INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Compressed MoE ASR Model Based on Knowledge Distillation and Quantization

Yuping Yuan[1,2,†], *Zhao You*[2,†], *Shulin Feng*[2], *Dan Su*[2], *Yanchun Liang*[1,3], *Xiaohu Shi*[1,3]*, *Dong Yu*[4]

[1]College of Computer Science and Technology, Jilin University, Changchun, China
[2]Tencent AI Lab, Shenzhen, China
[3]School of Computer Science, Zhuhai College of Science and Technology, Zhuhai, China
[4]Tencent AI Lab, Bellevue, WA, USA

{yuanyp20@mails.,ycliang@,shixh@}jlu.edu.cn, {dennisyou,shulinfeng,dansu,dyu}@tencent.com

Abstract

The mixture of experts (MoE)-based automatic speech recognition (ASR) model can achieve remarkable performance, but pose greater challenges to model deployment for its huge model size. Therefore, it is important to compress the model size and reduce the computational cost. In this paper, we propose a compressed MoE (CMoE) ASR model that simplifies the MoE structure by knowledge distillation and reduces parameter bit-width through quantization, and provide two pipelines (one-stage and two-stage pipelines) to deploy the compression. In quantization, we use binary weight network to quantize the weights to 1-bit for reducing the quantization error and use learned step size quantization to quantize the activations to 4-bit. Experimental results show that the quantized dense network compressed from the MoE based ASR model by our method reduces the size by 150x with very small accuracy loss. The proposed model is expected to be deployed on embedded devices.
Index Terms: speech recognition, mixture of experts, knowledge distillation, model quantization, extreme compression

1. Introduction

Various powerful network architectures such as Transformer, Conformer, etc. have recently made excellent progress in end-to-end automatic speech recognition (ASR) models[1, 2]. Besides, we have witnessed the larger models show promising performance on various speech recognition tasks. The mixture of experts (MoE)-based method, as an effective way to increase the model capacity, has attracted the attention of many researchers. MoE models have achieved impressive performance in many different domains, such as language modeling[3, 4, 5], image classification[6, 7, 8, 9], and speech recognition[10, 11, 12, 13]. Although the ASR model based on MoE can achieve remarkable performance and improve the performance of the dense model effectively, but the amount of model parameters is too large, which is not conducive to the application and deployment of embedded equipments. Previously, the scheme of compressing the MoE-based ASR model was converted into a non-MoE structure through knowledge distillation[14]. This method can effectively reduce the size of the model and improve the inference efficiency, but there is still a certain distance from the deployment of embedded devices with limited resources. The goal of further compression of the model can be solved by quantization, but the existing quantization of weights and activations to low bits will lead to a serious performance decline[15][16][17]. Thus, weights and activations quantization of ASR model, especially ultra-low bit quantization, which refers to 2-bit and 1-bit, is still worth exploring.

In order to break through the limitations mentioned above, this paper proposes a compressed MoE (CMoE) ASR model, which has an order of magnitude improvement in compression ratio compared with existing methods, compressing the float MoE ASR model into a binary (1-bit) quantized dense network. We conduct several experiments on the LibriSpeech with MoE-Conformer model to evaluate our proposed method. We make the following contributions:

- We propose a simple effective method that can compress MoE-Conformer ASR model to a binary Dense-Conformer ASR model while achieving promising performance. We distill the MoE layer of each Conformer block into the original half-step feed-forward layer to simplify the structure and quantize the Conformer block into ultra-low bit to reduce parameter bit-width for compressing as much as possible. We quantize the weights to 1-bit by binary weight network (BWN[18]) and quantize the activations to 4-bit by learned step size quantization (LSQ[19])
- We propose a two-stage pipeline to realize the above compression process separately, and further propose a one-stage pipeline to complete compression, which effectively shortens the pipeline. One-stage pipeline has simpler pipeline and comparable performance to the two-stage.
- We reduce the model size by 150x. It allows us to deploy complicated MoE models on embedded devices with low computational resources.

The rest of the paper is organized as follows. Section 2 reviews the previous works of MoE-Conformer model, BWN and LSQ. Section 3 presents our proposed CMoE method. The experimental setup and the experimental results are reported in Section 4. Finally, we conclude this paper in Section 5.

