

Содержание

От издательства	12
Предисловие	13
Часть I. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ И АРХИТЕКТУРА ГЕНЕРАТИВНОГО ИИ	15
Глава 1. Архитектура Transformer и механизм внимания.....	16
1.1. Базовая архитектура Transformer	16
1.1.1. Структура кодировщик–декодировщик.....	17
1.1.2. Механизмы внутреннего и многоголового внимания.....	20
1.1.3. Остаточная связь и нормализация слоев	22
1.2. Основные принципы работы механизма внимания	25
1.2.1. Сравнение механизмов скалярного и аддитивного внимания.....	25
1.2.2. Нормализация Softmax	27
1.2.3. Разреженность матрицы внимания и оптимизация ускорения	30
1.3. Расширение и оптимизация архитектуры Transformer	32
1.3.1. Применение динамического внимания.....	32
1.3.2. Дальнее и разреженное внимание	34
1.3.3. Диверсифицированное позиционное кодирование	37
1.4. Контекстное окно	40
1.4.1. Увеличение контекстного окна	40
1.4.2. Компромисс между памятью и вычислительной сложностью	42
1.4.3. Оптимизация контекстного окна в DeepSeek-V3.....	45
1.5. Баланс между стоимостью обучения и вычислительной эффективностью	47
1.5.1. Тенденция роста числа параметров и вычислительных требований.....	48
1.5.2. Вычисления на GPU в архитектуре Transformer.....	51
1.5.3. Как DeepSeek-V3 снижает затраты на обучение	53
1.6. Краткое содержание главы	55

Глава 2. Детальный анализ архитектуры ядра DeepSeek-V3 и ее технологии обучения	56
2.1. Архитектура MoE и ее основные концепции.....	56
2.1.1. Знакомство с архитектурой MoE	57
2.1.2. Механизм сигмоидной маршрутизации.....	60
2.1.3. Архитектура DeepSeek-V3 на основе MoE	62
2.2. Преимущества обучения со смешанной точностью FP8.....	63
2.2.1. Основные принципы вычислений со смешанной точностью	64
2.2.2. Применение FP8 в обучении моделей.....	66
2.2.3. Стратегия улучшения производительности DeepSeek-V3 с помощью FP8	68
2.3. Алгоритм DualPipe и оптимизация связи	72
2.3.1. Алгоритм DualPipe.....	72
2.3.2. Механизм межзудовой связи All-to-All.....	75
2.3.3. Оптимизация пропускной способности Infiniband и NVLink	77
2.4. Распределенное обучение больших моделей	79
2.4.1. Баланс между параллелизмом данных и параллелизмом моделей	79
2.4.2. Распределенная архитектура обучения DeepSeek-V3	82
2.4.3. Устройство и оптимизация динамического планировщика скорости обучения.....	83
2.4.4. Стратегия балансировки нагрузки без вспомогательных потерь.....	85
2.4.5. Обучение многотокенному прогнозированию.....	88
2.5. Механизм кеширования и управление токенами.....	90
2.5.1. Что такое попадания и промахи кеша.....	91
2.5.2. Определение и процесс кодирования токена	93
2.5.3. Продвинутое механизме кеширования DeepSeek-V3	96
2.6. Модели серии DeepSeek	98
2.6.1. DeepSeek LLM.....	98
2.6.2. DeepSeek Coder	100
2.6.3. DeepSeek Math	102
2.6.4. DeepSeek VL	104
2.6.5. DeepSeek-V2	105
2.6.6. DeepSeek Coder V2.....	107
2.6.7. DeepSeek-V3	108
2.7. Краткое содержание главы.....	112

Глава 3. Введение в разработку больших моделей на основе DeepSeek-V3..... 113

3.1. Сценарии применения больших моделей.....	113
3.1.1. Генерация и резюмирование текста	114
3.1.2. Вопросно-ответная система и генерация диалогов	115
3.1.3. Многоязычное программирование и генерация кода.....	116
3.2. Преимущества и направления применения DeepSeek-V3.....	118

