Towards Accurate Post-training Network Quantization via Bit-split and Stitching

Peisong Wang, Qiang Chen, Xiangyu He, Jian Cheng

Institute of Automation, Chinese Academy of Sciences

Outline

- Background
- Motivation
- Approach
- Experiments

Background

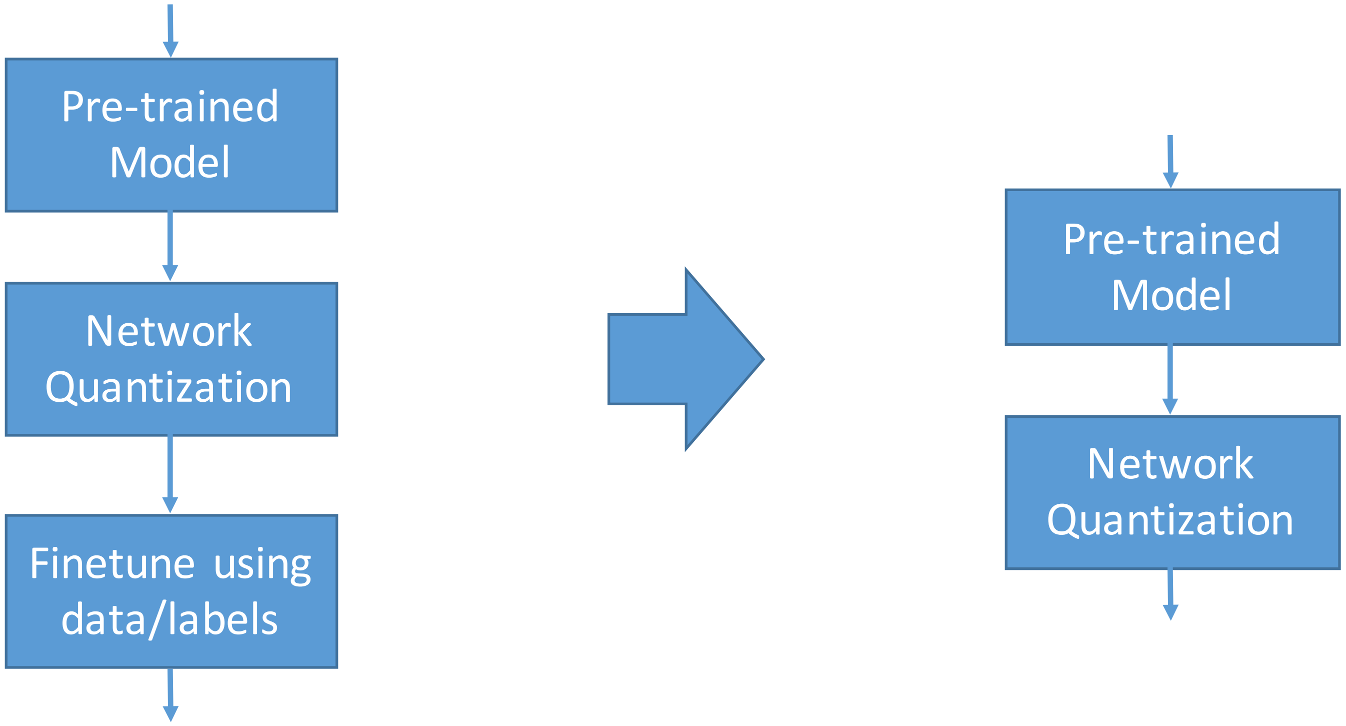
- **Low-bit quantization** has emerged as a promising compression technique
  - Robustness to network architectures
  - Hardware friendly

- **Problems:** low-bit quantization relies on
  - Training data
  - Large computational resources (CPUs, GPUs)
  - Quantization skills and expertise

Background

Training-aware quantization

Post-training Quantization



This work

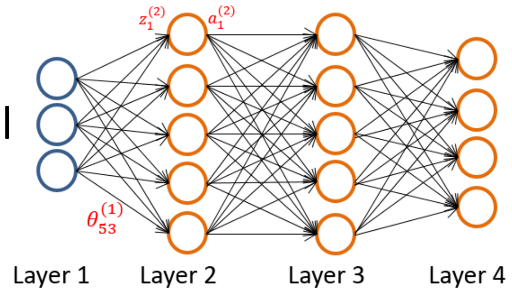
Data-free
BP-free
Easy to use

Krishnamoorthi, Raghuraman. "Quantizing deep convolutional networks for efficient inference: A whitepaper." arXiv preprint arXiv:1806.08342 (2018).

