二值神经网络那些事

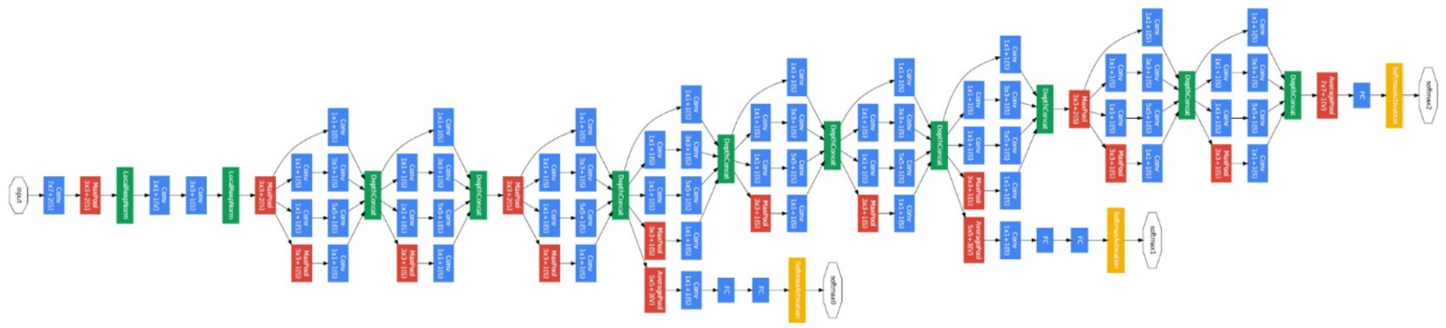
秦浩桐
北航高等理工学院 软件开发环境国家重点实验室 博一

2020-04-14

Contents

Contents

Background



Complicated Models

How to get better performance ？

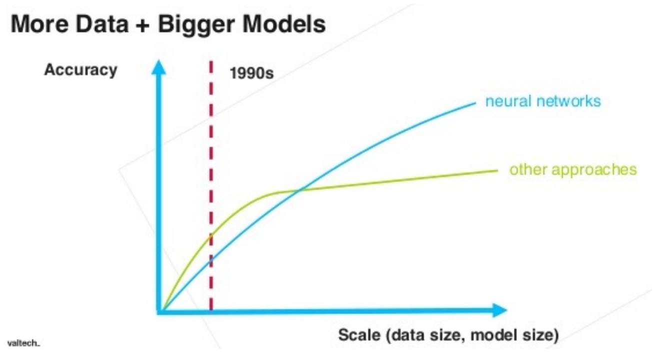
Big Data

HPC

Background

Challenges

- Limited computing resources
- Short response time
- Millions of parameters
- Complicated model architecture

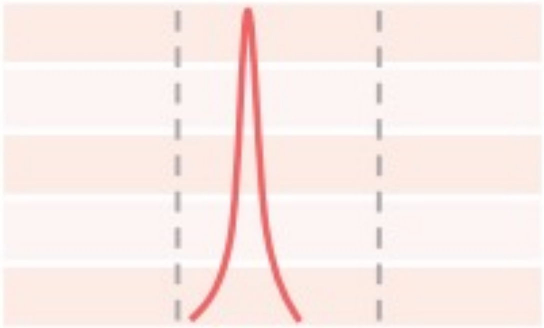


Model	Architecture	Parameters	Top-1 ERR	Top-5 ERR
AlexNet	8 Layers (5conv + 3fc)	~ 60 million	40.7%	15.3%
VGG	19 Layers (16conv + 3fc)	~ 144 million	24.4%	7.1%
GoogLeNet	22 Layers	~ 6.8 million	-	7.9%
MSRA	22 Layers (19conv + 3fc)	~ 200 million	21.29%	5.71%

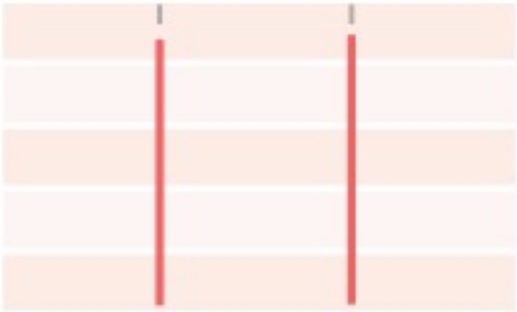
Thus, result in state-of-the-art models hard to be deployed

Lower Precision

$$\underset{\text{32-bit}}{\mathbb{R}} \longrightarrow \underset{\text{8-bit}}{\mathbb{I}} \longrightarrow \underset{\text{1-bit}}{\mathbb{B}} \in \{-1,+1\}$$



{-1,+1}	{0,1}
MUL	XNOR
ADD, SUB	Bit-Count (popcount)



Why binary?

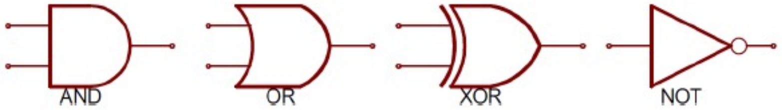
Extremely Low Memory Usage

32× memory savings

$$\underset{\text{32-bit}}{\mathbb{R}} \longrightarrow \underset{\text{8-bit}}{\mathbb{I}} \longrightarrow \underset{\text{1-bit}}{\mathbb{B}} \in \{-1, +1\}$$

