Stochastic Precision Ensemble: Self-Knowledge Distillation for Quantized Deep Neural Networks

Yoonho Boo,[1] **Sungho Shin,**[1] **Jungwook Choi,** [2] **and Wonyong Sung** [1]

[1] Seoul National University, Seoul, Korea
[2] Hanyang University, Seoul, Korea
dnsgh337@snu.ac.kr, sungho.develop@gmail.com, choij@hanyang.ac.kr, wysung@snu.ac.kr

Abstract

The quantization of deep neural networks (QDNNs) has been actively studied for deployment in edge devices. Recent studies employ the knowledge distillation (KD) method to improve the performance of quantized networks. In this study, we propose stochastic precision ensemble training for QDNNs (SPEQ). SPEQ is a knowledge distillation training scheme; however, the teacher is formed by sharing the model parameters of the student network. We obtain the soft labels of the teacher by randomly changing the bit precision of the activation stochastically at each layer of the forward-pass computation. The student model is trained with these soft labels to reduce the activation quantization noise. The cosine similarity loss is employed, instead of the KL-divergence, for KD training. As the teacher model changes continuously by random bit-precision assignment, it exploits the effect of stochastic ensemble KD. SPEQ outperforms the existing quantization training methods in various tasks, such as image classification, question-answering, and transfer learning without the need for cumbersome teacher networks.

Introduction

Deep neural networks (DNNs) have achieved remarkable accuracy for tasks in a wide range of applications, including image processing (He et al. 2016a), machine translation (Gehring et al. 2017), and speech recognition (Zhang et al. 2017). These state-of-the-art neural networks use very deep models, consuming hundreds of ExaOps of computation during training and GBytes of storage for model and data. This complexity poses a tremendous challenge for widespread deployment, especially in resource-constrained edge environments, leading to a plethora of explorations in model compression that minimize memory footprint and computational complexity while attempting to preserve the performance of the model. Among them, research on quantized DNNs (QDNNs) focuses on quantizing key data structures, namely weights and activations, into low-precision. Hence, we can save memory access overhead and simplify the arithmetic unit to perform reduced-precision computation. There have been extensive studies on QDNNs (Fengfu, Bo, and Bin 2016; Courbariaux, Bengio, and David 2015;

Choi et al. 2018; Hou and Kwok 2018), but most of them suffer from accuracy loss due to quantization.

To enhance the performance of low-capacity models, knowledge distillation (Hinton, Vinyals, and Dean 2015; Bucilu, Caruana, and Niculescu-Mizil 2006) (KD) has been widely adopted. KD employs a more accurate model as a teacher network to guide the training of the student model. For the same input, the teacher network provides its prediction as a soft label, which can be further considered in the loss function to guide the training of the student network. In the case of QDNNs, the quantized student network can compensate for its accuracy loss via supervision of the teacher model (Mishra and Marr 2018; Polino, Pascanu, and Alistarh 2018; Shin, Boo, and Sung 2019; Kim et al. 2019). However, the need for large and high-performance teacher models introduces significant overhead when applying KD. In particular, KD has not been successfully employed in the emerging study of on-device training for model adaptation and transfer learning, since the memory-intensive teacher models may not be available once the quantized models are deployed.

In this work, we propose a new practical approach to KD for QDNNs, called stochastic precision ensemble training for QDNNs (SPEQ). SPEQ is motivated by an inspiring observation about activation quantization. Table 1 shows that the accuracy of the WFA2 (float weight and 2-bit activation) model improves as the activation precision increases. However, the W2AF (2-bit weight and float activation) model shows the opposite characteristic. The accuracy drops as the weight precision increases for inference. This simple experiment reveals interesting insights: the activation quantization mostly adds noise to the decision boundary (Boo, Shin, and Sung 2020). Therefore, inference with various activation precision results in selective removal of such noise, leading to diverse guidance that can be exploited for self knowledge distillation.

In SPEQ, we form a teacher network that shares the quantized weights with the student but employs different bit precision for activation. The clipping levels of activation are also shared. In fact, the activation precision for the teacher is randomly selected between the low and high precision, such as 2 and 8-bit. Since the teacher stochastically applies the target low-bit activation quantization for its soft label computation, it can experience the impact of quantization for the

Trained precision	Test accuracy (%) / Inference precision			
2-bit W, float A (W2AF)	65.74 / W2AF	58.01 / W4AF	55.85 / W8AF	54.70 / WFAF
Float W, 2-bit A (WFA2)	66.93 / WFA2	68.48 / WFA4	68.77 / WFA8	68.71 / WFAF

Table 1: CIFAR100 test accuracy (%) in higher precision on the quantized model. ResNet20 is trained with 2-bit weight / float activation and float weight / 2-bit activation. (Details in Appendix A.)

guidance. Furthermore, we reveal that the cosine similarity loss is essential for distilling the knowledge of the teacher of stochastic quantization to the low-precision student.

