Accurate and Efficient 2-bit Quantized Neural Networks

Jungwook Choi [1] **Swagath Venkataramani** [1] **Vijayalakshmi Srinivasan** [1] **Kailash Gopalakrishnan** [1]
Zhuo Wang [1 2] **Pierce Chuang** [1 3]

Abstract

Deep learning algorithms achieve high classification accuracy at the expense of significant computation cost. In order to reduce this cost, several quantization schemes have gained attention recently with some focusing on weight quantization, and others focusing on quantizing activations. This paper proposes novel techniques that individually target weight and activation quantizations resulting in an overall quantized neural network (QNN). Our activation quantization technique, PArameterized Clipping acTivation (PACT), uses an activation clipping parameter α that is optimized during training to find the right quantization scale. Our weight quantization scheme, statistics-aware weight binning (SAWB), finds the optimal scaling factor that minimizes the quantization error based on the statistical characteristics of weight distribution without the need for an exhaustive search. Furthermore, we provide an innovative insight for quantization in the presence of shortcut connections, which motivates the use of high-precision for the shortcuts. The combination of PACT and SAWB results in a 2-bit QNN that achieves state-of-the-art classification accuracy (comparable to full precision networks) across a range of popular models and datasets. Using a detailed hardware accelerator system performance model, we also demonstrate that relative to the more recently proposed Wide Residual Network (WRPN) approach to quantization, PACT-SAWB not only achieves iso-accuracy but also achieves 2.7$\sim$3.1× speedup.

1 Introduction

Deep Convolutional Neural Networks (CNNs) have achieved remarkable accuracy for tasks in a wide range of application domains including image processing (He et al., 2016b), machine translation (Gehring et al., 2017), and speech recognition (Zhang et al., 2017). These state-of-the-art CNNs use very deep models, consuming 100s of ExaOps of computation during training and GBytes of storage for model and data. This complexity poses a tremendous challenge for widespread deployment, especially in resource constrained edge environments - leading to a plethora of explorations in compressed models that minimize memory footprint and computation while preserving model accuracy as much as possible.

Recently, a whole host of different techniques have been proposed to alleviate these computational costs. Among them, reducing the bit-precision of key CNN data structures, namely weights and activations, has gained attention due to its potential for significant reduction in both storage requirements and computational complexity. Several weight quantization techniques (Li & Liu, 2016; Zhu et al., 2017; Jan Achterhold, 2018; Antonio Polino, 2018; Lu Hou, 2018) have been proposed to reduce bit-precision of CNN weights but ended up sacrificing model accuracy. Furthermore, a straightforward extension of weight quantization schemes to activation quantization was also proposed, but it incurred significant accuracy degradation in large-scale image classification tasks such as ImageNet (Russakovsky et al., 2015). Lately, activation quantization schemes based on greedy layer-wise optimization were proposed (Park et al., 2017; Zhou et al., 2017; Cai et al., 2017), but they required expensive processing (e.g., Lloyd's algorithm for activation quantization in (Cai et al., 2017)), and achieved limited accuracy improvement.

Complementary to quantization schemes, increasing the network size has been shown to compensate accuracy loss due to quantization. For example, (Asit Mishra, 2018b) and (McDonnell, 2018) employed Wide Residual Network (Zagoruyko & Komodakis, 2016) for weight and activation quantization and demonstrated that increasing the number of channels reduces the impact of quantization error and allows more aggressive bit-width reduction. However, increasing the network size leads to a quadratic increase in the number of operations, which in turn increases the classification latency.

This work is motivated by the desire to significantly im-

[1]IBM T. J. Watson Research Center, New York, USA. [2]Now at Google. [3]Now at Facebook. Correspondence to: Jungwook Choi <choij@us.ibm.com>.

Proceedings of the 2^{nd} SysML Conference, Palo Alto, CA, USA, 2019. Copyright 2019 by the author(s).

Table 1. Comparison of the state of the art neural net quantization schemes.

QNN Scheme	Training Complexity	Quantization at inference	Enlarge Network size
WEQ (Park et al., 2017)	Weight clustering (iterative)	Non-uniform	N
BQ (Zhou et al., 2016)	Histogram equalization (iterative)	Non-uniform	N
HWGQ (Cai et al., 2017)	Lloyd's algorithm (iterative)	Non-uniform	N
LQ-Nets (Zhang et al., 2018)	Tune quantizers via block coordinate descent (iterative)	Non-uniform	N
QIP (Jung et al., 2018)	Tune interval parameters via backprop (one-pass)	Semi-uniform	N
WRPN (Asit Mishra, 2018b)	No parameter needed	Uniform	Y
Ours (PACT-SAWB)	Compute scales or tune via backprop (one-pass)	Uniform	N

prove quantization schemes and achieve accuracy comparable to full-precision models while requiring no changes to the network structure - thereby harnessing the full computational benefits of quantization. We propose individual techniques targeting activation and weight quantization resulting in an overall quantized neural network (QNN). Our activation quantization technique, PArameterized Clipping acTivation (PACT), uses an activation clipping parameter α that is optimized during training to find the right quantization scale. Our weight quantization scheme, statistics-aware weight binning (SAWB), finds the optimal scaling factor that minimizes the quantization error based on the statistical characteristics of weight distribution without performing an exhaustive search. Furthermore, we provide an innovative insight for quantization in the presence of shortcut connections, which motivates the use of high-precision for the shortcuts. As a result, We realize 2-bit QNNs with PACT-SAWB, which achieve the state-of-the-art classification accuracy comparable to full precision networks while incurring no larger than $\mathcal{O}(n)$ computational overhead. Using a detailed hardware accelerator system performance model, we demonstrate that relative to the WRPN approach for quantization (Asit Mishra, 2018b), PACT-SAWB not only achieves iso-accuracy but also achieves 2.7∼3.1× speedup.

The rest of the paper is organized as follows: Section 2 provides a summary of prior work on QNNs and challenges. We present a novel activation quantization scheme in Section 3 followed by a new weight quantization scheme in Section 4 and an in-depth discussion on ResNet quantization in Section 5. In Section 6, we demonstrate the effectiveness of our quantization schemes using a 2-bit QNN across a set of popular CNN models and datasets. Section 7 describes the hardware accelerator system-level performance model, and presents the utilization and speedup achieved by PACT-SAWB.

2 Prior Work in QNN

There has been extensive research on quantizing weight and activation to minimize CNN computation and storage costs. One of the earliest studies in weight quantization schemes (Hwang & Sung, 2014; Courbariaux et al., 2015) show that it is indeed possible to quantize weights to 1-bit (binary) or 2-bits (ternary), allowing an entire DNN model to fit in resource-constrained platforms (e.g., mobile devices). Effectiveness of weight quantization techniques has been further improved by ternarizing weight using its statistics (Li & Liu, 2016) or by tuning quantization scales during training (Zhu et al., 2017). To further reduce the cost of computing and storing activations, prior work (Kim & Smaragdis, 2015; Hubara et al., 2016a; Rastegari et al., 2016) proposed the use of fully binarized neural networks where activations are also quantized to 1-bit. More recently, activation quantization schemes using more general selections in bit-precision (Hubara et al., 2016b; Zhou et al., 2016) have been studied.

