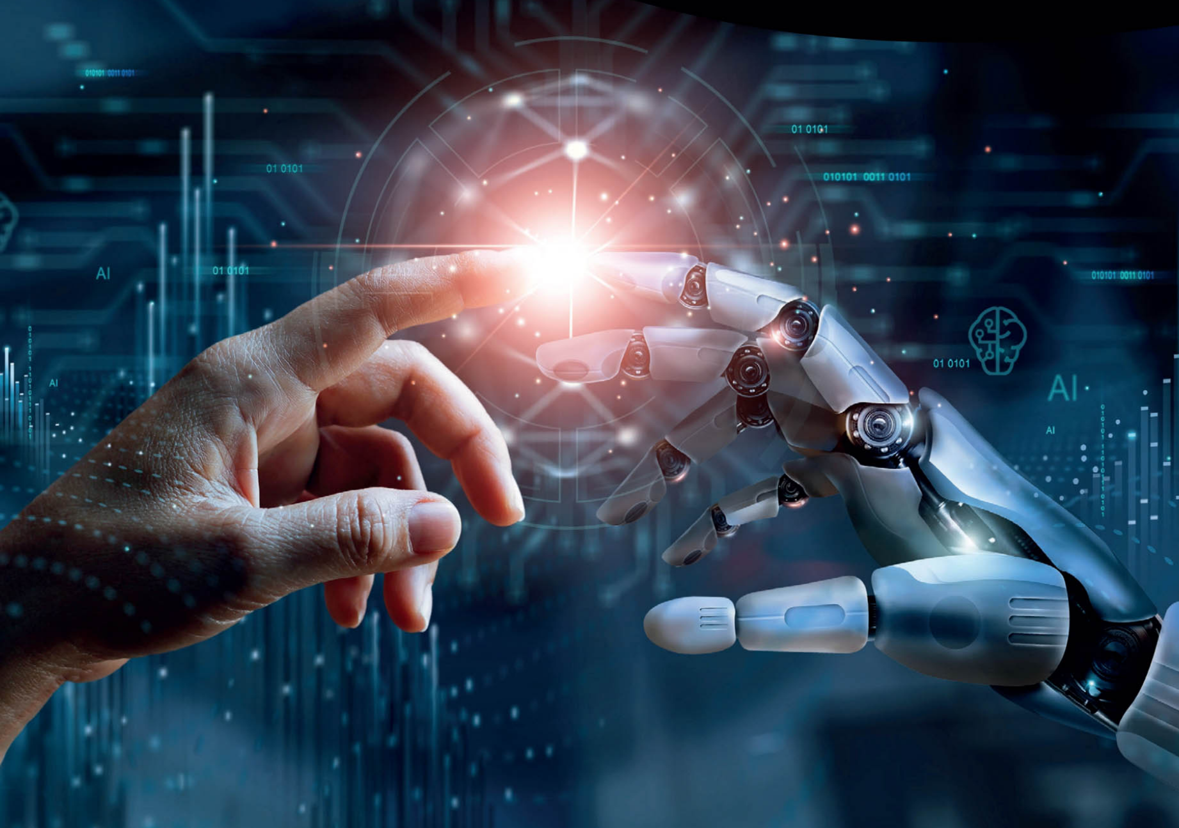
Embedded Artificial Intelligence

Real-Life Applications and Case Studies

Edited by Arpita Nath Boruah, Mrinal Goswami, Manoj Kumar, and Octavio Loyola-Gonzalez



A Chapman & Hall Book

CRC Press
Taylor & Francis Group

Embedded Artificial Intelligence

This book explores the role of embedded AI in revolutionizing industries such as healthcare, transportation, manufacturing, and retail. It begins by introducing the fundamentals of AI and embedded systems and specific challenges and opportunities. A key focus of this book is developing efficient and effective algorithms and models for embedded AI systems, as embedded systems have limited processing power, memory, and storage. It discusses a variety of techniques for optimizing algorithms and models for embedded systems, including hardware acceleration, model compression, and quantization.

Key features:

- Explores security experiments in emerging post-CMOS technologies using AI, including side-channel attack-resistant embedded systems.
- Discusses different hardware and software platforms available for developing embedded AI applications, as well as the various techniques used to design and implement these systems.
- Considers ethical and societal implications of embedded AI vis-a-vis the need for responsible development and deployment of embedded AI systems.
- Focuses on application-based research and case studies to develop embedded AI systems for real-life applications.
- Examines high-end parallel systems to run complex AI algorithms and comprehensive functionality while maintaining portability and power efficiency.

This reference book is for students, researchers, and professionals interested in embedded AI and relevant branches of computer science, electrical engineering, or artificial intelligence.

Embedded Artificial Intelligence

Real-Life Applications and Case Studies

Edited by
Arpita Nath Boruah, Mrinal Goswami,
Manoj Kumar, and Octavio Loyola-González

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
A CHAPMAN & HALL BOOK

Designed cover image: ShutterStock

First edition published 2025
by CRC Press
2385 NW Executive Center Drive, Suite 320, Boca Raton FL 33431

and by CRC Press
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

Library of Congress Cataloging-in-Publication Data
Names: Nath Boruah, Arpita, editor. | Goswami, Mrinal, editor. |
Kumar, Manoj, 1986- editor. | Loyola-González, Octavio, editor.
Title: Embedded artificial intelligence : real-life applications and case
studies / edited by Arpita Nath Boruah, Mrinal Goswami, Manoj Kumar,
Octavio Loyola-González.
Description: First edition. | Boca Raton : C&H/CRC Press, 2025. |
Includes bibliographical references and index.
Identifiers: LCCN 2024040905 (print) | LCCN 2024040906 (ebook) |
ISBN 9781032766607 (hardback) | ISBN 9781032770673 (paperback) |
ISBN 9781003481089 (ebook)
Subjects: LCSH: Artificial intelligence—Industrial applications.
Classification: LCC TA347.A78 E433 2025 (print) | LCC TA347.A78 (ebook) |
DDC 006.3—dc23/eng/20241218
LC record available at https://lccn.loc.gov/2024040905
LC ebook record available at https://lccn.loc.gov/2024040906

ISBN: 978-1-032-76660-7 (hbk)
ISBN: 978-1-032-77067-3 (pbk)
ISBN: 978-1-003-48108-9 (ebk)

DOI: 10.1201/9781003481089

Typeset in Times
by codeMantra

Contents

SECTION C Ethical Considerations in Embedded AI

Preface

In the rapidly evolving landscape of technology, the integration of artificial intelligence (AI) into embedded systems stands as a frontier of innovation. This book, *Embedded Artificial Intelligence: Real-Life Applications and Case Studies*, aims to bridge the gap between theoretical AI concepts and their practical implementation in embedded systems, offering readers a comprehensive guide to understanding and applying AI in real-world scenarios.

