FPGAs and Neural Networks/ZynqNet

Gernot Walser

December 14, 2021

Neural Networks

Motivation

- Computer Vision, Speech Recognition and Image Classification become increasingly important
- Hard coded algorithms are replaced by Machine Learning concepts
- Neural Networks have very high computational complexity
- Many applications need low latency and high efficency

Neural Networks

- Inspired by biological nervous sytem
- Predict simple functions through learnable weights
- Basic building blocks

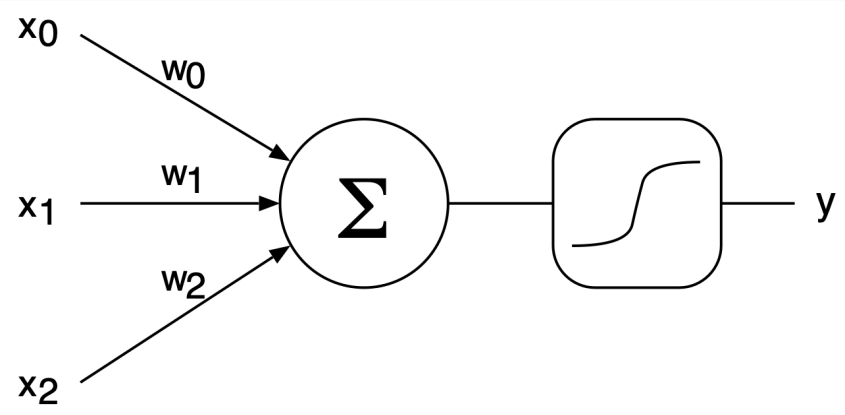

Figure 1: Aritifcal Neuron with inputs x, weights w and output y [1]

Neural Networks (cont'd)

- Multiple artificial neurons connected in layers
- Connection to each neuron in the adjacent layer
- Able to learn complex functions
- Adjust weights in training phase

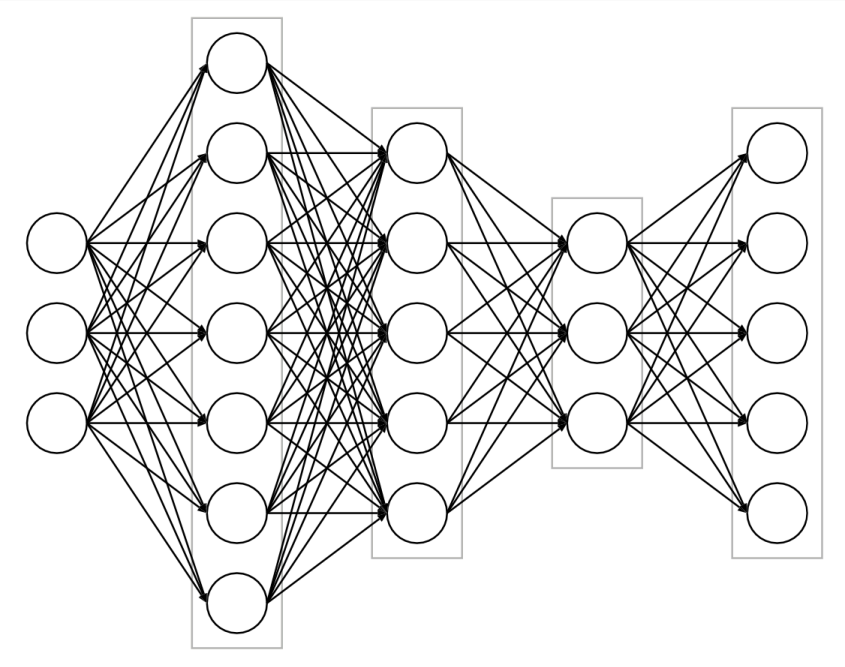

Figure 2: Fully Connected Neural Network [1]

Convolutional Neural Networks

- Specialized for image data
- Most information is local
- No fully connected architecture needed
- 2D Convolutions

