Why The Future of ML is Tiny and Bright

Challenges & Opportunities

Vijay Janapa Reddi, Ph. D. | Associate Professor |
John A. Paulson School of Engineering and Applied Sciences | Harvard University |
Web: http://scholar.harvard.edu/vijay-janapa-reddi
“Language”
What is Machine Learning?

1. **Machine Learning** is a subfield of **Artificial Intelligence** focused on developing algorithms that learn to **solve** problems by analyzing data for patterns.
What is (Deep) Machine Learning?

1. Machine Learning is a subfield of Artificial Intelligence focused on developing algorithms that learn to solve problems by analyzing data for patterns.

2. **Deep Learning** is a type of Machine Learning that leverages **Neural Networks** and **Big Data**.
Applications of Machine Learning
Applications of Machine Learning
Applications of Machine Learning
Image Classification
Object Detection
Segmentation
Machine Translation

1. Upload translated language pairs
   - 一扇門 a door
   - 兩個檯燈 two table lamps
   - 四個棉被 four quilts
   - 一壺茶 a pot of tea
   - 五部電話 five telephones
   - 六塊電池 six batteries

2. Train your model

3. Evaluate

AutoML Translation
Recommendations
Datacenter
Datacenter
TPUs/GPUs
But... Bigger Is Not Always Better.
High power
High bandwidth
High latency

Low power
Low bandwidth
Low latency
Endpoint Devices

Google Assistant
Endpoint Devices

Google Assistant
No Good Data Left Behind

5 Quintillion bytes of data produced every day by IoT

<1% of unstructured data is analyzed or used at all

Cisco, Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How?, Feb 5, 2018
Tiny Machine Learning
What is Tiny Machine Learning (TinyML)?
What is Tiny Machine Learning (TinyML)?

TinyML

Fastest-growing field of ML
What is Tiny Machine Learning (TinyML)?

TinyML

Fastest-growing field of ML

Algorithms, hardware, software
What is Tiny Machine Learning (TinyML)?

- TinyML
- Fastest-growing field of ML
- Algorithms, hardware, software
- On-device sensor analytics
What is Tiny Machine Learning (TinyML)?

- **TinyML**
- Fastest-growing field of ML
- On-device sensor analytics
- Low power consumption
- Algorithms, hardware, software
What is Tiny Machine Learning (TinyML)?

TinyML

- Fastest-growing field of ML
- Algorithms, hardware, software
- On-device sensor analytics
- Low power consumption
- Always-on ML
What is Tiny Machine Learning (TinyML)?

TinyML

- Fastest-growing field of ML
- Algorithms, hardware, software
- On-device sensor analytics
- Low power consumption
- Always-on ML
- Battery-operated
More Forward Looking Applications
More Forward Looking Applications
More Forward Looking Applications
More Forward Looking Applications
More Forward Looking Applications
Talking with whales

Project aims to translate sperm whale calls

By Leah Burrows | Press contact
April 22, 2021

This week, a team of scientists in partnership with the Government of Dominica and the National Geographic Society, officially launched an ambitious, interdisciplinary research initiative to listen to, contextualize, and translate the communication of sperm whales.

Project CETI (Cetacean Translation Initiative) will bring together leading cetacean researchers from around the world to examine the complex vocalizations of sperm whales. These vocalizations include bubble sequences, which are unique to each individual and serve as a form of identification, and other sounds that scientists believe may convey information about social interactions, migration, and marine environments.

The project aims to advance our understanding of sperm whale behavior and communication, potentially leading to new insights into the species' ecological roles and conservation needs.
ElephantEdge

Building The World’s Most Advanced Wildlife Tracker.
ElephantEdge

**Risk Monitoring**

“Know when an elephant is moving into a high-risk area and send real-time notifications to park rangers.”

**Conflict Monitoring**

“Sense and alert when an elephant is heading into an area where farmers live.”

**Activity Monitoring**

“Classify the general behavior of the elephant, such as when it is drinking, eating, sleeping, etc.”

