NJESR/January2023/Volume5/Issue1 E-ISSN-2582-5836
DOI-10.53571/NJESR.2023.5.1.1-11

Enabling Deep Learning On Embedded Systems For IoT Sensor Data Analytics

Subin Paul*1 Biju Peter *2

Head*1&*2

Department Of Computer Engineering

Government Polytechnic College

Vandiperiyar*1 Perumbavoor*2

(Received:2December2022/Revised:20December2022/Accepted:28December2022/Published:2January2023)

Abstract

An embedded system is an operating and controlling system that has been programmed to perform a specific task within a larger mechanical or electrical system. Typically, embedded systems have real-time computing constraints. It is embedded as part of a complete device, which frequently includes mechanical and hardware components. Many devices that are used every day are controlled by embedded systems[3]. Ninety-eight percent of all microprocessors are made as embedded system components. When compared to general-purpose counterparts, typical embedded computers have characteristics like low power consumption, small size, rugged operating ranges, and low cost per unit. This comes at the expense of limited processing resources, making programming and interacting with them significantly more challenging. However, one can optimizely manage the resources that are available at the unit and network levels as well as provide augmented functions that are well beyond those that are currently available by constructing intelligence mechanisms on top of the hardware, utilizing possible existing sensors, and utilizing the existence of a network of embedded units. For instance, embedded systems' power consumption can be managed with intelligent methods. Present day installed frameworks are much of the time in light of microcontrollers (for example Computer processors with coordinated memory or fringe interfaces)[7] yet standard microchips (involving outer chips for memory and fringe interface circuits) are likewise normal, particularly in more-complex frameworks. Regardless, the processor(s) utilized might be types going from universally useful to those worked in specific class of calculations, or even specially crafted for the current application. The digital signal processor (DSP) is a typical member of the standard class of dedicated processors. The embedded system can be optimized by design engineers to reduce the product's size and cost while increasing its reliability and performance due to its

specialized nature. Because of economies of scale, some embedded systems are produced in large quantities.

Embedded hardware platforms are being used more and more to make distributed IoT data analytics easier. Based on the limitations of embedded hardware platforms, it is necessary to optimize the execution of computation-intensive data analysis techniques like deep learning algorithms. The optimization strategies for deep learning approaches are summed up and the advantages and disadvantages of enabling deep learning on embedded hardware are discussed in this paper.

Keywords-Deep Learning, Internet Of Things, Embedded Systems, Sensor Data Analytics

Introduction

The amount of data generated by IoT devices has significantly increased in recent years. By 2020, Cisco predicts that 50 billion Internet of Things nodes will be connected to the internet[1]. The vast quantities of data generated by IoT sensors are not suitable for centralized cloud-based data analytics. The cloud integration is also more ambitious due to the heterogeneity of IoT devices, low latency response requirements, and mobility of IoT nodes. As a result, new intermediate platforms are required to facilitate real-time IoT sensor data analytics. As an answer, mist figuring and edge registering ideal models are acquainted with give the computational foundation nearer to the IoT gadgets. Decentralized data analytics greatly benefits from this paradigm shift. Utilizing high-performance devices throughout the hierarchy of data analytics is not economically feasible. Typically, low-power embedded hardware is used on platforms that are closer to IoT nodes. Fig. The hardware distribution and the IoT data analytics hierarchy are outlined in 1. The application-specific requirements and constraints must guide the selection of hardware and the distribution of data analytics. Within milliseconds, an application must respond to an external event in realtime (RT). It is anticipated that such time-sensitive data analytics will be carried out within the device itself or within an embedded hardware platform that is directly connected to the IoT sensor. This calculation stage is characterized as edge calculation foundation as meant in the lower layer of Fig. 1. The fog computing infrastructure, which is the second level of the hierarchy, is ideal for applications that require latency in the seconds to minutes range and are near real-time (NRT). Low-latency networks like local area networks (LANs) connect these fog nodes to the IoT nodes. They can be embedded hardware, networking devices, or medium-scale computing platforms. The cloud infrastructure's top layer is high-performance computing capability that can be used for business intelligence and transactional and historical data analytics applications. These days, AI based methods are vigorously utilized in IoT information examination for design acknowledgment, issue ID, condition checking, anticipating, and so on. Using an input dataset, machine learning algorithms statistically learn the knowledge that corresponds to the application domain. Training is the term for this process. After being trained, models of machine learning can mimic human brain function. Inference is the process of carrying out tasks with the knowledge that has been acquired. In the field of machine learning, recent developments like deep learning necessitate a large data set and specialized hardware accelerators for training. There are a variety of computational requirements for various aspects of machine learning algorithms.

