AdderNet

and its Minimalist Hardware Design for Energy-Efficient Artificial Intelligence

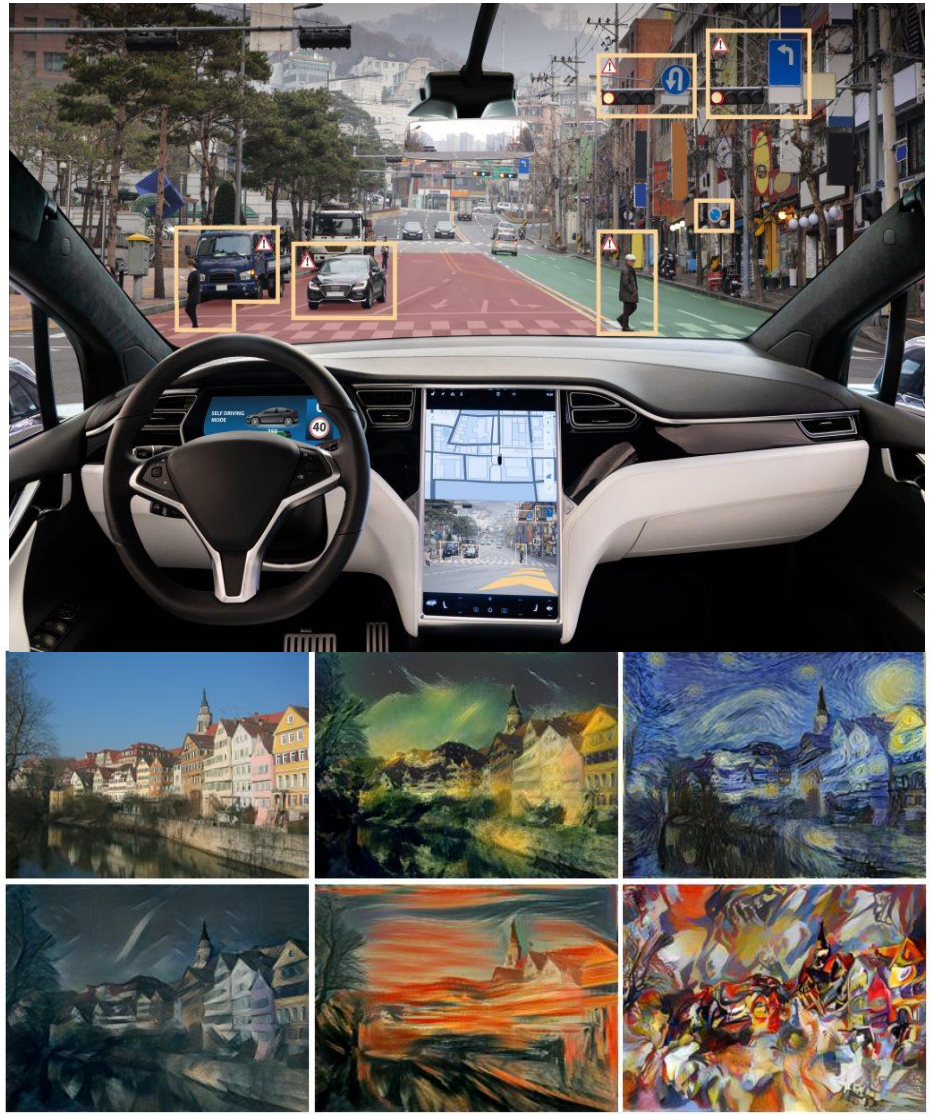
10.06.2021 | Jannik Hofmann

ML, CNNs, Convolution kernels

Introduction / Motivation

Machine Learning

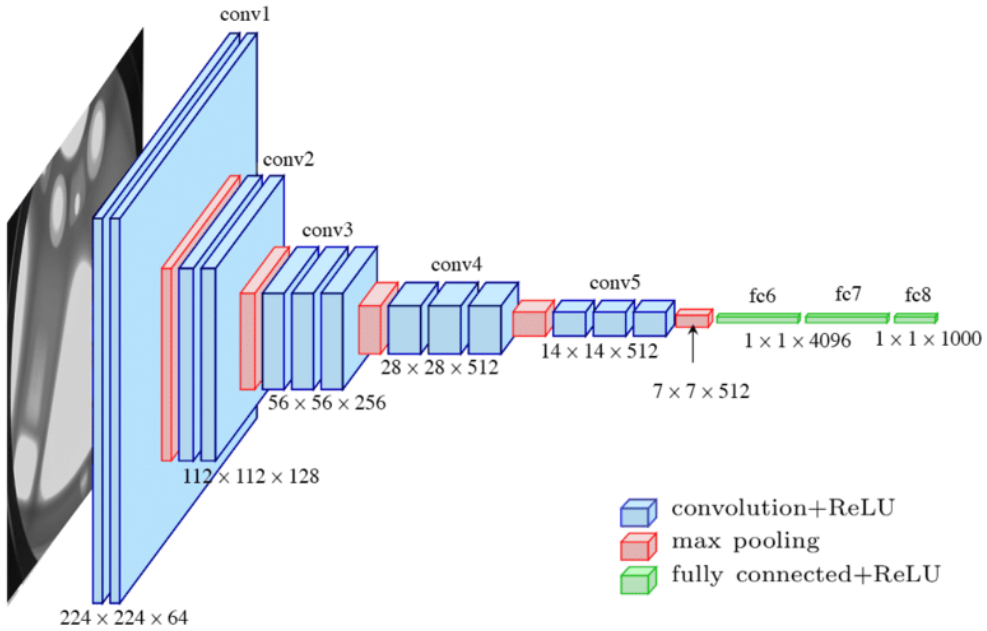
- Part of AI
- Countless applications
  - Computer vision
  - Image classification
  - Autonomous driving
  - Cybersecurity
  - Medical imaging
  - Media generation
  - Natural language processing



[Shaw K., 2019], [Cysmith, 2018]

Convolutional neural networks

- Important for computer vision, speech recognition, signal processing…
- Type of Deep neural network
- Achieves dimensionality reduction
  - e.g. extracting meaningful information from images
- Supervised learning
- Has convolutional layers
- Needs millions of multiplication-operations / layer



Architecture of VGG-16

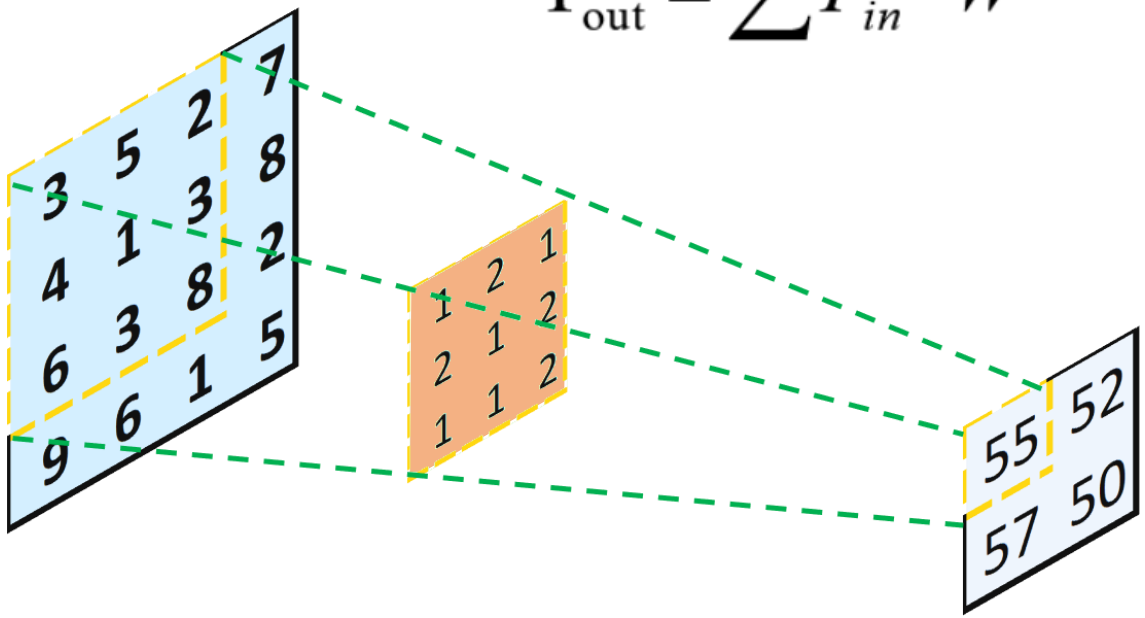
[Ferguson et al., 2017]

Convolution kernel

- Kernel size
- Weights
- Goes over array
  - Step size can reduce resolution
- Sum of weight * input
- Can have more dimensions

*3	*2	*(-1)
*2	*(-3)	0
*1	*2	*2

$$F_{out} = \sum F_{in} * W$$



[Wang et al., 2021], [Greenholt, 2020]

CNN drawbacks

- Computation power
- Energy consumption
- Embedded systems
  - Resource utilization targets
  - No high-performance graphics card
  - Needs reliability on weak hardware
- Millions of multiplication operations expensive
  - Need for kernel that doesn’t multiply

[Wang et al., 2021]

