Содержание

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 Transfomer 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 Transformer[1] 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* (эмбеддинга, 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.

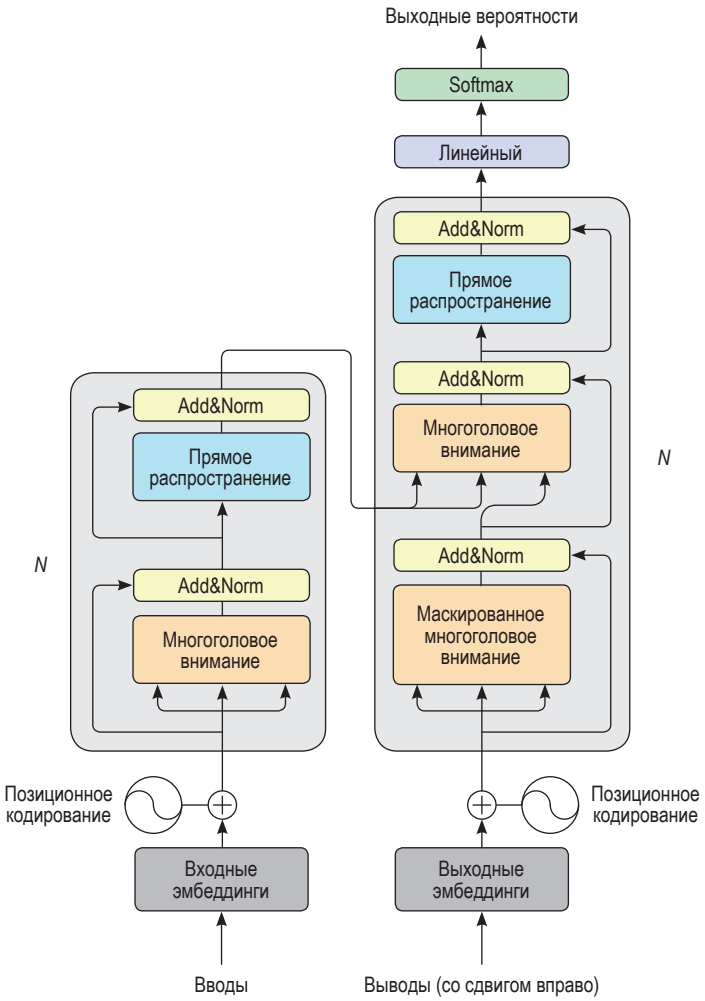

Рис. 1.1 ❖ Архитектура Transformer, состоящая из кодировщика и декодировщика

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.

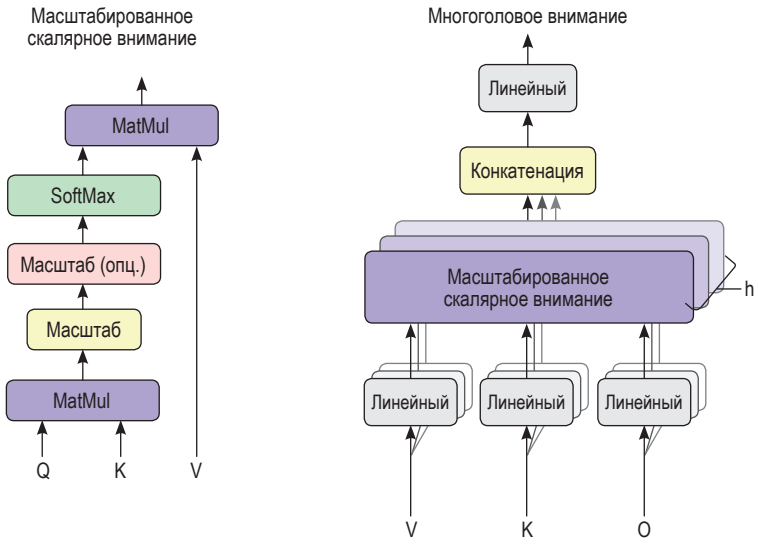

Рис. 1.2 ❖ Структурная схема скалярного и многоголового механизмов внимания

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.