Binary Neural Networks. Lossless picture quality for binary neural networks in pixel-level tasks.

A Preprint

Kirill Ovcharenko
Moscow Institute of Physics and Technology

Zharikov Ilya
Moscow Institute of Physics and Technology

Abstract

Image Super Resolution [SR] is a crucial class of image processing techniques that enhance quality of visual data. Deep Convolutional Neural Networks [DCNN] have recently shown great results in this field. However, application of DCNN on resource-limited devices remains challenging, as they demand significant amounts of memory, energy and computations. Binary neural networks [BNN] provide a promising approach to reduce computational complexity and speed up the inference of a model. To our best knowledge, there are not many papers devoted to applying BNN to SR tasks, as SR models are much more vulnerable to degradation in performance when decreasing the precision of weights, than image classification models. The paper proposes modification of a convolutional block to make it binary without performance decreasing.

Keywords Binary Neural Network · Single Image Super Resolution · Binarization · Model compression

1 Introduction

ToDo: Need to be extended in terms related works further.

Image Super Resolution [SR] aims to restore High Quality [HQ] image from corrupted Low Quality [LQ] counterpart. This task is important because of its various applications in medical imaging and surveillance instruments. In spite of the research in this field being active, it made some progress not a long ago, as some challenges were encountered. Main obstacle is desired output in SR tasks being much more diverse than input, so the model is required to do dense pixel-level prediction, hence is bound to be more complex.

Recent advances in the field of SR owe their success to Deep Neural Networks [DNN] which show state-of-the-art results in a wide range of computer vision problems, such as image classification, Semantic Segmentation etc. However, the models solving these tasks are usually complicated and demand a lot of space and computational resources, thus hindering their implementation on mobile devices, drones and other machines which are limited in GPU memory.

Lately, different methods of reducing complexity of these models were proposed. While some papers focus on pruning and knowledge distillation, other researches introduce quantization as a way to decrease memory needed.

The most extreme form of quantization is binarization. Binary Neural Networks [BNN] use only 1 bit to represent each parameter, thus drastically decreasing space demanded to store the model. Moreover, with all parameters of the model set to $\{-1, 1\}$, most of the the calculations can be conducted using XNOR and Bitcount operations. This approach seems promising, as it proposes new ways to design hardware that can help to handle and exploit complex neural networks.

However, it is obvious that BNN sacrifice precision and quality, as they have much less capacity and representational potential than Full-Precision [FP] networks. Previous works in this field propose different methods of maintaining competitive accuracy while achieving better performance. The paper Ma et al. [2019] focuses on residual block binarization, which helps to reduce a significant part of the model's parameters.

However, full-precision activations keep computational complexity of the model pretty high. ReactNet Liu et al. [2020] suggests generalized binarization and activations functions that help to shift distribution, which significantly increases representational capacity of the binary model. The BBCU Xia et al. [2022] proposed effective Convolutional Unit that can be used in any architecture that relies on residual connections. It provides much more efficient training and inference, but oversimplifies weight binarization. IR-Net Qin et al. [2020] reduces information loss by balancing weights to achieve maximum information entropy in forward propagation. After that, IR2Net Xue et al. [2022] proposes two essential components of the learning process: Information Restriction and Information Recovery. BNext Guo et al. [2022] also applies attention mechanism to obtain the key information from the full-precision activations and smooth out the loss landscape. However, last two papers investigate only the impact of these methods on performance of Image Classification models. Another way of extracting necessary information was proposed in Hu et al. [2018], where a squeeze-and-excitation block is added to every transformation to a feature map, so that it can learn dependencies between channels (which are expected to concentrate on different features). In contrast to regular approaches, the paper Zhao et al. [2020] presented a new method of attention that helps model to better get pixel-level dependencies and exhibits great results in SR task.

This paper adopts some techniques from the researches, mentioned above, and suggests further modifications of convolutional block that help to improve BNN's performance in SR tasks.

