ReActNet: Towards Precise Binary Neural Network with Generalized Activation Functions

Zechun Liu[1,2⋆], Zhiqiang Shen[2†], Marios Savvides[2], and Kwang-Ting Cheng[1]

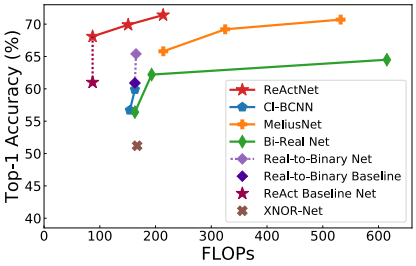
[1] Hong Kong University of Science and Technology, [2] Carnegie Mellon University
zliubq@connect.ust.hk, {zhiqians,marioss}@andrew.cmu.edu, timcheng@ust.hk

Abstract. In this paper, we propose several ideas for enhancing a binary network to close its accuracy gap from real-valued networks without incurring any additional computational cost. We first construct a baseline network by modifying and binarizing a compact real-valued network with parameter-free shortcuts, bypassing all the intermediate convolutional layers including the downsampling layers. This baseline network strikes a good trade-off between accuracy and efficiency, achieving superior performance than most of existing binary networks at approximately half of the computational cost. Through extensive experiments and analysis, we observed that the performance of binary networks is sensitive to activation distribution variations. Based on this important observation, we propose to generalize the traditional Sign and PReLU functions, denoted as RSign and RPReLU for the respective generalized functions, to enable explicit learning of the distribution reshape and shift at near-zero extra cost. Lastly, we adopt a distributional loss to further enforce the binary network to learn similar output distributions as those of a real-valued network. We show that after incorporating all these ideas, the proposed ReActNet outperforms all the state-of-the-arts by a large margin. Specifically, it outperforms Real-to-Binary Net and MeliusNet29 by 4.0% and 3.6% respectively for the top-1 accuracy and also reduces the gap to its real-valued counterpart to within 3.0% top-1 accuracy on ImageNet dataset. Code and models are available at: https://github.com/liuzechun/ReActNet.

1 Introduction

The 1-bit convolutional neural network (1-bit CNN, also known as binary neural network) [7,30], of which both weights and activations are binary, has been recognized as one of the most promising neural network compression methods for deploying models onto the resource-limited devices. It enjoys 32× memory compression ratio, and up to 58× practical computational reduction on CPU, as demonstrated in [30]. Moreover, with its pure logical computation (*i.e.*, XNOR operations between binary weights and binary activations), 1-bit CNN is both highly energy-efficient for embedded devices [8,40], and possesses the potential of being directly deployed on next generation memristor-based hardware [17].

⋆ Work done while visiting CMU. † Corresponding author.



Methods	OPs ($\times 10^8$)	Top-1 Acc(%)
Real-to-Binary Baseline [3]	1.63	60.9
Our ReAct Baseline Net	**0.87**	**61.1**
XNORNet [30]	1.67	51.2
Bi-RealNet [23]	1.63	56.4
Real-to-Binary [3]	1.65	65.4
MeliusNet29 [2]	2.14	65.8
Our ReActNet-A	**0.87**	**69.4**
MeliusNet59 [2]	5.32	70.7
Our ReActNet-C	**2.14**	**71.4**

Fig. 1. Computational cost vs. ImageNet Accuracy. Proposed ReActNets significantly outperform other binary neural networks. In particular, ReActNet-C achieves state-of-the-art result with 71.4% top-1 accuracy but being 2.5× more efficient than MeliusNet59. ReActNet-A exceeds Real-to-Binary Net and MeliusNet29 by 4.0% and 3.6% top-1 accuracy, respectively, and with more than 1.9× computational reduction. Details are described in Section 5.2.

Despite these attractive characteristics of 1-bit CNN, the severe accuracy degradation prevents it from being broadly deployed. For example, a representative binary network, XNOR-Net [30] only achieves 51.2% accuracy on the ImageNet classification dataset, leaving a $\sim$ 18% accuracy gap from the real-valued ResNet-18. Some preeminent binary networks [8,37] show good performance on small datasets such as CIFAR10 and MNIST, but still encounter severe accuracy drop when applied to a large dataset such as ImageNet.

In this study, our motivation is to further close the performance gap between binary neural networks and real-valued networks on the challenging large-scale datasets. We start with designing a high-performance baseline network. Inspired by the recent advances in real-valued compact neural network design, we choose MobileNetV1 [15] structure as our binarization backbone, which we believe is of greater practical value than binarizing non-compact models. Following the insights highlighted in [23], we adopt blocks with identity shortcuts which bypass 1-bit vanilla convolutions to replace the convolutions in MobileNetV1. Moreover, we propose to use a concatenation of two of such blocks to handle the channel number mismatch in the downsampling layers, as shown in Fig. 2(a). This baseline network design not only helps avoid real-valued convolutions in shortcuts, which effectively reduces the computation to near half of that needed in prevalent binary neural networks [30,23,3], but also achieves a high top-1 accuracy of 61.1% on ImageNet.

To further enhance the accuracy, we investigate another aspect which has not been studied in previous binarization or quantization works: activation distribution reshaping and shifting via non-linearity function design. We observed that the overall activation value distribution affects the feature representation, and this effect will be exaggerated by the activation binarization. A small distribution value shift near zero will cause the binarized feature map to have a disparate appearance and in turn will influence the final accuracy. This observation will be elaborated in Section 4.2. Enlightened by this observation, we propose a new gen-

eralization of Sign function and PReLU function to explicitly shift and reshape the activation distribution, denoted as ReAct-Sign (RSign) and ReAct-PReLU (RPReLU) respectively. These activation functions adaptively learn the parameters for distributional reshaping, which enhance the accuracy of the baseline network by $\sim 7\%$ with negligible extra computational cost.

