TRAINING BINARY NEURAL NETWORKS WITH REAL-TO-BINARY CONVOLUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper shows how to train binary networks to within a few percent points ($\sim 3-5\%$) of the full precision counterpart with a negligible increase in the computational cost. In particular, we first show how to build a strong baseline, which already achieves state-of-the-art accuracy, by combining recently proposed advances, and carefully tuning the optimization procedure. Secondly, we show that by attempting to minimize the discrepancy between the output of the binary and the corresponding real-valued convolution additional significant accuracy gains can be obtained. We materialize this idea in two complementary ways: (1) with a loss function, during training, by matching the spatial attention maps computed at the output of the binary and real-valued convolutions, and (2) in data-driven manner, by using the real-valued activations being available during inference *prior to* the binarization process for re-scaling the activations *right after* the binary convolution. Finally, we show that, when putting all of our improvements together, the resulting model reduces the gap to its real-valued counterpart to less than 3% and 5% top-1 error on CIFAR-100 and ImageNet, respectively, when using a ResNet-18 architecture.

1 INTRODUCTION

Following the introduction of the BinaryNeuralNet (BNN) algorithm (Courbariaux et al., 2016), binary neural networks emerged as one of the most promising approaches for training highly efficient neural networks that can be deployed on devices with limited computational resources. Binary networks are particularly appealing for two purposes: (a) Model compression: if the weights of the network are stored as bits in a 32-bit float, this implies a reduction of $32\times$ in memory usage. (b) Computational speed-up: computationally intensive floating-point multiply and add operations are replaced by efficient `xnor` and `pop-count` operations which have been shown to provide practical speed-ups of up to $58\times$ on CPU (Rastegari et al., 2016). Despite these appealing properties, binary neural networks have been criticized as binarization typically results in large accuracy drop, which renders their deployment in practical applications unlikely. For example, on ImageNet classification, there is a $\sim 18\%$ gap in top-1 accuracy between a ResNet-18 and its binary counterpart when binarized with XNOR-Net (Rastegari et al., 2016), which is the method of choice for neural network binarization.

But how far are we from training binary neural networks that are powerful enough to reach the accuracy levels of their real-valued counterparts? Our first contribution in this work is to take stock of some recent advances in training binary neural networks and train a very strong baseline which already results in state-of-the-art results without any increase in the model size or the computational cost. Our second contribution is a method for bridging most of the remaining gap which boils down to minimizing the discrepancy between the output of the binary and the corresponding real-valued convolution. This idea is materialized in our work in two complementary ways: **Firstly**, we match the spatial attention maps computed at the output of the binary and real-valued convolutions within a teacher-student training framework (Zagoruyko & Komodakis, 2017), where the binary network is the student and the real-valued network acts as the teacher. **Secondly**, while the aforementioned approach provides an extra supervisory signal during training, at test time, the guidance provided by the real-valued network is no longer available. Hence, we further propose to use the real-valued activations of the binary network being available during inference *prior to* the binarization process to compute scale factors that are used to re-scale the activations produced *right after* the application of

