Data science has not just remained a field of scientific computing and research. In the newly internet-connected world with pocket computers everywhere, **data science, machine learning, and artificial intelligence** are far more applicable than ever imagined. No doubt, the very first leap in the practical applications of machine learning and artificial intelligence happened when enterprise websites, including social media platforms, e-commerce portals, and video streaming websites, realized they needed meaningful insights from user data and behavior. This was when big tech companies started using ML and AI on their web servers and cloud infrastructures. Artificial intelligence has been computationally resource-intensive, and servers have barely had enough computational resources to run AI-backed applications 24×7.

Then, smartphones became a global phenomenon, and the concept of applying artificial intelligence to edge devices found its inception. Even smartphones had had enough hardware resources to run **machine learning** models. These pocket computers are regularly charged as they have that kind of use-case. Alphabet Inc. (Google) 's Android and Apple's iOS emerged as the most popular mobile operating systems across the globe. Several apps were developed for these mobile operating systems that were inherently using machine learning algorithms.

synonymous with TinyML as there is no other machine learning framework for microcontrollers. TensorFlow Lite has been developed to run on Android, iOS, Embedded Linux, and microcontrollers. At present, the only reference for TinyML is Pete Warden and Daniel Situnayake's book "**TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers**" published in 2020 and the **Tensorflow Lite documentation**.

What is TinyML?
TinyML is too young to have a proper technical definition. Though, at present, we can define TinyML as a subfield of machine learning that applies machine learning and deep learning models to embedded systems running on microcontrollers, digital signal processors, or other ultra-low-power specialized processors. Technically, these embedded systems must have a power consumption lower than 1 mW so that they can run for weeks, months, or even years without recharging or battery replacement.

Often, these embedded systems are IoT devices that remain connected to the internet. TinyML empowered embedded devices are designed to run some machine learning algorithm for a specific task – often a part of edge computing within the device.

A TinyML machine learning model running on the system is generally a few kilobytes in size and performs a particular cognitive function like recognizing a wake word, identifying people or objects, or deriving insights from data of a specific sensor. With the TinyML machine learning model running for 24×7 hours, the device still must have power consumption in the range of milliwatts or microwatts. The following features characterize a TinyML application:

1. It runs on a microcontroller, digital signal processor, low-power microcomputer hosting embedded Linux, or a mobile platform with explicitly limited RAM, flash memory, and battery power to implement a machine learning model.
2. It runs on ultra low power devices with power consumption in mW or µW while deriving inferences from the 24×7 running machine learning model.
3. It must be implementing machine learning at the edge of the network, therefore not requiring to transfer or exchange data with a cloud or server, i.e., the machine learning model must be executed within the edge device without any need of data communication over a network. The network used by the device must be used only to communicate results of inferences from the machine learning model to a server/cloud/controller if required.
4. The TinyML machine learning model must have a minimal footprint, typically a few tens of kilobytes, to be run on microcontrollers and microcomputers.

The training for these models is usually batch training in offline mode. The sensor data that has to be utilized for learning and deriving inferences is already determined according to the specific application. For example, if the model has to be trained to recognize a wake word, it is already designed to process a continuous audio stream from a microphone. The selection of dataset, the normalization, underfitting or overfitting of the model, regularization, data augmentation, training, validation, and testing, everything is already done with the help of a cloud platform like Google Colab in the case of TensorFlow Lite. After offline batch training, a full-trained model is finally converted and ported to the microcontroller, microcomputer, or digital signal processor.

Once ported to an embedded system, the model has no more training. Instead, it only consumes real-time data from sensors or input devices and applies the model to that data. Therefore, a TinyML machine learning model needs to be highly robust and may be retrained after years or may never be retrained. All possibilities of underfitting and overfitting of the model need to be checked out so that the model remains relevant for a very long period or, ideally, indefinitely. That is why TinyML machine learning models are usually passed through the rigorous train-test-and-validate procedure before conversion. The relevance of the model largely depends on the selection of the right and appropriate dataset and proper normalization and regularization of the dataset.

Why TinyML?

