Summer 2022 UROP

fpgaConvNet Hardware demonstrator development

19.8 - 30.9 (6 weeks)

Michal Palič

Imperial College London

EEE department

Supervised by:

Mr. Alexander Montgomerie-Corcoran

Dr. Christos Savvas Bouganis

The initial goal of this project was to develop a demonstrator for the fpgaConvNet framework [1] while basing the application on the work of Andrew Maclellan at the University of Strathclyde [2]. The demonstrator was intended to be a convolutional neural network in the role of a modulation classifier operating on time-series data. I was interested in this project because it would require gaining a degree of proficiency in the full stack of tools ranging from high level APIs for specifying machine learning models, down to FPGA programming. Over the course of the project the goals evolved in reaction to the limitations of the tool, while additional development contributions to the tool were made to resolve some of the limitations encountered.

The outcome of this project was partially successful. A demonstrator was developed, albeit for a simpler network than originally envisioned due to the limited feature support of the framework. This process was extensively documented in a ca. 20-page document, meant to serve as a tutorial for the framework and accelerate the onboarding of any other users seeking to utilize the framework. During the attempt to demonstrate the modulation classifier and the mnist-12 network, a fairly large number of issues were identified and communicated. My remaining time on the project was dedicated to resolving some of these issues that were encountered. Most of them related to parsing difficulties for the onnx format. The solutions were incorporated into this open-source project by its maintainer.

There were many skills that I developed over the course of the project due to the breadth of knowledge required in order to demonstrate and debug the inner workings of a framework operating at many layers of abstraction.

While I already possessed a basic understanding of many neural network concepts due to the preparation for last year's UROP, this year's UROP allowed me to apply this knowledge constructively. I gained experience with the specification, training and integration of neural network classifiers into software. The specified and trained network then served as the input to the fpgaConvNet tool.

The quirks of the fpgaConvNet tool often required the careful construction of its unput model, or the modification of existing networks. Due to the inexact mapping between the higher-level ML APIs such as Keras or Torch and the onnx format supported by fpgaConvNet, this project required me learn how to manipulate the encodings and structure of these files by hand. This allowed me to appreciate the design decisions made in this protocolbuffer based format. Here the performance and size of the serialized object were critical, as CNN models tend to have large file sizes. I also learned to operate on this model standard using the onnx-optimizer and onnx-simplifier frameworks.

The validation of the demonstrator required the computation of reference results from the developed models, which necessitated for me to get familiar with the available inference frameworks. This translated into the development of a working proficiency with the tools provided by the onnx-runtime library, a common way of embedding machine learning models in a wide spectrum of products ranging from web to desktop applications. I believe that this is also a powerful tool for the future due to its flexibility. While the onnx-runtime was the main utilized tool, Keras inference tools were also used for a part of the testing, related to the training the network in the said framework.

On the lower levels of abstraction, this project required me to gain familiarity with the Xilinx Vivado suite. I gained familiarity with the process for usage and testing of designs generated by the Vivado HLS compiler. I used the Vivado HDL flows and IP block designer to integrate the network IP and develop a digital system around it using the SoC capabilities of the provided ZC702 development board. I wrote the host and client code for this use case, which allowed for the demonstrator results to be compared with the software reference as well as for the performance of the system to be benchmarked and compared to the tool estimate.

The experience also served as my first encounter with collaborative work in the academic space. I worked with Alexander Montgomerie and the other postgraduate students involved in the tool development, to resolve the issues encountered. The original demonstrator model and use case was mainly based on some of the work of Andrew Maclellan from the University of Strathclyde, who answered many of my questions. It was a positive experience thanks to the willingness of everyone to donate their time.

Working together with the CAS research group provided me with a very valuable window into the experience of postgraduate students at Imperial College. I am thankful for this element of the UROP and the in-person nature of the project, because it allowed me to reflect more tangibly on the possibility of pursuing post-graduate study. I got to extensively witness the interactions between students, their advisors, academics in general as well as the atmosphere of the research group.

I also had the fortune to be present during a time when multiple visiting professors passed through the group and was invited to join several their talks. These presented the state of research in their fields, their work and the approaches being currently researched. The talks covered the areas of dynamic scheduling, CGRA modeling, and online arithmetic. I found these talks to be very beneficial as they expanded my awareness of the research being conducted around the world while providing a gentle introduction to the topics at hand.

References:

[1] CAS research group. (2017). *fpgaConvNet* [Online]. Available: https://cas.ee.ic.ac.uk/people/sv1310/fpgaConvNet.html

[2] A. Maclellan. (2022). *Streaming-CNN FPGA Architecture for Communications-based Applications* [Online]. Available: https://github.com/Axdy/rfsoc_modulation_classification