Towards Accurate Low Bit DNNs with Filter-wise Quantization

Hoseung Kim, Kwangbae Lee, Dongkun Shin
Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon, Korea
{ghtmd123, kblee93, dongkun}@skku.edu

Abstract

Execution of deep neural networks (DNNs) on a resource constraint device has become a rising issue of recent neural network research. Quantization using low bit-widths for networks is one of the most effective compression techniques. Although there have been many studies that use different bit-widths per layer to compress further the model instead of using a single bit-width, they achieved a limited reduction on the parameter size due to the layer-wise bit-width assignment. In this paper, we propose a more fine-grained and multi-precision quantization technique, called filter-wise quantization. Regularization is used while training networks to partition filters into various precision. In experiments, we show that our technique can provide better accuracy at a smaller parameter size at various DNN models for CIFAR-10 and CIFAR-100 data sets.

Keywords: Deep Neural Networks, Quantization, Regularization

1. Introduction

Recently, deep neural networks (DNNs) have shown a great performance in many tasks, such as autonomous driving, computer vision, and natural language processing. As the DNN architecture continues to go deeper to show high performance, it has more than a million parameters and billions of FLOPs, thus requiring a lot of execution time, energy consumption, and storage spaces. Those factors make it hard to handle real-time applications on resource-constrained devices, such as IoT devices or mobile devices.

The quantization technique is emerging to address this problem, which uses fewer bits to represent the parameters or the feature maps of the DNNs. Various studies have been conducted, such as quantization using only binary or ternary values [1-3], uniform [4-6], and non-uniform [7] quantization techniques. Jacob et al. [4] show that even simple methods such as linear quantization can compress the weights and the activations up to 8 bits without retraining while maintaining the accuracy. However, it is still a challenging problem to quantize to lower bits without loss of accuracy. Some techniques [5,6] quantize a network with different bit-widths per layer. Although this approach can achieve a higher compression ratio than using only a single bit-width across all layers, they achieve a limited reduction on the parameter size due to the layer-wise bit-width assignment. In this paper, we propose a more fine-grained bit-width assignment approach, called filter-wise quantization, to compress the DNNs further than the layer-wise approach. Our technique can reduce the overall model size by finding the filters which have a little impact on the accuracy loss even at a low bit quantization. Also, we use a regularization while training networks to partition filters into various precision. The regularization reduces the weight value range of unimportant filters and minimizes the overall quantization error. By assigning the different quantization bit-width per filters, we can get a more compressed model.

2. Related Work

2.1 Network Pruning

Han et al. [8] introduced a magnitude-based pruning method that prunes the weights lower than a threshold element-wisely. However, because element-wise pruning causes irregular memory access, there was little speed up occurs. To solve this problem, a structured pruning method was proposed, such as channel pruning. Channel pruning can accelerate the model by changing the model's overall structure by pruning all of the weights in each channel. Wen et al. [9] used group lasso [10] to learn sparse structures. Using group lasso, some of the structured groups shrink to zero, and we can get a pruned model.

2.2 Quantization

In order to reduce the model size, Courbariaux et al. [1] first used binary values to quantize weights, showed reasonable performance for MNIST and

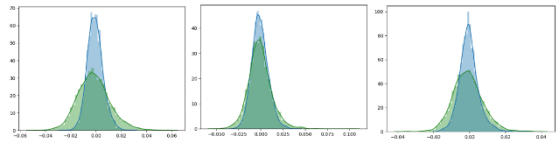

Figure 1: Weight distribution of two filters of 8th, 9th, and 10th layers of VGG-16 on CIFAR-100.

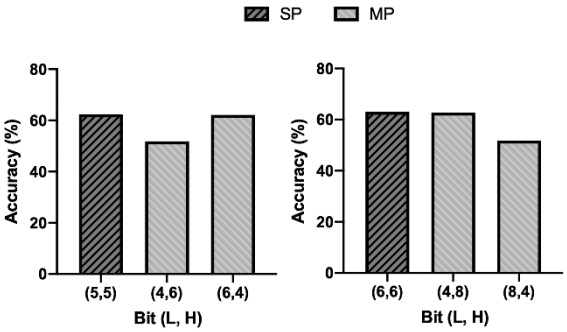

Figure 2: Accuracy of VGG-16 on CIFAR-100 when various bits are used for each filter.