Efficient Binary Instructions

58× faster convolutional operations



Low Power Devices

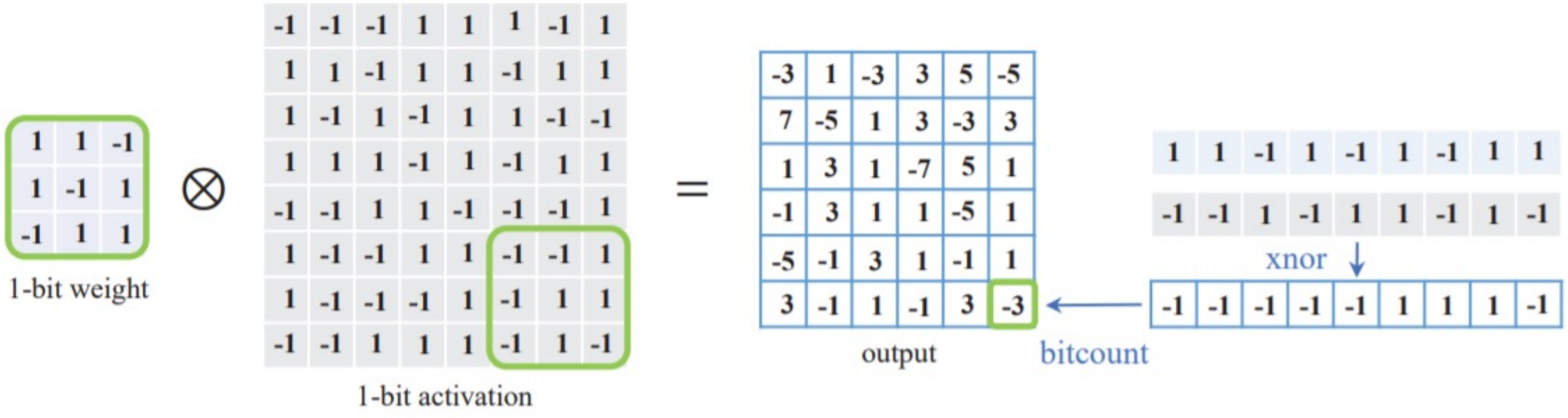


Formulation

$$Q_w(\mathbf{w}) = \alpha \mathbf{b_w}, \quad Q_a(\mathbf{a}) = \beta \mathbf{b_a}$$

$$\mathtt{sign}(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

$$\mathbf{z} = \sigma(Q_w(\mathbf{w}) \otimes Q_a(\mathbf{a})) = \sigma(\alpha\beta(\mathbf{b_w} \odot \mathbf{b_a}))$$



Binary Neural Networks

Full-Precision Neural Networks

- Massive Parameters
- Complex Computation
- High Memory Usage
- High Performance

Binarized Neural Networks

- Binarized Parameters
- Speedup
- Low Memory Usage
- **Significant Drop of Performance**

Contents

Design Accurate BNNs

How to design accurate binary neural networks?

Naive BNNs	Optimization Based BNNs			Tricks
	Minimize the Quantization Error	Improve Network Loss Function	Reduce the Gradient Error	

Binary Neural Networks: A Survey

Pattern Recognition

ArXiv: https://arxiv.org/abs/2004.03333

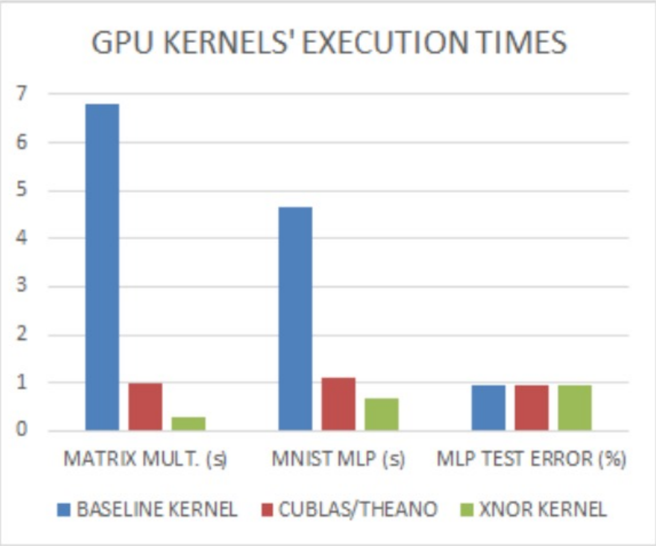
News: https://mp.weixin.qq.com/s/QGva6fow9tad_daZ_G2p0Q

Binarized Neural Networks

Forward:

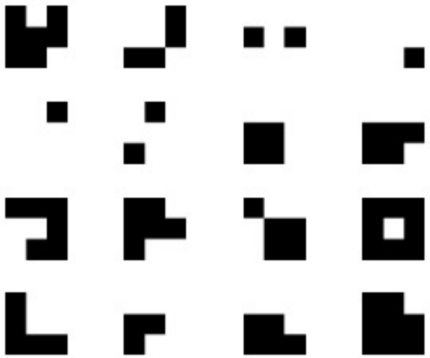
$$\texttt{sign}(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

$$w_b = \begin{cases} +1, & \text{with probability } p = \hat{\sigma}(w) \\ -1, & \text{with probability } 1 - p \end{cases}$$



Backward (STE):

$$\texttt{clip}(x, -1, 1) = \max(-1, \min(1, x)).$$



CIFAR-10

BNN[11]		AlexNet[1]	
Top-1	Top-5	Top-1	Top-5
27.9	50.42	56.6	80.2

Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or−1

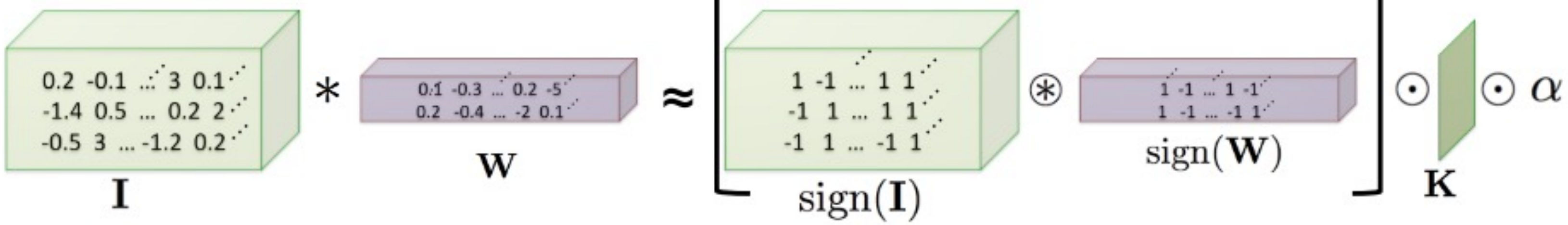
Minimize the Quantization Error

XNOR-Net

Solving the following optimization:

$$J(\mathbf{B}, \alpha) = \|\mathbf{W} - \alpha\mathbf{B}\|^2$$
$$\alpha^*, \mathbf{B}^* = \underset{\alpha, \mathbf{B}}{\operatorname{argmin}} J(\mathbf{B}, \alpha)$$