Alternative kernel designs

shift-CNN

shift core

>>1	>>2	<<1
>>2	<<3	0
0	>>1	<<2

Shift-CNN: Shifts bits of input value, multiplied with sign

$$F_{out} = \sum \text{shift}(F_{in}) \cdot \text{sign}()$$

XNORnet

XNOR logic core

==	==	!=
!=	==	==
==	!=	==

XNORnet: Works with bits, either flips or leaves them

$$F_{out} = \sum XNOR(F_{in}, W)$$

Kernels are subsets of CNN multiplication

<<3	<<2	<<1	0	>>1	>>2	>>3		==	!=
*8	*4	*2	*1	*0.5	*0.25	*0.125		*1	*-1

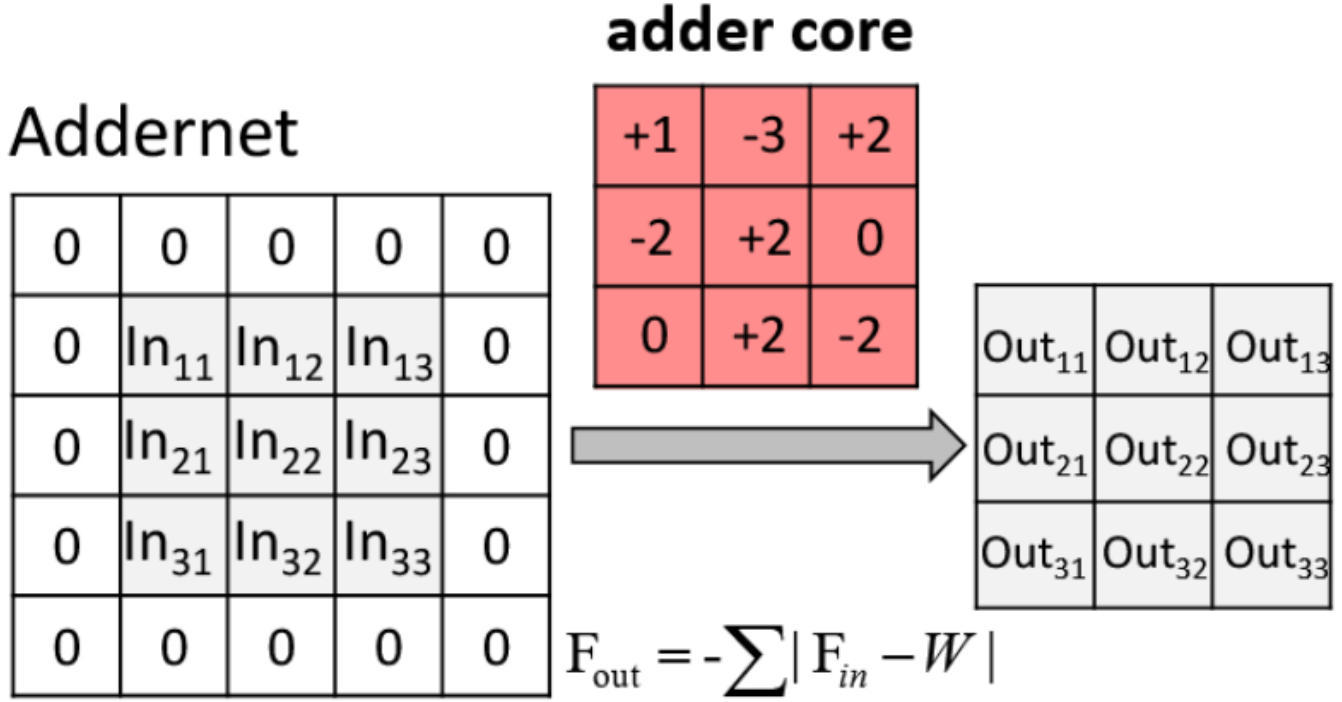
[Wang et al., 2021]

Adder core design

AdderNet Convolution

Addernet kernel design

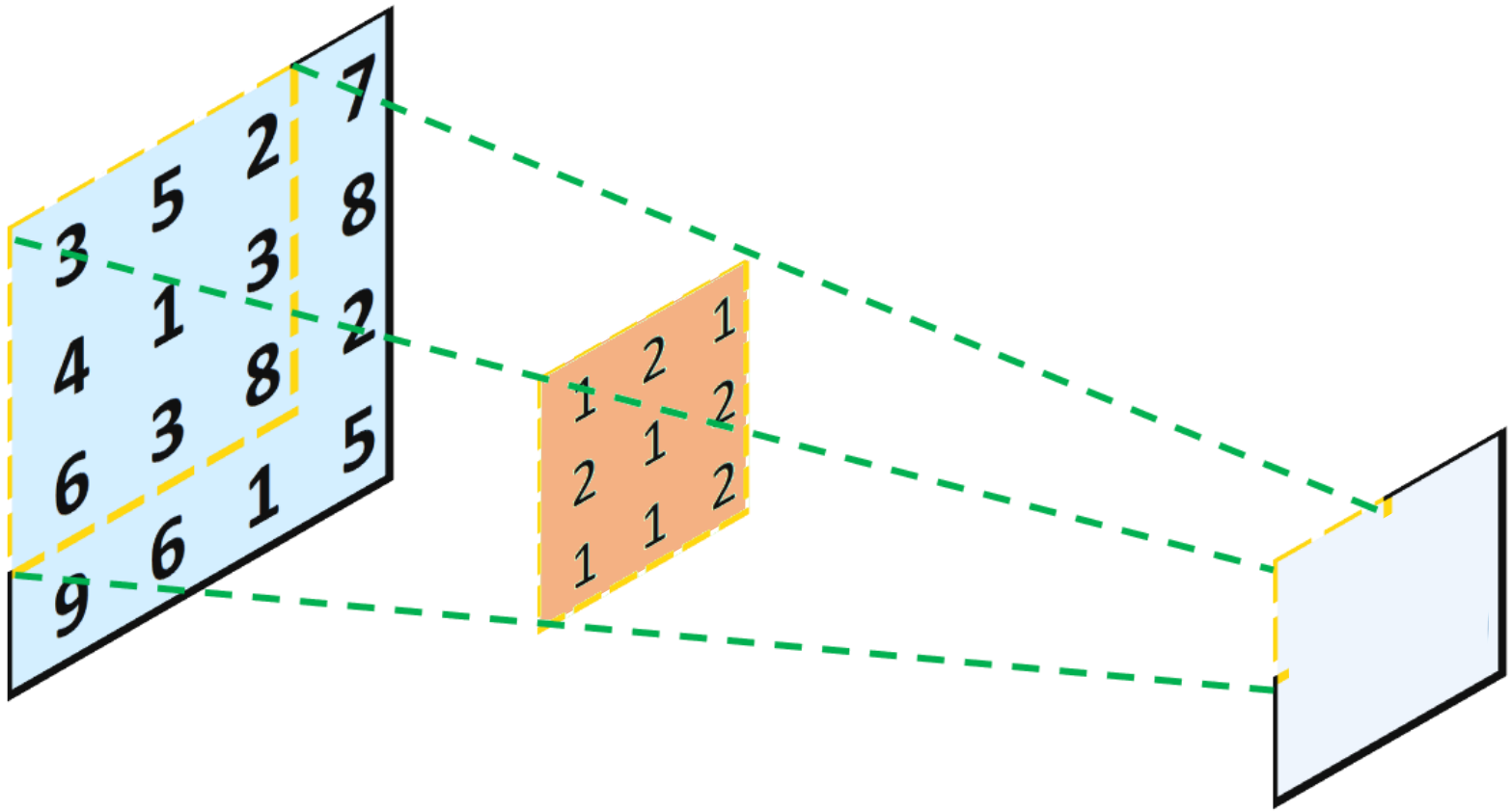
- Measures similarity between filters and inputs
- L1-norm



[Wang et al., 2021]

Addernet kernel design

- Measures similarity between filters and inputs
- L1-norm



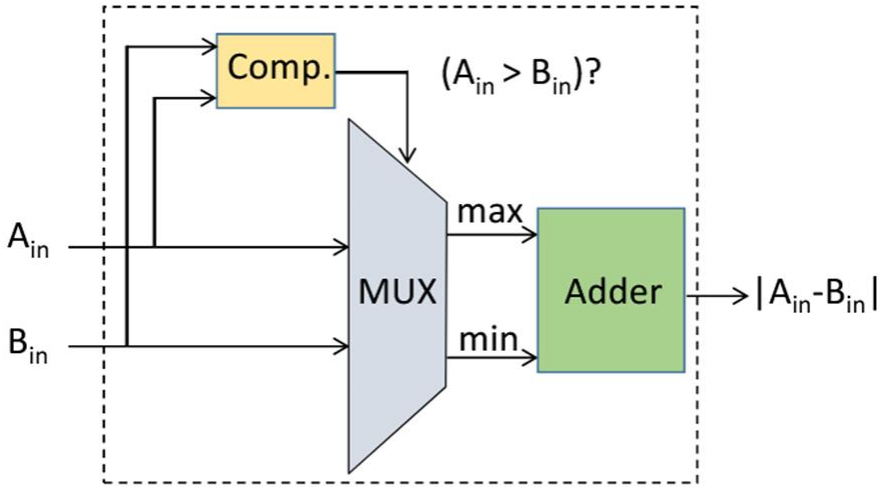
[Wang et al., 2021], [Greenholt, 2020]