Furthermore, we propose a distributional loss to enforce the output distribution similarity between the binary and real-valued networks, which further boosts the accuracy by $\sim 1\%$. After integrating all these ideas, the proposed network, dubbed as ReActNet, achieves 69.4% top-1 accuracy on ImageNet with only 87M OPs, surpassing all previously published works on binary networks and reduce the accuracy gap from its real-valued counterpart to only 3.0%, shown in Fig. 1.

We summarize our contributions as follows:

- We design a baseline binary network by modifying MobileNetV1, whose performance already surpasses most of the previously published work on binary networks while incurring only half of the computational cost.
- We propose a simple channel-wise reshaping and shifting operation on the activation distribution, which helps binary convolutions spare the computational power in adjusting the distribution to learn more representative features.
- We further adopt a distributional loss between binary and real-valued network outputs, replacing the original loss, which facilitates the binary network to mimic the distribution of a real-valued network.
- We demonstrate that our proposed ReActNet, which integrates the above mentioned contributions, achieves 69.4% top-1 accuracy on ImageNet, for the first time, exceeding the benchmarking ResNet-level accuracy (69.3%) while achieving more than 22× reduction in computational complexity. This result also outperforms the state-of-the-art binary network [3] by 4.0% top-1 accuracy while incurring only half the OPs[1].

2 Related Work

There have been extensive studies on neural network compression and acceleration, including quantization [46,39,43], pruning [9,12,24,22], knowledge distillation [14,33,6] and compact network design [15,32,25,41]. A comprehensive survey can be found in [35]. The proposed method falls into the category of quantization, specifically the extreme case of quantizing both weights and activations to only 1-bit, which is so-called network binarization or 1-bit CNNs.

Neural network binarization originates from EBP [34] and BNN [7], which establish an end-to-end gradient back-propagation framework for training the discrete binary weights and activations. As an initial attempt, BNN [7] demonstrated its success on small classification datasets including CIFAR10 [16] and MNIST [27], but encountered severe accuracy drop on a larger dataset such as

[1] OPs is a sum of binary OPs and floating-point OPs, i.e., OPs = BOPs/64 + FLOPs.

ImageNet [31], only achieving 42.2% top-1 accuracy compared to 69.3% of the real-valued version of the ResNet-18.

Many follow-up studies focused on enhancing the accuracy. XNOR-Net [30], which proposed real-valued scaling factors to multiply with each of binary weight kernels, has become a representative binarization method and enhanced the top-1 accuracy to 51.2%, narrowing the gap to the real-valued ResNet-18 to ∼18%. Based on the XNOR-Net design, Bi-Real Net [23] proposed to add shortcuts to propagate real-values along the feature maps, which further boost the top-1 accuracy to 56.4%.

Several recent studies attempted to improve the binary network performance via expanding the channel width [26], increasing the network depth [21] or using multiple binary weight bases [19]. Despite improvement to the final accuracy, the additional computational cost offsets the BNNs high compression advantage.

For network compression, the real-valued network design used as the starting point for binarization should be compact. Therefore, we chose MobileNetV1 as the backbone network for development of our baseline binary network, which combined with several improvements in implementation achieves $\sim 2\times$ further reduction in the computational cost compared to XNOR-Net and Bi-Real Net, and a top-1 accuracy of 61.1%, as shown in Fig. 1.

In addition to architectural design [2,23,28], studies on 1-bit CNNs expand from training algorithms [36,1,46,3], binary optimizer design [13], regularization loss design [8,29], to better approximation of binary weights and activations [30,11,37]. Different from these studies, this paper focuses on a new aspect that is seldom investigated before but surprisingly crucial for 1-bit CNNs accuracy, *i.e.* activation distribution reshaping and shifting. For this aspect, we propose novel *ReAct* operations, which are further combined with a proposed distributional loss. These enhancements improve the accuracy to 69.4%, further shrinking the accuracy gap to its real-valued counterpart to only 3.0%. The baseline network design and ReAct operations, as well as the proposed loss function are detailed in Section 4.

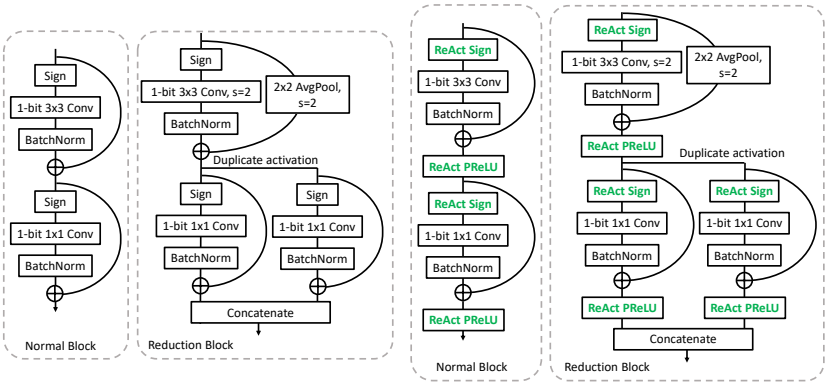
3 Revisit: 1-bit Convolution

In a 1-bit convolutional layer, both weights and activations are binarized to -1 and +1, such that the computationally heavy operations of floating-point matrix multiplication can be replaced by light-weighted bitwise XNOR operations and popcount operations [4], as:

$$\mathcal{X}_b * \mathcal{W}_b = \text{popcount}(\text{XNOR}(\mathcal{X}_b, \mathcal{W}_b)), \tag{1}$$

where $\mathcal{W}_b$ and $\mathcal{X}_b$ indicate the matrices of binary weights and binary activations. Specifically, weights and activations are binarized through a sign function:

$$x_b = \text{Sign}(x_r) = \begin{cases} +1, \text{ if } x_r > 0 \\ -1, \text{ if } x_r \leq 0 \end{cases}, \quad w_b = \frac{||\mathcal{W}_r||_{l1}}{n}\text{Sign}(w_r) = \begin{cases} +\frac{||\mathcal{W}_r||_{l1}}{n}, \text{ if } w_r > 0 \\ -\frac{||\mathcal{W}_r||_{l1}}{n}, \text{ if } w_r \leq 0 \end{cases} \tag{2}$$