(4) Convolution with XNOR-Bitcount



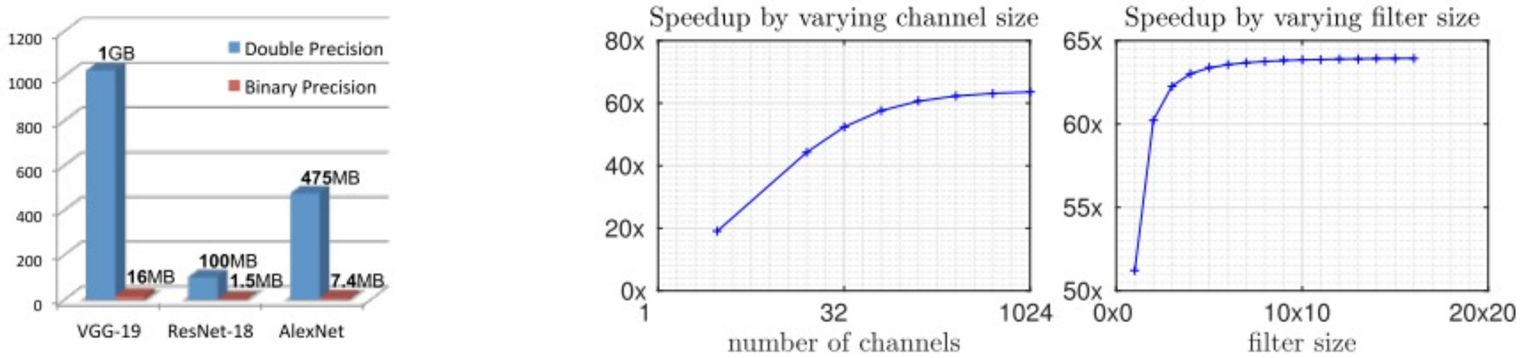
$$\alpha^*, \mathbf{B}^*, \beta^*, \mathbf{H}* = \underset{\alpha, \mathbf{B}, \beta, \mathbf{H}}{\operatorname{argmin}} \|\mathbf{X} \odot \mathbf{W} - \beta\alpha\mathbf{H} \odot \mathbf{B}\|$$

$$\alpha^* = \frac{\mathbf{W}^\mathsf{T} \operatorname{sign}(\mathbf{W})}{n} = \frac{\sum |\mathbf{W}_i|}{n} = \frac{1}{n}\|\mathbf{W}\|_{\ell 1}$$

XNOR-net: Imagenet classification using binary convolutional neural networks

Minimize the Quantization Error

XNOR-Net

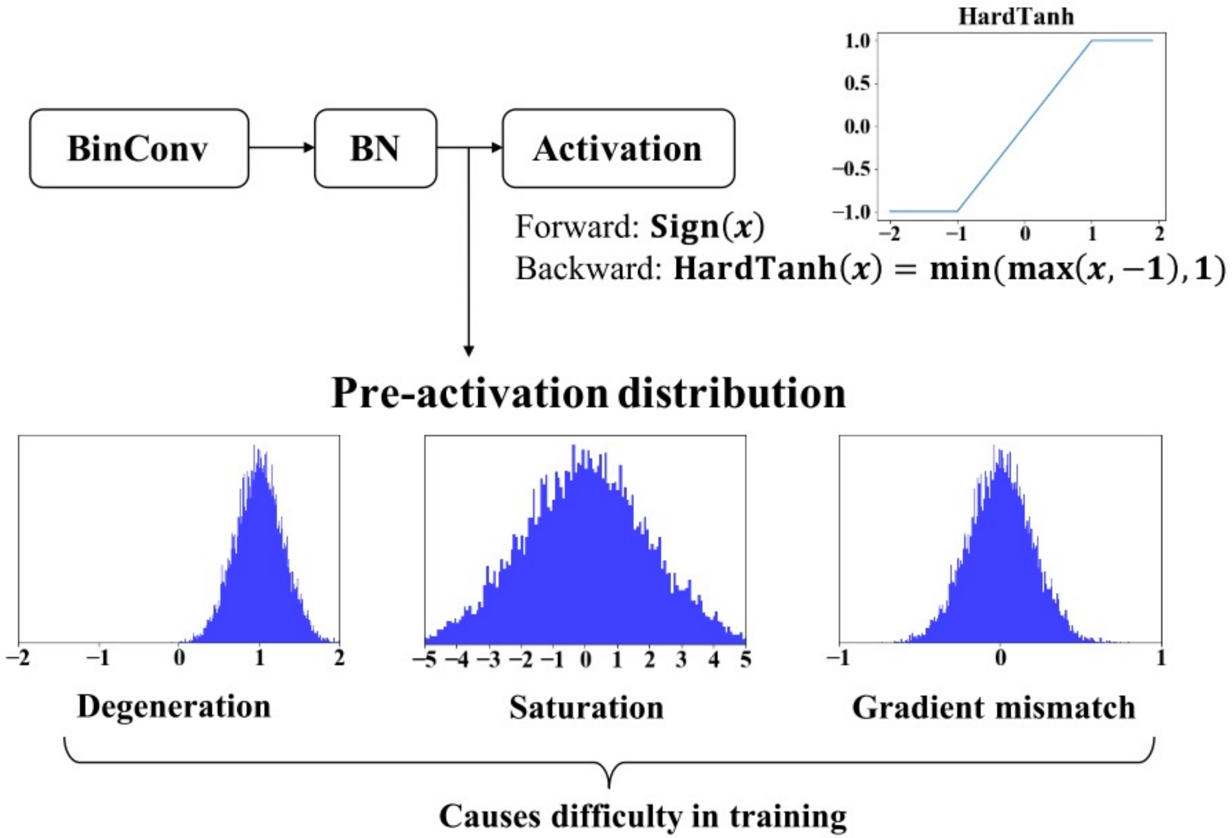


(a) (b) (c)

Classification Accuracy(%)									
Binary-Weight				Binary-Input-Binary-Weight				Full-Precision	
BWN		BC[11]		XNOR-Net		BNN[11]		AlexNet[1]	
Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
56.8	**79.4**	35.4	61.0	**44.2**	**69.2**	27.9	50.42	56.6	80.2

Table 1: This table compares the final accuracies (Top1 - Top5) of the full precision network with our binary precision networks; Binary-Weight-Networks(BWN) and XNOR-Networks(XNOR-Net) and the competitor methods; BinaryConnect(BC) and BinaryNet(BNN).

