TinyML

Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers

Pete Warden and Daniel Situnayake

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Table of Contents

Preface xiii

1. Introduction 1
Embedded Devices 3
Changing Landscape 4

2. Getting Started 5
Who Is This Book Aimed At? 5
What Hardware Do You Need? 6
What Software Do You Need? 7
What Do We Hope You'll Learn? 8

3. Getting Up to Speed on Machine Learning 11
What Machine Learning Actually Is 12
The Deep Learning Workflow 13
Decide on a Goal 14
Collect a Dataset 14
Design a Model Architecture 16
Train the Model 21
Convert the Model 26
Run Inference 26
Evaluate and Troubleshoot 27
Wrapping Up 28

4. The "Hello World" of TinyML: Building and Training a Model 29
What We're Building 30
Our Machine Learning Toolchain 32
Python and Jupyter Notebooks 32

Google Colaboratory 33
TensorFlow and Keras 33
Building Our Model 34
Importing Dependencies 35
Generating Data 38
Splitting the Data 41
Defining a Basic Model 42
Training Our Model 46
Training Metrics 48
Graphing the History 49
Improving Our Model 54
Testing 58
Converting the Model for TensorFlow Lite 60
Converting to a C File 64
Wrapping Up 65

5. The "Hello World" of TinyML: Building an Application..... 67
Walking Through the Tests 68
Including Dependencies 69
Setting Up the Test 70
Getting Ready to Log Data 70
Mapping Our Model 72
Creating an AllOpsResolver 74
Defining a Tensor Arena 74
Creating an Interpreter 75
Inspecting the Input Tensor 75
Running Inference on an Input 78
Reading the Output 80
Running the Tests 82
Project File Structure 85
Walking Through the Source 86
Starting with main_functions.cc 87
Handling Output with output_handler.cc 90
Wrapping Up main_functions.cc 91
Understanding main.cc 91
Running Our Application 92
Wrapping Up 93

6. The "Hello World" of TinyML: Deploying to Microcontrollers..... 95
What Exactly Is a Microcontroller? 96
Arduino 97
Handling Output on Arduino 98

Running the Example 101
Making Your Own Changes 106
SparkFun Edge 106
Handling Output on SparkFun Edge 107
Running the Example 110
Testing the Program 117
Viewing Debug Data 118
Making Your Own Changes 118
ST Microelectronics STM32F746G Discovery Kit 119
Handling Output on STM32F746G 119
Running the Example 124
Making Your Own Changes 126
Wrapping Up 126

7. Wake-Word Detection: Building an Application 127
What We're Building 128
Application Architecture 129
Introducing Our Model 130
All the Moving Parts 132
Walking Through the Tests 133
The Basic Flow 134
The Audio Provider 138
The Feature Provider 139
The Command Recognizer 145
The Command Responder 151
Listening for Wake Words 152
Running Our Application 156
Deploying to Microcontrollers 156
Arduino 157
SparkFun Edge 165
ST Microelectronics STM32F746G Discovery Kit 175
Wrapping Up 180

8. Wake-Word Detection: Training a Model 181
Training Our New Model 182
Training in Colab 182
Using the Model in Our Project 197
Replacing the Model 197
Updating the Labels 198
Updating command_responder.cc 198
Other Ways to Run the Scripts 201
How the Model Works 202

Visualizing the Inputs 202
How Does Feature Generation Work? 206
Understanding the Model Architecture 208
Understanding the Model Output 213
Training with Your Own Data 214
The Speech Commands Dataset 215
Training on Your Own Dataset 216
How to Record Your Own Audio 216
Data Augmentation 218
Model Architectures 219
Wrapping Up 219

9. Person Detection: Building an Application. 221
What We're Building 222
Application Architecture 224
Introducing Our Model 224
All the Moving Parts 225
Walking Through the Tests 227
The Basic Flow 227
The Image Provider 231
The Detection Responder 232
Detecting People 233
Deploying to Microcontrollers 236
Arduino 236
SparkFun Edge 246
Wrapping Up 257

10. Person Detection: Training a Model. 259
Picking a Machine 259
Setting Up a Google Cloud Platform Instance 260
Training Framework Choice 268
Building the Dataset 269
Training the Model 270
TensorBoard 272
Evaluating the Model 274
Exporting the Model to TensorFlow Lite 274
Exporting to a GraphDef Protobuf File 274
Freezing the Weights 275
Quantizing and Converting to TensorFlow Lite 275
Converting to a C Source File 276
Training for Other Categories 277
Understanding the Architecture 277

Wrapping Up 278

11. Magic Wand: Building an Application. 279
What We're Building 282
Application Architecture 283
Introducing Our Model 284
All the Moving Parts 284
Walking Through the Tests 285
The Basic Flow 286
The Accelerometer Handler 289
The Gesture Predictor 291
The Output Handler 294
Detecting Gestures 295
Deploying to Microcontrollers 298
Arduino 298
SparkFun Edge 312
Wrapping Up 327

