INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Extremely Low Bit Quantization for Mobile Speaker Verification Systems Under 1MB Memory

Bei Liu, Haoyu Wang, Yanmin Qian[†]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China

{beiliu, fayuge, yanminqian}@sjtu.edu.cn

Abstract

How to develop lightweight systems customized for mobile devices is an urgent and intriguing topic for speaker verification. In this paper, we investigate extremely low bit quantization for small-footprint speaker verification. Specifically, two different binary quantization schemes are proposed, namely static and adaptive quantizer. By applying them to the pre-trained full-precision ResNet, we successfully obtain binarized variants named as *b-vector* with a model size of under 1MB memory. Experiments on Voxceleb dataset illustrate that compared with the previous best small-footprint system, our best b-vector system achieves **38%**, **36%** and **30%** relative improvements on Vox1-O, E and H respectively, while maintaining almost identical model size. In addition, the analysis of the binarized weight histograms reveals that adaptive quantization scheme, when compared to the static method, can better match the real-valued distribution, and hence presents more effective representation ability.

Index Terms: speaker verification, mobile devices, neural network quantization, b-vector

1. Introduction

Speaker verification (SV) involves determining if enrollment and testing utterances are spoken by the same individual. The paradigm of SV systems undergoes the shift from the conventional i-vector [1] along with probabilistic linear discriminant analysis (PLDA) [2] towards the use of deep learning techniques for speaker embedding learning [3, 4, 5]. Recently, the performance of SV systems has been significantly improved with the utilization of much deeper and larger neural networks. For example, [6] proposes the depth-first version of ResNet and largely increases the depth of network to 233. Plus, [7] further pushes the depth of ResNet to 293 and obtains impressive performance gains. Although promising results have been achieved by large models, they generally consume substantial storage and computation resources, impeding the deployment on mobile devices. It is a challenging and demanding task of developing lightweight speaker verification systems that are customized for mobile devices.

In previous studies, researchers have explored several approaches for small-footprint speaker verification, including knowledge distillation [8, 9] and efficient architecture designs [10, 11, 12]. Knowledge distillation [13] is a commonly-used compression method which transfers knowledge from teacher networks to student ones. Despite the possibility of enhancing the performance of student networks without enlarging their model size, the deployment of these networks onto mobile devices still remains challenging due to the considerable number of parameters involved. On the other hand, many efforts have been made to manually devise more efficient calculation operators and network architectures. To reduce computational costs, researchers focus on leveraging lightweight convolution operations to substitute the computationally-intensive ones and introducing more efficient architectures tailored to embedded use cases. Although parameter number and computational complexity have significantly decreased, severe performance degradation can occur, which can barely meet the requirements of real-life SV applications.

In this paper, we investigate how to strike a better balance between performance and model size for small-footprint speaker verification. Neural network quantization is a compression technique employed to represent a 32-bit floating-point numbers with fewer bit width. The quantization of network weights can yield models with a smaller memory footprint. Specifically, we propose two distinct extremely low bit quantization schemes for SV systems, namely static and adaptive binary quantizer. Through the process of quantizing full-precision weights into 1-bit values, we successfully obtain a binarized variant of the pre-trained ResNet with a model size of less than 1MB. The experimental results on Voxceleb show that our best binarized model outperforms the previous state-of-the-art lightweight system, achieving significant relative improvements of **38%**, **36%** and **30%** on Vox1-O, E, and H respectively, while maintaining nearly identical model size. Furthermore, the analysis of the binarized weight histograms indicates that compared to the static method, adaptive quantization scheme can better align with the distribution of real-valued weights, thereby demonstrating superior representation capability.

2. Related Work

2.1. Small-footprint Speaker Verification

In recent years, small-footprint speaker verification has been an important and active research area. [8] proposes label-level and embedding-level distillation for small-footprint deep speaker embedding learning. [9] introduces a self-knowledge distillation framework to utilize enhanced features as teacher. In addition, [10] adopts QuartzNet [14] architecture with lightweight time channel separable 1-dimensional convolution (TCSConv1d) module. [11] develops a lite version of ECAPA-TDNN by squeezing feature mapping sizes and employing separable convolution. [12] presents a novel module called channel split time-channel-time separable 1-dimensional convolution (CS-CTCSConv1d) to enhance the performance of small-footprint SV systems.

