Yá’át’éezh 🙋

EASI-22

Edge AI Summer Institute 2022

with Navajo Tech
Our website!

tinyMLedu.org/EASI-22

home base for all information!
Workshop Agenda

Day 1
Introduction to AI and (Tiny)ML

Day 2
Keyword Spotting for the Navajo Language

Day 3
Bringing AI/ML from the Cloud to the Edge

Cloud ML
Mobile ML
Embedded ML
Workshop Agenda

Day 1  Introduction to AI and (Tiny)ML  Cloud ML
Day 2  Keyword Spotting for the Navajo Language  Mobile ML
Day 3  Bringing AI/ML from the Cloud to the Edge  Embedded ML
‘I’m Dhilan!
I study Electrical Engineering and Computer Science at Harvard.'
How ML works?

with Brian
Where ML works?

with Dhilan
Last time: **Phone**

![Phone screen with a chart showing listening percentages for 'NO', 'NOISE', 'UNKNOWN', and 'YES' with values 86, 85, and 84, respectively. A chart with a legend indicating 'Listening... unknown' with percentages 0.98, 1.00, and 0.91, respectively.]
Today: **TinyML Kit**
Today’s Agenda

- Review + Why Tiny?
- Hardware Basics
- Installing and Starting the Arduino IDE
- Testing Your TinyML Kit
- Deploying KWS model onto Arduino
- Summary & Next Steps
Today’s Agenda

Review + Why Tiny?

- Hardware Basics
- Installing and Starting the Arduino IDE
- Testing Your TinyML Kit
- Deploying KWS model onto Arduino
- Summary & Next Steps
Machine Learning Workflow

- Collect & Transform Data
- Design & Train a Model
- Deploy Model
- Make Inferences
Machine Learning Workflow
Machine Learning Workflow

1. Collect & Transform Data
2. Design & Train a Model
3. Deploy Model
4. Make Inferences

- Make a Guess!
- Check and count mistakes
- Improve your guess
Machine Learning Workflow

Collect & Transform Data

Design & Train a Model

Deploy Model

Make Inferences

Neural Network
Machine Learning Workflow

1. Collect & Transform Data
2. Design & Train a Model
3. Deploy Model
4. Make Inferences
5. Evaluate
6. Optimize
7. Convert Model
8. Deploy Model
9. Make Inferences
Machine Learning Workflow
Machine Learning Workflow

Collect & Transform Data → Design & Train a Model → Deploy Model → Make Inferences
Machine Learning Workflow

Battery & Memory

Less memory
Less compute power
Only focused on inference
deploy
to your tiny devices!
Datacenter
TPUs/GPUs
Bigger Is Not Always Better.
Latency

Power

Bandwidth

High

High

High

High latency

Low

Low

Low

Low latency
Endpoint Devices

Google Assistant
Endpoint Devices
Endpoints Have **Sensors**, Tons of Sensors

- **Motion Sensors**
  - Gyroscope, radar, magnetometer, accelerometer

- **Acoustic Sensors**
  - Ultrasonic, Microphones, Geophones, Vibrometers

- **Environmental Sensors**
  - Temperature, Humidity, Pressure, IR, etc.

- **Touchscreen Sensors**
  - Capacitive, IR

- **Image Sensors**
  - Thermal, Image

- **Biometric Sensors**
  - Fingerprint, Heart rate, etc.

- **Force Sensors**
  - Pressure, Strain

- **Rotation Sensors**
  - Encoders
Endpoints Have **Sensors**, Tons of Sensors

- **Motion Sensors**
  - Gyroscope, radar, magnetometer, accelerometer

- **Acoustic Sensors**
  - Ultrasonic, Microphones, Geophones, Vibrometers

- **Environmental Sensors**
  - Temperature, Humidity, Pressure, IR, etc.

- **Touchscreen Sensors**
  - Capacitive, IR

- **Image Sensors**
  - Thermal, Image

- **Force Sensors**
  - Pressure, Strain

- **Rotation Sensors**
  - Encoders

- **Biometric Sensors**
  - Fingerprint, Heart rate, etc.
Biometric Sensors

Fingerprint + Photoplethysmography (PPG)

Source: Jacobs School of Engineering/UC San Diego
Endpoints Have **Sensors**, Tons of Sensors

