

---

# ACCURATE AND EFFICIENT 2-BIT QUANTIZED NEURAL NETWORKS

---

Anonymous Authors<sup>1</sup>

## 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). The activation quantization technique, PArameterized Clipping acTivation (PACT), uses an activation clipping parameter  $\alpha$  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 the need for an exhaustive search. 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  $2.7 \times -3.1 \times$  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 GBs 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 to significantly reduce both storage requirements and computational complexity. Several weight quantization techniques (Li & Liu, 2016b; Zhu et al., 2017; Jan Achterhold, 2018; Antonio Polino, 2018; Lu Hou,

2018) have been proposed to reduce bit-precision of CNN weights but end up sacrificing model accuracy. Furthermore, a straightforward extension of weight quantization schemes to activation quantizations 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 quantization errors and allows more aggressive bit-width reductions. 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 improve 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 quantizations resulting in

---

<sup>1</sup>Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

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 (SAWB+PACT)	Compute scales (one-pass)	Uniform	N

an overall quantized neural network (QNN). The activation quantization technique, Parameterized Clipping acTivation (PACT), uses an activation clipping parameter  $\alpha$  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. Using PACT and SAWB we realize a 2-bit QNN which achieves state of the art classification accuracy comparable to full precision networks while incurring no larger than  $\mathcal{O}(n)$  computational overhead.

Furthermore, we use a detailed hardware accelerator system performance model and show that relative the WRPN approach to quantization (Asit Mishra, 2018b) and (McDonnell, 2018), PACT+SAWB not only achieves iso-accuracy but also achieves  $2.7 \times -3.1 \times$  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 CNNs. Section 7 describes the hardware accelerator system-level performance model, and presents the utilization and speedup achieved using 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), enabling an entire DNN model to fit effectively in resource-constrained platforms (e.g., mobile devices). Effectiveness of weight quantization techniques has been further improved (Li & Liu, 2016b; Zhu et al., 2017), by ternarizing weights using statistical distribution of weight values or by tuning quantization scales during training.

To reduce the overhead of 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 quantized using 1-bit as well. 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 (Rusakovsky 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)) uses iterative search algorithms for finding weight clusters or base/offset of logarithmic quantization for activation. Balanced quantization (BQ, (Zhou et al., 2017)) uses recursive partitioning for histogram equalization. Half-wave Gaussian quantization (Cai et al., 2017) finds the quantization scale via Lloyd optimization on Normal distribution. More recently, quantization schemes are learned in the context of training. LQ-Nets (Zhang et al., 2018) proposes learnable quantizers that can be updated via block-coordinate descent in the forward pass. Joint training of quantization interval parameters (QIP, (Jung et al., 2018)) parameterizes 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 to QNN focus on using simple quantization schemes but reduce quantization errors by performing more computations. One such approach is to increase the size of the network. Wide Residual Networks (Zagoruyko & Komodakis, 2016) demonstrates 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 Wide Residual Networks along with other modification in the training setting (e.g., warm-restart learning rate schedule) to improve accuracy for 1-bit weight quantiza-

tion. Another way of taking advantage of extra computation in quantization 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 and 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 increases the total flops for the network impacting inference latency.

In summary, prior quantization techniques either incur degradation in accuracy relative to full-precision, or use alternative approaches that require significantly more computations 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 without requiring changes to the network structure.

### 3 PARAMETERIZED CLIPPING ACTIVATION

#### 3.1 Challenge in Activation Quantization

Activation quantization becomes challenging when ReLU (the activation function most commonly used in CNNs) is used as the layer activation function. ReLU allows gradient of activations to propagate through deep layers and therefore achieves superior accuracy relative 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 the CIFAR10 dataset using ReLU and show that accuracy is significantly degraded when activation is quantized with ReLU.

