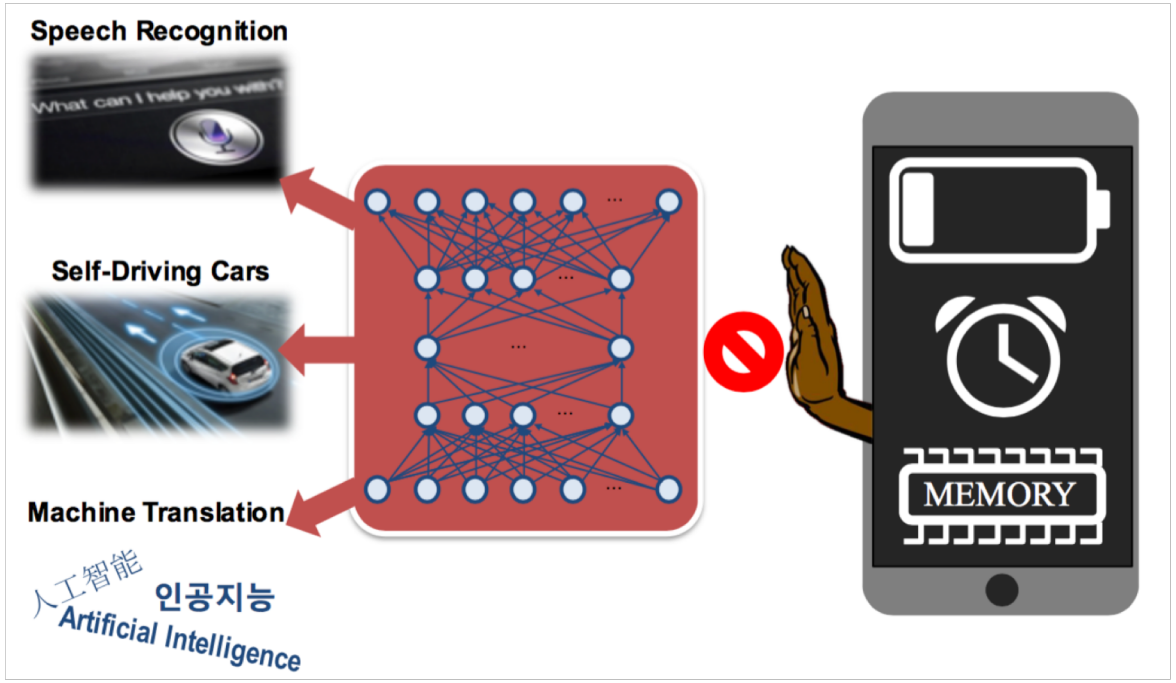
Projection Convolutional Neural Networks for 1-bit CNNs via Discrete Back Propagation

Jiaxin Gu,[1] Ce Li,[2] Baochang Zhang,[1*] Jungong Han,[3] Xianbin Cao,[1]
Jianzhuang Liu,[4] David Doermann[5]

[1]Beihang University,
[2]China University of Mining &Technology, Beijing, [3]Lancaster University
[4]Huawei Noah's Ark Lab, [5]University at Buffalo

Why 1 bit convolution?



- 58 × faster operation
- 32 × memory saving[1]

1. reported by XNOR-Net

58 × faster convolution operation

Dot product

$$x \odot y = 2 \times bitcount\big(XNOR(x, y)\big) - N$$

$$where\ x_i, y_i \in \{0,1\}\ \forall\ i$$

$$N\ is\ the\ total\ number\ of\ bits$$

i.e.

$A = 10010$

$B = 01111$, *here* 0 *represents* -1

Traditional product:

$A \odot B$ = (1*-1)+(-1*1)+(-1*1)+(1*1)+(-1*1)=**-3**

Accelerated product:

