

Spis treści

Słowo wstępne	13
Przedmowa	15
Część I. Tworzenie modeli	
1. Wprowadzenie do biblioteki TensorFlow	21
Czym jest uczenie maszynowe?	21
Ograniczenia programowania tradycyjnego	23
Od programowania do uczenia	25
Czym jest TensorFlow?	27
Użycie platformy TensorFlow	28
Instalowanie platformy TensorFlow za pomocą języka Python	29
Użycie platformy TensorFlow w środowisku PyCharm	30
Użycie platformy TensorFlow w środowisku Google Colab	32
Pierwsze kroki z uczeniem maszynowym	34
Czego nauczyła się sieć?	39
Podsumowanie	39
2. Wprowadzenie do widzenia komputerowego	41
Rozpoznawanie elementów odzieży	41
Dane: Fashion MNIST	42
Neurony widzenia komputerowego	43
Projektowanie sieci neuronowej	45
Cały kod programu	46
Trenowanie sieci neuronowej	48
Analiza wyników modelu	49
Trenowanie przez dłuższy czas — nadmierne dopasowanie	49
Zakończenie trenowania	50
Podsumowanie	51

3. Bardziej zaawansowane zagadnienie: wykrywanie cech w obrazach	53
Konwolucje	53
Pooling	55
Implementacja konwolucyjnych sieci neuronowych	57
Analiza sieci konwolucyjnej	59
Tworzenie konwolucyjnej sieci neuronowej rozróżniającej konie i ludzi	61
Zbiór danych Horses or Humans	61
Klasa ImageDataGenerator z pakietu Keras	62
Architektura konwolucyjnej sieci neuronowej przetwarzającej zbiór Horses or Humans	64
Tworzenie zbioru walidacyjnego	66
Testowanie obrazów ze zbioru Horse or Human	67
Generowanie dodatkowych obrazów	70
Uczenie transferowe	73
Klasyfikowanie wieloklasowe	77
Regularyzacja dropout	80
Podsumowanie	83
 4. Korzystanie za pomocą biblioteki TensorFlow Datasets z publicznie dostępnych zbiorów danych	 85
Pierwsze kroki z TFDS	86
Użycie biblioteki TFDS z modelami Keras	88
Wczytywanie określonych wersji	90
Użycie funkcji mapowania do generowania sztucznych danych	91
Użycie biblioteki TensorFlow Addons	91
Korzystanie z niestandardowych podzbiorów	92
Czym jest TFRecord?	93
Użycie procesu ETL do zarządzania danymi w TensorFlow	96
Optymalizacja fazy wczytywania	97
Zrównoleglenie procesu ETL w celu poprawy wydajności trenowania	98
Podsumowanie	100
 5. Wprowadzenie do przetwarzania języka naturalnego	 101
Zamiana języka na liczby	101
Pierwsze kroki z tokenizacją	102
Zamiana zdań na sekwencje	103
Usuwanie słów nieinformatywnych i porządkowanie tekstu	106
Obsługa realnych źródeł danych	108
Pobieranie tekstu za pomocą biblioteki TensorFlow Datasets	108
Pobieranie tekstów z plików CSV	111
Pobieranie tekstów z plików JSON	113
Podsumowanie	116

6. Programowa analiza emocji za pomocą osadzeń	117
Ustalanie znaczenia słów	117
Prosty przykład: względne znaczenie słów	117
Przykład bardziej zaawansowany: użycie wektorów	118
Osadzenia w bibliotece TensorFlow	119
Tworzenie detektora sarkazmu przy użyciu osadzania	120
Zmniejszanie nadmiernego dopasowania w modelach językowych	122
Stosowanie modelu do klasyfikowania zdań	132
Wizualizacja osadzeń	133
Korzystanie ze wstępnie wytrenowanych osadzeń przy użyciu TensorFlow Hub	135
Podsumowanie	137
7. Użycie rekurencyjnych sieci neuronowych do przetwarzania języka naturalnego	139
Podstawy rekurencji	139
Zastosowanie rekurencji w przetwarzaniu języka naturalnego	142
Tworzenie klasyfikatora tekstu przy użyciu rekurencyjnych sieci neuronowych	144
Łączenie warstw LSTM	146
Użycie wstępnie wytrenowanych osadzeń w rekurencyjnych sieciach neuronowych	150
Podsumowanie	156
8. Użycie biblioteki TensorFlow do generowania tekstu	157
Zamiana sekwencji na sekwencje wejściowe	158
Tworzenie modelu	161
Generowanie tekstu	163
Prognozowanie następnego słowa	163
Łączenie prognoz w celu generowania tekstu	164
Poszerzenie zbioru danych	165
Zmiana architektury modelu	166
Ulepszenie danych	167
Kodowanie oparte na znakach	170
Podsumowanie	171
9. Sekwencje i dane szeregów czasowych	173
Wspólne atrybuty szeregów czasowych	174
Tendencja	174
Sezonowość	174
Autokorelacja	175
Szum	176
Metody prognozowania szeregów czasowych	176
Prosta metoda prognozowania jako punkt odniesienia	176
Pomiar dokładności prognozy	178