We use EDSR Lim et al. [2017] as a backbone for our modified binary block, as it doesn't require batch normalization module and shows state-of-the-art results. We use BBCU Xia et al. [2022] as a baseline for our modification as it showed prominent performance in multiple Image Restoration tasks, particularly in SR. This paper proposes several adjustments for the binary convolutional block.

Firstly, we adopt the idea of using generalized activation and sign functions from Liu et al. [2020]. That helps to achieve better performance while preserving reasonable computational demands.

Secondly, we keep the idea of updating full-precision weights with respect to binarized weights from Ma et al. [2019]. We also maintain a learnable scale factor in contrast to Xia et al. [2022], where the optimal value is used, because the former method was proven to give better results.

Finally, we advance the idea from Xue et al. [2022] and Guo et al. [2022] to restrict information from the input to increase learning productivity by implementing attention modules into our binary network. We try different ways to compute attention maps, including the methods proposed in Zhao et al. [2020] and Hu et al. [2018].

2 Problem statement

ToDo: Need to be slightly reformulated during theory week.

Let $\{(X_i, Y_i)\}_{i=1}^n$ be our image dataset, where $X_i \in \mathbb{R}^{h_i \times w_i \times 3}$ denotes the low resolution image and $Y_i \in \mathbb{R}^{H_i \times W_i \times 3}$ - the high resolution one. Considering M to be the model, SR task targets optimization of

$$Q(M) = \frac{1}{n} \sum_{i=1}^{n} f(M(X_i), Y_i) \tag{1}$$

where f represents either $PSNR$ or $SSIM$ metric, defined as:

$$PSNR(x, y) = 10 \log_{10} \left(\frac{MAX_I^2}{MSE(x, y)} \right) \tag{2}$$

Here MAX_I is the maximum valid value for pixel, MSE is mean squared error.

$$SSIM(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

μ_x denotes the mean for x, μ_y is the mean for y, σ_x is the variance for x, σ_y is the variance for y, σ_{xy} is the covariation of x and y, c_1 and c_2 - two constants depending on dynamic pixel range.

Now let $B \in \mathcal{B}$ be our binarized representation of model M. $\mathcal{B}$ is space of BNN. Assuming L to be number of layers in $B \in \mathcal{B}$, $W_l \in \{-1, 1\}^{C_{out} \times C_{in} \times K_h \times K_w}$, $l \in \{1, \dots, L\}$. Here C_{out} is the number of output channels, C_{in} is the number of input channels, K_h is the kernel height, K_w is the kernel weight.

Thus, the problem of binarization can be expressed in finding B^* as

$$B^* = \arg\min_{B \in \mathcal{B}} [Q(M) - Q(B)] \tag{4}$$

3 Basic Binary Convolutional Block modification

3.1 Baseline block

In this section we define basic binarization operations that are used to build the Binary Convolutional Block.

Let $X_t^f \in \mathbb{R}^{H \times W \times C_{in}}$ and $W_t^f \in \mathbb{R}^{K_h \times K_w \times C_{in} \times C_{out}}$ be full-precision activations and full-precision convolution weights on the t-th layer respectively. Here H and W denote the input feature map height and width, C_{in} stands for the number of input channels and C_{out} is the number of output channels. Then $X_t^b \in \{-1, 1\}^{H \times W \times C_{in}}$, $W_t^b \in \{-1, 1\}^{\times K_w \times C_{in} \times C_{out}}$ would be the binary approximations for the corresponding full-precision parameters.

When using binary parameters, convolution operation $X^b * W^b$ can be effectively performed using XNOR and Bitcount operations:

$$X_t^b * W_t^b = Bitcount(XNOR(X_t^b, W_t^b))$$

We adopt the idea of using generalized sign function from Liu et al. [2020], as it was proven to help BNNs better learn distributions of activations. Thus, the binary representations can be acquired by applying RSign function:

$$x_{i,j,k}^b = RSign(x_{i,j,k}^f) = \begin{cases} +1, & x_{i,j,k}^f > \alpha_k \\ -1, & x_{i,j,k}^f \leq \alpha_k \end{cases}, \quad i \in [0, H), j \in [0, W), k \in [0, C_{in}) \tag{5}$$

Here $x_{i,j,k}^f \in X_t^f$, $x_{i,j,k}^b \in X_t^b$ are single full-precision and binary activations respectively.