**Motion Sensors**
- Gyroscope, radar, magnetometer, accelerator

**Acoustic Sensors**
- Ultrasonic, Microphones, Geophones, Vibrometers

**Environmental Sensors**
- Temperature, Humidity, Pressure, IR, etc.

**Touchscreen Sensors**
- Capacitive, IR

**Image Sensors**
- Thermal, Image

**Force Sensors**
- Pressure, Strain

**Rotation Sensors**
- Encoders

**Biometric Sensors**
- Fingerprint, Heart rate, etc.
Thinking Big
Thinking Big
Thinking Big

BIG
GPU / CPU
561mm²
Thinking **Small**

BIG
GPU / CPU
561mm²
Thinking Small

BIG
GPU / CPU
561mm²
Thinking **Small**

**BIG**
GPU / CPU
561mm$^2$

**SMALL**
Mobile SoC
83mm$^2$
Thinking Tiny

BIG
GPU / CPU
561mm$^2$

SMALL
Mobile SoC
83mm$^2$
Thinking Tiny

BIG
GPU / CPU
561mm²

SMALL
Mobile SoC
83mm²
Thinking Tiny

BIG
GPU / CPU
561mm$^2$

SMALL
Mobile SoC
83mm$^2$
Thinking Tiny

BIG
GPU / CPU
561mm²

SMALL
Mobile SoC
83mm²

TINY
Apple 0778
30mm²
We’re just getting started.
Thinking Record-breaking

BIG
GPU / CPU
561mm²

SMALL
Mobile SoC
83mm²

TINY
Apple 0778
30mm²

Kinetis KL03
3.2mm²
Thinking Record-breaking

BIG
GPU / CPU
561mm²

SMALL
Mobile SoC
83mm²

TINY
Apple 0778
30mm²

Kinetis KL03
3.2mm²

world’s smallest
ARM-Powered MCU
48MHz, 32KB flash, 20-pin
250 Billion

today
Challenges

- Latency & Bandwidth
- Accuracy & Personalization
- Security & Privacy
- Battery & Memory
Less memory

Less compute power

Only focused on *inference*
Even less memory

Even less compute power

Also, only focused on *inference*
What is Tiny Machine Learning (TinyML)?

- TinyML
- Fastest-growing field of ML
- Algorithms, hardware, software
- On-device sensor analytics
- Low power consumption
- Battery-operated
- Always-on ML
Measuring capacity by watt-hours lets us compare any type of battery

1 12 V car battery
≈ 133 9 V batteries
≈ 67 3.6 V smartphones

1 battery \times (12 \text{ V} \times 50 \text{ Ah}) \quad \frac{133 \text{ batteries}}{600 \text{ Wh}} \quad \frac{67 \text{ smartphones}}{600 \text{ Wh}}

\times (9 \text{ V} \times .5 \text{ Ah})

\times (3.6 \text{ V} \times 2.5 \text{ Ah})
TYPICAL BATTERY LIFE

Data Usage

50KB EVERY 2 HOURS
14.2 YRS.

200KB EVERY 2 HOURS
9 YRS.

50KB EVERY 24 HOURS
23.6 YRS.

200KB EVERY 24 HOURS
21.8 YRS.
TYPICAL BATTERY LIFE

Data Usage

- 50KB EVERY 2 HOURS: 14.2 YRS.
- 200KB EVERY 2 HOURS: 9 YRS.
- 50KB EVERY 24 HOURS: 23.6 YRS.
- 200KB EVERY 24 HOURS: 21.8 YRS.

transmit less data!
Today’s Agenda

- Review + Why Tiny?
- Hardware Basics
- Installing and Starting the Arduino IDE
- Testing Your TinyML Kit
- Deploying KWS model onto Arduino
- Summary & Next Steps
Today's Agenda

- Review + Why Tiny?
- Hardware Basics
  - Installing and Starting the Arduino IDE
  - Testing Your TinyML Kit
  - Deploying KWS model onto Arduino
  - Summary & Next Steps
<table>
<thead>
<tr>
<th>Binary</th>
<th>Hex</th>
<th>Decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>1010</td>
<td>A</td>
<td>10</td>
</tr>
<tr>
<td>1011</td>
<td>B</td>
<td>11</td>
</tr>
<tr>
<td>1100</td>
<td>C</td>
<td>12</td>
</tr>
<tr>
<td>1101</td>
<td>D</td>
<td>13</td>
</tr>
<tr>
<td>1110</td>
<td>E</td>
<td>14</td>
</tr>
<tr>
<td>1111</td>
<td>F</td>
<td>15</td>
</tr>
</tbody>
</table>
## Microcontroller

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>Read-Only Memory (ROM)</th>
<th>Read-Write Memory</th>
<th>Timer</th>
<th>I/O Port</th>
<th>Serial Interface</th>
</tr>
</thead>
</table>