Although this form of guidance resembles KD, there is a significant difference in that the same model is shared and any other auxiliary models, such as large teacher networks, are unnecessary. The forward-pass computation of the teacher and student in SPEQ can be performed economically as the same weight parameters can be loaded only once. Therefore, the SPEQ can improve the performance much without the overhead of teacher-model search or hyper-parameter tuning needed for conventional KD. Furthermore, since the stochastic precision ensemble provides distinctive knowledge, SPEQ can be combined with the conventional KD method to further improve the performance of the target QDNNs.

We demonstrate the superior performance and efficiency of our SPEQ on various applications, including CIFAR10/CIFAR100/ImageNet image classification and also transfer learning scenarios such as BERT-based question-answering and flower classification.

The contributions of our work are summarized as follows:

- We propose a new practical KD method called SPEQ that can enhance the accuracy of QDNNs QDNNs employing low-precision bit-widths for weights and activation signals. This method can yield better results compared to conventional KD-based QDNN optimization that utilizes large teacher models.

- We suggest cosine similarity as an essential loss function to effectively distill the knowledge of activation quantization in SPEQ training.

- We demonstrate that the proposed method outperforms the existing KD methods for training QDNNs with lower training overhead. We confirm this on various models and tasks including image classifications, question answering, and transfer learning.

- We show that the proposed method can be combined to the conventional KD method with a large teacher to further improve the performance of the target model.

Related Works

Quantization of Deep Neural Networks

QDNNs have been studied for a long time. Early works suggested stochastic gradient descent (SGD)-based training for QDNNs to restore the performance reduced by the quantization error (Courbariaux, Bengio, and David 2015; Hwang and Sung 2014; Zhu et al. 2017). The quantized SGD training maintains both full-precision and quantized weights.

Full-precision weights are exploited to accumulate the gradients, and the quantized weights are used for computing forward and backward propagation. Several techniques have been combined with the quantized SGD algorithm, which include data distribution (Zhou et al. 2017), stochastic rounding (Gupta et al. 2015), weight cluster (Park, Ahn, and Yoo 2017), trainable quantization (Zhang et al. 2018), fittable quantization scale (Cai et al. 2017), pow2-ternaization (Ott et al. 2016), stochastic weight averaging (Shin, Boo, and Sung 2020), increasing the size of the neural network (Kapur, Mishra, and Marr 2017), and quantization interval learning (Jung et al. 2019). A recent study suggested that quantization errors for weight and activation are different (Boo, Shin, and Sung 2020). Activation quantized models are known to be more vulnerable to the adversarial noise (Lin, Gan, and Han 2019). Architectural modifications of increasing the width or moving the location of activation and batch normalization have also been studied (Zagoruyko and Komodakis 2016; He et al. 2016b). In particular, increasing the number of parameters in CNNs reduces the quantization sensitivity (Mishra et al. 2017). However, considering the purpose of model compression, the number of parameters needs to be constrained.

Knowledge Distillation for Quantization

KD is a method to improve the accuracy of a target model (called a student) by transferring better representation power (i.e., "knowledge") of a larger or more complex model (called a teacher) (Hinton, Vinyals, and Dean 2015; Bucilu, Caruana, and Niculescu-Mizil 2006). Recently, several papers have adopted KD to restore the accuracy loss due to the quantization error of reduced-precision inference (Zhuang et al. 2018; Polino, Pascanu, and Alistarh 2018; Mishra and Marr 2018; Shin, Boo, and Sung 2019; Kim et al. 2019). Apprentice (Mishra and Marr 2018) proposed several approaches to apply KD for enhancing the accuracy of the quantized models. The importance of the hyperparameters of KD was studied in (Shin, Boo, and Sung 2019). More recently, quantization aware KD (Kim et al. 2019) (QKD) has been suggested, wherein the three training phases are coordinated as self-studying, co-studying, and tutoring. They train the full-precision larger teacher model using the soft labels of quantized student network to make the teacher understand the quantization errors of the student model. SP-Net (Guerra et al. 2020) also adopted self distillation but it focused on training models to robustly operate at various precisions. They employed only the full-precision pass as the teacher to improve accuracy for the other precision settings. Several studies have adopted deterministic self-distillation (Zhang et al. 2019a; Li et al. 2019; Phuong and Lampert 2019; Yu
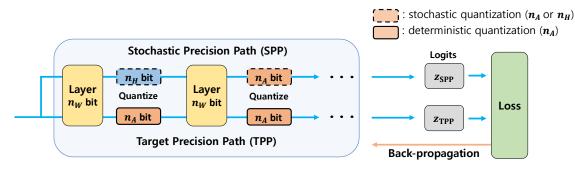
Figure 1: Structure of the proposed SPEQ training scheme for QDNNs. The QDNNs are trained for the target precision n_A through the 'target precision path'. The 'stochastic precision path' produces the teacher logits, $\mathbf{z}_{\text{SPP}}$ using the same model but with randomly assigned quantization precision for activation at every iteration. Note that the weights in the model are quantized to n_W bits.

and Huang 2019; Zhang et al. 2019b).