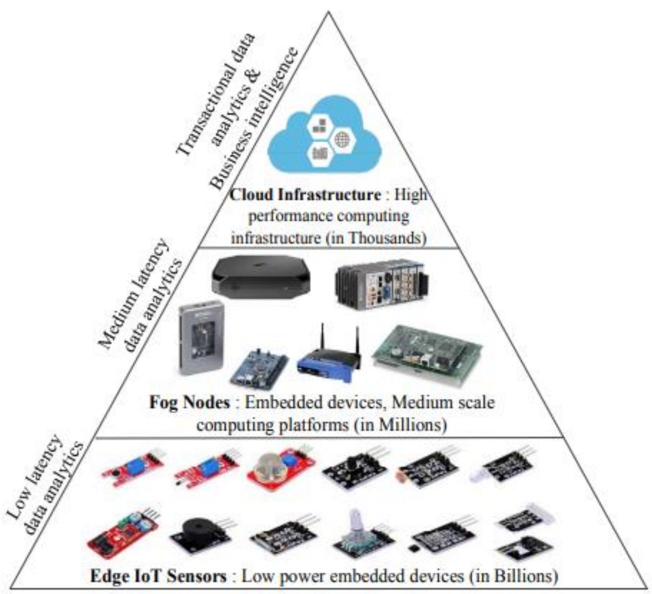

Figure 1. IoT Sensor Data Analytics Hierarchy & Hardware Distribution

The purpose of this paper is to draw attention to the difficulties and opportunities associated with deploying deep learning algorithms on top of embedded hardware at the edge or fog, as opposed to in a cloud server. This deployment offers significant advantages like improved data privacy and security, independence from connectivity, and lower latency. However, the majority of embedded devices must adhere to strict energy, processing, and memory requirements. To enable the effective processing of machine learning algorithms on embedded systems, it is necessary to carefully examine the effects of each constraint as well as the opportunities for improvement.

What is Embedded System

It's hard to give an accurate definition of embedded systems. To put it more succinctly, any and all computer systems other than a general-purpose computer are embedded technologies.

A system is a method of working, organizing, or carrying out one or more tasks in accordance with a predetermined program, plan, or set of rules. To put it another way, a structure in which all of the units come together and work together according to a plan or program. An embedded system is one in which software is embedded into the hardware, making it dedicated to a particular application, product, or component of a larger system. In order to control electromechanical equipment that may be a part of a larger system, it processes a predetermined set of instructions (not a computer with a keyboard, display, etc.).

Embedded systems are devices that are used to control, monitor, or assist in the operation of plant, machinery, or equipment, according to a general-purpose definition. Implanted mirrors the way that they are a basic piece of the framework. They may be so ingrained in a situation that a casual observer may not even notice them.

Embedded System

The foundation of the electronics industry is embedded systems. The purpose of embedded systems is specific. A device that incorporates both hardware and software is called an embedded system. It requires a microprocessor or microcontroller, depending on the system. The majority of embedded systems fall into two broad categories.

A. Based on requirements for performance and functionality. based on performance and functional requirements, which are further subdivided into four categories—real time, network, mobile, and standalone. Digital cameras, wireless MP3 players, and time- or deadline-based systems are examples of performance- and functional-based embedded systems.

B. In light of execution of microcontroller In view of execution of miniature regulator it very well may be sorted in three kinds called limited scope, medium scale and refined implanted framework. Regulator, temperature estimation framework, ATM and so on. are an illustration of a microcontroller.

The embedded system user can generate some raw data from the aforementioned source, but they are unable to make any decisions based on that data. Therefore, some data analytics methods are utilized to extract a meaningful insight from these data in order to make this system more decisive.

Characteristics

1. Single-function, application-specific embedded systems The application is beforehand known, and the programs are run repeatedly.

2. For embedded systems, efficiency is of the utmost importance. Their energy, code size, execution time, weight and dimensions, and cost are all optimized.

3. Typically, embedded systems are built to meet real-time constraints; A real-time system responds to operator or controlled object stimuli within the environment's time constraints. In real-time systems, correct responses that arrive too late or even too early are incorrect.

4. Typically reactive systems, embedded systems frequently interact (sense, manipulate, and communicate) with the outside world through sensors and actuators; A reactive system operates at a rate that is determined by the environment and is constantly in contact with it.

