Seminar Cloud Computing

From Concept to Production: Deploying TinyML in Industry

Trung Nguyen
Technische Universität München

November 2024

Abstract

In recent years, Artificial Intelligence (AI) and Machine Learning (ML) have received tremendous amount of attention in both industry and research world. However, conventional Machine Learning demands high computing capability which limits its usage to only larger computing units. The pardigm shift to Tiny Machine Learning (TinyML) is revolutionizing industries by enabling the deployment of machine learning models on low-power, resource-constrained devices. Being one of the most rapid developing field of Machine Learning, TinyML promises to benifits multiple industries. However, building a production-ready tinyML system poses different unique challenges. In this paper, we explore the key obstacles faced when developing and deploying TinyML models in production environments, including model optimization, hardware limitations, software integration, and maintaining performance in real-world conditions. Additionally, we present real-world use cases of TinyML in industrial settings, showcasing its transformative impact. We also discuss practical approaches and strategies presented by recent researches [8] to overcome these challenges, providing insights into how TinyML systems can be successfully scaled and implemented in production.

1 Introduction

Traditional Machine Learning Models, especially Deep Learning Models typically require substantial amount of computing capability to operate effectively. These models are often trained on powerful Graphics Processing Units (GPUs) and produce large models ranging from tens or hundreds of gigabytes (GB) down to smaller models in the range of 10 to 100 megabytes (MB). However, the memory requirements during runtime for these models far exceed what microcontrollers (MCUs) can handle. The pardigm shift to TinyML is driven by the prevailing number of Microcontroller Units (MCU) currently circulating in the industry. According to a recent report [2, 1], as of 2021, around 31 billion MCUs were shipped worldwide annually. The MCU market size is projected to increase in upcoming years [2]. This creates a big incentive for researchers and industry players to put effort into developing the technology further. TinyML aims to enable data processing or inferencing directly on embedded systems, particularly on Internet Of Things (IOT), instead of streaming to the cloud. TinyML model can run on energy-, and memory-constraint devices by limiting communication overhead with better suited architecture design and applying different compressing techniques such as: quantization and pruning. The advancement of tinyML has positively influenced multiple industries/sectors such as: industrial IOT [7], healthcare [3], agriculture [11], IOT in smart-city [5, 7].

Although various organizations have been putting efforts into ML project, merely 13% of them took off to production. In this paper, we aim to highlight what is required to transition a TinyML project into production and generate value for the industry. In Chapter 2, we provide an overview of the key concepts, techniques, and the development pipeline of TinyML. In Chapter 3, drawing on recent research, we explore the challenges of this development process and present approaches to address them. Next, in Chapter 4, we discuss real-world use cases of TinyML in industrial environments. Finally, in Chapter 5, we examine the future prospects of TinyML.

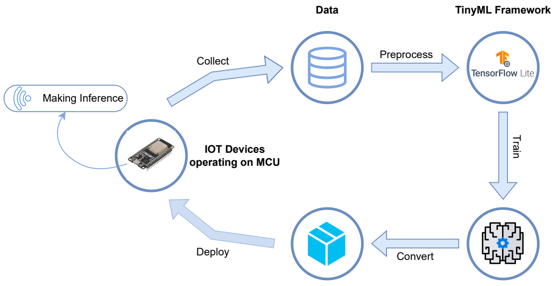

Figure 1: The caption explaining what can be seen in the image/figure. Readers often read captions first if they do not have much time. Thus, it is important to find a good short explanation.

2 TinyML Overview

2.1 Key Concepts and Techniques

Pruning

Quantization

With the TinyML model running on edge, the data is stored, processed and analysed internally rather than at an external server or cloud.

2.2 TinyML pipeline

Data Collection: • The workflow starts with collecting data from an IoT device. This data serves as the input for training machine learning models. The IoT device can gather various types of data depending on the application, such as sensor data in smart homes or environmental monitoring.