The motivation behind this book stems from the burgeoning need for intelligent, autonomous systems capable of performing complex tasks with minimal human intervention. From smart homes to autonomous vehicles, the potential applications of embedded AI are vast and transformative. Our goal is to equip engineers, researchers, and enthusiasts with the knowledge and tools necessary to navigate this exciting domain.

The structure of this book is designed to provide a balanced mix of foundational theory and practical insights. The initial chapters delve into the fundamentals of embedded AI, exploring key concepts such as machine learning, neural networks, and data processing techniques tailored for resource-constrained environments. As readers progress, they will encounter detailed case studies and real-life applications that demonstrate the implementation of AI in various industries.

Each case study is selected to showcase unique challenges and innovative solutions, offering a diverse perspective on the practicalities of embedding AI. These examples illustrate how AI can enhance functionality, improve efficiency, and create new possibilities across different sectors. We have meticulously curated these cases to include a range of applications, from healthcare and robotics to agriculture and smart cities.

Collaboration and interdisciplinary knowledge are crucial in the field of embedded AI. Throughout this book, we emphasize the importance of integrating insights from various domains, encouraging readers to think beyond traditional boundaries. By fostering a holistic understanding, we aim to inspire new ideas and foster innovation in the design and deployment of intelligent embedded systems.

The journey to mastering embedded AI is both challenging and rewarding. We hope this book serves as a valuable resource, guiding you through the intricacies of AI technologies and their application in embedded systems. Whether you are a seasoned professional or a newcomer to the field, we invite you to explore the pages that follow with curiosity and an open mind.

We extend our gratitude to the contributors, researchers, and industry experts who have shared their knowledge and experiences, enriching the content of this book. Their dedication and insights have been instrumental in shaping this comprehensive guide.

Embark on this journey with us and discover the transformative power of embedded AI. The future of intelligent systems awaits, and we are excited to be part of your exploration.

Arpita Nath Boruah
Mrinal Goswami
Manoj Kumar
Octavio Loyola-González

Editors

Arpita Nath Boruah is an Assistant Professor, Programme Computer Science and Engineering in the Faculty of Engineering and Technology at Assam Down Town University, Assam, India. She earned her doctorate degree from the National Institute of Technology, Silchar, Assam, in the Department of Computer Science and Engineering. She holds an MTech degree in Computer Science and Engineering from NERIST, Arunachal Pradesh, and a bachelor's degree in Computer Science and Engineering from Jorhat Engineering College, Assam. She is currently involved in research areas that include data mining, machine learning, data science, and transparent decision-making system. She has published several international journal articles in SCI/SCIE-indexed journals from various reputed publishers and has attended many international conferences.

Mrinal Goswami earned his BTech degree in Computer Science and Engineering from the North Eastern Regional Institute of Science and Technology, India, in 2012. He earned his MTech and PhD in Computer Science and Engineering from the National Institute of Technology, Durgapur, West Bengal, India, in 2014 and 2019, respectively. He has over eight years of research and teaching experience as an assistant professor. Currently, he is with the Faculty of Engineering and Technology, Programme of Computer Science and Engineering, Assam Down Town University, Guwahati, Assam, India, and working as an Associate Professor. Before this, he was an Assistant Professor in the Department of Computer Science and Engineering at the University of Petroleum and Energy Studies (NIRF Rank 54 & NACC A accredited) and North Eastern Regional Institute Science and Technology (Deemed-to-be-University under the Ministry of Education, Govt. of India). He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and Cellular Automata India Chapter. He has published several international research articles on design, testing, security, and fault-tolerance issues in digital systems, quantum dot cellular automata, quantum computing, and AI.

Manoj Kumar completed his PhD from The Northcap University and MSc (Information Security and Digital Forensics) from Technological University Dublin, Ireland, in 2013. He received a fully funded scholarship for the MTech and MSc programmes from the Irish Government and the MHRD, respectively. He has more than 13 years of research, teaching, and corporate experience. He is currently an Associate Professor of Cyber Security at the University of Wollongong in Dubai, UAE, and a Research Head for the Network and Cyber Security Cluster @UOWD. He has published over 170 articles in various journals and conferences. He published over 10 patents and delivered three international research-funded projects. He published three authored books and eight edited books. His specializations are digital forensics, machine learning, information security, image processing, IOT, and computer networking. He is a member of numerous renowned professional bodies including IEEE, ACM, IAENG, ISTS, and UACEE. He received the Best Researcher Award in 2020, an Outstanding Scientist Award in 2021, and a Young Researcher Award in 2021 from recognized international professional societies.

Octavio Loyola-González earned his PhD in Computer Science from the National Institute of Astrophysics, Optics, and Electronics, Mexico. He has won several awards from different institutions for his research work on applied projects. Consequently, he is a member of the National System of Researchers in Mexico (Rank 1). He worked as a distinguished professor and researcher at the Tecnologico de Monterrey, Campus Puebla, for undergraduate and graduate programmes of Computer Sciences. He was a Managing Director for Altair Management Consultants, having branches in the USA, UK, Spain, and Mexico, where he led several teams of data scientists for AI and ML, business intelligence reporting, and traditional consulting. Currently, he serves as an AI Executive Consulting Manager at NTT DATA. His responsibilities include bringing in clients from the automotive, manufacturing, real estate, infrastructure, and services sectors for deals related to advanced analytics (AI & GenAI). Additionally, he oversees the operations and activities of a team of data scientists, guiding them on both technical and business strategy aspects. Furthermore, he is actively involved in publishing scientific papers and books in reputable journals to promote advancements in AI research.

