

Training Quantized Neural Networks with a Full-precision Auxiliary Module

Bohan Zhuang¹ Lingqiao Liu¹ Mingkui Tan² Chunhua Shen¹ Ian Reid¹
¹The University of Adelaide ²South China University of Technology

Abstract

In this paper, we seek to tackle a challenge in training low-precision networks: the notorious difficulty in propagating gradient through a low-precision network due to the non-differentiable quantization function. We propose a solution by training the low-precision network with a full-precision auxiliary module. Specifically, during training, we construct a mix-precision network by augmenting the original low-precision network with the full precision auxiliary module. Then the augmented mix-precision network and the low-precision network are jointly optimized. This strategy creates additional full-precision routes to update the parameters of the low-precision model, thus making the gradient back-propagates more easily. At the inference time, we discard the auxiliary module without introducing any computational complexity to the low-precision network. We evaluate the proposed method on image classification and object detection over various quantization approaches and show consistent performance increase. In particular, we achieve near lossless performance to the full-precision model by using a 4-bit detector, which is of great practical value.

1. Introduction

Deep neural networks (DNNs) have made great strides in many computer vision tasks such as image classification [11, 20], segmentation [8, 10] and detection [37, 40]. Even though deep and/or wide models can achieve promising accuracy, their huge computational complexity makes them incompatible with energy constrained devices which usually have limited memory bandwidth and computational power. This has motivated the community to design energy-efficient models, often based on quantized precision, aiming not to sacrifice accuracy relative to the full-precision models. In this paper, we propose to improve the training of the low-precision networks.

The core challenge for quantization is the non-differentiability of the discrete quantizer. As a result, we cannot directly optimize the discretised network with

stochastic gradient descent. And the current solutions can be divided into two categories. The first category is to employ a surrogate of gradient. The most commonly used approach is the straight-through estimator (STE) [2]. Some recent works have been proposed to relax the discrete quantizer to be continuous for gradient-based optimization [1, 31]. Even though the discontinuity of the discretization operation during training can be partly solved by smoothing it appropriately, some important information may still be missed due to the approximation which can lead to an undesirable drop in accuracy. The second category is to seeking guidance from a full-precision model for discretised network training. For example, new training strategies such as knowledge distillation [32, 56, 57] have been proposed to learn a low-precision student network by distilling knowledge from a full-precision teacher network.

Our method falls into the second category and our approach is based on the idea of sharing parameters between a mixed-precision (partially fully-precision) model and a low-precision model. Specifically, our method constructs a full-precision auxiliary module which connects to multiple layers of a low-precision model (see Fig. 1). During training, the low-precision network and the full-precision auxiliary module will be combined to form an augmented mixed-precision network. Then, the mixed-precision network and the low-precision model are jointly optimized. Since the parameters from the low-precision model are shared, they can receive gradient from both full-precision connections and low-precision connections. Consequently, the parameters of low-precision model can be updated via two routes, and this can overcome the gradient propagation difficulty due to the discontinuity of the quantizer. Note that only the low-precision network will be utilized for inference and therefore there is no additional complexity introduced at the test stage.

In addition to image classification, we further extend the proposed approach to building quantized networks for object detection. Building low-precision networks for object detection is more challenging since detection needs the network outputs richer information, such as locations of bounding boxes. There has been several works in literature to address the quantized object detectors [16, 22, 48]. How-

Corresponding author, e-mail: chunhua.shen@adelaide.edu.au

ever, there still exists a significant performance drop of 4-bit or lower-precision quantized detectors comparing to their full-precision counterpart. We apply our techniques to train a 4-bit RetinaNet [25] detector and further propose a modification to RetinaNet to better accommodate the quantization design. Through extensive experiments on the COCO benchmark, we show that our 4-bit models can achieve near lossless performance comparing to the full-precision model, which has a significant value in practice.

Our contributions can be summarized as follows:

- We propose a new training method to account for the non-differentiability of the quantization operator in a low-precision network. Our method can lead to more accurate low-precision model without increasing the model complexity at the testing stage.
- We apply our learning approach and propose a new design modification to build a 4-bit quantized object detection which achieves comparable performance to its full-precision counterpart.

1.1. Related work

Network quantization. Quantized network represents the weights and activations with very low precision, thus yielding highly compact DNN models compared to their floating-point counterparts. Moreover, the convolution operations can be efficiently computed via bitwise operations. Quantization can be categorized into fixed-point quantization and binary neural networks (BNNs), in which fixed-point quantization can also be divided into uniform and non-uniform. Uniform approaches [17, 54, 57] design quantizers with a constant quantization step. To reduce the quantization error, non-uniform strategies [4, 52] propose to learn the quantization intervals by jointly optimizing parameters and quantizers. A fundamental problem of quantization is to approximate gradient of the non-differentiable quantizer. To solve this problem, some works have studied relaxed quantization [1, 31, 47, 57]. Moreover, with the popularity of automatic machine learning, some recent literature employs reinforcement learning to search for the optimal bitwidth for each layer [6, 46, 49]. BNNs [14, 36] constrain both weights and activations to binary values (*i.e.*, +1 or -1), which brings great benefits to specialized hardware devices. The development of BNNs can be classified into two categories: (i) a focus on improving the training of BNNs [13, 30, 36, 45]; (ii) multiple binarizations to approximate the full-precision tensor or structure [9, 23, 27, 28, 45, 58]. In this paper, we propose a general auxiliary learning approach that can work on all categories of quantization approaches.

