IQNN: Training Quantized Neural Networks with Iterative Optimizations

Shuchang Zhou[1,2,3(✉)], He Wen[3], Taihong Xiao[3], and Xinyu Zhou[3]

[1] University of Chinese Academy of Sciences, Beijing 100049, China
shuchang.zhou@gmail.com
[2] State Key Laboratory of Computer Architecture,
Institute of Computing Technology, Chinese Academy of Sciences,
Beijing 100190, China
[3] Megvii Inc., Beijing 100190, China
{wenhe,xiaotaihong,zxy}@megvii.com

Abstract. Quantized Neural Networks (QNNs) use low bitwidth numbers for representing parameters and intermediate results. The lowering of bitwidths saves storage space and allows for exploiting bitwise operations to speed up computations. However, QNNs often have lower prediction accuracies than their floating point counterparts, due to the extra quantization errors. In this paper, we propose a quantization algorithm that iteratively solves for the optimal scaling factor during every forward pass, which significantly reduces quantization errors. Moreover, we propose a novel initialization method for the iterative quantization, which speeds up convergence and further reduces quantization errors. Overall, our method improves prediction accuracies of QNNs at no extra costs for the inference. Experiments confirm the efficacy of our method in the quantization of AlexNet, GoogLeNet and ResNet. In particular, we are able to train a GoogLeNet having 4-bit weights and activations to reach 11.4% in top-5 single-crop error on ImageNet dataset, outperforming state-of-the-art QNNs. The code will be available online.

Keywords: Quantized Neural Network · Uniform quantization · Iterative quantization · Alternating least squares · Bitwise operation

1 Introduction

Deep Neural Networks have found wide-spread applications due to their ability to model nonlinear relationships in massive amount of data and robustness to real world noises. However, the modeling capacities of DNNs are roughly proportional to their computational complexities. For example, DNNs that are widely used in computer vision applications, like AlexNet [9], GoogLeNet [15] and ResNet [5], require billions of multiply-and-add operations for an input image of scale of 224. Such high resource requirements impede applications of DNNs to embedded devices and interactive scenarios.

Quantized Neural Networks (QNNs) [7,8,12] have been proposed as less resource-intensive variants of DNNs. By quantizing some of weights, activations

A. Lintas et al. (Eds.): ICANN 2017, Part II, LNCS 10614, pp. 688–695, 2017.
https://doi.org/10.1007/978-3-319-68612-7_78

and gradients to low bitwidth numbers, QNNs typically require less memory, storage space and computation, and have found applications in Image Classification, Segmentation, etc [16]. However, as the quantization introduces approximation errors, QNNs are in general worse than their floating point counterparts in terms of prediction accuracies.

In this paper, we focus on reducing the quantization errors of parameters of QNNs to improve prediction accuracies. We first propose an optimization formulation for the multi-bit quantization of weight parameters. As no closed-form solutions exist for the optimization, we construct Iterative Quantization, an Alternating Least Squares (ALS) [11] algorithm, to find the optimal scaling factors for the quantization. The iterative algorithm is designed to use only simple matrix operations, and can be readily integrated into the training process. Because the iterative optimization is only performed during training, there will be no overhead added to the inference. Moreover, we propose to initialize the optimization with values based on statistics of weights, to further reduce quantization errors and number of iterations required for the optimization.

Numerical experiments on the quantization of weights of neural networks confirm the efficacy of our method in reducing quantization errors. We also train QNNs by Iterative Quantization from scratch on the large-scale ImageNet [2] dataset, and outperform the state-of-the-art QNNs in terms of prediction accuracies.

