Distance-aware Quantization
Supplement

Dohyung Kim Junghyup Lee Bumsub Ham*  
School of Electrical and Electronic Engineering, Yonsei University
https://cvlab.yonsei.ac.kr/projects/DAQ

In this supplementary material, we present a detailed derivation of an output in Eq. (9) of our main paper, and an overall quantization process of DAQ in Sec. 1 and 2, respectively. We also provide more analysis on DAQ including the design and experimental results on the super-resolution (SR) task in Sec. 3 and 4, respectively.

1. Derivation of output in Eq. (9)

The output in Eq. (9) is obtained by plugging the adaptive temperature $\beta^*$ into the soft assignment function $\phi$ in Eq. (5) of the main paper. The soft assignment is defined as a weighted average using two nearest quantized values, $q_f$ and $q_c$, and corresponding distance probabilities $m_x$ as follows (See Eqs. (5-6) in the main paper):

$$\phi(x; \beta = \beta^*) = m_x(q_f; \beta = \beta^*)q_f + m_x(q_c; \beta = \beta^*)q_c = q_f \frac{\exp(\beta^* s_x(q_f)) + \exp(\beta^* s_x(q_c))}{\exp(\beta^* s_x(q_f)) + \exp(\beta^* s_x(q_c))} = q_f \frac{\exp(\beta^* (s_x(q_f) - s_x(q_c))) + q_c}{\exp(\beta^* (s_x(q_f) - s_x(q_c))) + 1}.$$  

We define the adaptive temperature $\beta^*$ as follows:

$$\beta^* = \frac{\gamma}{s_x(q_f) - s_x(q_c)},$$  

which can be expressed as:

$$\beta^* = \begin{cases} \frac{\gamma}{s_x(q_f) - s_x(q_c)}, & x \leq q_t \\ \frac{\gamma}{s_x(q_f) - s_x(q_c)}, & x > q_t \end{cases}.$$  

(3)

since the weighted score of $q_f$ is larger than that of $q_c$, i.e., $s_x(q_f) > s_x(q_c)$, when the normalized input $x$ is less than the transition point $q_t$, defined as $(q_f + q_c)/2$, and vice versa. Plugging the adaptive temperature $\beta^*$ in Eq. (3) into the soft assignment $\phi$ in Eq. (1), we obtain the following results:

$$\phi(x; \beta = \beta^*) = \begin{cases} q_f \exp(\gamma) + q_c, & x \leq q_t \\ q_f \exp(-\gamma) + q_c, & x > q_t \end{cases} = \begin{cases} q_f \exp(\gamma) + q_c, & x \leq q_t \\ q_c \exp(\gamma) + q_f, & x > q_t \end{cases}$$  

(4)

where $\lambda = 1/(\exp(\gamma) + 1)$ as stated in Sec. 3.3 of the main paper. We then reformulate this equation as follows:

$$\phi(x; \beta = \beta^*) = \begin{cases} (q_f \exp(\gamma) + q_f - q_f + q_c)\lambda, & x \leq q_t \\ (q_c \exp(\gamma) + q_c - q_c + q_f)\lambda, & x > q_t \end{cases} = \begin{cases} (q_f(1/\lambda) + (q_c - q_f))\lambda, & x \leq q_t \\ (q_c(1/\lambda) - (q_c - q_f))\lambda, & x > q_t \end{cases}$$  

(5)

Note that $q_c - q_f = 1$, since $q_f$ and $q_c$ are obtained by applying floor and ceil functions, respectively, to the normalized input $x$. Using this fact,

$$\phi(x; \beta = \beta^*) = \begin{cases} q_f + (q_c - q_f)\lambda, & x \leq q_t \\ q_c - (q_c - q_f)\lambda, & x > q_t \end{cases}$$  

(6)

which corresponds to Eq. (9) in the main paper.

2. Overall Process of DAQ

We show in Algorithm. 1 an overall quantization process of DAQ.

---

*Corresponding author
Algorithm 1 The distance-aware quantizer (DAQ)

1: Input:
\( \hat{x} \): an element of either full-precision weights or activations.

2: Output:
\( Q(\hat{x}) \): a quantized value with a b-bit precision.