It has been shown that this dynamic range problem can be alleviated by using a clipping activation function, which places an upper-bound on the output ([Hubara et al., 2016b; Zhou et al., 2016](#)). However, because of layer to layer and model to model differences - it is difficult to determine a globally optimal clipping value. In addition, as shown in Figure 1, even though the training error obtained using clipping with quantization is less than that obtained with quantized ReLU, the validation error is still noticeably higher than the baseline.

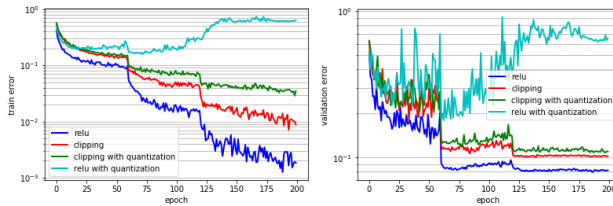


Figure 1. (a) Train error, (b) validation error when activation of CIFAR10 ResNet10 model with ReLU or clipping activation function is quantized. ReLU is more sensitive to activation quantization since it allows large dynamic range.

#### 3.2 Parameterized Clipping Activation Function

Building on these insights, we introduce PACT, a new activation quantization scheme in which the activation function has a parameterized clipping level,  $\alpha$ .  $\alpha$  is dynamically adjusted via gradient descent-based training with the objective of minimizing the accuracy degradation arising from quantization. In PACT, the conventional ReLU activation function in CNNs 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  $\alpha$  limits the 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, where

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

With this new activation function,  $\alpha$  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,

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

The larger the  $\alpha$ , 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  $\alpha$  in the loss function. Figure 2(b) illustrates how the value of  $\alpha$  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  $\alpha$  converges to values

much smaller than the initial value as the training epochs proceed, thereby limiting the dynamic range of activations and minimizing quantization loss.

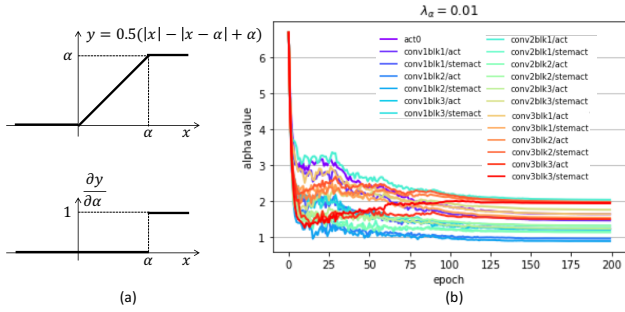


Figure 2. (a) PACT activation function and its gradient. The dynamic range of activation after PACT is bounded by  $\alpha$ , thus it is more robust to quantization. (b) Evolution of the trainable clipping parameter  $\alpha$  during training of CIFAR10 ResNet20.

### 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,  $\alpha$ , introduced in PACT, 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;  $x = w \cdot a$ ,  $y = \text{PACT}(x)$ , where  $a$  is input and  $w$  is weight. This network can be trained with SGD to find the output the network with ReLU would produce.

*Proof.* Consider a sample of training data  $(a, y^*)$ . For illustration purposes 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,

$$\frac{\partial L}{\partial \alpha} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial \alpha} = \frac{\partial L}{\partial y} \quad (4)$$

Therefore, when  $\alpha$  is updated by SGD,

$$\alpha_{new} = \alpha - \eta \frac{\partial L}{\partial \alpha} = \alpha - \eta \frac{\partial L}{\partial y} \quad (5)$$

where  $\eta$  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 MSE,  $\frac{\partial L}{\partial y} = (y - y^*)$ . Therefore, if  $y^* > x$ ,  $\alpha$  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  $y < y^* < x$ ,  $\alpha$  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.  $\square$

In general cases,  $\frac{\partial L}{\partial \alpha} = \sum_i \frac{\partial L}{\partial y_i}$ , and PACT considers output of 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  $\alpha$  is increased or decreased based on how close the overall output is to the target. Hence, there are configurations under which SGD would lead to a solution close to the one which the network with ReLU would achieve. Figure 3 demonstrates that CIFAR10 ResNet20 with PACT converges almost identical to the network with ReLU.

