Distance-aware Quantization

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

Abstract

We address the problem of network quantization, that is, reducing bit-widths of weights and/or activations to lighten network architectures. Quantization methods use a rounding function to map full-precision values to the nearest quantized ones, but this operation is not differentiable. There are mainly two approaches to training quantized networks with gradient-based optimizers. First, a straight-through estimator (STE) replaces the zero derivative of the rounding with that of an identity function, which causes a gradient mismatch problem. Second, soft quantizers approximate the rounding with continuous functions at training time, and exploit the rounding for quantization at test time. This alleviates the gradient mismatch, but causes a quantizer gap problem. We alleviate both problems in a unified framework. To this end, we introduce a novel quantizer, dubbed a distance-aware quantizer (DAQ), that mainly consists of a distance-aware soft rounding (DASR) and a temperature controller. To alleviate the gradient mismatch problem, DASR approximates the discrete rounding with the kernel soft argmax, which is based on our insight that the quantization can be formulated as a distance-based assignment problem between full-precision values and quantized ones. The controller adjusts the temperature parameter in DASR adaptively according to the input, addressing the quantizer gap problem. Experimental results on standard benchmarks show that DAQ outperforms the state of the art significantly for various bit-widths without bells and whistles.

1. Introduction

Convolutional neural networks (CNNs) have made significant progress in the field of computer vision, such as image recognition [27, 48], object detection [2, 43], and semantic segmentation [7, 34]. Deeper [15, 46] and wider [45] CNNs, however, require lots of parameters and FLOPs, making it difficult to deploy modern network architectures on edge devices (*e.g.*, mobile phones, televisions, or drones). Recent works focus on compressing networks to lighten the network architectures. Pruning [14] and distillation [16] are representative techniques for network compression. The pruning removes redundant weights in a network, and the distillation encourages a compact network to have features similar to the ones obtained from a large network. The networks compressed by these techniques still exploit floating-point computations, indicating that they are not suitable for edge devices favoring fixed-point operations for power efficiency. Network quantization [42] is an alternative approach that converts full-precision weights and/or activations into low-precision ones, enabling a fixed-point inference, while reducing memory and computational cost.

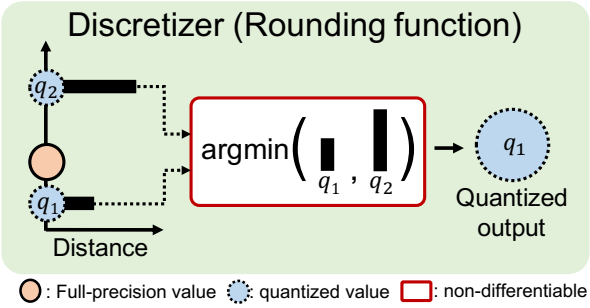

Figure 1: The discretizer takes a full-precision input, and then assigns it to the nearest quantized value, *e.g.*, q_1 in this example. We interpret the assignment process of a discretizer as follows: It first computes the distances between the full-precision input and quantized values, q_1 and q_2, and then applies an argmin operator over the distances to choose the quantized value. Since this operator is non-differentiable, the quantized network cannot be trained end-to-end with gradient-based optimizers. (Best viewed in color.)

Quantization methods typically use a staircase function as a quantizer, where it normalizes a full-precision value within a quantization interval, and assigns the normalized one to the nearest quantized value using a discretizer (*i.e.*, a rounding function) [11, 12, 22]. Since the derivative of the rounding is zero at almost everywhere, gradient-based optimizers could not be used to train quantized networks. To address this, the straight-through estimator (STE) [3] replaces the derivative of the rounding with that of identity or hard tanh functions for backward propagation. This, however, causes a gradient mismatch between forward and backward passes at training time, making the training process noisy and degrading

*Corresponding author