In-Depth Analysis of Vector Post-Training Quantization (VPTQ) for Large Language Models

19 октября 2024 г.

Аннотация

Vector Post-Training Quantization (VPTQ) представляет собой инновационный метод квантизации больших языковых моделей (LLMs), позволяющий существенно уменьшить размер модели до экстремально низкой битности без значительных потерь в точности. В данном документе рассматривается принцип работы VPTQ, его преимущества и недостатки по сравнению с существующими методами, влияние на сферу искусственного интеллекта, потенциальные бизнес-применения и направления для будущих разработок.

1 Введение

С увеличением размеров крупных языковых моделей (LLMs) возникают значительные сложности при их развертывании и инференсе, связанные с высоким потреблением памяти и вычислительных ресурсов. Квантизация весов модели является одним из ключевых методов оптимизации, позволяющим уменьшить размер модели и ускорить её работу. Vector Post-Training Quantization (VPTQ) представляет собой передовой подход в области квантизации, достигающий экстремально низкой битности (до 2 бит) без необходимости дополнительного обучения модели.

2 Принцип работы VPTQ

VPTQ использует векторную квантизацию для представления весов модели с помощью индексов и кодовых книг (lookup tables). Основная идея заключается в следующем:

1. **Разбиение весов на векторы:** Весовая матрица модели разбивается на векторы фиксированной длины.
2. **Кластеризация:** Каждый вектор сравнивается с предопределёнными центроидами (центрами кластеров) в кодовой книге, и ему присваивается ближайший индекс.
3. **Использование оптимизации второго порядка:** Для минимизации ошибок квантизации применяется оптимизация второго порядка, что позволяет точно подбирать центроиды и минимизировать влияние квантизации на производительность модели.
4. **Обновление ошибок квантизации:** После квантизации отдельные векторы корректируются с использованием оставшихся ошибок, что позволяет добиться более высокой точности.

3 Преимущества VPTQ по сравнению с другими методами

VPTQ имеет значительные преимущества перед другими методами квантизации, такими как GPTQ, QuIP и AQLM. Давайте сравним эти методы по ключевым показателям:

3.1 Точность при маленьком размере

VPTQ хорошо сохраняет точность даже при очень сильном сжатии (до 2 бит):

- **VPTQ против GPTQ**: На модели LLaMA-2 7B при 2-битной квантизации VPTQ достигает перплексии 6.13 на WikiText-2, а GPTQ даёт непригодную модель с перплексией больше 50.
- **VPTQ против QuIP**: Для LLaMA-2 13B при 2 битах VPTQ достигает перплексии 5.32 на WikiText-2, а QuIP - 5.35.
- **VPTQ против AQLM**: На LLaMA-2 70B при 2 битах VPTQ и AQLM показывают похожие перплексии (3.93 и 3.94), но VPTQ лучше справляется с вопросами и ответами (68.6

3.2 Эффективность вычислений

VPTQ значительно ускоряет работу модели:

- **Скорость**: Для LLaMA-2 7B при 2 битах VPTQ обрабатывает 39.9 токенов в секунду, QuIP - 4.4, а AQLM - 19.4.
- **Использование памяти**: VPTQ требует только 2.28 ГБ для LLaMA-2 7B при 2 битах, а AQLM - 2.16 ГБ.

3.3 Время квантизации

VPTQ быстрее выполняет процесс квантизации:

- **VPTQ против AQLM**: Для LLaMA-2 7B VPTQ требуется 2 часа, а AQLM - 11.07 часов.
- **VPTQ против GPTVQ**: Для LLaMA-2 13B VPTQ нужно 3.2 часа, а GPTVQ - 3.7 часа.

3.4 Гибкость и масштабируемость

VPTQ хорошо работает с моделями разного размера:

- **Модели LLaMA-3**: На LLaMA-3 8B при 2 битах VPTQ достигает перплексии 9.29 на WikiText-2, что намного лучше QuIP (85.1) и GPTQ (210.0).
- **Mistral-7B**: При 2-битной квантизации VPTQ достигает перплексии 5.64 на WikiText-2, а QuIP - 6.02, AQLM - 6.32.

3.5 Высокая точность при маленьком размере

Одно из главных преимуществ VPTQ - это способность сохранять высокую точность модели даже при очень сильном сжатии (до 2 бит). Это достигается благодаря использованию сложной математики и векторной квантизации.