Contributors

Sayandip Adhikari
School of Computer Engineering
KIIT University
India

Alka Agrawal
Computer Science and Engineering
Gulzar Group of Institutions
Mathura, India

R. Aishwarya
Sathyabama Institute of Science and Technology
India

Tina Babu
Alliance University
India

Aritra Banerjee
School of Computer Engineering
KIIT University
India

Sumit Kumar Banshal
Alliance College of Engineering and Design
Alliance University
Bangalore, India

Nazmin Begum
Computer Science Department
PES University
Bangalore, India

Angelina Chiran
ICFAI University
India

Amrita Choudhury
Gauhati University
India

Saswati Debnath
Alliance College of Engineering and Design
Alliance University
Bangalore, India

Lachit Dutta
Department of Electronics and Communication Engineering
Gauhati University
India

P. M. Ebin
Department of Computer Science and Engineering
Alliance University
India

Vishal Goyal
Computer Science and Engineering
Gulzar Group of Institutions
Mathura, India

Jasmin Gupta
CSE Department
ICFAI University
India

Siddharth Himanshee
School of Computer Engineering
KIIT University
India

S. Jaisiva
Sri Krishna College of Technology
India

T. Jesudas
Department of Mechatronics
Mahendra Engineering College
India

P. R. Jeyalakshmi
DR. John Mathai center, SMS
Regional Centre of Calicut University
India

Ramesh M. Kagalkar
Nagarjuna College of Engineering and Technology
India

S. Kalaiselvi
Information and Communication Engineering
Anna University
India

Mousumi Karmakar
Department of Computer Science and Engineering
Alliance University
India

Amitoj Kaur
Gulzar Group of Institutions
India

Harjit Kaur
Gulzar Group of Institutions
India

A. Krishnamoorthy
Vellore Institute of Technology
India

Ankita Maheshwari
Computer Science and Engineering
Gulzar Group of Institutions
Agra, India

G. Mathivanan
Department of CSE
Sathyabama Institute of Science and Technology
India

Hitesh Mohapatra
School of Computer Engineering
KIIT University
India

Sayan Shourya Muhuri
KIIT University
India

Annamalai Muthu
University of Technology and Applied Science – Nizwa
Sultanate of Oman

Shradha Naik
Machine Learning Department
BMS College of Engineering
Bangalore, India

Rekha R. Nair
Alliance University
India

Durgabati Podder
ICFAI University
India

C. Sakthi Gokul Rajan
Sreenivasa Institute of Technology and Management Studies
India

K. Ramalakshmi
Alliance College of Engineering and Design
Alliance University
Bangalore, India

Sukanya Saha
ICFAI University
India

Kandarpa Kumar Sarma
Gauhati University
India

Velappagari Sekhar
Kuppam Engineering College
India

M. Senbagavalli
Alliance College of Engineering and Design
Alliance University
Bangalore, India

Yagnyasenee Sengupta
Department of CSE
ICFAI University
India

Rajesh Sharma
Department of Computer Science and Engineering
Alliance University
India

Sujoy Shil
ICFAI University
India

Mittal Shit
KIIT University
India

Pratibha Soram
Gulzar Group of Institutions
India

Akey Sungheetha
Alliance University
India

Karna Sutradhar
ICFAI University
India

Tran Minh Tung
FPT School of Business and Technology (FSB)
FPT University
Vietnam

Section A

Overview of Embedded Systems and Artificial Intelligence

1 Unleashing Intelligence at the Edge

Exploring Machine Learning in Embedded Systems

Mousumi Karmakar

1.1 INTRODUCTION

Embedded systems, which are distinguished by their small footprint, restricted resources, and ability to operate in real time, are progressively evolving into centres of intelligence by incorporating machine learning algorithms. This introductory chapter delves into the convergence of machine learning and embedded systems, elucidating the complexities associated with the implementation of sophisticated algorithms on devices that have limited resources. Commencing with an examination of the machine learning algorithms that are appropriate for embedded systems, we assess their capability to accommodate the distinctive limitations of these platforms. Recognizing the significance of implementing machine learning on devices with limited resources, we explore the ramifications across multiple sectors, such as automotive, consumer electronics, IoT, and healthcare. Our objective is to outline the limitations, intriguing benefits, and optimal strategies to enhance embedded intelligence.

1.2 FUNDAMENTALS OF EMBEDDED SYSTEMS

Embedded systems encompass a diverse array of concepts central to the design, development, and deployment of specialized computing devices. Embedded systems are characterized by their integration into larger systems or products, often performing specific functions with real-time constraints and resource limitations.

Deploying machine learning on embedded devices poses significant constraints and challenges due to limited computational resources, including memory, processing power, and energy consumption (Cao et al., 2018). Optimizing machine learning models for deployment on embedded platforms requires striking a balance between model complexity and performance. Additionally, real-time processing requirements and stringent latency constraints further complicate the integration of machine learning algorithms into embedded systems.

These systems rely on low-power microcontrollers or microprocessors optimized for specific tasks, alongside integrated sensors and actuators. These architectures

DOI: 10.1201/9781003481089-2

prioritize compactness, efficiency, and real-time responsiveness, facilitating seamless integration into various embedded applications such as IoT devices, automotive systems, and industrial control systems.

Overall, mastering the fundamentals of embedded systems involves comprehending the intricate interplay between hardware and software components to create efficient, reliable, and cost-effective solutions for a wide range of applications, from consumer electronics to industrial automation and beyond.

1.3 MACHINE LEARNING ALGORITHMS FOR EMBEDDED SYSTEMS

1. *ML Algorithms on Embedded Systems*

   Machine learning algorithms tailored for embedded systems represent a convergence of computational ingenuity and resource efficiency. In the realm of embedded systems, the landscape of machine learning algorithms suitable for deployment is diverse and evolving (Yang et al., 2019). From lightweight classifiers like decision trees and k-nearest neighbours to more complex models such as neural networks and support vector machines, the spectrum offers solutions tailored to various applications and hardware constraints. Classifiers such as decision trees and support vector machines operate by learning decision boundaries from labelled data, making predictions based on the features of input samples. They are often favoured in embedded systems for their simplicity and efficiency. Neural networks, on the other hand, utilize interconnected layers of nodes to learn complex patterns and relationships in data, enabling more advanced tasks such as image recognition and natural language processing (Chen et al., 2020). While powerful, deploying neural networks on embedded systems can be challenging due to their computational demands and memory requirements. Efficient implementations, such as model quantization and hardware acceleration, are crucial for their successful deployment in resource-constrained environments.
2. *Optimization Techniques*

   Techniques for optimizing these algorithms within resource-constrained environments are pivotal, encompassing strategies like model compression, pruning, and quantization to reduce memory and computational overhead. Model quantization involves reducing the precision of model parameters, typically from floating-point to fixed-point representation, thereby reducing memory usage and computation complexity. Pruning techniques selectively remove redundant connections or parameters from neural networks, resulting in a smaller and more efficient model without significant loss of accuracy (Han et al., 2015). These optimization techniques are essential for ensuring efficient inference and real-time performance on resource-constrained embedded devices.
3. *Model Selection*

   Considerations for selecting algorithms extend beyond accuracy to encompass performance, memory footprint, and power consumption, guiding developers in choosing the most suitable models for specific embedded applications. These include computational complexity, which determines the feasibility of running the algorithm within the device's processing capabilities, as well as memory needs, as embedded systems typically have limited memory available. Real-time constraints are also critical, as certain applications require algorithms to make decisions or predictions within strict time limits. Additionally, energy efficiency is a crucial consideration, as embedded devices often run on battery power and must conserve energy to prolong battery life. By navigating this intricate terrain of algorithm selection and optimization, developers can unlock the potential of machine learning in embedded systems while meeting the demanding requirements of real-world deployments.

