve some of humanity's biggest problems. However, they predominantly solve embarrassingly parallel problems such as 3D rendering, where computational work can be trivially split and verified owing to its state independence. ML problems (besides niche tasks like hyperparameter optimisation) are inherently state dependent, requiring new methods for both parallelisation and verification. Volunteer networks alsofunction only by modelling participants as rational actors in a philanthropic system; adding financial transactions drastically changes the incentive mechanisms and introduces the spectre of exploitation.

(8) Volunteer networks ↓ (5)

Decentralised blockchain protocols extend the concept of grid computing into financially-incentivised, trustless environments. Specifically, Ethereum moved the space beyond the transaction use-cases of Bitcoin to more general on-chain computational work. This was achieved by incorporating a Turing-complete language (Solidity) and rewarding compute providers through variable gas fees.

(9) Ethereum too $$$ to compute on-chain

Ethereum, however, achieves trustless consensus only via extremely expensive on-chain replication of work. This is completely unsuitable for deep learning. Training a small MNIST neural network (~400M processor operations) takes ~8 minutes on an average laptop but would take ~80 days on Ethereum at a cost of approximately $32m. To address this, Truebit showed that it's possible to perform simple computational work off-chain (and thus with less overhead) and prove to the chain that it was performed correctly. They achieved this by modelling participants as financially-rational actors and carefully constructing incentive structures. Specifically, they solved the verifier's dilemma by intermittently requiring workers to produce incorrect work and awarding verifiers with a jackpot if they spot it.

(10) Truebit shows off-chain (3)

(11) Need off-chain compute system for computationally expensive DL

Despite these improvements, the work must still be replicated off-chain. This is unsuitable for activities with extreme computational expense (e.g. deep learning), and a cost-efficient off-chain compute system must exist if deep learning work is to be serviced in a trustless way.

Problem

A protocol which trustlessly connects and verifies off-chain deep learning work in a cost efficient way has five main challenges.

Work verification

In order to build a truly trustless compute network, with economic incentives for participation, the network must have a way to verify that deep learning computational work has actually been performed as promised. Central to this problem is the state dependencyof deep learning models; that is, each subsequent layer in a deep learning model takes as an input the output of the previous layer. Therefore, to validate work has been completed at a specific point, all work up to and including that point must be performed. We'll cover this in more detail later but it's a fundamental problem that until now has had no viable solutions.

Market

A marketplace for compute is subject to the same supply and demand issues that any new marketplace faces, with a few unique challenges too. Principally there is a cold-start issue, where supply and demand liquidity need to roughly match from the beginning in order to grow successfully. In order to capture latent compute supply, there must be a clear reward for participants to pledge their compute time. Computational work must be tracked and proportional payments made to the providers in a timely manner. For more traditional marketplaces, this is performed using intermediaries which handle administration and onboarding, with minimum payouts to reduce overheads. Unfortunately, this approach becomes costly to scale and results in a threshold equilibrium where only a small portion of the supply can be economically captured.

Ex-ante work estimation

Similar to Ethereum, ML computational work is subject to the halting problem - where it is at times impossible to quantify the amount of computational work required by a defined task and more specifically whether it will ever end (or halt). In the context of deep learning, this has become more significant relatively recently as models and frameworks have switched from static graph construction to dynamic construction and execution.

Privacy

With the growth of stronger personal privacy regulations around the world (e.g. GDPR, CCPA, LGPD), privacy-conscious design and development has become an expected practice for organisations. Whilst large amounts of ML research can be performed on open datasets, final model fine-tuning often uses proprietary user data. More specifically, in our interviews with ML engineers and CTOs, they indicated that data privacy was orders of magnitude more important than model privacy.

Parallelisation

State of the art deep learning models are typically trained in parallel over large clusters of hardware in order to access scale that is unachievable with a single device. The techniques required to achieve this parallelisation have improved drastically through recent research, with current state-of-the-art transformer models like Switch Transformers proposed by Fedus, Zoph, and Shazeer (2021) now inherently highlyparallelised by nature. Combining the performance requirements of the ML work with the untrusted and unreliable nature of the compute sources means that a high degree of parallelisation is essential in any solution.

