
Spis treści

Wstęp	11
1. Wprowadzenie	15
Urządzenia z systemem wbudowanym	17
Ciągły rozwój	17
2. Informacje wstępne	19
Do kogo skierowana jest ta książka?	19
Jaki sprzęt będzie Ci potrzebny?	20
Jakie oprogramowanie będzie Ci potrzebne?	21
Czego nauczysz się dzięki tej książce?	22
3. Wprowadzenie do uczenia maszynowego	23
Czym właściwie jest uczenie maszynowe?	24
Proces uczenia głębokiego	25
Określenie celu	25
Zebranie zestawu danych	26
Zaprojektowanie architektury modelu	28
Trenowanie modelu	32
Przekształcenie modelu	37
Uruchomienie procesu wnioskowania	37
Ocena i rozwiązanie ewentualnych problemów	38
Podsumowanie	39
4. „Witaj, świecie” TinyML: budowa i trenowanie modelu	40
Co będziemy budować?	41
Nasz zestaw narzędzi do uczenia maszynowego	43
Python i Jupyter Notebooks	43
Google Colaboratory	43
TensorFlow i Keras	44

Budowa naszego modelu	44
Importowanie pakietów	46
Generowanie danych	48
Rozdzielanie danych	50
Definiowanie podstawowego modelu	51
Trenowanie naszego modelu	55
Wskaźniki treningu	57
Wykres historii	58
Ulepszenie naszego modelu	61
Test	65
Konwertowanie modelu na potrzeby TensorFlow Lite	67
Konwertowanie na plik C	71
Podsumowanie	71
5. „Witaj, świecie” TinyML: budowanie aplikacji	72
Omówienie testów	73
Dodawanie zależności	74
Przygotowanie testów	75
Przygotowanie do rejestrowania danych	75
Mapowanie naszego modelu	76
Klasa AllOpsResolver	78
Alokacja pamięci dla modelu	79
Tworzenie interpretera	79
Sprawdzenie tensora wejścia	80
Uruchamianie procesu wnioskowania	82
Odczytywanie danych wyjściowych	84
Uruchamianie testów	86
Budowa pliku z projektem	89
Omówienie kodu źródłowego	90
Początek pliku main_functions.cc	90
Obsługa wyjścia za pomocą output_handler.cc	94
Koniec pliku main_functions.cc	94
Omówienie pliku main.cc	95
Uruchomienie aplikacji	95
Podsumowanie	96
6. „Witaj, świecie” TinyML: uruchomienie aplikacji na mikrokontrolerze	97
Czym właściwie jest mikrokontroler?	97
Arduino	99
Obsługa wyjścia na Arduino	99
Uruchomienie przykładu	102
Wprowadzanie własnych zmian	106

SparkFun Edge	107
Obsługa wyjścia na SparkFun Edge	107
Uruchomienie przykładu	110
Testowanie programu	117
Sprawdzanie danych o przebiegu programu	117
Wprowadzanie własnych zmian	117
Zestaw ST Microelectronics STM32F746G Discovery	118
Obsługa wyjścia na STM32F746G	118
Uruchomienie przykładu	122
Wprowadzanie własnych zmian	124
Podsumowanie	124
7. Wykrywanie słowa wybudzającego: budowanie aplikacji	125
Co będziemy tworzyć?	126
Architektura aplikacji	127
Wprowadzenie do naszego modelu	127
Wszystkie elementy aplikacji	129
Omówienie testów	131
Podstawowy przepływ danych	131
Element dostarczający dane audio	135
Element dostarczający cechy	136
Element rozpoznający polecenia	142
Element reagujący na polecenia	147
Nasłuchiwanie słów wybudzających	148
Uruchomienie naszej aplikacji	151
Uruchomienie aplikacji na mikrokontrolerach	152
Arduino	152
SparkFun Edge	158
Zestaw ST Microelectronics STM32F746G Discovery	168
Podsumowanie	172
8. Wykrywanie słowa wybudzającego: trenowanie modelu	173
Trenowanie naszego nowego modelu	174
Trenowanie w Colab	174
Wykorzystanie modelu w naszym projekcie	187
Zastępowanie modelu	187
Zmiana etykiet	188
Zmiany w kodzie <code>command_responder.cc</code>	188
Inne sposoby uruchamiania skryptów	190
Zasada działania modelu	192
Wizualizacja danych wejściowych	192
Zasada działania generowania cech	195