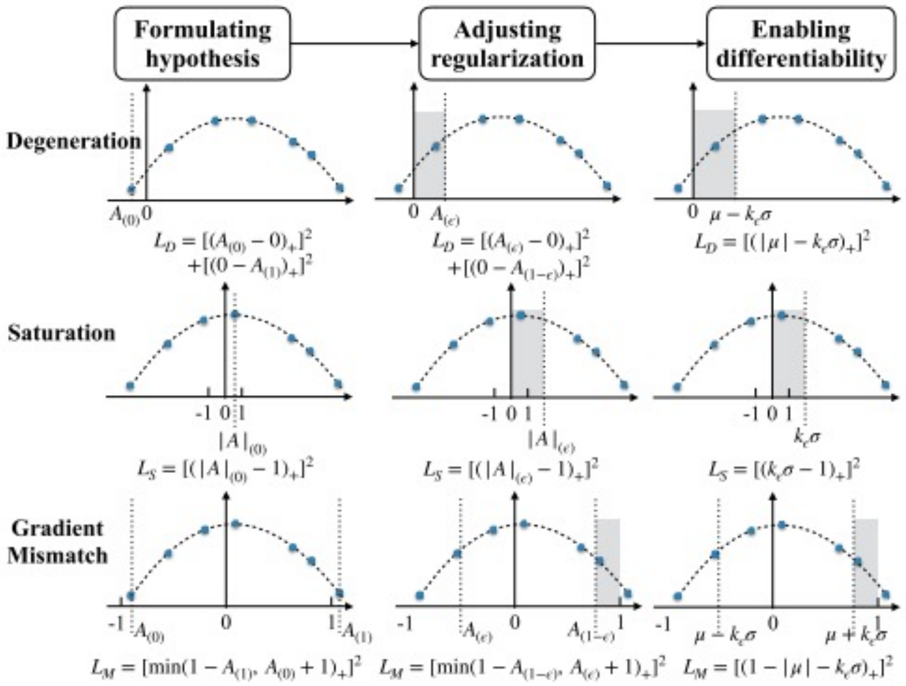
RAD



Regularizing activation distribution for training binarized deep networks

Improve Network Loss Function

RAD



Degeneration: $A_{(0)} \geq 0$ or $A_{(1)} \leq 0$

Saturation: $|A|_{(0)} \geq 1$

Gradient mismatch: $|A|_{(1)} \leq 1$

$$L^b_{DL} = \sum_{l,c} L^{b,l,c}_{DL} = \sum_{l,c} L^{b,l,c}_{D} + L^{b,l,c}_{S} + L^{b,l,c}_{M}$$

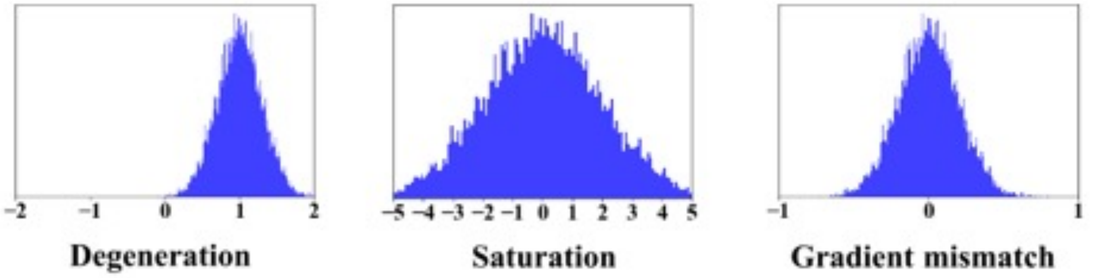
$$L^b_{total} = L^b_{CE} + \lambda L^b_{DL}$$

Improve Network Loss Function

RAD

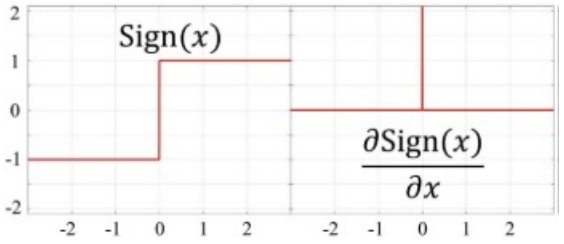
Table 4: Comparison with prior art using 1-bit weights and activations, in terms of accuracy and computation energy on different datasets. The best results are shown in bold face.

Dataset	Model	Pure-logical	Energy cost	Accuracy
CIFAR-10	BNN [22]	Yes	1×	87.13%
	XNOR-Net [38]	No	4.5×	87.38%
	LAB [19]	No	4.5×	87.72%
	BNN-DL	**Yes**	1×	**89.90%**
SVHN	BNN [22]	Yes	1×	96.50%
	XNOR-Net [38]	No	4.5×	96.57%
	LAB [19]	No	4.5×	96.64%
	BNN-DL	**Yes**	1×	**97.23%**
CIFAR-100	BNN [22]	No	1×	60.40%
	DQ-2bit [37]	No	-	49.32%
	BNN-DL	**No**	1×	**68.17%**

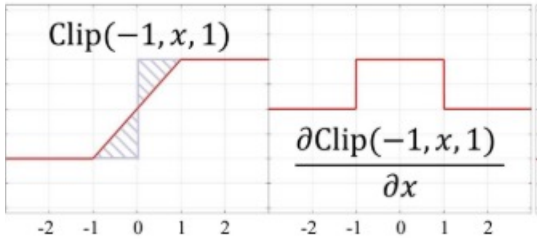


Reduce the Gradient Error

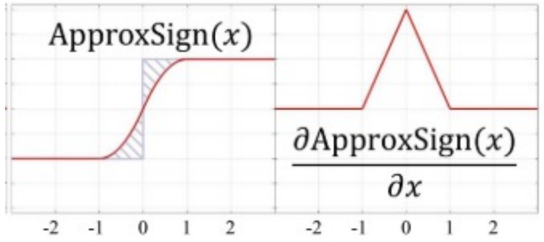
Bi-Real Net



(a)



(b)



(c)

