



Training Low Bit-width Convolutional Neural Networks on RRAM

Yi Cai, Tianqi Tang, Lixue Xia, Ming Cheng, Zhenhua Zhu,
Yu Wang, Huazhong Yang

Dept. of E.E., Tsinghua National Laboratory for Information
Science and Technology (TNList),
Tsinghua University, Beijing, China
E-mail: yu-wang@tsinghua.edu.cn



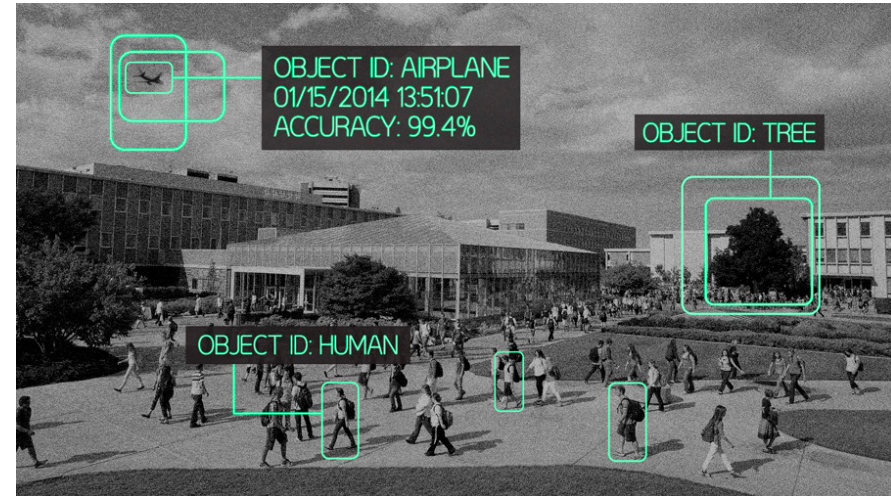
Outline

- Background & Motivation
- RRAM-based Low-bit CNN Training System
 - Overall Framework
 - Quantization and AD/DA Conversion
- Experimental Results
- Conclusion and Future Work
- Reference

Convolutional Neural Network



Video Tracking



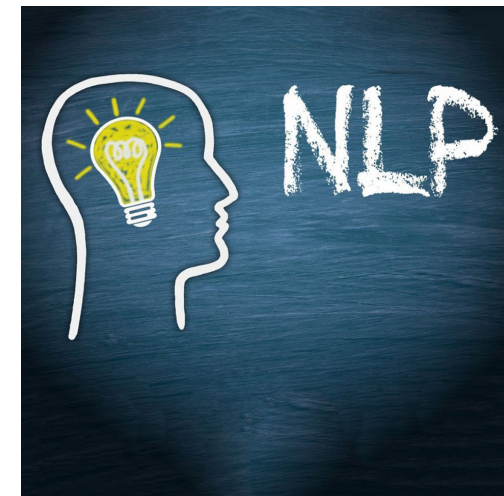
Object Detection

CNN

Speech Recognition



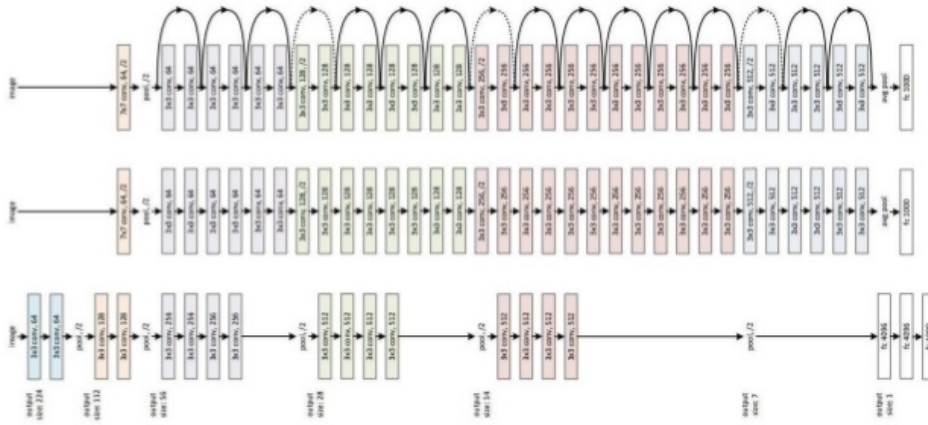
APPLICATIONS



Natural Language Processing

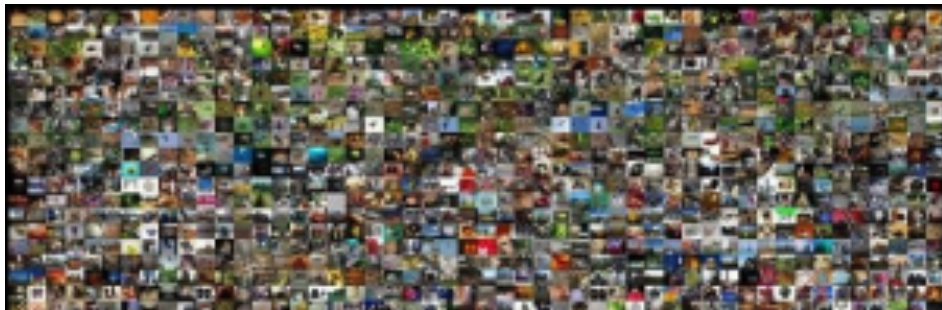
Energy-efficient platforms are required

Training CNNs is time-and-energy-consuming

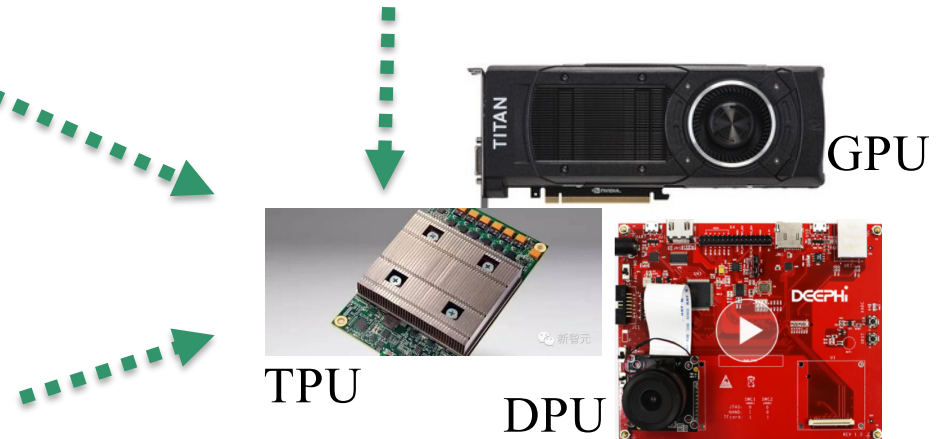


Real-time On-line NN Applications

Deeper Neural Networks



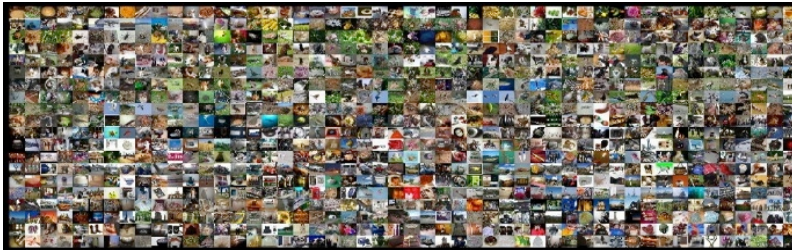
Exponentially Growing Data



Require Faster, more energy-efficient
NN Computing Accelerators

Accelerating Inference is not enough

Big Data



Training Data



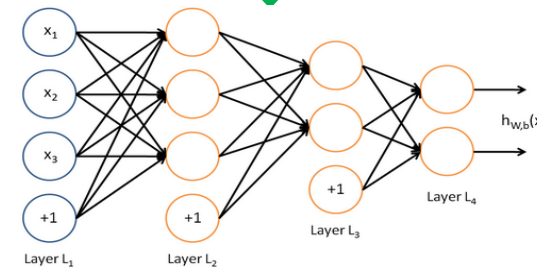
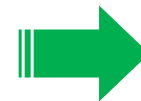
Server



Training

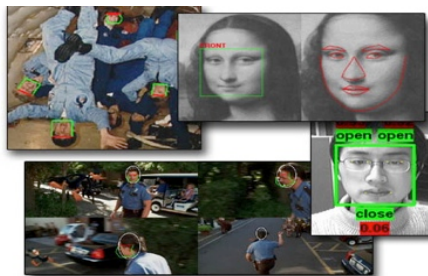
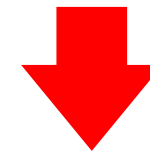


