Appendices

Parameter encoding for efficient storage Since the off-chip memory access is the bottleneck in terms of both energy consumption and delay, we take the similar weight encoding strategy as stated in [6]. As seen in Fig. 5c and Fig. 5d, different REL still shares the similar weight pattern owing to our ternarization scheme, which makes it possible to encode the weight for further model compression. Instead of directly store the ternarized weight for each REL, we encode and store the backend weight system. Thus, during the inference period, such encoded weights can be converted to the specific ternarized value for each REL.

We take our network ternarization scheme with one REL (T_{ex}=2) as an example. As discussed in Section 3.2, the network tenarization is equivalent to a multi-threshold function that divides the element in $\boldsymbol{w}$ into levels of $\gamma = \{-\alpha - \alpha_r, -\alpha, 0, \alpha, \alpha + \alpha_r\}$, with thresholds $\{-b, -a, a, b\}$. The weight levels seperated by $\{-b, -a, a, b\}$ can be encoded as 3-bit weight code $c \in \{001, 010, 000, 101, 110\}$ respectively, thus the compressed weight can be stored in the format of $\boldsymbol{w} \in c^{kh \times kw \times p \times q}$, where kh, kw, p, q is the weight tensor dimension to denote kernel height, kernel width, input channel, output channel respectively. For a N-bit weight code c, we assign the MSB as the *Sign bit* (c_{s}) and the rest number of bits as the *weight allocation index* (c_{wai}). $c_{\text{s}} = 0$ for negative weight and $c_{\text{s}} = 1$ for positive weight, and all bits of c are 0 for weight in zero.

During the computation, a weight conversion is required to format the encoded weight w^l in l-th layer into designated t-th RELs $w_r^{l,t}$ into ternary format. Note that, t is in the format of bit-string. The conversion process can be mathematically described as:

$$c_{\text{s}} = c[N-1];\ c_{\text{wai}} = c[N-2:0] \tag{6}$$

$$w_r^{l,t} = \begin{cases} +1, & \text{if } c_{\text{s}} = 1 \text{ and } c_{\text{wai}} \geq t \\ -1, & \text{if } c_{\text{s}} = 0 \text{ and } c_{\text{wai}} \geq t \\ 0, & \text{if } c_{\text{wai}} < t \text{ or } c = 0 \end{cases} \tag{7}$$

Taken REL with $T_{\text{ex}} = 2$ as example to explain the aforementioned conversion. A weight element encoded as $c = 101$ needs to be converted into tenary format for *Conv* (indexed by $t1$=01) and its corresponding REL *Conv*$_r$ (indexed by $t2$=10). Since c_{wai} is 01 where $t1 = c_{\text{wai}} < t2$, the ternary weight value is +1 in *Conv* and 0 in *Conv*$_r$ respectively.

Such parameter encoding scheme can significantly reduce the model size further. For example, the non-encoded ternarized model with two RELs ($t_{ex} = 2$) requires 6-bit, while the encoded version is merely 3-bit.