How to Innovate in the Field of Machine Learning Using FPGA

Daniel Lerner

Introduction

Machine learning is an ever-expanding field, with diverse applications in areas such as economics, biology, and medicine, just to name a few. With such a wide variety of use cases comes a wide range of different workloads, with different performance, power, and cost requirements to get the job done. Historically, researchers and application scientists have flocked to GPUs as their hardware of choice for machine learning workloads. Compared to its closest hardware competitor, the CPU, a GPU exhibits a higher aptitude for parallel computations, which are prevalent in machine learning. This advantage is especially pronounced when a cluster of GPUs is used to create a general purpose computing device, referred to as a GPGPU. With a software infrastructure surrounding them in NVIDIA's CUDA platform, GPGPUs make up ground in flexibility and ease of use, areas in which a CPU would typically outperform a standalone GPU. In more recent research, however, FPGAs have become an increasingly popular option for accelerating machine learning workloads due to their low power consumption, customizability, and ability to exploit pipeline parallelism [1]. As opposed to the data parallelism that GPUs excel at, computing the same operation on many pieces of data at once, pipeline parallelism involves performing and forwarding output from multiple different operations simultaneously. This research seeks to exploit the desirable qualities of the FPGA in order to identify and develop a new potential optimization to machine learning workloads.

Prior Work

There has been extensive innovation with FPGAs in the field of machine learning in recent years. Researchers are continuously developing new ways to use FPGAs to solve a variety of problems. Some examples include optimizing for specific use cases, creating a model for more general use, targeting a new workload, and partnering with complementary technologies. The following summary of prior work demonstrates examples of each of these types of innovations that can be used as inspiration for a future endeavor.

One study that looked at a specific use case was conducted in 2017 by researchers at ETH Zurich [2]. Kara et al. focused on Stochastic Gradient Descent (SGD), an algorithm that is commonly used in training machine learning models, and programmed an FPGA to perform this function on datasets of varying precision. They found that lowering the precision of the input data could speed up the training process anywhere from 2-11x depending on the workload. In this case, the team used the configurability and speed of an FPGA to assess the viability of a tradeoff in the machine learning model. Using FPGA as a tool in this way would be a reasonable scope for a new project in the realm of machine learning.

A more ambitious problem is tackled in the instance of DeepBurning [3], in which Wang, et al. provide a sort of meta-solution for using FPGAs to accelerate machine learning workloads. This study created a design automation tool that would allow users to configure customized FPGA designs that were optimized for their model of interest. This sort of research attempts to build a foundation that would make FPGAs more accessible for future use.

Another study, titled DianNao [4], applied FPGAs to large-scale workloads and compared their performance to that of a general purpose processor solution. While FPGAs had been commonly used in accelerating certain machine learning computations, there had been little effort in having them run entire state-of-the-art neural networks. Researchers here focused on a real-world use case that had been largely untapped by FPGAs, and designed a solution that beat their comparable standard by nearly 2 orders of magnitude. This paper lays the groundwork for another possible definition of success for a project, where FPGAs are either used in a completely new way or are used to significantly improve an existing workload.

One final possibility to consider when researching with FPGAs is implementing a heterogeneous solution - one with multiple types of hardware. A 2015 paper out of George Mason University [5] outlined a methodology to identify CPU-intensive workloads that would be best offloaded to an accelerator such as an FPGA. Researchers created a solution that employed both pieces of hardware to great success. They measured an overall speedup of close to 3x when offloading certain computations to an FPGA. This outline demonstrates yet another successful way to research with FPGAs - implementing a heterogeneous solution that can be directly compared against a general purpose processing solution.

