

O'REILLY®

Генеративное глубокое обучение

творческий
потенциал
нейронных
сетей



Дэвид Фостер

Generative Deep Learning

*Teaching Machines to Paint, Write,
Compose, and Play*

David Foster

Beijing • Boston • Farnham • Sebastopol • Tokyo

O'REILLY®

Генеративное глубокое обучение

творческий потенциал
нейронных сетей

Дэвид Фостер



Санкт-Петербург • Москва • Екатеринбург • Воронеж
Нижний Новгород • Ростов-на-Дону
Самара • Минск

2020

ББК 32.818
УДК 004.032.26
Ф81

Фостер Дэвид

Ф81 **Генеративное глубокое обучение. Творческий потенциал нейронных сетей.** — СПб.: Питер, 2020. — 336 с.: ил. — (Серия «Бестселлеры O'Reilly»).
ISBN 978-5-4461-1566-2

Генеративное моделирование — одна из самых обсуждаемых тем в области искусственного интеллекта. Машины можно научить рисовать, писать и сочинять музыку. Вы сами можете посадить искусственный интеллект за парту или мольберт, для этого достаточно познакомиться с самыми актуальными примерами генеративных моделей глубокого обучения: вариационные автокодировщики, генеративно-состязательные сети, модели типа кодер-декодер и многое другое.

Дэвид Фостер делает понятными и доступными архитектуру и методы генеративного моделирования, его советы и подсказки сделают ваши модели более творческими и эффективными в обучении. Вы начнете с основ глубокого обучения на базе Keras, а затем перейдете к самым передовым алгоритмам.

- Разберитесь с тем, как вариационные автокодировщики меняют эмоции на фотографиях.
- Создайте сеть GAN с нуля.
- Освойте работу с генеративными моделями генерации текста.
- Узнайте, как генеративные модели помогают агентам выполнять задачи в рамках обучения с подкреплением.
- Изучите BERT, GPT-2, ProGAN, StyleGAN и многое другое.

16+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.818
УДК 004.032.26

Права на издание получены по соглашению с O'Reilly. Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

Издательство не несет ответственности за доступность материалов, ссылки на которые вы можете найти в этой книге. На момент подготовки книги к изданию все ссылки на интернет-ресурсы были действующими.

ISBN 978-1492041948 англ.

Authorized Russian translation of the English edition of Generative Deep Learning

© 2019 Applied Data Science Partners Ltd

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

ISBN 978-5-4461-1566-2

© Перевод на русский язык ООО Издательство «Питер», 2020

© Издание на русском языке, оформление

ООО Издательство «Питер», 2020

© Серия «Бестселлеры O'Reilly», 2020

Краткое содержание

ЧАСТЬ I. ВВЕДЕНИЕ В ГЕНЕРАТИВНОЕ ГЛУБОКОЕ ОБУЧЕНИЕ 20

Глава 1. Генеративное моделирование	21
Глава 2. Глубокое обучение	56
Глава 3. Вариационные автокодировщики	89
Глава 4. Генеративно-сопоставительные сети	125

ЧАСТЬ II. УЧИМ МАШИНЫ РИСОВАТЬ, ПИСАТЬ, СОЧИНЯТЬ МУЗЫКУ И ИГРАТЬ В ИГРЫ 160

Глава 5. Рисование	162
Глава 6. Литературное творчество	197
Глава 7. Сочинение музыки	236
Глава 8. Играем в игры	275
Глава 9. Будущее генеративного моделирования	316
Глава 10. Заключение	342

Оглавление

ПРЕДИСЛОВИЕ	13
Цели и подходы	14
Уровень подготовки	15
Прочие ресурсы	15
Типографские соглашения	17
Использование программного кода примеров	17
Благодарности	18
От издательства	19
 ЧАСТЬ I. ВВЕДЕНИЕ В ГЕНЕРАТИВНОЕ ГЛУБОКОЕ ОБУЧЕНИЕ	20
Глава 1. Генеративное моделирование	21
Что такое генеративное моделирование?	21
Генеративное и дискриминативное моделирование	23
Достижения в машинном обучении	25
Появление генеративного моделирования	26
Основа для генеративного моделирования	29
Вероятностные генеративные модели	32
Привет, Ирма!	35
Ваша первая вероятностная генеративная модель	36
Наивная байесовская параметрическая модель	41
Привет, Ирма! Продолжение	44
Сложности генеративного моделирования	46
Обучение представлению	48
Настройка окружения	52
Итоги	55

Глава 2. Глубокое обучение 56

Структурированные и неструктурированные данные	56
Глубокие нейронные сети	58
Keras и TensorFlow	60
Ваша первая глубокая нейронная сеть	61
Загрузка данных	61
Конструирование модели	63
Компиляция модели	68
Обучение модели	70
Оценка модели	71
Улучшение модели	73
Сверточные слои	74
Пакетная нормализация	79
Слои прореживания	82
Соединяем все вместе	84
Итоги	88

Глава 3. Вариационные автокодировщики 89

Художественная выставка	89
Автокодировщики	92
Ваш первый автокодировщик	94
Кодировщик	94
Декодировщик	96
Объединение кодировщика и декодировщика	99
Анализ автокодировщика	101
Выставка вариационного искусства	104
Конструирование вариационного автокодировщика	106
Кодировщик	106
Функция потерь	112
Анализ вариационного автокодировщика	114
Использование вариационного автокодировщика для генерации изображений лиц	115
Обучение VAE	116
Анализ вариационного автокодировщика	119
Генерирование новых лиц	120

Арифметика скрытого пространства.	121
Преобразование одного лица в другое.	123
Итоги	124
Глава 4. Генеративно-состязательные сети	125
Ганимал	125
Введение в генеративно-состязательные сети	128
Ваша первая генеративно-состязательная сеть	129
Дискриминатор	130
Генератор	132
Обучение генеративно-состязательной сети	136
Проблемы генеративно-состязательных сетей	142
Колебания потерь	142
Коллапс модели	143
Неинформативные потери	144
Гиперпараметры	144
Решение проблем генеративно-состязательных сетей	145
Генеративно-состязательные сети с функцией потерь Вассерштейна	146
Функция потерь Вассерштейна	146
Ограничение Липшица	148
Усечение весов	149
Обучение WGAN	150
Анализ WGAN	151
WGAN-GP	152
Функция потерь штрафа за градиент	153
Анализ WGAN-GP.	157
Итоги	159
ЧАСТЬ II. УЧИМ МАШИНЫ РИСОВАТЬ, ПИСАТЬ, СОЧИНЯТЬ МУЗЫКУ И ИГРАТЬ В ИГРЫ	160
Глава 5. Рисование	162
Яблоки и апельсины	163
CycleGAN	166

