

Hardware-Aware Neural Network Optimization for Secure Two-Party Inference Using Lightweight Activation Functions

Jarwadi Suartini

Department of Computer Science, University of Houston, Houston, TX, USA.
jarwadi.suartini438@uh.edu

Itziar Segura

Department of Computer Science, Binghamton University, Binghamton, NY, USA.
itziar.segura@binghamton.edu

Abstract

Secure two-party inference has emerged as a critical capability for organizations that need to evaluate neural networks over sensitive inputs without revealing either the model or the data to the other participant. Although cryptographic protocols provide strong formal guarantees, their practical deployment remains constrained by communication overhead, latency, hardware heterogeneity, and the high cost of nonlinear activation functions. This paper examines hardware-aware neural network optimization for secure two-party inference, with particular attention to lightweight activation functions as a structural design lever. The analysis treats private inference not simply as a cryptographic problem but as a socio-technical infrastructure problem involving model architecture, accelerator selection, network topology, deployment governance, and long-term operational sustainability. We synthesize insights from secure computation, efficient deep learning, hardware-aware architecture search, and privacy-preserving machine learning systems. The discussion shows that activation function choice interacts with protocol design, memory hierarchy, batching strategy, and threat models in ways that cannot be resolved through isolated algorithmic improvements. Lightweight activation functions can reduce the number of expensive nonlinear operations, but they may also alter model accuracy, fairness, robustness, and auditability. The paper argues that sustainable private inference requires co-design across layers: model designers should treat activation functions as both accuracy parameters and cryptographic cost parameters; system architects should align protocols with hardware capabilities; and policymakers should require transparency about residual leakage, failure modes, and equitable access. The conclusion outlines a research agenda for hardware-aware, activation-conscious private inference that balances performance, privacy, fairness, and institutional accountability.

Keywords

secure two-party computation; private inference; hardware-aware optimization; lightweight activation functions; privacy-preserving machine learning; neural architecture search; deployment governance; socio-technical systems.

1. Introduction

Neural networks have become foundational to applications ranging from medical diagnosis and financial risk assessment to industrial monitoring and public-sector decision support [1], [2]. Their increasing scale has been matched by growing concern about the sensitivity of both

model parameters and inference inputs. In many settings, a service provider holds a proprietary model while a client holds personal or regulated data. Secure two-party inference allows the client to obtain a prediction without revealing its input, and allows the provider to protect model weights without exposing them to the client. Cryptographic protocols based on secure computation offer formal privacy guarantees [4], [5], but their deployment remains costly because every nonlinear operation, communication round, and memory access can become a bottleneck.

The practical difficulty is especially acute for activation functions. Modern neural networks rely on nonlinearities such as rectified linear units, sigmoid functions, and hyperbolic tangent functions to learn complex representations. In secure two-party computation, these nonlinearities often require garbled circuits, oblivious transfer, function secret sharing, or other interactive primitives. Linear layers can be accelerated through secret sharing and matrix multiplication optimizations, but activation functions frequently dominate latency and communication [6], [7], [8]. A model that is efficient in plaintext may therefore be inefficient, brittle, or economically unsustainable once placed inside a privacy-preserving protocol.

This paper focuses on hardware-aware neural network optimization for secure two-party inference using lightweight activation functions. The central argument is that activation function selection should not be treated as a minor accuracy trade-off. It is a structural decision that shapes protocol selection, hardware utilization, deployment cost, energy consumption, and governance obligations. Hardware-aware optimization seeks to align network architecture with the actual capabilities of processors, accelerators, memory systems, and network links. Lightweight activation functions reduce cryptographic burden, but they also raise questions about representational capacity, fairness, robustness, and auditability.

The paper contributes a system-level synthesis rather than a narrow algorithmic proposal. It reviews the private inference landscape, analyzes hardware-aware optimization strategies, and examines lightweight activation functions as a cross-layer design lever. It then considers deployment infrastructure, operational sustainability, fairness, robustness, and policy implications. The final sections outline forward-looking directions and conclude with a research agenda that treats private inference as an infrastructural capability rather than a standalone cryptographic primitive.