Architektura modelu	197
Dane wyjściowe modelu	200
Trenowanie modelu z własnymi danymi	201
Zestaw danych Speech Commands	201
Trenowanie modelu na własnych danych	202
Nagrywanie własnych dźwięków	203
Powiększenie zestawu danych	205
Architektury modeli	205
Podsumowanie	206
9. Wykrywanie osoby: budowanie aplikacji	207
Co będziemy budować?	208
Architektura aplikacji	209
Wprowadzenie do naszego modelu	210
Wszystkie elementy aplikacji	211
Omówienie testów	212
Podstawowy przepływ danych	213
Element dostarczający obrazy	216
Element reagujący na wykrycie człowieka	217
Wykrywanie ludzi	218
Uruchomienie aplikacji na mikrokontrolerach	221
Arduino	221
SparkFun Edge	229
Podsumowanie	239
10. Wykrywanie osoby: trenowanie modelu	240
Wybór maszyny	240
Konfiguracja instancji Google Cloud Platform	240
Wybór platformy programistycznej do treningu	248
Tworzenie zestawu danych	248
Trenowanie modelu	249
TensorBoard	251
Ocena modelu	253
Eksportowanie modelu do TensorFlow Lite	253
Eksportowanie do pliku GraphDef Protobuf	253
Zamrażanie wag	254
Kwantyzacja i konwertowanie na potrzeby TensorFlow Lite	254
Konwertowanie na plik źródłowy C	255
Trenowanie dla innych kategorii	255
Architektura MobileNet	256
Podsumowanie	256

11. Magiczna różdżka: budowanie aplikacji	257
Co będziemy tworzyć?	260
Architektura aplikacji	261
Wprowadzenie do naszego modelu	261
Wszystkie elementy aplikacji	262
Omówienie testów	263
Podstawowy przepływ danych	263
Element obsługujący akcelerometr	267
Element przewidujący gesty	268
Element reagujący na wykrycie gestu	271
Wykrywanie gestu	271
Uruchomienie aplikacji na mikrokontrolerach	274
Arduino	275
SparkFun Edge	286
Podsumowanie	300
12. Magiczna różdżka: trenowanie modelu	301
Trenowanie modelu	302
Trening w Colab	302
Inne sposoby uruchamiania skryptów	310
Zasada działania modelu	310
Wizualizacja danych wejściowych	310
Architektura modelu	313
Trenowanie modelu z własnymi danymi	318
Przechwytywanie danych	319
Modyfikacja skryptów trenujących	321
Trening	322
Wykorzystanie nowego modelu	322
Podsumowanie	322
Uczenie się uczenia maszynowego	323
Co dalej?	323
13. TensorFlow Lite dla mikrokontrolerów	324
Czym jest TensorFlow Lite dla mikrokontrolerów?	324
TensorFlow	324
TensorFlow Lite	325
TensorFlow Lite dla mikrokontrolerów	325
Wymagania	326
Dlaczego model potrzebuje interpretera?	328
Generowanie projektu	329
Kompilatory	330
Wyspecjalizowany kod	331

Pliki Makefile	335
Pisanie testów	337
Obsługa nowej platformy sprzętowej	338
Wyświetlanie rejestru zdarzeń	339
Wdrożenie funkcji DebugLog()	341
Uruchamianie wszystkich plików źródłowych	343
Integracja z plikami Makefile	343
Obsługa nowego IDE lub kompilatora	344
Integrowanie zmian w kodzie projektu z repozytoriami	345
Wnoszenie swojego wkładu do kodu z otwartym źródłem	346
Obsługa nowego akceleratora sprzętowego	347
Format pliku	348
Biblioteka FlatBuffers	349
Przenoszenie operacji TensorFlow Lite Mobile na wersję dla mikrokontrolerów	355
Oddzielanie kodu odniesienia	356
Utworzenie kopii operatora dla mikrokontrolera	356
Tworzenie wersji testów dla mikrokontrolerów	357
Tworzenie testu Bazel	358
Dodanie swojego operatora do obiektu AllOpsResolver	358
Kompilacja testu pliku Makefile	358
Podsumowanie	359
14. Projektowanie własnych aplikacji TinyML	360
Projektowanie	360
Czy potrzebny jest mikrokontroler, czy może być większe urządzenie?	361
Co jest możliwe?	362
Podążanie czyimiś śladami	362
Podobne modele do trenowania	363
Sprawdzenie danych	364
Magia „Czarnoksiężnika z krainy Oz”	365
Poprawnie działająca wersja na komputerze jako pierwszy etap	366
15. Optymalizacja prędkości działania programu	367
Prędkość modelu a prędkość ogólna aplikacji	367
Zmiany sprzętu	368
Ulepszenia modelu	368
Ocena opóźnienia modelu	369
Przyspieszanie modelu	370
Kwantyzacja	370
Etap projektowania produktu	372
Optymalizacje kodu	372
Profilowanie wydajności	373

