Exploring Bit-Slice Sparsity in Deep Neural Networks for Efficient ReRAM-Based Deployment

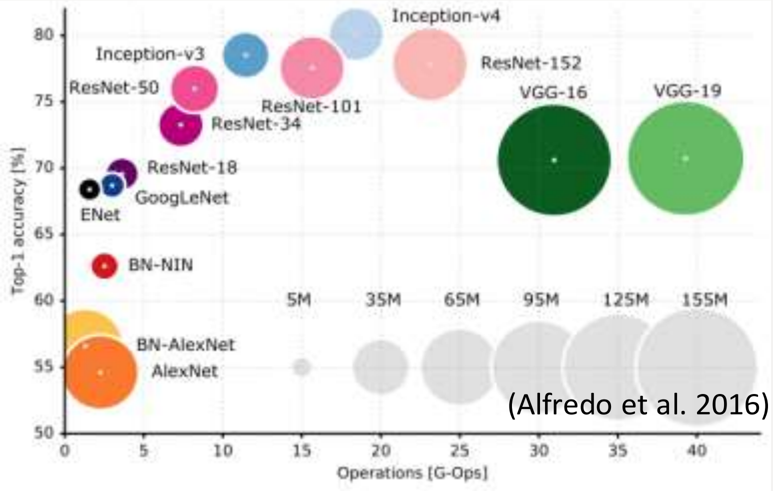
Jingyang Zhang[1], **Huanrui Yang[1]**, Fan Chen[1], Yitu Wang[2], Hai Li[1]

[1]Duke University, [2]Fudan University

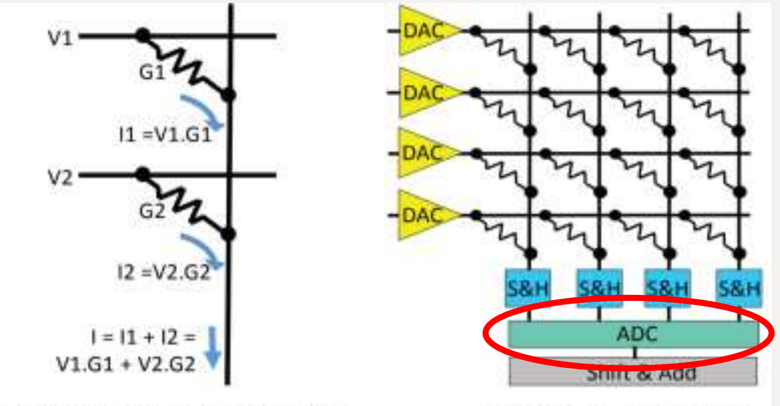
EMC2 Workshop @ NeurIPS 2019

Duke

Motivation: ReRAM-based DNN accelerator



(Alfredo et al. 2016)



(a) Multiply-Accumulate operation

(b) Vector-Matrix Multiplier

Two-order magnitude advantage in energy, performance and chip footprint

- High bit-resolution ADC accounts for >60% power and >30% area
- ADC resolution dictated by accumulated currents on bitlines: need sparsity in G
- Limited cell bit density: each XB only holds 2 bits (bit-slice) of the weight
- Need ***higher sparsity among bit-slice***

$$\begin{bmatrix} 0 & w_1 & 0 \\ 0 & 0 & \boxed{w_2} \\ w_0 & 0 & 0 \end{bmatrix} \Rightarrow [11\ 00\ 10\ 00]_2$$

Weight sparsity   Bit-slice sparsity

Canziani, Alfredo, Adam Paszke, and Eugenio Culurciello. "An analysis of deep neural network models for practical applications." *arXiv preprint arXiv:1605.07678* (2016).

A. Shafiee, A. Nag, N. Muralimanohar, R. Balasubramonian, J. P. Strachan, M. Hu, R. S. Williams, and V. Srikumar. Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars . In Proceedings of ISCA, 2016.