Weight sharing. Weight sharing has been attracting increasing attention for efficient, yet accurate computation. In visual recognition, region proposal networks (RPN) in Faster-RCNN [40] and Mask-RCNN [10] share the same

backbone with task-specific networks, which greatly saves testing time. For neural architecture search, ENAS [35] allows parameters to be shared among all architectures in the search space, which saves orders of magnitude GPU hours. In the network compression field, weight/activation quantization intends to partition the weight/activation distribution into clusters and use the centers of clusters as the possible discrete values. This strategy can be interpreted as a special case of weight sharing. Different from these approaches, we propose to utilize weight sharing for jointly optimizing the full-precision auxiliary module and the original low-precision network to improve the accuracy of the latter quantized model.

Auxiliary supervision. One straightforward way of adding auxiliary supervision is introducing additional losses into intermediate layers, which serves to combat the vanishing gradient problem while providing regularization. The effectiveness of additional losses has been demonstrated in some literature, like GoogLeNet [43], DSN [21], semantic segmentation [33, 53], etc. However, these methods are usually sensitive to the positions and scales of the guidance signals. Knowledge distillation (KD) is initially proposed for model compression, where a powerful wide/deep teacher distills knowledge to a narrow/shallow student to improve its performance [12, 41], which can also be treated as adding auxiliary supervisions. In terms of the definition of knowledge to be distilled from the teacher, existing models typically use teacher’s class probabilities [12] and/or intermediate features [15, 34, 41, 51, 57]. It is worth noting that our proposed auxiliary learning strategy uses weight sharing to assist optimization, where the motivation is very different from the KD methods. We does not need to pre-train a teacher network which is usually much deeper and may be the upper bound of the performance. Moreover, on network quantization, we show consistent superior performance over KD methods in Sec. 4.1.

Object detection. Object detection can be divided into two categories. As one of the dominant detection framework, two-stage detection methods [7, 8, 40] first generate region proposals and then refine them by subsequent networks. Another main category is the one-stage methods which are represented by YOLO [37–39], SSD [29] and RetinaNet [25]. The objective is to improve the detection efficiency by directly classifying and regressing the pre-defined anchors without the proposal generation step. The recent developing trends in object detection is designing light-weight frameworks for mobile applications [5, 44, 48], which usually requires real-time, low-power and fully embedded. In this paper, we explore to compress and accelerate detectors from the quantization perspective. Note that, we are the first to achieve near lossless 4-bit detectors in the literature.

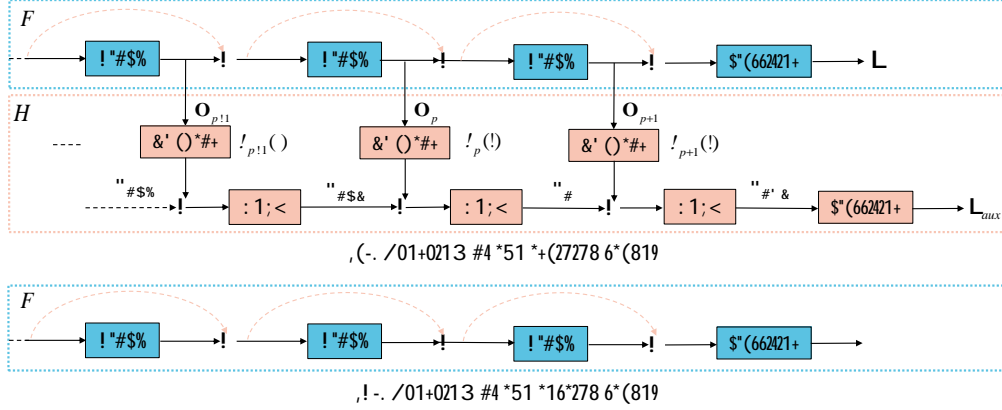


Figure 1: An overview of the proposed framework. Blue color represents low-precision operations while pink color denotes full-precision operations. During training, the full-precision H is connected to the quantized F to form a mixed-precision network $F + H$. Then the mixed-precision network and the low-precision F are jointly optimized with weight sharing. It is worth noting that only the learnt quantized network F is used during testing.

2. Method

In this section, we describe the proposed learning strategy for training a low-precision network. We first give an overview of the proposed approach in Sec. 2.1. Then we introduce the auxiliary module design in Sec. 2.2 and optimization in Sec. 2.3, respectively. After that, we provide discussions in Sec. 2.4 and explain how to extend it to quantized object detection in Sec. 3. Through the following part, we adopt the following terminologies: A layer is a standard parameterized layer in a network such as a dense or convolutional layer. A block is a collection of layers in which the output of its last layer is connected to the input of the next block (*e.g.*, a residual block).

2.1. Overview and motivation

The overview of the framework is shown in Fig. 1. The blue part, denoted as F , shows the low-precision network we aim to learn. The pink part, denoted as H , is a full-precision sub-network, which we call the auxiliary module. It connects to intermediate outputs from F . The input image is fed into F but generates two outputs, one is from the last layer in F and the other is from the last layer in H . In other words, the combination of F and H forms an augmented mix-precision network $F + H$ and the parameters of the low-precision network is shared by this augmented network. At the training time, two loss functions are applied to both outputs, and the mixed-precision network and low-precision network are trained jointly. After training, the auxiliary module H will be discarded and only F will be used at the test time.