Preprocessing: • After collecting data, it moves to the preprocessing stage. Here, the data is cleaned and prepared for model training. Preprocessing can involve tasks such as normalization, handling missing values, or feature extraction to improve the quality of the input data.

Model Training: • The preprocessed data is fed into an ML framework, where a machine learning model is trained. This stage involves using machine learning algorithms to find patterns in the data, building a predictive or analytical model that can generalize from the data.

Model Conversion: • Once the model is trained, it is converted into a format suitable for deployment on resource-constrained devices, like microcontrollers. This step might involve techniques such as quantization (reducing the precision of model weights) or pruning (removing unnecessary parts of the model) to reduce the model's size and computation needs.

Model Packaging: • After conversion, the model is packaged into a deployable form. This includes bundling the model with any necessary runtime components to allow it to be executed efficiently on the target device.

Deployment: • The next step involves deploying the packaged model onto an IoT device. This means transferring the model to the device, setting it up for real-time inference or operation in the field.

Inference: • Finally, the deployed model performs inference on the IoT device. Inference refers to the process of using the model to make predictions or decisions based on new data collected by the IoT device. The model operates locally on the device without needing continuous cloud connectivity, enabling real-time decision-making at the edge.

3 Enablements of TinyML in Industrial Setting

> "I think there is a world market for maybe five computers." (T.J. Watson, IBM, 1943)

The rest of the work (especially all the regular text) must be written/phrased by you. If you write about some results or fact stated in another paper, you should refer to it. The 'Analytical Engine" — a mechanical calculation machine — created by Charles Babbage in the year 1838 was based on the decimal system [5, 6, 4, 9, 10].

4 Use Cases of TinyML in Industrial Setting

Enumerations using bullet points:

- Agriculture
- Environmental Monitoring
- Industrial predictive maintenance
- Edge AI and Autonomous Systems

5 Future of TinyML

6 Conclusion

References

[1] Microcontroller market research, 2030.

[2] Microcontroller Market Size, Share & Trends By 2034.

[3] Mamta Bhamare, Pradnya V. Kulkarni, Rashmi Rane, Sarika Bobde, and Ruhi Patankar. Chapter 14 - TinyML applications and use cases for healthcare. pages 331–353, January 2024.

[4] Miguel de Prado, Manuele Rusci, Romain Donze, Alessandro Capotondi, Serge Monnerat, Luca Benini And, and Nuria Pazos. Robustifying the Deployment of tinyML Models for Autonomous mini-vehicles, July 2020.

[5] Dina Hussein, Dina Ibrahim, and Norah Alajlan. Original Research Article TinyML: Adopting tiny machine learning in smart cities. *Journal of Autonomous Intelligence*, 7:1–14, January 2024.

[6] Aditya Jyoti Paul, Puranjay Mohan, and Stuti Sehgal. Rethinking Generalization in American Sign Language Prediction for Edge Devices with Extremely Low Memory Footprint, February 2021. arXiv:2011.13741.

[7] Partha Pratim Ray. A review on TinyML: State-of-the-art and prospects. *Journal of King Saud University - Computer and Information Sciences*, 34(4):1595–1623, April 2022.

[8] Haoyu Ren, Darko Anicic, and Thomas Runkler. TinyOL: TinyML with Online-Learning on Microcontrollers. April 2021. arXiv:2103.08295.

[9] Haoyu Ren, Darko Anicic, and Thomas A. Runkler. The synergy of complex event processing and tiny machine learning in industrial IoT. In *Proceedings of the 15th ACM International Conference on Distributed and Event-based Systems*, DEBS '21, pages 126–135, New York, NY, USA, June 2021. Association for Computing Machinery.

[10] A. Navaas Roshan, B. Gokulapriyan, C. Siddarth, and Priyanka Kokil. Adaptive Traffic Control With TinyML. In *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pages 451–455, March 2021.

[11] Vasileios Tsoukas, Anargyros Gkogkidis, and Athanasios Kakarountas. A TinyML-based System For Smart Agriculture. pages 207–212, November 2022.