Greedy Search Algorithm for Mixed Precision in Post-Training Quantization of Convolutional Neural Network Inspired by Submodular Optimization

Satoki Tsuji TSUJI.SATOKI@FUJITSU.COM
Fujitsu Research, Fujitsu Ltd., 4-1-1 Kamikodanaka, Nakahara Ward, Kawasaki City, 2118588, Kanagawa Prefecture, Japan

Hiroshi Kawaguchi KAWAPY@GODZILLA.KOBE-U.AC.JP
Graduate School of Science, Technology and Innovation, Kobe University, 1-1 Rokkoudai, Nada Ward, Kobe City, 6578501, Hyogo Prefecture, Japan

Atsuki Inoue AINOUE@GODZILLA.KOBE-U.AC.JP
Graduate School of Science, Technology and Innovation, Kobe University, 1-1 Rokkoudai, Nada Ward, Kobe City, 6578501, Hyogo Prefecture, Japan

Yasufumi Sakai SAKAIYASUFUMI@FUJITSU.COM
Fujitsu Research, Fujitsu Ltd., 4-1-1 Kamikodanaka, Nakahara Ward, Kawasaki City, 2118588, Kanagawa Prefecture, Japan

Fuyuka Yamada YAMADA.FUYUKA@FUJITSU.COM
Fujitsu Research, Fujitsu Ltd., 4-1-1 Kamikodanaka, Nakahara Ward, Kawasaki City, 2118588, Kanagawa Prefecture, Japan

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

For lower bit-widths such as less than 8-bit, many quantization strategies include re-training in order to recover accuracy degradation. However, the re-training works against rapid deployment for wide distribution of quantized models. Therefore, post-training quantization has been getting more attention in recent years. In one example, partial quantization according to the layer sensitivity based on the accuracy after each quantization has been proposed; however, the effects of one layer quantization on the other layers has not taken into account. To further reduce the accuracy degradation, we propose a quantization scheme that considers the effects by continuously updating the accuracy after each layer quantization. Additionally, for more data compression, we extend that scheme to mixed precision, which applies a layer-by-layer fitted bit-width. Since the search space for bit allocation per layer increases exponentially with the number of layers N, existing methods require computationally intensive approach such as network training. Here, we derive practical solutions to the bit allocation problem in polynomial time $O(N^2)$ using a deterministic greedy search algorithm inspired by submodular optimization without any training. For example, the proposed algorithm completes a search on ResNet18 for ImageNet in 1 hour for a single GPU. Compared to the case without updating the layer sensitivity, our method improves the accuracy of the quantized model by more than 1% with multiple convolutional neural networks. For examples, 6-bit quantization of MobileNetV2 achieves 80.1% reduction of model size with -1.10% accuracy degradation. 4-bit quantization of ResNet50 achieves 82.9% size reduction with -0.194% accuracy degradation. Furthermore, results

show that the proposed method reduces the accuracy degradation by more than about 0.7% compared to various latest post-training quantization strategies.

Keywords: Deep learning; Greedy algorithm; Neural network; Quantization; Submodular optimization

1. Introduction

In the last few years, edge computing and edge AI are expected to be realized, with training and inference of deep neural network (DNN) calculated on edge devices such as sensors and smartphones. Because of the enormous amounts of computation necessary for deep learning, such central processing of calculations is commonly done in a cloud server with graphical processing unit (GPU) specialized for matrix operations. Nevertheless, various difficulties persist such as degradation of real-time performance and security robustness attributable to the huge amounts of data traffic. Therefore, to resolve these difficulties by markedly reducing the amount of data communication through edge computing, one must reduce computational costs such as latency, memory bandwidth, and power consumption in deep learning so that computations can run on edge devices with fewer computing resources.

Several model compression techniques have been proposed to reduce neural network computations: pruning (Molchanov et al. (2016)) to reduce the number of parameters, knowledge distillation (Hinton et al. (2015)) to inherit the knowledge of the larger teacher model to the smaller student model, and quantization (Zhou et al. (2016), Jacob et al. (2018)) to perform computations and store tensors such as weights and activations at smaller bit-width than floating point precision (usually 32-bit). Among these techniques, we specifically examine quantization because it is independent of the network architecture and because the development of DNN accelerators supporting smaller bit-width operations such as Tensor Processing Unit Jouppi et al. (2017) and Tensor Core Markidis et al. (2018) are progressing rapidly. Quantization can save the computational costs above and their accompanying needs for storage. However, simultaneously, its data precision loss degrades the inference accuracy. Smaller bit-widths usually entail greater quantization noise and accuracy degradation. Therefore, we aim at improving this tradeoff between the model compression ratio and accuracy degradation.

Broadly speaking, as introduced in a whitepaper Krishnamoorthi (2018), two main quantization methods exist. The first is post-training quantization, a method that is applicable to pre-trained models to obviate the need for re-training. This technique facilitates quantized model deployment that is rapid and simpler to use. However, at smaller bit-widths (e.g. less than 8-bit), the quantization error often becomes too large. Moreover, the inference accuracy drops considerably. The second is quantization aware training, which involves training during quantization to provide higher accuracy even at smaller bit-widths. The latter allows for considerable model compression with slight accuracy degradation, but we adopt the former post-training quantization for decreasing the quantization costs.