FPGA, quantization, parallelization, …

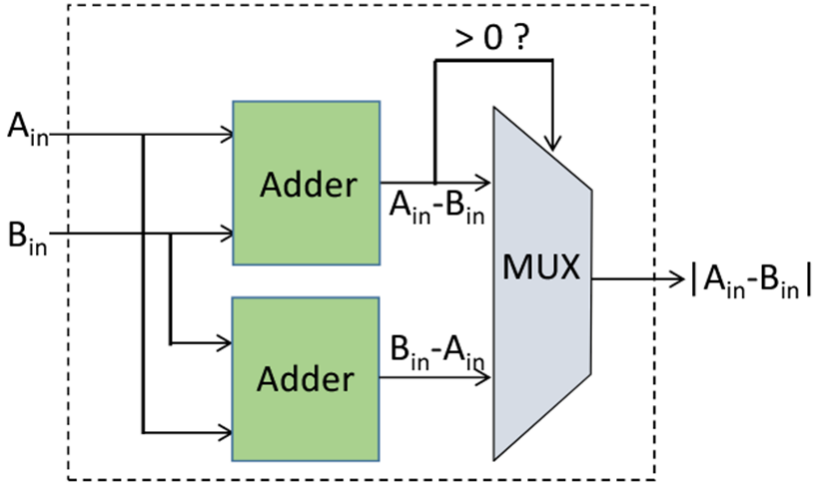
HARDWARE IMPLEMENTATION

Hardware implementation

- 1C1A: 1 Comparator, 1 Adder
  - Compare input with kernel weight
  - Return *larger value - smaller value*
  - Consumes less resources
  - Higher gate-delay => slower

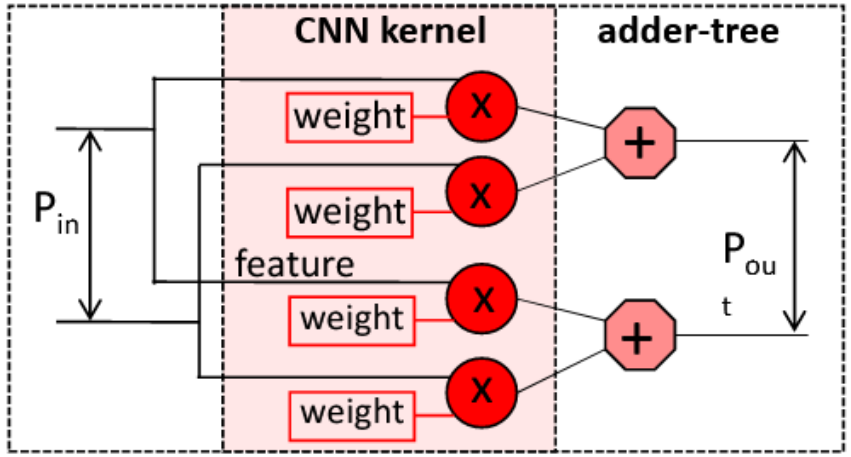


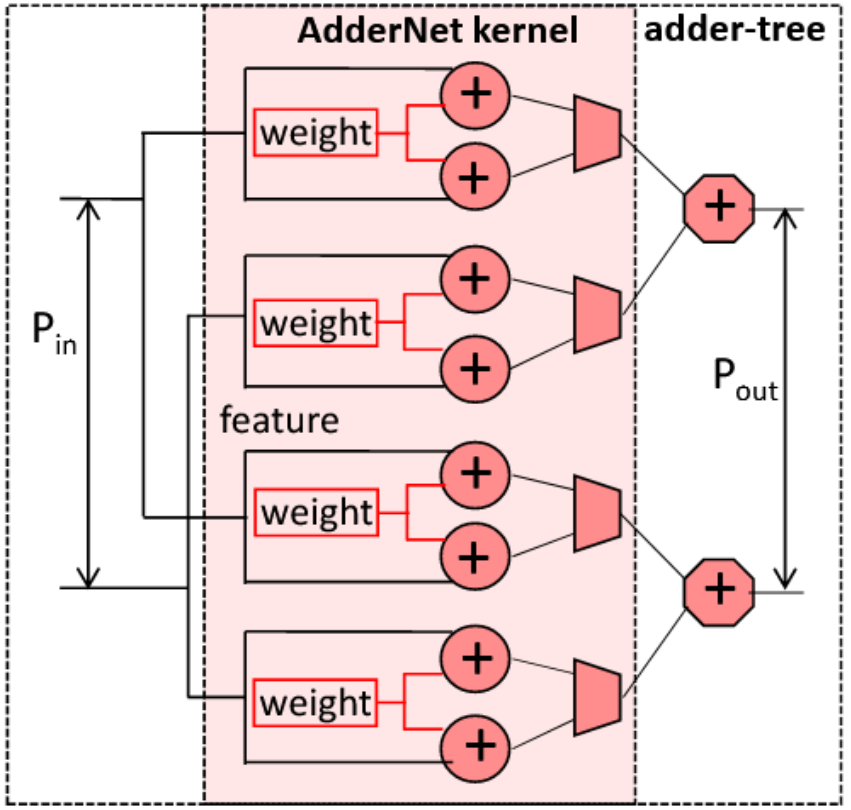
- 2A architecture: 2 Adders
  - Calc. *input - kernel* and *kernel - input*
  - Return the positive result
  - Parallel calculations => higher perform.
  - Higher circuit area



[Wang et al., 2021]

AdderNet implementation



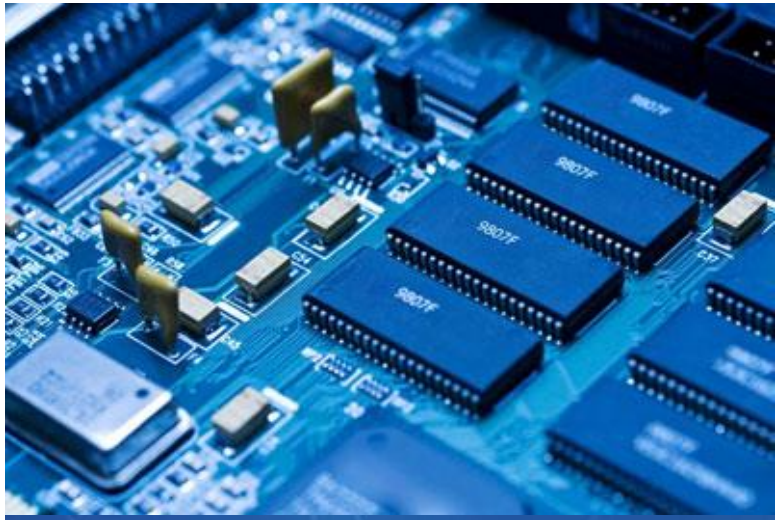


[Wang et al., 2021]

Hardware implementation

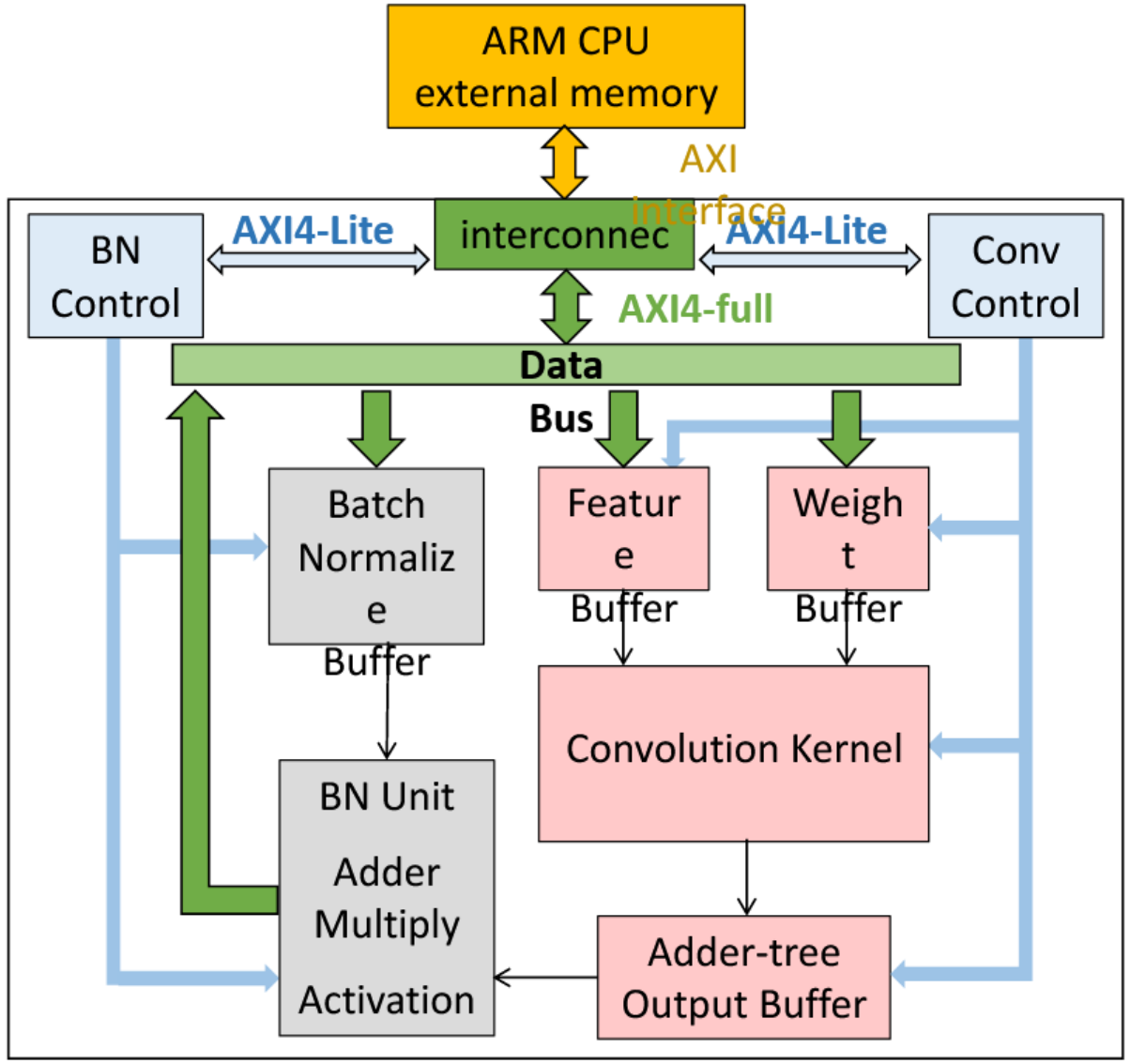
Field-programmable gate array (FPGA)

