- ORIGINAL ARTICLE -

A Mutual Learning Framework for Pruned and Quantized Networks

Un marco de aprendizaje mutuo para redes podadas y cuantificadas

Xiaohai Li[1,2,3], Yiqiang Chen[1,2] and Jindong Wang[4]

[1] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*
[2] *Beijing Key Laboratory of Mobile Computing and Pervasive Device, Beijing, China*
[3] *University of Chinese Academy of Sciences, Beijing, China*
[4] *Microsoft Research Asia, Beijing, China*
{lixiaohai, yqchen}@ict.ac.cn, jindong.wang@microsoft.com

Abstract

Model compression is an important topic in deep learning research. It can be mainly divided into two directions: model pruning and model quantization. However, both methods will more or less affect the original accuracy of the model. In this paper, we propose a mutual learning framework for pruned and quantized networks. We regard the pruned network and the quantized network as two sets of features that are not parallel. The purpose of our mutual learning framework is to better integrate the two sets of features and achieve complementary advantages, which we call feature augmentation. To verify the effectiveness of our framework, we select a pairwise combination of 3 state-of-the-art pruning algorithms and 3 state-of-the-art quantization algorithms. Extensive experiments on CIFAR-10, CIFAR-100 and Tiny-imagenet show the benefits of our framework: through the mutual learning of the two networks, we obtain a pruned network and a quantization network with higher accuracy than traditional approaches.

Keywords: Model Compression, Network Pruning, Quantization, mutual learning.

Resumen

La compresión de modelos es un tema importante en la investigación del aprendizaje profundo. Se puede dividir principalmente en dos direcciones: poda de modelos y cuantización de modelos. Sin embargo, ambos métodos afectarán más o menos la precisión original del modelo. En este artículo, proponemos un marco de aprendizaje mutuo para redes podadas y cuantificadas. Consideramos la red podada y la red quantized como dos conjuntos de características que no son paralelas. El propósito de nuestro marco de aprendizaje mutuo es integrar mejor los dos conjuntos de funciones y lograr ventajas complementarias, lo que llamamos aumento de funciones. Para verificar la efectividad de nuestro marco, seleccionamos una combinación por pares de 3 algoritmos de poda de última generación y 3 algoritmos de cuantificación de última generación. Extensos experimentos en CIFAR-10, CIFAR-100 y Tiny-imagenet muestran los beneficios de nuestro marco: a través del aprendizaje mutuo de las dos redes, obtenemos una red pruned y una red de cuantificación con mayor precisión que los enfoques tradicionales.

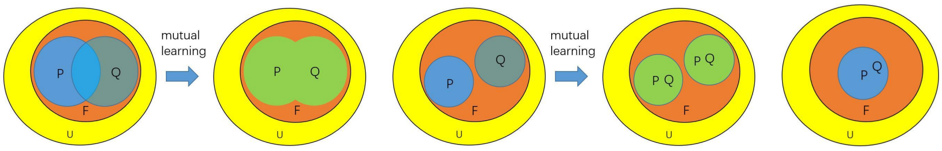
Palabras claves: Compresión de modelo, poda de red, cuantificación, aprendizaje mutuo.

1 Introduction

Deep learning is one of the important means to realize artificial intelligence. However, deep learning models often have large parameters and are difficult to deploy, specifically for edge computing devices with limited computing power and memory. Research on compressing deep learning models is on the rise. Model pruning and model quantization are the two main research directions of deep learning model compression.

A huge number of algorithms have been recently proposed to compress neural network in these two fields. In terms of model pruning, what they have in common is how to find and prune the weights or channels they think are unimportant, and make sure that the accuracy is not affected after pruning. As for quantification, the key is how to design a reasonable projection function to minimize the loss of quantization accuracy.

From the description above, it seems that these two fields of study are not related. Therefore, people tend to be independent researchers in both fields. However, we believe that the two are actually intrinsically related. From a linear algebra perspective, the essence of a neural network is to find an optimal mapping from input to output. This map is a representation of known data(training data) and can be used to predict unknown data(testing data). That is to say, when the data set is the same, the pruned network and the quantized network can be regarded as two different forms of representation. In most cases, the representational



(a) case 1 (b) case 2 (c) case 3

Figure 1: Motivation from a set-theoretical point of view.The sets P, Q, F, and U represent the representation capabilities of the pruned network, the quantitative network, the complete network, and the ideal network, respectively.

abilities of the two are inconsistent. Since they are inconsistent, this prompts us to think about whether the advantages of the two can be combined to obtain a better representation. This is exactly what our paper aims to do.

Perhaps our work is easier to understand from a set-theoretical point of view. Assuming that the representation capability of the pruned network is defined as the set P, the representation capability of the quantified network is defined as the set Q, the representation capability of the complete network is defined as the set F, and the representation capability of the ideal network is defined as the complete set U. Then there will be three cases, P and Q has intersection but does not overlap, P and Q has no intersection, P and Q overlaps as Figure 1 shown. As we mentioned earlier, the representational capabilities of pruning networks and quantization networks are often inconsistent. Usually we are faced with case1 and case2. In both cases, pruning and quantizing networks have the potential to improve representational power by learning from each other. So the key point is how to design a reasonable learning paradigm to guide the two to learn from each other. This is exactly what our paper examines.

In this paper, we propose MLPQ, a Mutual Learning Framework for Pruned and Quantized Networks to solve this problem. In our framework, the pruned network and the quantized network are trained at the same time. In the training process, the learning situation of each network is judged by the evaluation criteria, and then the part of the network that has been learned well relative to the other network is retained, and the part that is poorly learned learns the other. In this way, the two networks not only retain the good parts of each other, but also learn the good parts of each other.

In summary, our contributions are three-fold:

- We propose MLPQ to train the pruned network and the quantized network at the same time. Through the mutual learning between the two, a better performance than traditional approaches can be obtained.
- We design an effective evaluation criteria and a reasonable learning paradigm to guide the mutual learning between pruning and quantization networks.
- Extensive experiments prove that the performance of the pruning algorithm and the quantization algorithm has been significantly improved through our mutual learning framework.

This paper is organized as follows. We formulate the problem of Network Pruning and Quantization, and discuss its relationship with existing research areas in Section 2. Section 3 presents our approach: MLPQ. In Section 4, we describe experiments in detail. Finally, we conclude this paper in Section 5.

2 Related Work

2.1 Network Pruning

Model pruning is to find out the redundant parameters in the network through some evaluation criteria to delete. There are three key elements involved, one is an evaluation criterion to determine the redundant parameters of the model, the second is the granularity of pruning, and the third is the strategy of pruning. A schematic diagram of a simple network pruning is shown in Figure 2.

Problem Formulation A neural network NN f parameterized by $\mathbf{W}$ can be represented as $f(x;\mathbf{W})$. Neural network pruning entails taking as input a model $f(x;\mathbf{W})$ and producing a new model $f(x;\mathbf{M}\odot\mathbf{W}')$, where $\mathbf{W}'$ is a set of new parameters that may be different from $\mathbf{W}$, $\mathbf{M}\in\{0,1\}^{|\mathbf{W}'|}$ is a binary mask that fixes certain parameters to 0, and $\odot$ denotes element-wise production. Note that network pruning should retain the performance of the vanilla model (e.g., the classification accuracy should not drop) while minimizing the parameter size $|\mathbf{W}'|$.

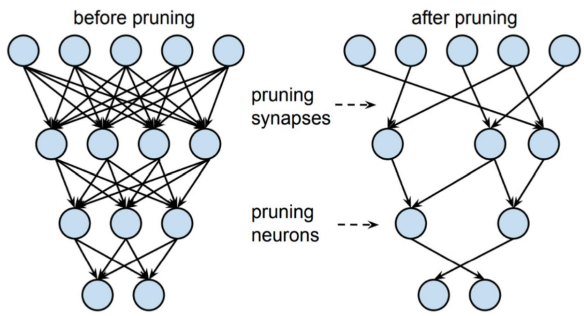


Figure 2: A schematic diagram of a simple network pruning.

In this paper, our focus is convolutional neural nets. Specifically, in a CNN classification network with N convolutional layers, we use $\mathbf{W}_i \in \mathbb{R}^{N_i \times C_i \times K_i \times K_i}(1 \leq i < N)$ to denote the weight matrix connecting the i_{th} and $i+1_{th}$ convolutional layers and N_i, C_i, K_i represent the number of output channels, the number of input channels, and the kernel size of filters, respectively. Similarly, the weights after pruning can be represented as $\mathbf{W}_i' \in \mathbb{R}^{N_i' \times C_i' \times K_i \times K_i}$. Therefore, the goal of pruning CNN is to minimize all N_i' and C_i' while retaining the vanilla accuracy.