Microcontroller: a complete package
# Embedded Systems

<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 2MB RAM</td>
<td>Accelerometer, Mic, Camera</td>
<td>None</td>
</tr>
<tr>
<td>Arduino Nano 33 BLE Sense</td>
<td>32-bit nRF52840</td>
<td>64 MHz</td>
<td>1MB flash 256kB RAM</td>
<td>Mic, IMU, Temp, Humidity, Gesture, Pressure, Proximity, Brightness, Color</td>
<td>BLE</td>
</tr>
<tr>
<td>SparkFun Edge 2</td>
<td>32-bit ArtemisV1</td>
<td>48 MHz</td>
<td>1MB flash 384kB RAM</td>
<td>Accelerometer, Mic, Camera</td>
<td>BLE</td>
</tr>
<tr>
<td>Espressif EYE</td>
<td>32-bit ESP32-D0WD</td>
<td>240 MHz</td>
<td>4MB flash 520kB RAM</td>
<td>Mic, Camera</td>
<td>WiFi, BLE</td>
</tr>
</tbody>
</table>
## Embedded Systems

<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 2MB RAM</td>
<td>Accelerometer, Mic, Camera</td>
<td>None</td>
</tr>
<tr>
<td><strong>Arduino Nano 33 BLE Sense</strong></td>
<td><strong>32-bit nRF52840</strong></td>
<td>64 MHz</td>
<td>1MB flash 256kB RAM</td>
<td>Mic, IMU, Temp, Humidity, Gesture, Pressure, Proximity, Brightness, Color</td>
<td>BLE</td>
</tr>
<tr>
<td>SparkFun Edge 2</td>
<td>32-bit ArtemisV1</td>
<td>48 MHz</td>
<td>1MB flash 384kB RAM</td>
<td>Accelerometer, Mic, Camera</td>
<td>BLE</td>
</tr>
<tr>
<td>Espressif EYE</td>
<td>32-bit ESP32-D0WD</td>
<td>240 MHz</td>
<td>4MB flash 520kB RAM</td>
<td>Mic, Camera</td>
<td>WiFi, BLE</td>
</tr>
</tbody>
</table>
Nano 33 BLE Sense (+ USB cable)

Purpose

AI-enabled developmental microcontroller board with USB-A to microB cable

Specifications

- nRF52840 MCU (ARM Cortex-M4): 3.3V, 64MHz, 1MB flash, 256 kB RAM
- Sensors on board: microphone, IMU, color, light, proximity, temperature, humidity, and more!
- BLE module with application-adjacent protocol layers (GAP, GATT) covered by ArduinoBLE library
OV 7675 Camera Module

Purpose

Breakout PCB for *tiny* camera, for use in person-detection exercises

Specifications

- Low-voltage, 0.3 MP CMOS VGA (can step down to QVGA, QQVGA) image sensor
- Serial Camera Control Bus (SCCB) + Camera Parallel Interface (CPI) / Digital Video Port (DVP) interface
- Breaks ribbon cable out to 2x10 pin array
Tiny Machine Learning Shield

Purpose

A daughter PCB designed to **breakout the I/O** from the Nano 33 BLE sense to permit easy, reliable **communication with** other local, **off-board elements**

Specifications

- Grove connectors (3.3V I2C and simple digital / analog - see pinouts)
- 2x10 pin array for OV7675 camera module
- Voltage input terminal block, accepts 4.5 to 21V (down regulated to 3.3V on Nano 33)
Today’s Agenda

- Review + Why Tiny?
- Hardware Basics
  - Installing and Starting the Arduino IDE
  - Testing Your TinyML Kit
- Deploying KWS model onto Arduino
- Summary & Next Steps
Today’s Agenda

- Review + Why Tiny?
- Hardware Basics

Installing and Starting the Arduino IDE

- Testing Your TinyML Kit
- Deploying KWS model onto Arduino
- Summary & Next Steps
Downloads

Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the Getting Started page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is hosted by GitHub. See the instructions for building the code. Latest release source code archives are available here. The archives are PGP-signed so they can be verified using this gpg key.

DOWNLOAD OPTIONS

- Windows Win 7 and newer
- Windows ZIP file
- Windows app Win 8.1 or 10
- Linux 32 bits
- Linux 64 bits
- Linux ARM 32 bits
- Linux ARM 64 bits
- Mac OS X 10.10 or newer

Release Notes

Checksums (sha512)
Downloads

Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the Getting Started page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is hosted by GitHub. See the instructions for building the code. Latest release source code archives are available here. The archives are PGP-signed so they can be verified using this gpg key.

