FTL: A universal framework for training low-bit DNNs via Feature Transfer

Kunyuan Du[1], Ya Zhang[1] ✉, Haibing Guan[1], Qi Tian[2], Shenggan Cheng[1], and James Lin[1]

[1] Shanghai Jiao Tong University
{dukunyuan, ya_zhang, wangyanfeng, chengshenggan, james}@sjtu.edu.cn
[2] Huawei Noah's Ark Lab tian.qi1@huawei.com

Abstract. Low-bit Deep Neural Networks (low-bit DNNs) have recently received significant attention for their high efficiency. However, low-bit DNNs are often difficult to optimize due to the the saddle points in loss surfaces. Here we introduce a novel feature-based knowledge transfer framework, which utilizes a 32-bit DNN to guide the training of a low-bit DNN via feature maps. It is challenge because feature maps from two branches lie in continuous and discrete space respectively, and such mismatch has not been handled properly by existing feature transfer frameworks. In this paper, we propose to directly transfer information-rich continuous-space feature to the low-bit branch. To alleviate the negative impacts brought by the feature quantizer during the transfer process, we make two branches interact via centered cosine distance rather than the widely-used p-norms. Extensive experiments are conducted on Cifar10/100 and ImageNet. Compared with low-bit models trained directly, the proposed framework brings 0.5% to 3.4% accuracy gains to three different quantization schemes. Besides, the proposed framework can also be combined with other techniques, e.g. logits transfer, for further enhacement.

Keywords: Low-bit DNN, Feature Transfer, Space mismatch

1 Introduction

The gains of Deep Neural Networks (DNNs) in various pattern analysis tasks have been accompanied by dramatic increases in model complexity. To mitigate this problem, Network quantization [8, 21, 15, 34, 30, 11, 10, 23], which converts 32-bit weights and activations into low-bit, has been proposed to deploy models to mobile platforms. For example, XNOR-net [21] can achieve 58× faster convolutional operations and 32× smaller model size. However, optimizing a low-bit DNN is often more difficult due to the noise in gradients [28] and the saddle points in its loss surface [23]. Various techniques have been developed to better train a low-bit model. Incremental quantization [33] gradually decreases the bit-width of the model to better adapt to the quantization noise. Logits Transfer [18, 20] supervises low-bit DNNs with soft labels from 32-bit DNNs to make use of

the correlation between labels. Attention Transfer [17] encourages low-bit DNNs to produce high quality attention maps for better training. And Feature Transfer [35, 27] guides low-bit DNNs via feature maps from 32-bit ones. Methods from different categories assist the optimization of low-bit DNNs from different aspects, and can be combined for further enhancement.

In this paper, we focus on the feature transfer approach. For non-quantized DNNs, a variety of feature transfer frameworks [26, 14, 16, 1, 25, 6] have been proposed, which directly minimizes the distance between feature maps from two DNN branches. Nevertheless, feature maps from low-bit DNNs lie in discrete space. Hence, the above studies cannot be directly applied to low-bit ones. To resolve this mismatch, [35, 27] quantizes the continuous knowledge to the same discrete space before transferring. However, transferring discrete knowledge has two main drawbacks. Firstly, regularization in discrete space introduces abrupt changes to gradients, especially for lower bit-width, which leads to unstable training process. Secondly, to convert 32-bit feature maps to discrete space, the specific form of the quantizer is needed. However the quantizer is adaptive for advanced quantization schemes [30, 11] and has no explicit expression. It is desired to design an universal framework, which can handle the mismatch problem for all low-bit DNNs.

To resolve the above problem, this paper explores to directly transfer knowledge from a 32-bit DNN to low-bit one without quantization. In other word, we propose to perform knowledge transfer before the quantizer. Since the quantizer is sensitive to distribution fluctuations during training, we introduce *centered cosine similarity* to replace the widely-used p-norms as the distance function, which focuses on the relative numerical relationship between feature elements and can better maintain the data distribution. We further reveal that the training of a low-bit DNN can be regarded as minimizing its distance to the corresponding 32-bit version. Because low-bit DNNs have much lower learning capacity than the 32-bit ones and may fail to follow its guidance, we further explore to relax the guidance of the 32-bit branch during training. It's worth noting that the proposed method is independent of the form of the quantizer. Therefore, it is an universal framework applicable to all low-bit DNNs.

To demonstrate the effectiveness of the proposed method in improving the performance of low-bit DNNs, we experiment with different benchmark datasets (i.e. Cifar-10/100 [13] and ImageNet [3]), different models (i.e. Alexnet, Vggnet and Resnet), different bit-width and different quantization algorithms (i.e. BNN [8], DoReFa-Net [34], LQ-Nets [30]). The proposed method consistently achieves 0.5% to 3.4% accuracy gains, and reaches *state-of-the-art* performance when taking LQ-Net [30] as the base model. Furthermore, experimental results show that FTL can be combined with other approaches, e.g. logits Transfer[7], for further enhancement.

Below we summary the main contributions of this paper.

- We propose to guide a low-bit DNN before its feature quantizer, which leads to more stable training process and more accurate guidance.

– We introduce centered cosine similarity for feature transfer, which ensures the consistency of guidance when going through the feature quantizer.
– We explore to relax the guidance when the learning capacities of the low-bit DNN prevent it from absorbing the knowledge from the 32-bit DNN.

2 Related Work

The proposed algorithm aims to enhance the classification performance of low-bit DNNs, leveraging a special case of knowledge transfer frameworks.