2. Related Works

In this section, we mainly describe the MoE-Conformer ASR model and the model quantization methods involved.

2.1. MoE-Conformer

The MoE-Conformer ASR model can achieve state of the art recognition performance with a large mixture of experts neural network[12]. The MoE-Conformer model is based on Connectionist Temporal Classification (CTC)/attention-based encoder-decoder(AED) framework. As shown in Fig. 1(a), the decoder has two branches, the CTC decoder and the attention decoder, which share the same encoder. The encoder layer shown in Fig. 1(b) consists of a MoE layer, non-expert layers and a shared embedding network. Each MoE layer consists of m experts (FFN modules) and a router layer. The embedding network is a small Conformer ASR encoder. The router concatenates previous layer's output and the embedding network's output o^e as

[†]Equal contribution.
[*]Corresponding author.

10.21437/Interspeech.2023-2544

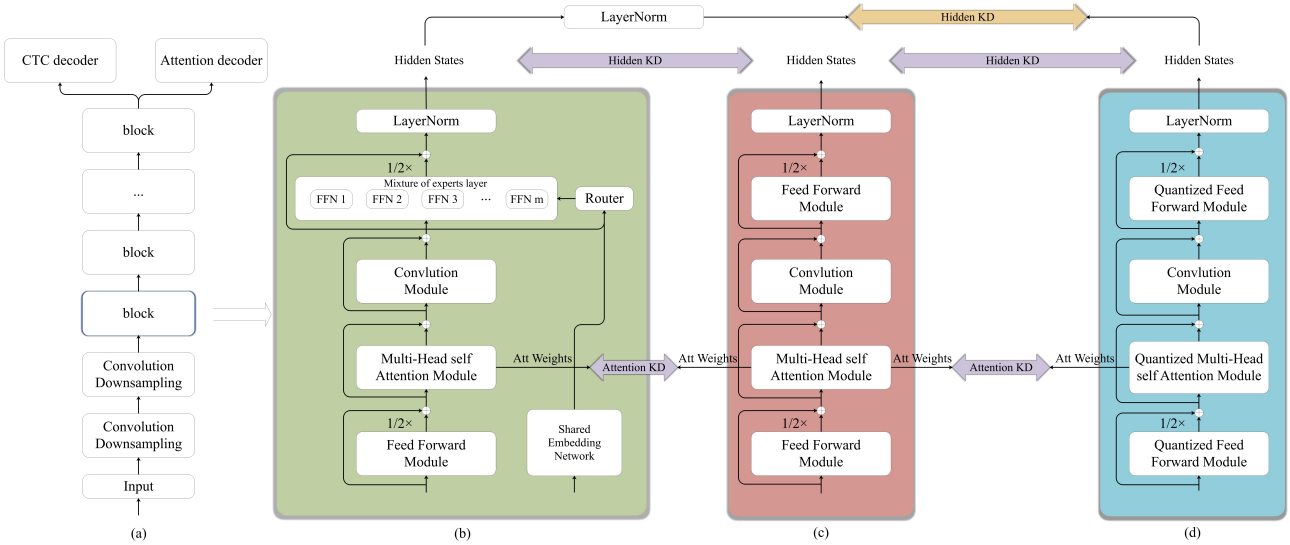

Figure 1: *CMoE model. (a) The overall architecture of the speech recognition model. (b), (c) and (d) are the architecture of Moe-Conformer block, Dense-Conformer block and binary Dense-Conformer block, any of them can be used as a block in (a). The purple arrow in the figure represents the knowledge distillation involved in two-stage pipeline, and the yellow arrow represents the knowledge distillation used in one-stage pipeline.*

input and routes each speech frame to the top-1 expert with the largest route probability. For the l-th MoE layer, let W_r^l and o^{l-1} be the router weights of the l-th MoE layer and the output of the previous layer, then the output E_q^l of selected expert q is also gated by router probability r^l to get the output y^l of the MoE layer:

$$r^l = W_r^l \cdot Concat(o^e; o^{l-1}) \tag{1}$$

$$y^l = \frac{exp^{r_q^l}}{\sum_{p=1}^{m} exp^{r_p^l}} E_q^l \tag{2}$$

2.2. Model quantization

2.2.1. Weight quantization

By estimating the binary weight values as close to the full-precision weights as possible, BWN quantizes the weights to $\{+s_w, -s_w\}$ to reduce the quantization error and make the quantized network performance close to the full-precision (FP32) network[18]. The scaling factor s_w for the weight is the average of the absolute weight values:

$$s_w = \frac{1}{k} \parallel w \parallel_1 \tag{3}$$

k represents the number of weight parameters. For each element in the binary representation w_i^b:

$$w_i^b = Binarize(w_i) = Sign(w_i) = \begin{cases} +1, & if \quad w_i \geq 0, \\ -1, & otherwise, \end{cases} \tag{4}$$

where w_i represents the real-value weight. The optimal estimation for w is the binary weight $\hat{w}$:

$$\hat{w} = s_w \cdot w^b \tag{5}$$

BWN will calculate the binary weights based on the actual weight values for forward propagation and back propagation. The real weight values are updated with the gradient calculated by the binary weights. In inference, we use binary weights to perform forward propagation.

2.2.2. Activation quantization

LSQ finds suitable scaling factor s_a by learning, rather than calculating[19]. Given data a, quantizer scaling factor s_a, $-Q_N$ and Q_P are the minimum and maximum quantized values, respectively. The quantized representation $\bar{a}$ of the activation a is defined as:

$$\bar{a} = Quantize(a) = \lfloor clip(a/s_a, -Q_N, Q_P) \rceil \tag{6}$$

Here, $\lfloor z \rceil$ rounds z to the nearest integer, and $clip(z, z_{min}, z_{max})$ returns z with values below z_{min} set to z_{min} and values above z_{max} set to z_{max}. The dequantize representation $\hat{a}$ of the activation a is defined as:

$$\hat{a} = Dequantize(\bar{a}) = \bar{a} \times s_a \tag{7}$$

Given bit-width b, for unsigned activations, $Q_N = 0$, $Q_P = 2^b - 1$, and for signed activations, $Q_N = 2^{b-1}$, $Q_P = 2^{b-1} - 1$. The gradient of the learnable scaling factor s_a is:

$$\frac{\partial \hat{a}}{\partial s_a} = \begin{cases} -a/s_a + \lfloor a/s_a \rceil & if \quad -Q_N < a/s_a < Q_P, \\ -Q_N & if \quad a/s_a \leq -Q_N, \\ Q_P & if \quad a/s_a \geq Q_P, \end{cases} \tag{8}$$

3. Proposed Method

To address the existing methods' limitations, this paper proposes a compressed MoE (CMoE) ASR model based on knowledge distillation and quantization, resulting in a dense model with the weights of 1-bit and the activations of 4-bit. We use BWN for quantizing weights because it can reduce the quantization error effectively, and use LSQ for quantizing activations because it can learn scaling factor compared with uniform quantization. Moreover, we developed two pipelines to achieve the CMoE model, namely two-stage compression and one-stage compression. These two pipelines will be introduced separately in the following subsection. The overall architecture of the speech recognition model adopted in this paper is described in Fig. 1(a). Knowledge distillation runs through all stages of our method, so the total loss of optimizing the student network is a combination of distillation loss and speech recognition task supervision loss, which is as follows:

$$L_s = \gamma L_{kd} + \lambda L_{ctc} + (1 - \lambda) L_{aed} \tag{9}$$

Among them, L_{kd} is the distillation loss, L_{ctc} and L_{aed} are the CTC loss and AED loss. γ , λ and 1-λ are the weighted values

for L_{kd}, L_{ctc} and L_{aed}, respectively. The specific distillation loss L_{kd} will be described in detail in the specific pipeline.

