Lattice Quantization

1st Clément Metz
List
CEA
Palaiseau, France
clement.metz@cea.fr

2nd Thibault Allenet
List
CEA
Palaiseau, France
thibault.allenet@cea.fr

3rd Johannes Thiele
Axelera.ai
Zurich, Switzerland
johannes.c.thiele@gmail.com

4th Antoine Dupret
Leti
CEA
Palaiseau, France
antoine.dupret@cea.fr

5th Olivier Bichler
List
CEA
Palaiseau, France
olivier.bichler@cea.fr

***Abstract*—Post-training quantization of neural networks consists in quantizing a model without retraining nor hyperparameter search, while being fast and data frugal. In this paper, we propose LatticeQ, a novel post-training weight quantization method designed for deep convolutional neural networks (DCNNs). Contrary to scalar rounding widely used in state-of-the-art quantization methods, LatticeQ uses a quantizer based on lattices – discrete algebraic structures. LatticeQ exploits the inner correlations between the model parameters to the benefit of minimizing quantization error. We achieve state-of-the-art results in post-training quantization. In particular, we achieve ImageNet classification results close to full precision on Resnet-18/50, with little to no accuracy drop for 4-bit models. Our code is available here, and a more thorough version of the paper here.**

***Index Terms*—Artificial Intelligence, Neural networks, Quantization, Post-training**

I. Introduction

Post-training quantization (PTQ) can be critical for rapid deployment of neural network models on embedded targets. We introduce a post-training quantization technique for DCNNs, which achieves state-of-the-art classification performance under various data availability hypotheses. Our method relies on a new quantizer that uses linear correlations between the parameters of convolution layers to minimize quantization error.

II. Preliminary observations

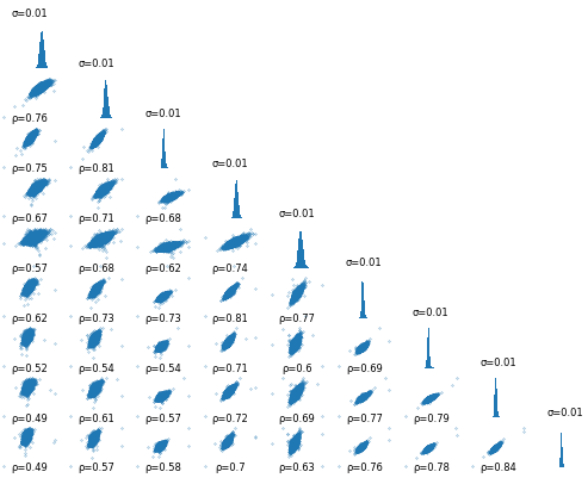

Fig. 1: Correlation diagram of layer 4.0 conv2 kernels (3×3).

On Figure 1, we plot in row i and column j the points (w_i, w_j) for each filter $f = (w_1, ..., w_9)$ in a chosen 3×3 convolution layer of a Resnet50. On the long diagonal, we plot the histogram of w_i. Kernel weights are strongly correlated. From this observation, we justify the main assumption of our method: a quantizer "shaped as a parallelogram" is more data efficient than a uniform quantizer "shaped as a square" (see Figure 2).

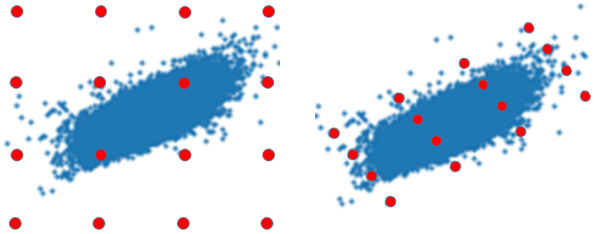

Fig. 2: Uniform quantization (left) and lattice quantization (right). Red dots are quantization points, blue dots are FP weights.

III. Related work in post-training quantization

[2] introduces bias correction and per-channel bit allocation. [3] designs a quantizer to minimize the MSE loss of the quantization operation. More recent approaches use a few samples of data. [5] tries to optimize weights rounding. [7] proceeds by bit optimization. [4] notices that blockwise optimization yields better results than usual layerwise optimization. Although degradations are still observed when quantizing to low precisions.

IV. Methods

A. Quantization

In order to quantize the weights, LatticeQ uses lattices, which are algebraic structures that discretize the notion of vector space. Each lattice has a basis, meaning that each point of the lattice can be written as an integer linear combination of the vectors of this basis. This integer linear combination is the encoding of LatticeQ. Thus, our quantization points are of the following form:

Let Λ be a lattice, and $\mathcal{B} = (\mathbf{b_i})_{1 \leq i \leq n} \in \mathbb{R}^n$ a basis of Λ. Given b the bitwidth, our quantization set is $Q = \{q \in \mathbb{R}^n, q = \sum_{i=1}^{n} f_i \mathbf{b_i},\ \forall i \in \{1, ..., n\},\ -2^{b-1} \leq f_i \leq 2^{b-1} - 1\}$.

