Содержание

Веб-сайт

www.deeplearning.book

Книгу сопровождает указанный выше сайт, где представлены упражнения, слайды, исправления ошибок и другие материалы, полезные читателям и преподавателям.

Благодарности

Эта книга не состоялась бы, если бы не помощь со стороны многих людей.

Мы благодарны тем, кто откликнулся на наше предложение о написании книги и помог спланировать ее содержание и структуру: Гийом Ален (Guillaume Alain), Кюнгхюн Чо (Kyunghyun Cho), Чаглар Гюльчехре (Çaglar Gülçehre), Дэвид Крюгер (David Krueger), Гуго Ларошель (Hugo Larochelle), Разван Паскану (Razvan Pascanu) и Томас Рохе (Thomas Rohée).

Мы также благодарны всем, кто присылал отзывы о самой книге, иногда даже о нескольких главах: Мартен Абади (Martín Abadi), Гийом Ален, Йон Андруцопулос (Ion Androutsopoulos), Фред Берщ (Fred Bertsch), Олекса Биланюк (Olexa Bilaniuk), Уфук Джан Бичиджи (Ufuk Can Biçici), Матко Босняк (Matko Bošnjak), Джон Буерсма (John Boersma), Грег Брокман (Greg Brockman), Александр де Бребиссон (Alexandre de Brébisson), Пьер Люк Каррье (Pierre Luc Carrier), Саратх Чандар (Sarath Chandar), Павел Чилински (Pawel Chilinski), Марк Дауст (Mark Daoust), Олег Дашевский, Лоран Дин (Laurent Dinh), Стефан Дрезейтль (Stephan Dreseitl), Джим Фан (Jim Fan), Мяо Фан (Miao Fan), Мейре Фортунато (Meire Fortunato), Фредерик Франсис (Frédéric Francis), Нандо де Фрейтас (Nando de Freitas), Чаглар Гюльчехре (Çaglar Gülçehre), Юрген Ван Гаэл (Jurgen Van Gael), Хавьер Алонсо Гарсиа (Javier Alonso García), Джонатан Хант (Jonathan Hunt), Гопи Джейярам (Gopi Jeyaram), Чингиз Кабытаев (Chingiz Kabytayev), Лукаш Кайзер (Lukasz Kaiser), Варун Канаде (Varun Kanade), Азифулла Хан (Asifullah Khan), Акиель Хан (Akiel Khan), Джон Кинг (John King), Дидерик П. Кингма (Diederik P. Kingma), Ян Лекун (Yann LeCun), Рудольф Матей (Rudolf Mathey), Матиас Маттамала (Matías Mattamala), Абхинав Мауриа (Abhinav Maurya), Кэвин Мерфи (Kevin Murphy), Олег Мюрк (Oleg Mürk), Роман Новак (Roman Novak), Огастес К. Одена (Augustus Q. Odena), Симон Павлик (Simon Pavlik), Карл Пичотта (Karl Pichotta), Эдди Пирс (Eddie Pierce), Кари Пулли (Kari Pulli), Руссель Рахман (Roussel Rahman), Тапани Райко (Tapani Raiko), Анурга Ранджан (Anurag Ranjan), Йоханнес Ройт (Johannes Roith), Михаэла Роска (Mihaela Rosca), Халис Сак (Halis Sak), Сезар Салгадо (César Salgado), Григорий Сапунов, Ёсинори Сасаки (Yoshinori Sasaki), Майк Шустер (Mike Schuster), Джулиан Сербан (Julian Serban), Нир Шабат (Nir Shabat), Кен Ширрифф (Ken Shirriff), Андрэ Симпело (Andre Simpelo), Дэвид Слейт (David Slate), Скотт Стэнли (Scott Stanley), Давид Суссилло (David Sussillo), Илья Суцкевер (Ilya Sutskever), Карлес Гелада Саец (Carles Gelada Sáez), Грэхэм Тейлор (Graham Taylor), Валентин Толмер (Valentin Tolmer), Массимильяно Томассоли (Massimiliano Tomassoli), Ан Тран (An Tran), Шубхенду Триведи (Shubhendu Trivedi), Алексей Умнов, Винсет Ванхоуке (Vincent Vanhoucke), Марко Висентини-Скарцанелла (Marco Visentini-Scarzanella), Матрин Вита (Martin Vita), Дэвид Уорд-Фарли (David Warde-Farley), Дастин Уэбб (Dustin Webb), Кэлвин Су (Kelvin Xu), Вэй Сюэ (Wei Xue), Ке Янг (Ke Yang), Ли Яо (Li Yao), Зигмунт Заяц (Zygmunt Zajac) и Озан Чаглаян (Ozan Çaglayan).