$XNOR(A, B) = 00010$

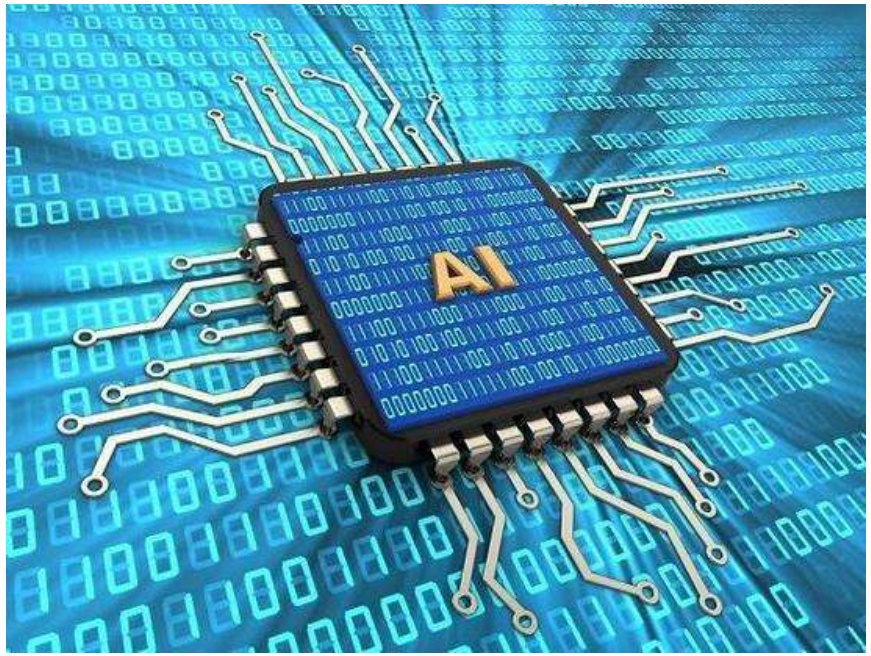
$bitcount\big(XNOR(x, y)\big)$ = 1

$A \odot B$= 2×1−5=**−3**

32 × memory saving

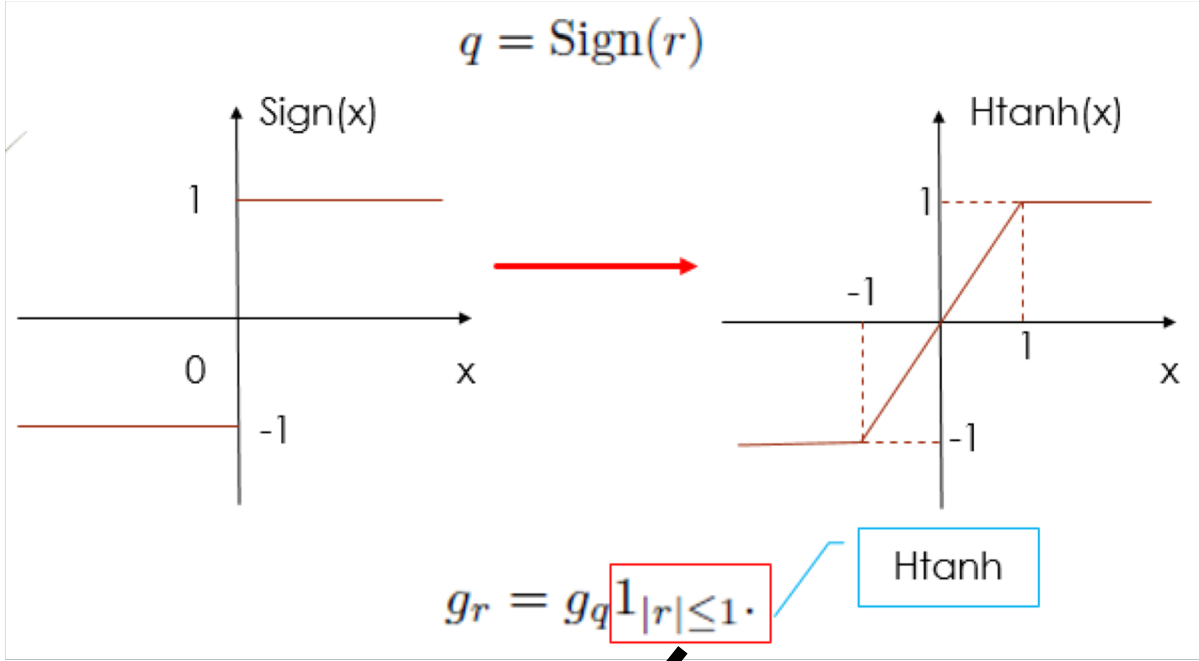
Model storage: FP32 -> 1 bit

To design specific AI chips



Related Work

BNN



Forward: **1 bit**

Backward: **full-precision**

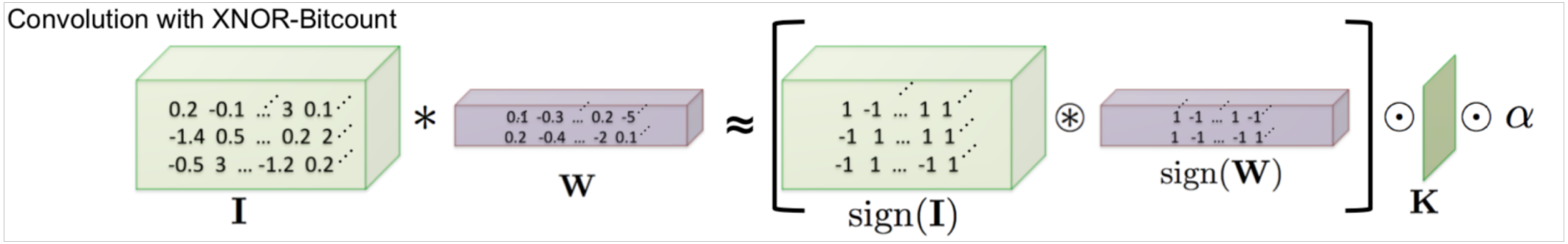
STE: Straight-through Estimator

Courbariaux, M; et. al, Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1.