A typical approach to maintain high inference accuracy with quantization is to reduce the quantization error. More specifically, the distance between the tensors before and after quantization is minimized to make the quantized model more closely resemble the original one. Existing work related to quantization have uniformly quantized entire models at the same bit-width (e.g. Gupta et al. (2015)). Nevertheless, effects of quantization on inference

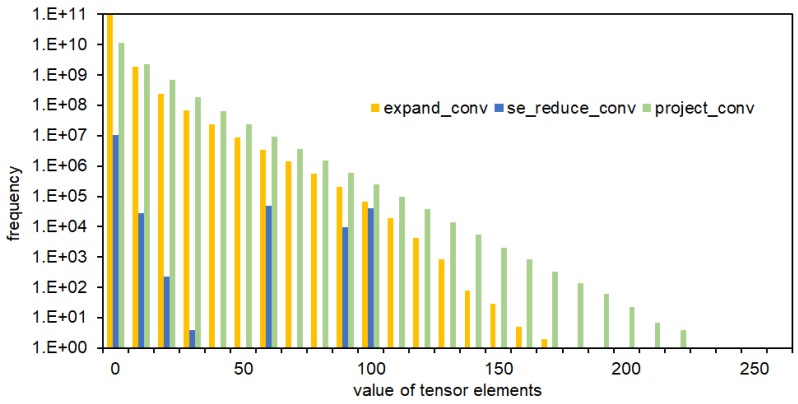


Figure 1: Differences among distributions of activation values for respective layers (in EfficientNet Tan and Le (2019) using weights trained using Xie et al. (2020)).

accuracy are known to differ among tensors such as layers and blocks. For example, as shown in Figure 1, it is apparent that each block has a different distribution of values of tensor elements. Consequently, adjusting the bit-width of each layer or block as the precision of quantization is effective to reduce the quantization error. A quantization scheme that incorporates this strategy is called mixed precision quantization. Various methods of bit-width allocation optimization have been reported. Nevertheless, it is difficult to optimize with a small search cost because the search space for bit-width allocation increases exponentially with the number of layers or blocks.

In light of this background, we propose a greedy post-training quantization scheme for mixed precision that has low computational complexity. The proposed method is so simple that it completes quantization in polynomial time without the need for re-training with back propagation and associated training data. Moreover, it enables model compression equivalent to 4-bit quantization, which was difficult to achieve without degrading accuracy in post-training quantization. Our deterministic search algorithm can improve the tradeoff between model compression and accuracy by greedily selecting the combination of quantization layers which minimizes the accuracy degradation.

The specific contributions of this paper are the following.

1. We propose a post-training quantization scheme for the bit allocation problem of mixed precision whose search space increases exponentially with the number of layers N. Our proposed method deterministically derives practical approximate solutions for the optimization problem in polynomial time of $O(N^2)$ using a greedy algorithm.

2. We present the need to consider the impact of partial quantization on other layers with continuously updating the quantized accuracy included in the evaluation function of the bit allocation problem. In addition, we show that incorporating the number of quantized parameters as a index of model compression into the evaluation function enables a more direct search for quantization-efficient layers.

3. We evaluated the performance of the greedy search algorithm in terms of the tradeoff between inference accuracy and model compression using multiple convolutional neural networks (CNNs) on ImageNet classification. Comparison with latest post-training quantization strategies shows that the proposed method reduces accuracy degradation by more than about 0.7% with 4-bit quantization of ResNet50.

2. Related work

2.1. Post-Training Quantization

Neural network quantization has received increasing attention along with the expansion of edge computing. In fact, various techniques have been proposed. Among them, we introduce several work related to post-training quantization, which can reduce the quantization cost. First, Lee et al. (2018) reported that introducing channel-wise quantization instead of layer-wise quantization reduces the accuracy degradation after 8-bit quantization without fine tuning. Furthermore, Choukroun et al. (2019) showed that a kernel-wise quantization scheme that minimizes the mean squared error is also effective for 4-bit quantization. Furthermore, as one clipping technique, a channel splitting method Zhao et al. (2019) to tackle distortion of the tensor distribution reduced the quantization noise by duplicating channels containing outliers. In terms of eliminating fine-tuning and hyperparameter selection, a data-free quantization Nagel et al. (2019) has also been presented. It achieves 6-bit quantization of MobileNetV2 with only slight accuracy degradation through weight range equalization. In a similar vein, post-training quantization with analytical clipping and per-channel bit allocation Banner et al. (2018) improved the tradeoff between inference accuracy and compression for 4-bit quantization. In studies particularly addressing the loss function of neural networks, Nahshan et al. (2019) use a quantization step size that minimizes cross-entropy loss. Also, Nagel et al. (2020) have proposed the use of approximated loss function for optimization of rounding up or down. In other approach, Wu et al. (2020) recommends trying partial quantization based on layer sensitivity examined individually, before adopting quantization-aware training. Based on these related work and proposals in a white paper Krishnamoorthi (2018), we adopt per-channel quantization for weights and per-layer quantization for activation.

