William Trimmer
Dr. Shimeng Yu
Emulating Neural Network with Resistor Network
Field Programmable Gate Arrays for Hardware Acceleration of Machine Learning

Introduction

Machine learning and neural networks have changed the way computing is done, both in consumer and enterprise devices. The problem with these two approaches is that they are purely software focused, and therefore slow by nature. Training a neural network can take hours for one dataset, and much longer for multiple datasets. Specialized hardware is required to offload the work of software and greatly speed up the entire process of NN training. Google currently uses TPUs for TensorFlow to accelerate their machine learning processes. But these devices can cost upwards of one hundred million dollars to design, develop, and maintain. FPGAs are the perfect device to be used for a hardware implementation of a neural network. They are simple to use, reprogrammable, allow for output to external devices. This technical review summarizes some commercially available FPGAs and provides methods of implementation for ideal operation in neural networks.

Commercial Availability of FPGA

There are several types of FPGAs that provide different benefits. The most popular type is the SRAM-based FPGA. These use volatile SRAM to store data while the FPGA is in use. There are two modes of programming with SRAM FPGAs. Master mode reads data from an external source, while slave mode allows the FPGA to be controlled by an external master device [1]. The best SRAM FPGA on the market is the Xilinx Virtex-7 Development Board ($3495), which boasts 1GB DDR3 RAM, 128MB of flash for PCIE, HDMI out, and 2 LCD displays, along with a plethora of I/O and control devices [2].

SRAM-based FPGAs don't have internal storage, so external storage is required. To make up for this shortcoming, there are also the SRAM-based FPGA with internal flash memory. This is like the SRAM-based FPGA, but with the inclusion of flash storage on-board, it eliminates the need for using external memory for loading data [1]. There aren't many of these FPGAs on the market, but the most popular is the Lattice Semiconductors LatticeXP2 ($43), which has 1024MB RAM, and less than 1MB of flash storage [3]. With such little flash storage, this type of FPGA is more for hobbyists and not for large development purposes.

There are FPGAs that use nothing but flash memory. These flash-based FPGAs don't use SRAM for configuration or storing data. The advantages for using flash memory include less power consumption, storing data without power, allowing the user to skip the configuration steps during boot, and a lower overall cost [1]. This is a new design method, so there aren't many solutions currently on the market. The best one right now is the Microchip Technology ProAsic3 ($320) with 516 Mb of RAM and 3 million

logic gates [4]. With the amount of memory limited on-board, the SRAM-based FPGA is still the best option.

The least common FPGA design is the antifuse-based FPGA. These FPGAs can only be programmed once. After being burned to conduct current, it cannot be reprogrammed [1]. The most common use of antifuse FPGAs include wireless electronics with high performance requirements, weapon systems, and medical instruments [5]. Microchip Technology produces the Axcelerator ($1270), which has a maximum frequency of 763Mhz and 2 million gates with a max supply voltage of 1.575V [6]. With the limited memory and the one-time use, this FPGA is the worst type to use for training neural networks since it can only be used once, and training requires adjusting values in the data multiple times.

Implementation of FPGAs in Machine Learning

The main use of FPGAs in machine learning is for hardware acceleration of model training [7]. The FPGA deals with all the input data, and the output is sent to another device which deals with the weight matrix. Develop control logic on the FPGA that takes in input data and evaluates each pixel, which will be either a 1 or 0, if the image black and white, and not RGB. If it is in RGB format, then more work will need to be done to convert it to a usable value. Convert these values to analog voltages and output to an external device, whether it be another FPGA or a network of resistors. The more voltages output, the more accurate the model will be. When the external device is done, it will return a value that the FPGA can evaluate.

The use of FPGAs in neural networks cannot be underestimated. They are used more than ASIC computers and general-purpose processors because FPGAs are more cost efficient and deal with concurrency better [8]. FPGAs are also easily programmable using a low-level language like VHDL, which allows for better performance. They are not used for learning because on-chip learning results in a loss of efficiency in a hardware implementation since it requires higher precision [8]. Off-chip learning is crucial for the success of any implementation because of this.

[1] 1-Core Technologies, “FPGA Architectures Overview,” 1-Core Technologies, Moscow, Russia. [Online], Available: https://www.pdx.edu/nanogroup/sites/www.pdx.edu.nanogroup/files/FPGA-architecture.pdf. [Accessed Oct. 20, 2018].

[2] Xilinx, “Xilinx Virtex-7 FPGA VC707 Evaluation Kit,” 2018. [Online]. Available: https://www.xilinx.com/products/boards-and-kits/ek-v7-vc707-g.html. [Accessed Oct. 20, 2018]

[3] Lattice Semiconductor, “Programmable Logic IC Development Tools Lattice XP2 Brevia Dev Kit,” 2018. [Online]. Available: https://www.latticestore.com/products/tabid/417/categoryid/59/productid/302/searchid/1/searchvalue/lfxp2-5e-b2-evn/default.aspx. [Accessed Oct. 20, 2018].

[4] Digi-Key, “ProASIC3,” 2018. [Online]. Available: https://www.digikey.com/catalog/en/partgroup/proasic3/14339. [Accessed Oct. 20, 2018].

[5] Microsemi, “Axcelerator,” 2018. [Online].Available: https://www.microsemi.com/product-directory/antifuse-fpgas/1700-axcelerator. [Accessed Oct. 21, 2018].

[6] Mouser Electronics, “Microsemi AX2000-1FG1152I,” 2018. [Online]. Available: https://www.mouser.com/ProductDetail/Microsemi/AX2000-1FG1152I?qs=sGAEpiMZZMvoScKlWpK8TFh2aURzvfKHo%252bZsT3uBtVM%3d. [Accessed Oct. 21, 2018].

[7] S. Yu, “Neuro-Inspired Computing with Emerging Nonvolatile Memory,” *Proceedings of the IEEE*, vol. 106, no. 2, Feb., pp. 260-285, 2018.

[8] A.R. Omondi, and J.C. Rajapaksa, Eds., *FPGA Implementations of Neural Networks*. Dordrecht, The Netherlands: Springer, 2006.