Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks

Yikai Wang, Yi Yang, Fuchun Sun, Anbang Yao

Tsinghua University, Intel Corporation

2021 ICCV OCTOBER 11-17 VIRTUAL

Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks

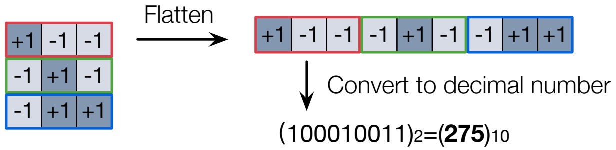
- Sub-bit Neural Networks (SNNs): The first method that simultaneously **compresses** and **accelerates** BNNs in a quantization pipeline with moderate accuracy drops.

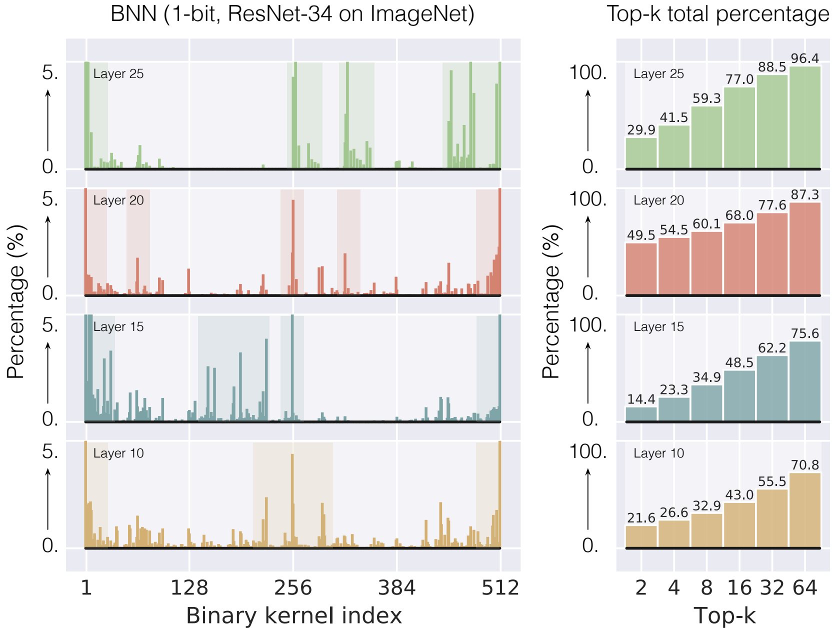
$$\bar{\mathbf{w}}^i = \mathrm{sign}(\mathbf{w}^i) = \underset{\mathbf{b}\in\{\pm1\}^{n^i}}{\arg\min} \|\mathbf{b} - \mathbf{w}^i\|_2^2.$$

$$\bar{\mathbf{w}}_c^i = \arg\min_{\mathbf{k}\in\mathbb{K}} \|\mathbf{k} - \mathbf{w}_c^i\|_2^2$$

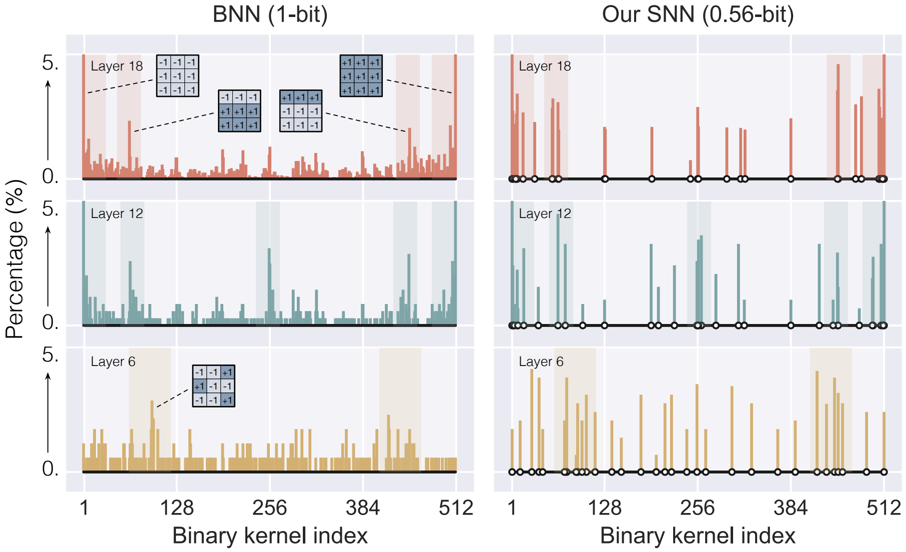
$$\mathbb{K} = \{\pm1\}^{w^i \cdot h^i}$$

$$|\mathbb{K}| = 2^9 = 512$$





Distributions of binary kernels for a standard BNN, where binary kernels are sparsely distributed.

