Team Falcons: Research Summary

Empowering Edge Systems: Distributed Machine Learning in Heterogeneous Environments

Raviteja Siginamsetty, Satish Babu Malempati, Vamshi Krishna Pathem, Narasimha Niteesh Chedalawada.

Introduction:

Our research explores the development of AI applications on edge devices, such as PCs, smartphones, and IoT devices, which locally train machine learning models for services such as smart surveillance and Industry AI Operations (AIOps). This distributed training method is critical for handling sensitive or extensive data locally without the inefficiencies and privacy concerns of central cloud processing. Instead of transmitting all data to a central server, edge devices compute model updates locally and share only parameter updates, significantly reducing bandwidth requirements and enhancing data privacy.

Addressing the challenge of heterogeneous computing capabilities among edge devices, this research introduces the Adaptive Synchronous Parallel (ADSP) model. ADSP improves upon traditional synchronous and asynchronous parameter synchronization methods by allowing faster devices to process continuously without waiting, while still ensuring periodic synchronization to maintain consistent model convergence. This method enhances resource utilization and accelerates the training process, offering a scalable and efficient solution for distributed machine learning in diverse edge computing environments.

Existing Methods:

The existing methods deals with the efficiency of various parameter synchronization models in distributed machine learning, highlighting three key types: **Bulk Synchronous Parallel (BSP)**, **Stale Synchronous Parallel (SSP)**, and **Totally Asynchronous Parallel (TAP)**. BSP, while ensuring model convergence, operates under stringent synchronization constraints that can slow down the training process as all nodes must wait for each other. SSP improves on BSP by allowing a degree of asynchrony; faster nodes can move ahead but must remain within a set lag of the slower ones, balancing speed with convergence assurance. In contrast, TAP allows complete asynchrony among workers without any convergence guarantee, which can lead to inconsistencies across the model's state.To optimize these synchronization strategies in systems with heterogeneous computing capabilities, the passage introduces the ADSP model and its innovative online search algorithm.

This algorithm dynamically adjusts commit rates for each worker based on real-time training conditions, effectively minimizing idle times and optimizing overall system performance. By balancing the commit rates according to the ongoing results and conditions, ADSP aims to significantly reduce convergence times and stabilize training, even in environments where the computing power of participating devices varies widely.

ADSP:

The ADSP (Adaptive Synchronous Parallel) model, a new parameter synchronization method designed for distributed machine learning using heterogeneous edge systems and a central parameter server (PS). ADSP aims to maximize the computational efficiency of each worker and balance the trade-off between hardware utilization and statistical efficiency to minimize the time required for model convergence. It also ensures that the model converges correctly despite varying training speeds and bandwidth constraints across different workers. In ADSP, training time is segmented into equal-sized intervals called check periods, marked by checkpoints. The process whereby a worker sends its computed gradients to the PS is termed a commit, and the frequency of these commits during a check period is defined as the commit rate.

The system comprises two main modules:

1) A novel synchronization model that allows faster systems to train more before each update, while ensuring uniform commit rates across all workers.
2) A global commit rate search algorithm that determines the optimal commit rate for all workers to facilitate rapid convergence.

Operationally, each worker calculates the number of commits it should have made by each checkpoint and adjusts its commit rate accordingly for the next period. Workers process data in mini-batches, updating the model parameters locally and then sending cumulative updates to the PS. The PS then integrates these updates into the global model using a global learning rate. This method ensures that all workers are contributing evenly and efficiently to the model training, regardless of their individual processing speeds.

Conclusion:

Through the implementation of an online search algorithm, ADSP dynamically identifies an optimal commit rate, optimizing the use of computational resources across diverse system capabilities and significantly speeding up the convergence process. Testbed experiments demonstrate that ADSP can accelerate convergence by up to 62.4% compared to traditional parameter synchronization models. The adaptability and efficiency of ADSP make it particularly suitable for large-scale machine learning applications and varying degrees of system heterogeneity.