[†] corresponding author

10.21437/Interspeech.2023-800

2.2. Neural Network Quantization

Neural network quantization is a widely-used compression method to represent full-precision numbers with fewer bit width. It has been extensively explored in various deep learning fields, including computer vision [15, 16, 17, 18], natural language processing (NLP) [19, 20] and speech recognition [21, 22]. [15] proposes k-means clustering based trainable quantization for image classification. [16] presents binary weight networks for challenging visual tasks. [17] introduces a novel non-uniform quantization scheme. [18] designs mixed-precision quantization strategy to assign different bit numbers for various layers. For NLP tasks, [19, 20] aim to quantize the large pre-trained language models to speed up the inference process. In addition, [21, 22] show that impressive compression ratio can be achieved for speech tasks without performance degradation.

3. Proposed Method

In this section, we first introduce the basic concepts of neural network quantization. Then, two distinct binary quantizers, the extreme case of quantization schemes, are proposed to quantize the full-precision weights of pre-trained ResNet system into 1-bit values for speaker verification based on quantization-aware training.

3.1. Preliminaries

In general, neural network quantization involves two operations: quantize and dequantize. In recent years, to bridge the non-negligible performance gap between full-precision models and their quantized counterparts, quantization-aware training [23] are introduced to minimize the quantization error in the training process.

quantize operation: This step aims to project real-valued numbers to low-precision integer values. For n-bit quantization, the integer set q is generally predefined as follows:

$$q \in \left\{0, \pm 1, \pm 2, \ldots, \pm(2^{n-1}-1)\right\} \tag{1}$$

The quantize operation can be achieved by *round* function.

dequantize operation: This operation is an affine mapping of integers to real-valued numbers. The specific calculation can be presented as follows:

$$Q = \alpha \times q \in \left\{0, \pm\alpha, \ldots, \pm\alpha \times (2^{n-1}-1)\right\} \tag{2}$$

where α represents the learnable full-precision scaling factor.

For a neural network, we can build a quantization integer set and the corresponding scaling factor for each of its layers.

3.2. Static Quantization

For static 1-bit quantization, the binary values are restricted to a fixed integer set, i.e. $\{-1, +1\}$ for all layers of a neural network, as shown in Figure 2 (left). Despite its simplicity, this quantization strategy ignores the fact that the weight distributions in different neural network layers are diverse. In addition, there exists a significant mismatch in magnitude between real-valued and the quantized weights. For example, in the pre-trained ResNet34 speaker verification system, the majority of weights reside in $10^{-3} \sim 10^{-2}$ magnitude. However, the quantized weights are generally spread within the interval of $[-1, 1]$. This phenomenon can incur substantial quantization error. Inspired by [24], we propose entropy preserving weight regularization to enhance the performance of binarized network.

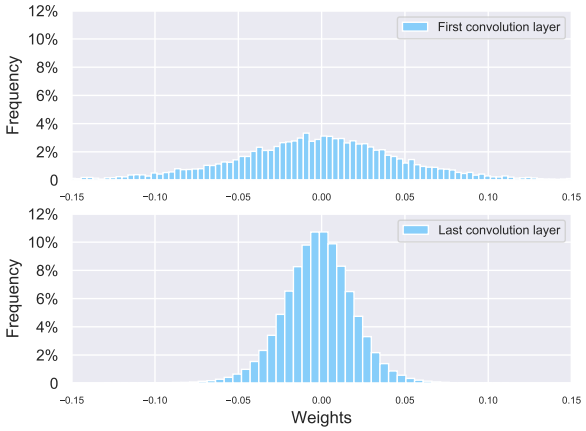

Figure 1: *Pre-trained weight distributions of the first and last convolution layers in ResNet34 speaker verification system.*

From the view of information theory, a distribution with higher entropy can preserve more information. Therefore, we introduce a weight regularizer which aims to preserve maximal entropy and minimize information loss in quantized weights. Theoretical analysis indicates that the maximum value of information entropy in quantized weights can be achieved when the real-valued weights are quantized into various quantization levels in equal proportions. Empirically, the corresponding quantized weights exhibit an approximately uniform distribution within binary integer set $\{-1, +1\}$ when the real-valued weights are regularized using the following equation:

$$W^{r\prime} = \frac{|W^r|}{\|W^r\|_{l1}} W^r \tag{3}$$

where W^r is the real-valued weight matrix. $|W^r|$ denotes the number of entries in the matrix. $\|W^r\|_{l1}$ stands for the L1 norm of the matrix.

