Эффективные архитектуры для решения задач на длинных последовательностях: обзор литературы

Каиров М.В.
kairov.mv@phystech.edu

Центр когнитивного моделирования МФТИ
Лаборатория нейронных систем и глубокого обучения МФТИ

2024 – Научно-исследовательский семинар

Table of Contents

- Introduction
- Problem Statement
- Sparse Attention
  - Longformer
  - Big Bird
  - Reformer
- Kernelized Attention
  - Linformer
  - ∞-former
  - Performer
- Recurrent Attention
  - Transformer-XL
  - Memformer
  - Recurrent Memory Transformer

Introduction

Motivation

- Transformers[1] are the most widely adopted type of architecture for solving tasks on sequences
- The attention mechanism, integral to Transformers, has quadratic computation complexity and memory consumption.
- On very long sequences (e.g. $> 10^4$ tokens long) traditional Transformers rapidly degrade in performance.

Research goal

The main goal of the current research is to analyze and compare the existing methods of optimizing the performance of transformer models on long sequences.

[1] A Vaswani. «Attention is all you need». В: *Advances in Neural Information Processing Systems* (2017).

Problem Statement

The optimization objective is the same for both the autoregressive and the long-context language modeling tasks:

$$\mathcal{L} = -\sum_{i=1}^{N} \log P\left(x_i | x_1, x_2, \ldots x_{i-1}\right).$$

Long-context language modeling invovlves handling sequences that are significantly longer than what a model can reasonably be trained on. Most long-context benchmarks implement the needle-in-a-haystack task, e.g. **BABILong**[2]. It requires the model to retrieve relevant facts from a vast pool of information.

[2] Yuri Kuratov и др. *BABILong: Testing the Limits of LLMs with Long Context Reasoning-in-a-Haystack*. 2024. arXiv: 2406.10149 [cs.CL]. URL: https://arxiv.org/abs/2406.10149.

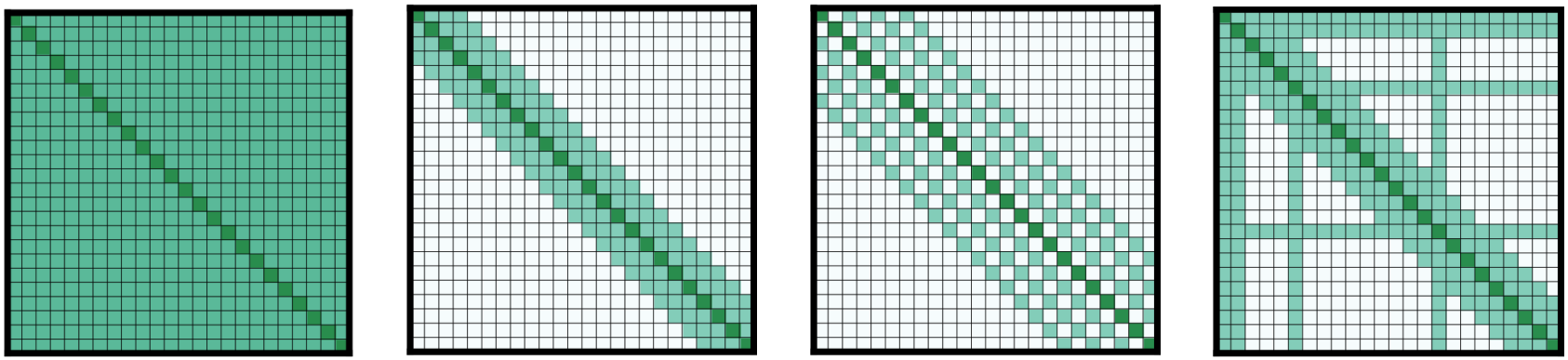
Sparse Attention

Sparse Attention mechanisms are designed to reduce the quadratic computational complexity of the standard self-attention by limiting the number of attention connections each query can make to a subset of keys, thus enabling the processing of much longer sequences efficiently. These techniques include but are not limited to:

- **Local Windowed Attention**: Restricts each token to attend only to a fixed-size window of neighboring tokens, which is particularly useful for capturing local context in sequences.
- **Block Sparse Attention**: Divides the input sequence into blocks and allows each block to attend to a few other blocks, creating a sparse attention pattern that reduces computation and memory requirements.
- **Random Sparse Attention**: Randomly samples a subset of keys for each query, leading to a more flexible and potentially more diverse attention pattern than fixed or local schemes.

Longformer[3]

- **Hybrid Sparse Attention**: Combines different sparse attention techniques for different tasks
- **Linear Complexity**: Reduces attention complexity from $O(N^2)$ to $O(N \cdot (w + g))$, where w is the width of the sliding window and g is the number of global tokens.



