High Efficiency Hardware Accelerators for Convolutional Neural Networks

Introduction

Convolutional Neural Networks (CNNs) are usually applied in computer vision tasks such as image detection, recognition, and classification. Unlike traditional sequential workloads that involve data dependencies and control flow dependencies and can be efficiently executed by Central Processing Units (CPUs), CNNs mainly perform matrix operations that can be massively paralleled and thus cannot be efficiently executed on CPUs [8]. Heterogeneous hardware accelerators are the most common solutions to accelerate the computation of CNNs. This technical review briefly summarizes some of the state-of-the-art CNN hardware accelerators, introduces underlying CNN computation mechanisms, and provides an ideal implementation of a hardware accelerator for CNN classification.

Commercial Hardware Accelerators

Due to the nature of convolutional neural networks which requires many parallel matrix operations, hardware accelerators are typically designed to maximize concurrent arithmetic operation throughput [8]. There are three common types of hardware accelerators for deep neural networks: General Purpose Graphics Processing Units (GPGPUs), like NVIDIA Tesla V100, are currently widely adopted for their general design purpose for massive parallelism [6]; Tensor Processing Units (TPUs), like Google TPU, are designed specifically for deep learning computation and are capable of doing fast matrix multiplications [2]; and Application Specific Integrated Circuits, or ASICs, are customized case-by-case for special neural network architectures and take inputs in a fixed format [3]. All three modern hardware accelerators are based on digital data pipelines and take advantage of data level parallelism to increase computation efficiency.

All processor cores inside a GPGPU are general purpose cores, and during real-time CNN inference computation, only a fraction of the logic units is used (mostly arithmetic multipliers) [10]. Typical GPGPU prices range from $300-$1000 and can achieve 500%-1000% speedup in parallel workloads when compared to a CPU [8]. Processors inside a TPU are specialized for matrix multiplication and can efficiently compute a batch of data within relatively fewer clock cycles, but still add extra scheduling overhead to deal with different sizes of matrix inputs and CNN layers. TPUs can achieve 280%-730% speedup in CNNs when compared to a GPGPU but are only available on Google Cloud Platform for $8.50/hour [1, 2]. ASICs are designed for special algorithms and architectures and can be more efficient in specific applications [3]. However, since recent neural network

architectures experience constant changes and revisions, few ASICs have been designed for these state-of-the-art architectures [4].

Underlying Mechanisms: Modern CNN Architectures

Convolutional Neural Networks consist of two major components: convolution layers, which perform matrix convolution operations and apply several filters to the target input to detect specific features in the input, and fully connected layers, which process the resulting detected features and multiply the data matrices with weight matrices to determine the class of the input [5]. Thus, computing the output of the neural network requires two major operations: matrix convolution and matrix multiplication.

Modern CNN architectures include both deep and shallow ones. Deep CNN architectures, like Vgg-16 or Vgg-19, use tens of convolutional layers followed by several layers of fully connected neurons [9]. Shallow CNN architectures, like LeNet-5, use only two to three convolutional layers followed by one to two layers of fully connected neurons [5]. Deeper CNNs tend to have 5%-30% higher accuracy but takes more computation power in orders of magnitudes to train and to compute inference than shallower ones [9].

Building Blocks of an Ideal Hardware Accelerator

To classify an image efficiently using a CNN, both software algorithms and hardware computations are required. Software algorithms are used to assign weights to each variable in the network to achieve efficient and accurate classification. To minimize unnecessary computation, the software algorithm must reduce the number of weight parameters in the network, while being able to retain a high classification accuracy [7]. Tradeoffs between accuracy and complexity must be carefully considered based on the requirements of the application. Hardware computing structures need to be fast and efficient as well to accelerate training and inference computation which could otherwise take hours or even days to complete. An efficient hardware architectural design should be able to maximize the throughput of the matrix operations without introducing significant error in floating point operations [3]. Thus, an ideal hardware accelerator for CNN classification should include both carefully designed weights and parameters in software algorithms, and low-latency high-throughput hardware designs.

References:

[1] "Benchmarking Google's new TPUv2 – RiseML Blog," *RiseML Blog*, 23-Feb-2018. [Online]. Available: https://blog.riseml.com/benchmarking-googles-new-tpuv2-121c03b71384. [Accessed: 22-Oct-2018].

[2] "Cloud TPU Documentation," *Google*. [Online]. Available: https://cloud.google.com/tpu/docs/. [Accessed: 21-Oct-2018].

[3] C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. Lecun, and E. Culurciello, "Hardware accelerated convolutional neural networks for synthetic vision systems," *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010.

[4] A. Joshi, "ASICs for Deep Learning Are Not Exactly ASICs," *Tractica*. [Online]. Available: https://www.tractica.com/artificial-intelligence/asics-for-deep-learning-are-not-exactly-asics/. [Accessed: 21-Oct-2018].

[5] Y. LeCun, *MNIST Demos on Yann LeCun's website*. [Online]. Available: http://yann.lecun.com/exdb/lenet/. [Accessed: 21-Oct-2018].

[6] "NVIDIA Tesla V100 Data Center GPU," *NVIDIA*. [Online]. Available: https://www.nvidia.com/en-us/data-center/tesla-v100/. [Accessed: 21-Oct-2018].

[7] Hinz, Magg, Sven, Wermter, and Stefan, "Speeding up the Hyperparameter Optimization of Deep Convolutional Neural Networks," *[1807.07362] Speeding up the Hyperparameter Optimization of Deep Convolutional Neural Networks*, 19-Jul-2018. [Online]. Available: https://arxiv.org/abs/1807.07362. [Accessed: 21-Oct-2018].

[8] S. R. Upadhyaya, "Parallel approaches to machine learning—A comprehensive survey," *Journal of Parallel and Distributed Computing*, vol. 73, no. 3, pp. 284–292, Mar. 2013.

[9] Simonyan, Karen, Zisserman, and Andrew, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *[1409.1556] Very Deep Convolutional Networks for Large-Scale Image Recognition*, 10-Apr-2015. [Online]. Available: https://arxiv.org/abs/1409.1556. [Accessed: 21-Oct-2018].

[10] S. Yalamanchili, *ECE8823 GPU Architecture*. [Online]. Available: http://ece8823-sy.ece.gatech.edu/. [Accessed: 21-Oct-2018].