Pattern Recognition 111 (2021) 107647

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Mixed-precision quantized neural networks with progressively decreasing bitwidth

Tianshu Chu[a], Qin Luo[a], Jie Yang[a,b,c], Xiaolin Huang[a,b,c,*]

[a] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China
[b] Institute of Medical Robotics, Shanghai Jiao Tong University, Shanghai 200240, China
[c] MOE Key Laboratory of System Control and Information Processing, Shanghai, 200240, China

ARTICLE INFO

Article history:
Received 2 November 2019
Revised 3 May 2020
Accepted 6 September 2020
Available online 24 September 2020

Keywords:
Model compression
Quantized neural networks
Mixed-precision

ABSTRACT

Efficient model inference is an important and practical issue in the deployment of deep neural networks on resource constraint platforms. Network quantization addresses this problem effectively by leveraging low-bit representation and arithmetic that could be conducted on dedicated embedded systems. In the previous works, the parameter bitwidth is set homogeneously and there is a trade-off between superior performance and aggressive compression. Actually, the stacked network layers, which are generally regarded as hierarchical feature extractors, contribute diversely to the overall performance. For a well-trained neural network, the feature distributions of different categories are organized gradually as the network propagates forward. Hence the capability requirement on the subsequent feature extractors is reduced. It indicates that the neurons in posterior layers could be assigned with lower bitwidth for quantized neural networks. Based on this observation, a simple yet effective mixed-precision quantized neural network with progressively decreasing bitwidth is proposed to improve the trade-off between accuracy and compression. Extensive experiments on typical network architectures and benchmark datasets demonstrate that the proposed method could achieve better or comparable results while reducing the memory space for quantized parameters by more than 25% in comparison with the homogeneous counterparts. In addition, the results also demonstrate that the higher-precision bottom layers could boost the 1-bit network performance appreciably due to a better preservation of the original image information while the lower-precision posterior layers contribute to the regularization of $k-$bit networks.

1. Introduction

The deep convolutional neural networks (CNNs) have achieved state-of-the-art results on computer vision tasks, such as image [1–5], object detection [6,7], and semantic segmentation [8–10]. These achievements depend on the complicated model that overfits the distribution of numerous training data. However, this also leads to over-parameterization and dramatic computation cost. A typical CNN often takes hundreds of MB memory space, i.e., 170MB for ResNet-101 [3], 250MB for AlexNet [1], 550MB for VGG-19 [2], and requires billions of FLOPs per image during inference that rely on powerful GPUs. This challenges the deployment of CNNs on the edge devices, such as mobile phones and drones. Thus the network compression and acceleration is an important issue in deep learning research and application.

Several techniques have been proposed to tackle this issue via compact neural architecture design [11,12], model pruning [13,14], and network quantization [15]. With the network topology unchanged, the quantization is able to reduce the model size greatly to only a fraction of the origin by utilizing low-precision representation of parameters [16]. Furthermore, the internal features could also be quantized. Then the model inference is accelerated significantly by converting the expensive floating-point arithmetic to the more effective fixed-point operations. Hence both the spatial and computational complexities are reduced notably by quantization.

Binary neural network (BNN) is a typical aggressive quantization method [17]. The model weights and activations are expressed as $\{-1, +1\}$ that could be stored by only 1-bit. Benefiting from the bitwise operations, the dot-product between binary weights and activations is replaced by XNOR and POPCOUNT arithmetics. Hence the deployment of BNN is no longer constrained by the GPUs. However, the naive BNN suffers from non-negligible performance degradation, especially on large-scale and complicated tasks [15]. Although some proposed techniques have alleviated the information loss through improved binarization scheme, network topology,

* Corresponding author at: Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail addresses: chutianshu@sjtu.edu.cn (T. Chu), tomqin@sjtu.edu.cn (Q. Luo), jieyang@sjtu.edu.cn (J. Yang), xiaolinhuang@sjtu.edu.cn (X. Huang).

https://doi.org/10.1016/j.patcog.2020.107647
0031-3203/

and training algorithm, there still exists nontrivial accuracy gap between BNN and the full-precision network [18–20]. Contemporarily, an effective method to boost the compact model performance is representing the model variables with fixed-point values, i.e., quantized neural network (QNN) [15]. As represented in [21,22], the QNNs are able to achieve comparable accuracy as the full-precision networks under the circumstance of 4-bit quantization. Nevertheless, larger bitwidth means the linear increase of model size and higher requirement on the hardware capacity. When the computing resources are extremely limited, it is necessary to make a trade-off between model accuracy and compression.

In this paper, we work on this trade-off issue by referring to mixed-precision approach. In fact, the network layers contribute diversely to the overall performance and each has different sensitivity to quantization. While the network propagating forward, the dissimilarity between hierarchical features is enhanced progressively. In the shallower layers, the internal features are distributed on complex manifolds. Accurate neurons are necessary to obtain the subsequent features. While in deeper layers, a rough convolutional filter is able to distinguish the previous local features as the deep semantic features are more separable. Hence the parameter precision could be designed flexibly based on the network structure and the distribution of hierarchical features. In this paper, a simple yet effective QNN with progressively decreasing bitwidth is proposed and the overview structure could be found in Fig. 1. The original information is preserved well by the high-precision bottom layers while the model size is compressed further due to the low-precision representation of top layers.

Our main contributions are:

1. Based on the observation on internal feature distributions and network structure, a mixed-precision QNN with progressively decreasing bitwidth is proposed.
2. Four typical classification CNNs, including VGG, AlexNet, and ResNet-18/20, and two object detection frameworks, SSD and Faster R-CNN, are quantized based on the proposed mixed-precision method. A heuristic of bitwidth assignment based on the quantitative separability for feature representation is given. The layer-wise bitwidth gradually reduces to 1-bit from 4-bit or 8-bit.
3. The re-designed QNNs are validated on several benchmark datasets, including CIFAR-10/100, ILSVRC-2012, and PASCAL VOC. The experimental results demonstrate that the mixed-precision networks could achieve preferable or very similar performance while requiring at least 25% less memory space for quantized parameters.

The rest of this paper is organized as follows. Section 2 provides a summary of related works. Based on the analysis on the feature distributions of different layers, a multi-level quantized structure with gradually decreasing bitwidth is proposed in Section 3. In Section 4, we demonstrate the effectiveness of the mixed precision framework via extensive experiments on several typical CNNs architectures and benchmark datasets. Section 5 ends this paper with some conclusions.

2. Related work

Network compression and acceleration is critical to the practical deployment of CNNs on edge devices. One kind of paradigm focuses on the network topology structure. Some researches focus on the design of compact neural architecture. Many lightweight networks are proposed, including ResNet [3], DenseNet [4], MobileNet [23], and ShuffleNet [24]. Besides, there exist some methods that search for an effective neural architecture via reinforcement learning [11,12,25]. Some other researches conduct model compression from the opposite direction. A tiny network is obtained via pruning and sparsity constraints on the basis of a well-trained complex network [13,26,27].