Requirements

The efforts described above pave multiple possible paths for a future successful FPGA research project. This new use for FPGA in the field of machine learning should satisfy at least one of the following requirements:

1. Use FPGA as a configurable tool to test the effects of a change in a parameter
2. Develop a methodology that makes using an FPGA more versatile and broadly impactful
3. Apply an FPGA solution to a specific workload to achieve a speedup
4. Implement a heterogeneous solution that pairs FPGA with an existing solution to achieve a speedup (see figure below)

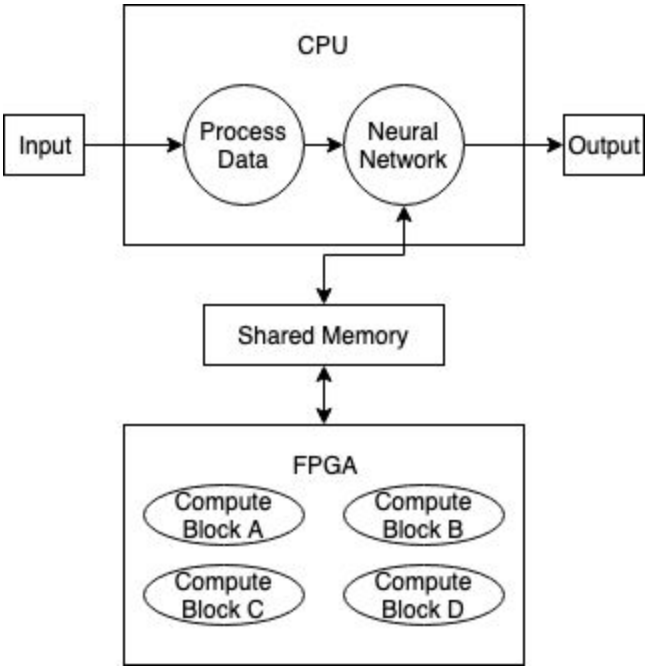

Figure 1: A Heterogeneous Solution for a Machine Learning Workload

In addition, the following testing criteria should be met -

- Any project intending to meet goals (1) or (2) should be tested on multiple different types of training data, as well as multiple datasets of the same type.

Solutions aiming to be generally applied should be proven to work with different types of data, a trait demonstrable with publicly available training sets. Furthermore, different datasets of the same type should be used to determine that the solution works for a type of data, as opposed to a particular dataset. Some popular choices include mnist and gisette, two sets of images of handwritten digits, text data such as Amazon reviews or Twitter posts, or audio like Free Music Archive.

- Any project planning to accomplish (3) or (4) should target an average of at least 1.4x overall speedup

The target speedup number represents the approximate annual rate needed to keep up with Moore's law of doubling a machine's capability every 2 years, and is a modest goal when compared to the examples provided. Some prior work shatters this number, recording orders of magnitude of performance improvement over the comparable existing solution.

Conclusion

In summary, existing publications establish several different blueprints for using FPGAs to accelerate machine learning models. A new FPGA could be used to test a potential optimization, improve performance a specific workload, complement an existing general purpose solution, or even revolutionize as a whole the way that researchers and application engineers work with FPGAs. FPGAs are a desirable hardware option because they are fast, low power, and configurable for both custom and general use. The main advantage that a current GPGPU solution would have over an FPGA is the software infrastructure, allowing for easier implementation and quicker turnaround time. Most current FPGA implementations target specific algorithms or functions in the machine learning process, but, as more and more work is done using FPGAs, a similar environment could emerge that rivals the current standard. There is ever growing demand for faster and denser machine learning hardware for countless real-world applications, and FPGAs are a great candidate for propelling innovation in the field as a whole.

Sources

[1] Lacey, G., Taylor, G. and Areibi, S. (2019). *Deep Learning on FPGAs: Past, Present, and Future*. [online] arXiv.org.
URL: https://arxiv.org/abs/1602.04283

[2] K. Kara, D. Alistarh, G. Alonso, O. Mutlu and C. Zhang, "FPGA-Accelerated Dense Linear Machine Learning: A Precision-Convergence Trade-Off," *2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, 2017, pp. 160-167.
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7966672&isnumber=7966626

[3] Wang, Y., Xu, J., Han, Y., Li, H. and Li, X. (2019). *DeepBurning*. [online]
URL: https://dl.acm.org/citation.cfm?id=2898003

[4] Chen, T., Du, Z., Sun, N., Wang, J., Wu, C., Chen, Y. and Temam, O. (2019). *DianNao*.
URL: https://dl.acm.org/citation.cfm?id=2541967

[5] K. Neshatpour, M. Malik and H. Homayoun, "Accelerating Machine Learning Kernel in Hadoop Using FPGAs," *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Shenzhen, 2015, pp. 1151-1154.
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7152609&isnumber=7152455