- Integrated circuit
- Reconfigurable hardware
  - Hardware description language
  - Interconnect programmable logic blocks
- Cheap, tiny, fast, reliable, easy to use
- Simulate hardware performance
  - without manufacturing expensive custom chips



[Thakur, 2019], [Eckstein, 2021]

Universal AdderNet accelerator

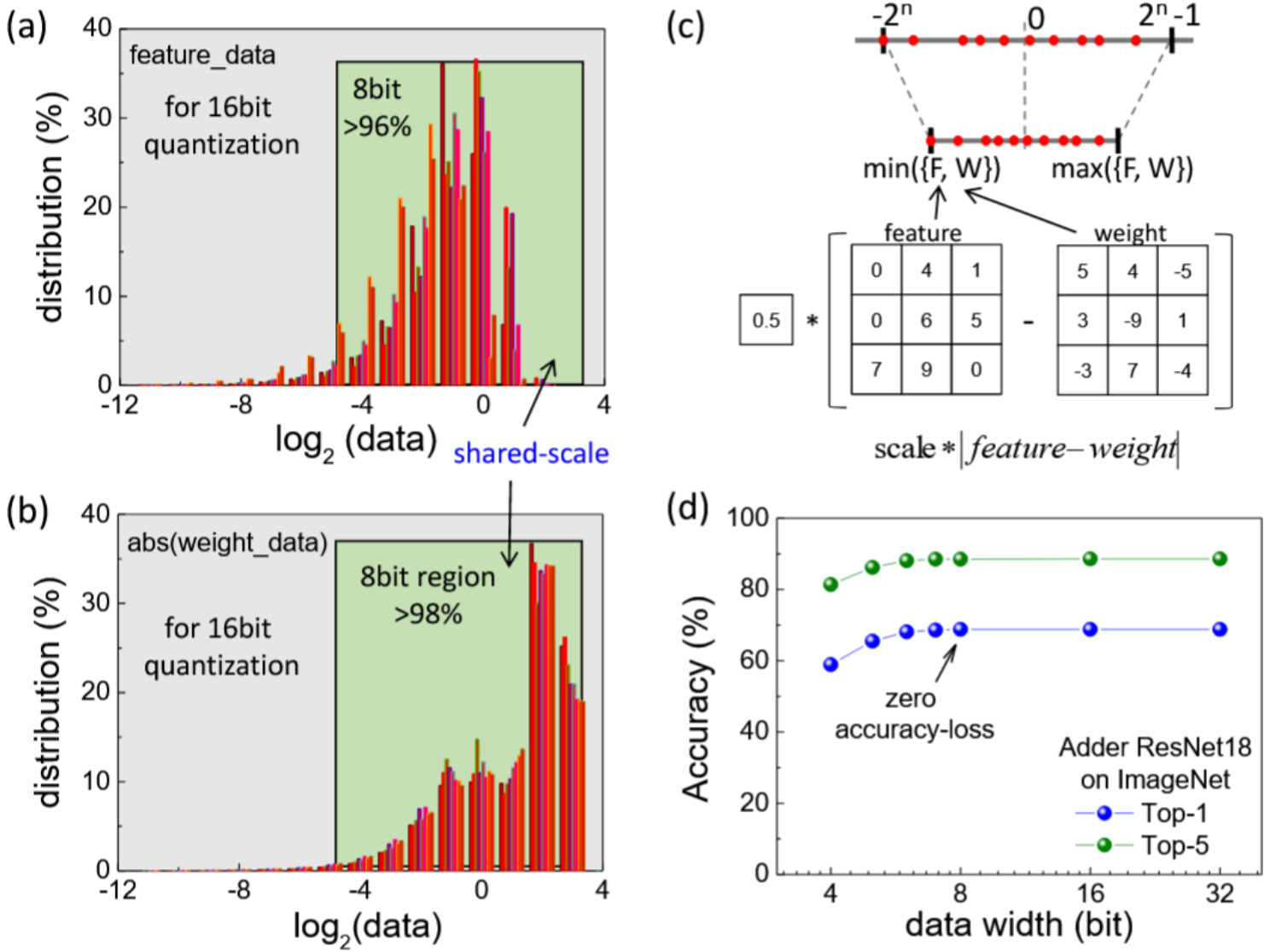


[Wang et al., 2021]

Quantization

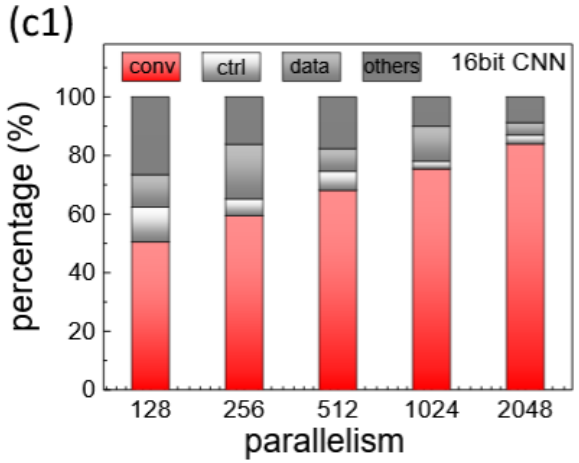
Data can be quantized in 8-bit integers

- Covers 98% of values with normalizing
- Zero accuracy-loss after training



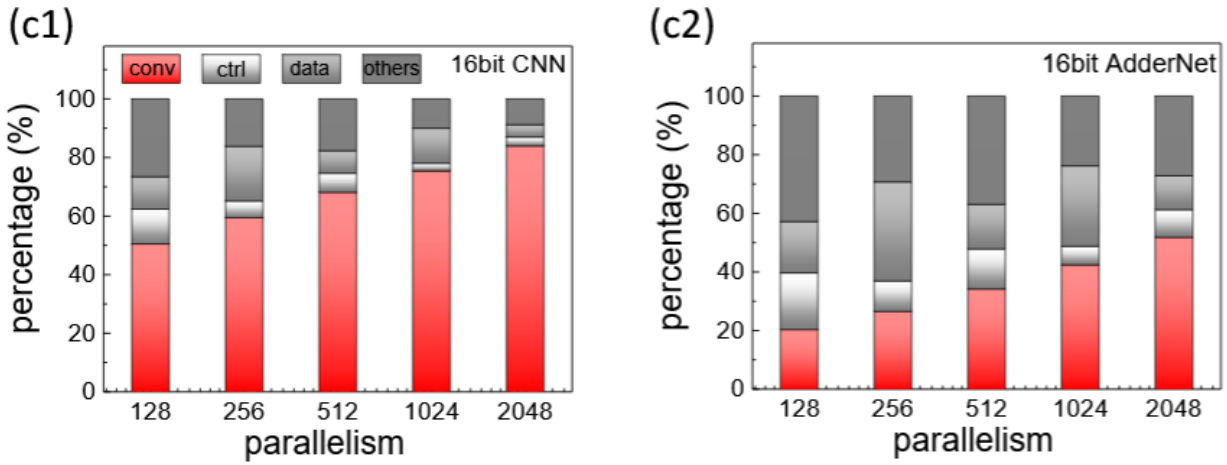
[Wang et al., 2021]

