Accelerating Transformer Models: FPGA-Based Hardware Optimization and Heterogeneous Computing Strategies

Ziyi Wang[1,a,*]

[1]James Watt School of Engineering, University of Glasgow, University Avenue, Glasgow, G12 8QQ, United Kingdom
a. 3026887w@student.gla.ac.uk
**corresponding author*

Abstract: The Transformer model has gained widespread application in natural language processing (NLP) and computer vision due to its remarkable global dependency modeling capability. However, its large parameter size and high computational complexity pose significant challenges for resource-constrained hardware platforms. This paper thoroughly analyzes the advantages and limitations of FPGA as a hardware acceleration platform, highlighting its potential in low-power, low-latency inference through its flexible architecture and parallel processing capabilities. At the same time, it reveals the bottlenecks caused by storage capacity and loading efficiency in deploying Transformer models. By leveraging techniques such as model compression, quantization, and dataflow optimization, the loading performance of FPGA is significantly improved. Furthermore, this paper explores the collaborative mechanisms of FPGA within heterogeneous computing systems, demonstrating its capability to work alongside CPUs and GPUs in addressing complex AI tasks. Against the backdrop of limitations associated with traditional general-purpose chips, this paper examines the optimized design of application-specific chips (such as Sohu chips) and their performance advantages in Transformer tasks, while pointing out their lack of flexibility. To address the computational bottlenecks of the self-attention mechanism in Transformer models, an optimization method combining relative position encoding is proposed, effectively reducing complexity and enhancing model performance. By integrating model and hardware in a deeply synergistic manner, this paper provides theoretical foundations and practical guidance for the efficient deployment of Transformer models, opening up new possibilities for applications in edge computing and embedded systems.

Keywords: NLP, Transformer, FPGA, hardware optimization, model compression, heterogeneous systems, self-attention optimization

1. Introduction

In recent years, natural language processing (NLP) has achieved significant progress driven by advancements in deep learning technologies, with the Transformer architecture emerging as a pivotal milestone in the field. Transformer-based models (e.g., BERT, GPT) have demonstrated exceptional performance in tasks such as machine translation, question answering, and sentiment analysis, owing to their remarkable ability to model global dependencies. However, current NLP systems still face numerous challenges, particularly in complex tasks. Most systems rely heavily on superficial features (e.g., word frequency) and struggle to capture the deeper meanings of language. This limitation hampers their performance in scenarios requiring contextual understanding or rapid adaptation to domain-specific language.

The Transformer model, as a novel architecture, enhances semantic understanding through its core self-attention mechanism. Simply put, this mechanism enables the model to evaluate the relationships between words in a text, leading to a more accurate comprehension of the overall meaning of a passage. Nevertheless, the enormous parameter size of Transformer models (e.g., BERT-LARGE with 340 million parameters) and their high computational complexity pose significant challenges for deployment in resource-constrained embedded devices and edge computing scenarios. Particularly in applications such as speech recognition, real-time video analysis, and robotic control, which demand efficient real-time inference, conventional hardware platforms (e.g., CPUs, GPUs) often struggle to balance high performance with low power consumption.

To address these challenges, field-programmable gate arrays (FPGAs) have emerged as an ideal choice for accelerating deep learning inference due to their hardware flexibility and efficient parallel computation capabilities. Compared to traditional hardware, FPGAs can dynamically reconfigure computational units and data paths based on specific tasks, thereby reducing power consumption while delivering high-performance computing support in embedded and edge environments. Furthermore, FPGAs excel in heterogeneous computing systems by collaborating with CPUs and GPUs to handle complex tasks, meeting the demands of real-time processing and multitasking. However, their application in loading and inference of Transformer models is hindered by issues such as limited storage capacity, insufficient bandwidth, and a complex development process.

Current research primarily focuses on either model compression or hardware acceleration as isolated approaches. These methods often operate independently, lacking a closed-loop framework that coordinates algorithm optimization and hardware resource utilization. To address this issue, this paper proposes a technical approach based on algorithm-hardware co-optimization, enabling synergistic optimization of Transformer model compression and FPGA acceleration. This study will emphasize the unique advantages and challenges of FPGA in deep learning hardware acceleration, supported by case studies demonstrating the effectiveness of co-optimization methods. Additionally, the paper will explore the potential of FPGAs in heterogeneous computing systems and embedded AI applications, offering insights and references for the efficient deployment of AI models in the future.