Difficult to fully generalized



Application scenarios vary

Deploy



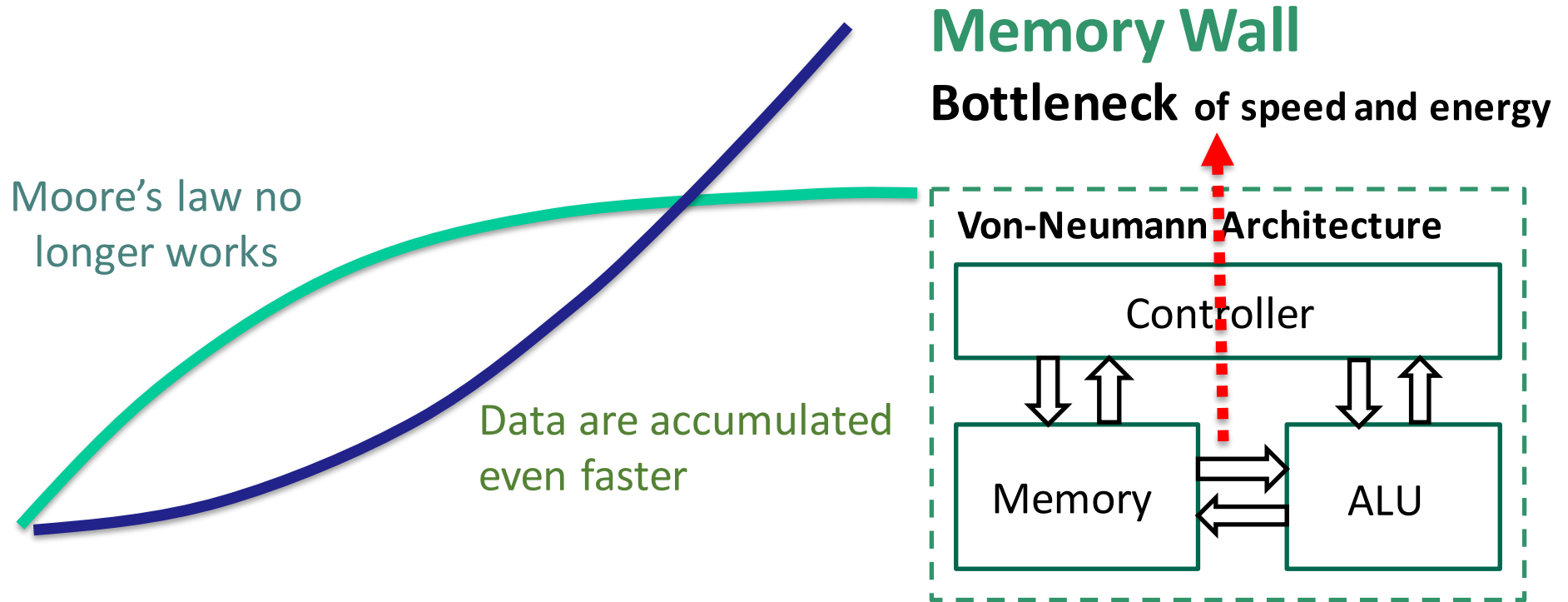
Inference



On-line training and learning are necessary

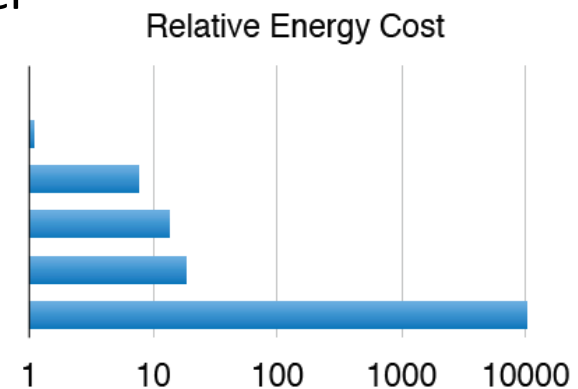
While most embedded devices are energy-limited

Memory Wall in von Neumann Architecture



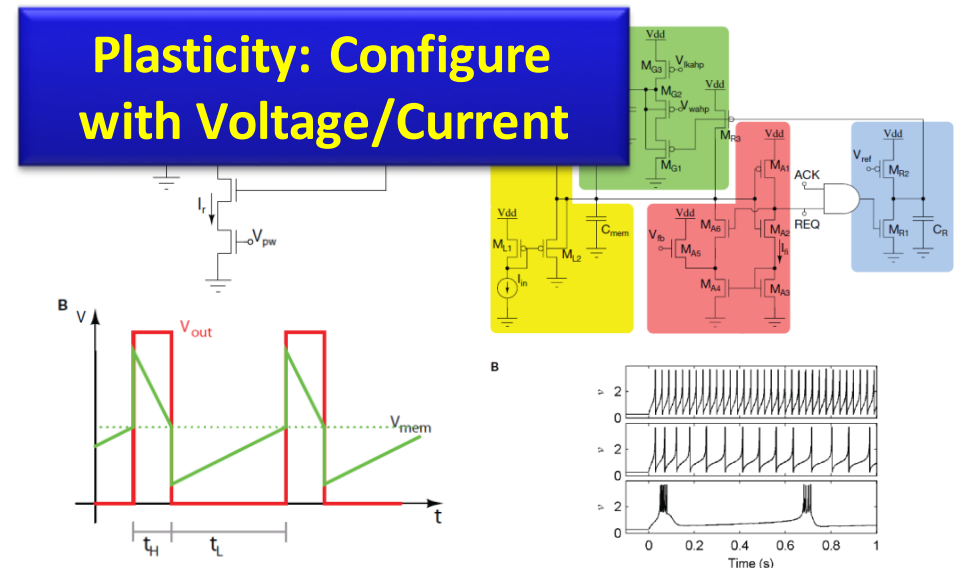
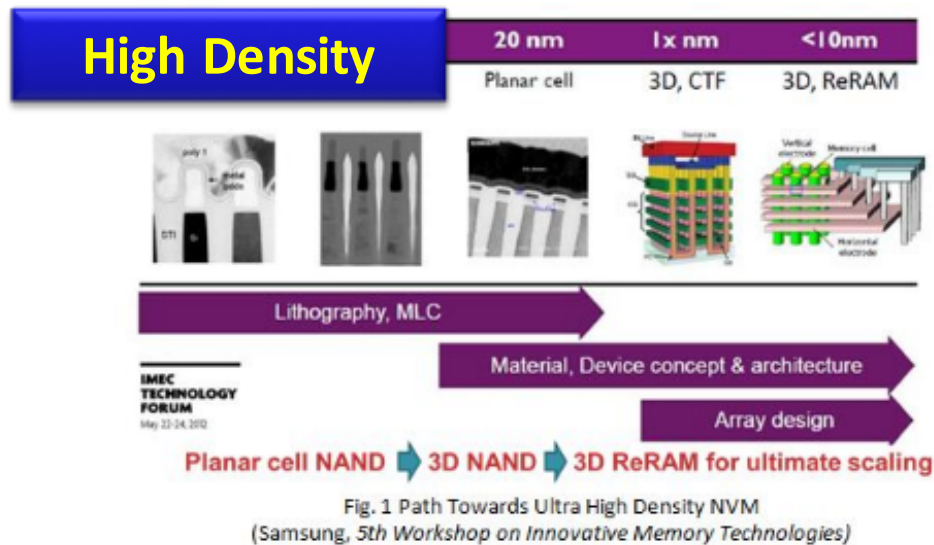
Too many memory accesses result in high power

Operation	Energy [pJ]	Relative Cost
16 bit int ADD	0.06	1
16 bit FP ADD	0.45	8
16 bit int MULT	0.8	13
16 bit FP MULT	1.1	18
32b LPDDR2 DRAM	640	10667