Solution

Gensyn Protocol

The Gensyn Protocol is a layer-1 trustless protocol for deep learning computation that directly and immediately rewards supply-side participants for pledging their compute time to the network and performing ML tasks. The protocol requires no administrative overseer or legal enforcement, rather facilitating task distribution and payments programmatically through smart contracts. As described above, the fundamental challenge in building this network is the verification of completed ML work. This is a highly complex problem that sits at the intersection of complexity theory, game theory, cryptography, and optimisation.

A simple solution is to check the honesty of workers by re-doing their work. At a bare minimum, this requires a doubling of the operations required ('single replication'); however, even with replication, the issue of trust remains unless the verifying party is the actual requestor of the work (in which case, they wouldn't request the work as they'd simply perform it themselves). Therefore, ensuring the honesty of the verifying party can generate an infinite chain of replication, where each new verifier is required to check the work of the previous verifier.

We solve this verification problem by interlocking three key concepts into a robust solution that is
\gt 1,350 \%>1,350%
more efficient than existing best-practice replication methods; in doing so, it solves the infinite-chain problem. The key concepts are:

Probabilistic proof-of-learning

Following Jia et al. (2021), we use the metadata from gradient-based optimisation processes to construct certificates of work performed, which can be verified quickly through replication

of certain stages.

Graph-based pinpoint protocol

Following Zheng et al. (2021), we use a multi-granular, graph-based pinpoint protocol and cross-evaluator consistent execution to allow verification work to be re-run and compared for consistency, and ultimately confirmed by the chain itself.

Truebit-style incentive game

Following Teutsch and Reitwießner (2019), we use staking and slashing to construct an incentive game ensuring each financially-rational participant behaves honestly and performs their intended tasks.

Participants

These concepts are used to construct a system with four main participants: Submitters, Solvers, Verifiers, and Whistleblowers.

Submitters

Submitters are the end-users of the system, providing tasks that will be computed and paying for units of work completed.

Solvers

Solvers are the main workers of the system, performing the model training and generating proofs to be checked by Verifiers.

Verifiers

Verifiers are key to linking the non-deterministic training process to a deterministic linear computation, replicating portions of the Solvers' proofs and comparing distances with expected thresholds.

Whistleblowers

Whistleblowers are the final line of defence, checking Verifiers' work and challenging in the hope of receiving a jackpot payout.

Usage

Typical protocol usage will pass through eight stages, with the above roles performing specific tasks.

(A) Task Submission

Tasks take the form of three specific pieces of information:

1. 1.
   Metadata describing the task and hyperparameters;
2. 2.
   A model binary (or skeleton architecture); and
3. 3.
   Publicly accessible, pre-processed training data.

(?) But isn't this a huge limitation to advancing ML? How to access more data beyond publicly avail?

In order to submit a task, Submitters specify the details of the task in a machine-readable format and submit these to the chain along with the publicly accessible locations of the model binary (or machine-readable architecture) and pre-processed training data. The publicly available data could be stored in a simple object store such as Amazon's S3 or in a decentralised store like IPFS, Arweave, or Subspace.

For privacy-preservation, models can be constructed using secure mapping layers (a form of functional encryption) as proposed by Lan, Liu, and Li (2020) and the publicly accessible training data encrypted. In this way, models can be trained on ciphertext with a small accuracy penalty (
\lto.5\%<0.5%
).

When submitting a task, an estimate of required work is generated by constructing and unrolling a computational graph into the required operations. These operations are weighted using values similar to Ethereum's Opcode gas values in order to calculate a rough sum of computational work to be performed. The transaction fee paid by the Submitter can then use this estimate, with any excess (e.g. due to pessimistic profiling) returned to the Submitter after computation. Crucially, unrolling the graph requires set limits to be placed on logic which can trigger the halting problem.

(S) to halting problem & ex-ante work estimation

Tasks form the smallest quantity of ML work that can be pushed to the protocol. Using parallelisation, larger computational workloads can be split into sets of tasks and pushed to the network asynchronously. Using this approach, large-scale language models and other state-of-the-art models can be built, as Diskin et al. (2021) demonstrated with volunteer compute.