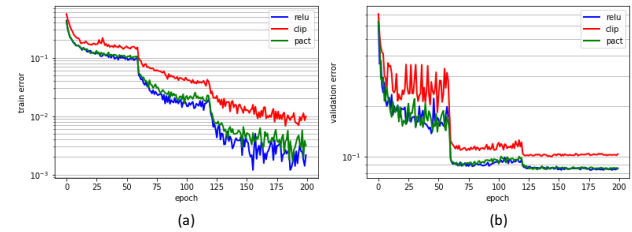


Figure 3. (a) Training error and (b) validation error of PACT for CIFAR10 ResNet20. Note that the convergence curve of PACT closely follow ReLU.

#### 3.3.2 Balancing Clipping and Quantization Errors

In Section 3.1, when we briefly discussed the challenges in activation quantization, we mentioned that there is a trade-off between errors due to clipping and quantization. As the clipping level increases, larger range of activation can be passed to the next layer of the neural network causing less clipping error ( $ErrClip_i = \max(x_i - \alpha, 0)$ ). However, the increased dynamic range incurs larger quantization error, since its magnitude is proportional to the clipping level ( $ErrQuant_i \leq 0.5 \cdot \frac{\alpha}{2^{k-1}}$ , with  $k$ -bit quantization). This imposes the challenge of finding a proper clipping level to balance between clipping and quantization errors.

This trade-off can be better observed in Figure 4(a), which



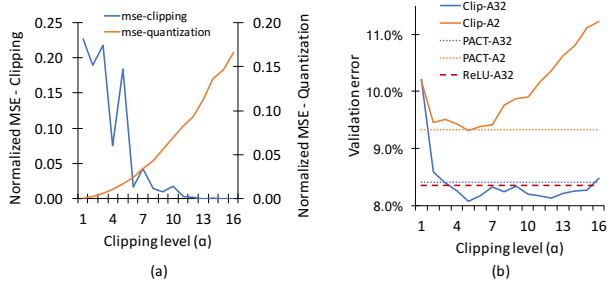


Figure 4. Experiments on CIFAR10 ResNet20 to validate that PACT balances clipping and quantization errors. (a) Trade-off between clipping and quantization error. (b) PACT achieving the lowest validation error that clipping activations can achieve without an exhaustive search over clipping levels  $\alpha$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

shows normalized mean-square-error caused by clipping and 2-bit quantization during training of the CIFAR10 ResNet20 with different clipping levels. It can be seen that activation functions with large dynamic range, such as ReLU, would suffer large quantization errors when the bit-precision is 2-bits. This explains why the network with ReLU fails to converge when the activation is quantized (Figure 1).

PACT can 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 during training. CIFAR10 ResNet20 is trained with clipping activation function with varying clipping level  $\alpha$  from 1 to 16. When activation is quantized with 2-bit, the network trained with clipping activation shows significant accuracy degradation as  $\alpha$  increases. This is consistent with the trend in quantization error we observed in Figure 4(a). In this case, PACT achieves the best accuracy amongst all the clipping functions, but without exhaustively sweeping over a range of clipping levels. In other words, PACT auto-tunes the clipping level to achieve the best accuracy without incurring significant computational overhead. PACT’s auto-tuning of the dynamic range 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  $\alpha$  is large. Surprisingly, some of the cases

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

even outperforms the ReLU network. In this case, PACT also achieves comparable accuracy to ReLU, confirming its expressivity discussed in Section 3.3.1.