The quantization process for a 3×3 layer is the following: we flatten the weights and group them by blocks of 3. Then, using the quantization basis, we search for the nearest quantization point to this block. The vector found is the quantization point

978-3-9819263-7-8/DATE23/© 2023 EDAA

for this block. For scalar quantization, the quantization operation relies on a simple round function, but to quantize on any lattice means to solve the Closest Vector Problem (CVP), which is finding the closest lattice point to a real vector: "Given $x \in \mathbb{R}^n$ and Λ a lattice of $\mathbb{R}^n$, find $\lambda \in \Lambda$ such that: $d(x, \lambda) = min\{d(x, l),\ l \in \Lambda\}$."

In order to solve the closest vector problem, we use the basic and well-known nearest plane algorithm [1].

B. Basis search

Once the problem of quantizing on a lattice is adressed, there remains to find the most relevant lattice. For the data free approach, we opt for a simple random search with restarts, where restarts simply consist in running the algorithm several times in a row and keeping the best result of all the runs. We look for a lattice that reduces the mean cube error loss (MCE) between the full precision weights of the layer (or channel), and their quantized version.

V. Experiments

A. Zero-shot LatticeQ

TABLE I: LatticeQ with per-channel quantization and bias correction from [2] on ImageNet.

Network	Method	Top-1 accuracy W4A32	W3A32	W2A32
Resnet-18 (69.8)	LatticeQ (Ours)	**69.0**	**66.7**	**41.7**
	Banner et al.	67.5	43.2	1.2
	OMSE+opt	67.1		
Resnet-50 (76.0)	LatticeQ (Ours)	**75.6**	**73.6**	**47.3**
	Banner et al.	74.8	67.4	0.4
	OMSE+opt	74.7		
VGG16-bn (73.4)	LatticeQ (Ours)	**72.9**	**70.9**	**40.7**
	Banner et al.	71.6	65.9	0.1
Densenet-121 (74.4)	LatticeQ (Ours)	**73.3**	**68.9**	**10.4**
	Banner et al.	69.8	54.2	0.5
	OMSE+opt	71.7		
Mobilenet-v2 (71.9)	LatticeQ (Ours)	**68.2**	**48.9**	**0.3**
	Banner et al.	62.8	13.9	0.1

B. LatticeQ with data samples

TABLE II: LatticeQ with 512 training images

Network	Method	Top-1 accuracy W4A32	W3A32
Resnet-18 (69.8)	LatticeQ (Ours)	**69.3**	**68.6**
	Adaround	68.6	
	Bitsplit	69.1	66.8
Preresnet-18 (71.0)	LatticeQ (Ours)	**70.5**	**69.3**
	Brecq	70.3	69.0

We evaluate our method on the ImageNet [6] classification task. We report bitwidths settings and top-1 accuracy for each tested model, and we also provide the results from [2], [3], [5] and [4], [7] for comparison. As we see, in both tested hypotheses, LatticeQ reaches state-of-the-art accuracy for quantization.

VI. Analysis

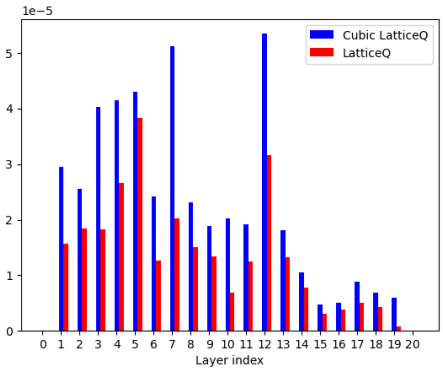

Fig. 3: Resnet18 per-layer quantization error comparison between LatticeQ and Cubic LatticeQ (scalar quantization). Vertical axis is MCE.

TABLE III: Comparison between baseline per-channel LatticeQ and baseline per-channel Cubic LatticeQ (scalar quantization).

Network	Method	FP32	W4A8
Resnet-18	LatticeQ	69.8	**67.2**
	Cubic LatticeQ	69.8	57.6

We experimentally show the advantages, both in quantization error (Figure 3), and task loss (Table III), in using a deformable lattice for quantization rather than a cubic lattice (i.e. uniform scalar quantization). This confirms our hypothesis that the inner correlations of the parameters of a neural network can be exploited for the purpose of quantization.

VII. Conclusion

In this paper, we introduced LatticeQ, a new post-training quantization method which exploits the flexibility of lattice quantizers for DCNN quantization, and gains from 2% on Resnet 4-bit weight quantization and up to 40% on 2-bit quantization. We believe that lattice quantization has potential beyond post-training, since our method also shows great performance when training data is available, and that our work could inspire other quantization methods outside of the realm of scalar quantization.

References

[1] L Babai. Nearest lattice point problem. 1986.

[2] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. Post-training 4-bit quantization of convolution networks for rapid-deployment. *NeurIPS*, 2019.

[3] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit Quantization of Neural Networks for Efficient Inference. *arXiv:1902.06822*, 2019.

[4] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the Limit of Post-Training Quantization by Block Reconstruction. *ICLR*, 2021.

[5] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? Adaptive rounding for post-training quantization. 2020.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575*, 2015.

[7] Peisong Wang, Qiang Chen, Xiangyu He, and Jian Cheng. Towards Accurate Post-training Network Quantization via Bit-Split and Stitching. *ICML*, 2020.