Related Work

XNOR-Net

- $$\alpha^* = \frac{\mathbf{W}^\mathsf{T}\,\mathrm{sign}(\mathbf{W})}{n} = \frac{\sum|\mathbf{W}_i|}{n} = \frac{1}{n}\|\mathbf{W}\|_{\ell 1}$$
- K is aborted

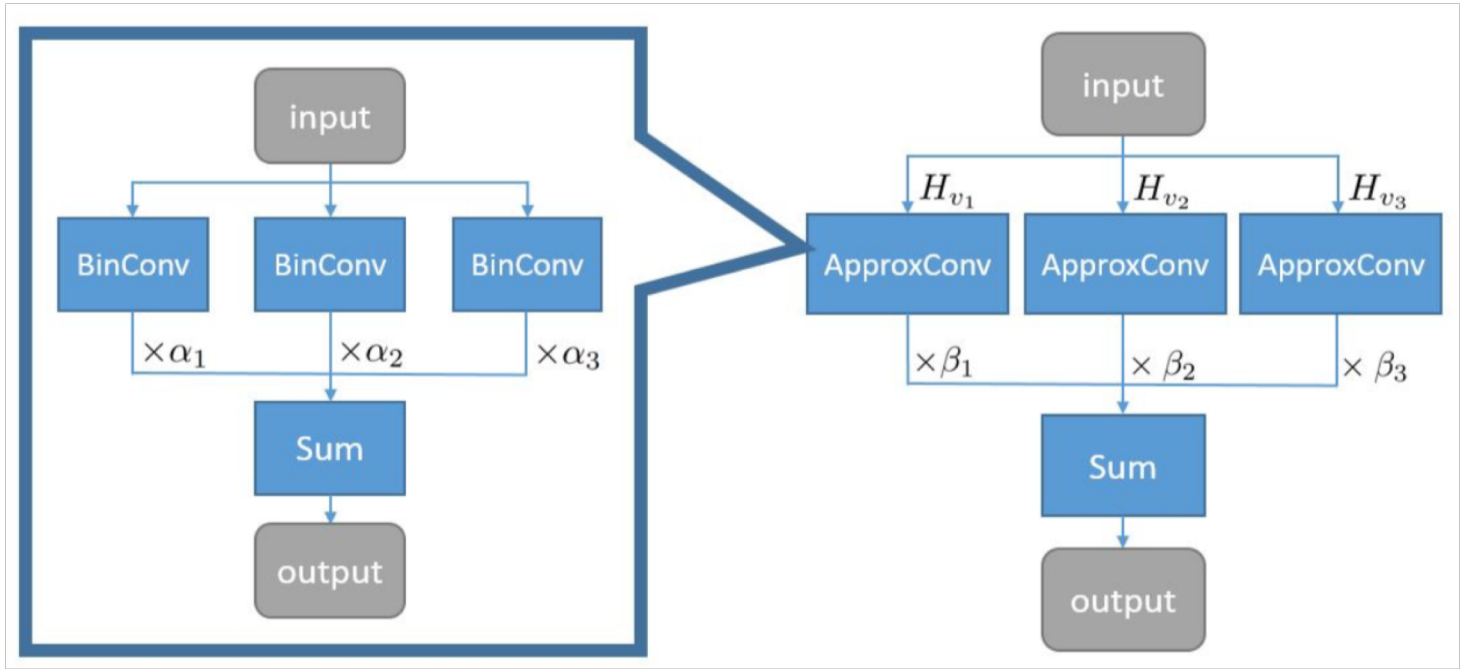


Rastegari, et.al., Xnor-net: Imagenet classification using binary convolutional neural networks.

Related Work

ABC-Net

- Linear combination of multiple binary weight bases
- Multiple binary activations



Lin, X et.al . Towards accurate binary convolutional neural network.

Bi-Real Net

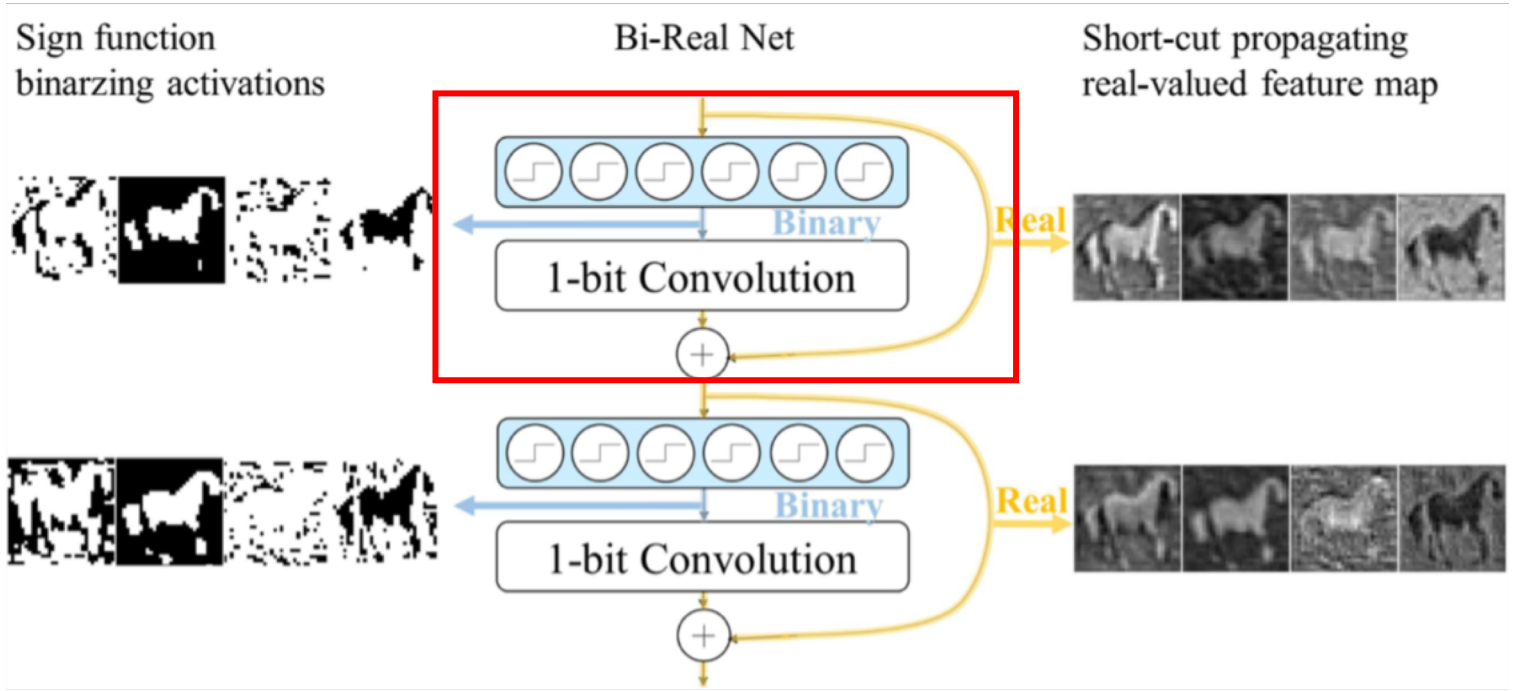
- Modified ResNet architecture
- Approximation to the derivative of the sign function with respect to activations
- Magnitude-aware gradient with respect to weights
- Initialization: progressive training