**Communication Monitoring**

“Listen for vocal communications between elephants via the onboard microphone.”
Massive tinyML opportunities in all verticals where machine intelligence meets physical world of billions of sensors
Technology for TinyML
What Makes TinyML?
250 Billion

MCUs today
IoT Microcontroller Market Size, Share & Trends Analysis Report By Product (8-bit, 16-bit, 32-bit) By Application (Industrial Automation, Smart Home, Consumer Electronics) And Segment Forecasts To 2022
MCU Pricing Forecast

Average Selling Price

Source: IC Insights
<table>
<thead>
<tr>
<th>Board</th>
<th>MCU / ASIC</th>
<th>Clock</th>
<th>Memory</th>
<th>Sensors</th>
<th>Radio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Himax WE-I Plus EVB</td>
<td>HX6537-A 32-bit EM9D DSP</td>
<td>400 MHz</td>
<td>2MB flash</td>
<td>Accelerometer, Mic, Camera</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2MB RAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arduino Nano 33 BLE Sense</td>
<td>32-bit nRF52840</td>
<td>64 MHz</td>
<td>1MB flash</td>
<td>Mic, IMU, Temp, Humidity, Gesture,</td>
<td>BLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>256kB RAM</td>
<td>Pressure, Proximity, Brightness,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Color</td>
<td></td>
</tr>
<tr>
<td>SparkFun Edge 2</td>
<td>32-bit ArtemisV1</td>
<td>48 MHz</td>
<td>1MB flash</td>
<td>Accelerometer, Mic, Camera</td>
<td>BLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>384kB RAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Espressif EYE</td>
<td>32-bit ESP32-D0WD</td>
<td>240 MHz</td>
<td>4MB flash</td>
<td>Mic, Camera</td>
<td>WiFi, BLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>520kB RAM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TF Micro

Hardware
- Heterogeneity
  - CPU
  - GPU
  - DSP
- Resource Constraints
  - NPU
  - Memory
  - Power

Software
- Missing Library Features
  - malloc
- Limited Operating System Support
  - ...

Resource Constraints
- Limited Operating System Support
- Missing Library Features
  - malloc
- ...

Heterogeneity
- Limited Operating System Support
- Missing Library Features
  - malloc
- ...

CPU

GPU

DSP

NPU

Memory

Power
TF Micro

Hardware

- Heterogeneity
  - CPU
  - GPU
  - DSP

- Resource Constraints
  - NPU
  - Memory
  - Power

Software

- Missing Library Features
  - malloc

- Limited Operating System Support
  - ...

Resource Constraints

- Limited Operating System Support
  - malloc

Heterogeneity

- CPU
  - GPU
  - DSP
Hardware
- Heterogeneity
  - CPU
  - GPU
  - DSP
- Resource Constraints
  - NPU
  - Memory
  - Power

Software
- Missing Library Features
  - malloc
- Limited Operating System Support
Micro
Himax
WE-I Plus EVB
SparkFun
Edge 2
Espressif
EYE
Arduino
BLE Sense 33
...
TensorFlow Lite Micro
in a Nutshell

Compatible with the TensorFlow training environment.
Built to fit on **embedded systems**:
- Very small binary footprint
- No dynamic memory allocation
- No dependencies on complex parts of the standard C/C++ libraries
- No operating system dependencies, can run on bare metal
- Designed to be **portable** across a wide variety of systems

---

**TensorFlow Lite Micro:**
**EMBEDDED MACHINE LEARNING ON TINYML SYSTEMS**

Robert Durrant 1, 2  
Jared Oden 1  
Adrián Zaldivar 3  
Rajeev Jha 1  
Vijaykumar Jagadish 1  
N.B. Jha 4  
Jen Li 5  
Nikhil Kundra 5  
Ian Napier 3  
Mihai Nica 3  
Mohamed Raja 6  
Rocky Wadle 7  
Thi Lan Ng 8  
Peter Warden 9

**ABSTRACT**

TinyML (TensorFlow Lite Micro, TFLM) is an open-source ML inference framework for running deep learning models on embedded systems. TFLM tackles the efficiency requirements imposed by embedded systems resource constraints and the fragmentation challenges that make cross-platform portability mostly impossible. This framework adopts a unique compiler-based approach that provides flexibility while overcoming these unique challenges. In this paper, we explain the design decisions behind TFLM and several of its implementations. We present an evaluation of TFLM to demonstrate its low resource requirements and minimal end-to-end performance overheads.

1 INTRODUCTION

TinyML is a compact technology that can be deployed on embedded systems where computational resources are limited. The model size and inference time are the key factors in determining the performance of a TinyML model. In this section, we introduce the key components of TinyML and compare it with other technologies.

