A. MicroNet-C10 & MicroNet-C100 Networks

The MicroNet-C10 and MicroNet-C100 networks were designed for the CIFAR-10 and CIFAR-100 datasets, respectively. They share the same architecture described in Table A.1, which consists of three sections of layers. The first section is represented by the input layer or "Stem Convolution". The next section has three stages, each one containing identical building blocks, whose elements are depicted in Figure A.1. This block was designed by introducing the building blocks PyramidNet [33] in the ResNet-44 architecture [34]. The third section consists of a global average-pooling layer followed by a fully-connected layer. Finally, as an important remark, when applying the Entropy-Constrained Trained Ternarization (EC2T) approach, the first and last layers are not quantized.

Table A.1. Architecture of MicroNet-C10 and MicroNet-C100 networks, where d and w are scaling factors for the networks' depth and width, respectively. For the baseline neworks (i.e., before applying compound-model-scaling), $d = w = 1$. The number of classes, $n_{classes}$, corresponds to 10 for CIFAR-10 and 100 for CIFAR-100.

Stage	**Operation**	**Resolution**	**Output Channels**	**Repetitions**
	Stem Convolution (3×3) + BN & ReLU	32×32	$16 \times w$	1
1	Building Block	32×32	$16 \times w$	$7 \times d$
2	Building Block	16×16	$32 \times w$	$7 \times d$
3	Building Block	8×8	$64 \times w$	$7 \times d$
	ReLU & Global Avg. Pooling	8×8	$64 \times w$	1
	Fully-Connected	1×1	$n_{classes}$	1

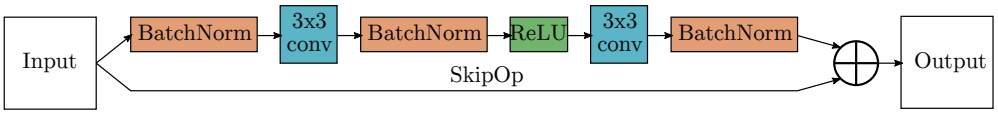

Figure A.1. Building block for the baseline models, MicroNet-C10 and MicroNet-C100.

B. Efficient Storage of Sparse & Ternary Weight Matrices

In addition to the trainable network parameters, we count those values that are needed to reconstruct the model from sparse matrix formats, i.e., binary masks or indices. Specifically, full-precision parameters (32-bits) count as one, while quantized parameters (with less than 32-bits) as a fraction of a parameter. For instance, a binary mask element counts as 1/32 with respect to a full-precision (32-bit) parameter.

If Compressed-Entropy-Row(CER)/Compressed-Sparse-Row (CSR) formats are not applied, a ternary convolution layer of size $\mathcal{N}K^2\mathcal{M}$ consists of two binary masks as illustrated in Figure B.1. One mask indicates the location of the centroid values (see Figure B.1b), while the other describes the sign of those values (see Figure B.1c). Thus, the parameter count for these masks is $1/32 \times \mathcal{N}K^2\mathcal{M}$ and $1/32 \times \sigma\mathcal{N}K^2\mathcal{M}$, respectively. In this notation, $\mathcal{N}$ is the number of effective input channels, K the kernel size, $\mathcal{M}$ the number of effective output channels, and $\sigma = 1 - \text{sparsity}$, with $\sigma \in [0, 1]$. The effective number of channels is computed as the original number of channels minus the number of channels pruned by the Entropy-Constrained Trained Ternarization (EC2T) approach. To calculate the layers' sparsity, we exclude the pruned channels. The third matrix in Figure B.1, uses two 16-bit numbers to represent the centroid values. Thus, they count as a single full-precision (32-bit) parameter (Figure B.1d). For the batch normalization layers, we add a 16-bit value (bias) per effective output channel. Therefore, their corresponding parameter count is $\mathcal{M}/2$.

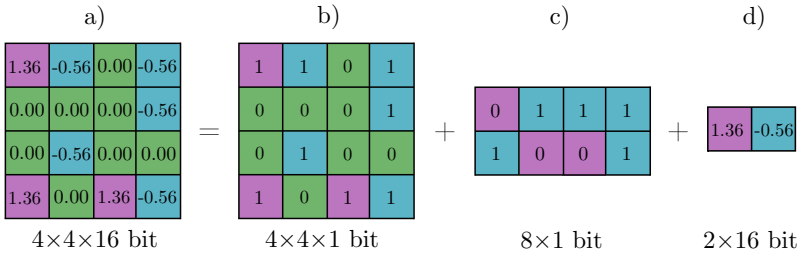

Figure B.1. Efficient storage of sparse and ternary weight matrices.