1.4 MODEL COMPRESSION AND QUANTIZATION TECHNIQUES

1. *Importance of Optimization*

   Model compression and quantization are vital techniques for deploying machine learning models on embedded systems, where resources such as memory and computational power are limited. By reducing the size and complexity of models, compression and quantization enable efficient deployment on resource-constrained devices without sacrificing performance. This is particularly crucial for edge computing applications, where lightweight models are essential for real-time processing and low-latency inference. Moreover, smaller models require less memory and bandwidth, making them more suitable for deployment in IoT devices and other embedded systems with limited storage and communication capabilities. Ultimately, model compression and quantization play a pivotal role in enabling the widespread adoption of machine learning in embedded deployment, unlocking new possibilities for intelligent edge computing applications.

2. *Overview of Compression Techniques*

   Compression techniques such as pruning, quantization, and knowledge distillation are essential for reducing the size and complexity of machine learning models, making them more suitable for deployment on resource-constrained embedded systems. Pruning involves removing redundant connections or parameters from neural networks, thereby reducing model size and computational requirements while preserving accuracy. Quantization reduces the precision of model parameters from floating-point to fixed-point representation, resulting in smaller memory footprint and faster inference. Knowledge distillation transfers knowledge from a larger, complex model (teacher) to a smaller, simplified model (student) through training, enabling efficient inference with reduced computational cost. These techniques collectively enable efficient deployment of machine learning algorithms on embedded systems, facilitating real-time processing and low-latency inference in diverse applications.

3. *Case Studies*
   i. **Drowsiness Detection:** Reddy et al. (2017) proposed one deep learning model utilizing pruning and quantization for driver's drowsiness detection which is deployable on an embedded device. Their approach involves model compression techniques to reduce computational complexity and memory usage, enabling efficient deployment on resource-constrained embedded devices. The system aims to enhance driver safety by accurately detecting drowsiness in real time, leveraging the advantages of deep learning while addressing the constraints of embedded environments. The study underscores that the effectiveness of model compression on embedded devices demonstrates substantial improvements in efficiency, enabling applications such as real-time image classification on edge devices, smart sensors, and wearable devices.
   ii. **Intrusion Detection System:** Nguyen et al. (2022) developed a lightweight intrusion detection system (IDS) specifically designed for Internet of Things (IoT) devices using embedded machine learning techniques. The system addresses the critical need for security in IoT environments by leveraging a compact and efficient machine learning model that can be deployed on resource-constrained devices. The IDS employs techniques such as feature selection, model compression, and optimized algorithms to detect network anomalies and potential intrusions in real time. The resulting system demonstrates high accuracy in detecting intrusions while maintaining low computational and energy overhead, making it suitable for practical IoT deployments.

1.5 HARDWARE ACCELERATION FOR MACHINE LEARNING

1. *Overview of Hardware Acceleration Techniques*

   The optimization of machine learning inference on embedded systems often relies on hardware acceleration techniques, which harness specialized computing architectures to enhance performance and efficiency. Various types of hardware accelerators, including Graphics Processing Units (GPUs), Field-Programmable Gate Arrays (FPGAs), and dedicated AI chips, are instrumental in accelerating the execution of machine learning algorithms (Deng et al., 2020). GPUs excel in parallel processing tasks, making them well suited for neural network computations. FPGAs offer flexibility through re-configurability, enabling customized hardware implementations tailored to specific algorithms or applications. Dedicated AI chips, designed specifically for machine learning tasks, deliver high performance and energy efficiency. These hardware acceleration techniques expedite computation-intensive tasks, facilitating real-time processing and low-latency inference on resource-constrained embedded devices, thus unlocking the full potential of machine learning for intelligent edge computing applications.

2. *Types of Hardware Accelerators*

   Hardware accelerators, including GPUs, FPGAs, and dedicated AI chips, revolutionize machine learning on embedded systems by providing specialized architectures tailored to specific tasks. GPUs excel in parallel processing, utilizing multiple cores optimized for matrix operations crucial in neural network computations. Their hierarchical structure comprises streaming multiprocessors, each containing hundreds to thousands of processing units. FPGAs offer versatility through reconfigurable logic blocks interconnected by programmable routing resources, allowing custom hardware designs optimized for machine learning algorithms. Dedicated AI chips, such as TPUs and NPUs, feature specialized processing units designed for the efficient execution of neural network operations, offering superior performance and energy efficiency. By harnessing the unique capabilities of these hardware accelerators, embedded systems can achieve unprecedented levels of computational power and efficiency, paving the way for intelligent edge computing applications.

1.5.1 Optimization Strategies for Leveraging Hardware Accelerators to Improve Performance and Efficiency

Optimization strategies for leveraging hardware accelerators focus on harnessing the unique capabilities of specialized architectures to enhance the performance and efficiency of machine learning algorithms on embedded systems. These strategies encompass various techniques, including algorithm mapping, parallelization, and data movement optimization. Algorithm mapping involves mapping computational tasks onto hardware accelerators in a manner that maximizes parallelism and minimizes data movement, thereby reducing processing latency. Parallelization techniques exploit the parallel processing capabilities of hardware accelerators, distributing tasks across multiple cores or processing units to expedite computation. Data movement optimization minimizes the transfer of data between the main memory and hardware accelerators, reducing energy consumption and latency. Additionally, techniques such as model quantization and sparsity exploitation can further optimize hardware utilization by reducing the computational and memory requirements of machine learning algorithms.

Utilizing these optimization strategies empowers developers to fully leverage the capabilities of hardware accelerators, leading to substantial enhancements in both performance and energy efficiency for machine learning inference on embedded systems.