2 Quantized Neural Networks

We first introduce some notations. We define a utility function quant_k that converts floating point numbers in the closed interval $[-\frac{1}{2}, \frac{1}{2}]$ to fixed point numbers as follows:

$$\text{quant}_k(\mathbf{W}) \stackrel{\text{def}}{=\!=} \frac{1}{2^k-1}\,\text{round}((2^k-1)(\mathbf{W}+\frac{1}{2}))-\frac{1}{2}, \quad -\frac{1}{2} \le w_{i,j} \le \frac{1}{2}\ \forall i,j, \tag{1}$$

where $w_{i,j}$ are entries of matrix $\mathbf{W}$, and the outputs of quant_k are among $-\frac{1}{2}, -\frac{1}{2}+\frac{1}{2^k-1}, -\frac{1}{2}+\frac{2}{2^k-1}, \cdots, \frac{1}{2}$.

When quantizing parameters of a Neural Network, we would need first map the parameters $\mathbf{W}$ to the closed interval $[-\frac{1}{2}, \frac{1}{2}]$ before applying quant_k:

Definition 1 (k-bit Uniform Quantization [7,17]).

$$\text{uniform-quant}_k(\mathbf{W}) \stackrel{\text{def}}{=\!=} 2\max(|\mathbf{W}|)\,\text{quant}_k\left(\frac{\mathbf{W}}{2\max(|\mathbf{W}|)}\right),$$

where the subscript k stands for k-bit quantization, and $|\mathbf{W}|$ is a matrix with values being the absolute values of corresponding entries in $\mathbf{W}$.

As $-\max(|\mathbf{W}|) \le w_{i,j} \le \max(|\mathbf{W}|)$, we have $-\frac{1}{2} \le \frac{w_{i,j}}{2\max(|\mathbf{W}|)} \le \frac{1}{2}$. We can then apply quant_k to get the fixed point values. Finally we restore the value range back to $[-\max(|\mathbf{W}|), \max(|\mathbf{W}|)]$ by multiplying $2\max(|\mathbf{W}|)$.

As its outputs are discrete values, any quantization function will have zero gradients, which invalidates the Back Propagation algorithm. To circumvent this problem, we need convert *quant* to a Straight Through Estimator [6], by substituting the gradients with respect to the quantized value for the gradients of the original value.

3 Iterative Quantization of Neural Network

QNNs often incur significant degradations in prediction accuracies when bitwidths are below 4-bit [7,12,17]. We note that for the uniform quantization defined in Definition 1, the scaling factors are determined from extremal values in one shot, which may be suboptimal. In this section we propose an algorithm that iteratively optimizes the scaling factors, which generalizes the uniform quantization method and reduces quantization errors.

3.1 Quantization as Optimization

To reduce quantization errors measured in Frobenius norm, we investigate the following optimization formulation for k-bit quantization:

$$\min_{\mathbf{\Lambda},\mathbf{Q}} \|\mathbf{\Lambda Q} - \mathbf{W}\|_F \tag{2}$$

where $\mathbf{W}$ contains weights of a fully-connected (convolutional) layer of a neural network, $\mathbf{\Lambda}$ is a diagonal matrix containing floating point scaling factors[1], and $\mathbf{Q}$ contains fixed-point values in the closed interval $[-\frac{1}{2}, \frac{1}{2}]$. Determining the scaling factor $\mathbf{\Lambda}$ is important as it affects the value of the fixed point part $\mathbf{Q}$. The product $\mathbf{\Lambda Q}$ will be used to replace $\mathbf{W}$ during the inference.

3.2 Solution by Iterative Algorithm

The objective function of Formula 2 is non-convex and lacks a closed-form solution except for the special case of 1-bit [12]. Nevertheless, it can be solved by the Alternating Least Squares algorithm, detailed in Algorithm 1.

It can be observed that only simple matrix operations are used in Algorithm 1. Hence the iterative optimization can be readily integrated into the computation graph of a QNN as a unrolled loop. Alternatively, as $(\mathbf{\Lambda}_i, \mathbf{Q}_i)$ are iteratively updated, the iterations can be implemented as a Recurrent Neural Network layer with $(\mathbf{\Lambda}_i, \mathbf{Q}_i)$ as state variables, which reduces memory footprint during training.

The uniform quantization method from Definition 1 can be formulated as a special case of Algorithm 1, by setting all entries of $\mathbf{\Lambda}_0$ to be $2\max(|\mathbf{W}|)$ and having the number of iterations $N = 1$.

[1] Floating point multiplication with $\mathbf{\Lambda}$ during inference can be avoided [17].

Algorithm 1. Iterative quantization for matrix $\mathbf{W} \in \mathbb{R}^{I \times J}$

Require: Initialization values $\mathbf{\Lambda}_0$
Ensure : Quantized weights $\mathbf{\Lambda}_N \mathbf{Q}_N \approx \mathbf{W}$
for $t = 1 \rightarrow N$ **do**
  $\mathbf{Q}_t \leftarrow \text{quant}_k(\text{clip}((\mathbf{\Lambda}_{t-1})^{-1}\mathbf{W}, -\frac{1}{2}, \frac{1}{2}))$;
  // The clipping function $\text{clip}(x, l, h) = \max(\min(x, h), l)$ is used to limit the value range to $[l, h]$.
  for $i = 1 \rightarrow I$ **do**
    $(\mathbf{\Lambda}_t)_i \leftarrow \frac{<(\mathbf{W})_i, (\mathbf{Q}_t)_i>}{\epsilon + <(\mathbf{Q}_t)_i, (\mathbf{Q}_t)_i>}$;
    // $< \cdot, \cdot >$ computes the inner product.
    // $(\mathbf{W})_i, (\mathbf{Q}_t)_i$ are the i-th row of $\mathbf{W}$ and $\mathbf{Q}_t$ respectively. $(\mathbf{\Lambda}_t)_i$ is the i-th diagonal entry of $\mathbf{\Lambda}_t$.
    // ϵ is a small constant thas is used to avoid division by zero.
  end
end

3.3 Distribution of Weights and Initialization

When weights follow a well known distribution, the scaling factors $\mathbf{\Lambda}$ may be determined from theoretical results of optimal uniform quantizers [14], which we list in Table 1. As the theoretical optima are fixed points of Iterative Quantization, when they are used as initialization values, the convergence of the iterative algorithm can be accelerated.

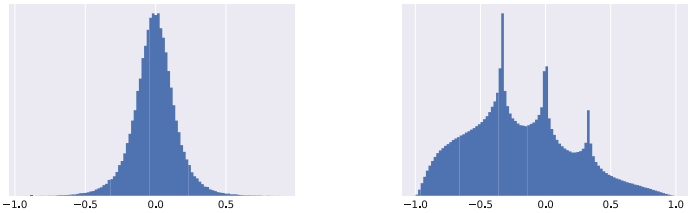
Table 1. Comparison of maximal quantized value of optimal uniform quantizers for uniform (over $[-1, 1]$) and standard normal distributions [14].

Statistics	Uniform	Normal
Maximum after optimal 2-bit quantization	0.5	0.798
Maximum after optimal 4-bit quantization	0.938	2.513
Mean of absolute value	0.5	0.798

However, the distribution of weights of Neural Networks may be quite complex. Two illustrative examples are given in Fig. 1, where the second one has many peaks. Hence it is not in general possible to determine optimal initialization values $\mathbf{\Lambda}_0$. Nevertheless, we observe that the ratio between mean of absolute value and maximal quantized value is quite stable across different distributions and different bitwidths. For example, when performing 2-bit quantization, the ratios are 1 for both Uniform and Normal distributions.

We propose to initialize $\mathbf{\Lambda}_0$ with mean of absolute values scaled by an coefficient γ when performing k-bit quantization as follows:

$$(\mathbf{\Lambda}_0)_i = \gamma \, \text{mean}((|\mathbf{W}|)_i), \tag{3}$$



(a) (b)

Fig. 1. (a) Distribution of weights from a convolution layer in a GoogLeNet model. (b) Distribution of weights before quantization from the last fully-connected layer in a QNN version of AlexNet.

where $(|\mathbf{W}|)_i$ is the i-th row of $|\mathbf{W}|$ and $(\mathbf{\Lambda}_0)_i$ is the i-th diagonal entry of $\mathbf{\Lambda}_0$. For 2-bit case we set $\gamma_{opt} = 2 = 2 \times \frac{0.5/0.5+0.798/0.798}{2}$, and for 4-bit we set $\gamma_{opt} = 5.02 \approx 2 \times \frac{0.938/0.5+2.513/0.798}{2}$. The γ coefficient is 2 times the average of ratios for Uniform and Normal distributions, as the maximual quantized values are mapped to $\frac{1}{2}$ in Definition 1.

4 Experiments

In this section, we conduct experiments to compare the performance of Iterative Quantization with the non-iterative method (Definition 1). Experiments are performed on machines equipped with Intel Xeon CPUs and NVidia TitanX Graphics Processing Units.

4.1 Iterative Quantization of Weights of a Layer

We first experiment on the quantization of weights of the last fully-connected layer of AlexNet, and test the convergence of our algorithm. Results are listed in Fig. 2. It can be seen that the quantization errors decrease monotonically with more iterations, which is a property of ALS. However, the initialization values significantly impact the speed of convergence. In fact, initialization with the approximate optimum $\gamma_{opt} = 5.02$ significantly reduces the number of iterations required for convergence. On the other hand, quantization errors are still substantially reduced even when initializing with γ_{opt}, which justifies performing the iterative optimization during training.

We will set $\#iter = 8$ in remaining experiments unless noted, to strike a balance between the training speed and the prediction accuracy.

4.2 Iterative Quantization for Training Neural Networks

We also apply Iterative Quantization to train QNNs from scratch. We use ImageNet dataset that contains 1.2 M images for training and 50 K images for validation. While testing, images are first resized so that the shortest edge contains

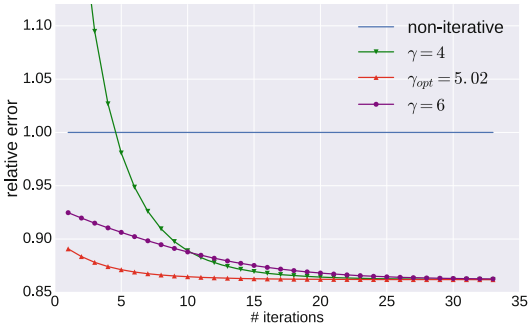


Fig. 2. Relative errors against number of iterations for the 4-bit quantization. Weights are from the last fully-connected layer of AlexNet. The errors are scaled so that the error of non-iterative method is 1.

256 pixels, then the center 224-by-224 crops will be used as inputs. Following the conventions, we report results in two measures: single-crop top-1 error rate and top-5 error rate over ILSVRC12 validation sets [13].

For all QNNs in this section, weights and activations of all convolutional and fully-connected layers have been quantized by specified bitwidths unless noted. The activations are quantized by the method of DoReFa-net [17].

Table 2. Comparison of classification errors of QNNs trained with different methods. FP stands for neural networks with floating point weights and activations. "FP weights + 2-bit activations" refers to models that have floating point weights and activations quantized to 2-bit numbers. Results in rows prefixed with "non-iterative" are produced from non-iterative uniform quantization.

Method	AlexNet		ResNet-18	
	Top-1 error	Top-5 error	Top-1 error	Top-5 error
FP	42.9%	20.6%	31.8%	12.5%
FP weights + 2-bit activations	43.5%	21.0%	38.9%	17.3%
Non-iterative 2-bit	45.3%	22.3%	42.3%	19.2%
Iterative 2-bit	**43.2%**	**20.8%**	**41.8%**	**19.0%**

Table 2 demonstrates the efficacy of Iterative Quantization for training QNNs, exhibited by the improved prediction accuracies. For AlexNet, the QNN trained with our method has almost the same top-5 error rate as the floating point one.

Table 3 compares the GoogLeNet quantized with our method against the state-of-the-art. The row marked with "QNN 4-bit" is from Hubara *et al.* [7]. To rule out factors like Image Augmentation, we also list the accuracies of their

Table 3. Comparison of classification errors of our method with the state-of-the-art for the quantization of GoogLeNet.

Method	Top-1 error	Top-5 error
FP [7]	28.4%	8.8%
Our FP	28.5%	10.1%
Ristretto [4] 8-bit	33.4%	-
QNN 4-bit [7]	33.5%	16.6%
Our 4-bit (#iter=4)	31.6%	11.9%
Our 4-bit (#iter=16)	**31.2%**	**11.4%**

floating point model. It can be seen that their FP model has better accuracies than our FP model. In contrast, our quantized model outperforms their quantized model. In particular, our method reduces the top-5 accuracy degradation, which is the difference in accuracy between a QNN and its floating point version, from 7.8 percentages to 1.3 percentages. In addition, with our initialization method, the top-5 error rate only slightly increases by 0.5 percentages if we reduce the number of iterations from 16 to 4.

5 Related Work

Our iterative quantization method is different from that of Gong *et al.* [3], because QNNs restrict the transformation of weights to scaling. Lin *et al.* [10] investigated optimal uniform quantization but did not integrate it into the training of DNNs, hence their method incurred severe accuracy degradations when bitwidths are below 6. Anwar *et al.* [1] investigated an iterative algorithm for quantization of pre-trained networks, which were later fine-tuned. However, such operations were only performed a few times during the whole training process. To the best of our knowledge, we are the first to integrate the iterative quantization into training of QNNs and perform experiments on a dataset of the scale of ImageNet.

6 Conclusion

In this paper, we propose the method of Iterative Quantization for training QNNs. We formulate the multi-bit quantization of weights of Neural Networks as an optimization problem, which is solved by an iterative algorithm to minimize quantization errors, during each forward pass of the training. Moreover, we propose a method to use statistics of weights as initial values, which further reduces quantization errors and the overhead added to training.

References

1. Anwar, S., Hwang, K., Sung, W.: Fixed point optimization of deep convolutional neural networks for object recognition. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19–24, 2015, pp. 1131–1135 (2015)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE (2009)
3. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2916–2929 (2013)
4. Gysel, P., Motamedi, M., Ghiasi, S.: Hardware-oriented approximation of convolutional neural networks. CoRR abs/1604.03168 (2016)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp. 770–778, June 2016
6. Hinton, G., Srivastava, N., Swersky, K.: Neural networks for machine learning. Coursera Video Lect. vol. 264 (2012)
7. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: training neural networks with low precision weights and activations. CoRR abs/1609.07061 (2016)
8. Kim, M., Smaragdis, P.: Bitwise neural networks. CoRR abs/1601.06071 (2016)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105, December 2012
10. Lin, D.D., Talathi, S.S., Annapureddy, V.S.: Fixed point quantization of deep convolutional networks. In: International Conference on Machine Learning (ICML2016) (2015)
11. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theor. **28**(2), 129–136 (1982)
12. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-Net: imagenet classification using binary convolutional neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 525–542. Springer, Cham (2016). doi:10.1007/978-3-319-46493-0_32
13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015)
14. Shi, Y.Q., Sun, H.: Image and Video Compression for Multimedia Engineering: Fundamentals, Algorithms, and Standards. CRC Press, Boca Raton (1999)
15. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015, pp. 1–9, June 2015
16. Wen, H., Zhou, S., Liang, Z., Zhang, Y., Feng, D., Zhou, X., Yao, C.: Training bit fully convolutional network for fast semantic segmentation. CoRR abs/1612.00212 (2016)
17. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: DoReFa-Net: training low bitwidth convolutional neural networks with low bitwidth gradients. CoRR abs/1606.06160 (2016)