The motivation of such a design is to create full-precision routes to update parameters of the low-precision model and thus alleviating the difficulty of propagating gradient in a quantized model. Specifically, the intermediate output of

each block in F can directly influence the output of the mixed-precision network $F + H$ through the full-precision connections in H . Consequently, the gradient from the loss of the second output will back propagate to the parameters of each block in the low-precision model.

2.2. Module design

We now elaborate the design for the auxiliary module H , which is made up of a sequential of adaptors and aggregators as shown in Fig. 1 (a). In particular, the auxiliary module H receives P output feature maps $\{O_p\}_{p=1}^P$ of the corresponding blocks in F . Let $\{B_1, \dots, B_P\}$ be the block indexes where we generate the feature maps. For the p -th input of H , we adopt a trainable adaptor $\rho_p(\cdot)$, which receives the output feature map O_p of the B_p -th block from F and outputs an adapted feature representation $\rho_p(O_p)$. The motivation of using the adaptor is to compensate the distribution discrepancy between the low-precision model and full-precision model. It ensures the quantized activations $\{O_p\}_{p=1}^P$ to be compatible to the full-precision calculation in H . We implement those adaptors by a simple 1×1 convolutional layer followed by a batch normalization layer in this paper.

In the auxiliary module H , the outputs of the adaptor are then sequentially aggregated. Formally, let g_p denotes the p -th aggregated feature. It is achieved by adding the adapted feature $\rho_p(O_p)$ from F and the $(p-1)$ -th aggregated feature from H followed by a $\text{ReLU}(\cdot)$ nonlinearity:

$$g_p = \text{ReLU}(\rho_p(O_p) + g_{p-1}). \quad (1)$$

At the last layer in H , a classifier layer is applied to g_p to make the class prediction. Then an auxiliary loss is employed. Note that the auxiliary module H is akin the skip connections in ResNet [11].

Algorithm 1: Joint training approach w.r.t. the main low-precision network F and the full-precision auxiliary module H .

Input: Current mini-batch $\{x_i, y_i\}$; parameter θ^F of the low-precision network F ; parameter θ^H of the full-precision auxiliary module H .

Output: Updated parameters $\{\theta^F, \theta^H\}$.

- 1 Obtain the quantized weight $Q^F = q(\theta^F)$, where $q(\cdot)$ is the quantization function;
- 2 $y_F, y_H = \text{Forward}(x_i, Q^F, \theta^H)$;
- 3 Compute the loss $L(y_i, y_F)$ for the main network F ;
- 4 Compute the loss $L_{\text{aux}}(y_i, y_H)$ for the auxiliary module H ;
- 5 $\frac{L}{Q^F}, \frac{L_{\text{aux}}}{Q^F}, \frac{L_{\text{aux}}}{\theta^H} = \text{Backward}(\frac{L}{y_F}, \frac{L_{\text{aux}}}{y_H}, Q^F, \theta^H)$;
- 6 Compute the gradient, in particular $Q^F = \frac{1}{2}(\frac{L}{Q^F} + \frac{L_{\text{aux}}}{Q^F})$;
- 7 Update parameters using Adam;

2.3. Optimization

Let $\{x_i, y_i\}_{i=1}^N$ be the training samples. The proposed method jointly optimize the main network F and the mixed-precision network which is the combination of F and H , denoting as $F \cup H$. The training objective is:

$$\min_{\{\theta^F, \theta^H\}} \sum_{i=1}^N L(F(x_i; \theta^F), y_i) + L_{\text{aux}}((F \cup H)(x_i; \theta^F, \theta^H), y_i), \quad (2)$$

where θ^F and θ^H represent the parameters for the backbone F and the auxiliary module H , respectively. L is the task objective and L_{aux} is the auxiliary loss. In the classification task, both terms are set to the cross-entropy loss. From Eq. (2), we can note that θ^F is shared among F and $F \cup H$. Following the chain rule, the gradient of θ^F will have an additional term comes from L_{aux} . As a result, the approximated gradient is averaged from both the mixed-precision network and the original low-precision network to achieve more accurate updating direction. In other words, the full-precision module H provides direct gradient for F using weight sharing during back-propagation. We summarize the proposed learning process for a quantized neural network in Algorithm 1.

2.4. Relationship to other methods

In this section, we will elaborate the relationship between the proposed auxiliary learning and other related approaches.

Differences to applying auxiliary classification losses to

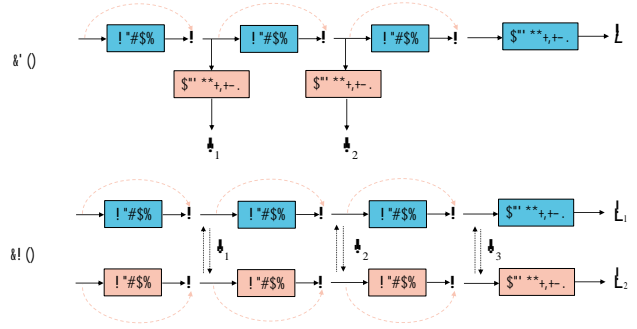


Figure 2: Overview of the two related approaches. (a): Adding additional classification losses to intermediate layers. (b): Knowledge distillation. In network quantization, the pink teacher network is full-precision while the blue student network is low-precision.