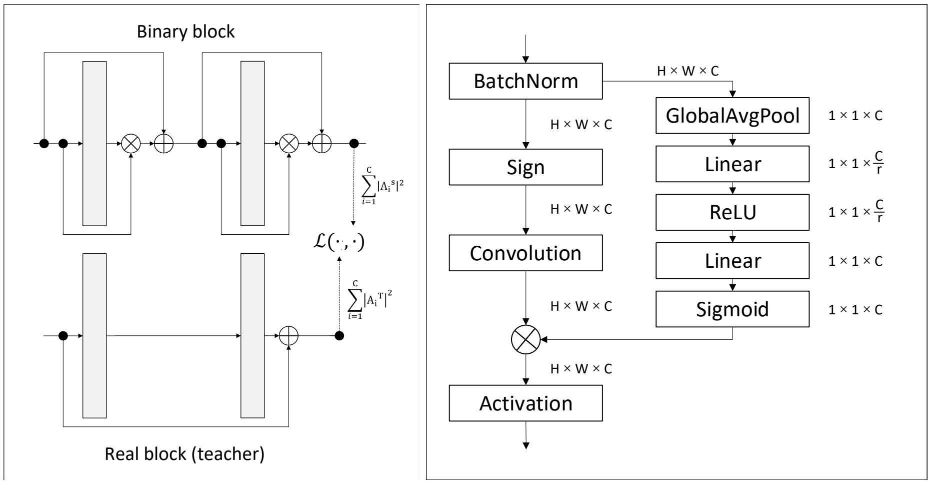
Binary block
$\sum_{i=1}^{C}|A_i^S|^2$
$\mathcal{L}(\cdot,\cdot)$
$\sum_{i=1}^{C}|A_i^T|^2$
Real block (teacher)
BatchNorm
H × W × C
GlobalAvgPool
1 × 1 × C
Sign
Linear
$1 \times 1 \times \frac{C}{r}$
ReLU
$1 \times 1 \times \frac{C}{r}$
Convolution
Linear
1 × 1 × C
Sigmoid
1 × 1 × C
Activation

Figure 1: **Left:** The proposed real-to-binary block. The diagram shows how spatial attention maps computed from a teacher real-valued network are matched with the ones computed from the binary network. Supervision is injected at the end of each binary block. See also Sub-section 4.2. **Right:** The proposed data-driven channel re-scaling approach. The left-hand side branch corresponds to the standard binary convolution module. The right-hand side branch corresponds to the proposed gating function that computes the channel-scaling factors from the output of the batch normalization. See also Sub-section 4.3.

the binary convolution. This is in line with recent works which have shown that re-scaling the binary convolution output results in large performance gains (Rastegari et al., 2016; Bulat & Tzimiropoulos, 2019) with the difference being that the computation of the scale factors in our case is data-driven, based on the real-valued activations produced by each layer prior to binarization.

Overall, we make the following **contributions**:

- We construct a very strong baseline by combining some recent insights on training binary networks, and by performing a thorough experimentation to find the most well-suited optimization techniques. We show that this baseline already achieves state-of-the-art accuracy, surpassing all previously published work on binary networks.
- We propose real-to-binary attention matching: this entails that matching spatial attention maps computed at the output of the binary and real-valued convolutions is particular suited for training binary neural networks. See Fig. 1, left and and Sub-section 4.2. We also devise a multi-stage attention matching strategy.
- We propose data-driven channel re-scaling: this entails using the real-valued activations of the binary network *prior* to the binarization process to compute the scale factors used to re-scale the activations produced *right after* the application of the binary convolution. See Fig. 1, right, and Sub-section 4.3.
- We show that our combined contributions provide, for the first time, competitive results on two standard datasets, achieving 76.2% top-1 performance on CIFAR-100, and 65.4% top-1 performance on ImageNet using a ResNet-18 –a gap bellow 3% and 5% respectively compared to their full precision counterparts– and with a negligible increase in the model size.

2 Related work

While being pre-dated by other works on binary networks (Soudry et al., 2014), the BNN algorithm (Courbariaux et al., 2016) established how to train networks with binary weights within the familiar back-propagation paradigm. The proposed training method relies on a real-valued copy of the weights, which is binarized during the forward pass, but is updated during back-propagation ignoring the binarization step. Unfortunately, BNN resulted in a staggering $\sim 28\%$ gap in top-1 accuracy compared to the full precision ResNet-18 on ImageNet.

In an attempt to bridge this gap, XNOR-Net (Rastegari et al., 2016) proposed to add a real-valued scaling factor to each output channel of a binary convolution. In fact, it has become standard for binary networks to use such scaling factors given the minimal increase in computational cost for a very large performance increase. It is worth noting that, in almost all works, the proposed binary networks do have a number of floating point operations; for example, the first convolution (a costly 7×7 kernel in ResNet), the fully connected layer as well as batch normalization layers are all real-valued. Thus, adding a small amount of real-valued operations is typically well-justified and accepted should it provide significant accuracy gains. For example, the authors of Bi-Real Net (Liu et al., 2018) argued that skip connections are fundamental for binary networks and observed that the flow of full precision activations provided by the skip connections is interrupted by the down-sample convolutions, which degrade the signal and make subsequent skip connections less effective. To alleviate this, they proposed making the downsample layers real-valued, obtaining around 3% accuracy increase in the process with a small increase in complexity.