The difference between the proposed method and the previous works is that the teacher is formed by sharing the student model and assigning stochastic bit precision to activation. There are two main advantages of this method: the teacher information contains the quantization noise induced in the target QDNN by model sharing (better performance) and pretrained teacher models or auxiliary training parameters are unnecessary (lower training cost).

Stochastic Precision Ensemble Training for QDNNs

Stochastic Precision Self-Distillation with Model Sharing

Changing the activation quantization precision in the same model affects the amount of noise injected into the model, as shown in Table 1. That is, the outputs obtained through high-precision activation have information when the model operates without noise.

The training procedure of the SPEQ is illustrated in Figure 1. Two outputs are computed through different paths using the same parameters. Note that the initial quantized weights and clipping levels for activation are determined using conventional QDNN optimization methods (Jung et al. 2019; Choi et al. 2018). The details of the employed quantization method are shown in Appendix B. The first output logits, $\mathbf{z}_{\text{TPP}}$, are obtained through the target precision path by quantizing the activation outputs to n_A bits. The goal of the SPEQ is to increase the performance of the QDNN with this target precision path. The second output logits, $\mathbf{z}_{\text{SPP}}$, are computed by quantizing the activation outputs using the stochastic bit precision, n_{SPP} which is defined as follows:

$$n_{\text{SPP}}^l = \begin{cases} n_A & \text{with probability} \quad u \\ n_H & \text{with probability} \quad 1-u, \end{cases} \quad (1)$$

where l denotes the layer index, n_A is the target precision, n_H is a precision higher than the target precision, and u is a quantization probability for the stochastic quantization path.

We set the high precision, n_H, to 8 bits. The impact of u is discussed in the next section. For readability, we denote the set of n_{SPP}^l as $\mathbf{n}_{\text{SPP}}$, that is, $\mathbf{n}_{\text{SPP}} = \{n_{\text{SPP}}^1, ...n_{\text{SPP}}^L\}$.

The output probability, $\mathbf{p}((z), \mathcal{T})$, is computed using the softmax operation with temperature, $\mathcal{T}$. The temperature softens the distribution of the softmax outputs by dividing the output logits (Hinton, Vinyals, and Dean 2015). Note that the soft labels, $\mathbf{p}(\mathbf{z}_{\text{SPP}}, \mathcal{T})$, are produced while sharing the parameters. Thus, they contain information of the quantization noise of the target model. We train the QDNN using these soft labels to reduce the activation quantization noise. The loss for the SPEQ training is the sum of the cross-entropy loss, $\mathcal{L}_{CE}$, and the cosine similarity loss, $\mathcal{L}_{CS}$, as follows.

$$\begin{aligned} \mathcal{L}_{SPEQ} = &\mathcal{L}_{CE}(y, \mathbf{p}(\mathbf{z}_{\text{TPP}}, 1)) \\ &+ \mathcal{L}_{CS}(\mathbf{p}(\mathbf{z}_{\text{SPP}}, \mathcal{T}), \mathbf{p}(\mathbf{z}_{\text{TPP}}, \mathcal{T})) \times \mathcal{T}^2. \end{aligned} \quad (2)$$

The effects of the cosine similarity loss function are discussed in Section . Note that the $\mathbf{z}_{\text{SPP}}$ is only used to produce the soft label for the $\mathcal{L}_{CS}$, thus, the back-propagation error only flows through the target precision path, as shown in Figure 1. Therefore, the only computational overhead for a training step is the computation of $\mathbf{z}_{\text{SPP}}$ by forward propagation. The SPEQ is based on the KD training but has the advantage that no other auxiliary model is required.

The proposed method can also adopt a larger teacher model to further improve the performance. In this case, the outputs through the SPP can be seen as the outputs of the teacher-assistant (Mirzadeh et al. 2019). The training loss with a larger teacher is computed as follows:

$$\mathcal{L}_{\text{T}} = \mathcal{L}_{KL}(\mathbf{p}(\mathbf{z}_{\text{T}}, \mathcal{T}), \mathbf{p}(\mathbf{z}_{\text{TPP}}, \mathcal{T})) \times \mathcal{T}^2 \quad (3)$$

$$\mathcal{L}_{SPEQ+KD} = \lambda \mathcal{L}_{SPEQ} + (1-\lambda)\mathcal{L}_T. \quad (4)$$

$\mathbf{z}_{\text{T}}$ is the logit from a large teacher model and thus $\mathcal{L}_T$ represents the distllation loss from the larger teacher to the shared model with the target precision.