However, these early techniques on weight and activation quantization show significant degradation in accuracy for ImageNet tasks (Russakovsky et al., 2015) when bit precision is reduced significantly ($\leq$ 2-bits). To improve QNN accuracy for ImageNet, more complex quantization schemes have been adopted. Weighted-entropy based quantization (WEQ, Park et al. 2017) used iterative search algorithms for finding weight clusters or base/offset of logarithmic quantization for activation. Balanced quantization (BQ, Zhou et al. 2017) used recursive partitioning for histogram equalization. Half-wave Gaussian quantization (HWGQ, Cai et al. 2017) found the quantization scale via Lloyd optimization on Normal distribution. More recently, quantization schemes were learned in the context of training. LQ-Nets (Zhang et al., 2018) proposed trainable quantizers that can be updated via block-coordinate descent in the forward pass. Joint training of quantization interval parameters (QIP, Jung et al. 2018) parameterized the quantization formula with trainable parameters tuned during back-propagation. These schemes demonstrated state-of-the-art QNN accuracy for neural networks for ImageNet, such as ResNet, but they often involve computationally expensive iterative algorithms during the training. Furthermore, they require non-uniform quantization levels during inference, which is a challenge for efficient hardware implementation.

Alternative approaches for QNN is to use simple quantization schemes but reduce impact of quantization error by performing more computations. One such approach is to increase the size of the network; (Zagoruyko & Komodakis, 2016) demonstrated that accuracy of a neural network can be improved by increasing the number of channels (called widening). Wide reduced-precision network (WRPN, Asit Mishra 2018b) exploited this increased accuracy for QNN, by doubling the channel size to compensate for quantization error. (McDonnell, 2018) also employed the idea of widening along with other modification in the training setting (e.g., warm-restart learning rate schedule) to improve accuracy for 1-bit weight quantization. Another way of taking advantage of the extra computation for QNN is knowledge distillation (e.g., Apprentice Asit Mishra 2018a), where a teacher network (which is typically large and trained in full precision) is employed to help train the student network (one that is quantized). In all of these approaches accuracy improvement for QNN comes at the expense of large computational cost.

Table 1 summarizes characteristics of these QNN schemes. Most of the state of the art schemes involve slow iterative algorithms such as clustering or Lloyd's algorithm, hindering adoption within the model training process. In addition, non-uniform quantizations often require additional data comparison operations in the hardware processing engines, adding significant area and power overhead. Although WRPN uses a simple quantization scheme, it requires larger number of computations for a network, thereby increasing the inference latency.

In summary, prior quantization techniques either incur noticeable degradation in accuracy relative to full-precision, or significantly increase computational complexity for training and/or inference to overcome quantization errors. In this work we set out to explore quantization schemes for both weights and activations to achieve accuracy comparable to full-precision models while maintaining quantization computation simple.

3 PARAMETERIZED CLIPPING ACTIVATION

3.1 Challenge in Activation Quantization

Activation quantization becomes challenging when ReLU (the most commonly used activation function in CNNs) is used as the layer activation function. ReLU allows gradient of activations to propagate through deep layers and therefore achieves superior accuracy compared to other activation functions (Nair & Hinton, 2010). However, as the output of the ReLU function is unbounded, the quantization after ReLU requires a high dynamic range (i.e., more bit-precision). This is particularly problematic when the target bit-precision is low, e.g., 2-bits. In Figure 1, we present the training and validation errors of ResNet20 with ReLU on the CIFAR10 dataset; its accuracy is significantly degraded when activation after ReLU is quantized into 2-bits.

It has been shown that this dynamic range issue can be alleviated by using a clipping activation function, which places an upper-bound on the magnitude of output activation (Hubara et al., 2016b; Zhou et al., 2016). As shown in Figure 1, the network with the clipping activation function shows lower training/validation errors than the ReLU case when activation is quantized. But, there is still noticeable accuracy degradation compared to the full-precision baseline. In addition, due to different characteristics of activation across the layers and models, it is difficult to determine a globally optimal clipping level. As an example, (Cai et al., 2017) used Lloyd optimization to find a proper scale for activation quantization, but its computation is expensive without approximation.

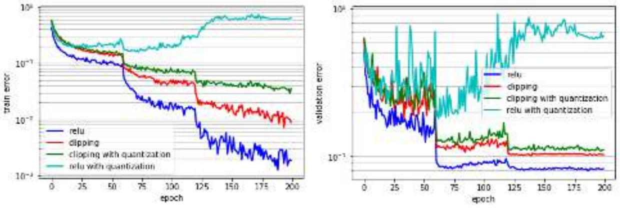

Figure 1. Train error (left) and validation error (right) when the activation of CIFAR10 ResNet10 with ReLU or clipping activation function (clipping level = 1.0) is quantized to 2-bits. ReLU is more sensitive to activation quantization due to its large dynamic range.

3.2 Parameterized Clipping Activation Function

Building on these insights, we introduce PArameterized Clipping acTivation Function (PACT), a new activation quantization scheme in which the activation function has a parameterized clipping level, α. This clipping level is dynamically adjusted via stochastic gradient descent (SGD)-based training with the goal of minimizing the accuracy degradation due to quantization error. In PACT, the conventional ReLU activation function in CNN is replaced with the following:

$$y = 0.5(|x| - |x-\alpha| + \alpha) = \begin{cases} 0, & x \in (-\infty, 0) \\ x, & x \in [0, \alpha) \\ \alpha, & x \in [\alpha, +\infty) \end{cases} \quad (1)$$

where α limits the dynamic range of activation to $[0, \alpha]$. This is illustrated in Figure 2(a). The truncated activation output is then linearly quantized to k-bits for the dot-product computations:

$$y_q = \text{round}(y \cdot \frac{2^k - 1}{\alpha}) \cdot \frac{\alpha}{2^k - 1} \quad (2)$$

With this new activation function, α is a variable in the loss function, whose value can be optimized during training. For back-propagation, gradient $\frac{\partial y_q}{\partial \alpha}$ can be computed using the Straight-Through Estimator (STE) (Bengio et al., 2013) to estimate $\frac{\partial y_q}{\partial y}$ as 1. Thus,

$$\frac{\partial y_q}{\partial \alpha} = \frac{\partial y_q}{\partial y}\frac{\partial y}{\partial \alpha} = \begin{cases} 0, & x \in (-\infty, \alpha) \\ 1, & x \in [\alpha, +\infty) \end{cases} \tag{3}$$

The larger the α, the more the parameterized clipping function resembles ReLU. To avoid large quantization errors due to a wide dynamic range, we include a L2-regularizer for α in the loss function. Figure 2(b) illustrates how the value of α changes during full-precision training of CIFAR10 ResNet20 starting with an initial value of 10 and using the L2-regularizer. It can be observed that α converges to the values much smaller than the initial value after epochs of training, thereby limiting the dynamic range of activations and reducing the quantization error. We empirically found that α per layer was easier to train than α per-channel.