Motivation

Post-training quantization

Pretrained model

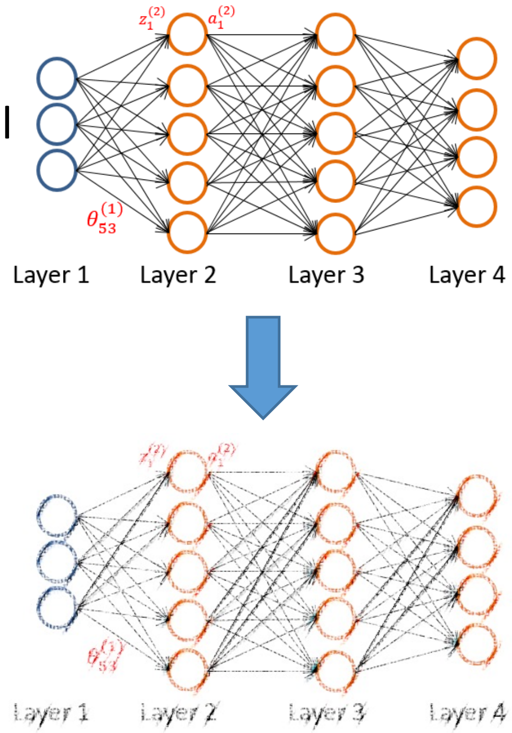


Motivation

Post-training quantization

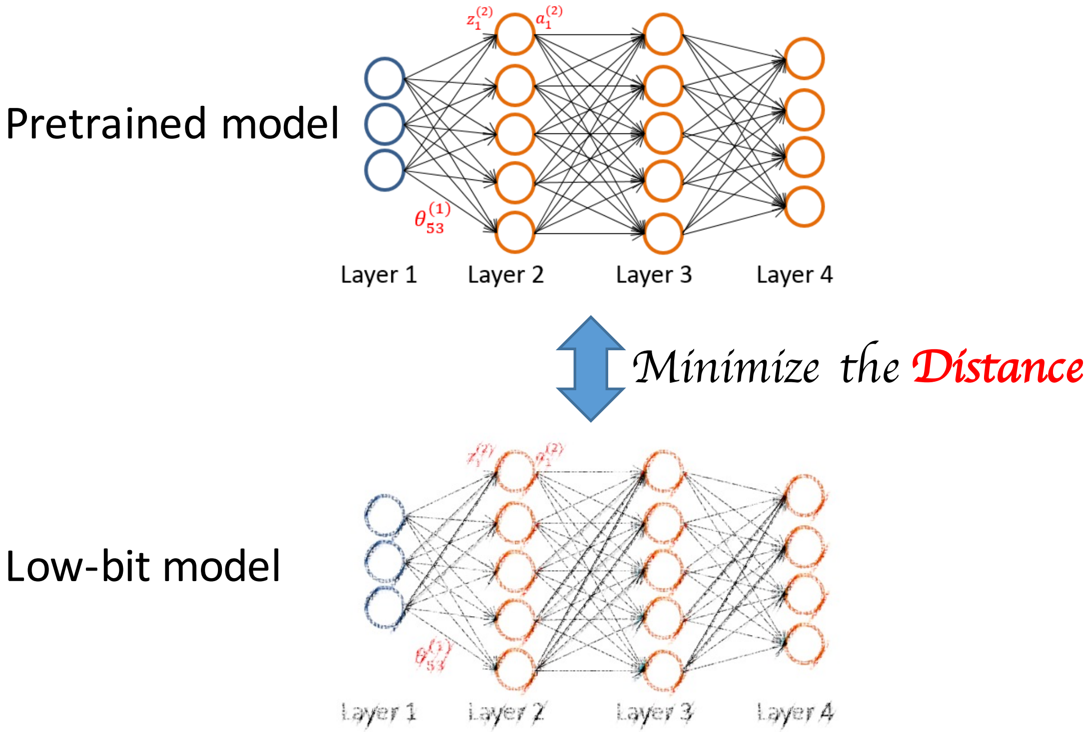
Pretrained model

Low-bit model



Motivation

Post-training quantization

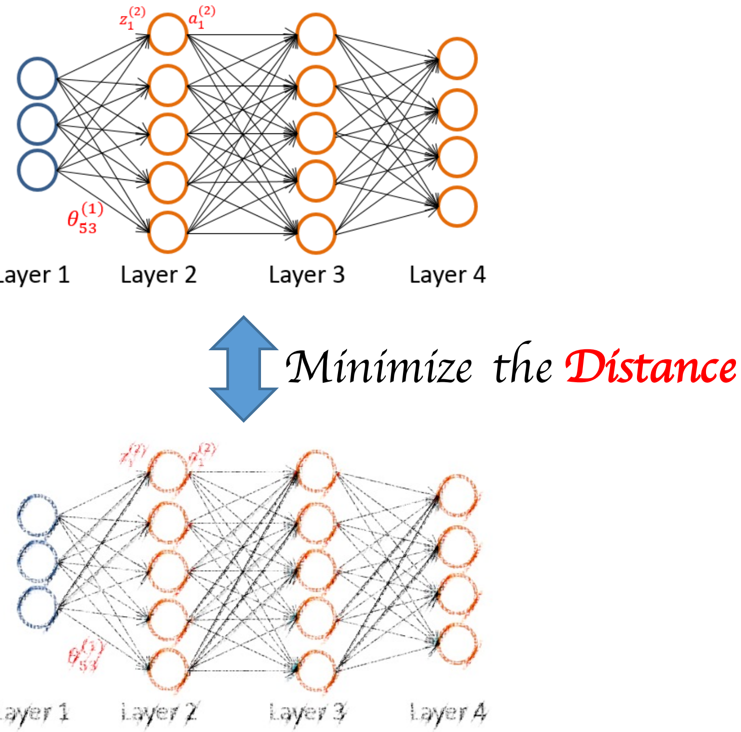


Motivation

Post-training quantization

Pretrained model

Low-bit model



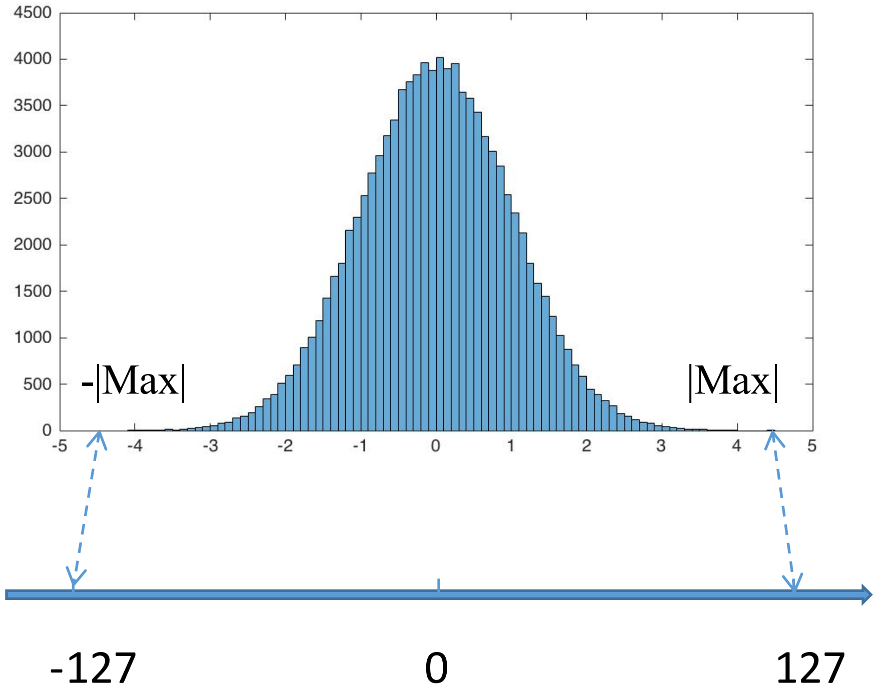
I. **Define the distance**

II. **Minimize the distance**

Related works

I. **Define the distance**

II. **Minimize the distance**



TF-lite

