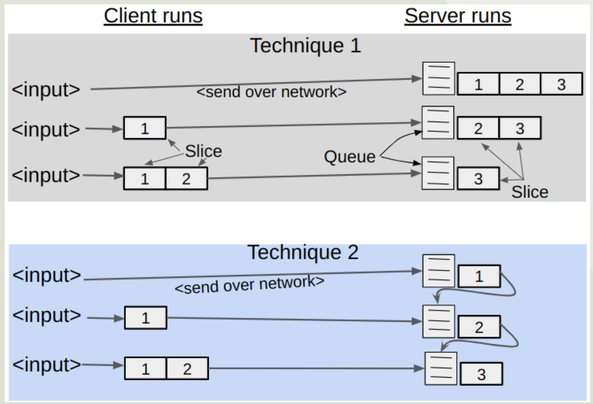
Batching Model Slices for Resource-efficient Execution of Transformer Models in Edge AI

Designed by Waleed Mubark Supervision: Dr. Md Yusuf Sarwar Uddin

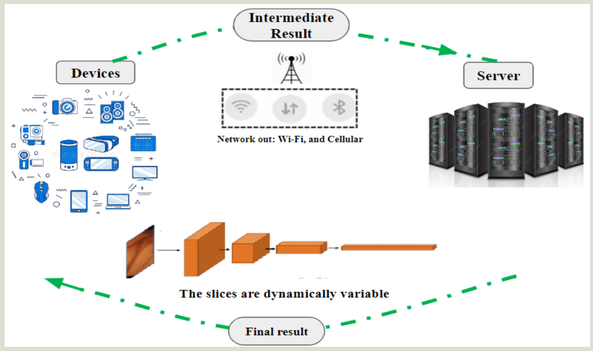
University of Missouri Kansas City - Department of Computer Science

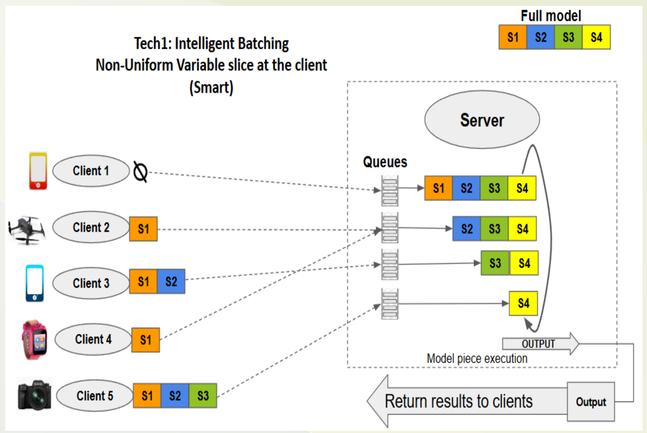
Abstract

Vision Transformer (ViT) models and their variants (e.g., Swin Transformers) have gained popularity for their high accuracy in vision AI applications. However, deploying them efficiently in edge computing environments (e.g., mobile phones, embedded platforms) is challenging due to their high computational demands. To address this, we propose a novel model slicing and smart batching approach to distribute workloads between resource-constrained clients and powerful edge servers. Model slicing partitions a model into smaller slices, allowing clients to execute initial head slices, while nearby edge servers handle the remaining tail slices. Smart batching enables the server to queue multiple client requests for efficient inference and improved GPU utilization. We introduce two batching strategies: one prioritizes speed but requires more GPU memory, while the other reduces memory usage with minimal overhead. Experimental results show that our approach reduces inference time by up to 67%, enhances GPU efficiency, and significantly improves inference speed, proving its effectiveness for distributed AI systems.

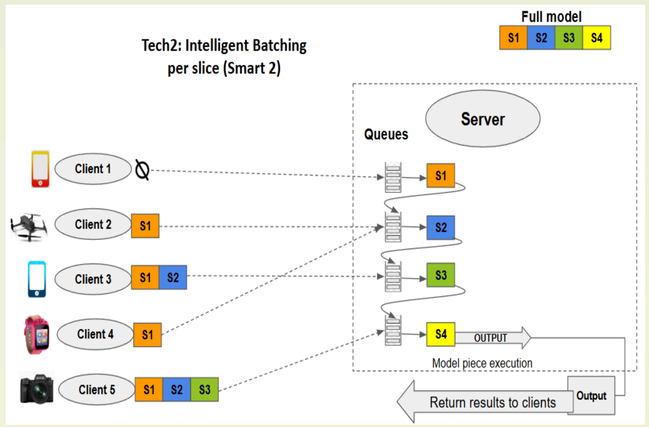


Two batching techniques proposed: A large model is partitioned into three slices, denoted as 1, 2, and 3. Clients run a few slices, sends out intermediate output to the server, and the server runs the remaining slices. The server batches requests for the same model slices. Technique 1 loads the entire server-side model slices as a single piece, whereas Technique 2 keeps separate a queue per slice. As per, Tech 1 loads six slices for serving whereas Tech 2 loads only three





Tech1: Smart batching architecture demonstrating dynamic slicing and batching.



Tech2: Architecture of model batch per slice.

Statement of the Problem

1. Transformer-based deep learning models are computationally demanding, making full deployment on edge devices.
2. Traditional solutions (full offloading to a server or fixed model partitioning) cause inefficiencies in computation, latency.
3. Edge environments are heterogeneous, with devices having varying computational capabilities.
4. A static slicing approach fails to adapt to different device capabilities and network conditions.
5. The proposed solution introduces dynamic batching for model slices, enabling clients to process initial layers locally and offload the rest to the server.
6. This approach optimizes inference performance, reduces latency, and improves resource utilization in edge AI settings.