2. Literature Review

Surface-level features (e.g., word co-occurrence frequencies or simple word vector representations) struggle to capture the deep semantics of language. This limitation is particularly pronounced in tasks requiring semantic reasoning or contextual dependencies. For example, tasks such as word sense disambiguation or the analysis of complex syntactic structures often demand stronger semantic understanding capabilities[1].

Additionally, the cross-domain adaptability of NLP models remains a significant challenge. Current models are often confined to training on specific corpora, making it difficult to adapt quickly to language expressions from other domains. For instance, in specialized fields like finance or medicine, models require the integration of domain knowledge to achieve accurate language processing. Existing models frequently experience performance degradation due to data scarcity or domain transfer, exposing limitations in their generalization capabilities.

To address these challenges, the Transformer model provides a new technological pathway for advancing NLP task performance. Leveraging its robust global modeling capability and efficient self-attention mechanism, the Transformer excels in capturing contextual dependencies and semantic relationships. Pre-trained language models (e.g., BERT and GPT), which learn rich semantic representations from large-scale corpora, have achieved unprecedented performance across numerous

NLP tasks. However, the large parameter sizes of these models impose computational and storage demands, limiting their widespread adoption in real-time applications.

From an algorithmic optimization perspective, the self-attention mechanism, as the core component of the Transformer model, significantly enhances semantic understanding. However, its computational complexity grows quadratically with the sequence length, creating a substantial performance bottleneck for long-sequence processing tasks[2]. Researchers have proposed various methods to optimize the efficiency of the attention mechanism. For example, Shaw et al. introduced relative position encoding to reduce redundant operations in attention computations, significantly lowering complexity while maintaining high performance in translation tasks. Furthermore, techniques such as pruning and quantization have played crucial roles in algorithmic optimization. Pruning removes less important parameters to increase model sparsity and substantially reduce computational costs[3]. Quantization, on the other hand, reduces the precision requirements for weight representations, dramatically lowering storage and computation demands. Despite these successes, challenges such as accuracy loss under high sparsity conditions persist. In particular, unstructured pruning, due to its hardware-unfriendliness, continues to face compatibility issues with hardware platforms[4].

In terms of hardware optimization, FPGAs, with their flexible architectures and efficient parallel computing capabilities, have become important platforms for deploying Transformer models. Compared to traditional CPUs and GPUs, FPGAs can dynamically reconfigure hardware resources, balancing performance and power consumption in embedded and edge computing scenarios. Studies indicate that optimizing sparse matrix storage formats (e.g., Compressed Sparse Row [CSR] and Coordinate [COO]) is a key technique to enhance FPGA compatibility[4]. These formats reduce storage requirements for non-zero elements, improve computational efficiency, and alleviate memory bandwidth pressure[5]. Additionally, researchers have proposed algorithm-hardware co-optimization frameworks[6], incorporating hardware feedback mechanisms to dynamically adjust model sparsity strategies, thereby achieving a balance between performance and resource utilization. This method has demonstrated potential advantages in real-time multi-task processing scenarios, offering technical support for efficient deployment.

However, most existing research focuses on single-task optimization, with limited applicability in real-time multi-task scenarios. Moreover, FPGA platforms face challenges such as limited storage capacity, insufficient bandwidth, and complex development processes when deploying Transformer models. For instance, low-end FPGAs (e.g., ZCU104) offer only 5MB of on-chip memory, while high-end FPGAs (e.g., Alveo U200) provide just 35MB, posing significant challenges for practical model deployment. Optimizing performance within constrained resources remains a critical direction for future research.

Building on the aforementioned research progress, this study centers on algorithm-hardware co-optimization to explore the deployment potential of Transformer models on FPGA platforms. By integrating model compression techniques and hardware optimization strategies, we propose a novel technical approach that emphasizes the applicability of co-optimization methods in multi-task scenarios and their practical effects in embedded devices. The research findings will provide theoretical support and technical guidance for the efficient deployment of Transformer models in the future.