Map the ***maximum weighs (activations)*** to the maximum low-bit number

$$distance\ (w,\ \alpha q)$$

$$\alpha = \frac{max(|w|)}{2^{M-1}-1}$$

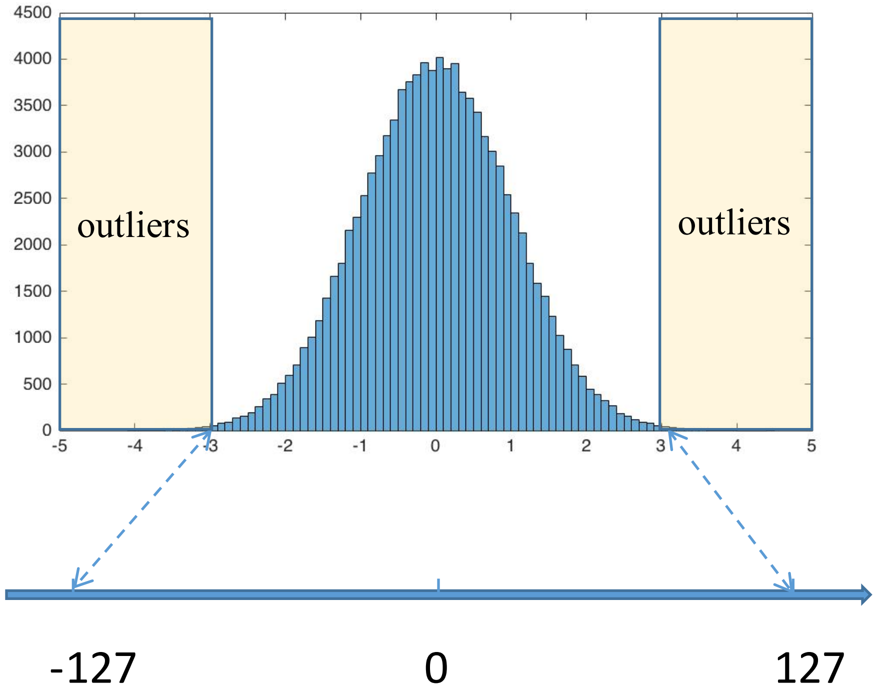
$$q = round(w/\alpha)$$

Krishnamoorthi, Raghuraman. "Quantizing deep convolutional networks for efficient inference: A whitepaper." arXiv preprint arXiv:1806.08342 (2018).

Related works

I. **Define the distance**

II. **Minimize the distance**



Szymon Migacz. 8-bit Inference with TensorRT. GTC 2017

TensorRT

Map the ***clip value***

to the maximum low-bit number

