6. Appendix

6.1. Energy cost for different types of binarization

We adopt a convolutional layer from VGG-16 for ImageNet to estimate the computation energy cost in Table 2. In the chose layer, both input and output channels are 256; both input and output feature maps are 56x56; the kernels are 3x3 with stride 1. To estimate the computational energy consumption for BNN [22], XNOR-Net [38] and ABC-Net [29], we first compute the number of XNORs, counts, fixed-point multiplications and additions for each of the binarization methods, and then add the energy consumption for all the operations together. For BNN and XNOR-Net, the number of XNORs is $C_{in}C_{out}HWK_hK_w$, where C_{in}, C_{out}, H, W, K_h, K_w are input channels, output channels, output feature map height and width, and kernel height and width, respectively. For ABC-Net the number of XNORs is $MNC_{in}C_{out}HWK_hK_w$ where M and N are the number of bases for weights and activations, respectively. In addition to the XNOR operations, BNN also needs counts and comparators, where the number of counts is roughtly the same as XNORs, and the number of comparators is $C_{out}HW$. XNOR-Net also needs $2C_{out}HW$ fixed-point multiplications and $C_{out}HWK_hK_w$ additions, as shown in Fig. 9, since the multiplications within the convolution between A^{avg} and k can be combined with the following scaling operation. ABC-Net needs approximately $MNC_{out}HW$ multiplications and additions as shown in Fig. 10.

6.2. XNOR-Net and ABC-Net blocks

The basic blocks of XNOR-Net and ABC-Net are shown in Fig. 9 and Fig. 10.

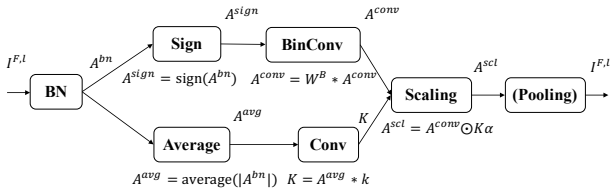

Figure 9: Basic block for XNOR-Net [38].

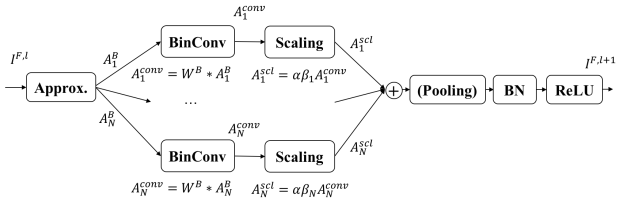

Figure 10: Basic block for ABC-Net [29].

6.3. Network structure for CIFAR-100

The basic block for ResNet-based BNN is shown in Fig. 11. Compared to the full-precision resnet, we add two convolutional layers, BN3 and BN4, to maintain a stable activation flow. For BNN-DL, the distribution loss is still applied to the activations prior to two sign activation functions.

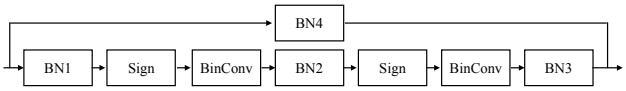

Figure 11: Basic block for ResNet-based BNN [29].

The network structure for CIFAR-100 is: xC-xB-xB-$2x$B-$2x$B-$4x$B-$4x$B-$8x$B-$8x$B-GP-100L, where xC indicates a convolutional layer with x filters, xB indicates a basic block with x filters for each convolutional layers, GP means global pooling, and xL means a linear layer with x output neurons. All the convolutional layers use 3×3 filter sizes. The first convolutional layers within the 2nd, 3rd and 4th blocks use stride 2 to reduce the feature map sizes, while the other convolutional layers use stride 1. We vary x from $\{128, 192, 256\}$.