Research Project Name: Project Large zkML

Improving Efficiency in Zero-Knowledge Proofs for LLM Inference Verification

Zero-Knowledge Proof is a cryptographic technique that enables one party, referred to as the prover, to demonstrate to another party, known as the verifier, that a particular statement is true without disclosing any additional information about the statement. This means the verifier can confirm the validity of the claim without learning any other details beyond the fact that the claim is true. Zero-Knowledge Proofs, which are based on arithmetic circuits, are widely used in blockchain applications due to their ability to verify computational integrity and ensure privacy-preservation.

Recently, researchers have begun exploring the potential of Zero-Knowledge Proofs within the field of machine learning. This emerging field, referred to as Zero-Knowledge Machine Learning (zkML), involves the application of zero-knowledge proofs to verify the correctness of machine learning model inferences. The growing interest in zkML is largely due to its ability to provide assurances to regulators and other stakeholders that a specific, authorized model is being used for an application, all while maintaining the confidentiality of the model's parameters.

In zkML, the parameters of a machine learning model are bound using a commitment scheme, which ensures that they cannot be altered after commitment. A zero-knowledge proof of the model's inference is then generated, which allows verifiers, such as regulators, to confirm that the inference was performed correctly and with the authorized model. This process prevents unauthorized modifications to the model and guarantees accountability in applications requiring high levels of trust.

In the context of blockchain applications, zkML can be implemented in two main ways: on-chain and off-chain. For instance, zkML can verify the inference of a model used to detect fraudulent transactions off-chain. Once the inference is completed, a proof of the inference can be generated and validated on-chain, ensuring that the process adheres to privacy requirements by concealing both the data and the model parameters. Alternatively, zkML can also be applied entirely on-chain. One example is its use in voter eligibility detection systems, where zkML ensures the correctness of the inference process while keeping sensitive details private.

EZKL, a prominent zkML framework, has been instrumental in advancing the field. It has successfully demonstrated the verification of inferences for various machine learning models, including classical algorithms such as tree-based and gradient-boosting models, as well as more complex deep neural networks like LSTMs, GANs, and autoencoders. Beyond these, researchers have developed techniques to adapt ZKPs for convolutional neural networks and other deep learning architectures, broadening the scope of zkML applications.

One notable recent development in zkML focuses on applying zero-knowledge proofs and lookup tables to large language models (LLMs). Traditional applications of ZKPs in machine learning have primarily addressed operations such as matrix multiplication and activation functions like ReLU. However, LLMs introduce unique challenges due to their architectural complexity and reliance on specialized activation functions like SwiGLU and softmax. Unlike traditional models

that employ softmax only at the output layer, LLMs integrate it throughout their architecture, resulting in significantly higher computational overhead in proof generation and verification.

The inaugural zkLLM paper marked a significant milestone by addressing these challenges, reducing the computational burden associated with proof generation and verification for LLMs. However, the scope of the paper was limited to smaller LLaMA 2 models (7B and 13B), which use traditional self-attention mechanisms. Larger models, such as LLaMA 2 70B and the LLaMA 3 Herd of models, which implement Grouped Query Attention, were not explored, leaving room for further advancements.

To address these challenges and enhance the efficiency of zkML systems, additional techniques can be employed. One such technique is the use of folding schemes, which aggregate multiple statements into a single proof. This approach enables the verification of the aggregated proof to imply the verification of each individual computation. By using folding schemes, the verification process can be streamlined, significantly reducing both proof generation and verification time. For example, instead of verifying each layer of an LLM—such as the RMS Normalization, FeedForward, and Attention layers—individually, a folding scheme can consolidate these computations into a single proof, resulting in drastic efficiency improvements.

Another promising approach involves converting models into ternary networks, which introduce sparsity into the model weights and reduce computational complexity. A ternary network converts model weights into three values: +1, -1, and 0. This sparsity eliminates the need for multiplication operations, which are computationally expensive, replacing them with simpler addition and subtraction operations. For LLMs, this conversion can lead to significant reductions in computational overhead while maintaining model performance consequently reducing the time required to generate and verify ZK proofs.

By integrating folding schemes and ternary networks, zkML frameworks can achieve substantial efficiency gain, which is crucial for handling powerful large language models with billions of parameters.