FPGA vs GPU for machine learning
Project background investigation

Buyuan Lin; buruce@bu.edu

1. Background

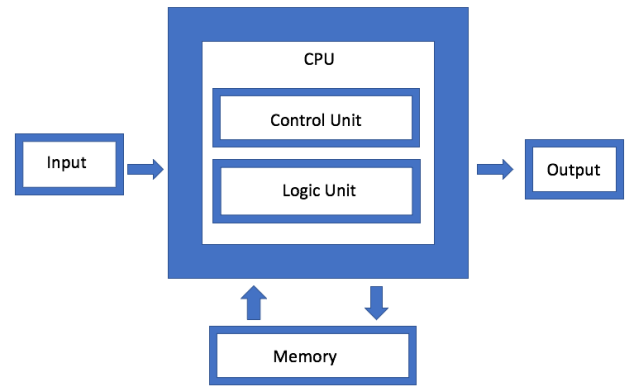
Ever since DNN (deep neural network) was invented, it has been wildly used in machine learning/ deep learning. First, it's because DNNs are capable of processing different tasks in different areas, e.x.CNN (convolutional neural network) for image recognition RNN (recurrent neural network) for natural language processing. Second is because DNNs have promising results in accuracy. However, there are related problems.

The major problem is that DNN relays heavily on computing power. At first, people tend to use CPU for all the work, but they soon find out that comparing to CPU, GPU is much more powerful in SIMD (single instruction multiple data) because of the difference in architecture. Simply speaking, GPU has much more cores which is exactly machine learning needs and that's the reason researchers and developers started to abuse GPU in their projects. Although GPU is already very powerful, it's still not enough for some researchers' crazy ideas: a normal DNN based on basic TensorFlow API takes couple hours to train, a customized DNN may take more hours or even days to train. An idea proposed by google called NAS (network architecture search), initially takes about 2000 GPU days to train according to estimation, which seems impossible even to google. This is the point where people started to think about machine learning and GPUs.

Like CPUs, despite all the core GPUs have, they still are based on Von Neumann architecture. This means, for every instruction that's fed to the GPU, the GPU has to do the following processes according to my knowledge in computer architecture: instruction fetch, instruction decode, data read, execution (ALU (Arithmetic & Logic Unit operations or memory address calculating) and data write back. Each of them will take one clock cycle. It may not seem to be a problem in executing normal programs since CPUs normally have 3~4G Hz clock frequency and even GPUs have around 1.5G Hz clock frequency. However, the instruction number in DNNs grows exponentially based on their architecture and data used in training. People started to realize that the Von Neumann architecture is not very efficient for this kind of task. But the commercial chip industry is already been set for decades and not every individual or organization is capable of developing a customized chip like Google did (Google's TPU, Tensor Processing Unit). As said, FPGA came to people's awareness.

The Von Neumann architecture.

(https://semiengineering.com/knowledge_centers/compute-architectures/von-neumann-architecture/)



FPGA (field programmable gate array) is basically a chip which contains only transistors, thus called the gate array. However, those transistors can be programmed to do the things that the user desires. In the case of machine learning, for example, it can be programmed specifically to do matrix multiplication without wasting too many clock cycles. As a result, the training time of a DNN could be dramatically decreased.

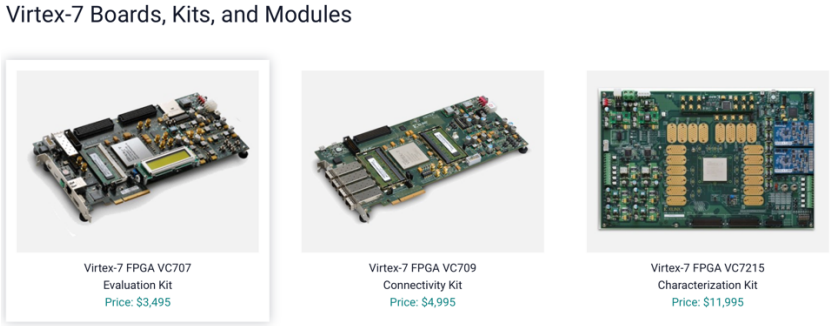
Another problem with the computer power craving machine learning is the cost. The cost to purchase and set up devices for machine learning projects. Top end GPUs designed for machine learning are costly, yet FPGAs are not that costly comparing to GPUs.