Evaluation criteria The existing evaluation criteria are varied, and it is difficult to judge which method is better. The mainstream evaluation criteria are as follow: Energy-based [1], Weight-based [2, 3], Correlation-based [4], Average Percentage of Zero (APoZ) [5] and Entropy-based [6–8]. Recently, Entropy-based methods show excellent performance.

Pruning granularity In addition to evaluation criteria, the pruning method can be roughly divided into 4 levels according to the pruning granularities. At the finest level, single weight is independently pruned [2, 3, 7, 9]. As a result, the network connection is not regular, and the memory usage needs to be reduced by sparse expression, which leads to a large amount of conditional judgment and extra space to indicate the 0 or non-zero parameter position in the forward propagation, and thus is not suitable for parallel computing. The next pruning granularity is intra-kernel weight pruning/nuclear thinning [10–13]. It imposes a restriction on the update of the weights to make it more sparse, so that most of the weights are 0. The third pruning granularity is middle hidden layer pruning [14]. Layer-wise pruning affects the depth of the network and a deep network can be converted into a shallow network. The last pruning granularity is Convolution kernel/Feature map/Channel/Filter pruning [15–21]. Feature map pruning affects the layer width. It directly leads to a thinner network and no sparse representation is needed. And it does not rely on any sparse convolution calculation library and special software.

Pruning strategy As with the evaluation criteria, the pruning strategy is also critical for network compression. There are usually two ways to deal with it: one-shot [22, 23] or iteratively [24, 25]. For a target pruning ratio, the iterative process gradually increases sparsity and repeats the process M times. The one-shot pruning induces the target pruning ratio in one step. However, it is difficult to avoid a drop in accuracy. As for iteratively pruning, the prior work [8] has shown that we can prune during the iteration, with no need for additional retraining. And it is found that fine-tuning after pruning can make up for the loss of precision caused by pruning, so many methods will do fine-tuning after pruning. Therefore, iterative pruning seems to be better.

2.2 Quantization

Quantization is a way of representing high-precision numbers with low-precision. Quantification has different classification methods according to different criteria. Firstly, according to the number of bits after quantization, it can be divided into 2-bit quantization, 3-bit quantization, 8-bit quantization, mixed-bit quantization, etc. Secondly, according to the quantization method, it can be divided into uniform quantization and non-uniform quantization. Finally, according to Fine-tuning Methods, it can be divided into Quantization Aware Training (QAT) and Post-Training Quantization (PTQ). A schematic diagram of Quantization is shown in Figure 3.

Problem Formulation Assume that the NN has L layers with learnable parameters, denoted as $\{W_1, W_2, \ldots, W_L\}$, with θ denoting the combination of all such parameters. Without loss of generality, we focus on the supervised learning problem, where the nominal goal is to optimize the following empirical risk minimization function:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} l\left(x_i, y_i; \theta\right) \tag{1}$$

where (x, y) is the input data and the corresponding label, $l(x, y; \theta)$ is the loss function (e.g., Mean Squared Error or Cross Entropy loss), and N is the total number of data points. Let us also denote the input hidden activations of the i^{th} layer as h_i , and the corresponding output hidden activation as a_i . We assume that we have the trained model parameters θ, stored in floating point precision. In quantization, the goal is to reduce the precision of both the parameters (θ), as well as the intermediate activation maps (i.e., h_i , a_i) to low-precision, with minimal impact on the generalization power/accuracy of the model. To do this, we need

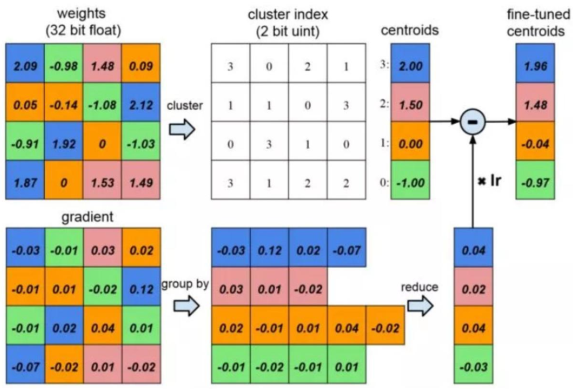


Figure 3: A schematic diagram of Quantization.

to define a quantization operator that maps a floating point value to a quantized one.

number of bits After Courbariaux proposed BinaryConnect [26], researchers begin to explore the mysteries of binary networks. BinaryConnect is a method which consists in training a DNN with binary weights during the forward and backward propagations, while retaining precision of the stored weights in which gradients are accumulated. Following in his footsteps, Darabi proposed an improved binary training method BNN+ [27], by introducing a regularization function that encourages training with weights around binary values. Then Phan proposed a novel neural network architecture, namely MoBiNet [28] - Mobile Binary Network in which skip connections are manipulated to prevent information loss and vanishing gradient, thus facilitating the training process. Shekhovtsov proposed a new method [29] combining sampling and analytic approximation steps. The method has a significantly reduced variance in the price of a small bias. Then Kim proposed to search architectures for binary networks BNAS [30] by defining a new search space for binary architectures and a novel search objective. Bulat proposed to address the inherent information bottleneck [31] in binary networks by introducing an efficient width expansion mechanism.

The above works are all based on binary networks. Later researchers found that it is difficult to improve the accuracy of binary networks, so some began to study mixed precision networks. Wang introduced the Hardware-Aware Automated Quantization HAQ [32] framework which leverages the reinforcement learning to automatically determine the quantization policy. And Uhlich proposed to parameterize the quantizer with the step size and dynamic range [33]. The bit width can then be inferred from them. Liu proposed multipoint quantization [34], a quantization method that approximates a full precision weight vector using a linear combination of multiple vectors of low-bit numbers. Yang proposed a novel learning-based algorithm [35] to derive mixed precision models end-to-end under target computation constraints and model sizes. Yang proposed bit-level sparsity quantization BSQ [36] to tackle the mixed-precision quantization from a new angle of inducing bit-level sparsity.

quantization method When the resulting quantized values (quantization levels) are uniformly in space, we call it so-called uniform quantization, otherwise Non-Uniform Quantization. Goncharenko proposed two methods to significantly optimize the training with the uniform quantization procedure [37]. The first one is introducing the trained scale factors for discretization thresholds that are separate for each filter. The second one is based on the mutual rescaling of consequent depth-wise separable convolution and convolution layers. Since uniform quantization has a larger quantization error than non-uniform quantization, most researches focus on the latter. Some work in the literature has also explored nonuniform quantization [38–44], where quantization steps as well as quantization levels are allowed to be non-uniformly spaced.

Fine-tuning Methods Quantization Aware Training QAT is an approach in which the usual forward and backward pass is performed on the quantized model in floating point, but the model parameters are quantized after each gradient update (similar to projected gradient descent). And accumulating the gradients in quantized precision can result in zero gradient or gradients that have a high error, especially in low-precision [26, 45–48]. An alternative to the expensive QAT method is Post-Training Quantization (PTQ) which performs the quantization and the adjustments of the weights, without any fine-tuning [34, 49–54]. Unlike QAT, which requires a sufficient amount of training data for retraining, PTQ has an additional advantage that it can be applied in situations where data is limited or unlabeled. However, this often comes at the cost of lower accuracy as compared to the QAT, especially for low-precision quantization.

2.3 In-Parallel Pruning-Quantization

In the process of studying pruning and quantification, some researchers try to prune and quantify at the same time, the so-called In-Parallel Pruning-Quantization. Tung combined network pruning and weight quantization in a single learning framework [55] that performs pruning and quantization jointly, and in parallel with fine-tuning. Wang devised to train a quantization-aware accuracy predictor [56] that is fed to the evolutionary search to select the best fit. And Wang framed neural network compression as a joint gradient-based optimization problem [57], trading off between model pruning and quantization automatically for hardware efficiency. Baalen introduces Bayesian Bits [58], a practical method for joint mixed precision quantization and pruning through gradient based optimization. Bayesian Bits employ a novel decomposition of the quantization operation, which sequentially considers doubling the bit width. Kim proposed the position-based scaled gradient PSG [59] that scales the gradient depending on the position of a weight vector to make it more compression-friendly.

It can be seen that these studies often ignore the inherent relationship between pruning and quantification, and regard them as two problems, usually using a joint optimization method. The difference between our research and the above research is that we believe that there is an inherent relationship between pruning and quantization. As long as this connection is fully utilized, we can maximize the performance of the pruned network and the quantization network. As Figure 1 shown, both P and Q are subsets of F, they both retain part of the information of the full network and also lose part of it. Specifically, the pruned network results in a loss of network structure complexity due to the deletion of some redundant parameters, while the quantization network results in a loss of parameter precision due to quantization. The two can make up for their shortcomings by learning from each other. This is exactly the starting point of our paper, and we'll show how this can be achieved by our proposed mutual learning framework.

