Paper Review - AdaInf: Data Drift Adaptive Scheduling for Accurate and SLO-guaranteed Multiple-Model Inference Serving at Edge Servers

Sai Venkat Malreddy

10 oktober 2024

1 Idea and Contribution

AdaInf proposes a complete scheduler that comprises of retraining and serving the incoming inference request at edge servers with in SLO at Edge Servers. The authors of the paper contributed to solving the problem of adapting to the new data and scenarios of evolving test beds in real life with managing the resources of edge servers effectively. Some of the main contributions by the authors in this paper are incremental retraining, implementing early exit strategies, improving(minimization) of the CPU-GPU memory communication, data drift aware GPU space and time allocation based on the offline profiling of various specific tasks (batching, GPU space and time allocations).

2 Positive Comments

- **Addressing the need for the data-drift aware applications**
  In real world, the data, scenarios always evolve and impossible to create a model to satisfy this and also using a edge server to run them. So utilizing the resources of edge server(less compared to cloud(larger)) to retrain, capture the impact degree for the need to retrain is definitely required for every publicly deployed models.
- **Effective utilization of resources - Early-Exit, Optimal Batch Size, GPU space and time allocation**
  As the resources of the edge servers is very less compare to cloud, the authors implemented various strategies like Early_Exit stages in model while serving the inferences, creating a optimal batch size based on the offline profiling and GPU space while reducing the worst case latency is crucial for managing the resources effectively and maintain the SLO.
- **DAG-Based Multi-Model Support**
  Unlike other approaches that primarily address single-model applications, AdaInf is built to handle the complexities of multi-model applications. In these applications, multiple deep learning models are arranged in a Directed Acyclic Graph (DAG), where the output of one model can be used as input for another.

3 Negative Comments

- **Need for Extensive Offline Profiling**
  AdaInf relies heavily on offline profiling to determine the best batch sizes, GPU allocations, and early-exit strategies for each application.

- **Lack of Support for Heterogeneous GPU Resources**
  The paper assumes that all retraining and inference tasks are performed on a homogeneous set of GPUs.

4 Questions unanswered about the paper

- **Task Prioritization in DAG Execution**
  The paper introduces a Directed Acyclic Graph (DAG) for multi-model applications, but more clarity is needed on how task prioritization is handled when multiple models in the DAG compete for the same GPU resources. How does AdaInf ensure critical tasks with tighter SLOs are prioritized over less urgent tasks, and how does it handle task dependencies efficiently? The paper could offer more insight into the underlying scheduling algorithms used for DAG execution under varying workloads

- **Scalability Across Multiple Edge Servers**
  The paper focuses on the efficiency of a single edge server but does not discuss how AdaInf could be extended to scale across multiple edge servers. A more detailed exploration of task distribution, load balancing, and coordination between servers would be helpful in understanding how the system performs in larger, distributed environments