2.1 Low-bit DNNs

For smaller model size and higher computational efficiency, both weights and activations of low-bit DNNs lie in discrete space [21]. According to bit-width, low-bit DNNs can be divided into two categories. With bit-width of 8 or 16, the low-bit DNNs can directly be obtained from the 32-bit DNN without additional training [10, 32, 19]. However, with bit-width equal or less than 4 bits [21, 15, 34, 30, 11], re-training considering quantization effect is required to mitigate accuracy degradation, namely quantization-aware training. In this paper, all low-bit DNNs refer to the second category unless otherwise specified. BNN [8] is one of the earliest work for extreme low-bit quantization. To improve the representational capacity of low-bit models, XNOR-Net [21] assigns scaling weights to each layer. Dorefa-net [34] extends XNOR-Net from binarization to arbitrary bit-width. LQ-nets further adopt adaptive quantizer to enhance the flexibility of the model. In [11], network pruning technique is adopted to optimize the quantization interval. In order to prove that our method can be applied to various quantization algorithms, we choose BNN [8], DoReFa-net [34] and LQ-nets [30] as our base models.

2.2 Knowledge Transfer

DNNs learn 'knowledge' from training data, and the 'knowledge' can be transferred from one DNN to another. In logits-based Knowledge Transfer [7], knowledge can be viewed as the soft label from the pre-trained teacher DNN, which is absorbed by student DNN via minimizing the Kullback-Leibler divergence between outputs of two DNNs. Such process can be repeated for multiple times for further enhancement [4]. In mutual learning [31], DNNs can learn from each other, rather than one way transfer from teacher to student.

Knowledge can also be transferred via feature maps. Since feature maps have much higher dimensions, it is more challenging to align two DNNs in middle layer than logits. To transfer knowledge, previous frameworks empirically minimize the p-norms between feature maps from two DNNs, without further explanation. Some methods [27, 26, 14] add 'attention' to original feature maps, or directly transfer the attention maps between different DNNs [29]. And generative adversarial learning is adopted in [16] to better align different branches. However,

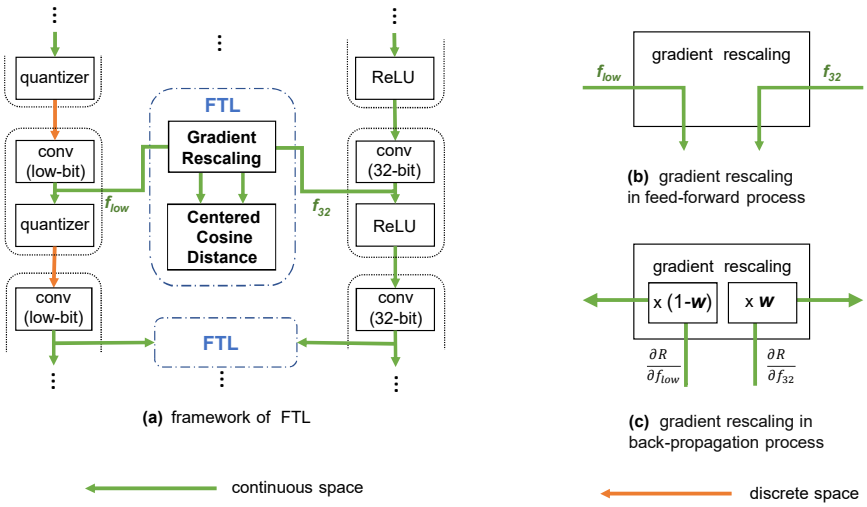

Fig. 1. Subfigure **(a)** is the overall framework of FTL. Subfigure **(b)** and **(c)** are further demonstration of gradient rescaling module in FTL.

most frameworks can not be applied to transfer feature-based knowledge from 32-bit DNNs to low-bit DNNs, because it is nontrivial to align feature maps with different numerical precision.

3 Feature Transfer for Low-bit DNNs

This section introduces the proposed method, which is specially designed for low-bit DNNs to overcome the space mismatch problem. Below we first introduce the overall framework, then the key components will be addressed in details.

3.1 Overall Framework

We attempt to utilize a 32-bit DNN to guide the training of a low-bit DNN via feature maps. Both two DNN branches are trained from scratch. This training scheme enables the low-bit DNN to learn the path to convergence [18]. For the sake of simplicity, the 32-bit DNN is constructed with the same hyper-parameters of the low-bit DNN. The only difference between two DNNs is that the low-bit one quantizes its weights and feature maps in each layer to discrete space.

Fig. 1 shows the overall framework of FTL. Intuitively, the low-bit DNN should learn from multiple layers of the 32-bit DNN to achieve more accuracy gains. However, feature maps of the first few layers often have weaker semantic information and more redundant details, which makes interaction via these layers a universal challenge for Knowledge Transfer frameworks. Since the focus of this paper is to alleviate the problem caused by the mismatch between continuous

