Sparsity for Long-Context LLM Inference

Zehao Fan

1. Introduction

Background Long-context large language models (LLMs) have achieved significant advancements across various fields. However, they face computational challenges due to high memory demands, particularly when the context window extends to lengths as large as 128K or even 1 million tokens. The self-attention mechanism, a core component in transformers, exhibits quadratic computational complexity with respect to sequence length, becoming a substantial bottleneck in managing such extended contexts. Efficient techniques are essential for practical deployment, especially for tasks that require lengthy contexts, such as dialogue systems, document summarization, and question answering.

Importance of Sparsity Introducing sparsity is crucial for handling these large-scale computations. By selectively reducing attention calculations, activating specific parameters, and optimizing key-value (KV) caching mechanisms, sparsity techniques enable models to retain essential information while minimizing computational costs. For example, SPARSEK(Lou et al., 2024) Attention optimizes sparse attention by limiting the number of key-value pairs each query attends to, achieving linear time complexity during generation and offering notable memory efficiency. This efficiency is increasingly critical for models like LLaMA and GPT-4, which are often deployed on resource-constrained devices.

Objective and Contribution This report aims to survey key sparsity techniques—specifically focusing on attention sparsity, activation sparsity, model structure sparsity, and KV cache optimization—to address the inference challenges in long-context LLMs. Each of these techniques represents a different approach to reducing computational load, preserving memory, and enhancing inference efficiency without the need for extensive retraining.

2. Problem Formulation

Challenges of Long-Context LLM Inference Managing long sequences in LLMs demands substantial computational resources. Due to the quadratic complexity of the attention mechanism, computation time and memory requirements increase exponentially with sequence length. For instance, a LLaMA-65B model requires approximately 365GB of KV cache memory for a batch size of 128 and a sequence length of 2048, which is nearly three times larger than the model's parameters. Although techniques like KV cache eviction have been proposed, they often lead to performance degradation when important tokens are inadvertently removed, resulting in context loss.

Defining Sparsity and Its Goals Sparsity involves selectively activating only the most critical parts of the model's computations or storage. Attention sparsity reduces the number of tokens each query attends to; activation sparsity deactivates unnecessary neurons; model structure sparsity prunes less critical components like attention heads or layers; and KV cache optimization focuses on retaining only essential tokens in memory to manage long-context dependencies. The ultimate goal is to balance memory usage and computational cost without compromising the model's ability to accurately process long contexts.

3. Techniques for Introducing Sparsity

3.1. Attention Sparsity

Sparse Attention Mechanisms Techniques such as SPARSEK(Lou et al., 2024) improve efficiency by reducing the quadratic complexity of self-attention to linear. SPARSEK(Lou et al., 2024) utilizes a differentiable top-K mask operator that allows each query to attend to only a fixed number of key-value pairs, thereby achieving constant memory usage and linear computational complexity during inference. Similarly, Low-Rank Approximation for Sparse Attention(Song et al., 2024) employs low-rank projections to approximate the attention map, selecting only the most relevant tokens for computation, which leads to reduced memory usage and enhanced speed.

Query-Aware Techniques Quest(Tang et al., 2024b) introduces a dynamic, query-dependent method by identifying critical tokens based on the current query vector. By selecting only the top-K pages relevant to each query, Quest(Tang et al., 2024b) minimizes the number of tokens processed in each attention calculation, significantly accelerating self-attention and achieving a 7.03× reduction in latency.

3.2. Activation Sparsity

Dynamic ReLU-based Sparsity TurboSparse(Song et al., 2024) optimizes activation sparsity using a novel dynamic

ReLU (dReLU) function that deactivates certain neurons based on the input, reducing unnecessary calculations without requiring additional fine-tuning. This selective deactivation effectively lowers both the number of active neurons and the overall computational load during inference.

Top-K Sparsification Q-Sparse(Wang et al., 2024) applies top-K sparsification to activations, enabling a fully sparse model where only the most critical activations are computed. This method scales well, providing efficiency gains in both computation and input/output operations, making it suitable for managing large models under limited computational budgets.

3.3. Model Structure Sparsity

Layer and Attention Head Pruning Techniques such as Mixture-of-Experts (MoE) selectively activate attention heads and layers based on task requirements. This approach reduces the number of active components during inference, significantly lowering resource demands without major loss in accuracy. LoRA-Sparse(Song et al., 2024) also demonstrates that removing redundant attention heads can further improve computational efficiency without sacrificing performance.