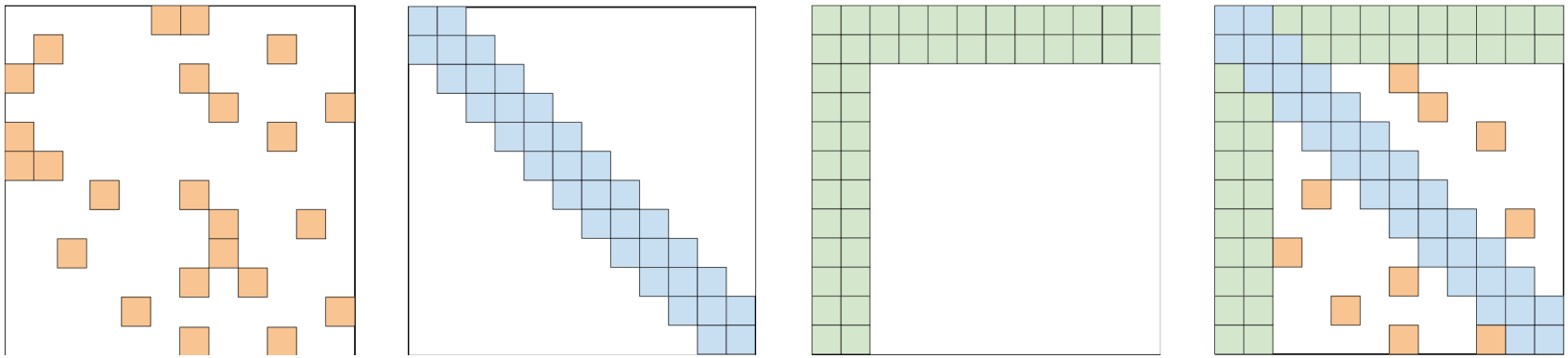
(a) Full attention (b) Sliding window attention (c) Dilated sliding window (d) Global + sliding window

Longformer attention patterns.

[3] Iz Beltagy, Matthew E. Peters и Arman Cohan. *Longformer: The Long-Document Transformer*. 2020. arXiv: 2004.05150 [cs.CL]. URL: https://arxiv.org/abs/2004.05150.

Big Bird[4]

- **Hybrid Sparse Attention**: Utilizes window attention, global attention and random attention.
- **Sub-quadratic Complexity**: Reduces attention complexity from $O(N^2)$ to $O(N \cdot \sqrt{N})$.



(a) Random attention

(b) Sliding window attention

(c) Global attention

(d) Big Bird

Big Bird attention patterns.

[4] Manzil Zaheer и др. «Big Bird: Transformers for Longer Sequences». В: *Advances in Neural Information Processing Systems*. Под ред. H. Larochelle и др. Т. 33. Curran Associates, Inc., 2020, с. 17283—17297. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf.

Reformer[5]

Center for Cognitive Modeling

- **Locally Sensitive Hashing (LSH)**: Replaces the traditional self-attention mechanism with a hash-based attention method, approximating attention to reduce memory and computation.
- **Reversible Layers**: Uses reversible residual layers to save memory during training by recomputing activations instead of storing them.
- **Complexity**: Reduces memory usage and computations to $O(N \cdot \log N)$.

[5] Nikita Kitaev, Łukasz Kaiser и Anselm Levskaya. «Reformer: The efficient transformer». В: *arXiv preprint arXiv:2001.04451* (2020).

Kernelized Attention

Kernelized Attention mechanisms transform the attention computation into a more efficient form by approximating the softmax function using kernel methods, allowing for linear-time complexity in attention computation and enabling the processing of very long sequences with minimal resource overhead.

- **Random Feature Maps**: Uses random projections to approximate the softmax function, converting the attention computation into a dot product operation that can be computed in linear time.
- **Generalized Kernel Functions**: Allows for the use of various kernel functions beyond the dot-product, providing flexibility in modeling different types of data and dependencies.
- **Efficient Matrix Multiplication**: Leverages fast matrix multiplication techniques to compute the attention weights, further optimizing the computational efficiency of the attention mechanism.

Linformer[6]

Center for Cognitive Modeling

- Replaces the full attention matrix with a linearly transformed one, reducing the complexity of attention computation from $O(n^2)$ to $O(n)$.
- Employs learned projections for keys and values to capture relevant information in a lower-dimensional space, preserving essential context.

[6] Sinong Wang и др. «Linformer: Self-attention with linear complexity». В: *arXiv preprint arXiv:2006.04768* (2020).