Metoda bardziej zaawansowana: wykorzystanie średniej ruchomej	179
Ulepszenie metody wykorzystującej średnią ruchomą	180
Podsumowanie	181
10. Tworzenie modeli uczenia maszynowego do prognozowania sekwencji	183
Tworzenie okna zbioru danych	184
Tworzenie okna zbioru danych szeregu czasowego	186
Tworzenie sieci DNN i jej trenowanie w celu dopasowania do danych sekwencji	187
Ocena wyników działania sieci DNN	189
Analiza ogólnej prognozy	190
Dostrajanie współczynnika uczenia	191
Dostrajanie hiperparametrów za pomocą narzędzia Keras Tuner	193
Podsumowanie	196
11. Użycie metod konwolucyjnych i rekurencyjnych w modelowaniu sekwencji	197
Użycie konwolucji z danymi sekwencyjnymi	197
Kodowanie konwolucji	198
Eksperymentowanie z hiperparametrami warstwy Conv1D	201
Korzystanie z danych pogodowych NASA	203
Odczytywanie danych GISS w Pythonie	204
Używanie sieci RNN do modelowania sekwencji	205
Korzystanie z większego zbioru danych	207
Użycie innych metod rekurencyjnych	210
Użycie dropoutu	210
Użycie dwukierunkowych sieci RNN	213
Podsumowanie	214
Część II. Używanie modeli	
12. Wprowadzenie do TensorFlow Lite	217
Czym jest TensorFlow Lite?	217
Przykład: utworzenie modelu i przekonwertowanie go do formatu TensorFlow Lite	219
Krok 1. — zapisanie modelu	220
Krok 2. — konwersja i zapamiętanie modelu	220
Krok 3. — wczytanie modelu TFLite i alokacja tensorów	221
Krok 4. — przeprowadzenie prognozy	222
Przykład: wykorzystanie uczenia transferowego	
w klasyfikatorze obrazów i jego konwersja na format TensorFlow Lite	223
Krok 1. — utworzenie i zapisanie modelu	223
Krok 2. — konwersja modelu do formatu TensorFlow Lite	225
Krok 3. — optymalizacja modelu	226
Podsumowanie	228

13. Użycie TensorFlow Lite w systemie Android	229
Czym jest Android Studio?	229
Tworzenie pierwszej aplikacji opartej na TensorFlow Lite dla systemu Android	230
Krok 1. — utworzenie nowego projektu	230
Krok 2. — edycja pliku układu	231
Krok 3. — dodanie zależności TensorFlow Lite	234
Krok 4. — dodanie modelu TensorFlow Lite	235
Krok 5. — utworzenie kodu umożliwiającego użycie modelu TensorFlow Lite do wnioskowania	236
Coś więcej niż „Witaj, świecie!” — przetwarzanie obrazów	239
Przykładowe aplikacje wykorzystujące bibliotekę TensorFlow Lite	242
Podsumowanie	243
14. Użycie TensorFlow Lite w systemie iOS	245
Tworzenie pierwszej aplikacji TensorFlow Lite za pomocą Xcode	245
Krok 1. — utworzenie prostej aplikacji iOS	245
Krok 2. — dodanie bibliotek TensorFlow Lite do projektu	246
Krok 3. — utworzenie interfejsu użytkownika	247
Krok 4. — dodanie i zainicjalizowanie klasy odpowiedzialnej za operację prognozowania	251
Krok 5. — przeprowadzenie operacji prognozowania	253
Krok 6. — dodanie modelu do aplikacji	255
Krok 7. — dodanie logiki obsługującej interfejs użytkownika	255
Coś więcej niż „Witaj, świecie!” — przetwarzanie obrazów	258
Przykładowe aplikacje wykorzystujące bibliotekę TensorFlow Lite	260
Podsumowanie	261
15. Wprowadzenie do TensorFlow.js	263
Czym jest TensorFlow.js?	263
Instalowanie i używanie środowiska programistycznego Brackets	265
Tworzenie pierwszego modelu wykorzystującego bibliotekę TensorFlow.js	266
Tworzenie klasyfikatora irysów	269
Podsumowanie	273
16. Rozwiązywanie problemów z zakresu widzenia komputerowego za pomocą biblioteki TensorFlow.js	275
Uwagi dla programistów używających biblioteki TensorFlow dotyczące języka JavaScript	276
Tworzenie konwolucyjnej sieci neuronowej za pomocą języka JavaScript	277
Stosowanie wywołań zwrotnych do wizualizacji	279
Trenowanie za pomocą zbioru MNIST	281
Przeprowadzanie wnioskowania dla obrazów przy użyciu biblioteki TensorFlow.js	285
Podsumowanie	286