2. Background and Related Work

Secure two-party computation originated in foundational work on secure function evaluation and general protocols for joint computation without trusted third parties [4], [5]. These theoretical advances established that arbitrary functions could be computed privately, but early constructions were far too slow for large neural networks. The rise of practical privacy-preserving machine learning changed this trajectory. SecureML demonstrated that linear algebra and stochastic gradient descent could be adapted to secret-sharing-based protocols, enabling scalable training and inference in specific settings [6]. MiniONN showed that neural network inference could be transformed into oblivious primitives, reducing interaction for certain model structures [7]. DeepSecure pursued a different balance by combining garbled circuits with optimized neural network evaluation, illustrating the importance of architecture-aware protocol design [8].

Subsequent systems expanded the performance frontier. CrypTFlow2 advanced two-party inference with optimized protocols for comparison, multiplication, and nonlinear evaluation [10]. Delphi explored cryptographic inference as a service, mixing preprocessing and online

phases to reduce latency [11]. BLAZE introduced faster privacy-preserving machine learning through improved garbled circuit and secret-sharing techniques [12]. CrypTen provided a software framework that made secure multi-party computation more accessible to machine learning practitioners, emphasizing usability and integration with tensor operations [13]. In parallel, fully homomorphic encryption systems such as CHET and TFHE explored compilation and encryption schemes that allow computation over encrypted data without interaction, albeit with different performance profiles [14], [15]. These works collectively show that private inference is not a single technology but a family of trade-offs among communication, computation, trust assumptions, and hardware requirements.

Hardware-aware neural network optimization has also matured independently. Efficient processing of deep neural networks depends on data reuse, memory bandwidth, arithmetic intensity, and accelerator characteristics [3]. Techniques such as pruning, quantization, and compact architecture design reduce model size and energy consumption [17]. Neural architecture search and once-for-all networks allow a single model family to be specialized for different deployment targets [16]. TAPAS explored tricks to accelerate encrypted prediction, highlighting how activation approximations and polynomial replacements can shift the cost profile of private inference [18]. More recently, ReLU-reduction strategies have been proposed to make two-party private inference more practical by reducing the number of expensive nonlinear operations while preserving accuracy [9]. This line of work is directly relevant because it treats activation functions as a first-class optimization target rather than an immutable component of the network.

The gap addressed by this paper is the absence of a fully integrated socio-technical account. Many studies optimize a protocol or a network in isolation. Fewer examine how hardware-aware architecture search, lightweight activation functions, deployment infrastructure, fairness constraints, and policy governance interact over time. A model that is efficient on one accelerator may be inefficient on another. A protocol that is fast in a laboratory may be impractical across geographic regions with variable bandwidth. An activation function that reduces cryptographic cost may also reduce interpretability or alter error distribution across demographic groups. These are not secondary concerns; they determine whether private inference becomes a sustainable public infrastructure or remains a niche service for well-resourced actors.

3. Hardware-Aware Neural Network Optimization for Secure Two-Party Inference

Hardware-aware optimization begins with the recognition that neural network computation is shaped by physical constraints. Processors differ in their support for vectorized operations, cryptographic instructions, memory bandwidth, and parallel execution. Graphics processing units excel at dense matrix multiplication but may perform poorly on irregular control flow and bit-level operations common in garbled circuits. Central processing units may handle branching and cryptographic primitives more flexibly but lack the throughput needed for large models. Specialized accelerators can improve specific operations, but they may not support the arbitrary precision or data-dependent behavior required by secure protocols. Consequently, a private inference system must be optimized for a target deployment profile rather than for an abstract computational graph.

In secure two-party inference, linear layers are often implemented through secret sharing, homomorphic operations, or oblivious matrix multiplication. Nonlinear layers are more challenging because they introduce comparisons, branching, or lookup operations. Hardware-aware optimization therefore seeks to reduce the frequency and complexity of nonlinear

operations while preserving model quality. This can involve architectural changes such as reducing depth, replacing activation functions, or introducing polynomial approximations. It can also involve scheduling changes such as batching, fusion, and pipelining to hide communication latency. The key insight is that the optimal network for private inference may differ from the optimal network for plaintext inference, even when both achieve similar accuracy.

