Universal Deep Neural Network Compression

Yoojin Choi   **Mostafa El-Khamy**   **Jungwon Lee**
SoC R&D, Samsung Semiconductor Inc.
San Diego, CA 92121, USA
`{yoojin.c,mostafa.e,jungwon2.lee}@samsung.com`

Abstract

In this paper, we investigate lossy compression of deep neural networks (DNNs) by weight quantization and lossless source coding for memory-efficient deployment. Whereas the previous work addressed non-universal scalar quantization and entropy coding of DNN weights, we for the first time introduce universal DNN compression by universal vector quantization and universal source coding. In particular, we examine universal randomized lattice quantization of DNNs, which randomizes DNN weights by uniform random dithering before lattice quantization and can perform near-optimally on any source without relying on knowledge of its probability distribution. Moreover, we present a method of fine-tuning vector quantized DNNs to recover the performance loss after quantization. Our experimental results show that the proposed universal DNN compression scheme compresses the 32-layer ResNet (trained on CIFAR-10) and the AlexNet (trained on ImageNet) with compression ratios of 47.1 and 42.5, respectively.

1   Introduction

Compression of deep neural networks (DNNs) has been actively studied in deep learning to develop compact DNN models for memory-efficient and computation-efficient deployment. Han et al. [1] showed impressive compression results by weight pruning, k-means clustering, and Huffman coding. It is further optimized in [2] using Hessian-weighted k-means clustering. Recently, it is shown how soft weight sharing or soft quantization can be employed for DNN weight quantization in [3, 4]. On the other hand, weight pruning is also extensively studied, e.g., in [5–9]. In this paper, we focus on DNN weight quantization, which can be used together with weight pruning to generate compressed models.

Vector quantization reduces the gap to the rate-distortion bound by jointly quantizing multiple symbols. Since conjectured by Gersho in [10], lattice quantization has been presumed to be the most efficient entropy coded vector quantization in the high resolution regime asymptotically, as the rate goes to infinity and the distortion diminishes [11]. Although lattice quantizers are simple and empirically shown to perform well even at finite rates, their efficiency depends on source statistics. Thus, we consider universal quantization that provides near-optimal performance for any source distribution [12]. Of particular interest is randomized lattice quantization, where uniform random dithering makes the distortion independent of the source, and the gap of its rate from the rate-distortion bound at any distortion level is provably no more than 0.754 bits per sample for any finite dimension [13].

From the classical lossy compression results, this paper establishes a universal DNN compression framework consisting of universal quantization and universal lossless source coding such as Lempel–Ziv–Welch [14–16] and the Burrows–Wheeler transform [17, 18]. In order to recover any accuracy loss resulting from weight quantization, we furthermore propose a fine-tuning algorithm for vector quantized DNNs. The gain of fine-tuning becomes larger as the vector dimension increases, due to the fact that the number of shared quantized values that are tunable (trainable) in a vector quantized model increases as the vector dimension increases.
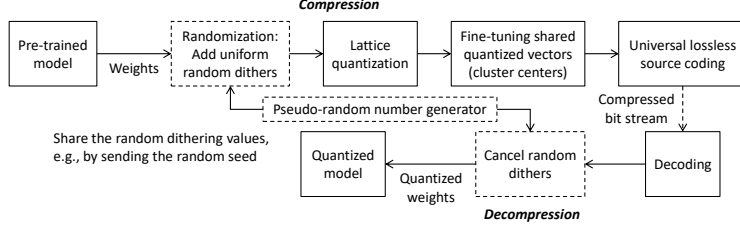
Figure 1: Universal DNN compression by universal quantization and universal source coding.

2 Universal DNN compression

For vector quantization, given N weights denoted by $w_1, \ldots, w_N$, we generate n-dimensional vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{\lceil N/n \rceil}$ by concatenating n distinct weights into one vector, e.g., as follows:

$$\mathbf{v}_i = [w_{ni-n+1} \; w_{ni-n+2} \; \cdots \; w_{ni}]^T, \quad 1 \le i \le \lceil N/n \rceil, \tag{1}$$

where $w_j = 0$ for $N + 1 \le j \le \lceil N/n \rceil n$, and $\lceil x \rceil$ is the smallest integer larger than or equal to x. Vector quantization partitions these n-dimensional vectors into a finite number of clusters, and the vectors in each cluster share their quantized value, i.e., the cluster center.

Randomized lattice quantization: Randomized lattice quantization [12] achieves universally good performance regardless of source statistics at any rates, and this leads us to the following universal DNN quantization method.