DOWNLOAD OPTIONS

Windows  Win 7 and newer
Windows  ZIP file
Windows app  Win 8.1 or 10  Get
Linux  32 bits
Linux  64 bits
Linux  ARM 32 bits
Linux  ARM 64 bits
Mac OS X  10.10 or newer
Downloads

Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the Getting Started page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is hosted by GitHub. See the instructions for building the code. Latest release source code archives are available here. The archives are PGP-signed so they can be verified using this gpg key.

DOWNLOAD OPTIONS

Windows  Win 7 and newer
Windows  ZIP file
Windows app  Win 8.1 or 10
Linux  32 bits
Linux  64 bits
Linux  ARM 32 bits
Linux  ARM 64 bits
Mac OS X  10.10 or newer

Release Notes

Checksums (sha512)
Getting Started with Arduino products

Welcome to Arduino! Before you start controlling the world around you, you'll need to set up the software to program your board.

The Arduino Software (IDE) allows you to write programs and upload them to your board. In the Arduino Software page you will find two options:

1. If you have a reliable Internet connection, you should use the online IDE (Arduino Web Editor). It will allow you to save your sketches in the cloud, having them available from any device and backed up. You will always have the most up-to-date version of the IDE without the need to install updates or community generated libraries.

2. If you would rather work offline, you should use the latest version of the desktop IDE.

Code online on the Arduino Web Editor

To use the online IDE simply follow these instructions. Remember that boards work out-of-the-box on the Web Editor, you only need to install Arduino Create Agent to get started.

Install the Arduino Desktop IDE

To get step-by-step instructions select one of the following link accordingly to your operating system.

- Windows
- Mac OS
- Linux
- Portable IDE (Windows and Linux)
- ChromeOS (Arduino Create App) in the Chrome Web Store

Choose your board in the list here on the right to learn how to get started with it and how to use it on the Desktop IDE.

Learn Arduino

- Read an Introduction on what is Arduino and why you'd want to use it.
- What is the Arduino Software (IDE) and how do I change the default language?
Getting Started with Arduino products

Welcome to Arduino! Before you start controlling the world around you, you'll need to set up the software to program your board.

The Arduino Software (IDE) allows you to write programs and upload them to your board. In the Arduino Software page you will find two options:

1. If you have a reliable Internet connection, you should use the online IDE (Arduino Web Editor). It will allow you to save your sketches in the cloud, having them available from any device and backed up. You will always have the most up-to-date version of the IDE without the need to install updates or community generated libraries.

2. If you would rather work offline, you should use the latest version of the desktop IDE.

Code online on the Arduino Web Editor

To use the online IDE simply follow these instructions. Remember that boards work out-of-the-box on the Web Editor, you only need to install Arduino Create Agent to get started.

Install the Arduino Desktop IDE

To get step by step instructions select one of the following link accordingly to your operating system.

- Windows
- Mac OS
- Linux
- Portable IDE (Windows and Linux)
- ChromeOS (Arduino Create App) in the Chrome Web Store

Choose your board in the list here on the right to learn how to get started with it and how to use it on the Desktop IDE.

Learn Arduino

- Read an Introduction on what is Arduino and why you'd want to use it.
- What is the Arduino Software (IDE) and how do I change the default language?
head to Arduino’s software website

www.arduino.cc/en/software

then download and install right now!
```
void setup() {  
  // put your setup code here, to run once:
}

void loop() {  
  // put your main code here, to run repeatedly:
}
```
Install Extras
Tools → Board → **Boards Manager**
Nano 33 BLE
<table>
<thead>
<tr>
<th>Boards Manager</th>
<th>Nano 33 BLE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arduino Mbed OS Nano Boards</strong></td>
<td><strong>3.1.1</strong></td>
</tr>
<tr>
<td>by Arduino</td>
<td>Install</td>
</tr>
<tr>
<td>Boards included in this package:</td>
<td></td>
</tr>
<tr>
<td>Online Help:</td>
<td>More Info</td>
</tr>
</tbody>
</table>