(S) to single device / scale via asynchronization

Ⓑ Profiling

The profiling process establishes a baseline distance threshold for the proof-of-learning verification. Verifiers will periodically grab profiling tasks and generate variation thresholds for proof-of-learning comparisons. To generate a threshold, a Verifier will deterministically run and re-run portions of the training with different random seeds, generating and checking their own proofs. In doing this, the Verifier will build up an aggregate expected distance threshold that can later be used as a threshold to validate the non-deterministic work of the Solvers.

In order to ensure the honesty of the Verifiers when generating the distance thresholds, Whistleblowers are expected to re-run the profiling work and challenge Verifiers where appropriate, using the same graph-based pinpoint challenge and contract arbitration mechanisms described below.

Ⓒ Training

Following profiling, the task enters the common task pool (analogous to the Ethereum mempool). A single Solver is selected to perform the task and the task is removed from the task pool. The Solver performs the task according to the metadata submitted by the Submitter and using the model and training data supplied. Whilst performing the training task, the Solver also generates a proof-of-learning by checkpointing at a scheduled interval and storing metadata from the training process (including parameters) so that the following optimisation step can be replicated as accurately as possible by a Verifier.

Ⓓ Proof generation

Proof generation follows the process outlined in Jia et al. (2021), whereby Solvers periodically store the model weights or updates along with the corresponding indices from the training dataset identifying the samples that were used to generate the weight updates. The checkpoint frequency can be tuned to provide stronger guarantees or to save on storage space. Proofs can be “stacked”, meaning that a proof can start from the random distribution used to initialise the weights or from pre-trained weights generated with their own proof. This allows the protocol to build up a set of already-proven, pre-trained base models (i.e. foundation models) which can be fine-tuned for more specific tasks.

Verification of proof

Following task completion, Solvers register the completion of the task with the chain and present their proof-of-learning in a publicly accessible location for access by Verifiers. Verifiers pick up verification tasks from a common task pool (again analogous to the

Ethereum mempool) and perform the computational work to re-run portions of the proof and perform distance calculations. The resulting distances are then used by the chain (along with the threshold calculated during the profiling stage) to determine whether the verification matches the proof.

Graph-based pinpoint challenge

Following verification of the proof-of-learning, Whistleblowers can replicate Verifier work in order to check that the verification work itself has been performed correctly. In the event that a Whistleblower believes that verification has been performed incorrectly (maliciously or not) they can challenge the Verifier to contract arbitration in order to receive a reward. This reward can come from Solver and Verifier deposits in the case of a true positive or from the jackpot treasury in the case of a false positive. The challenge process follows the procedure outlined in Zheng et al. (2021) and uses the chain itself to perform the arbitration. Following Teutsch and Reitwießner (2019), Whistleblowers (in their case Verifiers) are only expected to verify and subsequently challenge work in the event that they expect to receive appropriate compensation. In practice, this means that Whistleblowers are expected to join and leave the network depending on the number of other active (i.e. with live deposits and challenging) Whistleblowers. Therefore, the expected default strategy for any Whistleblower is to join the network when there are a low number of other Whistleblowers, post a deposit, randomly choose an active task, and begin their verification process. Following the conclusion of the first task, they would grab another random active task and repeat until the number of Whistleblowers increases above their determined payout threshold, whereupon they would leave the network (or more likely, switch to performing another role in the network--Verifier or Solver--depending on their hardware capabilities) until the situation reverses again.

Contract arbitration

When a Verifier is challenged by a Whistleblower, they enter a process with the chain to whittle down the location of a disputed operation or input, culminating in the chain performing the final basic operation and determining whether the challenge was justified. In order to maintain the honesty of the Whistleblowers and overcome the verifier's dilemma, the protocol introduces periodic forced errors with jackpot payouts, as proposed by Teutsch and Reitwießner (2019).

Settlement

In the settlement process, participants are paid according to the conclusions of the probabilistic and deterministic checks. Different payments are made in different scenarios depending on the outcome of the prior verification and challenges.

If the work is deemed to have been performed correctly and all checks have passed, the Solver and Verifier are both rewarded according to the operations performed.