When binarizing the convolution weights we use the regular Sign function:

$$w_{i,j,k,l}^b = Sign(w_{i,j,k,l}^f) = \begin{cases} +\alpha_l, & w_{i,j,k,l}^f > 0 \\ -\alpha_l, & w_{i,j,k,l}^f \leq 0 \end{cases}, \quad i \in [0, H), j \in [0, W), k \in [0, C_{in}), l \in [0, C_{out}) \tag{6}$$

Here full-precision and binary weights are denoted as $w_{i,j,k,l}^f \in W_t^f$, $w_{i,j,k,l}^b \in W_t^b$.

We preserve the idea from Ma et al. [2019] of using learnable α_l parameter instead of the optimal value $\alpha_l = \dfrac{|W_l|}{n}$, as optimality in the optimisation task is not necessary consistent with the optimality of target loss function.

We use RPReLU Liu et al. [2020] as activation function, because it achieves better performance by shifting the negative component of input's distribution, which is important for BNNs.

RPReLU is defined as follows:

$$RPReLU(x_{i,j,k}) = \begin{cases} x_{i,j,k} - \gamma_k + \zeta_k, & x_{i,j,k} > \gamma_k \\ \beta_k(x_{i,j,k} - \gamma_k) + \zeta_k, & x_{i,j,k} \leq \gamma_k \end{cases}, \quad i \in [0, H), j \in [0, W), k \in [0, C_{in}) \tag{7}$$

Where $x_{i,j,k} \in \mathbb{R}^{H \times W \times C_{in}}$ is an element of the input feature map, $\gamma_i \in \mathbb{R}$ and $\zeta_i \in \mathbb{R}$ are learnable shifts for moving the distribution, and $\beta_i \in \mathbb{R}$ is a learnable coefficient controlling the slope of the negative part.

Previous researches displayed the importance of the residual connection in the Binary Convolution Block, especially in the SR task, so we keep it for every binary convolution to transfer the full-precision information through the block. Moreover, BBCU Xia et al. [2022] shows that activation function narrows the negative part of the residual connection, thus losing negative full-precision information. On that account, we keep the idea of moving the residual connection out of the activation function.

EDSR Lim et al. [2017] showed that applying Batch Normalization has a negative impact on quality when dealing with pixel-level tasks, such as SR. But the experiments conducted in Xia et al. [2022] show that spreading the distribution of values is necessary for BNNs. For that reason, we adopt the amplification factor from BBCU Xia et al. [2022] that helps to avoid covering the full-precision information with the binary convolution output.

Taking in consideration all the points mentioned above, the baseline block can be expressed as follows:

$$X_{t+1}^{f} = RPReLU(X_t^b * W_t^b) + kX_t^f \tag{8}$$

Here X_t^b was also acquired from the amplified activations kX_t^f, but the learnable threshold from 5 helps to filter the amplitude.

3.2 Attention modules

When the full-precision model is being binarized, it is bound to lose some representational capacity and suffer a performance decrease. Previous researches Guo et al. [2022], Xue et al. [2022] focus on applying attention mechanism to help the model to capture the most important features and dependencies. Further advancing the idea of restricting information, we suggest attention modules that help the model to extract necessary features from the input.

We propose several different attention blocks: a simple squeeze-and-excitation attention, a spatial attention and a pixel attention.

