pyhf Hardware Acceleration Benchmarking with GPUS and TPUs

Project Description

This project aims to build a benchmarking suite, written as a pytest module, in the `pyhf` library to test and compare the performance of all the pyhf computational backends in fitting openly published likelihoods from the LHC [10.17182/hepdata.89408, 10.17182/hepdata.92006]. The `pyhf` library is a pure-python implementation of that statistical model for multi-bin histogram-based analysis. Its aim is to support modern computational graph libraries such as PyTorch and TensorFlow to make use of features such as auto differentiation and GPU acceleration.

In this project, functions to test and benchmark the performance increase of the hardware-accelerated backends on GPUs and TPUs are needed. For GPUs, I will firstly determine the key metrics to test and then write a benchmarking test suite for different backends. Currently, testing and comparing the outputs of `pyhf` and `roohf` is difficult. So, it is essential to enable test bench suite to generate `pyhf` JSON or generate `roohf` XML + ROOT and then workspace via hist2workspace. In this case, the user can evaluate the performance of hardware acceleration without analyzing visually or manually. As for TPUs, there exist some weird effects on TPUs (such as the performance of TPUs is slower than GPUs over Google Colab). More research is needed for the performance of TPUs. Besides, the benchmarking suite needs to be feasible for new data. In the end, detailed documentation over the evaluation of the performance of hardware-accelerated `pyhf` on GPUs and TPUs and a report for the pyhf website will be useful.

I propose to work under the mentorship of Matthew Feickert, Lukas Heinrich and Giordon Stark. I believe I can complete the project in three months because of my previous research experience in machine learning and big data. I am familiar with programming languages like Python, C++ and machine learning tools like Tensorflow, Keras, NumPy, Pandas. I took Statistics of Machine Learning(Python), Approximate Computing System for Big Data (Python) at Rice University, which can help me have a better understanding of the project and implementation of the work.

Schedule of Deliverables

Week 1

- Read developer documentation and get accustomed to the general community environment
- Communicate with community members about the project idea and its implementation and learning user experience from data analysts who use Data Retriever
- Get a head start

Week 2 - 3

- Implement a hardware acceleration(GPUs) benchmarking suite for Tensorflow

Week 4 - 5

- Implement a hardware acceleration(GPUs) benchmarking suite for PyTorch

Week 6 - 7

- Implement a hardware acceleration(GPUs) benchmarking suite for JAX

Week 8 - 9

- Do research related to TPUs
- Evaluate the hardware acceleration test suite using Google TPUs.

Week 10 - 11

- Write comparison plot generation code in Python for the performance of the GPU enabled backends against each other and the CPU backends.
- Evaluate the hardware acceleration test suite using Google TPUs.

Week 12

- Submit sample code to Mentors
- Document the performance benchmarking in the form of a case study.
- Profile the pyhf codebase