A survey of hardware error fault tolerance for deep learning

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This literature review provides an extensive survey of hardware error fault tolerance strategies in deep learning systems. As deep learning applications become increasingly integral to critical domains, the reliability of the supporting hardware is paramount. Hardware faults, arising from various sources such as manufacturing defects and environmental factors, pose significant risks to the accuracy and safety of deep learning computations. This survey encapsulates the latest research efforts aimed at evaluating and enhancing the fault tolerance of hardware utilized in deep learning tasks. It encompasses a range of topics including training methodologies for fault resilience, hardware-software co-design for error mitigation, and advanced techniques for fault detection and correction. The review also addresses the assessment of both transient and permanent hardware faults and discusses innovative frameworks that contribute to error resilience, such as clipped activation functions and sensitivity analysis. Concluding with a discussion on the broader challenges and future directions, this survey highlights the critical need for comprehensive testing methodologies, the implications of storage media errors, and the overarching requirement for robust machine learning systems that can withstand hardware imperfections. The synthesis of current knowledge and identification of research gaps presented herein aim to guide the development of dependable deep learning systems that maintain high performance standards despite hardware vulnerabilities.**

***Index Terms*—**

I. Introduction

The proliferation of deep learning (DL) in a wide array of applications, from autonomous vehicles to medical diagnostics, has underscored the critical need for reliable and robust hardware systems. As deep learning models become increasingly complex and pervasive, the underlying hardware that supports these computations faces a growing risk of faults and errors. These hardware errors can stem from a variety of sources, including manufacturing defects, environmental stress, aging, and radiation-induced events. The consequences of such errors can be catastrophic, particularly in safety-critical systems where even a single miscalculation can lead to dire outcomes.

In light of these challenges, the field of hardware error fault tolerance for deep learning has emerged as a vital area of research. This literature survey aims to provide a comprehensive overview of the state-of-the-art strategies and methodologies developed to enhance the fault tolerance of hardware systems running deep learning algorithms. The focus is on both the evaluation of hardware-related fault tolerance and the design of fault-tolerant systems, with the ultimate goal of mitigating computational errors caused by hardware circuit faults.

The survey is structured into several key sections, each addressing different aspects of hardware fault tolerance in deep learning. We begin by examining training and retraining strategies that aim to imbue deep learning models with resilience to hardware faults. This is followed by an exploration of hardware-software co-design approaches that seek to create synergies between DL algorithms and hardware design for improved error resilience. Subsequent sections delve into techniques for fault detection, prediction, and correction, as well as the evaluation and mitigation of both permanent and transient hardware faults.

Advanced techniques and frameworks that contribute to fault tolerance, such as clipped activation and sensitivity-based error resilience, are also discussed. Finally, the survey concludes with a broader perspective on the challenges and opportunities in building reliable deep learning systems, highlighting the importance of testing methodologies, the impact of storage media errors, and the need for robust machine learning systems that encompass hardware fault tolerance.

Through this literature survey, we aim to synthesize the collective knowledge in the field, identify gaps in current research, and suggest directions for future investigation. The ultimate objective is to contribute to the development of deep learning systems that are not only intelligent and efficient

Identify applicable funding agency here. If none, delete this.

but also resilient and dependable in the face of hardware imperfections.

II. Training and Retraining Strategies for Fault Tolerance

This section explores various training and retraining methodologies designed to enhance the fault tolerance of deep learning models against hardware errors. We will discuss fault-aware training, incremental network approximation, and retraining-based timing error mitigation techniques.

A. Fault-Aware Training Approaches

Fault-aware training (FAT) is a novel technique that incorporates error modeling during the training phase of neural networks to make them resilient to specific hardware fault models. The work by Zahid et al. [1] presents a comprehensive study on FAT, where faults are injected into the convolutional layers during the training of quantized neural networks (QNNs). This approach results in networks that are not only accurate but also exhibit a higher tolerance to errors. The authors demonstrate that QNNs trained with FAT can achieve higher worst-case accuracy at a lower hardware cost, which is validated across several classification tasks. This methodology is particularly relevant for safety-critical applications where functional safety is paramount, and it represents a significant step towards reducing the hardware cost associated with achieving such safety.