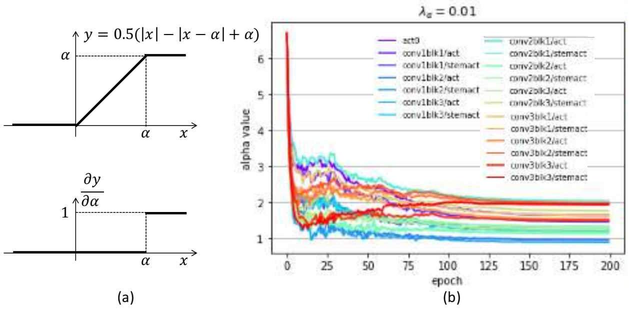

Figure 2. (a) PACT activation function and its gradient. The dynamic range of activation after PACT is bounded by α, thus it is more robust to quantization. (b) Evolution of the trainable clipping parameter α during training of CIFAR10 ResNet20.

3.3 Analysis

3.3.1 *PACT is as Expressive as ReLU*

When used as an activation function of the neural network, PACT is as expressive as ReLU. This is because the clipping parameter introduced in PACT, α, allows flexibility in adjusting the dynamic range of activation for each layer, thus it can cover large dynamic range as needed. We demonstrate in the simple example below that PACT can reach the same solution as ReLU via SGD.

Lemma 3.1. *Consider a single-neuron network with PACT; $x = w \cdot a$, $y = \mathrm{PACT}(x)$, where a is input and w is weight. This network can be trained with SGD to find the output the network with ReLU would achieve.*

Proof. Consider a sample of training data (a, y^*). For the purpose of illustration, consider mean-square-error (MSE) as the cost function: $L = 0.5 \cdot (y^* - y)^2$.

If $x \leq \alpha$, then clearly the network with PACT behaves the same as the network with ReLU.

If $x > \alpha$, then $y = \alpha$ and $\frac{\partial y}{\partial \alpha} = 1$ from (1). Thus,

$$\frac{\partial L}{\partial \alpha} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial \alpha} = \frac{\partial L}{\partial y} \tag{4}$$

Therefore, when α is updated by SGD,

$$\alpha_{new} = \alpha - \eta\frac{\partial L}{\partial \alpha} = \alpha - \eta\frac{\partial L}{\partial y} \tag{5}$$

where η is a learning rate. Note that during this update, the weight is not updated as $\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial x}(= 0) \cdot a = 0$.

From the MSE cost function, $\frac{\partial L}{\partial y} = (y - y^*)$. Therefore, if $y^* > x$, α is increased for each update of (5) until $\alpha \geq x$, then the PACT network behaves the same as the ReLU network.

Interestingly, if $y^* \leq y$ or $y < y^* < x$, α is decreased or increased to converge to y^*. Note that in this case, ReLU would pass erroneous output x to increase cost function, which needs to be fixed by updating w with $\frac{\partial L}{\partial w}$. PACT, on the other hand, ignores this erroneous output by directly adapting the dynamic range to match the target output y^*. In this way, the PACT network can be trained to produce output which converges to the same target that the ReLU network would achieve via SGD. □

In general, $\frac{\partial L}{\partial \alpha} = \sum_i \frac{\partial L}{\partial y_i}$, and PACT considers all the output neurons together to change the dynamic range. There are two options: (1) if output x_i is not clipped, then the network is trained via back-propagation of gradient to update weight, (2) if output x_i is clipped, then α is increased or decreased based on how close the overall output is to the target. Hence, there exist configurations under which SGD leads to a solution that the network with ReLU would achieve. Figure 3 demonstrates that CIFAR10 ResNet20 with PACT converges almost identical to the network with ReLU.

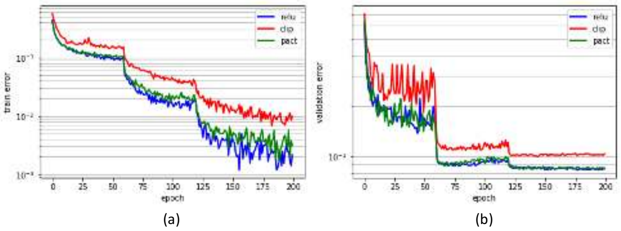

Figure 3. (a) Training error and (b) validation error of PACT for CIFAR10 ResNet20. Note that the convergence curve of PACT closely follow ReLU.

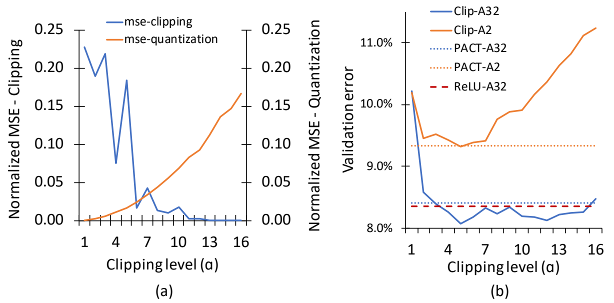

Figure 4. Experiments on CIFAR10 ResNet20 to validate that PACT balances clipping and quantization errors. (a) Trade-off between clipping and quantization error. (b) PACT achieving the lowest validation error that clipping activations can achieve without an exhaustive search over the clipping levels α.

3.3.2 Balancing Clipping and Quantization Errors

In Section 3.1, we discussed that there is a trade-off between errors due to clipping and quantization. A high clipping level α allows large dynamic range and decreases the clipping error, i.e., $ErrClip_i = max(x_i - \alpha, 0)$. However, it also increases the magnitude of the quantization error, i.e., $ErrQuant_i \leq 0.5 \cdot \frac{\alpha}{2^k - 1}$, with k-bit quantization.

The trade-off between clipping and quantization error can be better observed in Figure 4(a) where the normalized mean-square-error from clipping and 2-bit quantization during training of the CIFAR10 ResNet20 is shown for the different clipping levels. This figure explains why the network with ReLU and the clipping activation function fail to converge when the activation is quantized; ReLU suffers large quantization error, whereas the clipping activation function incurs significant clipping error. This imposes a burden of finding a proper clipping level to balance clipping and quantization errors.

PACT can effectively find a balancing point between clipping and quantization errors. As explained in Section 3.3.1, PACT adjusts the dynamic range based on how close the output is to the target. As both clipping and quantization errors distort the output away from the target, PACT would increase or decrease the dynamic range during training to minimize both clipping and quantization errors. Figure 4(b) shows how PACT balances the clipping and quantization errors for QNN. CIFAR10 ResNet20 is trained with a varying clipping level α from 1 to 16 for the clipping activation function. When activation is quantized to 2-bit, the network trained with the clipping activation function shows significant accuracy degradation as α increases. This is consistent with the trend in quantization error we observed in Figure 4(a). In this case, PACT achieves the best accuracy among all the clipping levels, but without exhaustively sweeping over α. In other words, PACT auto-tunes the clipping level to achieve the best accuracy without incurring significant computational overhead. The auto-tuning of the dynamic range in PACT is critical towards efficient yet robust training of large scale quantized neural networks, especially because it does not increase the burden for hyper-parameter tuning. In fact, we've used the same hyper-parameters as the original network structure for all the models we tested, except for replacing ReLU with PACT when we applied activation quantization.