intermediate activations. An alternatively way to create auxiliary full-precision routes for learning low-precision network is to apply classification losses to intermediate activations. The schematic illustration of this idea is shown in Fig. 2 (a), where each intermediate output is attached to a classifier to perform classification. In such a case, the final training objective becomes:

$$L_{\text{obj}} = L + \sum_{i=1}^M \alpha_i \ell_i, \quad (3)$$

where L is the classification loss of the original low-precision network, ℓ_i is the classification loss applies to the i -th intermediate output, and α_i is the weight associated to the i -th loss function. This scheme can directly propagate gradient to each block during training through the full-precision classifiers. However, its supervision is very restrictive since it essentially assumes that the intermediate output can be directly used for classification. In practice, we often find that choosing the positions of adding the additional supervisions or the weight α_i can be challenging. An inappropriate setting of those factors may lead to inferior performance than that achieved by directly training the low-precision model.

Differences with knowledge distillation. Knowledge distillation (KD) has been explored to assist the quantized model training [32, 56, 57]. In particular, a low-precision student network learns to generate similar posterior probabilities and/or feature representations of a full-precision teacher network (see Fig. 2 (b)). The training objective can be formulated as

$$\hat{L}_{\text{obj}} = \hat{L}_1 + \hat{L}_2 + \sum_{i=1}^M \hat{\alpha}_i \hat{\ell}_i, \quad (4)$$

where \hat{L}_1 and \hat{L}_2 are task-specific objectives for student and teacher networks respectively. $\hat{\ell}_i$ represents the i -th distillation loss.

Although both the proposed method and *KD* use a full-precision network to guide the training of a low-precision network, the ways of exerting this guidance are significantly different in *KD* and the proposed method. Specifically, in *KD*, the guidance from the full-precision model to the low-precision model is from the distillation losses while in our method this is achieved by making the parameters of a low-precision model shared with the mixed-precision model. There are many advantages of our method comparing to *KD*: (1) Our method only needs an extra memory to store the auxiliary module rather than a full-precision network. Comparing to *KD*, our method is more memory-efficient. (2) Our method only uses a single auxiliary loss but can create guidance signals to various blocks of the low-precision model. In contrast, *KD* needs multiple distillation losses to achieve this. Thus it usually involves more hyper-parameters, i.e., the weight of each loss term λ_i . Moreover, we empirically find that the proposed learning strategy performs consistently superior than *KD* for learning quantized networks in Sec. 4.1.3 and Sec. 4.1.4.

3. Extension to object detection

Most existing methods evaluate low-precision network with the classification task. Building a low-precision network for more difficult object detection task remains a challenge. To fill this gap, we further extend our method to build a quantized object detector. Following the work in [24, 25], we consider the object detection framework consisting of a backbone, a feature pyramid and prediction heads. We directly use the quantized network pretrained on the ImageNet classification task to initialize the detection backbone. We adopt the uniform quantization approach QIL [17] to quantize both weights and activations, where the quantization intervals are explicitly parameterized and jointly optimized with the network parameters. We add an individual auxiliary module for each prediction head while sharing a single module for the backbone.

Beside of applying the proposed auxiliary module and learning strategy to help training the quantized detector, we also propose a modification which we find beneficial. Specifically, unlike [22] which freezes the batch normalization (BN) statistics during training to stabilize optimization, we instead propose an alternative strategy where the BN statistics still keeps updating: we propose that except the last layers for classification and regression, parameters of the prediction heads are not shared across all feature pyramid levels. This is different from the common full-precision setup. The motivation of this design is that the multi-scale semantic information can not be encoded effectively due to the quantization process at different pyramid levels. For a full-precision network, using a shared head is sufficient to represent rich semantic information for classification and regression with the continuous activations. However, in the

low-precision setting, the representational capability of activations is highly degraded due to its discrete values. For the same reason, the batch statistics of quantized activations may differ drastically across different levels. Therefore, each head should learn independent parameters to capture the corresponding multi-scale information.

Remark: (1) The proposed modification does not share prediction heads and thus uses more parameters in the low-precision model. However, we should note that without sharing does not increase any additional computational complexity. Even though the number of parameters is increased, the memory consumption is still significantly reduced comparing to the full-precision model due to the low-bit storage. (2) We empirically find that not sharing heads may not improve (but reduce) the performance in the full-precision setting. So the proposed modification is only for low-precision networks. Please check the experiments in Sec. 4.2.3 for more discussions.

4. Experiments

In this section, we evaluate our proposed methods on image classification in Sec. 4.1 and object detection in Sec. 4.2, respectively. To investigate the effectiveness of the proposed method, we define several methods for comparison: **Auxi:** We optimize the network with the auxiliary module. **KD:** We employ the joint knowledge distillation in [56, 57] to improve the quantized network. **Additional loss:** We evenly insert classification losses at intermediate layers to assist training. Note that we will detail the settings in specific sections.

4.1. Experiments on image classification

We perform experiments on two standard image classification datasets: CIFAR-100 [19] and ImageNet [42]. The CIFAR-100 dataset consists of 60,000 color images of size 32×32 belonging to 100 classes. There are 50,000 training and 10,000 test images. ImageNet contains about 1.2 million training and 50K validation images of 1,000 object categories. To verify the effectiveness of the proposed auxiliary learning strategy, we experiment on various representative quantization approaches, including uniform fixed-point approach DoReFa-Net [54], non-uniform fixed-point method LQ-Net [52], as well as binary neural network approaches BiReal-Net [30] and Group-Net [58].