Ваша первая сеть CycleGAN	168
Обзор	168
Генераторы (U-Net)	170
Дискриминаторы	174
Компиляция CycleGAN	176
Обучение CycleGAN	178
Анализ CycleGAN	179
CycleGAN, рисующая в стиле Моне	181
Генераторы (ResNet)	182
Анализ CycleGAN	184
Нейронный перенос стиля	185
Потеря содержимого	187
Потеря стиля.	190
Потеря общей дисперсии.	193
Запуск нейронного переноса стиля	194
Анализ модели нейронного переноса стиля	195
Итоги	196

Глава 6. Литературное творчество 197

Литературное общество для проблемных правонарушителей	198
Сети с долгой краткосрочной памятью	200
Ваша первая сеть LSTM	201
Лексемизация	201
Создание набора данных	204
Архитектура модели LSTM	205
Слой Embedding	206
Слой LSTM.	207
Ячейка LSTM	209
Генерирование нового текста	212
Расширения RNN	216
Многослойные рекуррентные сети	217
Управляемые рекуррентные блоки.	218
Двунаправленные ячейки	220
Модели кодировщик-декодировщик	220
Генератор вопросов и ответов.	223

Набор данных с вопросами и ответами	224
Архитектура модели	226
Вычисление результатов	231
Результаты моделирования	233
Итоги	235
Глава 7. Сочинение музыки	236
Вступление	237
Нотная запись	237
Ваша первая сеть RNN для генерирования музыки	240
Внимание	242
Конструирование механизма внимания с помощью Keras	244
Анализ сети RNN с механизмом внимания	249
Механизм внимания в сетях типа кодировщик-декодировщик	254
Генерирование полифонической музыки	258
Музыкальный орган	258
Ваша первая сеть MuseGAN	260
Генератор MuseGAN	263
Аккорды, стиль, мелодия и дорожки	265
Генератор тактов	267
Объединяем все вместе	268
Критик	270
Анализ сети MuseGAN	271
Итоги	273
Глава 8. Играем в игры	275
Обучение с подкреплением	276
OpenAI Gym	278
Архитектура модели мира	280
Вариационный автокодировщик	281
Сеть MDN-RNN	282
Контроллер	283
Подготовка	284

Обзор процесса обучения	285
Сбор данных в ходе случайных прогонов	286
Обучение VAE	290
Архитектура VAE	291
Анализ VAE	291
Сбор данных для обучения RNN	296
Обучение сети MDN-RNN	297
Архитектура сети MDN-RNN	298
Выборка следующего состояния и вознаграждения из MDN-RNN	300
Функция потерь в MDN-RNN	300
Обучение контроллера	302
Архитектура контроллера	303
CMA-ES	304
Параллельное выполнение алгоритма CMA-ES	307
Вывод контроллера в процессе обучения	309
Обучение в мнимом окружении	310
Обучение контроллера в мнимом окружении	312
Недостатки обучения в мнимом окружении	314
Итоги	315

Глава 9. Будущее генеративного моделирования 316

Пять лет прогресса	317
Трансформер	319
Позиционное кодирование	319
Многоголовое внимание	322
Декодировщик	324
Анализ трансформера	325
BERT	326
GPT-2	327
MuseNet	328
Достижения в генерировании изображений	329
ProGAN	329
Self-Attention GAN (SAGAN)	332
BigGAN	333
StyleGAN	335

Области применения генеративного моделирования	339
Изобразительное творчество искусственного интеллекта	339
Музыкальное творчество искусственного интеллекта	340
Глава 10. Заключение	342
ОБ АВТОРЕ.	345
ОБ ОБЛОЖКЕ.	346

Предисловие

Чего не могу воссоздать, того не понимаю.

Ричард Фейнман

Способность творить — неотъемлемая часть человеческой природы. Еще на заре своего существования, живя в пещерах, человек искал возможность создавать оригинальные и красивые творения. Творчество первобытного человека выражалось в наскальных рисунках с дикими животными и абстрактными узорами, созданных с помощью пигментов, аккуратно и методично нанесенных на скалу. Эпоха романтизма подарила нам чудо симфоний Чайковского, с их способностью вызывать чувства триумфа и трагедии посредством звуковых волн, сплетенных в прекрасные мелодии и гармонии. И в наши дни мы, бывает, выскакиваем в полночь из дому, чтобы забежать в книжный магазин и купить продолжение истории о вымышленном волшебнике, потому что комбинация букв образует захватывающее повествование, заставляющее нас переворачивать страницу за страницей, чтобы узнать, какие еще приключения ждут нашего героя.

Поэтому неудивительно, что человечество задалось главным вопросом: можем ли мы создать что-то, что было бы творческим само по себе?

Поиск ответа на этот вопрос является целью генеративного моделирования. Благодаря последним достижениям в науке и технике мы можем создавать машины, способные рисовать оригинальные картины в определенном стиле, писать абзацы связного текста с хорошо прослеживаемой структурой, сочинять музыку, которую приятно слушать, и разрабатывать выигрышные стратегии для сложных игр, генерируя сценарии возможного развития событий. И это только начало генеративной революции, не оставляющей нам другого выбора, кроме отыскания ответов на некоторые из самых важных вопросов о механике творчества и, в конечном итоге, о том, что значит быть человеком.

Иными словами, сейчас самое время заняться изучением генеративного моделирования, так давайте приступим!

Цели и подходы

В этой книге рассматриваются ключевые методы, доминировавшие в ландшафте генеративного моделирования в последние годы и позволившие добиться впечатляющего прогресса в творческих задачах. Кроме знакомства с базовой теорией генеративного моделирования, в этой книге мы будем создавать действующие примеры некоторых ключевых моделей, заимствованных из литературы, и шаг за шагом рассмотрим реализацию каждой из них.

На протяжении всей книги вам будут встречаться короткие поучительные истории, объясняющие механику некоторых моделей. Пожалуй, один из лучших способов изучения новой абстрактной теории — сначала преобразовать ее во что-то менее абстрактное, например в рассказ, и только потом погружаться в техническое описание. Отдельные разделы теории будут более понятны в контексте, включающем людей, действия и эмоции, а не в контексте таких довольно абстрактных понятий, как, допустим, нейронные сети, обратное распространение или функции потерь.