Liu, Z.; et.al . Bi-real net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm.

Related Work

Bi-Real Net

Modified ResNet architecture is adopted



Liu, Z.; et.al., Bi-real net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm.

Problem

Accuracy degradation is *large* **on ImageNet**

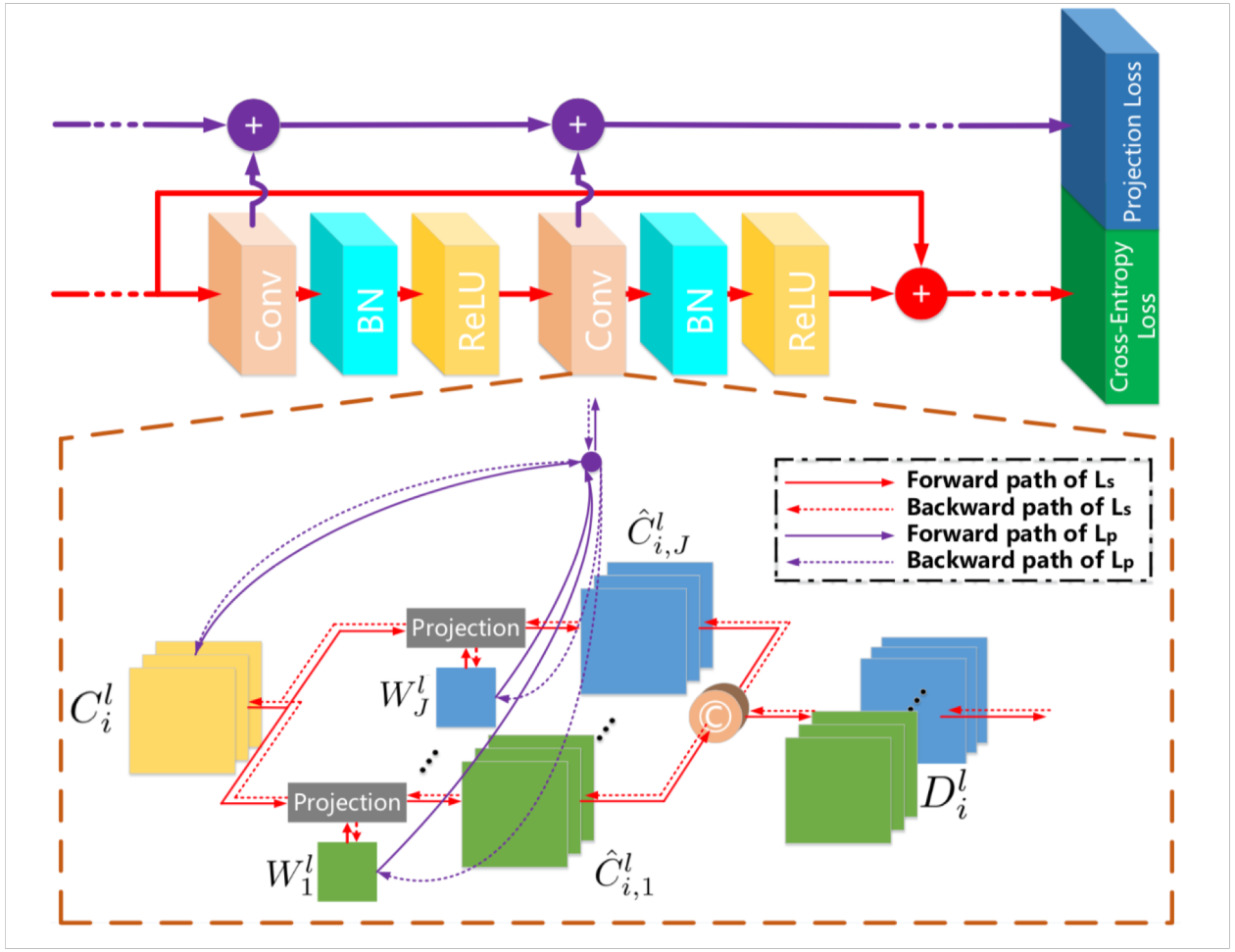
Model	W	A	Top-1	Top-5
Resnet18	32	32	69.3	89.2
BWN	1	32	60.8	83.0
DoReFa-Net	1	4	59.2	81.5
XNOR-Net	1	1	51.2	73.2
ABC-Net	1	1	42.7	67.6
BNN	1	1	42.2	67.1
Bi-Real Net	1	1	56.4	79.5
PCNN	1	32	$63.5^{\dagger}$, $66.1^{\ddagger}$	$85.1^{\dagger}$, $86.7^{\ddagger}$
PCNN	1	1	**57.3**†	**80.0**†