3. Applications and Challenges of FPGA in Artificial Intelligence Tasks

With the rapid advancement of artificial intelligence (AI) technologies, the demand for efficient computing hardware has been increasing steadily. Field-Programmable Gate Arrays (FPGAs), known for their flexibility and parallel processing capabilities, have become a vital hardware acceleration platform for AI tasks. Compared with traditional hardware such as GPUs and TPUs, FPGAs can dynamically configure hardware resources to achieve efficient inference with lower power consumption. This feature makes them particularly well-suited for resource-constrained edge computing scenarios with high real-time performance requirements.

The application scenarios of FPGAs span a wide range of fields, from computer vision to natural language processing. For instance, in real-time video analysis, FPGAs can rapidly perform image processing tasks, reducing latency through pipeline optimization and dataflow design, thereby significantly enhancing processing efficiency. In speech recognition tasks, FPGAs leverage their low-latency and high-throughput characteristics to efficiently process continuous audio input signals. Additionally, FPGAs are widely used in embedded systems, such as autonomous driving and robotic control, where parallel computing and task scheduling ensure real-time performance and stability.

However, deploying deep learning models on FPGAs presents several challenges. First, the large parameter size of Transformer models places significant demands on the on-chip memory capacity and bandwidth of FPGAs. For example, FPGA on-chip memory is typically less than 50MB, far smaller than the memory capacity of traditional GPUs, which constrains the loading speed and real-time inference capabilities of large models. Moreover, FPGA development requires proficiency in complex hardware description languages (e.g., VHDL or Verilog) and optimization toolchains, setting a high technical threshold for developers. Reducing the development complexity of FPGAs while improving model loading efficiency has become a key research focus.

In recent years, techniques such as model compression, quantization, and dataflow optimization have provided effective solutions for FPGA model deployment. Through techniques like pruning and knowledge distillation, researchers have significantly reduced model parameters, thereby alleviating demands on storage and bandwidth. Quantization techniques lower the precision of model weights and activation values, reducing storage requirements to 25% of the original while markedly improving loading speed. Additionally, dataflow optimization and pipeline design further enhance the real-time processing capabilities of FPGAs, enabling efficient model loading and inference in scenarios involving continuous input streams.

Nevertheless, the role of FPGAs in heterogeneous computing systems cannot be overlooked. By collaborating with CPUs and GPUs, FPGAs leverage their respective strengths for different tasks. For example, in edge video analysis scenarios, CPUs handle task scheduling, GPUs perform convolutional computations, and FPGAs focus on preprocessing or accelerating specific tasks. This division of labor not only improves overall system performance but also effectively reduces power consumption.

4. Challenges of Deploying Transformer Models on Traditional Chips

While FPGAs demonstrate unique advantages in AI tasks, comparing their limitations with traditional general-purpose hardware provides a clearer understanding of the technical necessity of FPGAs. However, the massive computational complexity and parameter size of Transformer models pose significant challenges to traditional general-purpose hardware, such as GPUs and TPUs. These challenges are particularly pronounced in edge computing and embedded device scenarios.

The design of traditional general-purpose chips must accommodate a variety of model architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Although this flexibility enhances versatility, it leads to reduced resource utilization efficiency. For example, in the NVIDIA H100 GPU, only about 3.3% of transistors are used for core matrix operations, with the remainder allocated to cache and control logic. This allocation limits efficiency in Transformer inference tasks. The low resource utilization rate makes it difficult for GPUs to simultaneously meet the high-performance and low-power requirements when handling the computation-intensive self-attention mechanism.

5. Deployment Frameworks for Transformers Using Specialized Chips and Algorithm-Hardware Co-Optimization

The deployment of Transformer models in embedded and resource-constrained scenarios faces significant challenges. One existing solution is the development of specialized hardware, such as the Sohu chip, which is fully optimized for the Transformer architecture. By eliminating support for other models, the Sohu chip concentrates most hardware resources on matrix computations. This design allows the Sohu chip to significantly outperform traditional GPUs in resource utilization and inference performance. For instance, an 8×Sohu server can process 500,000 tokens per second (based on the LLaMA 70B model) at lower power consumption, equivalent to the computing power of 160 NVIDIA H100 GPUs. This specialized design greatly enhances inference efficiency, providing robust hardware support for real-time tasks requiring Transformer models.