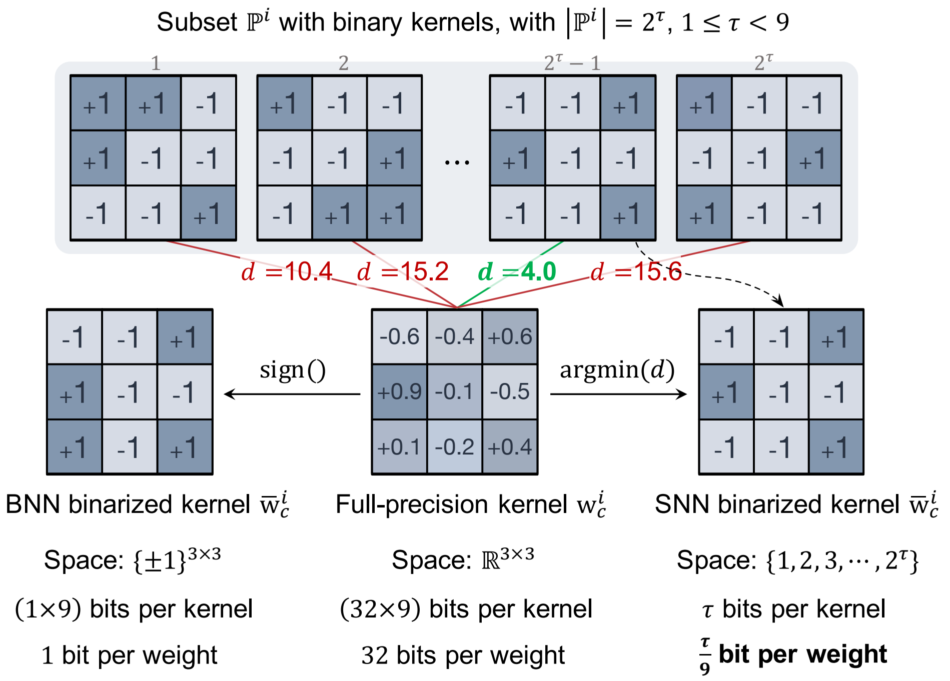


Frequencies of different binary kernels of a standard 1-bit BNN and our 0.56-bit SNN.

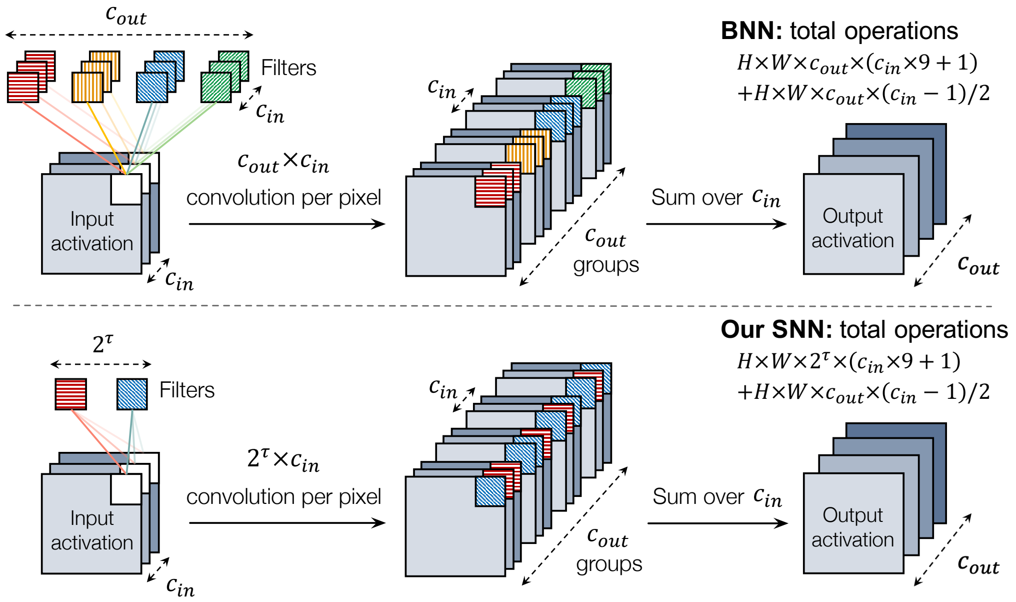
Yikai Wang, Yi Yang, Fuchun Sun, Anbang Yao. Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks. **ICCV 2021**.

Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks

- **Compression**: SNN leads to a compression ratio $\tau/9$
- **Acceleration** (with practical hardware design):
  - Bit-OPs of a BNN: $2 \cdot H \cdot W \cdot c_{out}^i \cdot (c_{in}^i \cdot w^i \cdot h^i + 1)$
  - SNN reduces this number with ratio $2^\tau / c_{out}^i$



Binarization comparison of a standard BNN model and SNN.



Comparison of convolution processes in a standard BNN and SNN.

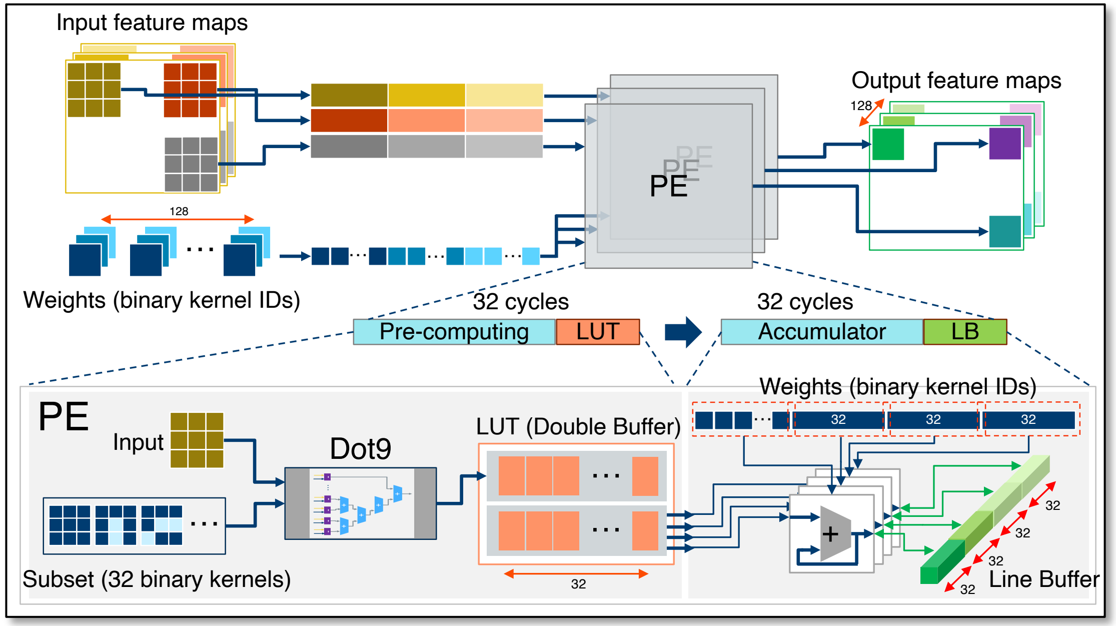
Theoretically, each kernel in the subset is precomputed per channel and per pixel of the input activation, and there are $2^\tau \times c_{in}^i \times W^i \times H^i$ precomputed results.

However, by well designing the computation flow, we can reduce the LUT size to 2^τ and thus decrease the lookup time costs (next page).

Yikai Wang, Yi Yang, Fuchun Sun, Anbang Yao. Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks. **ICCV 2021**.

Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks

- Acceleration (with **practical hardware design**):
  - Bit-OPs of a BNN: $2 \cdot H \cdot W \cdot c^i_{out} \cdot (c^i_{in} \cdot w^i \cdot h^i + 1)$
  - SNN reduces this number with ratio $2^\tau / c^i_{out}$