∞-former[7]

- Utilizes continuous attention: attention probability mass function over words is replaced by a probability density over a signal
- Implements a hybrid memory system that combines short-term and long-term memory using Gaussian basis functions for efficient attention computation across extensive input sequences.
- **Complexity**: $O(n)$ computations, $O(1)$ memory theoretically, but limited by floating point precision.

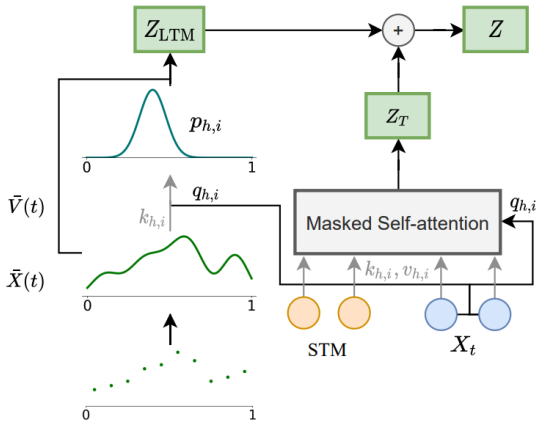


Рис.: ∞-former.

[7] Pedro Henrique Martins, Zita Marinho и André F. T. Martins. *∞-former: Infinite Memory Transformer*. 2022. arXiv: 2109.00301 [cs.CL]. URL: https://arxiv.org/abs/2109.00301

Performer[8]

- **Kernelized Attention (FAVOR+)**: Replaces softmax attention with a kernel-based approximation using positive orthogonal random features (FAVOR+).
- Supports various kernel functions beyond the dot-product attention, offering flexibility in modeling different types of data.
- **Complexity**: $O(n)$ time and memory complexity, making it suitable for large-scale sequence processing.

[8] Krzysztof Choromanski и др. «Rethinking attention with performers». В: *arXiv preprint arXiv:2009.14794* (2020).

Recurrent Attention

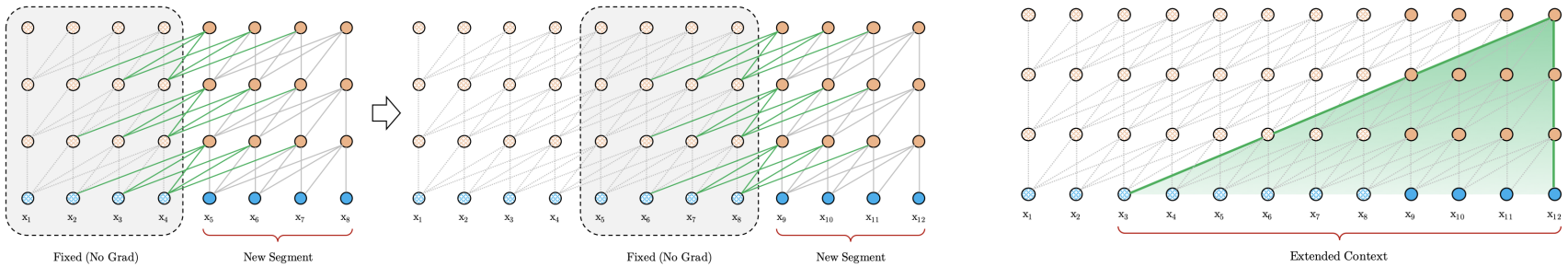
Center for Cognitive Modeling

Recurrent Attention mechanisms combine elements of recurrent neural networks (RNNs) with attention mechanisms, allowing models to process sequences sequentially while maintaining an internal state that can capture long-term dependencies, making them suitable for tasks requiring sequential processing and memory.

- **Sequential Processing**: Processes input sequences one element at a time, using an internal state that evolves as new elements are added, enabling the model to maintain a coherent understanding of the sequence context.
- **Attention over Past Inputs**: Attends to previous elements in the sequence to inform the processing of the current element, allowing for the integration of long-range dependencies in decision-making.
- **Memory-Efficient Computation**: Often requires less memory than traditional transformer models due to the sequential processing nature, making them more resource-efficient for long sequence processing.

Transformer-XL[9]

- Extends the context window across multiple segments by reusing hidden states from previous segments, effectively capturing longer contexts.
- Implements a segment-level recurrence mechanism to maintain coherence over segments, improving the model's ability to handle long-range dependencies.
- **Complexity**: $O(n)$ time complexity per segment, but with increased memory usage due to the retention of hidden states.



