Supplementary Material: Improving Accuracy of Binary Neural Networks using Unbalanced Activation Distribution

Hyungjun Kim[1]   Jihoon Park[1]   Changhun Lee[1]   Jae-Joon Kim[1,2]
[1]Department of Convergence IT Engineering, [2]Graduate School of Artificial Intelligence
Pohang University of Science and Technology (POSTECH), Korea
{hyungjun.kim, jihoon.park, changhun.lee, jaejoon}@postech.ac.kr

S1 Experimental setup

In this work, we used the vgg-small model as a baseline network for various experiments. The network consists of 4 convolution layers with 64, 64, 128, 128 output channels and 3 dense layers with 512, 512, and 10 output neurons. A batch normalization(BN) layer is placed between the convolution layer and the activation layer following the typical BNN block structure. Max pooling layer was used after each of the latter three convolution layers.

In Sec.3.1, to evaluate the effect of shifting the activation function along the x- or y-axis and that of increasing the range of hardtanh function, we used the vgg-small model with binary weight and full-precision activation. In Sec.3.2 and 3.4, we used the BNN version of the vgg-small model (binary weight and binary activation). For binary weight, we used the channel-wise high precision scaling factor following [4]. Neither weight decay nor weight clipping were used in this work. We trained both the full precision and binary models for 200 epochs using Adam optimizer [1] with an initial learning rate of 0.01. We used a batch size of 256, and scheduled the training using cosine annealing method [3] with 5 epochs of warmup.

In Sec.3.2.2, we tested two additional model architectures on MNIST dataset. The 2-layer MLP model has two fully connected layer with 512 and 10 output channels each. The model structure is as follows: FC1 - BN1 - BinAct - FC2 - BN2. The network is simplified as much as possible in order to rule out any effect other than threshold shifting of the single binary activation. The binary LeNet-5 has the network structure adopted from [2] and has an additional BN layer prior to every binary activation. We trained the networks for 30 epochs without warmup. All other hyperparameters are set the same as the ones used in the experiments conducted on CIFAR-10. We also investigated the sensitivity to the optimizer using SGD with momentum. We used an initial learning rate of 0.1 and momentum of 0.9 in this case.

In Sec.3.3, we evaluated several BNN models for ImageNet dataset. We followed the training recipe of the original authors for each model. We only modified the activation function to apply the proposed threshold shifting technique.

In Sec.3.5, we used ResNet-20 model to examine the effect of the additional activation function. The model architecture is described in Fig.9a. We used the same optimizer and training recipe as in Sec.3.2.

S2 BN bias initialization shifting as an alternative to the threshold shifting

As described in Sec.3.4.1, the threshold of binary activation function can be absorbed to the BN bias term in case the binary activation layer follows the BN layer. Therefore, the proposed threshold shifting technique can also be implemented by shifting initialization values of the BN bias instead. It can be easily seen that the threshold value can be merged into the bias term of the previous BN layer by simply subtracting the value from BN bias. Since the threshold value is fixed during the network optimization, the term can be merged prior to training, namely BN bias initialization. Instead of using a positive threshold value for binary activation, we can initialize the BN bias with the negative of that value to produce an unbalanced activation distribution in the network. Therefore, the proposed threshold shifting technique does not require any additional computation or resource during training and inference.

References

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

[2] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[3] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[4] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.