3 Methodology

For ease of understanding, we first define some concepts. Firstly, what we need to define is how good or bad the learning results of the pruned network and the quantized network is. Here we focus on the image classification task, assuming that the training data x_C^l is divided into L classes with label y_C, where C is the number of samples. For the neural network NN, the training data x_C^l enters the network to obtain the output y_C^l. If the output classification result y_C^l is consistent with the training data label y_C, the classification is correct. At this time, we think that the result learned is good. Conversely, we consider the learned results to be bad. Suppose the i_th output is y^i, We define y^{i+} to indicate that the output is consistent with the label, that is, a good result. And y^{i-} indicate the output is not consistent with the label. In our mutual learning framework, we define y_p^{i+}, y_p^{i-}, y_q^{i+} and y_q^{i-} similarly. The slight difference is that '$+$' and '$-$' here are relative concepts, which indicates closer or further away from the label.

3.1 Motivation

As Figure 1 shown, both P and Q are subsets of F, they retain part of the information of the full network and also lose part of it. Specifically, the pruned network results in a loss of network structure complexity due to the deletion of some redundant parameters, while the quantization network results in a loss of parameter precision due to quantization. Thus, we proposed a Mutual Learning Framework for Pruned and Quantized Networks, as Figure 4 shown. Training a pruned network and a quantized network with the same data yields different results, both good and bad, due to the respective limitations of the two networks. Our framework strives to make each other's good results correct the other's bad results so that the performance of each can be improved.

3.2 Our approach: MLPQ

Evaluation criteria In order for the two networks to learn their respective advantages from each other, the first and most critical point is to design a reasonable standard to judge the quality of network learning. Since our training data is image classification data, such as cifar10 and cifar100, the most intuitive way to judge whether it is good or bad is the accuracy of its classification results. To this end, we designed two kinds of candidates, one is the probability of the class with the highest probability, and the other is the ratio of the probability of the class with the highest probability of the second class. We find that it is better to directly use the probability maximum as the criterion. Therefore, we use it as a score to judge whether the network is good or bad so that the network output results are divided into two parts, good and bad. As Figure 4 shown, the pruned network (N_P), we define good results as y_p^{i+}, and bad results as y_p^{i-}. Similarly, for quantized networks(N_Q), we define good results as y_q^{i+}, and bad results as y_q^{i-}. After this division, you can find those learnable parts (bad parts) for learning to enhance the performance of the network.

Mutual learning In order to implement our mutual learning framework, some modifications to the original pruning algorithm and quantization algorithm are required.

For the pruned network, we assume that the loss function of the original pruning algorithm is expressed as follows:

$$L(P) = L_{\mathrm{CE}}(P,Y) + \alpha L_{\mathrm{PC}}(P) \tag{2}$$

where $L_{\mathrm{CE}}(P,Y)$ is the cross entropy loss function, $L_{\mathrm{PC}}(P)$ is the loss function related to pruning constraints, and α is used to balance the network accuracy and pruning strength. Then we revise the loss function by score in the following form:

$$L(P) = \lambda_{PCE} L_{\mathrm{CE}}(P^+,Y) + \lambda_{PCE'} L_{\mathrm{CE'}}(P^-,Q^+) + \alpha L_{\mathrm{PC}}(P) \tag{3}$$

where $L_{\mathrm{CE}}(P^+,Y)$ is the cross entropy loss function of y_p^{i+}, $L_{\mathrm{CE}}(P^-,Q^+)$ is the revised cross entropy function of y_p^{i-}, which replaces the label with the classification result of y_q^{i+} to calculate the cross entropy.

For the quantization network, we assume that the loss function of the original quantization algorithm is expressed as follows:

$$L(Q) = L_{\mathrm{CE}}(Q,Y) + \beta L_{\mathrm{QC}}(Q) \tag{4}$$

where $L_{\mathrm{CE}}(Q,Y)$ is the cross entropy loss function, $L_{\mathrm{QC}}(Q)$ is the loss function related to quantization constraints, and β is used to balance the network accuracy and QUANTIZATION strength. Then we revise the loss function by score in the following form:

$$L(Q) = \lambda_{QCE} L_{\mathrm{CE}}(Q^+,Y) + \lambda_{QCE'} L_{\mathrm{CE'}}(Q^-,P^+) + \beta L_{\mathrm{QC}}(Q) \tag{5}$$

where $L_{\mathrm{CE}}(Q^+,Y)$ is the cross entropy loss function of y_q^{i+}, $L_{\mathrm{CE}}(Q^-,P^+)$ is the revised cross entropy

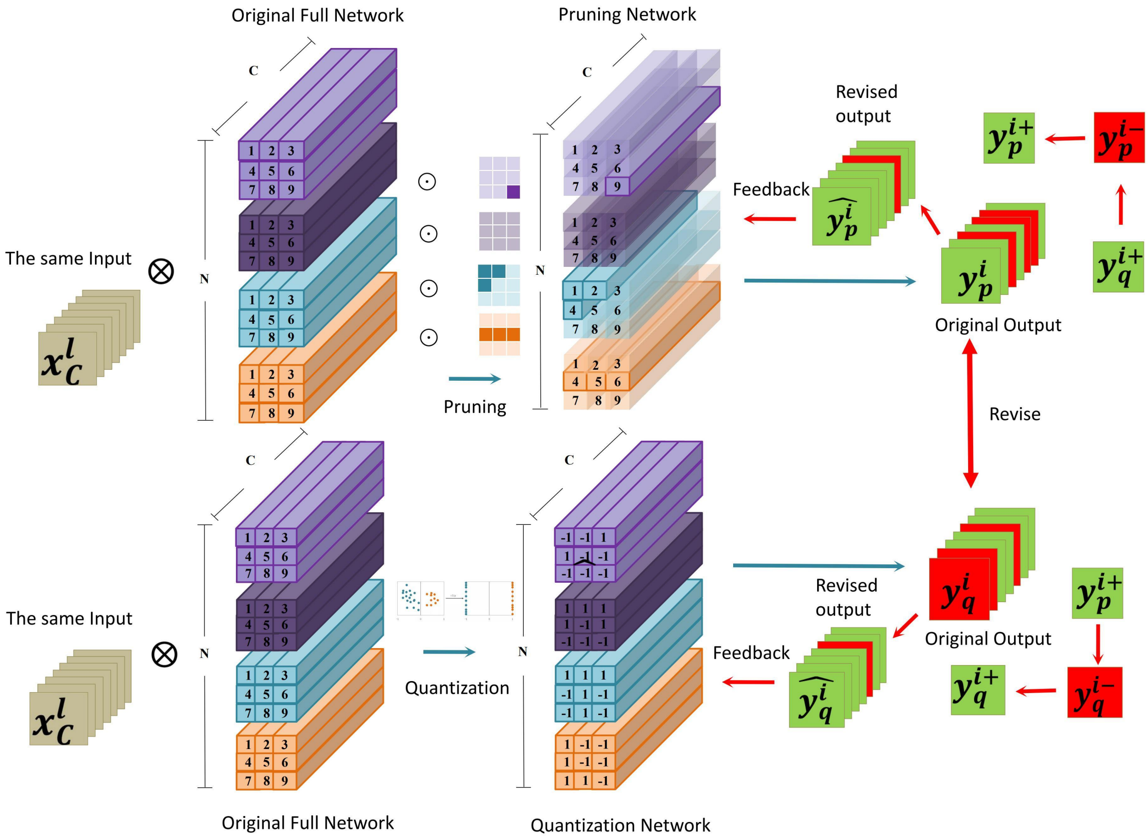


Figure 4: MLPQ Framework.

function of y_q^{i-}, which replaces the label with the classification result of y_p^{i+} to calculate the cross entropy.

λ_{QCE}, $\lambda_{QCE'}$, λ_{QCE} and $\lambda_{QCE'}$ are the coefficients that control the learning strength of the good and bad parts of the two networks respectively.

We believe that the mutual learning process of the two networks has gone through three stages. The first stage is that the learning capabilities of the two networks themselves converge, the second stage is that the two networks learn from each other, and the third stage is that the two networks cooperate in Learning ability convergence. There is an overlap between the three stages. At different stages, the learning strength of the two parts of the two networks should be different. So these four coefficients we called the learning coefficient (LC) are continuously changed during the training of the two networks in our mutual learning framework. The LC adjusts the learning intensity according to the current learning situation of the two networks. In the process of mutual learning between two networks, the part to be learned can be divided into two cases, according to the output results, one is that the classification results of the two networks are the same, and the other is that the classification results of the two networks are different. For the former, we think that the two networks are already in the third stage of mutual learning, and for the latter, we think it may still be in the first or second stage. We assume that the number of training samples with the same classification result in the part to be learned is n_{peq} and n_{qeq} respectively, and the number of samples with different classification results is n_{puneq} and n_{quneq} respectively. Then LC is related to three factors, which we call Convergence factor(F_{cv}), Contrast factor(F_c) and Self-feedback factor(F_{sf}). The F_{cv} should satisfy the following form:

$$F_{cv}(P) = n_{peq}/n_{puneq} \tag{6}$$

$$F_{cv}(Q) = n_{qeq}/n_{quneq} \tag{7}$$

For pnet, the larger the ratio n_{peq}/n_{puneq}, the closer to the third stage, the greater the strength of the learning part can be set. On the contrary, the smaller the ratio n_{peq}/n_{puneq}, the closer to the first stage, the smaller the intensity of the learning part can be set. For qnet it is similar.

