Transformers in Action

NICOLE KOENIGSTEIN
FOREWORD BY LUIS SERRANO



MANNING
SHELTER ISLAND

ТРАНСФОРМЕРЫ В ДЕЙСТВИИ

Николь Кёнигштайн

Астана
«АЛИСТ»
2026

УДК 004.89
ББК 32.973.26-018.1
К35

Кёнигштайн Н.

К35 Трансформеры в действии: пер. с англ. — Астана: АЛИСТ, 2026. — 320 с.: ил.

ISBN 978-601-12-9652-6

Основополагающая книга о практическом применении больших языковых моделей класса трансформеров (generative pretrained transformer, GPT). Рассказано, как устроены трансформеры на уровне архитектуры и какое место они занимают среди разнообразных нейронных сетей. В книге исследованы самовнимание, тонкая настройка моделей, логический вывод (инференс), обучение с подкреплением, адаптация трансформеров к работе в реальных условиях. Для практической отработки материала предлагаются готовые ноутбуки Jupyter. Основные акценты в примерах сделаны на извлечении данных и, в частности, на технологии RAG (генерация с дополненной выборкой), оптимизации производительности трансформеров, специализации трансформеров с подготовкой предметно-ориентированных языковых моделей, а также на расширении и эффективном использовании контекстного окна.

Для специалистов по большим языковым моделям и анализу данных

УДК 004.89
ББК 32.973.26-018.1

© 2026 ALIST LLP

Authorized translation of the English edition © 2026 Manning Publications. This translation is published and sold by permission of Manning Publications, the owner of all rights to publish and sell the same.

Авторизованный перевод английской редакции книги © 2026 Manning Publications. Перевод опубликован и продается с разрешения компании-правообладателя Manning Publications.

ISBN 978-1-63343-788-3 (англ.)
ISBN 978-601-12-9652-6 (каз.)

© Manning Publications, 2026
© Издание на русском языке. ТОО "АЛИСТ", 2026

Краткое содержание

- *Предисловие* 13
- *Вступление* 15
- *Благодарности* 17
- *О книге* 19
- *Об авторе* 23
- *Об изображении на обложке* 25
- ***Часть I. Основы современных моделей-трансформеров*** 27
- *Глава 1. Зачем нужны трансформеры* 29
- *Глава 2. Более пристальный взгляд на трансформеры* 39
- ***Часть II. Генеративные трансформеры*** 67
- *Глава 3. Семейства моделей и варианты архитектуры* 69
- *Глава 4. Стратегии генерации текста и методы промптинга* 94
- *Глава 5. Согласование предпочтений и генерация с дополненным поиском* 124
- ***Часть III. Специализированные модели*** 153
- *Глава 6. Мультимодальные модели* 155
- *Глава 7. Эффективные и специализированные малые языковые модели* 185
- *Глава 8. Обучение и оценка больших языковых моделей* 220
- *Глава 9. Оптимизация и масштабирование больших языковых моделей* 253
- *Глава 10. Этика и ответственность в разработке больших языковых моделей* 279
- *Ссылки* 309
- *Предметный указатель* 312

Содержание

Предисловие **13**

Вступление **15**

Благодарности **17**

О книге **19**

Кому следует прочесть эту книгу 19
Как организована книга: дорожная карта 20
О программном коде 21

Об авторе **23**

Об изображении на обложке **25**

ЧАСТЬ I. ОСНОВЫ СОВРЕМЕННЫХ МОДЕЛЕЙ-ТРАНСФОРМЕРОВ **27**

Глава 1. Зачем нужны трансформеры **29**

1.1. Трансформерный прорыв 30
1.1.1. Перевод до появления трансформеров 30
1.1.2. В чем отличие трансформеров? 31
1.1.3. Устройство механизма внимания 32
1.1.4. Возможности многоголового внимания 33

1.2. Использование трансформеров ... 34
1.3. Когда и зачем использовать трансформеры ... 35
1.4. От трансформеров к LLM: устойчивая модель ... 37
Резюме ... 38

Глава 2. Более пристальный взгляд на трансформеры ... 39

2.1. От моделей seq2seq к трансформерам ... 40
2.1.1. Сложности обучения рекуррентных нейронных сетей (RNN) ... 40
2.1.2. Введение механизмов внимания ... 41
2.1.3. Исчезающие градиенты: трансформеры приходят на помощь ... 42
2.1.4. Взрывающиеся градиенты: когда слишком большие значения градиентов нарушают процесс обучения ... 43
2.2. Архитектура модели ... 44
2.2.1. Стеки энкодера и декодера ... 45
2.2.2. Позиционное кодирование ... 49
2.2.3. Внимание ... 51
2.2.4. Позиционно-зависимые полносвязные сети прямого распространения ... 64
Резюме ... 66

Часть II. Генеративные трансформеры ... 67

Глава 3. Семейства моделей и варианты архитектуры ... 69

3.1. Модели только с декодером ... 69
3.2. Архитектура модели только с декодером ... 71
3.3. Модели только с энкодером ... 76
3.3.1. Маскированное моделирование языка в качестве стратегии предварительного обучения ... 80
3.4. Модели эмбеддингов и RAG ... 82
3.4.1. Что такое эмбеддинг? ... 83
3.5. MoE в больших языковых моделях (LLM) ... 87
3.5.1. Как работает MoE ... 87
Резюме ... 93

Глава 4. Стратегии генерации текста и методы промптинга ... 94

4.1. Методы декодирования и семплирования для генерации текста ... 95
4.1.1. Декодирование методом жадного поиска для генерации текста ... 95
4.1.2. Декодирование методом поиска по лучу для генерации текста ... 98
4.1.3. Top-k семплирование для генерации текста ... 101

4.1.4. Семплирование по ядру для генерации текста ... 104
4.1.5. Семплирование с учетом температуры для генерации текста ... 106
4.2. Искусство промптинга ... 109
4.2.1. Промптинг нулевого примера ... 110
4.2.2. Промптинг с одним и несколькими примерами ... 111
4.2.3. CoT-промптинг ... 113
4.2.4. Структурированная цепочка рассуждений (CoT) с инструктором ... 114
4.2.5. Контрастивный CoT-промптинг ... 115
4.2.6. CoVe-промптинг ... 116
4.2.7. ToT-промптинг ... 117
4.2.8. ThoT-промптинг ... 121
Резюме ... 123

Глава 5. Согласование предпочтений и генерация с дополненным поиском ... 124

5.1. Обучение, подкрепленное обратной связью от человека ... 125
5.1.1. От MDP к обучению с подкреплением ... 125
5.1.2. Улучшение моделей с помощью обратной связи от человека и обучения с подкреплением ... 127
5.2. Согласование больших языковых моделей с помощью прямой оптимизации предпочтений ... 130
5.2.1. Этап SFT ... 132
5.2.2. Обучение LLM с помощью DPO ... 136
5.2.3. Запуск инференса на LLM ... 139
5.2.4. Оптимизированные версии для DPO ... 140
5.2.5. Оптимизация групповой относительной политики ... 141
5.3. MixEval: бенчмарк для устойчивой и ресурсосберегающей оценки ... 143
5.4. Генерация с дополненным поиском (RAG) ... 144
5.4.1. Первое знакомство с RAG ... 145
5.4.2. Когда и почему стоит применять RAG? ... 146
5.4.3. Основные компоненты и проектные решения ... 147
Резюме ... 151

Часть III. Специализированные модели ... 153

Глава 6. Мультимодальные модели ... 155

6.1. Начало работы с мультимодальными моделями ... 156
6.2. Объединение модальностей из разных областей ... 157
6.3. Модально-специфичная токенизация ... 159
6.3.1. Изображения и визуальные эмбеддинги ... 160
6.3.2. Анализ изображений с помощью MLLM ... 163

6.3.3. От патчей изображений к видеокубам ... 165
6.3.4. Извлечение видеоинформации ... 166
6.3.5. Аудиоэмбеддинги ... 167
6.3.6. Конвейер только для аудио: извлечение и инференс ... 169
6.4. Мультимодальный RAG: от PDF к изображениям, таблицам и кросс-модельному сравнению ... 174
Резюме ... 183

Глава 7. Эффективные и специализированные малые языковые модели ... 185

7.1. Эффективность малых моделей ... 186
7.2. Малые модели как агенты в системах, составленных из специализированных моделей ... 188
7.3. Классификация с использованием малых языковых моделей ... 190
7.3.1. Оценка качества классификации ... 190
7.3.2. Точность и F1-мера ... 193
7.3.3. Дообучение малых языковых моделей на наборе данных Financial PhraseBank ... 193
7.4. Адаптация Gemma 3 270M для выражения эмпатии и просоциального тона ... 200
7.5. Адаптация Gemma 3 270M для перевода с английского на испанский ... 210
7.6. Более широкие варианты использования и дополнительные модели ... 215
Резюме ... 218

