Differentiable Soft Quantization: Bridging Full-Precision and Low-Bit Neural Networks

Ruihao Gong[1,2], Xianglong Liu[1*], Shenghu Jiang[2], Tianxiang Li[2,3], Peng Hu[2], Jiazhen Lin [2], Fengwei Yu[2], Junjie Yan[2]

[1]State Key Laboratory of Software Development Environment, Beihang University, [2]SenseTime Group Limited, [3]Beijing Institute of Technology

ICCV 2019 Seoul, Korea

Introduction

Background: Hardware-friendly network quantization (e.g., binary/uniform quantization) can efficiently accelerate the inference and meanwhile reduce memory consumption of the deep neural networks. But it faces unstable training problem and severe performance degradation. Given the bit width b and the floating-point activation/weight x following in the range (l, u), two existing hardware-friendly quantization functions can be formulated as:

$$\text{Binary Quantization: } Q_B(x) = \text{sgn}(x) = \begin{cases} +1, & \text{if } x \geq 0, \\ -1, & \text{otherwise.} \end{cases} \quad (1)$$

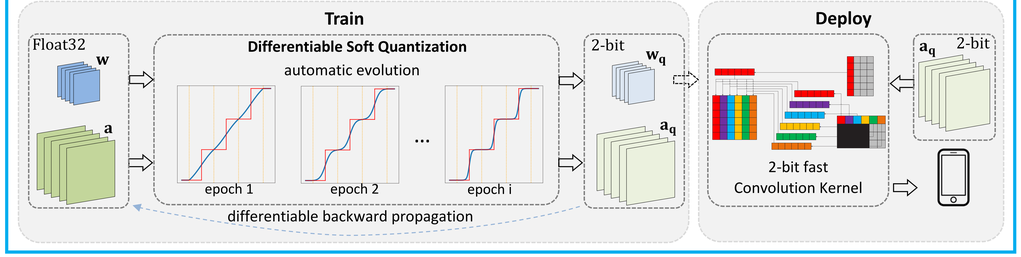
$$\text{Uniform Quantization: } Q_U(x) = \text{round}(\frac{x}{\Delta})\Delta, \ \ \Delta = \frac{u-l}{2^b-1}, \quad (2)$$

Key Contributions: We propose **Differentiable Soft Quantization (DSQ)** to well approximate the standard binary and uniform quantization process, which owns the following advantages:

- **Novel quantization**. The proposed DSQ function can well approximate the standard binary and uniform quantization in an evolution way.
- **Easy convergence**. The rectification with DSQ helps the backward propagation becomes more consistent with the forward pass.
- **Balanced loss**. DSQ jointly determines the clipping range and approximation of the quantization, and thus balances the quantization loss including clipping error and rounding error.
- **High efficiency**. DSQ can be implemented based on our fast computation kernels, and its inference speed surpasses most open-source high performance inference frameworks.
- **Strong flexibility**. DSQ is compatible with the binary or uniform quantization methods, easy to deploy in state-of-the-art network structures and able to get further accuracy improvement.

Framework

Overview: During training, we apply piecewise DSQ to redistribute the data and make it automatically evolve in each epoch to behave more like uniform quantization. After training, the piecewise DSQ can completely convert to the hard uniform quantization by sign operation, ensuring an easy and efficient deployment on resource limited devices.



DSQ Function

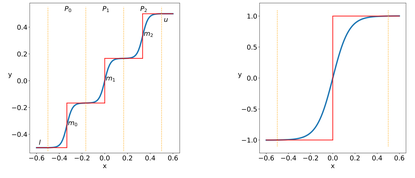
Formulation: Given the bit width b and the floating-point activation/weight x following in the range (l, u), the original range (l, u) is divided into $2^b - 1$ intervals $\mathcal{P}_i, i \in (0, 1, \ldots, 2^b - 1)$, and $\Delta = \frac{u-l}{2^b-1}$ is the interval length. The DSQ function handles the point x falling in different intervals $\mathcal{P}_i$:

$$m_i = l + (i + 0.5)\Delta \text{ and } s = \frac{1}{\tanh(0.5k\Delta)}. \quad (3)$$

$$\varphi(x) = s \tanh\left(k(x - m_i)\right), \quad \text{if } x \in \mathcal{P}_i, \quad (4)$$

$$Q_S(x) = \begin{cases} l, & x < l, \\ u, & x > u, \\ l + \Delta\left(i + \frac{\varphi(x)+1}{2}\right), & x \in \mathcal{P}_i \end{cases} \quad (5)$$

where the coefficient k determines the shape of DSQ. The larger k is, the more the asymptotic function behaves like the standard quantization function.