TinyML is designed for devices with limited computational resources, such as microcontrollers, sensors, and mobile devices. It is optimized to run on hardware that has limited memory and processing power.

TinyML can be used in various applications, such as image recognition, speech recognition, and natural language processing. It is becoming increasingly popular due to its ability to run on devices with limited resources.

**RELATED WORK**

Several open-source frameworks are available for deploying ML models on embedded devices. TensorFlow Lite is one such framework that is popular due to its flexibility and compatibility with various hardware platforms. However, TensorFlow Lite is not optimized for devices with limited resources.

**CONCLUSIONS**

In conclusion, TensorFlow Lite Micro (TFLM) is an open-source ML inference framework for running deep learning models on embedded systems. TFLM tackles the efficiency requirements imposed by embedded systems resource constraints and the fragmentation challenges that make cross-platform portability mostly impossible. This framework adopts a unique compiler-based approach that provides flexibility while overcoming these unique challenges. In this paper, we explain the design decisions behind TFLM and several of its implementations. We present an evaluation of TFLM to demonstrate its low resource requirements and minimal end-to-end performance overheads.

---

What Makes TinyML?

1. Embedded Systems Code
2. Training Data
Collect Data → Preprocess Data → Design a Model → Train a Model → Evaluate Model → Optimize Model → Convert Model → Deploy Model → Make Inferences
Speech commands for the whole planet?
● Speech commands for the **whole planet**?

● For **more than** just voice assistants
Data Engineering

Requirements

- Problem definition
- Permissions & rights
- Machine & human usable format
Data Engineering

Requirements
- Problem definition
- Permissions & rights
- Machine & human usable format

Gathering
- People
- Collection
- Labeling
- Data sources
Data Engineering

- Problem definition
- Permissions & rights
- Machine & human usable format

- People
- Collection
- Labeling
- Data sources

- Processing
- Validation
- Augmentation
Data Engineering

Requirements
● Problem definition
● Permissions & rights
● Machine & human usable format

Gathering
● People
● Collection
● Labeling
● Data sources

Refinement
● Processing
● Validation
● Augmentation

Sustainment
● Storage
● Security
● Errors
● Versioning
Data Engineering

Requirements
- Problem definition
- Permissions & rights
- Machine & human usable format

Gathering
- People
- Collection
- Labeling
- Data sources

Refinement
- Processing
- Validation
- Augmentation

Sustainment
- Storage
- Security
- Errors
- Versioning
Datasets require **significant effort**

These **massive** machine learning datasets are **constructed by hand**

- **Common Voice**—5000+ hours of spoken audio
- **Common Objects in Context (COCO)**—2.5M+ labeled images
- **ImageNet**—4M+ labeled images
- **Waymo**—1,950 20-second driving segments
- **KITTI 360**—73KM+ of annotated driving data

Data Engineering is costly and tedious.
Democratize Data Engineering
Automatic Keyword Dataset Generation

Specify Wanted Keywords

1. Up
2. Down
3. Yes
4. No
5. ...
6. ...
  ...
265. ...
Specify Wanted Keywords

1. Up
2. Down
3. Yes
4. No
5. ...
6. ...
... 265. ...

Search CV Sentences

en/validated.tsv

- He gazed up the steep bank. clip_29132.mp3
- "Yes," said Harry sullenly. clip_34212.mp3
- Pencils down, time is over. clip_54972.mp3
- Get that ladder up here. clip_28213.mp3
- There is no help for it. clip_38311.mp3
...

All sentences containing keywords
Automatic Keyword Dataset Generation

Specify Wanted Keywords
1. Up
2. Down
3. Yes
4. No
5. …
6. …
265. …

Search CV Sentences

<table>
<thead>
<tr>
<th>en/validated.tsv</th>
</tr>
</thead>
<tbody>
<tr>
<td>He gazed up the steep bank. clip_29132.mp3</td>
</tr>
<tr>
<td>“Yes,” said Harry sullenly. clip_34212.mp3</td>
</tr>
<tr>
<td>Pencils down, time is over. clip_54972.mp3</td>
</tr>
<tr>
<td>Get that ladder up here. clip_28213.mp3</td>
</tr>
<tr>
<td>There is no help for it. clip_38311.mp3</td>
</tr>
<tr>
<td>… …</td>
</tr>
</tbody>
</table>