3.4. KV Cache Techniques

Eviction H2O (Heavy-Hitter Oracle)(Zhang et al., 2023) retains only the most important tokens (heavy hitters) in the KV cache by analyzing accumulated attention scores. This strategy prioritizes tokens that contribute the most to downstream attention computations, reducing memory usage while maintaining inference quality.

Merging CaM(Tang et al., 2024a) and D2O(Wan et al., 2024) introduce merging strategies that combine tokens slated for eviction with others based on attention score similarity. By retaining essential information while compressing less critical data, these methods minimize performance degradation even with significant reductions in the KV cache.

Compression Quest(Tang et al., 2024b) employs page-level compression by selectively loading only the top-K pages of KV pairs, which reduces latency in self-attention calculations without compromising task accuracy. Quest's approach of loading only the critical pages enhances memory efficiency, particularly for long-context tasks.

4. Challenges and Open Questions

Scope Limitations Attention sparsity methods such as SPARSEK(Lou et al., 2024) may be less effective for tasks that require maintaining dense, contextually rich connections among all tokens. In such cases, reducing attention connections could impair performance. Similarly, KV merging techniques might not perform well for shorter contexts where every token is essential for accurate inference.

Balancing Sparsity and Accuracy A key challenge is balancing computational efficiency with model accuracy. While techniques like LoRA-Sparse(Song et al., 2024) demonstrate that sparse attention can sometimes enhance performance, there is a risk of diminishing returns or even accuracy degradation if sparsity is applied too aggressively.

Future Research Directions Future research could focus on developing adaptive sparsity techniques that adjust based on input context length or specific task requirements, further optimizing model efficiency. Additionally, exploring hybrid methods that integrate sparsity across attention, activation, and KV cache management may yield new insights into efficient long-context LLM deployment.

5. Conclusion

Sparsity techniques in attention, activation, model structure, and KV cache management play a crucial role in optimizing inference in long-context LLMs. By selectively reducing computational load and memory usage, these techniques enable efficient processing without significant loss of accuracy. Future research could explore more adaptive, task-aware sparsity techniques and further investigate sparsity for diverse LLM architectures. Such advancements will help scale these models for real-world applications that demand efficient long-context processing.

References

Lou, C., Jia, Z., Zheng, Z., and Tu, K. Sparser is Faster and Less is More: Efficient Sparse Attention for Long-Range Transformers. *arXiv e-prints*, art. arXiv:2406.16747, June 2024. doi: 10.48550/arXiv.2406.16747.

Song, L., Chen, Y., Yang, S., Ding, X., Ge, Y., Chen, Y.-C., and Shan, Y. Low-rank approximation for sparse attention in multi-modal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13763–13773, 2024. doi: 10.1109/CVPR52733.2024.01306.

Song, Y., Xie, H., Zhang, Z., Wen, B., Ma, L., Mi, Z., and Chen, H. Turbo Sparse: Achieving LLM SOTA Performance with Minimal Activated Parameters. *arXiv e-prints*, art. arXiv:2406.05955, June 2024. doi: 10.48550/arXiv.2406.05955.

Tang, J., Zhao, Y., Zhu, K., Xiao, G., Kasikci, B., and Han, S. Quest: Query-Aware Sparsity for Efficient Long-Context LLM Inference. *arXiv e-prints*, art. arXiv:2406.10774, June 2024a. doi: 10.48550/arXiv.2406.10774.

Tang, J., Zhao, Y., Zhu, K., Xiao, G., Kasikci, B., and Han, S. Quest: Query-Aware Sparsity for Efficient Long-Context LLM Inference. *arXiv e-prints*, art. arXiv:2406.10774, June 2024b. doi: 10.48550/arXiv.2406.10774.

Wan, Z., Wu, X., Zhang, Y., Xin, Y., Tao, C., Zhu, Z., Wang, X., Luo, S., Xiong, J., and Zhang, M. D2O: Dynamic Discriminative Operations for Efficient Generative Inference of Large Language Models. *arXiv e-prints*, art. arXiv:2406.13035, June 2024. doi: 10.48550/arXiv.2406.13035.

Wang, H., Ma, S., Wang, R., and Wei, F. Q-Sparse: All Large Language Models can be Fully Sparsely-Activated. *arXiv e-prints*, art. arXiv:2407.10969, July 2024. doi: 10.48550/arXiv.2407.10969.

Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., Wang, Z., and Chen, B. H_2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. *arXiv e-prints*, art. arXiv:2306.14048, June 2023. doi: 10.48550/arXiv.2306.14048.