CIFAR-10. To prevent accuracy degradation of binary networks for large datasets, Zhu et al. [2] proposed ternary weight networks that use ternary value and train the scaling factor for each non-zero values. They achieved 42.5% top-1 accuracy with AlexNet on ImageNet. DoReFa-Net [3] proposed a quantization method that can quantize weights, activations, and gradients with any bit-width.

There have been studies representing parameters in the layer-wise bit-width. Lin et al. [5] and Zhou et al. [6] considered the result of quantization as adding noise. They decided the layer-wise bit-width minimizing the quantization noise on the entire network based on the distribution of weights. However, they did not consider a more fine-grained approach, such as filter-wise, nor considered quantization aware retraining.

3. Motivation

To determine the bit-width of each filter, we have to figure out the importance and role of each filter. Figure 1 is the weight distribution of two filters of VGG-16 on some selected layers. As shown in Figure 1, the distribution of each filter is very diverse. If a filter consists of small values, it rarely affects the layer output, and if its value range is short, only small quantization error occurs. It is not necessary to use the same bit-width to these filters as others. By using the appropriate bit-width for each filter, we can minimize the overall parameter size.

We empirically explored the effect of each filter's role by quantizing weights filter-wisely without retraining. Figure 2 is the accuracy of quantized VGG-16 on CIFAR-100 to (L,H) bit. In Figure 2, SP means single-precision, and MP means multi-precision. We considered the filter's largest absolute value as the importance of the filter and used the L bits for the half

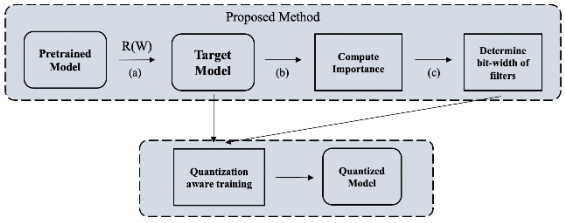

Figure 3: The whole process to get a quantized model.

of the filters with small importance value and the H bits for the others. In Figure 2(a), the accuracy when using 8bit as H bit and 4bit as L bit is 62.75%, more than 10% higher than using 8bit as L bit and 4bit as H bit. On the other hand, Figure 2(b) shows 62.19% accuracy when using 6 bits for L bit and 4 bits for H bit which more than 10% higher accuracy than the opposite case. It shows that we can't determine the importance of the filter with a simple method, and the accuracy varies considerably depending on how many bit-widths are used for which filter.

.

4. Method

In this section, we will discuss the whole process of the proposed method. Firstly, regularization for filter-wise quantization is introduced. Then, we propose an algorithm to assign bit-width to each filter based on the retrained network with a proposed regularization. Finally, we discuss the quantization scheme and other details used in this paper.

4.1 Learning importance of filters

Our regularization term is similar to group lasso [10] to give a penalty filter-wisely. Group lasso is commonly used when training sparse models because of the effect of variable selection to discard non-essential variables. At the same time, it shrinks the absolute value of non-essential variables. Our insight is if we adjust the regularization parameter, the difference between filters can be increased without loss of accuracy.

Before assigning the bit-width of each filter, we apply regularization to the pretrained model to get the target model, as shown in Figure 3(a). The target model means a model trained from a pretrained model to maximize the difference between filters. As will be discussed in section 4.2, from the target model, we derive the importance of filters and initialize the quantized model. Also, to substantially reduce the range of the filter, we weighted regularization term to large values. Consider the weight matrix of $l\text{-}th$ layer $W^{(l)} \in \mathbb{R}^{N^l \times C^l \times K^l \times K^l}$, where N^l, C^l and K^l are the number of filters, input channel size, and kernel size of the $l\text{-}th$ layer, respectively. The regularization term is defined as:

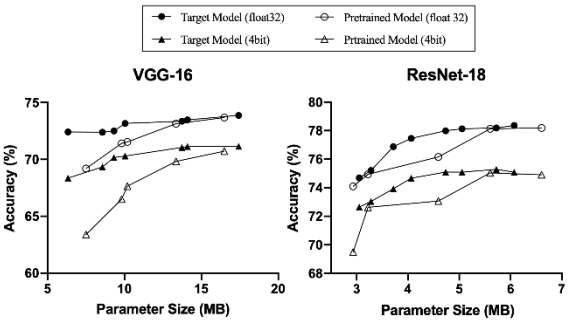

Figure 4: Accuracy of VGG-16 and ResNet-18 with various parameter size.

$$R(W^{(l)}) = \sum_{i=1}^{N^l} \sqrt{\sum_{j=1}^{C^l} \sum_{k=1}^{K^l \times K^l} \frac{\left(w_{i,j,k}^{(l)}\right)^2}{\left\| w_{max_i}^{(l)} - w_{i,j,k}^{(l)} + \lambda \right\|}} \quad (1)$$

The numerator is the group lasso penalty in the filter groups, and the denominator plays the role of increasing the group lasso penalty according to each weight value. If the λ is small, it will give more penalty to the larger weight. By adjusting the λ, it is possible to reduce the actual value range of the filter by avoiding the long-tail distribution caused when applying the group lasso penalty.

4.2 Filter-wise Weight Quantization

We consider the size and range of filters as the importance of quantization. The importance of filter can be determined effectively with the target model because the difference of each filter increased. We define the quantization importance I of $n\text{-}th$ filter of $l\text{-}th$ layer as:

$$I_n^{(l)} = \frac{\left\| W_n^{(l)} \right\|^2}{\sum_{i=1}^{N^l} \left\| W_i^{(l)} \right\|^2} (w_{max_n}^{(l)} - w_{min_n}^{(l)}) \quad (2)$$

Here, $w_{max_n}^{(l)}$ and $w_{min_n}^{(l)}$ and are the maximum and the minimum value of the $n\text{-}th$ filter of $l\text{-}th$ layer. By equation (2), if a filter has larger values than other filters in the same layer and its range is wide, it has a large importance value. Based on the obtained importance, the bit-width of each filter is determined, like Figure 3(c). We specify bit-widths in advance for the filters with the highest quantization importance as b_{max} and the lowest as b_{min}. We determine bit-width in proportion to its importance within a predefined bit-width range as:

$$b_n^{(l)} = \left[\frac{I_n^{(l)} - I_{min}^{(l)}}{I_{max}^{(l)} - I_{min}^{(l)}} (b_{max} - b_{min}) + b_{min} \right] \quad (3)$$

By assigning the bit-width to each filter obtained through all the processes in Figure 3(a) to (c), and initializing with the target model, we can generate a quantized model after retraining.

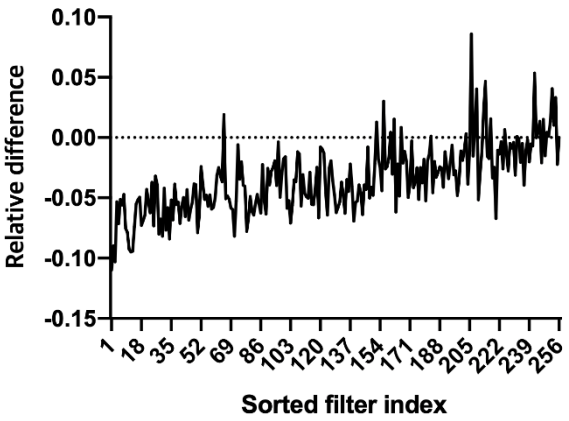

Figure 5: Relative difference of sum of filters after training with regularization on 5th layer of VGG-16 on CIFAR-100.

5. Experiments

5.1 Experimental Details

In experiments, we adopted the same quantization function as Jacob et al. [4]. During backpropagation, we used straight through estimator [11] to propagate gradients. The scaling factor is used as per filter groups using the same bit-width. We employed a small CNN model for CIFAR-10 and used VGG-16 and ResNet-20 for CIFAR-100 to evaluate.

