Journal of Computational Mathematics
Vol.37, No.3, 2019, 349–359.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1803-m2017-0301

QUANTIZATION AND TRAINING OF LOW BIT-WIDTH CONVOLUTIONAL NEURAL NETWORKS FOR OBJECT DETECTION*

Penghang Yin
Department of Mathematics, University of California, Los Angeles, CA 90095, USA
Email: yph@ucla.edu
Shuai Zhang,[1)] Yingyong Qi and Jack Xin
Department of Mathematics, University of California, Irvine, CA 92697, USA
Email: szhang3@uci.edu, yqi@uci.edu, jack.xin@uci.edu

Abstract

We present LBW-Net, an efficient optimization based method for quantization and training of the low bit-width convolutional neural networks (CNNs). Specifically, we quantize the weights to zero or powers of 2 by minimizing the Euclidean distance between full-precision weights and quantized weights during backpropagation (weight learning). We characterize the combinatorial nature of the low bit-width quantization problem. For 2-bit (ternary) CNNs, the quantization of N weights can be done by an exact formula in $O(N \log N)$ complexity. When the bit-width is 3 and above, we further propose a semi-analytical thresholding scheme with a single free parameter for quantization that is computationally inexpensive. The free parameter is further determined by network retraining and object detection tests. The LBW-Net has several desirable advantages over full-precision CNNs, including considerable memory savings, energy efficiency, and faster deployment. Our experiments on PASCAL VOC dataset show that compared with its 32-bit floating-point counterpart, the performance of the 6-bit LBW-Net is nearly lossless in the object detection tasks, and can even do better in real world visual scenes, while empirically enjoying more than 4× faster deployment.

Mathematics subject classification: 90C26, 90C10, 90C90.
Key words: Quantization, Low bit width deep neural networks, Exact and approximate analytical formulas, Network training, Object detection.

1. Introduction

Deep convolutional neural networks (CNNs) have demonstrated superior performance in various computer vision tasks [3,13–16,18,22–24]. However deep CNNs typically have hundreds of millions of trainable parameters which easily take up hundreds of megabytes of memory, and billions of FLOPs for a single inference. This poses a significant challenge for the deployment of deep CNNs on small devices with limited memory storage and computing power such as mobile phones. To address this issue, recent efforts have been made to compress the model size [7,9] and train neural networks with heavily quantized weights, activations, and gradients [1, 2, 6, 7, 9, 17, 20, 21, 26–28], which demand less storage and fewer FLOPs for deployment. These models include BinaryConnect [1], BinaryNet [2], XNOR-Net [21], TWN [17], TTQ [28],

* Received December 24, 2017 / Accepted March 13, 2018 /
Published online August 16, 2018 /
[1)] P. Yin and S. Zhang contributed equally to this work.

DoReFa-Net [27] and QNN [9], to name a few. In particular, binary (1-bit) and ternary (2-bit) weight models not only enable high model compression rate, but also eliminate the need of most floating-point multiplications during forward and backward propagations, which shows promise to resolve the problem. Compared with binary models, ternary weight networks such as TWN strike a better balance between model size and accuracy. It has been shown that ternary weight CNNs [17] can achieve nearly lossless accuracy on MNIST [16] and CIFAR-10 [12] benchmark datasets. Yet with fully ternarized weights, there is still noticeable drop in performance on larger datasets like ImageNet [4], which suggests the necessity of relatively wider bit-width models with stronger performance for challenging tasks.

An incremental network quantization strategy (INQ) is proposed in [26] for converting pre-trained full-precision CNNs into low bit-width versions whose weights are either zero or powers of two. A b bit-width model can have $2^{b-1}+1$ distinct candidate values, in which 2 bits are used for representing the zero and the signs, while the remaining $b-2$ bits for the powers. More precisely, the parameters are constrained to $2^s \times \{0, \pm 2^{1-2^{b-2}}, \pm 2^{2-2^{b-2}}, \ldots, \pm 1\}$ associated with a layerwise scaling factor 2^s, s an integer depending only on the weight maximum in the layer. At inference time, the original floating-point multiplication operations can be replaced by faster and cheaper binary bit shifting. The quantization scheme of [26] is however heuristic.

In this paper, we present the exact solution of the general b-bit approximation problem of a real weight vector W^f in the least squares sense. If $b = 2$ and the dimension of W^f is N, the computational complexity of the 2 bit solution is $O(N \log N)$. At $b \geq 3$, the combinatorial nature of the solution renders direct computation too expensive for large scale tasks. We shall develop a semi-analytical quantization scheme involving a single adjustable parameter μ to set up the quantization levels. The exponent s in the scaling factor can be calculated analytically from μ and the numbers of the downward sorted weight components between quantization levels. If the weight vector comes from a Gaussian ensemble, the parameter μ can be estimated analytically. However, we found that the weight vectors in CNNs (in particular ResNet) are strongly non-Gaussian. In this paper, μ is determined based on the object detection performance after retraining the network. This seems to be a natural choice in general as quantization is often part of a larger computer vision problem as is here. Therefore, the optimal parameter μ should not be decided by approximation (the least squares problem) errors alone. Indeed, we found that at $b \geq 4$, $\mu = \frac{3}{4}\|W^f\|_\infty$ gives the best detection performance, which suggests that a percentage of the large weights plays a key role in representing the image features and should be encoded during quantization.

Network retraining is necessary after quantization as a way for the system to adjust and absorb the resulting errors. Besides warm start, INQ [24] requires a careful layerwise partitioning and grouping of the weights which are then quantized and re-trained incrementally group by group rather than having all weights updated at once. Due to both classification and detection networks involved in this work, we opted for a simpler retraining method, a variant of the projected stochastic gradient descent (SGD) method (see [1, 17, 21] and references therein). As a result, our LBW-Net can be trained either from scratch or a partial warm start. During each iteration, besides forward and backward propagations, only an additional low cost thresholding (projection) step is needed to quantize the full-precision parameters to zero or powers of two. We train LBW-Net with randomly initialized weights in the detection network (R-FCN [3]), and pre-trained weights in ResNet [8]. We conduct object detection experiments on PASCAL VOC data sets [5] as in [3, 22]. We found that at bit-width $b = 6$, the accuracies of the quantized networks are well within 1% of those of their 32-bit floating-point counterparts on both