TINY MACHINE LEARNING

A Seminar Report Submitted in the partial fulfilment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE & ENGINEERING

SUBMITTED BY

M.V.N Vidya Sree

173C1A0557

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

D.M.S S.V.H COLLEGE OF ENGINEERING

(Affiliated to Jawaharlal Nehru Technological University, Kakinada, Approved by AICTE and GOVT of AP, An ISO 9001:2015 Certified Institution)

MACHILIPATNAM-521002, KRISHNA, A.P.

2020-2021

ABSTRACT

Tiny Machine Learning (TML) is a novel research area aiming at designing and developing Machine Learning (ML) techniques meant to be executed on Embedded Systems and Internet-of-Things (IOT) units. Such techniques, which take into account the constraints on computation, memory, and energy characterizing the hardware platform they operate on, exploit approximation and pruning mechanisms to reduce the computational load and the memory demand of Machine and Deep Learning (DL) algorithms.

Despite the advancement of the research, TML solutions present in the literature assume that Embedded Systems and IOT units support only the inference of ML and DL algorithms. The aim of this technology is to address such an open challenge by introducing an incremental algorithm based on transfer learning and k-nearest neighbour to support the on-device learning (and not only the inference) of ML and DL solutions on embedded systems and IOT units. This is also known as Tiny AI and the proposed solution is general and can be applied to different application scenarios.

M.V.N VIDYA SREE

(173C1A0557)

CONTENTS

1. INTRODUCTION

Tiny machine learning is broadly defined as a fast growing field of machine learning technologies and applications including hardware, algorithms and software capable of performing on-device sensor data analytics at extremely low power, typically in the mW range and below, and hence enabling a variety of always-on use-cases and targeting battery operated devices.

Over the past decade, we have witnessed the size of machine learning algorithms grow exponentially due to improvements in processor speeds and the advent of big data. Initially, models were small enough to run on local machines using one or more cores within the central processing unit (CPU).

1.1 GPU Resentment.

Shortly after, computation using graphics processing units (GPUs) became necessary to handle larger datasets and became more readily available due to introduction of cloud-based services such as SaaS platforms. At this time, algorithms could still be run on single machines.

While the achievements of GPT-3 and Turing-NLG are laudable, naturally, this has led to some in the industry to criticize the increasingly large carbon footprint of the AI industry.

Tinyml is a small framework for writing neural networks for educational purposes. The traditional idea of IoT was that data would be sent from a local device to the cloud for processing. Some individuals raised certain concerns with this concept: privacy, latency, storage, and energy efficiency to name a few.

2. LITERATURE SURVEY

An **ABI research** said that Global Shipments of TinyML Devices will reach 2.5 billion by 2030 and could reach more than US$ 70 billion in economic value. At that moment, there are several companies developing chips and frameworks to be used to build more efficient TinyML devices:

1. Arduino with a new **TinyML kit**
2. Google with **Tensorflow Lite for Microcontrollers**
3. **Microsoft Azure Sphere**, an integrated security platform to develop faster and secure IoT devices.

2.1 History of Tiny ML

The concept of TinyML is so new that it doesn't even have a Wikipedia entry , companies like Ericsson are already offering TinyML as-a-Service. The book TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers by Pete Warden and Daniel Situnayake came out in December 31, 2019.

It became clear to me that there was a whole new class of products emerging, with the key characteristics that they used ML to make sense of noisy sensor data, could run using a battery or energy harvesting for years

Making these products real required ways to turn raw sensor data into actionable information 1 locally, on the device itself, since the energy costs of transmitting streams

anywhere have proved to be inherently too high to be practical. This is where the idea of TinyML comes in. Long conversations with colleagues across industry and academia have led to the rough consensus that if you can run a neural network model at an energy cost of below 1 mW, it makes a lot of entirely new applications possible.