1.6 ENERGY-EFFICIENT MACHINE LEARNING

1. *Importance of Energy Efficiency in Embedded Systems*

   Energy efficiency holds paramount importance in embedded systems due to their inherent constraints and operational environments. These systems, often powered by batteries or operating in remote areas, necessitate optimal utilization of energy resources to prolong battery life and ensure continuous operation. Efficient energy consumption not only extends the device lifespan but also reduces operational costs and environmental impact. Moreover, in applications where power availability is limited or intermittent, such as IoT devices or remote sensors, energy efficiency becomes critical for uninterrupted functionality. By prioritizing energy-efficient design and optimization techniques, embedded systems can achieve prolonged uptime, enhanced reliability, and sustainable operation in diverse environments, thereby fulfilling their intended functions effectively.

2. *Optimization Techniques to Minimize Energy Consumption*

   Techniques for optimizing machine learning algorithms to minimize energy consumption are essential for enabling efficient operation of embedded systems with limited power resources. These techniques encompass various strategies aimed at reducing computational complexity, memory usage, and data movement, thereby mitigating energy consumption while maintaining performance. Model quantization reduces the precision of model parameters, typically from floating-point to fixed-point representation, resulting in smaller memory footprint and faster inference. Pruning techniques selectively remove redundant connections or parameters from neural networks, resulting in a more compact and efficient model. Furthermore, sparsity exploitation leverages the inherent sparsity in neural network weights to reduce computational workload and memory requirements (Vedder & Eaton, 2022). By applying these optimization techniques, developers can significantly reduce the energy footprint of machine learning algorithms on embedded systems, enabling prolonged battery life and sustainable operation in resource-constrained environments.

3. *Case Studies*

   i. **People Detection System:** A case study on energy-efficient people detection embedded system was presented in De Nardi & Monterio (2023). The system utilizes TinyML computer vision, i.e., amalgamation of ML, embedded systems, and IoT technology. The device was installed on the institution's premises, and it successfully provided real-time people count as well as energy consumption statistics. Involving solar power in the circuit design and turning off the components when not in use are two important strategies to increase the device's lifetime.

   ii. **Traffic Monitoring Systems:** Hoch & Kopetzky (2022) developed an energy-efficient traffic detection system using deep learning algorithms optimized for embedded IoT devices. The system leverages a compressed deep neural network (DNN) to perform real-time traffic detection and monitoring. Techniques such as model pruning, quantization, and lightweight architecture design were applied to reduce the computational complexity and energy consumption of the neural network. The optimized model was deployed on low-power embedded platforms, demonstrating the ability to accurately detect and classify traffic conditions with minimal energy usage. Field tests showed that the system could reliably monitor traffic flow and detect incidents in real time, making it suitable for smart city applications where energy efficiency and responsiveness are critical.

1.7 REAL-TIME CONSTRAINTS AND PERFORMANCE GUARANTEES

1. *Challenges in Meeting Real-Time Constraints*

   Meeting real-time constraints in embedded systems presents several challenges due to the inherent limitations of hardware resources and the complexity of the algorithms involved. One major challenge is ensuring the timely execution of tasks while meeting stringent deadlines, especially in applications requiring rapid decision-making or control. Additionally, the variability in processing times and unpredictable environmental factors further complicates real-time operation. Memory and computational constraints pose another challenge, as resource-intensive algorithms may exceed the available hardware resources, leading to delays or missed deadlines. Furthermore, achieving real-time performance often necessitates trade-offs between accuracy and speed, requiring careful optimization of algorithms and hardware. Ensuring deterministic behaviour and system reliability is crucial, as failures or delays in real-time tasks can have significant consequences in safety-critical applications. Addressing these challenges requires a holistic approach encompassing algorithmic optimizations, hardware acceleration, and robust system design to ensure the timely and predictable operation of embedded systems in diverse real-world scenarios.
2. *Ensuring Predictable Performance and Meeting Latency Requirements*

   Ensuring predictable performance and meeting latency requirements in embedded systems involves employing various techniques to manage and control task execution. One approach is through task scheduling, where real-time operating systems allocate CPU time to tasks based on their priority and deadline, ensuring timely execution. Additionally, techniques such as static analysis and worst-case execution time estimation help identify potential bottlenecks and predict the worst-case execution time of tasks, enabling developers to allocate resources accordingly. Hardware acceleration, including specialized processing units like GPUs or FPGAs, can expedite computation-intensive tasks, reducing latency and improving overall system performance. Moreover, employing efficient data structures and algorithms optimized for embedded systems can minimize processing overhead and memory usage, further enhancing predictability and meeting latency requirements. By combining these techniques, developers can design embedded systems capable of reliably meeting real-time constraints across diverse applications and environments.

1.8 SECURITY AND PRIVACY CONSIDERATIONS

1. *Security Threats and Vulnerabilities*

   Security threats and vulnerabilities in machine learning-enabled embedded systems pose significant risks to both data integrity and system reliability. Adversaries can exploit vulnerabilities in machine learning models to manipulate outputs or extract sensitive information, compromising the integrity of the system. Common attack vectors include adversarial examples, where maliciously crafted input data can deceive machine learning models into making incorrect predictions, and model inversion attacks, which attempt to reverse-engineer sensitive data from model outputs. Moreover, embedded systems are often deployed in uncontrolled environments, making them susceptible to physical attacks such as side-channel attacks or tampering. Ensuring the security of machine learning-enabled embedded systems requires robust defences such as secure model training, input validation, and encryption of sensitive data. Additionally, ongoing monitoring and updates are essential to detect and mitigate emerging threats, safeguarding both the integrity of the system and the privacy of user data.

2. *Techniques for Ensuring Data Privacy and Protecting against Adversarial Attacks*

   Securing embedded systems involves a multifaceted approach to safeguard sensitive information and mitigate potential threats. Secure model training techniques, such as federated learning or differential privacy, aim to protect sensitive data during the training process by aggregating model updates from multiple devices without exposing raw data. Additionally, input validation methods, including anomaly detection and input sanitization, help detect and mitigate adversarial attacks by identifying abnormal or malicious input data. Model robustness techniques, such as adversarial training or defensive distillation, fortify machine learning models against adversarial attacks by introducing perturbations during training to increase resilience to adversarial examples. Furthermore, encryption methods, such as homomorphic encryption (Natarajan & Dai, 2021) or secure multiparty computation, can be employed to protect data both in transit and at rest, ensuring privacy and confidentiality in machine learning-enabled embedded systems. By integrating these techniques into system design and implementation, developers can bolster data privacy and resilience against adversarial attacks, enhancing the security and trustworthiness of embedded systems in diverse applications.

