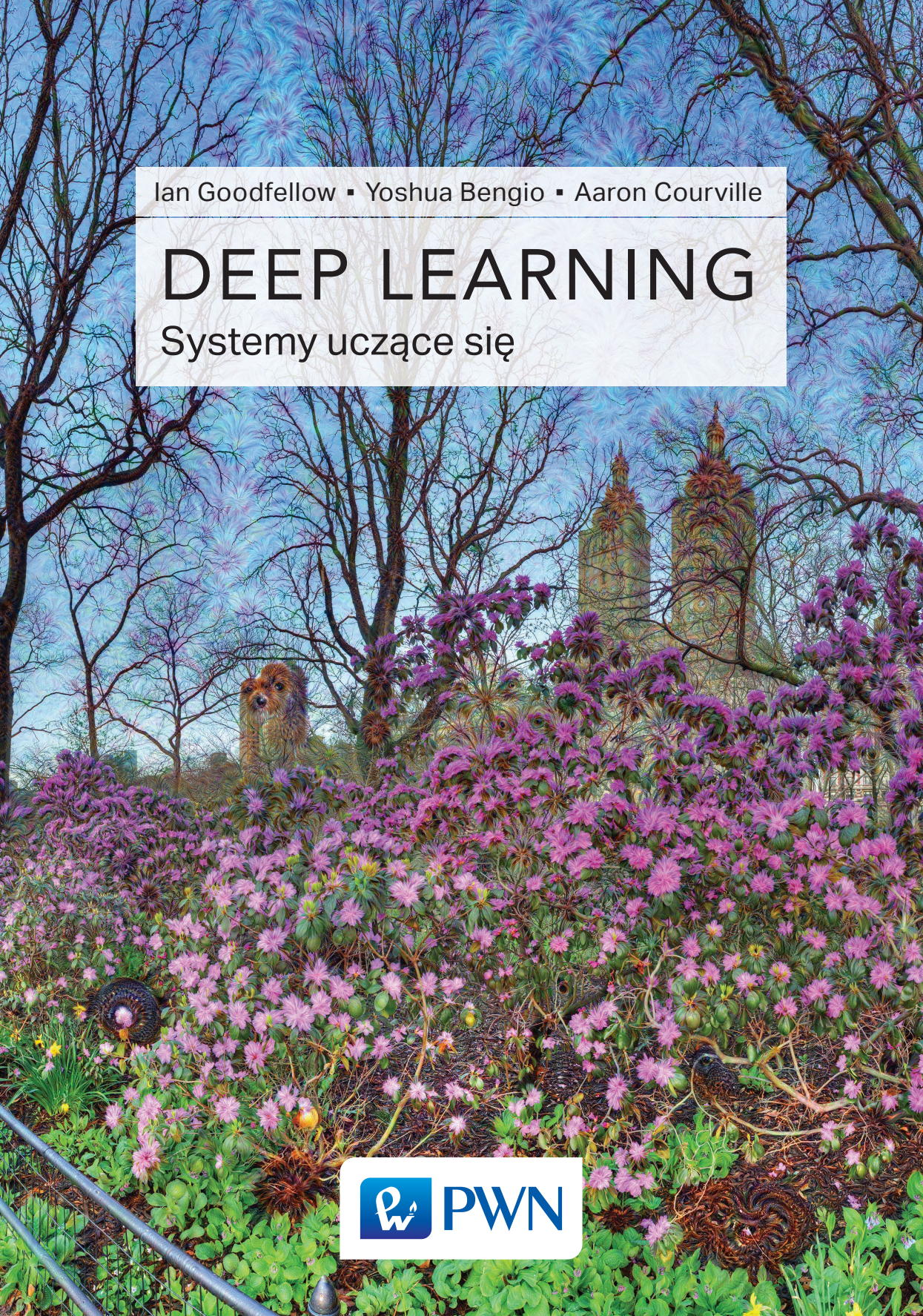

Ian Goodfellow ▪ Yoshua Bengio ▪ Aaron Courville
DEEP LEARNING
Systemy uczące się
PWN

DEEP LEARNING

Ian Goodfellow ▪ Yoshua Bengio ▪ Aaron Courville

DEEP LEARNING

Systemy uczące się

PWN

Przekład **WITKOM Witold Sikorski**

Projekt okładki polskiego wydania **Hubert Zacharski**

Ilustracja na okładce: Central Park Azalea Walk Dreamscape, autor – Daniel Ambrosi (danielambrosi.com) Dreamscapes Daniela Ambrosiego są tworzone dzięki zastosowaniu otwartego oprogramowania DeepDream Google'a, zmodyfikowanego przez Josepha Smarra (Google) i Chrisa Lamba (NVIDIA) tak, aby z powodzeniem działało na panoramicznych obrazach złożonych z wielu setek pikseli tworzonych przez Ambrosiego.