3.2.1. Фактическая производительность в различных областях	118
3.2.2. Возможности многоязычного программирования (на основе оценки Aider)	119
3.2.3. Анализ применения модели для разработки кода и решения математических задач	121
3.3. Теория и практика законов масштабирования	122
3.3.1. Связь между масштабом и производительностью модели	122
3.3.2. Эксперимент по применению законов масштабирования к малым моделям.....	124
3.4. Развертывание и интеграция модели.....	127
3.4.1. Вызов API и генерация в реальном времени.....	127
3.4.2. Локализованное развертывание	130
3.4.3. Стратегии оптимизации производительности	132
3.5. Распространенные проблемы и решения в разработке	135
3.5.1. Конструирование входных данных и управление генерацией	135
3.5.2. Проблемы предвзятости и надежности модели.....	138
3.5.3. Советы по решению конкретных проблем DeepSeek-V3	142
3.6. Краткое содержание главы	145

Часть II. РАЗРАБОТКА И ПРИМЕНЕНИЕ ПРИЛОЖЕНИЙ ГЕНЕРАТИВНОГО ИИ

146

Глава 4. Первый опыт работы с моделью DeepSeek-V3

147

4.1. Ведение диалогов и семантический анализ	147
4.1.1. Однораундовый и многораундовый диалоги.....	148
4.1.2. Контекстное взаимодействие	150
4.2. Способность к математическому мышлению	153
4.2.1. Ответы на типичные вопросы в области математики.....	153
4.2.2. Понимание и обоснование сложных проблем	155
4.3. Применение для вспомогательного программирования	160
4.3.1. Вспомогательная разработка алгоритмов	160
4.3.2. Разработка программного обеспечения	162
4.4. Краткое содержание главы	166

Глава 5. Открытая платформа DeepSeek и использование API

167

5.1. Знакомство с открытой платформой DeepSeek.....	167
5.1.1. Обзор основных модулей и сервисов платформы.....	168
5.1.2. Ключевые роли и сотрудничество в открытой экосистеме.....	171
5.2. Основные функции и примеры использования API DeepSeek	174
5.2.1. Механизм аутентификации и структура вызовов API.....	174
5.2.2. Часто применяемые интерфейсы и их назначение	177
5.3. Оптимизация производительности API и стратегия безопасности.....	181
5.3.1. Методы оптимизации производительности для уменьшения задержек.....	181

5.3.2. Защита данных и управление правами доступа	185
5.4. Краткое содержание главы	188

Глава 6. Диалоги, завершение текста и разработка специальных моделей

6.1. Основные принципы и методы генерации диалогов.....	189
6.1.1. Структура входных и выходных данных модели в задачах ведения диалога	190
6.1.2. Управление контекстом при взаимодействии на естественном языке.....	192
6.2. Принцип реализации и оптимизация автодополнения кода	195
6.2.1. Стратегия адаптации модели к языку программирования	196
6.2.2. Оптимизация функции глубокого завершения	198
6.3. Разработка индивидуальной модели на основе DeepSeek	202
6.3.1. Тонкая настройка модели и методы, ориентированные на конкретные задачи.....	202
6.3.2. Примеры использования персонализированных моделей для диалога и автодополнения	206
6.3.3. Более сложный случай: генерация кода и специализация на основе большой модели DeepSeek-V3	209
6.4. Краткое содержание главы	215

Глава 7. Продолжение диалога, формирование выходных данных FIM и JSON

7.1. Технические принципы и применение продолжения префикса диалога	216
7.1.1. Логика проектирования и реализация префиксного моделирования	217
7.1.2. Способы управления стилем продолжения	220
7.2. Анализ и применение технологии FIM	222
7.2.1. Пояснение к разным определениям задачи FIM.....	222
7.2.2. Способы оптимизации DeepSeek для задач FIM	225
7.3. Генерация выходных данных в формате JSON	227
7.3.1. Реализация модели для генерации структурированных данных.....	228
7.3.2. Применение вывода JSON на практике	230
7.3.3. Более сложный случай: многораундовый диалог и генерация структурированных данных.....	233
7.4. Краткое содержание главы.....	237

Глава 8. Функции обратного вызова и кеширование контекста на диске

8.1. Механизм обратного вызова и сценарии его применения	238
8.1.1. Функция обратного вызова и принципы ее реализации	239

8.1.2. Методы оптимизации обратного вызова на платформе DeepSeek	242
8.2. Основные принципы кеширования контекста на диске	245
8.2.1. Анализ влияния попаданий и промахов кеша	246
8.2.2. Реализация кеширования на жестком диске	249
8.3. Сочетание функций обратного вызова и механизма кеширования	252
8.3.1. Контекстно-ориентированное проектирование вызовов кеша	253
8.3.2. Пример повышения производительности за счет эффективного кеширования и комбинации обратного вызова	256
8.3.3. Сложный случай: применение DeepSeek и оптимизация интеллектуальной системы управления электростанцией	260
8.4. Краткое содержание главы	265

