

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, Ирм!	35
Ваша первая вероятностная генеративная модель	36
Наивная байесовская параметрическая модель	41
Привет, Ирм! 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-сопоставительные сети 125

Ганимал	125
Введение в генеративно-сопоставительные сети	128
Ваша первая генеративно-сопоставительная сеть	129
Дискриминатор	130
Генератор	132
Обучение генеративно-сопоставительной сети	136
Проблемы генеративно-сопоставительных сетей	142
Колебания потерь	142
Коллапс модели	143
Неинформативные потери	144
Гиперпараметры	144
Решение проблем генеративно-сопоставительных сетей	145
Генеративно-сопоставительные 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

Наивная байесовская параметрическая модель

Наивная байесовская параметрическая модель использует простое предположение, чтобы резко уменьшить количество параметров для оценки. Мы *наивно* предполагаем, что каждый признак x_j *не зависит* от любого другого признака x_k .¹ В отношении набора данных, полученных на Ирме, это означает, например, что выбор цвета волос не влияет на выбор типа одежды, а выбор типа очков не влияет на выбор прически. Более формально, для всех признаков x_j, x_k :

$$p(x_j|x_k) = p(x_j).$$

Это известно как *наивное байесовское* предположение. Чтобы применить его, сначала используется цепное правило вероятности для записи функции плотности в виде произведения условных вероятностей:

$$\begin{aligned} p(\mathbf{x}) &= p(x_1, \dots, x_K) = \\ &= p(x_2, \dots, x_K | x_1) p(x_1) = \\ &= p(x_3, \dots, x_K | x_2, x_1) p(x_2 | x_1) p(x_1) = \\ &= \prod_{k=1}^K p(x_k | x_1, \dots, x_{k-1}), \end{aligned}$$

где K — общее число признаков (то есть пять в примере с планетой Ирм).

Теперь применим наивное байесовское предположение, чтобы упростить последнюю строку:

$$p(\mathbf{x}) = \prod_{k=1}^K p(x_k).$$

Это — наивная байесовская модель. Задача сводится к оценке параметров $\theta_{kl} = p(x_k = l)$ для каждого признака в отдельности и их перемножению для определения вероятности любой возможной комбинации.

Сколько параметров нужно оценить в нашей задаче? Для каждого признака нужно оценить параметр для каждого значения, которое может принять этот

¹ Когда присутствует переменная отклика y , наивное байесовское предположение гласит, что существует *условная* независимость между каждой парой признаков x_j, x_k при заданном y .

признак. Следовательно, в примере с планетой Ирм эта модель определяется всего $7 + 6 + 3 + 4 + 8 - 5 = 23$ параметрами.¹

Оценка максимального правдоподобия $\hat{\theta}_{kl}$ вычисляется как

$$\hat{\theta}_{kl} = \frac{n_{kl}}{N},$$

где $\hat{\theta}_{kl}$ — число раз, когда признак k принимает значение l в наборе данных, а $N = 50$ — общее число наблюдений.

В табл. 1.2 показаны вычисленные параметры для набора данных с планеты Ирм.

Чтобы найти вероятность, с которой модель сгенерирует некоторое наблюдение \mathbf{x} , достаточно перемножить вероятности отдельных признаков. Например:

$$\begin{aligned} & p(\text{длинная стрижка, прямые волосы; рыжий; круглые очки;} \\ & \quad \text{футболка с круглым вырезом; белый}) = \\ & = p(\text{длинная стрижка, прямые волосы}) \times p(\text{рыжий}) \times p(\text{круглые очки}) \times \\ & \quad \times p(\text{футболка с круглым вырезом}) \times p(\text{белый}) = \\ & \quad = 0,46 \times 0,16 \times 0,44 \times 0,38 \times 0,44 = \\ & \quad = 0,0054 \end{aligned}$$

Обратите внимание: эта комбинация отсутствует в исходном наборе данных, но наша модель определяет для нее ненулевую вероятность, а значит, вполне может сгенерировать ее. Кроме того, вероятность этой комбинации выше, чем, например, *(длинная стрижка, прямые волосы; рыжий; круглые очки; футболка с круглым вырезом; синий01)*, потому что белый цвет одежды появляется в наборе наблюдений чаще, чем синий.

