AOC 2024 Spring

Paper Review 2

A Review of "Quantization Networks"

施宇庭 NN6124030

1 Motivation

The mainstream approaches of quantization can be divided into two categories: approximation-based and optimization-based. However, approximation-based methods may encounter gradient mismatch issues, while optimization-based methods tend to have higher training costs and are limited to weight compression only.

For example, the reference material [1] for lab 2 in this course adopts an approximation-based approach. Assume we are applying such quantization scheme on a fully-connected layer $y = Wx$ with loss function $L(y, \hat{y})$. In full-precision model, the gradient w.r.t. the weight W can be calculated by

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial W} = \frac{\partial L}{\partial y}x$$

In quantized model $y = Q(W)x$, however, the gradient should be

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial Q}\frac{\partial Q}{\partial W} = \frac{\partial L}{\partial y}x\frac{\partial Q}{\partial W}$$

Since the $Q(\cdot)$ is not differentiable, the approximation must be introduced, and hence cause the gradient of quantized model different from that of full-precision model (i.e. gradient mismatch problem).

Therefore, this article [2] attempts to formulate quantization from a different perspective to avoid the issue of gradient mismatch and proposes an approach suitable for both weight and activation with low training costs.

2 Proposed Method

2.1 Reformulation of Quantization

Inspired by the activation function in deep neural networks, the main idea of this work is to formulate the quantization as a differentiable non-linear function.

$$y = Q(x) = \alpha(\sum_{i=1}^{n} s_i \mathcal{A}(\beta x - b_i) - o) \tag{1}$$

where $x \in \mathbb{R}$ is the full-precision weight or activation, $y \in \mathcal{Y}$ is the quantized integer in a predefined set $\mathcal{Y}$, which has $n+1$ quantization intervals. β is the scale of input. $\mathcal{A}$ is the activation function. s_i and b_i is the scales and biases for the activation function. $s_i = \mathcal{Y}_{i+1} - \mathcal{Y}_i$. The global offset $o = \frac{1}{2}\sum_{i=1}^{n} s_i$ keeps the quantized output zero-centered. $n = |\mathcal{Y}| - 1$. α is the scale of output.

This approach creates the possibility of quantizing both weights and activations during the optimization process. By selecting an appropriate activation function $\mathcal{A}(\cdot)$, it is possible to achieve training with lower costs compared to traditional optimization-based quantization.

2.2 Inference of Quantization Networks

The quantization operation maps continuous real numbers $x \in \mathbb{R}$ to a set of discrete integers $y \in \mathcal{Y}$, which is equivalent to applying a step function (Eq. 2) to Eq. 1.

$$\mathcal{A}_{\text{hard}}(x) = \text{step}(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

2.3 Training of Quantization Networks

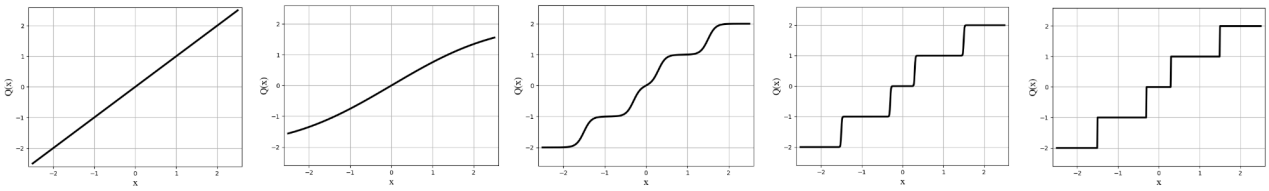
However, since the step function is not differentiable everywhere, it is not possible to directly obtain learnable parameters α, β, and b_i through backward propagation. Therefore, during the training phase, Eq. 3 is used as a replacement for the original step function.

$$\mathcal{A}_{\text{soft}}(x) = \sigma(Tx) = \frac{1}{1 + e^{-Tx}} \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid activation function, and the temperature T introduced by this paper is used to reduce the gap between the ideal/hard quantization function in inference stage and the soft quantization function in training stage.

$$\lim_{T \to \infty} \mathcal{A}_{\text{soft}}(x) = \lim_{T \to \infty} \frac{1}{1 + e^{-Tx}} = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} = \mathcal{A}_{\text{hard}}(x) \tag{4}$$

As T increases, the quantization function $Q(\cdot)$ approaches the ideal quantization function more closely, as shown in Eq. 4 and Fig. 1. However, the learning capacity is lower because most gradients of the quantization function become zero. Therefore, during training, this paper starts with a smaller value of temperature T and gradually increase the temperature (heating) as training progresses. This approach helps maintain better learning capacity in the initial stages of training and aims to closely approximate the ideal quantization function used during inference, thus avoiding the issue of gradient mismatch.



(a) No Quantization (b) T=1 (c) T=11 (d) T=121 (e) Complete Quantization

Figure 1: The quantization functions with different temperature T.

3 Experiment and Evaluation

This paper evaluates the effectiveness of the method using two tasks: image classification and object detection. It also discusses how the quantization operation affects the results. The experimental results for image classification and object detection can be found in the original work [2]. Below, we will discuss the details of the ablation study.