There have been also numerous ways to improve the optimization of training binary networks, such as the use of smooth approximations of the gradient, use of PReLU instead of ReLU (Bulat et al., 2019), two-stage training which binarizes the weights first and then the activations (Bulat et al., 2019) and progressive quantization (Bulat et al., 2019; Gong et al., 2019). The work in (Wang et al., 2019) proposed to learn channel correlations through reinforcement learning in order to better preserve the sign of a convolution output. A set of regularizers are added to the loss term in (Ding et al., 2019) so as to control the range of values of the activations, and guarantee good gradient flow. Other optimization aspects, such the effect of gradient clipping or batch-norm momentum, were empirically tested in (Alizadeh et al., 2019). In Sub-section 4.1, we show how to combine many of the insights provided in the aforementioned works with standard optimization techniques to obtain a very strong baseline that achieves state-of-the-art accuracy.

While the aforementioned works either maintain the same computational cost, or increase it by a fractional amount, some other lines of research have focused instead on relaxing the problem constraints by increasing the number of convolutions per layer by a large amount (typically a factor of 2 to 8 times), see for example the ABC-Net of (Lin et al., 2017), the structure approximation of (Zhuang et al., 2019), the circulant CNN of (Liu et al., 2019), and the binary ensemble of (Zhu et al., 2019). Note that the large increase of binary operations diminishes the efficiency claim that justifies the use of binary networks in first place. Furthermore, we will show that there is still a lot of margin in order to bridge the accuracy gap prior to resorting to such approaches [1].

Our attention matching approach described in Sub-section 4.2 is somewhat related to the feature distillation approach of Zhuang et al. (2018). However, Zhuang et al. (2018) tries to match the whole feature maps of the to-be-quantized network with the quantized feature maps of a real-valued network that is trained in parallel with the to-be-quantized network. Such an approach is shown to improve training of low-bit but not binary networks. Notably, our approach based on matching attention maps is much simpler and shown to be effective for the case of binary networks.

Our data-driven channel re-scaling approach, described in Sub-section 4.3, is related to the channel re-scaling approach of XNOR-Net and that of (Xu & Cheung, 2019; Bulat & Tzimiropoulos, 2019) which propose to learn the scale factors discriminatively through backpropagation. Contrary to (Xu & Cheung, 2019; Bulat & Tzimiropoulos, 2019), our method is data-driven and avoids using fixed scale factors learnt during training. Contrary to XNOR-Net, our method discriminatively learns how to produce the data-driven scale factors so that they are optimal for the task in hand.

3 BACKGROUND

This section reviews the binarization process proposed in (Courbariaux et al., 2016) and its improved version from (Rastegari et al., 2016) which is the method of choice for neural network binarization.

[1] There is also a large amount of work on using other low-bit quantization strategies but a review of these techniques goes beyond the scope of this section.

We denote by $\mathcal{W} \in \mathbb{R}^{o \times c \times k \times k}$ and $\mathcal{A} \in \mathbb{R}^{c \times w_{in} \times h_{in}}$ the weights and input features of a CNN layer, where o and c represent the number of output and input channels, (k, k) the width and height of the convolutional kernel, and w_{in} and h_{in} represent the spatial dimension of the input features $\mathcal{A}$. In (Courbariaux et al., 2016), both weights and activations are binarized using the sign function and then convolution is performed as $\mathcal{A} * \mathcal{W} \approx \text{sign}(\mathcal{A}) \circledast \text{sign}(\mathcal{W})$ where $\circledast$ denotes the binary convolution which can be implemented using bit-wise operations.

