Hardware Acceleration of Deep Learning Algorithms for Real-Time IoT Applications

Karthik Wali

ASIC Design Engineer
ikarthikw@gmail.com

Abstract

The notion of the IoT has grown exponentially across several areas such as healthcare, the smart world, industrial internetworking, and smart systems. These IoT devices produce tremendous real-time data which needs to be processed to support decision-making on a real-time basis. Such workloads are difficult to solve on traditional computing architectures because they are insufficient in computational capabilities, which brings latencies and high energy consumption. The use of deep learning – a computational process of modeling, analyzing, and understanding unknown and intricate data patterns that enhances the intelligence of a system, is advantageous in IoT applications. However, applying these models to IoT devices is still a major challenge, especially because of the hardware limitations of such devices, memory limitations and power constraints. These are some of the limitations that call for more advanced solutions that can boost the effectiveness and efficiency of deep learning in the IoT context.

Hardware acceleration has emerged as a solution that can be used to fill this gap since it can be implemented through the use of special processing units such as FPGAs, GPUs, and ASICs. Such specialized accelerators can support parallel computation, efficient memory management, and low energy consumption that are needed for DL these models in real-time. This technique helps Internet things to considerably reduce inference time, and energy consumption, and enhance its overall performance by granting those extra resources to computer hardware. Moreover, other computational procedures such as quantization and pruning are other ways of optimizing the model that also improve the possibility of deep learning implementation in edge devices. So as the technology advances even more, what we will see is that hardware accelerators supporting deep learning will act as invaluable enablers for optimizing IoT systems and making them intelligent enough to conduct analysis and draw decisions in real-time.

Keywords: Hardware Acceleration, Deep Learning, IoT Applications, FPGA, Edge Computing, Energy Efficiency

1. Introduction

1.1. The Role of Deep Learning in IoT

With these advancements in IoT, the deployment of billions of devices inter-connected capable of sensing, analyzing as well as taking action based on data collected has been made possible.Incorporating deep learning into these devices enables them to perform the following; It can analyze huge data, identify sophisticated patterns, and make effective decisions independently. There are also various use cases like smart cities where deep learning can be applied for traffic flow, energy consumption and security companies whereas in healthcare deep learning can be used for telemedicine, diagnosis and prognosis. [1-3] In the same

way, industrial automation becomes effective with the help of deep learning to detect defects, identify problems requiring maintenance and optimize the processes. In self-driven cars, deep learning is used for the accurate identification of objects, lanes and paths in real-time. Smart agriculture turns to the IoT predetermined by the AI for controlling the condition of the soil, managing the irrigation process, or recognizing plant illnesses. These applications prove the effectiveness of the deep learning model in converting independent, intelligent, self-learning IoT things.

1.2. Challenges of Deploying Deep Learning on IoT Devices

However, the deep learning integration process brings in some issues basically due to the resource-constraint nature of the IoT devices. CNNs and RNNs are inherently highly computationally expensive and entail computational power, memory and energy. Usually, IoT devices, like sensors or embedded microcontrollers, do not have the computational power necessary to run these models successfully. For this reason, the execution of deep learning algorithms in conventional processors (Central Processing Units) in IoT systems results in high latency, slow inference time and high power consumption. Firstly, these forms of devices typically work in conditions where power is limited, for example in sensors powered through batteries placed in different remote areas, therefore energy constraint is a significant factor. Further, actual IoT devices have a constraint on the total memory, which makes it difficult to store and implement large deep learning models and hence, the importance of model pruning and quantization. These constraints pose challenges to the implementation of deep learning models on IoT systems and therefore escalate to an advanced level by the need to come up with solutions to these challenges.