Глава 9. Библиотека промптов DeepSeek и ее дополнительные

возможности	266
9.1. Применение промптов при разработке кода	267
9.1.1. Доработка кода	267
9.1.2. Аннотирование кода	270
9.1.3. Генерация кода	272
9.2. Генерация и классификация контента	279
9.2.1. Классификация контента	280
9.2.2. Структурированный вывод	282
9.3. Ролевая игра	284
9.3.1. Ролевая игра (создание персонажа)	284
9.3.2. Ролевая игра (продолжение сценария)	286
9.4. Литературное творчество	287
9.4.1. Написание прозы	288
9.4.2. Написание стихов	290
9.5. Копирайтинг и продвижение	291
9.5.1. Создание плана копирайтинга	291
9.5.2. Генерация слоганов	294
9.6. Модель в роли эксперта-переводчика	296
9.6.1. Генерация промптов	296
9.6.2. Перевод в языковых парах	298
9.14. Краткое содержание главы	300

Часть III. ПРАКТИЧЕСКИЕ ПРОЕКТЫ И ИНТЕГРАЦИЯ ИИ В ПРИЛОЖЕНИЯ.....

301

Глава 10. Примеры интеграции, часть 1: разработка чат-клиента на основе LLM

302

10.1. Обзор чат-клиентов и их ключевые функции	302
10.1.1. Основная концепция дизайна чата	303
10.1.2. Анализ распространенных сценариев применения	305

10.2. Настройка и интеграция API DeepSeek	307
10.2.1. Получение и настройка ключа API	308
10.2.2. Вызовы стандартных интерфейсов	311
10.2.3. Интеграция API чат-клиента в приложение	315
10.3. Поддержка нескольких моделей и переключение между ними	318
10.3.1. Архитектура системы с переключаемыми моделями	318
10.3.2. Стратегии выбора модели в зависимости от задачи	321
10.3.3. Тестирование полного кода системы	325
10.4. Краткое содержание главы	329

Глава 11. Примеры интеграции, часть 2: разработка

интеллектуального помощника на основе ИИ

11.1. Начало эпохи ИИ-помощников и знакомство с технологией	331
11.1.1. Основные функции интеллектуального помощника на основе ИИ ...	331
11.1.2. Тенденции коммерческого применения ИИ-помощников	333
11.2. Конфигурация и применение API DeepSeek в ИИ-помощниках	335
11.2.1. Применение API DeepSeek в работе ИИ-помощника	336
11.2.2. Комбинация функций распознавания речи и обработки естественного языка	338
11.3. Внедрение и оптимизация функции интеллектуального помощника	340
11.3.1. Стратегии оптимизации для повышения точности вопросов и ответов	341
11.3.2. Технология непрерывного обучения и улучшения понимания контекста	344
11.4. Краткое содержание главы	347

Глава 12. Примеры интеграции, часть 3: разработка

вспомогательных плагинов на основе VS Code

12.1. Обзор и основные функции вспомогательных плагинов	349
12.1.1. Функциональное позиционирование вспомогательного плагины	349
12.1.2. Анализ полезных функций для разработчиков	354
12.2. Шаги по интеграции API DeepSeek в VS Code	358
12.2.1. Процесс вызова API в плагине	358
12.2.2. Оптимальное управление кешем вызовов API	360
12.3. Реализация автодополнения кода и рекомендаций по исправлению	364
12.3.1. Механизм автодополнения кода, основанный на глубоком семантическом понимании	364
12.3.2. Персонализированные рекомендации и гибкая настройка режимов разработки	368
12.4. Советы по использованию плагинов для повышения эффективности разработки	372

12.4.1. Применение инструментов для быстрого обнаружения и устранения ошибок	373
12.4.2. Автоматическая генерация кода	375
12.4.3. Быстрое создание комментариев и документации для крупных проектов	379
12.4.4. Управление проектами с помощью DeepSeek	383
12.4.5. Поддержка кодовой базы большого проекта	387
12.4.6. Генерация кода с поддержкой нескольких языков	391
12.4.7. Инструменты отладки, глубоко интегрированные в среду разработки	394
12.4.8. Оценка качества кода и генерация рекомендаций по оптимизации	397
12.5. Краткое содержание главы	401
Предметный указатель	402

От издательства

Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте www.dmkpress.com, зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу dmkpress@gmail.com; при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу http://dmkpress.com/authors/publish_book/ или напишите в издательство по адресу dmkpress@gmail.com.

Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу dmkpress@gmail.com. Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «ДМК Пресс» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты dmkpress@gmail.com.

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

Предисловие

В последние годы генеративный ИИ демонстрирует революционный прогресс. Благодаря выдающимся результатам в генерации текста и кода программ в работе с мультимодальными данными и в других областях он меняет привычные подходы к применению моделей искусственного интеллекта. В свое время новаторская архитектура Transformer легла в основу генеративного ИИ с его механизмом внутреннего внимания и модульной конструкцией. Большая модель DeepSeek фактически является результатом существенной оптимизации и расширения архитектуры Transformer. Она обеспечивает надежное решение масштабных задач генерации контента за счет интеграции таких технологий, как *смесь экспертов* (Mixture of Experts, MoE), *обучение со смешанной точностью FP8* и *распределенное обучение*.

DeepSeek-V3 – одна из больших моделей с открытым исходным кодом в серии DeepSeek. Она предназначена для решения таких задач, как генерация текста, автодополнение кода и мультимодальная генерация. Данная модель широко используется в диалоговых системах, интеллектуальных помощниках, программных модулях и других областях. Новаторские особенности этой модели заключаются в оптимизации, основанной на *законах масштабирования* (scaling laws) и объединении технологий *динамических контекстных окон* (dynamic context windows) и механизмов *разреженного внимания* (native sparse attention, NSA) для значительного повышения производительности и качества вывода модели при решении сложных задач. Эта книга посвящена DeepSeek-V3 и сочетает теоретический анализ с практическими примерами, что позволит читателям в полной мере изучить технологические основы и освоить прикладное применение большой модели с открытым исходным кодом.

Читателям будет представлено последовательное и полное руководство по освоению модели, от теоретических основ генеративного ИИ до технических особенностей архитектуры DeepSeek-V3, а затем и конкретных примеров разработки приложений. Независимо от того, являетесь ли вы исследователем технологий искусственного интеллекта или промышленным разработчиком, с помощью этой книги вы сможете быстро изучить архитектуру больших моделей DeepSeek и раскрыть потенциал ее применения в производственных и коммерческих сценариях.

Книга разделена на три части и состоит из 12 глав, охватывающих теоретические основы, описание подходов к разработке моделей и их практическое применение.

Первая часть (главы 1–3) посвящена теории и объясняет принципы работы архитектуры Transformer и механизма внимания, лежащих в основе

архитектуры DeepSeek-V3, а также основные подходы к построению моделей. Благодаря глубокому анализу архитектуры MoE, методов оптимизации контекстного окна и стратегий распределенного обучения перед читателем раскрываются уникальные преимущества DeepSeek-V3 с точки зрения стоимости обучения и эффективности вычислений. На этих главах будут основаны все дальнейшие технические рассуждения – не пропускайте их!

Вторая часть (главы 4–9) посвящена показателям эффективности и подходам к разработке приложений с использованием модели. В книге не только подробно описаны возможности DeepSeek-V3 в таких областях, как генерация диалогов, математические рассуждения и автодополнение кода, но и на подробных примерах кода показано, как использовать модель для точного решения задач. Кроме того, в книге детально рассмотрены дополнительные функции, такие как продолжение префикса диалога, режим генерации FIM и вывод JSON, которые помогают разработчикам эффективно подстраивать модели под свои потребности.

Третья часть (главы 9–12) охватывает широкий спектр примеров практического применения: от обратных вызовов функций и механизмов кеширования до разработки приложений. В книге представлено исчерпывающее руководство по всем аспектам – от простых вызовов API до оптимизации производительности посредством глубокого анализа особенностей открытой платформы DeepSeek и API. На примерах различных практических сценариев, включая разработку чат-агентов, интеллектуальных помощников и плагинов, в книге продемонстрированы обширные возможности применения DeepSeek-V3 в производственной среде.

В этой книге внимание в равной мере уделяется как теории, так и практике, и она помогает читателям быстро освоить основные навыки разработки больших моделей благодаря наличию подробных кейсов и детального технического анализа. К достоинствам книги следует отнести практическую интерпретацию законов масштабирования, демонстрацию примеров проектирования промптов и углубленное описание применения модели в промышленных сценариях. Эта книга будет полезна не только исследователям и разработчикам в области генеративного ИИ, но и энтузиастам передовых технологий, преподавателям и студентам университетов, желающим научиться применять большие модели на практике.