Рассказ и описание модели — это обычный прием объяснения одного и того же с двух точек зрения. Поэтому, изучая какую-то модель, иногда будет полезно вернуться к соответствующему рассказу. Если же вы уже знакомы с конкретным приемом, то просто получайте удовольствие, обнаруживая в рассказах параллели с каждым элементом модели!

В первой части книги представлены ключевые методы построения генеративных моделей, включая обзор глубокого обучения, вариационных автокодировщиков и генеративно-состязательных сетей. Во второй части эти методы применяются для решения нескольких творческих задач (рисование, сочинение рассказов и музыки) с помощью таких моделей, как CycleGAN, моделей типа кодер-декодер и MuseGAN. Мы увидим, как генеративное моделирование можно использовать для оптимизации выигрышной стратегии игры (World Models), рассмотрим самые передовые генеративные архитектуры, доступные сегодня: StyleGAN, BigGAN, BERT, GPT-2 и MuseNet.

Уровень подготовки

Эта книга предполагает, что у читателя есть опыт программирования на Python. Если вы не знакомы с Python, начните его изучение с сайта **LearningPython.org** (<https://www.learnpython.org/>). В интернете есть много бесплатных ресурсов, позволяющих приобрести достаточный объем знаний о Python для работы с примерами из этой книги.

Кроме того, некоторые модели описаны с использованием математических обозначений, поэтому будет полезно иметь достаточно полное представление о линейной алгебре (например, как выполняется умножение матриц и т. д.) и общей теории вероятностей.

Наконец, для опробования примеров кода из репозитория GitHub (https://github.com/davidADSP/GDL_code) книги вам потребуется программное окружение. Примеры для этой книги сознательно создавались так, чтобы не задействовались чрезмерно большие вычислительные ресурсы.

Многие ошибочно считают, что для обучения моделей глубокого обучения необходим графический процессор. Конечно, его наличие не будет лишним и поможет ускорить обучение моделей, но это совсем не обязательно. На самом деле, если вы делаете лишь первые шаги в области глубокого обучения, то начинайте с основ, экспериментируйте с небольшими примерами на ноутбуке и только потом тратьтесь на оборудование для ускорения процесса обучения своих моделей.

Прочие ресурсы

В качестве общего введения в машинное и глубокое обучение рекомендуются две книги:

- «Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems» Жерона Орельена (Geron Aurelien), вышедшая в издательстве O'Reilly;¹

¹ Жерон Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: Концепции, инструменты и техники для создания интеллектуальных систем. М.: Вильямс, 2018. — *Примеч. пер.*

■ «Deep Learning with Python» Шолле Франсуа (Francois Chollet), вышедшая в издательстве Manning.¹

Большинство статей, упоминаемых в этой книге, получены из **arXiv** (<https://arxiv.org/>), бесплатного репозитория научных статей. Сейчас многие авторы публикуют свои статьи в arXiv до их рецензирования. Обзор последних поступлений в репозиторий — отличный способ быть в курсе самых передовых разработок в области машинного обучения.

На сайте **Papers with Code** (<https://paperswithcode.com/>) можно найти самые свежие решения различных задач машинного обучения, ссылки на статьи и официальные репозитории GitHub. Это отличный ресурс для любого желающего выяснить, какие современные методы помогают достичь самых высоких результатов.

Наконец, полезным ресурсом для обучения моделей на высокопроизводительном оборудовании является **Google Colaboratory** (<https://colab.research.google.com/>). Это бесплатное окружение Jupyter Notebook, не требующее настройки и полностью работающее в облаке. Вы можете настроить выполнение своих сценариев на графическом процессоре, предоставляемом бесплатно (до 12 часов работы). Примеры из этой книги необязательно запускать на графическом процессоре, но такая возможность поможет ускорить процесс обучения ваших моделей. В любом случае Colab — отличный способ получить бесплатный доступ к ресурсам графического процессора.

¹ Шолле Франсуа. Глубокое обучение на Python. СПб.: Питер, 2018. — *Примеч. пер.*

Типографские соглашения

В этой книге приняты следующие типографские соглашения.

Курсив используется для обозначения новых терминов, имен файлов и расширений имен файлов.

Моноширинный шрифт применяется для оформления листингов программ и программных элементов внутри обычного текста: имен переменных и функций, баз данных, типов данных, переменных окружения, инструкций и ключевых слов.



Этот элемент используется для обозначения примечаний.

Использование программного кода примеров

Вспомогательные материалы (примеры кода, упражнения и т. д.) доступны для загрузки по адресу https://github.com/davidADSP/GDL_code.

В общем случае все примеры кода из этой книги вы можете использовать в своих программах и в документации. Вам не нужно обращаться в издательство за разрешением, если вы не собираетесь воспроизводить существенные части программного кода из этой книги, ограничиваясь использованием в программе лишь нескольких отрывков кода (в противном случае получение разрешения необходимо, см. ниже). Однако в случае продажи или распространения компакт-дисков с примерами из этой книги вам необходимо получить разрешение издательства O'Reilly. Если вы отвечаете на вопросы, цитируя данную книгу или примеры из нее, то получение разрешения не требуется.

Благодарности

Позвольте поблагодарить всех, кто помогал автору в работе над этой книгой.

Прежде всего, искреннее спасибо тем, кто нашел время для научного редактирования, в частности: Любе Эллиотт (Luba Elliott), Даррену Ричардсону (Darren Richardson), Эрику Джорджу (Eric George), Крису Шону (Chris Schon), Сигурдуре Скули Сигургейрссону (Sigurður Skúli Sigurgeirsson), Хао-Вен Донгу (Hao-Wen Dong), Дэвиду Ха (David Ha) и Лорне Барклай (Lorna Barclay).

Кроме того, автор выражает огромную благодарность коллегам из Applied Data Science Partners: Россу Витешчаку (Ross Witeszczak), Крису Шону (Chris Schon), Дэниелу Шарпу (Daniel Sharp) и Эми Булл (Amy Bull). Спасибо за вашу благосклонность в течение всего времени работы над книгой, с нетерпением жду новых проектов машинного обучения, реализацией которых в будущем мы будем заниматься вместе! Особое спасибо Россу за веру в меня как в делового партнера — если бы мы не решили начать совместный бизнес, то эта книга, возможно, никогда бы не появилась!

