# Neural Network Quantization with Scale-Adjusted Training

Qing Jin[1, 3]
jinqingking@gmail.com
Linjie Yang[1]
linjie.yang@bytedance.com
Zhenyu Liao[2]
liaozhenyu2004@gmail.com
Xiaoning Qian[3]
xqian@tamu.edu

[1] ByteDance Inc.
US AI Lab
Mountain View, California, USA

[2] Kwai Inc.
US AI Lab
Palo Alto, California, USA

[3] Texas A&M University
Department of Electrical and Computer Engineering
College Station, Texas, USA

**Abstract**

Quantization has long been studied as a compression and accelerating technique for deep neural networks due to its potential on reducing model size and computational costs, for both general hardware, such as DSP, CPU or GPU, and customized devices with flexible bit-width configurations, including FPGA and ASIC. However, previous works generally achieve network quantization by sacrificing on prediction accuracy with respect to their full-precision counterparts. In this paper, we investigate the underlying mechanism of such performance degeneration based on previous work of parameterized clipping activation (PACT). We find that the key factor is the weight scale in the last layer. Instead of aligning weight distributions of quantized and full-precision models, as generally suggested in the literature, the main issue is that large scale can cause over-fitting problem. We propose a technique called *scale-adjusted training (SAT)* by directly scaling down weights in the last layer to alleviate such over-fitting. With the proposed technique, quantized networks can demonstrate better performance than their full-precision counter-parts, and we achieve state-of-the-art accuracy with consistent improvement over previous quantization methods for light weight models including MobileNet V1/V2 on ImageNet classification.

## 1 Introduction

Deep neural networks have gained rapid progress in tasks including computer vision, natural language processing and speech recognition [25, 35, 36, 39, 48, 53, 56], and have been applied to real-world systems such as robotics and self-driving cars [19, 27]. However, it remains challenging to deploy the heavy deep models to resource-constrained platforms such as mobile phones and wearable devices. To make deep neural networks more efficient on model size, latency and energy, several approaches have been developed such as weight prunning [14], model slimming [2, 29, 54], and quantization [6, 7]. Recent works even apply

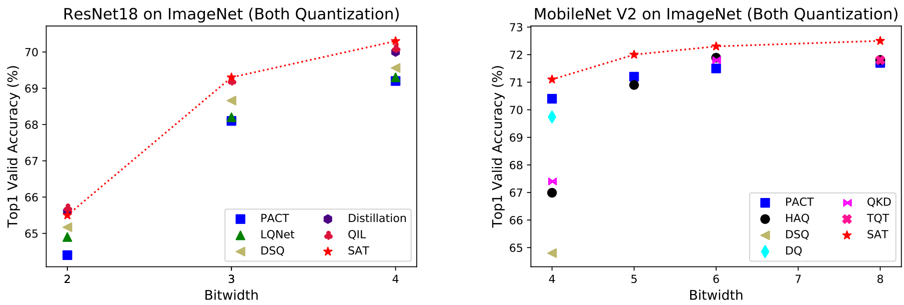



Figure 1: Comparison of quantization approaches with ResNet18 and MobileNet V2 on ImageNet under different quantization levels. Note that both weights and activations are quantized in these two plots. Left: ResNet18. Right: MobileNet V2. The bit-width represents equivalent computation cost for mixed-precision methods (AutoQB and HAQ).

neural architecture search (NAS) algorithms to determine the architecture of the model, to achieve the best trade-offs between resource budget and model performance [9, 26, 28, 33, 55]. As one of the promising methods, quantization provides the opportunity to embed bulky and computation-intensive models onto platforms with limited resources. By sacrificing the precision of weights [6, 14, 23, 24, 60] and even features [5, 7, 12, 17, 34, 37, 38, 40, 42, 52, 57, 58, 59, 62], model size can be shrunk to a large extent, and full-precision multiplication is replaced by low-precision fixed-point multiplication, addition or even bitwise operations, requiring much reduced latency and energy consumption during inference.

Despite these advantages on improving model efficiency, quantization is well known to suffer from significant accuracy reduction [3, 5, 22, 42, 45]. Recent works tackle this problem by adopting sophisticated training strategies [11, 18, 20, 21, 57] or by leveraging mixed precisions which are automatically searched [10, 31, 47, 49, 51]. However, previous works either lack a thorough study on a wide range of model structures, especially those light models such as MobileNets [11, 20, 57], or do not show consistently better performance of quantized models compared to their full-precision counterparts in all experiments [11, 18, 21]. It remains unclear whether we can achieve better performance from quantized neural networks than their full-precision counterparts. More fundamentally, what is the underlying reason of performance degeneration from network quantization.