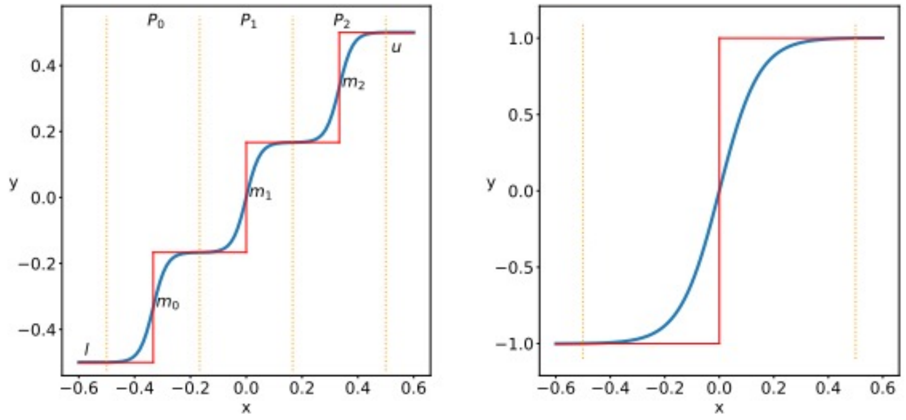
$$\text{sign:}\quad f(x) = \begin{cases} -1 & x < 0 \\ 1 & \text{otherwise} \end{cases} \quad f'(x) = \begin{cases} \infty & x = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{STE:}\quad f(x) = \begin{cases} -1 & x < -1 \\ x & x \in [-1, 1] \\ 1 & x > 1 \end{cases} \quad f'(x) = \begin{cases} 1 & x \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}$$

$$F(a_r) = \begin{cases} -1 & \text{if } a_r < -1 \\ 2a_r + a_r^2 & \text{if } -1 \leqslant a_r < 0 \\ 2a_r - a_r^2 & \text{if } 0 \leqslant a_r < 1 \\ 1 & \text{otherwise} \end{cases}, \quad \frac{\partial F(a_r)}{\partial a_r} = \begin{cases} 2 + 2a_r & \text{if } -1 \leqslant a_r < 0 \\ 2 - 2a_r & \text{if } 0 \leqslant a_r < 1 \\ 0 & \text{otherwise} \end{cases}$$

Bi-Real Net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm

DSQ



(a) Piecewise DSQ (b) Binary DSQ

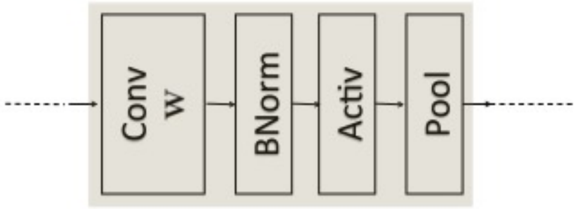
Binary DSQ:

$$f(x) = \tanh(kx)$$

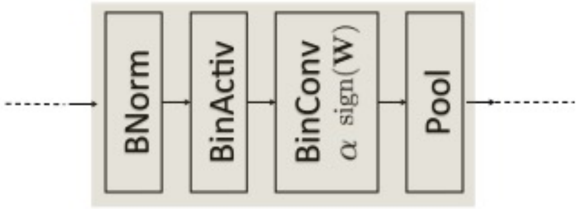
$$f'(x) = k(1 - \tanh^2(kx))$$

Tricks

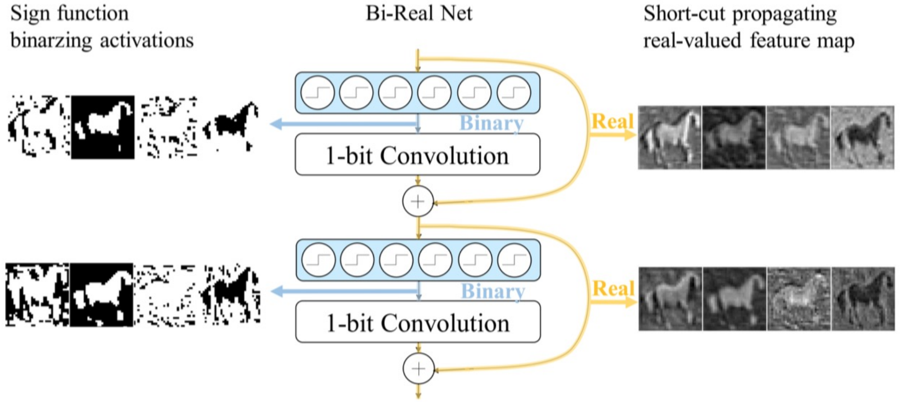
Structure Transformation



A typical block in CNN



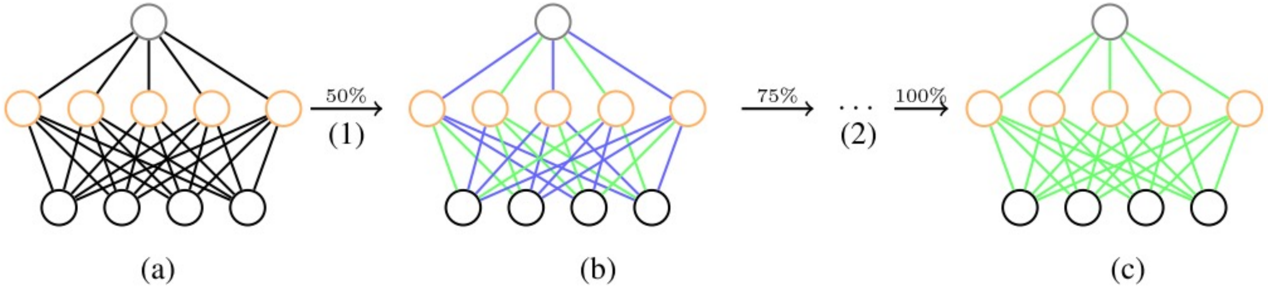
A block in XNOR-Net



Optimizer and Hyper-parameter Selection

ADAM optimizer; smaller weight decay; specific batch normalization' s momentum coefficient; *etc.*

Asymptotic Quantization



Challenges of BNNs

High Performance

- Higher accuracy
- Higher speedup
- Higher compression rate

Strong Versatility

- More type of tasks
- Fewer specific network structures
- Easier hardware deployment

Contents

3 IR-Net: 信息保留的二值神经网络

Forward and Backward Information Retention for Accurate Binary Neural Networks

CVPR 2020

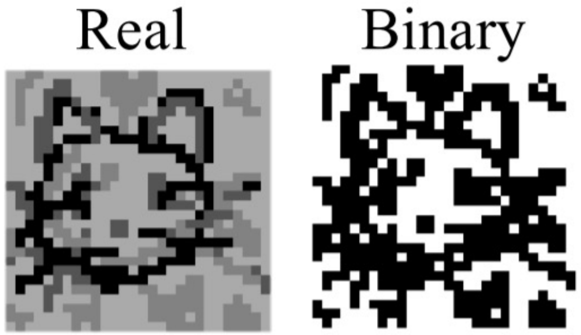
ArXiv: https://arxiv.org/abs/1909.10788

GitHub: https://github.com/htqin/IR-Net

News: https://mp.weixin.qq.com/s/cF14wwgnMcnvkBa864ox1Q

Why BNN suffer a significant accuracy drop?