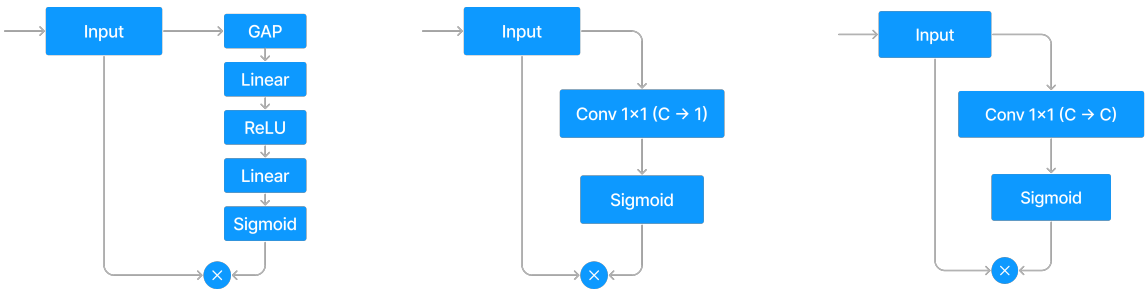
A squeeze-and-excitation block Hu et al. [2018] is depicted on 1a. It consists of Global Average Pooling (GAP), two linear layers with a non-linear activation function (ReLU) and a sigmoid function, which is defined as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

The structure of the block helps model to learn non-linear dependencies between channels, hence between different features.

Spatial attention block (1b) consists of one 1×1 convolution with 1 output channel followed by a sigmoid function. It focuses on getting a 2D attention map which helps it to learn relationships between channels for every pixel on the feature map.

Finally, pixel attention block (1c) has similar structure to the spatial attention block, except for using a 1×1 convolution with the same number of the output channels. It constructs a 3D attention map that learn connections between both the channels and the pixels. Previous works Zhao et al. [2020] proved it to be the best option for SR tasks as it helps to learn complex pixel-level dependencies.



(a) Squeeze-and-excitation attention (b) Spatial attention block (c) Pixel attention block

Рис. 1: Attention blocks

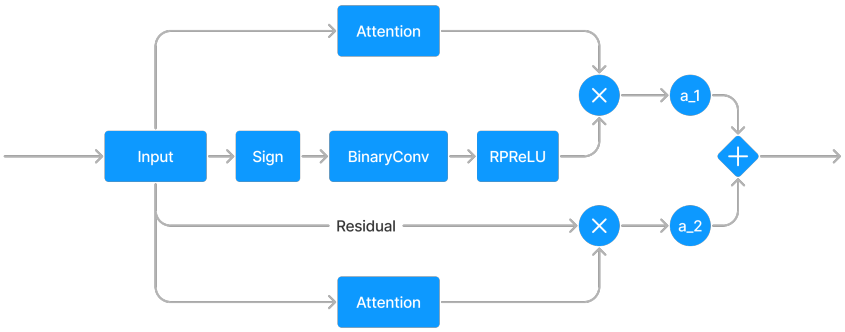


Рис. 2: Basic binary block modification

3.3 Proposed modification

In the baseline block, outputs of the binary convolution and activations from the previous layer are added to each other and have the same influence on the result. However, full-precision residual connections have information that cannot be effectively processed by the binary convolution due to its simple structure. Therefore, we need to provide a method to get the most important features from previous layer. Hence, we add an attention module to the residual branch of the convolutional block.

However, the outputs of the convolution branch can also contain unnecessary information that should not be propagated to the next layer. Thus, we add another attention block that is applied to the RPReLU output before connecting with residual information.

To give the model more freedom when deciding on the importance of information, we add scaling factors a_1 and a_2 to the binary convolution and residual branches respectively. Having these coefficients as learnable parameters help the model to propagate the most essential features to the next layers. Moreover, tuning these coefficients alleviates the need of using amplification factor, as a_1 and a_2 can maintain necessary amplitude of distribution of the residual information.

The block with all the mentioned modifications is presented on 2. Independently constructed attention maps are applied to both residual and binary branches and the outputs are connected using learnable scaling coefficients.

