CSE 5449 report #2

1. A. DDP for PyTorch Distributed:

**Model Parallel**: Model need to replicate once, and the model could be split into several GPUs. In this way, when DDP could use model parallel, all processes would use their model parallel separately.

**Data Parallel**: When data would be divided into different batches, the different GPU for the same model would run these data. In this way, the epoch would spend less time to finish it.

**DistributedSampler**: The DistributedSampler would make sure that each device could get its own non-overlapping input batch.

DistributedDataParallel uses data parallels to run across different devices or GPUs. By using the DDP, the multiple devices or GPUs could use the same model with different dataset to synchronize gradients and buffers. This distributed communication could support many kinds of communication patterns.

Data Parallel

Model Parallel

Distributed Data Parallel

PyTorch

NCCL

MPI

Gloo

B. Horovod for PyTorch Distributed:

**Ring all-reduce algorithm**: By using ring all-reduce algorithm, the model could operate efficient inter-GPU communication. In this way, the latencies between GPU would be short.

**Gloo**: Gloo is an open source communications library made by Facebook.

**Mpi4py**: This is a python library which is built based on mpi. It could help users to do the collective or point to point communication between multiple GPUs.

Horovod

PyTorch

Mpi4py

Gloo

2.