In addition to improving computational efficiency, specialized chips address several critical bottlenecks in hardware design. For example, the Sohu chip incorporates Continuous Batching technology to effectively alleviate memory bandwidth limitations. Furthermore, its simplified software stack supports only Transformer-related models, eliminating the need for complex CUDA or PyTorch debugging processes and significantly reducing development complexity.

Another promising solution is the algorithm-hardware co-optimization framework. This framework aims to balance performance and resource utilization through a closed-loop approach that integrates model compression techniques with hardware resource allocation strategies.

The core of the co-optimization framework lies in integrating two key components: model compression and hardware adaptation. Model compression reduces the parameter size of Transformer models through methods such as pruning, quantization, and knowledge distillation. For example, Hierarchical Pruning (HP), which combines block pruning and vector pruning techniques, enhances sparsity while maintaining accuracy. Compared to traditional unstructured pruning, this method significantly improves hardware compatibility and reduces the complexity of sparse matrix storage.

On the hardware adaptation side, the co-optimization framework employs a performance predictor to quickly evaluate the resource requirements of different hardware devices (e.g., FPGA, GPU, ASIC). By considering device bandwidth, memory capacity, and computational capabilities, the framework dynamically allocates resources to achieve optimal performance. For instance, in the multi-head attention mechanism of Transformer models, resource parallelism directly impacts inference latency. The framework adjusts computation parallelism factors and dataflow paths based on hardware characteristics, thereby enhancing overall efficiency.

Additionally, the framework supports user-defined latency and accuracy constraints, dynamically selecting suitable hardware devices and model configurations through reinforcement learning techniques. Experiments show that the co-optimization framework delivers excellent performance across various hardware platforms. For example, on FPGA platforms, Transformer models optimized with quantization and pruning achieved a 2.5× speed-up in inference and a 40% reduction in power consumption.

Despite its significant advantages, the complexity of the co-optimization framework introduces new challenges. For example, reinforcement learning-driven optimization strategies demand substantial computational resources, potentially increasing development time. Moreover, the framework’s dependency on model and hardware characteristics may require re-adjustments for new hardware or model architectures.

In summary, the algorithm-hardware co-optimization deployment framework provides robust technical support for the efficient deployment of Transformer models in embedded devices. By integrating model compression and hardware adaptation techniques, the framework not only achieves performance optimization but also establishes a reference paradigm for future AI model deployment.

In the following section, we will further explore the core role of the self-attention mechanism in optimizing Transformer models.

6. Performance Enhancement of Transformers Based on the Self-Attention Mechanism

Building upon the co-optimization framework, a deeper exploration of the Transformer model's core algorithm—the self-attention mechanism—can provide more refined strategies for performance optimization.

The Transformer model relies on its core self-attention mechanism to dynamically model global dependencies, demonstrating exceptional performance in natural language processing (NLP), computer vision, and other fields. However, the computational complexity of the self-attention mechanism grows quadratically with the input sequence length, posing a significant bottleneck when processing long sequences or operating in resource-constrained scenarios. Therefore, reducing computational complexity while maintaining the ability to model global dependencies has become a critical direction for optimizing Transformer performance.

Our approach conceptualizes the input sequence as a labeled directed graph, where nodes represent sequence elements and edges denote the relative distances between elements. By vectorizing these edge relationships and embedding them into the attention weight calculations, the model can naturally capture relative positional information. Specifically, the attention weight computation is modified from the original formula:

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d_z}} \tag{1}$$

to:

$$e_{ij} = \frac{(x_i W^Q)\left((x_j W^K) + a_{ij}^K\right)^T}{\sqrt{d_z}} \tag{2}$$

where a_{ij}^K represents the relative positional information, effectively capturing the relative positional relationships between elements in the sequence.