A second dimension is communication topology. Two-party inference often assumes a client and server connected by a network, but real deployments may involve edge devices, regional data centers, or hybrid cloud environments. Hardware-aware design must account for round-trip latency, bandwidth variability, packet loss, and energy constraints on mobile devices. A protocol that minimizes computation but requires many communication rounds may be inferior to one that performs more local computation but reduces interaction. Similarly, a protocol that requires large preprocessing tables may shift cost to off-peak periods or trusted hardware, creating new operational dependencies. These trade-offs are structural and cannot be resolved by optimizing a single metric such as multiply-accumulate count.

A third dimension is model specialization. Once-for-all networks and related approaches allow a single supernet to be specialized for different latency, energy, and memory targets [16]. In the private inference context, specialization could produce model variants that differ in activation function type, depth, width, and quantization level depending on the client's hardware and privacy requirements. A mobile client might use a shallower network with lightweight activations to reduce communication, while a data center client might use a deeper network with more accurate but costlier nonlinearities. This kind of portfolio strategy is promising, but it complicates model governance. Multiple variants must be validated, monitored, and audited, and the provider must ensure that privacy guarantees remain consistent across variants.

Hardware-aware optimization also intersects with cryptographic parameter selection. Larger security parameters increase robustness but also increase computation and communication. Hardware acceleration can partially offset this cost, but the relationship is nonlinear. For example, cryptographic operations that benefit from specialized instructions may be fast on one server but unavailable on another. A system designed for heterogeneous deployment must therefore either standardize on a common hardware profile or provide fallback paths that preserve privacy while accepting higher latency. This is a governance decision as much as an engineering one, because it determines who can participate in the private inference ecosystem and under what conditions.

4. Lightweight Activation Functions and Secure Computation

Activation functions are the primary source of nonlinearity in neural networks. In plaintext inference, functions such as rectified linear units are computationally cheap and well supported by hardware. In secure two-party computation, however, the same functions can become expensive because they require secure comparison, oblivious selection, or interactive protocols. A rectified linear unit essentially asks whether a secret value is positive and then returns either the value or zero. That comparison is trivial in plaintext but costly under encryption or secret sharing. Lightweight activation functions aim to reduce this cost by using operations that are more compatible with secure computation, such as low-degree polynomials, piecewise linear approximations, or simpler threshold functions.

The appeal of lightweight activations is straightforward. If a network uses fewer nonlinear operations, the private inference protocol requires fewer communication rounds and less cryptographic work. This can reduce latency, energy consumption, and cost. In some cases, replacing a rectified linear unit with a polynomial approximation allows the entire computation to be expressed in terms of additions and multiplications, which are more efficiently supported by secret-sharing and homomorphic encryption schemes [14], [15]. In other cases, reducing the number of activation layers or the number of neurons affected by nonlinearities can yield substantial savings without severe accuracy loss [9]. Hardware-aware design can then match the remaining nonlinearities to available accelerators, such as cryptographic coprocessors or vector units.

However, lightweight activation functions are not a free lunch. They alter the inductive bias of the network. A polynomial approximation may smooth decision boundaries in ways that affect robustness to adversarial examples or distribution shift. A piecewise linear approximation may introduce discontinuities that are difficult to audit. A simplified threshold function may reduce expressive power and require additional layers or width to compensate, potentially increasing linear computation and memory usage. These effects are not uniform across tasks. Image classification, natural language processing, fraud detection, and medical risk prediction have different sensitivities to activation choice. A design that works well for one domain may fail in another.

From a secure computation perspective, the cost of an activation function depends on the protocol. Garbled circuits can evaluate arbitrary Boolean functions but incur communication proportional to circuit size. Secret-sharing protocols can evaluate additions and multiplications efficiently but require interaction for comparisons. Homomorphic encryption can evaluate polynomial functions but struggles with branching and non-polynomial operations. Function secret sharing can handle comparisons and piecewise functions but may require preprocessing and key management. Therefore, a lightweight activation function is not universally lightweight; it is lightweight relative to a specific protocol and hardware environment. This relativity means that hardware-aware optimization must be co-designed with protocol selection.

A further concern is numerical stability. Secure computation often operates over finite fields or fixed-point representations, which can introduce rounding errors and overflow risks. Polynomial approximations of activation functions may behave poorly outside the range seen during training. If the private inference system encounters an out-of-distribution input, the approximation may produce large errors or even fail catastrophically. Robustness testing must therefore include adversarial and edge-case inputs, not only standard validation sets. The system should also include runtime checks or fallback mechanisms, although these can introduce additional leakage or complexity.