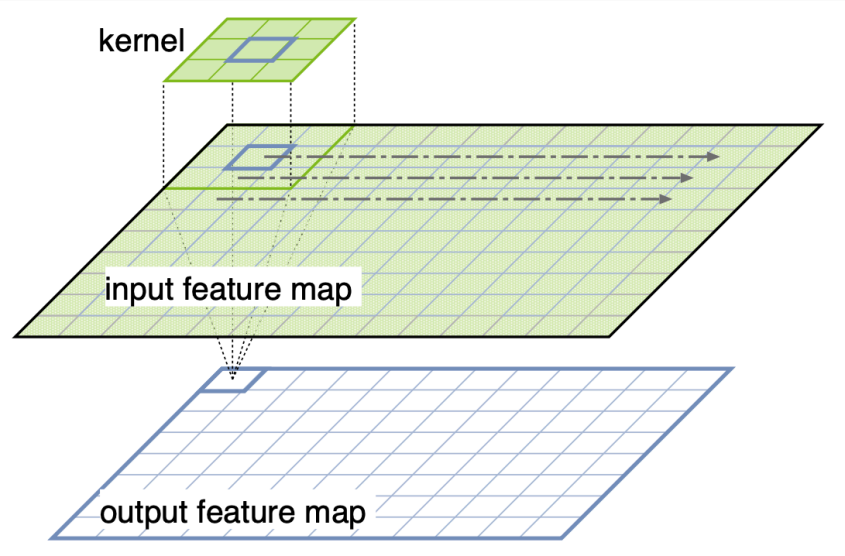

Figure 3: Convolutional Neural Network [1]

Parallelization

3x3 Convolutional kernel, with output O, input I and weights F:

$$O^{(co)}_{(y,x)} = \sum_{ci=0}^{ch_{in}-1} \left(\sum_{j=-1}^{1} \sum_{i=-1}^{1} I^{(ci)}_{(y-j,x-i)} \cdot F^{(ci,co)}_{(j,i)} \right)$$

- Independence of:
  - Layers
  - Pixel locations (y, x)
  - Input channels ci
  - Output channels co
  - Intra-kernel multiplications
- Lots of data reuse and parallelization potential

Hardware Accelerators

CPU vs. GPU

CPU:

- Highly optimized for serial workloads
- Few cores
- Pour performance per watt

GPU:

- Most popular platform for Neural Networks
- Highly optimized for independent parallel workloads
- Fast floating point operations
- Optimal for batches of data
- High power consumption

ASIC vs. FPGA

ASIC:

- Structure freezed at design time
- Typically only accelerators for specific parts
- High price
- Best performance per watt

FPGA:

- Reasonable flexibility
- Well suited for parallel workloads
- Good performance per watt
- Low latency

Applications

Low latency and low power consumption make FPGAs interesting for:

- Computer vision
  - Robots
  - Drohnes
  - Autonomous Driving
- Datacenters
  - Microsoft
  - Google (ASIC)

Challenges

- Large Neural Networks have millions of parameters
- Usually no floating point hardware available
- Off-chip memory access is slow
- Solutions:
  - Reduce parameter count
  - Use 16 bit float / binary weights
  - Data locality is key
  - Reuse on-chip data

ZynqNet

ZynqNet

FPGA-based Convolutional Neural Network (CNN) implementation, with focus on co-operation between Hardware and CNN.

- **ZynqNet CNN:** Custom FPGA optimized CNN architecture
- **ZynqNet FPGA Accelerator:** FPGA architecture for efficient acceleration of the ZnyqNet CNN
- Implemented on Zynq platform
- Trained offline on GPUs

ZynqNet CNN

- Based on SqueezeNet
- Consists of:
  - 1x1 and 3x3 convolutional layers
  - ReLU activations
  - Concatenation
  - Global average pooling

ZynqNet FPGA Accelerator

- 3x3 multiplications in parallel
- Partial parallelization of output channels
- Cache 2 full input image lines
- Cache all weights of current layer
- Cache single output elements