Я также хочу поблагодарить всех, кто когда-либо учил меня математике, — мне очень повезло, что в школе у меня были фантастические преподаватели, которые развили мой интерес к этому предмету и призывали меня продолжить развивать его в университете. Благодарю вас за вашу самоотдачу и за то, что щедро делились своими знаниями со мной.

Огромное спасибо сотрудникам издательства O'Reilly за то, что помогали мне в процессе работы над этой книгой. Особую благодарность хочу выразить Мишель Кронин (Michele Cronin), помогавшей на каждом этапе полезной обратной связью и посылавшей мне дружеские напоминания о необходимости продолжать писать главы! Спасибо также Кэти Тозер (Katie Tozer), Рейчел Хед (Rachel Head) и Мелани Ярбро (Melanie Yarbrough) за подготовку книги к печати и Майку Лоукидесу (Mike Loukides), который первым обратился ко мне с предложением написать книгу. Вы все оказали огромную поддержку этому проекту, и я хочу поблагодарить вас за то, что вы предоставили мне возможность написать книгу о том, что я люблю.

Все время, пока я писал книгу, моя семья оказывала мне всяческую поддержку. Огромное спасибо моей маме, Джиллиан Фостер, за проверку каждой строки текста на наличие опечаток и особенно за то, что научила

меня считать! Твое внимание к деталям чрезвычайно помогло при редактировании этой книги, и я очень благодарен за все возможности, которые предоставили мне ты и папа. Мой папа, Клайв Фостер, научил меня программированию — эта книга полна практических примеров, которые я сумел написать во многом благодаря его терпению в те годы, когда я изучал Бейсик и пытался писать футбольные игры. Мой брат, Роб Фостер, — самый скромный гений из всех гениев, особенно в области лингвистики, но беседы с ним об искусственном интеллекте и будущем машинного обучения на основе текстов оказались на удивление плодотворными. Наконец, я хотел бы поблагодарить мою сестру Нану — неиссякаемый источник вдохновения и радости. Ее любовь к литературе — одна из причин, по которой я решил, что написать книгу было бы захватывающим занятием.

Наконец, я хотел бы поблагодарить мою невесту (совсем скоро она станет моей женой) Лорну Барклай. Она не только тщательно просмотрела каждое слово в этой книге, но и оказывала мне бесконечную поддержку на протяжении всего процесса, заваривала чай, приносила разные вкусняшки и вообще делала все, чтобы превратить эту книгу в лучшее руководство по генеративному моделированию, уделяя огромное внимание деталям и делясь своими глубокими знаниями в области статистики и машинного обучения. Я едва ли смог бы завершить этот проект без тебя и благодарен за то время, которое ты потратила, чтобы помочь мне реорганизовать и расширить части книги, которые нуждались в дополнительном объяснении. Обещаю, что не буду говорить о генеративном моделировании за обеденным столом, по крайней мере, несколько недель после публикации книги.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На веб-сайте издательства www.piter.com вы найдете подробную информацию о наших книгах.

Часть I

Введение в генеративное глубокое обучение

Цель первых четырех глав — познакомить читателей с основными приемами, необходимыми для того, чтобы начать строить генеративные модели глубокого обучения.

В **главе 1** в общих чертах рассмотрена область генеративного моделирования, задачи, которые эта область решает, а также пример простой вероятностной генеративной модели. Здесь же поясняется, почему по мере роста сложности генеративной задачи необходимо применять методы глубокого обучения.

Глава 2 содержит руководство по инструментам и методам глубокого обучения, овладение которыми необходимо, чтобы начать создавать более сложные генеративные модели. Эта глава скорее практическое руководство по глубокому обучению, чем теоретический анализ. В частности, здесь представлен фреймворк Keras — инструмент для построения и обучения нейронных сетей с самыми передовыми архитектурами, упоминаемыми в современных источниках.

В **главе 3** рассмотрена наша первая генеративная модель глубокого обучения — вариационный автокодировщик, позволяющий не только создавать реалистичные изображения лиц, но и изменять существующие изображения, например, добавляя улыбку или изменяя цвет волос.

Глава 4 исследует один из самых успешных методов генеративного моделирования последних лет — генеративно-состязательную сеть. Она служит основой для структурирования задач генеративного моделирования и является основной движущей силой большинства современных генеративных моделей. Читатели узнают, как ее настраивать и адаптировать, чтобы постоянно расширять границы возможностей генеративного моделирования.

Генеративное моделирование

Эта глава является общим введением в генеративное моделирование. Сначала посмотрим, что подразумевается под названием *генеративная модель* и чем генеративное моделирование отличается от более широко известного *дискриминативного моделирования*. После этого я представлю базовые принципы и познакомлю с основными математическими понятиями, которые позволят нам сформировать обобщенный подход к задачам, требующим генеративного решения.

Затем, опираясь на новые знания, мы создадим наш первый пример генеративной модели (наивную байесовскую модель), который является вероятностным по своей природе. Он позволит нам генерировать новые образцы, находящиеся за пределами обучающего набора данных, а также исследовать причины, почему модели этого типа могут потерпеть неудачу с увеличением размера и сложности пространства возможных творений.

Что такое генеративное моделирование?

В общих чертах генеративную модель можно определить так:

Генеративная модель описывает, как генерируется набор данных, с точки зрения вероятностной модели. Используя эту модель, можно генерировать новые данные.

Допустим, у нас есть коллекция изображений лошадей и на ее основе нужно построить модель, способную генерировать новые изображения лошадей, которых никогда не существовало, но которые выглядят реальными, по-

тому что модель выучила общие правила, определяющие внешний вид лошади. Эту задачу можно решить с помощью генеративного моделирования. Краткое описание типичного процесса генеративного моделирования показано на рис. 1.1.

Прежде всего, необходим набор данных, состоящий из множества образцов сущности, которую нужно сгенерировать. Этот набор данных называется *обучающим набором*, а один образец данных в наборе называется *наблюдением*.

Каждое наблюдение состоит из множества *признаков* — в задачах генерации изображений роль признаков обычно играют отдельные пикселы. Наша цель — создать модель, способную генерировать новые наборы признаков, которые выглядят так, будто созданы с использованием тех же правил, что и исходные данные. Концептуально генерация изображений — невероятно сложная задача, учитывая огромное количество способов выбора значений для отдельных пикселов и относительно крошечное число вариантов такого их расположения, когда получается изображение, похожее на моделируемый объект.