Parallel computation: Logic resource utilization



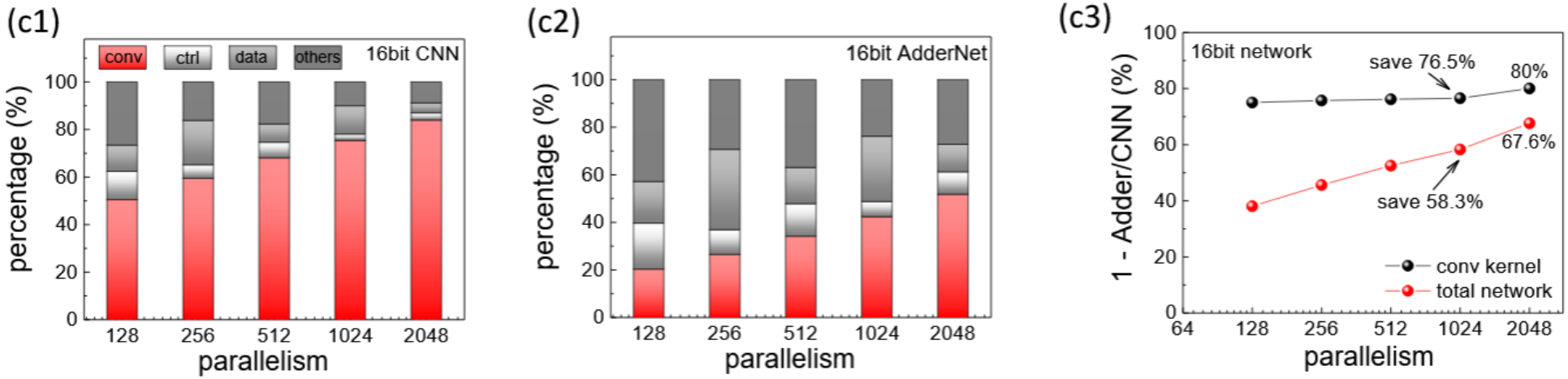
[Wang et al., 2021]

Parallel computation: Logic resource utilization



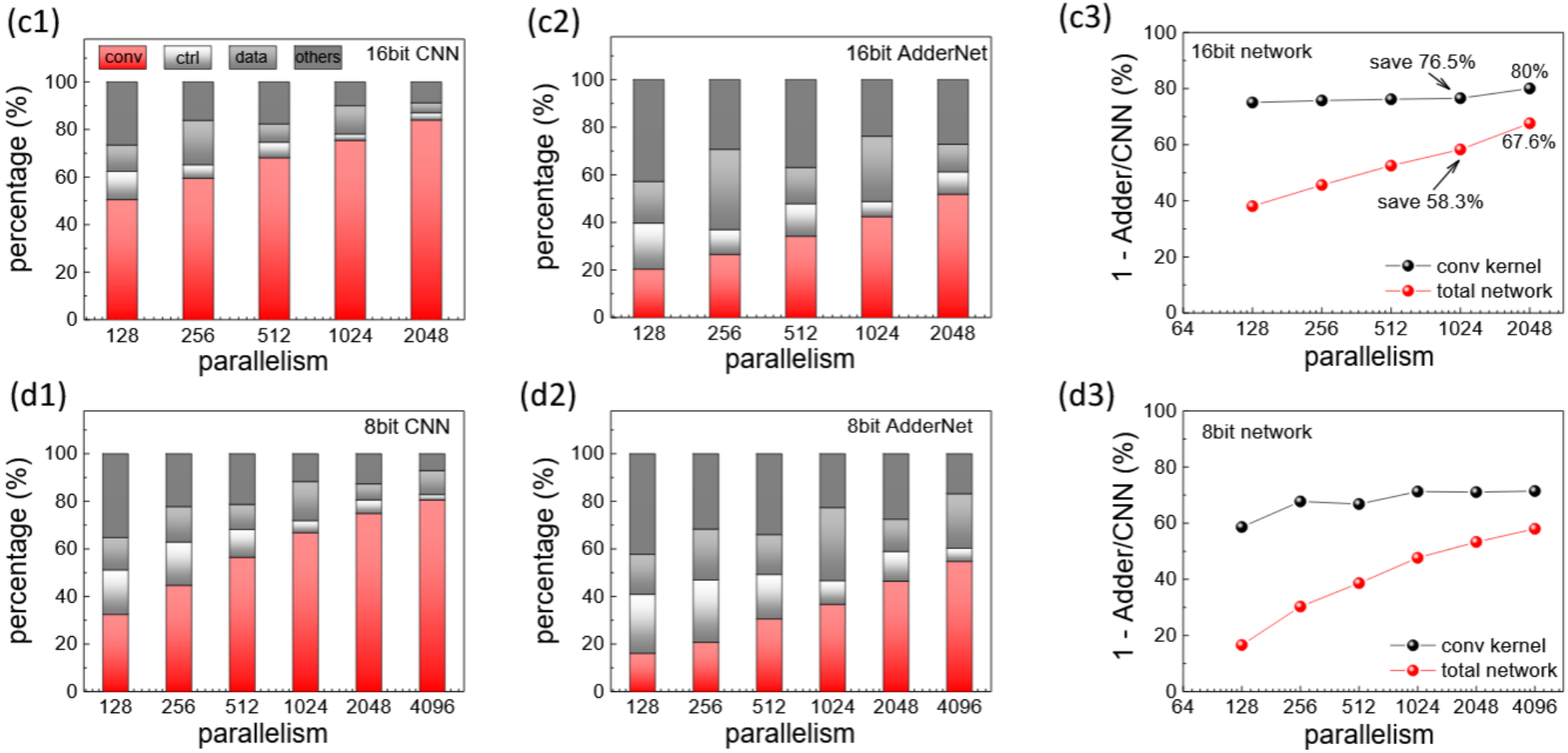
[Wang et al., 2021]

Parallel computation: Logic resource utilization



[Wang et al., 2021]

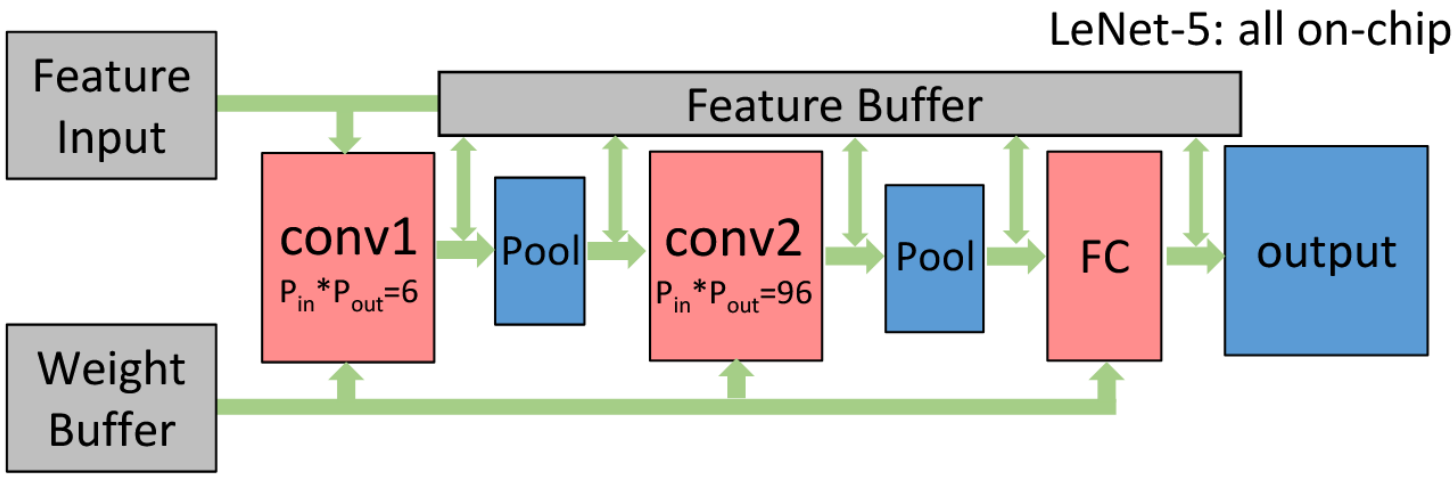
Parallel computation: Logic resource utilization



[Wang et al., 2021]

All on-chip design of AdderNet

- FPGA based CNN accelerator
- LeNet-5 network
- No data input/output needed, without off-chip data
- Data storage and calculations on-board



[Wang et al., 2021]

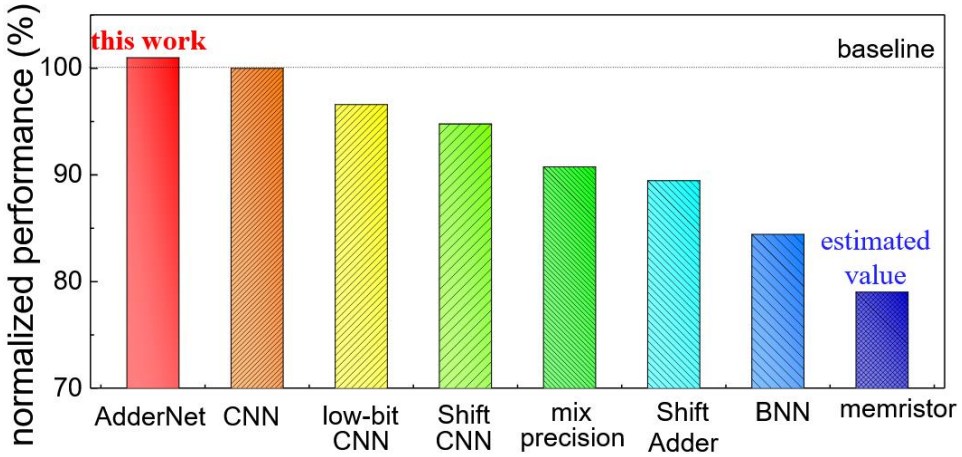
Accuracy, complexity, energy consumption, logic area

