Adaptive Random Forest on FPGA

Along with the demand of real-time applications, e.g., pedestrian detection, IP address search, or nanopartical sensors, learning from evolving data streams establishes a new fashion of machine learning, at which the boarder between training and inference becomes blur. Storing data prior to learning in a stationary manner is no longer feasible, since the concept of the data might drift dynamically and the underlying devices at edge might not have sufficient memory to store massive data for offline training. One prominent learning model is Adaptive Random Forest (ARF), which combines Hoeffding trees as base learners and a drift detection operation. When a drift is detected, some candidate trees will be replaced with new Hoeffding trees while providing theoretical guarantees on accuracy. However, differing from the conventional random forest and decision trees, the tree structures in ARF are changing on the fly. Although many recent techniques have demonstrated that FPGAs are premier candidates to accelerate conventional Random Forest with a low-power consumption, how to efficiently realize ARF on FPGA while accounting its dynamics is still an open question. This assignment will focus on exploring suitable methodologies to realize adaptive random forest on FPGAs.

The assignment includes the following steps:

1. Understanding Adaptive Random Forest and how state-of-the-art methods deploy random forest on FPGA
2. Design and implement suitable architectures to account for the dynamics of Adaptive Random Forest
3. Evaluating the performance of the FPGA implementation by comparing with relevant studies

Other suggestions and related topics are also welcome.

Contact: Kuan-Hsun Chen and Nikos Alachiotis