2.2. Mixed precision quantization

Because the tensor value distribution varies considerably for each part such as layers and channels of neural networks, allocating the appropriate bit-widths for each part is known to be effective for reducing the accuracy degradation caused by quantization. In terms of reducing the model size, a model compression technique Han et al. (2015) has been proposed to ascertain the bit-widths for convolution and full connect layers by human heuristics. For layer-wise bit-width optimization, several methods have been presented, such as estimation of the effect of quantization errors in individual layers on the overall inference accuracy (Zhou et al. (2018)), a solution of bit allocation using stochastic gradient descent (SGD) as a neural architecture search (NAS) problem (Wu et al. (2018)), using reinforcement learning with feedbacks of actual edge devices (Wang et al. (2019)), and automatically selecting the relative quantization bit-width of each layer based on the approximated Hessian spectrum

(Dong et al. (2019)). Various bit-widths are available (e.g. 1–8 bits) at each layer in mixed precision quantization. One can quantize neural networks without overfitting or underfitting, but many solutions of a layer-wise bit-width optimization problem entail considerably large computational costs because the search spaces for bit allocation are exponential in the number of layers. Therefore, we specifically examine a simple optimization approach with constrained computational complexity to generate a model quantized by mixed precision.

3. Approach

This section presents our proposed method for resolving difficulties of mixed precision quantization without retraining. First, we relax the layer-wise bit allocation problem and formulate it into a simple combinatorial optimization problem. Next, we describe submodular optimization, which is an effective approach to the combinatorial optimization problem. Subsequently, we propose a greedy search algorithm inspired by submodular optimization. Finally, we describe the quantization scale (and step), rounding, and the granularity of quantization applied in our experiments.

3.1. Problem formulation

The problem of layer-wise mixed precision quantization can be rephrased as which layers should be quantized and with what bit-widths to avoid degrading the inference accuracy of the quantized model. In this case, the search space is M^N, which is exponential with respect to the number of layers N. Also, M represents the number of candidates for the quantization bit-width (e.g. 1–8 bits). In this huge search space, reaching the optimal or approximate solution with low computation time is difficult. Therefore, we relax this search space to 2^N ($M = 2$) by constraining the number of candidates for quantization of bit-width. This scaling down enables us to emphasize those decisions to quantize or not for each layer. The problem can be simplified to a basic combinatorial optimization. For bit-width quantization, we adopt 8-bit as the baseline and 4-bit or 6-bit as the smaller bit-width. The reason for the former is that recent work have achieved post-training 8-bit quantization with almost no accuracy degradation (e.g. less than 1%). The latter is that quantization of less than 4-bit without quantization aware training tends to lead to great amounts of accuracy degradation. We approach the mixed precision quantization problem by starting with a baseline model in which all layers are in an 8-bit quantized state and by choosing layers to be quantized with a smaller bit-width.

Under the approach described above, the relaxed bit allocation problem can be regarded as a combinatorial optimization problem that entails determination of a combination of quantization layers that maximizes the inference accuracy. For example, we can formulate the problem as follows:

$$\begin{aligned} &\text{arg max} && \text{acc}(\boldsymbol{S}) \\ &\text{subject to} && |\boldsymbol{S}| = k \quad (k = 0, \cdots, N), \end{aligned} \tag{1}$$

by defining the objective function as accuracy. In that equation, $\boldsymbol{S}$ denotes the set of layers to be quantized at the smaller bit-width (4-bit or 6-bit). Also, $\text{acc}(\boldsymbol{S})$ is a set function that denotes the inference accuracy when layers $\boldsymbol{S}$ are quantized. Therein, k denotes the number

of quantization layers $\boldsymbol{S}$. Here, with the definition of $\boldsymbol{V}$ as a universal set of all layers a neural network to be quantized has, we apply 8-bit quantization to the layers $\boldsymbol{V} \setminus \boldsymbol{S}$. Problem formulation in Equation 1 maintains accuracy by searching for combinations of layers to be quantized at a smaller bit-width. However, because we aim at improving both the model compression ratio and accuracy through quantization simultaneously, Equation 1 is not a direct formulation. For more efficient quantization, we redefine the objective function in Equation 1 as follows:

$$\begin{aligned} &\text{arg max} && \text{acc}(\boldsymbol{S}) \times (\log \text{params}(\boldsymbol{S}))^{\beta} \\ &\text{subject to} && |\boldsymbol{S}| = k \quad (k = 0, \cdots, N), \end{aligned} \tag{2}$$

by introducing a factor $\text{params}(\boldsymbol{S})$ that denotes the number of quantized parameters where β is a coefficient to tune the relative importance of the quantized parameters as a model compression ratio. By adjusting β, one can search for quantization layers according to the priorities of model compression and accuracy. When $\beta = 0$, Equation 2 is simply a problem of maximizing accuracy without consideration of the number of quantized parameters as a factor related to model size. The formulation of Equation 2 can improve not only the accuracy degradation but also the model compression ratio through quantization.

