Neural Network Hardware Acceleration

Leveraging parallelism in FPGAs to improve neural network performance

Robert Lee, Dept. of Electrical and Computer Engineering

Supervised by Dr. Kin Fun Li, Dept. of Electrical and Computer Engineering

March 4, 2020

University of Victoria

This research was supported by the Jamie Cassels Undergraduate Research Awards.

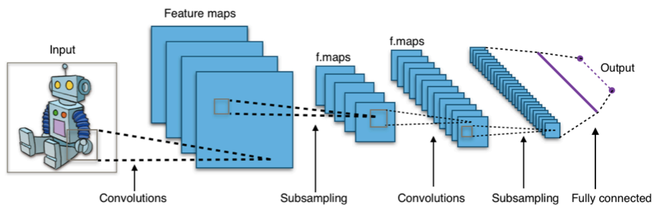
Background:

Artificial neural networks are prevalent in our modern world:

- **Computer vision:** Autonomous vehicles, super-resolution, object segmentation
- **Natural language processing:** speech-to-text, translation
- **Classification:** object & pattern recognition, medical imaging
- **Data processing:** regression analysis, biometrics, controls

Introduction:

Convolutional neural networks (CNNs) are commonly used. They consist of **multiple layers** in a **feed-forward structure**:



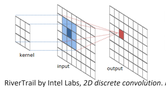
Aphex34, *Typical CNN architecture*. Wikipedia, 2015. Available: https://commons.wikimedia.org/wiki/File:Typical_cnn.png.

The network is defined by its **architecture** and **weights**.

Elements of a CNN:

The building blocks of CNNs include:

- **Convolution:** This operation **element-wise multiplies** an input region by a ***kernel***. This *kernel* is **moved across** the input layer to generate all values in an output layer.

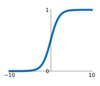


RiverTrail by Intel Labs, *2D discrete convolution*. Available: http://intellabs.github.io/RiverTrail/tutorial/.

- **Activation Functions:** This layer emulates the biological neuron's *"action potential"*, where a neuron *fires* if its input is **sufficiently high**. Popular functions are shown below:

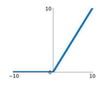
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



ReLU

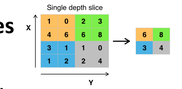
$$\max(0, x)$$

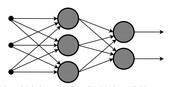


U. Udofia, *Activation Functions*. Medium, 2018. Available: https://medium.com/dataseries/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17.

- **Normalization:** Especially important for ReLU activations to prevent unbounded output, this layer helps **remove biases** and **improve training time**.
- **Pooling:** Also known as *subsampling,* this **reduces spatial size** by **discarding data**. A common operation is *max-pooling,* where the max is kept.
- **Fully-connected layer:** Once the spatial size is reduced, this layer **connects to all neurons** of one or more previous layers.

Aphex34, *max_pooling with 2x2 filter and stride = 2*. Wikipedia, 2015. Available: https://commons.wikimedia.org/wiki/File:Max_pooling.png.





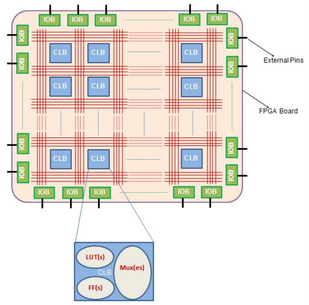
Offnfopt, *A Neural network with multiple layers*. Wikipedia, 2015. Available: https://commons.wikimedia.org/wiki/File:Multi-Layer_Neural_Network-Vector.svg.

CONV layers occupy **over 90%** of compute time during prediction.

What is a FPGA?

Short for ***F**ield **P**rogrammable **G**ate **A**rray*, FPGAs are ***integrated circuits*** that can be **reprogrammed** with a *hardware description language* (HDL). They consist of three *programmable* components:

- **Logic blocks:** These are the **computation** and **storage elements** of the system. They contain building blocks such as look-up tables, flip-flops, and multiplexers.
- **Interconnect:** This **connects logic and I/O blocks together** according to the specified design. It's built with multiplexers, transistors, and buffers.
- **Input/Output:** These connect the FPGA with desired **external components**. They must conform to many different voltage standards.