space and discrete space, we only make two DNNs interact via the output of the last convolution layer, which follows the practice in previous frameworks [12, 27]. We do not need to explicitly define the form of the quantizer $Q(\cdot)$ because the proposed framework is independent of $Q(\cdot)$. The interaction between two DNN branches is achieved by minimizing a distance function in continuous space. For clear description, we denote the training data set as $S_t = \{(x_i, y_i)\}_{i=1}^N$, where $\{x_i\}_{i=1}^N$ are the inputs and $\{y_i\}_{i=1}^N$ are corresponding targets. We further denote the feature maps in the 32-bit and low-bit branch as f_{32} and f_{low} respectively. Note that f_{low} is in continuous space. And it will be converted to discrete space via quantizer $Q(\cdot)$ before further forward computation. To design the distance function, two factors need to be taken into account. On the one hand, since f_{low} is followed by quantizer $Q(\cdot)$, the knowledge transferred to f_{low} may be degraded by $Q(\cdot)$. On the other hand, the guidance of f_{32} may cause f_{low} to deviate from the optimal numerical range for quantization operation. To handle both issues, we implement the distance function based on centered cosine similarity. Different from the widely-used p-norms, the centered cosine distance focuses on the relative numerical relationship between feature elements, rather than their numerical differences. The overall optimization object is to obtain:

$$\min \quad L_{C_{32}} + L_{C_{low}} + \lambda \cdot R(f_{low}, f_{32}), \tag{1}$$

where the first two terms $L_{C_{32}}$ and $L_{C_{low}}$ denote the widely-used cross entropy losses for the 32-bit DNN and the low-bit DNN respectively. The third term $R(f_{low}, f_{32})$ denotes the distance function, which is designed to add additional supervision on low-bit DNNs and can be viewed as regularization. $\lambda \geq 0$ is a balancing parameter.

In our framework, knowledge is transferred from f_{32} to f_{low}. It seems straightforward to make f_{32} not influenced by f_{low} (equivalent to set $\frac{\partial R}{\partial f_{32}} = 0$). However, due to its limited representational capacity, the low-bit DNN may fail to follow the strong guidance from the 32-bit DNN and the 32-bit DNN should 'realize' it and make some concessions, which in turn requires $\frac{\partial R}{\partial f_{32}} \neq 0$. To balance these two requirements, we should control how much $\frac{\partial R}{\partial f_{32}}$ is retained. Therefore, Gradient Rescaling Module is designed to balance the 'guidance' and 'concessions' from the 32-bit DNN. Next, we will introduce the design of regularization function in detail, and further combine it with Gradient Rescaling Module.

3.2 Distance Function

Definition In this section, we explore the design of distance function for the low-bit branch. The goal is to transfer knowledge from f_{32} to $Q(f_{low})$, where $Q(\cdot)$ is the feature quantizer. Since f_{32} and $Q(f_{low})$ lies in continous space and discrete space respectively, it is nontrivial to minimize the distance between $Q(f_{low})$ and f_{32}. Previous methods [35, 27] first quantize f_{32} and then utilize $Q(f_{32})$ to guide $Q(f_{low})$. However, minimizing the distance in discrete space brings abrupt changes to gradients, and quantization operation on f_{32} can lead to a loss of information for the knowledge. We instead directly minimize the

distance between f_{low} and f_{32}, and focus on their mismatch probability [22, 24]. To be specific, given any pair of feature elements $f_{32}^i \leq f_{32}^j$ in the 32-bit branch, it is desired that the low-bit branch produces $f_{low}^i \leq f_{low}^j$, so that feature elements in f_{low} and f_{32} have a high positive correlation. To achieve this purpose, we define the distance function based on centered cosine similarity, as shown in Eq.(2).

$$R(f_{32}, f_{low}) = 1 - \frac{(f_{low} - \overline{f_{low}}) \cdot (f_{32} - \overline{f_{32}})}{\left\| f_{low} - \overline{f_{low}} \right\|_2 \left\| f_{32} - \overline{f_{32}} \right\|_2}, \tag{2}$$

The distance function defined by Eq.(2) enables f_{low} to produce a high centered cosine similarity with f_{32}. Note that due to the non-decreasing property of $Q(\cdot)$, if $f_{low}^i \leq f_{low}^j$, the quantized feature maps in the low-bit branch also satisfy $Q(f_{low}^i) \leq Q(f_{low}^j)$, which ensures the consistency of knowledge passing through $Q(\cdot)$. Compared with p-norms, centered cosine distance only constrains the relative numerical relationship between elements of f_{low} rather than the magnitude of each element, which brings negligible changes to the data distribution of f_{low} during training. Such property is beneficial for the low-bit branch to converge, because f_{low} should maintain certain distribution to match with the quantizer $Q(\cdot)$, and the distribution fluctuation will be amplified by $Q(f_{low})$ due to its coarse feature pixel values.

Relationship with Mutual Information We assume that feature maps f_{low} and f_{32} are generated by variables $\boldsymbol{v_{low}}$ and $\boldsymbol{v_{32}}$ respectively. Since previous works [9, 2] have shown that both f_{low} and f_{32} follow a data distribution similar to Gaussian, we model both $\boldsymbol{v_{low}}$ and $\boldsymbol{v_{32}}$ as Gaussian variables $\boldsymbol{v_{low}} \sim \mathcal{N}(\mu_{low}, \sigma_{low}^2)$ and $\boldsymbol{v_{32}} \sim \mathcal{N}(\mu_{32}, \sigma_{32}^2)$. On this basis, the Mutual Information between $\boldsymbol{v_{low}}$ and $\boldsymbol{v_{32}}$ can be explicit formulated as Eq.(3) [5], where ρ is the correlation coefficient between $\boldsymbol{v_{low}}$ and $\boldsymbol{v_{32}}$, ranging from -1 to 1. Note that ρ can be estimated by the centered cosine similarity between f_{32} and f_{low}. Therefore, the minimizing the centered cosine distance can be viewed as to increase of Mutual Information between $\boldsymbol{v_{low}}$ and $\boldsymbol{v_{32}}$.

