4-10-2021

Machine Learning Meets Internet of Things: From Theory to Practice

Bharath Sudharsan
Confirm SFI Research Centre for Smart Manufacturing, Data Science Institute, NUI Galway, Ireland, bharath.sudharsan@insight-centre.org

Pankesh Patel
University of South Carolina - Columbia, ppankesh@mailbox.sc.edu

Publication Info

Preprint version *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2021)*, 2021.

Machine Learning Meets Internet of Things: From Theory to Practice

Bharath Sudharsan* and Pankesh Patel§
*Confirm SFI Research Centre for Smart Manufacturing, Data Science Institute, NUI Galway, Ireland
bharath.sudharsan@insight-centre.org
§ Artificial Intelligence Institute, University of South Carolina, Columbia, USA
ppankesh@mailbox.sc.edu

ABSTRACT

Standalone execution of problem-solving Artificial Intelligence (AI) on IoT devices produces a higher level of autonomy and privacy. This is because the sensitive user data collected by the devices need not be transmitted to the cloud for inference. The chipsets used to design IoT devices are resource-constrained due to their limited memory footprint, fewer computation cores, and low clock speeds. These limitations constrain one from deploying and executing complex problem-solving AI (usually an ML model) on IoT devices. Since there is a high potential for building intelligent IoT devices, in this tutorial, we teach researchers and developers; (i) How to deep compress CNNs and efficiently deploy on resource-constrained devices; (ii) How to efficiently port and execute ranking, regression, and classification problems solving ML classifiers on IoT devices; (iii) How to create ML-based self-learning devices that can locally re-train themselves *on-the-fly* using the unseen real-world data.

KEYWORDS

Resource-Constrained Devices, Offline Inference, Edge Intelligence, Multi-component Model Optimization, Internet of Things.

1 INTRODUCTION

During the past few years, AI has been used as the principal approach to solving a variety of real-world problems across domains such as machine translation, voice localization, handwriting recognition, computer vision, etc. However, such problem-solving AI is usually an ML model with complex architecture that demands a higher order of computational power and memory than what is available on most IoT devices. The majority of IoT devices like smartwatches, smart plugs, HVAC controllers, etc., are powered by MCUs and small CPUs that are highly resource-constrained. Hence, they lack multiple cores, parallel execution units, no hardware support for floating-point operations (FLOPS), low clock speed, etc.

The resource-constrained nature of IoT devices restricts the standalone execution of large ML models, forcing the device manufacturer to follow an online approach of transmitting local data to the cloud for analytics and inference. In such online settings, the cost of device hardware increases due to the addition of wireless modules (4G or WiFi), also increases the cyber-security risks and creates privacy concerns (GDPR restrictions). Additionally, a continuous connection between edge devices and the cloud infrastructure needs to be maintained, leading to the requirement of high network bandwidth and traffic. Finally, such internet-dependent devices are not self-contained ubiquitous systems.

The focus of this tutorial will be deep optimization and efficient deployment of machine learning models on resource constrained IoT devices.

Learning Objective I. In this tutorial, to address the above first set of issues, we teach how to perform deep optimization and execution of large ML models on tiny IoT devices. The methods we teach is generic since it applies to CNNs trained using various datasets, and it is executable on a variety of MCUs and small CPUs-based IoT devices. In addition to CNNs, we also cover ML classifiers like the decision tree (DTs) and Random Forests (RFs). We teach the audience how to perform porting and standalone execution of anomalies detection classifier models on IoT devices. By realizing the tutorial gathered information, the audience can execute various use-case trained ML models on their products/devices to make them perform offline analytics, thus eliminating the dependency on the internet and cloud subscriptions.

In the real world, every new scene generates unseen data patterns. When an ML model deployed on IoT devices sees fresh patterns which were not previously exposed during the training phase, it will either not know how to react to that specific scenario or can lead to false or less accurate results. In order to achieve a *truly autonomous local intelligence* at the device level, the devices must have the ability to *self-learn and understand* the data patterns offline, with no dependency on users or cloud services. These devices should be capable of locally gathering knowledge incrementally during the training/learning phase and should self-learn. Thus, transforming edge devices into intelligent machines capable of learning and performing analytics in any given environment.