Stochastic Ensemble Learning

Intuitively, using a good teacher when training QDNN with KD will improve the performance (Tang et al. 2020). In
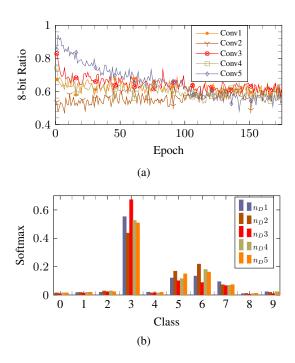
(a)

(b)

Figure 2: (a): The ratio of selected 8-bit precision when trained with the greedy strategy. (b) Softmax distributions generated by the stochastic precision path with the same images.

the proposed method, the activation quantization precision, $\mathbf{n}_{\text{SPP}}$, for each layer was determined stochastically. In this case, the total number of combinations for $\mathbf{n}_{\text{SPP}}$ is 2^L, and among them there will be the solution, $\mathbf{n}_{\text{SPP}}^*$, which shows the best performance. However, finding this solution is not practical for DNNs because 2^L inferences are needed for the exhaustive search. To investigate the performance of the best solution, we design a shallow CNN that consists of five convolutional layers and train the model by selecting the $\mathbf{n}_{\text{SPP}}$ according to the true label, y, for each step as follows:

$$\mathbf{n}_{\text{SPP}}^* = \underset{\mathbf{n}}{\operatorname{argmin}}\, \mathcal{L}_{CE}(y, \mathbf{z}_{\text{SPP}}|\mathbf{n}). \qquad (5)$$

Since the experiment is performed on a five-layer CNN, $L = 5$, we employ the greedy strategy that finds the $\mathbf{n}_{\text{SPP}}^*$ by inferencing the model 2^5 times with different combinations of the quantization precision. Note that $\mathbf{n}_{\text{SPP}}^*$ can change for each training step. The target model is trained using the soft label obtained with $\mathbf{n}_{\text{SPP}}^*$. Figure 2 (a) shows how the ratio of 8-bit selection changes during training for each layer. The model is pretrained to the 2-bit weights and activations and the target precision is also 2 bits. The floating-point and 2-bit models show accuracies of 89.9% and 87.8%, respectively.

The solution of Eq. (5) is not always 8-bit even at the beginning of the training. Note that the results in Table 1 show that using higher precision for the activation can achieve higher average accuracy. For each iteration, however, choosing 8-bit activation may not show the lowest loss. More importantly, the ratio of 8-bit selection decreases to 0.6 as the training progresses. This indicates that the best-performing
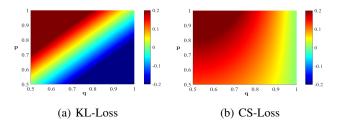

(a) KL-Loss   (b) CS-Loss

Figure 3: The gradients of the ground-truth logit according to the loss type. The x- and y-axes are the probabilities of the student and teacher for the logit, respectively.

solution, $\mathbf{n}_{\text{SPP}}^*$, selects 2- and 8-bit almost uniformly. As a result, the test accuracy of the 2-bit model with greedy training is 88.4%, which is better than the performance of always choosing 8-bit, 88.1%.

Another advantage of the SPEQ is that it has the effect of ensemble learning. By the stochastic selection of bit precision, soft labels with different distributions can be created for one training sample. In this case, the diversity of soft labels should be large enough to obtain the effect of the ensemble well (Chen et al. 2020). Figure 2 (b) shows the computed soft labels by quantizing the activation outputs of the ResNet20 with different bit precisions for a single training sample in the CIFAR10 dataset. Although we extract soft labels from the same parameters, the distribution of the soft labels varies according to the activation precision.

Based on our analysis, we apply the SPEQ training method with uniform n_A-bit and 8-bit selection probabilities to increase diversity. The sensitivity of bit-precision candidate or the quantization probability, u, is also examined.

Cosine Similarity Learning

In many KD approaches, KL-divergence is commonly used as a loss function to reflect the guidance of the teacher. In the setting of SPEQ, however, we claim that the cosine similarity loss (CS-Loss) functions better than KL-divergence loss (KL-Loss). The main difference is that the teacher in SPEQ may not be more reliable than the student. Note that activations are randomly quantized in SPEQ, thus the output prediction of the teacher might be significantly affected by the quantization noise. In this setting, it is important to reflect the guidance of the teacher selectively, as there is no guarantee that the teacher's prediction is more accurate than the student's. In this section, we explain that CS-Loss has such capability whereas KL-Loss does not.

To understand the situation more concretely, we compare the back-propagation errors (i.e., gradients w.r.t. each logit) for the two loss functions. Note that the student model is guided to increase (or decrease) the logit if the corresponding gradient is negative (or positive). When the predictions of the teacher and the student are $\mathbf{p}$ and $\mathbf{q}$, respectively, the KL-Loss and its back-propagation error for the ith logit, z_i,

Method (2-bit baseline accuracy: 90.73%)	KL-Loss	CS-Loss
KD w/ full precision ResNet20 as teacher	91.24±0.06	91.22±0.10
SPEQ ($u = 0$, always choose 8-bit for soft labels)	91.22±0.16	91.18±0.07
SPEQ ($u = 0.5$, 2-bit or 8-bit for soft labels)	90.83±0.07	**91.44±0.04**

Table 2: Comparison of the test accuracy of 2-bit ResNet20 on CIFAR10 according to the loss function for KD. The cosine similarity loss (CS-Loss) suits better than the KL-divergence loss (KL-Loss) for the proposed SPEQ method. The average test accuracy of 5 repeated experiments is reported with the standard deviation.

u	0.0	0.1	0.2	0.3	**0.4**	**0.5**	**0.6**	0.7	0.8	0.9	1.0	Mix
Test Acc.	91.22	91.23	91.22	91.29	**91.39**	**91.44**	**91.43**	91.21	91.24	90.96	90.74	91.23

