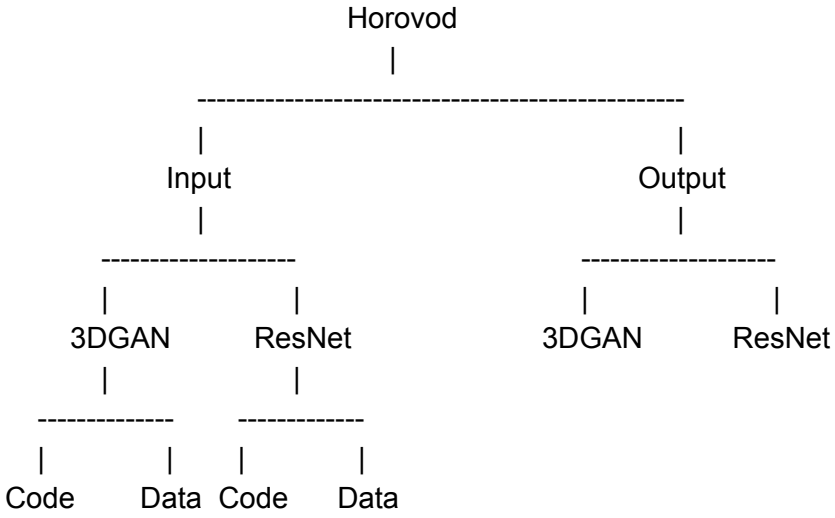
The Horovod section includes two problems: 1) 3D Convolutional GAN for high energy physics simulation (3DGAN) and 2) image classification with ResNet-50 on ImageNet dataset (ResNet).

1. Running instruction

1.1 Data Layout

The Horovod directory has the following structure:



Input/3DGAN/Code: the keras+tensorflow+horovod implementation of 3DGAN

Input/3DGAN/Data: dataset in the form of 20 hdf5 files

Input/ResNet/Code: the keras+tensorflow+horovod implementation of ResNet-50

Input/ResNet/Data: the imagenet dataset in files, please run “tar xf imagenet-1k.tar” to set the dataset ready

1.2 Software Version

The versions of software are recommended:

Keras: v2.2.2

TensorFlow: v1.8.0

Horovod: v0.13.4

1.3 Output

The output of each applications includes two parts of results: the trained model and standard output. The trained model of 3DGAN will be placed into Input/3DGAN/Code/3Dgan/keras/weights2D/ as discriminator_params_generator_epoch_029.hdf5 and generator_params_generator_epoch_029.hdf5.

The trained model of ResNet-50 will be placed into Input/ResNet/Code/checkpoint-90.h5. Please copy these model files to the respected Output directories.

To collect standard output, please attach the following string to your launching command "2>&1 | tee stdout.log".
For example, if you run "mpiexec.hydra -f hostfile -np 16 -ppn 4 python train_hvd.py" to launch training, now your have to run
"mpiexec.hydra -f hostfile -np 16 -ppn 4 python train_hvd.py 2>&1 | tee stdout.log".

Please upload the trained model and stdout.log without human changes to the Output/ directory accordingly.

1.4 Hyperparameter Tuning (What is allowed and what is not)

The principle of the Horovod problem is that only Horovod source code is considered as source code. So any code change in Horovod is allowed.

Given the variety of hardware of each team, we also allow limited hyper-parameter tuning with batch size and learning rate. All other hyperparameters tuning are not allowed.

1.5 Sample command lines

1.5.1 3DGAN

mpiexec.hydra -f hostfile -np 16 -ppn 4 python EcalEnergyTrain_hvd.py --datapath=/Input/3DGAN/data/EleEscan_*.h5 --batchsize 128 -lr 0.001 --optimizer=Adam --latentsize 200 --warmup 0 --nbepochs 30 --analysis=True 2>&1 | tee stdout.log

1.5.2 ResNet-50

mpiexec.hydra -f hostfile -np 16 -ppn 4 python keras_imagenet_resnet50.py 2>&1 | tee stdout.log

2. Score rubrics

2.1 Score Distribution

3DGAN -- 30%
ResNet-50 -- 50%
Interview -- 20%

2.2 Score Breakdown

In general, we score the results based on both model performance (test accuracy or error) and the time-to-solution. Model performance is preferred over time-to-solution.

2.2.1 3DGAN

We need one runs of 3DGAN: one on a single node and the other on the largest scale of each team.

1) Training for 30 epochs on multiple nodes -- 9%
2) Loss on multiple nodes within acceptable range -- 12%
3) Time-to-solution with 2) -- 9%

2.2.2 ResNet-50

Only a single run is required for ResNet-50. Run at the largest possible scale.

1) Training for 10 epochs -- 12%
2) Test accuracy within acceptable range -- 16%
3) Time-to-solution with 2) -- 12%
4) Training for 90 epochs with acceptable test accuracy -- 10%