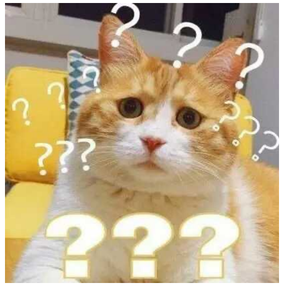
Forward



32-bit → 1-bit

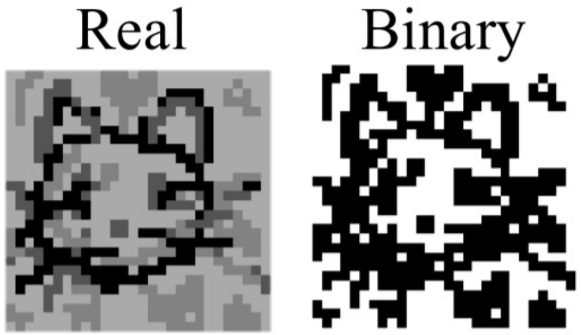
backward

The model’s diversity sharply decreases, while the diversity is proved to be the key of pursuing high accuracy of neural networks.



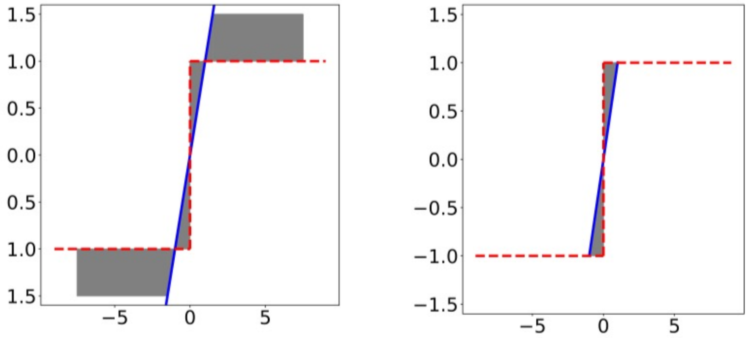
Why BNN suffer a significant accuracy drop?

Forward



32-bit → 1-bit

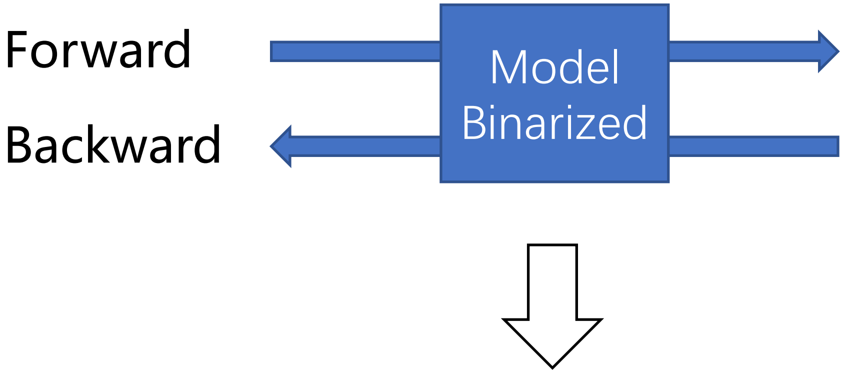
backward



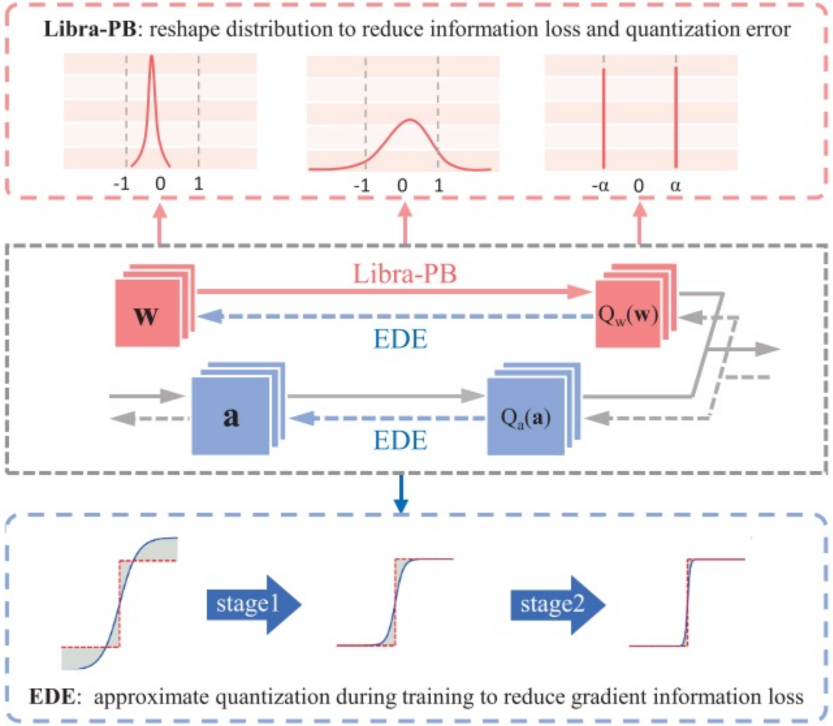
(a) Identity (b) Clip

The discrete binarization always leads to inaccurate gradients and the wrong optimization direction. (Saturation and Gradient mismatch)

Forward and backward information retention (IR-Net)



The information loss cause a significant accuracy drop



Forward: Libra-PB

Maximize the Information Entropy

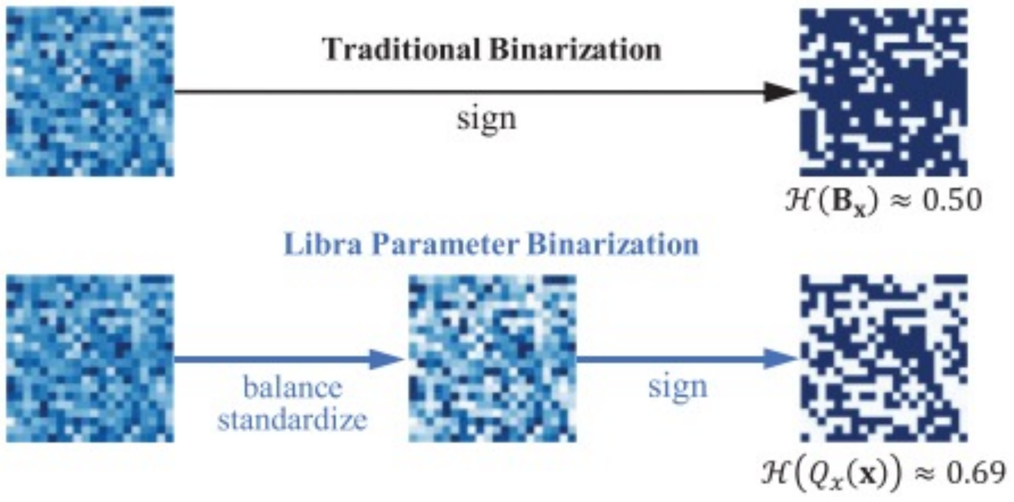
$$f(b)=\begin{cases} p, & if\ b=+1 \\ 1-p, & if\ b=-1, \end{cases}$$