3.1. Two-stage pipeline

In the two-stage pipeline, we realize progressive compression and call it two-stage quantization (2S-QT). We first distill the MoE model into the dense model, and then distill the dense model into binary dense model. The specific process is shown in Fig. 1. The first stage of two-stage pipeline compresses MoE-Conformer ASR model into Dense-Conformer ASR model through knowledge distillation. The difference between the teacher MoE-Conformer model (as shown in Fig. 1(b)) and the student Dense-Conformer model (as shown in Fig. 1(c)) lies in the MoE/FFN layer. The MoE-Conformer model is described in Section 2.1. The Dense-Conformer model can be regarded as a MoE-Conformer model containing an expert, but it can achieve similar performance to that of multiple experts. The second stage of the two-stage pipeline compresses Dense-Conformer ASR model into binary Dense-Conformer ASR model through knowledge distillation. The difference between the teacher Dense-Conformer model (as shown in Fig. 1(c)) and the student binary Dense-Conformer model (as shown in Fig. 1(d)) lies in the Modules/Quantized Modules. The binary Dense-Conformer model is obtained by quantizing Feed Forward Modules and Multi-Head Self Attention Modules in the Dense-Conformer model. Note that the Modules in the attention decoder have also been quantized accordingly. In the two-stage pipeline, the knowledge distillation of each stage (as shown in Fig. 1(b) to (c) and Fig. 1(c) to (d)) is in the same form[20], as follows:

$$L_{kd}^{2S} = \sum_{i=1}^{n} MSE(A^T, A^S) + \sum_{j=0}^{n} MSE(H^T, H^S) \quad (10)$$

L_{kd}^{2S} will be used as the L_{kd} in Eq. 9 for training student models. Among them, A^T and H^T refer to the attention weights and hidden states of the teacher model encoder layers; A^S and H^S refer to the attention weights and hidden states of the student model encoder layers. n refers to the number of encoder layers, and i and j refer to the encoder layer index. In particular, $i = 0$ represents the input hidden state of the first encoder layer.

For the binary Dense-Conformer model, the weights are represented by 1-bit through BWN and the activations are represented by 4-bit through LSQ in the quantized modules. Fig. 2 formally describes the running mechanism of quantization during training and inference. It is worth noting that quantizing the weights alone can reduce the size of the model, but it does not have a significant acceleration effect. The joint quantization of weights and activations can perform the integer matrix multiplication on hardware devices that support fixed-point operations, so that the model can reduce the size of the model while improving the runtime efficiency. To achieve faster computing operations, we further quantize the activations. To facilitate quantized training, the activation quantization we use usually dequantizes the quantization representation $\bar{a}$ to $\hat{a}$ to simulate the errors generated by quantization in training.

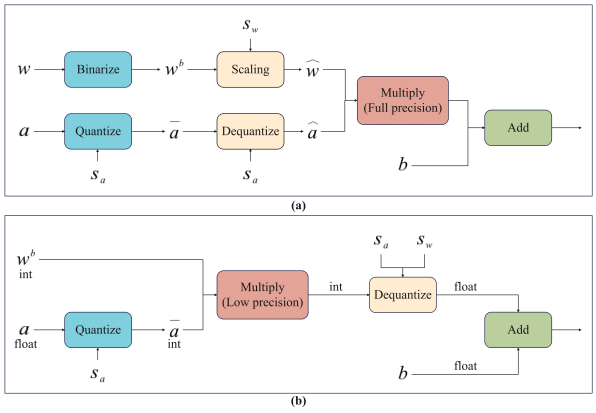

Figure 2: *Quantization Running Mechanism. (a) During training, the quantization process is carried out in a floating-point manner. (b) During inference, quantized weights and activations can perform low-precision operations, such as matrix multiplication.*

3.2. One-stage pipeline

In this paper, we explore the one-stage quantization (1S-QT), where the quantization process is conducted with one-stage that binary dense structure is obtained directly from MoE model. As a result, the pipeline of model compression is shortened. Specifically, the teacher is the FP32 MoE-Conformer model(as shown in Fig. 1(b)), and the student is the binary Dense-Conformer model (as shown in Fig. 1(d)) with 1-bit weights and 4-bit activations. The conversion from MoE to Dense compresses the model structurally. The conversion from FP32 to 1-bit further reduces the delay and model size by limiting the weight bit-width. In the two-stage compression pipeline, there is a transition process of knowledge transfer between the MoE-Conformer model and the binary Dense-Conformer model. In the one-stage pipeline, due to the large gap in knowledge between the teacher and the student, the learning of the hidden states is not easy to converge. Therefore, we use the different form[14] of knowledge distillation loss in one-stage pipeline:

$$L_{kd}^{1S} = \sum_{i=1}^{n} MSE(norm(H^T), H^S) \quad (11)$$

Distillation loss L_{kd}^{1S} only focus on learning the teacher's hidden states information, not attention weights information. Other variable's definitions are the same as Eq. 10. Similar to 2S-QT, L_{kd}^{1S} will be used as the L_{kd} in Eq. 9 for training student models.