Scale and cost-efficiency

Building the marketplace as a Web3 protocol removes the centralised overheads on scaling and reduces the barriers-to-entry for new supply participants, allowing the network to potentially encompass every computing device in the world. Connecting all devices through a single decentralised network provides a level of scalability that is currently impossible to achieve through any existing provider, giving unprecedented on-demand access to the entirety of the world's compute supply. For end-users, this completely dismantles the cost vs scale dilemma and provides a transparent, low-cost for potentially infinite scalability (up to worldwide physical hardware limits).

Creating a marketplace where prices are determined by market dynamics, and the market is open to all participants, allows the unit cost of ML compute to settle into its fair equilibrium. This sidesteps the usual moats that large providers enjoy, significantly drives down prices, and facilitates truly global competition at the resource level. Whilst current compute costs for end-users incorporate large margins for their oligopolistic suppliers, the Gensyn Protocol will ensure that the remaining margin, decreased by fair competition, is proportionally captured by every participant.

With Ethereum's move from proof-of-work to proof-of-stake in Eth2, many miners with powerful GPUs (e.g. NVIDIA V100s) will be left without a yield. These miners can currently expect a return of around $0.20 to $0.35 per hour, which even now, when subtracting amortized capital purchase and electricity costs, provides a tight marginal return. The delta between the current yield expected by these miners with ML-capable hardware and the average hourly cost of the same hardware from the main providers, alongside the likely disappearance of Eth mining, forms a huge opportunity for the Gensyn Protocol; it also allows the hardware to generate returns on useful processor cycles - as opposed to merely calculating hashes in proof-of-work systems. Capturing this mining supply, alongside other general sources of latent compute, leads to a projected hourly cost of around $0.40 per hour for NVIDIA V100-equivalent computation on the Gensyn Protocol, 80% cheaper than AWS on-demand.

Provider

Approximate hourly cost for ML training work (V100-equivalent)

Scalability

Ethereum

$15,700

Low

Truebit (+ Ethereum)

$12

Low

GCP on-demand

$2.50

Medium

AWS on-demand

$2

Medium

Golem Network

$1.20

Low

Vast.ai

$1.10

Low

AWS spot instances (unreliable)

$0.90

Medium

GCP spot instances (unreliable)

$0.75

Medium

Gensyn (projected)

$0.40

High

Single GPU in datacentre

$0.40

None

Single personal GPU

$0.28

None

Protocol evaluation

We evaluate our solution through Python simulations in order to assess the magnitude of performance gains delivered by the Gensyn protocol. In this instance, we gauge performance as the aggregate time in seconds taken to complete a 100 epoch training job on a small MNIST image classification model. We test this on a 6-Core Intel Core i7 processor.

Simulation test / compare

We compare the protocol with 3 alternative approaches: running the model locally (as opposed to using any protocol), running the model using Truebit-inspired replication (with 7 verifiers), and running the model on Ethereum.