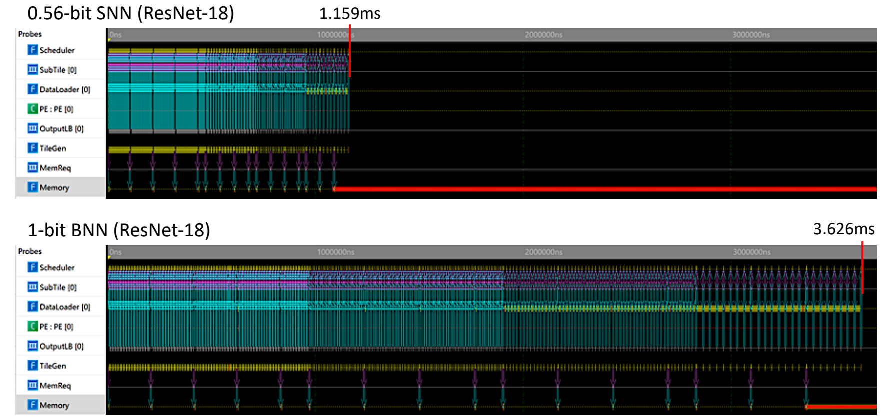


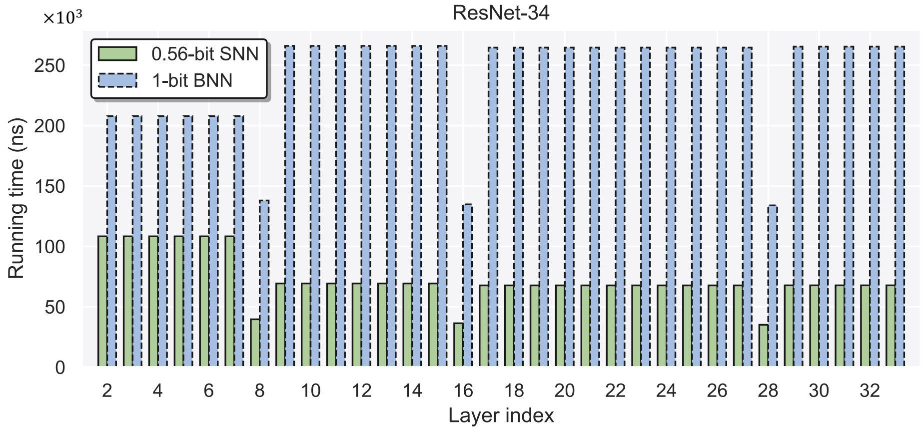
A hardware design case for the deployment of our 0.56-bit SNN, with 64 PEs and 4 parallel accumulators. Pre-computing and accumulating are performed simultaneously with the same cycles in a pipeline.

Backbone	Running time (ms)		Speed up
	1-bit BNN	0.56-bit SNN	
ResNet-18	3.626	**1.159**	3.13×
ResNet-34	7.753	**2.329**	3.33×

Speed tests of the practical deployment for BNNs and SNNs.

224×224 input on the hardware configuration of 64PEs@1GHz.





Yikai Wang, Yi Yang, Fuchun Sun, Anbang Yao. Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks. **ICCV 2021**.

ICCV 2021 OCTOBER 11-17 VIRTUAL

Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks

➢ Optimization method:

- Random Kernel Subsets Sampling

$$\text{Forward}: \bar{\mathbf{w}}_c^i = \arg\min_{\mathbf{k}\in\mathbb{P}^i} \|\mathbf{k} - \mathbf{w}_c^i\|_2^2,$$

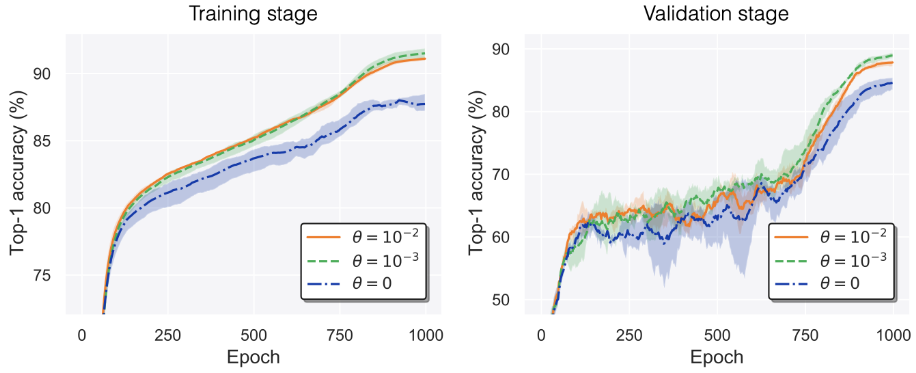
$$\text{Backward}: \frac{\partial\mathcal{L}}{\partial\mathbf{w}_c^i} \approx \begin{cases} \frac{\partial\mathcal{L}}{\partial\bar{\mathbf{w}}_c^i}, & \text{if } \mathbf{w}_c^i \in (-1,1)\,, \\ 0, & \text{otherwise.} \end{cases}$$