- We randomize the n-dimensional vectors in (1) by adding uniform random dithers as follows:

$$\tilde{\mathbf{v}}_i = \mathbf{v}_i + \mathbf{u}_i, \quad \mathbf{u}_i = [U_i \, U_i \; \cdots \; U_i]^T, \quad 1 \le i \le \lceil N/n \rceil, \tag{2}$$

  where each dithering vector $\mathbf{u}_i$ consists of n repetitions of a single uniform random variable U_i, and $U_1, \ldots U_{\lceil N/n \rceil}$ are independent and identically distributed (i.i.d.) uniform random variables of support $[-\Delta/2, \Delta/2]$; Δ is the quantization bin size. In each dimension, dithering values are i.i.d. uniform, which is sufficient to make the quantization error independent of source statistics.
- After dithering, uniform quantization in each dimension (i.e., lattice quantization) follows, i.e.,

$$\mathbf{q}_i = \Delta \, \text{round}(\tilde{\mathbf{v}}_i / \Delta), \tag{3}$$

  where $\mathbf{q}_i$ is the quantized vector of $\tilde{\mathbf{v}}_i$; the rounding and the scaling are element-wise operations in (3), where the rounding yields the closest integer values of the input.

Remark 1. Vector quantization theoretically provides a better rate-distortion trade-off. However, in practice, for compression of a finite number of data, the gain of vector quantization is limited by the codebook overhead, which becomes more considerable as dimension n increases and becomes the dominant factor that degrades the compression ratio after some point (see Figure 2 and compare the cases of $n = 4$ and $n = 6$).

Fine-tuning of vector quantized DNNs: We fine-tune the vector quantization codebook to recover the loss after quantization. Each element of a shared quantized vector in the codebook is fine-tuned separately. That is, if we have k_{VQ} clusters of n-dimensional vectors, we effectively divide weights into nk_{VQ} groups and fine-tune their shared quantized values separately. The average gradient of the network loss function with respect to weights is computed in each group and used to update their shared quantized value, as will be clarified below.

Let $\mathbf{c}_i = [c_{i,1} \; \cdots \; c_{i,n}]^T$ be the shared quantized vector present in the codebook for cluster i, and let $\mathcal{I}_{i,j}$ be the index set of all weights that are quantized to the same value $c_{i,j}$ from (3), for $1 \le i \le k_{VQ}$ and $1 \le j \le n$. The gradient descent for the shared quantized value $c_{i,j}$ is then given by

$$c_{i,j}(t) = c_{i,j}(t-1) - \eta \frac{1}{|\mathcal{I}_{i,j}|} \sum_{l \in \mathcal{I}_{i,j}} \nabla_{w_l} L(t-1), \tag{4}$$

where t is the iteration time, L is the network loss function, and η is the learning rate. The individual quantized vectors from (3) are also updated following their shared quantized vectors in the codebook.

In randomized lattice quantization, we would like to clarify that the average gradients are computed from the network loss function evaluated at the quantized weights after canceling random dithers (see (5)), while we fine-tune the shared values obtained before canceling random dithers (see (3)).
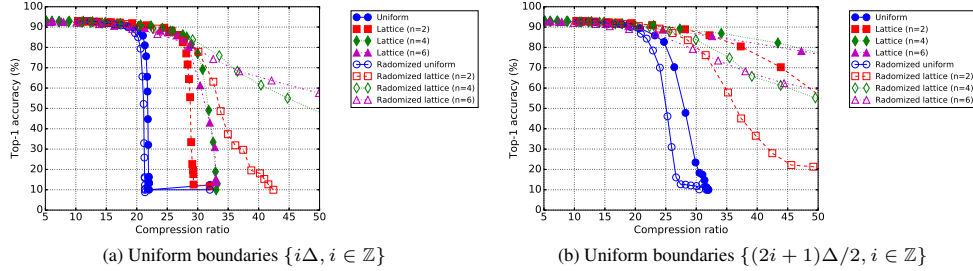
(a) Uniform boundaries $\{i\Delta, i \in \mathbb{Z}\}$ (b) Uniform boundaries $\{(2i+1)\Delta/2, i \in \mathbb{Z}\}$

Figure 2: Compression results of 32-layer ResNet by (randomized) lattice quantization and Huffman coding for different values of dimension n. No weight pruning is used in the results of this figure.

Universal source coding: Universal source coding algorithms are more convenient in practice than Huffman coding since they do not require us to estimate source statistics as in Huffman coding. Moreover, they utilize dictionary-based coding, where the codebook (i.e., dictionary) is built from source symbols adaptively in encoding and decoding, and therefore the codebook overhead is smaller than Huffman coding (see Remark 1).

For universal compression, the indices in the codebook of the lattice quantization output are passed as an input stream to a universal source coding scheme, which produces a compressed stream. The decoder needs to deploy the codebook that contains the indices and their corresponding fine-tuned shared quantized values for decompression.