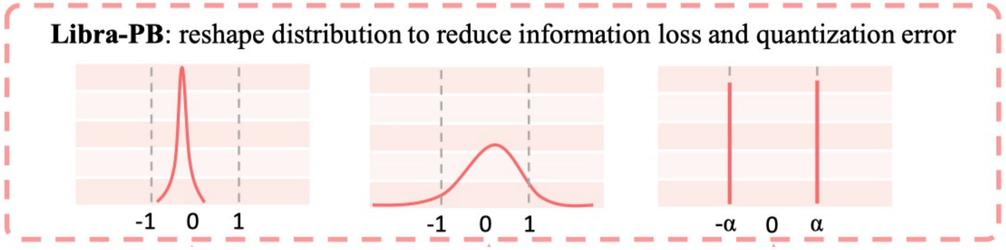
$$\mathcal{H}(Q_x(\mathbf{x}))=\mathcal{H}(\mathbf{B_x})=-p\ln(p)-(1-p)\ln(1-p).$$

$$\min J(Q_x(\mathbf{x}))-\lambda\mathcal{H}(Q_x(\mathbf{x})).$$

$$\hat{\mathbf{w}}_{\text{std}}=\frac{\hat{\mathbf{w}}}{\sigma(\hat{\mathbf{w}})},\quad \hat{\mathbf{w}}=\mathbf{w}-\overline{\mathbf{w}}.$$

$$\mathbb{E}[z]=Q_w(\hat{\mathbf{w}}_{\text{std}})^\top\mathbb{E}[Q_a(\mathbf{a})]=Q_w(\hat{\mathbf{w}}_{\text{std}})^\top\mu\mathbf{1}.$$



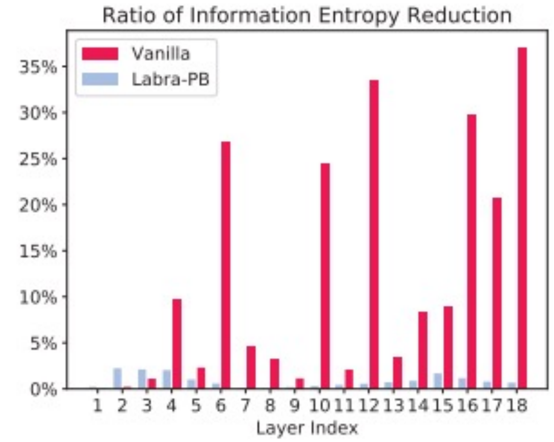


The information entropy of weight and activation is maximized.

Forward: Libra-PB

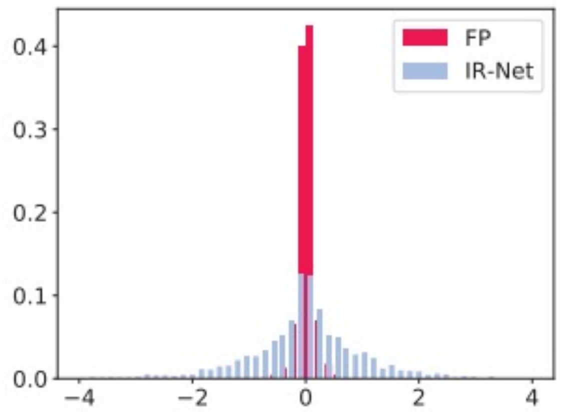
Maximize the information entropy

$$\hat{\mathbf{w}} = \mathbf{w} - \overline{\mathbf{w}}.$$



Stabilize the training process

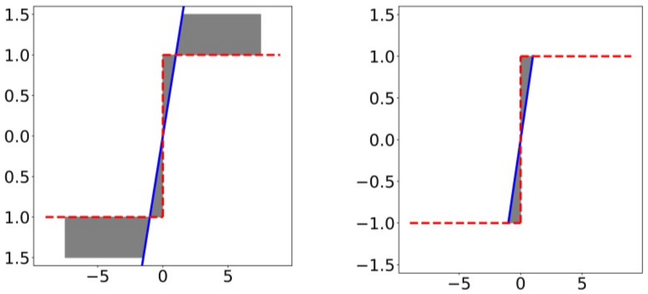
$$\hat{\mathbf{w}}_{\text{std}} = \frac{\hat{\mathbf{w}}}{\sigma(\hat{\mathbf{w}})},$$