5. The user interface is typically minimal or absent.

Fraunhofer researchers will demonstrate "Smart Farming"—the safe and secure operation of the interaction of machines in cyber-physical systems—at Embedded World 2013 (Nuremberg, February 26-28).

Agriculture is under pressure from climate change, population growth, and an increasing scarcity of resources. From the smallest amount of land, farmers must harvest as much as possible. Until now, the industry has dealt with this problem by coming up with new ideas in specific areas; For instance, intelligent systems regulate engines to conserve fuel.

The field work can be performed automatically by farming equipment with the assistance of satellites and sensor technology; They are able to spread seed, fertilizer, and pesticides on the land more effectively as a result of doing so. However, optimization is progressively reaching its limits. The following stage is to organize these singular frameworks into digital actual creation frameworks. These electronically map the entire process, from the farm computer to the harvesting operation, significantly improving pest identification efficiency and quality.

Population growth, a lack of resources, and climate change are putting a great deal of pressure on agriculture. Nowadays, farmers are required to harvest as much as possible from as little land as possible. As a result, something remarkable like an embedded system is needed to support this field. Farming is complicated in a number of ways because farmers need to know a lot about the weather and be able to change the way they farm to fit the weather. Farming methods even change depending on the state of the soil, so farmers benefit greatly from computational assistance. At Implanted World (corner 228 in Lobby B5) analysts from the Fraunhofer

Organization for can exhibit how farming will profit from arranged frameworks later on. The Experimental Software Engineering IESE in Kaiserslautern uses an agricultural diorama to move a piece of farm equipment across a plot of land for their exhibit. Two tablet computers are at the farmland's edge. They can be used by visitors to the trade show to start the farm equipment's automated control. The model farm is suspended above by six screens. They demonstrate how software manages functionality by displaying the processes that lie behind the automation. The purpose of the visualization is to assist visitors in comprehending the difficulties and solutions of interconnecting embedded and IT systems. Farmers can increase their farming productivity through intelligent networking. Simple task management for agricultural machinery is not the only application of agricultural operations networking. Geodata and weather data are being offered by sensor technology and data service providers, in addition to seed and fertilizer manufacturers; Smartphone apps are another way that scientists have developed a precision farming method that optimizes all aspects of agriculture to assist farmers. The goal of this procedure is to minimize input while maximizing output. KAU and ICFOSS are currently using this farming method in Kerala. They want to set up smart agriculture, which would use satellite data to provide actual soil data and recommend the best farming method based on that data.

Additionally, the precision farming process aims to provide farmers with market data, value-added options, and post-harvest guidance. In the future, this system also intends to solve labor issues by developing robotic farm equipment like sensor-based sprinklers that would carry out the same farming tasks as laborers currently do. Precision farming has gained a lot of importance in a number of countries, the most recent of which is Holland. Driverless tractors with Real Time Kinematic and GPS are currently being developed in this nation, which will prove to be efficient and cost-effective for large farmlands.

Emerging Opportunities For Embedded Data Processing In IOT

Through digital transformation, the majority of industrial operations from various application domains in this IoT era are becoming automated. Data is created and consumed at the network's edges in these applications. Additionally, the constraints and requirements for applications may differ depending on the domain. The following are examples of application domains that require embedded data processing to fulfill their functional requirements.

Computer Vision Applications Mechanical actuations are initiated by data analytics in computer vision applications like robotics and autonomous vehicle navigation. The surrounding environment is captured through the use of cameras and other sensors. The caught video transfer should be handled to distinguish the encompassing articles, traffic conditions and close by vehicles[2]. For cloud-based data analytics, the vehicle must continuously stream video to the cloud. This calls for a lot of bandwidth on the network. Additionally, vehicles' mobility may impede the creation of a continuous connection between the cloud and the vehicle. In addition, serious accidents can occur if nearby vehicles are not detected immediately. Due to the communication delay, cloud-based data analytics was unable to achieve the anticipated lower latency in such circumstances. The solution is to put the infrastructure for processing data closer to the vehicle. In order to achieve the best results, data processing in vehicles must be facilitated by embedded hardware because of their mobility.

Applications for the Smart Home In smart homes, household appliances like a television, air conditioner, and smart bulb are connected to the network for automation. The data generated by the home-based sensors are used to guide these devices' operations[3]. The data consumed by the devices in the same house are continuously reported by the sensors, including temperature readings, room occupancy, and human activities. Due to the requirement for bidirectional data transmission between the cloud and the home, the cloud computing paradigm is not the best option for these situations. For applications in which all of the sensor data are processed at an embedded home edge gateway, the edge computation architecture is identical. Device management is more flexible, privacy is protected, and bandwidth is saved thanks to this.