(a) Proposed baseline network block (b) Proposed ReActNet block

Fig. 2. The proposed baseline network modified from MobileNetV1 [15], which replaces the original (3×3 depth-wise and 1×1 point-wise) convolutional pairs by the proposed blocks. (a) The baselines configuration in terms of channel and layer numbers is identical to that of MobileNetV1. If the input and output channel numbers are equal in a dw-pw-conv pair in the original network, a normal block is used, otherwise a reduction block is adopted. For the reduction block, we duplicate the input activation and concatenate the outputs to increase the channel number. As a result, all 1-bit convolutions have the same input and output channel numbers and are bypassed by identity shortcuts. (b) In the proposed ReActNet block, ReAct-Sign and ReAct-PReLU are added to the baseline network.

The subscripts b and r denote binary and real-valued, respectively. The weight binarization method is inherited from [30], of which, $\frac{||\mathcal{W}_r||_{l1}}{n}$ is the average of absolute weight values, used as a scaling factor to minimize the difference between binary and real-valued weights. XNOR-Net [30] also applied similar real-valued scaling factor to binary activations. Note that with introduction of the proposed ReAct operations, to be described in Section 4.2, this scaling factor for activations becomes unnecessary and can be eliminated.

4 Methodology

In this section, we first introduce our proposed baseline network in Section 4.1. Then we analyze how the variation in activation distribution affects the feature quality and in turn influences the final performance. Based on this analysis, we introduce ReActNet which explicitly reshapes and shifts the activation distribution using ReAct-PReLU and ReAct-Sign functions described in Section 4.2 and matches the outputs via a distributional loss defined between binary and real-valued networks detailed in Section 4.3.

4.1 Baseline Network

Most studies on binary neural networks have been binarizing the ResNet structure. However, further compressing compact networks, such as the MobileNets, would be more logical and of greater interest for practical applications. Thus, we chose MobileNetV1 [15] structure for constructing our baseline binary network.

Inspired by Bi-Real Net [23], we add a shortcut to bypass every 1-bit convolutional layer that has the same number of input and output channels. The 3×3 depth-wise and the 1×1 point-wise convolutional blocks in the MobileNetV1 [15] are replaced by the 3×3 and 1×1 vanilla convolutions in parallel with shortcuts, respectively, as shown in Fig. 2.

Moreover, we propose a new structure design to handle the downsampling layers. For the downsampling layers whose input and output feature map sizes differ, previous works [23,37,3] adopt real-valued convolutional layers to match their dimension and to make sure the real-valued feature map propagating along the shortcut will not be "cut off" by the activation binarization. However, this strategy increases the computational cost. Instead, our proposal is to make sure that all convolutional layers have the same input and output dimensions so that we can safely binarize them and use a simple identity shortcut for activation propagation without additional real-valued matrix multiplications.

As shown in Fig. 2(a), we duplicate input channels and concatenate two blocks with the same inputs to address the channel number difference and also use average pooling in the shortcut to match spatial downsampling. All layers in our baseline network are binarized, except the first input convolutional layer and the output fully-connect layer. Such a structure is hardware friendly.

4.2 ReActNet

The intrinsic property of an image classification neural network is to learn a mapping from input images to the output logits. A logical deduction is that a good performing binary neural network should learn similar logits distribution as a real-valued network. However, the discrete values of variables limit binary neural networks from learning as rich distributional representations as real-valued ones. To address it, XNOR-Net [30] proposed to calculate analytical real-valued scaling factors and multiply them with the activations. Its follow-up works [38,4] further proposed to learn these factors through back-propagation.

In contrast to these previous works, this paper focuses on a different aspect: the activation distribution. We observed that small variations to activation distributions can greatly affect the semantic feature representations in 1-bit CNNs, which in turn will influence the final performance. However, 1-bit CNNs have limited capacity to learn appropriate activation distributions. To address this dilemma, we introduce generalized activation functions with learnable coefficients to increase the flexibility of 1-bit CNNs for learning semantically-optimized distributions.

Distribution Matters in 1-bit CNNs The importance of distribution has not been investigated much in training a real-valued network, because with weights

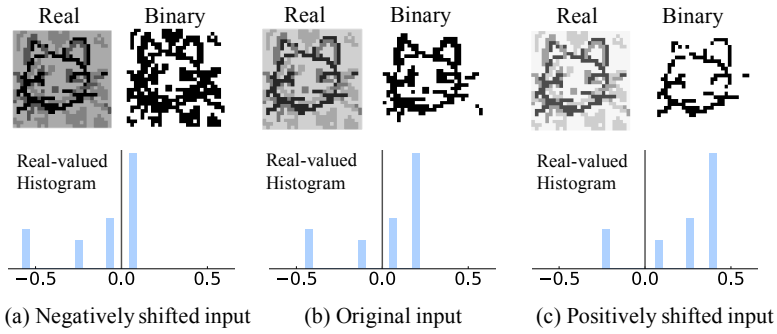


Fig. 3. An illustration of how distribution shift affects feature learning in binary neural networks. An ill-shifted distribution will introduce (a) too much background noise or (c) too few useful features, which harms feature learning.

and activations being continuous real values, reshaping or moving distributions would be effortless.