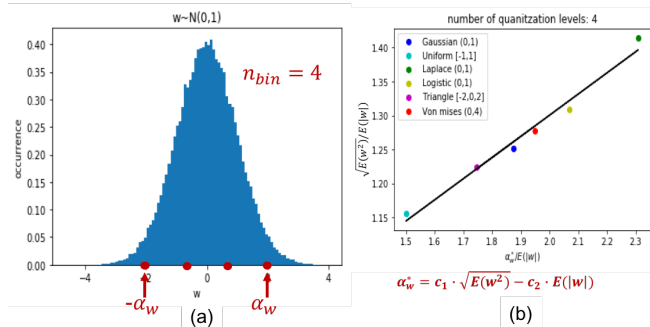


Figure 5. Statistics aware weight binning (SAWB). (a) Quantization levels determined by the scale  $\alpha$ . (b) Linear regression of the optimal scales from 6 distinct distributions shows that the optimal scale is well characterized by the first and second moments of the weight distribution.

## 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 both first and second moments of the weight distribution to minimize the quantization error. These statistics help the quantized states capture the shape changes in the weight distribution.

Figure 5(a) illustrates a 2-bit (4 bins) evenly-spaced SAWB scheme. We choose symmetrical and uniformly distributed quantization bins because it allows for a hardware friendly multiplication and accumulation (MAC) design. For SAWB, the scaling factor,  $\alpha_w$ , defines the largest quantization level and each weight is quantized to the nearest bin. For each weight distribution, there exists an optimal  $\alpha_w^*$  such that the MSE 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 a naive exhaustive search over  $\alpha_w$  is often not tractable, a popular approach is to assume the type of weight distribution and derive analytic solutions. For example, (Li & Liu, 2016a) conducted a case analysis with uniform and Normal distributions, motivated

by the fact that weights are often initialized with them, to derive an analytic formula for  $\alpha_w$  as a function of  $E(|w|)$ .

However, it is observed that the shape of the weight distribution is updated during back-propagation. This violates the base assumption, which in turn increases quantization errors. To overcome this limitation, we consider a larger variety of distributions with both  $E(|w|)$  and  $E(w^2)$ . Intuitively, the second moment  $E(w^2)$  captures the overall shape of distribution while the first moment  $E(|w|)$  captures the representative values. Therefore, by employing both  $E(|w|)$  and  $E(w^2)$ , the characteristics of the weight distribution can be better captured. Note that computation of  $E(w^2)$  does not incur iterative algorithms, so it is computationally cheap.

To validate this intuition, we first determine the optimal  $\alpha_w^*$  of the following six distributions for a wide range of quantization levels: 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 under consideration (points in different colors). Each point represents the optimal scale of that distribution obtained by a sweep over  $\alpha_w$  given the finite number of bins. We observe that the regression line closely fits the 6 points, thereby, validating our intuition that the optimal scale values can be mostly characterized by  $E(|w|)$  and  $E(w^2)$ . In fact, the behavior of  $\frac{\sqrt{E(w^2)}}{E(|w|)}$  is similar; the higher the variability in  $w$ , the larger the scale. This technique allowed us to capture the change in the shape of weight more faithfully.

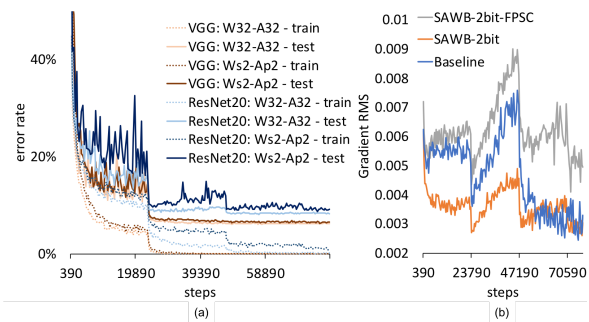


Figure 6. (a) Convergence plots for VGG and ResNet with and without quantization. While VGG is less sensitive to the quantization, ResNet suffers slower convergence when its shortcuts are quantized. (b) Impact of shortcut quantization visualized via root-mean-square (RMS) of weight gradient from the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