3.2. Submodular optimization

Submodular optimization, which uses discrete structures captured as convexity of set functions, is widely regarded as an effective approach to combinatorial optimization problems such as sensor placement (Krause et al. (2008)), graph mining (Thoma et al. (2009)), and active learning (Hoi et al. (2006)). Even for combinatorial optimization problems with a search space increasing exponentially, submodular optimization is often able to derive practical solutions in polynomial time. Particularly for maximization problems of a monotonic set function with submodularity, it is theoretically guaranteed that a greedy algorithm can derive an approximate solution with at least a minimum ratio of $1 - \frac{1}{e}$ (≈ 0.632) between the approximate value and the optimal value (Nemhauser and Wolsey (1981)). Additionally, a greedy algorithm is known to be able to find empirically or statistically practical and approximate solutions (Lin and Bilmes (2010), Tibshirani (1996)). A mathematical definition of submodularity (diminishing marginal utility) is similar to that shown in the following.

$$\begin{gathered} f(\boldsymbol{A} \cup \{j\}) - f(\boldsymbol{A}) \geq f(\boldsymbol{B} \cup \{j\}) - f(\boldsymbol{B}) \\ (\forall \boldsymbol{A} \subseteq \boldsymbol{B} \subseteq \boldsymbol{V}, \quad \forall j \in \boldsymbol{V} \setminus \boldsymbol{B}) \end{gathered} \tag{3}$$

Therein, $f(\cdot)$ denotes an arbitrary set function. Equation 3 shows that, with regard to sets $\boldsymbol{A}$ and $\boldsymbol{B}$ in the inclusion relation, the difference when adding a new element j is greater for the smaller set $\boldsymbol{A}$ than $\boldsymbol{B}$. A set function that satisfies Equation 3 increases less as the size of the input set expands. For example, information for humans entails such submodularity by putting information to input sets and the value of information to $f(\cdot)$ in Equation 3. Because more precise parameters of neural networks such as weights and activations are associated with the higher inference accuracy, we apply the greedy algorithm to Equation 2, assuming that the precision of parameters also has close properties to Equation 3 by putting quantization layers $\boldsymbol{S}$ to the input sets and the objective function in Equation 2 to $f(\cdot)$. In other words, we seek combinations of layers quantized at a small bit-width through a greedy algorithm that imitates submodular optimization.

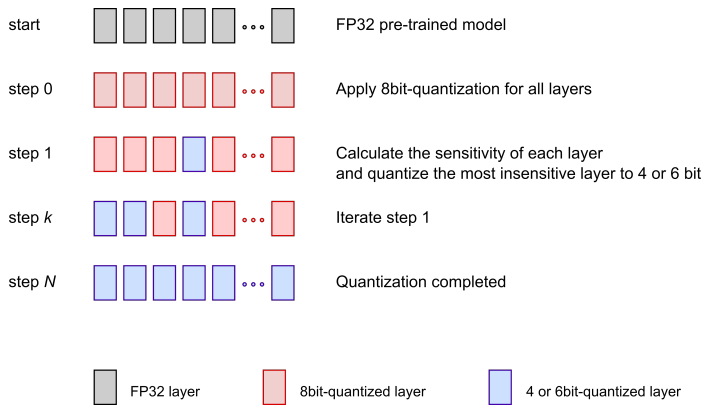


Figure 2: Overview of Greedy search algorithm for mixed precision quantization.

3.3. Greedy search algorithm

We propose a greedy search algorithm inspired by submodular optimization to derive practical and approximate solutions of the combinatorial optimization problem Equation 2. Our proposed algorithm presented in Figure 2 is simple: an algorithm that greedily selects a quantization layer of smaller bit-width (4-bit or 6-bit) at a time until quantization of all layers is completed. We set 4-bit or 6-bit in advance for the smaller bit-width used in the algorithm as step 0, according to quantization tolerance of the target model. The first step is to prepare a pre-trained FP32 model and quantize all its layers to INT8. The second step is to calculate (involve inference) the objective function in Equation 2 for all 8-bit quantized layers, select one layer with the largest value, and quantize it to smaller bit-width. This second step is iterated until all layers have been quantized to smaller bit-width. In other words, the layers are selected in order of quantization efficiency as defined by the objective function in Equation 2. Details of this proposed algorithm are presented in Algorithm 1. By plotting the inference accuracy after quantizing the selected layer to the smaller bit-width at each step, we eventually obtain an improved tradeoff between quantization progress as the model size and accuracy. Users can choose desirable quantization architectures of bit allocation among the models shown for each computational resource. For search space 2^N, the computational complexity of the proposed algorithm is only $O(N^2)$. The search is completed in $\frac{1}{2}N(N+1)$ times inference. For example, the greedy search algorithm finds the quantization architecture of MobileNetV2 Sandler et al. (2018) in about 2 hours on a GPU using 50,000 images for validation of the ImageNet-1k dataset Deng et al. (2009).