However, for 1-bit CNNs, learning distribution is both crucial and difficult. Because the activations in a binary convolution can only choose values from $\{-1, +1\}$, making a small distributional shift in the input real-valued feature map before the sign function can possibly result in a completely different output binary activations, which will directly affect the informativeness in the feature and significantly impact the final accuracy. For illustration, we plot the output binary feature maps of real-valued inputs with the original (Fig. 3(b)), positively-shifted (Fig. 3(a)), and negatively-shifted (Fig. 3(c)) activation distributions. Real-valued feature maps are robust to the shifts with which the legibility of semantic information will pretty much be maintained, while binary feature maps are sensitive to these shifts as illustrated in Fig. 3(a) and Fig. 3(c).

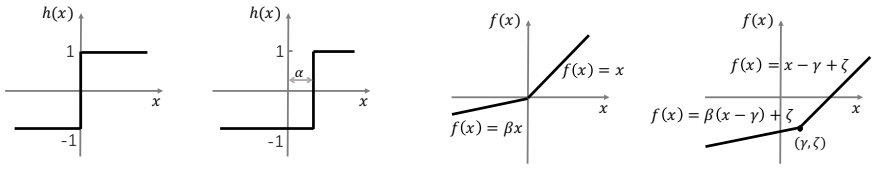
Explicit Distribution Reshape and Shift via Generalized Activation Functions Based on the aforementioned observation, we propose a simple yet effective operation to explicitly reshape and shift the activation distributions, dubbed as ReAct, which generalizes the traditional Sign and PReLU functions to ReAct-Sign (abbreviated as RSign) and ReAct-PReLU (abbreviated as RPReLU) respectively.

Definition

Essentially, RSign is defined as a sign function with channel-wisely learnable thresholds:

$$x_i^b = h(x_i^r) = \begin{cases} +1, & \text{if } x_i^r > \alpha_i \\ -1, & \text{if } x_i^r \leq \alpha_i \end{cases}. \tag{3}$$

Here, x_i^r is real-valued input of the RSign function h on the ith channel, x_i^b is the binary output and α_i is a learnable coefficient controlling the threshold. The subscript i in α_i indicates that the threshold can vary for different channels. The superscripts b and r refer to binary and real values. Fig. 4(a) shows the shapes of RSign and Sign.



(a) Sign vs. RSign (b) PReLU vs. RPReLU

Fig. 4. Proposed activation functions, RSign and RPReLU, with learnable coefficients and the traditional activation functions, Sign and PReLU.

Similarly, RPReLU is defined as

$$f(x_i) = \begin{cases} x_i - \gamma_i + \zeta_i, & \text{if } x_i > \gamma_i \\ \beta_i(x_i - \gamma_i) + \zeta_i, & \text{if } x_i \leq \gamma_i \end{cases}, \tag{4}$$

where x_i is the input of the RPReLU function f on the ith channel, γ_i and ζ_i are learnable shifts for moving the distribution, and β_i is a learnable coefficient controlling the slope of the negative part. All the coefficients are allowed to be different across channels. Fig. 4(b) compares the shapes of RPReLU and PReLU.

Intrinsically, RSign is learning the best channel-wise threshold (α) for binarizing the input feature map, or equivalently, shifting the input distribution to obtain the best distribution for taking a sign. From the latter angle, RPReLU can be easily interpreted as γ shifts the input distribution, finding a best point to use β to "fold" the distribution, then ζ shifts the output distribution, as illustrated in Fig. 5. These learned coefficients automatically adjust activation distributions for obtaining good binary features, which enhances the 1-bit CNNs' performance. With the introduction of these functions, the aforementioned difficulty in distributional learning can be greatly alleviated, and the 1-bit convolutions can effectively focus on learning more meaningful patterns. We will show later in the result section that this enhancement can boost the baseline networks top-1 accuracy substantially.

The number of extra parameters introduced by RSign and RPReLU is only $4 \times number\ of\ channels$ in the network, which is negligible considering the large size of the weight matrices. The computational overhead approximates a typical non-linear layer, which is also trivial compared to the computational intensive convolutional operations.

Optimization

Parameters in RSign and RPReLU can be optimized end-to-end with other parameters in the network. The gradient of α_i in RSign can be simply derived by the chain rule as:

$$\frac{\partial \mathcal{L}}{\partial \alpha_i} = \sum_{x_i^r} \frac{\partial \mathcal{L}}{\partial h(x_i^r)} \frac{\partial h(x_i^r)}{\partial \alpha_i}, \tag{5}$$

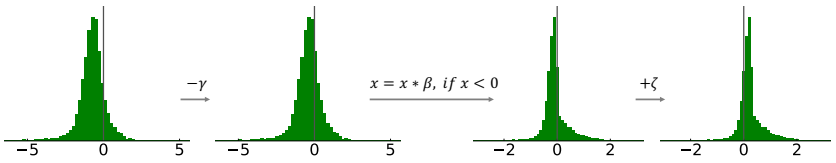


Fig. 5. An explanation of how proposed RPReLU operates. It first moves the input distribution by $-\gamma$, then reshapes the negative part by multiplying it with β and lastly moves the output distribution by ζ.

where $\mathcal{L}$ represents the loss function and $\frac{\partial \mathcal{L}}{\partial h(x_i^r)}$ denotes the gradients from deeper layers. The summation $\sum_{x_i^r}$ is applied to all entries in the ith channel. The derivative $\frac{\partial h(x_i^r)}{\partial \alpha_i}$ can be easily computed as

$$\frac{\partial h(x_i^r)}{\partial \alpha_i} = -1 \tag{6}$$

Similarly, for each parameter in RPReLU, the gradients are computed with the following formula:

$$\frac{\partial f(x_i)}{\partial \beta_i} = \mathbf{I}_{\{x_i \leq \gamma_i\}} \cdot (x - \gamma_i), \tag{7}$$

$$\frac{\partial f(x_i)}{\partial \gamma_i} = -\mathbf{I}_{\{x_i \leq \gamma_i\}} \cdot \beta_i - \mathbf{I}_{\{x_i > \gamma_i\}}, \tag{8}$$

$$\frac{\partial f(x_i)}{\partial \zeta_i} = 1. \tag{9}$$

Here, $\mathbf{I}$ denotes the indicator function. $\mathbf{I}_{\{\cdot\}} = 1$ when the inequation inside $\{\}$ holds, otherwise $\mathbf{I}_{\{\cdot\}} = 0$.