3. *Regulatory Considerations for Secure Embedded Machine Learning*

   Regulatory considerations and compliance requirements for secure embedded machine learning are pivotal aspects that must be addressed to ensure the ethical and legal deployment of these technologies. With the increasing integration of machine learning into embedded systems across various domains, regulators are emphasizing the need for robust frameworks to protect user privacy, mitigate risks, and ensure accountability. Regulations such as the General Data Protection Regulation (GDPR) in the European Union and the Health Insurance Portability and Accountability Act (HIPAA) in the United States impose stringent requirements on the collection, processing, and storage of sensitive data, including data used for machine learning purposes. Compliance with these regulations necessitates transparent data handling practices, informed consent mechanisms, and measures to safeguard against data breaches or unauthorized access.

Furthermore, adherence to industry standards and best practices, along with ongoing monitoring and audits, is essential to ensure compliance with regulatory requirements and maintain the trust and confidence of stakeholders in the secure deployment of embedded machine learning systems.

1.9 CASE STUDIES AND APPLICATIONS

1. **Autonomous Driving Systems:** Peng et al. (2020) presented a case study on the popular Apollo ADS.[1] They thoroughly examined the model architecture behind the system. The communication between components was reviewed. The system has successive versions supporting cruise on roads safely, avoiding collisions with obstacles, stopping at traffic lights, changing lanes, responding to geo-fencing and low-speed areas, etc. These algorithms process sensor data from cameras, lidar, and radar to detect objects, recognize road signs, and navigate complex environments autonomously.

   The case study emphasized the critical role of real-time processing and decision-making in autonomous driving systems. It underscored the importance of safety-critical validation and testing processes to ensure the reliability and robustness of machine learning algorithms in automotive applications.
2. **Wearable Health Monitoring Devices:** A non-invasive diabetes monitoring system has been developed by Valenzuela et al. (2020). The proposed device captures the readings of the patient and sends to the cloud application. Applying predictive analysis, alerts are sent to relatives and doctors in case of any discrepancy. Many more such devices have been proposed that analyse physiological data such as heart rate, activity levels, and sleep patterns to provide personalized health insights and early detection of health issues.

   Such kinds of systems have the inherent challenges of ensuring data privacy and security in healthcare applications. In the implementation of health monitoring systems, regulatory compliance and transparent communication with users regarding data collection and analysis play a major role. These facilitate and ease elderly person's life.
3. **Smart Home Assistants:** Companies now integrate machine learning algorithms into smart home assistants for voice recognition and natural language processing. VeRO is one such kind of smart assistant for blind persons (Triyono et al., 2021). The system enabled users to control smart home devices, access information, and perform tasks using voice commands. Although these systems ease human life, the need for continuous improvement and adaptation of machine learning models based on user feedback and usage patterns is emphasized.

   These case studies highlight the diverse applications of machine learning on embedded systems across different domains and provide valuable insights into the challenges, best practices, and lessons learned for successful deployment.

1.10 FUTURE DIRECTIONS AND EMERGING TRENDS

1. *Emerging Trends in Machine Learning Algorithms for Embedded Systems*

   The evolving landscape of machine learning algorithms for embedded systems is revolutionizing intelligent edge computing, presenting innovative solutions to tackle the constraints of deploying AI in resource-limited environments. One prominent trend is the development of lightweight and energy-efficient algorithms optimized for embedded devices, such as edge-friendly deep learning models or sparse neural networks. These algorithms prioritize efficiency without compromising performance, enabling real-time processing and low power consumption in edge devices. Another trend is the integration of federated learning techniques, which allow distributed devices to collaboratively train machine learning models while preserving data privacy. Additionally, there is a growing focus on hardware-aware algorithm design, leveraging specialized accelerators like TPUs or FPGAs to optimize performance and energy efficiency. Moreover, advancements in transfer learning and continual learning are enabling embedded systems to adapt and improve over time, enhancing their ability to address dynamic and evolving environments. By embracing these emerging trends, embedded systems are poised to become more intelligent, responsive, and capable of delivering transformative applications across diverse domains.

2. *Potential Future Advancements and Challenges*

   The trajectory of machine learning advancements for embedded systems promises transformative strides in intelligent edge computing. Enhanced lightweight and energy-efficient algorithms, tailored for embedded devices, offer optimized performance and scalability while conserving resources. The adoption of federated learning methodologies stands as a promising avenue, fostering collaborative model training across decentralized devices while preserving data privacy. Furthermore, ongoing advancements in hardware acceleration, inclusive of specialized AI chips and neuromorphic computing, present opportunities for further optimization and efficiency gains. Concurrently, the convergence of edge computing with 5G networks and edge-to-cloud architectures unfolds new vistas for real-time processing and distributed intelligence. Yet, amidst these strides, challenges persist, spanning robust security and privacy safeguards, interoperability concerns, and the intricacies of managing distributed systems. Overcoming these hurdles mandates interdisciplinary collaboration, pioneering research, and unified industry efforts to fully harness the potential of machine learning in embedded systems and propel the evolution of intelligent edge computing applications.

3. *Research Directions and Opportunities for Further Exploration*

   One key area of interest lies in the development of novel algorithms tailored specifically for edge computing environments, emphasizing resource efficiency, scalability, and adaptability to dynamic and heterogeneous networks. Exploring techniques for edge-to-edge communication and collaboration among distributed devices presents opportunities to enhance data sharing, model training, and decision-making processes in decentralized systems. Moreover, investigating the integration of machine learning with emerging technologies such as blockchain and edge computing can unlock new possibilities for secure and transparent data sharing and governance. Additionally, advancing research in explainable AI and model interpretability is essential for building trust and understanding in machine learning-enabled embedded systems, particularly in safety-critical applications. Furthermore, exploring the potential of edge intelligence for real-time anomaly detection, predictive maintenance, and autonomous decision-making in diverse domains offers promising avenues for research and practical applications. Embracing these research directions and opportunities will pave the way for the next generation of intelligent edge computing solutions, addressing pressing challenges and unlocking new frontiers in embedded systems research.