Bit-slice L1 for dynamic fixed-point quantization

- Dynamic range scaling (to [0,1])

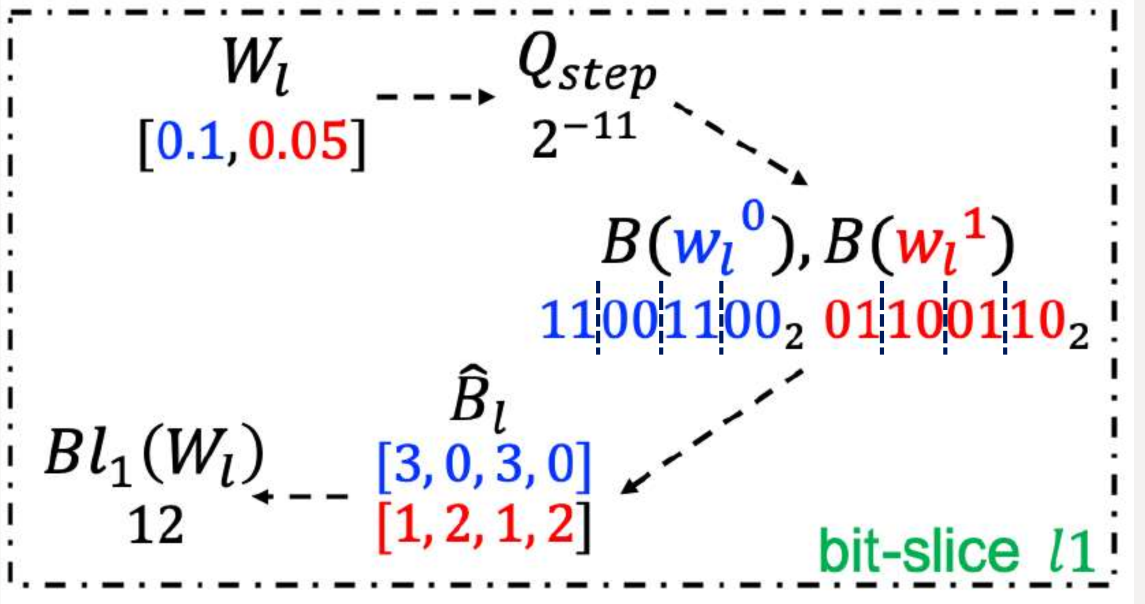
$$S(W_l) = \lceil \log_2 (\max_{w_l^i \in W_l} (|w_l^i|)) \rceil,$$

- N-bit uniform quantization

$$Q_{step} = 2^{S(W_l)-n}, \quad B(w_l^i) = \lfloor \frac{w_l^i}{Q_{step}} \rfloor.$$

- L1 regularization over all bit-slices

$$B(w_l^i) = \sum_{k=0}^{3} \hat{B}_l^{i,k} \cdot 2^{2k} \quad B\ell_1(W_l) := \sum_{i,k} \hat{B}_l^{i,k}.$$



W_l

[0.1, 0.05]

Q_{step}

2^{-11}

$B(w_l{}^0), B(w_l{}^1)$

$11|00|11|00_2$ $01|10|01|10_2$

$\hat{B}_l$

[3, 0, 3, 0]

[1, 2, 1, 2]

