Contribution ID: **104** Type: **Poster**

Dynamic Scheduling Support for Faster ML Inference in hls4ml

FPGAs are performant and flexible microchips well-suited for experimental physics that efficiently run anomaly detection algorithms and identify potential new physical phenomena. However, FPGAs are not easy to program: A significant gap exists between the algorithms used to discover new physics and the low-level hardware description languages (HDLs) required to program FPGAs. To tackle the absence of hardware expertise, collaboration between physicists and hardware designers led to the creation of the hls4ml project [1]: well-calibrated machine learning (ML) models—the core components in the algorithm used in experimental physics—are automatically translated into C code and embedded in the whole algorithm, which high-level synthesis (HLS) tools can then use to generate HDL.

The efficiency of the HDL code critically depends on how well the HLS tools can implement the underlying algorithm: hls4ml relies on HLS tools that use static scheduling, i.e., a predefined and fixed schedule dictates computations. These HLS tools are well-suited for problems with definite sizes and dimensions. Despite that, the computation involved in ML inference is usually highly regular; however, real-world events are often sparse and dynamic: certain operators may be conditionally bypassed, batch sizes can vary unpredictably, and data is frequently represented in a sparse format. Unfortunately, in the presence of variability, pipelining—the most prominent HLS optimization technique for fast and efficient hardware—is not achievable in static scheduling.

Dynamatic is an open-source HLS compiler that generates dynamically scheduled dataflow circuits [2]. Dataflow circuits deliver the best performance in the presence of variability: the circuit does not need to follow a predefined computation schedule that is designed for the worst-case scenario; computations progress as soon as the data is available. Dynamatic has been successfully used to demonstrate a performance advantage over traditional, statically scheduled HLS tools on conditionally skipped operations, unpredictable workloads, or sparse linear algebra.

This poster presents our ongoing effort in adding the Dynamatic HLS compiler as a backend of hls4ml. The purpose is to enable scientists to develop more dynamic and irregular algorithms and, ultimately, broaden the applicability of hls4ml. Our poster will discuss the following elements:
- We give a general overview of Dynamatic, as well as its strengths and weaknesses.
- We show that Dynamatic-produced implementations of common machine learning operators are on par with those produced by other HLS tools.
- We discuss our adaptations to hls4ml to emit HLS C code that is compatible with Dynamatic.
- We explore the benefits of dynamic scheduling in an anomaly detection algorithm, and compare the implementation quality with those achieved by existing hls4ml backends.

We hope that our poster will be of interest to users and developers of hls4ml, and we look forward to valuable interactions and feedback from the Fast ML community.

References
- [1] Duarte et al., JINST ’18
- [2] Josipovic et al., FPGA ’18

Authors: Mr GIRJOABA, Andrei (ETH Zurich); Ms KOSTIĆ, Andela (ETH Zurich); XU, Jiahui (ETH Zurich); Prof. JOSIPOVIĆ, Lana (ETH Zurich)

Presenter: Mr GIRJOABA, Andrei (ETH Zurich)

Session Classification: Posters and coffee