Fairness is another dimension. Activation functions can influence how models treat different subgroups, especially when data distributions are skewed or when model capacity is constrained. A lightweight activation that reduces accuracy overall may reduce it unevenly, widening disparities for groups that are already underrepresented. Because private inference is often justified as a privacy-enhancing technology for sensitive data, failures of fairness can undermine public trust. Governance frameworks should therefore require disaggregated evaluation of private inference models, even when the underlying data cannot be shared. Secure computation can support aggregate fairness metrics, but the design of those metrics and the interpretation of results remain institutional responsibilities.

5. System Architecture, Deployment, and Infrastructure

A private inference system is more than a protocol and a model. It is an architecture that includes client devices, server infrastructure, cryptographic key management, preprocessing services, network links, monitoring tools, and update mechanisms. Hardware-aware optimization must address each layer. At the client edge, limited memory and battery life may favor lightweight models and infrequent communication. At the server edge, high-throughput accelerators may favor batched requests and parallel cryptographic operations. Between them, the network may impose latency and bandwidth constraints that dominate overall performance. A sustainable deployment must therefore balance local computation, communication, and centralized processing.

Preprocessing is a particularly important architectural component. Many secure two-party protocols require correlated randomness, oblivious transfer extensions, or garbled tables before online inference begins. This preprocessing can be performed on trusted hardware, by a third party, or through expensive cryptographic protocols. It shifts cost away from the online phase but creates new dependencies. If preprocessing is centralized, it may become a single point of failure or a target for denial-of-service attacks. If it is distributed, it may increase coordination overhead. Hardware-aware design can optimize preprocessing for specific accelerators, but governance must ensure that preprocessing does not undermine the privacy guarantees of the overall system.

Deployment also involves model updates. Neural networks are frequently retrained or fine-tuned to address drift, improve accuracy, or patch vulnerabilities. In private inference, model updates must preserve compatibility with cryptographic protocols and hardware profiles. A new activation function may require new protocol parameters or new accelerator support. A new layer may increase communication rounds beyond acceptable thresholds. These constraints suggest that private inference systems should be designed with versioned interfaces, backward compatibility, and staged rollouts. The alternative is a brittle deployment that cannot evolve without breaking privacy or performance guarantees.

Operational sustainability is closely related to energy and cost. Secure computation is computationally intensive, and its energy footprint can be substantial when scaled across millions of requests. Hardware-aware optimization can reduce energy per inference, but it may also increase model complexity or require specialized infrastructure that is itself energy-intensive to manufacture and maintain. A holistic sustainability assessment should consider embodied energy, data center cooling, network transmission, and end-device battery consumption. Lightweight activation functions can contribute to sustainability by reducing computation, but they must be evaluated alongside accuracy and fairness. A system that is energy-efficient but biased or unreliable is not sustainable in a broader sense.

Infrastructure resilience is another concern. Private inference services may be used in healthcare, finance, government, and critical infrastructure, where downtime or degraded performance can have serious consequences. Redundancy, failover, and graceful degradation must be designed into the architecture. In some cases, graceful degradation may mean falling back to a less private but still acceptable mode, but this must be transparent to users and regulators. In other cases, the system should refuse to operate rather than weaken privacy guarantees. These choices are not purely technical; they reflect institutional risk tolerance and legal obligations.

6. Governance, Fairness, Robustness, and Sustainability

Governance of private inference must address the entire lifecycle of the system. This includes model design, data collection, training, cryptographic protocol selection, deployment, monitoring, and decommissioning. Hardware-aware optimization and lightweight activation functions affect each stage. If activation functions are chosen primarily to reduce cryptographic cost, then model quality, fairness, and robustness may be underweighted. Governance frameworks should require documentation of these trade-offs in model cards, system cards, or equivalent artifacts. Such documentation should be understandable to auditors, regulators, and affected communities, not only to engineers.