Глава 8. Обучение и оценка больших языковых моделей ... 220

8.1. Глубокое погружение в гиперпараметры ... 221
8.1.1. Как параметры и гиперпараметры влияют на градиентный спуск ... 221
8.2. Настройка модели и оптимизация гиперпараметров ... 224
8.2.1. Отслеживание экспериментов ... 230
8.3. Экономное дообучение LLM ... 234
8.3.1. Низкоранговая адаптация ... 236
8.3.2. Декомпозиция весов при низкоранговой адаптации ... 239
8.3.3. Квантование ... 241
8.3.4. Эффективное дообучение квантованных больших языковых моделей с помощью QLoRA ... 245
8.3.5. Низкоранговая адаптация с учетом квантования ... 248
8.3.6. Декомпозиция матриц на низкоранговые и квантованные компоненты ... 250
8.3.7. Подводя итог: выбор подходящей стратегии экономного дообучения (PEFT) ... 251
Резюме ... 251

Глава 9 .Оптимизация и масштабирование больших языковых моделей 253

9.1. Оптимизация моделей 254
9.1.1 Прунинг моделей 254
9.1.2. Дистилляция моделей 256
9.2. Шардинг для оптимизации памяти 258
9.3. Оптимизация инференса 261
9.4. Оптимизация на уровне GPU: тайлинг, потоки и память 265
9.4.1. FlashAttention: масштабируемое внимание на основе техники тайлинга 269
9.5. Расширение контекстных окон большой длины 272
9.5.1. Ротационные эмбеддинги и их усовершенствования 272
9.5.2. Усовершенствования: YaRN, позиционная интерполяция и iRoPE 274
Резюме 277

Глава 10. Этика и ответственность в разработке больших языковых моделей 279

10.1. Выявление и понимание предвзятостей в больших языковых моделях 280
10.1.1. Выявление предвзятостей 280
10.1.2. Интерпретируемость моделей и предвзятость в искусственном интеллекте 282
10.2. Прозрачность и объяснимость больших языковых моделей 284
10.2.1. Использование Captum ля анализа поведения генеративных языковых моделей 285
10.2.2. Использование локальных интерпретируемых модельно-независимых объяснений для интерпретации предсказания модели 290
10.3. Ответственное использование больших языковых моделей 293
10.3.1. Индекс прозрачности фундаментальных моделей 295
10.4. Защита вашей языковой модели 296
10.4.1. Джейлбрейки и уязвимости жизненного цикла 303
10.4.2. Защиты модели от вредоносного использования 304
Резюме 307

Ссылки 309

Предметный указатель 312

Предисловие

Трансформеры и созданные на их основе большие языковые модели находятся в центре современного искусственного интеллекта. Они знаменуют собой один из тех моментов, когда элегантная теоретическая идея приводит к огромным реальным эффектам. Если специализированное оборудование — это тело современных вычислений, то трансформеры — это их разум. Именно они являются той частью, которая обучается, рассуждает и создает. Практически каждый наблюдаемый сегодня крупный прорыв в области ИИ (искусственный интеллект) — от интеллектуальной генерации кода до мгновенного перевода и диалоговых ассистентов — восходит к одной-единственной идее: механизму внимания (attention) и невероятной параллелизации, которую он открыл. Если вы работаете с ИИ сегодня, то свободное владение языком трансформеров больше не является опциональным — оно просто необходимо.

Но следить за развитием этой области — задача не из легких. Кажется, что каждые несколько недель появляется новая архитектура, метод промптинга или техника масштабирования. Даже экспертам бывает трудно уследить за тем, что действительно имеет значение. Именно поэтому книга "Трансформеры в действии" является такой ценной и своевременной. Она не ограничивается объяснением принципа работы трансформеров, а помогает постичь их концептуальную основу. Формирует ту самую интуицию, которая позволяет видеть эти модели не как загадочные "черные ящики", а как системы, которые вы можете анализировать, адаптировать и улучшать. Именно такое понимание Николь Кёнигштайн (Nicole Königstein) и воплощает в жизнь.

Николь обладает редким сочетанием глубоких теоретических знаний и реального практического опыта. Она руководила командами по разработке ИИ, проектировала количественные системы в сфере финансов и работала над крупномасштабными внедрениями, где точность и надежность имеют первостепенное значение. Как исследователь с докторской степенью в области ИИ, главный специалист и консультант по данным и руководитель отдела количественных исследований в ИИ, она соединяет два мира, которые не всегда легко сочетаются: четкость теории и праг-

матизм производства. Николь прекрасно понимает, что создание успешных систем ИИ означает нечто большее, чем просто овладение математическим аппаратом. Это еще и баланс между инновациями и ответственностью, между техническим совершенством и здравым смыслом.

Данная книга прекрасно отражает такой баланс. Первые главы посвящены основам — почему механизм внимания стал таким прорывом и как трансформеры изменили наше представление о моделировании последовательностей. Далее рассматривается область генеративных моделей, исследуя продвинутые методы промптинга, согласование предпочтений для обеспечения безопасности и такие технические приемы, как генерация с расширенным поиском, которые удерживают модели в рамках достоверности. В последующих главах обсуждаются проблемы производственного внедрения: от мультимодальных систем и эффективных малых языковых моделей до таких методов оптимизации, как PEFT и LoRA. Все это объединяется в заключительном обсуждении этики и ответственности — темы, имеющей существенное значение для всех, кто формирует будущее искусственного интеллекта.

Книге "Трансформеры в действии" удается нечто особенное — она строга, но не суха; глубока, но всегда ясна. Автор делает сложные идеи интуитивно понятными и дает читателям уверенность, которая позволяет перейти от простого использования инструментов ИИ к настоящему созиданию на их основе. В области, которая развивается быстрее чем когда-либо, эта книга служит спокойным и надежным проводником, который сделает вас не просто осведомленными, но и вдохновленными.

Луис Серрано (Luis Serrano), PhD, основатель и генеральный директор Serrano Academy, автор книги "Grokking Machine Learning"

Вступление

Начав в 2019 году впервые использовать трансформеры, я сразу же увлеклась ими. Два года спустя я создала собственную архитектуру глубокого обучения с использованием механизма внимания (attention). Эта работа была опубликована в журнале Springer Nature, и оказалось, что трансформеры действительно окажутся преобразующим прорывом в буквальном смысле слова. Больше всего меня поразила не их сложность, а их простота. Механизм, который открыл революцию трансформеров, — это не сложная математика. Он основан на фундаментальных принципах линейной алгебры: умножении матриц, нормализации с помощью softmax и объединении векторов с использованием взвешенных сумм.

Удивительно, что на основе простых скалярных произведений и вероятностей мы пришли к системам с миллиардами параметров, способным рассуждать, работая с текстом, изображениями, аудио и видео. В этом и заключается история трансформеров: один элегантный механизм, примененный в масштабе, изменил ландшафт искусственного интеллекта. Данная книга посвящена именно этой истории — от истоков трансформеров до того, как мы теперь пользуемся на практике большими языковыми моделями (LLM) и мультимодальными системами.

Элегантность заключается в том, как эти простые шаги организованы и объединены. Каждый токен проецируется в запросы, ключи и значения. Модель вычисляет скалярные произведения между запросами и ключами, чтобы определить релевантность, применяет softmax для преобразования этих оценок в вероятности и использует их для формирования взвешенных сумм значений.

Если задуматься, это не так уж сильно отличается от того, что происходит во время самой генерации текста. Когда модель предсказывает следующий токен, она снова применяет softmax для получения вероятностей, а затем производит выборку из них, чтобы решить, что будет дальше. Оба механизма опираются на базовую теорию вероятностей. Именно поэтому, чтобы понять трансформеры, вам не нужно быть математиком. Их основы доступны, и настоящее чудо заключается в том, какая мощь возникает из таких простых операций.

Темпы инноваций в рамках этой архитектуры захватывают дух. В 2017 году статья "Attention Is All You Need"[1] ("Внимание — это все, что вам нужно") впервые применила трансформеры к задачам машинного перевода. Модель BERT продемонстрировала мощь предобучения и дообучения. То, что начиналось с перевода, теперь выросло до больших языковых моделей с миллиардами параметров: ChatGPT вывел трансформеры в поле общественного внимания, а такие модели, как DeepSeek, выводят эффективность и масштабируемость на новые рубежи. Благодаря непрерывным инновациям, таким как FlashAttention, все эти операции умножения матриц стали быстрее и эффективнее.

На протяжении всей этой книги вы проследите как за эволюцией трансформеров, так и за моим личным опытом работы с ними в контексте больших языковых моделей, одновременно формируя свой собственный путь и понимая, как двигаться вперед в этой области. Книга начинается с основ механизма внимания, а затем прослеживает, как трансформеры эволюционировали в генеративные и мультимодальные системы, которые мы знаем сегодня. По мере продвижения рассматриваются вопросы эффективности, стратегии масштабирования, а также ответственности, сопряженной с развертыванием столь мощных моделей.

Я надеюсь, что по мере чтения книги вы увидите и красоту, лежащую в основе простоты, и те невероятные возможности, которые из нее вырастают.