Decompression: Using randomized lattice quantization, the encoder and the decoder are assumed to share the information on random dithers or their random seed. Under this assumption, the decoder decompresses the fine-tuned lattice quantization output from the compressed stream and then cancels the dithering vectors to obtain

$$\bar{\mathbf{v}}_i = \mathbf{q}_i - \mathbf{u}_i, \tag{5}$$

which yields the deployed weights of the universally quantized DNN at the inference step.

3 Experiments and discussion

We evaluate the proposed DNN compression scheme first without pruning for the 32-layer ResNet [19] (ResNet-32) model on CIFAR-10 dataset [20]. We consider two cases for (randomized) lattice quantization where the uniform boundaries in each dimension are set from (a) $\{i\Delta, i \in \mathbb{Z}\}$ and (b) $\{(2i+1)\Delta/2, i \in \mathbb{Z}\}$, respectively; $\mathbb{Z}$ is the set of integers. The quantization bin size Δ is the same for both cases, but case (a) has the zero at a bin boundary while in case (b) the zero element is at the center of the middle bin.

For unpruned models, we often have a high volume of weights concentrated around zero, and thus case (b) that assigns one bin to in-

Table 1: Compression of pruned models.

Model	Method	Top-1 accuracy (%)	Compression ratio
ResNet-32 (CIFAR-10)	Pre-trained model	92.58	-
	Ours: Weight pruning (87.5%)* + Universal quantization + bzip2	92.67 92.57	7.07 32.87
	Ours: Weight pruning (92.7%)* + Universal quantization + bzip2	92.06 92.06	11.86 47.10
	Han et al. [1]	N/A	N/A
	Choi et al. [2]	92.68	22.17
	Agustsson et al. [4]	92.10	20.15
AlexNet (ImageNet)	Pre-trained model	57.22	-
	Ours: Weight pruning (90.4%)* + Universal quantization + bzip2	57.28 57.02	9.11 42.46
	Han et al. [1]	57.22	35.00
	Choi et al. [2]	56.20	40.65
	Agustsson et al. [4]	N/A	N/A

* These rows show the compression results from weight pruning only.

clude all the weights near zero is expected to outperform case (a), which is aligned with our lattice quantization results in Figure 2. However, it is interesting to observe that randomized lattice quantization provides similarly good performance in both cases, which is the main benefit of randomizing the source by uniform dithering before quantization. Figure 2 also shows that vector quantization provides additional gain over scalar quantization particularly when the compression ratio is large.

Finally, Table 1 summarizes the compression ratios that we obtain from our universal DNN compression method for pruned ResNet-32 and AlexNet [21] models. The proposed universal DNN compression scheme with the *bzip2* [18] universal source coding algorithm yields $47.10\times$ and $42.46\times$ compression for ResNet-32 and AlexNet, respectively. Compared with [1, 2, 4] which need to optimize and/or calculate source statistics for compression, we achieved a better trade-off between rate (compression ratio) and distortion (loss in accuracy) through the universal compression of DNNs.

References

[1] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in *International Conference on Learning Representations*, 2016.

[2] Y. Choi, M. El-Khamy, and J. Lee, "Towards the limit of network quantization," in *International Conference on Learning Representations*, 2017.

[3] K. Ullrich, E. Meeds, and M. Welling, "Soft weight-sharing for neural network compression," in *International Conference on Learning Representations*, 2017.

[4] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Advances in Neural Information Processing Systems*, 2017, pp. 1141–1151.

[5] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient DNNs," in *Advances In Neural Information Processing Systems*, 2016, pp. 1379–1387.

[6] D. Molchanov, A. Ashukha, and D. Vetrov, "Variational dropout sparsifies deep neural networks," in *International Conference on Machine Learning*, 2017, pp. 2498–2507.

[7] C. Louizos, K. Ullrich, and M. Welling, "Bayesian compression for deep learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 3290–3300.

[8] J. Lin, Y. Rao, J. Lu, and J. Zhou, "Runtime neural pruning," in *Advances in Neural Information Processing Systems*, 2017, pp. 2178–2188.

[9] B. Dai, C. Zhu, and D. Wipf, "Compressing neural networks using the variational information bottleneck," in *International Conference on Machine Learning*, 2018.

[10] A. Gersho, "Asymptotically optimal block quantization," *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 373–380, 1979.

[11] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.

[12] J. Ziv, "On universal quantization," *IEEE Transactions on Information Theory*, vol. 31, no. 3, pp. 344–347, 1985.