Learning Objective II. In this tutorial, in order to address the above second set of issues, we teach the audience how to create self-learning IoT devices. Thus they can make their IoT devices/products self-learn/train *on-the-fly*, using live IoT use-case data. Thus, devices can self-learn to perform analytics for any target IoT use cases.

2 AUDIENCE

The same people attending the ECML PDKK conference would be interested in our tutorial. We teach how to optimize the top-quality, large, ML framework-trained models in multiple aspects and comfortably execute them on limited memory devices. So this tutorial should be of interest to both the high-performance computing community (researchers who propose algorithms that get merged with ML frameworks like TensorFlow) and the TinyML practitioners (researchers who design resource-friendly algorithms for embedded systems). Overall the tutorial is expected to attract a large audience comprising of academia, industry, engineers, and other stakeholders in edge analytics, deep model optimization, IoT device design, etc. All levels of expertise are most welcome, as there is much information to be absorbed and a demo to be enjoyed.

3 AIMS AND LEARNING OBJECTIVES

Through this tutorial, we aim to interconnect the Software Engineering, Internet of Things, Machine Learning communities by bringing together the technology from each community in order to develop AI-enabled, self-learning, and offline inference performing autonomous IoT devices/products. The learning objectives of the tutorial are the following:

(1) For beginners, it will create an end-to-end understanding of how to optimize a given problem-solving ML model and deploy it on resource-constrained devices for offline analytics.
(2) Practitioners can improve the inference performance and compression levels of their use-case ML models, which they plan to deploy on their commercial IoT devices/products.
(3) Researchers, when benchmarking a ML model by executing it on real-world devices using the tutorial Part IV technique, can obtain superior experimental results in their papers.
(4) For ML experts, it will express the need of designing resource-friendly models in order to speed up the R & D phase (going from idea to product) of ML-powered IoT devices.

4 DETAILED DESCRIPTION

Our tutorial will consist of four parts/sessions:

PART I: ML for IoT Devices. The tutorial will start with an introduction aimed at setting a common context between ML ecosystem and IoT hardware for all participants. We will provide an understanding of commonly used terminologies by presenting a real-world ML-based IoT use case [8].

The **ML ecosystem** used to train and produce models for IoT devices is highly fragmented due to individualized implementations and programming frameworks (e.g., Tensorflow lite[1], Apache MXNet[2], Cloud[3]). So, at this initial stage, we pick and briefly explain the software components we use throughout the tutorial. To support and run the latest framework generate models, **IoT Hardware** manufacturers have designed low-power GPU-enabled devices that can perform ultra-fast edge-level inference. For instance, Intel's Movidius Neural Compute Stick (NCS) and Google's USB Coral Tensor Processing Unit (TPU) are popular co-processors that can be plugged into the USB port of an edge device. There are times when edge devices with such USB co-processors run out of resources (particularly RAM) while executing top-quality models. To address this, hardware manufacturers offer GPU-based boards dedicatedly designed to handle ML workloads and show top performance (both accuracy and speed). NVIDIA's Jetson Nano, Google Coral, LattePanda boards are successful examples.

Take Home Information. At the end of part I, the audience would have an idea of TensorFlow Lite, Arduino IDE, a basic ML model that we will later use during optimization, and the devices that we shall use to run the model.

PART II: Creating ML-based Self-learning IoT Devices. In this session, we initially present and demo three state-of-the-art frameworks; (i) Edge2Train [12] to enable onboard resource-friendly training of SVM models; (ii) Train++ [9] for ultra-fast incrementally onboard classifier training and inference; (iii) ML-MCU [11] to train up to a 50 class ML classifiers on a $ 3 device. Then leveraging the basics from part I and thus demonstrated frameworks, we will teach the audience how to create self-learning IoT devices.

Take Home Information. The audience would have learned how to make their IoT devices/products to self-learn/train *on-the-fly*, using live IoT use-case data. Thus, their devices can self-learn to perform analytics for any target IoT use cases.

PART III: Deep Optimizations of CNNs and Efficient Deployment on IoT Devices. We will start this session with a smart doorbell [8] motivation scenario. We show the audience how to utilize the multi-component ML model optimization work from [10, 13] to generate resource-friendly video analytics models (face, violence and theft detection, etc.) that can be deployed and executed on video doorbells, thus enabling even the resource-constrained doorbells to perform ultra-fast offline analytics without depending on internet and cloud subscriptions.