B. Incremental Network Approximation for Fault Tolerance

The concept of approximate computing has been introduced as a means to improve the hardware efficiency of deep neural networks (DNNs), especially in fault-tolerant applications. Liu et al. [2] propose an Incremental Network Approximation (INA) algorithm that addresses the convergence problem in re-training phases when applying highly approximate arithmetic operations, such as multipliers, to DNNs. The INA algorithm allows for a balance between accuracy and implementation cost, with experiments showing that models re-trained with INA can achieve up to 80% hardware reduction with less than 2% degradation in classification accuracy. This hardware-software co-design algorithm is a testament to the potential of INA in enhancing the fault tolerance of DNNs, making it a valuable addition to the repertoire of strategies for dealing with hardware errors.

C. Retraining-Based Timing Error Mitigation

Timing errors, caused by delay faults, process variations, and aging, pose a significant threat to the reliability of neural network accelerators, especially under nanoscale manufacturing processes. Deng et al. [3] explore the inherent error resiliency of neural networks and propose a retraining-based approach to mitigate timing errors in NN accelerators. By retraining the accelerators to update their weights, the method effectively circumvents critical timing errors, ensuring that the output results are not significantly affected. This strategy leverages the adaptive nature of neural networks and provides a practical solution for maintaining the reliability of hardware neural networks in the face of timing-related faults.

In addition to these specific strategies, Chiu's work [4] on training techniques for fault tolerance in neural networks provides a broader perspective on the subject. By coercing weights to lower magnitudes, adding artificial faults during training, and dynamically adjusting the network architecture, Chiu et al. demonstrate that neural networks can achieve enhanced robustness and fault tolerance. These methods offer valuable insights into the design of training strategies that can lead to more fault-tolerant deep learning models.

In summary, the strategies discussed in this section highlight the importance of training and retraining methodologies in achieving fault tolerance for deep learning models. By considering the inherent features of DNNs and the specific challenges posed by hardware faults, researchers can develop more resilient and efficient systems that are capable of operating reliably in safety-critical and resource-constrained environments.

III. Hardware-Software Co-Design for Error Resilience

In this section, we delve into the intersection of hardware and software design to achieve fault tolerance in deep learning systems. We will examine co-design algorithms, predictive design paradigms, and the role of algorithm-hardware co-design in resilient deep learning inference. The goal is to explore how these integrated approaches can mitigate the impact of hardware errors on deep learning computations, ensuring both efficiency and reliability.

A. Co-Design Algorithms for Fault Tolerance

One of the key strategies in achieving fault tolerance is through the development of co-design algorithms that can adapt to the limitations of hardware. Liu et al. [2] introduced the Incremental Network Approximation (INA) algorithm, which is a hardware-software co-design approach aimed at improving the fault tolerance of Deep Neural Networks (DNNs). INA addresses the convergence problem in re-training phase and allows for significant hardware reduction while maintaining acceptable accuracy levels. This method exemplifies how co-design algorithms can be tailored to balance the trade-offs between accuracy and implementation cost, making it a vital reference for understanding the potential of co-design in enhancing fault tolerance.

B. Predictive Design for Timing Error Resilience

The predictive design paradigm is another innovative approach that focuses on anticipating and mitigating errors before they occur. Pandey et al. [5] proposed GreenTPU, a predictive design paradigm for a low-power near-threshold Tensor Processing Unit (TPU). By identifying patterns in error-causing activation sequences, GreenTPU preemptively boosts the operating voltage of specific units to prevent timing errors. This method demonstrates how predictive designs can significantly enhance the performance and energy efficiency of hardware accelerators while maintaining high inference accuracy, making it a crucial contribution to the field of error resilience in deep learning hardware.

C. Adaptive Encodings for Resilient Inference

Adaptation in the face of hardware imperfections is a critical aspect of fault-tolerant deep learning systems. Tambe et al. [6] presented an algorithm-hardware co-design centered around AdaptivFloat, a novel floating-point inspired number format that dynamically adjusts its dynamic range for optimal encoding of neural network parameters. This approach has shown to maintain higher inference accuracies at low bit precision, demonstrating the effectiveness of adaptive encodings in resilient deep learning inference. The AdaptivFloat-quantized networks exhibit minimal degradation in performance metrics, such as BLEU score, even at reduced precision, highlighting the importance of such co-designs in maintaining accuracy under hardware constraints.