Without quantization, there is a trend that validation error is small when α is not large or small. Surprisingly, some of the cases even outperforms the ReLU network. PACT also achieves comparable accuracy to ReLU, confirming its expressivity discussed in Section 3.3.1.

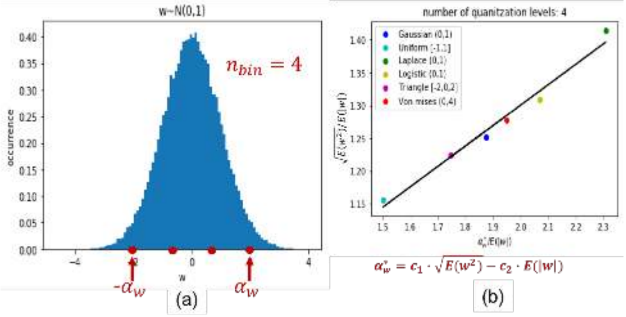

Figure 5. Statistics aware weight binning (SAWB). (a) Quantization levels determined by the scale α. (b) Linear regression of the optimal scales from 6 distinct distributions shows that the optimal scale is well characterized by the first and second moments ($E(|w|)$ and $E(w^2)$) of the weight distribution.

4 STATISTICS-AWARE WEIGHT BINNING

In addition to activation quantization, we also propose a novel weight quantization scheme, statistics-aware weight binning (SAWB). The main idea is to exploit the statistics of the weight distribution (i.e., the first and the second moments) when the quantization scale is determined, so that the dispersion of weight is better captured in the course of training. Figure 5(a) illustrates a 2-bit (i.e., $n_{bin} = 4$) evenly-spaced SAWB scheme. We choose symmetric and uniformly spaced quantization bins because it enables a hardware friendly multiplication and accumulation (MAC) design. All the values in weight are quantized to the nearest bins, and the quantization scale α_w defines their largest quantization level. Given a weight distribution, there exists an optimal scale α_w^* such that the quantization error is minimized. That is,

$$\alpha_w^* = \arg\min_{\alpha_w} ||w - w_q||^2 \quad (6)$$

This is a canonical problem that numerous prior weight quantization schemes try to solve. Since exhaustive search over α_w is often not tractable, a popular approach is to assume a type of distribution for weight and come up with an heuristics. For example, (Li & Liu, 2016) conducted a case analysis with Normal distribution motivated by the fact that weights are often initialized with them, then provided a formula that estimates α_w as a function of $E(|w|)$.

However, it is often observed that the shape of the weight distribution changes over the set of back-propagation passes during training and becomes different from Gaussian. This invalidates the base assumption of the prior work, which in turn increases quantization errors. To overcome this limitation, we consider not only $E(|w|)$ but also $E(w^2)$. Intuitively, the 2nd moment $E(w^2)$ captures the overall shape of distribution while the 1st moment $E(|w|)$ indicates the representative value. Therefore, by employing both the 1st and the 2nd moments, the weight distribution can be better characterized throughout the training.

From this intuition, we empirically derive a simple formula that efficiently finds the optimal scale α_w^* for a larger variety of distributions. We first investigate the relationship between the optimal scale and the 1st/2nd moments over the six well-known distributions: Gaussian, uniform, Laplace, logistic, triangle, and von Mises distributions. In Figure 5(b), we plot $\frac{\alpha_w^*}{E(|w|)}$ (x axis) versus $\frac{\sqrt{E(w^2)}}{E(|w|)}$ (y axis) for the six distributions (points in different colors). Each point represents the optimal scale for that distribution obtained by a sweep over α_w given the finite quantization levels (e.g., 4 bins). We observe that the regression line closely fits all the 6 points, thereby, validating our intuition that the optimal scale α_w^* can be mostly characterized by $E(|w|)$ and $E(w^2)$. In fact, similar to the *coefficient of variation*, the term $\frac{\sqrt{E(w^2)}}{E(|w|)}$ measures dispersion of the weight distribution; the higher the variability in w, the larger the optimal scale α_w^*. Thus, compared to the prior work, which mostly considers only $E(|w|)$, our technique with both $E(|w|)$ and $E(w^2)$ can capture the change in the shape of weight more faithfully.

From the empirical study, we found that the above relationship between the optimal scale and the 1st/2nd moments holds even for a different number of quantization levels (n_{bin}=2, 4, 8, 16). In other words, given a number of bins n_{bin}, we can find a linear regression that derives α_w^* as a function of $\sqrt{E(w^2)}$ and $E(|w|)$:

$$\alpha_w^* = c_1 * \sqrt{E(w^2)} - c_2 * E(|w|) \tag{7}$$

where the coefficients c_1 and c_2 are predetermined from the linear regression over the 6 distributions as discussed above (e.g., $c_1 = 2.587$ and $c_2 = 1.693$ when $n_{bin} = 4$).

As a result, for each mini-batch of training, SAWB first computes $E(|w|)$ and $E(w^2)$ from the full-precision copy of weight, then uses (7) to compute α_w^* per layer for quantization. The quantized weight w_q is used for both forward/backward passes to obtain the weight gradient, which is used to update the full-precision copy w for the next round. Note that computation of $E(|w|)$ and $E(w^2)$ does not incur iterative algorithms, so it is computationally cheap (i.e., $\mathcal{O}(n)$ complexity, where n is the number of weight values).

Table 2. Square Error (SE) of the optimal and SAWB estimated scaling factor of Layer 11 of CIFAR10 ResNet20 at different epoch with 2-bit weight quantization

Epoch	1	40	80	120	160	200
Optimal SE	6.77	12.21	10.45	8.24	7.92	7.75
SAWB SE	6.94	12.43	10.64	8.40	8.13	7.98

To evaluate the effectiveness of SAWB, we compared the quantization errors when using the optimal scale (obtained via exhaustive search) *vs.* one from SAWB. Table 2 summarizes the quantization error (in terms of square error, SE) of the weight in layer 11 of the 2-bit quantized CIFAR10 ResNet. It is evident that α_w^* determined by SAWB incurs less than 3% additional SE at anytime during the training.