3.4. Quantization function and granularity

This paragraph describes quantization details applied in our experiments to search for the bit allocation of each layer. We adopted linear quantization with zero bias (also called the offset, or zero-point). First, for the quantization scale, which is the range of values to which rounding is applied, we used multiples of a power of 2, as following:

$$scale = 2^e(2^{bit} - 1), \tag{4}$$

Algorithm 1 Greedy search algorithm for mixed precision quantization.

Input: Pre-trained FP32 model and the set of all its layers $\boldsymbol{V}(|\boldsymbol{V}| = N)$
Output: Ordered set $\boldsymbol{S}$

```
Quantize all layers of input FP32 model to INT8 as baseline
S ← ∅
% S is the set of layers to be quantized at the smaller bit-width (INT4 or 6 bit)
for k = 1, 2, ..., N do
    max_obj = 0
    % max_obj is the maximum value of the objective function in Equation 2
    for j ∈ V \ S do
        if max_obj < acc(S ∪ {j}) × (log params(S ∪ {j}))^β then
            max_obj = acc(S ∪ {j}) × (log params(S ∪ {j}))^β
            quantized_layer = j
        end if
    end for
    Push quantized_layer into S
end for
```

to enable facile quantization in hardware. Also, *bit* denotes the quantization bit-width; 2^e is quantization step, the smallest interval which the quantized value can represent. Step factor e was found using the following two metrics. The first one is min-max based, which sets the range to cover the minimum to maximum values of a distribution to be quantized such as tensor. In this case, we determine the quantization step factor e based on Equation 5 as

$$e = \text{ceil}(\log \frac{\max(\boldsymbol{W}_k) - \min(\boldsymbol{W}_k)}{2^{bit} - 1}), \tag{5}$$

where $\text{ceil}(a)$ is a ceiling function that maps a to the least integer greater than or equal to a and where $\boldsymbol{W}_k$ denotes a tensor to be quantized (such as weight and activation). Next, the second is based on mean squared error as the distance between the distributions before and after quantization, as follows:

$$e = \arg\max_x \text{MSE}(\boldsymbol{W}_k \parallel \text{quantize}(\boldsymbol{W}_k, x)). \tag{6}$$

In that equation, $\text{MSE}(\cdot \parallel \cdot)$ is the mean squared error corresponding to the distance between two tensors. Also, $\text{quantize}(\cdot, x)$ quantizes an input tensor with step factor x and as follows:

$$\text{quantize}(\boldsymbol{W}_k, x) = \text{round}(\frac{\text{clamp}(\boldsymbol{W}_k, scale)}{2^x}) \times 2^x. \tag{7}$$

Therein, $\text{clamp}(\cdot, b)$ is to truncate values into $[\min(\boldsymbol{W}_k), \min(\boldsymbol{W}_k) + b]$ and $\text{round}(\cdot)$ is a rounding function to the nearest even. For all quantization experiments, rounding in Equation 7 is applied.

Next, we describe the granularity to elucidate the quantization scale. In our quantization experiments, per-channel quantization is applied to weights of the target network to

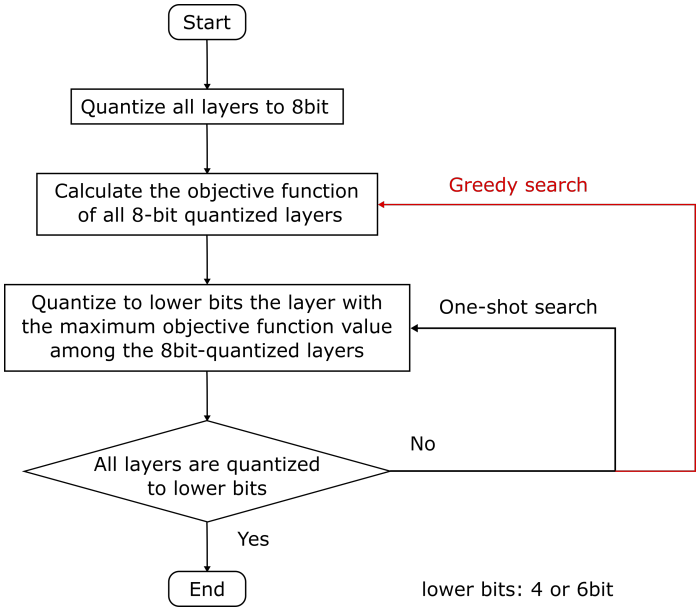


Figure 3: Comparison of One-shot search and Greedy search algorithms.

reduce the quantization error caused by approximation of the parameters. Therefore, the quantization scale is set to an appropriate value for each channel, which is the part of a layer. Here, that scale is found according to the step factor e in Equation 5 covering the min-max of the tensor which is less computational cost than the mean squared error in Equation 6. The reason is that per-channel scaling requires many calculations of scale for the same number of channels the target network has. Separately, we use per-layer quantization for activation. Because the activation tensor size e is many times greater than that of the weight tensors and because its per-channel scaling is a time-consuming process, a quantization scale is set for each layer as larger in granularity than the channel. In addition, because activation tends to be a greater quantization error as a result of its outliers, the scale is found based on the step factor e in Equation 6 using mean squared error to achieve robust quantization against the outliers.