Network quantization addresses the compression and acceleration issue from the perspective of data format while preserving the network architecture. In [28], the results show that half-precision model is able to acquire promising accuracy. This indicates that the parameters could be stored by lower bitwidth and the model size is scaled down. Moreover, the intermediate variables could also be represented by discrete values. Then the computationally expensive floating-point arithmetics are replaced by the fixed-point and bitwise operations which are able to be conducted on the dedicated hardware. As shown in Fig. 2, both the full-precision and low-bit variables are preserved in the computation graph during the training phase. To make backward-propagation feasible, the gradients flow through the non-differentiable quantizer straightly, i.e., by **s**traight-**t**hrough gradient **e**stimator (STE) [15,29]. Some training characteristics and theoretical analysis are demonstrated in [19,30–32]. After training, the full-precision weights are removed during deployment.

BNN is an aggressive form of network quantization. The weights and activations are expressed as $\{-1, +1\}$ according to the signs. Thus the memory space required for each variable is reduce to only 1-bit and the model size after binarization is nearly 1/32 of the origin [33]. In addition, the inference efficiency is improved substantially by leveraging the XNOR and POPCOUNT operations [15]. However, the extreme compression leads to heavy information loss during binarization. There exists nontrivial accuracy gap between BNN and the full-precision counterpart, especially on complicated tasks. Some techniques are proposed to alleviate the performance loss via modified binarization scheme [18,20] and network architecture [34]. These improvements are limited with extra full-precision arithmetic introduced.

Another effective way to improve the model capability is assigning larger bitwidth to the network variables, i.e., conservative quantization [15]. A general and flexible quantization method is proposed in [35] and achieves promising accuracy on ILSVRC-2012 [22] and [36] improve the QNN performance further by adjusting the quantization step size during back-propagation. In the case of 4-bit quantization, the QNN could achieve comparable results as the full-precision counterpart. However, the increase of bitwidth means scale-up of the model size. There is a trade-off between superior performance and aggressive compression.

Among the methods mentioned above, all the model weights are treated equally and assigned with the same bitwidth. Actually, the parameters in the stacked neural network contribute differently to the overall results. It indicates that the parameter bitwidth should be determined by its individual function. Moreover, some advance chips are released, including Apple A12 Bionic and Nvidia Turing GPU, that support mixed-precision arithmetic. Hence some researches tackle the QNN trade-off issue via mixed-precision method. In [37] and [38], the bitwidth of each parameter is set according to the quantization residual of a pre-trained network. Wang [39] fine-tune the bitwidth via reinforcement learning. In this paper, we explore the layer-wise bitwidth from another perspective and propose a simple but effective mixed-precision framework. In comparison with the previous work, this proposed method is more flexible and compatible with various quantization schemes.

3. Methodology

3.1. Quantization function

As Fig. 2 shows, the discrete data flow through the stacked neural cells which consist of quantization, multiply-accumulation

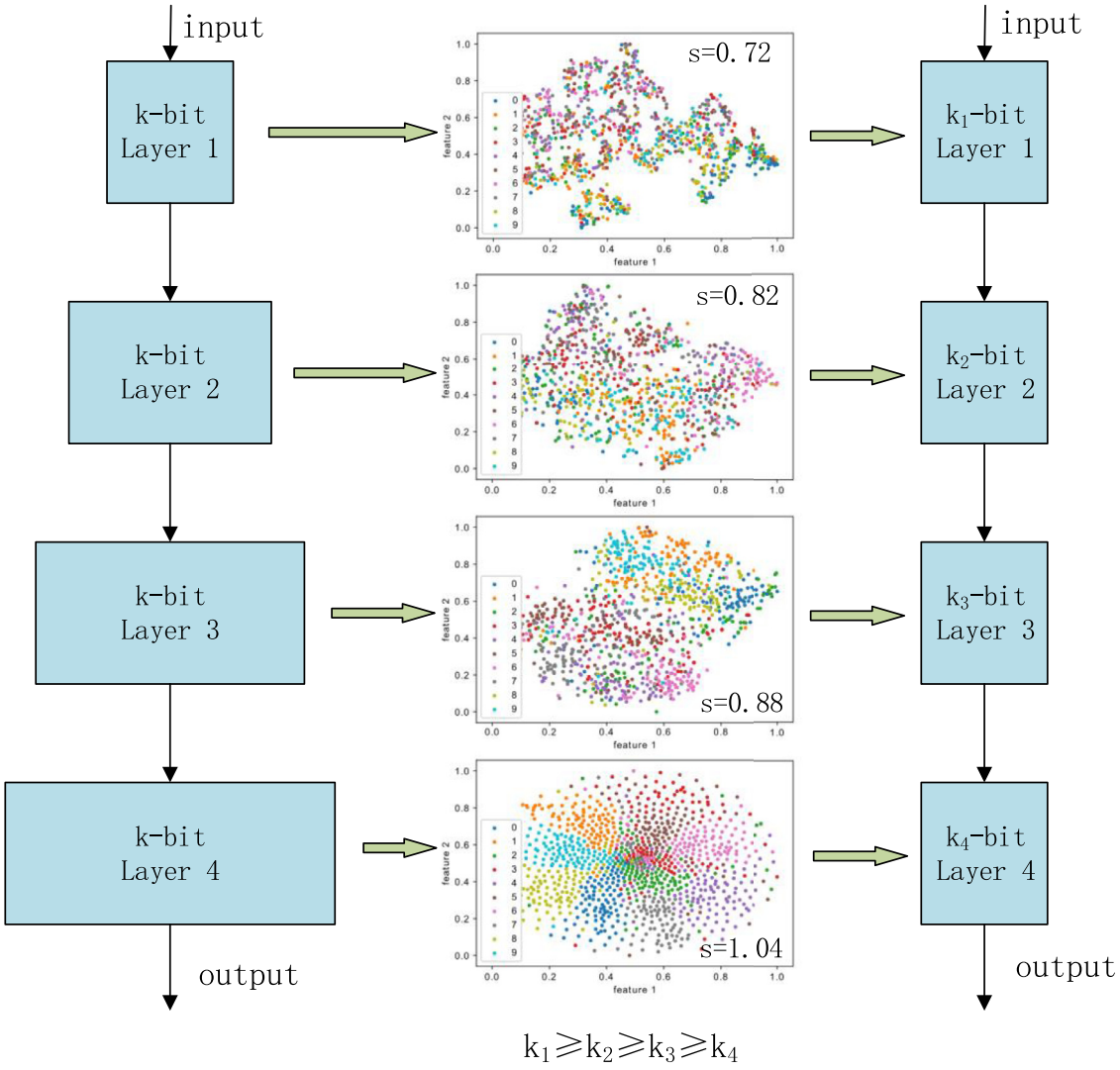

Fig. 1. The bitwidth settings of the k−bit homogeneous QNN (left) and the mixed-precision counterpart (right). The width of neural block will indicate the model size. The feature distributions after t-SNE transformation are depicted in the middle. In the initial layer, the quantitative separability s is low. Delicate neurons are required to distinguish the similar feature manifolds. As the network propagates forward, the feature distribution of the same category gathers gradually. In the deep hidden layers, a neuron with lower-precision parameters is able to extract robust feature.