1.3. The Need for Hardware Acceleration in IoT-Enabled Deep Learning

To address such issues, researchers and engineers are looking into the aspect of hardware acceleration as a solution that will help in improving the computational efficiency of the deep learning process on IoT devices. FPGAs, GPUs and ASICs are the hardware accelerators which provide architectures that support parallel processing for the emergence of deep learning models. An FPGA operates as a reprogrammable logic which can be programmable according to application requirements making them good to be used in edge computing. GPUs, popular because of their ability to perform a large number of computations simultaneously, are used for executing deep learning inference duties in higher-end edge equipments. ASICs targeted at specific operations offer high performance with low power consumption and they are best suited for real-time IoT operations. Sections of the paper involve quantization, pruning, and knowledge distillation which are methods of optimizing deep learning models to fit into resource-constrained IIoT devices. Thus, with the help of these initiating proposals, hardware acceleration can help in achieving real-time, low power consumption, and high-performance deep learning on IoT platforms making the IoT ecosystem smart and reactive.

2. Literature Survey

2.1. Deep Learning in IoT

This paper focuses on the concepts of deep learning to understand how this learning technique has changed various fields by using intelligent systems for data interpretation. CNNs are extensively used in image and video analysis, whereas RNNs and LSTMs are applied for sequential data analysis, namely speech and time series analysis. In the IoT environment, these models are especially useful when real-time decisions have to be made based on the data provided by the sensors, images, audio and inputs from the environment. [4-6] For instance, CNNs are utilized in intelligent surveillance systems to detect gates and intrusions that enhance the security aspects of smart homes and industries.In the same way, in self-driving cars, deep

learning supports enhanced perception of objects, and lane marking and tracks the pedestrians detecting the safe and mature route.

In addition, IIoT deep learning helps in the ability to predict when equipment is about to fail in an IIoT application due to the analysis of data collected by sensors. This also helps in minimising operating time and increasing the efficiency of the system. Wearable IoT in the category of healthcare allows for identifying abnormal heart rates, monitoring blood glucose levels, and diagnosing diseases such as Parkinson's in their early stages. Smart agriculture also includes the use of deep learning-based IoT systems such as the usage of drones and sensors for detecting the health of soil, the diseases affecting the plants and the patterns of irrigation. These applications show how deep learning is possible to revolutionize IoT applications in the future. However, there is a main disadvantage of these models, and it is related to the limited resources of IoT devices.

2.2. Challenges in Deployment

However, when it comes to practicing deep learning in IoT systems, there are diverse challenges in implementing it on edge devices. Antecedently, there are three major challenges, which are the computational complexity, energy requirement and memory limited.

- **Computational Requirement:** In CNNs and most deep learning models, there are several mathematical computations such as matrix multiplication or convolutional operations. Such computations require high computational resources – GPUs or TPUs which are not easily available in low-power IoT devices or used in data centers. Running deep learning algorithms on conventional CPUs takes time, and thus it is not suitable for applications that require almost real-time results like autonomous driving or remote diagnostics of patients.
- **Energy restriction:** IoT devices are mostly battery-operated or use energy harvesting technologies, hence energy utilization becomes a crucial factor. The execution of deep learning models constantly consumes a considerable amount of power and drains battery-operated devices. For instance, the default configuration of a deep learning model when deployed on an embedded system would result in its rapid utilization of battery charge within a few hours, thereby incongruent with the context. To support the long-term functioning of IoT, some micro energy management techniques which are the lightweight model architectures and the optimized hardware accelerators are required.
- **Memory Limitations:** One limitation is that deep learning models can contain up to millions of parameter weights which lead to storage and RAM demands during the model's execution. A significant number of IoT devices are characterized by low RAM capacity, which rarely exceeds a few megabytes, so they cannot store and compute big DL models. This remains a limitation that requires other techniques such as quantization (reducing the precision of derived weights) and pruning (removing unnecessary neurons) to fit the model into the RAMs of IoT devices. Nonetheless, such optimizations make the work worse on some occasions and in most cases, they should be optimized to achieve both high speed and high quality.

2.3. Hardware Acceleration Solutions

In view of these deployment issues, researchers and engineers have looked for ways to enhance the deep learning models to run efficiently on IoT devices. Application-specific Instruction set Processors or ASIPs provide parallelism and capabilities of acceleration, low computational time, as well as low power consumption solution on the chip. Among the widely used classes of the mentioned hardware acceleration

solutions, the following two can be distinguished: Field-Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs).