Мы признательны и тем, кто делился своим мнением об отдельных главах.

- Обозначения: Чжан Юань Хан (Zhang Yuanhang).
- Глава 1 «Введение»: Юсуф Акгуль (Yusuf Akgul), Себастьен Братьерес (Sebastien Bratieres), Самира Эбрахими (Samira Ebrahimi), Чарли Горичаназ (Charlie

Gorichanaz), Брендан Лоудермилк (Brendan Loudermilk), Эрис Моррис (Eric Morris), Космин Пярвулеску (Cosmin Pârvulescu) и Альфредо Солано (Alfredo Solano).

- Глава 2 «Линейная алгебра»: Амджад Алмахаири (Amjad Almahairi), Никола Баниц (Nikola Banic), Кэвин Беннетт (Kevin Bennett), Филипп Кастонге (Philippe Castonguay), Оскар Чанг (Oscar Chang), Эрик Фослер-Люссье (Eric Fosler-Lussier), Андрей Халявин, Сергей Орешков (Sergey Oreshkov), Иштван Петрас (István Petrás), Дэннис Прэнгл (Dennis Prangle), Томас Рохе (Thomas Rohée), Гитанджали Гулве Сехгал (Gitanjali Gulve Sehgal), Колби Толанд (Colby Toland), Алессандро Витале (Alessandro Vitale) и Боб Уэлланд (Bob Welland).
- Глава 3 «Теория вероятностей и теория информации»: Джон Филипп Андерсон (John Philip Anderson), Кай Арулкумаран (Kai Arulkumaran), Венсан Дюмуле (Vincent Dumoulin), Руй Фа (Rui Fa), Стефан Гоус (Stephan Gouws), Артем Оботуров, Антти Расмус (Antti Rasmus), Алексей Сурков и Фолькер Тресп (Volker Tresp).
- Глава 4 «Численные методы»: Тран Лам Аниан Фишер (Tran Lam AnIan Fischer) и Ху Ю Хуан (Hu Yuhuang).
- Глава 5 «Основы машинного обучения»: Дзмитри Бахданау (Dzmitry Bahdanau), Жюстен Доманге (Justin Domingue), Нихил Гарг (Nikhil Garg), Макото Оцука (Makoto Otsuka), Боб Пепин (Bob Pepin), Филип Попьен (Philip Popien), Бхарат Прабхакар (Bharat Prabhakar), Эммануэль Райнер (Emmanuel Rayner), Питер Шепард (Peter Shepard), Ки-Бонг Сонг (Kee-Bong Song), Чжен Сун (Zheng Sun) и Энди Ву (Andy Wu).
- Глава 6 «Глубокие сети прямого распространения»: Уриэль Бердуго (Uriel Berdugo), Фабрицио Боттарель (Fabrizio Bottarel), Элизабет Бэрл (Elizabeth Burl), Ишан Дуругкар (Ishan Durugkar), Джефф Хлыва (Jeff Hlywa), Ёнг Вук Ким (Jong Wook Kim), Давид Крюгер (David Krueger), Адития Кумар Прахарадж (Aditya Kumar Praharaj) и Стэн Сутла (Sten Sootla).
- Глава 7 «Регуляризация в глубоком обучении»: Мортен Колбэк (Morten Kolbæk), Читыдж Лауриа (Kshitij Lauria), Инкю Ли (Inkyu Lee), Сунил Мохан (Sunil Mohan), Хай Пхонг Пхан (Hai Phong Phan) и Джошуа Сэлисбэри (Joshua Salisbury).
- Глава 8 «Оптимизация обучения глубоких моделей»: Марсель Аккерман (Marcel Ackermann), Питер Армитейдж (Peter Armitage), Роуэл Атиенза (Rowel Atienza), Эндрю Брок (Andrew Brock), Теган Махарадж (Tegan Maharaj), Джеймс Мартенс (James Martens), Мостафа Натегх (Mostafa Nategh), Кашиф Расул (Kashif Rasul), Клаус Штробль (Klaus Strobl) и Никола Тэрнер (Nicholas Turner).
- Глава 9 «Сверточные сети»: Мартен Аржовски (Martín Arjovsky), Евгений Бревдо (Eugene Brevdo), Константин Дивилов, Эрик Йенсен (Eric Jensen), Мехди Мирза (Mehdi Mirza), Алекс Пайно (Alex Paino), Марджори Сэйер (Marjorie Sayer), Райан Стаут (Ryan Stout) и Вентао Ву (Wentao Wu).
- Глава 10 «Моделирование последовательностей: рекуррентные и рекурсивные сети»: Гёкчен Ераслан (Gökçen Eraslan), Стивен Хиксон (Steven Hickson), Разван Паскану (Razvan Pascanu), Лоренхо фон Риттер (Lorenzo von Ritter), Руй Родригес (Rui Rodrigues), Дмитрий Сердюк, Донгуй Ши (Dongyu Shi) и Кай Ю Яан (Kaiyu Yang).