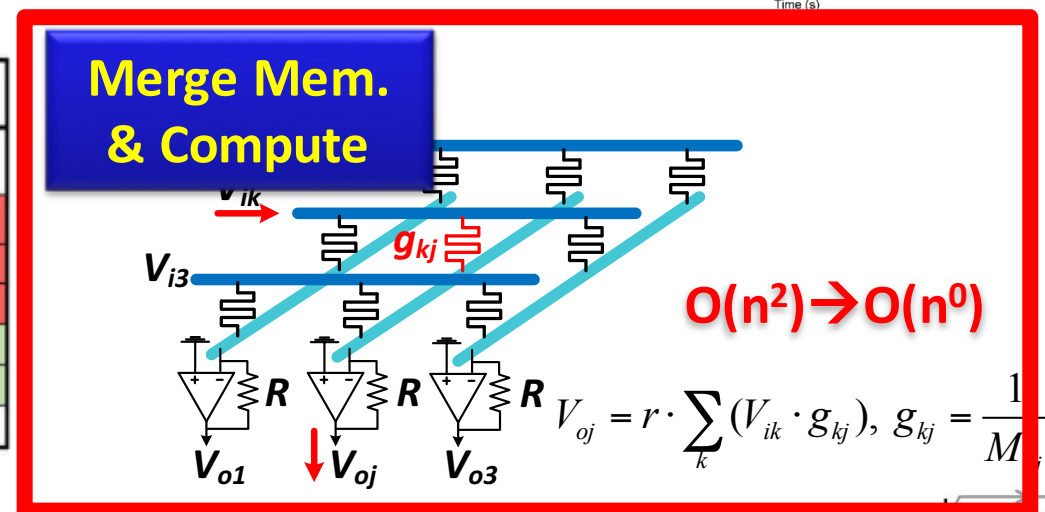
RRAM-based NN Computing Systems

RRAM has become a promising candidate for NN training

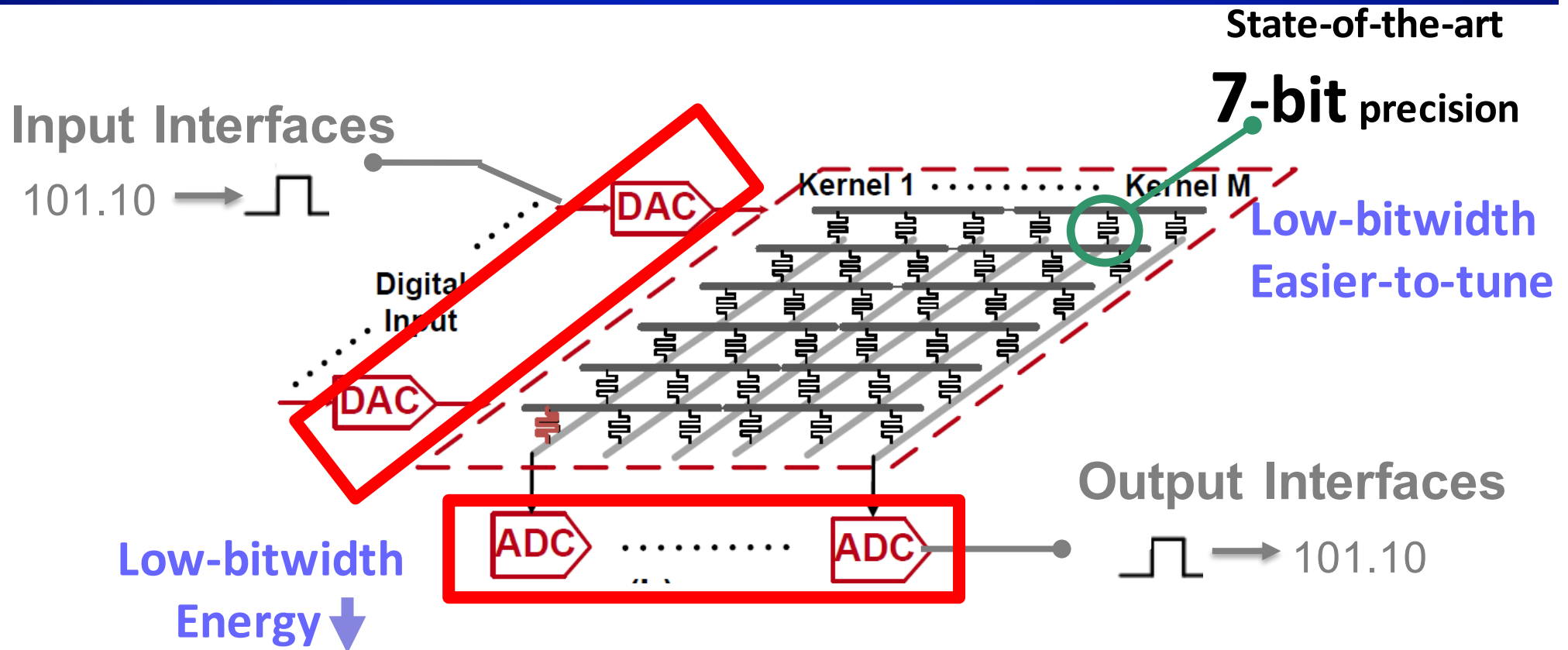


Non Volatile

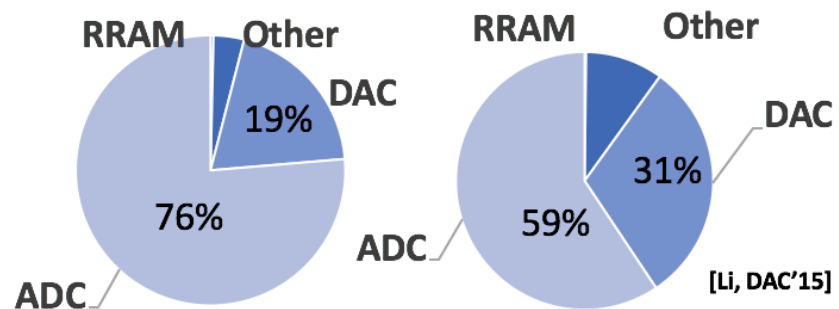
	or	PCM	STT-RAM	DRAM	Flash	HD
Chip area per bit (F ²)	4	8-16	14-64	6-8	4-8	n/a
Energy per bit (pJ) ²	0.1-3	2-100	0.1-1	2-4	10 ³ -10 ⁴	10 ⁶ -10 ⁷
Read time (ns)	<10	20-70	10-30	10-50	25,000	5-8x10 ⁶
Write time (ns)	20-30	50-500	13-95	10-50	200,000	5-8x10 ⁶
Retention	>10 years	<10 years	Weeks	<Second	~10 years	~10 years
Endurance (cycles)	~10 ¹²	10 ⁷ -10 ⁸	10 ¹⁵	>10 ¹⁷	10 ³ -10 ⁶	10 ¹⁵ ?
3D capability	Yes	No	No	No	Yes	n/a



High-precision data & weights are not supported in RRAM-based Systems



8bit ADC/DACs consume **>85%** energy of the system



Area & Power of RCS with 8bit ADC/DACs

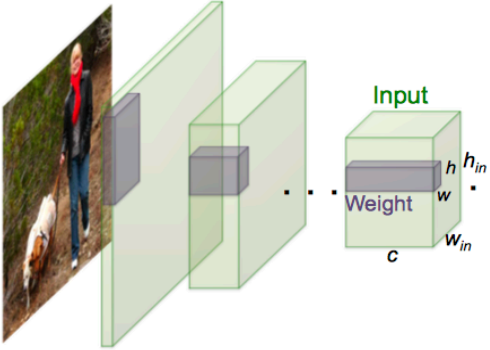
Training Low Bit-width CNN is Feasible

Binarized Neural Networks (BNN) [courbariaux2016binarynet]

- Training Neural Networks with **Weights and Activations** Constrained to +1 or -1
- Binary Weights and Activations (DNN)

XNOR-Net [rastegari2016xnor]

- Both **the filters and the input** to convolutional layers are binary (CNN)



	Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs $\begin{bmatrix} 0.11 & -0.21 & \dots & -0.34 \\ -0.25 & 0.61 & \dots & 0.52 \end{bmatrix}$ Real-Value Weights $\begin{bmatrix} 0.12 & -1.2 & \dots & 0.41 \\ -0.2 & 0.5 & \dots & 0.68 \end{bmatrix}$	+ , - , ×	1x	1x	%56.7
Binary Weight	Real-Value Inputs $\begin{bmatrix} 0.11 & -0.21 & \dots & -0.34 \\ -0.25 & 0.61 & \dots & 0.52 \end{bmatrix}$ Binary Weights $\begin{bmatrix} 1 & -1 & \dots & 1 \\ -1 & 1 & \dots & 1 \end{bmatrix}$	+ , -	~32x	~2x	%56.8
Binary Weights & Binary Input (XNOR-Net)	Binary Inputs $\begin{bmatrix} 1 & -1 & \dots & -1 \\ -1 & 1 & \dots & 1 \end{bmatrix}$ Binary Weights $\begin{bmatrix} 1 & -1 & \dots & 1 \\ -1 & 1 & \dots & 1 \end{bmatrix}$	XNOR , bitcount	~32x	~58x	%44.2

DoReFa-Net [zhou2016dorefanet]

- Using low bitwidth parameter gradients to train CNNs
- Can accelerate both **training and inference**

Contributions

The main contributions of this work include:

- In this paper, we propose an RRAM-based low-bitwidth CNN training system. We also propose the algorithm of training low-bitwidth convolutional neural networks, to enable a RRAM-based system to implement on-chip CNN training. And quantization and AD/DA conversion strategies are proposed to adapt to RCS and improve the accuracy of CNN model.
- We explore the configuration space of combinations of bitwidth of activations, convolution outputs, weights, and gradients by experiments of training LeNet-5 and ResNet-20 on proposed system, testing over the MNIST and CIFAR-10 datasets respectively. Moreover, a tradeoff of balancing between energy overhead and prediction accuracy is discussed.
- We analyze the probability distribution of RRAM's stochastic disturbance and make experiments to explore the effects of the disturbance on CNN's training.