*DEPRECATED - Please install standalone packages*
Tools → **Manage Libraries...**
Tools → Manage Libraries...
<table>
<thead>
<tr>
<th>Library Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arduino_TensorFlowLite</td>
<td>Allows you to run machine learning models locally on your device. This library runs TensorFlow machine learning models on microcontrollers, allowing you to build AI/ML applications powered by deep learning and neural networks. With the included examples, you can recognize speech, detect people using a camera, and recognize “magic wand” gestures using an accelerometer. The examples work best with the Arduino Nano 33 BLE Sense board, which has a microphone and accelerometer.</td>
</tr>
<tr>
<td>Adafruit TensorFlow Lite</td>
<td>This is a library to abstract away TensorFlow Lite for Adafruit/Arcada boards.</td>
</tr>
<tr>
<td>EloquentTinyML</td>
<td>An eloquent Interface to Tensorflow Lite for Microcontrollers</td>
</tr>
<tr>
<td>Harvard_TinyML</td>
<td></td>
</tr>
</tbody>
</table>
Arduino_TensorFlowLite

by Pete Warden

Allows you to run machine learning models locally on your device. This library runs TensorFlow machine learning models on microcontrollers, allowing you to build AI/ML applications powered by deep learning and neural networks. With the included examples, you can recognize speech, detect people using a camera, and recognize "magic wand" gestures using an accelerometer. The examples work best with the Arduino Nano 33 BLE Sense board, which has a microphone and accelerometer.

More info

EloquentTinyML

by Simone Salerno, eloquentarduino@gmail.com

An eloquent interface to Tensorflow Lite for Microcontrollers

More info
Arduino_TensorFlowLite

by Pete Warden

Allows you to run machine learning models locally on your device. This library runs TensorFlow machine learning models on microcontrollers, allowing you to build AI/ML applications powered by deep learning and neural networks. With the included examples, you can recognize speech, detect people using a camera, and recognize "magic wand" gestures using an accelerometer. The examples work best with the Arduino Nano 33 BLE Sense board, which has a microphone and accelerometer.

More info

Version 2.4.0-ALPHA

Install

Adafruit TensorFlow Lite

by Adafruit

This is a library to abstract away TensorFlow Lite for Adafruit/Arcada boards. This is a library to abstract away TensorFlow Lite for Adafruit/Arcada boards.

More info

EloquentTinyML

by Simone Salerno, eloquentarduino@gmail.com

An eloquent interface to TensorFlow Lite for Microcontrollers

More info

Harvard TinyML
Arduino_TensorFlowLite

by Pete Warden

Allows you to run machine learning models locally on your device. This library runs TensorFlow machine learning models on microcontrollers, allowing you to build AI/ML applications powered by deep learning and neural networks. With the included examples, you can recognize speech, detect people using a camera, and recognize "magic wand" gestures using an accelerometer. The examples work best with the Arduino Nano 33 BLE Sense board, which has a microphone and accelerometer.

More info

Version 2.4.0-ALPHA

Install
Install Extras
### Arduino_OV767X

**by Arduino**

Capture images from your OmniVision OV7670 camera in your Arduino sketches.

[More info]

### Harvard_TinyMLx

**by Brian Planche**  
**Version 1.1.0-Alpha INSTALLED**

Supports the TinyML edX Course and TinyML Shield. This library supports the TinyML Shield and provides examples that support the TinyML edX course. The examples work best with the Arduino Nano 33 BLE Sense board and the Tiny Machine Learning Kit from Arduino. It also includes a modified version of the Arduino_OV767X library version 0.0.2 and a fork of the TensorFlow_Lite version 2.4.0-Alpha Arduino examples.

[More info]
### Arduino_OV767X

**by Arduino**  
Capture images from your OmniVision OV7670 camera in your Arduino sketches.  
[More info](#)

**Version 0.0.2**

### Harvard_TinyMLx

**by Brian Plancher**  
Version 1.1.0-Alpha  
INSTALLED  
Supports the TinyML edX Course and TinyML Shield. This library supports the TinyML Shield and provides examples that support the TinyML edX course. The examples work best with the Arduino Nano 33 BLE Sense board and the Tiny Machine Learning Kit from Arduino. It also includes a modified version of the Arduino_OV767X library version 0.0.2 and a fork of the TensorFlow_Lite version 2.4.0-Alpha Arduino examples.  
[More info](#)
<table>
<thead>
<tr>
<th>Library Manager</th>
<th>Arduino_OV767X</th>
<th>767x</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong></td>
<td><strong>All</strong></td>
<td><strong>Topic</strong></td>
</tr>
<tr>
<td>Arduino_OV767X</td>
<td>Arduino</td>
<td>Capture images from your OmniVision OV7670 camera in your Arduino sketches.</td>
</tr>
</tbody>
</table>

**Arduino_OV767X**

Version 0.0.2
**Harvard_TinyMLx**

*Version 1.1.0-ALPHA*

Supports the TinyML edX Course and TinyML Shield. This library supports the TinyML Shield and provides examples that support the TinyML edX course. The examples work best with the Arduino Nano 33 BLE Sense board and the Tiny Machine Learning Kit from Arduino. It also includes a modified version of the Arduino_OV7670 library version 0.0.2 and a fork of the TensorFlow_Lite version 2.4.0-Alpha Arduino examples.
Install Extras
Arduino_LSM9DS1

Version 1.1.0

by Arduino

Allows you to read the accelerometer, magnetometer and gyroscope values from the LSM9DS1 IMU on your Arduino Nano 33 BLE Sense.