4. Experiment

As described in this section, after applying mixed precision quantization using our proposed algorithm to multiple models, we examine its performance. First, we introduce the algorithm to be compared with the proposed method. Next, we describe the environment and conditions under which we conducted our experiments.

4.1. Experiment terms

A quantization approach that incorporates an idea similar to our proposal has already been proposed. Its layers are quantized in order of less accuracy degradation (Wu et al. (2020)). Using this approach, the accuracy degradation because of quantization is calcu-

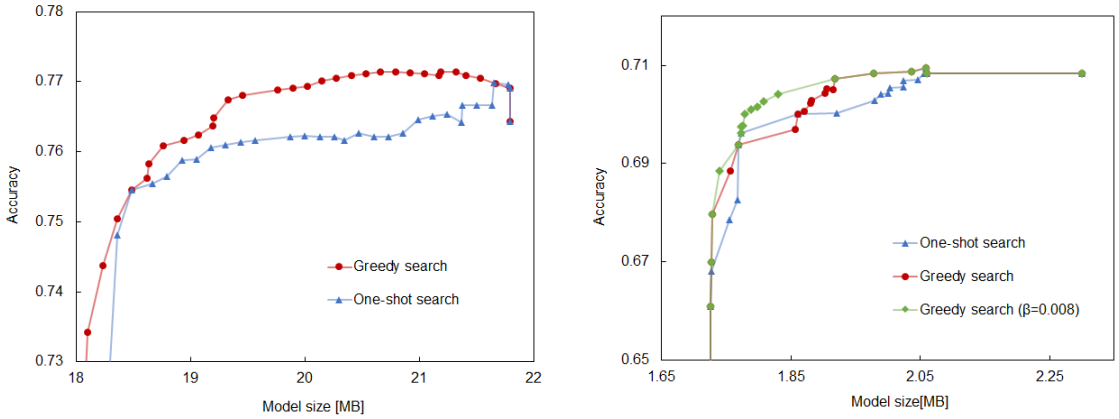
lated only once in advance as the sensitivity (no update) of each layer. The accuracy is maintained by quantizing the layers in order of descending value. We evaluated the quantization performance of the algorithm by extending this one-shot search method to our experimental conditions and by comparing it with our proposed method. Comparison between the flowcharts of the two algorithms is presented in Figure 3. Two main differences exist between the one-shot search and greedy search algorithm. The first is the difference in loop size. The one-shot search continually refers to the pre-calculated objective function value until the end, whereas our greedy search re-calculates and updates the value at each step. Therefore, the latter can incorporate consideration of changes in the objective function value of each layer as the quantization progresses. In other words, the one-shot search is based on approximately the quantization layers derived in step 1 of the greedy search. The second is the difference in objective function. The greedy search also introduces the number of quantized parameters as an index in model compression as in Equation 2, whereas the one-shot search deals only with the inference accuracy, as in Equation 1. By setting the objective function as Equation 2, the greedy search can improve the tradeoff between accuracy degradation and model compression more directly because of quantization. When β equals zero, both objective functions are equivalent. We compared the results found using these two algorithms for layers that should be quantized with the smaller bit-width.

Next, we describe the experiment environment. We evaluated the proposed algorithm for mixed precision quantization in ImageNet classification using various CNNs. All experiments were conducted using the deep learning framework, PyTorch Paszke et al. (2019). We quantized multiple CNNs according to the order of quantization layers searched by Algorithm 1 and showed the inference accuracy for the model size calculated based on the number of quantized parameters. Our experiment starts with 8-bit quantization. It is completed when all layers are quantized to decrease the bit-widths. Before the quantization experiment, we choose in advance whether to set 4-bit or 6-bit as the smaller bit-width for the target networks. The value of the hyperparameter β was set to make the rate of change in both the inference accuracy and quantized parameters approximately equal. More precisely, in step 1 of Algorithm 1, we tuned β to make the ratio of the minimum and maximum values of both factors in Equation 2 approximately equal.

4.2. Result

4.2.1. 6-bit Quantization

First, we present the results of quantizing both weights and activations to 6-bit as the smaller bit-width (W6A6). Figure 4 presents the results of quantization of Xception Chollet (2017) and MobileNetV2. The points shown in the figure reflect the relation between model size and inference accuracy when the layers are quantized one by one according to the order searched by the two algorithms. Therefore, the closer the shown point is to the upper left, the more efficient the model is. Unless otherwise stated, the value of hyperparameter β of the greedy search is 0. Our experiments start with an 8-bit quantized state of the entire model and end at the point where 6-bit quantization has been applied to all layers. As the points show, a marked accuracy degradation is attributable to quantization results from the few layers that are quantized at the end. Therefore, the appropriate combination of quantization layers results in model compression without sacrificing accuracy. In addition,



(a) Xception (b) MobileNetV2

Figure 4: 6-bit quantization (W6A6) with One-shot search and Greedy search of Xception and MobileNetV2.