4.1.1 Implementation details

Following previous approaches [14, 52, 54, 55, 57], we quantize all the convolutional layers to ultra-low precision except the first and last layers. However, to further improve the efficiency, we quantize the first convolutional layer and the last fully-connected layer to 8-bit. We first pre-train the full-precision counterpart as initialization and then fine-

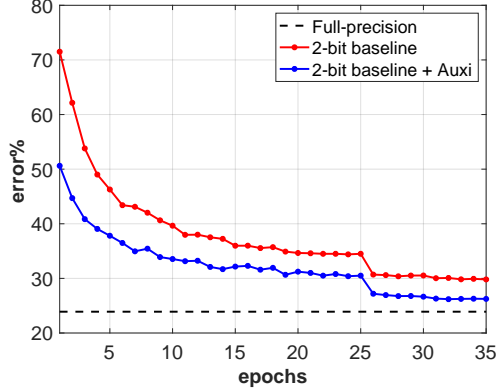


Figure 3: The convergence curves of 2-bit DoReFa-Net ResNet-50 baseline and the proposed auxiliary learning approach on ImageNet validation set during fine-tuning.

tune the quantized model. For all ImageNet experiments, training images are resized to 256×256 , and 224×224 patches are randomly cropped from an image or its horizontal flip, with the per-pixel mean subtracted. We use the single-crop setting for testing. No bias terms are used. We use SGD optimizer for the pre-training stage. For the fine-tuning stage, we adopt the Adam optimizer [18]. The mini-batch size is set to 256. We train a maximum 35 epochs and decay the learning rate by 10 at the 25-th and 30-th epochs. For fine-tuning the fixed-point methods [52, 54], the learning rate is initialized to $1e-3$. For fine-tuning binary neural networks [30, 58], the initial learning rate is set to $5e-4$. In practice, we take the output of each residual block [11] as the input of the auxiliary module. Our implementation is based on PyTorch.

4.1.2 Effect of the auxiliary module

Table 1: Accuracy (%) of different comparing methods on the ImageNet validation set.

model	method	Top-1 acc.	Top-5 acc.
ResNet-101	DoReFa-Net (2-bit)	70.8	89.6
	DoReFa-Net + Aux	74.6	91.9
ResNet-50	DoReFa-Net (2-bit)	70.2	89.1
	DoReFa-Net + Aux	73.8	91.4
ResNet-50	LQ-Net (3-bit)	74.2	91.6
	LQ-Net + Aux	75.4	92.4
ResNet-18	BiReal-Net	56.4	79.5
	BiReal-Net + Aux	58.6	81.2
ResNet-18	Group-Net (5 bases)	64.8	85.7
	Group-Net + Aux	66.0	86.5

In this section, we explore the effect of auxiliary module on assisting the low-precision network optimization. The results are reported in Table 1. By combining the baseline with *Aux*, we can observe a steady performance increase compared with the original baseline. This strongly

supports that the learned auxiliary gradient can facilitate the convergence of the low-precision model. In particular, the gradient of the shared weights is averaged from both the auxiliary module and the original low-precision network to achieve more accurate updating direction. Moreover, increasing gradient paths is important to solve the non-differentiability of the discrete quantization process has been proved by [3, 30]. To make it more clear, we plot the convergence curves of DoReFa-Net with ResNet-50 in Fig. 3. From the figure, we can observe that *baseline + Aux* converges much faster and better than the baseline. After the first epoch, the *baseline + Aux* outperforms *baseline* by 20% on the Top-1 accuracy. This result strongly justifies that the auxiliary module efficiently solves the non-differentiable problem by providing accurate hierarchical gradient for updating parameters during back-propagation.

It is worth noting that when the network becomes deeper (e.g., ResNet-50, 101), the improvement turns out to be more obvious. For instance, *Aux* brings **3.8%** Top-1 accuracy increase over the 2-bit baseline on ResNet-101. It can be attributed to that when the quantized network goes deeper, the optimization becomes more difficult due to the non-differentiable discretization process. However, the full-precision auxiliary module can provide direct hierarchical gradient to effectively solve this problem. Note that with the basic uniform DoReFa-Net baseline, we currently achieve the comparing results with state-of-the-arts on ResNet-50 and ResNet-101 [17, 31, 52] without advanced non-uniform or relaxation strategies.

4.1.3 Comparison with other related approaches

In this section, we compare the auxiliary module with the related approaches discussed in Sec. 2.4 and report the performance in Table 2. The experiments are based on the 2-bit DoReFa-Net with ResNet-18, ResNet-34 and ResNet-50 on ImageNet. For KD experiments, the results are directly cited from [56]. We can observe that introducing additional losses into intermediate layers does not show obvious improvement to the final performance. Compared to *KD*, we don't need to pre-train a complex teacher network whose quality is sensitive to the final performance. In contrast, we propose a simpler yet effective weight sharing strategy to jointly optimize the low-precision network and the floating-point auxiliary module. More detailed analysis on the difference between two approaches can be referred to Sec. 2.4. From the results, we can observe that *Aux* consistently outperforms *KD*. For instance, on ResNet-50, *Aux* exceeds *KD* by **2.4%** on the Top-1 accuracy. These results justify that the auxiliary module can effectively solve the non-differentiable problem during back-propagation in the low-precision network training. With a very different learning strategy, *Aux* shows consistently superior empirical results than *KD*. We therefore argue that the proposed auxil-

iary learning may be a substitute to *KD* methods [32, 57] in network quantization.