17. Konwersja modeli z Pythona do JavaScriptu i ponowne ich użycie	287
Konwersja modeli z Pythona do JavaScriptu	287
Użycie przekonwertowanych modeli	289
Użycie wcześniej przekonwertowanych modeli	291
Podsumowanie	299
18. Wykorzystanie uczenia transferowego w języku JavaScript	301
Uczenie transferowe przy użyciu biblioteki MobileNet	301
Krok 1. — pobranie modelu MobileNet i identyfikacja warstw do użycia	302
Krok 2. — utworzenie własnej architektury modelu, w której danymi wejściowymi są dane wyjściowe MobileNet	303
Krok 3. — uzyskanie i sformatowanie danych	306
Krok 4. — przeprowadzenie trenowania modelu	311
Krok 5. — przeprowadzenie wnioskowania za pomocą modelu	312
Uczenie transferowe przy użyciu repozytorium TensorFlow Hub	313
Użycie modeli z portalu TensorFlow.org	316
Podsumowanie	318
19. Wdrażanie modeli za pomocą usługi TensorFlow Serving	321
Czym jest TensorFlow Serving?	321
Instalowanie systemu TensorFlow Serving	324
Instalacja przy użyciu Dockera	324
Bezpośrednia instalacja w systemie Linux	325
Tworzenie i udostępnianie modelu	326
Konfigurowanie serwera	329
Podsumowanie	331
20. Sztuczna inteligencja a etyka, uczciwość i prywatność	333
Uczciwość w procesie programowania	334
Uczciwość w procesie uczenia maszynowego	337
Narzędzia związane z kwestiami uczciwości	338
What-If	338
Facets	340
Uczenie federacyjne	341
Krok 1. — identyfikacja dostępnych urządzeń, które można wykorzystać do trenowania	342
Krok 2. — identyfikacja odpowiednich urządzeń, które można wykorzystać do trenowania	342
Krok 3. — zainstalowanie modelu, który będzie używać zbioru treningowego	343

Krok 4. — zwrócenie wyników trenowania do serwera	344
Krok 5. — zainstalowanie modelu głównego w urządzeniach	344
Bezpieczna agregacja w uczeniu federacyjnym	345
Uczenie federacyjne przy użyciu TensorFlow Federated	346
Zasady firmy Google dotyczące sztucznej inteligencji	347
Podsumowanie	348

Szanowny Czytelniku,

Sztuczna inteligencja ma szansę przekształcić każdą branżę przemysłową. Jednak prawie każda aplikacja musi pasować do określonego zastosowania. System analizujący dotychczasową decyzję różni się od systemu wyznakającego ciśnień w fabryce, który z kolei jest czymś innym niż system rekomendacji produktów. Ale sztuczna inteligencja mogła wykazać pełną przydatność, mądrze i w sposób bezpieczny, które pomogą dostosować jej niesamowite możliwości do milionów konkretnych problemów czekających na rozwiązanie.

Gdy kłowałem zespołem Google Brain, zaczęliśmy tworzyć prekursor biblioteki TensorFlow o nazwie DistBelief, oparty na języku C++. Byliśmy przekonani, że wykorzystanie tysięcy procesorów do trenowania sieci neuronowej (na przykład użyliśmy 16 tysięcy procesorów do trenowania detektora kotów w niezarejestrowanych filmach na YouTube). Od tamtego czasu wiele się zmieniło! Do zadań, które kiedyś były niewykonalne, można obecnie za około 12 tysięcy złotych wykonać, odpowiadając zasoby w chmurze Google za pomocą tensorowych jednostek przetwarzania i procesorów graficznych rutynowo trenuje sieci neuronowe na skalę, która jeszcze kilka lat temu była niewyobrażalna.

Również rozwój biblioteki TensorFlow przeszedł długą drogę. Stała się dużo bardziej użyteczna i obecnie udostępnia wiele opcji, od modelowania, poprzez wykorzystywanie istniejących wytrenowanych modeli, po implementowanie aplikacji na urządzeniach brzożowych o niskiej mocy obliczeniowej. Dzięki niej setki tysięcy programistów tworzą modele uczenia głębokiego.

Laurance Moroney, największy apologeta sztucznej inteligencji w firmie Google, odegrał główną rolę w przekształceniu biblioteki TensorFlow w jedną z czołowych platform służących do programowania uczenia maszynowego. Miałem zaszczyt wspierać go za pomocą portalu deeplearning.ai i internetowych kursów Coursera. Dotarły one do ponad 80 tysięcy studentów i otrzymały liczne pochlebne recenzje.