It is now safe to say that machine learning has become a part of our day-to-day lives. With recent advancements, the demand on computational resources needed to conduct the computations has increased significantly, creating a need to outsource these computations. This in turn has introduced concerns about the trustworthiness of such computations, as they are no longer being performed locally under user's supervision. In response to this issue the cryptological solution has been proposed in the form of zero-knowledge proofs (ZKP) — small quickly verifiable proofs of validity of conducted computation with little overhead for the prover and disclosing no additional information (e.g. training dataset) to the verifier.

ZKP can be applied to various aspects of machine learning. A comprehensive survey on ZKP applications to machine learning was conducted by Xing et al. (2023), where the authors analyze and classify existing works based on their technical approaches. The authors distinguish two main issues outsourced computations pose: inference verification and training process verification. These issues led to two corresponding notions of proofs: proof-of-inference and proof-of-training, albeit the terminology may differ, considering this is an emerging area of computer science and cryptology. Due to sequential nature of the training and inference processes, most of the proving schemes use the so-called GKR-style protocols and various recursive proof compositions.

The GKR protocol, first introduced by Goldwasser et al. (2008), combines the sum-check protocol and introduces "wiring predicates" that allow for iterative proofs of arithmetic circuit evaluation, which is linear in circuit's depth. The protocol has been modernized multiple times improving time-complexity and applying overall simplifications to the algorithm. The most popular version of the protocol was proposed by Thaler (2015), where they improve on the protocol complexity by considering less general, but more practical scenario of binary finite fields, which allowed for many simplifications. In current schemes, GKR-style protocols are used to generate proofs of primitive operations, such as gradient descent. These proofs are then combined using the recursive proof compositions.

Valiant (2008) proposed one of the first recursive proof composition schemes designed to combine multiple proofs into one. A prover in the scheme uses time linear in the time of a classical prover and space polynomial in the space of a classical prover, while the verifier's time and space are constant. This approach is known as an "incrementally verifiable computation." Another construction has originated from the works of Chiesa and Tromer (2010) in the form of

"proof-carrying data". In this framework it is possible to encode certain data properties and propagate them through a chain of computations effectively without increasing time complexity. Both frameworks were developed and improved upon in future works. Such recursive constructions are now most often used to combine gradient descent proofs into a proof of one iteration. These proofs are later combined into proofs of training for machine learning epochs.

These advancements have led to the development of two previously mentioned notions: proof-of-inference and proof-of-training. As a direct response to the problem of trustworthiness, proof-of-inference strives to convince the verifier that the model output is indeed correct. One of the first results in this area goes back to Ghodsi et al. (2017). The proposed framework uses arithmetic circuit representation for a machine learning model and an interactive protocol system based on the GKR protocol. Later results are due to Kang et al. (2022), where the authors leverage the power of zk-SNARKs — succinct non-interactive arguments of knowledge, which gained a lot of attention in the recent years, mostly because of their universal construction, allowing for constructions of proofs independent of the problem at hand. Such approach allows for strong cryptographic properties while lacking performance and execution times. From these developments, another more convoluted notion of proof-of-training has emerged.

The notion of proof-of-training was first formally introduced by Hengrui et al. (2021). Originally called "proof-of-learning", it is roughly defined as a proof that a model was trained correctly, i.e. using a publicly known machine learning algorithm and public specifications, e.g. batch size and model architecture, and using a specific dataset. The idea was later developed by Abbaszadeh et al. (2024) and took a shape of a practical solution, called Kaizen, capable of handling large models, namely VGG-11, with 10 million parameters. This framework makes extensive use of GKR-style protocols and recursive proof constructions.

The prospects of using metamodels, e.g. gradient boosting, are however yet to be researched. Both proof-of-inference and proof-of-training papers only consider the case of "basic" models, where training process is aimed at the minimization of the loss function, leaving unexplored some of the most used production techniques.

REFERENCES:

1. Xing Z. et al. (2023). Zero-knowledge Proof Meets Machine Learning in Verifiability: A Survey. *arXiv preprint arXiv:2310.14848*.
2. Goldwasser S. et al. (2008). Delegating Computation: Interactive Proofs for Muggles. *ACM Symposium on Theory of Computing*, 113–122.
3. Thaler J. (2015). A Note on the GKR Protocol. *URL: https://people.cs.georgetown.edu/jthaler/GKRNote.pdf*.
4. Valiant P. (2008). Incrementally Verifiable Computation or Proofs of Knowledge Imply Time/Space Efficiency. *Canetti, R. (eds) Theory of Cryptography, TCC 2008, Lecture Notes in Computer Science, vol 4948, 1–18*.
5. Chisea A. & Tromer E. (2010). Proof-Carrying Data and Hearsay Arguments from Signature Cards. *Proceedings of the Symposium on Innovations in Computer Science, 310–331*.
6. Ghodsi Z. et. al. (2017). SafetyNets: verifiable execution of deep neural networks on an untrusted cloud. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 4675–4684*.
7. Kang D. et. al. (2024) Scaling up Trustless DNN Inference with Zero-Knowledge Proofs. *arXiv preprint arXiv:2210.08674*.
8. Hengrui J. et. al. (2021). Proof-of-Learning: Definitions and Practice. *2021 IEEE Symposium on Security and Privacy (SP), 1039–1056*.
9. Abbaszadeh K. et. al. (2024). Zero-Knowledge Proofs of Training for Deep Neural Networks. *Cryptology ePrint Archive*.