Method

Projection Convolutional Neural Networks

$$L = L_s + L_P$$

Layer-wise Projection loss



Method

Projection

$$P_\Omega(x) = \arg\min_{a_i} \|x - a_i\|, i \in \{1, ..., U\}$$

Optimization

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & \hat{x}_j = P_\Omega^j(\omega_j \circ x), \end{aligned}$$

Lagrangian method

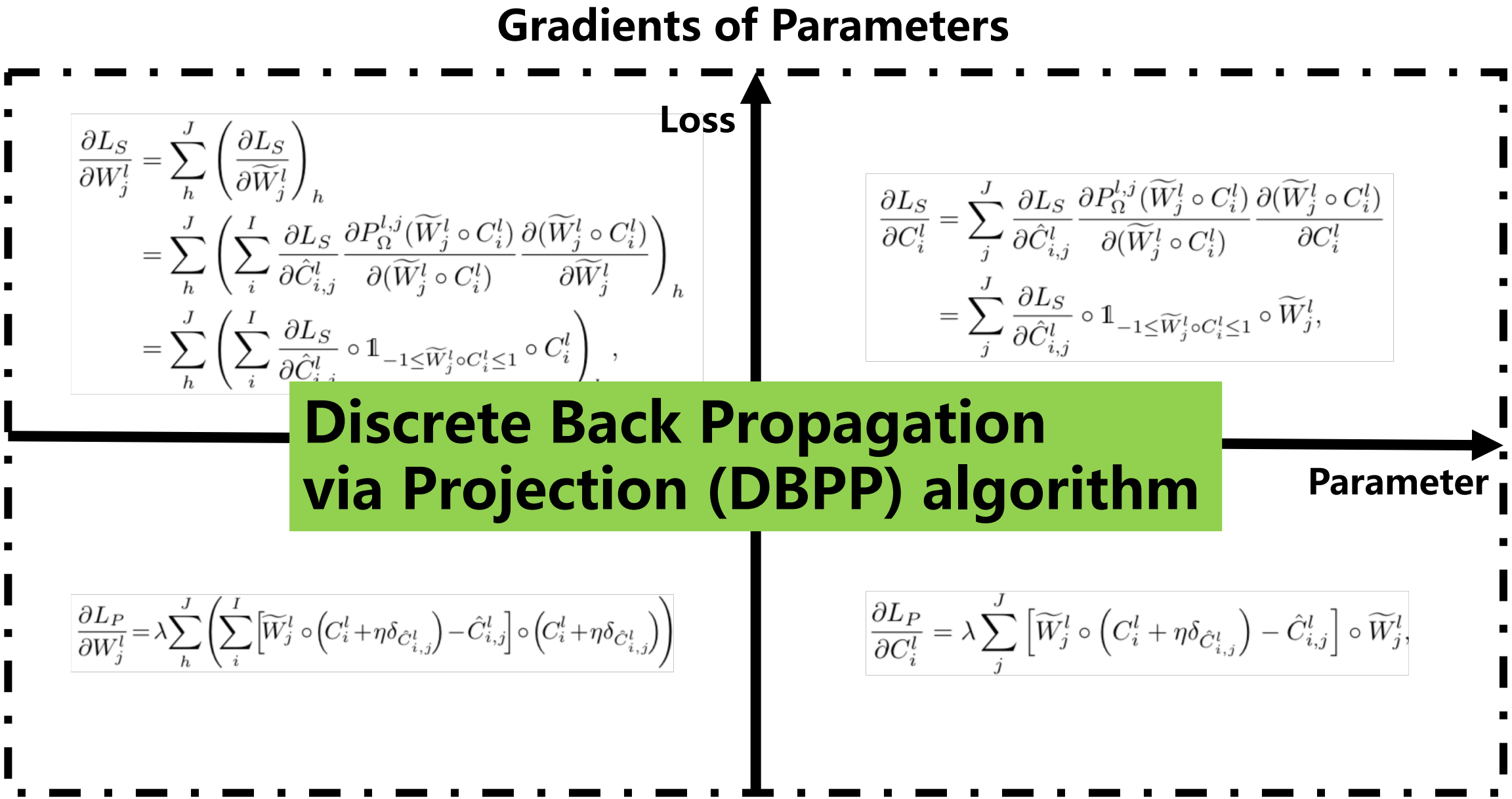
$$\min f(x) + \lambda \sum_j^J \|\hat{x}^{[k]} - \omega_j \circ (x + \eta \delta_{\hat{x}}^{[k]})\|^2$$

