Tutorial. Nov 18, 2024 1:30-5 pm

Distributed Deep Learning on GPU-based Clusters

Abhinav Bhatele, Siddharth Singh, Prajwal Singhania
Department of Computer Science

PARALLEL SOFTWARE AND SYSTEMS GROUP

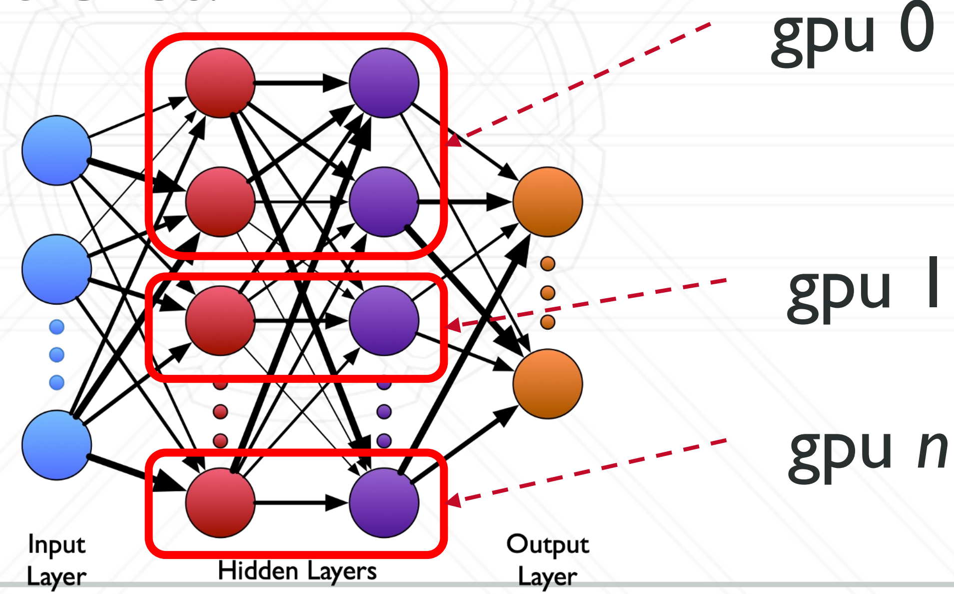
UNIVERSITY OF MARYLAND

Limitations of data parallelism

- DDP – Supports models of limited size
- FSDP – Large communication overheads, specially for small batch size tasks like IFT