12. Magic Wand: Training a Model. 329
Training a Model 330
Training in Colab 330
Other Ways to Run the Scripts 339
How the Model Works 339
Visualizing the Input 339
Understanding the Model Architecture 342
Training with Your Own Data 349
Capturing Data 349
Modifying the Training Scripts 352
Training 352
Using the New Model 352
Wrapping Up 353
Learning Machine Learning 353
What's Next 354

13. TensorFlow Lite for Microcontrollers. 355
What Is TensorFlow Lite for Microcontrollers? 355
TensorFlow 355
TensorFlow Lite 356
TensorFlow Lite for Microcontrollers 356
Requirements 357
Why Is the Model Interpreted? 359
Project Generation 360

Build Systems 361
Specializing Code 362
Makefiles 366
Writing Tests 369
Supporting a New Hardware Platform 370
Printing to a Log 371
Implementing DebugLog() 373
Running All the Targets 375
Integrating with the Makefile Build 376
Supporting a New IDE or Build System 376
Integrating Code Changes Between Projects and Repositories 377
Contributing Back to Open Source 379
Supporting New Hardware Accelerators 380
Understanding the File Format 381
FlatBuffers 382
Porting TensorFlow Lite Mobile Ops to Micro 388
Separate the Reference Code 389
Create a Micro Copy of the Operator 389
Port the Test to the Micro Framework 390
Build a Bazel Test 391
Add Your Op to AllOpsResolver 391
Build a Makefile Test 391
Wrapping Up 392

14. Designing Your Own TinyML Applications..... 393
The Design Process 393
Do You Need a Microcontroller, or Would a Larger Device Work? 394
Understanding What's Possible 395
Follow in Someone Else's Footsteps 395
Find Some Similar Models to Train 396
Look at the Data 397
Wizard of Oz-ing 398
Get It Working on the Desktop First 399

15. Optimizing Latency..... 401
First Make Sure It Matters 401
Hardware Changes 402
Model Improvements 402
Estimating Model Latency 403
How to Speed Up Your Model 404
Quantization 404
Product Design 406

Code Optimizations 407
Performance Profiling 407
Optimizing Operations 409
Look for Implementations That Are Already Optimized 409
Write Your Own Optimized Implementation 409
Taking Advantage of Hardware Features 412
Accelerators and Coprocessors 413
Contributing Back to Open Source 414
Wrapping Up 414

16. Optimizing Energy Usage..... 415
Developing Intuition 415
Typical Component Power Usage 416
Hardware Choice 417
Measuring Real Power Usage 419
Estimating Power Usage for a Model 419
Improving Power Usage 420
Duty Cycling 420
Cascading Design 421
Wrapping Up 422

17. Optimizing Model and Binary Size..... 423
Understanding Your System's Limits 423
Estimating Memory Usage 424
Flash Usage 424
RAM Usage 425
Ballpark Figures for Model Accuracy and Size on Different Problems 426
Speech Wake-Word Model 427
Accelerometer Predictive Maintenance Model 427
Person Presence Detection 427
Model Choice 428
Reducing the Size of Your Executable 428
Measuring Code Size 429
How Much Space Is Tensorflow Lite for Microcontrollers Taking? 429
OpResolver 430
Understanding the Size of Individual Functions 431
Framework Constants 434
Truly Tiny Models 434
Wrapping Up 435

18. Debugging..... 437
Accuracy Loss Between Training and Deployment 437

Preprocessing Differences 437
Debugging Preprocessing 439
On-Device Evaluation 440
Numerical Differences 440
Are the Differences a Problem? 440
Establish a Metric 441
Compare Against a Baseline 441
Swap Out Implementations 442
Mysterious Crashes and Hangs 442
Desktop Debugging 443
Log Tracing 443
Shotgun Debugging 444
Memory Corruption 444
Wrapping Up 445

19. Porting Models from TensorFlow to TensorFlow Lite..... 447
Understand What Ops Are Needed 447
Look at Existing Op Coverage in Tensorflow Lite 448
Move Preprocessing and Postprocessing into Application Code 449
Implement Required Ops if Necessary 450
Optimize Ops 450
Wrapping Up 450

20. Privacy, Security, and Deployment..... 453
Privacy 453
The Privacy Design Document 454
Using a PDD 456
Security 456
Protecting Models 457
Deployment 458
Moving from a Development Board to a Product 458
Wrapping Up 459

21. Learning More..... 461
The TinyML Foundation 461
SIG Micro 461
The TensorFlow Website 462
Other Frameworks 462
Twitter 462
Friends of TinyML 462
Wrapping Up 463

A. Using and Generating an Arduino Library Zip..... 465

B. Capturing Audio on Arduino..... 467

Index..... 475