$$distance\ (clip(w),\ \alpha q)$$

$$\alpha = \frac{ClipValue}{2^{M-1}-1}$$

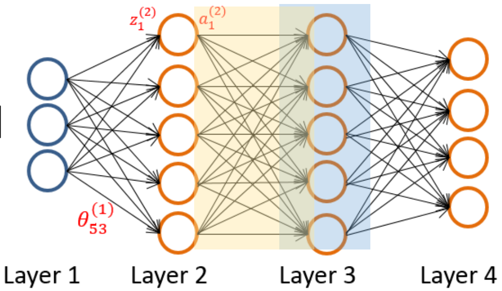
$$q = round(clip(w)/\alpha)$$

Method

I. **Define the distance**

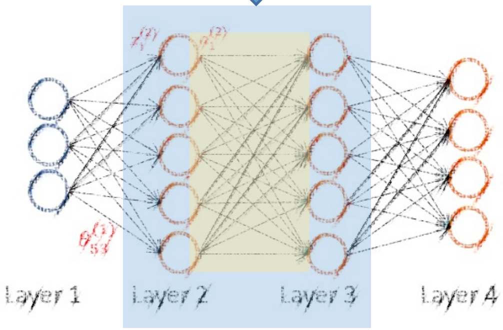
II. **Minimize the distance**

Pretrained model



Minimize the ***Distance***

Low-bit model



Objective

minimize: $f(x;\{w_l\}_{l=1}^{L}) \leftrightarrow f(x;\{q_l\}_{l=1}^{L})$

Previous work

minimize: $w_l \leftrightarrow q_l$

This work

Learns ***a low-bit mapping*** from input to the output of every convolution.