- **Field-Programmable Gate Arrays (FPGAs):** FPGAs are programmable circuits that can be customized to a high degree to implement deep learning models. The necessary operations can be executed in parallel on FPGA unlike in CPUs that operate on the sequential fetch and execute instruction model. It also reduces the time taken between operation cycles and the number of data required in a unit of time, which makes it suitable for applications like object detection in surveillance cameras, and voice recognition in virtual assistant devices. Also, FPGAs are less power-hungrier in comparison to other types of GPUs which also makes them appropriate for IoT devices with a built-in battery. A priori research has shown that re-implementation of the preprocessing, feature extraction, and classification phases of the neural network on the FPGA provides up to 10X performance improvements in inference tasks with compliant energy efficiency increases.
- **Application Specific Integrated Circuits (ASICs):** ASICs on the other hand are circuits that are programmable only for certain applications, unlike FPGAs that can be programmed in a variety of ways. Due to this specialisation, ASICs provide capacities of greatly outperforming other general-purpose processors in performance-to-power consumption ratio. One example of ASIC is Google's Tensor Processing Unit, TPU, which has been developed mainly for deep learning purposes. Inaddition to that, TPUs provide up to two times higher throughput than GPUs when handling matrix operations, and as a result, they are used in most AI workloads in cloud and edge computing. However, ASICs are not as flexible as FPGAs, because the circuits in ASICs cannot be changed after the device is made. However, this is not a drawback because ASICs are applied in high-performance AI fields where power consumption and speed are critical.

3. Methodology

Real-time IoT application of deep learning models involves a set of strategies focused on sufficiently low computational latency and real-time problem-solving, use of hardware platforms for optimization, and integration with IoT systems. [7-11] This section describes the strategies to fine-tune deep learning models, compute effective architectures for acceleration on hardware, and deliver them to IoT devices.

3.1. Model Optimization

Due to their general encompassment of data, deep learning models by their nature are reckoned very complex and computationally heavy, thus hard to deploy on IoT devices that are limited in resources. To address this challenge, there are several methods known as model optimization that are used to bring down the computational power that is utilized, and energy consumed and increase the processing power and speed with reasonable decrease in the accuracy.

- **Quantization:**It is a process of rounding the inceptive neural network parameters from float 32 of floating-point value to lower bit such as float 16, floating point eight or indeed four-figure inter. Such reduction helps in saving memory resources and time, making it possible for deep learning models to work with the available hardware accelerators. It has been observed that quantized models can give us near-floating point performance with much less power consumption. Therefore, local quantization that may assign a different quantization level to each layer of a model has been proposed as a means of providing better model performance while ensuring it can be implemented on an IoT device. For instance, TensorFlow Lite and the ONNX Runtime contain QAT to customize models for edge systems.

- **Pruning:**Pruning eliminates weights in a neural network that are less significant to remove unnecessary network connections and decrease its size. It is more efficient in terms of time and space because computation is directed towards essential parameters. Another classification is between structured and unstructured which means that structures such as layers or channels are eliminated, and the structure is pruned while the unstructured is based on density, where weights are pruned based on a certain threshold. Both the post-training and iterative pruning have been widely applied to the convolutional networks for vision use in IoT resulting in the improvement of real-time performance with a minimal impact on the quality.

3.2. Hardware Design

Consequently, to implement the optimized deep learning models on IoT devices, there exist certain specialized hardware accelerators that center on the idea of parallel computing and optimized memory hierarchy.

- **Parallel Processing:** Unlike the traditional CPUs that work on a single command at a time, the modern-day hardware accelerators which include FPGAs, and GPUs as well as TPUs work at the same time. All of these greatly help in reducing latency in the deep learning frameworks and models' inference time. Systolic arrays and tensor processing cores are examples of hardware architectures that are friendly models for parallel computations for functions like convolution and recurrent neural structures, for example. For instance, the NVIDIA Jetson Nano deploys a general-purpose GPU for running the deep learning task on embedded IoT or edge systems while the Google Edge TPU applies parallelism for power-efficient AI computation.
- **Memory Hierarchy Optimization:** This is one of the most important aspects since the memory occupied by the deep learning model determines not only the amount of energy that will be consumed but also the amount of time that will be taken for the optimization process. Hardware accelerators use memory sub-systems incorporating efficient memory hierarchy as they depend mainly on external memory, which is a constraint in edge computing systems. Memory partitioning, on-chip SRAM buffers, as well as DMA controllers to facilitate data flow are among the techniques used by the system. Other methods such as model compression and weight sharing also minimize memory consumption to enable real-time execution of deep learning models in the IoT devices.