Wydawca **Łukasz Łopuszański**

Redaktor prowadzący **Adam Kowalski**

Redaktor **Joanna Forysiak**

Koordynator produkcji **Anna Bączkowska**

Skład i łamanie **Fixpoint**

ISBN: 978-83-01-19583-0

Wydanie I
Warszawa 2018

Wydawnictwo Naukowe PWN SA
tel. 22 69 54 321; faks 22 69 54 288
infolinia 801 33 33 88
e-mail: pwn@pwn.com.pl, reklama@pwn.pl
www.pwn.pl

Druk i oprawa: OSDW Azymut Sp. z o.o.

Spis treści

1

Wprowadzenie

Wynalazcy od dawna marzyli o myślących maszynach. Te marzenia sięgają przynajmniej do czasów starożytnej Grecji. Mityczne postaci Pigmaliona, Dedala czy Hefajstosa można interpretować jako legendarnych wynalazców, a Galateę, Talosa i Pandorę rozpatrywać jako sztuczne życie (Ovid and Martin 2004, Sparkes 1996, Tandy 1997).

Już wtedy, gdy powstały pierwsze programowane komputery, ludzie zaczęli się zastanawiać, czy mogą one stać się inteligentne (Lovelace 1842) – ponad 100 lat wcześniej zanim zbudowano pierwsze inteligentne maszyny. Dziś **sztuczna inteligencja** (AI) to ekscytująca dziedzina z wieloma zastosowaniami praktycznymi i programami badawczymi. Wprowadza się inteligentne oprogramowanie, by zautomatyzować rutynowe prace, zrozumieć mowę lub obrazy, wykonywać diagnostykę medyczną i wspomagać podstawowe badania naukowe.

We wczesnych latach sztucznej inteligencji szybko rozwiązywane były problemy trudne do wykonania dla ludzi, ale dość proste dla komputerów – problemy, które można opisać jako listy sformalizowanych matematycznych reguł. Prawdziwym wyzwaniem dla sztucznej inteligencji okazało się rozwiązywanie zadań łatwych do wykonania dla ludzi, ale trudnych w opisie formalnym – problemów, które rozwiązujemy intuicyjnie, niejako automatycznie, jak rozpoznawanie słów lub twarzy na obrazach.

Książka ta poświęcona jest rozwiązywaniu właśnie tych intuicyjnych przeszkód. By je pokonać, trzeba pozwolić komputerom, aby uczyły się na swoich doświadczeniach i rozumiały świat w kontekście hierarchii pojęć, z których każde jest zdefiniowane przez jego związek z pojęciem prostszym. Dzięki gromadzeniu wiedzy na podstawie doświadczeń unika się potrzeby formalnego opisania przez ludzi całej wiedzy potrzebnej komputerowi. Hierarchia pojęć

pozwala mu na uczenie się skomplikowanych pojęć przez budowanie ich z pojęć prostszych. Jeśli narysujemy schemat, w którym pojęcia te nakładają się na siebie, to powstanie wiele warstw. Dlatego to podejście do AI nazywamy *głębokim uczeniem się*[*].

Wiele początkowych sukcesów na polu AI miało miejsce we względnie sterylnych i sformalizowanych środowiskach i nie wymagało od komputerów specjalnej wiedzy o świecie. Na przykład system gry w szachy komputera Deep Blue firmy IBM pokonał w 1997 roku mistrza świata Garriego Kasparowa (Hsu 2002). Szachy to prosty świat, obejmujący tylko 64 pola i 32 elementy, które mogą poruszać się według ściśle określonych reguł. Opracowanie prowadzącej do sukcesu strategii gry w szachy to niezwykłe osiągnięcie, ale nie wynika ono z trudności opisania komputerowi figur szachowych i dopuszczalnych ruchów. Szachy można dokładnie opisać za pomocą krótkiej listy całkiem sformalizowanych reguł podanych przez programistę.