Take Home Information. The audience can apply the learned techniques on the models from a growing number of use-cases such as anomaly detection, predictive maintenance, robotics, voice recognition, machine vision, etc., to enable standalone device-level execution. Thus, we believe this part of the tutorial opens future avenues for a broad-spectrum of applied research works.

Part IV: Porting and Execution of ML Classifiers on IoT Devices. We will start this session by introducing how Decision Trees (DT) and Random Forest (RF) classifiers can be used in an IoT setting to solve ranking, regression, and classification problems locally at the device level. Then we present and demo the sklearn-porter [5], m2cgen [2], emlearn [1] to show how to efficiently port and execute anomalies detection classifier models on IoT devices.

Take Home Information. The audience can use the explained generic end-to-end method to quickly port and execute various datasets trained ML classifiers on any of the resource-constrained devices of their choice/availability.

4.1 Tutorial Material

We will deliver the concepts using Powerpoint slides embedded with animations and small code snippets. During content delivery, the audience will be asked to perform small and quick exercises to make the tutorial interactive. We also will interleave live/recorded demonstrations throughout the tutorial to improve the audience's understanding and also to provide them opportunities to study technology in action. The tutorial will have the following four parts with hands-on exercises relevant to each session:

(1) **ML for IoT Devices** - 30 mins. Presenters: Pankesh Patel. Slides will be presented.
(2) **Creating ML-based Self-learning IoT Devices** - 60 mins. Presenters: Bharath Sudharsan & Pankesh Patel. Slides will be presented and followed by demos recorded illustrating our frameworks: Edge2Train, Train++ and ML-MCU.
(3) **Deep Optimizations of CNNs and Efficient Deployment on IoT Devices** - 60 mins. Presenters: Bharath Sudharsan & Pankesh Patel. Slides will be presented and followed by demos recorded illustrating our framework RCE-NN.

[1] https://www.tensorflow.org/lite
[2] https://mxnet.apache.org/
[3] https://aws.amazon.com/greengrass/ml/

(4) **Efficient Execution of ML Classifiers on IoT Devices** - 30 mins. Presenters: Bharath Sudharsan & Pankesh Patel. Slides will be presented, followed by demos recorded illustrating the process of porting and execution of ML classifiers on IoT devices.

The tutorial will wrap up with a focused discussion on specific learning of the audience and open questions.

4.2 Required Prior Knowledge

Since this tutorial makes use of concepts from both ML and IoT, the ideal preparation would be the basics of Arduino IDE, MCU boards, and basic ML models. Although our step-by-step tutorial will guide the audience on how to implement the tutorial covered technologies, the knowledge of programming languages such as Python, C/C++, and the set up of the Google Colab/Jupyter notebook would be beneficial for the hands-on session.

4.3 Length

We intend to deliver a half day (4 hours incl. one 30 minute break) tutorial.

4.4 Technical Requirements

Participants should install Arduino IDE in their laptop and download the Edge2Train [12], ML-MCU [11], Train++ [9], CNN_on_MCU [10] repositories (only a few MB in total).

5 PRESENTERS

Bharath Sudharsan. He is working towards his Ph.D. at the CONFIRM SFI Centre Research Centre for Smart Manufacturing, NUI Galway. His core research focuses on; (i) Designing multi-component sequences for deep optimization of ML models; (ii) Designing approaches for efficient execution of ML models on AIoT boards, small CPUs, and MCUs based IoT devices; (iii) Designing algorithms to create self-learning devices that can locally re-train themselves using the unseen real-world data. He has published papers in venues such as ICCPS, IoTDI, WF-IoT, IoT, PerCom, IEEE Internet Computing. He obtained his Masters from NUI Galway in Electronics and Computer Engineering. Prior to research, he was an Embedded System Engineer at Four Corners Technologies.
Homepage: https://bharathsudharsan.github.io/profile/

Pankesh Patel. Before joining Artificial Intelligence Institute, University of South Carolina, Dr. Pankesh Patel was Technology Consultant at Jupyter. He was hired at Jupyter, to develop AI- and Cloud-based "Intelligent Doorbell" products for the Australian market. Before joining these positions, he was a Senior Research Scientist at Fraunhofer USA and a Research Scientist in Industrial Software System (ISS) group at ABB Corporate Research-India. Both at Fraunhofer USA and ABB, he focused on the implementation of Industry 4.0 techniques and methodologies in commercial environments. His academic background and research work focus on building software development tools to easily develop applications in the cross-section of the Internet of Things/Industry 4.0, Artificial Intelligence, Edge, and Cloud Computing.