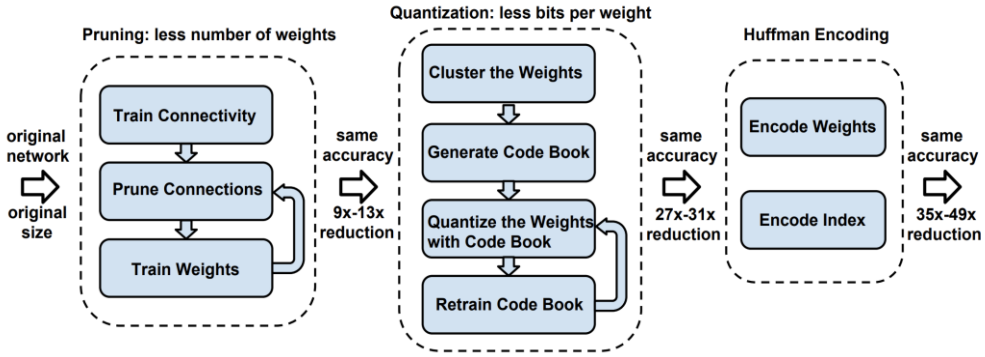
2.2 Technologies Used

- **Edge Computing** - a distributed information technology (IT) architecture in which client data is processed at the periphery of the network, as close to the originating source as possible. In simplest terms, edge computing moves some portion of storage and compute resources out of the central data center and closer to the source of the data itself.
- **Cloud Technology** – also popularly referred to as the cloud – has redefined the way we store and share information. It has helped us transcend the limitations of using a physical device to share and opened a whole new dimension of the internet.
- **The Internet of Things (IoT)-** describes the network of physical objects—"things"—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools

3.DESIGN AND IMPLEMENTATION

Software **design and implementation** is the stage in the software engineering process at which an executable software system is developed. In practice many of the techniques and skills used in selection, analysis and strategy development are continually applied during implementation.

3.1Block Diagram



3.2 Concept

Post-training is where the real tinyML block diagram begins, in a process often referred to as **deep compression**. The idea underlying deep compression is that larger networks have some sparsity or redundancy within them. While large networks have a high representational capacity, if the network capacity is not saturated it could be represented in a smaller network with a lower representation capacity.