TinyML started as an initiative to eradicate or reduce the dependence of the IoT on cloud platforms for simple small-scale machine learning tasks. This required implementation of machine learning models at the edge devices themselves. TinyML offers the following notable advantages:

1. Low footprint: A TinyML machine learning model is just a few tens of kilobytes in size. This can be easily ported to any microcontroller, DSP, or microcomputer device.
2. Low power consumption: A TinyML application must ideally consume power less than 1 milliWatt. A device can continue deriving inferences from the sensor data for months or years with such small power consumption, even if a coin battery powers it.
3. Low latency: A TinyML application does not require transferring or exchanging data over the network. All sensor data that it works on is captured locally, and an already trained model is applied to it to derive inferences. The result of inferences may be transferred to a server or cloud for logging or further processing, but the data exchange is not required for the device's functioning. This reduces the network latency and eradicates the dependence on a cloud or server for machine learning tasks.
4. Low bandwidth: Ideally, a TinyML application does not require communicating with a cloud or server. Even if the internet connection is used, it is used for tasks other than machine learning. The internet connection may only communicate inferences to a cloud/server. This does not impact the main

effective solution to run tiny machine learning applications on a large scale and is particularly useful where machine learning has to be applied in IoT applications.

How to get started?
To start with TinyML in TensorFlow Lite, first of all, you require a supported microcontroller board. TensorFlow Lite for Microcontrollers library has support for the following microcontrollers.

1. Arduino Nano 33 BLE Sense
2. SparkFun Edge
3. STM32F746 Discovery kit
4. Adafruit EdgeBadge
5. Adafruit TensorFlow Lite for Microcontrollers Kit
6. Adafruit Circuit Playground Bluefruit
7. Espressif ESP32-DevKitC
8. Espressif ESP-EYE
9. Wio Terminal: ATSAMD51
10. Himax WE-I Plus EVB Endpoint AI Development Board
11. Synopsys DesignWare ARC EM Software Development Platform
12. Sony Spresense

To run a machine learning model, these are 32-bit microcontrollers with sufficient flash memory, RAM, and clock frequency. The boards also come with several onboard sensors that can run any embedded application and apply machine learning models to the intended application.

Apart from a hardware platform, you require a laptop or computer to design a machine learning model. Different programming tools are available for each hardware platform that utilize the TensorFlow Lite for Microcontrollers library to build, train, and port machine learning models. The TensorFlow Lite is open source and can be used and modified without any license fees. To start with TinyML using TensorFlow Lite, you need just one of the embedded hardware platforms listed above, a computer/laptop, a USB cable, a USB-to-Serial converter – and a determination to learn machine learning with embedded systems.

Supported machine learning models in TinyML
TensorFlow Lite for Microcontrollers library supports a limited subset of machine learning operations. These operations can be seen from **all_ops_resolver.cc**. The TensorFlow Lite also hosts some example machine learning models at the **following link** that can be directly used for the learning purpose or development of a machine learning empowered embedded application. These example models include image classification, object detection, pose estimation, speech recognition, gesture recognition, image segmentation, text

book on TinyML written by Pete Warden and Daniel Situnayake.

Applications of TinyML

Though TinyML is still in its infancy, it has already found practical applications in many areas, including:

Industrial Automation: TinyML can be used to make manufacturing smarter, for example, by using predictive maintenance of machines and optimizing machine operations for higher productivity. TinyML can also improve machine performance to better product quality and early detection of faults and imperfections in a manufacturing product.

Agriculture: TinyML can be used to detect diseases and pests in plants. As TinyML operates independently of an internet connection, it can perfectly implement automation and IoT in agricultural farms.

Healthcare: TinyML is already in use for the early detection of mosquito-borne diseases. It can also be used in fitness devices and healthcare equipment.

Retail: TinyML can be used for automating inventory management in retail stores. A TinyML application can track items on store shelves and send an alert before they get out of stock with AI-enabled cameras. It can also aid in deriving inferences about customer preferences in the retail sector.

Transportation: TinyML applications can be used to monitor traffic and detect traffic jams. Such an application can be twined with traffic light management to optimize traffic in real-time. It can also be used for accident detection to make automatic alerts to the nearest trauma center.

Law enforcement: TinyML can be used to detect unlawful activities like riots and theft using machine learning and gesture recognition. A similar application can also be used for bank ATMs' security. A TinyML model can predict whether the user is a genuine customer making a transaction or an intruder trying to hack or break the ATM by monitoring user activity.

Ocean life conservation: TinyML applications are already in use for real-time monitoring of whales in the waterways of Vancouver and Seattle to avoid whales striking busy water lanes. Similar applications can monitor poaching, illegal mining, and deforestation. TinyML devices can also be deployed to monitor the well-being of coral reefs.