4.3 Distributional Loss

Based on the insight that if the binary neural networks can learn similar distributions as real-valued networks, the performance can be enhanced, we use a distributional loss to enforce this similarity, formulated as:

$$\mathcal{L}_{Distribution} = -\frac{1}{n} \sum_{c} \sum_{i=1}^{n} p_c^{\mathcal{R}_\theta}(X_i) \log\left(\frac{p_c^{\mathcal{B}_\theta}(X_i)}{p_c^{\mathcal{R}_\theta}(X_i)}\right), \tag{10}$$

where the distributional loss $\mathcal{L}_{Distribution}$ is defined as the KL divergence between the softmax output p_c of a real-valued network $\mathcal{R}_\theta$ and a binary network $\mathcal{B}_\theta$. The subscript c denotes classes and n is the batch size.

Different from prior work [46] that needs to match the outputs from every intermediate block, or further using multi-step progressive structural transition [3], we found that our distributional loss, while much simpler, can yield competitive

results. Moreover, without block-wise constraints, our approach enjoys the flexibility in choosing the real-valued network without the requirement of architecture similarity between real and binary networks.

5 Experiments

To investigate the performance of the proposed methods, we conduct experiments on ImageNet dataset. We first introduce the dataset and training strategy in Section 5.1, followed by comparison between the proposed networks and state-of-the-arts in terms of both accuracy and computational cost in Section 5.2. We then analyze the effects of the distributional loss, concatenated downsampling layer and the RSign and the RPReLU in detail in the ablation study described in Section 5.3. Visualization results on how RSign and RPReLU help binary network capture the fine-grained underlying distribution are presented in Section 5.4.

5.1 Experimental Settings

Dataset The experiments are carried out on the ILSVRC12 ImageNet classification dataset [31], which is more challenging than small datasets such as CIFAR [16] and MNIST [27]. In our experiments, we use the classic data augmentation method described in [15].
Training Strategy We followed the standard binarization method in [23] and adopted the two-step training strategy as [3]. In the first step, we train a network with binary activations and real-valued weights from scratch. In the second step, we inherit the weights from the first step as the initial value and fine-tune the network with weights and activations both being binary. For both steps, Adam optimizer with a linear learning rate decay scheduler is used, and the initial learning rate is set to 5e-4. We train it for 600k iterations with batch size being 256. The weight decay is set to 1e-5 for the first step and 0 for the second step.
Distributional Loss In both steps, we use proposed distributional loss as the objective function for optimization, replacing the original cross-entropy loss between the binary network output and the label.
OPs Calculation We follow the calculation method in [3], we count the binary operations (BOPs) and floating point operations (FLOPs) separately. The total operations (OPs) is calculated by OPs = BOPs/64 + FLOPs, following [30,23].

5.2 Comparison with State-of-the-art

We compare ReActNet with state-of-the-art quantization and binarization methods. Table 1 shows that ReActNet-A already outperforms all the quantizing methods in the left part, and also archives 4.0% higher accuracy than the state-of-the-art Real-to-Binary Network [3] with only approximately half of the OPs. Moreover, in contrast to [3] which computes channel re-scaling for each block

Methods	Bitwidth (W/A)	Acc(%) Top-1	Binary Methods	BOPs ($\times 10^9$)	FLOPs ($\times 10^8$)	OPs ($\times 10^8$)	Acc(%) Top-1
BWN [7]	1/32	60.8	BNNs [7]	1.70	1.20	1.47	42.2
TWN [18]	2/32	61.8	CI-BCNN [37]	–	–	1.63	59.9
INQ [42]	2/32	66.0	Binary MobileNet [28]	–	–	1.54	60.9
TTQ [44]	2/32	66.6	PCNN [11]	–	–	1.63	57.3
SYQ [10]	1/2	55.4	XNOR-Net [30]	1.70	1.41	1.67	51.2
HWGQ [5]	1/2	59.6	Trained Bin [38]	–	–	–	54.2
LQ-Nets [39]	1/2	62.6	Bi-RealNet-18 [23]	1.68	1.39	1.63	56.4
DoReFa-Net [43]	1/4	59.2	Bi-RealNet-34 [23]	3.53	1.39	1.93	62.2
Ensemble BNN [45]	(1/1) × 6	61.1	Bi-RealNet-152 [21]	10.7	4.48	6.15	64.5
Circulant CNN [20]	(1/1) × 4	61.4	Real-to-Binary Net [3]	1.68	1.56	1.83	65.4
Structured BNN [47]	(1/1) × 4	64.2	MeliusNet29 [2]	5.47	1.29	2.14	65.8
Structured BNN* [47]	(1/1) × 4	66.3	MeliusNet42 [2]	9.69	1.74	3.25	69.2
ABC-Net [19]	(1/1) × 5	65.0	MeliusNet59 [2]	18.3	2.45	5.32	70.7
Our ReActNet-A	1/1	–	–	**4.82**	**0.12**	**0.87**	**69.4**
Our ReActNet-B	1/1	–	–	**4.69**	**0.44**	**1.63**	**70.1**
Our ReActNet-C	1/1	–	–	**4.69**	**1.40**	**2.14**	**71.4**

Table 1. Comparison of the top-1 accuracy with state-of-the-art methods. The left part presents quantization methods applied on ResNet-18 structure and the right part are binarization methods with varied structures (ResNet-18 if not specified). Quantization methods include weight quantization (upper left block), low-bit weight and activation quantization (middle left block) and the weight and activation binarization with the expanded network capacity (lower left block), where the number times (1/1) indicates the multiplicative factor. (W/A) represents the number of bits used in weight or activation quantization.

with real-valued fully-connected layers, ReActNet-A has pure 1-bit convolutions except the first and the last layers, which is more hardware-friendly.