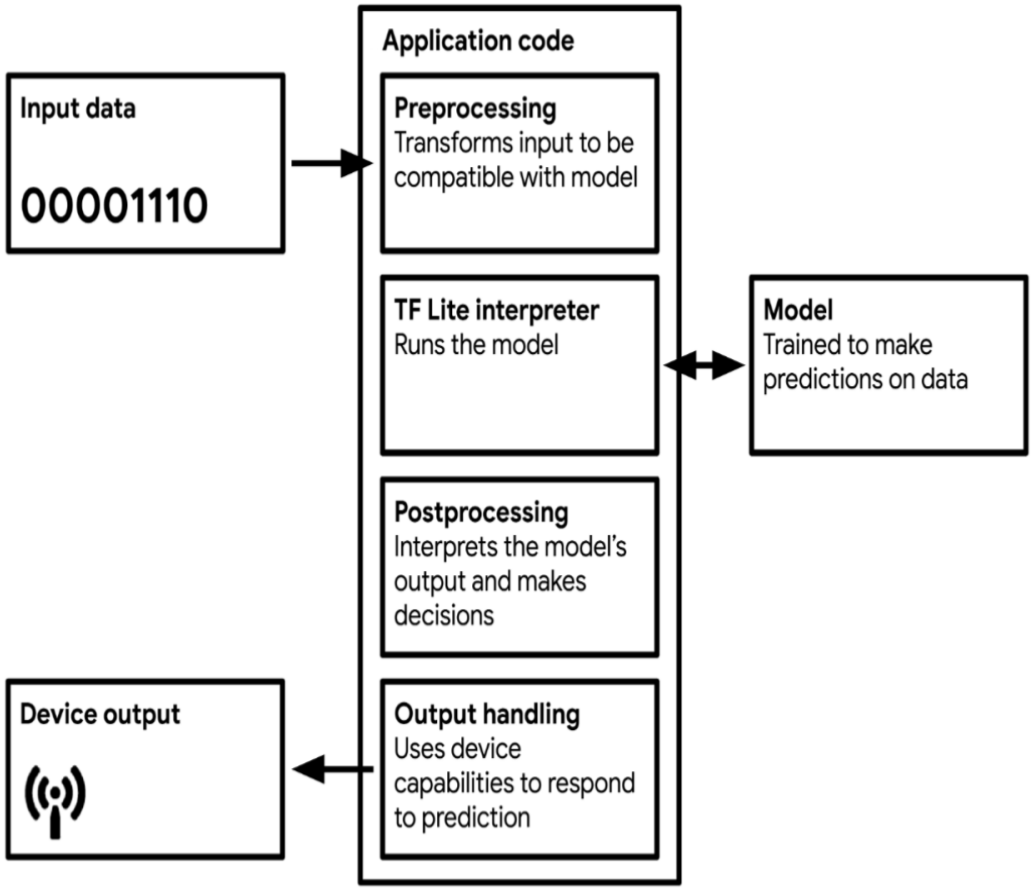
3.3 Description

Pruning, broadly speaking, attempts to remove neurons that provide little utility to the output prediction. The model is then quantized post-training into a format that is compatible with the architecture of the embedded device.

By **quantizing** the model, the storage size of weights is reduced by a factor of 4 (for a quantization from 32-bit to 8-bit values), and the accuracy is often negligibly impacted.

Encoding is an optional step that is sometimes taken to further reduce the model size by storing the data in a maximally efficient way: often via the famed **Huffman Encoding.**

3.4 Working Principle



The model has to be stored on the device, the model also has to be able to perform inference. This means the microcontroller must have a memory large enough that it can run

(1) its operating system and libraries

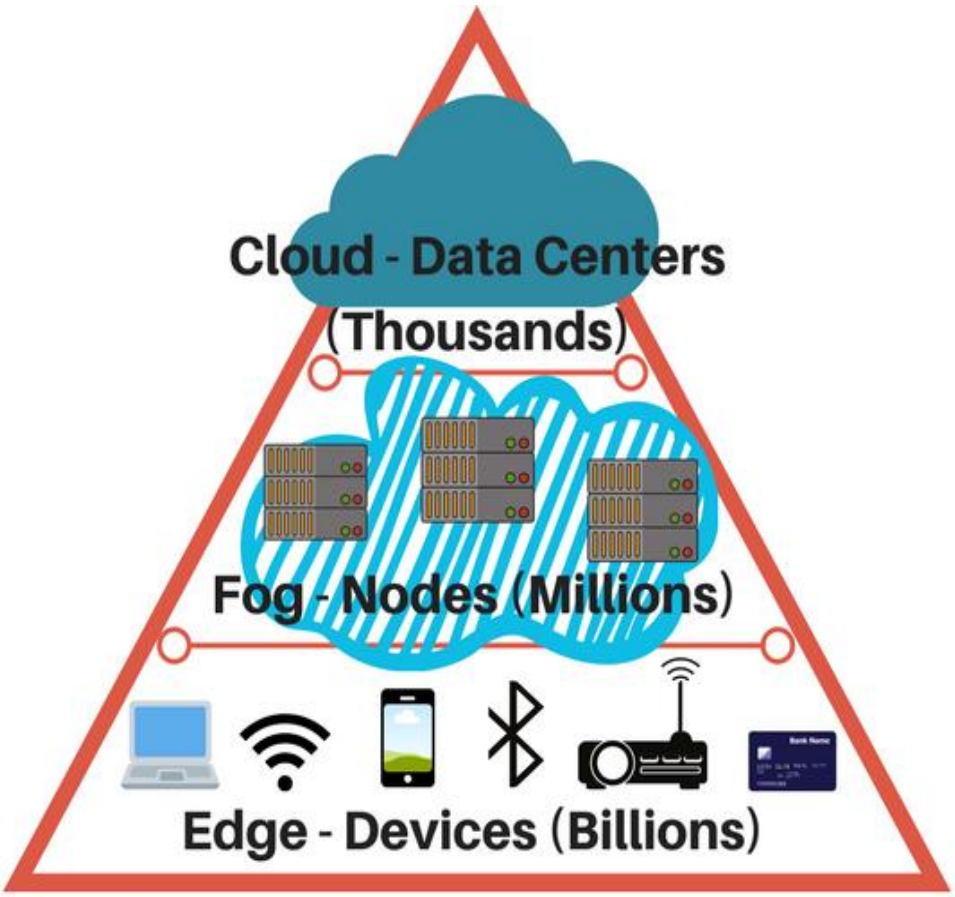
(2) a neural network interpreter such as TF Lite

(3) the stored neural weights and neural architecture

(4) the intermediate results during inference.