$$MI(\boldsymbol{v_{low}}, \boldsymbol{v_{32}}) = -\frac{1}{2} log(1 - \rho^2). \tag{3}$$

The question arising naturally is that can we directly maximize the Mutual Information in Eq.(3) rather than centered cosine similarity for guidance? Below we demonstrate two main drawbacks of Eq.(3). The first is gradients explosion, which can lead to unstable training process, as is shown in Eq.(4). The second problem is that maximizing Eq.(3) may induce $\rho \to -1$, which indicates that $\boldsymbol{v_{low}}$ and $\boldsymbol{v_{32}}$ have a strong negative correlation. However, $\boldsymbol{v_{low}}$ and $\boldsymbol{v_{32}}$ should instead have a positive correlation to ensure both of them are activated (or clipped), because most quantizers and activation functions only activate larger values while clipping smaller ones. Therefore, it is inappropriate to directly utilize Eq.(3) to guide the low-bit DNN.

$$\lim_{|\rho|\to 1}\left|\frac{\partial\ log(1-\rho^2)}{\partial\rho}\right| = +\infty. \tag{4}$$

3.3 Gradient Rescaling Module

In the field of knowledge transfer, two (or more) DNNs interact typically by two modes: teacher-student mode and mutual learning mode. The former transfers knowledge unidirectionally from 'teacher' DNN to 'student' DNN, while The latter allows DNNs to learn from each other.

Nevertheless, neither mode is optimal for the proposed algorithm. In this paper, we aim to transfer knowledge from the 32-bit DNN to the low-bit DNN, however, the latter may fail to absorb knowledge from the former due to its limited representational capacity and slow convergence speed. Simply with teacher-student mode, the 32-bit DNN cannot adjust itself according to the feedback from the low-bit DNN. Simply with mutual learning mode, the 32-bit DNN makes too many concessions and the noise from the low-bit DNN may worsen its performance, which in turn degrades its guidance for the low-bit DNN.

For the reasons above, Gradient Rescaling Module is designed to combine the advantages of both strategies, as is shown in Fig. 1(b)(c). In feed-forward process, Gradient Rescaling Module is simply an identity function and can be ignored. In back-propagation process, gradients of $R(f_{low}, f_{32})$ can be obtained, and Gradient Rescaling Module scales this gradients by the factor of $1-w$ and w for the low-bit DNN and the 32-bit DNN respectively. w is a hyper-parameter, with the range of 0 to 0.5. Then the gradients of Eq.(1) with respect to f_{low} and f_{32} can be denoted as Eq.(5) and Eq.(6):

$$\Delta_{f_{low}} L = \frac{\partial L_{C_{low}}}{\partial f_{low}} + (1-w)\cdot\lambda\cdot\frac{\partial R}{\partial f_{low}} \tag{5}$$

and

$$\Delta_{f_{32}} L = \frac{\partial L_{C_{32}}}{\partial f_{32}} + w\cdot\lambda\cdot\frac{\partial R}{\partial f_{32}}. \tag{6}$$

Following the same notation in Eq.(1), the low-bit DNN absorbs knowledge from the 32-bit DNN through $\frac{\partial R}{\partial f_{low}}$ in Eq.(5), while the latter receives feedback from the former via $\frac{\partial R}{\partial f_{32}}$ in Eq.(6). It can be seen that both teacher-student mode (when $w=0$, f_{32} is not influenced by f_{low}.) and mutual learning mode (when $w=0.5$, f_{32} and f_{low} affect each other to the same extent.) can be viewed as the extreme cases of Gradient Rescaling Module. With proper choice of hyper-parameter w (0 to 0.5), the 32-bit DNN can make 'appropriate' adjustments and concessions based on the feedback, which can relax its regularization on the low-bit DNN.

4 Experiments

In this section, we present experimental analysis on two widely used benchmark datasets, Cifar-10/100 [13] and ImageNet (ILSVRC12) [3]. Cifar-10 has

60,000 32×32 colour images in 10 classes, with 50,000 training images and 10,000 test images. Cifar-100 further divides Cifar-10 dataset to 100 classes. ImageNet (ILSVRC12) is a large scale dataset containing about 1.2 million training images and 50,000 validation images in 1,000 classes.

4.1 Implementation Details

To verify the effectiveness of the proposed algorithm, we experiment with three well-known low-bit DNNs: BNN [8], DoReFa-Net [34] and LQ-Nets [30]. To better expose the problem of space mismatch, we first experiment with simple quantization method BNN and Dorefa-Net, because simple methods suffer more from the mismatch problem. We further verify the performance on LQ-nets, one of the state-of-the-art quantization methods. All experiments are implemented based on the corresponding officially released source codes. To eliminate other distractions, we keep all experiment settings (e.g. network structure, data augmentation, hyper-parameters) consistent between standard low-bit DNNs (baseline) and guided low-bit DNNs (ours). The proposed algorithm has two hyper-parameters, i.e. λ and w. λ is a balancing parameter between empirical loss and regularization loss. We choose proper λ to make the gradients of the two loss functions comparable, so that both of them can contribute to the training. w in Gradient Rescaling Module controls the interaction mode between two DNNs, which ranges from 0 to 0.5. With a larger w, the low-bit DNN is more likely to converge while the 32-bit DNN suffers more noise from the low-bit DNN, which in turn degrades its guidance for the low-bit one. Thus, we start with $w = 0.5$ and reduce it by 10x each time until no significant performance degradation is observed for the 32-bit DNN. In fact, the significant performance of FTL is not due to excessive hyper-parameter adjustment. We set $w = 0.005$ (unless otherwise stated) for all experiments rather than searching for better choice for each model. And λ ranges from 1 to 3.