RRAM-based Low-bit CNN Training System

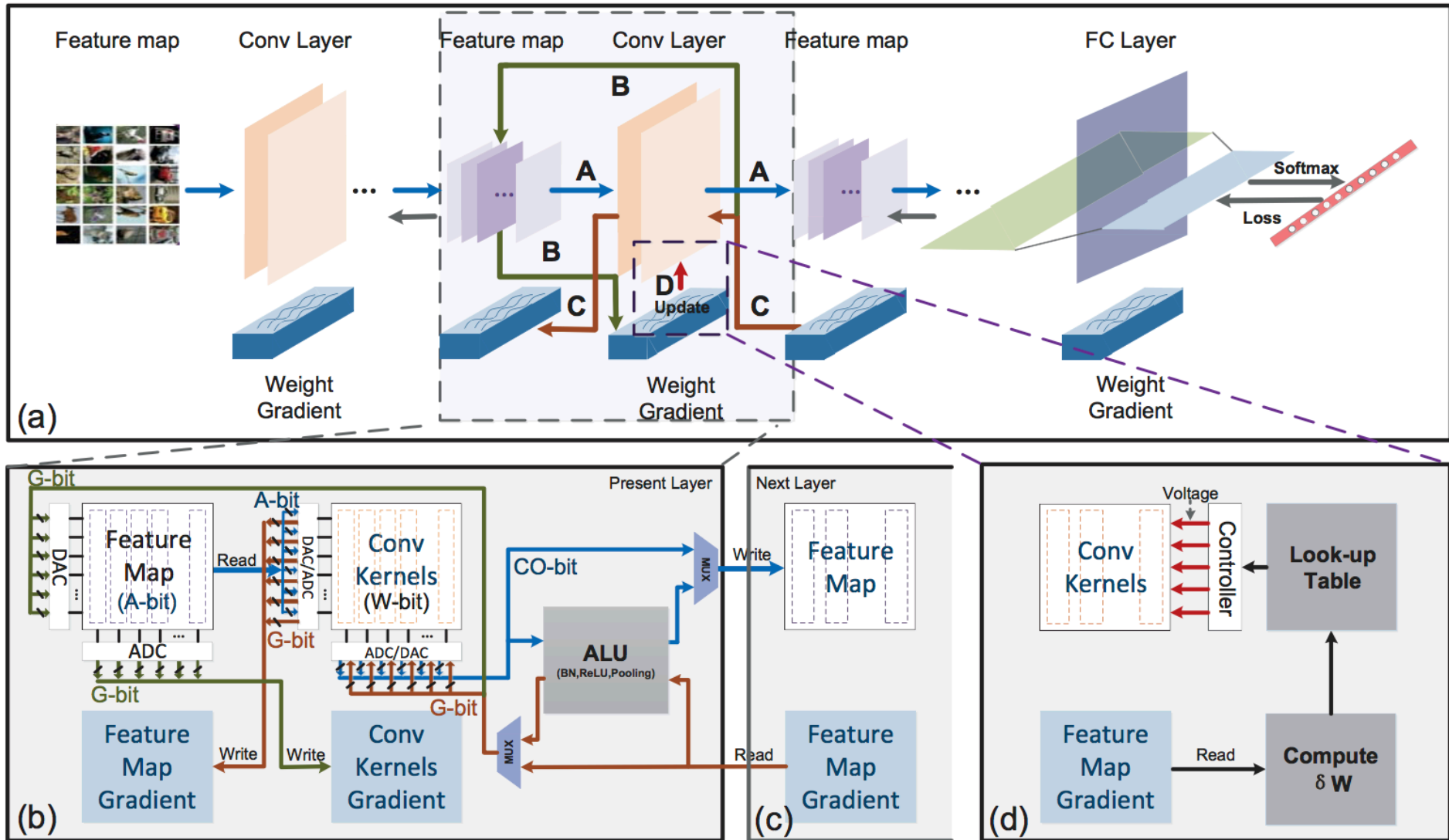
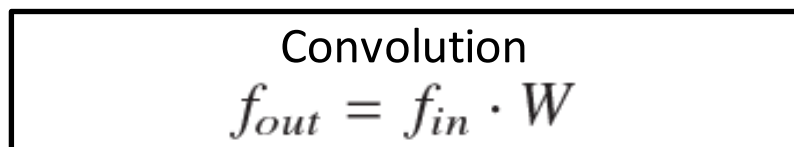
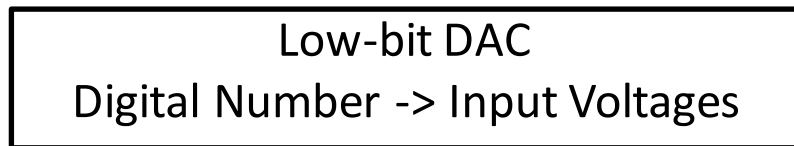


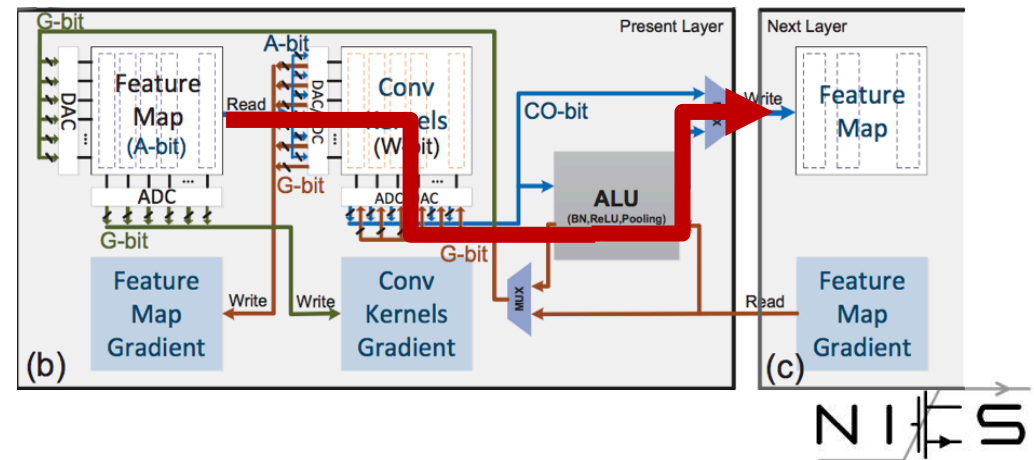
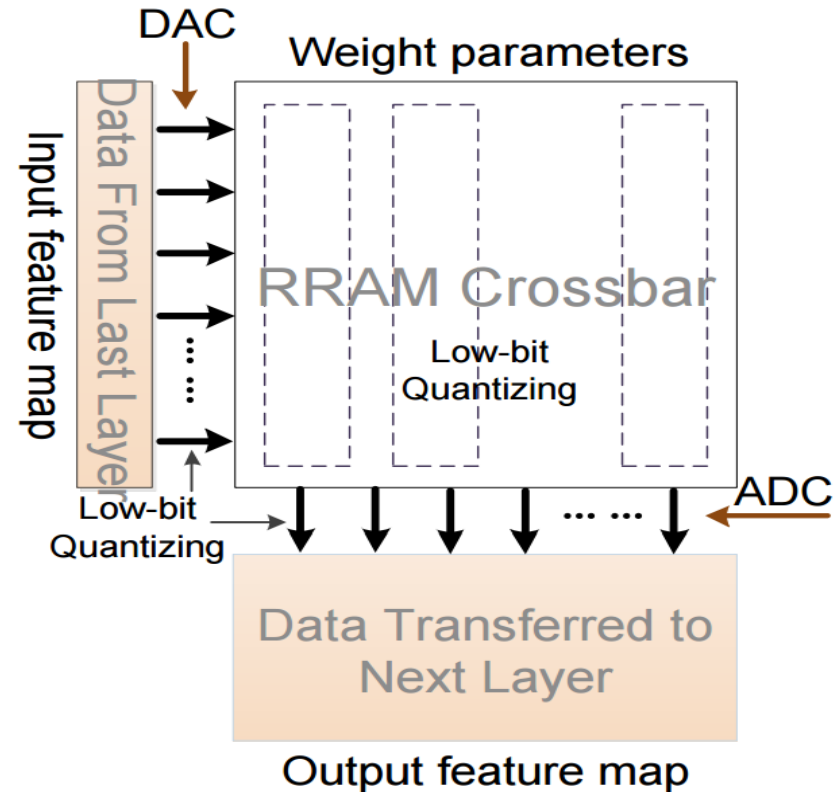
Fig. 2: Framework of RRAM-based low-bitwidth CNN training system.