In addition to the number of learning samples, the training loss can also reflect the learning situation of the network. Assume that the cross-entropy losses and revised the cross-entropy losses of pnet and qnet at the i_{th} epoch are $L_{\mathrm{CE_i}}(P)$, $L_{\mathrm{CE_i}}(P^+)$, $L_{\mathrm{CE'_i}}(P^-)$, $L_{\mathrm{CE_i}}(Q)$, $L_{\mathrm{CE_i}}(Q^+)$ and $L_{\mathrm{CE'_i}}(Q^-)$, respectively. Then F_c and

F_{sf} at the $i+1_{th}$ epoch should satisfy the following form:

$$F_c(P) = \ell_{\mathrm{CE}}(P)/L_{\mathrm{CE}}(Q^+) \tag{8}$$

$$F_c(Q) = L_{\mathrm{CE}}(Q)/L_{\mathrm{CE}}(P^+) \tag{9}$$

$$F_{sf}(P) = L_{\mathrm{CE}}(P)/L_{\mathrm{CE}}(P^+) \tag{10}$$

$$F_{sf}(Q) = L_{\mathrm{CE}}(Q)/L_{\mathrm{CE}}(Q^+) \tag{11}$$

For pnet, the larger the ratio $L_{\mathrm{CE}}(P)/L_{\mathrm{CE}}(Q^+)$, the better qnet learns than pnet, the greater the strength of the learning part can be set. On the contrary, the smaller the ratio $L_{\mathrm{CE}}(P)/L_{\mathrm{CE}}(Q^+)$, the smaller the intensity of the learning part can be set. And the larger the ratio $L_{\mathrm{CE}}(P)/L_{\mathrm{CE}}(P^+)$, the closer to the third stage, the greater the strength of the learning part can be set. On the contrary, the smaller the ratio $L_{\mathrm{CE}}(P)/L_{\mathrm{CE}}(P^+)$, the closer to the first stage, the smaller the intensity of the learning part can be set. For qnet it is similar.

Therefore, we control the learning intensity at different stages by dynamically adjusting the LC, to achieve the purpose of mutual learning between the two networks.

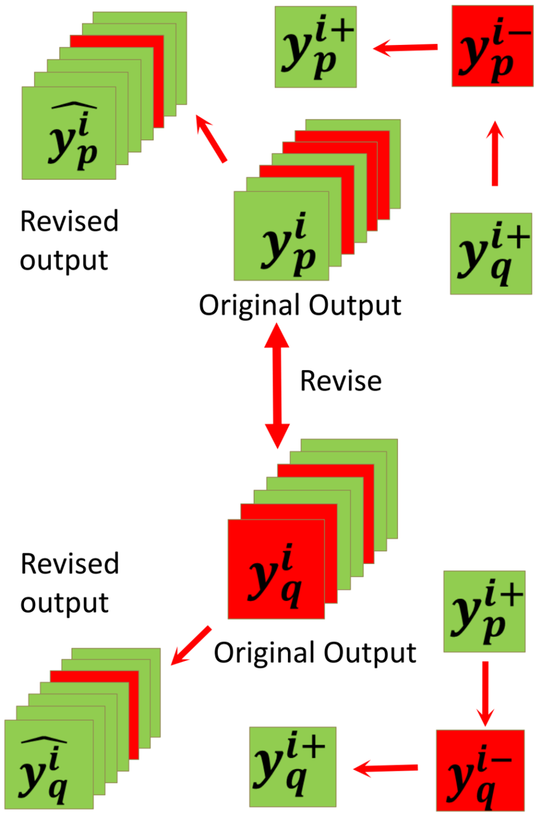


Figure 5: Mutual learning.

Training Process Our proposed mutual learning framework requires training both pruned and quantized networks. In the training process, according to their respective learning situations, they learn the other's good results in time to update themselves, to obtain a pruning network and a quantization network with better performance at the same time. To avoid convergence failures, we include an initial warm-up phase where the two networks are trained separately without mutual learning. Algorithms are shown as Algorithm 1.

Algorithm 1 Algorithm of MLPQ

Input: *pnet*, *qnet*, epoch m, n.
Output: *pnet'*, *qnet'*.

```
Initialize pnet, qnet.
For epoch k in [1, ..., m] do                ▷ warm up
    two networks are trained by Eq (2) and Eq (4).
End For
For k in [m, ..., n] do                      ▷ mutual learning
    calculate the score of pnet, qnet.
    calculate the LC by eqs. (6) to (11).
    mutual learning by Eq (3) and Eq (5).
End For
return pnet', qnet'
```

In the first stage of training, we use the original loss functions Eq (2) and Eq (4) to train two networks. This is to avoid interfering with each other when the two networks have not learned well, resulting in difficulty in convergence or falling into a local optimum. We call this phase the warm-up. Next is the so-called mutual learning phase. At this stage, for each epoch of training, we use the previously mentioned score to judge the relative learning of the two networks according to the network output. Then LC is calculated according to the relevant formula, and finally, the two networks are trained according to our proposed revised loss function Eq (3) and Eq (5).

4 Experiments

We conduct extensive experiments using different network architectures and various pruning and quantization algorithms to evaluate the performance of MLPQ.

4.1 Setup

Datasets CIFAR-10 and CIFAR-100 datasets consist of colored natural images with a size of 32×32 drawn from 10 and 100 classes, respectively. In each dataset, the train and test sets contain 50,000 and 10,000 images. Tiny-imagenet classification challenge is similar to the classification challenge in the full ImageNet ILSVRC. Tiny-imagenet contains 200 classes for training. Each class has 500 images. The test set contains 10,000 images. All images are 64x64 colored ones.

Baselines We implement our MLPQ with different network architectures: Resnet18, Resnet34, resnet20, resnet20+, resnet56. A network at the beginning of a capital letter indicates that the normal number of channels has not been reduced. The number of channels for Resnet18 and Resnet34 is [64,128,256,512], the values in parentheses are the number of channels in different layers. The number of channels for resnet20 and resnet56 is [16,32,64], and for resnet20+ is [80,160,320].

We compare MLPQ with various state-of-the-art pruning methods including Directional pruning (gRDA) [60], Towards Compact CNNs(Towards) [61] and Regularization-Pruning(Regu) [62]. Among them, gRDA and Towards are pruning while training, and Regu is pruning first and then finetuning. And gRDA is dynamic pruning according to the pruning strength, the parameter compression ratio is not fixed, while Towards and Regu are fixed-parameter compression ratios

And we compare MLPQ with various state-of-the-art Quantization methods including Reviving the Dead Weights (Re) [63], Any-Precision (Any) [64] and Element-wise Gradient Scaling (EW) [65]. Among them, Re is 1bit quantization, Any can achieve any bit quantization. In our experiment, for the convenience of comparison, we use 1bit and 2bit quantization with Any. EW is also 2bit quantized. And all three algorithms are Quantization Aware Training.

We put these algorithms in pairs in our mutual learning framework, and experiments have shown that through our framework, the performance of these algorithms has been improved. In the tables, the baseline row represents the result of the original algorithm, and the ours row represents the result of the original algorithm with our mutual learning framework.

Evaluation metrics We adopt two evaluation metrics:

- Classification accuracy (%) on test sets for two networks.
- parameter compression ratio (p_r): The ratio of number of non zero weights in the original model against the compressed network only for pruning network.