- Kernel Subsets Refinement by Optimization

$$\text{Forward}: \bar{\mathbf{w}}_c^i = \arg\min_{\mathbf{m}_j^i} \|\mathbf{m}_j^i - \mathbf{w}_c^i\|_2^2,$$

$$\mathbf{m}^i = \mathbf{m}^i \odot \mathbb{I}_{|\mathbf{p}^i|\le\theta} + \text{sign}(\mathbf{p}^i) \odot \mathbb{I}_{|\mathbf{p}^i|>\theta},$$

$$\text{Backward}: \text{Eq. (3)}, \quad \frac{\partial\mathcal{L}}{\partial\mathbf{p}^i} \approx \frac{\partial\mathcal{L}}{\partial\mathbf{m}^i},$$



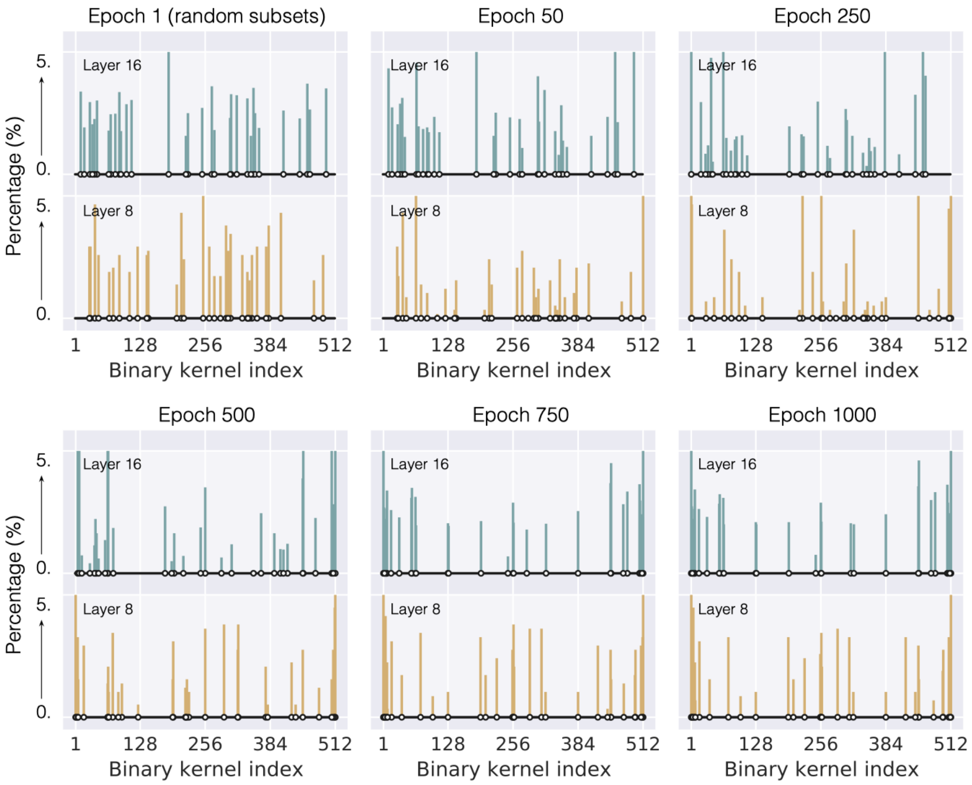
Algorithm 1 Training: forward and backward processes of Sub-bit Neural Networks (SNNs).

