ECE 332 Lab 4: Hand-Written digit recognition using DE1-SoC

Objectives:

- Train your neural network on your host machine
- Use your trained weights to perform inference on DE1-SoC board

Prerequisites

- Completion of Lab 3 and familiarity with the DE1-SoC board.
- Lab3 OpenCL kernel to program the board
- Lab3 capture_image.c program to capture image

Part 1:

Training NeuralNet on host machine (optional)

Required files are in the path tests/Inference

1. Open Jupyter notebook **TrainNN** and run all the cells

2. This will save the weights for each layer in “.bin” files

Testing the captured image inference : Print each layer output, helps debugging your C++ code

3. Capture the image using the Lab3 code **capture_image.cc** and steps

4. Run TestImage notebook, this will print each layer’s output. This will help in debugging and implementing each layer in the network

Part 2: Use your trained weights to perform inference on DE1-SoC board

Compiling aocx file on the host machine (Windows machine)

1. Extract de1_soc folder from the resources file
2. Setup OpenCL environment:
   go to the directory C:\intelFPGA\hld\ and type `init_opencl.bat` in the command prompt
3. Copy the **hardware** folder to a10_ref folder in the path
   C:\intelFPGA\hld\board\a10_ref\

Replace the hardware folder, this is because we don’t have permissions to change environment variables on the host machine. If you are working on your own system, instead of replacing the folder set the environment variable **AOCL_BOARD_PACKAGE_ROOT** to your own directory

4. Copy the **tests** folder to a10_ref folder in the path

5. In the command prompt type
   ```
   aoc -list-boards
   ```
   This should list all the boards that are listed, delete any other directories that we are not working with on the host machine

6. Launch Quartus prime, open the project in hardware directory named top.qpf and perform “Compile and Synthesis” (only compile and synthesis not the entire flow)

7. Now copy the .cl file you wrote in Lab 3 to the tests folder in de1-soc in the following path

   ```
   tests/Inference/device/
   ```

8. Close Quartus project and go to the tests/Inference directory and the
   ```
   aoc device/<your lab 3 .cl file> -o bin/<kernel name>.aocx -
   board=de1soc_sharedonly
   ```

   This will compile .aocx file that uses the hardware project and your multiplication kernel

Programming FPGA fabric with aocx file

1. Connect the SDcard that is flashed with the Linux image to the host machine using the card reader

2. Copy the .aocx file generated in the previous step to the SDcard (Don’t modify any files that are already present in the SDcard). This copied file will be available in the DE1-SoC linux in the path-
   ```
   /media/fat_partititon
   ```

3. Boot DE1-SoC by inserting the SDcard and set up OpenCL environment as described earlier.

   ```
   cd OpenCL/
   ```

```
source init_opencl.sh
```

4. After setting up OpenCL environment , copy the contents from SDcard which are in /media/fat_partititon/

   ```
   mv /media/fat_partititon/<file_name>.aocx  Labs/Exercise_3/solutions/
   ```

5. Now, we will program the FPGA fabric with with our design compiled in the previous section

   ```
   cd Labs/Exercise_3/solutions/
   aocl program /dev/ac10 bin/<file_name>.aocx
   ```

Inference logic on ARM CPU + FPGA

main.cpp is in the folder tests/Inference/host/src/

6. In Inference folder, following the instructions in main.cpp file to finish coding tasks and run the code on FPGA

7. You can run the main.cpp on the host machine by toggling the variable FPGA to 0 in the code. This will again print the outputs of each layer

8. On DE1-SoC board, to compile the main.cpp, you need to use Makefile. Copy a Makefile from any other project on the board to the tests/Inference folder.

Deliverables-

Part1 : A handwritten picture (example image from MNIST dataset) when held against the D5M camera, you should capture the image using capture_image.c and running detect the digit by running the inference network on CPU+FPGA