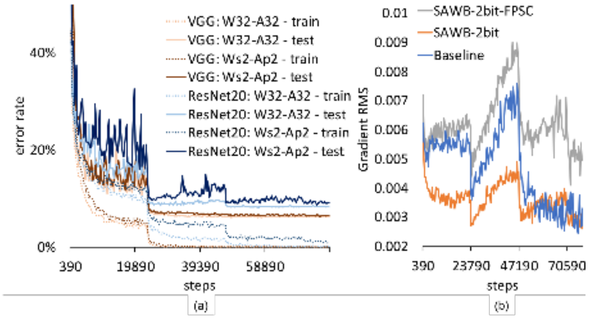

Figure 6. (a) Convergence plots for VGG and ResNet with and without quantization. While VGG is less sensitive to the quantization, ResNet suffers slower convergence when its shortcuts are quantized. (b) Impact of shortcut quantization visualized via root-mean-square (RMS) of weight gradient from the first shortcut of CIFAR10 ResNet. Note that SAWB-2bit suffers low gradient magnitude, implying slow progress in optimization.

5 Quantization in the Presence of Shortcut Connections

Throughout our quantization exploration, we observed that the feed-forward neural networks such as AlexNet and VGG behave more reliably compared to the networks with shortcut connections (e.g., ResNet, (He et al., 2016b)). Figure 6(a) demonstrates this issue with two training cases, VGG and ResNet20 models on CIFAR10, with and without 2-bit weight/activation quantization. As shown, VGG's training curves for full precision and the 2-bit network track closely, leading to almost identical test errors. On the other hand,

Table 3. Validation error of CIFAR10 ResNet20/32/44/56 for quantizing activation only, weight only, and both weight and activation. (p,s) indicates (PACT,SAWB), respectively. "fpsc" indicates full-precision shortcut. PACT-SAWB achieves accuracy $\leq 0.5\%$ for individual quantization, and $\leq 1\%$ for quantizing both weight and activation, compared to full precision baseline.

Layers	20	32	44	56
Full-Precision (32-bit)	8.49%	7.49%	6.84%	6.77%
W32-Ap2	9.51%	8.44%	8.02%	7.76%
W32-Ap2-fpsc	8.64%	7.72%	7.29%	7.01%
Ws2-A32	9.27%	8.80%	7.84%	7.45%
Ws2-A32-fpsc	9.08%	7.74%	7.39%	7.07%
Ws2-Ap2	10.77%	9.57%	9.39%	8.76%
Ws2-Ap2-fpsc	9.35%	8.36%	7.61%	7.48%

we can observe from the ResNet20 case that the training error for the 2-bit network decreases much slowly, indicating disruption of learning in the SGD optimization.

We made a key observation that naively applying quantization to the 1x1 convolution in shortcut connections negatively impact accuracy, because the shortcuts play an important role of flowing back-propagated gradient information across the layers but the quantization fundamentally would limit the flow of gradients. This phenomenon can be evidenced by measuring the magnitude of gradients. In Figure 6(b), we plot root-mean-square (RMS) values of gradients in the convolution in the first shortcut of CIFAR10 ResNet for the three cases: baseline, SAWB-2bit with and without the full-precision shortcut (FPSC). As shown in the figure, the magnitude of gradients is significantly small when the shortcut is quantized. This can obstruct the sgd optimizer from finding good local minima, slowing down the overall progress in optimization.

It is also mentioned in (He et al., 2016a) that keeping the shortcut connections more direct is beneficial to ease the optimization and reduce over-fitting. In the same vein, we explored an option of not quantizing the input activations or the weights in the shortcut paths. As will be discussed in depth in Section 6, our quantization with the full-precision shortcuts achieves accuracy close to the full-precision baseline for all the tested quantized ResNets. Note that the shortcut in ResNet takes a negligible portion of the overall compute, e.g., <1% for ImageNet-ResNet18.

6 Experiments

We implemented PACT and SAWB in Tensorflow (Abadi et al., 2015) using Tensorpack (Zhou et al., 2016) and studied several well-known CNNs: ResNet20/32/44/56 (He et al., 2016b) for the CIFAR10 dataset, and AlexNet and ResNet18/50 models for the ImageNet dataset. For comprehensive comparison with the state-of-the-art, we include most of the recent 2-bit QNNs that used the same experimental settings (including ternary quantizations that effectively use 2-bits), and compare the accuracy numbers from their papers: LQ-Nets (Zhang et al., 2018), TWN (Li & Liu, 2016), TTQ (Zhu et al., 2017), and QIP (Jung et al., 2018), BalancedQ (Zhou et al., 2017), WEQ (Park et al., 2017), WRPN (Asit Mishra, 2018b), and LearningReg (Choi et al., 2018). In case of ResNet50, for more exhaustive comparison, we further include notable QNN implementations that use more than 2-bits: Apprentice (Asit Mishra, 2018a), and UNIQ (Baskin et al., 2018). We also implemented DoReFa-Net (Zhou et al., 2016) using the same setting as PACT-SAWB (i.e., they share the same baseline).

Table 4. CIFAR10: Comparison on test accuracy (%) and degradation for different bit-precision settings.

Name	Baseline	Quantized	Degradation
< ResNet20: W2-A32 >			
SAWB-fpsc	**91.8**	**91.6**	**0.2**
DoReFa	91.8	90.9	1.0
LQ-Nets	92.1	91.8	0.3
TWN	91.8	90.9	0.9
TTQ	91.8	91.1	0.6
< ResNet20: W32-A2 >			
PACT-fpsc	**91.5**	**91.4**	**0.2**
DoReFa	91.5	90.1	1.4
ReLU6	91.5	91.0	0.5
< ResNet20: W2-A2 >			
PACT-SAWB-fpsc	**91.5**	**90.8**	**0.7**
DoReFa	91.5	88.2	3.3
LQ-Nets	92.1	90.2	1.9
< VGG: W2-A2 >			
PACT-SAWB-fpsc	**93.8**	**93.8**	**0.1**
LQ-Nets	93.8	93.5	0.3
QIP (lambda=0.5)	94.1	93.9	0.2

Detail information about our CNN implementation as well as our training settings can be found in Appendix A. Unless mentioned otherwise, ReLU following BatchNorm is used for activation function of the convolution (CONV) and fully-connected (FC) layers except the last FC using Softmax. Note that the same hyper-parameters from these full-precision CNN baselines are used for all the QNN experiments. For PACT, we only replace ReLU with PACT. The networks are trained from scratch and the first/last layers are not quantized, following the common practice. (Zhou et al., 2016; Asit Mishra, 2018b).

6.1 CIFAR10 Experiments

We first evaluate our activation and weight quantization schemes using CIFAR10 ResNet with (20,32,44,56) layers. Table 3 summarizes the accuracy for quantizing activation only, weight only, and both weight and activation. Individ-

Table 5. ImageNet: Comparison on top-1 test accuracy (%) and degradation. 2-bit QNNs except for Apprentice and UNIQ.