Optymalizowanie operacji	374
Implementacje już zoptymalizowane	375
Tworzenie własnej zoptymalizowanej implementacji	375
Wykorzystanie funkcjonalności sprzętu	378
Akceleratory i koprocesory	378
Wnoszenie swojego wkładu do kodu z otwartym źródłem	379
Podsumowanie	380
16. Optymalizacja poboru mocy	381
Rozwijanie intuicji	381
Pobór mocy standardowych elementów	382
Wybór sprzętu	383
Pomiar rzeczywistego poboru mocy	384
Oszacowanie poboru mocy modelu	385
Ulepszenia związane z zużyciem energii	385
Cykl pracy	386
Projektowanie kaskadowe	386
Podsumowanie	387
17. Optymalizacja modelu i rozmiaru pliku binarnego	388
Zrozumienie ograniczeń własnego systemu	388
Oszacowanie zużycia pamięci	389
Zużycie pamięci flash	389
Zużycie pamięci RAM	390
Szacunkowe wartości dokładności i rozmiaru modelu przy różnych problemach	391
Model rozpoznający słowa wybudzające	392
Model predykcyjnego utrzymania	392
Wykrywanie obecności człowieka	392
Wybór modelu	393
Zmniejszenie rozmiaru pliku wykonywalnego	393
Mierzenie rozmiaru kodu	393
Ile miejsca zajmuje TensorFlow Lite dla mikrokontrolerów?	394
OpResolver	394
Rozmiar pojedynczych funkcji	396
Stałe w platformie TensorFlow Lite	398
Naprawdę małe modele	399
Podsumowanie	399
18. Debugowanie	400
Różnica w dokładności między treningiem a wdrożeniem	400
Różnice we wstępnym przetwarzaniu danych	400
Debugowanie wstępnego przetwarzania danych	401
Ocena działania programu na urządzeniu docelowym	402

Różnice liczbowe	403
Czy różnice stanowią problem?	403
Ustalenie wskaźnika	403
Punkt odniesienia	404
Zamiana implementacji	404
Tajemnicze awarie	405
Debugowanie na pulpicie	405
Sprawdzanie rejestru	405
Debugowanie metodą strzelby	406
Błędy związane z pamięcią	407
Podsumowanie	408
19. Przenoszenie modelu z TensorFlow do TensorFlow Lite	409
Określenie wymaganych operacji	409
Operacje obsługiwane w TensorFlow Lite	410
Przeniesienie wstępnego i końcowego przetwarzania do kodu aplikacji	410
Implementacja niezbędnych operacji	412
Optymalizacja operacji	412
Podsumowanie	412
20. Prywatność, bezpieczeństwo i wdrażanie	413
Prywatność	413
PDD	413
Używanie PDD	415
Bezpieczeństwo	416
Ochrona modeli	416
Wdrożenie	417
Przejsięcie od płytki do produktu	418
Podsumowanie	418
21. Poszerzanie wiedzy	419
Fundacja TinyML	419
SIG Micro	419
Strona internetowa TensorFlow	420
Inne platformy programistyczne	420
Twitter	420
Przyjaciele TinyML	420
Podsumowanie	421
A Używanie i tworzenie biblioteki Arduino w formacie ZIP	423
B Przechwytywanie dźwięku na Arduino	425