In this paper, we study this problem by identifying the key factor impacting the prediction accuracy of quantized neural networks. Specifically, we focus on a recently popular technique of parameterized clipping activation (PACT) [5], which is based on the DoReFa quantization scheme [59]. We first study the effect of the clamping operation, which is a part of weight quantization in DoReFa, on weights in the last fully-connected layer. We find this clamping operation will enlarge the weight scale, especially for layers with a large number of neurons. Moreover, large weight scale will cause over-fitting, even for full-precision models, which is shown to be the main reason of performance degeneration for quantized networks. Based on this, we propose a simple yet effective technique named scale-adjusted training (SAT), with which the over-fitting problem is alleviated and quantized networks can even outperform their full-precision counterparts.

**Summary of Contributions** The main contributions of our work are summarized as follows:

1. We find that the large scale of weights caused by clamping operation in popular quantization schemes hampers the performance of the model, resulting in over-fitting issues. This draws a new respective compared to the previous claims that improper distribution of weights is the major reason for performance degeneration in quantization.

2. We propose a simple yet effective technique for neural network quantization and achieve state-of-the-art quantization performance for light weight models including MobileNet V1/V2 on ImageNet classification.

3. We demonstrate experimentally that quantized neural networks can outperform their full-precision counterparts and provide consistent results for MobileNet V1/V2 and ResNet18/50 on ImageNet classification.

## 2 Related Work

**Uniform-precision quantization.** Quantization of deep models has long been discussed since the early work of weight binarization [6, 7] and model compression [14]. Many previous methods enforce the same precision for weights/activations in different layers during quantization. Early approaches focus on minimizing the difference in values [42] or distributions [14, 57] between quantized weights/activations and full-precision ones. Recently, [57] proposes a learning-based quantization method, where the quantizer is trained from data. Regularizer for quantization is also proposed to implement binarized weights [3]. Ensemble of multiple models with low precision has also been studied [61], demonstrating improved performance than individual models under the same computation budget. [11] proposes a quantizer with trainable step size, and improves training convergence by balancing the magnitude of step size updates with weight updates, based on some heuristic analysis. However, this method focuses on training the step size, and scales the gradients, instead of analyzing the impact of model weights themselves on the training dynamics. Previous works have not shown consistently improved performance of quantized networks to their full-precision counterparts.

**Mixed-precision quantization.** Recent work attempts to use mixed-precision in one model, in which weights and activations in different layers are assigned different bit-widths, resulting in better trade-offs between efficiency and accuracy of neural networks. Towards this end, automated algorithms are adopted to determine the most appropriate bit-width for each layer. Reinforcement learning has been adopted to search for bit-width configurations with guidance based on memory and computation cost [10, 31] or latency and energy produced by hardware simulators [49]. [51] and [47] apply differentiable neural architecture search methods to efficiently explore the search space. Although these methods result in more flexible quantization architectures, their performances are still inferior to full-precision models.

**Weight Scaling and Generalization.** Previous works on quantization also apply scaling on integer weights [14, 42, 57, 60]. However, these methods mainly focus on aligning weight distributions of quantized and full-precision models. As another research topic, [41, 43] suggest that weight normalization/standardization is able to improve neural network performance, but their methods either require a trainable scale [43], or must have other normalization operations following batch or group normalization [41]. Moreover, none of these has studied the effect of weight scale on generalizability of neural networks. Neither do they discuss neural network quantization. [1, 4, 32, 44, 50] theoretically analyze the impact of weight scale on generalizability of neural networks but their analyses aim at general theoretical guidance rather than being directly practical in real-world tasks.

# 3 Scale Adjusted Training

## 3.1 DoReFa Scheme

Following previous work PACT [5], we adopt the DoReFa scheme [59] for weight quantization, and the PACT technique for activation quantization. The DoReFa scheme [59] involves two steps, clamping and quantization. Clamping transforms weights to values between 0 and 1, while quantization rounds weights to the nearest integers. We here analyze the impact of both steps on model performance.

### 3.1.1 Impact of Clamping

Before quantization, the weights are first clamped to the interval between 0 and 1. For a weight matrix $W$, we first clamp it to

$$\widetilde{W}_{ij} = \frac{1}{2}\left(\frac{\tanh(W_{ij})}{\max\limits_{r,s}|\tanh(W_{rs})|} + 1\right) \tag{1}$$

which is between 0 and 1. This transformation generally contracts the scale of large weights, and enlarges the difference of small scale elements. Thus, this clamping operation makes variables distributed more uniform in the interval $[0,1]$, which is beneficial for reducing quantization error.