Sneha H.L., *Internal architecture of a typical FPGA*. All About Circuits, 2017. Available: https://www.allaboutcircuits.com/technical-articles/purpose-and-internal-functionality-of-fpga-look-up-tables/.

Advantages of FPGAs:

FPGAs are the middle ground between general-purpose compute such as CPU and GPU and application specific integrated circuits (ASICs). FPGA strengths include:

- **Low latency:** Because hardware circuits are used, they can **perform computations much faster** with a **deterministic latency**.
- **Connectivity:** I/O are connected **directly to the pins** of the chip, instead of buses, greatly **improving the bandwidth**.
- **Reconfigurability:** Described with HDLs and synthesized onto a FPGA board, its internal structure can be **modified on demand**.
- **Energy efficiency:** FPGAs have high performance per Watt.
- **Parallelism:** Chips can be designed to perform multiple computations in **parallel** and **pipelined** architectures.

Optimizing CNNs for FPGAs:

Three levels of parallelism in CNN: feature map, neuron & synapse.

FPGA accelerators focus on convolution and fully-connected layers:

- **Kernel weights are shared** among neurons of a feature map.
- **Accessing off-chip DRAM** data consumes large power and time.

CONV is compute heavy: < 10% weights, but > 90% computations

FC is bandwidth heavy: > 90 weights, but < 10% computations

Other layers require much fewer resources in comparison.

Example Implementation:

An example by Qiu *et al.* employs a line buffer design followed by multipliers and adder trees to allow a convolution result to be **computed every cycle**. It leverages all three types of parallelism:

- **Operator-level** (fine-grained): the proposed 2D convolver.
- **Intra-output** (calculating an output from multiple input features): multiple convolvers work concurrently in a processing element.
- **Inter-output** (independent features are calculated concurrently): multiple processing elements can be used.

This system achieved **26.0x power** reduction compared to GPU and **1.4x** and **2.0x performance** increase vs. CPU and GPU, respectively.

Input/output of an example CNN depth estimation system:

C. Godard, O. M. Aodha, M. Firman and G. Brostow, "Digging Into Self-Supervised Monocular Depth Estimation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 3827-3837.

Limitations:

There are multiple implementation challenges:

- **Significant memory storage:** Millions or billions of model parameters for weights alone, which may **exceed available FPGA resources**. Careful consideration of processor design and maximizing resource sharing is required.
- **Variation in inter-layer requirements:** Different **parallelism** and **memory-access requirements** across different layers, so the designed accelerator must be **flexible** to optimize all layers.
- **Increasing CNN sizes:** More complex problems require larger number of layers, **increasing the number of weights**. To combat this, many weights in fully-connected layers have minimal impact on overall accuracy, and thus can be eliminated.

Conclusion:

Accelerating CNN training and prediction performance is highly desirable and an active research area. FPGAs offer greatly improved results on many metrics but still face bottlenecks in many scenarios.

Acknowledgements:

I would like to express my deep gratitude to Dr. Kin Fun Li for his mentorship, support, and guidance. This research was supported by the Jamie Cassels Undergraduate Research Awards, University of Victoria.

References:

[1] A. Shawahna, S. M. Sait and A. El-Maleh, "FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review," in *IEEE Access*, vol. 7, pp. 7823-7859, 2019.

[2] J. Qiu *et al.*, "Going deeper with embedded FPGA platform for convolutional neural network," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, 2016, pp. 26–35.

[3] C. Godard, O. M. Aodha, M. Firman and G. Brostow, "Digging Into Self-Supervised Monocular Depth Estimation," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 3827-3837.

[4] Mittal, Sparsh. (2018). A Survey of FPGA-based Accelerators for Convolutional Neural Networks. *Neural Computing and Applications*. 10.1007/s00521-018-3761-1.

[5] Sneha H.L., "Purpose and Internal Functionality of FPGA Look-Up Tables - Technical Articles," *All About Circuits*, 09-Nov-2017. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/purpose-and-internal-functionality-of-fpga-look-up-tables/. [Accessed: 02-Mar-2020].

[6] A. van der Ploeg, "Why use an FPGA instead of a CPU or GPU?," *Medium*, 14-Aug-2018. [Online]. Available: https://blog.esciencecenter.nl/why-use-an-fpga-instead-of-a-cpu-or-gpu-b234cd4f309c. [Accessed: 02-Mar-2020].