Methodology

Proposed Framework

Our framework optimizes distributed inference for transformer models using adaptive slicing and batching strategies. It consists of: Model Slicing & Batching: Efficiently processing inputs on the server.

1. Model Slicing Between Client and Server

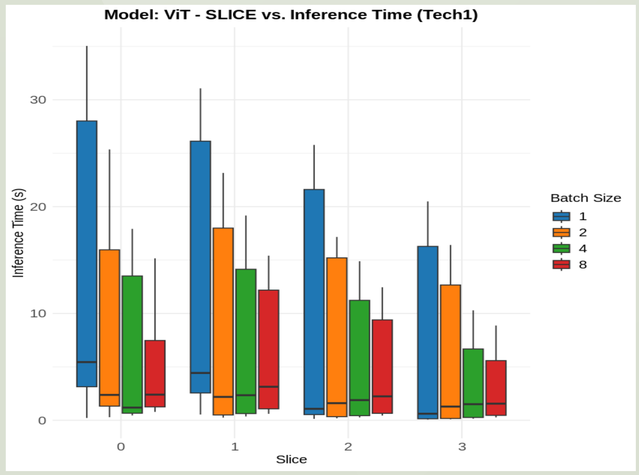
•The client preprocesses data, selects a model, and processes the head slice, and Intermediate outputs and split indices.

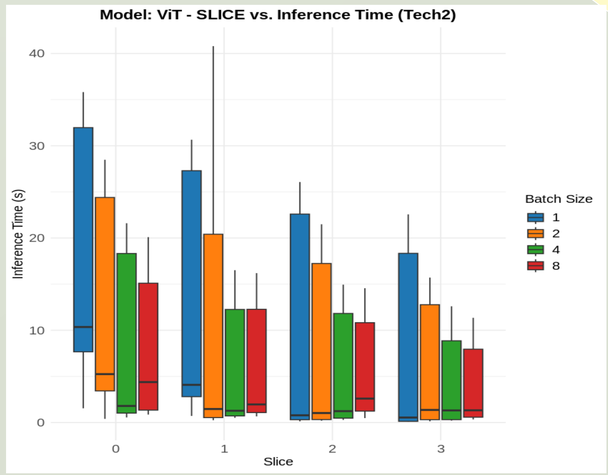
•The server processes the tail slice and returns final results and reduces client workload, offloading heavy computation.

2. Batching Inputs on the Server

•The server batches inputs from multiple clients.

•One technique loads the entire tail segment into GPU memory for faster inference but requires more GPU resources.

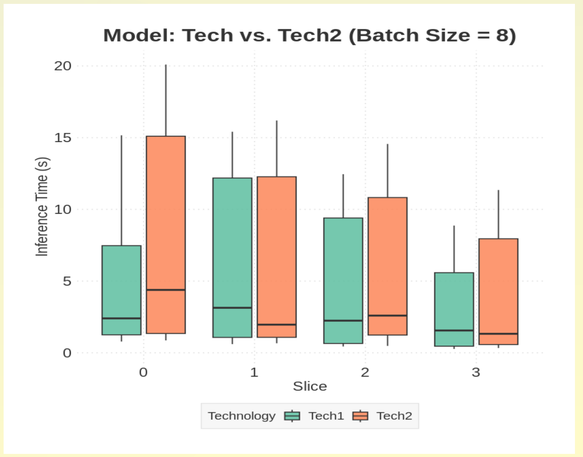




Observation and Results

The results, as depicted in Figure below compare the inference times of Vision Transformer (ViT) and ViT Large models using Tech1 and Tech2. The plot illustrates the relationship between batch size and inference time, demonstrating how each technique performs under varying conditions.

These results highlight the importance of selecting the appropriate technique based on the batch size and model architecture to optimize inference time. Tech1, while faster, requires more memory, making it suitable for systems with higher RAM availability. On the other hand, Tech2, though slower, is more memory-efficient, making it ideal for resource-constrained environments.



The plot shows the relationship between slice size and inference time in batch size=8, for ViT and ViT Large using Tech1 and Tech2.

Conclusion

The achieved model proves great potential for proposes a framework combining smart batching and model slicing to optimize inference time and GPU utilization for Vision Transformers (ViT). By distributing workloads between clients and servers, it enhances throughput, resource efficiency, and scalability, making it ideal for real-time applications.

References

1. Shengwei Gu, Xiangfeng Luo, Xinzhi Wang, and Yike Guo. Accelerating multi-exit bert inference via curriculum learning and knowledge distillation. International Journal of Software Engineering and Knowledge Engineering.
2. Seyed Morteza Nabavinejad, Masoumeh Ebrahimi, and Sherief Reda. Throughput maximization of DNN inference: Batching or multi-tenancy, 2023.
3. Jinghui Zhang, Weilong Xin, Dingyang Lv, Jiawei Wang, Guangxing Cai, and Fang Dong. Multi-exit dnn inference acceleration for intelligent terminal with heterogeneous processors. Sustainable Computing: Informatics and Systems, 40:100906, 2023
4. Kaihua Fu, Jiuchen Shi, Quan Chen, Ningxin Zheng, Wei Zhang, Deze Zeng, and Minyi Guo. Qos-aware irregular collaborative inference for improving throughput of dnn services. In SC22: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2022.