4. Experiments

4.1. Datasets

We evaluate our proposed method on the LibriSpeech dataset of English speech[21] on NVIDIA A100s. Evaluation is performed in terms of word error rate (WER) on LibriSpeech test-clean and test-other. The input speech uses 80-dimension log-Mel filterbank features, computed with a 25ms window and shifted every 10ms. Spec-Augment is applied 2 frequency masks with maximum frequency mask (F = 30) and 2 time masks with maximum time mask (T = 50) to alleviate overfitting. A global mean and variance normalization is used for data preparation.

4.2. Model configuration

The MoE-Conformer model consists of 12 MoE-Conformer blocks ($d_{ff} = 2048, n_{head} = 8, d_{att} = 512, CNN_{kernel} = 31$) in the encoder and 6 Transformer blocks ($d_{ff} = 2048, n_{head} = 8, d_{att} = 512$) in the decoder and the number of experts in MoE layer is 32. The shared embedding network consists of 6 Conformer encoder blocks and is pretrained with 4 Transformer decoder blocks to initial the MoE-Conformer. Each expert is a feed-forward network with two hidden layer of size 2048 activated by Swish. For the self-attention layer, we set the model dimension $d = 512$ and the number of heads $h = 8$. The Dense-Conformer model comprises 12 Conformer encoder blocks and 6 Transformer decoder blocks. All student

models in the paper are initialized by their teacher models. The second Macaron-FFN layers in MoE's student model are initialized by the most frequently used expert counted in the valid dataset in teacher's MoE layers. We don't quantize the convolutional kernel because it is very small and causes great performance damage[17]. Similar to BWN, our experiment uses a ternary weight network (TWN[22]) to quantize weights to 2 bits. The max number of epochs is 160. We set the average of last 30 models as the final model for testing. We use dynamic batch and the max number of frames in batch is 20000. We train models with learning rate of 0.001, except for a one-stage pipeline to obtain the best performance by searching the learning rate of 0.005.

4.3. Results of two-stage pipeline

In this section, we evaluate the performance of the binary Dense-Conformer model distilled by two stages. First, we distill the MoE-Conformer teacher model into the Dense-Conformer. In Table 1, we can see that the Dense-Conformer has a lower WER than the Conformer baseline model and achieves comparable performance as MoE-Conformer. We can reduce the model size by 7.5x compared to the MoE-Conformer.

Table 1: *Results of Dense-Conformer*

Model	test-clean	test-other	Model Size
Conformer	2.97 %	7.46 %	492 MB
MoE-Conformer	2.91 %	6.98 %	3.6 GB
Dense-Conformer	2.91 %	7.07 %	492 MB

Moreover, we evaluate the performance of binary Dense-Conformer and the results are shown in Table 2. The 'W**M**A**N**' means quantizing weights to M bits and quantizing activations to N bits. When N is equal to 32, it means that the activations are not quantized. From the results, we can see that our method can obtain a binary Dense-Conformer model with a small performance loss and 20x size reduction. We analyze the reasons for the decline in model performance and find that the quantizatin of weights, especially from 32 bits to 2 bits, is the main reason for the performance decline. For example, the WER of W2A32 increases by 10.31% compare with FP32 Dense-Conformer model on test-clean and 11.46% on test-other. In contrast, the performance gap between the model with 2-bit weights and the model with 1-bit is smaller. Specifically, WER of W1A32 increases by 4.67% compared with W2A32 on test-clean and WER of W1A4 increases by 6.31% compared with W2A4 on test-clean. The performance loss caused by activation quantization is even smaller. Specifically, WER of W2A4 increases by 3.74% compared with W2A32 on test-clean and WER of W1A4 increases by 5.36% compared with W1A32 on test-clean. The results show that the representation ability of the 2-bit model has been greatly damaged, and there is still room for development and progress in the quantization of the ultra-low bit of the speech recognition models.