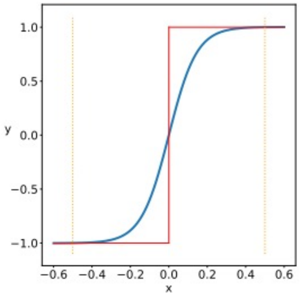
Backward: EDE

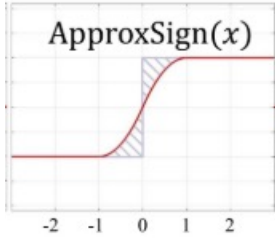
Retain the Information of Gradient

$\texttt{Identity}: y = x$ or $\texttt{Clip}: y = \texttt{Hardtanh}(x).$

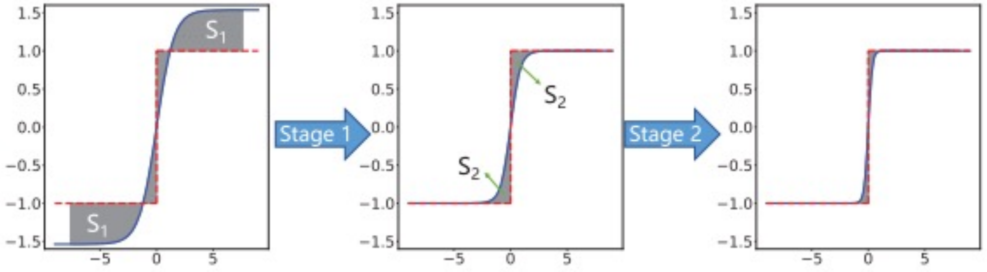


(a) `Identity` (b) `Clip`





(b) Binary DSQ

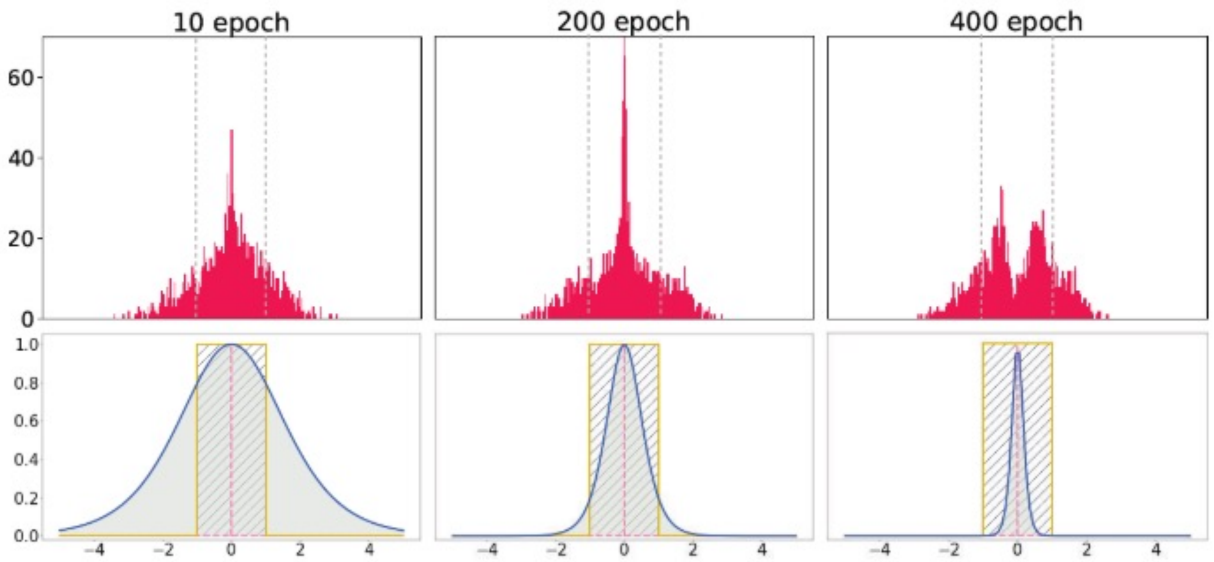


(c) EDE

$$g(x) = k \tanh tx$$

Backward: EDE

Minimize the Information Loss of Gradient



Results

Table 3: Accuracy comparison with SOTA methods on CIFAR-10.

Topology	Method	Bit-width (W/A)	Acc.(%)
ResNet-18	FP	32/32	93.0
	RAD	1/1	90.5
	Ours[1]	1/1	**91.5**
ResNet-20	FP	32/32	91.7
	DoReFa	1/1	79.3
	DSQ	1/1	84.1
	Ours[1]	1/1	**85.4**
	Ours[2]	1/1	**86.5**
	FP	32/32	91.7
	DoReFa	1/32	90.0
	LQ-Net	1/32	90.1
	DSQ	1/32	90.2
	Ours[1]	1/32	**90.8**
VGG-Small	FP	32/32	91.7
	LAB	1/1	87.7
	XNOR	1/1	89.8
	BNN	1/1	89.9
	RAD	1/1	90.0
	Ours	1/1	**90.4**

Table 4: Accuracy comparison with SOTA methods on ImageNet.

Topology	Method	Bit-width (W/A)	Top-1(%)	Top-5(%)
ResNet-18	FP	32/32	69.6	89.2
	ABC-Net	1/1	42.7	67.6
	XNOR	1/1	51.2	73.2
	BNN+	1/1	53.0	72.6
	DoReFa	1/2	53.4	–
	Bi-Real	1/1	56.4	79.5
	XNOR++	1/1	57.1	79.9
	Ours[2]	1/1	**58.1**	**80.0**
	FP	32/32	69.6	89.2
	SQ-BWN	1/32	58.4	81.6
	BWN	1/32	60.8	83.0
	HWGQ	1/32	61.3	83.2
	TWN	2/32	61.8	84.2
	SQ-TWN	2/32	63.8	85.7
	BWHN	1/32	64.3	85.9
	Ours[1]	1/32	**66.5**	**86.8**
ResNet-34	FP	32/32	73.3	91.3
	ABC-Net	1/1	52.4	76.5
	Bi-Real	1/1	62.2	83.9
	Ours[2]	1/1	**62.9**	**84.1**
	FP	32/32	73.3	91.3
	Ours[1]	1/32	**70.4**	**89.5**

[1]Results of ResNet with normal structure [22].
[2]Results of ResNet with Bi-Real structure [38].

High Performance and Strong Versatility

Results (Hardware Deployment)

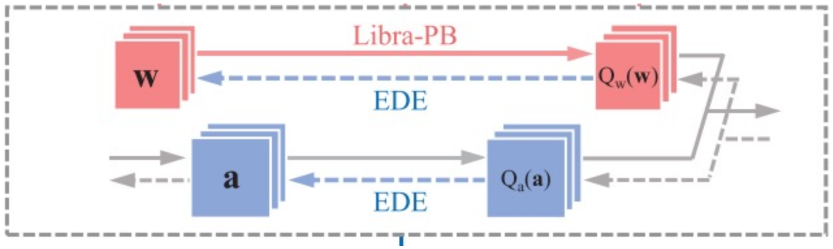


Table 5: Comparison of time cost of ResNet-18 with different bits (single thread).

Method	Bit-width (W/A)	Size (Mb)	Time (ms)
FP	32/32	46.77	1418.94
NCNN	8/8	–	935.51
DSQ	2/2	–	551.22
Ours (without bit-shift scales)	1/1	**4.20**	**252.16**
Ours	1/1	**4.21**	**261.98**

Based on daBNN (Open sourced by JD.com)

Conclusion

- Take away:
  - The IR-Net let the diversity of binary neural networks be kept as much as possible by forward and backward information retention.
  - On Hardware, the inference speed of IR-Net is much faster, and the model size of IR-Net can be greatly reduced.

- Further work:
  - Higher-performance and faster BNNs.
  - Apply BNNs to more tasks (detection, segmentation, *etc.*).

Q&A

Thank you!