Overview

● Why TinyML
● Why TFLite for Microcontrollers is exciting today
● Applications for TFLite for Microcontrollers
● Current and Future Work
● How to get involved
Opportunity

- There is a **growing market** for Microcontrollers

![MCU Market History and Forecast](source: IC Insights)
Problem

- Products demand increasingly sophisticated features
- Tight power constraints
Solution: Tiny ML

- CPU cycles and sensor reads are **power cheap**
- Communicating off-device is **power expensive**
- Tiny ML models can **condense** sensor data into meaningful results
Why Now

- Recently able to shrink TF models to fit on microcontrollers
- CVPR Visual Wakewords Challenge
- Excitement among researchers and companies
How: TFLite for Micros

Train Tiny Models Using TF Lite → Develop Portable Platform → Efficiently Run Models
Developing Tiny ML on Micros is Hard!
Audio Applications

- Speech detection
- Speaker ID
- Wake words
Video Applications

- Next gen cell phones
- Camera based sensors
Micro Vision Demo
// Load a model from an array of bytes
const tflite::Model* model =
    ::tflite::GetModel(g_person_detect_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 = 300 * 1024;
tfLite::TensorAllocator 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 image_provider(ImageFormat.FORMAT_RGB,
                                model_input->data.uint8);

// Perform image capture and populate the input array
image_provider.get_image();

// Run the model
TfLiteStatus invoke_status = interpreter.Invoke();
// Figure out the highest scoring category between person and not person
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_vision/main.cc
Sensor Fusion

- Industrial equipment
- Environmental sensing
- Lots of long-tail opportunities
Roadmap

- **Open source** vision demo on publicly available hardware
- Optimize interpreter **speed**
- Optimize interpreter **memory usage**
- Enable more platforms (Arduino, Mbed)
Get Involved

● Come afterwards to see the micro vision demo
● **Download** the demo from the Tensorflow GitHub and **try it yourself** when it’s available
● Do cool projects and share demos with us!
Questions?
njeff@google.com