Ограничения памяти в LLM-системах

Почему контекстное окно ограничено и что с этим делают

Студент: Салман Эль Фарисси Атмания

19 января 2026 г.

Математико-механический факультет

Структура доклада

1. Постановка задачи: что такое «память» в LLM
2. Ключевой механизм: self-attention и $O(N^2)$
3. Практические последствия длинного контекста
4. Подходы: оптимизации, разреженность, внешняя память (RAG)
5. Ограничения, риски, открытые проблемы

Введение и постановка задачи

- LLM эффективно работают с текстом, но обладают **ограниченной памятью**.
- На практике «память» чаще всего означает **контекстное окно** (число токенов на входе).
- Чем длиннее контекст, тем выше стоимость и тем сложнее сохранить качество.

Проблема

Почему архитектура Transformer ограничивает память, и почему масштабирование контекста сложно и дорого.

Рассматриваемый подход

- Современные LLM основаны на **Transformer**.
- «Память» реализуется **неявно** через attention между токенами.
- Каждый токен может «смотреть» на любой токен текущего контекста.

Ключевой вопрос

Почему нельзя просто сделать контекст *очень* большим (например, миллионы токенов)?

Ключевой механизм: self-attention

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

- Матрица внимания имеет размер $N \times N$, где N — длина контекста.
- Память и вычисления растут квадратично: $O(N^2)$.

Следствие

Увеличение контекста быстро упирается в ограничения GPU-памяти и времени инференса.

Практические последствия

- Ограничение длины диалога/документа: приходится обрезать историю.
- Растущая задержка ответа и стоимость при длинных запросах.
- Возможная деградация качества на длинных входах (модель «теряет» важное).

Наблюдение

Даже при больших окнах (32k–128k) качество не всегда линейно улучшается с длиной контекста.

Подходы к смягчению ограничений

Внутри attention

- **Оптимизация вычислений**: FlashAttention
- **Разреженное внимание**: Longformer
- **Аппроксимации**: Performer

Внешняя память

- **RAG**: извлечение фактов из базы/документов
- Короткий контекст + релевантные фрагменты

Важно

Ни один подход не «убирает» ограничение полностью — это компромиссы.

Ограничения и риски

- Разреженное внимание может терять глобальные зависимости.
- RAG зависит от качества retrieval: ошибки поиска $\Rightarrow$ ошибки ответа.
- Большой контекст не гарантирует «понимание»: модель может игнорировать часть ввода.
- Рост затрат: latency, VRAM, стоимость инференса.

Открытые проблемы

- Долгосрочная память: как хранить и обновлять знания между сессиями?
- Как делать это **надёжно** (без галлюцинаций и дрейфа)?
- Как измерять, что модель реально использует контекст, а не «делает вид»?

Идея направления

Переход от «длинного контекста» к системам с **управляемой памятью**: retrieval + summarization + контроль источников.

Выводы

- Ограничения памяти — следствие attention и квадратичной сложности.
- Масштабирование контекста дорого и не всегда даёт ожидаемый рост качества.
- Практика: комбинация методов (оптимизация attention + retrieval/внешняя память).
- Долгосрочная память и устойчивое обновление знаний остаются открытыми задачами.

Литература

- Vaswani et al. *Attention Is All You Need*. NeurIPS, 2017.
- Dao et al. *FlashAttention*. NeurIPS, 2022.
- Beltagy et al. *Longformer*. arXiv, 2020.
- Lewis et al. *Retrieval-Augmented Generation*. NeurIPS, 2020.