Training Process-Inference

Feature Map from bottom layer



Cascaded by BatchNorm/ReLU/Pooling



Training Process-Backpropagation-1

Gradients from top layer
 ↓
 Backpropagated through Pooling/ReLU/BN

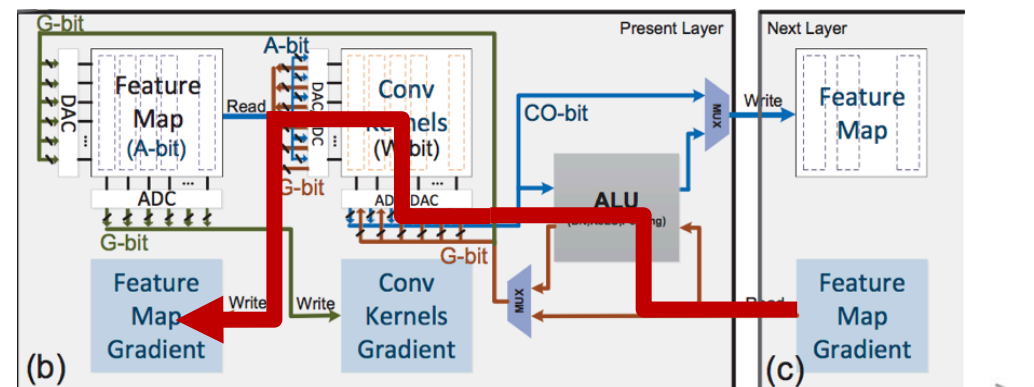
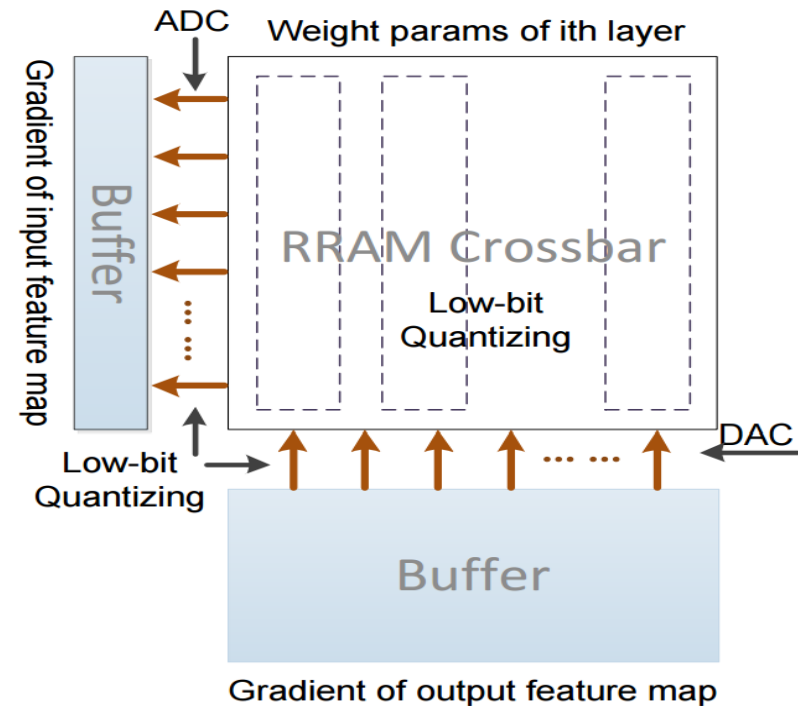
Low-bit DAC
 Digital Number -> Input Voltages

Gradient w.r.t input

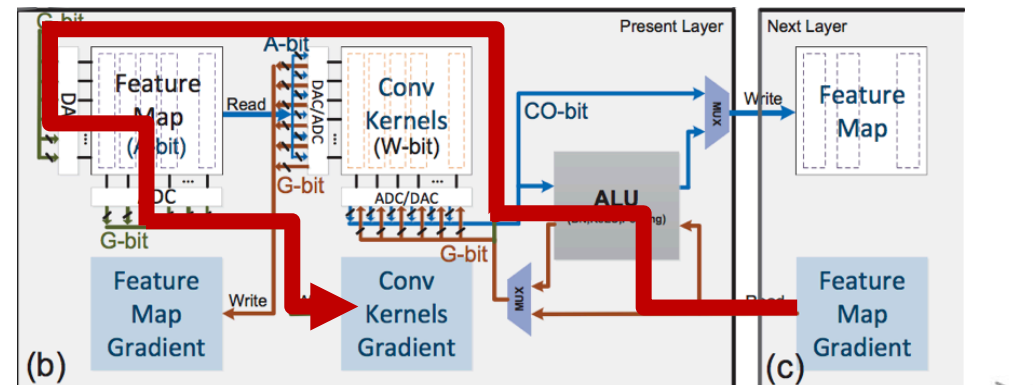
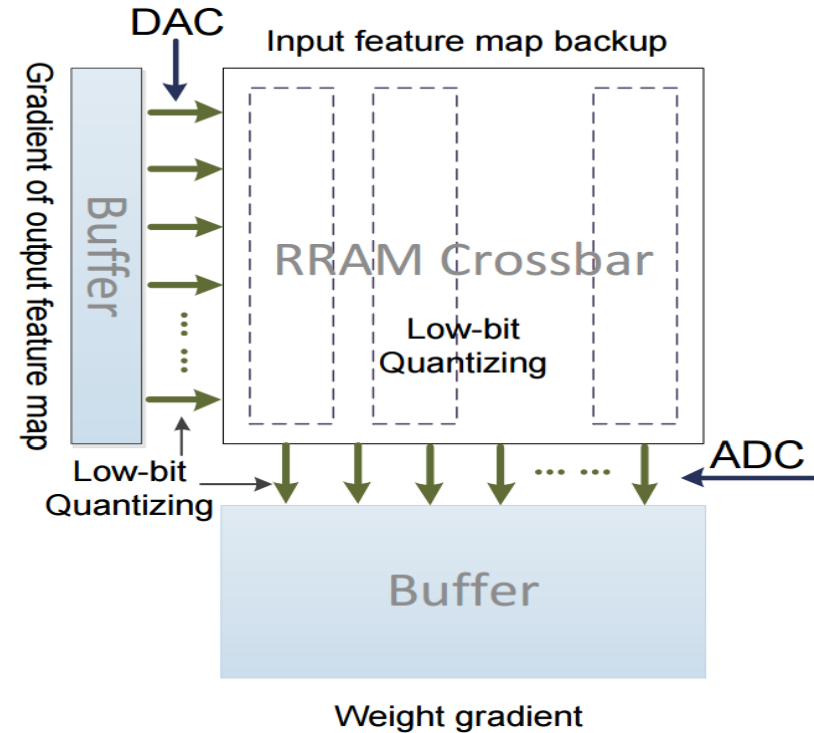
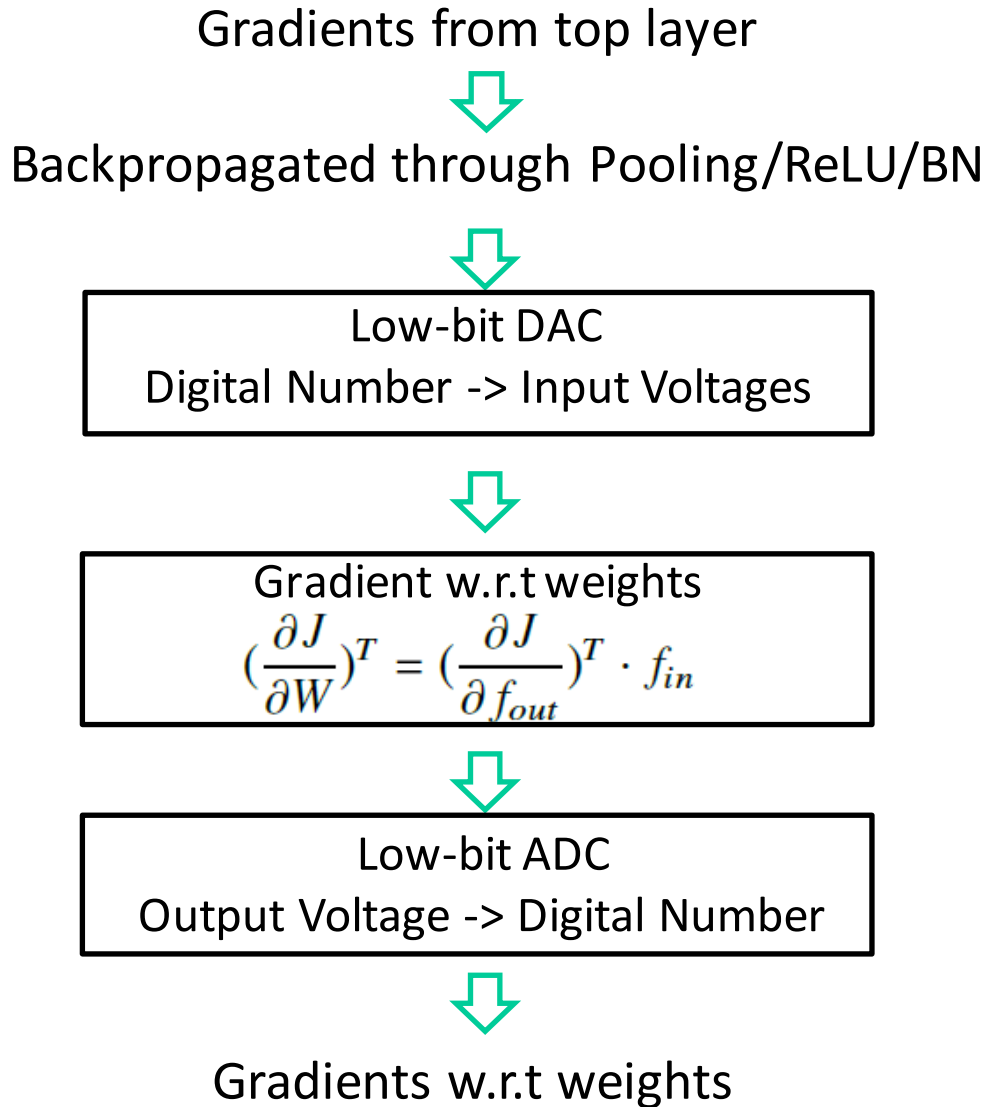
$$\frac{\partial J}{\partial f_{in}} = \frac{\partial J}{\partial f_{out}} \cdot W^T$$