Require: input data; full-precision weights $\mathbf{w}$; threshold θ; learning rate η.
for layer $i = 1 \to L$ **do**
  Randomly sample a layer-specific subset $\mathbb{P}^i \subset \mathbb{K}$ and there is $|\mathbb{P}^i| = 2^\tau$; Represent $\mathbb{P}^i$ as $\mathbf{p}^i \in \mathbb{R}^{w^i \cdot h^i \cdot |\mathbb{P}^i|}$.
  Initialize $\mathbf{m}^i = \text{sign}(\mathbf{p}^i)$.
for step $t = 1 \to T$ **do**
  Forward propagation:
    for layer $i = 1 \to L$ **do**
      Compute $\mathbf{m}^i = \mathbf{m}^i \odot \mathbb{I}_{|\mathbf{p}^i|\le\theta} + \text{sign}(\mathbf{p}^i) \odot \mathbb{I}_{|\mathbf{p}^i|>\theta}$.
      for channel $c = 1 \to c_{out}^i \cdot c_{in}^i$ **do**
        Compute $\bar{\mathbf{w}}_c^i = \arg\min_{\mathbf{m}_j^i} \|\mathbf{m}_j^i - \mathbf{w}_c^i\|_2^2$.
        Compute $\mathbf{a}_c^i = \lambda_c^i \cdot (\bar{\mathbf{w}}_c^i \circ \text{sign}(\mathbf{a}_c^{i-1}))$ in Sec. 3.
  Back propagation:
    for layer $i = L \to 1$ **do**
      for channel $c = 1 \to c_{out}^i \cdot c_{in}^i$ **do**
        Compute $\frac{\partial\mathcal{L}}{\partial\mathbf{w}_c^i}$ via Eq. (3).
      Compute $\frac{\partial\mathcal{L}}{\partial\mathbf{p}^i} \approx \frac{\partial\mathcal{L}}{\partial\mathbf{m}^i}$.
  Parameters Update:
    Update $\mathbf{w} = \mathbf{w} - \eta\frac{\partial\mathcal{L}}{\partial\mathbf{w}}$, $\mathbf{p} = \mathbf{p} - \eta\frac{\partial\mathcal{L}}{\partial\mathbf{p}}$.
  Check repetitive binary kernels in $\mathbf{m}^i$ and substitute these corresponding kernels in $\mathbf{p}^i$ with random new kernels.

Yikai Wang, Yi Yang, Fuchun Sun, Anbang Yao. Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks. **ICCV 2021**.

Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks

➢ Experiments



Visualization of how binary kernel subsets change during the training process of a 0.56-bit SNN.

Method	Bit-width (W/A)	#Params (Mbit)	Bit-OPs (G)	Top-1 Acc. (%)
ResNet-18				
Full precision	32/32	351.54	35.03	93.0
RAD [5]	1/1	10.99	0.547	90.5
IR-Net [23]	1/1	10.99	0.547	91.5
Vanilla-SNN | SNN	0.67/1	7.324 (1.5×)	0.289 (1.9×)	89.7 | 91.0
Vanilla-SNN | SNN	0.56/1	6.103 (1.8×)	0.164 (3.3×)	89.3 | 90.6
Vanilla-SNN | SNN	0.44/1	4.882 (2.3×)	0.097 (5.6×)	88.3 | 90.1
IR-Net* [23]	1/32	10.99	17.52	92.9
Vanilla-SNN | SNN	0.67/32	7.324 (1.5×)	9.236 (1.9×)	92.4 | 92.7
Vanilla-SNN | SNN	0.56/32	6.103 (1.8×)	5.239 (3.3×)	92.0 | 92.3
Vanilla-SNN | SNN	0.44/32	4.882 (2.3×)	3.106 (5.6×)	91.6 | 91.9
ResNet-20				
Full precision	32/32	8.54	2.567	91.7
DoReFa [33]	1/1	0.267	0.040	79.3
IR-Net [23]	1/1	0.267	0.040	86.5
Vanilla-SNN | SNN	0.67/1	0.178 (1.5×)	0.040	83.9 | 85.1
Vanilla-SNN | SNN	0.56/1	0.148 (1.8×)	0.034 (1.2×)	82.7 | 84.0
Vanilla-SNN | SNN	0.44/1	0.119 (2.3×)	0.025 (1.6×)	82.0 | 82.5
DoReFa [33]	1/32	0.267	1.283	90.0
LQ-Net [30]	1/32	0.267	1.283	90.1
IR-Net [23]	1/32	0.267	1.283	90.8
Vanilla-SNN | SNN	0.67/32	0.178 (1.5×)	1.283	88.7 | 90.0
Vanilla-SNN | SNN	0.56/32	0.148 (1.8×)	1.099 (1.2×)	87.8 | 88.9
Vanilla-SNN | SNN	0.44/32	0.119 (2.3×)	0.822 (1.6×)	87.1 | 87.6
VGG-small				
Full precision	32/32	146.24	38.66	92.5
LAB [9]	1/1	4.571	0.603	87.7
XNOR [24]	1/1	4.571	0.603	89.8
BNN [12]	1/1	4.571	0.603	89.9
RAD [5]	1/1	4.571	0.603	90.0
IR-Net [23]	1/1	4.571	0.603	90.4
IR-Net* [23]	1/1	4.571	0.603	91.3
Vanilla-SNN | SNN	0.67/1	3.047 (1.5×)	0.194 (3.1×)	90.3 | 91.0
Vanilla-SNN | SNN	0.56/1	2.540 (1.8×)	0.113 (5.3×)	89.8 | 90.6
Vanilla-SNN | SNN	0.44/1	2.032 (2.3×)	0.074 (8.1×)	89.2 | 90.0
IR-Net* [23]	1/32	4.571	19.30	92.5
Vanilla-SNN | SNN	0.67/32	3.047 (1.5×)	6.208 (3.1×)	92.0 | 92.4
Vanilla-SNN | SNN	0.56/32	2.540 (1.8×)	3.616 (5.3×)	91.7 | 92.1
Vanilla-SNN | SNN	0.44/32	2.032 (2.3×)	2.368 (8.1×)	91.3 | 91.9