Then, the regularized real-valued weight $W^{r\prime}$ are binarized through the following quantize and dequantize operation as Eq.4 and Eq.5 show.

$$q = round((clip(w^{r\prime}, -1, 1)+1)\times\frac{1}{2})\times 2 - 1 \in \{-1, 1\} \tag{4}$$

$$Q = \alpha \times q \tag{5}$$

where $clip$ is the function to clamp values between -1 and 1. $round$ is the function to map values to the nearest integer. α is the scaling factor.

3.3. Adaptive Quantization

Previous studies [15, 17] have demonstrated that weights in neural networks generally adhere to a bell-shaped distribution. However, our empirical findings indicate that the shape of this distribution varies across different layers of a neural network. As Figure 1 shows, weight distribution in shallow layer exhibit a wider range and larger variance, while that in deep layer is typically denser and narrower with the majority of weights centered around 0. Therefore, using a fixed integer set in static quantization limits the ability to provide binary diversity for various weight distributions, which ultimately constrains the representation capacity of the quantized network. In this section, we propose an adaptive quantization scheme that can dynamically determine the optimal binary set for each layer to achieve better alignment with the real-valued weight distribution.

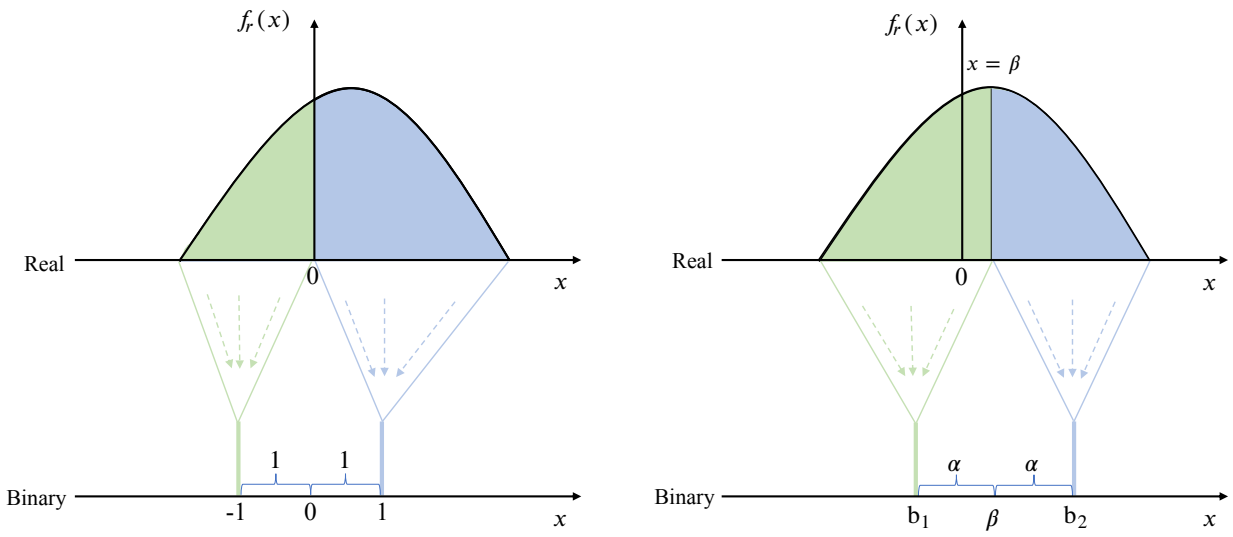

Figure 2: *The overview of static and adaptive binary quantization. Static quantization (left) map the real-valued weights into a fixed integer set $q \in \{-1, +1\}$ for all layers. In contrast, adaptive quantization (right) can dynamically determine the binary set $Q \in \{\beta - \alpha, \beta + \alpha\}$ for each layer to better match the real-valued weight distribution.*

Different from the static quantization where the binary integer set is fixed as $\{-1, +1\}$, we introduce two adaptive parameters α and β to better align with the distributions of real-valued weights in each layer, as Figure 2 (right) displays. The binarized weight can be obtained as follows:

$$Q = \begin{cases} \beta - \alpha, & w^r < \beta \\ \beta + \alpha, & w^r > \beta \end{cases} \tag{6}$$

where β is the center of binarized weights. α is the distance to the center. In this case, the binary set becomes $\{\beta - \alpha, \beta + \alpha\}$.