4.2 Results on Cifar10

The results of mutual learning on cifar10 are shown in Table 1. The three quantization algorithms are all QAT, with little difference. However, among the three pruning algorithms, gRDA pruned with a variable parameter compression ratio during training, Towards pruned with a fixed parameter compression ratio during training, and Regu pruned with a fixed parameter compression ratio and finetune. So we analyze the experimental results according to different pruning algorithms.

gRDA results As shown in Table 1, for gRDA & Re, we use two large networks Resnet18 and Resnet34. For Resnet18, with our mutual learning framework, the parameter compression ratio of the pruned network improves by about 2 times while maintaining the same accuracy. And the accuracy of the quantization algorithm has also improved, although not significantly. For Resnet34, The pruned network performed similarly, but the quantized network showed little improvement. This phenomenon shows that when the network is large enough, the performance of the quantization network has been compared with the full-precision network, so it is difficult to learn useful information from the pruned network. However, the network structure loss caused by pruning in the pruned network can be well compensated from the quantized network.

gRDA & EW and gRDA & Any used resnet20 and resnet20+ respectively. resnet20+ has 5 times as many channels as resnet20. From the experimental results, we can see that the pruned network can still improve the parameter compression ratio by about 2 times while keeping the accuracy unchanged. Even for small network structures, resnet20, the accuracy of the pruning algorithm is improved. And compared with the large model Resnet18 and Resnet34, the accuracy of the quantization algorithm is more obvious. This shows that when the network is small, the mutual learning potential of the pruned network and the quantization network is greater.

In general, when using gRDA and different quantization algorithms for mutual learning, the performance (parameter compression ratio) of gRDA is improved by 2 times, regardless of whether it is a small model or a large model. The performance gain of the quantization algorithm decreases with the size of the model. The experimental results are consistent with our motivation as Figure 1 shown.

Towards results As shown in Table 1, for Towards, We adopted 3 network structures, resnet56, resnet20 and resnet20+. From the experimental results, we can see that the performance improvement of pruning is limited in the case of the fixed parameter compression ratio. Comparing the two different parameter compression ratio settings with resnet20+, we can see that when the pruned network parameter compression ratio is larger, the accuracy is improved more, obviously. This shows that when the accuracy is not close to the complete network, the greater the network pruning, the greater the mutual learning potential.

For the quantitative network, the accuracy has been improved. The improvement effect of resnet56 is more obvious than that of resnet20. When the network is too small, the expressive ability of the quantitative network will be limited, thus affecting the effect of mutual learning. In resnet20+, when the parameter compression ratio is 4, the accuracy improvement of the quantization network is higher than that when it

Table 1: mutual learning on cifar10

mutual learning	Net	method	*Paccuracy*	*Qaccuracy*	*PParam*(*compressed*/*original*) (M)	p_r
gRDA & Re	Resnet18	baseline	93.03	92.59	0.145/11.2	77×
		ours	93.13↑0.10	93.06↑ 0.47	0.069/11.2	162× ↑ 85
	Resnet34	baseline	93.50	93.70	0.151/21.3	140×
		ours	93.50	93.70	0.073/21.3	289×↑ 149×
gRDA & EW	resnet20	baseline	89.56	84.27	0.065/0.270	4.1×
		ours	90.23↑0.67	85.19↑ 0.92	0.028/0.270	9.6×↑ 5.5×
gRDA & Any	resnet20+	baseline	93.42	92.26	0.14/6.7	47.9×
		ours	93.42	93.06↑ 0.80	0.08/6.7	83.8×↑ 35.9×
Towards & Re	resnet56	baseline	92.50	84.44	0.416/0.832	2×
		ours	92.82↑0.32	86.05↑ 1.61	0.416/0.832	2×
Towards & EW	resnet20	baseline	89.25	84.27	0.123/0.246	2×
		ours	89.44↑0.19	85.11↑ 0.83	0.123/0.246	2×
Towards & Any	resnet20+	baseline	94.75	92.47	3.1/6.2	2×
		ours	94.77↑0.02	92.60↑0.13	3.1/6.2	2×
		baseline	93.77↑0.17	92.47↑0.58	1.55/6.2	4×
		ours	93.94	93.05	1.55/6.2	4×
Regu & Re	resnet56	baseline	92.50	84.61	0.488/0.488	1×
		ours	92.67↑0.17	85.22↑0.61	0.488/0.488	1×
	resnet20	baseline	90.99	81.54	0.154/0.154	1×
		ours	91.34↑0.35	82.29↑0.75	0.154/0.154	1×
Regu & EW	resnet56	baseline	92.58	83.84	0.488/0.488	1×
		ours	92.75↑0.17	86.97↑3.07	0.488/0.488	1×
	resnet20	baseline	90.96	82.16	0.154/0.154	1×
		ours	91.21↑0.25	83.78↑1.62	0.154/0.154	1×
Regu & Any	resnet56	baseline	92.61	86.45	0.488/0.488	1×
		ours	93.02↑0.41	87.10↑0.65	0.488/0.488	1×
	resnet20	baseline	91.00	81.41	0.154/0.154	1×
		ours	91.25↑0.25	81.90↑0.49	0.154/0.154	1×

is 2. This shows that when the expressive ability of the two networks has a large gap, it will also affect the effect of mutual learning.

In general, when Towards is adopted with MLPQ, the performance (accuracy) of the pruned network obtained a limited improvement while keeping the parameter compression ratio the same. This shows that the fixed parameter ratio limits the effect of mutual learning, and the gradual dynamic pruning can better utilize the information of the quantized network in the training process to make up for the loss of accuracy caused by the loss of network structure. For example, in Towards & Any, when the parameter compression ratio is increased from 2 to 4, the accuracy is reduced from 94.77 to 93.94. In gRDA & Any, the parameter compression ratio is almost improved, and the accuracy is 93.42 almost unchanged.

Regu results As shown in Table 1, for Regu, We adopted 2 network structures, resnet56, resnet20. Note that Regu pruned with fixed parameters and then finetune. So the p_r is always equal to 1. For the pruned network, whether in resnet56 or resnet20, the performance (accuracy) has been improved, but it is limited. This is similar to Towards' result. For quantitative networks, the accuracy of both network structures is improved. And the accuracy of 2bit quantization EW is higher than that of 1bit (Re and Any). Comparing the two 1bit quantization algorithms, on resnet56, the result of Any is significantly better than that of Re, and on resnet20, the results of the two are similar. Note that here Any uses 1bit quantization to achieve the same accuracy as EW uses 2bit quantization on resnet56. This shows that when the network structure is not large enough, the improvement provided by our MLPQ is limited by the network structure and affected by quantization algorithms.

4.3 Results on Cifar100

The results of mutual learning on cifar100 are shown in Table 2. For cifar100, we did 4 sets of experiments, gRDA & Re, Regu & Re, Regu & EW and Regu & Any. We divide the experimental results into two groups according to the different pruning algorithms.

gRDA results For gRDA & Re, we adopted 2 network structures, Resnet18 and Resnet34. From the experimental results, it can be seen that the accuracy of the pruned network with our MLPQ is still improved

Table 2: mutual learning on cifar100

mutual learning	Net	method	*Paccuracy*	*Qaccuracy*	*PParam*(*compressed*/*original*) (M)	p_r
gRDA & Re	Resnet18	baseline	74.60	71.67	0.56/11.3	20×
		ours	75.57↑0.97	73.52↑1.85	0.23/11.3	49×↑29×
	Resnet34	baseline	75.03	73.38	0.54/21.4	40×
		ours	75.35↑0.32	75.06↑1.68	0.25/21.4	86×↑46×
Regu & Re	resnet56	baseline	70.47	50.69	0.494/0.494	1×
		ours	71.22↑0.75	52.89↑2.20	0.494/0.494	1×
	resnet20	baseline	65.95	43.74	0.160/0.160	1×
		ours	66.21↑0.26	46.21↑2.47	0.160/0.160	1×
Regu & EW	resnet56	baseline	70.50	57.23	0.494/0.494	1×
		ours	71.67↑1.17	60.35↑3.12	0.494/0.494	1×
	resnet20	baseline	65.84	52.09	0.160/0.160	1×
		ours	66.15↑0.31	52.90↑0.82	0.160/0.160	1×
Regu & Any	resnet56	baseline	70.24	56.65	0.494/0.494	1×
		ours	71.61↑1.37	57.78↑1.13	0.494/0.494	1×
	resnet20	baseline	66.23	59.35	0.160/0.160	1×
		ours	66.44↑0.21	61.17↑1.82	0.160/0.160	1×

when the parameter compression ratio is increased by more than 2 times. The improvement on Resnet18 is more obvious than that of Resnet34. And the improvement of quantization network accuracy is also more obvious than that of cifar10. This shows that when the complexity of the training data set is increased, the network expression ability is not enough to fully express the data. At this time, the potential of the two networks to learn from each other is even greater.

Regu results For Regu, we adopted 2 network structures, resnet56 and resnet20, with 3 kinds of quantization algorithms. Note that in this group of experiments, Re uses 1bit quantization, EW uses 2bit quantization, Any uses 2bit quantization on resnet20, and 1bit on resnet56. It can be seen from the experimental results that the accuracy of the pruned network is improved more obvious on resnet56 than on resnet20. This shows that when the network is too small and the training data is complex, the pruned network will have limited expression ability due to severe structural loss, thereby reducing the potential for mutual learning. For the quantitative network, there is no obvious rule, some have more obvious improvement on resnet20, and some have more obvious improvement on resnet56. This shows that the mutual learning effect of the quantization network is affected by the adopted quantization algorithm.