Генеративная модель также должна быть *вероятностной*, а не *детерминированной*. Если модель просто представляет фиксированные вычисления, например, выбирает среднее значение каждого пиксела в наборе данных, то она не будет генеративной, потому что каждый раз будет давать один и тот же результат. Модель должна включать *стохастический* (случайный) элемент, который влияет на отдельные выборки, генерируемые моделью.

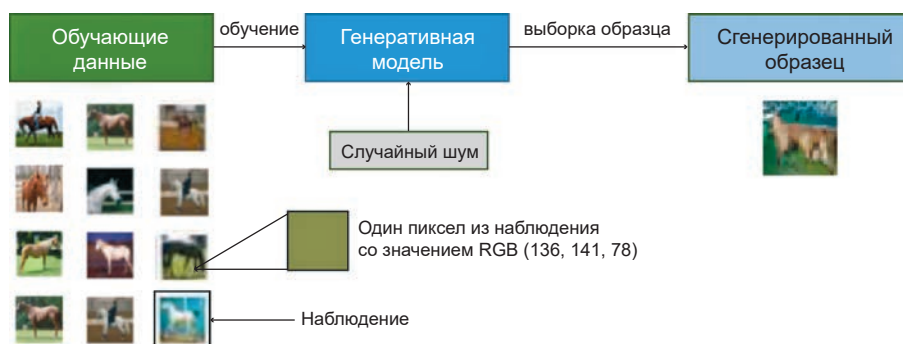


Рис. 1.1. Процесс генеративного моделирования

Другими словами, мы можем представить, что существует какое-то неизвестное вероятностное распределение, объясняющее, почему одни изображения могли бы присутствовать в обучающем наборе, а другие нет. Наша задача — создать модель, максимально точно имитирующую это распределение, а затем произвести выборку из нее, чтобы сгенерировать новые наблюдения, которые выглядят так, будто могли бы иметься в исходном обучающем наборе.

Генеративное и дискриминативное моделирование

Чтобы по-настоящему понять цель и важность генеративного моделирования, полезно сравнить его со своим аналогом, *дискриминативным моделированием*. Знакомые с машинным обучением знают, что большинство задач, с которыми вы столкнетесь, скорее всего, носят дискриминативный характер. Рассмотрим пример, чтобы понять разницу.

Предположим, у нас есть набор данных с коллекцией картин, часть которых написаны Ван Гогом, а часть — другими художниками. Имея достаточный объем данных, мы сможем обучить дискриминативную модель, способную предсказать, была ли данная картина написана Ван Гогом. Наша модель может выучить, какие цвета, формы и текстуры с большей вероятностью будут указывать на принадлежность картины кисти голландского мастера, и в соответствии с этими характеристиками оценивать свой прогноз. На рис. 1.2 показан процесс дискриминативного моделирования — обратите внимание, как он отличается от процесса генеративного моделирования, изображенного на рис. 1.1.



Рис. 1.2. Процесс дискриминативного моделирования

Одно из ключевых отличий состоит в том, что при выполнении дискриминативного моделирования каждое наблюдение в обучающих данных имеет *метку*. Для задачи бинарной классификации, такой как определение принадлежности картины, картины Ван Гога будут помечены меткой *1*, а картины других художников — меткой *0*. Исследовав этот набор, наша модель научится различать эти две группы и выведет вероятность, что новое наблюдение имеет метку *1*, то есть что эта картина нарисована Ван Гогом.

По этой причине дискриминативное моделирование часто называют *обучением с учителем* или определением функции отображения входных данных в выходные с использованием маркированного набора данных. Для построения генеративных моделей обычно используются наборы данных без меток (то есть это форма обучения без учителя), хотя также можно использовать наборы данных с метками, чтобы узнать, как генерировать наблюдения из каждого отдельного класса.

Давайте рассмотрим некоторые математические обозначения, помогающие описать разницу между генеративным и дискриминативным моделированием.

Дискриминативная модель оценивает $p(y|\mathbf{x})$ — вероятность метки y для данного наблюдения \mathbf{x} .

Генеративная модель оценивает $p(\mathbf{x})$ — вероятность получения наблюдения \mathbf{x} .

Если набор данных содержит метки, то можно построить генеративную модель, оценивающую распределение $p(\mathbf{x}|y)$.

Иначе говоря, дискриминативная модель оценивает вероятность того, что наблюдение \mathbf{x} относится к категории y . Генеративная модель не учитывает метки наблюдений и оценивает вероятность того, что сгенерированное наблюдение похоже на остальные наблюдения.

Важно отметить, что даже если бы мы были в состоянии создать идеальную дискриминативную модель для идентификации картин Ван Гога, то она все равно не смогла бы создать картину, похожую на картины Ван Гога. Она сможет лишь определять вероятности для существующих изображений, потому

что ее учили именно этому. Нам же нужно обучить генеративную модель, которая сможет генерировать наборы пикселей, с высокой вероятностью принадлежащие исходному набору обучающих данных.

Достижения в машинном обучении

Чтобы понять, почему генеративное моделирование можно считать следующим рубежом машинного обучения, сначала нужно разобраться, почему дискриминативное моделирование послужило движущей силой для большинства достижений в методологии машинного обучения за последние два десятка лет, как в науке, так и в промышленности.

С академической точки зрения прогресс в дискриминативном моделировании легко проследить, так как есть возможность измерить показатели производительности по определенным задачам классификации и выявить лучшую методологию в своем классе. Оценить генеративные модели труднее, особенно когда оценка качества получаемых результатов в значительной степени субъективна. Поэтому в последние годы большое внимание уделялось обучению дискриминативных моделей для достижения надежности классификации изображений или текста, сравнимой с человеческой или даже превосходящей ее.

Например, в 2012 году произошел важнейший прорыв в классификации изображений, когда команда во главе с Джеффом Хинтоном (Geoff Hinton) из университета в Торонто выиграла конкурс по распознаванию образов ImageNet Large Scale Visual Recognition Challenge (ILSVRC) со своей глубокой сверточной нейронной сетью. По условиям конкурса требовалось классифицировать изображения и отнести их к одной из тысячи категорий, а полученные результаты использовались для сравнения новейших современных методов. Модель глубокого обучения ошиблась в 16 % случаев, оказавшись намного лучше своего ближайшего преследователя с его 26,2 % ошибок. Это вызвало бум в развитии глубокого обучения и повлекло дальнейшее снижение доли ошибок с каждым годом. Победитель конкурса 2015 года впервые добился сверхчеловеческих результатов с долей ошибок 4 %, а современная модель достигла уровня ошибок всего 2 %. Теперь многие считают эту проблему решенной.