Table 3: 2-bit ResNet20 test accuracy according to the quantization probability for the stochastic precision path, u. The average test accuracy of 5 repeated experiments is reported. The result of 'Mix' is obtained by selecting all precisions from 2 to 8 bits uniformly.

is represented as follows (Hinton, Vinyals, and Dean 2015):

$$\mathcal{L}_{KL}(\mathbf{p}, \mathbf{q}) = -\sum_{i=0}^{C-1} p_i \log \frac{q_i}{p_i}, \qquad (6)$$

$$\frac{\partial \mathcal{L}_{KL}}{\partial z_i} = q_i - p_i. \qquad (7)$$

Eq. (7) indicates that the $\mathcal{L}_{KL}$ produces back-propagation errors in the direction of decreasing the difference between the p_i and q_i. That is, KL-Loss guides the student to always follow the teacher. Such guidance is regarded as "re-weighting" (Tang et al. 2020), but it is helpful under a condition that the teacher's prediction is more confident than the student's. Since the teacher in SPEQ is not as reliable as the large teacher models in typical KD methods, the gradients from KL-Loss can be misleading.

In comparison, CS-Loss between predictions of the teacher and the student after the normalization is given as follows:

$$\mathcal{L}_{CS}(\mathbf{p}, \mathbf{q}) = 1 - \mathbf{p} \cdot \mathbf{q}, \qquad (8)$$

$$\frac{\partial \mathcal{L}_{CS}}{\partial z_i} = -\sum_{j=0}^{C-1} p_j (q_j \delta_{ij} - q_j q_i). \qquad (9)$$

The gradients of $\mathcal{L}_{CS}$ are more cognizant of the confidence of both the teacher and the student. Assume that the ith label is the ground-truth and the teacher's prediction is also confident about it, i.e. $p_i \gg p_j$ ($i \neq j$), Eq. (9) is approximated as follows:

$$\frac{\partial \mathcal{L}_{CS}}{\partial z_i} \approx -p_i q_i (1 - q_i). \qquad (10)$$

Eq. (10) indicates that the gradients is proportional to $q_i(1 - q_i)$. This is particularly helpful when the confidence of the student's prediction is not high; when $0 < q_i < 1$, the student is guided to increase the confidence for q_i. If the student's prediction has high confidence, the gradients become zero so that they will be highly penalized by the CE-Loss if the prediction is wrong.

In addition, as the prediction of the teacher itself is ambiguous, the back-propagation error decreases. When the prediction of the teacher goes to a uniform distribution, Eq. (9) can be approximated as:

$$\frac{\partial \mathcal{L}_{CS}}{\partial z_i} \approx -p_i q_i (1 - \sum_{j=0}^{C-1} q_i) = 0 \qquad (11)$$

Note that the gradients in this case are almost zero, implying that the CS-Loss will be neglected when the confidence of the teacher's prediction is small. Additional case-studies and the examples are provided in Appendix C.

The impact of the relationship between the teacher and the student predictions to the gradients is illustrated in Figure 3. As can be seen, the gradient of KL-Loss flips its direction when the student's confidence is higher than the teacher. This is detrimental for SPEQ-based knowledge distillation as the prediction of the teacher is prone to noise. Whereas, the CS-Loss allows selective adoption of the teacher's information; the gradients guide to follow the teacher more if it has high confidence. If not, the guidance is neglected.

To distinguish the effects of two loss terms on general and our KD, we trained the 2-bit ResNet20 using various teachers as shown in Table 2. The baseline model is trained without applying KD. When the output probability of the teacher is computed using a better teacher model or a deterministic precision on the shared model, the two loss terms show almost the same result. However, when the teacher outputs are generated while changing the bit precision, only the cosine loss shows the performance improvement compared to the baseline model.

Experimental Results

The training procedures in our experiments consist of three steps: train the floating-point DNN (pretrain), train the QDNN to the target precision initialized from the floating-point parameters (retrain (Hwang and Sung 2014; Choi et al. 2018)), and train the QDNN using the SPEQ method initialized with the retrained parameters. The details of the experimental settings for each task explained in Appendix D.

Methods	Precision (W / A)	CIFAR10		CIFAR100	
		VGG16	ResNet20	ResNet32	MobileNetV2
Baseline	F / F	93.6	92.1	70.3	76.8
Retrain	2 / 2	92.5	90.7	66.9	73.0
PACT-SWAB-8brc (Choi et al. 2019)	2 / 2	-	90.7	-	-
QKD (Kim et al. 2019)	2 / 2	-	90.5	66.4	-
SPEQ	2 / 2	**93.1**	**91.4**	**69.1**	**74.4**
Retrain	F / 2	92.9	91.8	67.9	74.5
SPEQ	F / 2	**93.5**	**92.1**	**69.7**	**75.2**

Table 4: Test accuracy (%) of quantized CNNs on CIFAR10 and CIFAR100 datasets. 'F' denotes the floating-point precision.