He is a winner of the prestigious Marie-Curie fellowship at SFI Confirm Centre for Smart Manufacturing, Data Science Institute, NUIG Galway, Ireland. In the past 7 years, he has published several research articles(40+ publications, h-index: 17) in prestigious conferences and delivered several talks as a keynote and invited speaker. He finished his Ph.D. in Computer Science (with a Très Honorable" award) from the University of Paris VI (UPMC), France. His Ph.D. was funded by the French National Institute for Research in Computer Science and Control (INRIA)–Paris, France.

6 PREVIOUS TUTORIALS

In the following, we briefly describe our past relevant tutorials.

(1) At ISWC 2016, we delivered a half-day tutorial [3] on how to rapidly develop semantics-based Web of Things (WoT) applications, demonstrated how semantic web technologies can be employed for semantic annotation and reasoning on data to build interoperable IoT/WoT applications.
(2) At WWW 2017, we delivered a half-day tutorial [4] to familiarize audience with the open-source tools designed by various semantic Web, IoT, WoT based projects and provided them a rich hands-on experience to use these tools and build smart applications with minimal effort. We showcased real-world use case scenarios designed using our semantically-enabled frameworks such as CityPulse, FIESTA-IoT and M3.
(3) At APSEC 2015 [6], we delivered a 3-hour tutorial where we explained the latest research from corporate R&D organizations and described the on-ground concerns while architecting and developing real-life IoT applications.
(4) We delivered a 3-hour tutorial [7] explaining all aspects of developing software-powered Physical-Cyber-Social (PCS) systems and discussed the challenges in designing and implementing software artifacts that operate between the Physical and Cyberworld. We also shared our real-world industrial experiences in prototyping IoT products, utilizing big-data tools and conceptual models to design efficient PCS systems.

REFERENCES

[1] 2020. emlearn. https://github.com/emlearn/
[2] 2020. m2cgen: Code-generation for various ML models into native code.
[3] Amelie Gyrard, Pankesh Patel, et al. 2016. Semantic web meets internet of things (iot) and web of things (wot). In *International Conference on Semantic Web (ISWC)*.
[4] Amelie Gyrard, Pankesh Patel, et al. 2017. Semantic web meets internet of things and web of things. In *26th International Conference on World Wide Web*.
[5] Darius Morawiec. 2020. sklearn-porter: Transpile trained scikit-learn models.
[6] Pankesh Patel, Vikrant Kaulgud, et al. 2015. Building enterprise-grade internet of things applications. In *Asia-Pacific Software Engineering Conference (APSEC)*.
[7] Pankesh Patel, Ashok Kumar, et al. 2017. Engineering Smart Physical-Cyber-Social Systems. In *Proceedings of Innovations in Software Engineering Conference*.
[8] Tapan Pathak, Pankesh Patel, et al. 2020. A distributed framework to orchestrate video analytics across edge and cloud: a use case of smart doorbell. In *10th International Conference on the Internet of Things*.
[9] B Sudharsan. 2020. Code. https://github.com/bharathsudharsan/Train_plus_plus
[10] B Sudharsan. 2020. Code. https://github.com/bharathsudharsan/CNN_on_MCU
[11] B Sudharsan. 2020. ML-MCU. https://github.com/bharathsudharsan/ML-MCU
[12] Bharath Sudharsan, J. G. Breslin, and M.I. Ali. 2020. Edge2train: a framework to train machine learning models (svms) on resource-constrained iot edge devices. In *Proceedings of the 10th International Conference on the Internet of Things*.
[13] Bharath Sudharsan, J. G. Breslin, and M.I. Ali. 2020. RCE-NN: a five-stage pipeline to execute neural networks (cnns) on resource-constrained iot edge devices. In *Proceedings of the 10th International Conference on the Internet of Things*.