D. Training for Hardware Imperfection Tolerance

Addressing hardware imperfections through training methods and co-design methodologies is essential for the longevity and accuracy of DNN accelerators. Song et al. [7] proposed a software-hardware co-design methodology that leverages the self-healing capability of neural networks to map large-weight synapses away from imperfect memristors in RRAM-crossbar-based accelerators. This approach ensures minimal accuracy loss despite resistance variations and stuck-at-faults, showcasing how training methods can be combined with hardware design to create robust deep learning systems.

In conclusion, the integration of hardware and software design is pivotal for the development of error-resilient deep learning systems. The co-design algorithms, predictive designs, adaptive encodings, and training methodologies discussed in this section provide a comprehensive overview of the current state-of-the-art approaches in the field. These strategies not only enhance the fault tolerance of deep learning hardware but also open up new avenues for research in achieving the optimal balance between efficiency, accuracy, and reliability in the face of hardware imperfections.

IV. Fault Detection, Prediction, and Correction Techniques

This section focuses on the mechanisms for detecting, predicting, and correcting faults in deep learning hardware. We will cover methods for neuron resilience prediction, fault detection and remedy frameworks, and safety design techniques for error localization and correction.

A. Neuron Resilience Prediction

The reliability of neural network accelerators, particularly in safety-critical applications, is paramount. As such, the ability to predict neuron resilience to hardware faults is a significant area of research. Schorn et al. [8] introduced a novel method for predicting the error resilience of neurons in deep neural networks. Their approach significantly improves the accuracy and interpretability of resilience predictions, which is crucial for managing the trade-off between reliability and efficiency in hardware accelerators. Similarly, Schorn et al. [9] proposed a method that allows for flexible reliability management by simulating hardware faults and protecting neurons based on resilience estimations. These methods are essential for ensuring that neural network accelerators can maintain high performance while operating reliably in environments with stringent safety requirements.

B. Fault Detection and Remedy Frameworks

The detection and correction of faults in deep learning hardware are critical for maintaining system integrity and performance. Li et al. [10] presented RRAMedy, a framework designed to detect and correct faults in ReRAM-based neural accelerators. This framework utilizes adversarial example testing for on-device fault detection and employs an edge-cloud collaborative mechanism for fault-masking retraining and model updating. This approach minimizes communication overhead and effectively protects the neural accelerator throughout its lifecycle. Additionally, Wang et al. [11] introduced Adversarial Testing, an online fault detection method that leverages adversarial deep learning techniques to detect hardware variations with minimal runtime overhead. These innovative methods contribute significantly to the robustness of deep learning processors against hardware-induced failures.

C. Safety Design Techniques for Error Localization and Correction

Ensuring the safety of neural network accelerators involves not only detecting errors but also localizing and correcting them efficiently. Xu et al. [12] proposed safety design techniques, including Algorithm Based Atomic Error Checking, for a convolutional neural network accelerator. These techniques enable on-the-fly error detection and diagnosis, allowing for online fault management and recovery with minimal area and power overhead. Furthermore, Khoshavi et al. [13] developed SHIELDeNN, an accelerated framework for fault-tolerant deep neural network architectures. SHIELDeNN synergizes mitigation approaches with computational resources to achieve a low-overhead, error-resilient neural network overlay, enhancing the fault tolerance of deep learning systems.

In summary, the literature presents a variety of methods and frameworks aimed at predicting, detecting, and correcting hardware faults in deep learning accelerators. These contributions are vital for the deployment of reliable and efficient deep learning systems, especially in safety-critical applications. Future research could focus on extending these techniques to a broader range of hardware architectures and exploring the potential of integrating multiple fault tolerance strategies to achieve even higher levels of system reliability.

V. Evaluation and Mitigation of Permanent and Transient Hardware Faults

Here, we review the impact of both permanent and transient faults on deep learning hardware and discuss various mitigation strategies. We will consider the resilience of different neural network architectures to hardware errors and the role of fault-tolerant design in accelerators.

A. Impact and Mitigation of Permanent Faults

Permanent faults in hardware, such as those caused by manufacturing defects or wear-out, can have a significant impact on the reliability of deep learning systems. In the context of Quantized Neural Networks (QNNs), [14] presents a comprehensive study on the vulnerability of QNNs to permanent faults and proposes two mitigation strategies: selective channel replication and fault-aware scheduling. Their findings challenge the common belief that QNNs are inherently robust to hardware faults, showing that convolutional layers in QNNs can suffer from accuracy drops of up to 10% due to permanent faults.