4.4 Results on Tiny-imagenet

The results of mutual learning on Tiny-imagenet are shown in Table 3.

For Tiny-imagenet, we did 3 sets of experiments on resnet20 and resnet56, Regu & Re, Regu & EW and Regu & Any. Note that in this group of experiments, Re and Any uses 1bit quantization, EW uses 2bit quantization. It can be seen from the experimental results that the improvement on resnet20 is not very obvious for the pruned network. This is because the training data is too complex, and the network structure is too simple, resulting in insufficient expression ability and low mutual learning potential. On resnet56, the improvement of Regu & EW is more obvious than that of Regu & Any and Regu & Any, because Regu & EW learns a quantitative network with higher accuracy and stronger expression ability.

For quantitative networks, the pattern is not very obvious. But comparing the results of cifar100, it is found that the two have similarities. First, for Regu & Re, the accuracy improvement of the quantization network on resnet56 and resnet20 are both obvious. Second, for Regu & EW, only the accuracy improvement of the quantization network on resnet56 is obvious. Finally, for Regu & Any, the accuracy improvement of the quantization network on resnet56 and resnet20 are both not obvious. The reasons for this may be the following. The reasons for this situation may be as follows. First, the algorithm design of Any is more reasonable and effective, so there is not much room for improvement through mutual learning. It can be seen that its 1bit quantization on resnet56 for cifar10 can reach 2bit quantization of EW. Second, the algorithm design of Re is poor, so the room for improvement through mutual learning is larger. Third, the network structure resnet20 is too small, and the expressive ability is limited, so the improvement is not obvious for Regu & EW.

4.5 Analysis of LC

Since there are too many experimental results, we only select the results of the two groups of algorithms

Table 3: mutual learning on Tiny-imagenet

mutual learning	Net	method	$Paccuracy$	$Qaccuracy$	$PParam(compressed/original)$ (M)	p_r
Regu & Re	resnet56	baseline	57.47	32.98	0.50/0.50	1×
		ours	57.68↑0.21	39.97↑6.99	0.50/0.50	1×
	resnet20	baseline	49.36	22.70	0.167/0.167	1×
		ours	49.56↑0.20	28.06↑5.36	0.167/0.167	1×
Regu & EW	resnet56	baseline	57.16	50.77	0.50/0.50	1×
		ours	58.15↑0.99	54.17↑3.40	0.50/0.50	1×
	resnet20	baseline	49.18	37.06	0.167/0.167	1×
		ours	49.40↑0.22	37.64↑0.58	0.167/0.167	1×
Regu & Any	resnet56	baseline	57.25	44.52	0.50/0.50	1×
		ours	57.42↑ 0.17	45.09↑0.57	0.50/0.50	1×
	resnet20	baseline	49.20	35.20	0.167/0.167	1×
		ours	49.44↑0.24	35.48↑0.28	0.167/0.167	1×

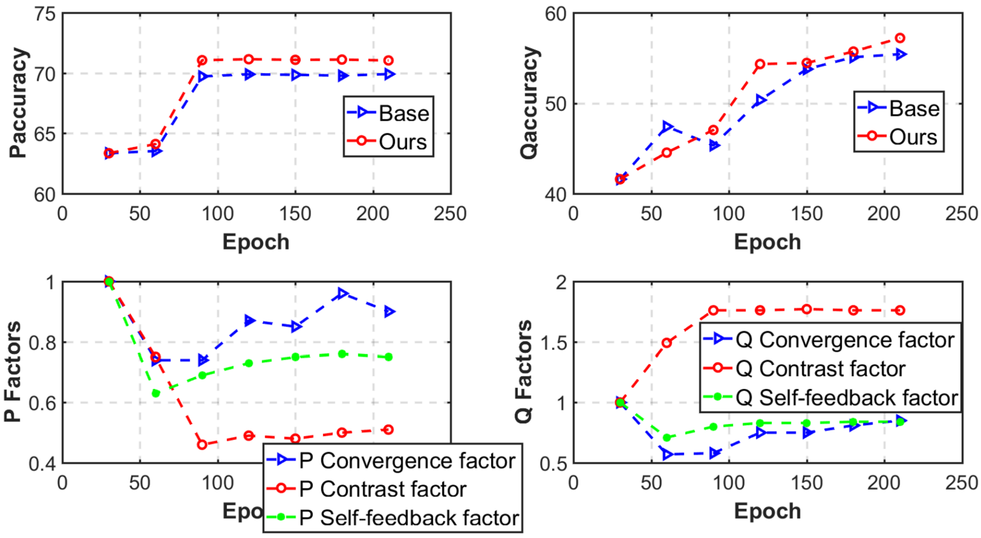


Figure 6: Regu & Any: Trends in accuracy and LC factor during training on resnet56 for cifar100.

related to Regu on resnet56 on cifar100 and Tiny-imagenet for analysis.

Regu & Any As Figure 6 and Figure 7 shown, the trends of Convergence factor(F_{cv}) and Self-feedback factor(F_{sf}) on pnet are similar to qnet on both datasets. Convergence factor(F_{cv}), Contrast factor(F_c), and Self-feedback factor(F_{sf}) all eventually flatten out. Note that the final convergence value of F_{cv} and F_{sf} is different on cifar100 and Tiny-imagenet. On cifar100, the final values of F_{cv} and F_{sf} are larger, both for pnet and qnet. This shows that they are more likely to enter the convergence stage of mutual learning(stage 3). Therefore, the improvement obtained by the mutual learning framework is more obvious. Comparing the accuracy curves of pnet on cifar100 and Tiny-imagenet can have a very intuitive feeling. F_c reflects the gap with the network to be learned. When it is too large, it will cause learning difficulties and affect performance improvement. This is why for qnet, the accuracy improvement on cifar100 is about the same as it on Tiny-imagenet.

Regu & EW As Figure 8 and Figure 9 shown, the trends of Convergence factor(F_{cv}) and Self-feedback factor(F_{sf}) on pnet are also similar to qnet on both datasets. The same as the previous experiment, F_c and F_{sf} gradually level off as the training progresses, the difference is that F_{cv} still maintains an upward trend. This shows that the speed of mutual learning entering the convergence is faster, so the accuracy improvement is more obvious. Comparing Figure 9 and Figure 7, it can be seen that when 3 is larger, the improvement obtained by the mutual learning framework is more obvious. Comparing Figure 8 and Figure 6, it can be found that when F_c is large, it will affect the effect of mutual learning, but when F_{cv} is large enough, this effect will be weakened.

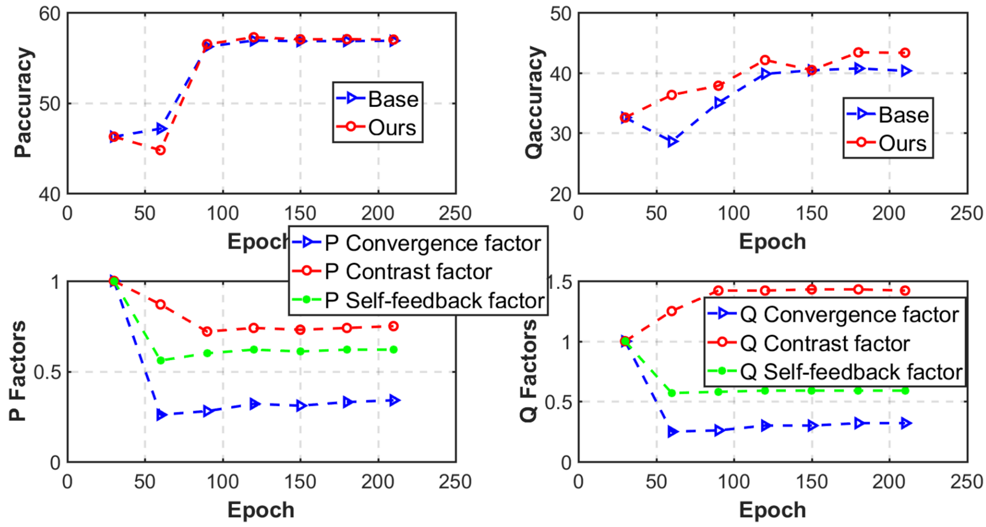


Figure 7: Regu & Any: Trends in accuracy and LC factor during training on resnet56 for Tiny-imagenet.