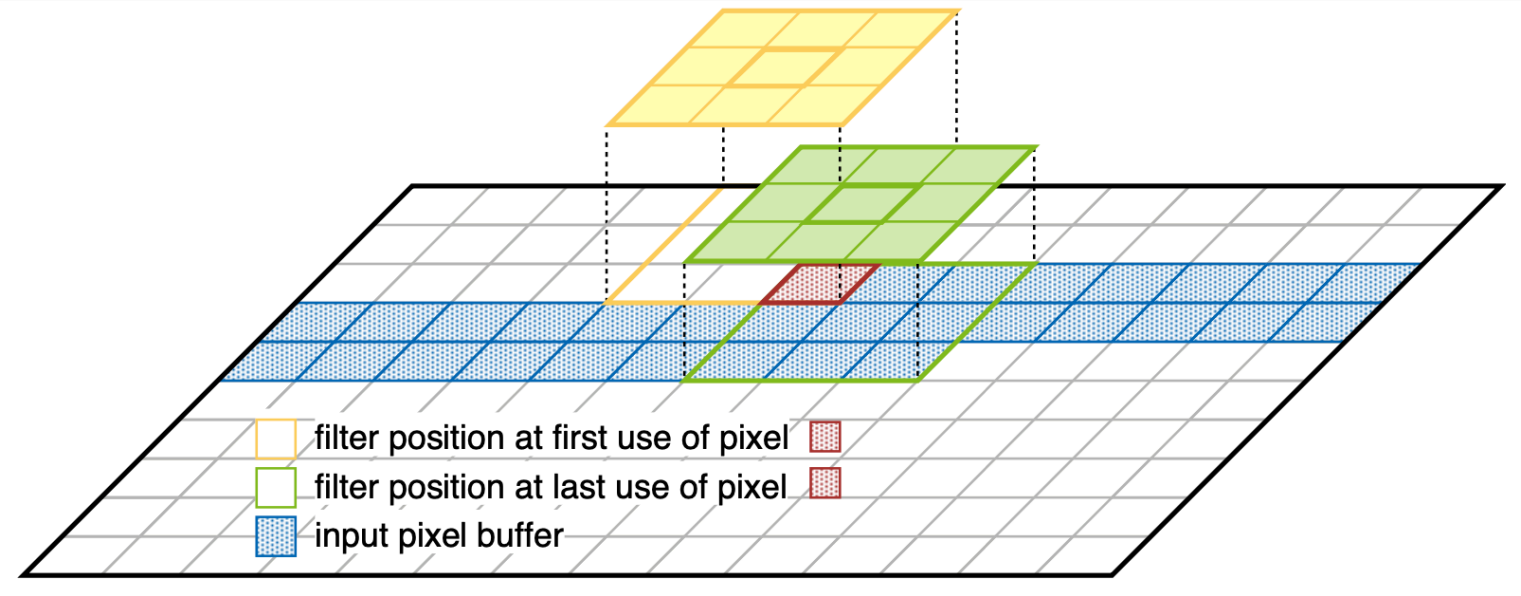

Figure 4: Input line caching [1]

ZynqNet FPGA Accelerator (cont'd)