Low-bit ADC
 Output Voltage -> Digital Number

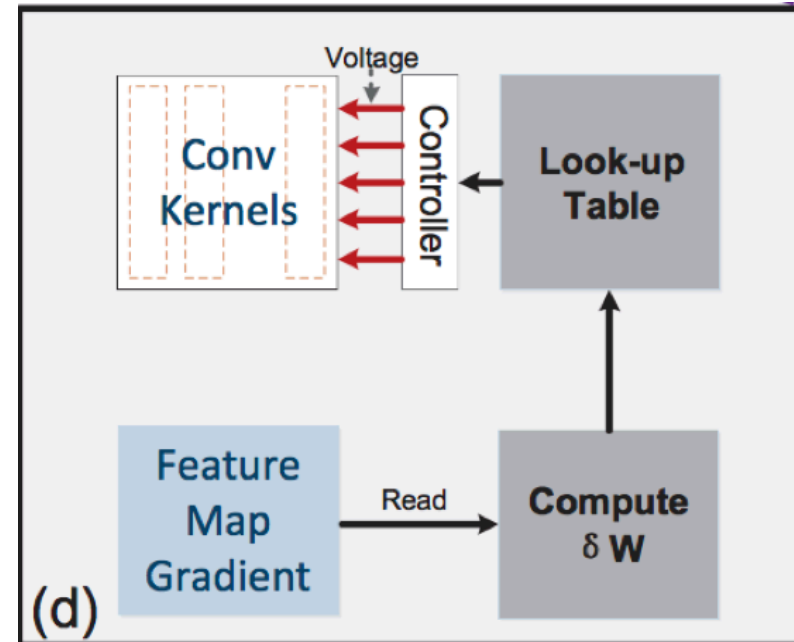
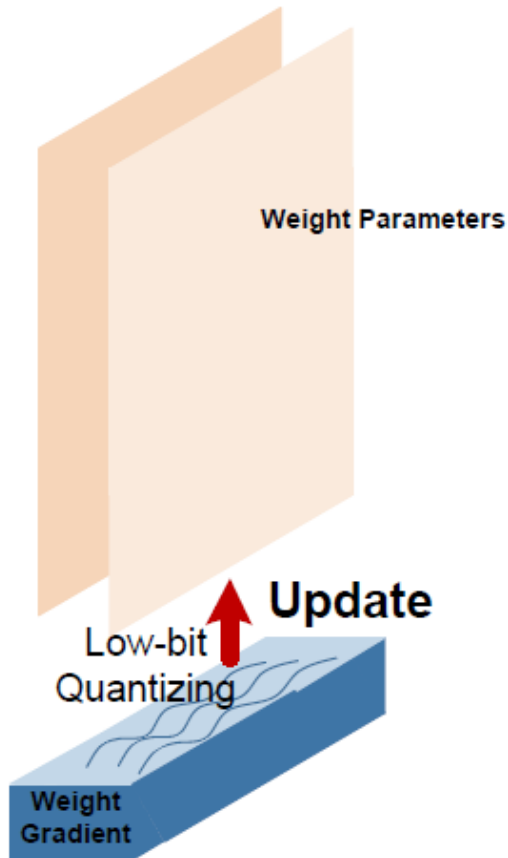
Gradients w.r.t input



Training Process-Backpropagation-2



Training Process-Parameters Update



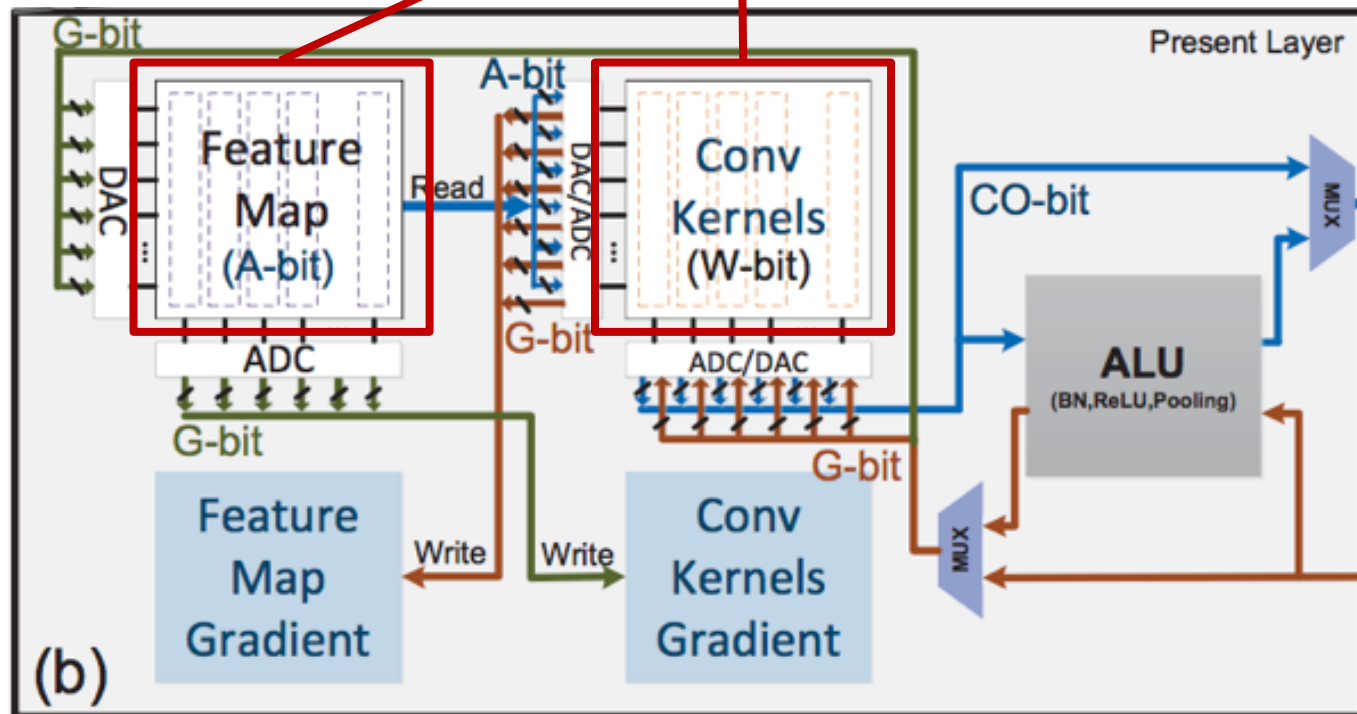
$$\text{GetDeltaW}\left(\frac{\partial J}{\partial K}, \chi, m\right) = \Delta K = \underbrace{\chi}_{\text{Learning rate}} \cdot \frac{\partial J}{\partial K} + \underbrace{m}_{\text{momentum}} \cdot \Delta K_{last}$$

Quantization and AD/DA Conversion

- Quantizing floating-point x to k -bit fixed-point number

$$\text{Quantize}_k(x) = \frac{1}{2^{k-1} - 1} \text{round}((2^{k-1} - 1) \cdot x)$$

Uniform quantization

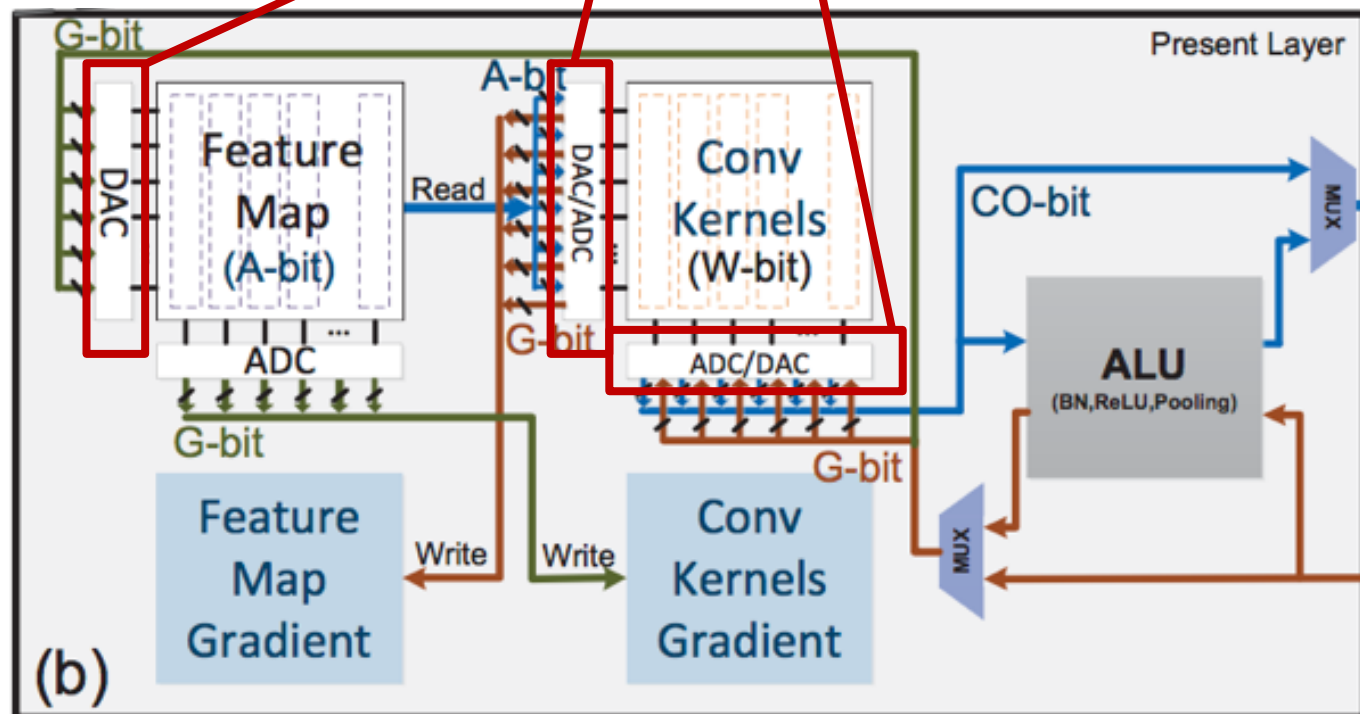


Quantization and AD/DA Conversion

- **Digital-to-Analog conversion strategy in the input interfaces of RRAM Crossbars**