Results on the CIFAR10 dataset.

Method	Bit-width (W/A)	#Params (Mbit)	Bit-OPs (G)	Top-1 Acc. (%)
ResNet-18				
Full precision	32/32	351.54	107.28	69.6
XNOR [24]	1/1	10.99	1.677	51.2
BNN+ [12]	1/1	10.99	1.677	53.0
Bi-Real [21]	1/1	10.99	1.677	56.4
XNOR++ [2]	1/1	10.99	1.677	57.1
IR-Net [23]	1/1	10.99	1.677	58.1
Vanilla-SNN | SNN	0.67/1	7.324 (1.5×)	0.883 (1.9×)	55.7 | 56.3
Vanilla-SNN | SNN	0.56/1	6.103 (1.8×)	0.501 (3.3×)	54.6 | 55.1
Vanilla-SNN | SNN	0.44/1	4.882 (2.3×)	0.297 (5.6×)	52.5 | 53.0
BWN [24]	1/32	10.99	53.64	60.8
HWGQ [17]	1/32	10.99	53.64	61.3
BWHN [11]	1/32	10.99	53.64	64.3
IR-Net [23]	1/32	10.99	53.64	66.5
FleXOR [15]	0.80/32	8.788 (1.3×)	53.64	63.8
FleXOR [15]	0.60/32	6.591 (1.7×)	53.64	62.0
Vanilla-SNN | SNN	0.67/32	7.324 (1.5×)	28.26 (1.9×)	63.7 | 64.7
Vanilla-SNN | SNN	0.56/32	6.103 (1.8×)	16.03 (3.3×)	62.8 | 63.4
Vanilla-SNN | SNN	0.44/32	4.882 (2.3×)	9.504 (5.6×)	60.1 | 60.9
ResNet-34				
Full precision	32/32	674.88	225.66	73.3
Bi-Real [21]	1/1	21.09	3.526	62.2
IR-Net [23]	1/1	21.09	3.526	62.9
Vanilla-SNN | SNN	0.67/1	14.06 (1.5×)	1.696 (2.1×)	60.6 | 61.4
Vanilla-SNN | SNN	0.56/1	11.71 (1.8×)	0.965 (3.7×)	59.5 | 60.2
Vanilla-SNN | SNN	0.44/1	9.372 (2.3×)	0.581 (6.1×)	58.1 | 58.6
IR-Net [23]	1/32	21.09	112.83	70.4
Vanilla-SNN | SNN	0.67/32	14.06 (1.5×)	54.27 (2.1×)	67.5 | 68.0
Vanilla-SNN | SNN	0.56/32	11.71 (1.8×)	30.88 (3.7×)	66.3 | 66.9
Vanilla-SNN | SNN	0.44/32	9.372 (2.3×)	18.59 (6.1×)	64.5 | 65.1

Results on the ImageNet dataset.

Yikai Wang, Yi Yang, Fuchun Sun, Anbang Yao. Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks. **ICCV 2021**.

Thanks for your listening!

Yikai Wang, Yi Yang, Fuchun Sun, Anbang Yao. Sub-bit Neural Networks: Learning to Compress and Accelerate Binary Neural Networks. **ICCV 2021**.