То есть наивная байесовская модель способна выявить некоторую структуру данных и использовать ее для создания новых образцов, отсутствующих в исходном наборе. Модель оценила вероятность встретить каждое значение признака независимо от других, поэтому при использовании наивного байесовского предположения можно перемножить эти вероятности, чтобы построить полную функцию плотности, $p_{\theta}(\mathbf{x})$.

¹ Последний член в выражении -5 отражает тот факт, что последний параметр для каждого признака подбирается так, чтобы сумма его параметров была равна 1.

Таблица 1.2. Оценки максимального правдоподобия для параметров в наивной байесовской модели

Прическа	n	$\hat{\theta}$	Цвет волос	n	$\hat{\theta}$	Цвет одежды	n	$\hat{\theta}$
нет волос	7	0,14	черный	7	0,14	черный	0	0,00
длинные волосы, собранные в пучок	0	0,00	блонд	6	0,12	синий01	4	0,08
длинная стрижка, волнистые волосы	1	0,02	каштановый	2	0,04	серый01	10	0,20
длинная стрижка, прямые волосы	23	0,46	пастельный розовый	3	0,06	пастельный зеленый	5	0,10
короткая стрижка, волнистые волосы	1	0,02	рыжий	8	0,16	пастельный оранжевый	2	0,04
короткая стрижка, прямые волосы	11	0,22	серебристо-серый	24	0,48	розовый	4	0,08
короткая стрижка, курчавые волосы	7	0,14	<i>Всего</i>	<i>50</i>	<i>1,00</i>	красный	3	0,06
<i>Всего</i>	<i>50</i>	<i>1,00</i>	<i>Всего</i>	<i>50</i>	<i>1,00</i>	белый	22	0,44
						<i>Всего</i>	<i>50</i>	<i>1,00</i>

Вид очков	n	$\hat{\theta}$
нет	11	0,22
круглые	22	0,44
солнцезащитные	17	0,34
<i>Всего</i>	<i>50</i>	<i>1,00</i>

Тип одежды	n	$\hat{\theta}$
Худи	7	0,14
комбинезон	18	0,36
футболка с круглым вырезом	19	0,38
футболка с V-образным вырезом	6	0,12
<i>Всего</i>	<i>50</i>	<i>1,00</i>

На рис. 1.8 показаны 10 наблюдений, выбранных моделью.

Для этой простой задачи наивное байесовское предположение о независимости признаков является разумным и, следовательно, дает хорошую генеративную модель.

Теперь посмотрим, что получится, если это предположение оказывается ошибочным.



Рис. 1.8. Десять новых стилей для ирмян, сгенерированных наивной байесовской моделью

Привет, Ирм! Продолжение

Вы испытываете определенное чувство гордости, глядя на десять новых творений, созданных вашей наивной байесовской моделью. Воодушевленные своим успехом, вы обращаете внимание на другую сторону задачи, и на этот раз она не выглядит такой же простой.

Набор данных с незамысловатым названием Planet Pixel, который был вам предоставлен, не содержит пяти высокоуровневых признаков, которые вы видели выше (*цвет волос, тип аксессуара* и т. д.), а только значения 32×32 пикселей, составляющих каждое изображение. То есть каждое наблюдение теперь имеет $32 \times 32 = 1024$ признака и каждый признак может принимать любое из 256 значений (отдельные цвета в палитре).

Изображения из нового набора данных показаны на рис. 1.9, а выборка значений пикселей для первых десяти наблюдений показана в табл. 1.3.