[1] Авторами статьи являются исследователи из Google Brain и Google Research: Эшиш Васвани, Ноам Шазир, Ники Пармар, Якоб Ушкорейт, Ллион Джонс, Эйдан Гомес, Лукаш Кайзер и Илья Полосухин. — *Примеч. ред.*

Благодарности

Написание книги или построение карьеры, особенно в такой области, как искусственный интеллект, никогда не бывает делом одиночки, даже если бóльшая часть работы выполняется в тихие часы исследований, программирования и написания черновиков. Поэтому я хочу воспользоваться моментом, чтобы поблагодарить людей, которые поддерживали меня на ключевых этапах этого пути.

Прежде всего, я хочу поблагодарить Маркуса Оэмана (Markus Oehmann), который поощрял и поддерживал меня в самом начале моего пути в области изучения искусственного интеллекта. Хотя наши жизненные пути в итоге разошлись, его поддержка в те первые годы дала мне уверенность в правильности выбора исследований как своего истинного призвания. За это он заслуживает быть здесь упомянутым.

Я глубоко благодарна профессору доктору Кристофу Денцлеру (Christoph Denzler), который поверил в меня на раннем этапе, при необходимости шел на уступки вопреки правилам и поддержал мою первую диссертацию. Именно эта диссертация в конечном итоге поставила меня на путь, приведший к нынешней карьере. Его поддержка заложила фундамент для моей первой опубликованной статьи и для всего за ней последовавшего.

Спасибо всем рецензентам: Элу Пезевски (Al Pezewski), Александару Бабичу (Aleksandar Babic), Али Шакибе (Ali Shakiba), Анимику Айчу (Animikh Aich), Анкиту Вирмани (Ankit Virmani), Антону Петрову (Anton Petrov), Анупу Парикху (Anup Parikh), Артуро Гайгелю (Arturo Geigel), Бруно Куриолу (Bruno Couriol), Чуньсю Тангу (Chunxu Tang), Дэвиду Каррану (David Curran), Дхирендре Чоудхари (Dhirendra Choudhary), Фернандо Байону (Fernando Bayon), Джорджу Гейнсу (George Gaines), Хобсону Лейну (Hobson Lane), Якубу Лангру (Jakub Langr), Якубу Моравски (Jakub Morawski), Джеймсу Лю (James Liu), Джереми Чену (Jeremy Chen), Джереми Цайднеру (Jeremy Zeidner), Джону Уильямсу (John Williams), Марку Лю (Mark Liu), Мартину Хедигеру (Martin Hediger), Мэттью Шарпу (Matthew Sharp), Морин Мецгер (Maureen Metzger), Навину Ачьюте (Naveen Achyuta), Олене Сокол (Olena Sokol), Полу Силистяну (Paul Silisteanu), Филиппу Диттриху (Philipp Dittrich), Прадипу Сарасвати (Pradeep Saraswati), Приянке Нилакришнан (Priyanka

Neelakrishnan), Раджу Кумару (Raj Kumar), Равешу Шарме (Ravesh Sharma), Ричарду Майнзену (Richard Meinsen), Россу Тернеру (Ross Turner), Руи Лю (Rui Liu), Саммиту Сонаване (Sameet Sonawane), Сидхарту Сомантану (Sidharth Somanathan), Симону Чшёке (Simon Tschöke), Симоне Де Бонису (Simone De Bonis), Симоне Сгуацце (Simone Sguazza), Шри Раму Мачарле (Sri Ram Macharla), Субханкару Раю (Subhankar Ray), Суканье Конатам (Sukanya Konatam), Тони Холдройду (Tony Holdroyd), Вахиду Мирджалили (Vahid Mirjalili), Видхье Винай (Vidhya Vinay), Виноту Нагешварану (Vinoth Nageshwaran), Вибхавредди Каммиредди Чангалредди (Vybhavreddy Kammireddy Changalreddy), Вальтеру Александру Мате Лопесу (Walter Alexander Mata López), Вэй-Мэн Ли (Wei-Meng Lee). Ваши предложения и замечания помогли сделать эту книгу лучше.

Я хочу поблагодарить весь коллектив издательства Manning, который помогал мне в работе над этой книгой, и особенно Марину Майклс (Marina Michaels) за ее внимание к деталям на протяжении всего процесса разработки, а также всю команду, работающую "за кулисами" производства. Также выражаю благодарность техническим редакторам: Дэвиду Пачеко Аснару (David Pacheco Aznar), вычислительному математику и специалисту по данным и Майку Эрлихсону (Mike Erlihson), доктору математики из Техниона (Technion), а также Карстену Стробеку (Karsten Strøbæk), техническому корректору издательства Manning, который проверил и протестировал весь код.

Наконец, я хочу поблагодарить всех коллег, студентов и единомышленников, которые вдохновляли меня своими дискуссиями, сотрудничеством и общей страстью к искусственному интеллекту. Каждый из вас внес свой вклад, большой или малый, в формирование идей и взглядов, отраженных в этой книге.

О книге

Книга "Трансформеры в действии" — это всестороннее руководство по пониманию и применению моделей-трансформеров в области обработки естественного языка и мультимодальных систем. Эти модели являются фундаментом таких современных систем искусственного интеллекта, как ChatGPT и Gemini. Цель книги — предоставить вам прочную основу для использования этих моделей в собственных проектах: начиная с ключевых концепций трансформеров и переходя к таким практическим и более продвинутым приложениям, как мультимодальные системы поиска информации.

Вы узнаете, почему трансформеры устроены именно так и как они работают, что даст вам как теоретическое понимание, так и практические навыки для их эффективного использования. По ходу изложения вы увидите, когда целесообразнее применять малые языковые модели (SLM), а когда более оправданными являются архитектурные решения, такие как модели только с энкодером или только с декодером.

Кому следует прочесть эту книгу

Эта книга предназначена для специалистов по работе с данными и инженеров машинного обучения, которые хотят научиться создавать и применять модели на основе трансформеров для решения задач в области обработки естественного языка и мультимодальных задач. Цель книги — снабдить вас необходимыми знаниями для формирования прочного фундамента, помогающего с уверенностью переходить к более продвинутым моделям и подходам.

Как организована книга: дорожная карта

Книга разделена на три части, охватывающие 10 глав.

- *Часть 1 "Основы современных моделей-трансформеров"* объясняет основы моделей-трансформеров.
  - *Глава 1 "Зачем нужны трансформеры"* знакомит с предпосылками появления трансформеров, объясняет, почему более ранние модели для работы с последовательностями сталкивались с трудностями, и показывает, как механизм внимания позволяет преодолеть возникшие ограничения.
  - *Глава 2 "Более пристальный взгляд на трансформеры"* посвящена исследованию полной архитектуры трансформеров, включая стеки энкодера и декодера, позиционное кодирование, слои внимания и полносвязные сети.
- *Часть 2 "Генеративные трансформеры"* посвящена генеративным трансформерам.
  - *Глава 3 "Семейства моделей и варианты архитектуры"* рассматривает основные архитектурные варианты трансформеров, включая модели только с декодером, только с энкодером, эмбеддинг-модели и архитектуры со смесью экспертов.
  - *Глава 4 "Стратегии генерации текста и методы промптинга"* представляет стратегии генерации текста и технические приемы промптинга, включая жадный поиск и поиск по лучу, top-k-семплирование и семплирование по ядру, семплирование с учетом температуры, а также шаблоны промптов — от промптинга нулевого примера и до "дерева рассуждений".
  - *Глава 5 "Согласование предпочтений и генерация с дополненным поиском"* посвящена согласованию предпочтений и генерации с дополненным поиском (retrieval-augmented generation, RAG). В ней рассматриваются обучение, подкрепленное обратной связью от человека, прямая оптимизация предпочтений и надежные методы оценки, а также демонстрируется, как строить обоснованные системы с использованием RAG.
- *Часть 3 "Специализированные модели"* посвящена специализированным и продвинутым моделям.
  - *Глава 6 "Мультимодальные модели"* знакомит с мультимодальными моделями, объединяющими текст с изображениями, аудио и видео. В ней объясняются специфичные для модальностей методы токенизации, визуальные и аудиоэмбеддинги, а также мультимодальная RAG для работы со сложными документами.
  - *Глава 7 "Эффективные и специализированные малые языковые модели"* посвящена малым языковым моделям (SLM). Вы узнаете, как SLM могут выступать в роли эффективных специалистов, и рассмотрите практические примеры их применения для задач классификации, перевода и дообучения на

эмпатию и просоциальную тональность. В этой главе также показано, как SLM могут выполнять роль агентов в более крупных рабочих процессах.

