Differentiable Fine-grained Quantization for Deep Neural Network Compression

Hsin-Pai Cheng*
ECE Department
Duke University
Durham, NC 27708
hc218@duke.edu

Yuanjun Huang*
University of Science
and Technology of China
Anhui, China
yjhuang@mail.ustc.edu.cn

Xuyang Guo*
Tsinhua University
Beijing, China
guoxuyang1997@gmail.com

Feng Yan
Computer Science
and Engineering
University of Nevada, Reno
Reno, NV 89557
fyan@unr.edu

Yifei Huang
Nanjing University
Jiangsu, China
161240027@smail.nju.edu.cn

Wei Wen
ECE Department
Duke University
Durham, NC 27708
wei.wen@duke.edu

Hai Li
ECE Department
Duke University
Durham, NC 27708
hai.li@duke.edu

Yiran Chen
ECE Department
Duke University
Durham, NC 27708
yiran.chen@duke.edu

Abstract

Neural networks have shown great performance in cognitive tasks. When deploying network models on mobile devices with limited resources, weight quantization has been widely adopted. Binary quantization obtains the highest compression but usually results in big accuracy drop. In practice, 8-bit or 16-bit quantization is often used aiming at maintaining the same accuracy as the original 32-bit precision. We observe different quantization schemes have different accuracy impact on different layers. Thus judiciously selecting different precision for different layers/structures can potentially produce more efficient models compared to traditional quantization methods by striking a better balance between accuracy and compression rate. In this work, we propose a fine-grained quantization approach for deep neural network compression by relaxing the search space of quantization bitwidth from discrete to a continuous domain. The proposed approach applies gradient descend based optimization to generate a mixed-precision quantization scheme that outperforms the accuracy of traditional quantization methods under the same compression rate.

1 Introduction

State-of-the-art neural networks have demonstrated promising performance in tasks such as image classification and object detection [1][2][3][4]. These network models are designed for high accuracy with less consideration in the computational cost and inference delay. Thus deploying them on resource-constrained platform such as mobile phones is usually inefficient or even infeasible. Even

*Equal Contribution and Co-First Authors

Accepted by NIPS 2018 Workshop on Compact Deep Neural Network Representation with Industrial Applications, Montréal, Canada.

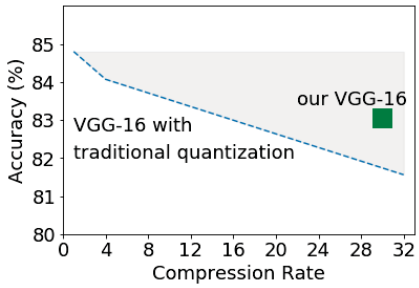

Figure 1: Accuracy VS compression rate. The dotted line is traditional quantization and the shaded area is our optimization goal. The box is our preliminary result.

the recently proposed MobileNet [5], which is customized for mobile platforms, has a relatively large 4.24M parameters. Extensive studies have been carried out in developing network models for resource constraint platforms. Quantization is one of the most popular approaches [6][7]. For example, Bai *et al.* recently proposed a proximal operators for training low-precision deep neural netwokrs [8]. Courbariaux *et al.* proposed a radical binary representation of the inputs, weights, and activations [6]. Rastegari *et al.* [7] theoretically analyzed the binary network and introduced a scaling scheme for their XNOR-Net—a network based on [6] with higher accuracy. Despite the significant improvement on inference accuracy, none of the above networks is able to achieve comparable accuracy as the full-precision counterparts.

We observe that different layers may have different accuracy sensitivity of quantization, thus a fine-grained quantization for each layer has the penitential to preserve accuracy under the same compression rate (defined as the ratio of original model size and compressed model size) compared to traditional course-grained quantization that uses the same quantization for the entire model. The dotted line in Figure 1 shows the trade-off between accuracy and compression rate in traditional quantization for VGG-16. Our goal is to push the trade-off between accuracy and compression rate into the shaded region of Figure 1 to achieve better compression efficiency, i.e., higher accuracy under the same compression rate. To achieve this, we propose a fine-grained quantization approach that relaxes the search space of quantization bitwidth from discrete to continuous domain and applies gradient descend optimization to generate best quantization scheme for each layer, i.e., applies lower bit for less quantization sensitive layers while preserving high bit precision for quantization sensitive layers. Our experimental results show that the proposed approach outperforms the accuracy of traditional quantization methods under the same compression rate.

2 Proposed Approach

In this section, we propose a methodology to judiciously determine the best quantization scheme for each layer based on each layer's accuracy sensitivity of quantization. For easy description, we only use two-level quantization: binary and 8-bit quantization as an example to illustrate our approach and conduct a preliminary evaluation. It is straightforward to extend to more quantization levels. We relax the discrete variables to a continuous domain as the finer granularity of which can provide more accurate indication in quantization searching. We adopt gradient descent based searching algorithm as it is fast and can be easily deployed in different machine learning frameworks.

We use *Softmax* function to relax the search space from discrete to continuous. We denote the output of layer i with continuous relaxation as q_i. For example, binary and 8-bit quantization represented as q_{i_0} and q_{i_1}. *Softmax*(q_{i_0}, q_{i_1}) can be translated as the probability of binary and 8-bit quantization, respectively. Thus q_i can be computed as

$$q_i = \frac{\sum_{j=0}^{1} exp(\alpha_{i_j}) \mathcal{B}(q_{i_j})}{\sum_{j=0}^{1} exp(\alpha_{i_j})}, \tag{1}$$

where $\mathcal{B}$ is the batch normalization operation. The output q_i is used as the input of the following layer. The search space for a network with k layer is $\alpha = \{\alpha_0, \alpha_1, ..., \alpha_{k-1}\}$. To explore the trade-off

between different quantization schemes, we model the target objective function as

$$\min_{\alpha} \mathcal{G}(\alpha), \tag{2}$$

$$s.t. \quad \mathcal{L}_{val}(w^*, \alpha) - \theta \leq 0, \tag{3}$$

where

$$w^* = \arg\min_{w} \mathcal{L}_{train}(w^*, \alpha). \tag{4}$$

Here $\mathcal{G}$ represents the model size, $\mathcal{L}$ is the cross entropy loss, θ is the expected maximum loss, w denotes the weights of the model, and α represents the coefficient of either quantization method (binary or 8-bit) in a certain layer. In our model, (3) is the constraint for optimization problem (2). We can rewrite the above as a bi-level optimization problem:

$$\begin{aligned} &\min_{\alpha} \max_{\lambda \geq 0} \left(\mathcal{G}(\alpha) + \lambda(\mathcal{L}(\alpha, w^*) - \theta)\right) \\ &\quad s.t. \qquad w^* = \arg\min_{w} \mathcal{L}_{train}(w, \alpha). \end{aligned} \tag{5}$$

To solve this bi-level optimization problem, we adopt the approximate algorithm in [9]. First, we retrain the network to find the weights that result in the minimal loss on the training set. Then the Lagrange multiplier problem is solved by fixing the weights. As shown in Algorithm 1, solving the Lagrange multiplier problem starts with maximizing the target function w.r.t. λ: if $\mathcal{L}(\alpha, w^*) - \theta \leq 0$, λ approaches 0; otherwise λ approaches infinite. Here θ is a tunable hyperparameter representing the tolerance of accuracy drop. Larger θ tolerates less accuracy drop but may also result in smaller compression rate. While smaller θ can potentially achieve a higher compression rate, it may cause larger accuracy drop. Our setting of θ is using the (expected or target) loss in full precision model. Finally we minimize the target function w.r.t. α. Once obtaining the hyperparameter set α with the best trade-off, we retrain the quantization and fine tune the quantized weights to generate the final network model.

3 Experimental Evaluation

We evaluate our proposed methodology on a pretrained 2-layer depth-wise separable convolution neural network using MNIST data set as well as VGG-16 neural network model using CIFAR-10 data set. For each model, we compare our approach with the following baselines: 32-bit floating point (full-precision) model, 8-bit fixed precision model, and binary fixed precision model. As shown in Table 1, the results of MNIST experiment suggest that our algorithm is capable of find a quantization scheme that achieves 28x compression rate while keeping the accuracy drop less than 0.5%. In CIFAR-10 experiment, we set θ as 0.6 for VGG-16. Comparing to whole binary quantization, our approach obtains a compression rate that is very close to binary quantization while gaining 1.5% more accuracy. Figure 2 shows the memory consumption of our model and the original 32-bit full precision model. The memory usage is dramatically decreased especially at the middle layers. It is worth mentioning that our method is orthogonal to weight pruning. Combing with state-of-the-art pruning methods [10][11] which achieve approximately 30x compression rate, the overall compression rate can be up to approximately 900x.

Algorithm 1: Differentiable fine-grained quantization

Initialization;
while not converged **do**
  Update weights w by descending $\nabla_w \mathcal{L}_{train}(w, \alpha)$;
  if $\mathcal{L}_{valid}(w, \alpha) - \theta \leq 0$ **then**
    $\lambda \leftarrow 0$;
  else
    $\lambda \leftarrow inf$;
  end
  Update probability α by descending $\nabla_\alpha(\mathcal{G}(\alpha) + \lambda(\mathcal{L}_{valid}(w^*, \alpha) - \theta))$;
end

Table 1: Comparison of different quantization schemes.

	MNIST		CIFAR-10	
Quant.	Comp.	Accu.(%)	Comp.	Accu.(%)
float32	1×	98.66	1×	84.80
8-bit	4×	98.48	4×	84.07
ours	**28×**	**98.20**	**30×**	**83.06**
binary	32×	96.34	32	81.56

Figure 2: Pretrained 32 bit VGG-16 vs. the mixed precision model generated by our algorithm. The width of a rectangle denotes the size (i.e., memory consumption) of the corresponding layer.

4 Conclusion and On-going Work

In this paper, we propose a differentiable mixed-precision search method for compressing deep neural networks efficiently. Unlike the traditional quantization methods, our approach relaxes quantization bitwidths to a continuous domain and combined with loss function. Deep neural networks can be either quantized from the start of training phase or from a pretrained model using our proposed methodology. Moreover, our approach ensures quantized model remain a similar accuracy while being compressed up to 30X.

The proposed methodology is not tied into any specific neural network topology, so it can potentially be extended to mixed-precision quantization of different neural network architectures, such as RNN and LSTM. We are currently working on providing more quantization options for each layer. For example, each layer can be quantized to x bits, and $x \in \{1, 2, ..., 32\}$. These new quantization options drastically increase the search space. Therefore, We plan to design a predictor combined with autoencdoer-decoder architecture to expedite the search process of layer-wise quantization.

Acknowledgement

This work is supported in part by the following grants: National Science Foundation CCF-1756013, IIS-1838024, 1717657 and Air Force Research Laboratory FA8750-18-2-0057.

References

[1] C. Szegedy and et al. Going deeper with convolutions. In CVPR, 2015.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[3] K. He and et al. Deep residual learning for image recognition. In IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[4] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In ECCV. Springer, 2014.

[5] A. Howard and et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[6] M. Courbariaux and et al. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830, 2016.

[7] M. Rastegari and et al. Xnor-net: Imagenet classification using binary convolutional neural networks. In ECCV, pages 525–542. Springer, 2016.

[8] Yu Bai, Yu-Xiang Wang, and Edo Liberty. Proxquant: Quantized neural networks via proximal operators. arXiv preprint arXiv:1810.00861, 2018.

[9] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055, 2018.

[10] Tianyun Zhang, Kaiqi Zhang, Shaokai Ye, Jiayu Li, Jian Tang, Wujie Wen, Xue Lin, Makan Fardad, and Yanzhi Wang. Adam-admm: A unified, systematic framework of structured weight pruning for dnns. arXiv preprint arXiv:1807.11091, 2018.

[11] Shaokai Ye and et al. Progressive weight pruning of deep neural networks using admm. arXiv preprint arXiv:1810.07378v1, 2018.