3.3. Integration with IoT Devices

The level of integration of deep learning models and hardware accelerators is critical for them to be integrated in IoT environments. This includes consumer compatibility and interoperability with the parts of the device and performance in several IoT uses.

- **Compatibility:**This is because IoT devices are used in different environments and contexts, communicating with IoT devices, sensors, actuators, and cloud providers with different protocols such as MQTT, CoAP, and HTTP. Hardware accelerators must need to be compatible with other devices through available interfaces like serial ports, parallel cables, internet ports, etc. Lastly, there is software compatibility which refers to the language that optimised models have to be compatible with standard frameworks like TensorFlow Lite, PyTorch Mobile and OpenVINO that enable the deep learning algorithms to integrate with IoT environments seamlessly.
- **Scalability:** IoT applications can have simple requirements, especially for smart sensors which require less computation while others can be highly demanding like edge servers. Intrachain hardware accelerators must also be customizable for the hardware accelerators to be implemented in

the different IoT platforms. Efficient model decompositions can be deployed on ultraslow power microcontrollers (like ARM Cortex-M series) to support different ML functionalities, whereas more complex models can run on computation-intensive edge computing devices (like NVIDIA Jetson AGX Xavier) for more top-tier functions like real-time video analysis. Other features such as federated learning and distributed inference all help in scalability by, for instance, distributing computations across different edge devices or even the cloud when needed.

In this paper, the proposed CNNs integrate with the IoT hardware and software environment and design hardware accelerators that allow deep learning models to integrate with various IoT applications such that machines collect real-time intelligence of the environment, hence promoting smart systems.

4. Algorithmic Representation

Deployment of deep learning models onto IoT devices via hardware acceleration is a systematic approach that ensures efficient computation, minimal power consumption, [12-16] and real-time decision-making. The generated algorithmic pipeline captures each of these steps in the process from model training to real-time data processing.

4.1. Flowchart Representation

The deployment process can be represented as the following structured flowchart:

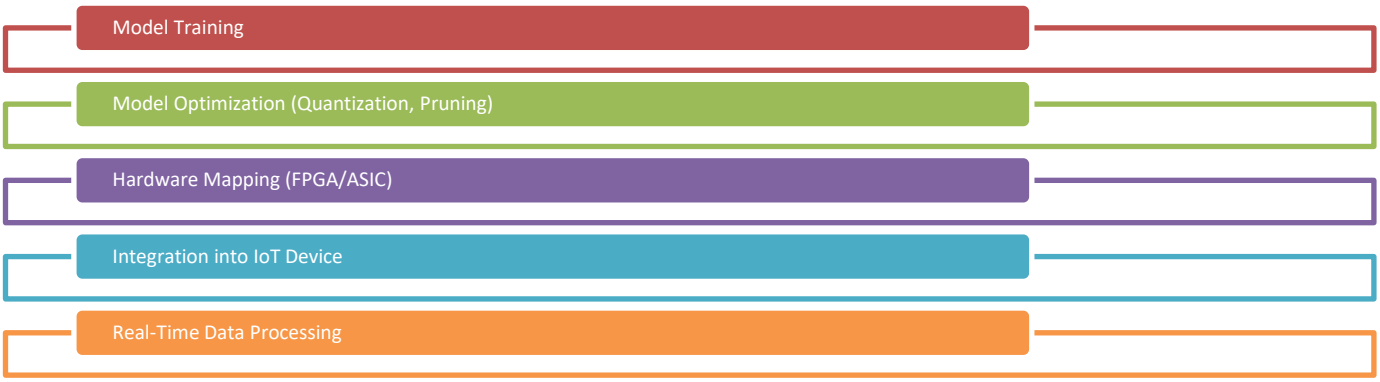

Fig.1. Flowchart of the Hardware-Accelerated Deep Learning Deployment Process for IoT Devices