However, this direct binarization approach introduces a high quantization error that leads to low accuracy. To alleviate this, XNOR-Net (Rastegari et al., 2016) proposes to use real-valued scaling factors to re-scale the output of the binary convolution as

$$\mathcal{A} * \mathcal{W} \approx (\text{sign}(\mathcal{A}) \circledast \text{sign}(\mathcal{W})) \odot \mathcal{K}\boldsymbol{\alpha}, \tag{1}$$

where $\odot$ denotes the element-wise multiplication, $\boldsymbol{\alpha}$ and $\mathcal{K}$ are the weight and activation scaling factors, respectively, calculated in Rastegari et al. (2016) in an analytic manner. More recently, Bulat & Tzimiropoulos (2019) proposed to fuse $\boldsymbol{\alpha}$ and $\mathcal{K}$ into a single factor Γ that is learned via backpropagation resulting in further accuracy gains.

4 METHOD

This section firstly introduces our strong baseline. Then, we present two ways to improve the approximation of Eq. (1): Firstly, we use a loss based on matching attention maps computed from the binary and a real-valued network (see Sub-section 4.2). Secondly, we make the scaling factor a function of the *real-valued* input activations $\mathcal{A}$ (see Sub-section 4.3).

4.1 BUILDING A STRONG BASELINE

Currently, almost all works on binary networks use XNOR-Net and BNN as baselines. In this section, we show how to construct a strong baseline by incorporating insights and techniques described in recent works as well as standard optimization techniques. We show that our baseline already achieves state-of-the-art accuracy. We believe that this is an important contribution towards understanding the true impact of proposed methodologies and towards assessing the true gap with real-valued networks.

Following prior work in binary networks, we based our method on the ResNet-18 architecture and apply the improvements listed below:

Block structure: It is well-known that a modified ResNet block must be adapted to provide optimal results when training binary networks. We used the widely-used setting where the operations are ordered as BatchNorm → Binarization → BinaryConv → Activation. The skip connection is to the last operation of the block (Rastegari et al., 2016).

Residual learning: We used double skip connections, as in (Liu et al., 2018).

Activation: We used PReLU (He et al., 2015), which is known to facilitate the training of binary networks (Bulat et al., 2019).

Scaling factors: We used discriminatively learnt scaling factors via backpropagation as in (Bulat & Tzimiropoulos, 2019).

Downsample layers: We used real-valued downsample layers, and found the reported large accuracy boost (around $3 - 4\%$ top-1 improvement on ImageNet) to be consistent across our experiments (Liu et al., 2018).

We used the following training strategies to train our strong baseline:

Initialization: When training binary networks, it is crucial to use a 2-stage optimization strategy. In particular, we adopted the strategy of binarizing the activations first, and then using the resulting model as initialization to train a network with both weights and activations binarized (Bulat et al., 2019).

Weight decay: Setting up weight decay carefully is surprisingly important. We use $1e - 5$ on stage 1, and set it to 0 on stage 2 (Bethge et al., 2019).

Data augmentation: For CIFAR-100 we use the standard random crop, random horizontal flip and random rotation ($\pm 15°$). For ImageNet, we found that random cropping, flipping and colour jitter augmentation worked the best. However, colour jitter is disabled for stage 2.

Mix-up: We found that mix-up (Zhang et al., 2017) is crucial for CIFAR100, while it slightly hurts performance for ImageNet – this is due to the higher risk of overfitting on CIFAR100.

Warm-up: We used warm-up for 5 epochs during stage 1, and no warm-up for stage 2.

Optimizer: We used Adam (Kingma & Ba, 2014) with a stepwise scheduler. The learning rate is set to $1e-3$ for stage 1, and $2e-4$ for stage 2. For CIFAR-100, we trained for 350 epochs, with steps at epochs 150, 250 and 320. For ImageNet, we train for 75 epochs, with steps at epochs 40, 60 and 70. Batch sizes are 256 for ImageNet and 128 for CIFAR-100.