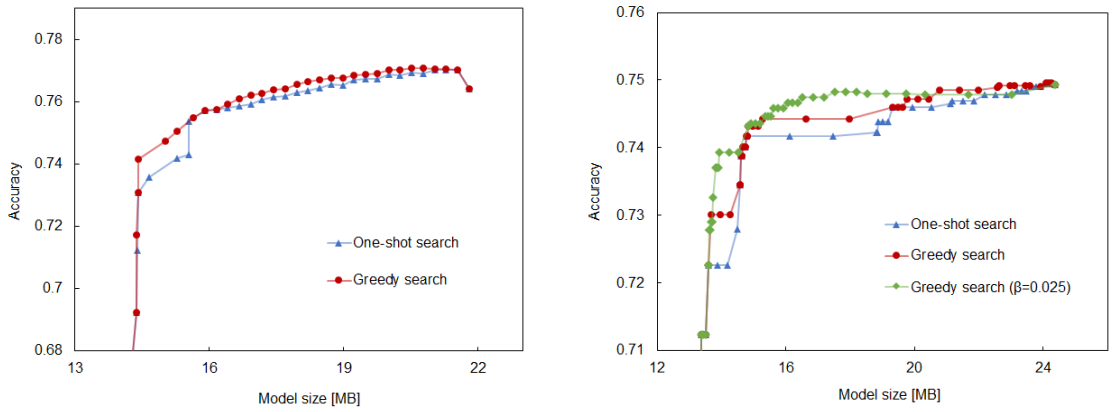
the greedy search finds more efficient combinations of quantization layers than the one-shot search does. More specifically, the greedy search achieved 77.8% reduction of model size with -0.868% accuracy degradation compared to FP32 model (hereinafter, same comparison applied) in Xception 4a. Furthermore, we reached more efficient quantization by introducing a factor that denotes the model compression ratio as in Equation 2. The greedy search with tuned $\beta = 0.008$ achieved 80.1% reduction of model size with -1.10% accuracy degradation in MobileNetV2 4b.

4.2.2. 4-bit Quantization

Second, we present the results of quantizing weights to 4-bit as the smaller bit-width (W4A8). Figure 5 presents results of quantization of Xception and ResNet50 He et al. (2016). In Xception, the accuracy difference in some layers at the end between the two algorithms is greater than 1%. The greedy search achieved 83.5% reduction of model size with -3.44% accuracy degradation 5a. Furthermore, the greedy search of ResNet50 considering the number of quantized parameters reached efficient quantization that generates the lighter and more accurate model. For example, the size of quantized models searched with $\beta = 0.025$ achieved 84.7% size reduction with -0.626% accuracy degradation, or 85.7% size reduction with -1.05% accuracy degradation 5b.

Third, we present the results of applying 4-bit quantization to both weights and activations (W4A4). Figure 6 shows the result obtained for quantization of ResNet50. Changes in inference accuracy for all algorithms show similar behavior to that in Figure 5b. The quantized models searched by our greedy search with $\beta = 0.02$ achieved 83.5% size reduction with -0.414% accuracy degradation or 85.7% size reduction with -1.22% accuracy degradation.

Finally, we compare the proposed algorithm using various strategies used recently for post-training quantization. Figure 7 presents results of applying 4-bit quantization to both weights and activations in ResNet50 using each method (AdaRound Nagel et al. (2020) is



(a) Xception (b) ResNet50

Figure 5: 4-bit quantization (W4A8) with One-shot search and Greedy search of Xception and ResNet50.

W4A8 quantization). The vertical axis shows the accuracy degradation after quantization because the inference accuracy before quantization depends on each experimental environment and differs in the literature. The horizontal axis shows the relative compression ratio, where the size of the original-FP32 model is 1. However, for methods where the precise compression ratio could not be confirmed from the paper, we set a uniform value of 0.125. In this visualization format, points shown in the upper left of the figure are compact and accurate models. This result demonstrates that the proposed quantization using greedy search has a smaller accuracy degradation than other strategies for post-training quantization. More detailed figures can be found in Appendix A

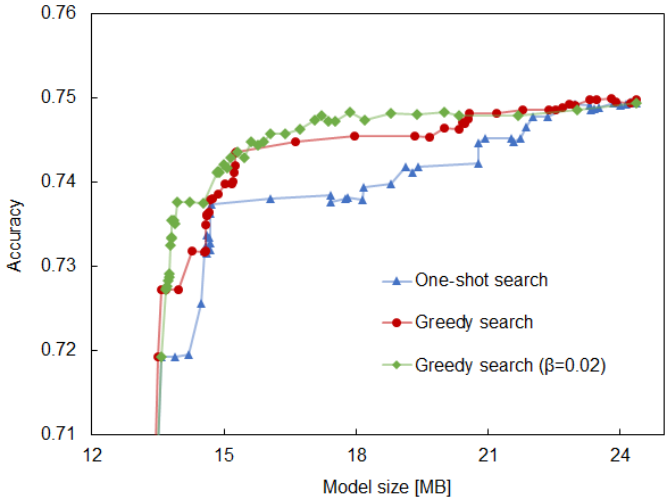


Figure 6: 4-bit quantization (W4A4) with One-shot search and Greedy search of ResNet50.