$Bl_1(W_l)$

12

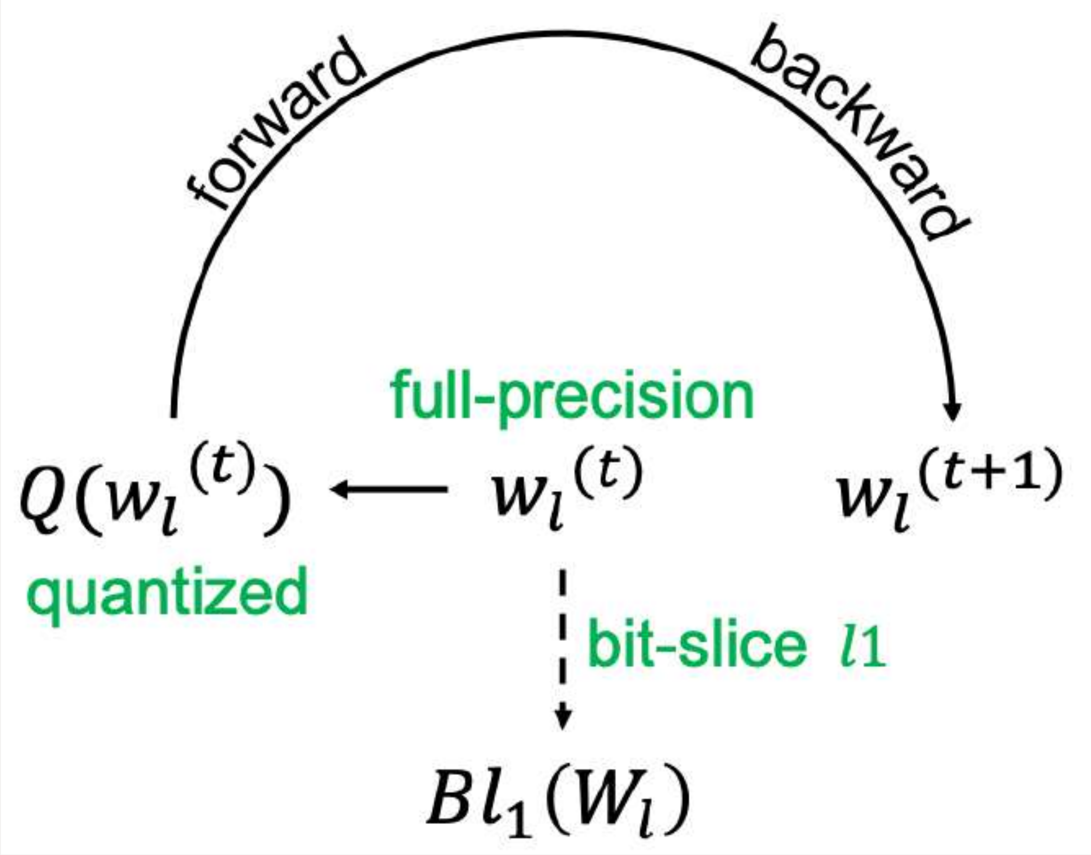
bit-slice $l1$

Training routine

- Dynamic range recovery

$$Q(w_l^i) = B(w_l^i) \cdot Q_{step}$$

- Training routine
  - FP and BP with quantized weight
  - Gradient update on full-precision weight
  - Add Bit-slice L1 to the objective



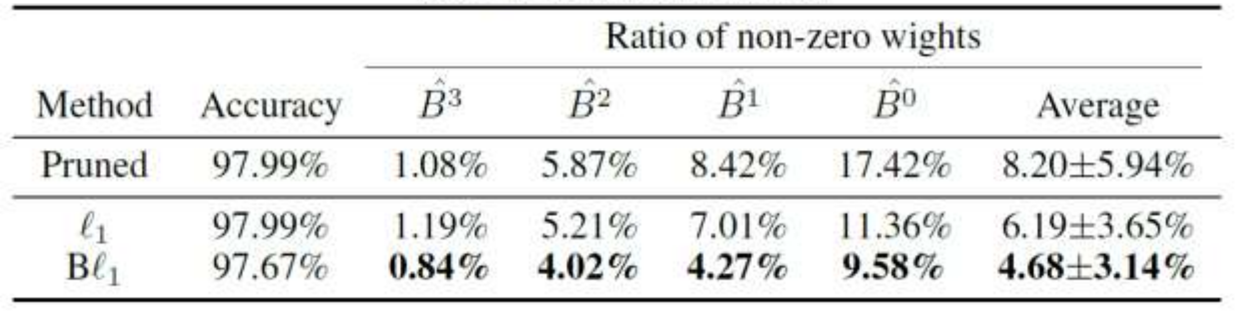
$$q^{(t)} = Q(w_l^{(t)}),$$

$$w_l^{(t+1)} = q^{(t)} - lr \times (\nabla_q \mathcal{L}_{CE}(q^{(t)}) + \alpha \nabla_q B\ell_1(q^{(t)}))$$