$$DA_convert_k(d) = \alpha_{in} \cdot \sum_{i=0}^{k-1} 2^{-(i+1)} d[i]$$

Determined by experiments

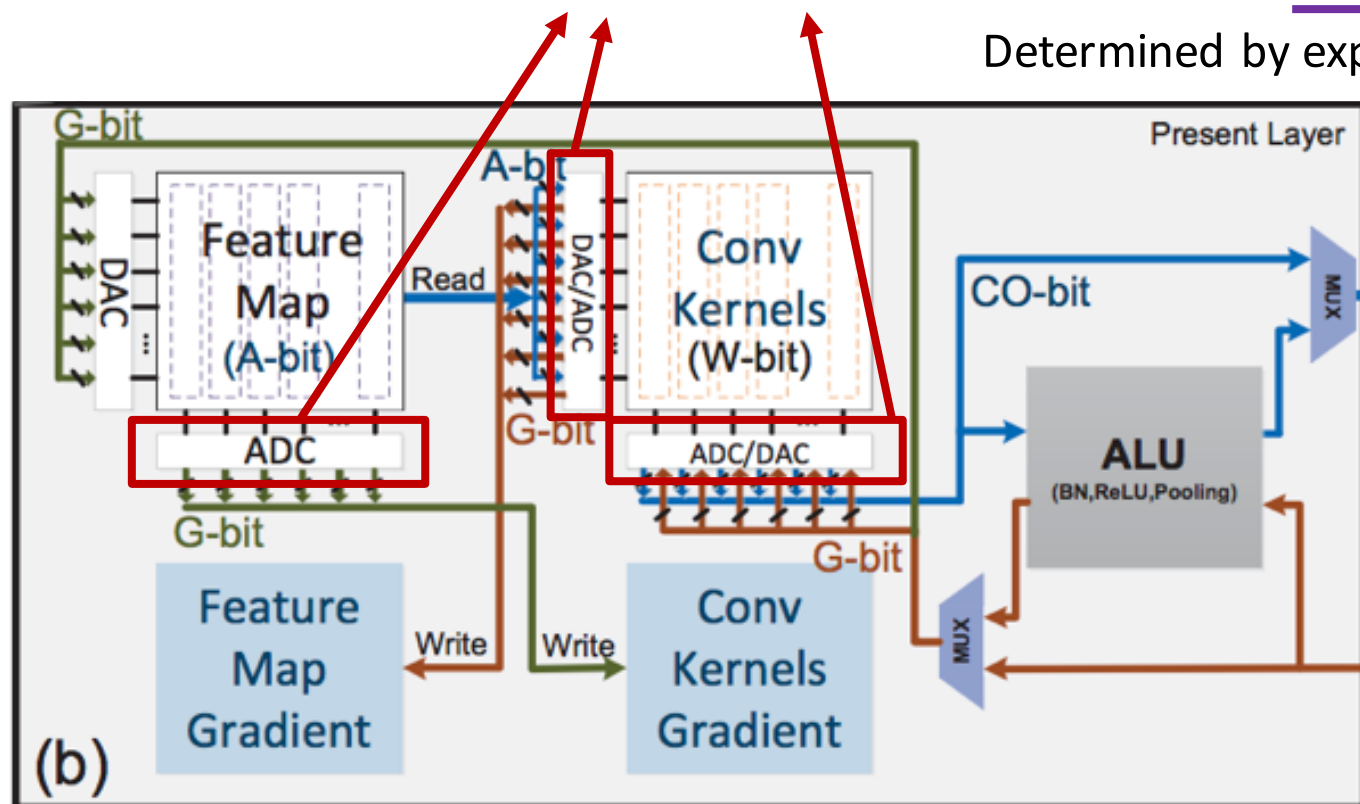


Quantization and AD/DA Conversion

- Analog-to-Digital conversion strategy in the output interfaces of RRAM Crossbars

$$AD_convert_k(V_{out}) = \text{Quantize}_k(\min_1(\max_{-1}(\frac{V_{out}}{\alpha_{out}})))$$

Determined by experiments



Experimental Results

- Benchmarks
 - LeNet on MNIST dataset
 - ResNet-20 on CIFAR-10 dataset
- Evaluation
 - Energy efficiency
 - Accuracy
- Disturbance setup
 - Default: Satisfying $\frac{Weight_{disturbance}}{Weight_{expected}} \sim U[-5\%, 5\%]$
- Comparisons
 - GPU: NVIDIA Titan X (Pascal)
 - CPU: Intel (R) Core(TM) i7-6900K CPU @ 3.20GHz

Experimental Results - Accuracy

TABLE I: The classification accuracy for MNIST test dataset with different combinations of bitwidth in LeNet-5. A, CO, W, G are bitwidth of activations, convouts, weights, and gradients.

A	CO	W	G	Accuracy without disturbance	Accuracy with disturbance	
32 ^a	32	32	32	0.9914	*	
8	8	8	8	0.9828	0.9825	Disturbance- Tolerant
6	6	6	6	0.9745	0.9733	
4	4	4	4	0.9767	0.9797	
3	3	2	4	0.9687	0.9736	
2	2	2	2	0.9670	0.9752	Better Generalization
2	2	1	4	- ^b	-	
2	2	2	1	0.9633	0.9647	
1	1	2	1	0.9416	0.9375	
1	1	1	1	-	-	

^a bitwidth=32 means 32-bit floating-point numbers.

^b '-' means failing to train a convergent model under such bitwidth.