In addition, Kullback-Leibler divergence (KLD) is adopted to measure the distribution similarity between binarized and real-valued weights as follows:

$$D_{KL}(P_r \parallel P_b) = \int P_r(x) \log \frac{P_r(x)}{P_b(x)} \, dx \tag{7}$$

where the $P_r(x)$ and $P_b(x)$ denote the probability distribution of real-valued and binarized weights respectively.

Given a real-valued weight matrix W^r, we firstly align the center of binary value β to the mean of the real-valued weight distribution. Therefore, β can be obtained via:

$$\beta = \frac{1}{c \times k \times k} \sum_{m=0}^{c-1} \sum_{j=0}^{k-1} \sum_{i=0}^{k-1} W^r_{m,j,i} \tag{8}$$

where c and k represent the channel number and kernel size respectively. m, j and i are the index to iterate through the channel number c, kernel_size1 k and kernel_size2 k respectively.

As inferred in 3.2, we assume that the binarized weights conform to a uniform distribution, which means $P_b(\beta - \alpha) = P_b(\beta + \alpha) = 0.5$. For real-valued weights, the distribution is roughly a bell-shaped curve, which is widely believed to obey Gaussian distribution. To minimize the KL distance, we empirically observe that α should be on the position of standard deviation of W^r. Finally, α can be estimated via:

$$\alpha = \frac{\|W^r - \beta\|_2}{\sqrt{c \times k \times k}} \tag{9}$$

In the proposed adaptive quantization scheme, α and β can be dynamically updated along with the real-valued weights during the training process for each network layer.

4. Experimental Setup

4.1. Datasets

We conduct experiments on Voxceleb1&2 [25, 26] datasets, using the development set of Voxceleb2 as the training data and Voxceleb1 as the testing data. Performance is evaluated on the three official trials: Vox1-O, Vox1-E and Vox1-H. Plus, three data augmentation techniques are employed to enhance the diversity of training data, including online data augmentation [27] with MUSAN [28] and RIR dataset [29], specaugment [30], speed perturb [31] with 0.9 and 1.1 times speed changes.

4.2. Training Strategies

Our training process consists of two stages. The first stage aims to obtain a full-precision speaker verification system. Then, we apply the proposed 1-bit quantization schemes to the pre-trained network, yielding the corresponding binarized models.

stage 1: In the experiments, we adopt ResNet34 as the speaker embedding extraction model. Firstly, a ResNet34-based SV system is trained in full precision. For training data, a 200-frame segment is randomly chunked from each utterance. The input features are 80-dimensional Fbank with a window length of 25ms and a shift of 10ms. AAM-softmax [32] with a margin of 0.2 and a scale of 32 is employed as the loss function. The optimizer is SGD with momentum of 0.9 and weight decay of $1e$-4. The extracted speaker embedding is 256-dimension.

stage 2: Subsequently, the pre-trained ResNet34 full-precision model is re-loaded and fine-tuned for 40 epochs using the proposed 1-bit quantization schemes. During training, online data augmentation and spec-augment are discarded. The remaining settings are kept the same in stage 1.

4.3. Evaluation Metrics

Cosine distance is adopted to measure the embedding similarity. Then, we normalize the resulting scores using adaptive score normalization (AS-Norm) [33] with an imposter cohort size of 600. Performance is evaluated in terms of the equal error rate (EER) and the minimum detection cost function (MinDCF) with the settings of $P_{target} = 0.01$ and $C_{FA} = C_{Miss} = 1$.

Table 1: *EER and MinDCF results of previous small-footprint systems and our proposed b-vector on the Voxceleb1 dataset.*

System	Proceeding	Model Size	Voxceleb-O		Voxceleb-E		Voxceleb-H	
			EER(%)	MinDCF	EER(%)	MinDCF	EER(%)	MinDCF
ECAPA-TDNNLite [11]	ICASSP'22	1.27MB	3.07	0.296	3.00	0.318	5.20	0.436
Julien et al. [10]	ICASSP'21	0.95MB	2.91	0.284	3.04	0.292	4.79	0.396
CS-CTCSConv1d [12]	INTERSPEECH'22	0.96MB	2.77	0.280	2.83	0.282	4.49	0.383
ResNet34 (full-precision)	–	26.7MB	0.89	0.098	1.01	0.121	1.85	0.184
b-vector (static)	Ours	0.97MB	**1.90**	**0.212**	**1.99**	**0.215**	**3.40**	**0.298**
b-vector (adaptive)		0.97MB	**1.72**	**0.200**	**1.81**	**0.197**	**3.14**	**0.278**

5. Results and Analysis

5.1. The Performance of b-vector

The performance and model size measured in MegaBytes (MB) of recent small-footprint SV systems and our proposed binarized models are presented in Table 1.