2-bit weights / 2-bit activations				Float weights / 2-bit activations		
Method	AlexNet (60.8)	ResNet18 (70.3)	ResNet34 (73.6)	Method	AlexNet (60.8)	ResNet18 (70.3)
DoReFa† (Zhou et al. 2016)	46.4	62.6	-	BalancedQ (Zhou et al. 2017)	56.5	61.1
QIL (Jung et al. 2019)	58.1	65.7	70.6	QN (Yang et al. 2019)	-	65.7
PACT_SWAB (Choi et al. 2019)	57.2	67.0	-	DoReFa† (Zhou et al. 2016)	54.1	66.9
Retrain	56.9	66.6	70.5	PACT (Choi et al. 2018)	54.9	67.5
SPEQ	**59.3**	**67.4**	**71.5**	SPEQ	**60.8**	**68.4**

Table 5: Top-1 validation accuracy (%) on the ImageNet dataset. Values in the parentheses are the accuracy of pretrained floating-point models. Results with the symbol † are from (Choi et al. 2018).

Results on the CIFAR10 / CIFAR100 Datasets

We first studied how the stochastic quantization probability, u, affects the performance of the SPEQ method. To this end, we trained 2-bit quantized ResNet20 models with various values of u from 0.0 to 1.0 on the CIFAR10 dataset. The results are reported in Table 3. It should be noted that 0.0 and 1.0 of u indicate that the stochastic precision path selects only 8- and 2-bit precisions, respectively. The best test accuracy is observed when u is between 0.4 and 0.6. This result indicates that the SPEQ shows the best performance when the stochastic precision path is selected to some degree evenly rather than being biased to either precision. For comparison, we employed SPEQ by uniformly selecting all precisions from 2 to 8 bits. The result is shown as 'Mix' in Table 3. The result is 0.21% lower than that of the training using only two precisions, 2 and 8 bits. Note that Quantization errors are the largest in 2-bit precision and the lowest in 8-bit precision. Rather than intermediate precision, selecting precision either the lowest or highest precision increases the diversity of teacher outputs, which has a great influence on performance (Chen et al. 2020). The uniform selection between the lowest and highest precision results in a better ensemble effect by increasing the diversity of teacher outputs. For all the rest of the experiments, we set the quantization probability, u, to 0.5.

We evaluated the proposed SPEQ scheme using the CIFAR10 and CIFAR100 datasets. The performance of the SPEQ and existing methods are shown in Table 4. The test accuracy of 2-bit ResNet20 before applying SPEQ, denoted as 'Retrain', is 90.7% on the CIFAR10 dataset. The SPEQ significantly improves the performance of 2-bit ResNet20 and achieves 91.4% test accuracy. This result is better than the QKD (Kim et al. 2019), which employs a large teacher. Furthermore, when only activations are quantized to 2 bits, SPEQ shows almost the same performance as the full-precision models for CIFAR10. The SPEQ shows consistent improvements on various CNNs.

Results on the ImageNet Dataset

The performance of the SPEQ on the ImageNet dataset is shown in Table 5. The retraining scheme shows 56.9%, 66.6%, and 70.5% top-1 accuracy for the 2-bit AlexNet, ResNet18, and ResNet34, respectively. By SPEQ training, the top-1 accuracy increases approximately 1% for ResNet18 and ResNet34. The SPEQ training on 2-bit AlexNet improves the top-1 accuracy noticeably, showing 59.3% top-1 accuracy. This result is only a 1.5% accuracy drop compared to the full-precision AlexNet. The results for 2-bit activation-quantized CNNs indicate that the proposed SPEQ method is very effective for reducing the activation quantization noise.

We also optimized the 3-bit EfficientNet-b0 with the ImageNet dataset. We obtained the pretrained full-precision model from the Tensorflow official Github [1] and retrained the model using the same hyper-parameters and data augmentation methods as those for ResNet. The performance

---

[1] https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

Methods	Retrain	SPEQ	AP†	QKD†
Top-1 Acc (%)	68.4	**69.5**	68.4	69.2

Table 6: ImageNet valiation top-1 accuracy on 3-bit weight and activation quantized EfficientNet-b0. The top-1 accuracy of the full-precision model is 76.7%. Results with the symbol † are from (Kim et al. 2019).

Method	ResNet18		ResNet34	
	Teacher	Acc(%)	Teacher	Acc(%)
Retrain	w/o KD	66.6	w/o KD	70.5
AP†	ResNet34	66.8	ResNet50	71.1
QKD†	ResNet34	67.4	ResNet50	71.6
SPEQ	-	67.4	-	71.5
SPEQ+AP	ResNet34	**67.8**	ResNet50	**72.1**

Table 7: 2-bit ImageNet quantization Top-1 validation accuracy (%) compared with other KD applied QDNNs. "Acc" with the symbol † are from (Kim et al. 2019).

of 3-bit EfficientNEt-b0 is compared in Table 6. Note that the AP (Mishra and Marr 2018) and QKD (Kim et al. 2019) employed the full-precision EfficientNet-b1 as a teacher for KD. SPEQ outperformed the existing KD methods on this recently developed CNN without a large teacher model.