- Глава 11 «Практическая методология»: Даниэль Бекштейн (Daniel Beckstein).
- Глава 12 «Приложения»: Джордж Дал (George Dahl), Владимир Некрасов (Vladimir Nekrasov) и Рибана Рошер (Ribana Roscher).
- Глава 13 «Линейные факторные модели»: Джейнт Кушик (Jayanth Koushik).
- Глава 15 «Обучение представлений»: Кунал Гхош (Kunal Ghosh).
- Глава 16 «Структурные вероятностные модели в глубоком обучении»: Минь Ле (Minh Lê) и Антон Варфолом (Anton Varfolom).
- Глава 18 «Преодоление трудностей, связанных со статической суммой»: Сэм Боумен (Sam Bowman).
- Глава 19 «Приближенный вывод»: Ю Цзя Бао (Yujia Bao).
- Глава 20 «Глубокие порождающие модели»: Николас Чападос (Nicolas Chapados), Даниэль Галвес (Daniel Galvez), Вэн Мин Ма (Wenming Ma), Фади Медхат (Fady Medhat), Шакри Мохамед (Shakir Mohamed) и Грегуар Монтавон (Grégoire Montavon).
- Библиография: Лукас Михельбахер (Lukas Michelbacher) и Лесли Н. Смит (Leslie N. Smith).

Мы также благодарим авторов, давших нам разрешение использовать в тексте изображения, рисунки и данные из их публикаций. Источники указываются в подрисуночных подписях.

Мы благодарны Лю Вану (Lu Wang), написавшему программу pdf2htmlEX, которой мы пользовались для подготовки варианта книги для веба, а также за помощь в улучшении качества получившегося HTML-кода.

Спасибо супруге Яна Даниэле Флори Гудфеллоу за терпеливую поддержку Яна на всем протяжении работы над книгой и за помощь в выверке текста.

Спасибо команде Google Brain за создание атмосферы интеллектуального общения, позволившей Яну уделять много времени работе над книгой и получать отзывы и наставления от коллег. Особенно хотим поблагодарить бывшего начальника Яна, Грега Коррадо (Greg Corrado), и его нынешнего начальника, Сами Бенджио (Samy Bengio), за поддержку этого проекта. Наконец, мы благодарны Джеффри Хинтону (Geoffrey Hinton), который подбадривал нас, когда было трудно.

Обозначения

В этом разделе приведен краткий перечень обозначений, используемых в книге. Большая часть соответствующего математического аппарата описана в главах 2–4.