4.2 Real-to-Binary Attention Matching

We make the reasonable assumption that if a binary network is trained so that the output of each binary convolution more closely matches the output of a real convolution in the corresponding layer of a real-valued network, then significant accuracy gains can be obtained. Notably, a similar assumption was made in (Rastegari et al., 2016) where analytic scale factors were calculated so that the error between binary and real convolutions is minimized. Instead, and inspired by the attention transfer method of Zagoruyko & Komodakis (2017), we propose to enforce such a constraint via a loss term at the end of each convolutional block between attention maps calculated from the binary and real-valued activations. Such supervisory signals provide the binary network with much-needed extra guidance, as it is well-known that backpropagation for binary networks is not as effective as for real-valued ones. By introducing such loss terms at the end of each block, gradients do not have to traverse the whole network and suffer a degraded signal.

Assuming that attention matching is applied at a set of $\mathcal{J}$ transfer points within then network, the total loss can be expressed as:

$$\mathcal{L}_{att} = \sum_{j=1}^{\mathcal{J}} \| \frac{\mathcal{Q}_S^j}{\|\mathcal{Q}_S^j\|_2} - \frac{\mathcal{Q}_T^j}{\|\mathcal{Q}_T^j\|_2} \|, \tag{2}$$

where $\mathcal{Q}^j = \sum_{i=1}^{c} |\mathcal{A}_i|^2$ and $\mathcal{A}_i$ is the $i-$th channel of activation map $\mathcal{A}$. Moreover, at the end of the network, we apply standard logit matching (Hinton et al., 2015).

Multi-stage training: We observed that teacher and student having as similar architecture as possible is very important in our case. We thus devise a multi-stage teacher-student optimization strategy that creates a sequence of teacher-student pairs that bridge the differences between the real network and the binary network in small increments:
Stage 0: The teacher is the real-valued network with the standard ResNet architecture. The student is another real-valued network, but with the same architecture as the binary ResNet-18 (e.g. double skip connection, layer ordering, PReLU, etc). Furthermore, a soft binarization (a Tanh function) is applied to the activations so that the resulting network more closely resembles a network with binary activations.
Stage 1: The network resulting from Stage 0 is used as teacher. A network with binary activations and real-valued weights is the student.
Stage 2: The network resulting from Stage 1 is the teacher. A network with binary weights and activations is the student. In this stage, only logit matching is used.

4.3 Data-driven channel re-scaling

While the approach of the previous section provides better guidance for the training of binary networks, the representation power of binary convolutions is still limited, hindering its capacity to approximate the real-valued network. Here we describe how to boost the representation capability of a binary neural network while incurring in a negligible increment on the number of real-valued operations.

Previous works have shown the effectiveness of re-scaling binary convolutions with the goal of better approximating real convolutions and in turn achieving large accuracy gains. XNOR-Net (Rastegari et al., 2016) proposed to compute these scale factors analytically while (Bulat & Tzimiropoulos, 2019; Xu & Cheung, 2019) proposed to learn them discriminatively in an end-to-end manner, showing additional accuracy gains. For the latter case, during training, the optimization aims to find a set of *fixed* scaling factors that minimize the average expected loss for the training set. We propose instead to go beyond this and obtain input-dependent scaling factors – thus, at test time, these scaling factors will *not* be fixed but rather inferred from data.

Let us first recall what the signal flow is when going through a binary block. The activations entering a binary block are actually real-valued. Batch normalization centers the activations, which are then binarized, thus losing a large amount of information. Binary convolution, re-scaling and eventually PReLU follow. We propose to use the full-precision activation signal, *prior to* the large information loss incurred by the binarization operation, to predict the scaling factors used to re-scale the output of the binary convolution channel-wise. Specifically, we propose to approximate the real convolution as follows:

$$\mathcal{A} * \mathcal{W} \approx (\text{sign}(\mathcal{A}) \circledast \text{sign}(\mathcal{W})) \odot G(\mathcal{A}; \mathcal{W}_G), \tag{3}$$

where $\mathcal{W}_G$ are the parameters of the gating function G. Such function computes the scale factors used to re-scale the output of the binary convolution, and uses the pre-convolution real-valued activations as input. Fig. 1 shows our implementation of function G. The design is inspired by Hu et al. (2018), but we use the gating function to predict ahead rather than performing a self-referential attention mechanism.