(MAC), batch normalization, and activation. While the storage and computation cost is reduced notably, the information loss is inevitable due to quantization error during this process. An appropriate quantization module which is able to preserve the valuable information in the continuous variables is crucial for the network performance.

3.1.1. Binarization

An extreme quantization method is to store the discrete value by 1-bit, i.e., binarization. Given a variable $x \in \mathbf{R}^n$, the binary value x_b is determined by the sign. In order to enlarge the value range, a scaling factor $\frac{\|x\|_1}{n}$ is introduced. Then MAC is conducted by XNOR and POPCOUNT operations. However, the binarization function $B(\cdot)$ maps a continuous set $\mathbf{R}^n$ onto a discrete set $\{-1, +1\}^n$. The non-differentiability is an obstacle during the backward propagation and challenges the training of QNN. To address this issue, the STE is proposed to bypass the quantizer [15,29]. The forward and backward computations of binarization are shown as follows. $I(\cdot)$ is the indicator function. If the condition is satisfied, the indicator returns 1. Otherwise, it returns 0.

$$\textbf{Forward:} \quad x_b = B(x) = \frac{\|x\|_1}{n}\text{sign}(x),$$

$$\textbf{Backward:} \quad \frac{\partial B}{\partial x} \approx I(|x| < 1).$$

3.1.2. Quantization

The conservative quantization can improve the model capacity significantly by utilizing a larger bitwidth $k > 1$. A general linear function $Q(\cdot)$ is defined as

$$\textbf{Forward:} \quad x_q = Q(x) = \frac{1}{2^k - 1}\lfloor (2^k - 1)x \rceil,$$

$$\textbf{Backward:} \quad \frac{\partial Q}{\partial x} \approx 1,$$

where $x \in [0, 1]^n$ and $x_q \in [0, 1]^n$ denote the full-precision and quantized values, and $\lfloor \cdot \rceil$ represents the rounding operation. The STE gradient is utilized either in the backward of $Q(\cdot)$. With this function, the model weights and activations could be discretized after proper preprocessing as follows.

3.1.3. Weight quantization

For a continuous weight tensor $W \in \mathbf{R}^{m \times n}$, it is necessary to project the unbounded elements into the specified interval [0, 1].

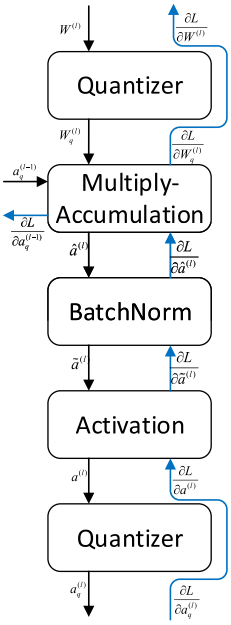

Fig. 2. The computation graph of a neural cell in QNN. The black arrows depict the forward data flow and the blue ones show the backward-propagation. Both the full-precision and quantized values are remained during training. The non-differentiable quantizer module is bypassed in the computation graph. After training, full-precision weights are discarded during deployment. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

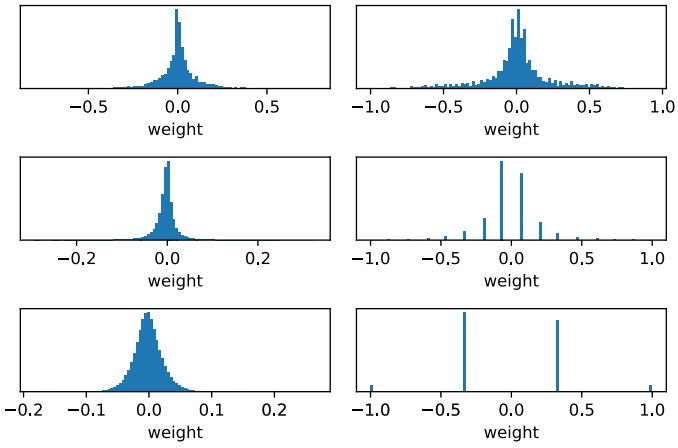

Fig. 3. The the histogram of network weight parameters. The first column depicts the distribution of weights from three different layers in well-trained full-precision network. The second column demonstrates the quantized weights from the according layers in a well-trained QNN. The images from top to bottom in the second column represent the 8-bit, 4-bit and 2-bit quantization results respectively.

The most straightforward normalization is scaling and shifting after dividing the largest absolute value. However, the majority of the continuous weight values distribute around the zero-point as Fig. 3 shows. The straightforward division would make the normalization dominant by the outliers and lead to additional round-off quantization error. Hence a non-linear transformation, the hyperbolic tangent function, is introduced to alleviate the impact of long-tail distribution. The saturation effect of $\tanh(\cdot)$ can suppress the variation of large values and avoid outliers during training. It is also worth noticing that the MAC operations are conducted channel-wise,

$$\hat{a}_i = W_i \cdot a, \quad W_i^T, a \in \mathbf{R}^n.$$

The MAC results are related to the weight values in the corresponding channels. Hence it is more suitable to do channel-wise normalization. The extra scaling factors can be merged into the batch normalization and no addition computation cost is introduced during deployment. Thus the overall quantization procedure for weights is as follows,

$$\hat{W} = \tanh(W),$$
$$M_i = \max_j(|\hat{W}_{ij}|),$$
$$W_{q,ij} = 2 \cdot Q\left(\frac{\hat{W}_{ij}}{2 \cdot M_i} + \frac{1}{2}\right) - 1.$$

3.1.4. Activation quantization

For the activation quantization, it is theoretically feasible to adopt the similar strategy as weight parameters. But the model efficiency will drop dramatically due the additional floating-point operations in preprocessing. Therefore a clamp function is usually applied as the activation function to confine the features to the specified interval [0, 1] before quantization.

$$a = \text{clamp}(\tilde{a}, 0, 1),$$
$$a_q = Q(a).$$

3.2. Hierarchical feature distribution and mixed-precision QNN

One of the important advantages that contribute to the remarkable achievements of deep neural networks is that a delicate feature representation could be learned automatically by end-to-end training. Based on the network topology structure, the hierarchical features are organized by the MAC operations and non-linear transformations layer-wise. As the network propagates forward, the variation of each categorical feature distribution is reduced gradually while the margins between each other increase. Consequently, the feature distributions are mapped from complex manifolds in high-dimension to several clusters in low-dimension and a linear classifier is able to achieve great accuracy by leveraging the final semantic features.

