Efficient Streaming Language Models with Attention Sinks

Amit Kumar

AI/NLP Engineer at E42.ai

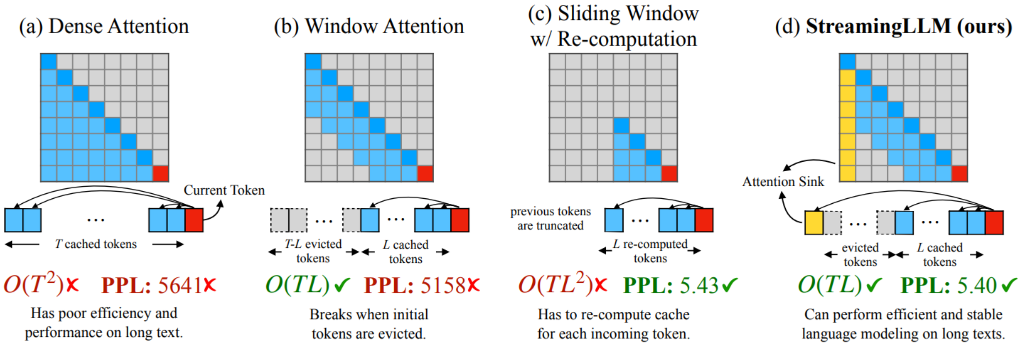
Table of contents

Introduction

- To unleash the full potential of pretrained LLMs, they should be able to efficiently and accurately perform long sequence generation. For example, an ideal ChatBot assistant can stably work over the content of recent day-long conversations.
- The longest acceptable sequence length for lengthy inputs remains finite, which prevents these models from being deployed for extended periods of time.
- When applying LLMs for infinite input streams, two primary challenges arise:
    1. During the decoding stage, Transformer-based LLMs cache the Key and Value states (KV) of all previous tokens, which can lead to excessive memory.
    2. Existing models have limited length extrapolation abilities, i.e., their performance degrades when the sequence length goes beyond the attention window size set during pre-training.

Can we deploy an LLM for infinite-length inputs without sacrificing efficiency and performance?

StreamingLLM vs. existing methods



Attention sink

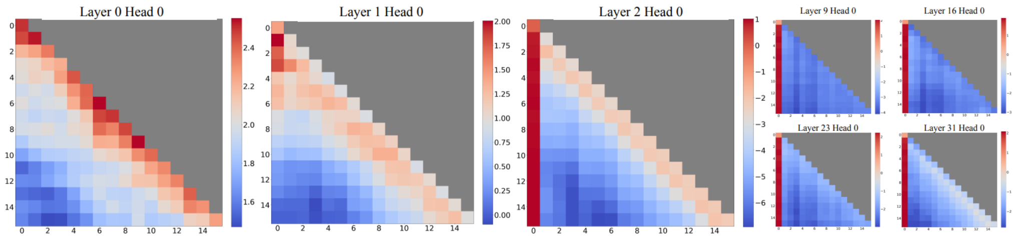


Figure 2: Visualization of the *average* attention logits in Llama-2-7B over 256 sentences, each with a length of 16. Observations include: (1) The attention maps in the first two layers (layers 0 and 1) exhibit the "local" pattern, with recent tokens receiving more attention. (2) Beyond the bottom two layers, the model heavily attends to the initial token across all layers and heads.

StreamingLLM approach

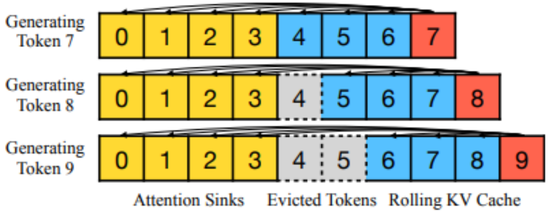
Why do LLMs break when removing *initial* tokens' KV? We visualize attention maps from all layers and heads of the Llama-2-7B and models in Figure 2. We find that, beyond the bottom two layers, the model consistently focuses on the initial tokens across all layers and heads. The implication is clear: removing these initial tokens' KV will remove a considerable portion of the denominator in the SoftMax function (Equation 1) in attention computation. This alteration leads to a significant shift in the distribution of attention scores away from what would be expected in normal inference settings.

$$\text{SoftMax}(x)_i = \frac{e^{x_i}}{e^{x_1} + \sum_{j=2}^{N} e^{x_j}}, \quad x_1 \gg x_j, j \in 2, \dots, N \tag{1}$$

There are two possible explanations for the importance of the initial tokens in language modeling: (1) Either their semantics are crucial, or (2) the model learns a bias towards their absolute position. To distinguish between these possibilities, we conduct experiments (Table 1), wherein the first four tokens are substituted with the linebreak token "\n". The observations indicate that the model still significantly emphasizes these initial linebreak tokens. Furthermore, reintroducing them restores the language modeling perplexity to levels comparable to having the original initial tokens. This suggests that the absolute position of the starting tokens, rather than their semantic value, holds greater significance.

Result

Window attention has poor performance on long text. The perplexity is restored when we reintroduce the initial four tokens alongside the recent 1020 tokens (4+1020). Substituting the original four initial tokens with linebreak tokens (4+1020) achieves comparable perplexity restoration.



Llama-2-13B	PPL (↓)
0 + 1024 (Window)	5158.07
4 + 1020	5.40
4"\n"+1020	5.60

Conclusion

Deploying LLMs in streaming apps is crucial but has challenges. Window attention helps but falters when tokens are excluded. StreamingLLM, a simple solution to handle long texts without fine-tuning.

- StreamingLLM uses ”attention sinks” with recent tokens.
- It can model texts up to 4 million tokens efficiently.
- Pre-training with a dedicated sink token enhances streaming performance.
- StreamingLLM separates pre-training window size from text generation length for seamless streaming LLM deployment.

References

- NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation.
- Things I'm learning while training superhot., 2023
- How Long Can Open-Source LLMs Truly Promise on Context Length?
- FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning

Thank You