1. Multi-Task Federated Learning

Multi-Task Federated Learning
- Different Learning tasks with Imbalance Datasets
- Heterogeneous Models
- Learning Speed Multi-Task Synchronization

Multi-Task Deployment with Parallel Model Structures

Challenges
- Multi-Task Parallel Computing Deployment
- Different Learning Speed Multi-Task Coordination

We propose a full-stack multi-task FL optimization scheme:
- intra-device GPU scheduling with a competitive resource sharing scheme;
- inter-device FL coordination with realistic GPU runtime synchronization.

2. Intra-Device Multi-Task Resource Allocation

Multi-Task Parallel Deployment Mechanism

How to achieve effective multi-task dedicated GPU resource allocation.

Challenges:
- Identify the parallel computing issue;
- Establish the relationship between performance and resource allocation.

Comprehensive Analysis with Competitive Resource Sharing Cases under unification

Isolated Spatial Resource Allocation
Exclusive resource assignment causes under-utilization.

Spatial Sharing with Light Competition
Higher resource allocation flexibility and better resource utilization.

Sharing with Excessive Competition
Leading to resource competition and considerable contention overhead.

Extreme Contention Kills Parallelism
Push back to temporal scheduling when fully sharing the resource without partitioning.

Optimization Insights!
Controlling an appropriate degree of resource competition and sharing is the key to achieving the optimal performance.

“Virtual Resource” management to establish the relationship between performance and resource allocation.

Definition of Virtual Resource:
Virtual resource is a number between (0%~100%) x l (l of tasks).

As shown in the right figure, the four points correspond to the above four cases.

We propose a machine learning approach to estimate the GPU throughput P and achieve the maximum throughput.

3. Inter-device Multi-Task FL coordination

Coordination Design Motivation:
We rethink the FL coordination from a GPU scheduling perspective under imbalance dataset, model workload and resource allocation.

Our Objectives:
1. Each device could be fully utilized during each synchronization cycle;
2. Maximize the overall GPU throughput R.

Tri-party Optimization to Achieve Objective 1:
1. Larger Dataset D need larger Workload O;
2. Larger Workloads O compete for more Resources;
3. More Resources train more data each cycle.

4. Experimental Results

We can apply our resource allocation method on various multi-task training scenarios under multiple DNN models’ combinations.

Intra-Device Multi-Task GPU Throughput

We use a FL system with several devices, each device has three tasks with different model structure and imbalance data volume.

Joint optimization of workload and resource allocation allows our method to utilize GPU resources during the whole synchronization cycle.

5. Contribution and Conclusion

- We analyze the competitive resource sharing mechanism propose an intra-device multi-task dedicated GPU scheduling method.
- We further bring competitive resource sharing mechanism into the inter-device FL cluster and rethink the FL coordination from a GPU scheduling perspective.
- Beyond some FL methods which resolve the heterogeneous problems from algorithmic level, FL’s heterogeneous research should also dive into computation level due to imbalance dataset, heterogeneous model and different learning speed per task.

6. Reference