Comparison to other kernels

Comparison to other kernels

Performance / Prediction accuracy

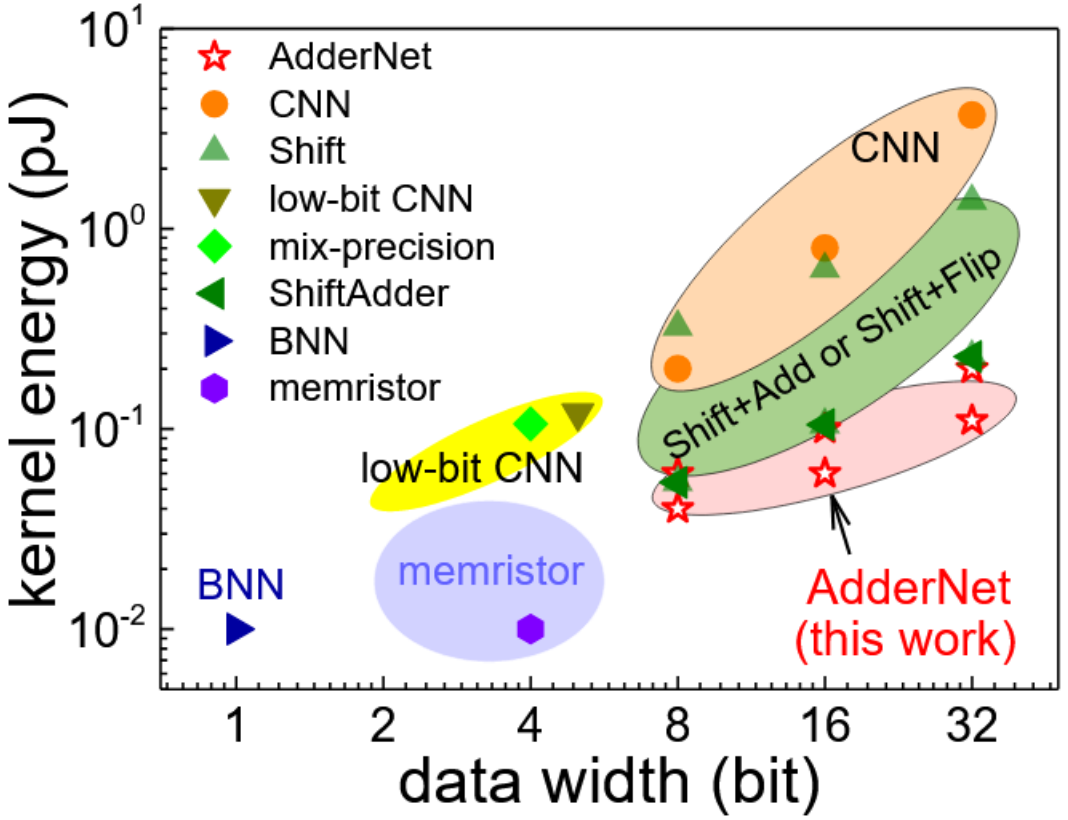
Neural Network	ResNet-18 on ImageNet		ResNet-50 on ImageNet		VGG-16 on ImageNet		ResNet-20 on CIFAR100
Intrinstic performance	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	accuracy
AdderNet (this work)	**68.80%**	**88.60%**	**76.80%**	**93.30%**	**71.40%**	**90.40%**	**69.93%**
CNN (baseline)	**69.76%**	**89.08%**	**76.13%**	**92.86%**	**71.59%**	**90.38%**	**68.75%**
low-bit CNN (5bit)	65.00%	85.90%	70.10%	89.70%	-	-	-
XNOR (BNN)	51.20%	73.20%	55.80%	78.40%	-	-	50.50%
DeepShift (6bit-W)	65.63%	86.33%	75.29%	92.55%	70.87%	90.09%	-



[Wang et al., 2021]

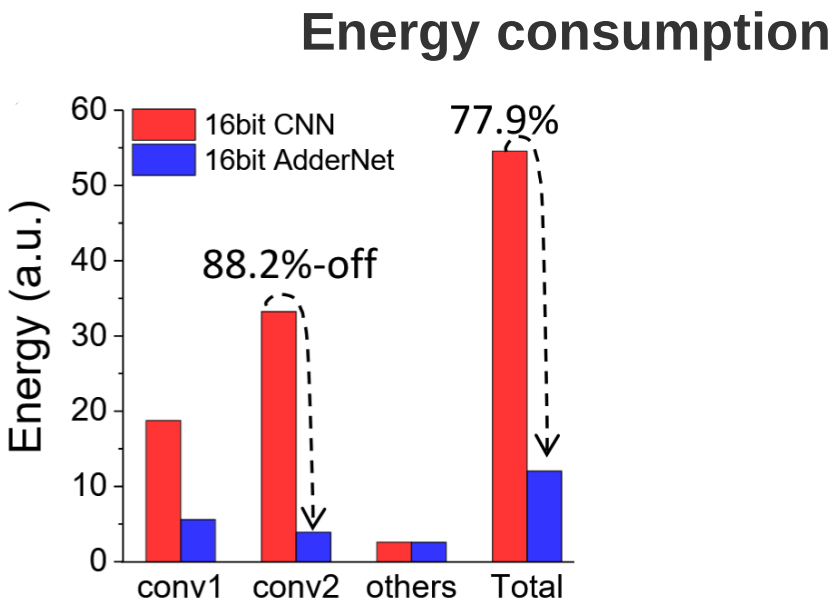
Comparison to other kernels

Energy consumption



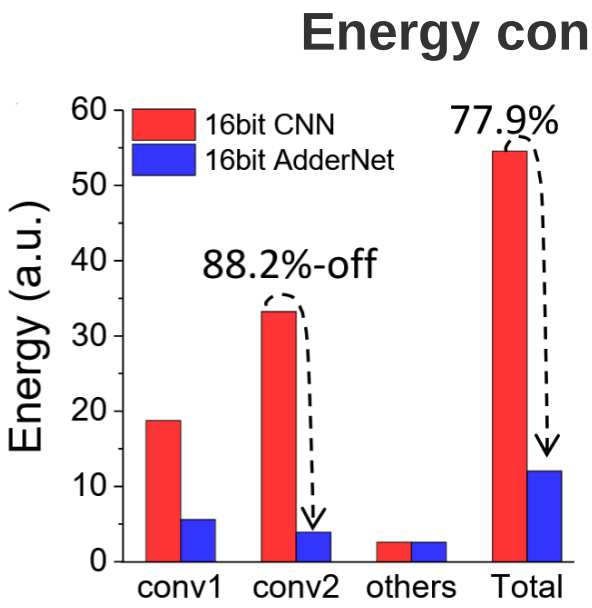
[Wang et al., 2021]

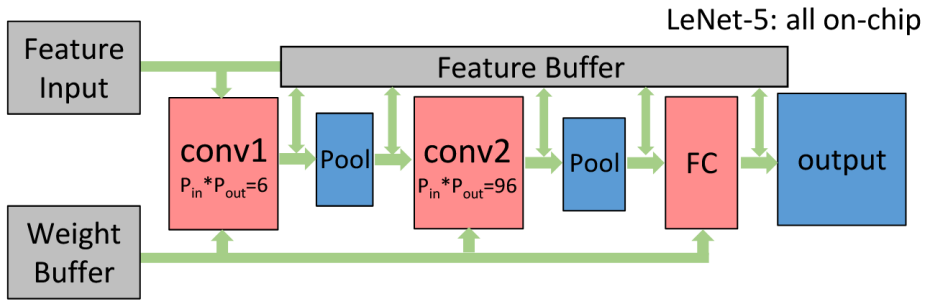
Comparison to other kernels



[Wang et al., 2021]