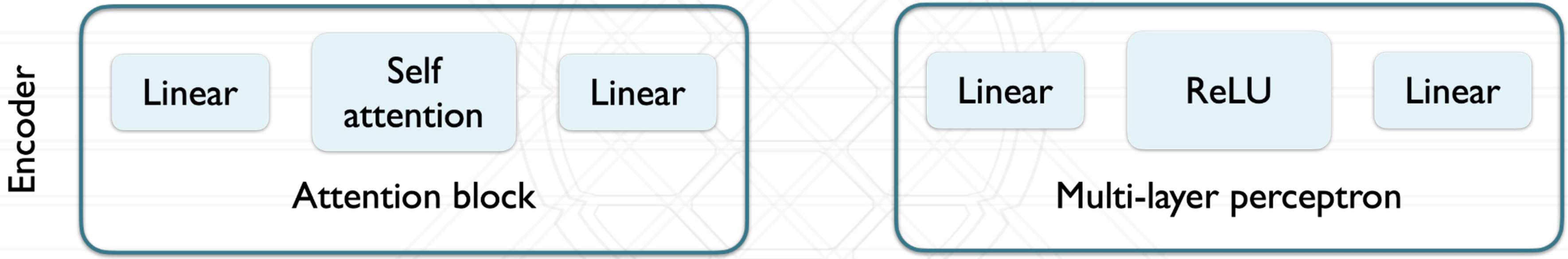
Tensor parallelism

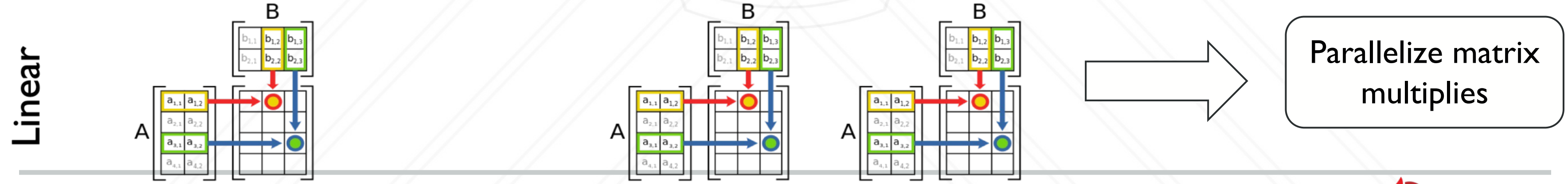
- Divide parameters and compute of every layer of a neural network on multiple GPUs.



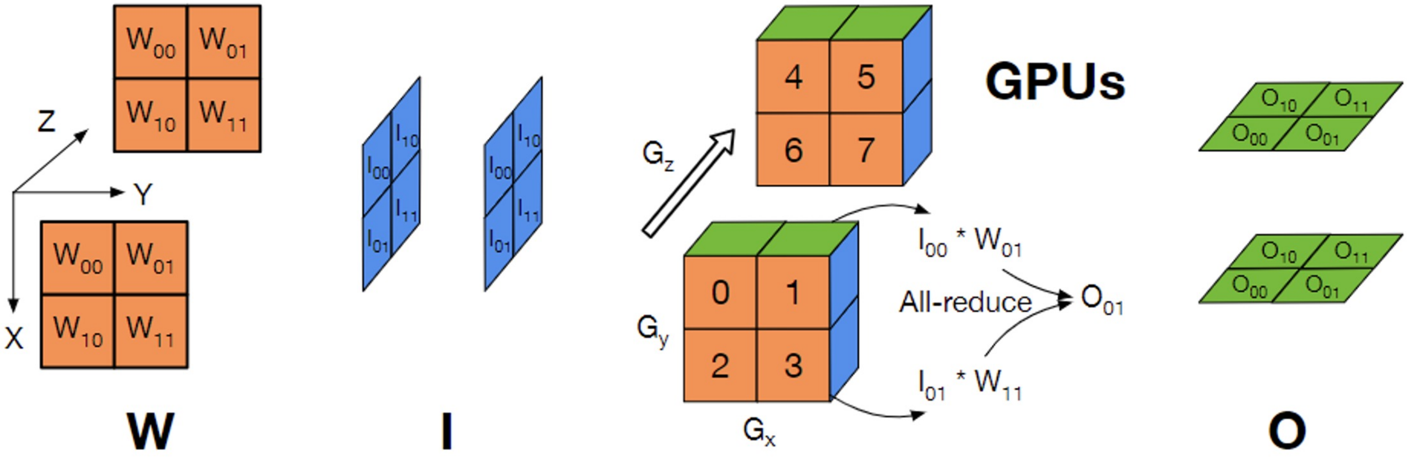
Tensor parallelism

- Divide parameters and compute of every layer of a neural network on multiple GPUs.





AxoNN's 3D Tensor Parallelism



Parallelizing a matrix multiplication (I.W=O) using AxoNN on 8 GPUs

Creating an AxoNN Lightning Strategy

```
from axonn.lightning import AxonnStrategy

pl_strategy = AxonnStrategy(
                G_intra_x=..
                G_intra_y=..
                G_intra_z=..
                overlap_communication=True,
)
```

3D tensor parallel grid dimensions

Running the code (Tensor/Intra-Layer)

- Code – `train.py`

```
CONFIG_FILE=configs/axonn.json
sbatch --ntasks-per-node=4  train.sh
```

Let's try different AxoNN configurations

- In config/axonn.json - tp_dimensions is [2, 2, 1].
- Now change it to [4, 1, 1] and rerun your code.
- Does it become faster?

How to set the parallel configuration?

- Small batch-sizes - Use x and/or y
  - Example - finetuning and inference
- Large batch sizes - Use z (+data parallelism)
  - Example - pretraining

Alternative way to use AxoNN

```
from axonn.intra_layer import auto_parallelize
with auto_parallelize():
    net = FC_Net(args.num_layers, args.image_size**2, args.hidden_size, 10).cuda()
```

Zero code changes required in your model definition!

fabric.init_module() calls AxoNN's auto-parallelize function