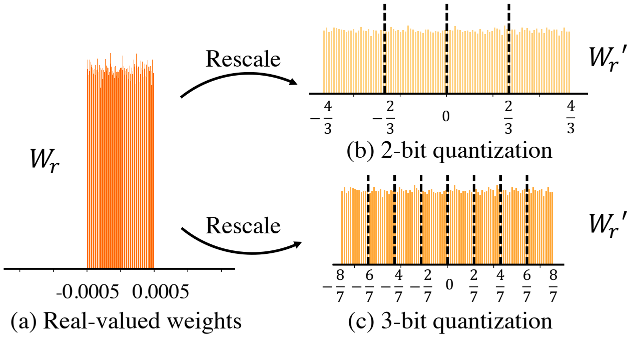
Nonuniform-to-Uniform Quantization: Towards Accurate Quantization via Generalized Straight-Through Estimation - Appendix

Zechun Liu[1,2,4] Kwang-Ting Cheng[1] Dong Huang[2] Eric Xing[2,3] Zhiqiang Shen[2,3]
[1]Hong Kong University of Science and Technology [2]Carnegie Mellon University
[3]Mohamed bin Zayed University of Artificial Intelligence [4]Reality Labs, Meta Inc.
{zliubq, timcheng}@ust.hk epxing@cs.cmu.edu {dghuang,zhiqians}@andrew.cmu.edu

In this appendix, we provide details omitted in the main text, including:

- Section A: Illustration and more details about the proposed Entropy Preserving Weight Regularization.
- Section B: Visualization of learned parameters.
- Section C: The results of keeping down-sampling layers to be real-valued in ResNet structures.

A. Entropy Preserving Weight Regularization



(a) Real-valued weights

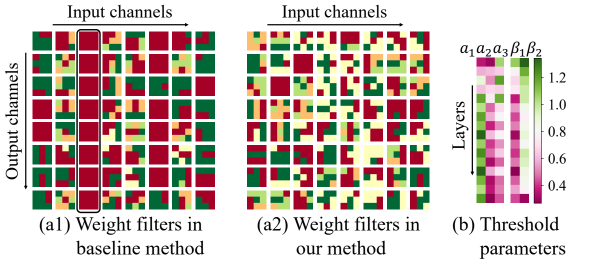
(b) 2-bit quantization

(c) 3-bit quantization

Figure A. Illustration of the proposed entropy preserving weight regularization.

The weight regularization function proposed in the main paper: $W^{r\prime} = \frac{2^{(n-1)}}{2^n-1}\frac{|W^r|}{||W^r||_{l1}}W^r$ aims to rescale the real-valued weights for preserving information entropy in the corresponding quantized weights. Specifically, $\frac{|W^r|}{||W^r||_{l1}}$ scales the W^r to have the absolute mean value equal to 1. When the real-valued weights are initialized as uniformly and symmetrically distributed [1,2], $\frac{|W^r|}{||W^r||_{l1}}W^r$ will be evenly distributed in $[-2, 2]$. The factor $\frac{2^{(n-1)}}{2^n-1}$ further spread the real-valued weight distribution to $[-\frac{2^n}{2^n-1}, \frac{2^n}{2^n-1}]$, for which, the corresponding quantized weights after the quantization function $F_Q = round((Clip(-1, W^{r\prime}, 1) + 1) \times \frac{2^n-1}{2}) \times \frac{2}{2^n-1} - 1$ will be approximately uniformly quantized to 2^n levels as shown in Fig. A. During training, the real-valued weight distributions are not always uniform, in which case, regularization helps to better distribute the weights. After training, this regularization factor can be calculated offline from the optimized weights and be absorbed by the BatchNorm layers (if used) after the quantized convolutional layers as mentioned in [3].

B. Learned Parameters Visualization



(a1) Weight filters in baseline method

(a2) Weight filters in our method

(b) Threshold parameters

Figure B. Quantized weights and learnable thresholds visualization

We visualize the optimized weights in the trained 2-bit ResNet-18. As shown in Fig. B, many 3 × 3 weight matrices learn the same value in the baseline method, which can hardly extract useful features. In contrast, this phenomenon is much rarer with the proposed weight regularization. Specifically, for 2-bit case, 4.37% of quantized 3×3 weight matrices contain the same values in baseline method, while this number is reduced to 1.69% in our method. Further, in Fig. B (b), the learned threshold parameters in the quantized ResNet-18 network have clear patterns. The parameters in first convolutional layers of the residual blocks (*i.e.*, odd rows in Fig. B (b)) often have larger values in the threshold intervals a, and smaller values in the first scaling factors β_1. We deem it is because the real-valued activations in these layers are the summation of the residual connection and the previous layer output, thus are larger in magnitude, for which, larger a and smaller β_1 are learned to better represent these activations in fixed bits.

C. Results without Quantizing Downsampling Layers

In previous quantization works, there is a practice [4, 5] of keeping the down-sampling layers to be full-precision and quantizing the rest of the convolutional layers. We follow these studies and conduct experiments on ResNet. As shown in Table A, for lower bits, real-valued 1x1 downsampling layers can boost the accuracy for ∼0.3%, while this effect becomes marginal for higher bits.

Table A. Accuracy comparison of quantizing the downsampling layers in ResNet. Both weights and activations are quantized to 2-bit 3-bit or 4-bit. * denotes keeping the weights and activations to be full-precision in 1×1 downsampling layers and quantizing all the remaining convolutional and fully-connected layers except the first and the last one.

Network	Method	2-bit		3-bit		4-bit	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
ResNet-18	N2UQ	69.4	88.4	71.9	90.5	72.9	90.9
	N2UQ*	**69.7**	**88.9**	**72.1**	**90.5**	**73.1**	**91.2**
ResNet-34	N2UQ	73.3	91.2	75.2	92.3	76.0	92.8
	N2UQ*	**73.4**	**91.3**	**75.3**	**92.4**	**76.1**	**92.8**
ResNet-50	N2UQ	75.8	92.3	77.5	93.6	78.0	93.9
	N2UQ*	**76.4**	**92.9**	**77.6**	**93.7**	**78.0**	**94.0**

References

[1] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 1

[3] Zechun Liu, Wenhan Luo, Baoyuan Wu, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Binarizing deep network towards real-network performance. *International Journal of Computer Vision*, pages 1–18, 2018. 1

[4] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*, pages 722–737, 2018. 2

[5] 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. 2