Table 2: *Results of binary Dense-Conformer in two-stage pipeline*

Model	test-clean	test-other	Model Size
Dense-Conformer FP32	2.91 %	7.07 %	492 MB
2S-QT W2A32	3.21 % (+10.31 %)	7.88 % (+11.46 %)	39 MB
2S-QT W2A4	3.33 % (+14.43 %)	8.27 % (+16.97 %)	39 MB
2S-QT W1A32	3.36 % (+15.46 %)	8.36 % (+18.25 %)	24 MB
2S-QT W1A4	3.54 % (+21.65 %)	8.71 % (+23.20 %)	24 MB

4.4. Results of one-stage pipeline

In this section, we present the results of one-stage compressing method that obtain the binary Dense-Conformer from the MoE-Conformer directly. In Table 3, we can see that the proposed one-stage compressing method can achieve 150x compressing rate with comparable performance compared to the 2S-QT process. In addition, we also compare one-stage pipeline with other quantization methods. We emphasize that the proposed method quantizes both weights and activations to low bits, which helps to compress the size of the model and improve computing efficiency. The BOPs[23] of quantized W1A4 layer is about 1/256 (256=32*8) of FP32 layer. The method of Refs.[24, 25]only quantizes the weights and the calculation cannot be significantly accelerated by quantizing the weight alone. Besides, from prior results (such as the baseline Refs.[15] in Table 2), it can observe that when the weights are quantized to more than 8 bits and activations to more than 8 bits, model quantization will hardly cause performance loss. However, when it is further quantized to a lower bit width, performance is severely declined. In summary, our proposed method provides a more compressing rate and lower performance loss compared to the methods of Refs.[15, 24]. Besides, we can observe that 2S-QT W2A4 with 39MB in Table 2 significantly outperform the performance of W6A8 with 93MB in Ref.[15].

Table 3: *WER comparison with different quantization methods*

Model	test-clean	test-other	Model Size
Dense-Conformer FP32	2.91 %	7.07 %	492 MB
2S-QT W1A4	3.54 % (+21.65 %)	8.71 % (+23.20 %)	24 MB
MoE-Conformer FP32	2.91 %	6.98 %	3.6 GB
1S-QT W1A4 (ours)	3.55 % (+21.99 %)	8.77 % (+25.64 %)	24 MB
FP32[24]	8.68 %	22.29 %	240 MB
W8A32[24]	8.70 % (+0.23 %)	22.36 % (+0.31 %)	60 MB
W6A32[24]	8.90 % (+2.53 %)	22.82 % (+2.38 %)	45 MB
W5A32[24]	9.76 % (+12.44 %)	24.10 % (+8.12 %)	38 MB
W4A32[24]	16.43 % (+89.29 %)	35.69 % (+60.12 %)	30 MB
FP32[15]	2.78 %	6.19 %	495 MB
W8A8[15]	3.06 % (+10.07 %)	7.06 % (+14.05 %)	124 MB
W6A8[15]	4.03 % (+44.96 %)	8.48 % (+37.00 %)	93 MB

5. Conclusions and Future Works

In this paper, we propose an extreme compression method based on knowledge distillation and quantization for building a quantized ASR model with 150x compressing rate that can directly compress from a float-point MoE ASR model to a binary Dense ASR model. Through several experiments on LibriSpeech, we can observe that one-stage compression shortened the pipeline can achieve comparable performance with two-stage compression. Thus, we can deploy compressed models derived from complicated MoE models on embedded devices. The second stage of the two-stage pipeline shows that our quantization method is not limited to the MoE models. In the future work, we expect to achieve a larger compression ratio with layer-reduction knowledge distillation and convolution quantization. As the one-stage compression, we find that the configuration of learning rate is the key factor for final compressing performance and hope for higher accuracy with a learning rate search.

6. References

[1] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 5884–5888.

[2] A. Gulati, J. Qin, C. C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, 2020, pp. 5036–5040.

[3] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," *arXiv preprint arXiv:2006.16668*, 2020.

[4] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232–5270, 2022.