(a) Piecewise DSQ

(b) Binary DSQ

Figure 1: Curves of differentiable soft quantization function for (a)uniform quantization, (b) binary quantization, a special case of (a)

Auto Evolution

Definition of α:

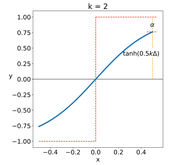
$$\alpha = 1 - \tanh(0.5k\Delta) = 1 - \frac{1}{s}. \quad (6)$$

Evolution of α:

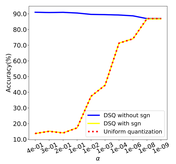
$$\min_{\alpha} \mathcal{L}(\alpha; x, y) \quad \text{s.t. } ||\alpha||_2 < \lambda \quad (7)$$

$$\frac{\partial y}{\partial \alpha} = \begin{cases} 0, & x < l, \\ 0, & x > u, \\ \frac{\partial Q_S(x)}{\partial \alpha}, & x \in P_i \end{cases} \quad (8)$$

$\alpha \to$ **0, DSQ** $\to$ **standard quantization.**



(a) Characteristic of α



(b) Effect of α

Figure 2: The characteristic variable α in DSQ.

Optimization of clipping value:

$$\frac{\partial y}{\partial l} = \begin{cases} 1, & x < l, \\ 0, & x > u, \\ 1 + q\frac{\partial \Delta}{\partial l} + \frac{\Delta}{2}\frac{\partial \varphi(x)}{\partial l}, & x \in P_i \end{cases} \quad (9)$$

$$\frac{\partial y}{\partial u} = \begin{cases} 0, & x < l, \\ 1, & x > u, \\ q\frac{\partial \Delta}{\partial u} + \frac{\Delta}{2}\frac{\partial \varphi(x)}{\partial u}, & x \in P_i \end{cases} \quad (10)$$

where

$$q = i + \frac{1}{2}(\varphi(x) + 1). \quad (11)$$

Efficient Low-bit Implementation

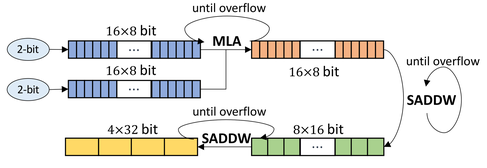


Figure 3: Data flow in our fast 2-bit ARM GEMM implementation.

Table 1: Ratio of time for calling MLA (8-bit register) and SADDW (16-bit register) with respect to different number of quantized bits.

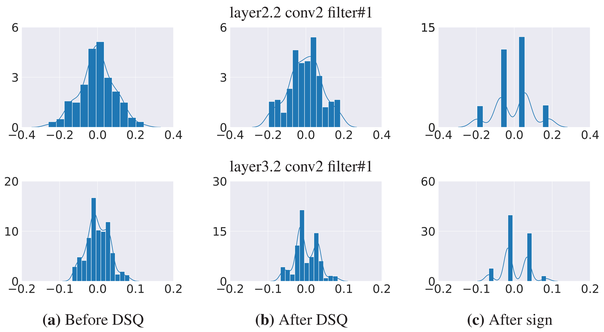
b	2	3	4
MLA/SADDW	31/1	7/1	1/1

Table 2: Comparison of time cost of ResNet-18 on Raspberry Pi 3B using different inference frameworks with different bits (single thread).

	DSQ 2-bit	NCNN 8-bit [6]
time (ms)	**551.22**	935.51

* NCNN was tested with commit d263cd5 on 2019.3.15.