Числа и массивы

a	скаляр (целый или вещественный)
$\boldsymbol{a}$	вектор
$\boldsymbol{A}$	матрица
$\mathbf{A}$	тензор
$\boldsymbol{I}_n$	единичная матрица с n строками и n столбцами
$\boldsymbol{I}$	единичная матрица, размер которой определяется контекстом
$\boldsymbol{e}^{(i)}$	стандартный базисный вектор [0, ..., 0, 1, 0..., 0], содержащий 1 в i-й позиции
diag($\boldsymbol{a}$)	квадратная диагональная матрица, на диагонали которой находятся элементы вектора $\boldsymbol{a}$
a	случайная скалярная величина
$\mathbf{a}$	случайный вектор
$\mathbf{A}$	случайная матрица

Множества и графы

$\mathbb{A}$	множество
$\mathbb{R}$	множество вещественных чисел
$\{0, 1\}$	множество из двух элементов: 0 и 1
$\{0, 1, ..., n\}$	множество целых чисел от 0 до n включительно
$[a, b]$	замкнутый интервал вещественной прямой от a до b, включающий границы
$(a, b]$	интервал вещественной прямой, не включающий a, но включающий b
$\mathbb{A}\setminus\mathbb{B}$	разность множеств, т. е. множество, содержащее все элементы $\mathbb{A}$, не являющиеся элементами $\mathbb{B}$
$\mathcal{G}$	граф
$Pa_{\mathcal{G}}(\mathrm{x}_i)$	родители x_i в $\mathcal{G}$

Индексирование

a_i	i-й элемент вектора $\boldsymbol{a}$, индексирование начинается с 1
a_{-i}	все элементы вектора $\boldsymbol{a}$, кроме i-го
$A_{i,j}$	элемент матрицы $\boldsymbol{A}$ в позиции (i, j)
$\boldsymbol{A}_{i,:}$	i-я строка матрицы $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	i-й столбец матрицы $\boldsymbol{A}$
$\mathbf{A}_{i,j,k}$	элемент трехмерного тензора $\mathbf{A}$ в позиции (i, j, k)
$\mathbf{A}_{:,:,i}$	двумерная срезка трехмерного тензора $\mathbf{A}$
a_i	i-й элемент случайного вектора $\mathbf{a}$

Операции линейной алгебры

$\boldsymbol{A}^\top$	матрица, транспонированная к $\boldsymbol{A}$
$\boldsymbol{A}^+$	псевдообратная матрица Мура-Пенроуза

$\boldsymbol{A} \odot \boldsymbol{B}$	поэлементное произведение матриц $\boldsymbol{A}$ и $\boldsymbol{B}$ (произведение Адамара)
$\det(\boldsymbol{A})$	определитель $\boldsymbol{A}$

Математический анализ

$\frac{dy}{dx}$	производная y по x
$\frac{\partial y}{\partial x}$	частная производная y по x
$\nabla_{\boldsymbol{x}} y$	градиент y по $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	матрица производных y относительно $\boldsymbol{X}$
$\nabla_{\mathbf{X}} y$	тензор производных y относительно $\mathbf{X}$
$\frac{\partial f}{\partial \boldsymbol{x}}$	матрица Якоби $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ функции $f\colon \mathbb{R}^n \to \mathbb{R}^m$
$\nabla_{\boldsymbol{x}}^2 f(\boldsymbol{x})$ или $\boldsymbol{H}(f)(\boldsymbol{x})$	гессиан функции f в точке $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	определенный интеграл по всей области определения $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	определенный интеграл по множеству $\mathbb{S}$