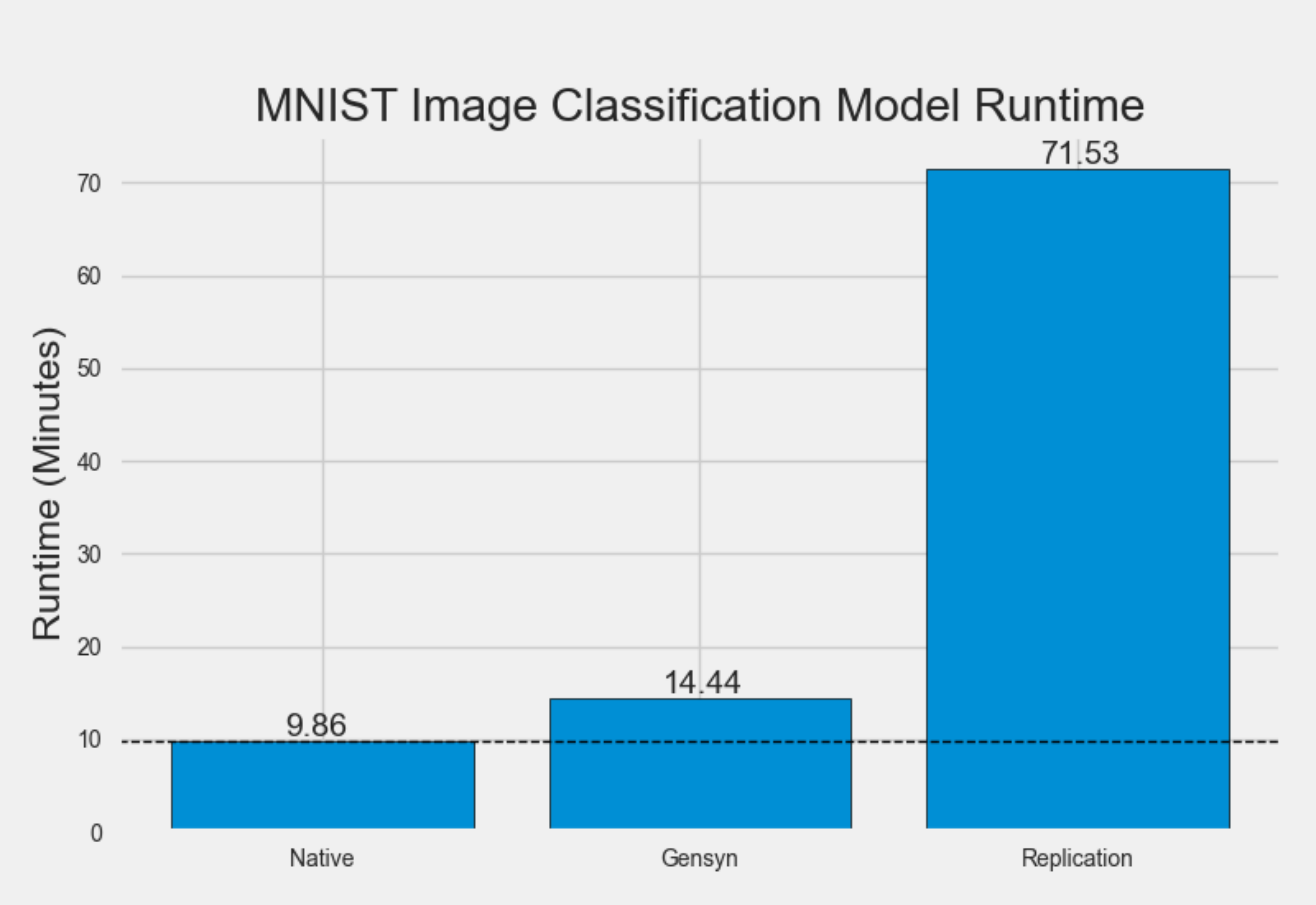
% perf. gains vs Truebit + Eth

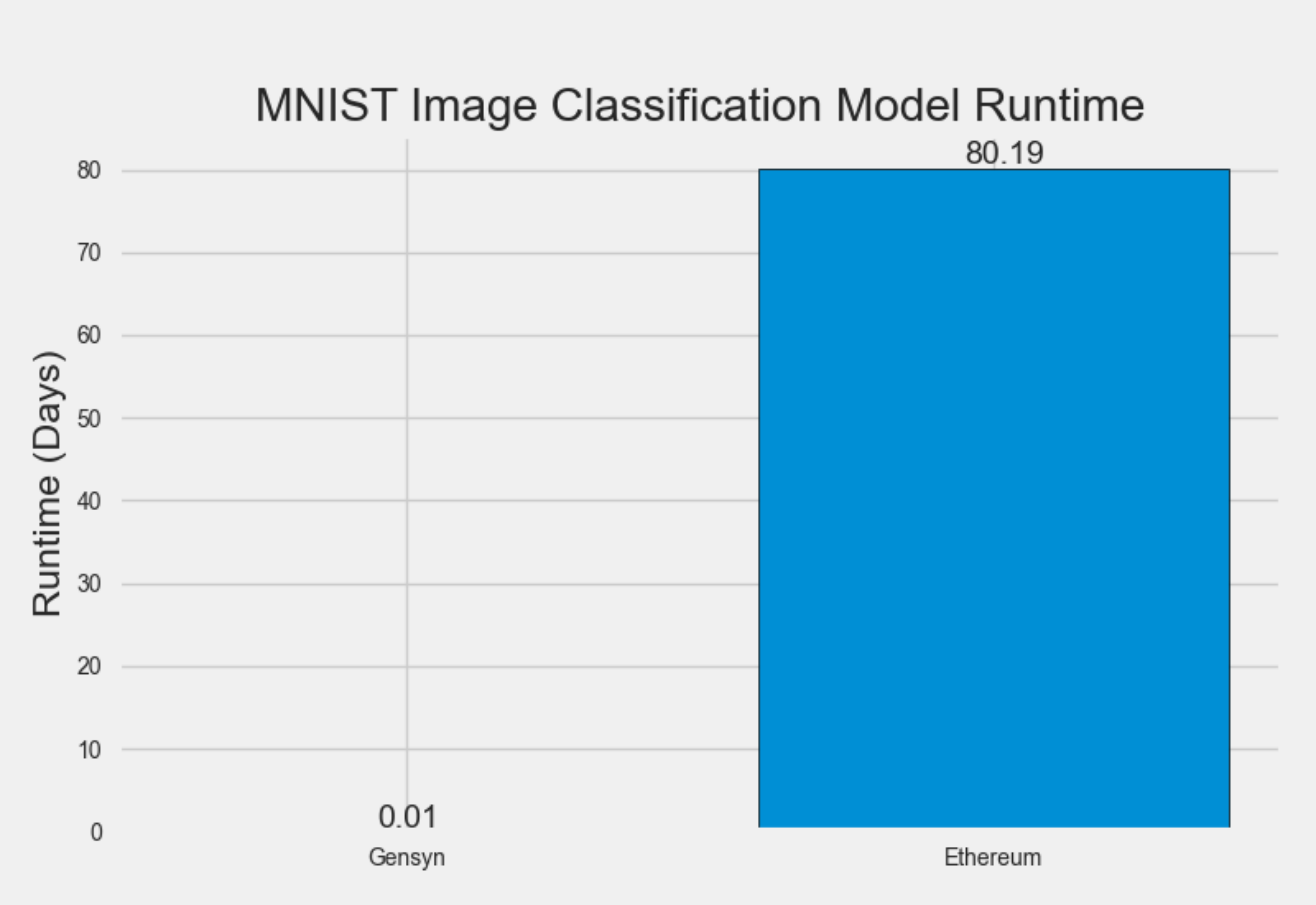
Despite the code lacking production-level optimisations, the results show that the Gensyn protocol adds a ~46% time overhead to model training representing a 1,350% performance gain versus Truebit-style replication and 2,522,477% gain versus Ethereum.

Runtime comparison between Gensyn and Truebit-style replication for an MNIST image classification model

Runtime comparison between Gensyn and Ethereum (theoretical) for an MNIST image classification model

Underlying source: Gensyn
Annotations: Sandeep C. Ramesh
More: sandeepramesh.com





Decentralisation and governance

Governance

Gensyn Limited is the initial entity that is developing the protocol, hiring the team, and managing the IP (prior to open source launch). Gensyn Limited is a fully remote company, hiring talent from all over the world. Following the Token Generation Event (TGE), Gensyn Limited will handle technical development and the Gensyn Foundation will represent the interests of the protocol.

• Foundation & Develop-ment gov. model

Tokens will be issued at the TGE by the Gensyn Foundation, which will be governed in a decentralised manner by an elected council and make decisions based on on-chain proposals and referenda. Initially, members of the council will be tightly mapped to core members of Gensyn Limited and the early community in order to quickly develop the protocol. As time goes on, the council will become more decentralised.

• Phased decent. approach

The Gensyn Foundation will also control a treasury that will be directed by proposals to further the aims of the protocol by funding the continued development of the protocol itself and the overall ecosystem. The treasury will primarily be funded by taking a very small

percentage of each task fee.

Future development

Research

We will continue our research into three main areas to improve the protocol: probabilistic verification of ML training using metadata from the optimisation process, pinpoint verification of deterministic ML work for on-chain proof, and parallelisation of ML models over heterogeneous hardware with latency constraints.

This research will strengthen the work verification guarantees and expand the utility of the protocol to include more model primitives and a wider variety of model types.

Development

Development of the Gensyn protocol will follow three high-level phases: testnet, canarynet, mainnet.

Testnet

Initial development will focus on building a testnet implementation of the core technology. Tokens used by the testnet will be non-permanent, and users of the testnet will be early adopters and core members of the community who will be rewarded at the TGE.