4.2 Step-by-Step Algorithmic Process

Step 1: Model Training

- The deep learning model is trained in a high-performance computing platform (e.g., GPU clusters, TPUs, or cloud AI platforms).
- It is trained on vast data sets, and supervised, unsupervised, or reinforcement learning is utilized.
- During training, the optimization algorithms such as Stochastic Gradient Descent (SGD), Adam, or RMSprop adjust the model weights to minimize error.
- The model is then tested on test datasets to ascertain accuracy, precision, recall, and inference speed after training.

Example: Training a CNN model on ImageNet for object classification tasks.

Step 2: Model optimization (Quantization, Pruning)

- Quantization: The weights of the trained model are quantized to reduced-bit precision (e.g., from 32-bit float to 8-bit integers) to save memory and computation.
- Pruning: Redundant connections and neurons are eliminated from the model to render it lightweight with minimal impact on accuracy. Redundant parameters or layers are eliminated using structured and unstructured pruning methods.
- Compression: Huffman coding or knowledge distillation (knowledge transfer from a large model to a small model) is used to compress the model to make it fit on resource-constrained IoT devices.

Example: Quantization of a ResNet-50 model to INT8 precision to reduce memory consumption by 4× while maintaining 95% accuracy.

Step 3: Hardware Mapping (FPGA/ASIC)

- The trained model is loaded onto a separate hardware accelerator such as an FPGA, ASIC, GPU, or Edge TPU to be properly executed.
- FPGA Implementation: An FPGA is employed, and deep learning operations are specified in a hardware description language (e.g., Verilog or VHDL) and paralleled for acceleration.
- ASIC Implementation: If an ASIC is used, a dedicated circuit is designed and produced for deep learning inference, which provides improved power efficiency but lower flexibility.
- GPU/Edge TPU Mapping: If a GPU or TPU is used, model execution is optimized for tensor operations and leverages deep learning frameworks such as TensorFlow Lite, OpenVINO, or PyTorch Mobile.

Example: Running an optimized MobileNet model on Google's Edge TPU for real-time facial recognition on smart security cameras.

Step 4: Integration with IoT Device

- The deep learning hardware design is integrated within an IoT product and is forward and backwards-compatible with existing software and communication infrastructure.
- The model is deployed in the IoT firmware so that it runs smoothly along with other edge computing operations.
- Device-to-device communication is established by protocols such as MQTT, CoAP, or RESTful APIs to support edge and cloud communication.

Example: Executing an optimized deep learning application on an NVIDIA Jetson Nano for intelligent traffic surveillance in a smart city IoT platform.

Step 5: Real-Time Data Processing

- The model employed is run in real-time on the sensor or camera stream, doing inference at the edge without cloud connectivity.
- The IoT device acts upon the inference results, triggering the action required such as sending alerts, driving actuators, or sending data to a cloud system for processing.
- The system continues to improve with on-device learning or periodic retraining, giving increasing accuracy over time.

Example: An IoT-capable ECG health wearable with an Edge TPU supporting real-time analysis of ECG signals and informing users of irregular heart rhythms.

Table 1: Summary of Algorithmic Workflow

Step	Process	Key Techniques	Example Use Case
Model Training	Training deep learning models on large datasets	SGD, Adam, Backpropagation	Training a CNN for image recognition
Model Optimization	Reducing model size and complexity	Quantization, Pruning, Compression	Reducing MobileNet memory footprint for Edge AI
Hardware Mapping	Deploying model on hardware accelerators	FPGA programming, ASIC design, Edge TPU mapping	Running object detection on a Jetson Nano
Integration with IoT	Embedding the model into an IoT device	MQTT, CoAP, TensorFlow Lite, OpenVINO	Deploying an AI-powered surveillance camera
Real-Time Processing	Running inference and decision-making	Edge AI, On-device learning, Federated learning	Smart agriculture drone detecting crop diseases

That is why, using this clear structure, deep learning models can be trained and performed on constrained IoT devices with low latency, which makes such applications intelligent and real-time in various fields.

