tinyML® EMEA
Enabling Ultra-low Power Machine Learning at the Edge

June 26 - 28, 2023

www.tinyML.org
How to build an ML-powered doorbell notifier

Sandeep Mistry
June 26, 2023

© 2023 Arm
Machine Learning + Microcontroller based solution

On-device ML Inferencing = Privacy Preserving
Audio Classification – FSD50K subset

- Doorbell - 

- Music - 🎶

- Domestic and home sounds - 🏡

- Human voice - 🗣

- Hands (clapping, finger snapping) - 👏 🫰
How-to Get Started with Machine Learning on Arduino

October 15, 2019

A guest post by Sandeep Misty & Dominic Pajak of the Arduino team

Arduino is on a mission to make Machine Learning simple enough for anyone to use. We’ve been working with the TensorFlow Lite team over the past few months and are excited to show you what we’ve been up to together: bringing TensorFlow Lite Micro to the Arduino Nano 33 BLE Sense. In this article, we’ll show you how to install and run several new TensorFlow Lite Micro examples that are now available in the Arduino Library Manager.

The first tutorial below shows you how to install a neural network on your Arduino board to recognize simple voice commands.
ML Model
“tiny_conv” ML model

TensorFlow v1 vs TensorFlow v2 with Keras

```python
import tensorflow as tf

first_conv = tf.keras.layers.Conv2D(8, kernel_size=(3, 3), activation='relu', padding='same', input_shape=(49, 49, 1))

# ... more code...
```

© 2023 Arm
🎶 Input Signal

1 second of audio @ 16 kHz = 16,000 samples
Preprocessing

Sample Rate = 16 kHz

- Frame Length = 30 ms
  \[
  \frac{30}{1000} \times 16000 = 480 \text{ samples}
  \]

- Frame Step = 20 ms
  \[
  \frac{20}{1000} \times 16000 = 320 \text{ samples}
  \]

- FFT Size = 256

In this section we discuss spectrograms, the preprocessed speech input to the model. Here's an illustration of the process:

The model doesn't take in raw audio sample data, instead it works with spectrograms which are two dimensional arrays that are made up of slices of frequency information, each taken from a different time window.
tf.signal - spectrogram

```python
import tensorflow as tf

# samples = 1 second of audio = 16,000 samples

spectrogram = tf.math.abs(
    tf.signal.stft(
        samples,
        frame_length=480,
        frame_step=320,
        fft_length=256,
        window_fn=tf.signal.hann_window,
        pad_end=False,
    )
)

# spectrogram.shape = (49, 129) Not 49 x 40 😞
```

tf.signal - Mel Weight Matrix

Human perception of audio frequencies

```python
import tensorflow as tf

mel_weight_matrix = tf.signal.linear_to_mel_weight_matrix(
    num_mel_bins=40,
    num_spectrogram_bins=129,
    sample_rate=16000,
    lower_edge_hertz=0,
    upper_edge_hertz=8000,
)

# mel_weight_matrix.shape = (129, 40)
```
tf.io - Mel spectrogram

```python
1 import tensorflow_io as tfio
2
3 # spectrogram.shape = (49, 129)
4
5 mel_spectrogram = tfio.audio.melscale(
6     spectrogram,
7     rate=16000,
8     mels=40,
9     fmin=0,
10    fmax=8000
11 )
12
13 # mel_spectrogram.shape = (49, 40)
```
tf.io - Mel power spectrogram (dB)

```python
1 import tensorflow as tf
2 import tensorflow_io as tfio
3 # mel_power = 10 * log(mel * mel) / log(10)
4 mel_spectrogram = tf.maximum(1e-6, mel_spectrogram)
5 dbscale_mel_spectrogram = tfio.audio.dbscale(
  6    mel_spectrogram,
  7    top_db=80
  8 )
```

Mel power spectrogram of doorbell sound
DSP + ML Model

16,000 samples

49 x 40 “2d image”
Model Training Flow

tf.data.Dataset pipeline

- Read Audio File: `tf.io.read_file(…)`
- Decode Wave Data: `tf.audio.decode_wav(…)`
- Re-sample to 16 kHz: `tfio.audio.resample(…)`
- Trim Silence: `tfio.audio.trim(…)`
- Frame: `tf.signal.frame(…)`
- Mel Power Spectrogram: # steps from previous slides

