ELE548: Computer Architecture
Project Proposal
Neural Network Embedded Platform Performance Metrics

By: Matthew Lima and Krishu Nakarmi

Abstract

***Abstract* - Deep learning is a rapidly expanding field of data science and is being used increasingly in everyday life. This field is unique in that the calculations necessary rely heavily on matrix operations which have been optimized to be performed on Graphics Processing Units (GPUs) due to their highly parallel nature. While GPUs are still widely used during the training phase of Deep Neural Networks (DNNs), research is being done towards the best platform for deploying trained DNNs in increasingly complicated environments. We will analyze the YOLOv3-tiny DNN on an ARM-based system (Raspberry Pi 4b), an x86-64 based system (AMD Ryzen 7 5800X3D), an x86-64 based system with a GPU accelerator (AMD Ryzen 7 5800X3D with an RTX3060), and an FPGA based system (Xilinx PYNQ-Z2 development board) to capture quantitative metrics and determine the most suitable application for each platform.**

Description

Our project involves running the same neural network (or as close as possible with the given hardware) on various platforms with the intention of determining which platform is best suited for different deployment requirements. For example, running a facial recognition network for a security system in a bank would have different deployment requirements than an object recognition network for a self-driving car. One requires the highest accuracy possible while having a nearly unlimited power ceiling, while the other needs to be able to run in real-time while maintaining a finite power draw.

The object detection model that will be used for this testing is the YOLOv3-tiny convolutional neural network (CNN). This model was chosen because it is based off of a more complicated CNN (YOLOv3) and was reduced to maintain a high level of accuracy while being able to run more efficiently. Ideally the full YOLOv3 network (or a newer version, such as the YOLOv7 CNN) would be run, but due to hardware and budget constraints the reduced model will be used.

The hardware chosen for the project was selected because of its availability and cost. Most of the hardware is already available for use, such as the Raspberry Pi 4B, Ryzen 7 5800X3D, and the RTX3060. The FPGA that will be used is the Zynq-7000 SoC XC7Z020-1CLG400C on the XUP PYNQ-Z2 development board. This FPGA was chosen because of its low-cost, high-performance FPGA, and high-speed off-chip interfaces. We imagine the biggest performance bottleneck for the FPGA would be feeding the image data into the CNN accelerator, so the DDR3 memory on board will help to alleviate that issue. We can also investigate using the USB interface for running the CNN in real time and displaying the results over the HDMI interface, but the primary goal will be to run the network on images initially stored on the SD card.

The benefit of using the pre-trained YOLOv3-tiny network is that there are many examples online of how to run the system on all the systems we are testing. This will help to alleviate the time constraint by reducing the amount of time spent porting the network to all the different testing platforms. The most difficult platform to port the network to will be the FPGA because of how open-ended the system is, and securing a definitive implementation will be key in how quickly we can get the system running.

Because of the increasing use of artificially intelligent systems, most of which rely on DNN-based technology, determining the best deployment platform for a specific neural network is becoming an increasingly more difficult task. As embedded systems engineers with a focus on FPGA-based design, we feel as though this project is something that can help other teams determine what platform is the best for deploying their neural network based on their own specific deployment requirements.

Project Outline

Week 1:

- Acquire PYNQ-Z2 development board.
- Setup hardware platforms: Raspberry Pi, Ryzen 7 5800X3D, RTX3060, PYNQ-Z2.
- Install the necessary software dependencies for each platform.

Week 2:

- Convert and transfer the pre-trained model to the Raspberry Pi and Windows platforms.
- Develop a test environment on the Raspberry Pi and Windows machine to collect and store relevant data.
- Begin work on porting the model to the PYNQ-Z2.

Week 3:

- Collect test data for the Raspberry Pi and Windows platforms.
- Finish porting the model to the PYNQ-Z2.
- Develop a test environment for the PYNQ-Z2 board.

Week 4:

- Collect test data for the PYNQ-Z2.
- Compare results of all testing environments.

Week 5:

- Write final report.
- Create power point presentation to present results.

Evaluation Methodology

Accuracy

Evaluating the accuracy of a DNN is a fairly straight-forward process: give the system an input with known outputs, and compare the outputs of the system to the true metrics. Computer Vision is a rapidly growing field of computer science and because of this there are many competitive datasets used to evaluate the accuracy of certain models to compare against others. For the YOLOv3-tiny model we will use the Common Objects in Context (COCO) dataset. This dataset was chosen because it is free to download and has already been evaluated on the YOLOv3-tiny network. This will ease the process of getting the system running as there is an in-depth procedure listed on how the system was originally evaluated.

Performance

The performance metric for the system will be based on the evaluation time for a series of images run through the network on each platform. Since this network is designed to be used in real-time for computer vision applications, this evaluation time metric can then be used to determine what the theoretical frames-per-second value could be expected from each platform, which is what most real-time applications would use to quantify model performance.

Power Draw

To evaluate the power draw of the system, a simple power meter will be placed in-line with each of the testing platforms. For the Windows PC, the power supply of the computer will be plugged directly into the meter. For the Raspberry Pi and the PYNQ-Z2, the AC/DC converter used to power the boards will be plugged into the meter.

This setup will have errors inherent with the data collection because of the difficulty of isolating the power drawn by the execution of the model vs. the power drawn by the various sub-components and sub-processes (i.e. the efficiency of the AC/DC converter, the other applications running on a traditional Windows installation, etc.). These errors will be factored into the final power calculation to provide a confidence interval of where the true power draw lies.

References

- [1]J. Redmon and A. Farhadi, YOLOv3: An Incremental Improvement. 2018. [1804.02767] YOLOv3: An Incremental Improvement (arxiv.org)
- [2]T.-Y. Lin et al., Microsoft COCO: Common Objects in Context. 2015. [1405.0312] Microsoft COCO: Common Objects in Context (arxiv.org)
- [3]A. Boutros et al., "Beyond Peak Performance: Comparing the Real Performance of AI-Optimized FPGAs and GPUs," in 2020 International Conference on Field-Programmable Technology (ICFPT), 2020, pp. 10–19. doi: 10.1109/ICFPT51103.2020.00011. Beyond Peak Performance: Comparing the Real Performance of AI-Optimized FPGAs and GPUs | IEEE Conference Publication | IEEE Xplore
- [4]J. Wang and S. Gu, "FPGA Implementation of Object Detection Accelerator Based on Vitis-AI," in 2021 11th International Conference on Information Science and Technology (ICIST), 2021, pp. 571–577. doi: 10.1109/ICIST52614.2021.9440554. FPGA Implementation of Object Detection Accelerator Based on Vitis-AI | IEEE Conference Publication | IEEE Xplore
- [5]F. Al-Ali, T. D. Gamage, H. W. Nanayakkara, F. Mehdipour, and S. K. Ray, "Novel Casestudy and Benchmarking of AlexNet for Edge AI: From CPU and GPU to FPGA," in 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2020, pp. 1–4. doi: 10.1109/CCECE47787.2020.9255739. Novel Casestudy and Benchmarking of AlexNet for Edge AI: From CPU and GPU to FPGA | IEEE Conference Publication | IEEE Xplore