5.2 Performance Evaluation

We compare single precision and filter-wise quantization, and the results are listed in Table 1. The baseline accuracy of CIFAR-10 is 93.24%, VGG16 is 73.81%, ResNet-18 is 78.23% and ResNet-20 is 66.61%. As Table 1 shows, using filter-wise quantization provides better accuracy at smaller parameter sizes. For example, the floating-point activation of VGG-16 shows a more than 1.5% performance improvement compared to quantized to 3bit fixed bit-width, with 15% small parameter size. Since most filters are assigned a small bit-width, the overall parameter size is reduced.

5.3 Effectiveness of proposed method

Figure 4 shows the accuracy of filter-wise quantization in the target model and pretrained model, respectively, of VGG-16 and ResNet-18, on CIFAR-100. The target model always shows better accuracy. In extreme cases, the target model was more than 3% higher accuracy than the pretrained model at a similar parameter size. Figure 5 shows the relative change of the sum of filters normalized between 0 to 1 on the fifth layer of the VGG-16 when trained with the target model. We can see that the difference between the filters increases because the filters that were originally

Table 1: Comparison of the single-precision to proposed filter-wise quantization.

Model	Method	A	W (b_{max}/b_{min})	# of filters for each bit-width	Params (KB)	Acc.
CIFAR10	SP	4	2	-	2271	90.61
			3	-	3407	91.54
	MP		1/9	[1360 938 316 120 49 15 10 9 9]	2001	91.68
			2/9	[1465 936 265 97 30 15 8 10]	3015	92.64
VGG16	SP	4	3	-	12449	69.17
			4	-	16599	70.7
	MP		2/8	[8464 2478 986 412 126 31 19]	10283	70.29
			3/9		14433	71.13
	SP	Float32	3	-	12449	71.58
			4	-	16599	73.44
	MP		2/8	[8464 2478 986 412 126 31 19]	10283	73.16
			3/9		14433	73.47
ResNet-18	SP	4	3	-	4105	73.11
			4	-	5474	74.03
	MP		2/5	[2070 2490 293 47]	3808	73.94
			3/6		5176	75.11
ResNet-20	SP	Float32	3	-	101	62.94
			4	-	135	66.06
	MP		2/3	[601 283]	79	63.32
			3/5	[265 537 82]	131	66.22

small becomes smaller.

6. Conclusion

In this paper, we propose a regularization and filter-wise quantization method. The regularization allows the model to create more low-bit filters by increasing the difference between the filters and can be a guidance of assigning the bit-width of filters. Filter-wise quantization can further compress the model using low bit-width for less important filters. In the experiment, we show better accuracy even with a smaller parameter size than using single bit-width. We will further study the effectiveness of our method in various quantization schemes.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.IITP-2017-0-00914, Software Framework for Intelligent IoT Devices)

References

[1] Courbariaux, Matthieu, Yoshua Bengio, and Jean-Pierre David. "Binaryconnect: Training deep neural networks with binary weights during propagations." *Advances in neural information processing systems.* 2015.

[2] Zhu, Chenzhuo, et al. "Trained ternary quantization." *arXiv preprint arXiv:1612.01064* (2016).

[3] Zhou, Shuchang, et al. "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients." *arXiv preprint arXiv:1606.06160* (2016).

[4] Jacob, Benoit, et al. "Quantization and training of neural networks for efficient integer-arithmetic-only inference." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[5] Lin, Darryl, Sachin Talathi, and Sreekanth Annapureddy. "Fixed point quantization of deep convolutional networks." *International Conference on Machine Learning.* 2016.

[6] Zhou, Yiren, et al. "Adaptive quantization for deep neural network." *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[7] Zhang, Dongqing, et al. "Lq-nets: Learned quantization for highly accurate and compact deep neural networks." *Proceedings of the European conference on computer vision (ECCV).* 2018.

[8] Han, Song, et al. "Learning both weights and connections for efficient neural network." *Advances in neural information processing systems.* 2015.

[9] Wen, Wei, et al. "Learning structured sparsity in deep neural networks." *Advances in neural information processing systems.* 2016.

[10] Yuan, Ming, and Yi Lin. "Model selection and estimation in regression with grouped variables." *Journal of the Royal Statistical Society: Series B* (Statistical Methodology) 68.1 (2006): 49-67.

[11] Bengio, Yoshua, et al."Estimating or propagating gradients through stochastic neurons for conditional computation." *arXiv preprint arXiv:1308.3432* (2013).