Applications for Smart Monitoring The Internet of Things (IoT) is frequently utilized in applications for smart monitoring to digitally transform the physical world into machine-readable data. Asset obliged sensors and implanted information obtaining frameworks (DAQ) are utilized to detect the natural circumstances. The systems, such as those for monitoring habitat, faulty condition, and natural disasters, use the captured data. Each checking applications requires various sorts of natural circumstances at various time goals. Because it directly affects the size of the data that is generated, the time resolution of the data plays a crucial role in these applications. In real time, larger amounts of data are generated by higher sampling rates[4]. Additionally, low latency data analytics are required for some monitoring scenarios, such as natural disaster monitoring. In addition, these monitoring applications necessitate the correlation of sensor data from different locations. In contrast to the centralized cloud, the data processing infrastructure also needs to be distributed in order to process the data locally and extract key information like abnormalities. Instead of sending a lot of raw data to the cloud, this deployment makes it easier for adjacent nodes to transform key features to detect an event. For local data processing, embedded hardware with limited resources is typically attached to the DAQ systems. As a result, the challenges posed by cutting-edge IoT applications from various fields can be addressed by embedded data processing at the network edges. The primary benefits of decentralized embedded data processing include lower latency[5] bandwidth efficiency, privacy protection, mobility support, and geographical distribution.

Overview Of Deep Learning Techniques & Computational Requirements

Deep learning is a subfield of machine learning as a whole. The methods used in machine learning are based on how the human brain works. Brain networks are the most well known and generally utilized AI approach which is gotten from interfacing the vitally computational component of a human cerebrum called neuron. To scale the input signal to a neuron, each connection to the neurons has a weight-based scaling factor. Through its associated activation function, a neuron is able to produce the input signal's output. A neural network typically consists of layers of neurons. On the basis of the aforementioned building blocks, structures that are more computationally complex are now constructed. Deep learning is now a new area of machine learning as a result. Because of its high accuracy, deep learning methods are increasingly being used as solutions for a wide range of artificial intelligence (AI) tasks, including speech recognition, robotics, and image recognition. Deep neural networks, in contrast to conventional neural networks, have more layers—ranging from five to thousands—and are able to extract higher-level features from raw input. In addition, the configurations and arrangement of the layers are significantly different from those of conventional neural networks. In deep learning architectures, there are three main types of connections that are used a lot: as depicted in Fig., fully connected, sparsely connected, and recurrently connected. All input activations contribute to the output in fully connected layers. In contrast, the input's receptive field is only taken into consideration by sparsely connected layers. In deep learning architectures, the most common sparsely connected layer is the convolutional layer. In networks with internal memory, recurrent connections are frequently used to store intermediate results for later processing. Pooling and normalization are two other common functionalities that have been utilized frequently in deep learning. Down sampling and dimensionality reduction both benefit from pooling. The network's learn ability and training speed are both improved through the use of various normalization techniques. Since the input data for a network is typically represented as a matrix with varying dimensionalities, all of these implementations necessitate manipulating matrices. The fundamental operations for obtaining weighted sums of the input matrix are additions and multiplications. To produce an output, popular deep learning structures like AlexNet[6], GoogLeNet[7], and ResNet[8] require 724 million, 1.43 billion, and 3.9 billion, respectively. With embedded devices, it is impossible to meet such computational requirements. In the beginning, deep learning structures are created with the intention of maximizing task accuracy rather than taking into account the complexity of implementation. In order to make deep learning approaches more applicable to real-world situations, recent studies have argued that deep learning models need to be constructed with the goal of maximizing accuracy while simultaneously reducing the amount of energy required and the computational complexity involved.