To make further comparison with previous approaches that use real-valued convolution to enhance binary networks accuracy [23,3,2], we constructed ReActNet-B and ReActNet-C, which replace the 1-bit 1×1 convolution with real-valued 1×1 convolution in the downsampling layers, as shown in Fig. 6(c). ReActNet-B defines the real-valued convolutions to be group convolutions with 4 groups, while ReActNet-C uses full real-valued convolution. We show that ReActNet-B achieves 13.7% higher accuracy than Bi-RealNet-18 with the same number of OPs and ReActNet-C outperforms MeliusNet59 by 0.7% with less than half of the OPs.

Moreover, we applied the ReAct operations to Bi-RealNet-18, and obtained 65.5% Top-1 accuracy, increasing the accuracy of Bi-RealNet-18 by 9.1% without changing the network structure.

Considering the challenges in previous attempts to enhance 1-bit CNNs performance, the accuracy leap achieved by ReActNets is significant. It requires an ingenious use of binary networks special property to effectively utilize every precious bit and strike a delicate balance between binary and real-valued information. For example, ReActNet-A, with 69.4% top-1 accuracy at 87M OPs, outperforms the real-valued 0.5× MobileNetV1 by 5.7% greater accuracy at

Network	Top-1 Acc(%)
Baseline network † *	58.2
Baseline network †	59.6
Proposed baseline network *	61.1
Proposed baseline network	62.5
Proposed baseline network + PReLU	65.5
Proposed baseline network + RSign	66.1
Proposed baseline network + RPReLU	67.4
ReActNet-A (RSign and RPReLU)	69.4
Corresponding real-valued network	72.4

Table 2. The effects of different components in ReActNet on the final accuracy. († denotes the network not using the concatenated blocks, but directly binarizing the downsampling layers instead. * indicates not using the proposed distributional loss during training.)

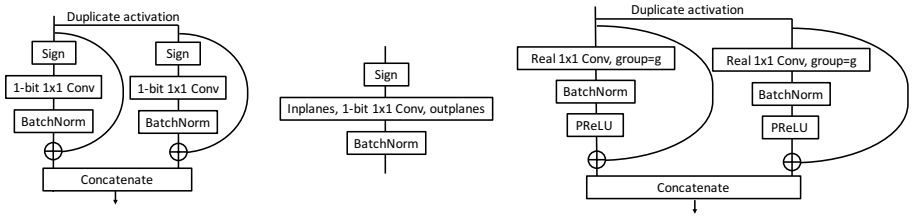
41.6% fewer OPs. These results demonstrate the potential of 1-bit CNNs and the effectiveness of our ReActNet design.

5.3 Ablation Study

We conduct ablation studies to analyze the individual effect of the following proposed techniques:

Block Duplication and Concatenation Real-valued shortcuts are crucial for binary neural network accuracy [23]. However, the input channels of the downsampling layers are twice the output channels, which violates the requirement for adding the shortcuts that demands an equal number of input and output channels. In the proposed baseline network, we duplicate the downsampling blocks and concatenate the outputs (Fig. 6(a)), enabling the use of identity shortcuts to bypass 1-bit convolutions in the downsampling layers. This idea alone results in a 2.9% accuracy enhancement compared to the network without concatenation (Fig. 6(b)). The enhancement can be observed by comparing the $2nd$ and $4th$ rows of Table 2. With the proposed downsampling layer design, our baseline network achieves both high accuracy and high compression ratio. Because it no longer requires real-valued matrix multiplications in the downsampling layers, the computational cost is greatly reduced. As a result, even without using the distributional loss in training, our proposed baseline network has already surpassed the Strong Baseline in [3] by 0.1% for top-1 accuracy at only half of the OPs. With this strong performance, this simple baseline network serves well as a new baseline for future studies on compact binary neural networks.

Distributional Loss The results in the first section of Table 2 also validate that the distributional loss designed for matching the output distribution between binary and real-valued neural networks is effective for enhancing the performance of propose baseline network, improving the accuracy by 1.4%, which is achieved independent of the network architecture design.



(a) Concatenated 1-bit convolutional block in Proposed Baseline Network

(b) Normal 1-bit convolutional block for downsampling in baseline network

(c) Real valued convolutional block for downsampling in ReActNet-B and C

Fig. 6. Variations in the downsampling layer design.

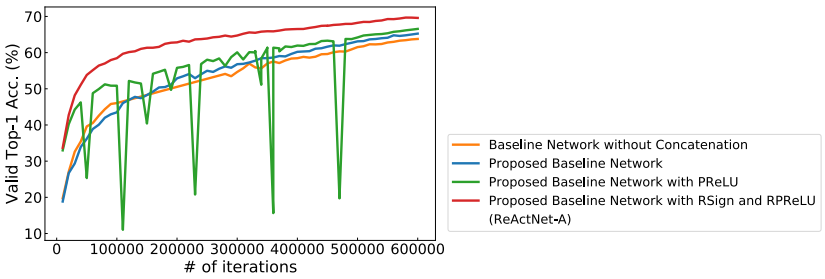


Fig. 7. Comparing validation accuracy curves between baseline networks and ReActNet. Using proposed RSign and RPReLU (red curve) achieves the higher accuracy and is more robust than using Sign and PReLU (green curve).