Projection loss in CNNs

$$L_p = \frac{\lambda}{2} \sum_{l,i}^{L,I} \sum_j^J ||\hat{C}_{i,j}^{l,[k]} - \widetilde{W}_j^{l,[k]} \circ (C_i^{l,[k]} + \eta \delta_{\hat{C}_{i,j}^{l,[k]}})||^2$$

Parameters to be learned

Method


Gradients of Parameters
Loss
Parameter
Discrete Back Propagation via Projection (DBPP) algorithm
$\frac{\partial L_S}{\partial W_j^l} = \sum_h^J \left(\frac{\partial L_S}{\partial \widetilde{W}_j^l}\right)_h = \sum_h^J \left(\sum_i^I \frac{\partial L_S}{\partial \hat{C}_{i,j}^l} \frac{\partial P_\Omega^{l,j}(\widetilde{W}_j^l \circ C_i^l)}{\partial(\widetilde{W}_j^l \circ C_i^l)} \frac{\partial(\widetilde{W}_j^l \circ C_i^l)}{\partial \widetilde{W}_j^l}\right)_h = \sum_h^J \left(\sum_i^I \frac{\partial L_S}{\partial \hat{C}_{i,j}^l} \circ \mathbb{1}_{-1 \le \widetilde{W}_j^l \circ C_i^l \le 1} \circ C_i^l\right)_h,$
$\frac{\partial L_S}{\partial C_i^l} = \sum_j^J \frac{\partial L_S}{\partial \hat{C}_{i,j}^l} \frac{\partial P_\Omega^{l,j}(\widetilde{W}_j^l \circ C_i^l)}{\partial(\widetilde{W}_j^l \circ C_i^l)} \frac{\partial(\widetilde{W}_j^l \circ C_i^l)}{\partial C_i^l} = \sum_j^J \frac{\partial L_S}{\partial \hat{C}_{i,j}^l} \circ \mathbb{1}_{-1 \le \widetilde{W}_j^l \circ C_i^l \le 1} \circ \widetilde{W}_j^l,$
$\frac{\partial L_P}{\partial W_j^l} = \lambda \sum_h^J \left(\sum_i^I \left[\widetilde{W}_j^l \circ \left(C_i^l + \eta \delta_{\hat{C}_{i,j}^l}\right) - \hat{C}_{i,j}^l\right] \circ \left(C_i^l + \eta \delta_{\hat{C}_{i,j}^l}\right)\right)$
$\frac{\partial L_P}{\partial C_i^l} = \lambda \sum_j^J \left[\widetilde{W}_j^l \circ \left(C_i^l + \eta \delta_{\hat{C}_{i,j}^l}\right) - \hat{C}_{i,j}^l\right] \circ \widetilde{W}_j^l,$

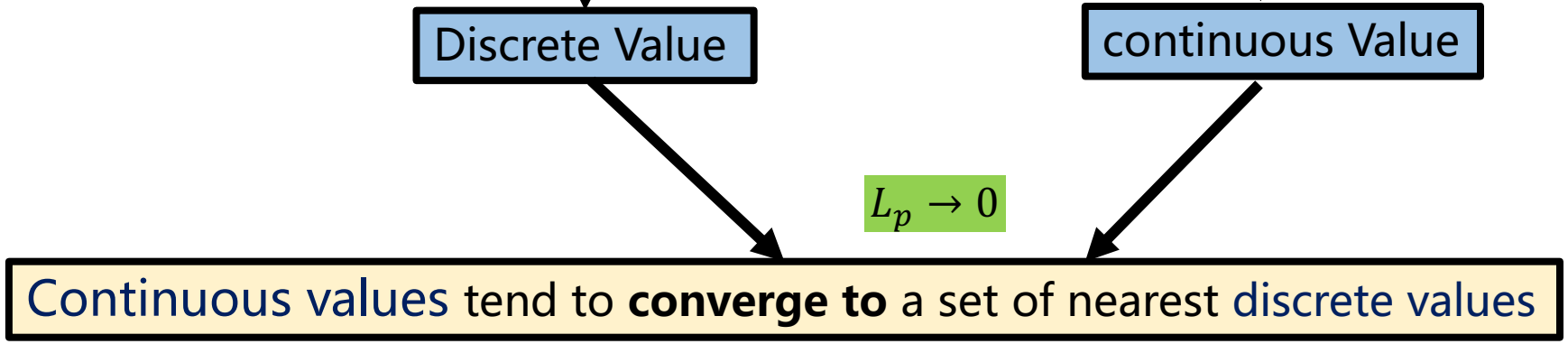
Contributions:

- DBPP algorithm is proposed with multiple projections to learn a set of diverse quantized kernels.
- A projection loss is theoretically achieved in DBPP.
- PCNNs achieve the state-of-the-art results on ImageNet and CIFAR dataset.