3.6 Эффективность вычислений

VPTQ значительно ускоряет работу модели по сравнению с обычными методами квантизации. Благодаря сжатию модели и оптимизации процесса раскодирования, VPTQ позволяет обрабатывать в 1.6-1.8 раз больше данных по сравнению с лучшими существующими методами.

3.7 Уменьшение требований к памяти

VPTQ позволяет уменьшить объём памяти, нужный для хранения модели, до 10-15

3.8 Простота использования

VPTQ можно легко использовать с существующими библиотеками для глубокого обучения, такими как PyTorch и TensorFlow. Это упрощает его применение в разных проектах и исследованиях.

4 Недостатки и ограничения VPTQ

4.1 Высокие вычислительные затраты на этапе квантизации

Несмотря на улучшенную эффективность, процесс квантизации с использованием VPTQ всё ещё требует значительных вычислительных ресурсов. Например, для квантизации модели LLaMA-2 70B требуется 4 GPU A100 с 80 ГБ памяти каждый в течение 19 часов.

4.2 Зависимость от архитектуры модели

Эффективность VPTQ может варьироваться в зависимости от архитектуры модели. Например, на LLaMA-3 70B при 2-битной квантизации VPTQ достигает перплексии 5.6, в то время как на LLaMA-2 70B — 3.93. Это указывает на то, что результаты могут различаться в зависимости от конкретной модели.

4.3 Ограниченная поддержка многоязычных моделей

Текущие тестирования VPTQ проводились преимущественно на английских текстах. Необходимо провести дополнительные исследования для подтверждения эффективности метода в многоязычных сценариях.

4.4 Необходимость специализированной настройки

Для достижения оптимальных результатов VPTQ может потребовать тонкой настройки параметров квантизации, таких как длина вектора и размер кодовой книги, что может быть сложным для пользователей без глубоких знаний в области квантизации.

5 Влияние VPTQ на сферу искусственного интеллекта

5.1 Оптимизация ресурсов

VPTQ помогает уменьшить размер моделей ИИ. Это позволяет экономить память и вычислительные ресурсы. В результате можно использовать более мощные модели на существующем оборудовании.

5.2 Расширение возможностей развертывания

Уменьшение размера моделей открывает новые возможности для их использования на устройствах с ограниченными ресурсами. Это включает некоторые мобильные устройства и устройства интернета вещей. Однако важно отметить, что полноценное применение на смартфонах пока ограничено.

5.3 Снижение затрат на инфраструктуру

Компании, предоставляющие AI-сервисы, могут снизить затраты на инфраструктуру. Это происходит за счет уменьшения требований к памяти и повышения пропускной способности систем.

5.4 Вклад в устойчивое развитие

Уменьшение размера моделей ведет к снижению энергопотребления. Это способствует экологической устойчивости и уменьшает углеродный след, связанный с работой больших языковых моделей.

6 Бизнес-случаи использования VPTQ

6.1 Оптимизация инфраструктуры AI

Компании, предоставляющие облачные AI-сервисы, могут использовать VPTQ для уменьшения затрат на хранение и обработку больших моделей. Это позволяет предлагать более доступные решения для клиентов.

6.2 Потенциал для мобильных AI-приложений

Сжатие моделей до 2-8 ГБ открывает перспективы для будущей интеграции мощных языковых моделей в мобильные устройства. Однако на данный момент это остается областью активных исследований и разработок.

6.3 Интернет вещей (IoT)

Устройства с ограниченными вычислительными ресурсами могут использовать VPTQ для выполнения некоторых AI-задач на месте. Это может повысить скорость реакции и защитить конфиденциальность данных.

6.4 Образование и исследования

Сокращение требований к ресурсам делает VPTQ полезным инструментом для образовательных учреждений и исследовательских центров. Это позволяет им работать с большими моделями без необходимости вкладывать средства в дорогостоящую инфраструктуру.

7 Направления будущих разработок

7.1 Расширение языковой поддержки

Необходимо протестировать VPTQ на моделях, обученных на различных языках. Это поможет подтвердить его эффективность в многоязычных сценариях.

7.2 Интеграция с аппаратными ускорителями

Работа над оптимизацией VPTQ для специальных аппаратных ускорителей, таких как TPU и FPGA, может повысить производительность и снизить энергопотребление.

7.3 Поддержка более широкого спектра архитектур моделей

Расширение поддержки VPTQ для разных архитектур моделей позволит использовать метод в большем количестве приложений. Это увеличит его универсальность и применимость.