3.1 Ablation Study

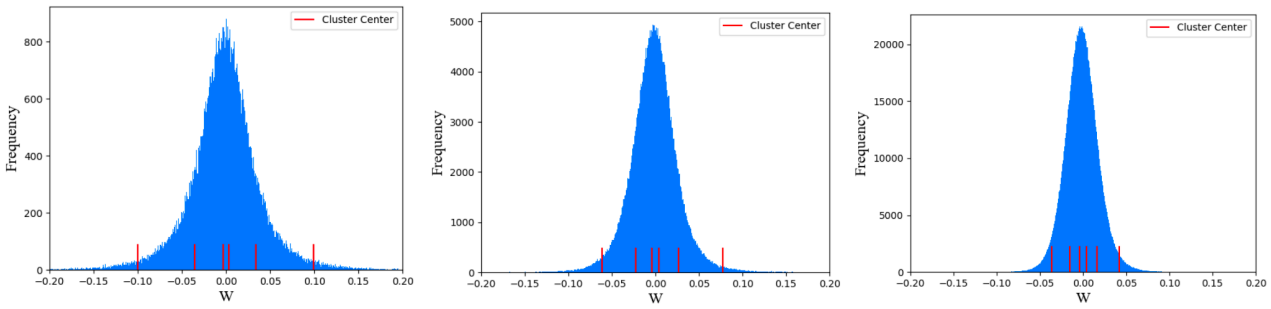
The settings of the quantization network are discussed from training process of AlexNet and ResNet-18 on ImageNet in this section.

3.1.1 Configuration of Bias b

As shown in Fig. 3, the distribution of full-precision parameters of pre-trained model is roughly subjected to Gaussian distribution (neither linear nor logarithmic), a non-uniform quantization (e.g. K-means clustering) is more suitable for this. Fig 2 shows that non-uniform quantization outperforms linear quantization.

Quantization methods	W/A	Top-1	Top-5
linear	2/32	60.6	82.8
non-uniform	2/32	**60.9**	**83.2**
linear	3(±4)/32	60.7	83.0
non-uniform	3(±4)/32	**61.9**	**83.6**

Figure 2: Linear v.s. non-uniform quantization for AlexNet on ImageNet classification.



(a) Conv2-1 (b) Conv3-1 (c) Conv4-1

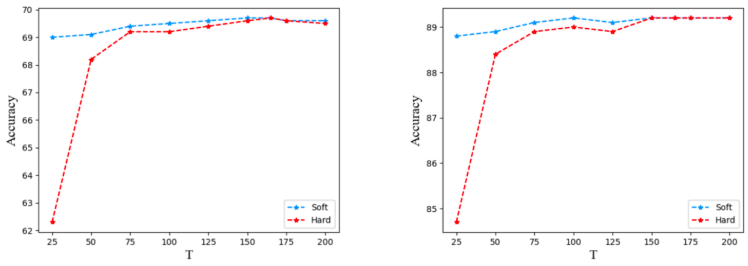
Figure 3: The distribution of full-precision parameters in ResNet-18.

3.1.2 Effect of Layer-wise Quantization

As depicted in Fig. 3, the magnitudes of parameters vary significantly from layer to layer. It makes more sense to use different quantization functions (different α, β, and b_i) for activations and weights in different layers.

3.1.3 Effect of Temperature T

In Eq. 1, the temperature T controls the gap between the soft and hard quantization functions, and this is reflected in the model performance, as shown in Fig. 4. As T increases, the gap between training accuracy and testing accuracy becomes smaller and smaller.



(a) Top-1 (b) Top-5

Figure 4: The gap between training and testing model w.r.t. training epoch for ResNet-18 {-4, +4}.

3.1.4 Training from Pre-trained Model

When training the model from scratch, the temperature may become too high before the network is well-converged, and the saturated neurons will slow down the training process. Fine-tuning with a pre-trained model has better performance than training the model from scratch.

3.1.5 Convergence of Temperature T

The heating speed is also a tunable hyper-parameter, which controls the speed of convergence from the soft quantization function to the hard quantization function. Since this paper considers the difference between the soft quantization function during training and the hard quantization function during inference as the main source of quantization error (similar to the gradient mismatch problem in approximation-based methods), a larger heating speed can accelerate the convergence of the quantization function, as shown in Fig. 5. Thus, the effect of heating speed is similar to that of the learning rate.

4 Analysis

In summary, this paper addresses the limitations of traditional optimization-based quantization, which is only applicable to weights and incurs high training costs, by reinterpreting quantization operations using differentiable non-linear activation functions. It also introduces a temperature parameter T and applies a relaxation method during training to gradually adjust the quantization function, thus mitigating the gradient mismatch issue in traditional approximation-based quantization methods.