Improving the bit-slice sparsity

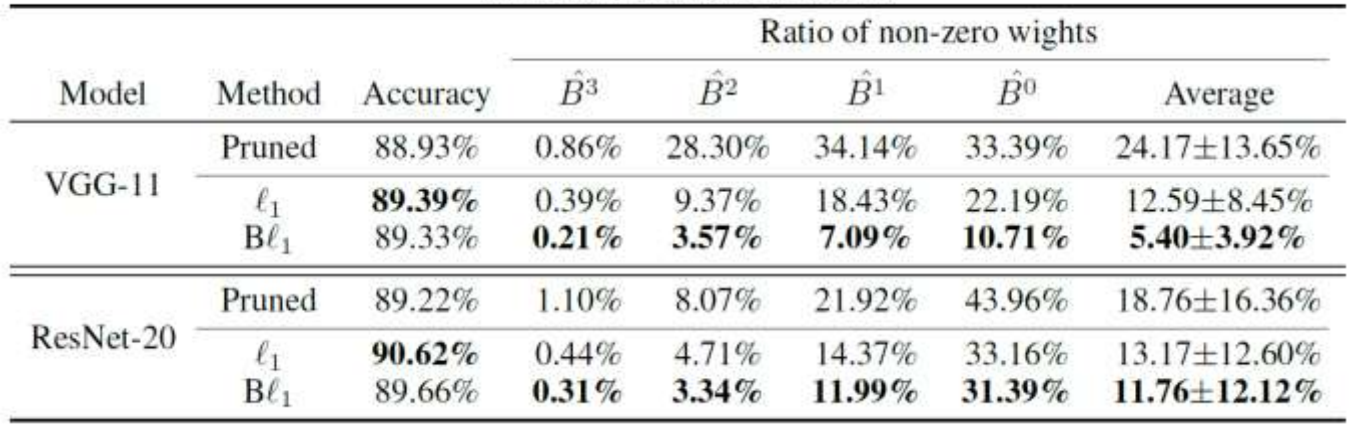
- Up to 2x less nonzero bit-slices than traditional L1

Table 1: Results on MNIST



		Ratio of non-zero wights				
Method	Accuracy	$\hat{B}^3$	$\hat{B}^2$	$\hat{B}^1$	$\hat{B}^0$	Average
Pruned	97.99%	1.08%	5.87%	8.42%	17.42%	8.20±5.94%
ℓ_1	97.99%	1.19%	5.21%	7.01%	11.36%	6.19±3.65%
Bℓ_1	97.67%	**0.84%**	**4.02%**	**4.27%**	**9.58%**	**4.68±3.14%**

Table 2: Results on CIFAR-10



			Ratio of non-zero wights				
Model	Method	Accuracy	$\hat{B}^3$	$\hat{B}^2$	$\hat{B}^1$	$\hat{B}^0$	Average
VGG-11	Pruned	88.93%	0.86%	28.30%	34.14%	33.39%	24.17±13.65%
	ℓ_1	**89.39%**	0.39%	9.37%	18.43%	22.19%	12.59±8.45%
	Bℓ_1	89.33%	**0.21%**	**3.57%**	**7.09%**	**10.71%**	**5.40±3.92%**
ResNet-20	Pruned	89.22%	1.10%	8.07%	21.92%	43.96%	18.76±16.36%
	ℓ_1	**90.62%**	0.44%	4.71%	14.37%	33.16%	13.17±12.60%
	Bℓ_1	89.66%	**0.31%**	**3.34%**	**11.99%**	**31.39%**	**11.76±12.12%**