To illustrate the separability of hierarchical feature distribution quantitatively, the ratio between the inter-class distance and the inner-class distance is selected. Specifically for an image sample x_i, $x_i^{(l)}$ is the corresponding feature map after the lth layer of network. Due to that the convolution operation focuses on the local pattern and is conducted patch-wise, the average feature patch is extracted as the overall local representation,

$$z_i^{(l)} = \frac{1}{H_l \times W_l} \sum_{m=1}^{H_l \times W_l} x_{i,m}^{(l-1)},$$

where W_l and H_l is the width and height of $x_i^{(l)}$, respectively, and $x_{i,m}^{(l)}$ is the mth local patch of $x_i^{(l)}$. Given that $d_{ij}^{(l)} = \|z_i^{(l)} - z_j^{(l)}\|_2^2$ is the squared distance between the overall local representations $z_i^{(l)}$ and $z_j^{(l)}$, the feature separability of dataset $\{(x_i, y_i)\}_{i=1}^N$ could be measured by

$$s^{(l)} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\sum_{j:y^{(j)}=y^{(i)}} d_{ij}^{(l)} / \sum_j I(y^{(j)} = y^{(i)})}{\sum_{j:y^{(j)}\neq y^{(i)}} d_{ij}^{(l)} / \sum_j I(y^{(j)} \neq y^{(i)})} \right), \quad (1)$$

where $I(\cdot)$ is the indicator function.

During the forward-propagation process, the feature transformation is conducted by the neurons in each layer. Each individual neuron works as a simple classifier to extract target feature. The input complexities of the network layers differ from each other, which means that the precision requirements on the neurons are also different. Based on this observation, we argue that the neurons in the shallower layers are more sensitive to quantization. As

Table 1
Number of weight parameters in typical networks.

Layer	1	2	3	4	5	6	7
ResNet-20	432	13,824	51,200	204,800	-	-	-
ResNet-18	1728	147,456	524,288	2,097,152	8,388,608	-	-
VGG-7	3456	147,456	294,912	589,824	1,179,648	2,359,296	8,388,608
AlexNet	41,472	307,200	884,736	663,552	442,368	37,748,736	16,777,216

the feature distributions overlap mutually, finite neurons are unable to distinguish the samples and extract meaningful intermediate representations without suitable precision. Once the advanced features are obtained explicitly, the following layers become more robust to the quantization error. Thus it is feasible to design the QNN structure more flexibly rather than $k-$bit homogeneous networks. The general bitwidth setting for the model progressively decreases from the initial k-bit as the QNN propagates forward. The quantitative separability of features of each layer could be measured by (1), which also serves as a hint to determine the used bitwidth for each layer.

It is worth noticing that the majority of model parameters are concentrated in the deeper layers as Table 1 shows. The rise of bitwidth at the bottom layers has little effect on the model size in comparison with low-bit network. But the original information would be preserved better. On the other hand, the model size of mixed-precision QNN is much smaller than the $k-$bit homogeneous one due to lower parameter precision. Hence the mixed-precision QNN is more compact and has the potential to achieve promising performance.

We could implement the proposed mixed-precision framework with progressively decreasing bitwidth in many neural networks. Here we discuss four typical CNNs, including VGG-7[1], AlexNet, ResNet-20, and ResNet-18, which will also be validated numerically, to show the specific settings of bitwidth. The suggested bitwidth k of each layer is determined by the input separability heuristically as,

$$k = \begin{cases} 4 \times t, & \text{if } s \leq 0.8; \\ 2 \times t, & \text{if } 0.8 < s \leq 0.85; \\ 1 \times t, & \text{if } s > 0.85. \end{cases} \tag{2}$$

According to the performance and memory requirements, the suggested bitwidth setting could be scaled generally by adjusting the integer t. In addition, there exists randomness during separability calculation due to dataset sampling. One can also trim the original bitwidth individually based on the value of s.

VGG-Net and AlexNet are the representatives of plain CNNs. The VGG-7 in this paper is designed for CIFAR-10/100 dataset. All the weight parameters are quantized except that of the output layer as the linear classifier is related to the final results directly and requires enough precision. 1000 CIFAR-10 samples, 100 per class, are selected randomly to calculate the feature separability. According the bitwidth rule (2), the suggested weight bitwidth setting is (4-4-2-2-1-1/1). Based on this setting, a modified bitwidth combination that decreases from 8-bit to 1-bit layer-wise with a factor 1/2 as shown in Table 2 is also validated in this paper. Although the initial bitwidth is higher than the homogeneous counterpart, the average model bitwidth is reduced to 1.10 and 1.06 respectively. AlexNet, which contains 5 convolution and 2 latent fully-connected layers, is proposed for the high-resolution image recognition task ILSVRC-2012 [1]. The input and output layers are maintained full-precision as [18,35] for a fair comparison. As the ILSVRC-2012 dataset contains 1000 categories of samples, it is unaffordable to obtain the complete distance matrix $\{d_{ij}\}$. Hence 10 categories and 100 samples per-class are sampled to approach the distance matrix. The suggested bitwidth setting is shown in Table 4. The overall average bitwidth is 1.10.

[1] VGG-7 architecture: 2 × (128-Conv3 × 3) + MP2 + 2 × (256-Conv3 × 3) + MP2 + 2 × (512-Conv3 × 3) + MP2 + 1024-FC + Output-FC.

Table 2
CIFAR-10 Experimental results.

Model	Method	k_w	k_a	Test Acc. %
ResNet-20	FP [3]	32	32	91.60
	DoReFa [35]	2	2	88.20
		4	4	90.50
	Ours/heuristic ($t = 1$)	1.91(4-1-2)	2	88.22
	Ours/manual	1.34 (4-2-1)	2	88.33
	Ours/heuristic ($t = 2$)	3.82(8-2-4)	4	89.56
	Ours/manual	2.68(8-4-2)	4	90.54
VGG-7	FP	32	32	92.48
	BNN [33]	1	1	89.85
	HWGQ [41]	1	2	92.51
	DoReFa [35]	1	2	92.33
		2	2	92.83
	Ours/heuristic ($t = 1$)	1.10(4-4-2-2-1-1/1)	2	93.21
	Ours/ manual	1.06 (8-4-2-1-1-1/1)	2	93.22

ResNet is the pioneer of networks with shortcuts. The ResNet-20, which consists of 3 residual stages, is initially proposed for the CIFAR-10 task [3]. For a fair comparison with related work [18,35], the weight bitwidths of residual stages are determined by s as (4-1-2) and modified to (4-2-1) as shown in Table 2. As ResNet-20 has only 64 filters at the final stage, it is uncertain that the 64-dim pooling features obtained by aggressively quantized neurons could satisfy the classification requirement, especially for CIFAR-100 task. The doubled bitwidth models with more powerful capacity are also validated in this paper. By contrast, ResNet-18, containing 4 residual stages, is much wider and has 512 filters at the final residual stage. The suggested bitwidth according to s reduces from 8-bit to 1-bit, which is shown in Table 4. The activation bitwidths of the mixed-precision networks are set the same with the homogeneous counterparts to maintain comparable representation capability.