Why is this important?: An optimal mechanism to modulate the output of the binary convolution clearly should not be the same for all examples as in Bulat & Tzimiropoulos (2019) or Xu & Cheung (2019). Note that in Rastegari et al. (2016) the computation of the scale factors depends on the input activations. However the analytic calculation is sub-optimal with respect to the task at hand. To circumvent the aforementioned problems, our method learns, via backpropagation for the task at hand, to predict the modulating factors using the real-valued input activations. By doing so, more than $1/3$ of the remaining gap with the real-valued network is bridged.

4.4 COMPUTATIONAL COST ANALYSIS

Table 1 details the computational cost of the different baseline binary network methodologies. We differentiate between the number of binary and floating point operations, including operations such as skip connections, pooling layers, etc. It is possible to see that our method leaves the number of binary operations constant, and that the FLOPs increases by only a 1% of the total floating point operation count. To put this into perspective, the magnitude is similar to the operation increase incurred by the XNOR-Net with respect to its predecessor, BNN. Similarly, the double skip connections proposed in (Liu et al., 2018) adds again a comparable amount of operations.

Method	BOPS	FLOPS
BNN (Courbariaux et al., 2016)	1.695×10^9	1.314×10^8
XNOR-Net (Rastegari et al., 2016)	1.695×10^9	1.333×10^8
Double Skip ((Liu et al., 2018)	1.695×10^9	1.351×10^8
Bi-Real (Liu et al., 2018)	1.676×10^9	1.544×10^8
Ours	1.676×10^9	1.564×10^8
Full Precision	0	1.826×10^9

Table 1: Breakdown of floating point and binary operations for variants of binary ResNet-18.

5 RESULTS

In this section, we present two main sets of experiments. We used ImageNet (Russakovsky et al., 2015) as a benchmark to compare our method against other state-of-the-art approaches in Sec. 5.1. ImageNet is the most widely used dataset to report results on binary networks and, at the same time, allows us to show for the first time that binary networks can perform competitively on a large-scale dataset. We further used CIFAR-100 (Krizhevsky & Hinton, 2009) to perform a set of ablation studies (Sec. 5.2).

5.1 Comparison with the State-of-the-Art

Table 2 shows a comparison between our method and relevant state-of-the-art methods, including low-bit quantization methods.

Vs. other binary networks: Our strong baseline already comfortably achieves state-of-the art results, surpassing the previously best-reported result by about 1% Wang et al. (2019). Our full method further *improves over the state-of-the-art by* 5.5% *top-1 accuracy*. When comparing to binary models that scale the capacity of the network (second set of results on Tab. 2), only Zhuang et al. (2019) outperforms our method, surpassing it by 0.9% top-1 accuracy - yet, this is achieved using 4 times the number of binary blocks.

Vs. real-valued networks: Our method reduces the performance gap with its real-valued counterpart to $\sim 4\%$ top-1 accuracy, or $\sim 5\%$ if we compare against a real-valued network trained with attention transfer.

Vs. other low-bit quantization: Table 2 also shows a comparison to the state-of-the-art for low-bit quantization methods (first set of results). It can be seen that our method surpasses the performance of all methods, except for TTQ (Zhu et al., 2016), which uses 2-bit weights and full-precision activations - thus incurring a computational cost several times larger than ours.