Similarly, [15] evaluates the effects of radiation-induced errors on neural networks implemented in FPGAs. The authors propose a selective hardening strategy that targets the most vulnerable layers of the neural network, achieving significant fault masking with minimal overhead. This approach contrasts with traditional methods such as Triple Modular Redundancy (TMR), which are often considered too costly in terms of computational resources.

B. Transient Faults and Their Mitigation

Transient faults, such as Single Event Upsets (SEUs), can also pose a threat to the reliability of deep learning hardware. [16] analyzes the sensitivity of CNNs to SEUs on FPGA accelerators and introduces a fault-tolerant design that detects and masks errors with lower overhead than TMR. This work highlights the importance of understanding the specific error characteristics of deep learning accelerators to develop efficient fault-tolerant mechanisms.

The robustness of modern deep learning architectures against transient faults is further explored by [17], which investigates the resilience of various neural network architectures to SEUs. The study reveals that architectures like ResNet50 and InceptionV3 exhibit higher robustness compared to VGG16, potentially due to features like batch normalization and shortcut connections.

C. Fault Tolerance in Hardware Accelerators

The design of fault-tolerant hardware accelerators for deep learning is an active area of research. [18] proposes strategies to enhance the fault tolerance of systolic array-based DNN accelerators, which have become increasingly popular due to their efficient parallel processing capabilities.

On the other hand, [19] focuses on fault characterization and mitigation at the Register-Transfer Level (RTL) of NN accelerators. The authors present a fault mitigation technique that corrects bit flips more effectively than existing methods, emphasizing the role of application-level and architectural-level specifications in fault severity.

Lastly, [20] introduces a novel fault-tolerant neural network architecture that addresses weight disturbances in emerging DNN accelerator technologies like ReRAM. This work demonstrates the potential for low-cost, effective rectification of accuracy degradation without the need for retraining.

In summary, the evaluation and mitigation of both permanent and transient hardware faults are critical for the reliable deployment of deep learning systems. The literature suggests that a combination of architectural innovations, selective hardening strategies, and efficient fault detection and correction mechanisms can significantly enhance the fault tolerance of deep learning hardware.

VI. Advanced Techniques and Frameworks for Fault Tolerance

This section presents advanced techniques and frameworks that contribute to the fault tolerance of deep learning systems. We will explore the use of clipped activation, sensitivity-based techniques, dynamic quantization, and Bayesian approaches for assessing and improving hardware error resilience.

A. Clipped Activation for Fault Mitigation

The robustness of Deep Neural Networks (DNNs) to hardware faults is critical for their deployment in safety-critical applications. A novel approach to enhance the fault tolerance of DNNs is the use of clipped activation functions. Hoang et al. [21] introduced FT-ClipAct, a technique that mitigates the impact of hardware faults by squashing high-intensity faulty activation values. This is achieved by replacing unbounded activation functions with their clipped counterparts and systematically defining the clipping values to increase network resilience. Their experiments on networks like AlexNet and VGG-16 trained on the CIFAR-10 dataset demonstrated an average improvement of 68.92% in classification accuracy under certain fault conditions. This technique exemplifies how modifying the activation function can significantly bolster the fault tolerance of DNNs.

B. Sensitivity-Based Error Resilience

Another avenue for enhancing fault tolerance is by exploiting the inherent algorithmic error resilience of DNNs. Choi et al. [22] proposed sensitivity-based error resilient techniques that enable aggressive voltage scaling for energy-efficient DNN accelerator design. By using first-order Taylor expansion, they rapidly evaluated filter/channel-level weight sensitivities and assigned computations with more sensitive weights to robust multiply-accumulate (MAC) units. This approach led to substantial energy savings while maintaining accuracy within acceptable bounds. The sensitivity-based technique underscores the potential of fine-tuning hardware resources according to the error resilience characteristics of different DNN components.