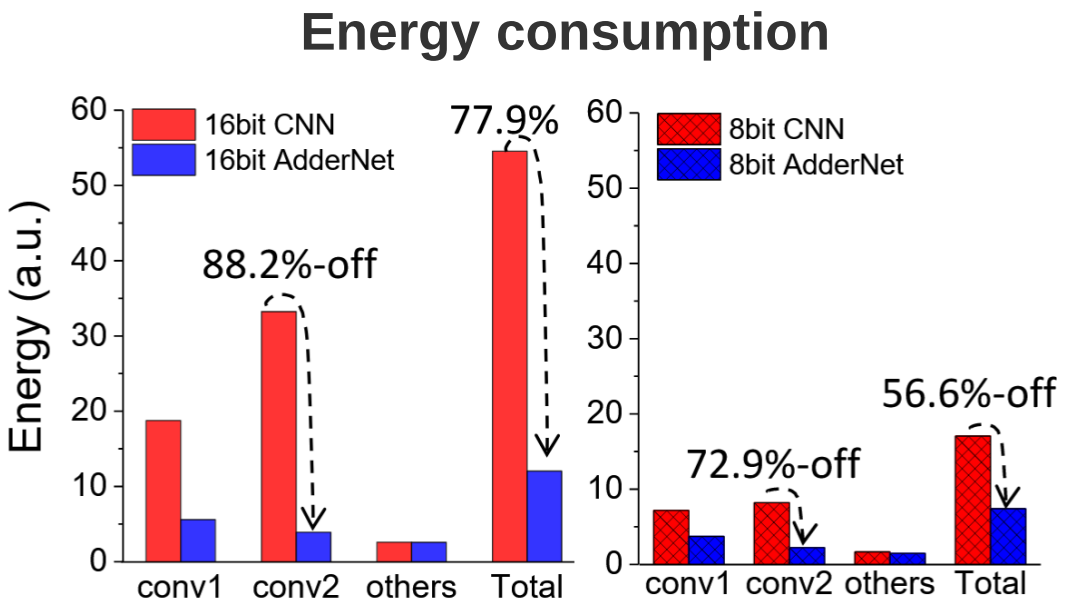
Comparison to other kernels

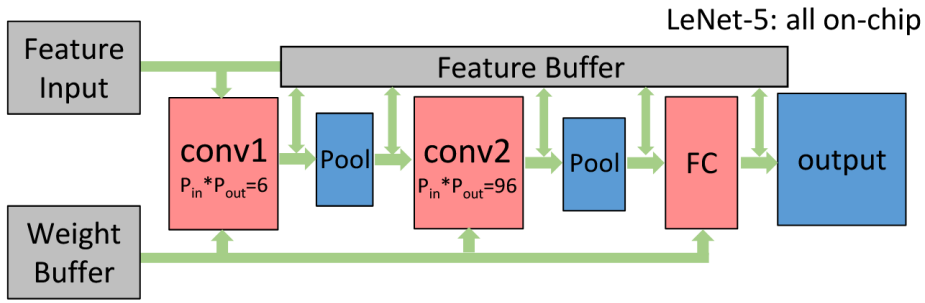




[Wang et al., 2021]

Comparison to other kernels

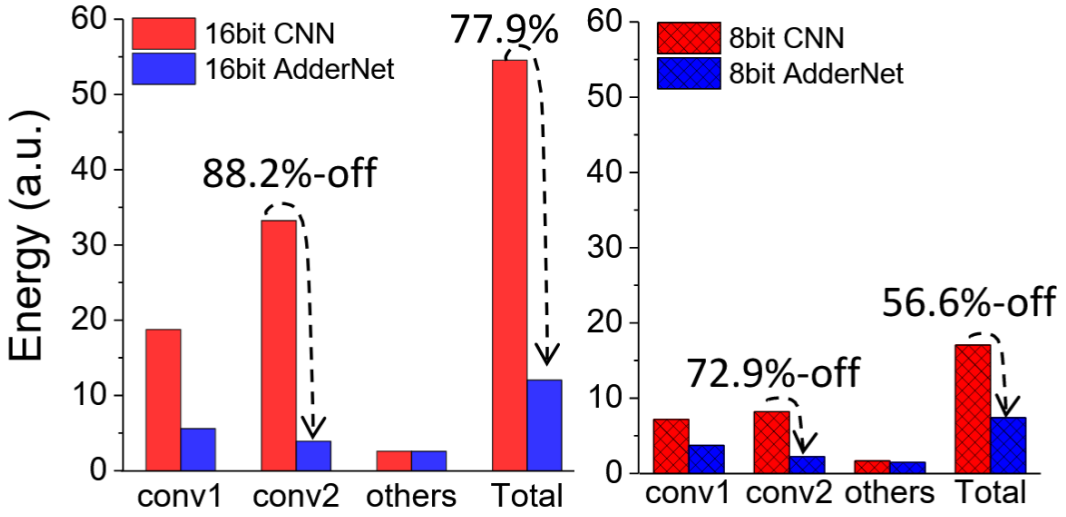




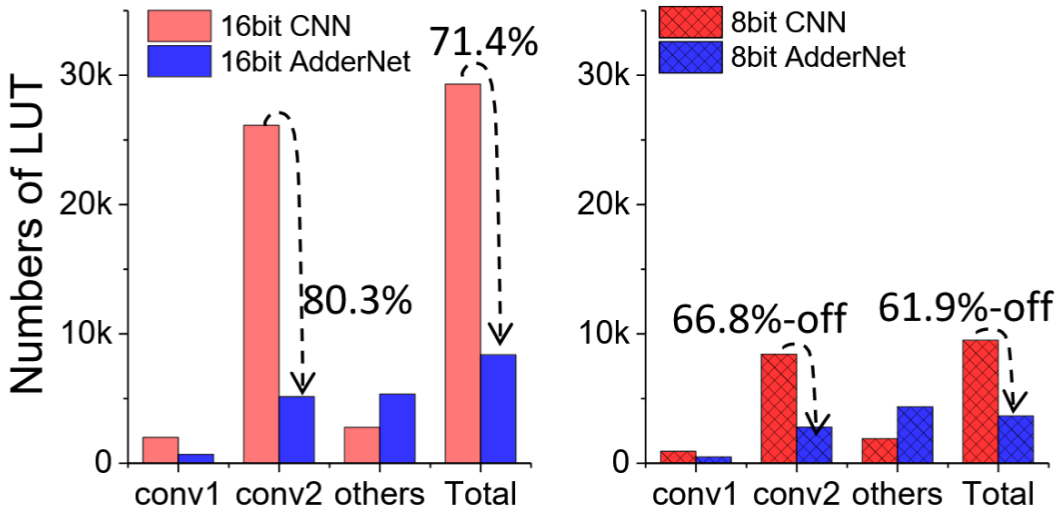
[Wang et al., 2021]

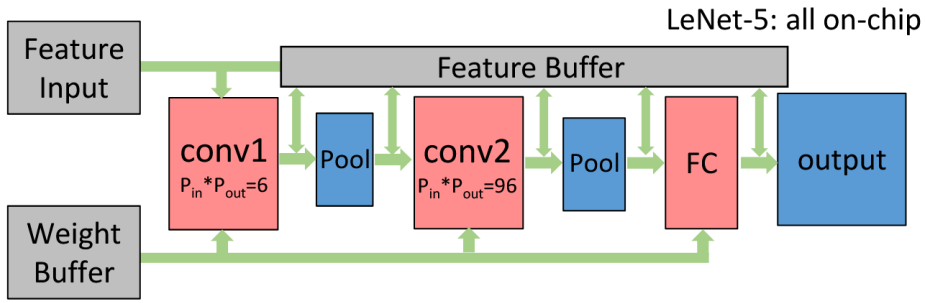
Comparison to other kernels

Energy consumption



Logic circuit area utilization





[Wang et al., 2021]

FUTURE RESEARCH &

FURTHER IMPLEMENTATIONS

Further implementations

DNN Training without Multiplications

- Add 2 floats like int to approx. multiplication (~12.5% accuracy)
- Competitive or equal performance on ResNet-50

ShiftAddNet

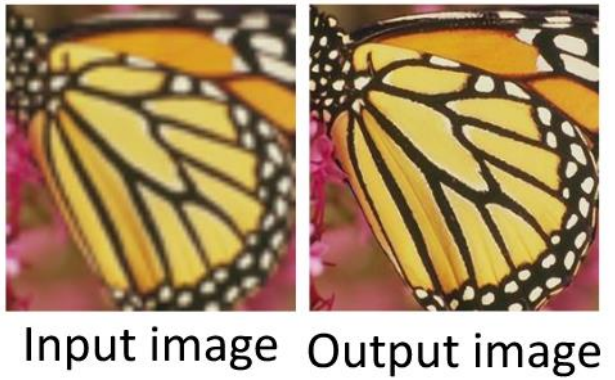
- Multiplication by a constant
- Sequence of bit-shifts and adds
- High performance due to hardware implementations

[Mogami, 2020], [You et al., 2020]

Further implementations

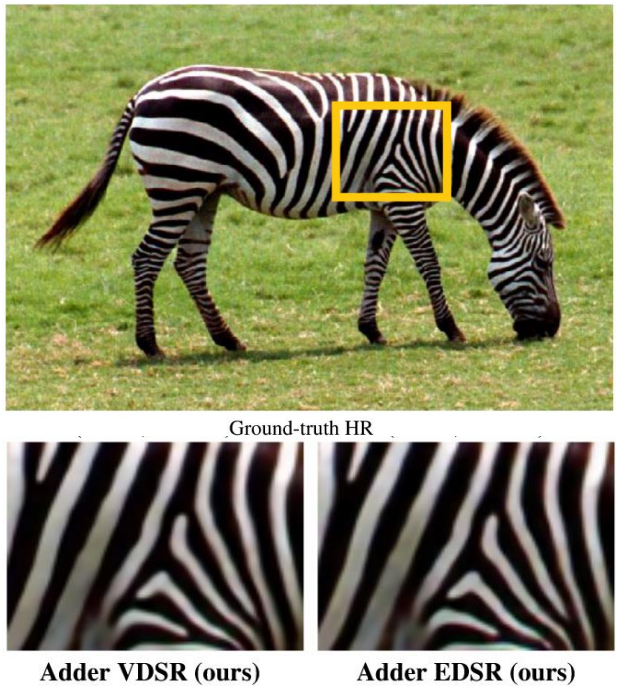
Adder Super Resolution

- Use AdderNets with shortcuts as enhancements
- Super-resolution network
- Achieves similar performance as normal CNNs