- *Глава 8 "Обучение и оценка больших языковых моделей"* посвящена обучению и оценке больших языковых моделей, включая гиперпараметры, отслеживание экспериментов, параметрически эффективное дообучение и такие технические приемы квантования, как QLoRA.
- *Глава 9 "Оптимизация и масштабирование больших языковых моделей"* посвящена оптимизации и масштабированию. В ней объясняются методы прунинга, дистилляции, шардинга, оптимизации инференса, повышения эффективности на уровне GPU, а также рассматриваются FlashAttention и расширения для работы с длинным контекстом.
- *Глава 10 "Этика и ответственность в разработке больших языковых моделей"* посвящена вопросам этики и ответственности в работе с искусственным интеллектом. В ней рассматриваются обнаружение предвзятости, инструменты обеспечения прозрачности и объяснимости, ответственное развертывание моделей, а также защитные механизмы против джейлбрейков и злоупотреблений.

Книгу можно читать от начала до конца или начать с *части 1*, чтобы освоить основы, а затем перейти к темам, наиболее актуальным для вашей работы, в *частях 2* и *3*.

О программном коде

Книга задумана так, чтобы дать не только прочную теоретическую базу, но и позволить приобрести практические навыки. Поэтому в ней содержится множество примеров исходного кода — как в виде нумерованных листингов, так и в тексте. В обоих случаях исходный код оформлен моноширинным шрифтом, отделяющим его от обычного текста. Иногда код также выделяется полужирным начертанием, чтобы подчеркнуть изменения по сравнению с предыдущими шагами в главе — например, когда новая функция добавляется в уже существующую строку кода.

Во многих случаях исходный код был переформатирован: добавлены разрывы строк и изменены отступы, чтобы уместить его в доступное пространство страницы книги. В редких случаях, когда этого оказалось недостаточно, в листингах используются маркеры продолжения строки (➥). Кроме того, комментарии в исходном коде часто были удалены из листингов в тех случаях, когда код описывался в тексте. Многие листинги сопровождаются аннотациями, выделяющими важные концепции. Я рекомендую использовать блокноты Jupyter напрямую, а не копировать код из печатных листингов, поскольку исходный код был переформатирован. Это позволит вам легче использовать их в качестве основы для создания собственных приложений.

Весь исходный код доступен в специализированном репозитории на GitHub по адресу: **https://github.com/Nicolepcx/Transformers-in-Action**. Репозиторий организован по главам и содержит блокноты Jupyter, которые делают примеры интерактивными и удобными для расширения. Каждый блокнот снабжен кнопкой **Open in Colab** (Открыть в Colab), позволяющей запускать код напрямую. Некоторые примеры из-за требований к памяти могут запрашивать Colab Pro или аналогичного доступа к графическому процессору (GPU).

Вы можете получить исполняемые фрагменты кода из онлайн-версии этой книги на платформе liveBook по адресу: **https://livebook.manning.com/book/transformers-in-action**. Полный код примеров из книги доступен для скачивания на веб-сайте Manning по ссылке: **https://livebook.manning.com/books/transformers-in-action**.

Об авторе



Николь Кёнигштайн (Nicole Königstein) — основатель, генеральный директор и соруководитель по вопросам искусственного интеллекта (Co-Chief AI Officer) глубоко технологичной компании, специализирующейся на системах больших языковых моделей (LLM) и сложных мультиагентных архитектурах. Она является всемирно признанным экспертом в области больших языковых моделей и агентных архитектур, с особым акцентом на их трансформационном применении в количественных финансах. Николь выступала в качестве внешнего эксперта Европейской комиссии по вопросам финансирования исследований в области больших языковых моделей, а также в Международной организации комиссий по ценным бумагам (IOSCO), консультируя по вопросам внедрения генеративного ИИ в регулируемых отраслях. Она также входит в консультативные советы QuantMinds International и QuantStrats — двух крупнейших и наиболее влиятельных мировых конференций, посвященных финтеху, искусственному интеллекту и количественным стратегиям. Николь неоднократно выступала в качестве приглашенного лектора, докладчика на ключевых сессиях и ведущего технических воркшопов в разных странах мира — в том числе в Оксфордском университете, Корнеллском университете, Bank of America, J.P. Morgan и Международной школе финансов, а также на таких международных мероприятиях, как European Women in Technology, ACM Tech Talk и ODSC Europe.

Николь является преподавателем Института искусственного интеллекта в финансах (Artificial Intelligence in Finance Institute) в Нью-Йорке, где она проводит занятия по программированию на Python, машинному обучению, глубокому обучению, большим языковым моделям и агентному ИИ. Она — автор книги "Math for Machine Learning" (издательство Manning Publications), а ее следующие книги "Transformers: The Definitive Guide" и "AI Agents: The Definitive Guide" будут опубликованы издательством O'Reilly Media.

Об изображении на обложке

Рисунок на обложке книги "Трансформеры в действии", озаглавленный "L'étudiant en droit" ("Студент-правовед"), взят из книги Луи Кюрмера, изданной в 1841 году, где каждая иллюстрация выполнена вручную с тонкой проработкой деталей и раскрашена от руки.

В те времена по одежде было легко определить, где человек живет, чем занимается и какое положение занимает в обществе. Издательство Manning подчеркивает изобретательность и инициативность компьютерного бизнеса, создавая обложки книг, основанные на богатом разнообразии региональной культуры многовековой давности, которые оживают благодаря иллюстрациям из таких коллекций, как эта.

Часть 1

Основы современных моделей-трансформеров

С момента своего появления на 31-й Конференции по нейронным системам обработки информации (NIPS 2017) трансформеры стали весьма популярной темой исследований. *Трансформер* представляет собой тип модели машинного обучения, которая, в отличие от своих предшественников, способна понимать контекст слов в предложении, одновременно сосредотачиваясь на разных частях предложения. Именно этот факт делает трансформеры высокоэффективными средствами решения широкого спектра задач обработки естественного языка (NLP), включая классификацию текста, его суммирование и генерацию. Они широко используются во многих приложениях, подчеркивая важность глубокого понимания этих моделей.

Хотя вам, возможно, не терпится приступить к делу и научиться создавать собственную модель ChatGPT, способную отвечать на любые ваши вопросы и даже строить прогнозы, крайне важно сначала усвоить принципы работы трансформерной модели. Дело в том, что для достижения успеха в данной области и успешного применения более продвинутых моделей необходимо заложить под свою работу прочную основу.

Представьте себе, что освоение применения трансформеров при решении ваших конкретных задач похоже на выращивание сада. Прежде чем выращивать изысканные и экзотические растения, нужно подготовить почву, т. е. заложить под это дело прочную основу. Именно для этого и предназначена первая часть данной этой книги, предоставляя вам необходимые базовые знания для закладки прочного фундамента. Первая часть дает вам более глубокое понимание основных концепций и базовых математических основ трансформеров, объясняет, почему они устроены именно так и как они работают. И с этим знанием данная часть готовит вас к более сложным и современным моделям, а также к продвинутым подходам, изложенным во второй и третьей частях книги.

1 Зачем нужны трансформеры

Темы этой главы

- Как трансформеры революционизировали обработку естественного языка (NLP)
- Механизмs внимания — ключевой архитектурный компонент трансформеров
- Как использовать трансформеры
- Когда и зачем применять трансформеры

Область машинного обучения (ML, Machine Learning), и в особенности обработки естественного языка (NLP, Natural Language Processing), претерпела революционные изменения благодаря изобретению нового класса нейронных сетей, называемых трансформерами. Эти модели, отличающиеся своей способностью понимать и генерировать естественный язык, являются основой широко используемых приложений генеративного искусственного интеллекта, таких как ChatGPT от OpenAI и Claude от Anthropic.

Трансформеры, а также такие их производные, как большие языковые модели (LLM, Large Language Model), используют уникальный архитектурный подход, включающий инновационный компонент, известный как "механизм внимания". Данный механизм позволяет модели концентрироваться в различной степени на отдельных сегментах входных данных, улучшая ее способность обрабатывать и осмысливать сложные последовательные данные. Эта особенность критически важна для обработки естественного языка в LLM, а также применима и к широкому спектру использования трансформеров при обработке аудиопотоков, изображений и видео.

Для начала сравним трансформеры с их предшественниками — моделями с долгой краткосрочной памятью (LSTM, Long Short-Term Memory), а затем рассмотрим каждый компонент трансформера более подробно.

1.1. Трансформерный прорыв

Человеческий мозг обладает поразительной способностью воспринимать большое количество данных и быстро устанавливать внутри них связи между соответствующими сегментами. Моделям машинного обучения долгое время не удавалось справиться с этой базовой задачей, отчасти потому, что им сложно было выявлять ключевые последовательности в большом потоке данных.

Чтобы преодолеть эти ограничения, Васвани (Vaswani) и др. [1][1] представили трансформер — архитектуру глубокого обучения, построенную вокруг механизма внимания. Позволяя каждому элементу в последовательности непосредственно взаимодействовать с любым другим элементом, трансформеры не только ускоряют обработку, но и повышают точность решения задач, связанных последовательностями. После публикации эпохальной статьи "Внимание — все, что вам нужно" ("Attention Is All You Need") трансформеры ознаменовали собой существенный отход от таких ранее популярных рекуррентных нейронных сетевых архитектур, как LSTM. До появления трансформеров основным выбором для обработки последовательных данных, включая естественный язык, была долгая краткосрочная память (LSTM), являющаяся особым видом рекуррентной нейронной сети (RNN, Recurrent Neural Network), которая, исходя из названия, обрабатывает последовательности путем итерации элементов и сохраняет своего рода "память" о ранее обработанной информации. Давайте вкратце рассмотрим, как модели машинного обучения, предшествующие трансформерам, решали такие задачи, как перевод с английского на французский.