As stated in section 4.2, we firstly pre-train a ResNet34-based system in full precision. Although the full-precision model demonstrates promising performance, its practical deployment on edge devices is impeded by prohibitive memory requirements. By applying the proposed static and adaptive 1-bit quantization schemes, two different binarized models are obtained. Specifically, we name the embedding extracted from the resulting quantized models as *b-vector*. Form Table 1, it can be observed that the model size is effectively reduced to less than 1MB, resulting in a significant 27x compression ratio.

Regarding model performance, it is evident that b-vector (adaptive) achieves much better results than b-vector (static), indicating that adaptive quantization scheme exhibits superior speaker representation capability. Compared to recently published small-footprint systems, both static and adaptive b-vector systems achieve a new state-of-the-art performance with similar model size. Specifically, Julien et al. [10] present a variant of QuartzNet customized for embedded systems. Furthermore, CS-CTCSConv1D [12] proposes several architectural enhancements to Julien et al.'s model, resulting in the best reported performance to date for small-footprint speaker verification. By comparison, our best system b-vector (adaptive) obtains an average relative improvement in EER by 35% and in MinDCF by 29% while maintaining nearly identical model size. Additionally, our proposed b-vector systems outperform ECAPA-TDNNLite by a significant margin with 24% fewer memory costs. The above analysis illustrates that b-vector systems achieve a much better trade-off on performance and model size in the context of small-footprint speaker verification.

5.2. Weight Distribution Analysis

In this section, we provide an analysis of the histograms of binarized weights for both static and adaptive quantization schemes. As depicted in Figure 3, the distributions of pre-trained real-valued weights exhibit a significant distinction between the first and last convolution layers. Static quantization employs a fixed integer set, resulting in highly similar binarized outcomes for the first and last convolution layers (0.07 vs. 0.06). This impedes its ability to match the distribution of real-valued weights accurately. For example, most of the weights are concentrated within the range $[-0.05, 0.05]$ in the last layer. Nonetheless, static quantization generates two binarized weights ± 0.06 which can incur significant quantization error. In contrast, adaptive quantization demonstrates superior representation capabilities owing to its ability to adaptively determine the binary set based on the distribution of real-valued weights. For instance, in the first layer, where the distribution is wider, adaptive quantization maps the weights to ± 0.05. On the other hand, it produces two binarized weights of -0.02 and 0.02 in the last layer due to a denser and narrower distribution. This exemplifies the ability of the adaptive scheme to better align with the distribution of real-valued weights, leading to enhanced performance.

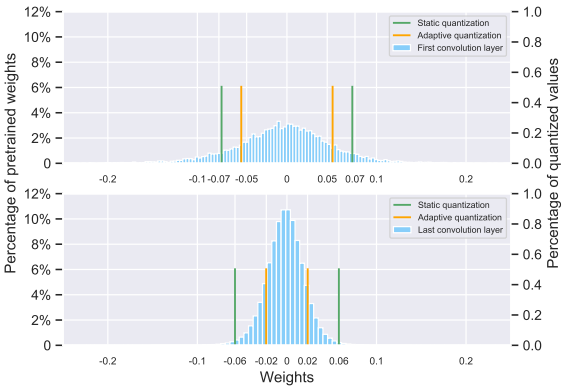

Figure 3: *The distributions of pre-trained real-valued weights and binarized weights in the first and last convolution layers.*

6. Conclusions

In this paper, we explore extremely low bit quantization for small-footprint speaker verification. Specifically, two distinct binary quantization schemes, static and adaptive quantizer, are proposed. By applying them to the pre-trained full-precision ResNet, we successfully obtain binarized variants named as *b-vector* with a model size of less than 1MB memory. Experiments on Voxceleb show that our best b-vector system outperforms the previous best small-footprint system by **38%**, **36%**, and **30%** on Vox1-O, E and H respectively, while maintaining nearly identical model size. In addition, the binarized weight histogram indicates adaptive quantization scheme exhibits superior representation capability over the static method.