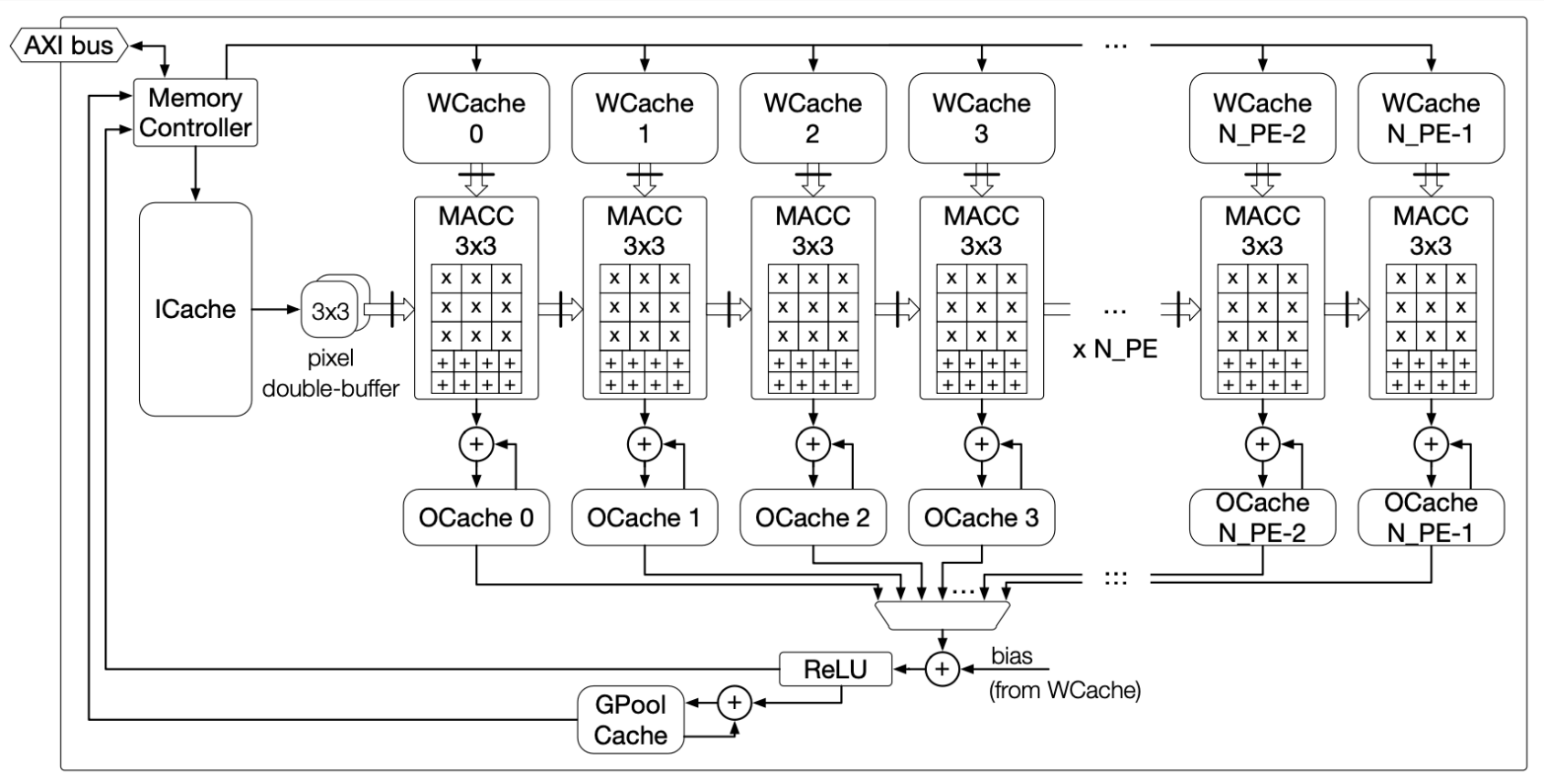

Figure 5: Block diagram [1]

Utilization and Performance

- Implemented on Zynq XC-7Z045 FPGA
- 80% DSP slices used
- 90% Block RAM used
- 1955ms per frame

	#conv. layers	#MACCs [millions]	#params [millions]	#activations [millions]	ImageNet top-5 error
ZynqNet CNN	**18**	**530**	**2.5**	**8.8**	**15.4%**
AlexNet	5	1 140	62.4	2.4	19.7%
Network-in-Network	12	1 100	7.6	4.0	~19.0%
VGG-16	16	15 470	138.3	29.0	8.1%
GoogLeNet	22	1 600	7.0	10.4	9.2%
ResNet-50	50	3 870	25.6	46.9	7.0%
Inception v3	48	5 710	23.8	32.6	5.6%
Inception-ResNet-v2	96	9 210	31.6	74.5	4.9%
SqueezeNet	18	860	1.2	12.7	19.7%
SqueezeNet v1.1	18	390	1.2	7.8	19.7%

Figure 6: Block diagram [1]

Questions?

References i

[1] David Gschwend. *ZynqNet: An FPGA-Accelerated Embedded Convolutional Neural Network*. https://arxiv.org/pdf/2005.06892.pdf. Online; accessed 13 December 2021.

[2] Ahmad Shawahna, Sadiq M. Sait and Aiman El-Maleh. *FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review*. https://ieeexplore.ieee.org/document/8594633. Online; accessed 13 December 2021.

References ii

[3] Eriko Nurvitadhi, David Sheffield, Jaewoong Sim, Asit Mishra, Ganesh Venkatesh and Debbie Marr. *A FPGA-based Hardware Accelerator for Multiple Convolutional Neural Networks*. `https://ieeexplore.ieee.org/document/8565657`. Online; accessed 13 December 2021.

[4] Kalin Ovtcharov, Olatunji Ruwase, Joo-Young Kim, Jeremy Fowers, Karin Strauss and Eric S. Chung. *Accelerating Deep Convolutional Neural Networks Using Specialized Hardware*. `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/CNN20Whitepaper.pdf`. Online; accessed 13 December 2021.

References iii

[5] Stacey Higginbotham. *Google takes unconventional route with homegrown machine learning chips*. https://www.nextplatform.com/2016/05/19/google-takes-unconventional-route-homegrown-machine-lea Online; accessed 13 December 2021.

[6] Lukas Cavigelli. *FPGA System Design for Computer Vision with Convolutional Neural Networks*. https://iis-projects.ee.ethz.ch/index.php/FPGA_System_Design_for_Computer_Vision_with_Convolutional_Neural_Networks. Online; accessed 13 December 2021.