Теория вероятностей и теория информации

$\mathrm{a} \perp \mathrm{b}$	случайные величины a и b независимы
$\mathrm{a} \perp \mathrm{b} \mid \mathrm{c}$	они условно независимы при условии c
$P(\mathrm{a})$	распределение вероятности дискретной случайной величины
$p(\mathrm{a})$	распределение вероятности непрерывной случайной величины или величины, тип которой не задан
$\mathrm{a} \sim P$	случайная величина a имеет распределение P
$\mathbb{E}_{\mathrm{x} \sim P}[f(x)]$ или $\mathbb{E} f(x)$	математическое ожидание $f(x)$ при заданном распределении $P(\mathrm{x})$
$\mathrm{Var}(f(x))$	дисперсия $f(x)$ при заданном распределении $P(\mathrm{x})$
$\mathrm{Cov}(f(x), g(x))$	ковариация $f(x)$ и $g(x)$ при заданном распределении $P(\mathrm{x})$
$H(\mathrm{x})$	энтропия Шеннона случайной величины x
$D_{\mathrm{KL}}(P \Vert Q)$	расхождение Кульбака–Лейблера между P и Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	нормальное распределение случайной величины $\boldsymbol{x}$ со средним $\boldsymbol{\mu}$ и ковариацией $\boldsymbol{\Sigma}$

Функции

$f\colon \mathbb{A} \to \mathbb{B}$	функция f с областью определения $\mathbb{A}$ и областью значений $\mathbb{B}$
$f \circ g$	композиция функций f и g
$f(\boldsymbol{x}; \boldsymbol{\theta})$	функция от $\boldsymbol{x}$, параметризованная $\boldsymbol{\theta}$ (иногда, чтобы не утяжелять формулы, мы пишем $f(\boldsymbol{x})$, опуская аргумент $\boldsymbol{\theta}$)
$\log x$	натуральный логарифм x
$\sigma(x)$	логистическая сигмоида, $1 / (1 + \exp(-x))$
$\zeta(x)$	функция $\log(1 + \exp(x))$
$\Vert \boldsymbol{x} \Vert_p$	норма L^p вектора $\boldsymbol{x}$
$\Vert \boldsymbol{x} \Vert$	норма L^2 вектора $\boldsymbol{x}$
x^+	положительная часть x, т. е. $\max(0, x)$
$\mathbf{1}_{\mathrm{condition}}$	1, если условие *condition* истинно, иначе 0

Иногда мы применяем функцию f со скалярным аргументом к вектору, матрице или тензору: $f(\boldsymbol{x})$, $f(\boldsymbol{X})$ или $f(\mathbf{X})$. Это означает, что функция f применяется к каждому элементу массива. Например, запись $\mathbf{C} = \sigma(\mathbf{X})$ означает, что $C_{i,j,k} = \sigma(X_{i,j,k})$ для всех i, j, k.

Наборы данных и распределения

p_{data}	распределение, порождающее данные
$\hat{p}_{\text{data}}$	эмпирическое распределение, определенное обучающим набором
$\mathbb{X}$	обучающий набор примеров
$\boldsymbol{x}^{(i)}$	i-й пример из входного набора данных
$y^{(i)}$ или $\boldsymbol{y}^{(i)}$	метка, ассоциированная с $\boldsymbol{x}^{(i)}$ при обучении с учителем
$\boldsymbol{X}$	матрица $m \times n$, в строке $\boldsymbol{X}_{i,:}$ которой находится входной пример $\boldsymbol{x}^{(i)}$

Глава 1

Введение

Изобретатели давно мечтали создать думающую машину. Эти мечты восходят еще к Древней Греции. Персонажей мифов – Пигмалиона, Дедала, Гефеста – можно было бы назвать легендарными изобретателями, а их творения – Галатею, Талоса и Пандору – искусственной жизнью (Ovid and Martin, 2004; Sparkes, 1996; Tandy, 1997).

Впервые задумавшись о программируемых вычислительных машинах, человек задался вопросом, смогут ли они стать разумными, – за сотню с лишним лет до построения компьютера (Lovelace, 1842). Сегодня **искусственный интеллект** (ИИ) – бурно развивающаяся дисциплина, имеющая многочисленные приложения. Мы хотим иметь интеллектуальные программы, которые могли бы автоматизировать рутинный труд, понимали речь и изображения, ставили медицинские диагнозы и поддерживали научные исследования.