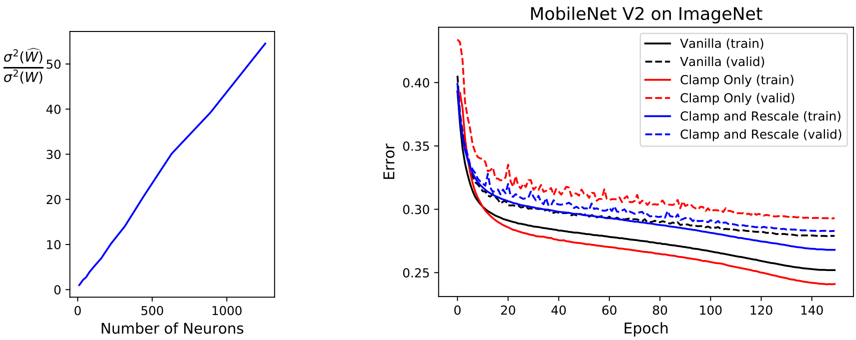
To understand the effect of clamping on prediction accuracy, we first analyze a model using clamped weights without quantization, following the DoReFa scheme

$$\widehat{W}_{ij} = 2\widetilde{W}_{ij} - 1 \tag{2}$$

Fig. 2a gives the ratio between variances of the clamped and the original weights with respect to the number of neurons. As a common practice [15], the original weights $W$ are sampled from a Gaussian distribution of zero mean and variance proportional to the reciprocal of the number of neurons. We find that for large neuron numbers, the variance of weights can be enlarged to tens of their original values.

To see the effect of such scale enlargement, we train a MobileNet V2 on ImageNet with and without clamping, and compare their learning curves. As shown in Fig. 2b, clamping impairs the training procedure significantly, reducing the final accuracy by as much as 1%. Also, we notice that clamping makes the model more prone to the over-fitting issue, which is consistent with the previous literature claiming that increasing weight variance in neural networks might worsen their generalization property [1, 4, 32, 44, 50]. In S1 we provide more detailed analysis on this problem. Moreover, since the number of output neurons of the last linear layer is determined by the number of class labels, we expect large datasets such as ImageNet [8] to be more vulnerable to this problem than small datasets such as CIFAR10. This partially explains the situation that some of previous methods gave good results on small datasets but failed to work on large datasets.

To deal with this problem, we propose a method named scale-adjusted training (SAT) to restore the scale of weights. We directly multiplies the normalized weight with the square root of the reciprocal of the number of neurons in the linear layer as in Eq. (3). Here $\mathbb{VAR}[\widehat{W}_{rs}]$ is the sample variance of elements in the weight matrix, calculated by averaging the square of elements in the weight matrix. In back-propagation, $\mathbb{VAR}[\widehat{W}_{rs}]$ is viewed as constant and receives no gradient. The factor $\sqrt{\widehat{n}_{\text{out}}}$ in the denominator is inspired by the




(a) (b)

Figure 2: Effect of weight clamping. (a) The ratio of variances with respect to the number of neurons. Note that the plot is only a sampling result and different samples can give different results, but the order of magnitude remains meaningful. (b) Learning curves with different settings. Here, "clamp only" refers to using clamped weight without quantization, following the DoReFa scheme [59].

condition of Kaiming initialization [15], where $\widehat{n}_{\text{out}}$ represents the number of output features of the last fully-connected layer. This simple strategy is named *constant rescaling* and works well empirically across all of the experiments. Note that here we have ignored the difference between weight variances across channels and just use variance of the weights in the whole layer for simplicity.

$$W_{ij}^* = \frac{1}{\sqrt{\widehat{n}_{\text{out}} \mathbb{VAR}[\widehat{W}_{rs}]}} \widehat{W}_{ij} \tag{3}$$

Fig. 2b compares the learning curves of the vanilla method, and weight clamping with and without constant rescaling. It shows that SAT alleviates the over-fitting issue and improves the validation accuracy significantly after weight clamping. We also experiment with an alternative rescaling approach in S2 and notice similar performance. In the following experiments we will always use constant rescaling. For MobileNet V2, we only need to apply SAT to the last fully-connected layer. For other models where convolution is not directly followed by BN such as full pre-activation ResNet [16], we find that it is important to also apply SAT to all such convolution layers (see S3 for more details, and this also applies to fully-connected layers without BN following, such as those in VGGNet [46]). Before further discussion, we want to emphasize that the clamping is only a preprocessing step for quantization and there is no quantization operation involved up to now.