Jak na ironię, abstrakcyjne i sformalizowane zadania, należące do najtrudniejszych dla człowieka, są najłatwiejszymi dla komputera. Komputery od dawna mogą pokonać najlepszego szachistę, ale dopiero ostatnio zaczęły nabywać niektóre zwykłe ludzkie umiejętności, jak rozpoznawanie obiektów czy mowy. Życie codzienne wymaga sporej wiedzy o świecie. Jej duża część jest subiektywna i intuicyjna, dlatego trudno ją formalnie opisać. Aby działać inteligentnie, komputery muszą zdobyć tę samą wiedzę. Jednym z kluczowych wyzwań w dziedzinie sztucznej inteligencji jest zatem przekazanie tej nieformalnej wiedzy komputerowi.

W kilku projektach z dziedziny sztucznej inteligencji poszukiwano twardej wiedzy o świecie podanej językiem sformalizowanym – komputer może automatycznie tworzyć argumenty na podstawie tych instrukcji za pomocą logicznych reguł wnioskowania. Nazywa się to podejściem do sztucznej inteligencji **opartym na wiedzy**. Żaden z tych projektów nie zakończył się sukcesem. Najbardziej znany jest Cyc (Lenat and Guha 1989). Cyc to mechanizm wnioskowania i baza danych instrukcji w języku o nazwie CycL. Instrukcje te są wprowadzane przez personel złożony z ludzi nadzorujących projekt. Jest to uciążliwy proces. Ludzie męczą się, tworząc sformalizowane reguły o takiej złożoności, aby opisywały świat. Cyc nie potrafił zrozumieć opowieści o człowieku imieniem Fred, który rano się golił (Linde 1992). Jego mechanizm rozumowania wykrył niespójność w tej opowieści: wiedział,

[*]Termin *machine learning* jest tłumaczony na język polski na kilka sposobów. W tłumaczeniu przyjęto określenie *systemy uczące się*, aby podkreślić jednoznacznie, że systemy same się uczą, a nie uczą kogoś, co sugeruje sformułowanie *uczenie maszynowe*. Podobnie więc mamy *głębokie uczenie się*. W tekście zachowano jednak określenie *deep learning*, jako nowe pojęcie, którego przekład nie jest w Polsce utrwalony (wszystkie przypisy oznaczone * pochodzą od tłumacza).

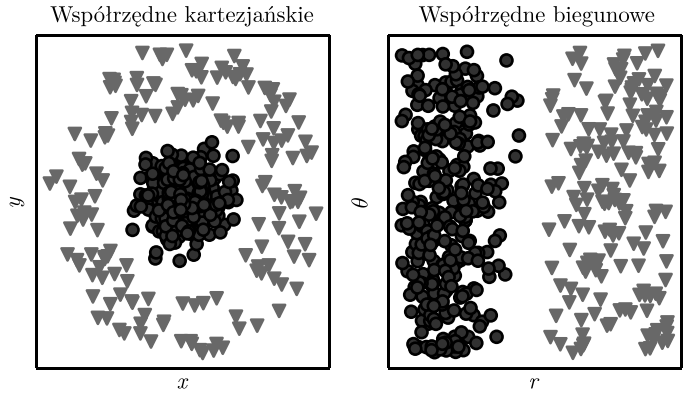
że ludzie nie mają części elektrycznych, ale Fred miał w ręku elektryczną golarkę, więc uwierzył, że jednostka „FredPodczasGolenia" zawierała elementy elektryczne. Stąd pojawiło się pytanie, czy podczas golenia Fred nadal jest osobą.

Trudności systemu opierającego się na twardo zapisanej wiedzy sugerują, że systemy AI potrzebują zdolności do przyswajania własnej wiedzy wyodrębnianej jako wzorce z surowych danych. Takie zdolności określa się mianem **systemów uczących się**. Ich wprowadzenie pozwoliło komputerom na zajęcie się problemami obejmującymi wiedzę o świecie i podejmowanie decyzji, które wydają się subiektywne. Prosty algorytm samouczenia się zwany regresją logistyczną pozwala określić, czy zalecać cesarskie cięcie (Mor-Yosef et al. 1990). Inny prosty algorytm samouczenia zwany „naiwny Baynes" potrafi oddzielić ważne e-maile od spamu.