Adafruit LSM9DS1 Library

version 1.1.0

Install
Tools → Board: “[something]”

→ Arduino Mbed OS Nano Boards

→ Arduino Nano 33 BLE
Plug in via MicroUSB to your computer right now!
Tools → Port: “[???] (Arduino Nano 33 BLE)”

→ <???> (Arduino Nano 33 BLE)

Mac → /dev/cu.usbmodem<#>
Windows → COM<#>
Linux → ttyUSB<#> or ttyACM<#>
Today’s Agenda

- Review + Why Tiny?
- Hardware Basics
- Installing and Starting the Arduino IDE

Testing Your TinyML Kit

- Deploying KWS model onto Arduino
- Summary & Next Steps
File → Examples
→ Harvard_TinyMLx
→ test_microphone
#include <PDM.h>
#include <TinyMLShield.h>

// PDM buffer
short sampleBuffer[256];
volatile int samplesRead;
bool record = false;
bool commandRecv = false;

void setup() {
    Serial.begin(9600);
    while (!Serial);
    // Initialize the TinyML Shield
    initializeShield();
}

Done uploading.
Locked : none
Security : false
Erase flash
Done in 0.081 seconds
Write 93144 bytes to flash (23 pages)
[==================================] 100% (23/23 pages)
Done in 3.575 seconds
An error occurred while uploading the sketch

/home/plancher/Arduino/libraries/test_image_2_inferencing/src/edge-impulse-sdk/CMSIS/NN/Source/PoolingFunctions/arm_pool_q7_HW *
SIMD32(pCnt)++ = QADD16(vo2, In);

/home/plancher/Arduino/libraries/test_image_2_inferencing/src/edge-impulse-sdk/tensorflow/lite/core/api/op_resolver.cpp: In function 'builtin_code< BuiltInOperator_MIN> {'

Sketch uses 224024 bytes (22%) of program storage space. Maximum is 983946 bytes.
Global variables use 58872 bytes (22%) of dynamic memory, leaving 203472 bytes for local variables. Maximum is 262144 bytes.
An error occurred while uploading the sketch

Device unsupported

Double Tap Reset for Bootloader Mode!
Tools → **Serial Monitor**
Tools → **Serial Monitor**

Welcome to the microphone test for the built-in microphone on the Nano 33 BLE Sense

Use the on-shield button or send the command 'click' to start and stop an audio recording.
Open the Serial Plotter to view the corresponding waveform.
Tools → Serial Monitor

Welcome to the microphone test for the built-in microphone on the Nano 33 BLE Sense

Use the on-shield button or send the command 'click' to start and stop an audio recording.

Open the Serial Plotter to view the corresponding waveform.

9600 baud

Both! NL & CR
Welcome to the microphone test for the built-in microphone on the Nano 33 BLE Sense

Use the on-shield button or send the command 'click' to start and stop an audio recording
Open the Serial Plotter to view the corresponding waveform
Tools → **Serial Monitor**

![Serial Monitor Window](image)

**COM5**

**NUMBERS FLYING BY**
Tools → Serial Plotter
Tools → Serial Plotter

How clean of a wave can you get?
Today’s Agenda

- Review + Why Tiny?
- Hardware Basics
- Installing and Starting the Arduino IDE
- Testing Your TinyML Kit

Deploying KWS model onto Arduino

- Summary & Next Steps
DEPLOYMENT (TEST IMAGE 2)

Deploy your impulse

You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. Read more.

Create library

Turn your impulse into optimized source code that you can run on any device.

Build firmware

Get a ready-to-go binary for your development board that includes your impulse.
Select optimizations (optional)

Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.

Enable EON™ Compiler
Same accuracy, up to 50% less memory. Open source.

Available optimizations for Transfer learning:

<table>
<thead>
<tr>
<th>Optimization</th>
<th>RAM USAGE</th>
<th>LATENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantized (Int8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Currently selected</td>
<td>66.1K</td>
<td>58 ms</td>
</tr>
<tr>
<td>Unoptimized (float32)</td>
<td>108.1K</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optimization</th>
<th>RAM USAGE</th>
<th>LATENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unoptimized (float32)</td>
<td>155.6K</td>
<td>43 ms</td>
</tr>
<tr>
<td></td>
<td>193.8K</td>
<td>-</td>
</tr>
</tbody>
</table>
Select optimizations *(optional)*

Model optimizations can increase on-device performance but may reduce accuracy. Click below to analyze optimizations and see the recommended choices for your target. Or, just click Build to use the currently selected options.