7. Acknowledgement

This work was supported in part by China STI 2030-Major Projects under Grant No.2021ZD0201500, in part by China NSFC projects under Grants 62122050 and 62071288, and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

8. References

[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[2] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 531–542.

[3] D.Snyder, D.Garcia-Romero, G.Sell, D.Povey, and S.Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[4] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "But system description to voxceleb speaker recognition challenge 2019," *arXiv preprint arXiv:1910.12592*, 2019.

[5] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, pp. 3830–3834.

[6] B. Liu, Z. Chen, S. Wang, H. Wang, B. Han, and Y. Qian, "Df-resnet: boosting speaker verification performance with depth-first design," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2022, pp. 296–300.

[7] Z. Chen, B. Liu, B. Han, L. Zhang, and Y. Qian, "The sjtu x-lance lab system for cnsrc 2022," *arXiv preprint arXiv:2206.11699*, 2022.

[8] S. Wang, Y. Yang, T. Wang, Y. Qian, and K. Yu, "Knowledge distillation for small foot-print deep speaker embedding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6021–6025.

[9] B. Liu, H. Wang, Z. Chen, S. Wang, and Y. Qian, "Self-knowledge distillation via feature enhancement for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7542–7546.

[10] J. Balian, R. Tavarone, M. Poumeyrol, and A. Coucke, "Small footprint text-independent speaker verification for embedded systems," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6164–6168.

[11] Q. Li, L. Yang, X. Wang, X. Qin, J. Wang, and M. Li, "Towards lightweight applications: asymmetric enroll-verify structure for speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7067–7071.

[12] L. Cai, Y. Yang, X. Chen, W. Tu, and H. Chen, "Cs-ctcsconv1d: small footprint speaker verification with channel split time-channel-time separable 1-dimensional convolution," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2022, pp. 326–330.

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[14] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6124–6128.

[15] 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 (ICLR)*, 2016.

[16] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 525–542.

[17] Y. Li, X. Dong, and W. Wang, "Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks," in *International Conference on Learning Representations (ICLR)*, 2020.

[18] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, and K. Keutzer, "Hawq: Hessian aware quantization of neural networks with mixed-precision," in *International Conference on Computer Vision (ICCV)*, 2019, pp. 293–302.

[19] S. Kim, A. Gholami, Z. Yao, M. W. Mahoney, and K. Keutzer, "I-bert: Integer-only bert quantization," in *International Conference on Machine Learning (ICML)*, 2021, pp. 5506–5518.

[20] C. Tao, L. Hou, W. Zhang, L. Shang, X. Jiang, Q. Liu, P. Luo, and N. Wong, "Compression of generative pre-trained language models via quantization," in *Association for Computational Linguistics (ACL)*, 2022, pp. 4821–4836.

[21] J.Xu, J.Yu, X.Liu, and H.Meng, "Mixed precision dnn quantization for overlapped speech separation and recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7297–7301.

[22] J. Xu, S. Hu, X. Liu, and H. Meng, "Towards green asr: Lossless 4-bit quantization of a hybrid tdnn system on the 300-hr switchboard corpus," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2022, pp. 2128–2132.

[23] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2704–2713.

[24] Z. Liu, K. Cheng, D. Huang, E. Xing, and Z. Shen, "Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 4942–4952.

[25] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 2616–2620.

[26] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1086–1090.

[27] W. Cai, J. Chen, J. Zhang, and M. Li, "On-the-fly data loader and utterance-level aggregation for speaker and language recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 28, pp. 1038–1051, 2020.

[28] D. Snyder, G. Chen, and D. Povey, "Musan: a music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[29] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.

[30] D. S. Park, W. Chan, Y. Zhang, C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: a simple data augmentation method for automatic speech recognition," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2613–2617.

[31] W. Wang, D. Cai, X. Qin, and M. Li, "The dku-dukeece systems for voxceleb speaker recognition challenge 2020," *arXiv preprint arXiv:2010.12731*, 2020.

[32] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2019, pp. 2873–2877.

[33] Z. N. Karam, W. M. Campbell, and N. Dehak, "Towards reduced false-alarms using cohorts," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 4512–4515.