Table 2: Accuracy (%) of different supervision strategies on the ImageNet validation set based on 2-bit DoReFa-Net on ResNet-18, ResNet-34 and ResNet-50.

model	method	Top-1 acc.	Top-5 acc.
ResNet-18	baseline (2-bit)	64.7	86.0
	baseline + Additional loss	64.9	86.1
	baseline + KD	65.6	86.3
	baseline + Aux	66.7	87.0
ResNet-34	baseline (2-bit)	68.2	88.1
	baseline + Additional loss	68.5	88.2
	baseline + KD	69.0	88.6
	baseline + Aux	71.2	89.8
ResNet-50	baseline (2-bit)	70.2	89.1
	baseline + Additional loss	70.5	89.3
	baseline + KD	71.4	90.0
	baseline + Aux	73.8	91.4

4.1.4 Experiments on plain networks

Table 3: Accuracy (%) of the proposed approaches on the ImageNet validation set. All the cases are 2-bit and without skip connections except for the baselines. We can observe that the auxiliary module can significantly improve the plain network performance.

model	method	Top-1 acc.	Top-5 acc.
DoReFa-Net on ResNet-18	baseline (2-bit)	64.7	86.0
	plain	61.5	84.3
	plain + KD	62.7	85.0
	plain + Aux	63.9	85.5
DoReFa-Net on ResNet-34	baseline (2-bit)	68.2	88.1
	plain	62.1	83.9
	plain + KD	64.5	85.4
	plain + Aux	66.4	86.8
LQ-Net on ResNet-34	baseline (2-bit)	69.8	89.1
	plain	63.5	84.6
	plain + KD	65.7	86.8
	plain + Aux	68.6	88.5

Table 4: Accuracy (%) of 2-bit DoReFa-Net using ResNet-18 on the CIFAR-100 dataset.

model	method	Top-1 acc.	Top-5 acc.
ResNet-18	full-precision	70.7	91.3
	baseline (2-bit)	67.6	90.2
	plain	64.6	88.3
	plain + Aux	67.9	90.0

We further explore an interesting by-product of the auxiliary module for network quantization. We assume that the auxiliary module mimics the effect of skip connections and can partially share its effect. We therefore analyze training a plain low-precision network without skip connections. The results can be referred in Table 3 and Table 4. *plain* represents we directly optimize a low-precision plain network without skip connections. By comparing *plain* and *plain* +

Aux, we observe apparent accuracy increase by incorporating *Aux*. For example, in LQ-Net ResNet-34 based experiments, introducing *Aux* can boost the Top-1 accuracy by 5.1%. On tiny CIFAR-100 dataset, *plain* + *Aux* even outperforms the Top-1 baseline. Moreover, same as the observation in Sec. 4.1.3, *Aux* still performs consistently better than *KD*. From the plain network setting, we can strongly justify that the auxiliary module can provide hierarchical gradient to promote convergence of the quantized network.

However, we still observe performance gap between *plain* + *Aux* and the baseline on large-scale ImageNet. This can be attributed to two assumptions of skip connections. First, the skip connections may improve the convergence of training, as indicated by the improvement observed when using *Aux*. Second, the skip connection and the feature map after one convolution are added through a tensor addition. Then the representational capability (i.e., the value range) of each entry in the added activations is significantly enhanced. In other words, the plain network has less representational capability than its residual counterpart.

4.1.5 Effect of different auxiliary architectures

We further explore the influence of different auxiliary module architectures in Table 5. From the table, we observe that increasing the complexity of the auxiliary module can further boost the performance. For example, by replacing the 1×1 convolution in the adaptor with a larger kernel of 3×3 , we further get slightly performance gain. This can be attributed to that the gradient of shared parameters is averaged from F and $F + H$, where better representational capability of H can result in more accurate gradient update.

Table 5: Accuracy (%) of using different adaptors. We use DoReFa-Net on ImageNet as our baseline.

model	method	Top-1 acc.	Top-5 acc.
ResNet-18	baseline (2-bit)	64.7	86.0
	baseline + 1×1 Aux	66.7	87.0
	baseline + 3×3 Aux	66.9	87.1

4.2. Experiments on quantized object detection

In this section, we evaluate the proposed approach on the general object detection task. Our experiments are conducted on the large-scale detection benchmark COCO [26]. Following [24, 25], we use the COCO *trainval35k* split (115K images) for training and *minival* split (5K images) for validation. We conduct experiments based on RetinaNet [25] and compare with the state-of-the-art FQN [22].

4.2.1 Training details

Training is divided into two stages. In the first stage, the detection framework is kept to full-precision. The backbone is initialized with the classification model pre-trained

on the ImageNet dataset. Unless specified, we use the same hyper-parameters with RetinaNet. Specifically, all training and evaluation images are resized so that their shorter edges are 800 pixels. We augment training images by random horizontal flipping while no evaluation augmentations are performed. Our network is trained with stochastic gradient descent (SGD) for 90K iterations with the initial learning rate being 0.01 and the batch size of 16. The learning rate is decayed by a factor of 10 at iterations 60K and 80K, respectively. In the second stage, we use the converged model in the first stage as initialization and fine-tune with quantization. This stage uses identical settings as the full-precision training, except that we use Adam optimizer and the initial learning rate is set to 1e-3. Our implementation is based on Detectron2 [50].