Conclusion

TinyML itself is a revolutionary idea combining embedded systems with machine learning. The technology can emerge as a major subfield in machine learning and artificial intelligence as the narrow AI peaks in various verticals and domains. The TinyML offers a solution to many problems currently faced by the IoT industry and the experts applying machine learning to various domain-specific fields. The idea of using machine learning at edge devices with minimal computational footprint and power consumption can bring a significant change in how embedded systems and robots are designed. The current frameworks require more community support and support from the chip designers. TinyML is destined to go mainstream as the supported hardware and programming tools and libraries expand soon.

You may also like:

FPGA vs microcontrollers: Another approach to embedded design

What is machine learning?

What are different types of Artificial Intelligence ?

What is Artificial Intelligence, Machine Learning, Deep Learning, and Natural...

Introduction to Robotics

Artificial Intelligence vs. Intelligence Augmentation

Filed Under: Featured, What Is
Tagged With: Tensorflow Lite, TensorFlow Lite for Microcontrollers, TinyML

← **Previous Article** **Next Article →**

Search this website GO

HAVE A QUESTION?

Have a technical question about an article or other engineering questions? Check out our engineering forums EDABoard.com and Electro-Tech-Online.com where you can get those questions asked and answered by your peers!

EDA BOARD

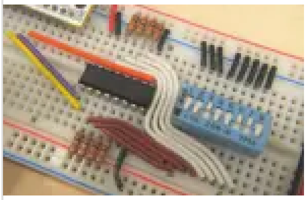
ELECTRO-TECH-ONLINE

EDABOARD.COM DISCUSSIONS

- problem using uint32_t return argument in EFR32FG14
- How do you implement offset voltage in SMPS Load share circuitry?
- Steps to Load pull, Stabilize and Match a power amplifier
- What are these structures in RF Amplifier
- ENC28j60 simulation in Proteus over Windows10

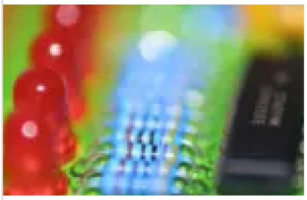
ELECTRO-TECH-ONLINE.COM DISCUSSIONS

- Hardware ESC 8xSERVO CONTROL on PIC (Oshonsoft BASIC)

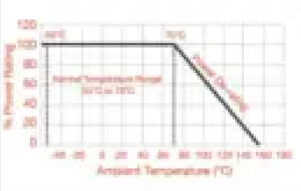


Basic Electronics 01 – Beginners guide to setting up an electronics lab

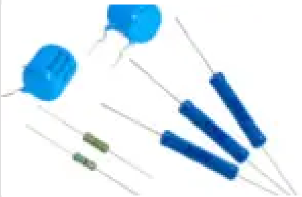
Basic Electronics 02 – Common mistakes made by electronics beginners



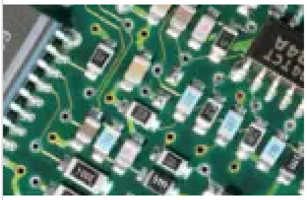
Basic Electronics 03 – Practical guide to resistors



Basic Electronics 04 – Practical guide to resistors



Basic Electronics 05 – Fixed Resistors: composition, film type and cermet



Basic Electronics 06 – Fixed Resistors: wirewound, foil and semiconductor resistors

STAY UP TO DATE

other insightful tech content.

EE TRAINING CENTER

EE CLASSROOMS

EE DESIGN GUIDES

RECENT ARTICLES

- The top 3D-printed robotic arms for 2023
- STMicroelectronics expands its range of USB-PD digital controllers
- Digi-Key releases new Boards Guide and augmented-reality app
- Infineon launches new PDM microphone with low power consumption
- Vishay offers automotive-grade polymer tantalum chip capacitors

SUBMIT A GUEST POST

ANALOG IC TIPS

CONNECTOR TIPS

DESIGNFAST

EDABOARD FORUMS

EE WORLD ONLINE

ELECTRO-TECH-ONLINE FORUMS

MICROCONTROLLER TIPS

POWER ELECTRONIC TIPS

SENSOR TIPS

TEST AND MEASUREMENT TIPS

5G TECHNOLOGY WORLD

ABOUT US

CONTACT US

ADVERTISE