Beyond classification, the proposed mixed-precision strategy could also be used for other tasks, e.g., object detection, which is a much more complicated task. In addition to predict categories of

Table 3
CIFAR-100 Experimental results.

Model	Method	k_w	k_a	Test Acc. %
ResNet-20	FP	32	32	66.29
	DoReFa [35]	2	2	60.42
		4	4	63.86
	Ours/heuristic ($t = 1$)	1.91(4-1-2)	2	61.57
	Ours/manual	1.34(4-2-1)	2	57.82
	Ours/heuristic ($t = 2$)	3.82(8-2-4)	4	64.28
	Ours/manual	2.68(8-4-2)	4	63.36
VGG-7	FP	32	32	72.03
	XNOR [18]	1	1	57.74
	DoReFa [35]	1	2	69.64
		2	2	71.44
	Ours/heuristic ($t = 1$)	1.10(4-4-2-2-1-1/1)	2	70.42
	Ours/manual	1.06(8-4-2-1-1-1/1)	2	71.53

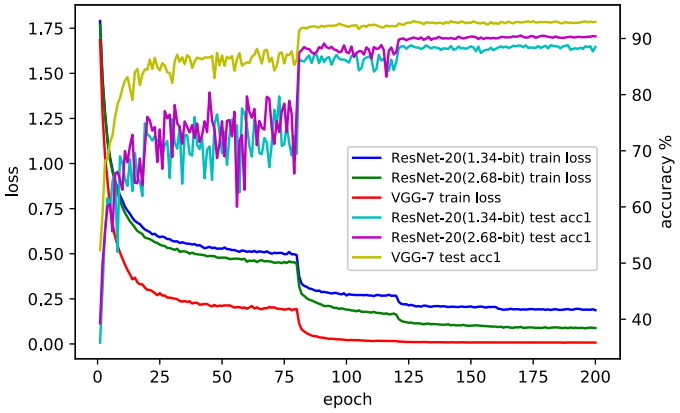

Fig. 4. The training curve of ResNet-20 and VGG-7 on CIFAR-10. The training of VGG-7 is more stable and enjoys larger learning rate due to network redundancy. On the contrary, ResNet-20 is a compact network with shortcuts. The training process is stabilized by lower learning rate.

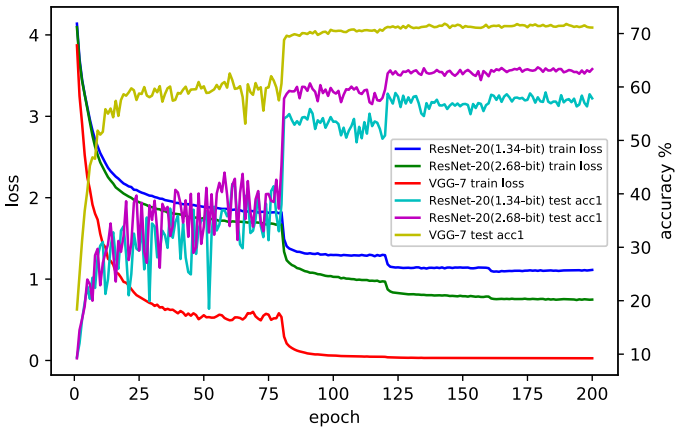

Fig. 5. The training curve of ResNet-20 and VGG-7 on CIFAR-100. The higher model redundancy makes the VGG-7 also competent for the complicated task. On the contrary, the performance of ResNet-20, which is originally designed for CIFAR-10, degrades significantly on CIFAR-100 due to constrained model capacity. In this case, the higher precision could boost the model accuracy notably.

multiple objects in an image, the detector also needs to regress the coordinates of bounding boxes. This requires greater feature extracting capability of the network. To investigate the performance of mixed-precision QNN on object detection task, a VGG-16 based single shot detector (SSD) [40] and a ResNet-50 with feature pyramid network based Faster R-CNN [7] are quantized and validated in this paper. The weight parameters of VGG-16 backbone are discretized utilizing the similar bitwidth setting as VGG-7. To improve the feature extraction capability at the final stage, the bitwidth of extra layers is set to 4-bit. The output layers remain full-precision. The final average bitwidth is 1.42. For the Faster R-CNN, the weight bitwidth of ResNet-50 backbone is set to (8-4-2-1) for the residual stages. The final average bitwidth is 1.52.

4. Experiments

To validate the performance of QNN with progressively decreasing bitwidth, we conduct extensive experiments on CIFAR-10/100, ILSVRC-2012, and Pascal VOC datasets. The training codes of the mentioned classification and object detection neural networks will be available on-line.[2]

[2] https://github.com/ariescts/mp-qnn.

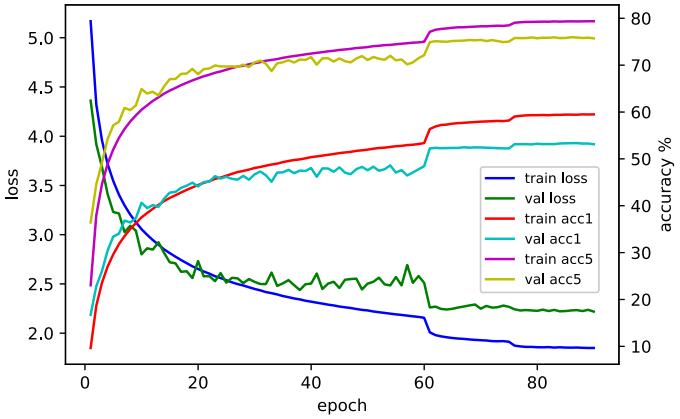

Fig. 6. The training curve of AlexNet. Although the model weights are compressed to 1.1-bit aggressively, the training process converges effectively and obtain competitive final results.

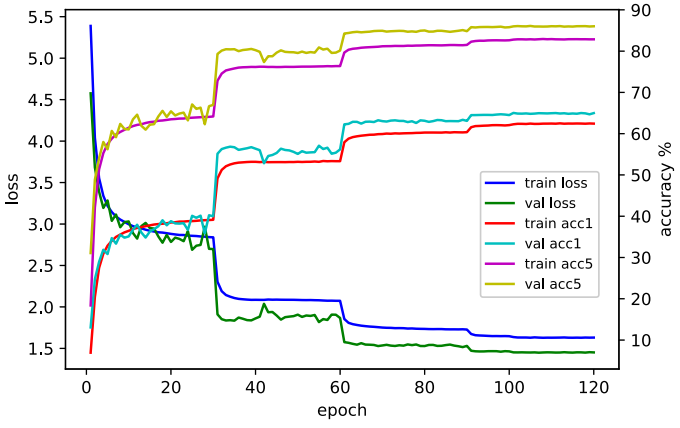

Fig. 7. The training curve of ResNet-18. The generalization fo mixed-precision ResNet-18 is better than AlexNet and prefers multi-stage learning rate scheduler.