Мы хотели бы выразить нашу благодарность сообществу разработчиков ПО с открытым исходным кодом и техническим командам, которые принимали участие в разработке и применении DeepSeek-V3. Их усилия способствовали быстрому развитию технологии генеративного ИИ и послужили источником богатого содержательного материала для этой книги. Мы надеемся, что она станет для читателей надежным подспорьем в области генеративного ИИ и найдет применение в реальных проектах.

ЧАСТЬ I



Теоретические основы и архитектура генеративного ИИ

Главы 1–3 раскрывают теоретические основы и технические особенности архитектуры генеративного ИИ, закладывая фундамент для изучения DeepSeek-V3. В них подробно рассмотрены такие компоненты и методы архитектуры Transformer, как кодировщик и декодировщик, механизм внимания, позиционное кодирование и увеличение контекстного окна. Особое внимание уделяется ключевым особенностям DeepSeek-V3, таким как *динамическое внимание* (dynamic attention), *разреженное внимание* (sparse attention) и оптимизация *дальней зависимости*¹ (long-range dependency, LRD), освещаются инновации в проектировании моделей и стратегии оптимизации их производительности, а также даются исчерпывающие пояснения устройства и принципов работы больших моделей.

В то же время здесь подробно анализируется основная архитектура и технология обучения DeepSeek-V3, включая технические детали проектирования экспертной маршрутизации на основе MoE, обучения со смешанной точностью FP8 и распределенного обучения. В этом разделе посредством объяснения архитектуры графического процессора, оптимизации полосы пропускания и динамического планировщика скорости обучения показано, как DeepSeek-V3 обеспечивает баланс между вычислительной эффективностью и стоимостью обучения больших моделей с помощью технологических инноваций. Кроме того, исследование законов масштабирования дает теоретическую основу для изучения взаимосвязи между масштабом модели и ее производительностью, позволяя читателям более четко понимать технологическую эволюцию и логику оптимизации больших моделей.

¹ Долгосрочная зависимость является мерой ослабления статистической зависимости. – Прим. ред.

Глава 1

Архитектура Transformer и механизм внимания

Уникальный механизм внимания и модульная конструкция архитектуры Transformer со временем стали стандартом де-факто в области обработки естественного языка, способствуя быстрому развитию технологии больших моделей. Механизм внимания обеспечивает эффективное решение проблемы моделирования сложных данных путем динамического захвата зависимостей между элементами последовательности, в то время как механизмы многоголового внимания и остаточных связей дополнительно улучшают масштабируемость и стабильность модели.

В этой главе анализируется базовая структура и математические основы архитектуры Transformer, а также изучаются стратегии ее применения и оптимизации при обработке длинных контекстов, что закладывает прочную основу для понимания принципов работы больших моделей, таких как DeepSeek-V3.

1.1. Базовая архитектура Transformer

Архитектура Transformer¹ стала важной вехой в истории глубокого обучения благодаря своей гибкой модульной конструкции и мощным возможностям параллельных вычислений. В основе лежит хорошо известная структура кодировщик–декодировщик, но применяемая в сочетании с инновационной конструкцией механизма внутреннего внимания и многоголового внимания для точного моделирования сложных взаимосвязей во входной последовательности.

В то же время применение методов остаточных связей и нормализации слоев эффективно устраняет такие проблемы, как исчезновение градиента

¹ Модели, построенные по архитектуре Transformer, мы будем далее называть просто *трансформерами*, как принято в русскоязычной среде. – Прим. перев.

и нестабильность обучения. Далее в этой главе мы подробно проанализируем основные модули архитектуры Transformer и заложим техническую основу для глубокого понимания принципов работы больших моделей.

1.1.1. Структура кодировщик–декодировщик

Базовые принципы

Структура *кодировщик–декодировщик* (или *кодер–декодер*, *encoder-decoder*) лежит в основе архитектуры Transformer и в основном применяется для решения задач моделирования последовательности. Кодировщик преобразует входную последовательность в промежуточное представление, а затем декодировщик превращает это представление в целевую (выходную) последовательность.