ReAct Operations The introduction of RSign and RPReLU improves the accuracy by 4.9% and 3.6% respectively over the proposed baseline network, as shown in the second section of Table 2. By adding both RSign and RPReLU, ReActNet-A achieves 6.9% higher accuracy than the baseline, narrowing the accuracy gap to the corresponding real-valued network to within 3.0%. Compared to merely using the Sign and PReLU, the use of the generalized activation functions, RSign and RPReLU, with simple learnable parameters boost the accuracy by 3.9%, which is very significant for the ImageNet classification task. As shown in Fig. 7, the validation curve of the network using original Sign + PReLU oscillates vigorously, which is suspected to be triggered by the slope coefficient β in PReLU changing its sign which in turn affects the later layers with an avalanche effect. This also indirectly confirms our assumption that 1-bit CNNs are vulnerable to distributional changing. In comparison, the proposed RSign and RPReLU functions are effective for stabilizing training in addition to improving the accuracy.

5.4 Visualization

To help gain better insights, we visualize the learned coefficients as well as the intermediate activation distributions.

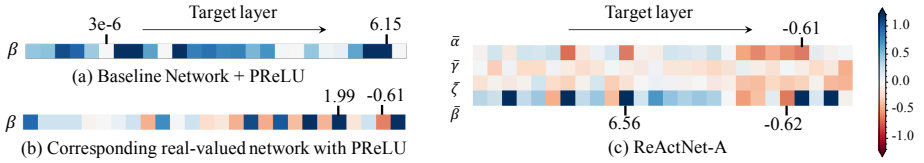


Fig. 8. The color bar of the learned coefficients. Blue color denotes the positive values while red denotes the negative, and the darkness in color reflects the absolute value. We also mark coefficients that have extreme values.

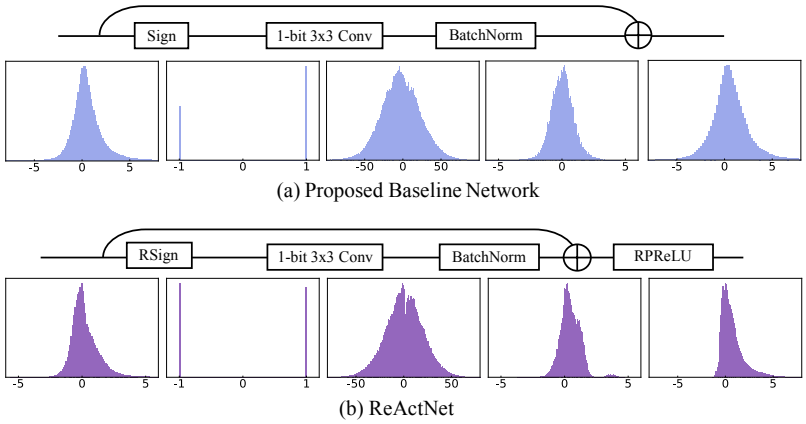


Fig. 9. Histogram of the activation distribution

Learned Coefficients For clarity, we present the learned coefficients of each layer in form of the color bar in Fig. 8. Compared to the binary network using traditional PReLU whose learned slopes β are positive only (Fig. 8(a)), ReActNet using RPReLU learns both positive and negative slopes (Fig. 8(c)), which are closer to the distributions of PReLU coefficients in a real-valued network we trained (Fig. 8(b)). The learned distribution shifting coefficients also have large absolute values as shown in Rows 1-3 of Fig. 8(c), indicating the necessity of their explicit shift for high-performance 1-bit CNNs.

Activation Distribution In Fig. 9, we show the histograms of activation distributions inside the trained baseline network and ReActNet. Compared to the baseline network without RSign and RPReLU, ReActNets distributions are more enriched and subtle, as shown in the forth sub-figure in Fig. 9(b). Also, in ReActNet, the distribution of -1 and +1 after the sign function is more balanced, as illustrated in the second sub-figure in Fig. 9(b), suggesting better utilization of black and white pixels in representing the binary features.

6 Conclusions

In this paper, we present several new ideas to optimize a 1-bit CNN for higher accuracy. We first design parameter-free shortcuts based on MobileNetV1 to propagate real-valued feature maps in both normal convolutional layers as well as the downsampling layers. This yields a baseline binary network with 61.1% top-1 accuracy at only 87M OPs for the ImageNet dataset. Then, based on our observation that 1-bit CNNs performance is highly sensitive to distributional variations, we propose ReAct-Sign and ReAct-PReLU to enable shift and reshape the distributions in a learnable fashion and demonstrate their dramatical enhancements on the top-1 accuracy. We also propose to incorporate a distributional loss, which is defined between the outputs of the binary network and the real-valued reference network, to replace the original cross-entropy loss for training. With contributions jointly achieved by these ideas, the proposed ReActNet achieves 69.4% top-1 accuracy on ImageNet, which is just 3% shy of its real-valued counterpart while at substantially lower computational cost.