Name	Baseline	Quantized	Degradation
	< AlexNet >		
PACT-SAWB-fpsc	**58.3**	**57.2**	**1.1**
LQ-NETs	61.8	57.4	4.4
QIP (lambda=0.0)	58.1	55.7	2.4
DoReFa-Net	55.1	53.6	1.5
HWGQ	58.5	52.7	5.8
BalancedQ	57.1	55.7	1.4
WEQ	57.1	50.6	6.5
WRPN-x1	57.2	51.3	5.9
WRPN-x2	60.5	55.8	4.7
WRPN-x2, W2-A4	60.5	57.2	3.3
LearningReg	58.0	54.1	3.9
	< ResNet18 >		
PACT-SAWB-fpsc	**70.4**	**67.0**	**3.4**
LQ-NETs	70.3	64.9	5.4
QIP (lambda=0.0)	69.2	65.4	3.8
DoReFa	70.2	62.6	7.6
HWGQ	67.3	59.6	7.7
BalancedQ	68.2	59.4	8.8
LearningReg	68.1	61.7	6.4
	< ResNet50 >		
PACT-SAWB-fpsc	**76.9**	**74.2**	**2.7**
LQ-NETs	76.4	71.5	4.9
Apprentice (W2-A8)	76.2	71.5	3.4
UNIQ (W4-A8)	76.0	73.4	2.6

ual quantization incurs only marginal accuracy degradation, and the full-precision shortcut (post-fix "-fpsc") further improves the accuracy to be within 0.5% of the full-precision accuracy. Putting it all together, PACT-SAWB with the full-precision shortcut achieves accuracy within 1% of the full-precision accuracy across all the variants of CIFAR10 ResNets.

Next, we compare accuracy results of PACT-SAWB on CIFAR10 with the other state-of-the-art QNN schemes. To present a larger coverage, we compare results from both ResNet and VGG. As shown in Table 4, PACT and SAWB achieve accuracy among the highest and accuracy degradation among the lowest. In particular, SAWB-fpsc and PACT-SAWB-fpsc achieve slightly lower accuracy than LQ-Nets and QIP, respectively, but has lower accuracy degradation compared to them.

6.2 ImageNet Experiments

Next, we evaluate accuracy of our 2-bit QNN schemes on ImageNet dataset. As can be seen in Table 5, PACT-SAWB achieves the highest absolute accuracy as well as the lowest accuracy degradation for 2-bit QNN on all three networks (AlexNet, ResNet18 and ResNet50). In fact, to the best of our knowledge, PACT-SAWB's ResNet accuracies (67.0% and 74.2% for ResNet18 and ResNet50) are the highest ever reported. The absolute ResNet50 accuracy is even higher than the accuracy results obtained with higher bit-width in Apprentice and UNIQ. This first demonstration of 2-bit ResNet50 showing less than 3% accuracy degradation is a significant achievement with great potential for practical use.

Table 6. Impact of shortcut precision for CIFAR10 ResNet20.

Shortcut precision	16-bit	8-bit	4-bit	2-bit
Validation error (%)	9.28	9.31	9.40	10.7

WRPN-x2 W2-A4 achieves accuracy on-par with our 2-bit QNN, since it can regain the loss in accuracy by increasing the network size (i.e., doubling output channel length). But the enlarged network size can forfeit the system-level performance gain achieved by the bit-width reduction. The impact of reduced precision inference on the system performance will be investigated deeper in Section 7.

6.3 Impact of Shortcut Precision

Table 6 shows the validation error for various shortcut precision in CIFAR10 ResNet20 (the rest of the layers except the first/last layers are quantized into 2-bit). It is very interesting to see that the accuracy degradation is negligible if 8-bit or larger bit-precision is assigned to the shortcuts. This implies that the negative impact from quantizing 1x1 CONV in shortcuts can be greatly relieved without keeping their precision to 32-bits. It opens up a potential for reducing the bit-precision for the sensitive layers of the neural nets such as the first/last layers as well as the shortcut to gain better system-level performance.

7 System-Level Performance Gain

In this section, we demonstrate the gain in system performance as a result of bit-precision reduction from PACT-SAWB.

7.1 System Setup

We consider a DNN accelerator system comprising of a DNN accelerator chip, comprising of multiple cores, interfaced with an external memory. Similar to the state-of-the-art accelerator architectures such as (Fleischer et al., 2018), each core consists of a 2D-systolic array of fixed-point multiply-and-accumulate (MAC) processing elements on which DNN layers are executed. Each core also contains an on-chip memory, which stores the operands that are fed into the processing array. Figure 7(a) describes the architecture template of this system.

To estimate system performance at different bit-precision, we studied different configurations of the DNN accelerator each comprising the same amount of on-chip memory, ex-

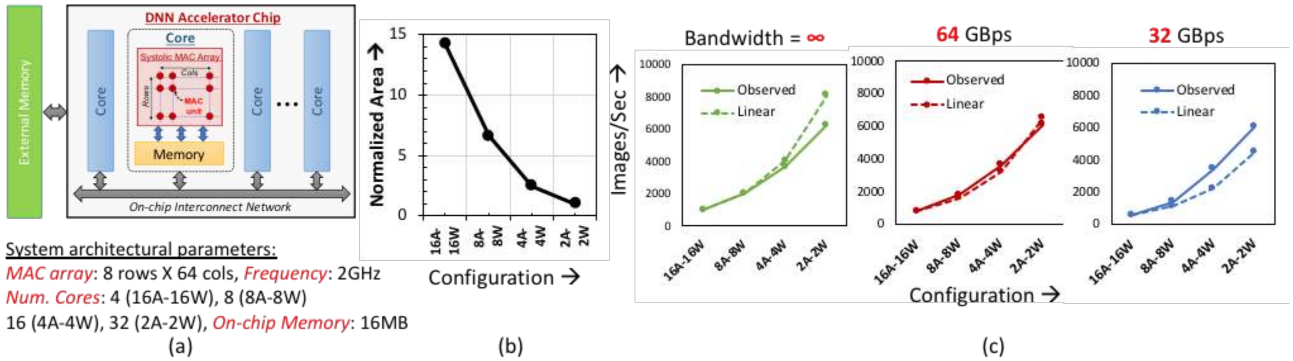

Figure 7. (a) System architecture and parameters. (b) Variation in MAC area with bit-precision. (c) Speedup at different quantizations for inference using ResNet50 model.

ternal memory bandwidth, and occupying iso-silicon area. First, we implemented a precision-configurable MAC unit using a high-end commercial technology (14nm CMOS) and measured the MAC area for the scaled bit-precision. Note that MAC design for PACT-SAWB is simple with little control overhead since it employs uniform quantization. As shown in Figure 7(b), we achieved ∼14× improvement in density when the bit-precision of both activations and weights are uniformly reduced from 16 to 2-bits.

Next, to translate the reduction in area to improvement in overall performance, we built an accelerator system with a varying number of cores whose bit-precision for MAC can be modulated. The peak compute capability (a number of operations, OPs) of the system is varied such that we achieve iso-area at each precision. Using the 2d-array size of 8x64 MACs, we can investigate the system with a large range of peak performance from 8 to 64 TOPs (corresponding to 16 to 2-bit MAC), covering the peak OPs of the off-the-shelf accelerator platforms such as Xilinx Alveo U200 (= 18.6 TOPs, 8-bits). Note that the total on-chip memory and external bandwidth remains constant at all precisions. We estimate the overall system performance using DeepMatrix (Venkataramani et al.), a framework that systematically characterizes and analyzes the performance of DNNs on a shared memory accelerator via exploration of the design space configurations (e.g., number of cores, dataflow, memory bandwidth) defined on an architectural template similar to Fig 7(a).