Relative position encoding has emerged as a critical optimization technique to reduce the computational complexity of self-attention while maintaining its ability to model global dependencies. Unlike traditional absolute position encoding, relative position encoding encodes the relative distances between sequence elements, enabling the model to capture inter-element positional information more efficiently. For instance, Shaw et al. (2018) demonstrated that incorporating relative position encoding into Transformer models led to significant improvements in BLEU scores for machine translation tasks.

Another key optimization direction is the sparse attention mechanism. By limiting the scope of attention computations, sparse attention reduces computational complexity from $O(n^2)$ to $O(nlogn)$ or lower. For example, the BigBird model introduced a combination of global, local, and random attention, achieving efficient computation in long-sequence tasks. Additionally, graph-based attention mechanisms extend the application of sparse attention by representing input sequences as graphs of inter-element relationships, enhancing the model's ability to process structured data.

To adapt to resource-constrained hardware environments, quantization techniques have been widely applied to optimize the self-attention mechanism. By compressing attention weights and activation values into low-precision representations (e.g., 8-bit or lower), the model's storage requirements and computational load are significantly reduced. Moreover, hardware-friendly quantization designs enable FPGA and ASIC platforms to execute self-attention computations more efficiently, providing robust support for deploying Transformer models in embedded scenarios.

Although these optimization strategies significantly reduce the computational complexity of the self-attention mechanism, they introduce new challenges. For instance, sparse attention mechanisms may reduce the ability to capture global dependencies, potentially impacting model performance. Additionally, precision loss during the quantization process requires compensation through calibration and distillation techniques.

7. Future Directions and Conclusion

The Transformer model has become a cornerstone in the field of AI due to its powerful performance. However, its significant computational complexity and storage requirements pose substantial challenges for hardware platforms. This paper has explored the current technical difficulties and solutions surrounding the optimized deployment of Transformer models on hardware and has highlighted future research directions.

Future research should focus on achieving further breakthroughs in the co-optimization of models and hardware. For the Transformer's complex matrix operations and attention mechanisms, techniques such as sparse attention, quantization, and hierarchical pruning can further reduce computational complexity and adapt to hardware resource constraints. In hardware design, specialized chips like the Sohu chip have demonstrated substantial performance improvements. However, their highly targeted designs may lack flexibility. Future research should explore more modular architectures to accommodate the evolving requirements of new AI models.

Additionally, the flexibility and parallelism of FPGAs offer strong support for deploying Transformer models. However, challenges such as high development barriers and limited storage capacity remain significant bottlenecks. Future advancements should aim at developing more intelligent toolchains, combined with pipeline optimization and dynamic resource scheduling techniques, to improve FPGA loading efficiency and inference performance. Verified with FPGA, the industry would expect to see more dedicated hardware such as ASICs to be developed for transformer-centric applications.

In conclusion, The complexity and diversity of natural language processing (NLP) demand continuous optimization of Transformer models in both technical and hardware aspects. By integrating optimizations of the self-attention mechanism with hardware support, NLP models have achieved notable improvements in semantic reasoning capabilities and cross-domain adaptability. This progression not only provides clear direction for future NLP research but also broadens the prospects for real-world applications of Transformer models, paving the way for transformative advancements in AI technologies.

References

[1] *Saini, Vaishali, and Nithin Joseph. "Artificial Intelligence in Robotics Using NLP." (2022).*

[2] *Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani. "Self-attention with relative position representations." arXiv preprint arXiv:1803.02155 (2018).*

[3] *Vadera, Sunil, and Salem Ameen. "Methods for pruning deep neural networks." IEEE Access 10 (2022): 63280-63300.*

[4] *He, Yang, et al. "Learning filter pruning criteria for deep convolutional neural networks acceleration." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.*

[5] *Da Silva, Bruno, et al. "Comparing and combining GPU and FPGA accelerators in an image processing context." 2013 23rd International Conference on Field programmable Logic and Applications. IEEE, 2013.*

[6] *Qi, Panjie, et al. "Accelerating framework of transformer by hardware design and model compression co-optimization." 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD). IEEE, 2021.*