### 3.1.2 Impact of Weight Quantization

With weights clamped to $[0,1]$, the DoReFa scheme [59] further quantizes weights with the following function

$$q_k(x) = \frac{1}{a} \left\lfloor ax \right\rceil \tag{4}$$

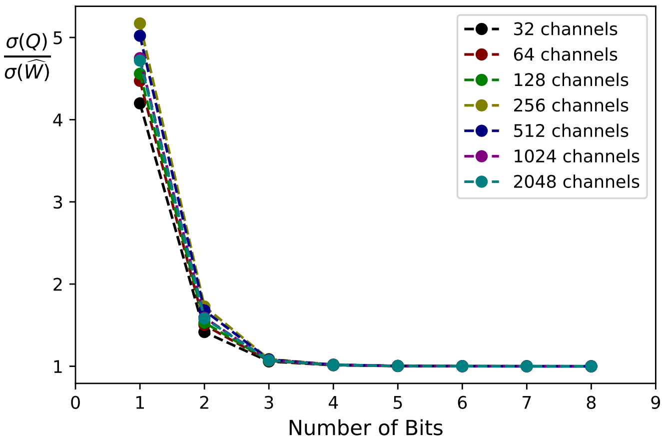



Figure 3: Impact of weight quantization on the variance of effective weight under different channel numbers.

Here, $\lfloor \cdot \rceil$ indicates rounding to the nearest integer, and $a$ equals $2^k - 1$ where $k$ is the number of quantization bits. Quantized weights are given by

$$Q_{ij} = 2q_k(\widetilde{W}_{ij}) - 1 \tag{5}$$

To see the impact of quantization on the model performance, we compare the variance of the quantized weight $Q_{ij}$ with the variance of the full-precision clamped weight $\widehat{W}_{ij}$ . Fig. 3 shows the ratio between the standard deviations of them with respect to the number of bits for different channel numbers, which determines the variances of the original non-clamped weights $W$. We can find that for precision higher than 3 bits, quantized weights have nearly the same variance as the full-precision weights, indicating quantization itself introduces little impact on model performance. However, the discrepancy increases significantly for low precision such as 1 or 2 bits, thus we should use the variance of quantized weights $Q_{ij}$ for standardization, rather than that of the clamped weights $\widehat{W}_{ij}$. For simplicity, we apply constant scaling to the quantized weights of linear layers without BN by

$$Q^*_{ij} = \frac{1}{\sqrt{\widehat{n}_{\text{out}} \mathbb{VAR}[Q_{rs}]}} Q_{ij} \tag{6}$$

We also notice that different channel numbers give similar results.

For typical models such as MobileNets and ResNets, only the last fully-connected layer needs to be rescaled, and such rescaling is only necessary during training.For inference, the scaling factor (which is positive) can be discarded, with the bias term being modified accordingly, introducing no additional operations. For models with several fully-connected layers such as VGGNet [46], or with convolution layers not followed by BN layers, such as fully pre-activation ResNet, the scaling factors for these layers can be applied after computation-intensive convolutions or matrix multiplications, adding marginal computational cost.

# 4 Experiments

## 4.1 Basic Quantization Strategy

Historically, quantization of neural networks follows different conventions and settings [22]. Here we describe the settings adopted in this paper to avoid unnecessary ambiguity. We first train the full-precision model, which is used as the baseline for comparison. For quantized models, we use the pretrained full-precision model as the initialization, and apply the same training hyperparameters and settings as full-precision model training (including initial learning rate, learning rate scheduler, weight decay, the number of epochs, optimizer, batch size, etc.) to finetune the quantized model. For the input images to the model, we use unsigned 8bit integer (uint8) without standardization (neither demeaning nor normalization). Previous works sometimes avoid quantizing the first and last layers due to accuracy drop [51, 59]. We follow a more practical setting to quantize weights in both layers with a minimum precision of 8bit [5] in our main results. To investigate the effect of quantization levels in these two layers, additional results are shown in S3. The input to the last layer is quantized with the same precision as other layers. As a widely adopted convention [5, 49], bias in the last fully-connected layer(s) and the batch normalization (BN) layers (including weight, bias and the running statistics) are not quantized. Note that no bias term is used in convolution layers.

## 4.2 Experiment Details & Discussion

We apply the SAT technique to popular models including MobileNet V1, MobileNet V2, ResNet18, ResNet50 on ImageNet. For all experiments, we use the cosine learning rate scheduler [30] without restart. Learning rate is initially set to 0.05 and updated every iteration for 150 epochs. We use SGD optimizer, Nesterov momentum with a momentum weight of 0.9 without damping, and weight decay of $4 \times 10^{-5}$. The batch size is set to 2048 , and we adopt the warmup strategy suggested in [13] by linearly increasing the learning rate every iteration to a larger value (batch size$/256 \times 0.05$) for the first five epochs before using the cosine annealing scheduler. The input image is randomly cropped to $224 \times 224$ and randomly flipped horizontally, and is kept as 8 bit unsigned integer with no standardization applied. Note that we use full-precision models with clamped weights as initial points to fine tune quantized models.

