Contents

Plan for Milestone 1

In this phase,there were three main objectives. The first objective was to develop and test a plugin for OpenVINO and the second objective was to generate kernel codes (in order to run Simple CNN topology on a single FPGA) using Tensor Virtual Machine (TVM). The third objective was to customize TVM codes for the above mentioned plugin and to synthesize the codes for Stratix 10 board.

Development of OpenVINO Plugin

The development of OpenVino Plugin is a crucial objective of our project.This chapter contains the steps taken to create the first version of the plugin. This chapter also describes the tasks which were not completed (partially or completely).

2.0.1 Plugin development

- The Deep Learning Deployment Toolkit (dldt) github repository was integrated into the university gitlab server.
- A folder named Noctua_plugin was created inside dldt and another folder named kernels was created inside Nocuta_plugin to store the bitstreams (.aocx files).
- The development process started by editing the classification sample application to parse the IR of the given input topology.
- This parsed IR, which is stored in a data structure called CNNNetwork, is then passed on to the plugin.
- A single file named fpga_plugin.cpp retrieved the layer information from the IR, launched the kernels on the FPGA after passing them the required arguments.
- Caffe was used as the framework to generate protobuf file. OpenVINO does not support Tensorflow or Caffe 2.
- A customized Simple CNN kernel code (AOCL file) was used to test the plugin.
- A single MNIST PNG file was used to test the launching capabilities of OpenVino. The image was correctly classified on emulator. Binary image

files are not compatible with OpenVino. Thus , images in PNG format are passed to the plugin.

- OpenVino supports only PNG format images. Due to time constraints, only one PNG image was tested. Testing with more images is needed and will be done in Milestone 2.

Using TVM to generate kernels

Tensor Virtual Machine(TVM) is being used as a code generator. This chapter describes the steps taken to setup the TVM tool and to generate kernel codes. This chapter also contains the tasks which were not completed.

3.0.1 Development

- Relay is imported which helps provide intermediate representation (IR) for the model.
- Tensorfow or caffe model (protobuf file) is imported and graph definition is created from the file with the help of tensorflow GraphDef class.
- On creating graph definition, it is imported into relay frontend to generate relay expression and parameters from tensorflow protobuf file.
- The expression and parameters along with target host (OpenCL) are parsed to relay build function to create the final graph.
- TVM does not support custom architectures so the kernel codes for Simple CNN could not be generated.

Customizing and Synthesizing TVM generated kernels

The third objective was to customize and synthesize the kernel code generated by TVM to suit the needs of OpenVino plugin. In this chapter, the steps taken to meet this objective are discussed.

4.0.1 Custom code

- A kernel code compatible with OpenVino was written manually because we were not able to use TVM to generate kernel codes for Simple CNN.
- Due to time constraints , the code was not compiled (synthesized) for Stratix 10. But functional tests were carried out on the emulator.

Problems faced

- TVM does not support custom models due to which kernels for Simple CNN could not be generated.
- Hand-built kernel code file was used for launching kernels through the OpenVINO plugin. Tis means we still do not have experience fine tuning (or customizing) TVM generated kernel code.
- The plugin was tested on an emulator due to time constraints.
- Since kernels codes for CNN could not be generated using TVM, the idea of Simple CNN model has been scrapped and the focus is now on GoogLeNet.
- Generation of kernel code using TVM and the customization and synthesis of the kernel code on Stratix 10 will be achieved in the early stages of Milestone 2.

Conclusion

To summarize, not all the objectives as per the plan were achieved in Milestone 1. But enough progress has been made to move to the next milestone (Milestone 2).

- As discussed, a plugin was successfully developed for OpenVINO and the developed plugin supports convolution, max pooling and fully connected layers as required by the Simple CNN topology.
- OpenVino plugin is in a git repository named dldt and needs to be merged with pg-custonn2-2018.
- The plugin was partially tested on emulator.
- The final graph created in TVM is used to generate the OpenCL code. However, kernel codes were not generated using TVM due to the fact that TVM does not support custom models.
- Hence, the focus has now shifted to GoogleNet, which will be achieved in Milestone 2.
- Hand-built kernel code was used instead of TVM generated kernel code.
- Synthesis of the kernel code and running on Stratix 10 has been pushed to early stages of Milestone 2 due to time constraints.