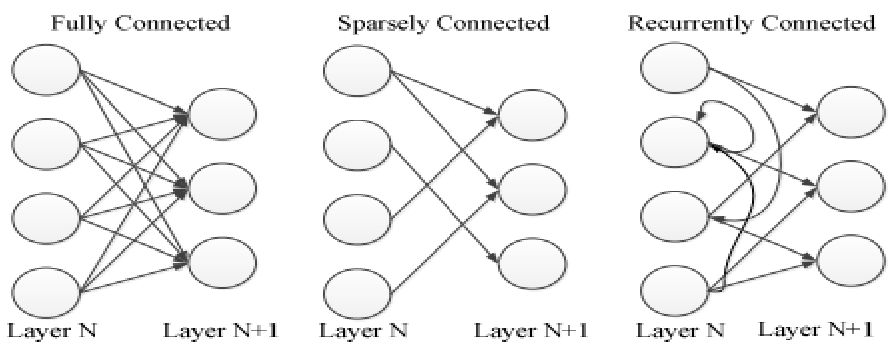

Figure 2. Types Of Connections In Deep Neural Networks

Conclusion

The impact and future applications of embedded systems can be seen in this paper's conclusion. As the authors of this paper, we intend to make extensive use of data analytics to extend our study of embedded systems in manufacturing sectors. We have discussed embedded systems today and in the future in this paper, and the second half also outlines the fundamentals of data analytics. Since data is automatically captured, it is anticipated that the data generated by embedded systems will be more pure. When data is captured by machine and end-less processes are incorporated, the speed of the data will be very high. If a well-designed and sophisticated embedded system captures this enormous amount of data, it may be demonstrated to be an essential input for data analytics. This paper discusses some optimization methods that can be applied to real-time Internet of Things applications and provides a summary of the advantages and disadvantages of incorporating deep learning-based approaches into embedded hardware.

References

[1]. Sirojan, T., Phung, T., & Ambikairajah, E. (2018, December). Enabling deep learning on embedded systems for iot sensor data analytics: Opportunities and challenges. In 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS) (pp. 1-5). IEEE.

[2]. Lee, J., Stanley, M., Spanias, A., & Tepedelenlioglu, C. (2016, December). Integrating machine learning in embedded sensor systems for Internet-of-Things applications. In 2016 IEEE international symposium on signal processing and information technology (ISSPIT) (pp. 290-294). IEEE.

[3]. Saba, T., Rehman, A., Haseeb, K., Bahaj, S. A., & Jeon, G. (2022). Energy-Efficient Edge Optimization Embedded System Using Graph Theory with 2-Tiered Security. Electronics, 11(18), 2942.

[4]. Ouafiq, E. M., Saadane, R., & Chehri, A. (2022). Data Management and Integration of Low Power Consumption Embedded Devices IoT for Transforming Smart Agriculture into Actionable Knowledge. Agriculture, 12(3), 329.

[5]. Kopetz, H., & Steiner, W. (2022). Internet of things. In Real-time systems (pp. 325-341). Springer, Cham.

[6]. Hudson, J. (2022). Internet of Medical Things-driven Remote Monitoring Systems, Big Healthcare Data Analytics, and Wireless Body Area Networks in COVID-19 Detection and Diagnosis. American Journal of Medical Research, 9(1), 81-96.

[7]. Lachance, G. P., Boisselier, É., Boukadoum, M., & Miled, A. (2022). Towards an advanced neurotechnological system: colorimetric sensing with a novel grism-based spectrometer, functionalized gold nanoparticles and a heterogeneous embedded system. Philosophical Transactions of the Royal Society A, 380(2228), 20210016.

[8]. Huang, W., Zhang, Y., & Wan, S. (2022). A sorting fuzzy min-max model in an embedded system for atrial fibrillation detection. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 18(2s), 1-18.

[9]. Varshney, A., Yan, W., & Dutta, P. (2022, June). Judo: addressing the energy asymmetry of wireless embedded systems through tunnel diode based wireless transmitters. In Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services (pp. 273-286).

[10]. Saddik, A., Latif, R., El Ouardi, A., Elhoseny, M., & Khelifi, A. (2022). Computer development based embedded systems in precision agriculture: tools and application. Acta Agriculturae Scandinavica, Section B—Soil & Plant Science, 72(1), 589-611.

[11]. Dey, S., Dasgupta, P., & Chakrabarti, P. P. (2022). SymDNN: Simple & Effective Adversarial Robustness for Embedded Systems. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3599-3609).

[12]. Safari, S., Ansari, M., Khdr, H., Gohari-Nazari, P., Yari-Karin, S., Yeganeh-Khaksar, A., & Henkel, J. (2022). A Survey of Fault-Tolerance Techniques for Embedded Systems From the Perspective of Power, Energy, and Thermal Issues. IEEE Access, 10, 12229-12251.

[13]. Yun, J., Rustamov, F., Kim, J., & Shin, Y. (2022). Fuzzing of Embedded Systems: A Survey. ACM Computing Surveys (CSUR).