We compare our method with techniques in recent publications, including uniform-precision quantization algorithms such as PACT [5], LQNet [57], LSQ [11], QKD [21], and mixed-precision approaches such as HAQ [49]. Validation accuracy with respect to quantization levels for ResNet18 and MobileNet V2 where both weights and activations are quantized is plotted in Fig. 1. It is obvious that our method gives significant and consistent improvement over previous methods under the same resource constraint. More thorough comparisons for quantization on MobileNets with or without quantized activation are listed in Table 1 and 2, respectively. Surprisingly, the quantized models with our approach not only outperform all previous methods under all quantization levels, including mixed-precision algorithms, but even outperform full-precision ones when the quantization is moderate ( $\geq 5$ bits for both quantization and 4 bits for weight-only quantization).

Table 3 compares different quantization techniques on ResNet18 and ResNet50. Results of compared methods are from corresponding papers [5, 11, 20, 57]. Since the topology of ResNets has several different versions with significantly different performance even for the full-precision models, we think the accuracy gap between quantized and full-precision models is a more reasonable metric for comparison. Thus, in Table 3, we also list accuracy drop next to the absolute value for each quantization level, where a positive value indicates that the quantized model achieves better performance. We find that our technique is better than existing ones for deeper architectures such as ResNet50, and among the top two in all the experiments. Moreover, our method is able to give consistent improvement over full-precision counterparts for moderate quantization of 4bits.

Table 1: Comparison of quantization techniques with both weights and activation quantized.

| | | MobileNet V1 | | MobileNet V2 | |
|---|---|---|---|---|---|
| Quant. Method | Bit-widths | Acc.-1 | Acc.-5 | Acc.-1 | Acc.-5 |
| PACT [5] | 4bits | 70.3 | 89.2 | 70.4 | 89.4 |
| HAQ [49] | flexible | 67.40 | 87.90 | 66.99 | 87.33 |
| SAT (Ours) | 4bits | **71.3** | **89.9** | **71.1** | **90.0** |
| PACT [5] | 5bits | 71.1 | 89.6 | 71.2 | 89.8 |
| HAQ [49] | flexible | 70.58 | 89.77 | 70.90 | 89.91 |
| SAT (Ours) | 5bits | **71.9** | **90.3** | **72.0** | **90.4** |
| PACT [5] | 6bits | 71.2 | 89.2 | 71.5 | 90.0 |
| HAQ [49] | flexible | 71.20 | 90.19 | 71.89 | 90.36 |
| SAT (Ours) | 6bits | **72.3** | **90.4** | **72.3** | **90.6** |
| PACT [5] | 8bits | 71.3 | 89.7 | 71.7 | 89.9 |
| HAQ [49] | flexible | 70.82 | 89.85 | 71.81 | 90.25 |
| SAT (Ours) | 8bits | **72.6** | **90.7** | **72.5** | **90.7** |
| PACT [5] | FP | 72.1 | 90.2 | 72.1 | 90.5 |
| SAT (Ours) | FP | 71.7 | 90.2 | 71.8 | 90.2 |

From another view, compared with model pruning techniques, our results prove that quantization is more effective on reducing model size and computational cost, and introduces much less impairment on the predicting capability of the compressed model. As a simple comparison, quantizing MobileNet V2 to 6-bit compresses the model size by roughly $4.74\times$ and reduces the BitOPs by $14.25\times$, while scaling the model's channel numbers by a width-multiplier of $0.35\times$ only shrinks the model size by $2.06\times$ and cuts down the FLOPs by $5.10\times$ [54]. 6-bit MobileNet V2 demonstrates better predictive accuracy than the full-precision model, while reducing the channel numbers to $0.35\times$ will significantly impair its performance [54]. A recent pruning method Knapsack Pruning with Inner Distillation [2] shows 0.27% accuracy drop with 40.64% reduction of FLOPs for ResNet50 on ImageNet, and another work Network Pruning via Transformable Architecture Search [9] obtains 1.26% accuracy drop with FLOPs pruning ratio of 43.5%. In comparison, our technique produces 0.4% accuracy improvement with BitOPs reduction ratio of roughly 96.8% for 4-bit ResNet50 on ImageNet (67.32B for 4-bit and 2.16T for floating point). Although it is not completely fair to compare quantization with model pruning due to different hardware implementations, this point highlights that network quantization can serve as a strong proxy for complexity-performance trade-offs.

