Leave No Context Behind: Efficient Infinite Context Transformers with Infini-attention

Amit Kumar

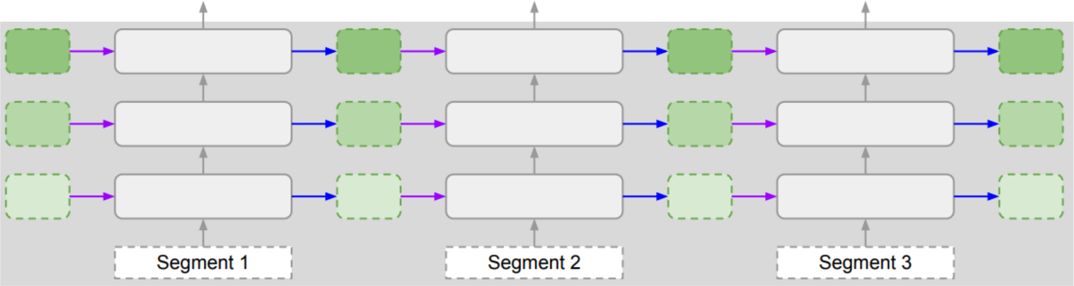
AI/NLP Engineer at E42.ai

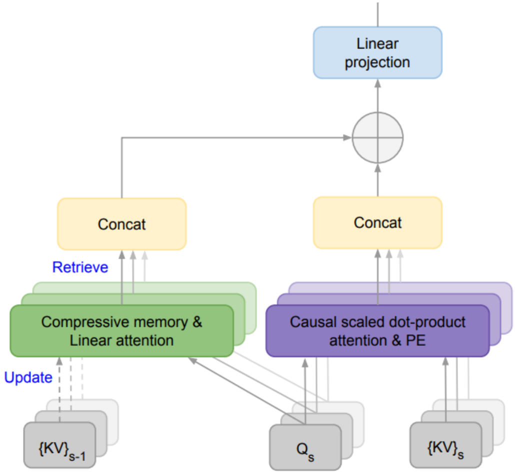
Table of contents

Introduction

- **Efficient Scaling**: Introduces a method to scale Transformer-based LLMs to handle infinitely long inputs with limited memory and computation.
- **Compressive Memory**: Infini-attention incorporates a compressive memory into the standard attention mechanism.
- **Combined Mechanisms**: It combines masked local attention and long-term linear attention within a single Transformer block.
- **Performance**: Demonstrates effectiveness on benchmarks like 1M sequence length context retrieval and 500K length book summarization.
- **Minimal Memory**: The approach uses minimal bounded memory parameters.

Infini-attention





- Infini-attention has an additional compressive memory with linear attention for processing infinitely long contexts. KV_{s-1} and KV_s are attention key and values for current and previous input segments, respectively and Qs the attention queries.

Vanilla Multi Head Attention

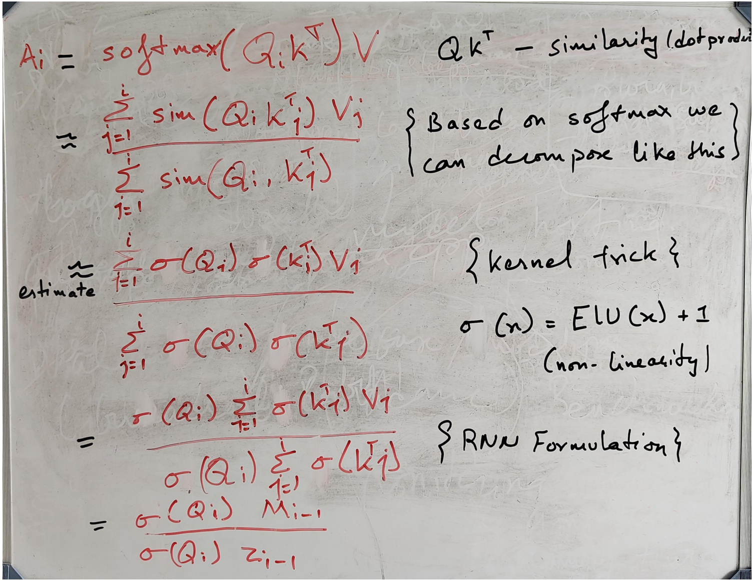
A single head in the vanilla MHA computes its attention context $\mathbf{A}_{\text{dot}} \in \mathbb{R}^{N \times d_{\text{value}}}$ from sequence of input segments $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{model}}}$ as follows. First, it computes attention query, key, and value states:

$$\mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V, \quad \mathbf{Q} = \mathbf{X}\mathbf{W}_Q.$$