Top end machine learning GPU price

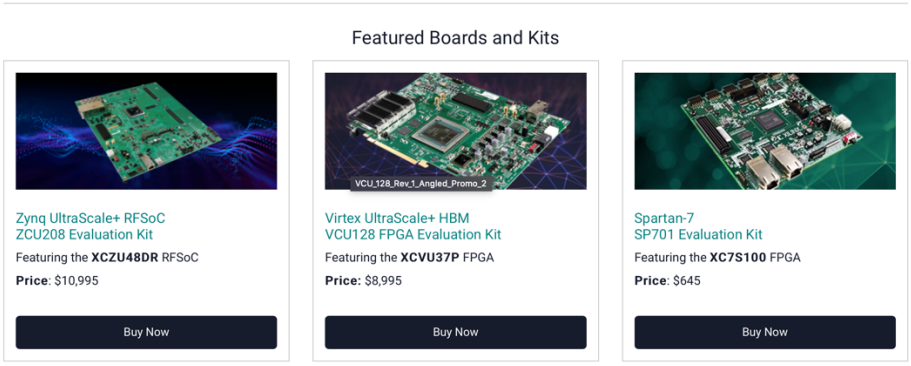
Tesla V100 Price

Tesla GPU model	**Price**	**Double-Precision Performance (FP64)**
Tesla V100 PCI-E 16GB or 32GB	**$10,664*** **$11,458*** for 32GB	7 TFLOPS
Tesla P100 PCI-E 16GB	**$7,374***	4.7 TFLOPS
Tesla V100 SXM 16GB or 32GB	**$10,664*** **$11,458*** for 32GB	7.8 TFLOPS
Tesla P100 SXM2 16GB	**$9,428***	5.3 TFLOPS

Virtex-7 FPGA price (used in related researches)
https://www.xilinx.com/products/boards-and-kits/device-family/nav-virtex-7.html



Xilinx FPGA price (https://www.xilinx.com/products/boards-and-kits.html)



Those are two major motivations of the research area of FPGA based machine learning according to my knowledge and research.

2. Mission and Purpose of the Research Area and Products

As said, the major goal of FPGA based machine learning is to:

1. Accelerate the training/testing process of a DNN
2. Reduce the cost of machine learning projects.

This research area is relatively new, and I haven't seen any commercial product in the market right now. But I my opinion, the products have to be able to solve above problems, FPGA is a start, after proper developments, chip manufactures can move on to ASICs (application specific integrated chips) for even lower cost and easier deployments.

3.Product Use Cases

From a developer's perspective, they want to save time when designing DNNs. While software improvement is one possible approach, a hardware improvement is more straight forward. However, purchasing new GPU is a costly thing to do. An FPGA accelerator would save money while providing similar results.

From a research's perspective, this seems to be a great area to revolutionize the

machine learning area.

From a chip manufacture's perspective, this is a profitable area to invest time and money.

4. Related Researches and Analysis:

A GPU-Outperforming FPGA Accelerator Architecture for Binary Convolutional Neural Networks: https://arxiv.org/abs/1702.06392

This research proposed a possible approach to combine binarized neural network algorithms with FPGA

Caffeinated FPGAs: FPGA Framework for Convolutional Neural Networks: https://arxiv.org/pdf/1609.09671.pdf

This research proposed a possible framework using FPGA by smartly combining CPU, GPU and FPGA to get the best result.

Accelerating a random forest classifier: multi-core, GP-GPU, or FPGA? https://ieeexplore.ieee.org/document/6239820

This research proposed some test results of accelerating a random forest classifier using different hardware including multi-core, GP-GPU and FPGA

Binarized Neural Network: Training Deep Neural Network with Weights and Activations Constrained to +1 or -1: https://arxiv.org/abs/1602.02830

This research proposed a possible algorithm to convert the float number weights and activations in DNNs while remaining competitive test results.

5. Proposal:

I think combining FPGA with Binarized Neural Network is a promising approach. Since the way FPGAs work is digital, which is perfect for binarized operations. Besides, most of the researches only use FPGA for CNN yet RNN is neglected. If we can apply the Binarized Neural Network algorithms/idea to RNN that would be great. However, the issues are FPGA programming is complicated and developing new algorithms for RNN may be too much for me or a group as graduate students.

What's more: Sorry I don't have the proper equipment to reproduce the experiments yet. Since I just seen AWS FPGA and the SpoNN projects in Wednesday's lecture. I may need more time to investigate those projects. Hope we can talk about this and my approach of doing this project. Thank you!