Our method reveals that over-fitting caused by large weight scale in the last fully-con-

Table 2: Comparison of quantization techniques with only weights quantized.

| Quant. Method | Weights | MobileNet V1 Acc.-1 | MobileNet V1 Acc.-5 | MobileNet V2 Acc.-1 | MobileNet V2 Acc.-5 |
|---|---|---|---|---|---|
| Deep Compression [14] | 2bits | 37.62 | 64.31 | 58.07 | 81.24 |
| HAQ [49] | flexible | 57.14 | 81.87 | **66.75** | **87.32** |
| SAT (Ours) | 2bits | **66.3** | **86.8** | **66.8** | 87.2 |
| Deep Compression [14] | 3bits | 65.93 | 86.85 | 68.00 | 87.96 |
| HAQ [49] | flexible | 67.66 | 88.21 | 70.90 | 89.76 |
| SAT (Ours) | 3bits | **70.7** | **89.5** | **71.1** | **89.9** |
| Deep Compression [14] | 4bits | 71.14 | 89.84 | 71.24 | 89.93 |
| HAQ [49] | flexible | 71.74 | **90.36** | 71.47 | 90.23 |
| SAT (Ours) | 4bits | **72.1** | 90.2 | **72.1** | **90.6** |
| Deep Compression [14] | FP | 70.90 | 89.90 | 71.87 | 90.32 |
| HAQ [49] | FP | 70.90 | 89.90 | 71.87 | 90.32 |
| SAT (Ours) | FP | 71.7 | 90.2 | 71.8 | 90.2 |

nected layer is indeed the main reason for performance degeneration of network quantization. With proper scaling, the quantized models achieve comparable or even better performance than their full-precision counterparts. In this case, we have to rethink about the doctrine in the model quantization literature that quantization itself hampers the capacity of the model. It seems with mild quantization, the generated models do not sacrifice in capacity, but benefit from the quantization procedure. The clamping and rescaling technique does not contribute to the gain in quantized models since they are already used in full-precision training. One potential reason is that quantization acts as a favorable regularization during training and help the model to generalize better. The underlying mechanism is not clear yet. We left in-depth exploration as future work.

# 5 Conclusion

This paper studies the main reason for performance degeneration of quantized neural networks. By investigating the impact of clamping operation on weight scale and the learning curve of the full-precision model, we find that enlargement of weight scale in the last fully-connected layer will cause over-fitting issue, regardless of the weight/activation precision in the model. Based on this, we propose a scale-adjusted training technique to alleviate this problem. Our method yields state-of-the-art performance on quantized neural network for light models such as MobileNet V1/V2, and consistently better performance than the full-precision counterparts for MobileNet V1/V2 and ResNet18/50 under moderate bit-widths.

# References

[1] Madhu S Advani and Andrew M Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv preprint arXiv:1710.03667*, 2017.

Table 3: Comparison of quantization techniques on ResNet18 and ResNet50 with both weight and activation quantized.

| Bit-widths | Quant. Method† | ResNet18 | | ResNet50 | |
|---|---|---|---|---|---|
| | | Acc.-1 | Acc.-5 | Acc.-1 | Acc.-5 |
| 2bits | PACT [5] | 64.4 (-6.0) | 85.6 (-4.0) | 72.2 (-4.7) | 90.5 (-2.6) |
| | LQNet [57] | 64.9 (-5.4) | 85.9 (-3.6) | 71.5 (-3.9) | 90.3 (-2.9) |
| | QIL [20] | 65.7 (-4.5) | - | - | - |
| | LSQ [11] | 67.6 (**-2.9**) | 87.6 (**-2.0**) | 73.7 (-3.2) | 91.5 (-1.9) |
| | SAT (Ours) | 65.5 (-4.9) | 86.3 (-3.3) | 73.3 (**-2.6**) | 91.3 (**-1.4**) |
| 3bits | PACT [5] | 68.1 (-2.3) | 88.2 (-1.4) | 75.3 (-1.6) | 92.6 (-0.5) |
| | LQNet [57] | 68.2 (-2.1) | 87.9 (-1.6) | 74.2 (-2.2) | 91.6 (-1.6) |
| | QIL [20] | 69.2 (-1.0) | - | - | - |
| | LSQ [11] | 70.2 (**-0.3**) | 89.4 (**-0.1**) | 75.8 (-1.1) | 92.7 (-0.7) |
| | SAT (Ours) | 69.3 (-0.9) | 88.9 (-0.6) | 75.9 (**0.0**) | 92.7 (**0.0**) |
| 4bits | PACT [5] | 69.2 (-1.2) | 89.0 (-0.6) | 76.5 (-0.4) | 93.2 (**+0.1**) |
| | LQNet [57] | 69.3 (-1.0) | 88.8 (-0.7) | 75.1 (-1.3) | 92.4 (-0.8) |
| | QIL [20] | 70.1 (-0.1) | - | - | - |
| | LSQ [11] | 71.1 (**+0.6**) | 90.0 (**+0.4**) | 76.7 (-0.2) | 93.2 (-0.2) |
| | SAT (Ours) | 70.3 (+0.1) | 89.5 (0.0) | 76.3 (**+0.4**) | 92.8 (**+0.1**) |
| FP | PACT [5] | 70.4 | 89.6 | 76.9 | 93.1 |
| | LQNet [57] | 70.3 | 89.5 | 76.4 | 93.2 |
| | QIL [20] | 70.2 | - | - | - |
| | LSQ [11] | 70.5 | 89.6 | 76.9 | 93.4 |
| | SAT (Ours) | 70.2 | 89.5 | 75.9 | 92.7 |