The key difference in the training procedure between SPEQ and previous works is that SPEQ shares the same model for the teacher and the student. This simple choice leads to the significant savings in the training computation. Although our approach is based on the self-distillation, the larger teacher can also be employed to further improve the performance of the target model. The SPEQ training with the KD method is also compared with other KD training methods for QDNNs in Table 7. We applied the KD by combining the Apprentice (Mishra and Marr 2018) (AP) and the SPEQ scheme. The combined training improves the top-1 accuracy of ResNet18 and ResNet34 by 0.4% and 0.6%, respectively.

Results on Transfer Learning

Because SPEQ does not require a teacher model, it is convenient for transfer learning with only one base-model available. We optimized the low-precision BERT (Devlin et al. 2018) using SPEQ training. The pretrained BERT-Base model is obtained from the Google research [2] and fine-tuned using the Stanford Question Answering Dataset (SQuAD1.1) (Rajpurkar et al. 2016). The performance improvements of the SPEQ on the quantized BERT are shown in Table 8. The fine-tuned floating-point BERT shows 81.1 F1 and 88.6 EM scores. When the activation is quantized to n_A bits, the stochastic precision for computing soft labels is chosen between n_A and 8 bits.

We expand the experiment for transfer learning using Oxford Flowers-102 (Nilsback and Zisserman 2008). We em-

---

SQuAD1.1	W3/A3		W4/A4	
	EM	F1	EM	F1
FixedBERT	71.5	81.4	74.2	83.1
SPEQ	**76.4**	**85.1**	**78.0**	**86.6**

Table 8: The performance (EM and F1 scores) of low-precision BERT on the SQuAD1.1 dev dataset. FixedBERT results are from (Boo and Sung 2020)

# training samples	Float RN18 (CE Loss)	2-bit RN18 (CE Loss)	2-bit RN18 (SPEQ)
510 (5 / label)	69.18	70.67	**71.21**
1020 (10 / label)	78.04	77.99	**78.36**
2040 (20 / label)	84.57	83.91	**85.02**

Table 9: Validation accuracy (%) on the Flowers-102 dataset according to the training method and the feature extractor. RN is an abbreviation for ResNet.

ploy the ResNet18 as a feature extractor, which is frozen when fine-tuning. The SPEQ is evaluated by changing the number of training samples and the results are shown in Table 9. SPEQ improves the performance significantly compared to the 2-bit model retraining using the cross-entropy loss. Moreover, the SPEQ-trained 2-bit ResNet18 achieves better results than the floating-point model. In practice, the conventional KD method is hard to be applied due to the need of auxiliary models. Therefore, the SPEQ method is an invaluable solution to apply KD on transfer learning.

Concluding Remarks

In this work, we proposed a novel KD method for quantized DNN training. The proposed method, SPEQ, does not require a cumbersome teacher model; it assigns the same parameters for the teacher and student networks. The teacher model is formed by assigning the stochastic precision to the activation of each layer, by which it can produce the soft labels of stochastically ensembled models. The cosine similarity loss is used for KD training to render reliable operation even when the confidence of the teacher is lower than that of the student. The SPEQ outperforms the existing quantized training methods in various tasks. Furthermore, the SPEQ can be easily used for low-precision training even when no larger teacher model is available.

Acknowledgements

References

Boo, Y.; Shin, S.; and Sung, W. 2020. Quantized Neural Networks: Characterization and Holistic Optimization. *arXiv preprint arXiv:2006.00530* .

Boo, Y.; and Sung, W. 2020. Fixed-Point Optimization of Transformer Neural Network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1753–1757. IEEE.

Bucilu, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 535–541. ACM.

Cai, Z.; He, X.; Sun, J.; and Vasconcelos, N. 2017. Deep learning with low precision by half-wave gaussian quantization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5406–5414. IEEE.

Chen, D.; Mei, J.-P.; Wang, C.; Feng, Y.; and Chen, C. 2020. Online Knowledge Distillation with Diverse Peers. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Choi, J.; Venkataramani, S.; Srinivasan, V.; Gopalakrishnan, K.; Wang, Z.; and Chuang, P. 2019. Accurate and efficient 2-bit quantized neural networks. In *Proceedings of the 2nd SysML Conference*.

Choi, J.; Wang, Z.; Venkataramani, S.; Chuang, P. I.-J.; Srinivasan, V.; and Gopalakrishnan, K. 2018. PACT: Parameterized Clipping Activation for Quantized Neural Networks. *arXiv preprint arXiv:1805.06085* .

Courbariaux, M.; Bengio, Y.; and David, J.-P. 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems (NIPS)*, 3123–3131.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre–training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Fengfu, L.; Bo, Z.; and Bin, L. 2016. Ternary weight networks. In *NIPS Workshop on EMDNN*, volume 118, 119.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1243–1252. JMLR. org.

Guerra, L.; Zhuang, B.; Reid, I.; and Drummond, T. 2020. Switchable Precision Neural Networks. *arXiv preprint arXiv:2002.02815* .