1. *Роль кодировщика*: преобразование входной последовательности в многомерное представление фиксированной длины, которое содержит семантическую и контекстную информацию входной последовательности.
2. *Роль декодировщика*: генерация выходного сигнала в целевой последовательности на основе промежуточного представления, сгенерированного кодировщиком, и исторической информации о целевой последовательности.

Эта архитектура особенно хорошо подходит для таких задач, как машинный перевод и генерация текста. Например, при переводе предложения с одного языка на другой кодировщик может абстрагировать скрытые взаимосвязи исходного языка, а затем декодировщик генерирует контент на целевом языке.

Принцип работы модуля кодировщика

Кодировщик состоит из нескольких слоев, каждый из которых содержит две части: механизм внутреннего внимания и нейронную сеть прямого распространения.

1. *Механизм внутреннего внимания (self-attention)*: вычисляет взаимосвязь между каждым элементом в последовательности и динамически корректирует представление каждого элемента таким образом, чтобы оно могло улавливать контекстную информацию всей входной последовательности.
2. *Нейронная сеть прямого распространения (feedforward neural network)*: дополнительно обрабатывает выходные данные механизма внутреннего внимания для генерации представлений признаков более высокого уровня.

На вход кодировщика могут поступать векторы слов или другие формы *встроенного представления* (эмбединга, *embedding*). Выходные данные каж-

дого слоя выступают в качестве входных данных следующего слоя, постепенно улучшая способность модели абстрактно понимать семантику. Архитектура трансформера показана на рис. 1.1.

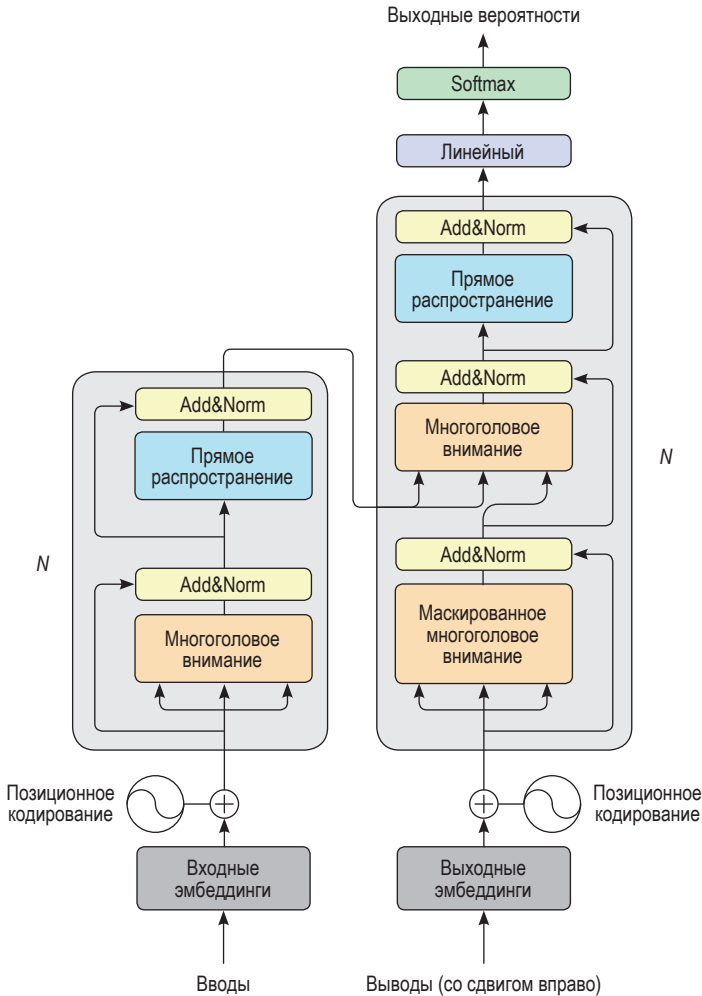


Рис. 1.1 ❖ Архитектура Transformer, состоящая из кодировщика и декодировщика

Принцип работы модуля декодировщика

Декодировщик похож на кодировщик и также состоит из нескольких последовательных слоев, но его рабочий процесс более сложен и состоит из трех основных частей.

1. **Механизм внутреннего внимания:** подобно кодировщику, механизм внутреннего внимания декодировщика отвечает за моделирование

внутренних взаимосвязей целевой последовательности, чтобы гарантировать, что каждое сгенерированное слово сохраняет согласованность с предыдущим словом.