Fairness is especially challenging because private inference protects data in ways that can also obscure disparities. If the client's data is never revealed, how can the system detect discriminatory outcomes? One approach is to use secure computation to compute aggregate fairness metrics without revealing individual records. Another is to conduct pre-deployment audits on representative data and then monitor aggregate outcomes in production. Both approaches have limitations. Aggregate metrics may hide intersectional harms, and pre-deployment audits may not capture distribution shift. A robust governance regime should combine multiple methods and require periodic re-evaluation, especially when activation functions or architectures change.

Robustness is similarly multi-dimensional. Private inference systems must resist adversarial inputs, protocol attacks, side-channel leakage, and infrastructure failures. Lightweight activation functions may alter the adversarial surface of the model. Smooth polynomial approximations can be more susceptible to certain gradient-based attacks, while discontinuous functions can create unstable decision boundaries. Hardware-aware optimization may introduce timing side channels if different inputs trigger different execution paths. Constant-time implementations and oblivious access patterns are therefore essential. These requirements increase engineering complexity and cost, but they are necessary for systems that handle sensitive data.

Sustainability should be understood in environmental, economic, and institutional terms. Environmental sustainability requires minimizing energy and material consumption across the lifecycle. Economic sustainability requires that private inference be affordable for smaller organizations, public agencies, and individual users, not only for large technology firms. Institutional sustainability requires that the system remain trustworthy over time, with clear accountability, transparent updates, and mechanisms for redress. Lightweight activation functions can support sustainability by reducing computational burden, but they must not become a shortcut that sacrifices accuracy or fairness for cost reduction.

Policy implications follow from these considerations. Policymakers should avoid treating private inference as a purely technical solution to privacy problems. It is a socio-technical capability that depends on standards, procurement rules, competition policy, and digital infrastructure. Governments can support research into hardware-aware secure computation, fund public-interest deployments, and require transparency about residual leakage and failure modes. They can also promote interoperability so that private inference services are not locked into proprietary hardware or protocols. At the same time, they must recognize that no system provides absolute privacy. Risk assessments should be honest about what is protected, what is revealed, and under what conditions.

7. Policy Implications and Forward-Looking Perspectives

The future of private inference will likely involve greater specialization. Rather than one protocol or one model architecture, deployments may use portfolios of models and protocols tailored to different devices, tasks, and privacy requirements. Hardware-aware neural architecture search can automate part of this specialization, but it must be constrained by fairness, robustness, and governance objectives. Lightweight activation functions will remain an important lever, but their selection should be based on empirical evaluation across domains and populations. Research should therefore move beyond single-benchmark comparisons toward multi-objective optimization that includes accuracy, latency, energy, fairness, and auditability.

Another promising direction is hybrid computation. Some layers may be evaluated with homomorphic encryption, others with secret sharing, and still others with garbled circuits. Hardware accelerators may be assigned to the operations they handle best. This heterogeneity can improve performance, but it complicates security proofs and system verification. Formal methods and automated verification could help ensure that hybrid protocols preserve privacy and correctness. Standardized interfaces could make it easier to compose components from different vendors without sacrificing trust.

A third direction is transparency-preserving privacy. Private inference protects data and models, but it can also reduce accountability if no one can inspect the computation. Techniques such as zero-knowledge proofs, secure logging, and confidential computing may allow selective disclosure to auditors without revealing sensitive inputs. These techniques are not mature, and they add overhead. Nevertheless, they point toward a future in which privacy and accountability are balanced rather than treated as opposites. Lightweight activation functions and hardware-aware optimization can create the performance headroom needed to support such mechanisms.

Finally, private inference should be evaluated as public infrastructure. If it becomes essential for healthcare, finance, or government services, then access, reliability, and fairness become policy concerns. Infrastructure should be resilient, interoperable, and governed by clear rules. It should not depend on a single vendor or a single hardware ecosystem. It should be sustainable in energy and cost. And it should be accountable to the people whose data it protects. These goals are ambitious, but they provide a coherent direction for research and practice.

8. Conclusion

This paper has examined hardware-aware neural network optimization for secure two-party inference using lightweight activation functions. The central finding is that activation function choice is a cross-layer design decision that shapes cryptographic cost, hardware utilization, model accuracy, fairness, robustness, and governance. Lightweight activation functions can reduce the burden of nonlinear operations in secure computation, but they are not universally beneficial. Their value depends on the protocol, the hardware, the task, and the population being served. Hardware-aware optimization must therefore be integrated with model design, system architecture, and institutional governance.