Kernel Based Progressive Distillation for AdderNNs

- Train normal CNN as teacher network
- Teacher guides learning of AdderNet
- Both have same architecture
- Better performance than directly training AdderNet



[Song et al., 2020], [Xu et al., 2020]

Conclusion

Conclusion

- Possible to create CNNs without multiplications
  - Needs less resources
    - 16% faster
    - 47.85% - 77.9% less power consumption
    - 67.4% - 71.4% less logic resource utilization
  - Competitive results
  - Implementable in FPGA, no custom manufacturing
  - Useful for AI in embedded systems
- Accuracy could be different on other network types
- Recommended kernel depends on use case / hardware req.
- Existing alternatives to avoid multiplications

[Wang et al., 2021]

Sources

- Cysmith. (2018, August 3). cysmith/neural-style-tf. GitHub. https://github.com/cysmith/neural-style-tf.
- Eckstein, M. (2021, June 2). Adaptierbare Embedded-Systeme erfolgreich entwickeln. https://www.elektronikpraxis.vogel.de/adaptierbare-embedded-systeme-erfolgreich-entwickeln-a-1027845/.
- Ferguson, M., Ak, R., Lee, Y. T. T., & Law, K. H. (2017, December). Automatic localization of casting defects with convolutional neural networks. In 2017 IEEE international conference on big data (big data) (pp. 1726-1735). IEEE.
- Greenholt, V. (2020). Social Network for Programmers and Developers. RSS. https://morioh.com/p/6349bbe063b2.
- Mogami, T. (2020). Deep Neural Network Training without Multiplications. arXiv preprint arXiv:2012.03458.
- Shaw, K. (2019, October 29). PBS Science Show NOVA Shines its Spotlight on Self-Driving Cars. Robotics Business Review. https://www.roboticsbusinessreview.com/unmanned/unmanned-ground/pbs-science-show-nova-shines-its-spotlight-on-self-driving-cars/.
- Song, D., Wang, Y., Chen, H., Xu, C., Xu, C., & Tao, D. (2020). Addersr: Towards energy efficient image super-resolution. arXiv preprint arXiv:2009.08891.
- Thakur, A. (2019, July 12). Field Programmable Gate Array (FPGA) : An Overview. Engineers Garage. https://www.engineersgarage.com/field-programmable-gate-array-fpga-an-overview/.
- Wang, Y., Huang, M., Han, K., Chen, H., Zhang, W., Xu, C., & Tao, D. (2021). AdderNet and its Minimalist Hardware Design for Energy-Efficient Artificial Intelligence. arXiv preprint arXiv:2101.10015.
- Xu, Y., Xu, C., Chen, X., Zhang, W., Xu, C., & Wang, Y. (2020). Kernel based progressive distillation for adder neural networks. arXiv preprint arXiv:2009.13044.
- You, H., Chen, X., Zhang, Y., Li, C., Li, S., Liu, Z., ... & Lin, Y. (2020). Shiftaddnet: A hardware-inspired deep network. arXiv preprint arXiv:2010.12785.

Questions?

Content

Introduction / Motivation

- ML, CNNs, Convolution kernels
- CNN Drawbacks, alternative designs

AdderNet convolution

- Adder core design

Hardware implementation

- FPGA, quantization, parallelization, …

Comparison to other kernels

- Accuracy, complexity, energy consumption, logic area

Further implementations, future research

Conclusion

Convolution kernel formulas

Traditional multiply core

$$g(x,y) = \omega * f(x,y) = \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} \omega(dx,dy) f(x+dx, y+dy)$$

Shift core

$$g(x,y) = \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} \omega_{sign}\,(dx,dy) 2^{\omega_{exponent}(dx,dy)} f(x+dx, y+dy)$$

XNOR logic core

$$g(x,y) = \sum_{dx=-a}^{a} \sum_{dy=-b}^{b} XNOR(\omega(dx,dy), f(x+dx, y+dy))$$

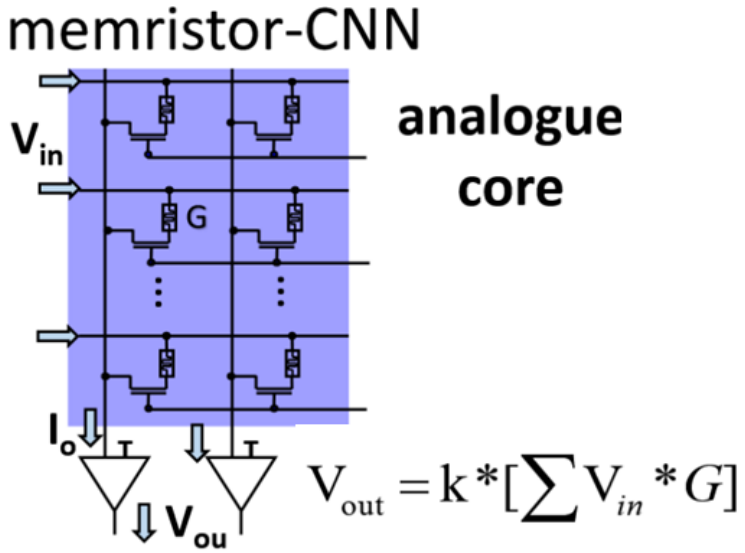
Adder core

$$g(x,y) = -\sum_{dx=-a}^{a} \sum_{dy=-b}^{b} |\omega(dx,dy) - f(x+dx, y+dy)|$$

[Wang et al., 2021]

Memristor-CNN

- Analogue multiplier using Memristor
- Analogue implementation of multiplier
- Consists of 2 parallel 1-Transistor-1-Memristor, 1 Differential circuit



[Wang et al., 2021]

Circuit complexity

- Kernels are made up of…

▪ CNN multiplication	1 multiplier
▪ Shift-CNN	1 Serial-Shift-Register, 1 Multiplexer, 1 N-bit*1-bit Multiplier (sign) (more Shift-Registers and Adders if data-width of weight > 1)
▪ XNOR	several AND/NAND gates
▪ AdderNet	2 Adders (or 1 Comparator, 1 Adder)

[Wang et al., 2021]

Performance / Prediction accuracy

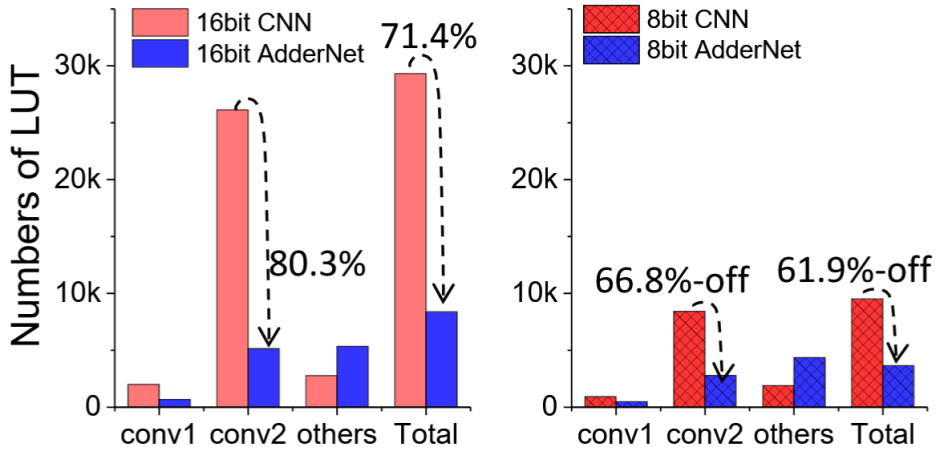
Kernel operation	Neural Network	ResNet-18 on ImageNet		ResNet-50 on ImageNet		VGG-16 on ImageNet		ResNet-20 on CIFAR100
	Intrinstic performance	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	accuracy
adder	**AdderNet (this work)**	**68.80%**	**88.60%**	**76.80%**	**93.30%**	**71.40%**	**90.40%**	**69.93%**
multiplication	**CNN (baseline)**	**69.76%**	**89.08%**	**76.13%**	**92.86%**	**71.59%**	**90.38%**	**68.75%**
	low-bit CNN (5bit)	65.00%	85.90%	70.10%	89.70%	-	-	-
bit-level operation	mix-precision	59.47%	-	-	-	-	-	62.36%
	XNOR (BNN)	51.20%	73.20%	55.80%	78.40%	-	-	50.50%
analogue $I_{out}=\sum(V_{in}/R)$	memristor-CNN	-						
shift (with sign flip) (or with adder)	DeepShift (1bit-W)	41.53%	67.29%	41.30%	65.10%	65.25%	86.30%	
	DeepShift (6bit-W)	65.63%	86.33%	75.29%	92.55%	70.87%	90.09%	-
	ShiftAdder	-	-	-	-	-	-	61.50%

[Wang et al., 2021]

Logic circuit area utilization

data width	AdderNet		CNN		DeepShift		XNOR	memristor
	1Comparator + 1Adder	2 Adders	multiplier	low-bit multiplier	1bit-weight	6bit-weight		
1bit							~1	
4bit				~18				~2
8bit	58	72	282					
16bit	112	134						
32bit	227	274	3495					
FP32bit		8368	7700					

Fig. 12: Detailed comparison of logic circuit area in different convolution kernels [33].



[Wang et al., 2021]