$$distance\ (w^T X,\ \alpha q^T X)$$

$$= distance\ (y,\ \alpha q^T X)$$

$$\underset{\alpha, q}{\text{minimize}} \parallel y - \alpha q^T X \parallel_F^2$$

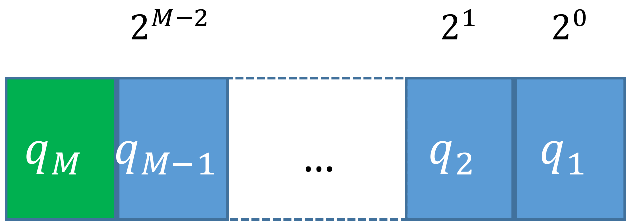
Method

I. Define the distance

$$\underset{\alpha, q}{\text{minimize}} \parallel y - \alpha q^T X \parallel_F^2$$

II. Minimize the distance (Bit-split)

$$\underset{\alpha, \{q_1, \cdots, q_{M-1}\}}{\text{minimize}} \parallel y - \alpha(2^0 q_1^T + \cdots + 2^{M-2} q_{M-1}^T) X \parallel_F^2,$$
$$s.t. \;\; q_m \in \{-1, 0, +1\}^{(C \cdot K_h \cdot K_w)} \text{ for } m = 1, \cdots, M-1$$



Method

Optimize α

$$\underset{\alpha,q}{\text{minimize}} \parallel y - \alpha q^T X \parallel_F^2$$

$$\alpha = \frac{y^T X^T q}{q^T X X^T q}$$

Optimize m-th bit

$$\underset{\alpha,\{q_1,\cdots,q_{M-1}\}}{\text{minimize}} \parallel y - \alpha(2^0 q_1^T + \cdots + 2^{M-2} q_{M-1}^T) X \parallel_F^2,$$
$$s.t.\ \ q_m \in \{-1,0,+1\}^{(C \cdot K_h \cdot K_w)} \text{ for } m = 1,\cdots, M-1$$

$$\underset{q_m}{\text{minimize}} \parallel y_m - \alpha_m q_m^T X \parallel_F^2,$$
$$s.t.\ \ q_m \in \{-1,0,+1\}^{(C \cdot K_h \cdot K_w)}$$

$$\begin{cases} y_m = y - \alpha \sum_{i \neq m} 2^{m-1} q_i^T X, \\ \alpha_m = \alpha 2^{m-2} \end{cases}$$

Wang, P., Hu, Q., Zhang, Y., Zhang, C., Liu, Y. and Cheng, J., 2018. Two-step quantization for low-bit neural networks. In Proceedings of the IEEE Conference on computer vision and pattern recognition (pp. 4376-4384).

Bit-Split for Post-training Network Quantization

Problem:

$$\underset{\alpha,q}{\text{minimize}} \parallel y - \alpha q^T X \parallel_F^2$$

Optimization:

$$\underset{\alpha,\{q_1,\cdots,q_{M-1}\}}{\text{minimize}} \parallel y - \alpha(2^0 q_1^T + \cdots + 2^{M-2} q_{M-1}^T) X \parallel_F^2,$$

$$s.t.\ \ q_m \in \{-1, 0, +1\}^{(C \cdot K_h \cdot K_w)} \text{ for } m = 1, \cdots, M-1$$