Table 4
ILSVRC-2012 Experimental results.

Model	Method	k_w	k_a	Top1 Acc%	Top5 Acc%
AlexNet	FP	32	32	56.60	80.20
	XNOR [18]	1	1	44.20	69.20
	DoReFa [35]	1	2	47.70	-
	Ours	1.10 (8-4-2-1/1-1)	2	53.18	75.90
ResNet-18	FP	32	32	69.30	89.20
	XNOR [18]	1	1	51.20	73.20
	Bi-Real [34]	1	1	56.40	79.50
	DoReFa [35]	2	2	62.60	84.40
	PACT [22]	2	2	64.40	85.60
	Ours	1.42 (8-4-2-1)	2	65.03	86.00

4.1. CIFAR-10/100

There are 10 classes of 50,000 training images and 10,000 test ones in CIFAR-10 dataset. The image size is 32 × 32 pixels. The CIFAR-100 dataset consists of the same number of images from 100 categories. One tenth of training samples are selected as validation set.

We follow the data augmentation in [3] for training. At test time, the original images are sampled directly. We use SGD optimizer with momentum of 0.9 and learning rate starting from 0.1 and scaled by 0.1 at epoch 80, 120, 160. L2-regularizer with decay of 2e-4 is applied to weight parameters. The mini-batch size is 128. After 200 epochs of training from scratch, the test accuracy

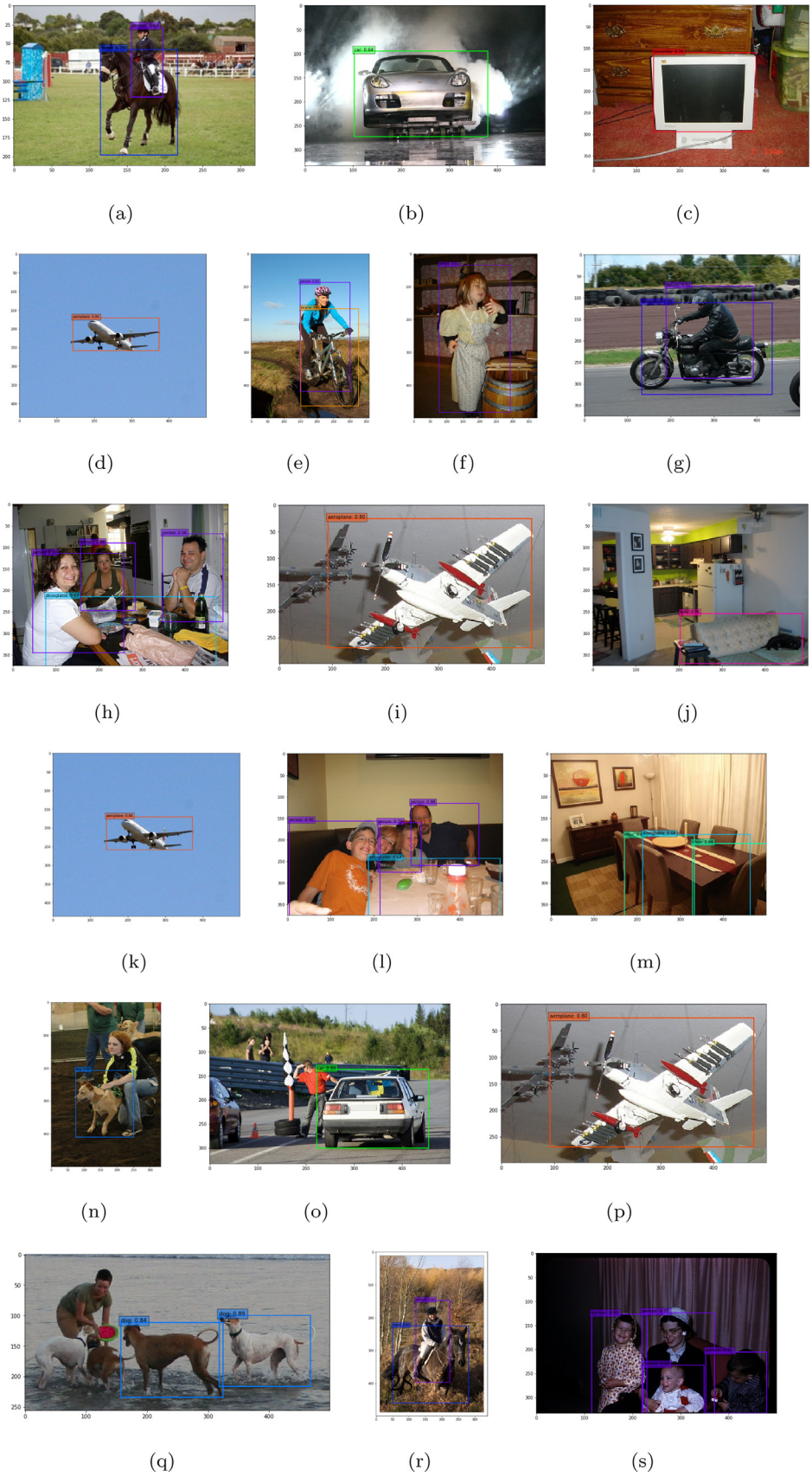

Fig. 8. The sampled detection results of the mixed-precision SSD. The quantized detector is able to locate the distinct and significant objects precisely while neglecting the ones which are overlapped or located at the boundary of images.

Table 5
Pascal VOC Experimental results.

Model	Method	kw	ka	map	aero table	bike dog	bird horse	boat mbike	bottle person	bus plant	car sheep	cat sofa	chair train	cow tv
SSD-300	FP	32	32	75.10	76.92	82.08	74.83	68.10	47.80	83.23	83.99	88.26	56.65	79.88
					74.56	85.80	84.96	81.48	76.39	43.40	73.88	77.08	87.57	75.14
	Dorefa	2	2	60.66	67.06	73.52	47.77	50.39	22.89	70.48	78.28	72.27	41.93	57.04
					63.61	66.49	75.54	74.63	68.57	25.59	58.22	63.48	75.61	59.83
	Ours	1.22	2	62.21	70.96	76.08	51.64	54.97	25.33	72.37	78.79	74.07	44.30	56.27
					62.43	66.91	78.71	73.82	69.57	27.19	58.90	64.00	77.20	60.69
Faster R-CNN	FP	32	32	77.32	81.00	84.44	76.60	65.18	69.72	83.44	88.00	86.97	61.82	75.82
					73.22	82.71	84.51	83.51	84.38	55.54	80.82	72.91	81.58	74.23
	Dorefa	2	4	74.06	79.23	81.99	70.64	63.99	66.71	79.97	86.65	81.58	59.53	67.32
					70.82	75.64	81.06	82.28	83.60	49.75	74.69	70.40	80.40	74.95
	Ours	1.52	4	74.52	79.77	83.58	72.57	62.34	67.48	81.14	87.14	81.33	58.73	70.40
					69.86	75.94	81.37	80.90	84.02	52.26	75.74	71.34	78.75	75.84

associated with the best validation performance is reported as the final result.