Когда наука об искусственном интеллекте только зарождалась, были быстро исследованы и решены некоторые задачи, трудные для человека, но относительно простые для компьютеров – описываемые с помощью списка формальных математических правил. Настоящим испытанием для искусственного интеллекта стали задачи, которые легко решаются человеком, но с трудом поддаются формализации, – задачи, которые мы решаем интуитивно, как бы автоматически: распознавание устной речи или лиц на картинке.

Эта книга посвящена решению таких интуитивных задач. Цель заключается в том, чтобы компьютер мог учиться на опыте и понимать мир в терминах иерархии понятий, каждое из которых определено через более простые понятия. Благодаря приобретению знаний опытным путем этот подход позволяет исключить этап формального описания человеком всех необходимых компьютеру знаний. Иерархическая организация дает компьютеру возможность учиться более сложным понятиям путем построения их из более простых. Граф, описывающий эту иерархию, будет глубоким – содержащим много уровней. Поэтому такой подход к ИИ называется **глубоким обучением**.

Ранние успехи ИИ в большинстве своем были достигнуты в относительно стерильной формальной среде, где от компьютера не требовались обширные знания о мире. Взять, к примеру, созданную IBM шахматную программу Deep Blue, которая в 1997 году обыграла чемпиона мира Гарри Каспарова (Hsu, 2002). Шахматы – это очень простой мир, состоящий всего из 64 клеток и 32 фигур, которые могут ходить лишь строго определенным образом. Разработка успешной стратегии игры в шахматы – огромное достижение, но трудность задачи – не в описании множества фигур и допустимых ходов на языке, понятном компьютеру. Для полного описания игры достаточно очень короткого списка формальных правил, который заранее составляется программистом.

Забавно, что абстрактные, формально поставленные задачи, требующие значительных умственных усилий от человека, для компьютера как раз наиболее просты. Компьютеры давно уже способны обыграть в шахматы сильнейших гроссмейстеров, но лишь в последние годы стали сопоставимы с человеком в части распознавания объектов или речи. В повседневной жизни человеку необходим гигантский объем знаний о мире. Знания эти субъективны и представлены на интуитивном уровне, поэтому выразить их формально затруднительно. Но чтобы вести себя «разумно», компьютерам нужны такие же знания. Одна из основных проблем искусственного интеллекта – как заложить эти неформальные знания в компьютер.

Авторы нескольких проектов в области ИИ пытались представить знания о мире с помощью формальных языков. Компьютер может автоматически рассуждать о предложениях на таком языке, применяя правила логического вывода. В основе таких подходов лежит **база знаний**. Ни один из этих проектов не привел к существенному успеху. Одним из самых известных был проект Cyc (Lenat and Guha, 1989) – машина логического вывода и база утверждений на языке CycL. За ввод утверждений отвечал штат учителей-людей. Процесс оказывается крайне громоздким. Люди изо всех сил пытаются придумать формальные правила, достаточно сложные для точного описания мира. Например, Cyc не сумел понять рассказ о человеке по имени Фред, который бреется по утрам (Linde, 1992). Его машина вывода обнаружила в рассказе противоречие: он знал, что в людях нет электрических деталей, но, поскольку Фред держал электрическую бритву, система решила, что объект «БреющийсяФред» содержит электрические детали. И задала вопрос: является ли Фред по-прежнему человеком, когда бреется.

Трудности, с которыми сталкиваются системы, опирающиеся на «зашитые в код» знания, наводят на мысль, что система с искусственным интеллектом должна уметь самостоятельно накапливать знания, отыскивая закономерности в исходных данных. Это умение называется **машинным обучением**. С появлением машинного обучения перед компьютерами открылась возможность подступиться к задачам, требующим знаний о реальном мире, и принимать решения, кажущиеся субъективными. Простой алгоритм машинного обучения – **логистическая регрессия** – может решить, следует ли рекомендовать кесарево сечение (Mor-Yosef et al., 1990). Другой простой алгоритм – **наивный байесовский классификатор** – умеет отделять нормальную электронную почту от спама.