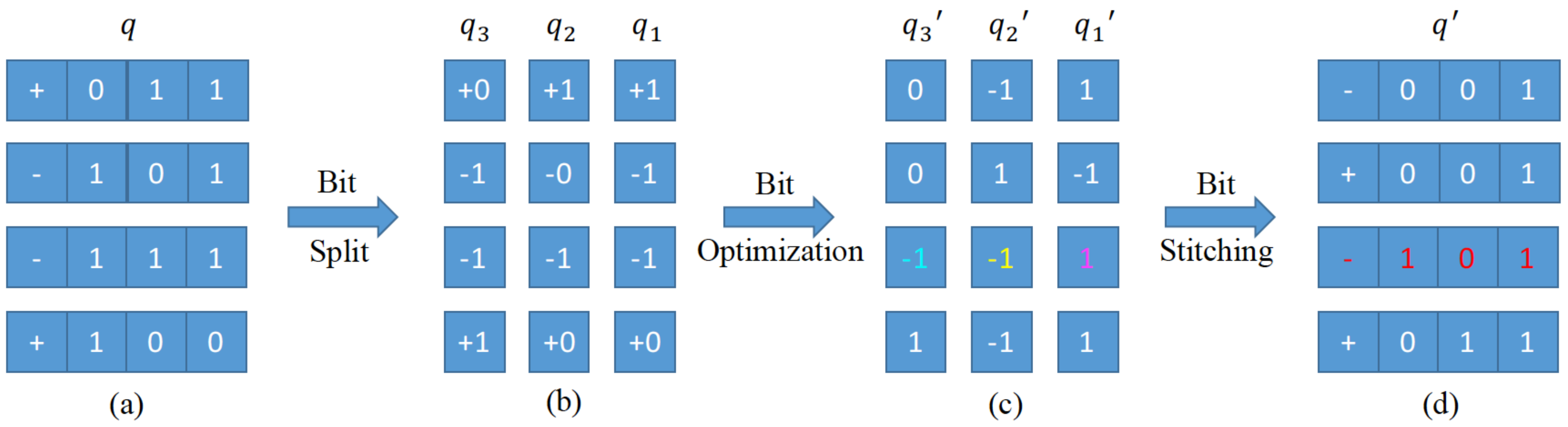


Figure 1. An illustration of Bit-Split and Stitching (Bit-split) framework for 4-bit weight quantization. In the first step of bit-split stage, each 4-bit value is split into 3 ternary values, which can be optimized separately in the second bit-optimization stage. The third stage stitching optimized bits back into integers, taking the third value for example, $2^0 \cdot \mathbf{1} + 2^1 \cdot (\text{-}\mathbf{1}) + 2^2 \cdot (\text{-}\mathbf{1}) = -5 = \textbf{-101}\text{b}$.

Bit-Split Results

Weight Quantization:

Model		8-bit		7-bit		6-bit		5-bit		4-bit		3-bit	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ResNet-18	TF-Lite	69.63	88.96	69.67	89.02	69.06	88.72	66.81	87.39	55.53	79.21	0.85	2.68
(69.76, 89.08)	Bit-split	69.79	**89.15**	**69.84**	**89.15**	**69.83**	**89.12**	**69.70**	88.93	**69.11**	**88.69**	**66.76**	87.45
	Bit-split (A8)	**69.82**	**89.15**	69.82	89.05	69.80	**89.12**	69.64	**88.98**	69.10	**88.69**	66.75	**87.46**
ResNet-50	TF-Lite	76.12	92.88	76.07	92.86	75.87	92.82	75.17	92.50	70.14	89.57	4.22	11.53
(76.15, 92.87)	Bit-split	**76.20**	**92.97**	**76.16**	**92.91**	**76.17**	**92.90**	**76.05**	**92.82**	**75.58**	**92.57**	**73.64**	**91.61**
ResNet-101	TF-Lite	77.32	93.57	77.28	93.51	77.06	93.47	76.25	93.05	72.67	90.87	9.19	20.05
(77.47, 93.56)	Bit-split	**77.55**	**93.59**	**77.44**	**93.59**	**77.51**	**93.60**	**77.55**	**93.59**	**76.89**	**93.31**	**74.98**	**92.42**
VGG-16-BN	TF-Lite	73.36	91.51	73.34	91.48	73.12	91.36	72.37	90.86	66.36	87.26	1.16	4.49
(73.37, 91.50)	Bit-split	**73.43**	**91.61**	**73.37**	**91.52**	**73.22**	**91.53**	**73.37**	**91.50**	**72.97**	**91.35**	**72.11**	**90.77**