Given a bin level  $n_{bin}$ , linear regression can be applied to derive  $\alpha_w^*$  as a function of  $\sqrt{E(w^2)}$  and  $E(|w|)$ . As a result, during each mini-batch of training, we first computed  $E(|w|)$  and  $E(w^2)$ , and then applied the empirically determined linear coefficients  $c_1$  and  $c_2$  to calculate  $\hat{\alpha}_w^*$ , which is used for quantizing weights.  $c_1$  and  $c_2$  needs to

be determined off-line only once; we use  $c_1 = 2.587$  and  $c_2 = 1.693$  when  $n_{bin} = 4$ . SAWB can quickly determine  $\hat{\alpha}_w^*$  for a variety of weight distribution based on the distribution’s statistic (i.e.,  $E(w^2)$  and  $E(|w|)$ ) and results in a  $\mathcal{O}(n)$  complexity, where  $n$  is the number of weight.

To evaluate its effectiveness, we compared the quantization errors when using optimal vs. SAWB scaling factor. Table 2 summarizes the square error (SE) of layer 11’s weights based on the optimal and SAWB estimated scaling factor in a 2-bit quantized CIFAR10 ResNet.  $\alpha_w^*$  is determined through an off-line exhaustive search. It is evident that the  $\hat{\alpha}_w^*$  determined by the proposed SAWB incurs less than 7% additional SE at anytime during the training.

## 5 QUANTIZATION IN THE PRESENCE OF RESIDUAL CONNECTIONS

Throughout our quantization exploration, we observed that the feed-forward neural networks such as AlexNet and VGG behave more reliably compared to Residual Networks. Figure 6(a) demonstrates this issue with two training cases on VGG and CIFAR10 ResNet20 models with and without 2-bit weight/activation quantization. As shown, VGG’s training curves for full precision and the 2-bit networks track closely, leading to almost identical test errors. On the other hand, from the training curves for CIFAR10 ResNet20 we observe that the training error for the 2-bit network decreases slowly, indicating disruption during SGD optimization.

We made a key observation that applying quantization naively to the 1x1 convolution in shortcut (residual) paths impact accuracy because they play a key role in back-propagating gradient information across the layers and quantization fundamentally limits the flow of gradients. This could be evidenced by observing 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 baseline, SAWB-2bit with the full-precision shortcut (FPSC), and SAWB-2bit. As shown in the figure, the magnitude of gradients is significantly small when the shortcut is quantized. This can obstruct optimizer from finding good local minima, slowing down the overall progress of the optimization.

It is also mentioned in (He et al., 2016a) that keeping the shortcut connections more direct is beneficial for ease of optimization and to reduce over-fitting. In the same vein, we did not quantize the input activations or the weights in the residual path. Our quantization results show that when using full-precision only for the weights and activations in the shortcut path the accuracy of all the tested quantized ResNets is within  $\leq 1\%$  of the full-precision ResNets. Note that the shortcut in ResNet takes a negligible portion of the

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. “fpssc” 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-fpssc	8.64%	7.72%	7.29%	7.01%
Ws2-A32	9.27%	8.80%	7.84%	7.45%
Ws2-A32-fpssc	9.08%	7.74%	7.39%	7.07%
Ws2-Ap2	10.77%	9.57%	9.39%	8.76%
Ws2-Ap2-fpssc	9.35%	8.36%	7.61%	7.48%

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 QNN works that used the same experimental settings (including ternary quantizations that effectively use 2-bits), and report the accuracy numbers from their paper: LQ-Nets (Zhang et al., 2018), TWN (Li & Liu, 2016a), 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). For ResNet50, to make the comparison more exhaustive, we further include notable QNN implementations, Apprentice (Asit Mishra, 2018a), and UNIQ (Baskin et al., 2018) that use more than 2-bits. We implemented DoReFa-Net (Zhou et al., 2016) from the same setting (i.e., they share the same baseline).