4.2.2 Performance evaluation

We report the performance of the proposed quantized detection framework in Table 6. From the results, we can observe that our 4-bit detector can achieve near lossless results over the full-precision counterparts, which meets the requirement for practical deployment. Moreover, we can achieve significant performance boost over FQN on all comparing architectures. For example, on ResNet-50, the improvement reaches to 3.6 on AP.

Table 6: Performance on the COCO validation set with 4-bit quantization.

Backbone	Method	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
ResNet-50	RetinaNet	36.5	56.5	39.2	21.4	40.4	46.9
	FQN [22]	32.5	51.5	34.7	17.3	35.6	42.6
	Ours	36.1	55.8	38.9	21.2	39.9	46.3
ResNet-34	RetinaNet	35.2	54.3	37.6	19.5	38.5	46.2
	FQN [22]	31.3	50.4	33.3	16.1	34.4	41.6
	Ours	34.7	53.7	36.9	19.3	38.0	45.9
ResNet-18	RetinaNet	32.1	50.5	34.1	16.9	34.8	42.6
	FQN [22]	28.6	46.9	29.9	14.9	31.2	38.7
	Ours	31.9	50.4	33.7	16.5	34.6	42.3

4.2.3 Ablation studies

We now perform ablation studies to give comprehensive analysis and insights of the quantized detection framework. The results are reported in Table 7. “Backbone only” indicates we only quantize the backbone to 4-bit while other parts are kept to full-precision. “Baseline” represents we directly quantize the RetinaNet using QIL [17].

When we only quantize the backbone network, we don’t observe AP drop at least on ResNet-18. This justifies that 4-bit backbone can encode accurate features for further decoding. However, when quantizing all the components including feature pyramid and prediction heads, we can find obvious precision drop. The baseline result indicates that quantizing the continuous features to a fixed range of integers can cause great multi-scale information loss.

We now explore the effect of the two strategies described in Sec. 3. First, we incorporate the proposed auxiliary

Table 7: Ablation studies on the COCO validation set with 4-bit quantization.

Method	AP	AP ₅₀	AP ₇₅	AP _S	AP _M	AP _L
RetinaNet (w/ sharing)	32.1	50.5	34.1	16.9	34.8	42.6
RetinaNet (w/o sharing)	31.2	49.2	32.9	15.8	34.3	41.5
Backbone only (w/ sharing)	32.1	50.7	34.0	16.7	34.7	42.7
Baseline (w/ sharing)	29.2	47.1	31.0	14.4	31.5	38.5
Baseline + Auxl (w/ sharing)	30.6	48.8	32.8	15.5	33.1	40.2
Baseline + Auxl (w/o sharing)	31.9	50.4	33.7	16.5	34.6	42.3

learning strategy to assist the convergence of quantized detector. Specifically, *Baseline + Auxl* can boost the AP of *Baseline* by 1.4. Second, not sharing heads is particularly designed for the low-precision detector. We should note that this strategy deteriorates the performance in the full-precision setting. The reason is that a separate head can only see a certain range size of objects. In contrast, not sharing heads can boost the AP by 1.3 in the quantization setting compared with the sharing heads counterpart. It can be attributed that when quantizing both weights and activations to 4-bit, the representational capability of each head is very limited and the statistics of activations differ a lot. Therefore, each head should learn independent parameters to transform the corresponding level feature for classification and regression.

5. Conclusion

In this paper, we have proposed an auxiliary learning strategy to tackle the non-differentiable quantization process in training low-bitwise convolutional neural networks. Specifically, we have explicitly utilized weight sharing to construct a full-precision auxiliary module. During training, the auxiliary module is combined with the low-precision network to form a mix-precision network, which is jointly optimized with the low-precision model. In this way, the full-precision auxiliary module can provide direct hierarchical gradient during back-propagation to assist the optimization of the low-precision network. In the testing phase, the auxiliary module is removed without introducing any additional computational complexity. Moreover, we have also worked on quantized object detection and proposed several practical solutions. We have conducted extensive experiments based on various quantization approaches and observed consistent performance increase on the image classification and object detection. To be emphasized, we have achieved near lossless results using 4-bit detectors.

Acknowledgement MT was in part supported by Guangdong Provincial Scientific and Technological Funds under Grants 2018B010107001. CS was in part supported by ARC DP Project ‘Deep learning that scales’. IR was in part supported by ARC Laureate Fellowship FL130100102. We acknowledge the support of the Australian Research Council through the Centre of Excellence for Robotic Vision.