Both Weight and Activation Quantization:

Model		8-bit		7-bit		6-bit		5-bit		4-bit		3-bit	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ResNet-18	TF-Lite	69.57	89.02	69.46	88.87	67.95	88.02	61.47	83.43	18.84	36.33	0.13	0.61
(69.76, 89.08)	Bit-split	**69.74**	**89.09**	**69.68**	**89.07**	**69.58**	**88.96**	**69.28**	**88.77**	**67.56**	**87.76**	**61.30**	**83.47**
ResNet-50	TF-Lite	**76.05**	**92.93**	75.75	92.70	73.83	91.66	65.46	86.34	10.40	22.36	0.11	0.54
(76.15, 92.87)	Bit-split	75.96	92.83	**76.09**	**92.84**	**75.90**	**92.75**	**75.38**	**92.59**	**73.71**	**91.62**	**66.22**	**87.18**
ResNet-101	TF-Lite	76.78	93.31	74.07	91.79	31.78	55.96	0.82	2.65	0.25	0.98	0.09	0.54
(77.47, 93.56)	Bit-split	**77.23**	**93.55**	**77.20**	**93.47**	**76.93**	**93.42**	**76.07**	**92.95**	**74.68**	**92.18**	**63.96**	**85.65**
VGG-16-BN	TF-Lite	73.31	91.53	72.94	91.25	70.65	89.77	54.45	78.18	3.41	10.17	0.18	0.78
(73.37, 91.50)	Bit-split	**73.43**	**91.54**	**73.43**	**91.55**	**73.34**	**91.45**	**72.89**	**91.22**	**71.14**	**90.29**	**66.11**	**86.92**

Comparison with State-of-the-arts

Table 4. Comparison results of different post-training quantization approaches. Bold values indicate the best results.

Model		Per-layer	Unified-precision	ResNet-18	ResNet-50	ResNet-101	VGG-16-BN
Full-precision		-	-	69.76	76.15	77.47	73.37
A8W4	TF-Lite (Krishnamoorthi, 2018)	√	√	55.5	70.1	72.6	66.4
	ACIQ (Banner et al., 2019)	×	√	67.4	74.8	76.3	71.7
	ACIQ-Mix (Banner et al., 2019)	×	×	68.3	75.3	**76.9**	72.4
	Bit-split	√	√	**69.1**	**75.6**	**76.9**	**73.0**
A4W4	TF-Lite (Krishnamoorthi, 2018)	√	√	18.8	10.4	0.3	3.4
	TensorRT (Migacz, 2017)	√	√	31.9	46.2	49.9	-
	LAPQ (Nahshan et al., 2019)	√	√	59.8	70.0	59.2	-
	ACIQ-Mix (Banner et al., 2019)	×	×	67.0	73.8	75.0	**71.8**
	Bit-split	√	√	67.6	73.7	74.7	71.1
	Bit-split-per-channel	×	√	**68.1**	**74.2**	**75.3**	**71.8**

Results on Detection and Instance segmentation

Table 5. Object detection (bounding box AP) and instance segmentation (mask AP) results on COCO minival set.

Model		$AP_{0.5:0.95}$	$AP_{0.5}$	$AP_{0.75}$
RetinaNet (Box)	Full-precision	30.7	49.1	32.4
	A8W4	30.1	48.2	31.8
	A6W4	30.2	48.2	31.9
	A4W4	29.6	47.6	31.0
Mask R-CNN (Box)	Full-precision	33.1	54.3	35.2
	A8W4	32.4	53.5	34.4
	A6W4	32.3	53.3	34.2
	A4W4	32.0	52.9	34.0
Mask R-CNN (Mask)	Full-precision	30.7	51.2	32.4
	A8W4	30.1	50.5	31.6
	A6W4	30.1	50.4	31.5
	A4W4	29.6	49.7	31.2

Thanks for your attention.

Codes are available at https://github.com/wps712/BitSplit

peisong.wang@nlpr.ia.ac.cn