1.11 CONCLUSION

In conclusion, this exploration of machine learning on embedded systems has revealed key insights into the evolving landscape of intelligent edge computing. The summary of key findings underscores the importance of optimizing algorithms for resource-constrained environments while addressing real-time constraints and ensuring data privacy and security. Looking to the future, the implications of these findings suggest a promising trajectory for the integration of machine learning into embedded systems, with potential applications spanning diverse domains such as IoT, automotive, healthcare, and consumer electronics. To capitalize on these opportunities, recommendations for researchers, practitioners, and policymakers include fostering interdisciplinary collaboration, prioritizing energy efficiency and performance optimization, and establishing regulatory frameworks to safeguard data privacy and security. By embracing these recommendations and building on the insights gained from this exploration, stakeholders can collectively drive innovation and unlock the full potential of machine learning in embedded systems, paving the way for a smarter, more connected, and more resilient future.

NOTE

1 https://github.com/ApolloAuto/apollo, accessed on 20/5/2024.

REFERENCES

Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick P., and D. S. Nikolopoulos, "Challenges and Opportunities in Edge Computing," *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, New York, NY, USA, 2016, pp. 20–26, doi: 10.1109/SmartCloud.2016.18.

Chen, Chunlei, Peng Zhang, Huixiang Zhang, Jiangyan Dai, Yugen Yi, Huihui Zhang, and Yonghui Zhang. "Deep learning on computational-resource-limited platforms: A survey." *Mobile Information Systems* 2020, no. 1 (2020): 8454327.

De Nardi, Adriel Monti, and Maxwell Eduardo Monteiro. "Evaluation of the energy viability of smart IoT sensors using TinyML for computer vision applications: A case study." In *IFIP International Internet of Things Conference*, pp. 18–33. Cham: Springer Nature Switzerland, 2023.

Deng, Lei, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. "Model compression and hardware acceleration for neural networks: A comprehensive survey." *Proceedings of the IEEE* 108, no. 4 (2020): 485–532.

Han, Song, Huizi Mao, and William J. Dally. "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding." arXiv preprint arXiv:1510.00149 (2015).

Hoch, Thomas, and Theodorich Kopetzky. "Energy-efficient Internet of Things solution for traffic monitoring." In Schirrer, A., Gratzer, A.L., Thormann, S., Jakubek, S., Neubauer, M., Schildorfer, W. (Eds.), *Energy-Efficient and Semi-Automated Truck Platooning: Research and Evaluation*, pp. 129–141. Cham: Springer International Publishing, 2022.

Natarajan, Deepika, and Wei Dai. "SEAL-embedded: A homomorphic encryption library for the Internet of Things." *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2021, no. 3 (2021): 756–779.

Nguyen, Xuan-Ha, Xuan-Duong Nguyen, Hoang-Hai Huynh, and Kim-Hung Le. "Realguard: A lightweight network intrusion detection system for IoT gateways." *Sensors* 22, no. 2 (2022): 432.

Peng, Zi, Jinqiu Yang, Tse-Hsun Chen, and Lei Ma. "A first look at the integration of machine learning models in complex autonomous driving systems: A case study on apollo." In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Virtual Event USA, pp. 1240–1250, 2020.

Reddy, Bhargava, Ye-Hoon Kim, Sojung Yun, Chanwon Seo, and Junik Jang. "Real-time driver drowsiness detection for embedded system using model compression of deep neural networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Honolulu, HI, USA, pp. 121–128, 2017.

Triyono, Liliek, Yudantoro, Tri Raharjo, Sukamto, Sukamto, and Hestinigsih, Idhawati. "VeRO: Smart home assistant for blind with voice recognition." *IOP Conference Series. Materials Science and Engineering* 1108, no. 1 (2021).

Valenzuela, Francisco, Armando García, Erica Ruiz, Mabel Vazquez, Joaquin Cortez, and Adolfo Espinoza. "An IoT-based glucose monitoring algorithm to prevent diabetes complications." *Applied Sciences* 10, no. 3 (2020): 921.

Vedder, Kyle, and Eric Eaton. "Sparse pointpillars: Maintaining and exploiting input sparsity to improve runtime on embedded systems." In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, pp. 2025–2031. IEEE, 2022.

Yang, Jie, Jinfan Xu, Xiaolei Zhang, Chiyu Wu, Tao Lin, and Yibin Ying. "Deep learning for vibrational spectral analysis: Recent progress and a practical guide." *Analytica Chimica Acta* 1081 (2019): 6–17.

2 Fusion of Edge Computing in AI-Enabled Embedded Technologies

R. Aishwarya and G. Mathivanan

2.1 INTRODUCTION

Artificial intelligence (AI) focuses on developing systems that act and think like humans and construct rational decisions. The establishment of AI dates back to the mid-20th century when Allen Newel and Herbert A. Simon along with J.C. Shaw initiated the “Logic Theorist”, marking the origination of symbolic AI to represent rules and knowledge for decision-making based on AI using symbols in early form. Eventually, the field progressed to include expert systems and neural networks. Frank Rosenblatt founded Perceptron in 1958. A neural network contains input layers along with randomized weights-based hidden layers and learning connection-based output layers. This growth deposits the preliminary work for more complex neural networks. In AI development around 1975–1985, the establishment of evolutionary algorithms such as genetic algorithm and swarm algorithm further hurled AI development.

As AI technology advanced, the hardware supporting it also made progress. Embedded systems saw improvements in memory and CPU capabilities moving from 4- to 64-bit processors while becoming more compact. The advent of Large-Scale Integration (LSI) and Very Large-Scale Integration (VLSI) technologies notably boosted computing power. Initially, basic algorithms were created, followed by pseudo-second-order algorithms. However, the introduction of neural networks posed challenges like the vanishing gradient issue. Despite algorithms taking a week to process, data advancements led to algorithms that could generate results within milliseconds. Although AI software demonstrated proficiency, integrating it seamlessly with adaptable hardware execution remained a hurdle, prompting the merging of AI into embedded systems known as embedded firmware. This progression gave rise to control systems where specialized software stored in flash memory chips controlled device functions. This integration has had an impact across applications, including efforts to address global warming. Edge computing, a method that processes data closer to its origin than solely relying on cloud servers, plays a vital role in reducing the environmental footprint of embedded systems. By reducing the amount of data exchanged and enhancing processing effectiveness, edge computing enables the integration of embedded AI leading to increased sustainability and efficiency. This method does not boost the functionality of AI-powered applications. Also, it tackles the rising worries regarding environmental preservation.