Here, $\mathbf{W}_K \in \mathbb{R}^{d_{\text{model}} \times d_{\text{key}}}$, $\mathbf{W}_V \in \mathbb{R}^{d_{\text{model}} \times d_{\text{value}}}$ and $\mathbf{W}_Q \in \mathbb{R}^{d_{\text{model}} \times d_{\text{key}}}$ are trainable projection matrices. Then, the attention context is calculated as a weighted average of all other values as

$$\mathbf{A}_{\text{dot}} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{\text{model}}}}\right)\mathbf{V}.$$

Memory retrieval operation



Memory retrieval operation

In Infini-attention, we retrieve new content $\mathbf{A}_{\text{mem}} \in \mathbb{R}^{N \times d_{\text{value}}}$ from the memory $\mathbf{M}_{s-1} \in \mathbb{R}^{d_{\text{key}} \times d_{\text{value}}}$ by using the query $\mathbf{Q} \in \mathbb{R}^{N \times d_{\text{key}}}$ as:

$$\mathbf{A}_{\text{mem}} = \frac{\sigma(\mathbf{Q})\mathbf{M}_{s-1}}{\sigma(\mathbf{Q})\mathbf{z}_{s-1}}$$

Here, $\sigma \in \mathbb{R}^{d_{\text{key}}}$ is a nonlinear activation function.

Hidden state update

- Once the retrieval is done, we update the memory and the normalization term with the new KV entries and obtain the next states as

$$\mathbf{M}_s \leftarrow \mathbf{M}_{s-1} + \sigma(\mathbf{K}^T\mathbf{V})$$

$$\mathbf{z}_s \leftarrow \mathbf{z}_{s-1} + \sum_{t=1}^{N} \sigma(\mathbf{K}^T)$$

- **Delta rule**: The delta rule attempts a slightly improved memory update by first retrieving existing value entries and subtracting them from the new values before applying the associative bindings as new update.

$$\mathbf{M}_s \leftarrow \mathbf{M}_{s-1} + \sigma(\mathbf{K}^T)(\mathbf{V} - \frac{\sigma(\mathbf{K})\mathbf{M}_{s-1}}{\sigma(\mathbf{K})\mathbf{z}_{s-1}})$$

Combining local attention and RNN

Long-term context injection: We aggregate the local attention state $\mathbf{A}_{\text{dot}}$ and memory retrieved content $\mathbf{A}_{\text{mem}}$ via a learned gating scalar β:

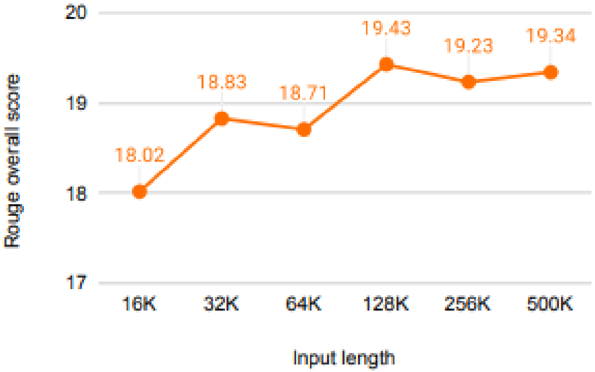
$$\mathbf{A} = \text{sigmoid}(\beta) \odot \mathbf{A}_{\text{mem}} + (1 - \text{sigmoid}(\beta)) \odot \mathbf{A}_{\text{dot}}.$$

Experimental Results

- 500K length book summarization (BookSum) results. The BART, PRIMERA and Unlimiformer results are from Bertsch et al. (2024).

Model	Rouge-1	Rouge-2	Rouge-L	Overall
BART	36.4	7.6	15.3	16.2
BART + Unlimiformer	36.8	8.3	15.7	16.9
PRIMERA	38.6	7.2	15.6	16.3
PRIMERA + Unlimiformer	37.9	8.2	16.3	17.2
Infini-Transformers (Linear)	37.9	8.7	17.6	18.0
Infini-Transformers (Linear + Delta)	**40.0**	**8.8**	**17.9**	**18.5**

Experimental Results



- Infini-Transformers obtain better Rouge overall scores with more book text provided as input.

Conclusion

1. **Memory Matters**: Having a good memory system is essential not just for understanding long texts with Large Language Models (LLMs), but also for other cognitive tasks like reasoning, planning, and learning how to learn.
2. **Introducing a Memory Module**: This work introduces a memory module that closely integrates with the standard attention mechanism used in LLMs. This modification helps LLMs handle very long contexts using limited memory and computational resources.
3. **Scalability**: The proposed approach allows LLMs to effectively process extremely long input sequences, up to a million tokens, while still outperforming existing methods on tasks like language modeling and summarization of books.
4. **Generalization**: The approach also demonstrates promising generalization capabilities across different lengths of input sequences. For example, a model trained on shorter sequences could effectively solve problems involving much longer sequences.

Thank You