Gupta, S.; Agrawal, A.; Gopalakrishnan, K.; and Narayanan, P. 2015. Deep learning with limited numerical precision. In *International Conference on Machine Learning (ICML)*, 1737–1746.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *European conference on computer vision*, 630–645. Springer.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

Hou, L.; and Kwok, J. T. 2018. Loss-aware weight quantization of deep networks. *International Conference on Learning Representations (ICLR)* .

Hwang, K.; and Sung, W. 2014. Fixed-point feedforward deep neural network design using weights +1, 0, and -1. In *Signal Processing Systems (SiPS), 2014 IEEE Workshop on*, 1–6. IEEE.

Jung, S.; Son, C.; Lee, S.; Son, J.; Han, J.-J.; Kwak, Y.; Hwang, S. J.; and Choi, C. 2019. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4350–4359.

Kapur, S.; Mishra, A.; and Marr, D. 2017. Low Precision RNNs: Quantizing RNNs Without Losing Accuracy. *arXiv preprint arXiv:1710.07706* .

Kim, J.; Bhalgat, Y.; Lee, J.; Patel, C.; and Kwak, N. 2019. QKD: Quantization-aware Knowledge Distillation. *arXiv preprint arXiv:1911.12491* .

Li, H.; Zhang, H.; Qi, X.; Yang, R.; and Huang, G. 2019. Improved techniques for training adaptive deep networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 1891–1900.

Lin, J.; Gan, C.; and Han, S. 2019. Defensive quantization: When efficiency meets robustness. *International Conference on Learning Representations (ICLR)* .

Mirzadeh, S.-I.; Farajtabar, M.; Li, A.; and Ghasemzadeh, H. 2019. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393* .

Mishra, A.; and Marr, D. 2018. Apprentice: Using Knowledge Distillation Techniques To Improve Low-Precision Network Accuracy. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=B1ae1lZRb.

Mishra, A.; Nurvitadhi, E.; Cook, J. J.; and Marr, D. 2017. WRPN: wide reduced-precision networks. *arXiv preprint arXiv:1709.01134* .

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 722–729. IEEE.

Ott, J.; Lin, Z.; Zhang, Y.; Liu, S.-C.; and Bengio, Y. 2016. Recurrent neural networks with limited numerical precision. *arXiv preprint arXiv:1608.06902* .

Park, E.; Ahn, J.; and Yoo, S. 2017. Weighted-entropy-based quantization for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5456–5464.

Phuong, M.; and Lampert, C. H. 2019. Distillation-based training for multi-exit architectures. In *Proceedings of the IEEE International Conference on Computer Vision*, 1355–1364.

Polino, A.; Pascanu, R.; and Alistarh, D. 2018. Model compression via distillation and quantization. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=S1XolQbRW.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* .

Shin, S.; Boo, Y.; and Sung, W. 2019. Empirical Analysis of Knowledge Distillation Technique for Optimization of Quantized Deep Neural Networks. *arXiv preprint arXiv:1909.01688* .

Shin, S.; Boo, Y.; and Sung, W. 2020. SQWA: Stochastic Quantized Weight Averaging for Improving the Generalization Capability of Low-Precision Deep Neural Networks. *arXiv preprint arXiv:2002.00343* .

Tang, J.; Shivanna, R.; Zhao, Z.; Lin, D.; Singh, A.; Chi, E. H.; and Jain, S. 2020. Understanding and Improving Knowledge Distillation. *arXiv preprint arXiv:2002.03532* .

Yang, J.; Shen, X.; Xing, J.; Tian, X.; Li, H.; Deng, B.; Huang, J.; and Hua, X.-s. 2019. Quantization networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7308–7316.

Yu, J.; and Huang, T. S. 2019. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE International Conference on Computer Vision*, 1803–1811.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* .

Zhang, D.; Yang, J.; Ye, D.; and Hua, G. 2018. LQ-Nets: Learned quantization for highly accurate and compact deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 365–382.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019a. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, 3713–3722.

Zhang, L.; Tan, Z.; Song, J.; Chen, J.; Bao, C.; and Ma, K. 2019b. SCAN: A scalable neural networks framework towards compact and efficient models. In *Advances in Neural Information Processing Systems*, 4027–4036.

Zhang, Y.; Pezeshki, M.; Brakel, P.; Zhang, S.; Bengio, C. L. Y.; and Courville, A. 2017. Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv preprint arXiv:1701.02720* .

Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; and Zou, Y. 2016. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160* .

Zhou, S.-C.; Wang, Y.-Z.; Wen, H.; He, Q.-Y.; and Zou, Y.-H. 2017. Balanced quantization: An effective and efficient approach to quantized neural networks. *Journal of Computer Science and Technology* 32(4): 667–682.

Zhu, C.; Han, S.; Mao, H.; and Dally, W. J. 2017. Trained ternary quantization. *International Conference on Learning Representations (ICLR)* .

Zhuang, B.; Shen, C.; Tan, M.; Liu, L.; and Reid, I. 2018. Towards effective low-bitwidth convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7920–7928.