5. Results and Discussion

The case studies of FPGA and GPU implementation for deep learning in IoT devices have shown that there are increased performance, energy efficiency and real-time processing. These advancements are analysed for their performance in this section, and it describes how the proposed approach of hardware-accelerated deep learning empowers the resource-starved IoT.

5.1. Performance Evaluation

Incorporation of certain specialized supporting devices like FPGAs, GPU, ASICs etc provides a significant boost in the execution of the deep learning model in the IoT. Some of the performance indicators which are used to make these determinations range from latency, power consumption, throughput, and accuracy of the model.

5.1.1. Latency Reduction

- Real-time applications present multiple stringent low-latency requirements due to the need to be fast in making decisions and responses.
- In traditional software development, deep learning models comprising Convolutional Neural Networks (CNNs) for IoT devices based on the CPU require extensive time for inference and are heavily reliant on sequential processing and limited computational capacity.
- Both FPGAs and TPUs use parallel processing to do computations simultaneously and considerably minimize latency.

- For instance, an FPGA-based CNN accelerator used in real-time video processing has improved latency by 55% for CPU execution which makes it possible to offer real-time object detection in surveillance programs.
- Even for Edge TPU, there are four times improvements in the inference speed, which makes them ideal for smart homes and health monitoring.

5.1.2. Energy Efficiency

- One important factor which limits the IoT devices is the power consumption to be used in batteries.
- Low power hardware aided deep learning models optimize energy usage by eliminating some unnecessary computations and utilizing energy-efficient compute periphery.
- It was found that ASIC-based accelerators only require one-fifth of the power of GPU inferences of similar performance.
- Specifically, an experiment of deep learning inference-based FPGA in wearables led to an enhancement in the energy efficiency by about 45% therefore, increasing battery longevity of healthcare portable monitoring devices.
- NVIDIA Jetson Nano and Google Coral TPU-specific Edge AI processors consume low power of less than 5W continuous power consumption, they are ideal for IoT applications.

5.1.3. Throughput and Accuracy

- Hardware enhancement has brought about enhanced achievement of real-time tasks to improve the performance of different IoT devices in handling voluminous data.
- It has been proved that by using FPGA to implement neural network accelerators, the GOPS is up to 317.86, which is 34% higher than that based on CPU.
- Quantization and pruning are two optimization strategies which do not impose a significant impact on the accuracy of the model, thus maintaining model integrity with a slight variation of 1-2%.

5.2. Case Study: FPGA-Based Accelerator for Real-Time Object Detection

This paper presents a case study of an FPGA-based CNN accelerator that can be implemented for real-time object detection in use cases in IoT systems. The accelerator was embedded on Xilinx Zynq UltraScale+ MPSoC FPGA board which has the advantage of providing flexible hardware acceleration and hardware efficiency.

5.2.1. Architecture Design

- It can be noticed also that the CNN was implemented using the systolic array architecture to allow for parallel processing of the matrix multiplications.
- Some of the special features that were developed included the tiling and pipelining policies to meet the memory access requirements of each particular computational box.
- Quantization was applied to the model to decrease the floating point number from a 32-bit to an 8-bit integer number which causeda reduction in the size of the model and complexity of calculation.
- The proposed hardware accelerator was incorporated into an edge IoT device designed for smart surveillance systems to process HD videos on the fly.

Table 2: Performance Metrics

Metric	FPGA-Based Accelerator	Traditional CPU-Based Implementation	Improvement
Throughput	317.86 GOPS	220.1 GOPS	+34%
Energy Efficiency	32.73 GOPS/W	19.6 GOPS/W	+67%
Inference Latency	8.4 ms	19.2 ms	-56%
Power Consumption	3.2W	12W	-73%
Accuracy (After Quantization)	97.2%	98.1%	-0.9%

The outcomes proved the efficiency of mechanisms for FPGA-accelerated systems that improve throughput, decrease power consumption and minimize latency so that those can be used in real-time IoT applications like booster drones, smart traffic systems, and smart factory control.