Canarynet

Following successful testnet iteration, the protocol will launch as a canary network parachain on the Kusama relay chain. This phase will involve launching the canary utility token that will have real economic value. The canary network can be seen as a beta version of the protocol with access to the newest features and some risk associated with its use. Long-term, canary networks typically offer slightly lower prices and access to bleeding-edge R&D functionality in exchange for this slight risk.

Mainnet

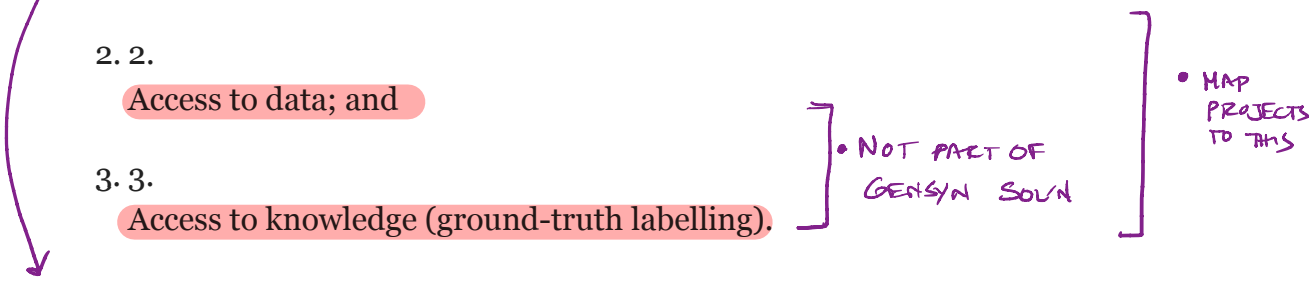
Following a successful parachain launch on the Kusama relay chain, the next phase will be to launch the final live parachain on the Polkadot relay chain. This phase will include the launch of the mainnet utility token that will be the main utility token for the protocol. The mainnet

will be the hardened, live protocol for full use by any organisation or individual. Features or changes will go through testnet and canarynet iteration before launching on the mainnet.

Ecosystem

The Gensyn Protocol will be a foundational layer for ML compute, similar to Ethereum for smart contract execution. Going forward, we expect others to build on top of the protocol to provide rich user experiences and specific functionality in numerous niches. We expect this burgeoning ecosystem to start with expert-knowledge-based applications, allowing non-experts to build and deploy ML solutions using abstractions similar to existing Web2 solutions such as Amazon's SageMaker and DataRobot.
Besides human knowledge in model design, there are three fundamental problems slowing the progress of applied ML:

1. 1. Access to compute power;
2. 2. Access to data; and
3. 3. Access to knowledge (ground-truth labelling).



Gensyn solves the first problem by providing on-demand access to globally scalable compute at its fair market price. The Gensyn Foundation will seek to encourage solutions to two and three through research, funding, and collaborations with other protocols.

Long-term vision

The Gensyn Protocol will enable anyone to train ML models for any task using a self-organising network that encompasses every source of compute power in existence.

As Web3 Dapps increase in complexity and infrastructure requirements, they are forced to fall back onto Web2 where Web3 resources don't exist. By decentralising ML compute, the Gensyn Protocol brings a crucial infrastructure component natively to Web3 - reducing reliance on Web2 and further strengthening and decentralising the entire ecosystem.

Deep learning has shown incredible generalisation power and looks set to play a huge part in the future of ML. Foundation models, trained on the Gensyn Protocol, will be decentralised and globally owned - allowing humanity to equally benefit from collaborative ML development and training. Building on these foundation models using fine-tuning will be as

simple as defining a task and paying a fair market price for the fine-tuning work - removing the barriers that currently exist.

For decades, ML has progressed in silos, both academic and industrial. The Gensyn Protocol connects these silos through a common infrastructure with decentralised ownership, allowing all of humanity to quickly and collectively explore the future of artificial intelligence as equal pioneers. Combining this network with hierarchically-trained and collectively-owned foundation models provides a path towards a true realisation of AGI - the next step for humanity.

Get involved

You can follow our progress on Twitter and join our community on Discord. If you're interested in contributing compute resources, using the network for ML tasks, or joining us then please send us a message. We'd love to chat.

Last modified 5mo ago