In the following section, we investigate two variants of system performance evaluations, first to assess impact of bit-precision reduction on system performance, and second to compare the system performance gain with WRPN. In the first evaluation, we use 8-bit for the high-precision computation (e.g., first/last layers) since we empirically observed no accuracy loss with this setting. As discussed in Section 6.3, we chose 8-bit for the high-precision computation to evaluate the most bit-optimized scenario. In the second evaluation, on the other hand, we chose to have a fair comparison by conservatively using 16-bit for the high-precision computation for both ours and WRPN.

7.2 Gain in Inference Performance

Figure 7(c) shows the gain in inference performance for ResNet50. We study the performance improvement using different external memory bandwidths, namely, a bandwidth unconstrained system (infinite memory bandwidth) and two bandwidth constrained systems at 32 and 64 GBps. In the bandwidth unconstrained scenario, the performance gain is limited by how amenable it is to parallelize the work using all the MAC units available in the system. In this case, we see a near-linear increase in performance for up-to 4 bits and a small drop at extreme quantization levels (2 bits).

Practical systems, whose bandwidths are constrained, (surprisingly) exhibit a super-linear growth in performance with quantization. For example, when external bandwidth is limited to 32 GBps, quantizing from 16 to 4 bits leads to a 4× increase in peak OPs but a 4.5× improvement in performance. This is because, the total amount of on-chip memory remains constant, and at very low precision some of the data-structures begin to fit within the memory present in the cores, thereby *avoiding* data transfers from the external memory. Consequently, in bandwidth limited systems, reducing the amount of data transferred from off-chip can provide an additional boost in system performance beyond the increase in peak OPs. Note that for the 4 and 2 bit precision configurations, we still used 8 bit precision to execute the first and last layers of the DNN. If we are able to quantize the first and last layers as well to 4 or 2 bits, we estimate an additional 1.24× improvement in performance, motivating the need to explore ways to aggressively quantize the first/last layers.

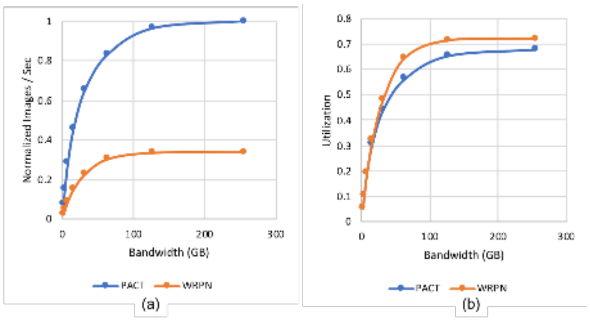

Figure 8. Comparison of system performance between PACT and WRPN. (a) Normalized images/sec, (b) Hardware utilization.

7.3 PACT-SAWB *vs.* WRPN

In PACT-SAWB, we compensate the impact of quantization error by incorporating it during the training process. An orthogonal approach, adopted in WRPN (Asit Mishra, 2018b), increases the number of channels in the layers, which would add redundancy to the data-structures making them more amenable to quantization. In this section, we analyze how each approach impacts system performance. We use AlexNet as the benchmark for this study, as PACT-SAWB and WRPN achieve iso-accuracy (57.2%), albeit at different quantization levels. PACT-SAWB quantizes both weights and activations to 2 bits, whereas WRPN quantizes weights to 2 bits and activations to 4 bits, while also doubling the number of channels in the quantized layers. The first and last layers are not quantized in both cases.

Figure 8(a) shows the classification throughput (normalized number of images classified per second) achieved by both techniques as the external memory bandwidth is varied. We find that PACT-SAWB achieves 2.7$\sim$3.1$\times$ speedup over WRPN. The benefits stem from 3 factors: (i) PACT-SAWB uses fewer computations, (ii) PACT-SAWB quantizes both weights and quantization to 2-bits, as opposed to just weights in WRPN, which allows us to pack more OPs in given area budget (Figure 7(b)), and (iii) PACT-SAWB uses less memory foot-print, which enables data-structures to fit on-chip and benefit performance when the external memory bandwidth is small. It is interesting to note that although doubling channels leads to a 4$\times$ increase in OPs, our benefits are roughly limited to 3$\times$. This is because, the first and last layers are not quantized and carried out in 16-bit. Although the constant overhead this adds is relatively small in terms of number of MAC operations, it is amplified by nearly an order of magnitude difference in the cost of 2-bit and 16-bit operations.

To further understand the reasons behind the speedup, Figure 8(b) compares the utilization of MAC array in case of PACT-SAWB and WRPN. At lower bandwidths, PACT-SAWB achieves a slightly better utilization compared to WRPN. This is because of PACT-SAWB's ability to have a smaller memory footprint, which reduces the pressure on the external memory bandwidth. However, at higher bandwidths, the utilization achieved by WRPN is larger, as doubling channels leads to increased data-reuse (higher OPs/Byte). Nevertheless, the increase in utilization is not commensurate with the higher OPs and bit-precision that WRPN demands, which results in PACT-SAWB eventually achieving substantial speedup.

8 Conclusion

In this paper, we propose novel techniques that target weight and activation quantizations separately resulting in an overall quantized neural network (QNN). The activation quantization technique, PArameterized Clipping acTivation (PACT), uses an activation clipping parameter α that is optimized during training to find the right quantization scale. The weight quantization scheme, statistics-aware weight binning (SAWB), finds the optimal scaling factor that minimizes the quantization error based on the statistical characteristics of the distribution of weights without performing an exhaustive search. Furthermore, we provide an innovative insight for quantization in the presence of shortcut connections, which motivates the use of high-precision for the shortcuts. Our evaluations show that the proposed quantization scheme outperforms the other prior quantization techniques. We demonstrate that using a 2-bit QNN, we achieve <1% accuracy loss for CIFAR10 tasks, and the best accuracy relative to other state-of-the-art quantization techniques for ImageNet tasks. Our demonstration of 2-bit ResNet50 showing less than 3% accuracy degradation is a significant achievement with great potential for practical use. Using a detailed hardware accelerator system performance model, we show that relative to the more recently proposed WRPN approach for quantization (Asit Mishra, 2018b), PACT-SAWB not only achieves iso-accuracy but also achieves 2.7$\sim$3.1$\times$ speedup.

Acknowledgements

The authors would like to thank Chia-Yu Chen, Naigang Wang, Xiao Sun, Ankur Agrawal, I-Hsin Chung, Ming-Hung Chen for helpful discussions and suggestions. This research was supported by IBM Research, IBM SoftLayer, and IBM Cognitive Computing Cluster (CCC).