[5] S. Rajbhandari, C. Li, Z. Yao, M. Zhang, R. Y. Aminabadi, A. A. Awan, J. Rasley, and Y. He, "Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale," in *International Conference on Machine Learning*, 2022, pp. 18 332–18 346.

[6] S. Gross, M. Ranzato, and A. Szlam, "Hard mixtures of experts for large scaleweakly supervised vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5085–5093.

[7] K. Ahmed, M. H. Baig, and L. Torresani, "Network of experts for large-scale image categorization," in *European Conference on Computer Vision*, 2016, pp. 516–532.

[8] X. Wang, F. Yu, L. Dunlap, Y.-A. Ma, R. Wang, A. Mirhoseini, T. Darrell, and J. E. Gonzalez, "Deep mixture of experts via shallow embedding," in *Uncertainty in artificial intelligence*, 2020, pp. 552–562.

[9] S. Cai, Y. Shu, and W. Wang, "Dynamic routing networks," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3588–3597.

[10] Z. You, S. Feng, D. Su, and D. Yu, "Speechmoe: Scaling to large acoustic models with dynamic routing mixture of experts," in *INTERSPEECH 2021 – 22nd Annual Conference of the International Speech Communication Association*, 2021, pp. 2077–2081.

[11] ——, "Speechmoe2: Mixture-of-experts model with improved routing," in *International Conference on Acoustics Speech and Signal Processing*, 2022, pp. 7217–7221.

[12] ——, "3m: Multi-loss, multi-path and multi-level neural networks for speech recognition," in *13th International Symposium on Chinese Spoken Language Processing*, 2022, pp. 170–174.

[13] K. Kumatani, R. Gmyr, F. C. Salinas, L. Liu, W. Zuo, D. Patel, E. Sun, and Y. Shi, "Building a great multi-lingual teacher with sparsely-gated mixture of experts for speech recognition," *arXiv preprint arXiv:2112.05820*, 2021.

[14] F. C. Salinas, K. Kumatani, R. Gmyr, L. Liu, and Y. Shi, "Knowledge distillation for mixture of experts models in speech recognition," Microsoft, Tech. Rep. MSR-TR-2022-6, May 2022.

[15] S. Kim, A. Gholami, Z. Yao, N. Lee, P. Wang, A. Nrusimha, B. Zhai, T. Gao, M. W. Mahoney, and K. Keutzer, "Integer-only zero-shot quantization for efficient speech recognition," in *International Conference on Acoustics Speech and Signal Processing*, 2022, pp. 4288–4292.

[16] C.-F. Yeh, W.-N. Hsu, P. Tomasello, and A. Mohamed, "Efficient speech representation learning with low-bit quantization," *arXiv preprint arXiv:2301.00652*, 2022.

[17] S. Ding, P. Meadowlark, Y. He, L. Lew, S. Agrawal, and O. Rybakov, "4-bit conformer with native quantization aware training for speech recognition," in *INTERSPEECH 2022 – 23rd Annual Conference of the International Speech Communication Association*, 2022, pp. 1711–1715.

[18] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *14th European Conference on Computer Vision*, 2016, pp. 525–542.

[19] S. K. Esser, J. L. McKinstry, D. Bablani, R. Appuswamy, and D. S. Modha, "Learned step size quantization," *arXiv preprint arXiv:1902.08153*, 2019.

[20] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *International Conference on Acoustics Speech and Signal Processing*, 2015, pp. 5206–5210.

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

[23] M. van Baalen, C. Louizos, M. Nagel, R. A. Amjad, Y. Wang, T. Blankevoort, and M. Welling, "Bayesian bits: Unifying quantization and pruning," *Advances in neural information processing systems*, vol. 33, pp. 5741–5752, 2020.

[24] H. D. Nguyen, A. Alexandridis, and A. Mouchtaris, "Quantization aware training with absolute-cosine regularization for automatic speech recognition," in *INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, 2020, pp. 3366–3370.

[25] K. Zhen, H. D. Nguyen, R. Chinta, N. Susanj, A. Mouchtaris, T. Afzal, and A. Rastrow, "Sub-8-bit quantization aware training for 8-bit neural network accelerator with on-device speech recognition," in *INTERSPEECH 2022 – 23rd Annual Conference of the International Speech Communication Association*, 2022, pp. 3033–3037.