**Enable EON™ Compiler**
Same accuracy, up to 50% less memory. Open source.

### Available optimizations for Transfer learning

<table>
<thead>
<tr>
<th></th>
<th>RAM USAGE</th>
<th>LATENCY</th>
<th>ACCURACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantized (Int8)</td>
<td>66.1K</td>
<td>58 ms</td>
<td></td>
</tr>
<tr>
<td>Unoptimized (float32)</td>
<td>155.6K</td>
<td>43 ms</td>
<td></td>
</tr>
<tr>
<td><strong>Currently selected</strong></td>
<td><strong>108.1K</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Unoptimized</strong></td>
<td><strong>193.8K</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Estimate for A规律o ARM Cortex-M7 (Corex-M7 400 MHz)

[Build]
Quantization

Reduces the precision of numbers used in a model which results in:

- **smaller** model size
- **faster** computation
Reducing the Precision

4 bytes per model parameter

1 byte per model parameter

float32

int8

max(|x|)

min(|x|)

max: 3.40282e+38
min: 1.17549e-38
## Tradeoff

<table>
<thead>
<tr>
<th>Model</th>
<th>Floating-point Baseline</th>
<th>After Quantization</th>
<th>Accuracy Drop</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileNet v1 1.0 224</td>
<td>71.03%</td>
<td>69.57%</td>
<td>▼1.46%</td>
</tr>
<tr>
<td>MobileNet v2 1.0 224</td>
<td>70.77%</td>
<td>70.20%</td>
<td>▼0.57%</td>
</tr>
<tr>
<td>Resnet v1 50</td>
<td>76.30%</td>
<td>75.95%</td>
<td>▼0.35%</td>
</tr>
</tbody>
</table>
### Model

#### Last training performance (validation set)

**ACCUACY**

- **float32**: 100.0%
- **int8**: 70.0%

**LOSS**

- **float32**: 0.25
- **int8**: 0.33

#### Confusion matrix (validation set)

<table>
<thead>
<tr>
<th></th>
<th>CAR</th>
<th>TRUCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAR</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>TRUCK</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

#### Feature explorer (full training set)

- car - correct
- truck - correct
- car - incorrect
- truck - incorrect

#### On-device performance

- **float32**: INFERENCING TIME 43 ms., PEAK RAM USAGE 155.6K, FLASH USAGE 193.8K
- **int8**: INFERENCING TIME 58 ms., PEAK RAM USAGE 66.1K, FLASH USAGE 108.1K
**float32**

**Model**

Last training performance (validation set)

- **ACCUacy**: 100.0%
- **LOSs**: 0.25

Confusion matrix (validation set)

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Dog</td>
<td>0%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Feature explorer (validation set)

On-device performance

- **INFERENCING TIME**: 43 ms.
- **PEAK RAM USAGE**: 155.6K
- **FLASH USAGE**: 193.8K

**int8**

**Model**

Last training performance (validation set)

- **ACCUacy**: 70.0%
- **LOSs**: 0.33

Confusion matrix (validation set)

<table>
<thead>
<tr>
<th></th>
<th>Cat</th>
<th>Dog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>80%</td>
<td>20%</td>
</tr>
<tr>
<td>Dog</td>
<td>20%</td>
<td>80%</td>
</tr>
</tbody>
</table>

Feature explorer (validation set)

On-device performance

- **INFERENCING TIME**: 58 ms.
- **PEAK RAM USAGE**: 66.1K
- **FLASH USAGE**: 108.1K
Built Arduino library

Add this library through the Arduino IDE via:

Sketch > Include Library > Add .ZIP Library...

Examples can then be found under:

File > Examples > YOUR_PROJECT_NAME
Sketch → Include Library

→ Add .ZIP Library
Built Arduino library

Add this library through the Arduino IDE via:

Sketch > Include Library > Add .ZIP Library...