Вы решаете попробовать наивную байесовскую модель еще раз, на этот раз обученную на наборе данных пикселей. Модель оценит параметры максимального правдоподобия, определяющие распределение цвета каждого пиксела, чтобы на основе этого распределения сгенерировать новые наблюдения. Однако, закончив модель, вы понимаете, что что-то пошло не так. Вместо новых образцов моды модель вывела десять похожих друг на друга изображений, на которых нельзя различить ни аксессуары, ни четкие признаки прически или одежды (рис. 1.10). Почему так случилось?



Рис. 1.9. Образцы моды в Planet Pixel

Таблица 1.3. Значения пикселей 458–467 для первых десяти наблюдений в Planet Pixel

№ образца	px_458	px_459	px_460	px_461	px_462	px_463	px_464	px_465	px_466	px_467
0	49	14	14	19	7	5	5	12	19	14
1	43	10	10	17	9	3	3	18	17	10
2	37	12	12	14	11	4	4	6	14	12
3	54	9	9	14	10	4	4	16	14	9
4	2	2	5	2	4	4	4	4	2	5
5	44	15	15	21	14	3	3	4	21	15
6	12	9	2	31	16	3	3	16	31	2
7	36	9	9	13	11	4	4	12	13	9
8	54	11	11	16	10	4	4	19	16	11
9	49	17	17	19	12	6	6	22	19	17



Рис. 1.10. Десять новых стилей, сгенерированных наивной байесовской моделью на основе набора данных Planet Pixel

Сложности генеративного моделирования

Во-первых, поскольку наивная байесовская модель выбирает пиксели независимо друг от друга, она не понимает, что два смежных пиксела обычно имеют похожие оттенки, потому что, например, являются частью одного и того же предмета одежды. Модель может сгенерировать цвет лица и рта, поскольку все эти пиксели в обучающем наборе имеют примерно одинаковый оттенок во всех наблюдениях; однако цвет пикселей для футболки выбирается случайно из множества цветов, присутствующих в обучающем наборе, независимо от цвета соседних пикселей. Кроме того, отсутствует механизм формирования пикселей в области вокруг глаз, позволяющий получить круглые очки, или в верхней части изображения, чтобы получить конкретную прическу.

Во-вторых, на этот раз в выборочном пространстве имеется непостижимо большое количество возможных наблюдений. Лишь небольшая часть из них — узнаваемые лица, а еще меньшее подмножество — лица, которые придерживаются правил моды в Planet Pixel. Следовательно, если наивная байесовская модель обучается на сильно коррелированных значениях пикселей, то вероятность найти удовлетворительную комбинацию значений оказывается очень мала.

В первом случае мы имели независимые признаки и относительно небольшое выборочное пространство, поэтому наивная байесовская модель показала неплохие результаты. Во втором случае, когда данные были представлены наборами пикселей, предположение о независимости значений пикселей оказалось несостоятельным. Значения пикселей сильно коррелированы, а выборочное пространство огромно, поэтому получить правильное изображение путем независимой выборки пикселей практически невозможно. Это объясняет, почему наивные байесовские модели плохо работают с обработанными изображениями.

Рассмотренный нами пример высветил две ключевые проблемы, которые должна преодолеть генеративная модель, чтобы добиться успеха.

СЛОЖНОСТИ ГЕНЕРАТИВНОГО МОДЕЛИРОВАНИЯ

- Как модель может справиться с высокой условной взаимозависимостью признаков?
- Как модель может отыскать одну из крошечных пропорций, чтобы получить удовлетворительное наблюдение в многомерном выборочном пространстве?

Ключом к решению обеих этих проблем является глубокое обучение.

Нам нужна модель, способная выявить релевантную структуру в данных, которая не требует делать каких-либо предположений заранее. Именно с этим прекрасно справляется глубокое обучение, и именно поэтому этот метод стал движущей силой последних достижений в генеративном моделировании.

Тот факт, что глубокое обучение может формировать свои признаки в пространстве более низкой размерности, означает, что это — форма *обучения представлению*. Прежде чем приступить к глубокому обучению в следующей главе, мы должны разобраться с ключевыми понятиями обучения представлению.