DOI: 10.1201/9781003481089-3

2.2 FUNDAMENTALS OF EDGE COMPUTING AND AI INTEGRATION

1. *Understanding Edge Computing Definition and Principles of Edge Computing*

   Edge computing is a middleware where it does computation, data processing, and storage closer to the end devices, which results in fast processes and lower latency. The main key principles of edge computing include real-time data analysis, distributed data processing, scalability, and resilience of resources at the edge of the device network for fast responsive applications.

   i. **Advantages and Challenges of Edge Computing**

      Edge computing mainly helps to reduce latency, increase privacy, and increase efficiency bandwidth. But even though edge computing provides lots of benefits, it also goes through many challenges such as data management when it comes to bulk of data, scalability when it comes to growing number of edge devices, network connectivity, and integration with cloud.

2. *The Relevance of AI Applications in Embedded Systems*

   Nowadays, many AI-based applications are racing to find relevance in embedded systems, transforming the potential of these devices. As I discussed before, embedded systems are specialized systems for computing specific tasks, which are now embraced with AI algorithms to improve their performance. These will lead to performing complex tasks in an efficient manner such as image recognition, speech recognition, and autonomous decision-making in real time. Exploiting AI with embedded systems can result in faster data, more informed decisions, and the ability to adapt to dynamic environments. The integration of AI with embedded systems brings many innovations across various fields such as healthcare, industrial automation, and electronics.

3. *Key Concepts for AI Integration in Embedded Systems*

   i. **An Overview of AI-Enabled Embedded Systems**

      The fusion of AI and embedded technologies is transforming various fields by immersing intelligent functionality into devices. These integrate AI algorithms with hardware and make real-time decision-making with advanced potential in constrained environments. From smart appliances to autonomous vehicles, AI-embedded systems enhance adaptability, efficiency, and autonomy. They resist machine learning, deep learning, and other AI methods to analyse, recognize, and make decisions without depending on external resources (LeCun et al. 2015). By locally doing data processing, the system has low latency, improves privacy, and operates offline. By facilitating AI-embedded systems, edge computing lowers the bandwidth and enhances response. As AI algorithms evolve and hardware capabilities advance, the potential applications of AI-enabled embedded systems continue to expand, promising transformative impacts across industries ranging from healthcare and manufacturing to smart cities and IoT ecosystems.

ii. **Machine Learning Algorithms and Neural Networks for Embedded Devices**

The principle of autonomous and intelligent functionalities in mannered environments is machine learning algorithms and neural networks in embedded devices. The ML algorithms make a wide range of methods that make devices to learn and train from data, make decisions, and recognize patterns. The ML algorithms typically operate with low processing power and memory when implemented in embedded devices. As a result, efficient ML algorithms such as support vector machine, decision tree, and k-nearest neighbours are chosen for deployment in embedded systems. These ML algorithms perform tasks such as regression, classification, and anomaly detection.

On the other hand, Neural networks represent a class of ML model that is inspired by the human brain structure and function. This Neural network model consists of interconnected layers of neurons. Each neuron performs simple computation and transfers its outputs to next layers. The tasks such as image recognition, time series prediction and natural language processing, and neural networks show remarkable performance. The main challenge is deploying the neural networks on embedded devices due to memory requirements and demands on computation.

To address these challenges, researchers and engineers developed convolution neural networks and recurrent neural networks. The quantization, model compression, and pruning are techniques used to decrease the size and complexity of neural network models and make them suitable for deployment on embedded devices. Integrating ML algorithms and neural networks into embedded devices allows them to perform intensive and intelligent tasks locally without any continuous connectivity to cloud-based resources. This kind of localization of intelligence provides various benefits such as privacy preservation, real-time responsiveness, and improved reliability with limited network access. Later by doing computation offloading from central servers to distributed edge devices, the overall system latency is reduced (Aishwarya and Mathivanan 2023). Also, it enables faster decision-making and provides a better user experience. As embedded devices are rapidly growing in various domains such as IoT, Healthcare, and industrial automation, the integration of ML algorithms and neural networks will play a very important role in opening their full potential for intelligent performance.

2.3 THE INTEGRATION OF EDGE COMPUTING AND AI IN EMBEDDED SYSTEMS

1. *Edge-Based AI Processing*

   In this section, we will look at how edge computing helps to enhance AI processing especially while integrated with embedded systems (Gupta et al. 2020).

2. *Grasp of Edge-Based AI Processing*

   i. **The Definition and Principles of Edge-Based AI Processing**

   The edge-based AI processing helps to conduct the algorithms of AI and its calculations, bringing them closer to the edge devices, which rely on a centralized cloud environment (Makhzani and Shrivastava 2020).

This kind of technique transfers the capability of computers to the area of generation of data and also helps to avoid the travelling of data for a long distance to reach the CPU. The processing of Edge-Based AI consists of below principles:

- **Proximity to the Data Source:**

  The edge processing significance is the need of AI computations close to the possible input source. This kind of proximity helps to reduce the data transmission time and associated delay.
- **Real-Time Responsiveness:**

  The main objective of edge-based AI processing is to increase the response time. It enables quicker decision-making, which is a very important factor for very low latency applications such as self-driving vehicles, IoT devices, and robotics due to local data processing.
- **Reduced Reliance on Cloud:**

  In contrast to traditional centralized cloud environments, AI and edge computing remove the need for a centralized cloud environment. This is mainly when internet connectivity is irregular or prevents efficient information transfer due to constraints of bandwidth.
- **Distributed Computing Paradigm:**

  Edge-based AI-embedded processing coheres to the principles of distributed computing. The burden of computing is distributed among edge devices, which include actuators, sensors, or other embedded devices. This will lead to a more scattered and scalable approach to AI.

3. *Method of Edge Computing in Reducing Latency and Improving Flow of Data in Embedded System*

   The edge computing reduces the latency and improves the data flow within the embedded system through vital mechanisms.

   i. **Local Processing:**

   Local processing on devices on edge crucially lowers the latency. This is situated in crisp contrast to normal cloud processing in which data should transfer over the network to a centralized system and then back, leading to delay.

   ii. **Data Filtering and Prioritization:**

   Based on the machine learning models, edge devices can filter the data and prioritize the data locally. This processing ensures that only applicable data is transferred to the cloud and minimizes overall data volume and latency.

   iii. **Real-Time Decision-Making:**

   Edge computing delegates real-time decision-making by accessing data at the point of origin. This is crucially important in many applications such as industrial automation, wherein short time decisions can have considerable results on operating safety and efficiency.