### 6.1 CNN Implementation Details

In this section, we summarize details of our CNN implementation as well as our training settings, which is based on the default networks provided by Tensorpack (Zhou et al., 2016). Unless mentioned otherwise, ReLU following BatchNorm is used for activation function of the convolution (CONV) layers, and Softmax is used for the fully-connected (FC) layer. Note that the baseline networks use the same hyper-parameters and ReLU activation functions as described in the references. For PACT experiments, we only replace ReLU with PACT but otherwise use the same hyper-parameters. When SAWB is applied, the quantized

weight values are used in the forward and backward pass computation, but the full-precision copy is maintained for the gradient update. The networks are trained from scratch and the first/last layers are not quantized during training following common practice. (Zhou et al., 2016; Asit Mishra, 2018b).

The CIFAR10 dataset (Krizhevsky & Hinton, 2010) is an image classification benchmark containing  $32 \times 32$  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  $3 \times 3$  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, 2016a). We used stochastic gradient descent (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  $5 \times 10^{-6}$  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  $3 \times 3$  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 stochastic gradient descent (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.

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

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

## 6.2 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 weights and activations. Individual quantization incurs only marginal accuracy degradation, and the full-precision shortcut (postfix -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 show 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 SAWB-PACT-fpsc achieve slightly lower accuracy than LQ-Nets and QIP, respectively, but has lower accuracy degradation compared to them.

## 6.3 ImageNet Experiments

Next, we evaluate accuracy of our 2-bit QNN schemes on ImageNet dataset. As can be seen in Table 5, SAWB-PACT 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, SAWB-PACT’s ResNet accuracies (67.0% and 74.2% for ResNet18 and ResNet50) are the highest ever reported. The absolute ResNet50 accuracy is even higher

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

Name	Baseline	Quantized	Degradation
< AlexNet >			
<b>SAWB-PACT-fpsc</b>	<b>58.3</b>	<b>57.2</b>	<b>1.1</b>
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 >			
<b>SAWB-PACT-fpsc</b>	<b>70.4</b>	<b>67.0</b>	<b>3.4</b>
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 >			
<b>SAWB-PACT-fpsc</b>	<b>76.9</b>	<b>74.2</b>	<b>2.7</b>
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

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.

WRPN-x2 W2-A4 achieves accuracy on-par with our 2-bit QNN. Although WRPN-x2 regains the loss in accuracy by increasing the network size (i.e., doubling output channel length), the enlarged network size could negatively impact system-level performance gain achieved by the bit-width reduction. The following section investigates the impact of reduced precision inference on the system performance.

## 7 SYSTEM-LEVEL PERFORMANCE GAIN

### 7.1 System Settings

In this section, we demonstrate the gain in system performance as a result of the reduction in bit-precision achieved using PACT-SAWB. To this end, as shown in Figure 7(a), we consider a DNN accelerator system comprising of a DNN accelerator chip, comprising of multiple cores, interfaced with an external memory. 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.