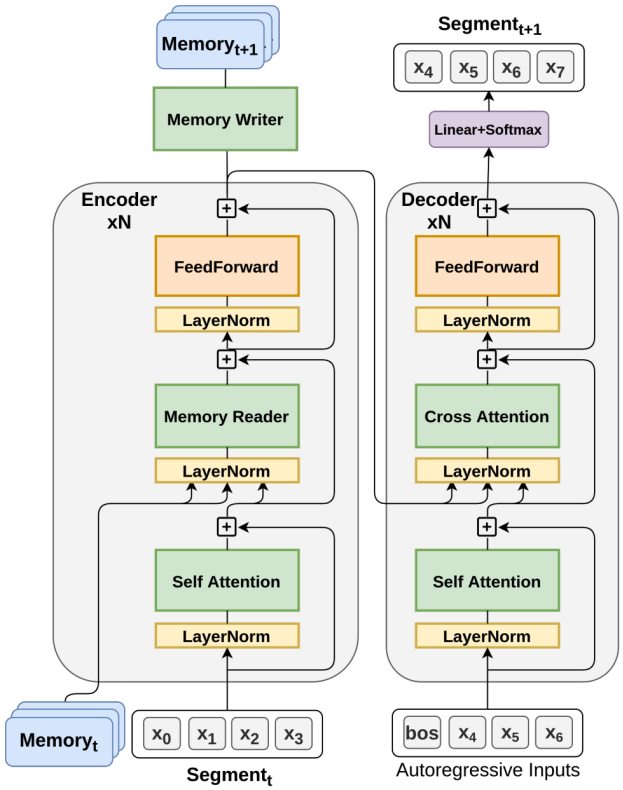
(a) Training phase. (b) Evaluation phase.

Рис.: Transformer-XL

[9] Zihang Dai и др. *Transformer-XL: Attentive Language Models Beyond a*

Memformer[10]

- Uses a memory module that selectively retains and updates information based on relevance, enabling the model to focus on critical parts of the sequence.
- Employs a gating mechanism to control the flow of information into and out of the memory module, optimizing resource allocation.
- **Complexity**: $O(n)$ time complexity, with potential for reduced memory usage through selective memory updating.

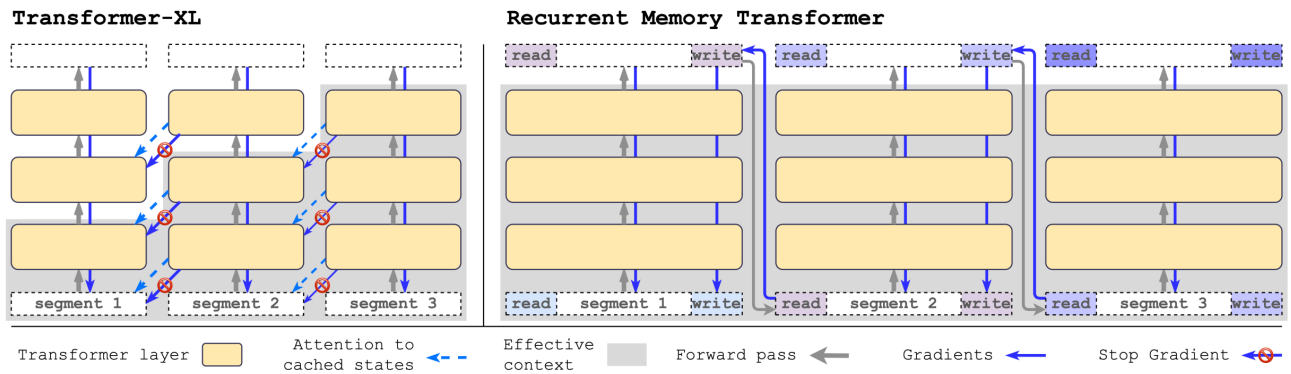


Memformer block.

[10] Qingyang Wu и др. *Memformer: A Memory-Augmented Transformer for Sequence Modeling*. 2022. arXiv: 2010.06891 [cs.CL]. URL: https://arxiv.org/abs/2010.06891.

Recurrent Memory Transformer[11]

- Combines the strengths of RNNs and transformers by using a recurrent mechanism to process sequences and a transformer-like architecture for parallelization.
- Enables transfer learning by implementing a recurrent wrapper for pre-trained transformer models.
- **Complexity**: $O(n)$ time complexity per time step, with potential for reduced memory usage through the use of memory cells.



Recurrent Memory Transformer.

[11] Aydar Bulatov, Yury Kuratov и Mikhail Burtsev. «Recurrent memory transformer». В: *Advances in Neural Information Processing Systems* 35 (2022), с. 11079—11091.

Спасибо за внимание!

kairov.mv@phystech.edu