4.2 Performance Evaluation

The proposed algorithm aims to use the 32-bit DNN to guide training of the low-bit DNN. In this section, we evaluate the performance of the proposed algorithm on Cifar-10/100 [13] and ImageNet [3].

Performance on Cifar-10/100 Table 1 presents the experimental results of Resnet-small model and Vgg-small model. Since advanced low-bit DNNs (e.g. DoReFa-Net [34] and LQ-Nets [30]) have already achieved excellent performance on small scale datasets like Cifar-10/100, we only experiment with BNN [8], a naive quantization algorithm. Table 1 shows that the proposed algorithm brings 0.79% to 3.00% accuracy gains over the baseline for Cifar-10/100, where "W/A/G" denotes the bit-width of weights/activations/gradients.

Table 1. Top-1 accuracy (average of 5 runs) on Cifar10/100. "W/A/G" denotes the bit-width of weights/activations/gradients. "Full precision" denotes the classification accuracy of the 32-bit DNN. "Baseline" represents the low-bit DNN trained without guidance.

Dataset	Model	Bit-width(W/A/G)	Baseline	Ours	Accuracy gain	Full precision
Cifar10	Resnet-small	1/1/32	88.16	90.88	**+2.72**	93.7
	Vgg-small	1/1/32	88.98	89.82	**+0.84**	93.0
Cifar100	Resnet-small	1/1/32	61.20	64.20	**+3.00**	70.9
	Vgg-small	1/1/32	63.76	64.55	**+0.79**	69.1

Table 2. Top-1 accuracy on ImageNet. "Quantizer type" refers to the type of quantizer in certain quantization method. "Method" means different quantization algorithm. "W/A/G" denotes the bit-width of weights/activations/gradients. "Full precision" denotes the classification accuracy of the 32-bit DNN. "Baseline" represents the low-bit DNN trained without FTL.

Method	Model	Bit-width(W/A/G)	Baseline	Ours	Accuracy gain	Full precision
BNN	Alexnet	1/1/32	36.6	37.9	**+1.3**	60.6
	Resnet-18	1/1/32	46.3	46.9	**+0.6**	69.6
DoReFa-Net	Alexnet	1/2/32	52.6	54.0	**+1.4**	59.7
	Alexnet	1/2/4	41.5	44.9	**+3.4**	59.7
	Resnet-18	1/2/32	56.1	57.0	**+0.9**	69.6
	Resnet-18	1/2/4	52.1	53.7	**+1.6**	69.6
LQ-Net	Alexnet	1/2/32	55.7	56.3	**+0.6**	61.8
	Resnet-18	1/2/32	62.6	63.1	**+0.5**	70.3
	Resnet-34	1/2/32	66.3	67.4	**+1.1**	73.8
	Resnet-50	1/2/32	68.7	69.6	**+0.9**	76.4
	Resnet-50	2/2/32	70.3	71.4	**+1.1**	76.4

Performance on ImageNet Table 2 presents the experimental results on the ImageNet. Though various quantization algorithms have been developed, low-bit DNNs still struggle to achieve satisfying performance on such large scale dataset. Hence, we conduct various experiments on ImageNet with different qunatization methods, different models and different bit-width. Experimental results show 0.5% to 3.4% accuracy gains. The enhancement for different models varies a little, which is mainly because we simply assign the same hyper-parameter value for all experiments without further tuning. Note that with quantized gradients of 4-bit (equivalent to adding more noise to gradients), the proposed algorithm can bring more accuracy gains. This phenomenon demonstrates that additional supervision on middle layers can alleviate the negative impact of noise in gradients [28].

Comparison with the State-of-the-art As is introduced above, various techniques have been proposed to better train a low-bit DNN, which can be divided into Incremental quantization [33], Logits Transfer [18, 20], Attention Transfer [17] and Feature Transfer [35, 27]. Since different kinds of methods can be

Table 3. Comparison between the state-of-the-art and ours on Cifar-10. "JGT" denotes the framework proposed in [35]. Since no experiment is conducted based on BNN in the original paper, we implement "JGT" ourselves. All results are average of 5 runs.

Model	Baseline	JGT	**Ours**
Resnet-small	88.16	88.89	**90.88**
Vgg-small	88.98	89.10	**89.82**

Table 4. Comparison between the state-of-the-art and ours on ImageNet with Alexnet. "JGT" and "JGT*" denotes the guided framework proposed in [35] with different training strategy. We directly quote the experimental results for JGT and JGT* from the original paper.