Наряду с простотой публикации измеримых результатов в академической среде, дискриминативное моделирование исторически проще применить для решения практических задач, чем генеративное моделирование. При

решении практических задач нас, как правило, не волнует, *как* были сгенерированы данные, нас интересует лишь, как классифицировать или оценить новый образец. Например:

- при изучении спутникового снимка сотрудника министерства обороны будет интересовать только вероятность присутствия на нем вражеских подразделений, а не вероятность того, что это изображение является спутниковым снимком;
- менеджеру по работе с клиентами будет интереснее узнать, какой эмоциональный настрой имеет полученное им электронное письмо, положительный или отрицательный, и он едва ли посчитает полезной генеративную модель, которая может генерировать образцы электронных писем клиентов, которых не существует в действительности;
- врачу полезнее будет знать вероятность наличия признаков глаукомы на данном снимке сетчатки глаза, а не возможность генерировать новые изображения, похожие на снимки задней стенки глазного яблока.

Поскольку большинство решений, имеющих практическое применение, относится к области дискриминативного моделирования, наблюдается рост числа инструментов вида *машинное обучение как услуга* (Machine-Learning-as-a-Service, MLaaS), нацеленных на коммерциализацию дискриминативного моделирования в промышленности и автоматизацию процессов сборки, проверки и мониторинга, являющихся универсальными для почти всех задач дискриминативного моделирования.

Появление генеративного моделирования

Несмотря на то что прогресс в области машинного обучения до сих пор в значительной степени обеспечивало дискриминативное моделирование, наиболее интересные достижения в этой области в последние 3–5 лет стали результатом применения методов глубокого обучения для задач генеративного моделирования.

В частности, возросло внимание средств массовой информации к таким проектам генеративного моделирования, как StyleGAN от NVIDIA¹, который

¹ Теро Каррас (Tero Karras), Самули Лейн (Samuli Laine) и Тимо Айла (Timo Aila), «A Style-Based Generator Architecture for Generative Adversarial Networks», 12 декабря 2018, <https://arxiv.org/abs/1812.04948>.

способен создавать очень реалистичные изображения человеческих лиц, и языковая модель GPT-2 от OpenAI¹, которая может завершить отрывок текста по короткому вступительному абзацу.

На рис. 1.3 показан поразительный прогресс в области создания фото-реалистичных изображений лица, достигнутый с 2014 года². Эти достижения с успехом могут использоваться в таких отраслях, как разработка компьютерных игр и кинематография, и в этих же областях наверняка найдут практическое применение усовершенствованные модели автоматического создания музыки. Еще неизвестно, будем ли мы в обозримом будущем читать новостные статьи или романы, написанные генеративной моделью, но последние достижения в этой области ошеломляют и позволяют надеяться, что этот день когда-нибудь настанет. Однако, несмотря на восхищение от новых достижений, возникают также этические вопросы, связанные с распространением фальшивого контента в интернете, а это означает, что доверять всему, что поступает к нам по общедоступным каналам связи, становится все труднее.



Рис. 1.3. Качество изображений лиц, сгенерированных генеративными моделями, значительно улучшилось за последние четыре года³

Помимо практического применения (многие варианты которого пока не обнаружены), существуют еще три причины, по которым генеративное моделирование можно считать ключом к гораздо более сложным формам искусственного интеллекта, недоступным для дискриминативного моделирования.

¹ Алек Редфорд (Alec Radford) и др., «Language Models Are Unsupervised Multitask Learners», 2019, <https://paperswithcode.com/paper/language-models-are-unsupervised-multitask>.

² Майлз Брандейдж (Miles Brundage) и др., «The Malicious Use of Artificial Intelligence: Forecasting, Prevention, and Mitigation», февраль 2018, https://www.eff.org/files/2018/02/20/malicious_ai_report_final.pdf.

³ Источник: Майлз Брандейдж (Miles Brundage) и др., 2018.

Во-первых, с чисто теоретической точки зрения мы не должны довольствоваться успехами в области классификации данных, но должны стремиться к более полному пониманию, как эти данные были созданы. Это, несомненно, более трудная задача из-за высокой размерности пространства возможных выходных данных и относительно небольшого числа творений, которые мы классифицировали бы как принадлежащие набору данных. Однако, как мы увидим далее, многие из методов, которые привели к развитию дискриминативного моделирования, такие как глубокое обучение, могут использоваться и в генеративных моделях.

Во-вторых, весьма вероятно, что генеративное моделирование окажет существенное влияние на выбор направления будущих разработок в других областях машинного обучения, таких как обучение с подкреплением (когда обучаемый агент получает возможность оптимизировать цели методом проб и ошибок). Например, обучение с подкреплением можно использовать, чтобы обучить робота ходить по данной местности. Для этого можно построить компьютерную имитацию местности, а затем провести множество экспериментов, дав агенту возможность опробовать разные стратегии. Со временем агент выявит наиболее успешные стратегии и, следовательно, будет постепенно совершенствоваться. Типичная проблема этого подхода состоит в том, что физика окружающей среды часто очень сложна и ее необходимо рассчитывать на каждом временном шаге, чтобы передать информацию агенту для принятия решения о его следующем шаге. Однако если агент сможет моделировать свое окружение с помощью генеративной модели, то ему не придется проверять стратегию в компьютерной модели или в реальном мире, поскольку он сможет учиться в своем *воображаемом* окружении. В главе 8 мы увидим эту идею в действии, на примере обучения автомобиля максимально быстрому движению по трассе, позволив ему учиться непосредственно на собственной галлюцинации.