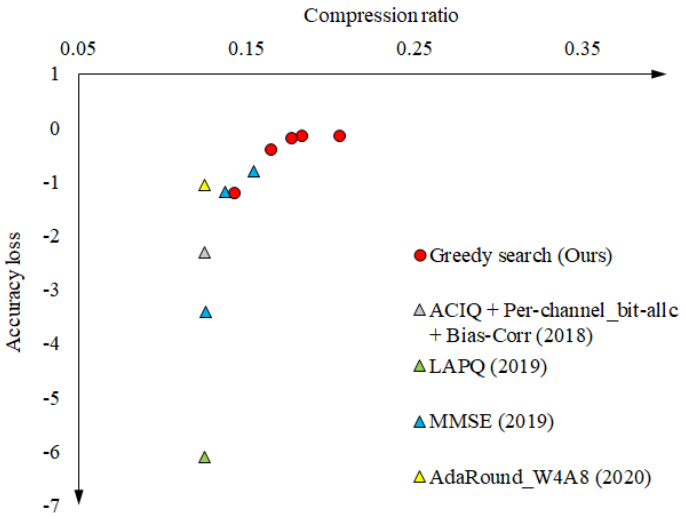


Figure 7: Comparison among state-of-the-art strategies for Post-training quantization in 4-bit quantization (W4A4) of ResNet50.

5. Conclusion

We introduced a mixed precision quantization strategy to realize high inference performance of neural networks using limited hardware resources. For this approach, we introduced a new evaluation function for the per-layer bit allocation problem required by the strategy presented above. Then we proposed a practical greedy approach that completes in polynomial time for post-training quantization. By applying per-channel scaling to the weight parameters simultaneously, the proposed algorithm was able to reduce the accuracy degradation considerably, even for 4-bit quantization with no training. Specifically, the size of MobileNetV2 is 80.1% reduced with -1.10% accuracy degradation in 6-bit quantization, and the size of ResNet50 is 83.5% reduced with -0.414% accuracy degradation in 4-bit quantization. Additionally, we have observed that the quantized models generated by our proposed greedy search outperform the various latest post-training quantization strategies in terms of maintaining inference accuracy. Although the accuracy transitions of our experimentally obtained results do not seem to have strict convexity, they show that the greedy algorithm is sufficiently effective to search for quantization layers.

References

Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. Post-training 4-bit quantization of convolution networks for rapid-deployment. *arXiv preprint arXiv:1810.05723*, 2018.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *ICCV Workshops*, pages 3009–3018, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 293–302, 2019.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR, 2015.

Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Steven CH Hoi, Rong Jin, Jianke Zhu, and Michael R Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424, 2006.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12, 2017.

Andreas Krause, H Brendan McMahan, Carlos Guestrin, and Anupam Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(12), 2008.

Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

Jun Haeng Lee, Sangwon Ha, Saerom Choi, Won-Jo Lee, and Seungwon Lee. Quantization for rapid deployment of deep neural networks. *arXiv preprint arXiv:1810.05488*, 2018.

Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920, 2010.

Stefano Markidis, Steven Wei Der Chien, Erwin Laure, Ivy Bo Peng, and Jeffrey S. Vetter. Nvidia tensor core programmability, performance precision. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 522–531, 2018. doi: 10.1109/IPDPSW.2018.00091.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1334, 2019.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020.

Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *arXiv preprint arXiv:1911.07190*, 2019.

George L Nemhauser and Laurence A Wolsey. Maximizing submodular set functions: formulations and analysis of algorithms. In *North-Holland Mathematics Studies*, volume 59, pages 279–301. Elsevier, 1981.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

Marisa Thoma, Hong Cheng, Arthur Gretton, Jiawei Han, Hans-Peter Kriegel, Alex Smola, Le Song, Philip S Yu, Xifeng Yan, and Karsten Borgwardt. Near-optimal supervised feature selection among frequent subgraphs. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 1076–1087. SIAM, 2009.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8612–8620, 2019.

Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv preprint arXiv:1812.00090*, 2018.

Hao Wu, Patrick Judd, Xiaojie Zhang, Mikhail Isaev, and Paulius Micikevicius. Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*, 2020.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. In *International conference on machine learning*, pages 7543–7552. PMLR, 2019.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

Yiren Zhou, Seyed-Mohsen Moosavi-Dezfooli, Ngai-Man Cheung, and Pascal Frossard. Adaptive quantization for deep neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Appendix A. Comparison on accuracy degradation

Baseline is the accuracy of the original FP32 model used in each paper. W-comp denotes the relative compression ratio, where the size of the FP32 model is 1.

Method	Year	W/A	Baseline[%]	Acc[%]	Acc loss[%]	W-comp
ACIQ+Bit-alloc+Bias-corr	2018	4/4	76.1	73.8	-2.3	0.125
LAPQ	2019	4/4	76.1	70.0	-6.1	0.125
MMSE	2019	4/4	76.012	75.198	-0.814	0.154
AdaRound	2020	4/8	76.07	75.01	-1.06	0.125
Greedy search (ours)	2021	4/4	74.98	74.786	**-0.194**	0.177

Table 1: Comparison on accuracy degradation caused by 4-bit quantization of ResNet50 with latest post-training quantization strategies