1.1.1. Перевод до появления трансформеров

Предположим, нам нужно перевести предложение "I don't speak French" ("Я не говорю по-французски") на французский язык: "Je ne parle pas français". Чтобы перевести предложение с использованием RNN, архитектурный компонент этой нейронной сети, называемый энкодером (от англ. *encoder* — кодировщик), последовательно обрабатывал бы каждое слово предложения "I don't speak French", обновляя свое состояние с каждым словом и формируя так называемый контекстный вектор английского предложения. Этот контекстный вектор заключает в себе семантику исходного предложения и служит мостиком к следующему компоненту RNN — декодеру (от англ. *decoder* — декодировщик).

Затем декодер использует этот контекстный вектор для порождения выходной последовательности "Je ne parle pas français", и опять пословно, используя внутреннее (рекуррентное) состояние для запоминания ранее сгенерированных слов. На протяжении всего процесса рекуррентная природа архитектуры позволяет каждому этапу основываться на предыдущих, создавая связный перевод. Обработка на основе последовательностей позволяет RNN улавливать временные зависимости внутри

[1] В конце книги в *разделе "Ссылки"* представлены источники информации. — *Примеч. ред.*

предложения, но при этом также может привести к трудностям в обработке дальних зависимостей между словами. Данный процесс показан на рис. 1.1.

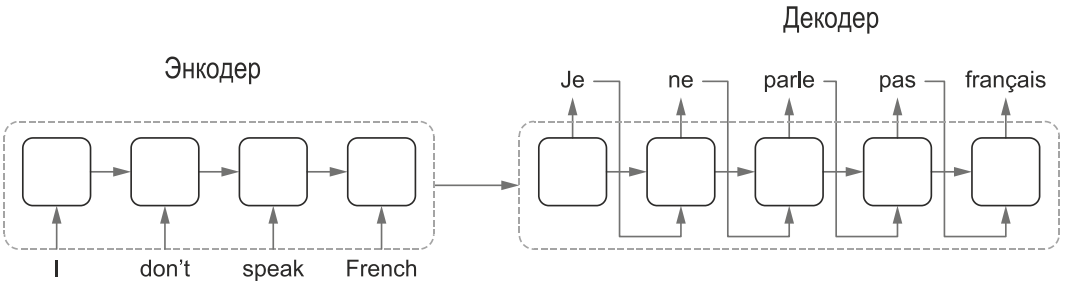

Рис. 1.1. Общий обзор потока последовательных данных в RNN. Энкодер в RNN обрабатывает входное предложение пословно, обновляя свое внутреннее состояние с каждым новым словом. Итогом становится финальный контекстный вектор, представляющий все предложение целиком. Затем декодер использует этот контекстный вектор для генерации выходного предложения, опять же пословно, используя свое внутреннее (рекуррентное) состояние для запоминания ранее сгенерированных слов

На рис. 1.1 показано, что "рекуррентное состояние" обновляется для каждого элемента выходной последовательности, фиксируя контекстную информацию о тех частях последовательности, которые уже были обработаны. Но у RNN имеются недостатки: эти сети плохо справляются с длинными последовательностями, что снижает эффективность улавливания в данных долгосрочных зависимостей.

1.1.2. В чем отличие трансформеров?

Трансформеры положили начало совершенно иной стратегии. В отличие от RNN, трансформеры способны при обработке каждой отдельной взятой части одновременно учитывать сразу несколько частей входной последовательности. Вместо поэлементной обработки последовательности и передачи вперед единственного рекуррентного состояния, модель трансформера рассчитывает "оценку внимания" для каждого элемента относительно всех остальных элементов последовательности.

Этот фундаментальный переход делает трансформеры гораздо более распараллеливаемыми и, следовательно, более эффективными. Он также значительно увеличивает способность модели выявлять в данных тонкие закономерности и долговременные зависимости. В отличие от традиционных моделей типа "последовательность-к-последовательности", трансформеры для создания более взаимосвязанного представления всей последовательности используют механизм внимания, что повышает способность модели выдавать точные предсказания. Кроме того, архитектура трансформеров позволяет одновременно обрабатывать все элементы последовательности, существенно сокращая время вычислений.

Давайте в качестве примера вновь рассмотрим задачу перевода предложения "I don't speak French" ("Я не говорю по-французски") на "Je ne parle pas français". В отличие от RNN, энкодер модели трансформера извлекает признаки из входного предложения "I don't speak French". Затем эти признаки обрабатываются трансформером с помощью механизма внимания для расчета оценок внимания для каждого элемента. Эти оценки эффективно отражают контекстные отношения между каждым словом и остальной частью предложения. Декодер, являющийся еще одной составляющей модели трансформера, использует эти оценки внимания и извлеченные признаки для генерации переведенного предложения "Je ne parle pas français". Схематично процесс перевода с использованием модели трансформера показан на рис. 1.2.

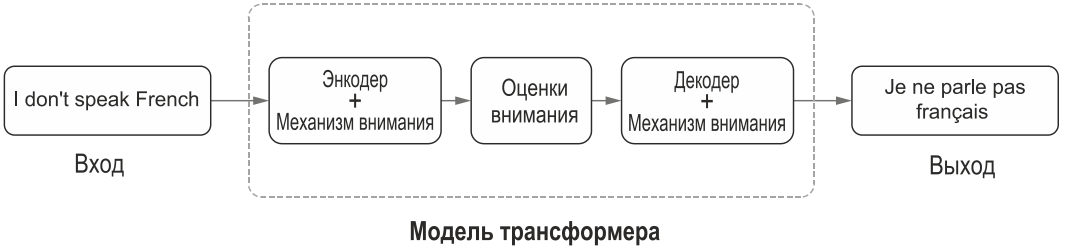

Рис. 1.2. Схематичное представление потока последовательных данных в архитектуре трансформера. Слева направо входное предложение поступает в трансформер, где оно сначала обрабатывается энкодером совместно с механизмом внимания. Результаты, представленные в виде оценок внимания (также называемых контекстными представлениями), передаются декодеру, имеющему свой собственный механизм внимания. На выходе получается переведенное предложение

На рис. 1.2 показано, что механизм внимания в модели трансформера позволяет не обновлять "рекуррентное состояние" для каждого элемента, как это делают RNN, а параллельно обрабатывать всю последовательность. Такая конструкция позволяет трансформерам эффективно улавливать контекстные взаимоотношения между всеми частями последовательности, преодолевая проблемы, испытываемые такими традиционными моделями, как RNN, которые сталкиваются с затруднениями при работе с длинными последовательностями. Результат — более эффективная и мощная модель, способная справляться с имеющимися в данных сложными паттернами и долгосрочными зависимостями.

1.1.3. Устройство механизма внимания

Используемый в трансформерах *механизм внимания* позволяет модели при вычислении представления последовательности присваивать различным позициям разные веса. По сути, в рамках архитектуры энкодер-декодер механизм внимания оценивает релевантность различных входных векторов и назначает наибольшие веса наиболее важным из них. Это принципиально отличается от рекуррентных нейронных

сетей (RNN), например, от моделей LSTM, ведущих поэлементную обработку входных последовательностей.

Давайте в качестве пояснения более подробно рассмотрим наш пример перевода с английского на французский. Идеальная модель перевода должна понимать каждое слово в контексте других слов в предложении. В данном случае перевод слова "speak" в "parle" зависит от окружающих его слов "I", "don't" и "French". Это проиллюстрировано на рис. 1.3, где связь показана толщиной линий, представляющих оценки внимания, назначаемые моделью, и демонстрируя, как слова связаны друг с другом.



Рис. 1.3. Пример перевода, сделанного трансформерной моделью с механизмом внимания. Толщина и цвет линий демонстрируют оценки внимания, назначенные моделью

Данный механизм внимания является ключевым компонентом трансформерной модели, наделяя ее способностью обработки последовательностей со сложными взаимосвязями между элементами. Но даже наше простое предложение требует, чтобы модель учитывала с помощью механизма внимания взаимосвязи между различными словами в последовательности. Более того, по мере усложнения последовательностей и повышения изощренности задач становится все более очевидной важность таких продвинутых стратегий внимания, как многоголовое внимание (MHA, Multi-Head Attention). Давайте рассмотрим многоголовое внимание на примере более сложной задачи обработки естественного языка: классификации тональности текста.