# create 1 s slices with 0.1 s of overlap
Model Training Flow
Train “tiny_conv” model and convert

Train baseline model
Using ESC-50K data, 50 classes

Train model
Using subset of FSD50K data and Transfer Learning
--> same DepthWiseConv2D weights as baseline model
--> new classification head, 5 classes

Convert model to .tflite
8-bit inputs and outputs, quantized weights
Sending SMS messages

Twilio REST API

Twilio Console

HTTP POST

- URL
  
  https://api.twilio.com/2010-04-01/Accounts/<Account SID>/Messages.json

- Auth Header = HTTP Basic Auth
  - Username = Account SID
  - Password = Auth Token

- Body = application/x-www-form-urlencoded
  - To = the phone # to send the message to
  - From = the Twilio # the message is from
  - Body = the message text
SparkFun AzureWave ThingPlus
Realtek RTL8721DM SoC - compute, connectivity, and audio

- Arm Cortex-M33 compatible Real-M300 CPU @ 200 MHz
- 512 KB of SRAM and 4 MB of PSRAM
- 4 MB of flash
- Built-in 2.4 GHz and 5 GHz Wi-Fi connectivity
- Built-in audio codec with two analog inputs

Arduino IDE support with the Realtek Arduino core
MEMS Microphone Input

SparkFun SPH8878LR5H-1

<table>
<thead>
<tr>
<th>SparkFun AzureWave Thing+</th>
<th>Analog MEMS Microphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>3V3</td>
<td>VCC</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>22 (PA4)</td>
<td>AUD</td>
</tr>
</tbody>
</table>
# Arduino Libraries

<table>
<thead>
<tr>
<th>Name</th>
<th>Provider</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AudioCodec</td>
<td>Realtek</td>
<td>Used to control and manage the hardware Audio Codec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Provides audio data 1024 bytes at a time = 512 samples @ 16-bit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>With sample rate = 16,000 Hz have 32 ms to process audio in real-time</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 32 ms = 0.320 s = 512 samples / 16,000 samples per second</td>
</tr>
<tr>
<td>WiFi</td>
<td>Realtek</td>
<td>Used to control and manage the Wi-Fi interface and UDP or TCP sockets</td>
</tr>
<tr>
<td>Ameba_TensorFlowLite</td>
<td>Realtek</td>
<td>Provides TensorFlow Lite for Microcontroller (TFLM) support for the board</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Includes Arm's <strong>CMSIS-NN</strong> library, which provides optimized Neural Network compute kernels for Arm Cortex-M processors</td>
</tr>
<tr>
<td>CMSIS-DSP</td>
<td>Arm</td>
<td>Optimized Digital Signal Processing on Arm Cortex-M</td>
</tr>
<tr>
<td>ArduinoHttpClient</td>
<td>Arduino</td>
<td>Used to interact with HTTP + REST API’s</td>
</tr>
</tbody>
</table>
Arduino Sketch pseudo code

```cpp
setup()

// Initialize DSP pipeline
Initialize DSP pipeline

// Initialize ML Model
Initialize ML Model

// Connect to the Wi-Fi Network
Connect to the Wi-Fi Network

// Start Audio Codec
Start Audio Codec

Mono 16-bit @ 16 kHz
```
Arduino Sketch pseudo code

```cpp
loop()

Wait for new Audio samples

Update Mel Spectrogram with new Audio samples

ML Inferencing

Exponential Smoothing

\[ s_t = \alpha x_t + (1 - \alpha)s_{t-1} \]

Doorbell sound detected and over 30 s since last detection

Send SMS
```
Processing Step | Time
--- | ---
Mel Power Spectrogram | ~3.4 ms
Model Inferencing | ~14.0 ms
Total | ~17.4 ms

Under the 32 ms goal for real-time processing!
Recap

Audio
Compute
Connectivity

Twilio
Learn More ...

- Demo at Arm booth

- Hackster.io
  - https://www.hackster.io/sandeep-mistry/how-to-build-an-ml-powered-doorbell-notifier-0a781e

- GitHub
  - https://github.com/ArmDeveloperEcosystem/aiot-doorbell-notifier-example-for-ameba
Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
ありがとう
Asante
Merci
감사합니다
धन्यवाद
شكرًا
ধন্যবাদ
תודה
Copyright Notice

This presentation in this publication was presented as a tinyML® EMEA Innovation Forum. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org