	ImageNet		
Method	Bitwidth (W/A)	Top-1	Top-5
BWN (Rastegari et al., 2016)	1/32	60.8	83.0
TTQ (Zhu et al., 2016)	2/32	66.6	87.2
HWGQ (Cai et al., 2017)	1/2	59.6	82.2
LQ-Net (Zhang et al., 2018)	1/2	62.6	84.3
SYQ (Faraone et al., 2018)	1/2	55.4	78.6
DOREFA-Net (Zhou et al., 2016)	2/2	62.6	84.4
ABC-Net (Lin et al., 2017)	(1/1)×5	65.0	85.9
Circulant CNN (Liu et al., 2019)	(1/1)×4	61.4	82.8
Struct Appr (Zhuang et al., 2019)	(1/1)×4	64.2	85.6
Struct Appr** (Zhuang et al., 2019)	(1/1)×4	66.3	86.6
Ensemble (Zhu et al., 2019)	(1/1)×6	61.0	–
BNN (Courbariaux et al., 2016)	1/1	42.2	69.2
XNOR-Net (Rastegari et al., 2016)	1/1	51.2	73.2
Trained Bin (Xu & Cheung, 2019)	1/1	54.2	77.9
Bi-Real Net (Liu et al., 2018)**	1/1	56.4	79.5
CI-Net (Wang et al., 2019)	1/1	56.7	80.1
XNOR-Net++ (Bulat & Tzimiropoulos, 2019)	1/1	57.1	79.9
CI-Net (Wang et al., 2019)**	1/1	59.9	84.2
Strong Baseline (ours)**	1/1	60.9	83.0
Real-to-Bin (ours)**	1/1	**65.4**	**86.2**
Real valued	32/32	69.3	89.2
Real valued T-S	32/32	70.7	90.0

Table 2: Comparison with SOTA methods on Binary Networks on ImageNet. ** indicates the use of real-valued downsample. The second column indicates the number of bits used to represent weights and activations. Methods include low-bit quantization (upper section), and methods multiplying the capacity of the network (second section). For the latter case, the second column includes the multiplicative factor of the network capacity used.

5.2 Ablation Studies

In order to conduct a more detailed ablation study we provide results on CIFAR-100.

Multi-Stage Teacher Student: We trained a ResNet-18 full precision network to serve as the real-valued baseline, and further trained another version using ResNet-34 as its teacher. The use of teacher supervision on CIFAR-100 yields $\sim 1\%$ top-1 accuracy increase. Instead, our multi-stage teacher-student strategy yields $\sim 5\%$ top-1 accuracy gain, showing that it is a fundamental tool when training binary networks.

Performance gap: We observe that, for CIFAR-100, we close the gap with real-valued networks to about 2% when comparing with the full-precision ResNet-18, and to about 3% when comparing with full-precision using teacher supervision. The gap is consistent to that on ImageNet, where the error rate of our method increases 13% relative to a real-valued network, while on CIFAR-100 the increase is of 10% relative error.

Binary downsample: We also show that the performance increase respect to the baseline is 6.4% top-1 accuracy when using binary downsample, compared to an improvement of 6.6% when using real-valued downsample. It is also noticeable that the gating mechanism is less effective when using binary downsampling. This is reasonable given that, when using real-valued downsample, the activations have a path through skip connections that avoids binary convolutions altogether. In this way, the activation signal used by the gating function as input has better quality.

Scaling factors without attention matching: It is also remarkable that the gating module is less effective in the absence of attention matching. It seems clear from this result that both are interconnected: the extra supervisory signal is necessary to properly guide the training, while the extra flexibility added through the gating mechanism boosts the capacity of the network to mimic the attention map.

	Stage 1	Stage 2	
Method	Top-1 / Top-5	Top-1 / Top-5	ΔTop-1
Strong Baseline	69.31 / 88.70	66.32 / 88.62	–
SB + Att Trans	72.18 / 90.39	70.31 / 90.87	+3.99
SB + Att Trans + HKD	73.05 / 91.23	71.07 / 90.94	+4.75
SB + G	67.20 / 87.01	63.19 / 84.98	-3.13
SB + Multi-Stage TS	73.77 / 91.49	72.34 / 89.80	+6.02
Real-to-Bin	**74.98 / 92.16**	**72.68 / 91.57**	+6.36
Strong Baseline**	72.14 / 89.92	69.56 / 89.20	–
SB + Att Trans**	74.34 / 91.25	72.64 / 91.37	+3.08
SB + Att Trans + HKD**	75.43 / 92.15	73.93 / 91.24	+4.37
SB + G**	72.01 / 89.78	70.86 / 89.26	+1.30
SB + Multi-Stage TS**	75.72 / 92.11	74.62 / 91.79	+5.06
Real-to-Bin**	**76.49 / 92.81**	**76.15 / 92.67**	+6.59
Full Precision (our impl.)	78.28 / 93.63		
Full Precision + TS (our impl.)	79.26 / 94.38		