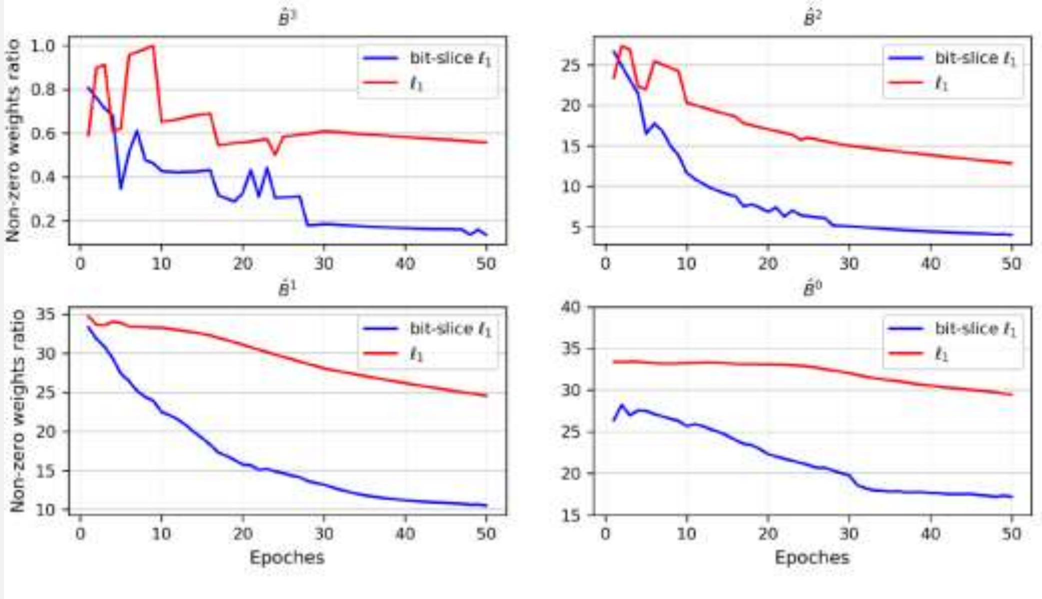


Figure 2: Bit-slice sparsity of VGG-11 on CIFAR-10 during training.

- Codes available at: https://github.com/zjysteven/bitslice_sparsity

Duke

Reducing ADC overhead

- High sparsity in bit-slices enables the use of low-resolution ADC
- Low resolution reduces ADC overhead

- Simulation results for mapping to 128x128 ReRAM XBs

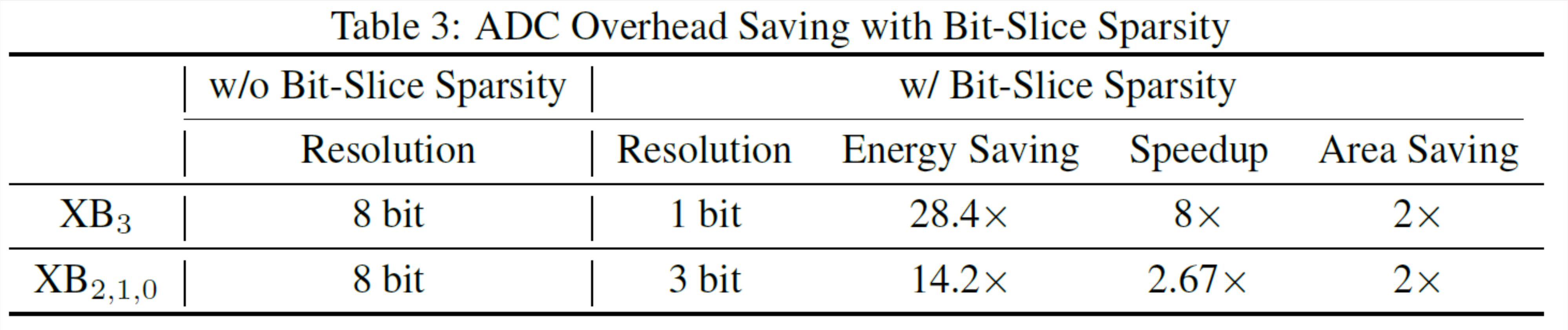


Table 3: ADC Overhead Saving with Bit-Slice Sparsity

	w/o Bit-Slice Sparsity	w/ Bit-Slice Sparsity			
	Resolution	Resolution	Energy Saving	Speedup	Area Saving
XB_3	8 bit	1 bit	28.4×	8×	2×
$XB_{2,1,0}$	8 bit	3 bit	14.2×	2.67×	2×