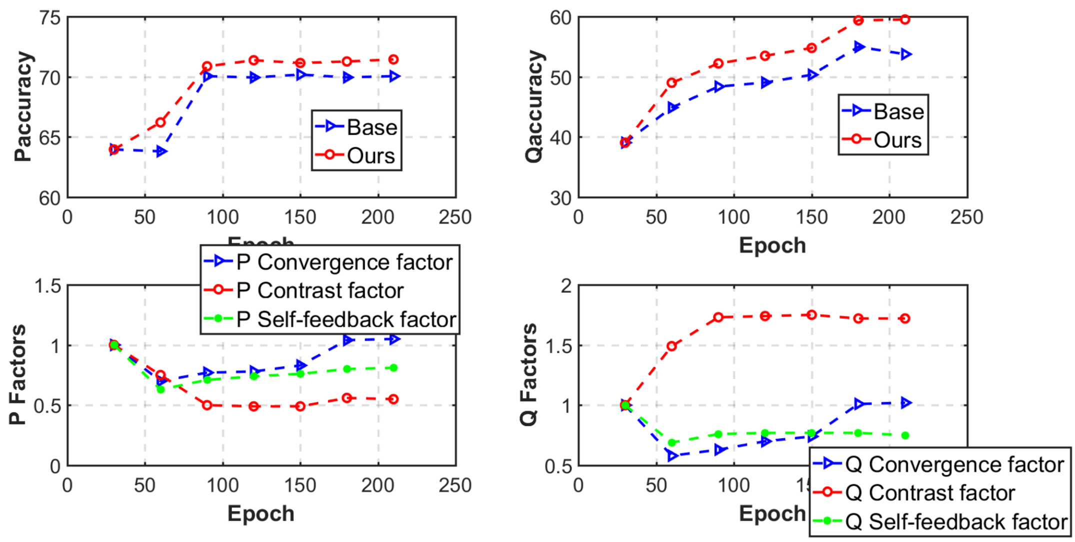


Figure 8: Regu & EW: Trends in accuracy and LC factor during training on resnet56 for cifar100.

4.6 lessons learned

In general, it can be seen from the above experiments that through our mutual learning framework MLPQ, the performance of the pruning algorithm and the quantization algorithm can be improved. The degree of performance improvement is affected by the following factors, network structure complexity, training data complexity, pruning algorithm strategy and quantification algorithm effectiveness. Specifically, when the complexity of the network structure is not enough to support the perfect expression of the training data complexity, the pruned network and the quantization network cannot significantly improve the performance through the mutual learning framework. In addition, the results of dynamic pruning are better than that of finetune after one shot. The former can take advantage of the mutual learning framework to make up for the loss of accuracy caused by the pruning process. Finally, when the performance of the two networks is close and the mutual learning potential is large (neither limited by the network structure and training data complexity, nor far from the optimal expression, such as gRDA & Re on Resnet18 and Resnet34 for cifar100), the performance can be effectively improved through our mutual learning framework.

5 Conclusions

In this paper, we propose a mutual learning framework for pruned and quantized networks. We regard the pruned network and the quantized network as two sets

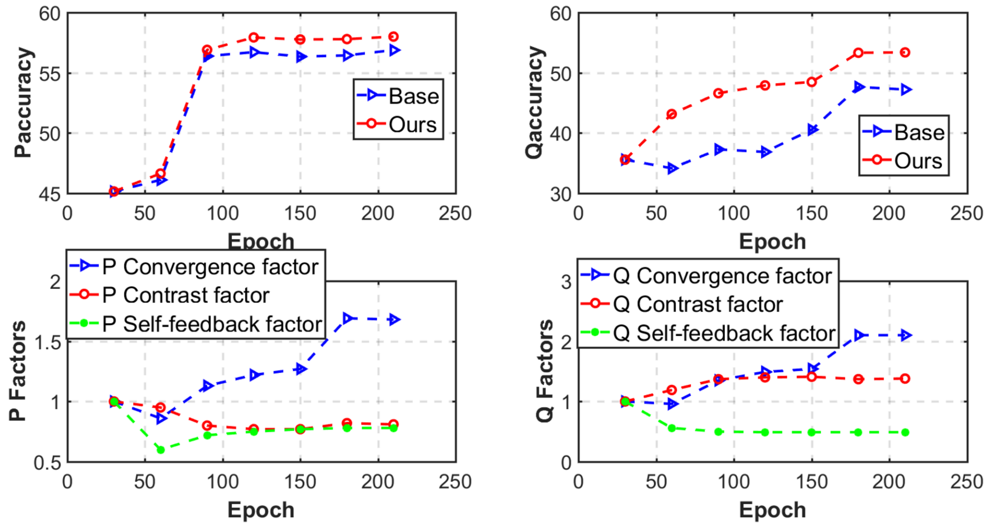


Figure 9: Regu & EW: Trends in accuracy and LC factor during training on resnet56 for Tiny-imagenet.

of features that are not parallel. Our mutual learning framework can better integrate the two sets of features and achieve complementary advantages, which we call feature augmentation. The core of our mutual learning framework is how to judge the training situation of the two networks and how to use the better features of the two networks to learn from each other according to the current training situation. To this end, we designed the score to determine the learning situation of the network and LC coefficient to dynamically adjust the learning strategies of the two networks. Extensive experiments prove that the performance of pruning and quantization algorithms can be improved by our mutual learning framework. When certain conditions are met, the improvement effect is particularly obvious.

Competing interests

The authors have declared that no competing interests exist.

Authors' contribution

XL wrote the program, conducted the experiments, analyzed the results and wrote the manuscript; XL and JW conceived the idea, conducted the experiments and analyzed the results; XL, JW and YC analyzed the results and revised the manuscript. All authors read and approved the final manuscript.

Acknowledgements

This work was supported by National Key Research Development Program of ChinaNo. 2020YFC2007104 Natural Science Foundation of China No.61972383, Science and Technology Service Network Initiative, Chinese Academy of Sciences (No. KFJ-STS-QYZD-2021-11-001), the science and technology innovation Program of Hunan Province.

References

[1] A. Ankit, T. Ibrayev, A. Sengupta, and K. Roy, "Transsformer: Clustered pruning on crossbar-based architectures for energy-efficient neural networks," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2361–2374, 2020.

[2] S. Wang, H. Cai, J. A. Bilmes, and W. S. Noble, "Training compressed fully-connected networks with a density-diversity penalty," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[3] F. Meng, H. Cheng, K. Li, H. Luo, X. Guo, G. Lu, and X. Sun, "Pruning filter in filter," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[4] W. Wang, C. Fu, J. Guo, D. Cai, and X. He, "COP: customized deep model compression via regularized correlation-based filter-level pruning," *CoRR*, vol. abs/1906.10337, 2019.

[5] Y. Woo, D. Kim, J. Jeong, Y. W. Ko, and J. Lee, "Zero-keep filter pruning for energy efficient deep neural network," in *International Conference on Information and Communication Technology Convergence, ICTC 2020, Jeju Island, Korea (South), October 21-23, 2020*, pp. 1288–1292, IEEE, 2020.

[6] C. Zhao, B. Ni, J. Zhang, Q. Zhao, W. Zhang, and Q. Tian, "Variational convolutional neural network pruning," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 2780–2789, Computer Vision Foundation / IEEE, 2019.

[7] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in *5th International Conference on*

Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.

[8] B. Dai, C. Zhu, B. Guo, and D. P. Wipf, "Compressing neural networks using the variational information bottleneck," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (J. G. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 1143–1152, PMLR, 2018.

[9] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 1135–1143, 2015.

[10] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds.), pp. 2074–2082, 2016.

[11] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient dnns," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds.), pp. 1379–1387, 2016.

[12] X. Ding, G. Ding, X. Zhou, Y. Guo, J. Han, and J. Liu, "Global sparse momentum SGD for pruning very deep neural networks," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada* (H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds.), pp. 6379–6391, 2019.

[13] V. Sanh, T. Wolf, and A. M. Rush, "Movement pruning: Adaptive sparsity by fine-tuning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[14] Y. Ro and J. Y. Choi, "Autolr: Layer-wise pruning and auto-tuning of learning rates in fine-tuning of deep networks," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 2486–2494, AAAI Press, 2021.

[15] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017.

[16] J. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 5068–5076, IEEE Computer Society, 2017.

[17] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 1398–1406, IEEE Computer Society, 2017.

[18] M. Lin, R. Ji, Y. Wang, Y. Zhang, B. Zhang, Y. Tian, and L. Shao, "Hrank: Filter pruning using high-rank feature map," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 1526–1535, Computer Vision Foundation / IEEE, 2020.

[19] J. Chen, Z. Zhu, C. Li, and Y. Zhao, "Self-adaptive network pruning," in *Neural Information Processing - 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12-15, 2019, Proceedings, Part I* (T. Gedeon, K. W. Wong, and M. Lee, eds.), vol. 11953 of *Lecture Notes in Computer Science*, pp. 175–186, Springer, 2019.