Having similar structure with BBCU, the block can be used in any SR model architecture with minimal effort. In the next section we implement the modified block into EDSR and prove it to show state-of-the-art results on the benchmark datasets.

4 Computational experiment

4.1 Preliminary report

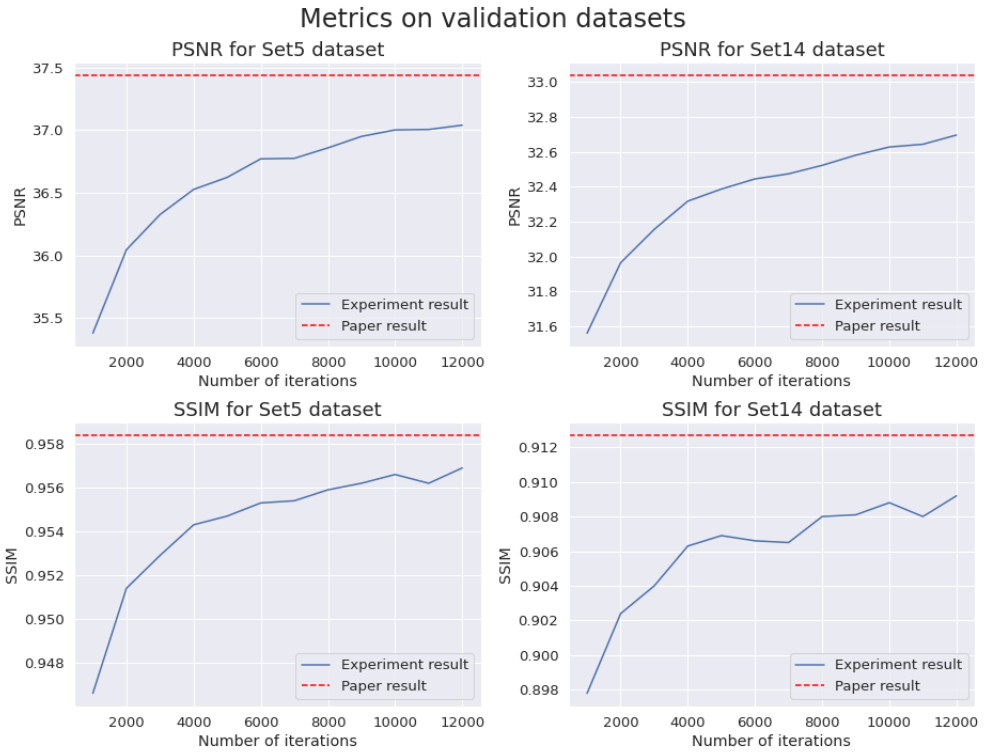
We expect our modification to show the same trend of PSNR and SSIM metrics, but the goal of the research is to increase the performance while preserving the amount of computational resources required.

Список литературы

Yinglan Ma, Hongyu Xiong, Zhe Hu, and Lizhuang Ma. Efficient super resolution using binarized neural network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019.

Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In Computer Vision–ECCV 2020: 16th European

Рис. 3: Validation metric plots for baseline (BBCU)



Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, pages 143–159. Springer, 2020.

Bin Xia, Yulun Zhang, Yitong Wang, Yapeng Tian, Wenming Yang, Radu Timofte, and Luc Van Gool. Basic binary convolution unit for binarized image restoration network. arXiv preprint arXiv:2210.00405, 2022.

Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. Forward and backward information retention for accurate binary neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2250–2259, 2020.

Ping Xue, Yang Lu, Jingfei Chang, Xing Wei, and Zhen Wei. Ir2net: Information restriction and information recovery for accurate binary neural networks. arXiv preprint arXiv:2210.02637, 2022.

Nianhui Guo, Joseph Bethge, Christoph Meinel, and Haojin Yang. Join the high accuracy club on imagenet with a binary neural network ticket. arXiv preprint arXiv:2211.12933, 2022.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.

Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel attention. In Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pages 56–72. Springer, 2020.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 136–144, 2017.