Distributed Training Overview

Why Distributed? Single GPU insufficient for large models/datasets

Data Parallelism

M M M

Same model on multiple GPUs, different data batches

Model Parallelism

M_1 M_2 M_3

Split model across GPUs (layers or tensor sharding)

Synchronous Training

Common

All GPUs sync gradients each step. More stable, consistent results.

Asynchronous Training

GPUs update independently. Faster but less stable convergence.

PyTorch Standard Implementation

DistributedDataParallel (DDP) for multi-GPU training

`torch.nn.parallel.DistributedDataParallel`

Communication

All-reduce operation for gradient synchronization across GPUs

Efficiency

Linear speedup ideal (2 GPUs = 2x), but overhead exists