TensorFlow Lite for microcontrollers

Daniel Situnayake
@dansitu
Agenda

- What are TensorFlow and TensorFlow Lite?
- Why TensorFlow for Microcontrollers?
- Demo (hotword detection)
- Code walkthrough
- How to get started
- → Deep dive with Nat Jeffries
What is TensorFlow?

End-to-end open source ML platform

TensorFlow is a machine learning platform that enables you to **develop, train and deploy** machine learning and deep learning models anywhere.
1. Load & preprocess data
   - TENSORFLOW
   - SWIFT FOR TF
   - TFX

2. Build, train, and reuse models
   - TENSORFLOW
   - TENSORFLOW.JS
   - SWIFT FOR TF
   - TFX

3. Deploy
   - TENSORFLOW
   - TENSORFLOW.JS
   - SWIFT FOR TF
   - TENSORFLOW LITE
   - TFX
   - TENSORBOARD
TensorFlow.js

---

TF.js
TFLite
ML Kit

---

Desktop & mobile web

Embedded Linux (Raspberry Pi)

Android & iOS

Microcontrollers

ML Kit

TensorFlow.js

TensorFlow Lite
TensorFlow (estimator or Keras) → Saved Model → TF Lite Converter → TF Lite Model
Get up and running

Load your model → Transforming data → Run inference → Use the resulting output
What are they?

Small computer on a single circuit

- No operating system
- Tens of KB of RAM & Flash
- Only CPU, memory & I/O peripherals
TensorFlow provides you with a single framework to deploy on Microcontrollers as well as phones.
Demo

Hotword detection with a 20kb model
// Load a model from an array of bytes
const tflite::Model* model =
    ::tflite::GetModel(g_tiny_conv_micro_features_model_data);

// Pull in all the operation implementations we need
tflite::ops::micro::AllOpsResolver resolver;

// Create an area of memory to use for input, output, and intermediate arrays
const int tensor_arena_size = 10 * 1024;
uint8_t tensor_arena[tensor_arena_size];
tflite::SimpleTensorAllocator tensor_allocator(tensor_arena,
                                               tensor_arena_size);

// Build an interpreter to run the model with
tflite::MicroInterpreter interpreter(model, resolver, &tensor_allocator,
                                      error_reporter);
// Get information about the memory area to use for the model's input
TfLiteTensor* model_input = interpreter.input(0);

// Prepare to generate the features that will be input to the neural network
FeatureProvider feature_provider(kFeatureElementCount,
        model_input->data.uint8);

// Perform feature extraction and populate the input array
feature_provider.PopulateFeatureData(...)

// Run the model
TfLiteStatus invoke_status = interpreter.Invoke();
TfLiteTensor* output = interpreter.output(0);

uint8_t top_category_score = 0, top_category_index = 0;

for (int category_index = 0; category_index < kCategoryCount;
     ++category_index) {
    const uint8_t category_score = output->data.uint8[category_index];
    if (category_score > top_category_score) {
        top_category_score = category_score;
        top_category_index = category_index;
    }
}

https://github.com/tensorflow/tensorflow/blob/master/tensorflow/
lite/experimental/micro/examples/micro_speech/main.cc
MCU
Is there any sound?

MCU
Is that human speech?

Deeper Network

Application Processor
Roadmap

- More device support (Arduino coming soon)
- Optimizations (CMSIS-NN)
- More sample code
Get started

tensorflow.org/lite/microcontrollers/overview
TinyML
Machine Learning with TensorFlow on Arduino and Ultra-Low Power Micro-Controllers

Pete Warden & Daniel Situnayake