C. Dynamic Quantization for Acceleration and Fault Tolerance

Dynamic quantization is a technique that can indirectly contribute to fault tolerance by maintaining accuracy despite hardware constraints. Song et al. [23] introduced DRQ, a dynamic region-based quantization method that adjusts the precision of a DNN model based on sensitive regions in the feature map. This approach not only accelerates DNN inference but also preserves accuracy, achieving significant performance gains and energy reductions compared to other quantization accelerators. By dynamically adjusting precision, DRQ can potentially mitigate the effects of hardware errors on DNN accuracy, making it a relevant technique for fault tolerance.

D. Bayesian Approach for Fault Tolerance Assessment

Assessing the fault tolerance of DNNs is as crucial as improving it. Banerjee et al. [24] proposed a Bayesian Deep Learning based Fault Injection (BDLFI) methodology for this purpose. BDLFI uses Bayesian Deep Learning to model fault propagation and Markov Chain Monte Carlo inference to quantify the effect of faults on NN outputs. This methodology provides a novel perspective on fault injection and assessment, challenging pre-existing results in the field. The Bayesian approach offers a systematic way to evaluate and understand the resilience of DNNs to hardware faults, which is essential for developing more robust systems.

In summary, the advanced techniques and frameworks discussed in this section offer promising directions for improving the fault tolerance of deep learning systems in the presence of hardware errors. From clipped activation functions to sensitivity-based design and dynamic quantization, each approach contributes to the resilience of DNNs. Furthermore, Bayesian methods for fault assessment provide a deeper understanding of fault impacts, guiding future fault-tolerant design strategies.

VII. Broader Perspectives on Reliability and Robustness in Deep Learning Systems

In the final section, we provide a broader perspective on the challenges and opportunities in building reliable deep learning systems. We will discuss the importance of testing methodologies, the impact of storage media errors, and the need for robust machine learning systems that encompass hardware fault tolerance.

A. State-of-the-Art in DNN Reliability and Hardware Error Resilience

The reliability of deep neural networks (DNNs) is of paramount importance, especially as they are increasingly deployed in mission-critical applications. Mittal's survey [25] provides a comprehensive overview of the state-of-the-art techniques for studying and optimizing the reliability of DNN accelerators and architectures. It highlights the need for designing reliability into the system from the ground up rather than retrofitting it. Similarly, Torres-Huitzil's review [26] delves into fault tolerance in neural networks, emphasizing the intrinsic properties of neural models that can be harnessed to achieve fault tolerance passively. These foundational works set the stage for understanding the broader context of DNN reliability and the taxonomy of fault tolerance techniques that are crucial for hardware error resilience.

B. Innovative Fault Tolerance Methods and Their Applications

Recent advancements have introduced innovative methods to boost the reliability of DNNs in the face of hardware errors. Ozen et al. [27] propose Sanity-Check, a fault tolerance method that utilizes checksums to protect the integrity of computations in DNNs, demonstrating its efficacy in safety-critical applications like autonomous driving. Hari et al. [28] explore algorithm-based error detection techniques for convolutions in CNNs, offering a cost-effective alternative to full duplication. These methods represent significant strides in developing low-cost, high-coverage fault tolerance solutions for deep learning systems.

C. Emerging Challenges and Future Directions

While progress has been made, there are still numerous challenges and opportunities for future research. The work by Leung et al. [29] on fault-tolerant algorithms for radial basis function (RBF) networks underlines the complexity of dealing with concurrent weight failures. Chaudhuri et al. [30] study the classification of structural faults in AI accelerators, emphasizing the need for sophisticated machine learning methods to assess functional criticality. These studies, along with the contributions of Mandal et al. [31] on error correction in FPGA configuration memory and Ma et al. [32] on mitigating process variation in CNN accelerators, underscore the ongoing need for research that addresses the multifaceted nature of hardware fault tolerance in deep learning systems.

In conclusion, the pursuit of robust deep learning systems that can withstand hardware errors is a multifaceted challenge that spans across various layers of the system stack. From the design of fault-tolerant neural architectures [33] to the development of testing methodologies that ensure dependable operation [34], the field is ripe with opportunities for innovation. As we move forward, it is crucial to integrate these broader perspectives on reliability and robustness to create deep learning systems that are not only high-performing but also resilient and trustworthy.

References

[1] U. Zahid, G. Gambardella, N. J. Fraser, M. Blott, and K. Vissers, "FAT: Training Neural Networks for Reliable Inference under Hardware Faults," *Proceedings - International Test Conference*, vol. 2020-Novem, 2020.