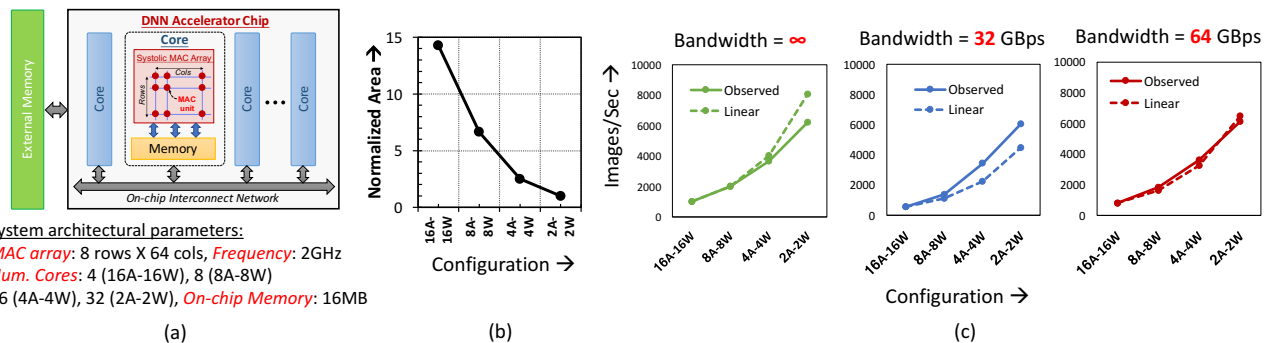


Figure 7. (a) System architecture and parameters. (b) Variation in MAC area with bit-precision. (c) Speedup at different quantizations for inference using ResNet50 model.

To estimate system performance at different bit precisions, we studied different versions of the DNN accelerator each comprising the same amount of on-chip memory, external memory bandwidth, and occupying iso-silicon area. First, using real hardware implementations in a up-to-date technology (14 nm CMOS), we accurately estimate the reduction in the MAC area achieved by aggressively scaling bit precision. As shown in Figure 7(b), we achieve  $\sim 14\times$  improvement in density when the bit-precisions 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 a precision-configurable MAC unit, whose bit precision can be modulated dynamically. The peak compute capability (FLOPs) of the MAC unit varied such that we achieve iso-area at each precision. Note that the total on-chip memory and external bandwidth remains constant at all precisions. We estimate the overall system performance using DeepMatrix, a detailed performance modelling framework for DNN accelerators ((Venkataramani et al.)).

## 7.2 Gain in Inference Performance

Figure 7(c) shows the gain in inference performance for the ResNet50 benchmark. 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 gain in performance is limited by how amenable it is to parallelize the work. 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 64 GBps, quantizing from 16 to 4 bits leads to a  $4\times$  increase in peak FLOPs but a  $4.5\times$  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 FLOPs. 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  $1.24\times$  improvement in performance, motivating the need to explore ways to quantize the first and last layers.

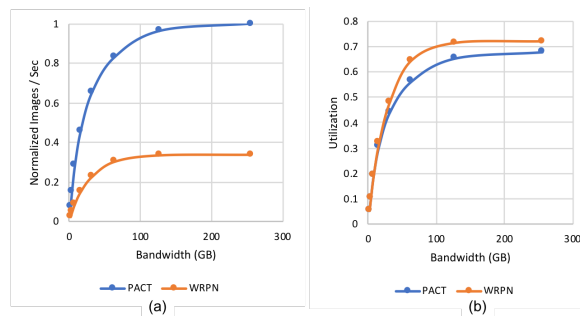


Figure 8. Comparison of system performance between PACT and WRPN. (a) Normalized images/sec, (b) Hardware utilization.

## 7.3 PACT-SAWB vs. WRPN

In PACT-SAWB, we heal the effect of quantization 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  $2.7\times-3.1\times$  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 FLOPs 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 FLOPs, our benefits are roughly limited to  $3\times$ . This is because, the first and last layers are not quantized and carried out in FP16. 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 INT2 and FP16 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 FLOPs/Byte). Nevertheless, the increase in utilization is not commensurate with the higher FLOPs 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, PArmeterized Clipping aCTivation (PACT), uses an activation clipping parameter  $\alpha$  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. Our evaluations show that our quantization scheme achieves best overall accuracy relative to

other prior quantization techniques. We demonstrate that using a 2-bit QNN, we achieve iso-accuracy relative to full-precision baseline for AlexNet, ( $< 1\%$ ) accuracy loss for CIFAR10 tasks, and best accuracy relative to other 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. Furthermore, we used a detailed hardware accelerator system performance model and showed that relative to the more recently proposed WRPN approach to quantization (Asit Mishra, 2018b) and (McDonnell, 2018), PACT-SAWB not only achieves iso-accuracy but also achieves  $2.7\times-3.1\times$  speedup.