The paper has argued for a socio-technical approach to private inference. Secure two-party computation provides strong cryptographic guarantees, but practical deployment depends on infrastructure, energy, policy, and trust. A system that is fast and private but biased, fragile, or exclusionary is not a successful system. A system that is fair and robust but too costly to operate at scale is also incomplete. The research agenda ahead should focus on multi-

objective design, hybrid protocols, transparent auditability, and public-interest deployment. Lightweight activation functions will remain one important tool in this agenda, but they must be evaluated as part of a larger architecture of responsibility. Only then can private inference become a sustainable and equitable capability for sensitive machine learning.

References

1. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1097-1105.
2. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>
3. Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12), 2295-2329. <https://doi.org/10.1109/JPROC.2017.2761740>
4. Yao, A. C. (1986). How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science* (pp. 162-167). IEEE. <https://doi.org/10.1109/SFCS.1986.25>
5. Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play any mental game. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing* (pp. 218-229). ACM. <https://doi.org/10.1145/28395.28420>
6. Mohassel, P., & Zhang, Y. (2017). SecureML: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy* (pp. 19-38). IEEE. <https://doi.org/10.1109/SP.2017.12>
7. Liu, J., Juuti, M., Lu, Y., & Asokan, N. (2017). Oblivious neural network predictions via MiniONN transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 619-631). ACM. <https://doi.org/10.1145/3133956.3134056>
8. Rouhani, B. D., Riazi, M. S., & Koushanfar, F. (2018). DeepSecure: Scalable provably-secure deep learning. In *Proceedings of the 55th Annual Design Automation Conference* (pp. 1-6). ACM. <https://doi.org/10.1145/3195970.3196023>
9. Peng, H., Huang, S., Zhou, T., Luo, Y., Wang, C., Wang, Z., Zhao, J., Xie, X., Li, A., Geng, T., Mahmood, K., Ding, C., & Wen, W. (2023). Rrnet: Towards relu-reduced neural network for two-party computation based private inference. *arXiv*. <https://arxiv.org/abs/2302.02292>
10. Rathee, D., Rathee, M., Kumar, N., Chandran, N., Gupta, D., Rastogi, A., & Sharma, R. (2020). CryptFlow2: Practical 2-party secure inference. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (pp. 325-342). ACM. <https://doi.org/10.1145/3372297.3417274>
11. Mishra, P., Lehmkuhl, R., Srinivasan, A., Zheng, W., & Popa, R. A. (2020). Delphi: A cryptographic inference service for neural networks. In *Proceedings of the 29th USENIX Security Symposium* (pp. 2505-2522). USENIX Association.

12. Patra, A., Schneider, T., Suresh, A., & Yalame, H. (2020). BLAZE: Blazing fast privacy-preserving machine learning. In Proceedings of the 2020 Network and Distributed System Security Symposium. Internet Society. <https://doi.org/10.14722/ndss.2020.24098>
13. Knott, B., Venkataraman, S., Hannun, A., Sengupta, S., & van der Maaten, L. (2021). CrypTen: Secure multi-party computation meets machine learning. In Advances in Neural Information Processing Systems 34 (pp. 4961-4973).
14. Dathathri, R., Saarikivi, O., Chen, H., Laine, K., Lauter, K., Maleki, S., & Musuvathi, M. (2019). CHET: An optimizing compiler for fully-homomorphic neural-network inferencing. In Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (pp. 142-156). ACM. <https://doi.org/10.1145/3314221.3314628>
15. Chillotti, I., Gama, N., Georgieva, M., & Izabachène, M. (2020). TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33(1), 34-91. <https://doi.org/10.1007/s00145-019-09319-x>
16. Cai, H., Gan, C., Wang, T., Zhang, Z., & Han, S. (2020). Once-for-all: Train one network and specialize it for efficient deployment. In International Conference on Learning Representations.
17. Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In International Conference on Learning Representations.
18. Sanyal, A., Kusner, M. J., Gascón, A., & Kanade, V. (2018). TAPAS: Tricks to accelerate (encrypted) prediction as a service. In Proceedings of the 35th International Conference on Machine Learning (pp. 4490-4499). PMLR.