[2] Z. Liu, K. Jia, W. Liu, Q. Wei, F. Qiao, and H. Yang, "INA: Incremental network approximation algorithm for limited precision deep neural networks," *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, vol. 2019-Novem, 2019.

[3] J. Deng, Y. Rang, Z. Du, Y. Wang, H. Li, O. Temam, P. Ienne, D. Novo, X. Li, Y. Chen, and C. Wu, "Retraining-based timing error mitigation for hardware neural networks," *Proceedings -Design, Automation and Test in Europe, DATE*, vol. 2015-April, no. 2011, pp. 593–596, 2015.

[4] C.-t. Chiu, K. Mehrotra, C. K. Mohan, and S. Ranka, "Training Techniques to Obtain Fault Tolerant Neural Networks," *Proceedings of IEEE 24th International Symposium on Fault-Tolerant Computing*, pp. 360—-369, 1994.

[5] P. Pandey, P. Basu, K. Chakraborty, and S. Roy, "GreenTPU: Predictive Design Paradigm for Improving Timing Error Resilience of a Near-Threshold Tensor Processing Unit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 7, pp. 1557–1566, 2020.

[6] T. Tambe, E. Y. Yang, Z. Wan, Y. Deng, V. Janapa Reddi, A. Rush, D. Brooks, and G. Y. Wei, "Algorithm-hardware co-design of adaptive floating-point encodings for resilient deep learning inference," *Proceedings - Design Automation Conference*, vol. 2020-July, 2020.

[7] Z. Song, Y. Sun, L. Chen, T. Li, N. Jing, X. Liang, and L. Jiang, "ITT-RNA: Imperfection Tolerable Training for RRAM-Crossbar-Based Deep Neural-Network Accelerator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 1, pp. 129–142, 2021.

[8] C. Schorn, A. Guntoro, and G. Ascheid, "Accurate neuron resilience prediction for a flexible reliability management in neural network accelerators," *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*, vol. 2018-Janua, pp. 979–984, 2018.

[9] ——, "Accurate neuron resilience prediction for a flexible reliability management in neural network accelerators," *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*, vol. 2018-Janua, pp. 979–984, 2018.

[10] W. Li, Y. Wang, H. Li, and X. Li, "RRAMedy: Protecting ReRAM-based neural network from permanent and soft faults during its lifetime," in *Proceedings - 2019 IEEE International Conference on Computer Design, ICCD 2019*, 2019, pp. 91–99.

[11] Y. Wang, "Adversarial Testing : A Novel On-line Testing Method for Deep Learning Processors," *VTS*, 2021.

[12] Z. Xu and J. Abraham, "Safety design of a convolutional neural network accelerator with error localization and correction," in *Proceedings - International Test Conference*, vol. 2019-Novem, 2019, pp. 1–10.

[13] N. Khoshavi, A. Roohi, C. Broyles, S. Sargolzaei, Y. Bi, and D. Z. Pan, "SHIELDeNN: Online accelerated framework for fault-tolerant deep neural network architectures," in *Proceedings - Design Automation Conference*, vol. 2020-July, 2020.

[14] G. Gambardella, J. Kappauf, M. Blott, C. Doehring, M. Kumm, P. Zipf, and K. Vissers, "Efficient Error-Tolerant Quantized Neural Network Accelerators," *arXiv*, 2019.

[15] F. Libano, B. Wilson, J. Anderson, M. J. Wirthlin, C. Cazzaniga, C. Frost, and P. Rech, "Selective hardening for neural networks in FPGAs," *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 216–222, 2019.

[16] W. Li, G. Ge, K. Guo, X. Chen, Q. Wei, Z. Gao, Y. Wang, and H. Yang, "Soft Error Mitigation for Deep Convolution Neural Network on FPGA Accelerators," *Proceedings - 2020 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2020*, pp. 1–5, 2020.

[17] A. P. Arechiga and A. J. Michaels, "The Robustness of Modern Deep Learning Architectures against Single Event Upset Errors," *2018 IEEE High Performance Extreme Computing Conference, HPEC 2018*, 2018.

[18] J. J. Zhang, K. Basu, and S. Garg, "Fault-Tolerant Systolic Array Based Accelerators for Deep Neural Network Execution," *IEEE Design and Test*, vol. 36, no. 5, pp. 44–53, 2019.