## 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.

- 550 Courbariaux, M., Bengio, Y., and David, J. BinaryConnect:  
551 Training Deep Neural Networks with binary weights dur-  
552 ing propagations. *CoRR*, abs/1511.00363, 2015.
- 553 Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin,  
554 Y. N. Convolutional Sequence to Sequence Learning.  
555 *CoRR*, abs/1705.03122, 2017.
- 556 He, K., Zhang, X., Ren, S., and Sun, J. Identity Mappings in  
557 Deep Residual Networks. *CoRR*, abs/1603.05027, 2016a.
- 558 He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual  
559 Learning for Image Recognition. *IEEE Conference on  
560 Computer Vision and Pattern Recognition (CVPR)*, pp.  
561 770–778, 2016b.
- 562 Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and  
563 Bengio, Y. Binarized Neural Networks. *NIPS*, pp. 4107–  
564 4115, 2016a.
- 565 Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and  
566 Bengio, Y. Quantized Neural Networks: Training Neural  
567 Networks with Low Precision Weights and Activations.  
568 *CoRR*, abs/1609.07061, 2016b.
- 569 Hwang, K. and Sung, W. Fixed-point Feedforward Deep  
570 Neural Network Design Using Weights +1, 0, and -1. In  
571 *IEEE Workshop on Signal Processing Systems (SiPS)*, pp.  
572 1–6, Oct. 2014.
- 573 Jan Achterhold, Jan Mathias Koehler, A. S. T. G. Variational  
574 Network Quantization. *ICLR*, 2018.
- 575 Jung, S., Son, C., Lee, S., Son, J., Kwak, Y., Han, J.-J., and  
576 Choi, C. Joint training of low-precision neural network  
577 with quantization interval parameters. *arXiv preprint  
578 arXiv:1808.05779*, 2018.
- 579 Kim, M. and Smaragdīs, P. Bitwise Neural Networks. *ICML  
580 Workshop on Resource-Efficient Machine Learning*, 2015.
- 581 Krizhevsky, A. and Hinton, G. Convolutional deep belief  
582 networks on cifar-10. *Unpublished manuscript*, 40, 2010.
- 583 Krizhevsky, A., Sutskever, I., and Hinton, G. E. ImageNet  
584 Classification with Deep Convolutional Neural Networks.  
585 In *Advances in Neural Information Processing Systems  
586 25 (NIPS)*, pp. 1097–1105, 2012.
- 587 Li, F. and Liu, B. Ternary weight networks. *CoRR*,  
588 abs/1605.04711, 2016a. URL [http://arxiv.org/  
589 abs/1605.04711](http://arxiv.org/abs/1605.04711).
- 590 Li, F. and Liu, B. Ternary Weight Networks. *CoRR*,  
591 abs/1605.04711, 2016b.
- 592 Lu Hou, J. T. K. Loss-aware Weight Quantization of Deep  
593 Networks. *International Conference on Learning Repre-  
594 sentations*, 2018.
- 595 McDonnell, M. D. Training wide residual networks for  
596 deployment using a single bit for each weight. *ICLR*,  
597 2018.
- 598 Nair, V. and Hinton, G. E. Rectified Linear Units Im-  
599 prove Restricted Boltzmann Machines. *27th International  
600 Conference on Machine Learning (ICML)*, pp. 807–814,  
601 2010.
- 602 Park, E., Ahn, J., and Yoo, S. Weighted-Entropy-Based  
603 Quantization for Deep Neural Networks. *IEEE Con-  
604 ference on Computer Vision and Pattern Recognition  
(CVPR)*, July 2017.
- Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A.  
XNOR-Net: ImageNet Classification Using Binary Con-  
volutional 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 convolu-  
tional 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 Rep-  
resentations (ICLR)*, 2017.