* PACT and LSQ use full pre-activation ResNet, QIL and SAT use vanilla ResNet, and LQNet uses vanilla ResNet without convolution operation in shortcut (type-A shortcut).

† PACT, LQNet and QIL use full-precision for the first and last layers, while LSQ and SAT use 8bit for both layers .

[2] Yonathan Aflalo, Asaf Noy, Ming Lin, Itamar Friedman, and Lihi Zelnik. Knapsack pruning with inner distillation. *arXiv preprint arXiv:2002.08258*, 2020.

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

[4] Yamini Bansal, Madhu Advani, David D Cox, and Andrew M Saxe. Minnorm training: an algorithm for training over-parameterized deep neural networks. *arXiv preprint arXiv:1806.00730*, 2018.

[5] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

[6] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.

[7] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Xuanyi Dong and Yi Yang. Network pruning via transformable architecture search. In *Advances in Neural Information Processing Systems*, pages 760–771, 2019.

[10] Ahmed T. Elthakeb, Prannoy Pilligundla, FatemehSadat Mireshghallah, Amir Yazdanbakhsh, Sicun Gao, and Hadi Esmaeilzadeh. Releq: an automatic reinforcement learning approach for deep quantization of neural networks. *arXiv preprint arXiv:1811.01704*, 2018.

[11] Steven K. Esser, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S. Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

[12] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. *arXiv preprint arXiv:1908.05033*, 2019.

[13] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[14] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[17] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[18] Sambhav R Jain, Albert Gural, Michael Wu, and Chris H Dick. Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. *arXiv preprint arXiv:1903.08066*, 2(3):7, 2019.

[19] Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *arXiv preprint arXiv:1704.05519*, 2017.

[20] Sangil Jung, Changyong Son, Seohyung Lee, Jinwoo Son, Jae-Joon Han, Youngjun Kwak, Sung Ju Hwang, and Changkyu Choi. Learning to quantize deep networks by optimizing quantization intervals with task loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4350–4359, 2019.

[21] Jangho Kim, Yash Bhalgat, Jinwon Lee, Chirag Patel, and Nojun Kwak. Qkd: Quantization-aware knowledge distillation. *arXiv preprint arXiv:1911.12491*, 2019.

[22] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[23] Cong Leng, Zesheng Dou, Hao Li, Shenghuo Zhu, and Rong Jin. Extremely low bit neural network: Squeeze the last bit out with admm. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[24] Fengfu Li, Bo Zhang, and Bin Liu. Ternary weight networks. *arXiv preprint arXiv:1605.04711*, 2016.

[25] Yingwei Li, Zhuotun Zhu, Yuyin Zhou, Yingda Xia, Wei Shen, Elliot K. Fishman, and Alan L. Yuille. Volumetric medical image segmentation: A 3d deep coarse-to-fine framework and its adversarial examples. In *Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics*, pages 69–91, 2019.

[26] Yingwei Li, Xiaojie Jin, Jieru Mei, Xiaochen Lian, Linjie Yang, Cihang Xie, Qihang Yu, Yuyin Zhou, Song Bai, and Alan Yuille. Neural architecture search for lightweight non-local networks. In *CVPR*, 2020.

[27] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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

[29] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[31] Qian Lou, Lantao Liu, Minje Kim, and Lei Jiang. Autoqb: Automl for network quantization and binarization on mobile devices. *arXiv preprint arXiv:1902.05690*, 2019.

