Accelerating Graph Neural Network
Work so far

- Environment Setup
- Code & paper review
- GNN Training

TrackML dataset ~ HL-LHC silicon
https://indico.cern.ch/event/831165/contributions/3717124/
GNN Training Result

Result
Accuracy: 0.912819
Precision (purity): 0.887547
Recall (efficiency): 0.976627

In train_edge_classifier_kaggle_share.yaml
layers: [7, 8, 9, 10, 24, 25, 26, 27, 40, 41]
endcaps: true
real_weight: 0.33
fake_weight: 1.00
Queries

- Where does the bottleneck in the conversion of deep learning model lies? Is it because of latency, or memory or conversion time of trained deep learning model to HDL code?
- In the NeurIPS 2019 paper, the GNN model is trained on two sub-detector regions in the endcaps having 7 layers, one layer in the barrel of TrackML detector. Is this consistent with the settings?
- On page 9 of slides presented in the previous meeting, why is CTD 2020 paper linked rather than NeurIPS 2019 paper?
- In the last meeting, there was a discussion about breaking into subgraphs and then performing parallel computation. Wouldn’t there be an impact on edge efficiency, which is an important metric GNN efficiency?
- Why are fake weights greater than real weights?