[19] B. Salami, O. S. Unsal, and A. C. Kestelman, "On the Resilience of RTL NN Accelerators: Fault Characterization and Mitigation," *Proceedings - 2018 30th International Symposium on Computer Architecture and High Performance Computing, SBAC-PAD 2018*, pp. 322–329, 2019. [Online]. Available: http://arxiv.org/abs/1806.09679

[20] T. Liu, W. Wen, L. Jiang, Y. Wang, C. Yang, and G. Quan, "A fault-tolerant neural network architecture," in *Proceedings - Design Automation Conference*, 2019.

[21] L. H. Hoang, M. A. Hanif, and M. Shafique, "FT-ClipAct: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation," *arXiv*, pp. 1241–1246, 2019.

[22] W. Choi, D. Shin, J. Park, and S. Ghosh, "Sensitivity based error resilient techniques for energy efficient deep neural network accelerators," *Proceedings - Design Automation Conference*, 2019.

[23] Z. Song, B. Fu, F. Wu, Z. Jiang, L. Jiang, N. Jing, and X. Liang, "DRQ: Dynamic Region-based Quantization for Deep Neural Network Acceleration," *Proceedings - International Symposium on Computer Architecture*, vol. 2020-May, pp. 1010–1021, 2020.

[24] S. S. Banerjee, J. Cyriac, S. Jha, Z. T. Kalbarczyk, and R. K. Iyer, "Towards a Bayesian Approach for Assessing Fault Tolerance of Deep Neural Networks," *Proceedings - 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume, DSN-S 2019*, pp. 25–26, 2019.

[25] S. Mittal, "A survey on modeling and improving reliability of DNN algorithms and accelerators," *Journal of Systems Architecture*, vol. 104, no. August 2019, p. 101689, 2020. [Online]. Available: https://doi.org/10.1016/j.sysarc.2019.101689

[26] C. Torres-Huitzil and B. Girau, "Fault and Error Tolerance in Neural Networks: A Review," *IEEE Access*, vol. 5, pp. 17 322–17 341, 2017.

[27] E. Ozen and A. Orailoglu, "Sanity-Check: Boosting the Reliability of Safety-Critical Deep Neural Network Applications," *Proceedings of the Asian Test Symposium*, vol. 2019-Decem, pp. 7–12, 2019.

[28] S. K. S. Hari, M. B. Sullivan, T. Tsai, and S. W. Keckler, "Making Convolutions Resilient via Algorithm-Based Error Detection Techniques," pp. 1–12, 2020.

[29] C. S. Leung, W. Y. Wan, and R. Feng, "A Regularizer Approach for RBF Networks Under the Concurrent Weight Failure Situation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 6, pp. 1360–1372, 2017.

[30] A. Chaudhuri, J. Talukdar, F. Su, and K. Chakrabarty, "Functional Criticality Classification of Structural Faults in AI Accelerators," *Proceedings - International Test Conference*, vol. 2020-Novem, pp. 1–5, 2020.

[31] S. Mandal, S. Sarkar, W. M. Ming, A. Chattopadhyay, and A. Chakrabarti, "Criticality aware soft error mitigation in the configuration memory of SRAM based FPGA," *Proceedings - 32nd International Conference on VLSI Design, VLSID 2019 - Held concurrently with 18th International Conference on Embedded Systems, ES 2019*, pp. 257–262, 2019.

[32] M. Ma, J. Tan, X. Wei, and K. Yan, "Process variation mitigation on convolutional neural network accelerator architecture," *Proceedings - 2019 IEEE International Conference on Computer Design, ICCD 2019*, no. Iccd, pp. 47–55, 2019.

[33] W. Li, X. Ning, G. Ge, X. Chen, Y. Wang, and H. Yang, "FTT-NAS: Discovering Fault-Tolerant Neural Architecture," in *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, vol. 2020-Janua, 2020, pp. 211–216.

[34] S. Gerasimou, H. F. Eniser, A. Sen, and A. Cakan, "Importance-Driven Deep Learning System Testing," in *Proceedings - 2020 ACM/IEEE 42nd International Conference on Software Engineering: Companion, ICSE-Companion 2020*, 2020, pp. 322–323.