Thus, the peak memory usage of a quantized algorithm is often quoted in tinyML

3.5 Architecture



3.6 Characteristics of TinyML

1. It enables low-latency, low power and low bandwidth model inference at edge devices

2.The thousand times less power consumption.

3. TinyML devices to run unplugged on batteries for weeks, months, and in some cases, even years

3.7 Hybrid Components

Hardware

The kit consists of a powerful board equipped with a microcontroller and a wide variety of sensors (Arduino Nano 33 BLE Sense). The board can sense movement, acceleration, rotation, temperature, humidity, barometric pressure, sounds, gestures, proximity, color, and light intensity. The kit also includes a camera module (OV7675) and custom Arduino shield to make it easy to attach your components and create your very own unique TinyML project. We will be able to explore practical ML use cases using classical algorithms as well as deep neural networks powered by TensorFlow Lite Micro.

Software

1.TensorFlow Lite is an open-source deep learning framework developed by Google for inference on embedded devices . This framework consists of two main tools, Converter and Interpreter.

TensorFlow Converter transforms the TensorFlow code into a particular format, reduces the model's size, and optimizes the code for minimal accuracy loss. **TensorFlow Interpreter** is a library that executes the code on the embedded device.

2. Embedded Learning Library is an open-source embedded framework developed by Microsoft that supports the development of machine learning models for several platforms based on ARM Cortex-A and Cortex-M architectures, such as

ELL can be described as a cross-compiler toolchain for ML/AI models.

3. ARM-NN is an open source Linux software for machine learning inference on embedded devices, developed by Arm. The core of Arm's framework is the CMSIS NN software library, which is a collection of efficient neural network kernels developed to maximize the performance and minimize the memory footprint of neural networks on Cortex-M processor cores.

4. APPLICATIONS

1.TinyML has great potential to enhance various monitoring and personal health products .

2.TinyML focuses mostly on developing neural network algorithms for computer vision and audio processing, which are the foundation of surveillance applications**.**

3.The embedded device may run pre-trained machine learning models to detect the anomalies in network traffic pattern.

4. TinyML can address the needs for in-place data analytics and time-critical constraints of some industrial monitoring applications.

5. TinyML technology can take part in many tasks of smart space applications while preserving the user's privacy.

5. ADVANTAGES AND DISADVANTAGES

Advantages

1.Keeping your data on the edge devices reduces this risk and strengthens your data privacy.

2.Saving only necessary data can reduce expenses.

3.Completing all processing and inferencing on edge devices can help serve users with extremely low latency.

4. Due to a more distributed architecture, you don't have to rely on one model for all predictions.

Disadvantages

1.Lack of benchmarking tools

2.Adaptation and lifelong learning

3.Need for other types of machine learning models

4.Lack of appropriate datasets

6.CONCLUSION

Tiny machine learning (TinyML) has the potential to unlock an entirely new class of smart applications across industrial and consumer spaces. This field is still at a very early stage of development, but it evolves rapidly. Currently, the focus of TinyML is on understanding the performance boundaries and trade-offs among different integral parts (machine learning

algorithms, hardware, software) of resource-constrained embedded systems. The edge computing landscape is broad, and TinyML deployments are often highly customized.

The diversity of hardware and software needs to be addressed through benchmarking policies for TinyML to gain wide support.

7.REFERENCES.

[1] Hinton, Geoffrey & Vinyals, Oriol & Dean, Jeff. (2015). Distilling the Knowledge in a Neural Network.

[2] D. Bankman, L. Yang, B. Moons, M. Verhelst and B. Murmann, "An always-on 3.8μJ/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28nm CMOS," *2018 IEEE International Solid-State Circuits Conference — (ISSCC)*, San Francisco, CA, 2018, pp. 222–224, doi: 10.1109/ISSCC.2018.8310264.

[3] Warden, P. (2018). Why the Future of Machine Learning is Tiny. Pete Warden's Blog.

[4] Ward-Foxton, S. (2020). AI Sound Recognition on a Cortex-M0: Data is King. EE Times.