1.1.4. Возможности многоголового внимания

Модель трансформера вводит мощное расширение механизма внимания, известное как *многоголовое внимание*. Оно расширяет способность модели улавливать множественные взаимосвязи внутри последовательности. Данный подход позволяет модели одновременно фокусироваться на различных позициях входных данных, выделяя различные аспекты информации. Это означает, что модель может поддерживать несколько "точек зрения", чтобы лучше понимать имеющиеся в данных сложные закономерности.

Для демонстрации преимущества многоголового внимания рассмотрим предложение "The movie was not bad" ("Фильм был неплохим"). В данном случае многоголовое внимание трансформера позволяет модели одновременно анализировать взаимодействие "не" и "плохим", что приводит к пониманию общего положительного тона высказывания. Взаимодействие слов в предложении показано на рис. 1.4.



Рис. 1.4. Пример различных взаимосвязей в предложении

В отличие от этого, такие рекуррентные модели, как LSTM, зачастую испытывают трудности с идентификацией в предложении подобных длинных зависимостей, что приводит к потенциальным проблемам в распознавании таких вот тонких нюансов. Трансформеры, обладая архитектурой, основанной на внимании, обходят эту проблему, эффективно улавливая сложные взаимосвязи в текстовых данных.

Короче говоря, многоголовое внимание позволяет трансформерам одновременно понимать различные аспекты входных данных, совершая существенный скачок по сравнению с такими предыдущими моделями, как LSTM. Используя данный прием, трансформер смог достичь показателя 41 по метрике BLEU (Bilingual Evaluation Understudy)[2]. Это очень высокий результат, свидетельствующий о том, что сгенерированный трансформером перевод очень близок к эталонному тексту. Еще интереснее то, что для достижения такого высокого результата первой модели трансформера понадобилось всего лишь 3,5 дня обучения, что составляло ничтожную долю от времени, требуемого для достижения сопоставимого высокого результата в той же задаче перевода с английского на французский предыдущим передовым сетям, таким как LSTM.

Более того, в сравнении с другими моделями на основе RNN (например, LSTM) успех трансформеров и больших языковых моделей (LLM) становится еще более заметным с точки зрения их революционного влияния на область обработки естественного языка (NLP). Хотя LSTM представляли значительные улучшения для обработки последовательностей, скорость завоевания LLM и трансформерами этой области, а также темп непрерывной разработки современных моделей не имеют аналогов. Благодаря стремительному и прорывному успеху трансформеров спустя всего лишь пять лет после появления первой модели трансформера стало возможным создание таких передовых языковых моделей, как ChatGPT.

1.2. Использование трансформеров

Начало работы с трансформерами значительно упрощается благодаря библиотеке Transformers от платформы машинного обучения Hugging Face. Эта библиотека

[2] BLEU (Bilingual Evaluation Understudy) — это алгоритм автоматической оценки качества машинного перевода, который измеряет степень лексической близости сгенерированного текста к эталонным человеческим переводам. — *Примеч. ред.*

предоставляет предварительно обученные модели, облегчающие решения широкого спектра задач: от перевода текста и его генерации до классификации и анализа тональности.

Термин "предварительно обученные" относится к моделям, которые были заранее обучены на обширных наборах данных, содержащих обычно миллионы документов, охватывающих широкий спектр тем. Такое обучение позволяет моделям усваивать тонкие закономерности языка, включая синтаксис, семантику и контекст. Использование таких предварительно обученных моделей фактически дает возможность воспользоваться накопленными ими знаниями, избавляя вас от необходимости проведения самостоятельного обучения модели с нуля. Такой подход позволяет сэкономить существенное количество вычислительных ресурсов и времени, поскольку теперь вам нужно только лишь дообучить модель под решение вашей конкретной задачи. Например, для проведения анализа тональности текста можно воспользоваться размеченным набором данных, состоящим из пар предложений и меток "нейтрально", "негативно" или "позитивно".

При использовании трансформеров следует учитывать и такой фактор, как требуемая вычислительная мощность. Трансформеры, учитывая их сложность, могут быть весьма ресурсоемкими. И здесь в игру уже вступают графические процессоры (GPU, Graphics Processing Unit), предназначенные для одновременного выполнения множества операций, что делает их идеально приспособленными для матричных вычислений и параллельной обработки данных, характерных для обучения и использования трансформеров. Даже при отсутствии у вас высокопроизводительного GPU доступ к такому GPU предоставляется многими облачные сервисами, например, Google Colab, что делает использование трансформеров доступным широкой аудитории.

Таким образом, получив через библиотеку Transformers от Hugging Face доступ к предварительно обученным моделям, и вычислительные мощности, предоставляемые облачными сервисами, вы будете полностью подготовлены к тому, чтобы воспользоваться возможностями трансформеров для решения ваших собственных задач по обработке естественного языка.

1.3. Когда и зачем использовать трансформеры

Трансформеры стали неотъемлемой частью современного ландшафта машинного обучения. Они предлагают беспрецедентные возможности в области обработки естественного языка, но их потенциал выходит далеко за рамки этой сферы. Трансформеры продемонстрировали многообещающие результаты в таких областях, как компьютерное зрение и распознавание речи, что делает их в будущем наиболее универсальной архитектурой машинного обучения.

Но настоящая ценность трансформеров заключается в их доступности. Библиотека Transformers от Hugging Face содержит широкий спектр предварительно обученных моделей, облегчающих начало работы специалистам-практикам. В отличие от ар-

хитектур на основе LSTM, которым часто требуется обучения с нуля, трансформеры через Hugging Face сразу доступны в предварительно обученном виде, позволяя существенно экономить время и вычислительные ресурсы.

Это преимущество становится еще весомее благодаря активному сообществу пользователей библиотеки Transformers от Hugging Face. Имея открытый исходный код, библиотека непрерывно развивается за счет вклада и улучшения, вносимых энтузиастами машинного обучения со всего мира. Поэтому дообучить готовую модель трансформера — например, для анализа тональности текста — можно за считаные часы.

Кроме того, сферу применения трансформеров расширило развитие методов обучения без примеров (zero-shot) и с малым количеством примеров (few-shot). Обучение без примеров означает способность модели справляться с задачами, для которых она специально не была обучена. По сути, она может понимать и выполнять новые задачи, опираясь на свою широкую предварительную подготовку. Обучение с малым количеством примеров подразумевает, что модель способна быстро научиться выполнению новых задач после тренировки на весьма скромном количестве данных, относящихся к этой задаче. Этот метод использует преимущества уже накопленных моделью знаний, полученных в ходе ее масштабного предварительного обучения. В сочетании с активностью сообщества эти передовые методы делают трансформеры привлекательным выбором для широкого круга задач в области обработки естественного языка (NLP).

И все же, при весьма впечатляющих возможностях этих разнообразных моделей у них имеется ряд ограничений, особенно если рассматривать так называемые модели с миллиардами параметров, которыми обозначается масштаб и число настраиваемых параметров или "весов" этих моделей трансформеров. Например, ранние модели трансформеров имели 110 миллионов параметров, но прогресс в этой области привел к созданию еще более крупных моделей, некоторые из которых обладают сотнями миллиардов параметров.

Модели с миллиардами параметров демонстрируют впечатляющую способность генерировать связные и контекстно-релевантные ответы в диалоге. Но, несмотря на их огромный масштаб и выдающиеся показатели производительности, у этих моделей также имеются ограничения. В определенных специализированных сферах, таких как финансы или здравоохранение, где решающее значение имеет специфический контекст предметной области, их практическая применимость может снижаться. Внедрение этих моделей, особенно в системах реального времени, которым необходимы быстрые и точные результаты, из-за высоких требований к вычислительной мощности и памяти может затрудняться.

Тем не менее потенциал данных разновидностей моделей-трансформеров нельзя недооценивать. Они раздвигают границы возможного в понимании и генерации естественного языка, классификации текста, переводе и многом другом. Но, как и в любом инструменте, ключ кроется в сбалансированном подходе, ориентированном на конкретные задачи. По мере продвижения вперед мы исследуем этот баланс бо-

лее подробно, сосредоточившись на том, как использовать сильные стороны различных типов моделей с одновременным учетом их специфических ограничений.

1.4. От трансформеров к LLM: устойчивая модель

Архитектура трансформера является фундаментальной основой современных передовых больших языковых моделей (LLM). По своей сути эти мощные модели представляют собой нейронные сети на основе трансформеров, прошедшие интенсивное обучение на огромных массивах текстовых данных, что позволяет им превосходно справляться с широким спектром задач обработки естественного языка. Несмотря на многочисленные усовершенствования и уточненные варианты, появившиеся с течением времени, все современные передовые большие языковые модели наследуют, по существу, архитектуру и ключевые механизмы, впервые представленные в знаковой статье 2017 года "Attention Is All You Need" ("Внимание — это все, что вам нужно"). Хотя с тех пор появилось множество последующих инноваций, включая специализированные варианты энкодер-декодер, архитектуры декодеров, оптимизированные для генерации языка, а также такие усовершенствования механизмов внимания, как многоголовое внимание, разреженное внимание и улучшение позиционного кодирования, основные принципы архитектуры остались практически неизменными. В частности, определять сущность трансформеров и их возможности по-прежнему продолжает использование механизма внимания, являющегося основным средством улавливания контекстных взаимосвязей внутри данных.