References

1. Alizadeh, M., Fernández-Marqués, J., Lane, N.D., Gal, Y.: An empirical study of binary neural networks' optimisation (2018) 4
2. Bethge, J., Bartz, C., Yang, H., Chen, Y., Meinel, C.: Meliusnet: Can binary neural networks achieve mobilenet-level accuracy? arXiv preprint arXiv:2001.05936 (2020) 2, 4, 11
3. Brais Martinez, Jing Yang, A.B.G.T.: Training binary neural networks with real-to-binary convolutions. International Conference on Learning Representations (2020) 2, 3, 4, 6, 9, 10, 11, 12
4. Bulat, A., Tzimiropoulos, G.: Xnor-net++: Improved binary neural networks. British Machine Vision Conference (2019) 4, 6
5. Cai, Z., He, X., Sun, J., Vasconcelos, N.: Deep learning with low precision by half-wave gaussian quantization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5918–5926 (2017) 11
6. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: Advances in Neural Information Processing Systems. pp. 742–751 (2017) 3
7. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. arXiv preprint arXiv:1602.02830 (2016) 1, 3, 11
8. Ding, R., Chin, T.W., Liu, Z., Marculescu, D.: Regularizing activation distribution for training binarized deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11408–11417 (2019) 1, 2, 4
9. Ding, X., Zhou, X., Guo, Y., Han, J., Liu, J., et al.: Global sparse momentum sgd for pruning very deep neural networks. In: Advances in Neural Information Processing Systems. pp. 6379–6391 (2019) 3
10. Faraone, J., Fraser, N., Blott, M., Leong, P.H.: Syq: Learning symmetric quantization for efficient deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4300–4309 (2018) 11

11. Gu, J., Li, C., Zhang, B., Han, J., Cao, X., Liu, J., Doermann, D.: Projection convolutional neural networks for 1-bit cnns via discrete back propagation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8344–8351 (2019) 4, 11
12. He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1389–1397 (2017) 3
13. Helwegen, K., Widdicombe, J., Geiger, L., Liu, Z., Cheng, K.T., Nusselder, R.: Latent weights do not exist: Rethinking binarized neural network optimization. In: Advances in neural information processing systems. pp. 7531–7542 (2019) 4
14. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015) 3
15. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017) 2, 3, 5, 6, 10
16. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009) 3, 10
17. Li, B., Shan, Y., Hu, M., Wang, Y., Chen, Y., Yang, H.: Memristor-based approximated computation. In: Proceedings of the 2013 International Symposium on Low Power Electronics and Design. pp. 242–247. IEEE Press (2013) 1
18. Li, F., Zhang, B., Liu, B.: Ternary weight networks. arXiv preprint arXiv:1605.04711 (2016) 11
19. Lin, X., Zhao, C., Pan, W.: Towards accurate binary convolutional neural network. In: Advances in Neural Information Processing Systems. pp. 345–353 (2017) 4, 11
20. Liu, C., Ding, W., Xia, X., Zhang, B., Gu, J., Liu, J., Ji, R., Doermann, D.: Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnns with circulant back propagation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2691–2699 (2019) 11
21. Liu, Z., Luo, W., Wu, B., Yang, X., Liu, W., Cheng, K.T.: Bi-real net: Binarizing deep network towards real-network performance. International Journal of Computer Vision pp. 1–18 (2018) 4, 11
22. Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, K.T., Sun, J.: Metapruning: Meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3296–3305 (2019) 3
23. Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., Cheng, K.T.: Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In: Proceedings of the European conference on computer vision (ECCV). pp. 722–737 (2018) 2, 4, 6, 10, 11, 12
24. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2736–2744 (2017) 3
25. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 116–131 (2018) 3
26. Mishra, A., Nurvitadhi, E., Cook, J.J., Marr, D.: Wrpn: wide reduced-precision networks. arXiv preprint arXiv:1709.01134 (2017) 4
27. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 5 (2011) 3, 10

28. Phan, H., Liu, Z., Huynh, D., Savvides, M., Cheng, K.T., Shen, Z.: Binarizing mobilenet via evolution-based searching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13420–13429 (2020) 4, 11
29. Qin, H., Gong, R., Liu, X., Shen, M., Wei, Z., Yu, F., Song, J.: Forward and backward information retention for accurate binary neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2250–2259 (2020) 4
30. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision. pp. 525–542. Springer (2016) 1, 2, 4, 5, 6, 10, 11
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015) 4, 10
32. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018) 3
33. Shen, Z., He, Z., Xue, X.: Meal: Multi-model ensemble via adversarial learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4886–4893 (2019) 3
34. Soudry, D., Hubara, I., Meir, R.: Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In: Advances in Neural Information Processing Systems. pp. 963–971 (2014) 3
35. Sze, V., Chen, Y.H., Yang, T.J., Emer, J.S.: Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE **105**(12), 2295–2329 (2017) 3
36. Tang, W., Hua, G., Wang, L.: How to train a compact binary neural network with high accuracy? In: Thirty-First AAAI conference on artificial intelligence (2017) 4
37. Wang, Z., Lu, J., Tao, C., Zhou, J., Tian, Q.: Learning channel-wise interactions for binary convolutional neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 2, 4, 6, 11
38. Xu, Z., Cheung, R.C.: Accurate and compact convolutional neural networks with trained binarization. British Machine Vision Conference (2019) 6, 11
39. Zhang, D., Yang, J., Ye, D., Hua, G.: Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 365–382 (2018) 3, 11
40. Zhang, J., Pan, Y., Yao, T., Zhao, H., Mei, T.: dabnn: A super fast inference framework for binary neural networks on arm devices. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 2272–2275 (2019) 1
41. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018) 3
42. Zhou, A., Yao, A., Guo, Y., Xu, L., Chen, Y.: Incremental network quantization: Towards lossless cnns with low-precision weights. arXiv preprint arXiv:1702.03044 (2017) 11
43. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. arXiv preprint arXiv:1606.06160 (2016) 3, 11
44. Zhu, C., Han, S., Mao, H., Dally, W.J.: Trained ternary quantization. arXiv preprint arXiv:1612.01064 (2016) 11

45. Zhu, S., Dong, X., Su, H.: Binary ensemble neural network: More bits per network or more networks per bit? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4923–4932 (2019) 11
46. Zhuang, B., Shen, C., Tan, M., Liu, L., Reid, I.: Towards effective low-bitwidth convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7920–7928 (2018) 3, 4, 9
47. Zhuang, B., Shen, C., Tan, M., Liu, L., Reid, I.: Structured binary neural networks for accurate image classification and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 413–422 (2019) 11