Analysis of DSQ



(a) Before DSQ **(b)** After DSQ **(c)** After sign

Figure 4: Rectification: Data distribution before and after DSQ (2-bit).

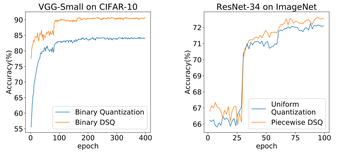
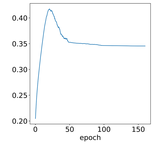
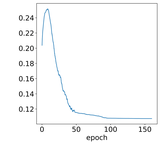


Figure 5: Training quantized model with/without DSQ.



(a) α for activations



(b) α for weights

Figure 6: Automatic evolution of α during training.

Experiments

Table 3: Ablation study on 2-bit uniform quantization.

Method	Top-1 (%)	Top-5 (%)
Standard Quantization	86.63	99.35
Fixed α	86.95	99.50
Learnt α	87.25	99.49
Learnt α, l, u	**88.44**	**99.50**

Table 4: Comparison of 1-bit quantized models on CIFAR-10.

Model	Method	Bit-Width (W/A)	Accuracy (%)
VGG-Small	FP	32/32	91.65
	BNN [3]	1/1	89.90
	XNOR [7]	1/1	89.80
	Ours	1/1	**91.72**
ResNet-20	FP	32/32	90.78
	DoReFa [11]	1/1	79.30
	Ours	1/1	**84.11**
	DoReFa [11]	1/32	90.00
	LQ-Net [10]	1/32	90.10
	Ours	1/32	**90.24**

Table 5: Comparison of different quantized models on ImageNet.

Model	Method	Bit-Width (W/A)	Accuracy (%)
ResNet-18	FP	32/32	69.90
	BWN [7]	1/32	60.80
	HWGQ [1]	1/32	61.30
	TWN [4]	2/32	61.80
	Ours	1/32	**63.71**
	PACT [2]	2/2	64.40
	LQ-Net [10]	2/2	64.90
	Ours	2/2	**65.17**
	ABC-Net [5]	3/3	61.00
	PACT [2]	3/3	68.10
	LQ-Net [10]	3/3	68.20
	Ours	3/3	**68.66**
	BCGD [9]	4/4	67.36†
	Ours	4/4	**69.56**†
ResNet-34	FP	32/32	73.80
	LQ-Net [10]	2/2	69.80
	Ours	2/2	**70.02**
	ABC-Net [5]	3/3	66.70
	LQ-Net [10]	3/3	71.90
	Ours	3/3	**72.54**
	BCGD [9]	4/4	70.81†
	Ours	4/4	**72.76**†
Mobile-NetV2	FP	32/32	71.87
	PACT [2, 8]	4/4	61.40
	Ours	4/4	**64.80**

* The † represents the results of full quantization for activations and weights across all convolution layers.

References

[1] Z. Cai, X. He, J. Sun, and N. Vasconcelos. Deep learning with low precision by half-wave gaussian quantization. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[2] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

[3] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Binarized neural networks. In *Advances in Neural Information Processing Systems 29*, pages 4107–4115. Curran Associates, Inc., 2016.

[4] F. Li, B. Zhang, and B. Liu. Ternary weight networks. *arXiv preprint arXiv:1605.04711*, 2016.

[5] X. Lin, C. Zhao, and W. Pan. Towards accurate binary convolutional neural network. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 345–353. Curran Associates, Inc., 2017.

[6] nihui et al. Ncnn. `https://github.com/Tencent/ncnn`, 2017.

[7] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. *Lecture Notes in Computer Science*, page 525–542, 2016.

[8] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han. Haq: Hardware-aware automated quantization. *arXiv preprint arXiv:1811.08886*, 2018.

[9] P. Yin, S. Zhang, J. Lyu, S. Osher, Y. Qi, and J. Xin. Blended coarse gradient descent for full quantization of deep neural networks. *Research in the Mathematical Sciences*, 6(1):14, 2019.

[10] D. Zhang, J. Yang, D. Ye, and G. Hua. Lq-nets: Learned quantization for highly accurate and compact deep neural networks. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[11] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.