Именно этими унаследованными особенностями и обуславливается удивительная универсальность трансформеров. Например, модели типа энкодер-декодер преуспевают в задачах перевода, где критически важно понимание контекста из входных последовательностей, тогда как архитектуры, использующие только декодер, получили широкое распространение в таких генеративных задачах, как завершение текста или диалоговый искусственный интеллект. Точно так же усовершенствования методов позиционного кодирования и механизмов внимания значительно расширили способность модели обрабатывать более длинные последовательности и более сложные контексты, но базовая концепция все же остается неизменной.

По сути исходная статья о трансформерах предложила схему, которая оказалась чрезвычайно прочной и адаптируемой. Современные большие языковые модели строятся на этой схеме, масштабируя ее до беспрецедентных размеров и совершенствуя стратегии обучения с помощью таких методов, как контролируемая тонкая настройка, неконтролируемое предварительное обучение и обучение, подкрепленное обратной связью с человеком. Таким образом, понимание базовой архитектуры трансформера крайне важно не только из-за его исторического значения, но и потому, что оно продолжает активно влиять на формирование архитектуры и возможностей современных генеративных систем искусственного интеллекта.

Резюме

- Модель трансформера использует механизмы внимания и многоголового внимания. Применяя эти инструменты, она эффективно ориентируется в различных частях предложения, выделяя, то есть уделяя больше внимания тем словам, которые играют ключевую роль в формировании общего смысла и понимания контекста.
- Механизм внимания позволяет модели сосредоточиться на ключевых частях входных данных и выделить наиболее важную информацию.
- Компонент многоголового внимания в модели трансформера помогает выявлять в последовательности многочисленные связи между словами, что сделало ее одной из самых широко используемых моделей в области обработки естественного языка (NLP).
- Трансформеры продемонстрировали выдающиеся результаты в решении задач обработки естественного языка благодаря своей способности справляться в последовательных данных с долговременными зависимостями.
- Они произвели революцию в этой области, сделав возможным обучение модели трансдукции не за недели или месяцы, а за считаные дни, при этом превосходя по качеству передовые нейросетевые архитектуры.
- Такие передовые методы, как обучение без примеров (zero-shot) или с малым количеством примеров (few-shot), позволяют крупным моделям делать выводы и обобщать информацию о новых задачах на основе уже имеющейся подготовки, без необходимости явного переобучения, что обеспечивает более эффективное использование ресурсов и времени.
- Несмотря на впечатляющие возможности, чрезвычайно крупные языковые модели не лишены ограничений. Одной из потенциальных проблем является снижение эффективности в некоторых специализированных областях, в числе которых можно выделить финансы и здравоохранение. Кроме того, их внедрение может затрудняться по причине высоких требований к вычислительным ресурсам и памяти. Поэтому выбор наиболее подходящей модели требует тщательного анализа конкретной задачи и сбалансированности сложности модели и ее практической применимости.

2 Более пристальный взгляд на трансформеры

Темы этой главы

- Моделирование последовательностей до появления трансформеров
- Ключевые компоненты модели-трансформера
- Механизм внимания и его разновидности
- Роль трансформеров в стабилизации распространения градиентов

Тем, кому уже приходилось пользоваться инструментами на основе трансформеров, например ChatGPT, могут на собственном опыте сформировать представление о степени эффективности интерпретации и генерации естественного языка при использовании больших языковых моделей (LLM). Но для успешного применения этих моделей к собственным задачам одного лишь импорта готовых конвейеров недостаточно. Для тонкой настройки LLM, устранения неожиданных проблем с производительностью, оптимизации ресурсов GPU или изучения передовых архитектур, таких как смесь экспертов (mixture-of-experts, MoE) или параметрически эффективных методов вроде LoRA, вам потребуется глубокое понимание внутренней работы трансформеров.

В этой главе мы постараемся развеять кажущуюся сложность архитектуры трансформеров, разложив ее на такие базовые концепции, как самовнимание (self-attention), многоголовое внимание (multihead attention), полносвязные нейросети (feed-forward networks, FFN) и позиционное кодирование. Понимание этих ключевых компонентов позволит вам не только уверенно использовать существующие языковые модели, но и эффективно адаптировать и оптимизировать их для реальных производственных сценариев.

2.1. От моделей seq2seq к трансформерам

Как уже ранее упоминалось, до появления трансформеров задачи машинного перевода обычно решались с помощью архитектуры "энкодер–декодер" на основе рекуррентных нейронных сетей (RNN), которая позволяла считывать предложение на исходном языке и формировать его представление фиксированной длины с последующей передачей этого представления декодеру (рис. 2.1).



Рис. 2.1. Иллюстрация обучения по принципу "последовательность-к-последовательности". RNN-энкодер принимает входную последовательность x и формирует промежуточное представление фиксированной длины z. Затем декодер обрабатывает это промежуточное представление, чтобы предсказать итоговый результат

Декодер формирует предложение на целевом языке, предсказывая по одному слову в авторегрессионном режиме, основываясь на предыдущих словах и на фиксированном по длине представлении от энкодера. Моделирование "последовательность-в-последовательность" играет ключевую роль в задачах, где крайне важен учет контекста. Главное преимущество такого подхода — способность понимать контекстуальную семантику предложения, — критически важный аспект при переводе. Например, слово "date" может означать как финик, так и свидание, и не хотелось бы, конечно, чтобы эти значения были перепутаны при переводе.

2.1.1. Сложности обучения рекуррентных нейронных сетей (RNN)

Обучение энкодера и декодера, архитектура которых основана на рекуррентных нейронных сетях, сопряжено с существенными сложностями, особенно когда модель должна выполнять перевод между языками с сильно различающимся синтаксисом или словарным запасом. Эти сложности зачастую возникают из-за накопления ошибок на протяженных временны́х последовательностях. Для их смягчения требуется тщательная инициализация (подбор начальных значений параметров) и использование ненасыщающихся функций активации, позволяющих избегать "застревания" в определенных диапазонах значений. Это обеспечивает эффективный градиентный поток в процессе обратного распространения ошибки и пакетной нормализации, способствуя повышению стабильности работы сети. В контексте сетей долгой краткосрочной памяти (LSTM) "стабильность" означает способность модели точно и надежно обучаться и представлять закономерности и взаимосвязи данных без чрезмерной чувствительности к незначительным изменениям или

шуму во входной информации. Проблемы обучения RNN, особенно глубоких RNN, подробно проанализированы в статье "On the Difficulty of Training Recurrent Neural Networks" ("О сложности обучения рекуррентных нейронных сетей") под авторством Glorot и др.

Зачастую необходимость применения архитектуры глубокой нейронной сети обуславливается предоставлением модели возможности выявления в данных сложных закономерностей и взаимосвязей, а также эффективного улавливания в последовательных данных долгосрочных зависимостей. В случае с LSTM это обычно достигается за счет последовательного наложения нескольких рекуррентных слоев друг на друга, что со временем позволяет сети осваивать все более сложные представления входных данных. Но с ростом числа слоев в сети усложняется распространение сигналов об ошибках в процессе обучения. А это, в свою очередь, может спровоцировать нестабильность в виде рассматриваемой далее проблемы исчезающего градиента.

2.1.2. Введение механизмов внимания

Хотя рекуррентные нейронные сети (RNN) можно спроектировать так, чтобы они выборочно запоминали или забывали информацию из прежних шагов, эффективное усвоение подобных зависимостей им все еще дается нелегко. Учитывать долгосрочные зависимости в LSTM моделям позволяет введение механизмов внимания, и все же они уступают в эффективности тем моделям, что используются в трансформерах, поскольку входные данные по-прежнему подаются в сеть последовательно, что замедляет обучение и вывод результатов.

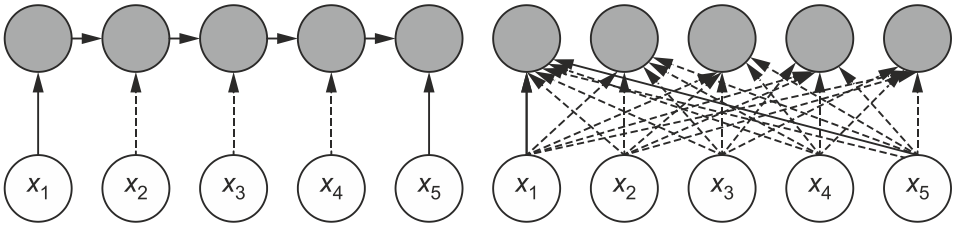

Рис. 2.2. Сравнение архитектур RNN (слева) и механизма самовнимания (справа). На схеме показано, что трансформеры, использующие самовнимание, могут напрямую учитывать информацию со всех позиций в последовательности, независимо от их временно́го положения. Пунктирными линиями обозначен поток информации или зависимости. Переменные $x_1 – x_5$ представляют входные данные, а круги над ними обозначают для RNN состояние сети на каждом временно́м шаге. В отличие от этого, для механизма самовнимания круги выше обозначают, как каждое состояние учитывает информацию от всех остальных состояний, а не только из непосредственного предшественника

В трансформерах применяется архитектура внимания, обеспечивающая более эффективную обработку длинных зависимостей (рис. 2.2). Еще одним преимуществом архитектуры трансформеров является использование многоголового внимания, по-

зволяющего модели параллельно улавливать различные аспекты входных данных, дополнительно усиливая ее способность эффективно обрабатывать долгосрочные зависимости. На рисунке показано, что трансформеры предоставляют альтернативный метод улавливания долгосрочных взаимосвязей в последовательных данных, не требующий ни последовательной обработки, ни глубокого наложения слоев.