Method

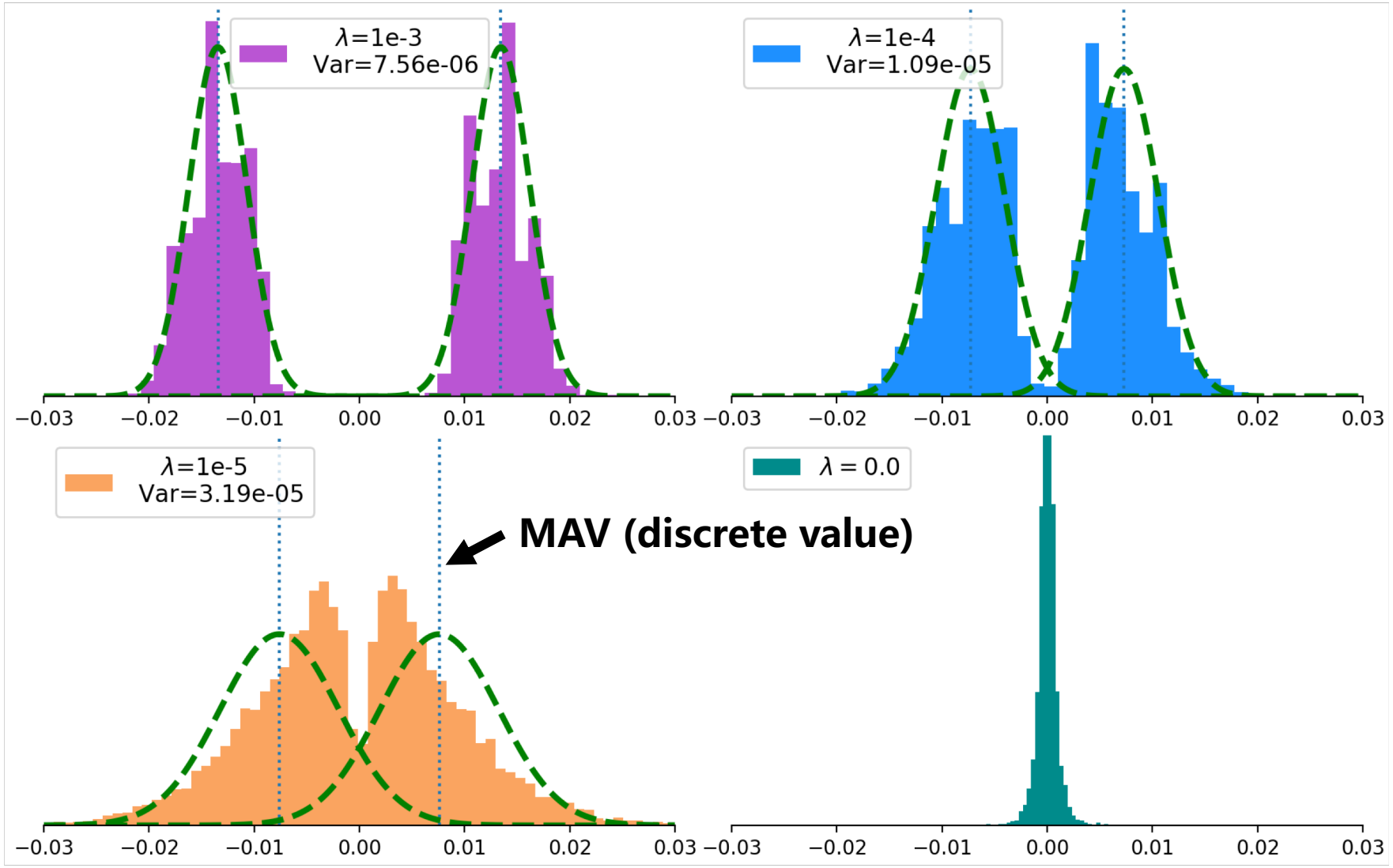
A close look at Lp

$$L_p = \frac{\lambda}{2} \sum_{l,i}^{L,I} \sum_{j}^{J} \left| \hat{C}_{i,j}^{l,[k]} - \widetilde{W}_{j}^{l,[k]} \circ \left(C_i^{l,[k]} + \eta \delta_{\hat{C}_{i,j}^{l,[k]}} \right) \right|^2$$