[32] Charles H Martin and Michael W Mahoney. Rethinking generalization requires revisiting old ideas: statistical mechanics approaches and complex learning behavior. *arXiv preprint arXiv:1710.09553*, 2017.

[33] Jieru Mei, Yingwei Li, Xiaochen Lian, Xiaojie Jin, Linjie Yang, Alan Yuille, and Jianchao Yang. Atom{nas}: Fine-grained end-to-end neural architecture search. In *International Conference on Learning Representations*, 2020.

[34] Naveen Mellempudi, Abhisek Kundu, Dheevatsa Mudigere, Dipankar Das, Bharat Kaul, and Pradeep Dubey. Ternary neural networks with fine-grained quantization. *arXiv preprint arXiv:1705.01462*, 2017.

[35] Xin Miao, Xiantong Zhen, Xianglong Liu, Cheng Deng, Vassilis Athitsos, and Heng Huang. Direct shape regression networks for end-to-end face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5040–5049, 2018.

[36] Xin Miao, Xin Yuan, Yunchen Pu, and Vassilis Athitsos. $\lambda$-net: Reconstruct hyperspectral images from a snapshot measurement. In *IEEE/CVF Conference on Computer Vision (ICCV)*, volume 1, 2019.

[37] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852*, 2017.

[38] Asit Mishra, Eriko Nurvitadhi, Jeffrey J Cook, and Debbie Marr. Wrpn: wide reduced-precision networks. *arXiv preprint arXiv:1709.01134*, 2017.

[39] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7: 19143–19165, 2019.

[40] Eunhyeok Park, Junwhan Ahn, and Sungjoo Yoo. Weighted-entropy-based quantization for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5456–5464, 2017.

[41] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv preprint arXiv:1903.10520*, 2019.

[42] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.

[43] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems*, pages 901–909, 2016.

[44] Hyunjune Sebastian Seung, Haim Sompolinsky, and Naftali Tishby. Statistical mechanics of learning from examples. *Physical review A*, 45(8):6056, 1992.

[45] Tao Sheng, Chen Feng, Shaojie Zhuo, Xiaopeng Zhang, Liang Shen, and Mickey Aleksic. A quantization-friendly separable convolution for mobilenets. In *2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2)*, pages 14–18. IEEE, 2018.

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

[47] Stefan Uhlich, Lukas Mauch, Kazuki Yoshiyama, Fabien Cardinaux, Javier Alonso Garcia, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Differentiable quantization of deep neural networks. *arXiv preprint arXiv:1905.11452*, 2019.

[48] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.

[49] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8612–8620, 2019.

[50] Timothy LH Watkin, Albrecht Rau, and Michael Biehl. The statistical mechanics of learning a rule. *Reviews of Modern Physics*, 65(2):499, 1993.

[51] Bichen Wu, Yanghan Wang, Peizhao Zhang, Yuandong Tian, Peter Vajda, and Kurt Keutzer. Mixed precision quantization of convnets via differentiable neural architecture search. *arXiv preprint arXiv:1812.00090*, 2018.

[52] Chen Xu, Jianqiang Yao, Zhouchen Lin, Wenwu Ou, Yuanbin Cao, Zhirong Wang, and Hongbin Zha. Alternating multi-bit quantization for recurrent neural networks. *arXiv preprint arXiv:1802.00150*, 2018.

[53] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13:55–75, 08 2018. doi: 10.1109/MCI.2018.2840738.

[54] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018.

[55] Qihang Yu, Yingwei Li, Jieru Mei, Yuyin Zhou, and Alan L Yuille. Cakes: Channel-wise automatic kernel shrinking for efficient 3d network. *arXiv preprint arXiv:2003.12798*, 2020.

[56] Lei Yue, Xin Miao, Pengbo Wang, Baochang Zhang, Xiantong Zhen, and Xianbin Cao. Attentional alignment networks. In *BMVC*, volume 2, page 7, 2018.

[57] Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 365–382, 2018.

[58] Shu-Chang Zhou, Yu-Zhi Wang, He Wen, Qin-Yao He, and Yu-Heng Zou. Balanced quantization: An effective and efficient approach to quantized neural networks. *Journal of Computer Science and Technology*, 32(4):667–682, 2017.

[59] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[60] Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. *arXiv preprint arXiv:1612.01064*, 2016.

[61] Shilin Zhu, Xin Dong, and Hao Su. Binary ensemble neural network: More bits per network or more networks per bit? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4923–4932, 2019.

[62] Bohan Zhuang, Jing Liu, Mingkui Tan, Lingqiao Liu, Ian Reid, and Chunhua Shen. Effective training of convolutional neural networks with low-bitwidth weights and activations. *arXiv preprint arXiv:1908.04680*, 2019.