References

- [1] Yu Bai, Yu-Xiang Wang, and Edo Liberty. Proxquant: Quantized neural networks via proximal operators. In *Proc. Int. Conf. Learn. Repren.*, 2019. **1, 2**
- [2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. **1**
- [3] Joseph Bethge, Marvin Bornstein, Adrian Loy, Haojin Yang, and Christoph Meinel. Training competitive binary neural networks from scratch. *arXiv preprint arXiv:1812.01965*, 2018. **6**
- [4] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5918–5926, 2017. **2**
- [5] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 742–751, 2017. **2**
- [6] Yukang Chen, Gaofeng Meng, Qian Zhang, Xinbang Zhang, Liangchen Song, Shiming Xiang, and Chunhong Pan. Joint neural architecture search and quantization. *arXiv preprint arXiv:1811.09426*, 2018. **2**
- [7] Ross Girshick. Fast r-cnn. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1440–1448, 2015. **2**
- [8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 580–587, 2014. **1, 2**
- [9] Yiwen Guo, Anbang Yao, Hao Zhao, and Yurong Chen. Network sketching: Exploiting binary structure in deep cnns. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5955–5963, 2017. **2**
- [10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2980–2988, 2017. **1, 2**
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016. **1, 3, 6**
- [12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Proc. Adv. Neural Inf. Process. Syst. Workshops*, 2014. **2**
- [13] Lu Hou, Quanming Yao, and James T Kwok. Loss-aware binarization of deep networks. In *Proc. Int. Conf. Learn. Repren.*, 2017. **2**
- [14] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 4107–4115, 2016. **2, 5**
- [15] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised knowledge distillation using singular value decomposition. In *Proc. Eur. Conf. Comp. Vis.*, 2018. **2**
- [16] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2704–2713, 2018. **1**
- [17] Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4350–4359, 2019. **2, 5, 6, 8**
- [18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learn. Repren.*, 2015. **6**
- [19] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. **5**
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 1097–1105, 2012. **1**
- [21] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, 2015. **2**
- [22] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2810–2819, 2019. **1, 5, 7, 8**
- [23] Zefan Li, Bingbing Ni, Wenjun Zhang, Xiaokang Yang, and Wen Gao. Performance guaranteed network acceleration via high-order residual quantization. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2584–2592, 2017. **2**
- [24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2117–2125, 2017. **5, 7**
- [25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2980–2988, 2017. **2, 5, 7**
- [26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, pages 740–755, 2014. **7**
- [27] Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 344–352, 2017. **2**
- [28] Chunlei Liu, Wenrui Ding, Xin Xia, Baochang Zhang, Jiaxin Gu, Jianzhuang Liu, Rongrong Ji, and David Doermann. Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnn with circulant back propagation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2691–2699, 2019. **2**
- [29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proc. Eur. Conf. Comp. Vis.*, pages 21–37, 2016. **2**
- [30] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proc. Eur. Conf. Comp. Vis.*, pages 722–737, 2018. **2, 5, 6**

- [31] Christos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, and Max Welling. Relaxed quantization for discretized neural networks. In *Proc. Int. Conf. Learn. Repren.*, 2019. **1, 2, 6**
- [32] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *Proc. Int. Conf. Learn. Repren.*, 2018. **1, 4, 7**
- [33] Vladimir Nekrasov, Hao Chen, Chunhua Shen, and Ian Reid. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9126–9135, 2019. **2**
- [34] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019. **2**
- [35] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *Proc. Int. Conf. Mach. Learn.*, pages 4092–4101, 2018. **2**
- [36] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proc. Eur. Conf. Comp. Vis.*, pages 525–542, 2016. **2**
- [37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 779–788, 2016. **1, 2**
- [38] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7263–7271, 2017. **2**
- [39] Joseph Redmon and Ali Farhadi. Yolo3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. **2**
- [40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 91–99, 2015. **1, 2**
- [41] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *Proc. Int. Conf. Learn. Repren.*, 2015. **2**
- [42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comp. Vis.*, 115(3):211–252, 2015. **5**
- [43] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1–9, 2015. **2**
- [44] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnas-net: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019. **2**
- [45] Wei Tang, Gang Hua, and Liang Wang. How to train a compact binary neural network with high accuracy? In *Proc. AAAI Conf. on Arti. Intel.*, pages 2625–2631, 2017. **2**
- [46] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019. **2**
- [47] Peisong Wang, Qinghao Hu, Yifan Zhang, Chunjie Zhang, Yang Liu, Jian Cheng, et al. Two-step quantization for low-bit neural networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4376–4384, 2018. **2**
- [48] Yi Wei, Xinyu Pan, Hongwei Qin, Wanli Ouyang, and Junjie Yan. Quantization mimic: Towards very tiny cnn for object detection. In *Proc. Eur. Conf. Comp. Vis.*, pages 267–283, 2018. **1, 2**
- [49] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv preprint arXiv:1812.00090*, 2018. **2**
- [50] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. **8**
- [51] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Proc. Int. Conf. Learn. Repren.*, 2017. **2**
- [52] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *Proc. Eur. Conf. Comp. Vis.*, pages 365–382, 2018. **2, 5, 6**
- [53] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2881–2890, 2017. **2**
- [54] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016. **2, 5, 6**
- [55] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. *Proc. Int. Conf. Learn. Repren.*, 2017. **5**
- [56] Bohan Zhuang, Jing Liu, Mingkui Tan, Lingqiao Liu, Ian Reid, and Chunhua Shen. Effective training of convolutional neural networks with low-bitwidth weights and activations. *arXiv preprint arXiv:1908.04680*, 2019. **1, 4, 5, 6**
- [57] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Towards effective low-bitwidth convolutional neural networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7920–7928, 2018. **1, 2, 4, 5, 7**
- [58] Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Structured binary neural network for accurate image classification and semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 413–422, 2019. **2, 5, 6**