Extract Keyword Utterances

Input 5-10 second mp3
* gazed up the steep.*

Output
1 second wav, silence-padded
Automatic Keyword Dataset Generation

1. Specify Wanted Keywords
   - 1. Up
   - 2. Down
   - 3. Yes
   - 4. No
   - 5. ...
   - 6. ...
   - 265. ...

2. Search CV Sentences
   - en/validated.tsv
   - He gazed up the steep bank. clip_29132.mp3
   - "Yes," said Harry sullenly. clip_34212.mp3
   - Pencils down, time is over. clip_54972.mp3
   - Get that ladder up here. clip_28213.mp3
   - There is no help for it. clip_38311.mp3
   - ...
   - All sentences containing keywords

3. Extract Keyword Utterances
   - Input 5-10 second mp3
   - "...gazed up the steep..."
   - Output
     - 1 second wav, silence-padded

He gazed **up** the steep bank.
"Yes," said Harry sullenly.
Get that ladder **up** here.
Pencils **down**, time is over.
There is no help for it.

**Estimate Per-Word Timing**

**Extract Keywords**

**Add**

Large Keyword Dataset
4.3M Utterances
3,126 Keywords
22 Languages
Automatic Keyword Dataset Generation

Wanted Keywords
1. Up
2. Down
3. Yes
4. No
5. …
6. …
…
265. …

Forced Alignment

<table>
<thead>
<tr>
<th>Input 5-10 second mp3</th>
<th>Output 1 second wav, silence-padded</th>
</tr>
</thead>
</table>

Word Extraction

- He gazed up the steep bank.
  - en/validated.tsv
- "Yes," said Harry sullenly.
  - clip_29132.mp3
- Pencils down, time is over.
  - clip_54972.mp3
- Get that ladder up here.
  - clip_28213.mp3
- There is no help for it.
  - clip_38311.mp3

All sentences containing keywords

Train KWS Model

- Large Keyword Dataset
  - 17M Utterances
  - 344,286 Keywords
  - 50 Languages

Estimate Per-Word Timing

Extract Keywords

Add
Nine-Language Embedding Model

Multilingual Embedding Model

760 Frequent Words in 9 Languages

49x40 spectrogram

EfficientNet-B0 Backbone

Fully-Connected ReLU (2048)

Fully-Connected ReLU (2048)

Fully-Connected SELU (1024)

761-category output [background, words]
Nine-Language Embedding Model

Multilingual Embedding Model
- 760 Frequent Words in 9 Languages
- Embedding Representation
- EfficientNet-B0 Backbone
  - Fully-Connected ReLU (2048)
  - Fully-Connected ReLU (2048)
  - Fully-Connected SELU (1024)
- 49x40 spectrogram
- 761-category output [background, words]

Five-Shot Keyword Spotting Model
- 5 Target Keyword Examples
- 5K Non-Target Examples
- 49x40 spectrogram
- Embedding Representation
- Softmax Layer
  - Target
  - Unknown
  - Background
Generalizing to **Any** Language

![Graph showing True Positive Rate vs False Positive Rate with 9 languages seen by the embedding model]
Generalizing to Any Language

- 9 languages seen by the embedding model
- 13 languages not seen by the embedding model

BETTER
Challenge:
1000 Words in 1000 Languages
Widening Access to Applied ML
10 Million
4 Billion
Over 250 Billion

Many Courses
Handful
Few

General ML
Mobile ML
TinyML
Need for Full-Stack ML Developers
Course 1: Fundamentals of TinyML
- Neural Network
- Filters
- Regression
- Loss Function
- Data augmentation
- Inference
- CNNs/DNNs
- Classification
- Preprocessing
- Responsible AI
- Gradient Descent

Applications of TinyML

Course 2: Deploying TinyML
- Keyword Spotting
- Visual Wake Words
- Gesture Recognition
Responsible AI: Human-Centered Design

Course 1
Fundamentals of TinyML

Course 2
Applications of TinyML

Course 3
Deploying TinyML
Course 1
Fundamentals of TinyML

- **What** am I building?
- **Who** am I building this for?
- What are the **consequences** for the user if it **fails**?