Działanie tych algorytmów mocno zależy od reprezentatywności otrzymanych danych. Na przykład, gdy regresja logistyczna jest używana do wskazania cesarskiego cięcia, system AI nie bada pacjenta. To lekarz podaje systemowi kilka odpowiednich informacji, jak obecność blizn macicznych lub ich brak. Każdy fragment informacji ujęty w opisie pacjentki jest określany jako cecha. Regresja logistyczna uczy się, jaki związek ma każda z cech pacjentki z rozmaitymi rezultatami. Nie może jednak w żaden sposób wpływać na sposób definiowania tych cech. Jeśli regresja logistyczna otrzyma skan rezonansu magnetycznego (MRI) pacjentki zamiast formalnego raportu lekarza, nie potrafi dać użytecznych wskazań. Pojedyncze piksele skanowania MRI mają pomijalną korelację z jakimikolwiek komplikacjami, jakie mogą pojawić się przy porodzie.

Ta zależność od sposobu reprezentacji jest ogólnym zjawiskiem w całej dziedzinie informatyki, a także w życiu codziennym. W naukach komputerowych działania takie, jak przeszukiwanie zbioru danych, są w wykładniczo szybsze, jeśli zbiór ma strukturę i jest inteligentnie indeksowany. Ludzie z łatwością wykonują działania arytmetyczne na cyfrach arabskich, ale działania na cyfrach rzymskich zajmują im znacznie więcej czasu. Nie jest więc zaskoczeniem, że wybrany sposób reprezentacji ma ogromny wpływ na wydajność algorytmów samouczących się. Prosty obrazowy przykład pokazano na rysunku 1.1.

Wiele zadań z zakresu sztucznej inteligencji można rozwiązać, projektując odpowiedni dla danego zadania zestaw cech, a następnie dostarczając te cechy prostemu algorytmowi uczącemu się. Na przykład użyteczną cechą identyfikacji mówcy na podstawie dźwięku jest ocena rozmiaru jego toru głosowego. Cecha ta daje nam możne przesłanki do określenia, czy mówiący jest kobietą, mężczyzną czy dzieckiem. Jednak dla wielu zadań trudno określić



Rysunek 1.1. Przykłady różnych reprezentacji. Przypuśćmy, że chcemy oddzielić dwie kategorie, rysując linię między nimi jako wykres punktowy. Na wykresie po lewej mamy dane we współrzędnych kartezjańskich, a zadanie jest niemożliwe do wykonania. Na rysunku po prawej dane są we współrzędnych biegunowych i zadanie staje się proste do rozwiązania za pomocą linii pionowej (rysunek utworzono we współpracy z Davidem Warde-Farley'em)

wybrane cechy. Przypuśćmy na przykład, że chcemy napisać program do identyfikacji samochodów na zdjęciach. Wiemy, że samochody mają koła, możemy więc jako cechę określić obecność koła. Niestety, trudno w pikselach opisać dokładnie, jak wygląda koło. Ma ono co prawda prosty kształt geometryczny, ale jego obraz może być skomplikowany z powodu padającego cienia, słońca odbijającego się od metalowych części felgi, błotnika, obiektu na pierwszym planie zasłaniającego część koła i tak dalej.

Jednym z rozwiązań tego problemu będzie zastosowanie systemów uczących się, aby brać pod uwagę nie tylko odwzorowanie reprezentacji na wyniki, lecz także sam sposób reprezentacji. To podejście znane jest jako uczenie się przez reprezentację. Opanowanie reprezentacji często daje dużo lepszą wydajność niż to, co można osiągnąć, wprowadzając reprezentację ręcznie. Umożliwia to także systemom SAI szybką adaptację do nowych zadań z minimalnym udziałem człowieka. Algorytm uczenia się reprezentacji może odnaleźć właściwy zestaw cech dla prostego zadania w ciągu kilku minut, a dla złożonego w ciągu godzin lub miesięcy. Ręczne tworzenie cech dla skomplikowanego zadania wymaga wiele ludzkiej pracy i może zająć całe dekady dużej grupie badaczy.

Typowym przykładem algorytmu uczenia się reprezentatywnego jest autokodowanie. Jest to połączenie funkcji kodowania, która przekształca dane wejściowe na inną postać, z funkcją dekodera, która przekształca nową reprezentację z powrotem do formatu początkowego. Systemy autokodowania