Experiments

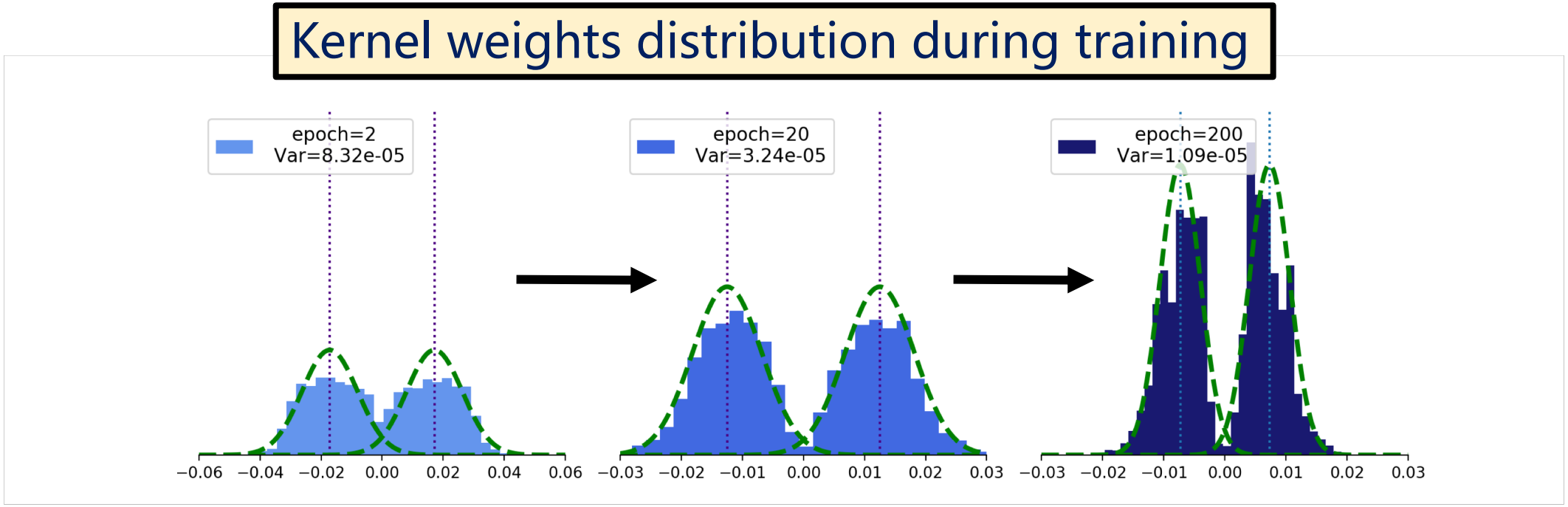
Kernel weights distribution



Larger λ

Smaller Var.

Experiments

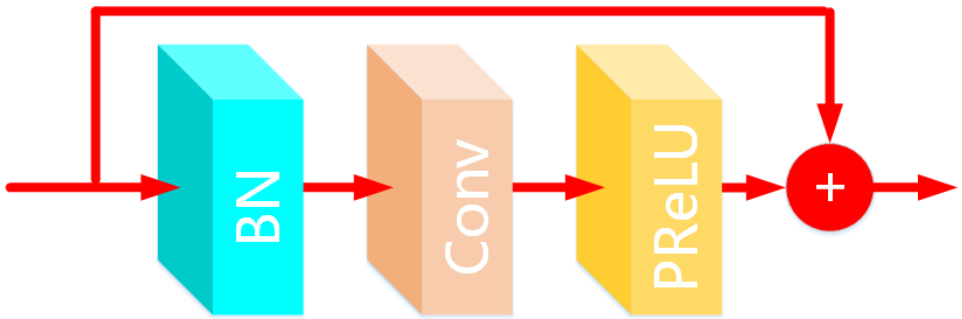


The most suitable λ

Model	λ			
	$1e-3$	$1e-4$	$1e-5$	0
PCNN-22	91.92	92.79	92.24	91.52
PCNN-40	92.85	93.78	93.65	92.84

Experiments

ResNet18 architecture modification



Experiments

Results on CIFAR and ImageNet dataset

Table 3: Test accuracy on CIFAR10/100 dataset. PCNNs are based on WRN-22. The numbers of parameters refer to the models on CIFAR10.

Model	#Para.	Dataset	
		CIFAR-10	CIFAR-100
WRN (i=16)	0.27M	92.62	68.83
WRN (i=64)	4.29M	95.75	77.34
BinaryConnect	14.02M	91.73	-
BNN	14.02M	89.85	-
LBCNN	14.02M	92.99	-
BWN	14.02M	90.12	-
XNOR-Net	14.02M	89.83	-
(McDonnell 2018)	4.30M	93.87	76.13
PCNN (i=16, J=1)	0.27M	89.17	62.66
PCNN (i=16, J=2)	0.54M	91.27	66.86
PCNN (i=16, J=4)	1.07M	92.79	70.09
PCNN (i=64, J=1)	4.29M	94.31	76.93
PCNN (i=64, J=4)	17.16M	95.39	78.13

Table 4: Test accuracy on ImageNet. 'W' and 'A' refer to the weight and activation bitwidth respectively. The first two PCNNs are based on Resnet18, while the last one is based on VGG16. † and ‡ indicate J=1 and J=2 respectively.

Model	W	A	Top-1	Top-5
Resnet18	32	32	69.3	89.2
BWN	1	32	60.8	83.0
DoReFa-Net	1	4	59.2	81.5
XNOR-Net	1	1	51.2	73.2
ABC-Net	1	1	42.7	67.6
BNN	1	1	42.2	67.1
Bi-Real Net	1	1	56.4	79.5
PCNN	1	32	$63.5^{\dagger}$, $66.1^{\ddagger}$	$85.1^{\dagger}$, $86.7^{\ddagger}$
PCNN	1	1	**57.3**†	**80.0**†
VGG16	32	32	73.0	91.2
PCNN	1	32	$69.0^{\dagger}$	$89.1^{\dagger}$

Experiments

Memory usage and efficiency comparison

Negligible additional computation cost

Table 5: Memory usage and efficiency of convolution comparison with XNOR-Net, full-precision Resnet18, and PCNN (J=1). PCNN is based on Resnet18.

Model	Memory usage	Memory saving	Speedup
PCNN	33.7 Mbit	11.10 ×	58 ×
XNOR-Net	33.7 Mbit	11.10 ×	58 ×
Resnet18	374.1 Mbit	-	-

Thanks