After 5 runs of each experiment, the average test accuracies of CIFAR-10 are recorded in Table 2. Here, FP and 32-bit denote the full-precision network with floating-point parameters. As the analysis in Section 3.2, the mixed-precision networks obtain higher accuracies than the homogeneous counterparts while the model size is smaller. For ResNet-20, the suggested bitwidth settings and the modified ones achieve very similar results. Due to the incorporation with manual fine-tune, the model size under modified setting is smaller. And the corresponding mixed-precision network with less than 3-bit for weights 4-bit for activations is able to achieve comparable final result as the full-precision network. However, at the beginning of training process, the generalization ability of mixed-precision QNN fluctuates obviously as Fig. 4 shows. This is due to that the quantized values change back and forth due to a large learning rate. When the learning rate decays, the training process is stabilized. In addition, the mixed-precision VGG-Nets obtain better result than both the 2-bit and even the full-precision one. And the manual setting outperforms the suggested one slightly. We argue that the better information preservation in the initial layers due to higher bitwidth boosts the performance evidently. Meanwhile, the VGG-7 is a very "wide" network. The redundancy stabilizes the training process as Fig. 4 shows. But once sufficient and meaningful information is obtained by the bottom layer, the redundant parameters in the subsequent layers may lead to overfitting. Hence, the suitable bitwidth setting contributes to the model regularization.

The results on CIFAR-100 dataset are recorded in Table 3 and Fig. 5. The performance is consistent with that of CIFAR-10 generally. For ResNet-20, the suggested settings achieve higher accuracy than the manual ones due to greater model capacity. It is noticeable that our modified ResNet-20 result at the fifth line is 3% lower than the homogeneous bitwidth network. The reason is that ResNet-20 is a very "narrow" network that is originally designed for CIFAR-10. After the average pooling layer, the dimension of semantic feature, 64, is less than that number of classes. Hence the 1-bit neurons in deep layers would induce significant information loss. Once the final or the overall bitwidth increases, the performance bottleneck is broken. While for the wider network, VGG-Net, it is unnecessary to worry about that. The numerous 1-bit neurons in deep layer guarantee meaningful semantic features. In comparison with the 2-bit network, the mixed-precision model is able to compress memory space for quantized parameters to nearly a half while achieving very competitive accuracy. In addition, the manual re-designed quantized network outperforms the suggested one by 1% due to better information preservation at the bottom layer.

4.2. ILSVRC-2012

ILSVRC-2012 is a 1000-category dataset which consists of 1.2 million training images and 50 thousands of validation ones. Compared to the CIFAR task, ILSVRC is much more challenging due to larger and more diverse images. For training, the images are resized to 256 $\times$ 256 and cropped randomly to 224 $\times$ 224. For validation, the center crops are used as inputs.

In the training process, an Adam optimizer with initial learning rate of 2e-4 and no weight-decay is applied to AlexNet. For ResNet-18, we take an SGD optimizer with an initial learning rate of 0.1 and weight-decay of 1e-4. The learning rate is scaled by 0.1 at the 60th and 75th of the 90 total epochs and at the 30th, 60th, 90th and 100th of 120 total epochs respectively. After training, the Top-1 and Top-5 validation accuracies are reported in Table 4. The training process is illustrated in Fig. 6 and Fig. 7, which correspond to AlexNet and ResNet-18, respectively. It is clear that the mixed-precision QNNs have advantages over the ordinary ones in terms of both performance and model size. In comparison with the full-precision networks, the results are still acceptable.

4.3. Pascal VOC

Pascal VOC is a benchmark dataset for object detection, which consists of 20 categories of objects in general. To validate the performance of the proposed method on more challenging tasks, we select SSD and Faster R-CNN as baseline detectors and train our models on VOC2007 trainval and VOC2012 trainval datasets (16,551 images) after quantizing the backbone network with mixed-precision. Then the resulted model is evaluated on the VOC2007 test dataset (4,952 images). For SSD, an SGD optimizer with weight-decay of 1e-4 is applied for 8000 iterations of training. The learning rate 1e-3 is used for the first 4000 iterations and then continue training for 2000 iterations with 1e-4 and 1e-5. For Faster R-CNN, an SGD optimizer with weight-decay of 1e-4 is applied for 10 epochs of training. The learning rate starts from 0.01 and is divided by 10 at the 4th, 6th and 8th epoch.

The comparison results are illustrated in Table 5. The mixed-precision networks still outperform the homogeneous ones. The 62.21% mAP and detailed AP results of SSD demonstrate that the mixed-precision one-stage detector has the fundamental capability to detect obvious objects which are significant enough and located at the center of images, as demonstrated in Fig. 8. But compared to the full-precision counterpart, the performance of the quantized networks degrade notably with aggressive quantization bitwidth. This is due to that the detection task is much more challenging. The quantization error makes it difficult to predict the object location directly.

By introducing the region proposal network and the feature pyramid network, the performance of two-stage detector, Faster R-CNN with quantized backbone, is improved evidently in comparison with SSD. There are marginal gaps between the precision of quantized detectors and that of full-precision ones. In addition, the mixed-precision detector achieves better performance than the homogeneous one while taking less memory space.

5. Conclusions

In this paper, a novel QNN framework with multiple bitwidths is proposed. Based on the observation of layer-wise hierarchical feature distributions and network structure, we propose a quantitative separability of feature representation and a progressively decreasing bitwidth setting to address the trade-off issue between aggressive compression and excellent performance.

Extensive experiments on typical CNNs and benchmark datasets demonstrate the effectiveness of our method. For image categorization, the re-designed mixed-precision QNN could save at least 25% memory space for quantized parameters while achieving preferable performance in comparison with the k−bit homogeneous counterparts. Specifically, the low bitwidth in the deep layers contributes to model regularization apart from compression for the redundant networks like VGG-7. For the compact network on complex tasks, the model performance is boosted significantly due to better preservation of original image information via higher bitwidth in the shallower layers.