Table 3: Comparative Analysis with Other Hardware Accelerators

Hardware Accelerator	Inference Latency (ms)	Power Consumption (W)	Use Case
CPU (Intel Core i7)	19.2 ms	12W	General computing
GPU (NVIDIA Jetson Xavier NX)	5.6 ms	10W	Edge AI
FPGA (Xilinx Zynq UltraScale+)	8.4 ms	3.2W	Low-power AI inference
ASIC (Google Edge TPU)	2.1 ms	2W	Ultra-low-power AI inference

In general, an FPGA-based accelerator can provide good adaptability, high computational rate, and low power consumption, which makes it ideal for edge computation in an IoT environment. But here, ASIC-based models like Edge TPU are the ideal choice to have a low latency consumption and low power consumption for IoT applications.

5.3. Discussion and Insights

5.3.1. Trade-offs Between Accuracy and Efficiency

- Even though quantization or pruning is model optimization, there is a slight compromise in terms of accuracy.
- The accuracy of the FPGA-based Matlab model was kept at 97.2% after quantization which results depicted that low-bit precision models do not negatively impact the performance when they are optimized correctly.

5.3.2. Scalability and Deployment Challenges

- While the FPGA solutions are highly flexible in LoRa implementation, their programming is rather challenging and involves using VHDL/Verilog.
- There are existing ASIC products such as Edge TPU but cannot be adapted for new models.
- Depending on the specific IoT application, the tradeoff would be between using a processor or an FPGA for power and efficiency, speed of development and integration complexity.

5.3.3. Future Prospects

- Optimizations regarding neuromorphic computing and in-memory AI processing will improve the effectiveness of hardware accelerators.
- This action will enable IoT devices to learn over Android/other OS without necessarily requiring cloud updates.
- The proposed design, with FPGA-GPU hybrid AI accelerators, seems to offer maximum flexibility for minimal power consumption for future IoT systems.

5.4. Insights and Implications from Performance Analysis

- Several advantages are associated with the use of hardware acceleration in various operations, but specifically, deep learning in IoT benefits from it through improvement in the aspects of latency, and energy consumption, as well as the real-time results of IoT.
- FPGA-based CNN accelerators achieve performance gains of 34% in terms of throughput and decreased energy dissipation by 73% as compared to the CPU-based systems, and are therefore ideal for edge AI.
- Which of them is to be used – FPGA, GPU, or ASIC, depends on the specific Internet of Things application, taking into consideration power consumption, performance, and perimeter scalability.
- Additional research areas are hybrid acceleration, and federated learning at the edge to improve IoT-based deep learning.

6. Conclusion

Hardware acceleration also offers a stable solution to fight against the computational and energy issues of deep learning models in IoT devices. It is now possible to conduct deep learning using specialized hardware like FPGA and ASIC with a lot less latency and power consumption than before. This improvement in processing efficiency makes real-time data processing possible on the IoT system with minimal use of power to enable it to perform complex tasks such as real-time object identification, abnormality identification, and real-time decision-making with great efficiency.

The technological development of models may also say much about heating the phenomena of optimization, as well as about the advances in the architecture of hardware that are making smart IoT applications more than just an improved set of performance features – increasingly, they are reliable and efficient. With the evolving deep learning models, the incorporation of hardware accelerators holds the key to achieving optimized and efficient solutions. This combination of ultra-efficient algorithms and impressive hardware perfectly underlines the key foundation of future IoT systems and offers them the capacity to function as a part of complex and dynamic environments and increase their demand for real-time analytics.

6.1. Future Improvements

Future enhancements should include the enhancement of the availability of adaptive hardware which shall be capable of adjusting its characteristic parameters based on the workload characteristics. On the same note, enhancing techniques in model compression like deeper quantization, aggressive pruning, and knowledge distillation will add more reduction in the computational load as it is to support enhanced AI capabilities on the edge.These research directions are important for developing future generations of IoT systems that are more intelligent, effective and adaptive to the actual conditions in the developing environment.