Experimental Results - Accuracy

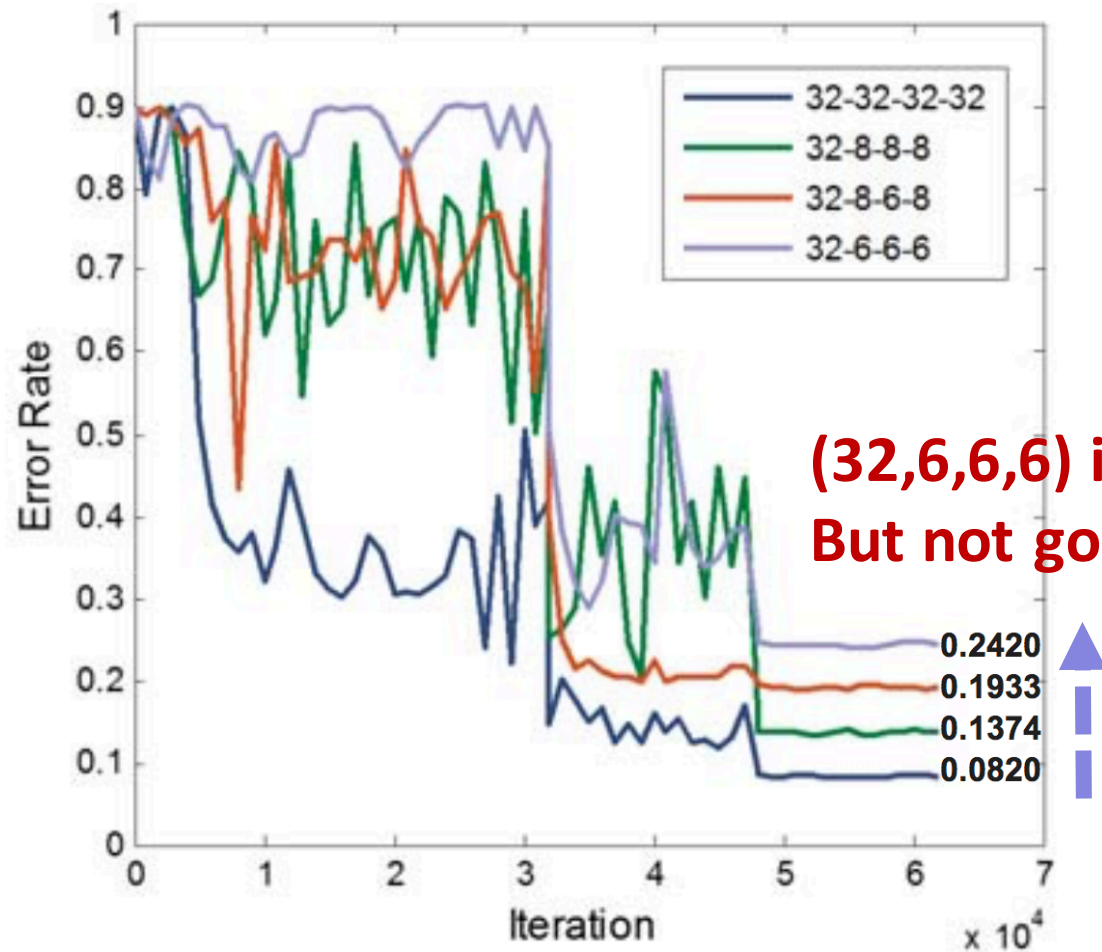
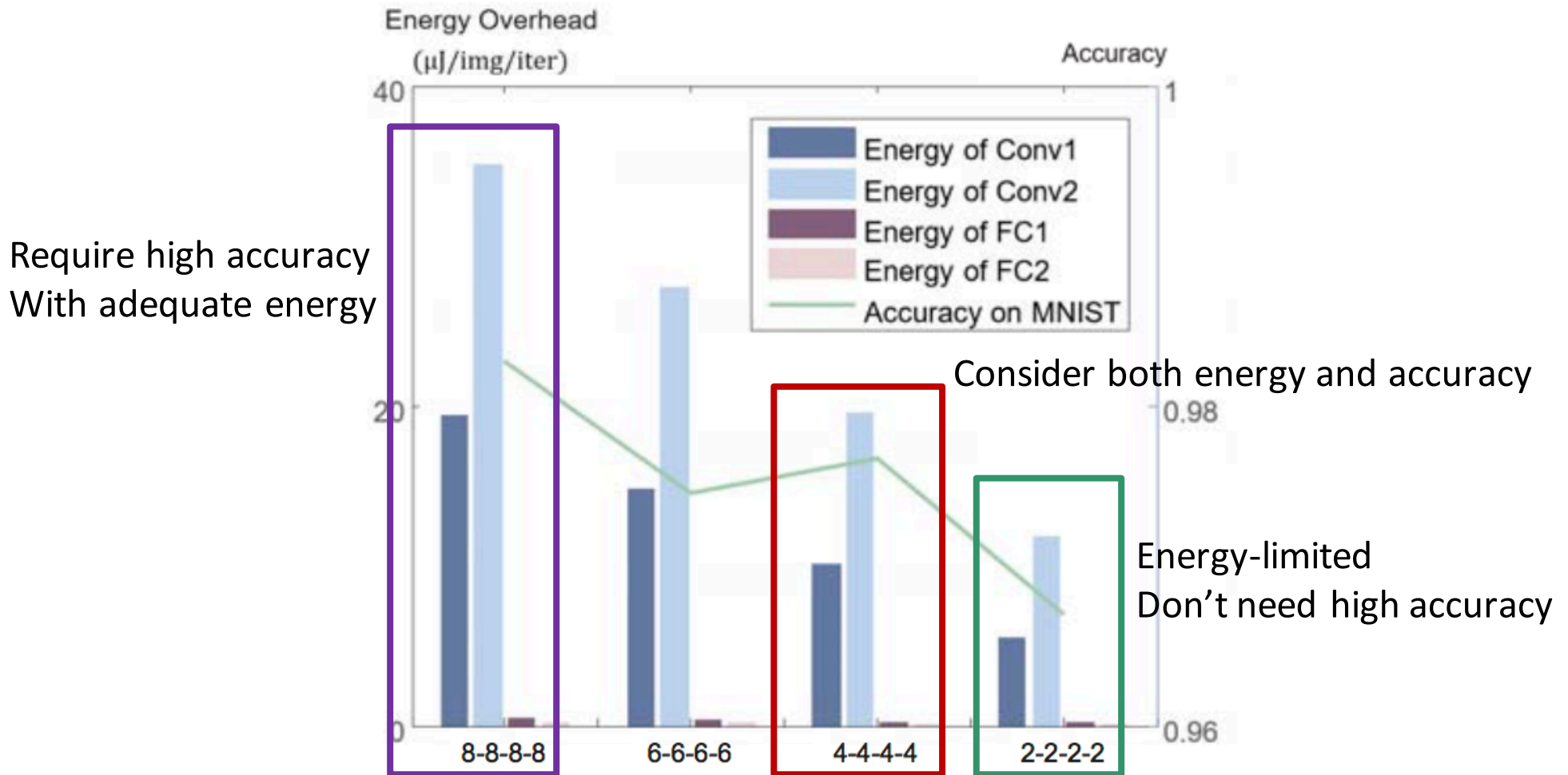


Fig. 3: Error Rate Curves of ResNet-20 on CIFAR-10: Accuracy Under Different Combinations of Bitwidth (A,CO,W,G).

Experimental Results - Tradeoff



Tradeoff: Energy & Accuracy

Experimental Results - Energy

TABLE II: Energy Overhead Estimation of CNNs in Different Training Platforms

Database	Platform	CNN Model	Energy($\mu\text{J}/\text{img}/\text{iter}$)			
			Conv+FC	Others	All	
MNIST	CPU ^a	LeNet-5	Conv+FC	7997.6	88.7%	16.8x
			Others	1015.6	11.3%	
			All	9013.2	100.0%	
	GPU ^b		Conv+FC	11824.9	96.0%	23.0x
			Others	491.3	4.0%	
			All	12316.2	100.0%	
	RRAM		Conv+FC	44.96	8.4%	1.0x
			Others	491.3	91.6%	
			All	536.26	100.0%	
CIFAR-10	CPU	ResNet-20	Conv+FC	262523.4	77.9%	8.9x
			Others	74414.1	22.1%	
			All	336937.5	100.0%	
	GPU		Conv+FC	133066.9	79.4%	4.4x
			Others	34465.2	20.6%	
			All	167532.1	100.0%	
	RRAM		Conv+FC	3653.7	9.6%	1.0x
			Others	34465.2	90.4%	
			All	38118.9	100.0%	

^a Intel(R) Core(TM) i7-6900K CPU @ 3.20GHz.

^b NVIDIA TITAN X (Pascal).

Conclusion

Challenges:

- Can not efficiently support **high-precision data & weights**
- **Disturbance** on RRAM's resistance

Solutions:

- Low-bit CNN training system & algorithm design
- Disturbance analysis and noise-tolerant training

Future Work:

- Improving the training algorithm to get higher accuracy
- Enabling RRAM-based logic computing

Reference

- [1] K. Simonyan et al., “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
- [2] J. Fan et al., “Human tracking using convolutional neural networks.” IEEE Transactions on Neural Networks, vol. 21, no. 10, pp. 1610–1623, 2010.
- [3] G. Hinton et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.
- [4] A. Karpathy et al., “Deep visual-semantic alignments for generating image descriptions,” in Computer Vision and Pattern Recognition, 2015. [5] B. Li et al., “Merging the interface: Power, area and accuracy co-optimization for rram crossbar-based mixed-signal computing system,” in DAC, 2015, p. 13.
- [6] L. Xia et al., “Selected by input: Energy efficient structure for rram-based convolutional neural network,” in DAC, 2016.
- [7] P. Chi et al., “Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory,” in ISCA, vol. 43, 2016.
- [8] A. Shafiee et al., “Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars,” in Proc. ISCA, 2016.
- [9] F. Alibart et al., “High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm,” Nanotechnology, vol. 23, no. 7, p. 075201, 2012.
- [10] R. Degraeve et al., “Causes and consequences of the stochastic aspect of filamentary rram,” Microelectronic Engineering, vol. 147, pp. 171–175, 2015.
- [11] M. Courbariaux et al., “Binarized neural network: Training deep neural networks with weights and activations constrained to +1 or -1,” arXiv preprint arXiv:1602.02830, 2016.

Reference

- [12] M. Rastegari et al., “Xnor-net: Imagenet classification using binary convolutional neural networks,” arXiv preprint arXiv:1603.05279, 2016.
- [13] S. Zhou et al., “Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients,” arXiv preprint arXiv:1606.06160, 2016.
- [14] T. Tang et al., “Binary convolutional neural network on rram,” in Design Automation Conference (ASP-DAC), 2017 22nd Asia and South Pacific. IEEE, 2017, pp. 782–787.
- [15] M. Cheng et al., “Time: A training-in-memory architecture for memristor-based deep neural networks,” in Proceedings of the 54th Annual Design Automation Conference 2017. ACM, 2017, p. 26.
- [16] Z. Jiang et al., “A compact model for metal–oxide resistive random access memory with experiment verification,” IEEE Transactions on Electron Devices, vol. 63, no. 5, pp. 1884–1892, 2016.
- [17] S. Yu et al., “Scaling-up resistive synaptic arrays for neuro-inspired architecture: Challenges and prospect,” in Electron Devices Meeting (IEDM), 2015 IEEE International. IEEE, 2015, pp. 17–3.
- [18] Y. LeCun et al., “Comparison of learning algorithms for handwritten digit recognition,” in International conference on artificial neural networks, vol. 60. Perth, Australia, 1995, pp. 53–60.
- [19] K. He et al., “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
- [20] C. M. Bishop, “Training with noise is equivalent to tikhonov regularization,” Neural computation, vol. 7, no. 1, pp. 108–116, 1995.
- [21] A. F. Murray and P. J. Edwards, “Enhanced mlp performance and fault tolerance resulting from synaptic weight noise during training,” IEEE Transactions on neural networks, vol. 5, no. 5, pp. 792–802, 1994.
- [22] X. Dong et al., “Nvsim: A circuit-level performance, energy, and area model for emerging non-volatile memory,” TCAD, vol. 31, no. 7, pp.994–1007, 2012.

**THANKS FOR
WATCHING**