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Antonio Polino, Razvan Pascanu, D. A. Model compression via distillation and quantization. *ICLR*, 2018.

Asit Mishra, D. M. Apprentice: Using Knowledge Distillation Techniques To Improve Low-Precision Network Accuracy. *International Conference on Learning Representations*, 2018a.

Asit Mishra, Eriko Nurvitadhi, J. J. C. D. M. WRPN: Wide Reduced-Precision Networks. *ICLR*, 2018b.

Baskin, C., Schwartz, E., Zheltonozhskii, E., Liss, N., Giryes, R., Bronstein, A. M., and Mendelson, A. Uniq: Uniform noise injection for the quantization of neural networks. *arXiv preprint arXiv:1804.10969*, 2018.

Bengio, Y., Léonard, N., and Courville, A. C. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *CoRR*, abs/1308.3432, 2013.

Cai, Z., He, X., Sun, J., and Vasconcelos, N. Deep Learning With Low Precision by Half-Wave Gaussian Quantization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Choi, Y., El-Khamy, M., and Lee, J. Learning low precision deep neural networks through regularization. *arXiv preprint arXiv:1809.00095*, 2018.

Courbariaux, M., Bengio, Y., and David, J. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. *CoRR*, abs/1511.00363, 2015.

Fleischer, B., Shukla, S., Ziegler, M., Silberman, J., Oh, J., Srinivasan, V., Choi, J., Mueller, S., Agrawal, A., Babinsky, T., et al. A scalable multi-teraops deep learning processor core for ai training and inference. In *VLSI Circuits, 2018 Symposium on*. IEEE, 2018.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional Sequence to Sequence Learning. *CoRR*, abs/1705.03122, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Identity Mappings in Deep Residual Networks. *CoRR*, abs/1603.05027, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016b.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Binarized Neural Networks. *NIPs*, pp. 4107–4115, 2016a.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. *CoRR*, abs/1609.07061, 2016b.

Hwang, K. and Sung, W. Fixed-point Feedforward Deep Neural Network Design Using Weights +1, 0, and -1. In *IEEE Workshop on Signal Processing Systems (SiPS)*, pp. 1–6, Oct. 2014.

Jan Achterhold, Jan Mathias Koehler, A. S. T. G. Variational Network Quantization. *ICLR*, 2018.

Jung, S., Son, C., Lee, S., Son, J., Kwak, Y., Han, J.-J., and Choi, C. Joint training of low-precision neural network with quantization interval parameters. *arXiv preprint arXiv:1808.05779*, 2018.

Kim, M. and Smaragdis, P. Bitwise Neural Networks. *ICML Workshop on Resource-Efficient Machine Learning*, 2015.

Krizhevsky, A. and Hinton, G. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40, 2010.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pp. 1097–1105, 2012.

Li, F. and Liu, B. Ternary weight networks. *CoRR*, abs/1605.04711, 2016. URL http://arxiv.org/abs/1605.04711.

Lu Hou, J. T. K. Loss-aware Weight Quantization of Deep Networks. *International Conference on Learning Representations*, 2018.

McDonnell, M. D. Training wide residual networks for deployment using a single bit for each weight. *ICLR*, 2018.

Nair, V. and Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. *27th International Conference on Machine Learning (ICML)*, pp. 807–814, 2010.

Park, E., Ahn, J., and Yoo, S. Weighted-Entropy-Based Quantization for Deep Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. *CoRR*, abs/1603.05279, 2016.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Venkataramani, S., Choi, J., Srinivasan, V., Gopalakrishnan, K., and Chang, L. POSTER: Design Space Exploration for Performance Optimization of Deep Neural Networks on Shared Memory Accelerators. *Proc. PACT 2017*.

Zagoruyko, S. and Komodakis, N. Wide Residual Networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhang, D., Yang, J., Ye, D., and Hua, G. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. *arXiv preprint arXiv:1807.10029*, 2018.

Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Laurent, C., Bengio, Y., and Courville, A. C. Towards End-to-End Speech Recognition with Deep Convolutional Neural Networks. *CoRR*, abs/1701.02720, 2017.

Zhou, S., Ni, Z., Zhou, X., Wen, H., Wu, Y., and Zou, Y. DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients. *CoRR*, abs/1606.06160, 2016.

Zhou, S., Wang, Y., Wen, H., He, Q., and Zou, Y. Balanced Quantization: An Effective and Efficient Approach to Quantized Neural Networks. *CoRR*, abs/1706.07145, 2017.

Zhu, C., Han, S., Mao, H., and Dally, W. J. Trained Ternary Quantization. *International Conference on Learning Representations (ICLR)*, 2017.

A CNN Implementation Details

The CIFAR10 dataset (Krizhevsky & Hinton, 2010) is an image classification benchmark containing 32×32 pixel RGB images. It consists of 50K training and 10K test image sets. We used the “full pre-activation” ResNet structure (He et al., 2016a) as well as the VGG (Simonyan & Zisserman, 2014) structure, which are most popular. The ResNet consists of a CONV layer followed by 3 ResNet blocks (18/30/42/54 CONV layers with 3x3 filter, depending on choice) and a final FC layer. The VGG consists of 3 blocks of 2-CONV followed by a max-pooling layer, followed by one FC layer and an SVM, similar to (Li & Liu, 2016). We used SGD with momentum of 0.9 and learning rate starting from 0.1 and scaled by 0.1 at epoch 60, 120. L2-regularizer with decay of 0.0002 is applied to weight. The mini-batch size of 128 is used, and the maximum number of epochs is 200.

The ImageNet dataset (Russakovsky et al., 2015) consists of 1000-categories of objects with over 1.2M training and 50K validation images. Images are first resized to 256 256 and randomly cropped to 224224 prior to being used as input to the network. We used a modified AlexNet, ResNet18 and ResNet50.

We used AlexNet network (Krizhevsky et al., 2012) in which local contrast renormalization (R-Norm) layer is replaced with BatchNorm layer. We used ADAM with epsilon 10^{-5} and learning rate starting from 10^{-4} and scaled by 0.2 at epoch 56 and 64. L2-regularizer with decay factor of 5×10^{-6} is applied to weight. The mini-batch size of 128 is used, and the maximum number of epochs is 100.

ResNet18 consists of a CONV layer followed by 8 ResNet blocks (16 CONV layers with 3x3 filter) and a final FC layer. "full pre-activation" ResNet structure (He et al., 2016a) is employed. ResNet50 consists of a CONV layer followed by 16 ResNet “bottleneck” blocks (total 48 CONV layers) and a final FC layer. "full pre-activation" ResNet structure (He et al., 2016a) is employed.

For both ResNet18 and ResNet50, we used SGD with momentum of 0.9 and learning rate starting from 0.1 and scaled by 0.1 at epoch 30, 60, 85, 95. L2-regularizer with decay of 10^{-4} is applied to weight. The mini-batch size of 256 is used, and the maximum number of epochs is 110.