	Top-1	Top-5		Top-1	Top-5
JGT	50.0	74.1	Baseline (JGT)	48.8	72.2
JGT*	51.6	76.2	Baseline (JGT*)	50.9	74.9
Ours	**54.0**	**77.2**	Baseline (Ours)	52.6	76.0

applied simultaneously for further enhancement, it is proper to only make comparison within the same category. This paper focuses on the Feature Transfer approach. Jointly Guided training (JGT) proposed in [35] is the state-of-the-art framework to guide the low-bit DNN via feature maps. In this section, we make a comparison between JGT and FTL. There is no results related to TCO [27] because TCO is specially designed for object detection task. Table 3 and Table 4 demonstrate the results implemented on BNN [8] and DoReFa-Net [34] respectively. No experiment is conducted on LQ-nets [30] because JGT framework is not applicable to low-bit DNNs with adaptive quantizers (e.g. LQ-nets [30]), which is a serious limitation compared with our framework. More than that, experimental results show that FTL can even bring more accuracy gains when training BNN and DoReFa-Net. The gains are mainly from the information-rich continuous-space knowledge.

4.3 Ablation Study

FTL consists of Centered Cosine Distance and Gradient Rescaling Module. To verify their effectiveness, ablation study is conducted based on BNN [8].

Centered Cosine Distance

Results To analyze the effectiveness of centered cosine distance implemented with centered cosine similarity, we make a comparison between the centered cosine distance and the widely used p-norm $\|\cdot\|_p$. Following [12, 35], we take $p = 1$ for $\|\cdot\|_p$, denoted as L1 norm. Since the performance of each distance function is greatly affected by the balancing parameter λ in Eq.(1), we search for the 'optimal' λ for L1 norm in $\{0.04, 0.2, 1, 5, 25, 100\}$ and demonstrate the

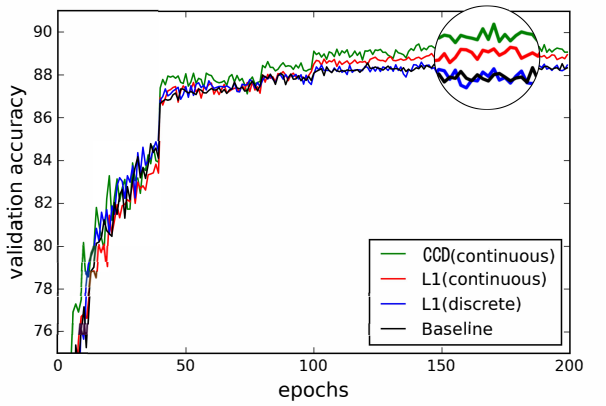

Fig. 2. Effectiveness of centered cosine distance. "CCD(continuous)" represents the proposed centered cosine distance, which is applied in continuous space. "L1(discrete)" and "L1(continuous)" curves represent L1 norm applied in discrete and continuous space, while "Baseline" is standard training of the low-bit DNN. Compared to "Baseline", "MIR(continuous)", "L1(continuous)" and "L1(discrete)" achieve 0.84%, +0.45% and -0.07% accuracy gains respectively.

best performance. In order to reduce random fluctuations in training, each curve is obtained by averaging 5 runs. In our framework, centered cosine distance is applied in continuous space. So we also implement L1 norm in continuous space, which is denoted as "L1(continuous)" in Fig. 2. It can be seen that our centered cosine distance (CCD) obviously outperforms L1 norm when both applied in continuous space.

Besides that, in order to verify whether guiding the low-bit branch in continuous space can bring more performance gains, we also experiment with L1 norm in discrete space, which is represented as "L1(discrete)" in Fig. 2. It can be seen that "L1(discrete)" can hardly bring performance gains over "Baseline" while "L1(continuous)" achieve better performance, which is consistent with our analysis in Introduction section.

Further analysis Below we further analyze why centered cosine distance outperforms p-norm. The feature maps from the low-bit DNN and the 32-bit DNN are denoted as f_{low} and f_{32}, respectively. According to Eq.(2), centered cosine distance inclines f_{low} to mimic the relative numerical relationship in f_{32}. However, regularized by p-norm, f_{low} tends to learn the magnitude of each element in f_{32}, which is a stronger regularization than centered cosine distance. In consequence, p-norm may affect the overall distribution of f_{low}. In low-bit DNNs, the distribution of f_{low} is of vital importance and it should match the pre-defined quantizer. Otherwise the quantization noise will increase and degrade the performance of low-bit DNNs.

An empirical experiment is conducted to demonstrate the difference between centered cosine distance and p-norm. We initialize a low-bit DNN with a pre-

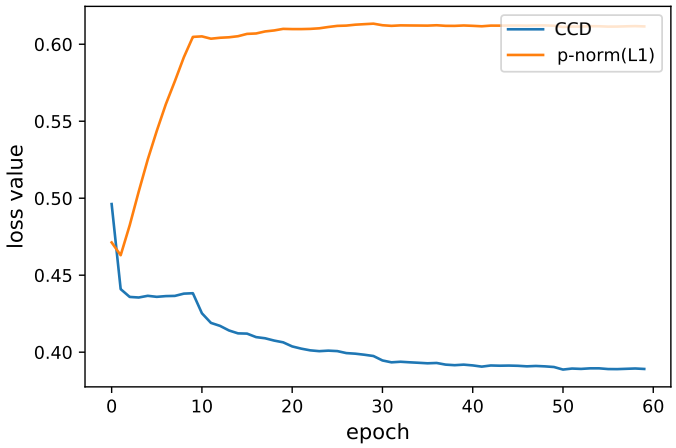

Fig. 3. Curves of centered cosine distance (CCD) and p-norm during fine-tuning. Gradients of both regularization are not back-propagated. Results are averaged of 5 runs.