Object detection is a much more sophisticated task than categorization and has higher requirements on the hierarchical feature map. While the quantized one-stage detectors degrade notably in comparison with the full-precision counterpart, the mixed-precision network still outperforms the bitwidth homogeneous one in both model size and precision. Meanwhile, the performance of two-stage detector with quantized backbone is improved evidently. And the mixed-precision method is able to reduce the memory space remarkably while maintaining excellent accuracy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was partially supported by National Key Research Development Project (No. 2018AAA0100702), National Natural Science Foundation of China (Nos. 61977046, 61876107, U1803261) and Committee of Science of Technology, Shanghai, China (No. 19510711200).

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.
[3] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[4] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[5] S. Matiz, K.E. Barner, Inductive conformal predictor for convolutional neural networks: applications to active learning for image classification, Pattern Recognit. 90 (2019) 172–182.
[6] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[7] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[8] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[9] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[10] J. Wei, Y. Xia, Y. Zhang, M3Net: a multi-model, multi-size, and multi-view deep neural network for brain magnetic resonance image segmentation, Pattern Recognit. 91 (2019) 366–378.
[11] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, Learning transferable architectures for scalable image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8697–8710.
[12] J. Chang, Y. Guo, G. Meng, S. Xiang, C. Pan, et al., Data: differentiable architecture approximation, in: Advances in Neural Information Processing Systems, 2019, pp. 876–886.
[13] J. Frankle, M. Carbin, The lottery ticket hypothesis: finding sparse, trainable neural networks, in: International Conference on Learning Representations, 2018.
[14] Z. Liu, H. Mu, X. Zhang, Z. Guo, X. Yang, K.-T. Cheng, J. Sun, MetaPruning: meta learning for automatic neural network channel pruning, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3296–3305.
[15] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, Y. Bengio, Quantized neural networks: training neural networks with low precision weights and activations, J. Mach. Learn. Res. 18 (1) (2017) 6869–6898.
[16] M. Courbariaux, Y. Bengio, J.-P. David, BinaryConnect: training deep neural networks with binary weights during propagations, in: Advances in Neural Information Processing Systems, 2015, pp. 3123–3131.
[17] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, N. Sebe, Binary neural networks: a survey, Pattern Recognit. (2020) 107281.
[18] M. Rastegari, V. Ordonez, J. Redmon, A. Farhadi, XNOR-Net: ImageNet classification using binary convolutional neural networks, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 525–542.
[19] J. Bethge, M. Bornstein, A. Loy, H. Yang, C. Meinel, Training competitive binary neural networks from scratch, arXiv preprint arXiv:1812.01965 (2018).
[20] L. Hou, Q. Yao, J.T. Kwok, Loss-aware binarization of deep networks, in: International Conference on Learning Representations, 2017.
[21] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, D. Kalenichenko, Quantization and training of neural networks for efficient integer-arithmetic-only inference, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2704–2713.
[22] J. Choi, Z. Wang, S. Venkataramani, P.I.-J. Chuang, V. Srinivasan, K. Gopalakrishnan, PACT: parameterized clipping activation for quantized neural networks, arXiv preprint arXiv:1805.06085 (2018).
[23] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).
[24] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: an extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.
[25] M. Tan, Q. Le, EfficientNet: rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, 2019, pp. 6105–6114.
[26] P. Molchanov, S. Tyree, T. Karras, T. Aila, J. Kautz, Pruning convolutional neural networks for resource efficient inference, in: International Conference on Learning Representations, 2017.
[27] R. Yu, A. Li, C.-F. Chen, J.-H. Lai, V.I. Morariu, X. Han, M. Gao, C.-Y. Lin, L.S. Davis, NISP: pruning networks using neuron importance score propagation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9194–9203.
[28] S. Gupta, A. Agrawal, K. Gopalakrishnan, P. Narayanan, Deep learning with limited numerical precision, in: International Conference on Machine Learning, 2015, pp. 1737–1746.
[29] Y. Bengio, N. Léonard, A. Courville, Estimating or propagating gradients through stochastic neurons for conditional computation, arXiv preprint arXiv:1308.3432 (2013).
[30] W. Tang, G. Hua, L. Wang, How to train a compact binary neural network with high accuracy? in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2625–2631.
[31] M. Alizadeh, J. Fernández-Marqués, N.D. Lane, Y. Gal, An empirical study of binary neural networks' optimisation, in: International Conference on Learning Representations, 2018.
[32] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, J. Xin, Understanding straight-through estimator in training activation quantized neural nets, in: International Conference on Learning Representations, 2019.
[33] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, Y. Bengio, Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or -1, arXiv preprint arXiv:1602.02830 (2016).
[34] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, K.-T. Cheng, Bi-Real Net: enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 722–737.
[35] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, Y. Zou, DoReFa-Net: training low bitwidth convolutional neural networks with low bitwidth gradients, arXiv preprint arXiv:1606.06160 (2016).

[36] S.K. Esser, J.L. McKinstry, D. Bablani, R. Appuswamy, D.S. Modha, Learned step size quantization, in: International Conference on Learning Representations, 2019.
[37] J. Fromm, S. Patel, M. Philipose, Heterogeneous bitwidth binarization in convolutional neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 4006–4015.
[38] Y. Zhou, S.-M. Moosavi-Dezfooli, N.-M. Cheung, P. Frossard, Adaptive quantization for deep neural network, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[39] K. Wang, Z. Liu, Y. Lin, J. Lin, S. Han, HAQ: hardware-aware automated quantization with mixed precision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8612–8620.
[40] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[41] Z. Cai, X. He, J. Sun, N. Vasconcelos, Deep learning with low precision by half-wave gaussian quantization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5918–5926.

Tianshu Chu received the M.S. degree in control science and engineering from Northeastern University, Shenyang, China, in 2013. He is currently pursuing the Ph.D. degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China. His research interests include machine learning, optimization and network compression.

Qin Luo received the B.S. degree in instrument science and engineering from Shanghai Jiao Tong University, China, in 2019. He is currently pursuing the Master degree at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, China. His research interests include kernel method, optimization and neural network compression.

Jie Yang received his Ph.D. from the Department of Computer Science, Hamburg University, Hamberg, Germany, in 1994. Currently, he is a professor at the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. He has led many research projects (e.g., National Science Foundation, 863 National High Technique Plan), has one book published in Germany, and authored more than 300 journal papers. His major research interests are object detection and recognition, data fusion and data mining, and medical image processing.

Xiaolin Huang received the B.S. degree in control and engineering, and the B.S. degree in applied mathematics from Xin Jiaotong University, Xin, China in 2006. In 2012, he received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China. From 2012 to 2015, he worked as a postdoctoral researcher in ESAT-STADIUS, KU Leuven, Leuven, Belgium. After that, he was selected as an Alexander von Humboldt Fellow and working in Pattern Recognition Lab, the Friedrich-Alexander-Universitat Erlangen-Nurnberg, Erlangen, Germany. From 2016, he has been an Associate Professor at Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. In 2017, he was awarded by 1000-Talent Plan (Young Program). His current research areas include machine learning and optimization.