Наконец, если мы действительно зададимся целью создать машину, обладающую интеллектом, сравнимым с человеческим, то нам определенно потребуется использовать приемы генеративного моделирования. Один из ярких примеров генеративной модели в природе — человек, читающий эту книгу. Отвлекитесь на минутку и проведите мысленный эксперимент, который поможет вам понять, насколько невероятной генеративной моделью вы являетесь. Вы можете закрыть глаза и представить, как будет выглядеть

слон под любыми возможными углами зрения. Вы можете вообразить ряд возможных концовок вашего любимого телешоу и спланировать свою неделю наперед, прорабатывая разные варианты будущего в своем воображении. Современная нейробиологическая теория предполагает, что наше восприятие реальности — это не сложная дискриминативная модель, получающая информацию от органов чувств и производящая предсказания на ее основе, а генеративная модель, которая с рождения обучается моделированию нашего окружения, точно соответствующего будущему. Некоторые теории даже предполагают, что результатом этой генеративной модели является наше непосредственное восприятие реальности. Очевидно, что глубокое понимание того, как создавать машины, обладающие этими способностями, будет иметь ключевое значение для нашего дальнейшего понимания работы мозга в частности и искусственного интеллекта в целом.

А теперь, после краткого знакомства с историей и перспективами, начнем наше путешествие в захватывающий мир генеративного моделирования. Для начала рассмотрим простейшие примеры генеративных моделей и некоторые идеи, которые помогут нам перейти к более сложным архитектурам, с которыми мы столкнемся позже в книге.

Основа для генеративного моделирования

Для начала поиграем в генеративное моделирование в двух измерениях. Я выбрал правило, согласно которому был сгенерирован набор точек \mathbf{X} , изображенных на рис. 1.4. Назовем это правило p_{data} . Ваша задача состоит в том, чтобы выбрать другую точку $\mathbf{x} = (x_1, x_2)$ в пространстве, которая выглядит так, будто она сгенерирована тем же правилом.

Выбрали? Вы, вероятно, использовали свои знания о существующих точках, чтобы построить ментальную модель p_{model} и определить, где в пространстве должна находиться эта точка. В этом отношении p_{model} является *оценкой* p_{data} . Возможно, вы решили, что p_{model} должна выглядеть, как показано на рис. 1.5, — прямоугольное поле, в котором могут находиться точки, и область за границами прямоугольника, где точки находиться не должны. Чтобы сгенерировать новое наблюдение, можно просто выбрать случайную точку внутри поля или, выражаясь более формальным языком, *выбрать образец* из распределения p_{model} . Поздравляю, вы только что разработали свою первую генеративную модель!

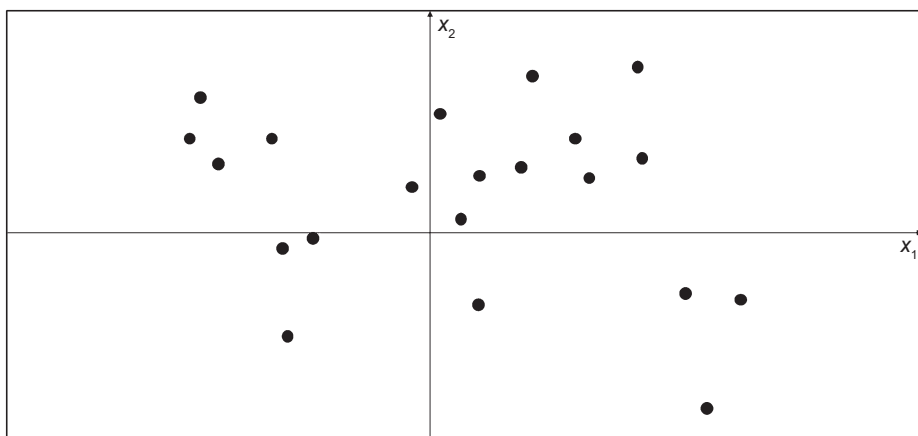


Рис. 1.4. Набор точек на двумерной плоскости, сгенерированный с использованием неизвестного правила p_{data}

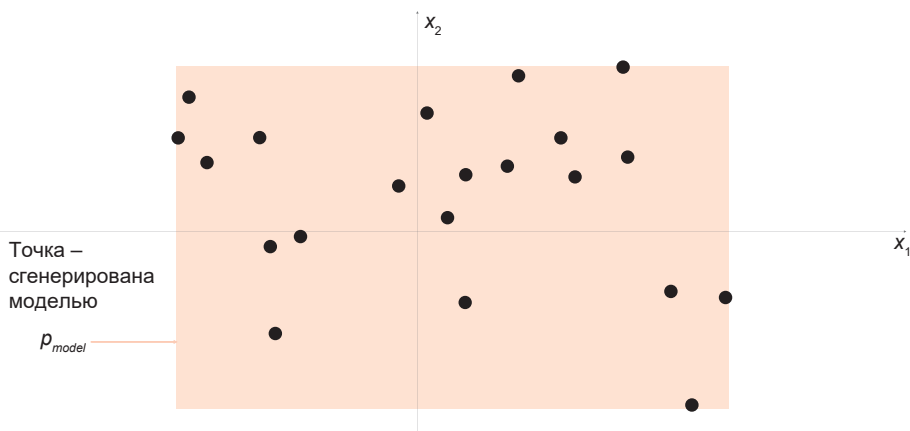


Рис. 1.5. Оранжевый прямоугольник p_{model} — это оценка фактического распределения p_{data}

Это не самый сложный пример, но его можно использовать, чтобы понять цель генеративного моделирования. Следующие базовые принципы определяют наши мотивы.

Теперь раскроем истинное распределение данных p_{data} и посмотрим, как базовые принципы применяются к этому примеру.

Как показано на рис. 1.6, правило, на основе которого получены данные, — это просто равномерное распределение точек по земной суше.

Очевидно, что наша модель p_{model} — это упрощенное представление p_{data} . Точки А, В и С соответствуют трем наблюдениям, сгенерированным моделью p_{model} с разной степенью успеха:

- **Точка А** нарушает правило 1 базовых принципов генеративного моделирования — ясно видно, что она не принадлежит распределению p_{data} , поскольку находится посреди моря.
- **Точка В** настолько близко расположена от точки из исходного набора данных, что мы не впечатлены способностью модели создать такую точку. Если все образцы, сгенерированные моделью, будут расположены так же близко к точкам из исходного набора, то это можно считать нарушением правила 2 базовых принципов генеративного моделирования.
- **Точку С** можно рассматривать как успех, потому что она вполне могла быть получена из распределения p_{data} и существенно отличается от всех точек в исходном наборе.