Course 2
Applications of TinyML

Course 3
Deploying TinyML
Responsible AI: Human-Centered Design

Course 1
Fundamentals of TinyML
- What am I building?
- Who am I building this for?
- What are the consequences for the user if it fails?

Course 2
Applications of TinyML
- What data will be collected to train the model?
- Is the dataset biased?
- How can we ensure the model is fair?

Course 3
Deploying TinyML
Responsible AI: Human-Centered Design

**Course 1**
*Fundamentals of TinyML*
- What am I building?
- Who am I building this for?
- What are the consequences for the user if it fails?

**Course 2**
*Applications of TinyML*
- What data will be collected to train the model?
- Is the dataset biased?
- How can we ensure the model is fair?

**Course 3**
*Deploying TinyML*
- How will model drift be monitored?
- How should security breaches be addressed?
- How should the user’s privacy be protected?
Widening Access to Applied ML

- Broaden the reach of applied AI/ML resources globally
- From the Big Tech & Ivory Tower to the Greater Commons
- Focus on end-to-end ML application development

Widening Access to Applied Machine Learning

Vijay Janapa Reddi, Brian Plascher, Susan Kennedy, Laurence Moroney, Pete Warden, Anant Agarwal, Colby Barburn, Massimo Banzi, Benjamin Brown, Sharad Chitlangia, Radhika Ghonial, Rupert Jaeger, Srivatsan Krishnan, Daníel Leiker, Mark Manzulder, Dominic Vajda, Bhikam Rastaprasad, J. Evan Smith, Matthew Stewart, Dustin Tingley

Harvard University

Google

Abstract

Despite the expanding role of machine learning (ML) in applications in just a few countries and organizations, there is an attrition among the end-users. The required computing hardware and software are not available to everyone, and most of the applications are designed for a specific use case. The current approach is to focus on end-to-end ML application development, which requires a deep understanding of the development process from data collection to deployment. Therefore, we must understand the ethical implications of their designs before deploying them. To address this issue, we have developed a course on TinyML, a TinyML course that provides application-driven instruction on the development of end-to-end solutions using TinyML. The course is available on GitHub and has been successfully used to train learners on the development process of TinyML. The course introduces learners to real-world applications and the ethical considerations of their design. The course is designed to be accessible to anyone with a basic understanding of programming and deployment of TinyML applications in both the cloud and on edge devices. The course materials are open-source, and contributors are encouraged to contribute to the course.

1 Introduction

The past two decades have seen dramatic progress in machine learning (ML) from a purely academic discipline to a widespread commercial technology that enables a range of sectors. ML allows developers to innovate new processes and products that can be deployed through data-driven optimization. Given applied ML's agility and

Figure 1: We designed a new applied ML course motivated by real-world applications, covering not only the software (ML algorithms) and hardware (embedded systems) but also the product life cycle and responsible AI. To make it accessible and scalable, as well as to provide hands-on components, we focused on the emerging TinyML domain and released the course as a NOOC on edX.

success, it is an important course for everyone. The course provides a wide range of applications, including artificial intelligence (AI), autonomous vehicles (AVs), robotics (Ro), health care (HC), transportation (TR), and security (S), education (E), etc. New use cases are rapidly emerging, and the list above is not exhaustive. The main reason for the proliferation of this technology and associated jobs is the great potential to improve security and uncover new opportunities for technological innovation, societal prosperity, and individual growth. But it is difficult to have the technical skills required to perform this kind of work. To address this issue, we have developed a course on TinyML, a TinyML course that provides application-driven instruction on the development of end-to-end solutions using TinyML. The course is available on GitHub and has been successfully used to train learners on the development process of TinyML. The course is designed to be accessible to anyone with a basic understanding of programming and deployment of TinyML applications in both the cloud and on edge devices. The course materials are open-source, and contributors are encouraged to contribute to the course.

1 Introduction

The past two decades have seen dramatic progress in machine learning (ML) from a purely academic discipline to a widespread commercial technology that enables a range of sectors. ML allows developers to innovate new processes and products that can be deployed through data-driven optimization. Given applied ML's agility and
Conclusion

- Why AI is going tiny
  - BLERP

- How it can change the world
  - Unlocking real-time AI
  - AI for Social Good

- What shrinks it
  - Challenges in terms of “Code” and “Data”
The Future of AI is Tiny and Bright

Challenges & Opportunities

Vijay Janapa Reddi
Harvard University