[13] R. Zamir and M. Feder, "On universal quantization by randomized uniform/lattice quantizers," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 428–436, 1992.

[14] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, 1977.

[15] ——, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, 1978.

[16] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 6, no. 17, pp. 8–19, 1984.

[17] M. Effros, K. Visweswariah, S. R. Kulkarni, and S. Verdú, "Universal lossless source coding with the Burrows Wheeler transform," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1061–1081, 2002.

[18] J. Seward, "bzip2," 1998. [Online]. Available: www.bzip.org

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[20] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[22] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems*, 2015, pp. 1135–1143.

(a) Before fine-tuning the quantization codebook   (b) After fine-tuning the quantization codebook
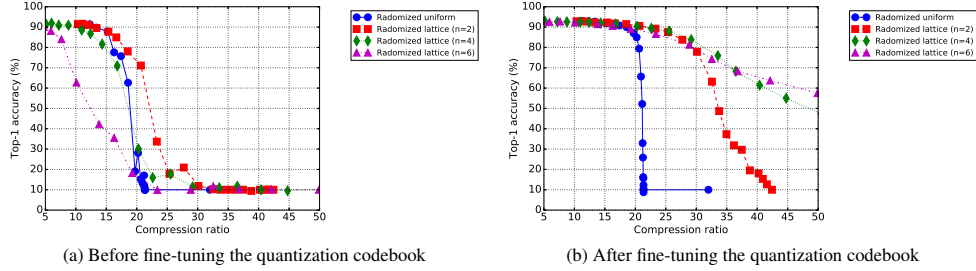
Figure 3: Performance comparison before and after fine-tuning the codebook for the 32-layer ResNet models, quantized with randomized lattice quantization for different values of vector dimension n, where the uniform boundaries in each dimension are from $\{i\Delta, i \in \mathbb{Z}\}$.

Table 2: Additional compression results for the pruned AlexNet model.

Model	Method	Top-1 accuracy (%)	Compression ratio
	Pre-trained model	57.22	-
	Weight pruning (90.4%)	57.28	9.11
AlexNet	Weight pruning (90.4%) + Universal quantization + Huffman coding	57.02	41.43
(ImageNet)	Weight pruning (90.4%) + Universal quantization + LZW	57.02	41.82
	Weight pruning (90.4%) + Universal quantization + bzip2	57.02	42.46

A   Appendix

A.1   Experimental details for the results in Figure 2

Figure 2 shows the compression results of the unpruned ResNet-32 model. In particular, we compare the trade-off between the accuracy and the compression ratio for lattice quantization and randomized lattice quantization with different values of vector dimension n (see (1)); here, uniform quantization corresponds to lattice quantization with dimension $n = 1$. Given the vector dimension n, the weights from all layers of the pre-trained ResNet-32 model are vectorized as in (1) for vector quantization. Then, lattice quantization or randomized lattice quantization follows. In plain lattice quantization, no random dithering is added before quantization, i.e., we set $\mathbf{u}_i = \mathbf{0}$ for all i in (2). We fine-tune the quantization codebook as explained in Section 2. We simply use Huffman coding only in this experiment to get the compressed models.

The gain of randomized lattice quantization over lattice quantization can be found in Figure 2(a) in particular for $n \geq 2$ and large compression ratios. We note that randomized lattice quantizers provide similarly good performance in both cases (a) and (b). Lattice quantization performs well only in case (b), where the quantization bins are optimized for given weight distribution. We emphasize that randomized lattice quantization is applicable for any network models blindly, regardless of their weight distribution and with no optimization, while it is guaranteed to yield a good rate-distortion trade-off close to the optimum within a fixed gap [12].

A.2   Additional results that show the impact of fine-tuning

In Figure 3, we show the performance of universally quantized ResNet-32 models before and after fine-tuning the codebook. The gain of vector quantization becomes more significant after fine-tuning, in particular, as the vector dimension n increases, since there are a more number of shared quantized values trainable in vector quantized models.

A.3   Contribution of pruning, quantization, and source coding

Table 2 shows the incremental improvement of the compression ratio for the AlexNet model. Given a pre-trained AlexNet model, we prune its 90.4% weights and fine-tune remaining unpruned weights, as suggested in [22], which yields the compression ratio of 9.11 with the top-1 accuracy of 57.28%. Using universal quantization and *bzip2* universal source coding for the pruned AlexNet model, we achieve the compression ratio of 42.46 with the top-1 accuracy of 57.02%. We also compare Huffman coding to the universal source coding algorithms, i.e., Lempel–Ziv–Welch (LZW) and *bzip2* that deploys the Burrows–Wheeler transform [17]. Both LZW and *bzip2* provide better compression ratios than Huffman coding in our experiments.