Рис. 1.6. Оранжевые прямоугольники p_{model} — это оценка истинного распределения p_{data} (серые области)

БАЗОВЫЕ ПРИНЦИПЫ ГЕНЕРАТИВНОГО МОДЕЛИРОВАНИЯ

- У нас есть набор данных с наблюдениями \mathbf{X} .
- Предполагается, что наблюдения сгенерированы в соответствии с некоторым неизвестным распределением p_{data} .
- Генеративная модель p_{model} пытается имитировать p_{data} . Правильно подобрав модель p_{model} , мы сможем с ее помощью генерировать наблюдения, которые выглядят так, будто были получены из p_{data} .
- Модель p_{model} впечатлит нас, если:
 - ✓ Правило 1: она сможет генерировать образцы, которые выглядят так, будто получены из p_{data} .
 - ✓ Правило 2: она сможет генерировать образцы, отличающиеся от наблюдений в \mathbf{X} . То есть модель не должна просто воспроизводить уже известные ей наблюдения.

Область генеративного моделирования разнообразна, и определения задач могут принимать самые разные формы. Однако в большинстве сценариев базовые принципы генеративного моделирования достаточно полно отражают то, насколько широко мы должны думать о решении задачи.

А теперь создадим наш первый пример генеративной модели.

Вероятностные генеративные модели

Сразу хочу успокоить тех, кто никогда не изучал теорию вероятностей. Чтобы построить и опробовать большинство моделей глубокого обучения, которые мы увидим далее в этой книге, необязательно иметь глубокое понимание математической статистики. Но чтобы получить полное представление об истории задачи, которую мы пытаемся решить, стоит попробовать создать генеративную модель, которая опирается не на глубокое обучение, а исключительно на теорию вероятностей. Это поможет заложить основы для понимания всех генеративных моделей, основанных и не основанных на глубоком обучении, с одной и той же вероятностной точки зрения.



Если вы хорошо знакомы с теорией вероятностей, это здорово, и тогда большая часть следующего раздела будет вам знакома. Но в середине этой главы есть забавный пример, не пропустите его!

Первым делом определим четыре ключевых термина: *выборочное пространство*, *функция плотности*, *параметрическое моделирование* и *оценка максимального правдоподобия*.

ФУНКЦИЯ ПЛОТНОСТИ ВЕРОЯТНОСТИ

Функция плотности вероятности (или просто *функция плотности*), $p(\mathbf{x})$, отображает точку \mathbf{x} из выборочного пространства в число от 0 до 1. Сумма¹ функции плотности по всем точкам в выборочном пространстве должна быть равна 1, то есть это четко определенное распределение вероятностей.²

В примере с картой мира функция плотности нашей модели равна 0 за пределами оранжевого прямоугольника и постоянна внутри него.

ПАРАМЕТРИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Параметрическая модель $p_\theta(\mathbf{x})$ — это семейство функций плотности, которое можно описать с использованием конечного числа параметров θ .

Семейство всех возможных прямоугольников, которые можно было бы нарисовать на рис. 1.5, является примером параметрической модели. В данном случае мы имеем четыре параметра: координаты левого нижнего (θ_1, θ_2) и правого верхнего (θ_3, θ_4) углов прямоугольника.

Таким образом, каждая функция плотности $p_\theta(\mathbf{x})$ в этой параметрической модели (то есть каждый прямоугольник) может быть уникально представлена четырьмя числами: $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$.

¹ Или интеграл, если выборочное пространство непрерывное.

² Если выборочное пространство дискретно, тогда $p(\mathbf{x})$ — это просто вероятность принадлежности точки \mathbf{x} к множеству наблюдений.

ПРАВДОПОДОБИЕ

Вероятность $\mathcal{L}(\theta|\mathbf{x})$ набора параметров θ является функцией, которая измеряет правдоподобие θ с учетом некоторой наблюдаемой точки \mathbf{x} .

Определяется она как

$$\mathcal{L}(\theta|\mathbf{x}) = p_{\theta}(\mathbf{x})$$

То есть правдоподобие θ некоторой наблюдаемой точки \mathbf{x} определяется как значение функции плотности, параметризованной θ , в точке \mathbf{x} .

При наличии полного набора данных \mathbf{X} независимых наблюдений мы могли бы написать:

$$\mathcal{L}(\theta | \mathbf{X}) = \prod_{\mathbf{x} \in \mathbf{X}} p_{\theta}(\mathbf{x})$$

Поскольку с вычислительной точки зрения найти это произведение довольно сложно, вместо него часто используется логарифм правдоподобия ℓ :

$$\ell(\theta | \mathbf{X}) = \sum_{\mathbf{x} \in \mathbf{X}} \log p_{\theta}(\mathbf{x})$$

Есть вполне определенные статистические причины, объясняющие, почему правдоподобие определяется именно так, но нам достаточно будет интуитивного понимания. Мы просто определяем правдоподобие того, что набор параметров θ равен вероятности увидеть данные в рамках модели, параметризованной θ .

В примере с картой мира оранжевый прямоугольник, покрывающий только левую половину карты, имел бы правдоподобие 0 — он не сможет сгенерировать набор данных, потому что мы наблюдаем точки и в правой половине карты. Оранжевый прямоугольник на рис. 1.5 имеет положительное правдоподобие, так как функция плотности положительна для всех точек данных в этой модели.

Существует только одна истинная функция плотности p_{data} , которая, как предполагается, сгенерировала наблюдаемый набор данных, и бесконечное число функций плотности p_{model} , которые можно использовать для

Дэвид Фостер

**Генеративное глубокое обучение.
Творческий потенциал нейронных сетей**

Перевел с английского А. Киселев

Заведующая редакцией
Ведущий редактор
Литературный редактор
Корректоры
Художественный редактор
Верстка

*Ю. Сергиенко
К. Тульцева
М. Рогожин
С. Беляева, Н. Викторова
В. Мостипан
Л. Соловьева*

Изготовлено в России. Изготовитель: ООО «Прогресс книга».
Место нахождения и фактический адрес: 194044, Россия, г. Санкт-Петербург,
Б. Сампсониевский пр., д. 29А, пом. 52. Тел.: +78127037373.

Дата изготовления: 03.2020.
Наименование: книжная продукция.
Срок годности: не ограничен.

Налоговая льгота — общероссийский классификатор продукции ОК 034-2014,
58.11.12 — Книги печатные профессиональные, технические и научные.

Импортер в Беларусь: ООО «ПИТЕР М», 220020, РБ, г. Минск,
ул. Тимирязева, д. 121/3, к. 214, тел./факс: 208 80 01.

Подписано в печать 26.02.20. Формат 70×100/16. Бумага офсетная.
Усл. п. л. 28,380. Тираж 1000. Заказ 0000.