trained 32-bit DNN, and fine-tune it without any feature guidance. Such initialization is to make f_{low} and f_{32} have certain similarities at the beginning. As the fine-tuning progresses, the low-bit DNN gradually adapts itself to the pre-defined quantizer. We report the loss value of centered cosine distance and p-norm between f_{low} and f_{32} at every epoch in Fig. 3. Note that no gradients of both guidance are back-propagated. We observe an interesting phenomenon that p-norm curve shows an upward trend, which suggests that p-norm negatively impacts the adaptation of low-bit DNNs to the pre-defined quantizer. In contrast, centered cosine distance decreases as the fine-tuning progresses. Since we do not back-propagate the gradients of the centered cosine distance (CCD), intuitively, the curve of CCD should increase or remain stable at best. However, only trained with empirical loss, the feature maps of the low-bit branch can also minimize its centered cosine distance with the 32-bit branch, which provides some insights for explaining the training of low-bit DNNs. In Fig. 3, the CCD between two branches is converged to ≈ 0.4. In the proposed framework, since we back-propagate the gradient of CCD, it can further decrease to ≈ 0.2. Due to their differences between CCD and p-norms, the former can bring more accuracy gains to low-bit DNNs.

Gradient Rescaling Module In this section, we analyze whether Gradient Rescaling Module can explore better interaction modes between two DNNs than teacher-student mode [7] and mutual learning mode [31]. We conduct experiments on Cifar-10 with vgg-small variant (only **0.5**$\times$ channel numbers to save training time). All experimental results are average of 5 runs. As is shown in Fig. 4, we change w (hyper-parameter in Gradient Rescaling Module) from 1e-5

to 5e-1 while other conditions remain unchanged, which corresponds to point "A" to point "G" respectively. Among them, G ($w = 0.5$) is equal to mutual learning mode, and A ($w \approx 0$) can be approximately considered as teacher-student mode. Both modes fail to bring performance gains. This can be explained from two aspects. On the one hand, low-bit DNNs have much smaller representational capacity than 32-bit DNNs. Merely with teacher-student mode ($w = 0$), the low-bit DNN fails to mimic the feature maps from the latter since the 32-bit one makes no concession and adjustments. On the other hand, with mutual learning mode ($w = 0.5$), the 32-bit DNN absorbs a large amount of feedback (can be viewed as noise) from the low-bit DNN, which in turn worsens its guidance to the latter. However, our Gradient Rescaling Module enables the exploration (different choices of w) for better interaction modes (e.g. point "C" and "D") instead of having to choose between "A" and "G".

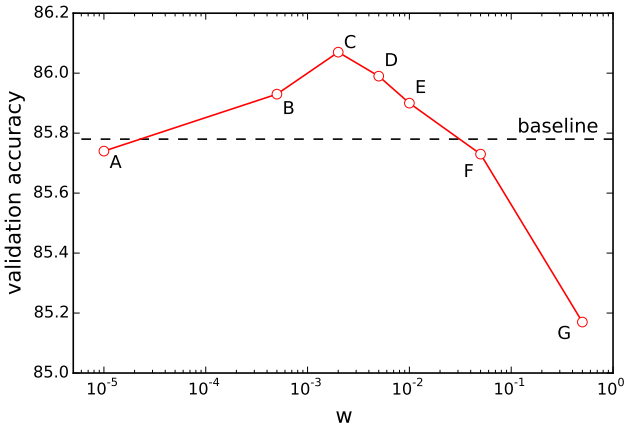

Fig. 4. Impact of Gradient Rescaling Module. The X-axis is the hyper-parameter w in Gradient Rescaling Module. A to G represent w =1e-5, 5e-4, 2e-3, 5e-3, 1e-2, 5e-2 and 5e-1 respectively. "Baseline" represents directly training for the low-bit DNN. Compared to "Baseline", "A" to "G" achieve -0.04%, +0.15%, +0.29%, +0.21%, +0.12%, -0.05%, -0.61% accuracy gains respectively.

4.4 Combination with other Methods

Except the proposed algorithm, there exist other methods to assist training of low-bit DNNs such as Logits Transfer and fine-tuning from pre-trained 32-bit DNN. Since these techniques and our FTL enhance performance of low-bit DNNs from different aspects, we explore whether combining FTL with these methods leads to better performance. We conduct experiments on Cifar-10 with Vgg-small model, as is shown in Table 5. For "Fine-tuning" and "Fine-tuning + ours" method, we only train for 60 epochs (200 in others) since it has better

Table 5. Combination with other Methods. "Logits Transfer" denotes Knowledge Distillation propsed in [7]. "Fine-tuning" denotes fine-tuning from a pre-trained 32-bit DNN. All methods are trained for 200 epochs except "Fine-tuning" and "Fine-tuning + Ours".

Method	Validation accuracy
Baseline	88.98
Ours	**89.82**
Fine-tuning (60 epochs)	89.31
Fine-tuning + Ours (60 epochs)	**89.62**
Logits Transfer	89.49
Logits Transfer + Ours	**89.96**

initialization. All experimental results are average of 5 runs. It can be seen that the proposed algorithm can be combined with other methods for fast training or better performance.

5 Conclusion

We analyze the difficulty in optimizing low-bit DNNs and propose a universal framework named FTL to assist its training. In FTL, an auxiliary 32-bit DNN is constructed to provide middle layer supervision for the low-bit one. Different from traditional discrete space supervision, we make two DNNs interact in continuous space. Considering the quantization operation in the low-bit DNN, we guide the low-bit DNN with centered cosine distance, which has better performance compared to empirically used p-norms. Besides, Gradient Rescaling Module is designed to coordinate the training of two DNNs, which can combine the advantages of teacher-student mode and mutual learning mode.

