Implementation of graph neural networks on CPU + FPGA co-processors for scalable track reconstruction tasks

Aneesh Heintz, Cornell University (ah2263@cornell.edu)
PI: Isobel Ojalvo, Princeton University

June 15, 2020

1 Background

Run 2 of the LHC on the CMS detector produced a data rate on the scale of hundreds of terabytes per second [1]. Being able to reduce the data within a few milliseconds and sift through the data in a reasonable time frame to produce meaningful results is crucially important. Future increases in instantaneous luminosity, meaning more proton-proton collisions per bunch-crossing, will lead to data produced at increasingly larger rates, causing scalability issues in traditional particle track reconstruction algorithms. The operations in tracking algorithms scale exponentially with input data size. Trained machine learning models can address this problem more efficiently as the online evaluation process requires far fewer operations.

The Level-1 trigger work being done in Dr. Ojalvo's lab at Princeton University uses machine learning to replace the cpu-intensive parts of particle track reconstruction algorithms in an attempt to remedy this inference time problem. Deep neural networks scale linearly with input data size [2] and are able to learn highly non-linear representations of the data. The data resulting from proton-proton collisions is inherently geometric and graphs as a mathematical object are good at representing them. The track reconstruction problem can therefore be considered as an edge classification task. By representing the flexible geometric data as graphs, Graph neural networks (GNN) are specially designed deep neural networks to work more efficiently and effectively than more standard types of neural networks.

2 Proposed Project

As part of this initiative, this project proposes to implement a graph network that can be evaluated on a CPU that has a FPGA co-processors. This will allow trained networks to be run online in a highly parallelized fashion, greatly accelerating data throughput. Work conducted over the next three months will start by becoming familiar with the CPU + FPGA co-processor. This involves repeating some of the work that has been done on the FPGA co-processors to date and verifying that the graph network architecture size is suitable for implementation. Following this initial investigation, I will expand on what has been done and scale up the data sizes, focusing on the message passing aggregation part of the GNN process. The project's end goal is to extract the message passing aggregation part and successfully run it on the FPGA co-processors. Along the way, I will investigate the effects scalability has on the co-processors. This involves studying data transfer rates and the load the networks have on them. Additionally, I plan to test the effects different types of networks have on the co-processors, again examining efficiency and scalability metrics. All research will be done remotely and in collaboration with Dr. Ojalvo's group.

3 Deliverables

- Creation & Successful implementation of the message passing aggregation part of the GNN process.
- An analysis of co-processor performance with a range of data input sizes and network types / architectures.

4 Timeline

Week(s)	Activity
1-3	Project setup; Familiarization with existing setup & repeating work already done on the co-processors.
4-5	Investigate data scalability.
6-9	Design & implement message passing aggregation part of the GNN process on co-processors.
10-11	Study effects different network architectures & data scalability has on co-processors.
12	Prepare findings.

5 Student Background

My interest in this work stems from the experience I have gained conducting research involving point cloud data and graph neural networks. My PhD work is in aerospace engineering; specifically, the development and implementation of deep learning algorithms for spacecraft. However, my interest in the proposed project stems from the significant overlap with high-energy physics in developing scalable deep learning algorithms and their practical implementation on different hardware architectures to efficiently analyze large data sets. My research so far has resulted in significant work related to the use of point clouds and graph neural networks (see CV for more details). The proposed project will allow me to learn to use FPGAs and how to implement a neural network on them while working on a meaningful problem. The knowledge learned will be used during the rest of my PhD work to demonstrate the practical use of deep learning methods using FPGAs in a space exploration context.

References

[1] A. Klimentov, M. Grigorieva, A. Kiryanov, and A. Zarochentsev. 12(06):C06044–C06044, jun 2017.

[2] Steven Farrell et al. The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking. EPJ Web Conf. , 150:00003, 2017