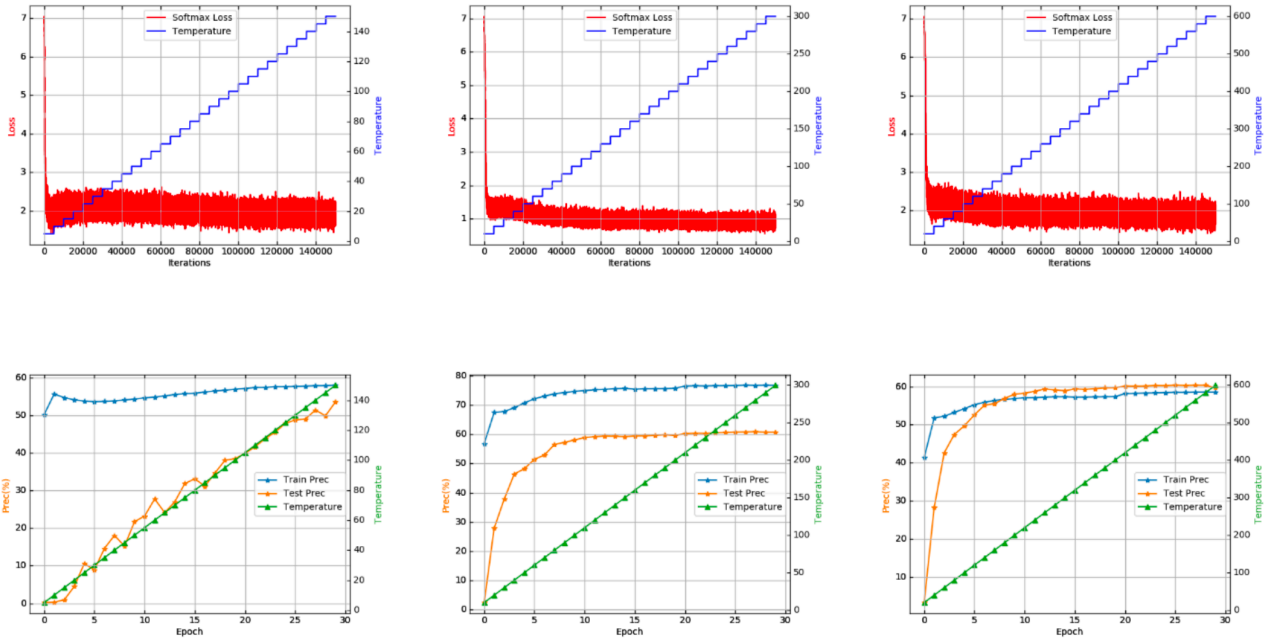


Figure 5: The training loss and training/validation accuracy for AlexNet quantization with $T = 5/10/20 * \text{epoch}$ (left to right).

Based on the ablation study in this work, I have the following observations:

First, the effects of training from scratch versus fine-tuning from a pre-trained model on the performance were evaluated. It was found that if the temperature rises too quickly, the parameters trained from scratch will be failed to converge. Thus, the fine-tuning from a pre-trained model is superior. However, adopting a lower heating speed may potentially avoid this situation.

In addition, when investigating the effect of temperature on model performance, it was observed that as the temperature increased during the training process, the testing accuracy and training accuracy gradually approached and converged. However, this result shown in Fig. 4 could also be contributed by parameters other than the quantization parameters, and the experiments did not exclude this situation. Supplementing the training process with fixed temperature could provide a more direct insight into the impact of temperature on model performance.

5 Future Work

This paper proposes a novel and effective method for quantization networks, but there are still some aspects that can be improved and explored further:

1. Quantization Function Form Selection The paper employs a quantization function composed of multiple linear combinations of sigmoid functions without thoroughly explaining why this form is chosen. Are there alternative quantization function forms, and how do their advantages and disadvantages compare?

2. Generalizability to Other Tasks and Models While the paper's experiments mainly focus on image classification and object detection tasks, the method's applicability to different network architectures (e.g., RNNs, transformers) and tasks (e.g., speech recognition) remains unverified.

3. Compatibility with Mixed-Precision Quantization This paper adopts layer-wise quantization, meaning that quantization functions for different layers can be independently defined. As long as different sets of quantized integer outputs $\mathcal{Y}$ are used, this approach can be combined with mixed-precision quantization methods.

4. Configuration of Quantization Biases The paper employs K-means clustering to obtain quantization biases in order to fit the Gaussian distribution of parameters in the model. However, K-means requires more computational resources compared to linear or logarithmic methods and is difficult to deploy on general computation hardware (e.g., GPU and CPU). It may be beneficial to consider using fine-grained sub-channel uniform quantization instead. This approach divides each channel into multiple groups and utilizes different quantization parameters for each group, thereby balancing computational costs and accuracy.

References

[1] Benoit Jacob et al. *Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference*. arXiv:1712.05877 [cs, stat]. Dec. 2017. DOI: 10.48550/arXiv.1712.05877. URL: http://arxiv.org/abs/1712.05877 (visited on 10/23/2023).

[2] Jiwei Yang et al. *Quantization Networks*. arXiv:1911.09464 [cs, stat]. Nov. 2019. DOI: 10.48550/arXiv.1911.09464. URL: http://arxiv.org/abs/1911.09464 (visited on 10/23/2023).