3: Parameters:
\( \gamma \): a positive constant, \( l \): a lower bound, \( u \): an upper bound.

4: Training:
5: Clip and normalize the input \( \hat{x} \):
\( x = (2^b - 1)(\text{clip}(\hat{x}, \min = l, \max = u) - l)/(u - l). \)
6: Find the two nearest quantized values, \( q_f \) and \( q_c \):
\( q_f = \text{floor}(x), q_c = \text{ceil}(x). \)
7: Compute distance scores and adjust the temperature \( \beta^* \):
\( d_s(q_i) = \exp(-|x - q_i|), \ i \in \{f,c\} \)
\( \beta^* = \gamma/|s_s(q_f) - s_s(q_c)|, \)
where \( s_s(q) \) is a weighted score, defined as \( k_{s}(q)d_{s}(q). \)
8: Compute distance probabilities \( m_x \) for quantized values, \( q_f \) and \( q_c \), with an adaptive temperature \( \beta^* \):
\( m_x(q_i; \beta = \beta^*) = \exp(\beta^*s_s(q_i)) / \sum_{i \in \{f,c\}} \exp(\beta^*s_s(q_i)), \ i \in \{f,c\} \)
9: Compute the assignment using the continuous assignment function \( \phi(x; \beta = \beta^*) \) with the corresponding temperature \( \beta^* \):
\( \phi(x; \beta = \beta^*) = \sum_{i \in \{f,c\}} m_x(q_i; \beta = \beta^*)q_i. \)
10: Rescale the assignment \( \phi(x; \beta = \beta^*) \) using the following function:
\( f(y) = (y - q_t)/(1 - 2\lambda) + q_t \),
where \( \lambda = 1/(e^\gamma + 1) \) and \( q_t = (q_f + q_c)/2. \)
11: Set \( Q(\hat{x}) = f(\phi(x; \beta = \beta^*)). \)
12: Inference:
13: Clip and normalize the input \( \hat{x} \):
\( x = (2^b - 1)(\text{clip}(\hat{x}, \min = l, \max = u) - l)/(u - l). \)
14: Find the nearest quantized value using a rounding function:
\( Q(\hat{x}) = \text{round}(x). \)

3. Design of DAQ

We design functions in DAQ for distance scores and kernels satisfying the following conditions. The distance score \( d_x \) increases as the input \( x \) approaches the quantized value \( q \), and the kernel function \( k_x \) retains values nearby its center. Under these conditions, we can use other functions for distance scores and kernels. To test this, we apply variants of them to our quantizer, where the parameters of \( \gamma \) and standard deviation of the kernel are fixed to the values chosen from the original functions. Specifically, we quantize ResNet-20 in an 1/1-bit setting, and test the model on the test split of CIFAR-10. 1) For the distance score, we use two variants, \( d_x^1 \) and \( d_x^2 \), as follows:
\[
\begin{align*}
    d_x^1(q) &= 1 - |x - q|, \quad d_x^2(q) = \frac{1}{1 + |x - q|}. \quad (7)
\end{align*}
\]
With these functions but using the kernel as our original Gaussian one, our quantizer achieves top-1 accuracies of 84.4 and 84.2, respectively. 2) For the kernel, we replace the Gaussian kernel with the Laplacian one while fixing the distance score as our original one in Eq. (4) of the main paper, achieving the top-1 accuracy of 83.7. These results suggest that switching the distance or kernel functions provides comparable results with the original one (85.8), even without tuning hyperparameters.

4. Experiments on the SR Task

We apply our method to the SR task on the Set5 [1] dataset. We quantize weights and activations for FSRCNN [3] in 3/3, 4/4, and 8/8-bit settings, achieving the average peak signal to noise ratio (PSNR) of 35.99dB, 36.56dB, and 37.18dB, respectively, with the scale factor of 2, where the full-precision model shows the PSNR of 37.05dB. We can see the experimental results from the SR task show trends similar to those for the classification task in that the high-bit quantized model (i.e., 8/8-bit setting) provides a better result than the full-precision one. This suggests that DAQ could be effective in the regression task as well. Note that most quantization methods [2, 4, 5, 6, 7, 8] show their effectiveness only to the classification task, making it difficult to compare them directly with ours on the SR task.
References