Table 3: Top-1 and Top-5 classification accuracy using ResNet-18 on CIFAR-100. ** indicates real-valued downsample layers. G indicates that the gating function of Sec. 4.3 is used.

6 Conclusion

In this work we showed how to train binary networks to within a few percent points of their real-valued counterpart, turning binary networks from hopeful research into a compelling alternative to real-valued networks. We did so by training a binary network to not only predict training labels, but also mimic the behaviour of real-valued networks. To this end, we devised a multi-stage attention matching strategy to drive optimization, and combined it with a gating strategy for scaling the output of binary convolutions to increase representation power of the convolutional block. The two strategies combine perfectly to boost the state-of-the-art of binary networks by 5.5% performance top-1 accuracy.

References

Milad Alizadeh, Javier Fernández-Marqués, Nicholas D. Lane, and Yarin Gal. An empirical study of binary neural networks' optimisation. In *International Conference on Learning Representations*, 2019.

Joseph Bethge, Haojin Yang, Marvin Bornstein, and Christoph Meinel. Back to simplicity: How to train accurate BNNs from scratch? *arXiv preprint arXiv:1906.08637*, 2019.

Adrian Bulat and Georgios Tzimiropoulos. XNOR-Net++: Improved binary neural networks. In *British Machine Vision Conference*, 2019.

Adrian Bulat, Georgios Tzimiropoulos, Jean Kossaifi, and Maja Pantic. Improved training of binary networks for human pose estimation and image recognition. *arXiv preprint arXiv:1904.05868*, 2019.

Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

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

Ruizhou Ding, Ting-Wu Chin, Zeye Liu, and Diana Marculescu. Regularizing activation distribution for training binarized deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Julian Faraone, Nicholas J. Fraser, Michaela Blott, and Philip H. W. Leong. SYQ: learning symmetric quantization for efficient deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Xiaofan Lin, Cong Zhao, and Wei Pan. Towards accurate binary convolutional neural network. In *Advances on Neural Information Processing Systems*, 2017.

Chunlei Liu, Wenrui Ding, Xin Xia, Baochang Zhang, Jiaxin Gu, Jianzhuang Liu, Rongrong Ji, and David Doermann. Circulant binary convolutional networks: Enhancing the performance of 1-bit dcnns with circulant back propagation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-Real Net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm. In *European Conference on Computer Vision*, 2018.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-Net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal on Computer Vision*, 115(3):211–252, 2015.

Daniel Soudry, Itay Hubara, and Ron Meir. Expectation backpropagation: Parameter-free training of multilayer neural networks with continuous or discrete weights. In *Advances on Neural Information Processing Systems*, 2014.

Ziwei Wang, Jiwen Lu, Chenxin Tao, Jie Zhou, and Qi Tian. Learning channel-wise interactions for binary convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Zhe Xu and Ray C.C. Cheung. Accurate and compact convolutional neural networks with trained binarization. In *British Machine Vision Conference*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017.

Dongqing Zhang, Jiaolong Yang, Dongqiangzi Ye, and Gang Hua. LQ-Nets: Learned quantization for highly accurate and compact deep neural networks. In *European Conference on Computer Vision*, 2018.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv*, 2016.

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

Shilin Zhu, Xin Dong, and Hao Su. Binary ensemble neural network: More bits per network or more networks per bit? In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian D. Reid. Towards effective low-bitwidth convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, and Ian Reid. Structured binary neural networks for accurate image classification and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.