Experimental results suggest that with FTL, the classification accuracy of three different low-bit DNNs increases by 0.5% to 3.4%. Moreover, our framework can be well combined with other existing methods (e.g. Knowledge Distillation) to train a more accurate low-bit DNN. For future work, we plan to provide supervision for multiple middle layers to better guide training. Furthermore, the 'attention' mechanism can also be considered to improve the quality of the guidance.

Acknowledgement

Acknowledgement: This work is supported by the National Key Research and Development Program of China (No. 2019YFB1804304), SHEITC (No. 2018-RGZN-02046), 111 plan (No. BP0719010), and STCSM (No. 18DZ2270700), and State Key Laboratory of UHD Video and Audio Production and Presentation.

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9163–9171 (2019)
2. Cai, Z., He, X., Sun, J., Vasconcelos, N.: Deep learning with low precision by half-wave gaussian quantization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5918–5926 (2017)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
4. Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. In: International Conference on Machine Learning. pp. 1607–1616 (2018)
5. Gel'Fand, I., Yaglom, A.: Abouta random function contained in another such function". Eleven Papers on Analysis, Probability and Topology **12**, 199 (1959)
6. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1921–1930 (2019)
7. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
8. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks. In: Advances in neural information processing systems. pp. 4107–4115 (2016)
9. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
10. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2704–2713 (2018)
11. Jung, S., Son, C., Lee, S., Son, J., Han, J.J., Kwak, Y., Hwang, S.J., Choi, C.: Learning to quantize deep networks by optimizing quantization intervals with task loss. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4350–4359 (2019)
12. Kim, J., Park, S., Kwak, N.: Paraphrasing complex network: Network compression via factor transfer. In: Advances in Neural Information Processing Systems. pp. 2760–2769 (2018)
13. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)
14. Li, X., Xiong, H., Wang, H., Rao, Y., Liu, L., Huan, J.: DELTA: DEEP LEARNING TRANSFER USING FEATURE MAP WITH ATTENTION FOR CONVOLUTIONAL NETWORKS. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=rkgbwsAcYm
15. Lin, X., Zhao, C., Pan, W.: Towards accurate binary convolutional neural network. In: Advances in Neural Information Processing Systems. pp. 345–353 (2017)
16. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2604–2613 (2019)

17. Martinez, B., Yang, J., Bulat, A., Tzimiropoulos, G.: Training binary neural networks with real-to-binary convolutions. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=BJg4NgBKvH
18. Mishra, A., Marr, D.: Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. arXiv preprint arXiv:1711.05852 (2017)
19. Nagel, M., Baalen, M.v., Blankevoort, T., Welling, M.: Data-free quantization through weight equalization and bias correction. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1325–1334 (2019)
20. Polino, A., Pascanu, R., Alistarh, D.: Model compression via distillation and quantization. arXiv preprint arXiv:1802.05668 (2018)
21. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European Conference on Computer Vision. pp. 525–542. Springer (2016)
22. Sakr, C., Kim, Y., Shanbhag, N.: Analytical guarantees on numerical precision of deep neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 3007–3016. JMLR. org (2017)
23. Sun, X., Choi, J., Chen, C.Y., Wang, N., Venkataramani, S., Srinivasan, V.V., Cui, X., Zhang, W., Gopalakrishnan, K.: Hybrid 8-bit floating point (hfp8) training and inference for deep neural networks. In: Advances in Neural Information Processing Systems. pp. 4901–4910 (2019)
24. Sun, X., Choi, J., Chen, C.Y., Wang, N., Venkataramani, S., Srinivasan, V.V., Cui, X., Zhang, W., Gopalakrishnan, K.: Hybrid 8-bit floating point (hfp8) training and inference for deep neural networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 4900–4909. Curran Associates, Inc. (2019), http://papers.nips.cc/paper/8736-hybrid-8-bit-floating-point-hfp8-training-and-inference-for-deep-neural-networks.pdf
25. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1365–1374 (2019)
26. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4933–4942 (2019)
27. Wei, Y., Pan, X., Qin, H., Ouyang, W., Yan, J.: Quantization mimic: Towards very tiny cnn for object detection. In: European Conference on Computer Vision. pp. 274–290. Springer (2018)
28. Yin, P., Lyu, J., Zhang, S., Osher, S., Qi, Y., Xin, J.: Understanding straight-through estimator in training activation quantized neural nets. arXiv preprint arXiv:1903.05662 (2019)
29. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928 (2016)
30. Zhang, D., Yang, J., Ye, D., Hua, G.: Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In: European Conference on Computer Vision. pp. 373–390. Springer (2018)
31. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4320–4328 (2018)
32. Zhao, R., Hu, Y., Dotzel, J., De Sa, C., Zhang, Z.: Improving neural network quantization without retraining using outlier channel splitting. In: International Conference on Machine Learning. pp. 7543–7552 (2019)

33. Zhou, A., Yao, A., Wang, K., Chen, Y.: Explicit loss-error-aware quantization for low-bit deep neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
34. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016)
35. Zhuang, B., Shen, C., Tan, M., Liu, L., Reid, I.: Towards effective low-bitwidth convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7920–7928 (2018)