Несмотря на рост популярности трансформеров, продемонстрировавших выдающуюся эффективность в решении многих задач, следует все же признать сохраняющуюся актуальность RNN и таких их вариантов, как LSTM. Эти модели попрежнему приносят пользу в различных областях применения, где важны специфические характеристики последовательных данных и временна́я динамика, например, в некоторых задачах прогнозирования временны́х рядов. По мере изучения деталей работы трансформерных моделей понимание как сильных сторон, так и ограничений RNN помогает осознать, почему новые методы стали существенным прорывом в области глубокого обучения.

2.1.3. Исчезающие градиенты: трансформеры приходят на помощь

В модели на основе RNN заложено принципиальное ограничение, называемое "проблемой исчезающего градиента (vanishing gradient problem)", затрудняющее распространение ошибок по сети и обновление параметров модели в процессе обратного распространения (backpropagation). Это усложняет обучение долгосрочным зависимостям и корректное моделирование последовательных данных, что, как показано на рис. 2.3, необходимо для многих задач преобразования "последовательность-в-последовательность" (sequence-to-sequence transduction). Проблема исчезающего градиента характерна для нейронных сетей, в которых градиенты становятся крайне малыми. В результате весá сети обновляются неэффективно, что приводит к замедлению процесса обучения и снижению производительности.

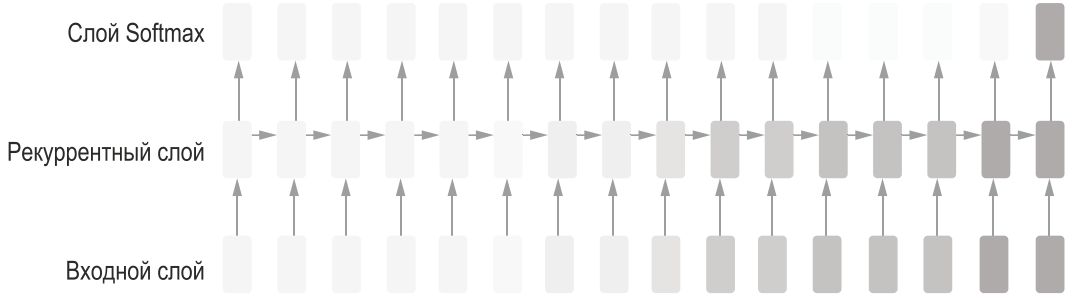

Рис. 2.3. Упрощенный пример проблемы исчезающих градиентов, где вклад градиента от более ранних шагов (слева) становится незначительным

Именно здесь на помощь приходит архитектура трансформеров, предлагающая решение этой проблемы за счет использования механизма внимания, позволяющего,

как показано на рис. 2.2, улавливать зависимости между всеми позициями во входной последовательности. На рис. 2.4 на примере предложения: "The quick brown fox jumps over the lazy dog and runs through the fields, while the lazy dog barks loudly and then chases the quick brown fox through the forest" ("Шустрая бурая лисица перепрыгивает через ленивую собаку и бежит по полям, в то время как ленивая собака громко лает, а затем гонится за шустрой бурой лисицей по лесу") показано сравнение градиентов для архитектур LSTM и трансформера. Здесь наглядно демонстрируется уменьшение градиентов в LSTM до очень малых величин при увеличении длины предложения. Попробуйте запустить код из репозитория книги, чтобы увидеть, как различная длина предложений сказывается на градиентах этих двух архитектур.

Осознав, что механизм внимания в трансформере позволяет снизить остроту проблемы исчезающих градиентов, давайте перейдем к рассмотрению его общей архитектуры. Начнем с изучения двух основных компонентов трансформера — энкодера и декодера — и того, как они работают вместе, преобразуя входную последовательность в выходную. Затем подробнее изучим ключевое нововведение архитектуры трансформера — механизм самовнимания, а в завершение рассмотрим принцип работы позиционного кодирования.

```
Gradients for RNN model:
tensor([-1.1910e-08, -5.4378e-09, -1.7932e-08, -1.0601e-07,  4.1367e-08,
        -9.4355e-08,  3.2711e-07,  2.2510e-08, -2.8239e-07, -1.7337e-06,
        -3.7372e-06, -3.9611e-07,  3.1367e-06, -2.0029e-05, -1.4689e-05,
         3.0543e-05,  6.7980e-05,  9.9472e-05,  4.7608e-06, -4.2451e-04,
        -6.8603e-06,  1.0076e-03,  6.5036e-04,  3.5185e-03,  8.8883e-03,
         4.6364e-03,  5.6579e-03,  1.5272e-03, -1.9715e-02, -3.8232e-01])

Gradients for self-attention model:
tensor([-0.0010, -0.0011, -0.0007, -0.0013, -0.0019, -0.0011, -0.0010, -0.0014,
        -0.0017, -0.0009, -0.0010, -0.0010, -0.0010, -0.0012, -0.0008, -0.0012,
        -0.0014, -0.0017, -0.0016, -0.0015, -0.0009, -0.0009, -0.0014, -0.0010,
        -0.0011, -0.0007, -0.0013, -0.0010, -0.0010,  0.0032])
```

Рис. 2.4. Сравнение градиентов для архитектур LSTM и трансформера на основе предложения: "The quick brown fox jumps over the lazy dog and runs through the fields, while the lazy dog barks loudly and then chases the quick brown fox through the forest"

2.1.4. Взрывающиеся градиенты: когда слишком большие значения градиентов нарушают процесс обучения

В моделях на основе RNN могут возникать не только проблемы исчезающих градиентов, но и их полные противоположности — проблемы взрывающихся градиентов. Взрывающиеся градиенты (exploding gradients) возникают, когда величины гра-

диентов, вычисляемые в процессе обратного распространения ошибки, становятся чрезмерно большими. Это приводит к резким и нестабильным обновлениям весов модели. Такая нестабильность зачастую приводит к хаотичному поведению в процессе обучения и может даже полностью препятствовать сходимости модели.

Рассмотрим, к примеру, обучение модели типа "последовательность-в-последовательность" на длинных, сильно повторяющихся текстовых данных либо числовых последовательностях с экстремальными значениями. По мере увеличения длины или сложности последовательности каждый рекуррентный шаг умножает градиенты, что приводит к нарастанию их величины. Без надлежащих методов смягчения последствий, например, без отсечения градиентов, столь большие градиенты могут вызывать слишком резкие изменения весов модели, что в итоге дестабилизирует процесс обучения и препятствует эффективному усвоению долгосрочных зависимостей. *Отсечение градиентов* (gradient clipping) — это метод, ограничивающий величину градиентов в ходе обучения нейронной сети. Он предотвращает их чрезмерный рост, тем самым стабилизируя процесс обучения и позволяя избежать проблемы взрывающихся градиентов.

Хотя большим языковым моделям (LLM) в значительной степени удается избегать этих проблем благодаря механизмам самовнимания, обеспечивающим прямые связи между элементами последовательности без последовательной обработки, важно понимать обе связанные с градиентами проблемы, чтобы в полной мере оценить те улучшения, которые предлагаются трансформерами.

2.2. Архитектура модели

Модель трансформера, несмотря на ее принципиальное отличие от традиционных RNN, в своей основе все же сохраняет структуру "энкодер–декодер". Эта приверженность доказывает устойчивость парадигмы "энкодер–декодер", которая продолжает служить прочной основой для передовых моделей.

Модель трансформера достигает своих уникальных возможностей за счет использования многослойного механизма внимания и покомпонентных полносвязных слоев как в энкодере, так и в декодере. Благодаря архитектурным решениям, показанным в левой и правой частях рис. 2.5, получается весьма гибкая и масштабируемая модель, превосходно справляющаяся с широким спектром задач предсказания "последовательность-в-последовательность". Масштабируемость и гибкость раскрывают истинную силу архитектуры трансформеров, делая ее краеугольным камнем в стремительно развивающемся мире искусственного интеллекта.

Теперь, получив наглядное представление об архитектуре трансформера, давайте займемся детальным изучением каждого ее компонента. Начнем с обсуждения энкодера, его конкретной структуры и функций, после чего перейдем к другим важным аспектам модели трансформера.