[20] Y. Wang, X. Zhang, X. Hu, B. Zhang, and H. Su, "Dynamic network pruning with interpretable layer-wise channel selection," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 6299–6306, AAAI Press, 2020.

[21] S. Chao, Z. Wang, Y. Xing, and G. Cheng, "Directional pruning of deep neural networks," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[22] I. Jung, K. You, H. Noh, M. Cho, and B. Han, "Real-time object tracking via meta-learning: Efficient model adaptation and one-shot channel pruning," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 11205–11212, AAAI Press, 2020.

[23] M. S. Zhang and B. C. Stadie, "One-shot pruning of recurrent neural networks by jacobian spectrum evaluation," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.

[24] H. Tanaka, D. Kunin, D. L. K. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems*

2020, NeurIPS 2020, December 6-12, 2020, virtual (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[25] T. Jiang, X. Yang, Y. Shi, and H. Wang, "Layer-wise deep neural network pruning via iteratively reweighted optimization," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 5606–5610, IEEE, 2019.

[26] M. Courbariaux, Y. Bengio, and J. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 3123–3131, 2015.

[27] S. Darabi, M. Belbahri, M. Courbariaux, and V. P. Nia, "BNN+: improved binary network training," *CoRR*, vol. abs/1812.11800, 2018.

[28] H. Phan, D. Huynh, Y. He, M. Savvides, and Z. Shen, "Mobinet: A mobile binary network for image classification," in *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pp. 3442–3451, IEEE, 2020.

[29] A. Shekhovtsov, V. Yanush, and B. Flach, "Path sample-analytic gradient estimators for stochastic binary networks," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[30] D. Kim, K. P. Singh, and J. Choi, "Learning architectures for binary networks," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XII* (A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, eds.), vol. 12357 of *Lecture Notes in Computer Science*, pp. 575–591, Springer, 2020.

[31] A. Bulat, B. Martínez, and G. Tzimiropoulos, "High-capacity expert binary networks," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[32] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "HAQ: hardware-aware automated quantization with mixed precision," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8612–8620, Computer Vision Foundation / IEEE, 2019.

[33] S. Uhlich, L. Mauch, F. Cardinaux, K. Yoshiyama, J. A. García, S. Tiedemann, T. Kemp, and A. Nakamura, "Mixed precision dnns: All you need is a good parametrization," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, OpenReview.net, 2020.

[34] X. Liu, M. Ye, D. Zhou, and Q. Liu, "Post-training quantization with multiple points: Mixed precision without mixed precision," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 8697–8705, AAAI Press, 2021.

[35] L. Yang and Q. Jin, "Fracbits: Mixed precision quantization via fractional bit-widths," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 10612–10620, AAAI Press, 2021.

[36] H. Yang, L. Duan, Y. Chen, and H. Li, "BSQ: exploring bit-level sparsity for mixed-precision neural network quantization," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[37] A. Goncharenko, A. Denisov, S. Alyamkin, and E. Terentev, "On practical approach to uniform quantization of non-redundant neural networks," in *Artificial Neural Networks and Machine Learning - ICANN 2019: Deep Learning - 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17-19, 2019, Proceedings, Part II* (I. V. Tetko, V. Kurková, P. Karpov, and F. J. Theis, eds.), vol. 11728 of *Lecture Notes in Computer Science*, pp. 349–360, Springer, 2019.

[38] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave gaussian quantization," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5406–5414, IEEE Computer Society, 2017.

[39] J. Faraone, N. J. Fraser, M. Blott, and P. H. W. Leong, "SYQ: learning symmetric quantization for efficient deep neural networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 4300–4309, Computer Vision Foundation / IEEE Computer Society, 2018.

[40] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015* (F. R. Bach and D. M. Blei, eds.), vol. 37 of *JMLR Workshop and Conference Proceedings*, pp. 1737–1746, JMLR.org, 2015.

[41] S. Jung, C. Son, S. Lee, J. Son, J. Han, Y. Kwak, S. J. Hwang, and C. Choi, "Learning to quantize deep networks by optimizing quantization intervals with task loss," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 4350–4359, Computer Vision Foundation / IEEE, 2019.

[42] Z. Liao, R. Couillet, and M. W. Mahoney, "Sparse quantized spectral clustering," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[43] E. Park, J. Ahn, and S. Yoo, "Weighted-entropy-based quantization for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 7197–7205, IEEE Computer Society, 2017.

[44] E. Park, S. Yoo, and P. Vajda, "Value-aware quantization for training and inference of neural networks," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), vol. 11208 of *Lecture Notes in Computer Science*, pp. 608–624, Springer, 2018.

[45] P. Gysel, J. J. Pimentel, M. Motamedi, and S. Ghiasi, "Ristretto: A framework for empirical study of resource-efficient inference in convolutional neural networks," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 29, no. 11, pp. 5784–5789, 2018.

[46] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds.), pp. 4107–4115, 2016.

[47] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), vol. 9908 of *Lecture Notes in Computer Science*, pp. 525–542, Springer, 2016.

[48] S. A. Tailor, J. Fernández-Marqués, and N. D. Lane, "Degree-quant: Quantization-aware training for graph neural networks," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[49] Y. Cai, Z. Yao, Z. Dong, A. Gholami, M. W. Mahoney, and K. Keutzer, "Zeroq: A novel zero shot quantization framework," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 13166–13175, Computer Vision Foundation / IEEE, 2020.

[50] Y. Choukroun, E. Kravchik, F. Yang, and P. Kisilev, "Low-bit quantization of neural networks for efficient inference," in *2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019*, pp. 3009–3018, IEEE, 2019.

[51] X. He and J. Cheng, "Learning compression from limited unlabeled data," in *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), vol. 11205 of *Lecture Notes in Computer Science*, pp. 778–795, Springer, 2018.

[52] E. Meller, A. Finkelstein, U. Almog, and M. Grobman, "Same, same but different: Recovering neural network quantization error through weight factorization," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 4486–4495, PMLR, 2019.

[53] M. Nagel, M. van Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 1325–1334, IEEE, 2019.

[54] R. Zhao, Y. Hu, J. Dotzel, C. D. Sa, and Z. Zhang, "Improving neural network quantization without retraining using outlier channel splitting," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 7543–7552, PMLR, 2019.

[55] F. Tung and G. Mori, "CLIP-Q: deep network compression learning by in-parallel pruning-quantization," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 7873–7882, Computer Vision Foundation / IEEE Computer Society, 2018.

[56] T. Wang, K. Wang, H. Cai, J. Lin, Z. Liu, H. Wang, Y. Lin, and S. Han, "APQ: joint search for network architecture, pruning and quantization policy," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 2075–2084, Computer Vision Foundation / IEEE, 2020.

[57] Y. Wang, Y. Lu, and T. Blankevoort, "Differentiable joint pruning and quantization for hardware efficiency," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIX* (A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, eds.), vol. 12374 of *Lecture Notes in Computer Science*, pp. 259–277, Springer, 2020.

[58] M. van Baalen, C. Louizos, M. Nagel, R. A. Amjad, Y. Wang, T. Blankevoort, and M. Welling, "Bayesian bits: Unifying quantization and pruning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[59] J. Kim, K. Yoo, and N. Kwak, "Position-based scaled gradient for model quantization and pruning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[60] S. Chao, Z. Wang, Y. Xing, and G. Cheng, "Directional pruning of deep neural networks," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), 2020.

[61] Y. Li, S. Lin, J. Liu, Q. Ye, M. Wang, F. Chao, F. Yang, J. Ma, Q. Tian, and R. Ji, "Towards compact cnns via collaborative compression," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 6438–6447, Computer Vision Foundation / IEEE, 2021.

[62] H. Wang, C. Qin, Y. Zhang, and Y. Fu, "Neural pruning via growing regularization," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021.

[63] Z. Xu, M. Lin, J. Liu, J. Chen, L. Shao, Y. Gao, Y. Tian, and R. Ji, "Recu: Reviving the dead weights in binary neural networks," in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 5178–5188, IEEE, 2021.

[64] H. Yu, H. Li, H. Shi, T. S. Huang, and G. Hua, "Any-precision deep neural networks," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 10763–10771, AAAI Press, 2021.

[65] J. Lee, D. Kim, and B. Ham, "Network quantization with element-wise gradient scaling," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 6448–6457, Computer Vision Foundation / IEEE, 2021.

Citation: X. Li, Y. Chen and J. Wang. *A Mutual Learning Framework for Pruned and Quantized Networks*. Journal of Computer Science & Technology, vol. 23, no. 1, pp. 1–17, 2023.
DOI: 10.24215/16666038.23.e01
Received: July 12, 2022 **Accepted:** November 28, 2022.