Examples can then be found under:

File > Examples > YOUR_PROJECT_NAME
File → Examples →

YOUR_PROJECT_NAME →

nano_ble33_sense_microphone
done?
Double Tap Reset for Bootloader Mode!
Tools → Serial Monitor
Confidence that the audio is one of the choices (0-1 scale)
Today’s Agenda

- Review + Why Tiny?
- Hardware Basics
- Installing and Starting the Arduino IDE
- Testing Your TinyML Kit
- Deploying KWS model onto Arduino

Summary & Next Steps
Machine Learning

WE PROVIDE

LABELS

INPUTS

RULES

THE COMPUTER LEARNS

ANSWERS
Deep Learning with Neural Networks
Features can be found with **Convolutions**

<table>
<thead>
<tr>
<th>-2</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
The (Tiny) Machine Learning Workflow

Collect & Transform Data
Design & Train a Model
Deploy Model
Make Inferences

ML is everywhere → collect GOOD data RESPONSIBLY
The (Tiny) Machine Learning Workflow

Collect & Transform Data

Design & Train a Model

Deploy Model

Make Inferences

No Loud

No Loud

No Loud
The (Tiny) Machine Learning Workflow

- Collect & Transform Data
- Design & Train a Model
- Deploy Model
- Make Inferences

The infographic shows a flowchart with steps for each category, followed by a neural network diagram with layers and dimensions.
The (Tiny) Machine Learning Workflow

Edge Impulse Simplifies Training and Deployment
The (Tiny) Machine Learning Workflow

1. Collect & Transform Data
2. Design & Train a Model
3. Deploy Model
4. Make Inferences

Even less memory
Even less compute power
Also, only focused on inference
Confidence that the audio is one of the choices (0-1 scale)
Workshop Agenda

Day 1
- Introduction to AI and (Tiny)ML
- Cloud ML

Day 2
- Keyword Spotting for the Navajo Language
- Mobile ML

Day 3
- Bringing AI/ML from the Cloud to the Edge
- Embedded ML
Workshop Agenda

Day 1
Introduction to AI and (Tiny)ML
Cloud ML

Day 2
Keyword Spotting for the Navajo Language
Mobile ML

Day 3
Bringing AI/ML from the Cloud to the Edge
Embedded ML
Practice with more machine learning!
Experiments with Google

AI + MUSIC

**FREDDIEMETER**
by Google Research, Google Creative Lab, YouTube Music
An AI-powered singing challenge that rates how closely your singing matches the voice

**AI DUET**
by Yotam Mann
A piano that responds to you.

**SEMI-CONDUCTOR**
by Google Creative Lab
Conduct your own orchestra in the browser by moving your arms

[experiments.withgoogle.com](http://experiments.withgoogle.com)
Teachable Machine

Train a computer to recognize your own images, sounds, & poses.

A fast, easy way to create machine learning models for your sites, apps, and more – no expertise or coding required.

Get Started

teachablemachine.withgoogle.com
Classification result

Summary

Name: helloworld.jan5.wav.1ncrr7qm.s17
Expected outcome: helloworld

CATEGORY | COUNT
---|---
helloworld | 0
noise | 0
unknown | 1
uncertain | 0

Detailed result

<table>
<thead>
<tr>
<th>TIMESTAMP</th>
<th>HELLOWORLD</th>
<th>NOISE</th>
<th>UNKNOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.36</td>
<td>0.01</td>
<td>0.62</td>
</tr>
</tbody>
</table>

Raw data: `helloworld.jan5.wav.1ncrr7qm.s17`

Raw features:

- 37, 34, 42, 36, 14, 1, -3, -9, -7, -18, -28, -29, -26, -21, -23, -

MFCC (1,649 samples)

X Axis
- Visualization layer

Y Axis
- Visualization layer

Z Axis
- Visualization layer

- helloworld
- noise
- unknown
- classification 0
Neural Networks

From the ground up

3Blue1Brown
The Future of ML is Tiny and Bright

Courses in this program

- HarvardX's Tiny Machine Learning (TinyML) Professional Certificate
  - Fundamentals of TinyML
  - Applications of TinyML
  - Deploying TinyML

edx.org
Explore projects

- Getting Started with the Raspberry Pi Pico
  Arduino "havingIt" Guy
  - 15 views, 2.5K members

- 64-Key Prototyping Keyboard Matrix for Arduino
  Cameron Coward
  - 19 views, 6.8K members

- ML-Based Bird and Squirrel Detector (Raspberry Pi and AWS)
  Mike Sadowski
  - 31 views, 5.3K members

- Self-Playing Melodica 🎹
  touchmysound
  - 30 views, 3.9K members

[Link to hackster.io]
Our website!

tinyMLedu.org/EASI-22

home base for all information!
survey!

bit.ly/EASI22-Post

help us make this workshop better!
Our team!

with help from many more
hágoónee’ 🙋

thank you and keep exploring with ML