2. *Механизм перекрестного внимания* (cross-attention): промежуточное представление, сгенерированное кодировщиком, объединяется с представлением целевой последовательности, сгенерированным декодировщиком, чтобы гарантировать, что информация входной последовательности может быть полностью использована в процессе декодирования.
3. *Нейронная сеть прямого распространения*: дополнительно извлекает и преобразует выходные данные механизма внимания для обеспечения генерации целевых последовательностей.

Улучшения кодировщика–декодировщика в DeepSeek-V3

Хотя основная идея структуры кодировщика–декодировщика в DeepSeek-V3 осталась неизменной, для повышения эффективности и качества результатов в ней были оптимизированы некоторые важные детали.

1. *Улучшенный механизм внимания*: DeepSeek-V3 представляет технологию *потенциального многоголового внимания* (multi-head potential attention), которая обладает повышенной способностью захватывать детали контекста входной последовательности посредством многоканальной обработки информации.
2. *Стратегия балансировки нагрузки без дополнительных потерь* (load balancing strategy): для решения распространенной проблемы неравномерного распределения ресурсов при обучении крупномасштабных моделей DeepSeek-V3 использует инновационные стратегии, гарантирующие, что вычислительные ресурсы могут быть полностью использованы как на этапах кодирования, так и на этапах декодирования.
3. *Параллельное прогнозирование целевых токенов* (multi-token prediction): декодировщик может прогнозировать несколько целевых токенов одновременно, что повышает скорость генерации и демонстрирует очевидные преимущества в производительности при выполнении задач генерации длинных последовательностей.

Практическое значение оптимизации кодировщика–декодировщика

Структура кодировщика–декодировщика устраняет ограничения традиционных моделей при обработке длинных последовательностей, позволяя трансформерам эффективно моделировать сложные отношения ввода-вывода, и служит технологической основой для последующей разработки *больших языковых моделей* (large language model, LLM).

Благодаря оптимизации DeepSeek-V3 из этой структуры извлекается максимум возможного. В результате она не только хорошо справляется с задача-

ми моделирования языка, но и обеспечивает мощную поддержку генерации кода, математических рассуждений и других задач, связанных с генерацией контента.

1.1.2. Механизмы внутреннего и многоголового внимания

Что такое внутреннее внимание?

Внутреннее внимание – это ключевой механизм трансформера, который используется для выявления и захвата корреляции между различными позициями во входной последовательности. Его функция заключается в том, чтобы позволить каждому элементу ввода (например, слову) динамически корректировать свое собственное представление на основе информации в других позициях. Эта способность позволяет модели глубже понимать контекстные отношения в последовательности. Структурная диаграмма механизмов *скалярного внимания* (dot product attention) и *многоголового внимания* (multi-head attention) показана на рис. 1.2.

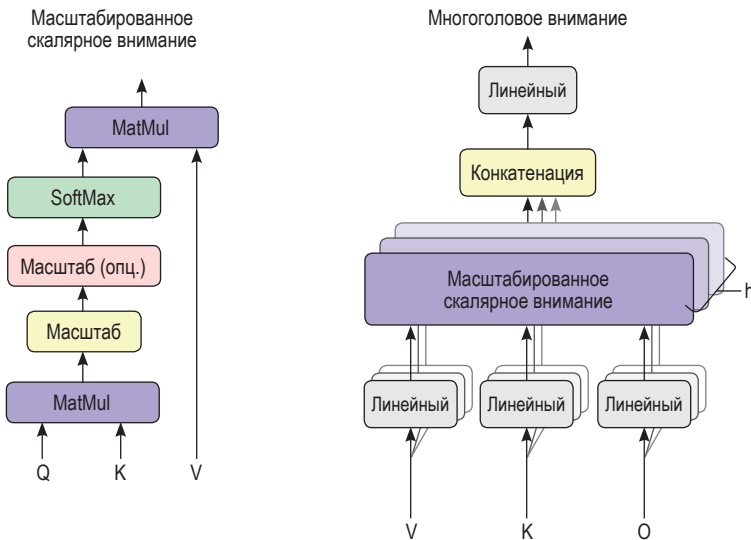


Рис. 1.2 ❖ Структурная схема скалярного и многоголового механизмов внимания

Рабочий процесс стандартного механизма внимания состоит из трех шагов.

1. *Расчет релевантности:* сравнивая каждый входной элемент со всеми другими элементами в последовательности, получаем набор оценок релевантности.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru