Research on Quantitative Optimization Method Based on Incremental Optimization

Ying Chen(✉), Youjun Huang, and Lichao Gao

Xiamen Identity Check Network Technology CO., LTD., Xiamen, China
{cheny,huangyoujun,gaolc}@xmigc.com

Abstract. Existing automatic mixed-precision quantization algorithms focus on search algorithms, ignoring the huge search space and inaccurate performance evaluation criteria. In order to narrow the search space, this paper analyzes the influence of quantization truncation error and rounding error on the performance of quantization model from the perspective of progressive optimization. It was found that for a given model, the quantization truncation error is a constant, while the quantization rounding error is a function of the quantization accuracy. Based on this, this paper proposes a finite-error progressive optimization quantization algorithm. In order to solve the problem of inaccurate performance evaluation criteria, based on quantitative loss analysis and reasoning, this paper proposes a performance evaluation criteria based on Hessian matrix. Adam's second-order gradient is used as proxy information to reduce the computational complexity of Hessian matrix. The method obtains a model that satisfies the hardware constraints in an end-to-end manner. Rigorous mathematical derivation and comparative experiments have proved the rationality of the algorithm, and its performance far exceeds the current mainstream algorithms. For example, on the ResNet-18 network, while achieving a search space reduction of 10^{19}x, the computational efficiency of the model performance evaluation standard is increased by 12 times, and the mixed precision model only loses 0.3% of performance, while achieving a 5.7x compression gain.

Keywords: Neural network quantization · Incremental optimization · Compression and acceleration

1 Introduction

Quantization is a common and well-established algorithm for compression and acceleration of deep convolutional neural networks. This algorithm sets all the convolutional layers of the neural network to a unified low-precision, and performs convolution operations on the low-precision multiplier-adder to achieve the purpose of compression and acceleration. However, different neural network layers have different sensitivities and different degrees of redundancy to different quantization precision settings, and also have different performances on hardware. Reflected on the performance of the entire network, there will be different impacts. Without loss of generality, this uniform precision setting is not optimal. To solve this problem, mixed-precision quantization came into being. For

D.-S. Huang et al. (Eds.): ICIC 2022, LNAI 13395, pp. 729–740, 2022.
https://doi.org/10.1007/978-3-031-13832-4_60

a given neural network with N network layers, assuming that the size of the optional quantization precision space is m, the mixed precision quantization is intended to find the optimal quantization precision for each layer of the neural network. Combined with the network architecture search algorithm, the traditional mixed-precision quantization algorithm is generally divided into three steps: (1) design an optional mixed-precision search space; (2) design a performance evaluation index to measure the performance of each mixed-precision model; (3) Select an appropriate search algorithm and explore in the alternative precision space based on the performance evaluation criteria. Traditional algorithms [2, 24, 26] usually focus on the design of iterative search algorithms, such as reinforcement learning [15, 24, 25], evolutionary learning [2], and gradient-based update [26]. The mixed-precision search space is generally set manually. As for the performance evaluation indicators, the existing mixed-precision quantization algorithms are generally based on the Performance Ranking Hypothesis. For a given network A and network B, if the verification performance of network A is higher than that of network B in the initial training stage, then After both A and B have converged, the performance of network A is often better than that of network B), and the model performance based on one training frequency (Epoch) is used as the model evaluation criterion. In the step-by-step iteration process, the search algorithm and performance evaluation criteria are used to find the optimal mixed-precision strategy in the entire search space. Although these methods have improved the performance of the model, there are still two important and urgent problems in mixed-precision quantization: (1) huge mixed-precision search space; (2) imprecise performance criteria.

The huge mixed-precision search space is an exponential $O(m^2N)$ complexity problem. An effective search space approximation method is urgently needed to achieve the purpose of speeding up the search. The traditional method simply reduces the number of candidate precisions [5] by hand, namely: $m << 32$. However, this method does not completely solve the problem of exponential search space. Manually designing the search space also requires a lot of experimentation and deployment experience, which is also unacceptable in the actual application process. In addition, the limited candidate precision space also greatly reduces the effectiveness of the search algorithm. The second problem of mixed-precision quantization comes from the model performance evaluation criteria. The existing methods are based on the performance ranking assumption, and the model performance after one training frequency (Epoch) is equivalent to the performance after model training convergence. There is a time-consuming problem with performance evaluation criteria, and it is not friendly to heavy neural networks and large datasets. In addition, this paper further studies the correlation between this performance ranking assumption and mixed-precision quantization, and finds that during the iterative update process, due to factors such as parameter sharing and parameter inheritance, the performance ranking assumption is effective in the mixed-precision quantization model search process. Based on the above observations, this paper will focus on the study of the optimization problem of mixed-precision quantization algorithms with huge search space and invalid performance criteria.

Different from the brute force search method in exponential mixed-precision space, this paper studies the problem of parameter sharing and parameter inheritance in the mixed-precision search process from the perspective of asymptotic optimization. The

relationship between progress length, quantization parameters, network parameters and quantization perceptual loss, through the method of incremental optimization, the quantization perceptual loss is limited by the gradual interval in each round of iteration of the quantization model. The search space is greatly reduced, and the efficiency of mixed-precision search is improved. Aiming at the problem of performance evaluation criteria, this paper models the quantization perceptual loss, and proposes a new evaluation criterion based on Hessian matrix information to measure the performance of mixed precision models. In addition, limited by the computational complexity of the Hessian matrix, this paper is further based on the assumption of positive semi-definite diagonal approximation of the Hessian matrix, and introduces the proxy information based on Adam's second-order gradient approximation, realizing that only one batch of data can be used to calculate The goal of performance evaluation criteria further optimizes the entire search process. In addition, in the iterative search process of the whole algorithm, a small amount of iterative training needs to be performed on the intermediate state network to make the performance evaluation index more accurate. In addition, it is noted that this end-to-end search makes the intermediate state model retrained, so only one search is needed to obtain the mixed-precision quantization model with the best performance under the constraints of different parameters and computational constraints.

In order to prove the effectiveness of the algorithm, based on the deep learning framework of Pytorch [20], the author uses the classic ResNet [7], DenseNet [9], MobileNet [22] and other network models as the skeleton structure, CIFAR-100 [11] and ImageNet-2012 [3] and other datasets. Compared with the traditional algorithms, the algorithms proposed in this paper have obtained the best model performance. For example, on the CIFAR-10 dataset, the method in this paper enables the VGG network model to achieve optimal performance under 2-bit accuracy, and it is also effective on heavier neural networks such as ResNet [7] and DenseNet [9]. It compensates for the performance loss due to uniform quantization accuracy [3]. Compared with some of the latest mixed-precision quantization algorithms [5], the algorithm proposed in this paper can achieve better performance at a higher compression rate. In addition, on the ImageNet dataset, compared with the traditional method, the algorithm can ensure that the performance of low-precision deep neural network can be improved while effectively compressing and accelerating the network model. Specifically, on the ResNet-18 network, the mixed-precision model achieves 5.7 times the compression gain while losing 0.3.

2 Related Work

The core idea of neural network quantization is to convert floating-point operations and floating-point representations into fixed-point operations and fixed-point representations, so as to achieve the purpose of network model compression and acceleration. Taking the quantization accuracy as the standard, neural network quantization can be divided into the following two types: (1) low-precision quantization; (2) high-precision quantization. In addition, there is a special class of quantization algorithms called codebook quantization.

Low-precision quantization generally adopts a quantization precision of less than 4-bit, and converts 32-bit floating-point weights into 4, 3, 2, 1-bit fixed-point weights, so as to achieve the purpose of network compression and acceleration. Due to the low quantization accuracy, the performance of the model after quantization is seriously degraded.

The method in [4] directly converted the floating-point network weight parameters to 1-bit, which greatly reduced the memory storage pressure of the model. In addition, this research creatively proposes the Straight-Through Estimator (STE) algorithm, which solves the problem that the gradient of the Sign function is 0 everywhere. Based on this, the quantization model can be directly updated through reverse gradient training to make up for the quantization loss caused by the decrease in quantization accuracy. In addition,

Rastegari et al. [21] used the exclusive-or (XNOR) operation to quantize the network weights and activations into 1-bit at the same time, so that the network model can be compressed and under the condition of special hardware support, it can obtain effective acceleration. The research work also introduces the floating scale factor corresponding to the weight and activation binary quantization, and obtains the analytical solution of the scale factor by optimizing the quantization error. Due to its excellent performance, the scale factor was inherited and developed by subsequent research work, which promoted the research of quantitative algorithms. Wan et al. [23] quantized the network weights and activations into 2- or 3-bit, and converted floating-point matrix multiplication into low-precision logical operations (XNOR, AND) through reasonable design to achieve fast convolution calculations. Lin et al. [16] reduced the quantization loss through the combination of multiple binary expressions, and at the same time alleviated the problem of gradient disappearance to a certain extent. Hu et al. [8] combined the hash algorithm with binary quantization for the first time, converted the objective function into an optimized hash code, and used the hash function to map floating-point weights or activations to the low-precision expression space. [17] proposed Cyclic Filters (CiFs) and Cyclic Binary Convolution (CBConv) to enhance the feature extraction capability of binary convolutions, and proposed Cyclic Back Propagation (CBP) to train the network model. In addition to optimizing the low-precision quantization expression space, Cai et al. [5] proposed a half-wave Gaussian quantizer, which focused on the optimization of gradient mismatch problems and improved the performance of low-precision models. Gong et al. [6] also paid attention to the problem of gradient error caused by STE, and proposed to optimize the Tanh function scale and bias factor through training, and at the same time convert the quantized truncation parameters into learnable parameters, and use gradient approximation to improve the performance of binary networks. In addition, in order to further improve the accuracy of the binary network, Bai et al. [1] proposed a low-precision model training strategy with alternating update optimization. Martinez et al. [19] proposed a binary module, combined with knowledge distillation to improve the flow of feature information. Bi-Real [18] introduced residual connections to transfer floating point values of binary networks. These algorithms all improve the performance of binary networks from different perspectives.

Quantization is an efficient neural network compression and acceleration algorithm that is orthogonal to other compression and acceleration algorithms. However, as mentioned above, the actual acceleration and compression effects are strongly dependent on hardware implementation. Since current general-purpose hardware accelerators can only support 16-bit and 8-bit acceleration, special hardware may support lower-precision model acceleration. Therefore, the underlying hardware also limits the application and deployment of quantization algorithms to a certain extent.

3 Method

3.1 Problem Formulation

In the traditional quantization method, the quantization accuracy of all activation values and weight values of the deep convolutional neural network is uniformly set to k-bit. However, different neural network layers have different sensitivities to different quantization accuracy. In order to effectively improve the performance of the neural network model after quantization, it is necessary to set different quantization precisions for the neural network layers with different sensitivities. In general, higher quantization precision should be set for those network layers that are sensitive to precision, and vice versa. Therefore, more and more researchers are devoted to finding an efficient method to determine the quantization accuracy of each neural network layer.

Given a special dataset D, a pre-trained deep convolutional neural network (N network layers) and a candidate precision space (K, the general case Next, $K = [1, 2, \ldots, 32]$). Mixed-precision quantization strives to find the most appropriate quantization accuracy for each computationally intensive network layer of the neural network within the range of the smallest possible loss of accuracy. However, there is currently no efficient way to explore such a large search space ($|K|^{2N}$). Some scholars [2, 24, 26] proposed to approximate the entire candidate precision space through a learning-based method, that is, directly reduce the number of optional quantization precisions ($K << 32$). However, this method does not really solve the problem of huge precision search space. In addition, in order to be able to judge which mixed-precision strategy is better, it is necessary to detect the impact of quantization accuracy on the entire deep convolutional neural network. Performance Ranking Hypothesis is a commonly used method. However, this method requires a lot of computation for both training and testing, and its time complexity is very high for heavy-duty neural networks and large datasets. This paper introduces a new mixed-precision search space approximation and model performance evaluation method, which are respectively called: asymptotic-based quantization algorithm (ProQ) and Hessian matrix information-aware performance evaluation standard (Hessian-Aware Indicator), collectively referred to as ProQHA.

3.2 Progressive Optimization Mixed Precision Quantization

In order to solve the problem of the huge search space, this section proposes a quantization algorithm based on incremental optimization, which gradually optimizes the high-precision model to the corresponding low-precision model step by step, namely: 32-bit → 8-bit → 4bit → 2-bit. Specifically, given a candidate precision space K, where $k_m < k_{m-1} < \ldots < k_1$, km and k1 correspond to the minimum and maximum quantization precisions in the candidate precision space, respectively. As shown in (b) and (c) of Fig. 1, in a deep convolutional neural network, the search space of any layer is iterated with progressive intervals: for a given network layer, The number in each diamond represents the candidate quantization accuracy, the number in each circle represents the accuracy of the pre-trained model, and τ is the progressive quantization interval. Figure 1(a) represents a traditional search space where candidate precisions are randomly sampled at each iteration step. Figure 1(b) and (c) are both progressively quantized search spaces,

where τ is equal to 1 and 2, respectively. At any iteration step, the mixed-precision search space of this layer depends only on the accuracy of the pre-trained model and the progressive quantization interval (τ) in the previous iteration.

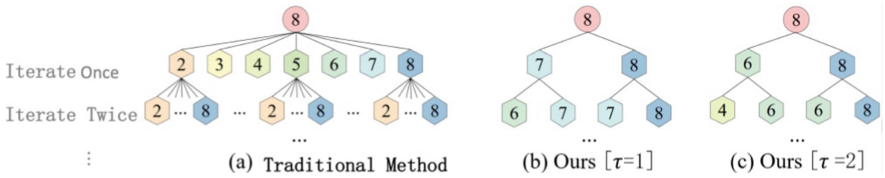

Fig. 1. Comparative analysis of single-layer search space

The mixed-precision space after asymptotic optimization effectively solves the problem that the mixed-precision search space is too large. In addition, it can be noticed that the quantized model of (k − τ)-bit inherits the weights of the k-bit model, which brings additional advantages, namely: the optimized solution based on the k-bit model, (k − τ)-bit model is easier to optimize.

3.3 Model Structure

To express the hardware constraints for model compression and acceleration, this paper defines the model cost as:

$$\text{cost} = \frac{\sum_{w_i \in W} P_{w_i} \times K_{w_i}}{P_w} \tag{1}$$

where *P(·)* and *K(·)* represent the number of neural network parameters and the corresponding accuracy of the quantization layer, respectively. *P(·)* can also use floating point operands (FLOPs) to represent hardware resource constraints. Figure 2 shows the quantization structure based on ResBlock. As shown in the figure, the paper makes appropriate changes to the network structure, the batch normalization layer and activation quantization layer are performed before the convolutional layer. The algorithm performs a mixed-precision search at the input and the convolution kernel, respectively. The network parameter quantization uses the algorithm based on Dorefa [29], and the activation value uses the algorithm based on PACT [3] to perform the quantization operation of the corresponding precision. Furthermore, this paper does not quantify the first and last layers of the model to ensure stable performance. See Algorithm 1 for the complete flow of the automatic mixed-precision quantization algorithm based on progressive optimization.

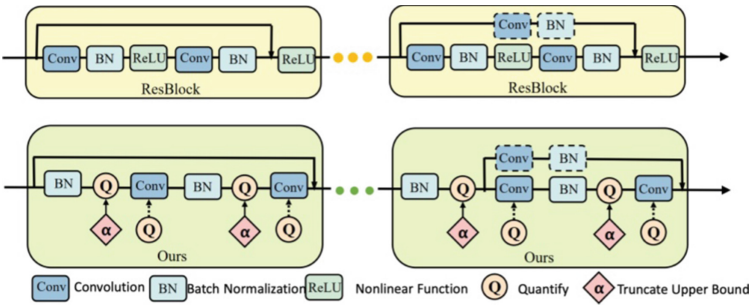

Fig. 2. Schematic diagram of quantization structure based on ResBlock

Algorithm 1: Based on progressive optimization mixed precision quantization (ProQHA) algorithm flow

Input: training dataset D, candidate precision space $K = [k_1, k_2, ...k_m]$, pretrained model M with all layer precision k1, random sampling number p, random sampling ratio r and fixed asymptotic interval T and Resource Constraints C.

Output: Mixed Precision Model M_s;

Randomly sample a single batch of data d from the entire dataset D;
while *cost ge C* **do**
  Generate asymptotic accuracy k based on the asymptotic interval T and the accuracy at the current iteration step of the model;
  for *j=1,..,p* **do**
    Based on the sampling probability r, set the accuracy of the sampling layer to the progressive accuracy h; keep the accuracy of the unsampled layer unchanged;
    Based on formula 3.10 and a single batch of data d, calculate the performance evaluation standard value of each layer;
  end
  Based on the performance evaluation standard value, select the optimal mixed precision setting S;
  Set the precision of the model M to S to generate a mixed precision model M_s;
  Retrain the model M_s;
  Calculate the current model cost cost based on Equation 3.11;
end
Return M_s;

4 Experiment

The algorithm is based on Pytorch, and the corresponding verification experiments are done on the CIFAR-100 and ImageNet-2012 datasets. This section expounds the effectiveness of the algorithm in terms of experimental setup, performance comparison experiments, ablation analysis experiments, and visualization experiments, respectively.

4.1 Experimental Setup

In the initial pre-training stage of the model, for CIFAR-10, we train by using the stochastic gradient descent (SGD) optimization algorithm with momentum 0.9, where the initial learning rate and learning rate decay are set to 0.1 and 5e−4, respectively, for a total of 78,200 iterations. The experimental settings of ImageNet mainly refer to previous work [28]. In the mixed-precision search stage, the mixed-precision search algorithm based on progressive optimization will search with a data batch size of 128, and perform a total of one iteration. Adam is used for optimization during this process, while 600 samples are randomly sampled to calculate performance evaluation metrics. The optimal mixed-precision model was selected to re-execute the iterative optimization process for 30 training epochs (Epoch) to restore model performance. During the whole retraining process, the initial learning rate is 0.001, and at the 10th and 20th Epochs, the learning rate is reduced to 10% of the original. For models that meet the hardware constraints of the model, this paper reports the highest validation performance after retraining the model after 100 training epochs.

Table 1. Experimental results of quantifying the model on the CIFAR-100 dataset.

Model	Precision	ResNet-18		ResNet-50		DenseNet-40		MobileNetV2	
	WA	Acc.-1	Cost	Acc.-1	Cost	Acc.-1	Cost	Acc.-1	Cost
PACT [3]	2/2	66.49	0.0625	66.68	0.0625	71.01	0.0625	NA	0.0625
ProQHA	MP	68.80	0.0625	71.48	0.06	72.39	0.0625	68.88	0.0625
PACT [3]	4/4	70.68	0.1250	71.98	0.1250	73.63	0.125	9.93	0.1250
ProQHA	MP	70.26	0.1077	73.40	0.1215	73.69	0.1253	72.54	0.1231
PACT [3]	6/6	70.46	0.1875	72.73	0.1875	73.90	0.1875	50.29	0.1875
ProQHA	MP	70.87	0.1746	73.49	0.1810	74.25	0.1828	74.61	0.1717
PACT [3]	8/8	70.48	0.2500	72.60	0.2500	73.73	0.2500	68.21	0.2500
ProQHA	MP	70.94	0.2064	73.72	0.2311	74.05	0.2349	74.56	0.2356
Baseline	32	71.05	1.000	74.34	1.000	74.68	1.000	69.61	1.000

4.2 CIFAR-100

This paper further conducts experimental verification on CIFAR-100. Table 1 shows that the performance of the algorithm proposed in this paper is higher than that based on the uniform precision quantization algorithm. Compared with PACT [3], the quantization algorithm based on progressive optimization achieves higher model compression rate and higher computational speedup ratio. In particular, the size of the mixed-precision model obtained based on the algorithm in this paper is 1 of the original floating-point model, but it achieves similar performance to the original model. Such as: ResNet-50 (18.18%) v.s. MobileNetV2 (17.17%). For a more compact model (MobileNetV2), the traditional method cannot obtain effective accuracy when the quantization accuracy is

2, while the quantization algorithm based on asymptotic optimization can still obtain better accuracy on the compact model under the condition of lower accuracy.

Table 2. Based on ImageNet [12], the quantization algorithm comparison of uniform precision setting with ResNet-18 as the model skeleton. "Parameter Accuracy/Activation Accuracy" represents the accuracy used for the quantization of neural network model parameters and activation values. Based on a single 1080Ti GPU, each iteration of ProHQA takes one day.

Model	Parameter accuracy/Activation accuracy	Model performance	Performance drop	Parameter compression	Activate compression
Baseline	32/32	70.20	0.00	1.00	1.00
ABC-Net [16]	5/5	65.00	−5.20	6.40	6.40
Dorefa [29]	5/5	68.40	−1.80	6.40	6.40
ProQHA	**MP/MP**	**70.01**	**−0.10**	**6.46**	**7.01**
ABC-Net [16]	3/3	61.00	−9.20	10.67	10.67
Dorefa [29]	3/3	67.50	−2.70	10.67	10.67
PACT [3]	3/3	68.10	−2.10	10.67	10.67
LQ-Nets [27]	3/3	68.20	−2.00	10.67	10.67
ProQHA	**MP/MP**	**68.34**	**−1.86**	**10.67**	**10.04**
Dorefa [29]	2/2	62.60	−7.60	16.00	16.00
PACT [3]	2/2	64.40	−5.80	16.00	16.00
LQ-Nets [27]	2/2	64.90	−5.30	13.84	12.19
DSQ [6]	2/2	65.20	−5.00	16.00	16.00
QIL [10]	2/2	65.70	−4.50	11.20	1.00
ProQHA	**MP/MP**	**66.18**	**−4.02**	**13.87**	**11.90**

4.3 ImageNet

In order to further verify the effectiveness of the algorithm, this paper also conducts comparative experiments on the ImageNet dataset. The results are shown in Table 2. Based on a single 1080Ti GPU, each iteration of ProHQA takes one day. For example, a total of 12 iterations are used to obtain the result 65.72 in the following table. It is easy to know that while ProQHA obtains a higher compression rate, it obtains performance similar to that of the floating-point model. For example, compared with floating-point ResNet-18, the model generated by ProQHA achieves better model performance with a smaller model cost. That is: ProQHA can effectively improve the performance of lower precision models. It is easy to know that the stability of ProQHA is better than other algorithms. It is worth noting that random search may induce suboptimal mixed precision

settings. But ProQHA still achieves the best model performance. For example, compared with BRECQ [14], ProQHA can obtain more stable model performance with a similar compression rate. This proves from the experimental point of view that the quantization algorithm based on asymptotic optimization can optimize the low-precision model within the bounded error, which greatly improves the effect of search stability and demonstrates the superiority of ProQHA over other mixed-precision quantization algorithms.

5 Conclusion

In the field of mixed-precision quantization, the method proposed in this paper is the first algorithm to simultaneously optimize the search space and performance estimation. A novel automatic mixed-precision quantization framework is proposed, which includes a progressive optimization-based mixed-precision quantization (ProQ) algorithm and a Hessian matrix information-aware-based model performance evaluation metric. The algorithm performs a progressive quantization algorithm on randomly sampled layers to reduce the mixed-precision search space. At the same time, the theory demonstrates the stability of asymptotic quantization for smaller intervals. In the iterative process of incremental optimization, the model performance evaluation criteria based on Hessian matrix information perception are incorporated to select the current optimal mixed-precision strategy. Finally, this paper proves the rationality and effectiveness of ProQHA with a large number of experiments, and finally achieves the goal of optimization based on the quantization precision algorithm.

References

1. Bai, Y., Wang, Y.-X., Liberty, E.: ProxQuant: quantized neural networks via proximal operators. arXiv preprint arXiv:1810.00861 (2018)
2. Chen, Y., et al.: Joint neural architecture search and quantization. arXiv preprint arXiv:1811.09426 (2018)
3. Choi, J., Wang, Z., Venkataramani, S., Chuang, P.I.-J., Srinivasan, V., Gopalakrishnan, K.: PACT: parameterized clipping activation for quantized neural networks. arXiv preprint arXiv:1805.06085 (2018)
4. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: training deep neural networks with weights and activations constrained to $+1$ or -1. arXiv preprint arXiv:1602.02830 (2016)
5. Dong, Z., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: HAWQ: Hessian AWare Quantization of neural networks with mixed-precision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 293–302 (2019)
6. Gong, R., et al.: Differentiable soft quantization: bridging full-precision and low-bit neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4852–4861 (2019)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. Hu, Q., Wang, P., Cheng, J.: From hashing to CNNs: training binary weight networks via hashing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
10. Jung, S., et al.: Learning to quantize deep networks by optimizing quantization intervals with task loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4350–4359 (2019)
11. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25 (2012)
13. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710 (2016)
14. Li, Y., et al.: BRECQ: pushing the limit of post-training quantization by block reconstruction. arXiv preprint arXiv:2102.05426 (2021)
15. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
16. Lin, X., Zhao, C., Pan, W.: Towards accurate binary convolutional neural network. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
17. Liu, C., et al.: Circulant binary convolutional networks: enhancing the performance of 1-bit DCNNs with circulant back propagation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2691–2699 (2019)
18. Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., Cheng, K.-T.: Bi-Real Net: enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11219, pp. 747–763. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01267-0_44
19. Martinez, B., Yang, J., Bulat, A., Tzimiropoulos, G.: Training binary neural networks with real-to-binary convolutions. arXiv preprint arXiv:2003.11535 (2020)
20. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)
21. 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). https://doi.org/10.1007/978-3-319-46493-0_32
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
23. Wan, D., et al.: TBN: convolutional neural network with ternary inputs and binary weights. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 322–339. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_20
24. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: HAQ: hardware-aware automated quantization with mixed precision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8612–8620 (2019)
25. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**(3), 279–292 (1992)
26. Wu, B., Wang, Y., Zhang, P., Tian, Y., Vajda, P., Keutzer, K.: Mixed precision quantization of ConvNets via differentiable neural architecture search. arXiv preprint arXiv:1812.00090 (2018)
27. Zhang, D., Yang, J., Ye, D., Hua, G.: LQ-Nets: learned quantization for highly accurate and compact deep neural networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11212, pp. 373–390. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01237-3_23

28. Zheng, X., Ji, R., Tang, L., Zhang, B., Liu, J., Tian, Q.: Multinomial distribution learning for effective neural architecture search. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1304–1313 (2019)
29. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: DoReFa-Net: training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016)