Качество этих простых алгоритмов сильно зависит от представления исходных данных. Так, система ИИ, выдающая рекомендации о показанности кесарева сечения, не осматривает пациента. Вместо этого врач сообщает системе относящуюся к делу информацию, например о наличии или отсутствии рубца на матке. Каждый отдельный элемент информации, включаемый в представление о пациенте, называется **признаком**. Алгоритм логистической регрессии анализирует, как признаки пациента коррелируют с различными результатами. Но он не может никаким образом повлиять на определение признаков. Если алгоритму предложить снимок МРТ, а не формализованные врачом сведения, то он не сможет выдать полезную рекомендацию. Отдельные пиксели снимка практически не коррелированы с осложнениями, которые могут возникнуть во время родов.

Эта зависимость от представления является общим явлением, проявляющимся как в информатике, так и в повседневной жизни. Если говорить об информатике, то такие операции, как поиск в коллекции данных, будут производиться многократно

быстрее, если коллекция структурирована и подходящим образом индексирована. Люди же легко выполняют арифметические операции с числами, записанными арабскими цифрами, но тратят куда больше времени, если используются римские цифры. Неудивительно, что выбор представления оказывает огромное влияние на качество и производительность алгоритмов машинного обучения. На рис. 1.1 приведен простой наглядный пример.

Многие задачи ИИ можно решить, если правильно подобрать признаки, а затем предъявить их алгоритму машинного обучения. Например, в задаче идентификации говорящего по звукам речи полезным признаком является речевой тракт. Он позволяет с большой точностью определить, кто говорит: мужчина, женщина или ребенок.

Но во многих задачах нелегко понять, какие признаки выделять. Допустим, к примеру, что мы пишем программу обнаружения автомобилей на фотографиях. Мы знаем, что у автомобилей есть колеса, поэтому могли бы счесть присутствие колеса признаком. К сожалению, на уровне пикселей трудно описать, как выглядит колесо. Колесо имеет простую геометрическую форму, но распознавание его изображения может быть осложнено отбрасыванием теней, блеском солнца на металлических деталях автомобиля, наличием щитка, защищающего колесо от грязи, или объектов на переднем плане, частично загораживающих колесо, и т. д.

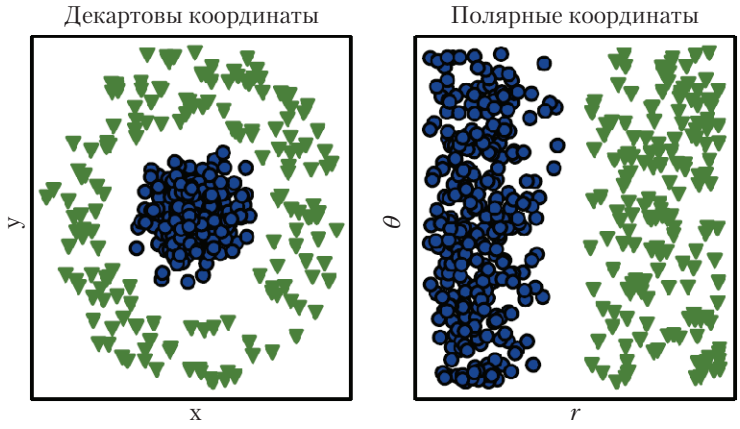

Рис. 1.1 ❖ Пример различных представлений: предположим, что требуется разделить две категории данных, проведя прямую на диаграмме рассеяния. На левом рисунке данные представлены в декартовых координатах, и задача неразрешима. На правом рисунке те же данные представлены в полярных координатах и разделяются вертикальной прямой (рисунок подготовлен совместно с Дэвидом Уорд-Фарли)

Одно из решений этой проблемы – воспользоваться машинным обучением не только для того, чтобы найти отображение представления на результат, но и чтобы определить само представление. Такой подход называется **обучением представлений**. На представлениях, полученных в ходе обучения, часто удается добиться гораздо более высокого качества, чем на представлениях, созданных вручную. К тому же это позволяет системам ИИ быстро адаптироваться к новым задачам при минималь-