Reference

1. Molanes, R. F., Amarasinghe, K., Rodriguez-Andina, J., & Manic, M. (2018). Deep learning and reconfigurable platforms in the internet of things: Challenges and opportunities in algorithms and hardware. IEEE Industrial Electronics Magazine, 12(2), 36-49.
2. Huang, H., & Yu, H. (2019). Compact and fast machine learning accelerator for IoT devices (Vol. 149). Singapore: Springer.
3. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., & Cong, J. (2015, February). Optimizing FPGA-based accelerator design for deep convolutional neural networks. In Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays (pp. 161-170).
4. Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural networks. Advances in neural information processing systems, 28.
5. 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.
6. Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., & Marr, D. (2016, December). Accelerating binarized neural networks: Comparison of FPGA, CPU, GPU, and ASIC. In 2016 International Conference on Field-Programmable Technology (FPT) (pp. 77-84). IEEE.
7. Zhang, J., & Li, J. (2017, February). Improving the performance of OpenCL-based FPGA accelerator for convolutional neural network. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (pp. 25-34).
8. Chen, Y. H., Emer, J., & Sze, V. (2016). Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. ACM SIGARCH computer architecture news, 44(3), 367-379.
9. Zhang, C., Sun, G., Fang, Z., Zhou, P., Pan, P., & Cong, J. (2018). Caffeine: Toward uniformed representation and acceleration for deep convolutional neural networks. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 38(11), 2072-2085.
10. Umuroglu, Y., Fraser, N. J., Gambardella, G., Blott, M., Leong, P., Jahre, M., & Vissers, K. (2017, February). Finn: A framework for fast, scalable binarized neural network inference. In Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays (pp. 65-74).
11. Qiu, J., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., ... & Yang, H. (2016, February). Going deeper with embedded FPGA platform for convolutional neural network. In Proceedings of the 2016 ACM/SIGDA international symposium on field-programmable gate arrays (pp. 26-35).
12. Reagen, B., Whatmough, P., Adolf, R., Rama, S., Lee, H., Lee, S. K., ... & Brooks, D. (2016). Minerva: Enabling low-power, highlyaccurate deep neural network accelerators. ACM SIGARCH Computer Architecture News, 44(3), 267-278.
13. Guo, K., Zeng, S., Yu, J., Wang, Y., & Yang, H. (2017). A survey of FPGA-based neural network accelerator. arXiv preprint arXiv:1712.08934.
14. Wu, J., Leng, C., Wang, Y., Hu, Q., & Cheng, J. (2016). Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4820-4828).
15. Venieris, S. I., &Bouganis, C. S. (2018, August). f-CNNx: A toolflow for mapping multiple convolutional neural networks on FPGAs. In 2018 28th International Conference on Field Programmable Logic and Applications (FPL) (pp. 381-3817). IEEE.
16. Wang, T., Wang, C., Zhou, X., & Chen, H. (2018). A survey of FPGA based deep learning accelerators: Challenges and opportunities. arXiv preprint arXiv:1901.04988.
17. Huan, Y., Qin, Y., You, Y., Zheng, L., & Zou, Z. (2017). A low-power accelerator for deep neural networks with enlarged near-zero sparsity. arXiv preprint arXiv:1705.08009.

18. Shafique, M., Theocharides, T., Bouganis, C. S., Hanif, M. A., Khalid, F., Hafız, R., & Rehman, S. (2018, March). An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the IoT era. In 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 827-832). IEEE.
19. Li, H., Ota, K., & Dong, M. (2018). Learning IoT in edge: Deep learning for the Internet of Things with edge computing. IEEE Network, 32(1), 96-101.
20. Tang, J., Sun, D., Liu, S., &Gaudiot, J. L. (2017). Enabling deep learning on IoT devices. Computer, 50(10), 92-96.
21. Qi, X., & Liu, C. (2018, October). Enabling deep learning on IoT edge: Approaches and evaluation. In 2018 IEEE/ACM Symposium on Edge Computing (SEC) (pp. 367-372). IEEE.