INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

2-bit Conformer quantization for automatic speech recognition

Oleg Rybakov, Phoenix Meadowlark, Shaojin Ding, David Qiu, Jian Li, David Rim, Yanzhang He

Google Research

{rybakov,meadowlark,shaojinding,qdavid,jianlijianli,davidrim,yanzhanghe}@google.com

Abstract

Large speech models are rapidly gaining traction in research community. As a result, model compression has become an important topic, so that these models can fit in memory and be served with reduced cost. Practical approaches for compressing automatic speech recognition (ASR) model use int8 or int4 weight quantization. In this study, we propose to develop 2-bit ASR models. We explore the impact of symmetric and asymmetric quantization combined with sub-channel quantization and clipping on both LibriSpeech dataset and large-scale training data. We obtain a lossless 2-bit Conformer model with 32% model size reduction when compared to state of the art 4-bit Conformer model for LibriSpeech. With the large-scale training data, we obtain a 2-bit Conformer model with over 40% model size reduction against the 4-bit version at the cost of 17% relative word error rate degradation.

Index Terms: speech recognition, model quantization, low-bit quantization

1. Introduction

Modern automatic speech recognition models are mostly based on an end-to-end approach [1, 2, 3, 4]. One of the popular methods to improve accuracy of such models is to increase model size [5]. With the growing success and size of these models, compressing them with neutral quality impact is becoming an important research topic.

The most popular approaches for compressing neural networks are pruning [6, 7], knowledge distillation [8], and quantization [9, 10]. In this paper we are focused on quantization as the most straight forward approach. It can be applied on activations and weights. If both activations and weights are quantized [11], then it reduces memory footprint and can give speed up due to memory footprint reduction and low bits multiplication (the latter one requires hardware support). If quantization is applied only on weights then it reduces memory footprint and can provide speed up due to lower memory usage. It also does not require special hardware support for low bits numbers multiplication. That is why in this work we are focused on weights only quantization.

Quantization methods can be divided into post training quantization (PTQ) and quantization aware training (QAT). PTQ is successfully applied on speech applications [12, 13] because it is easy to use (e.g. with TFLite [14]) and it works well at int8 precision. PTQ with lower bits has limited support in TFLite [14] and can have significant accuracy degradation with no or minimal tools to address it. Hence, we are focusing on quantization aware training. QAT can be applied after model is pre-trained (i.e., fine-tuning stage), or it can be applied from the beginning of the model training (i.e., QAT training from scratch). In this work we are focused on training from scratch, although our approach can be used for fine-tuning too. Quantization of a tensor can be done with dynamic quantization [15] or static quantization [16, 17]. In this work we are focused on dynamic quantization because it does not require additional variables during training and it works well for speech applications e.g. [18]. Tensor can be quantized using non-uniform quantization (e.g., with float8 [19]). Not all hardware supports it, so in this work we are focused on uniform 2-bit integer quantization, also called fixed-point quantization [16].

QAT with 4-bit is successfully applied on multiple speech models [20, 21, 18, 22, 23] with minimal accuracy impact. Lower than 4-bit weight quantization is explored for different applications [24, 25, 26], but there is not much research on 2-bit quantization of ASR models. One work [27] addresses lower than 4-bit quantization, but the authors showed significant accuracy degradation with 2-bit and 4-bit quantization. Note that in [27], the authors quantize both activation and weights. Here we are focused on 2-bit Conformer weights only quantization with minimal accuracy impact. Our main contributions are outlined as below:

- We present a new 2-bit asymmetric dynamic sub-channel QAT technique with adaptive per channel clipping (based on greedy search). It is open sourced in Praxis [28].
- We benchmark several proposed approaches of QAT and demonstrate that Conformer ASR on LibriSpeech data shows minimal or no accuracy loss with 2-bit weights when comparing to state of the art float model. We reduced model size by 32% relative to the 4-bit model and establish state of the art ASR model in terms of model size and WER.
- We evaluate the effectiveness of the best 2-bit setup on a Conformer model that is trained on large-scale datasets. We achieve over 40% model size reduction against the 4-bit version at the cost of 17% relative word error rate degradation.

2. Quantization aware methods

2.1. Symmetric quantization: *I2Wsym*

The standard method of weights only QAT is based on symmetric quantization [29]. State of the art 4-bit symmetric quantization is presented in [18]. We use approach described in [18] as the baseline, configure it for 2-bit quantization, and label it as *I2Wsym*. Note that 2-bit symmetric quantization under-utilizes the 2-bit quantization buckets (because it uses only three values). As a result, symmetric quantization can degrade accuracy.

2.2. Asymmetric quantization: *I2Wasym*

Asymmetric quantization [29] allows us to use all four quantization buckets by estimating the minimum value and subtracting it from the input tensor. We label 2-bit asymmetric quantization as *I2Wasym*. In Figure1, we show an example with per channel

10.21437/Interspeech.2023-1012

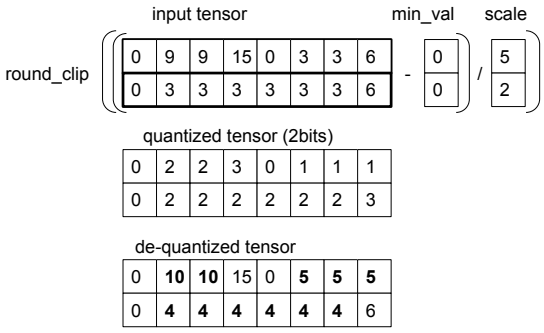

Figure 1: *Per-channel asymmetric quantization.*

asymmetric quantization. The input tensor (weight matrix) has two rows (every row is called a channel). We quantize it using the function *quantize* (shown on Figure 5). It computes min value (*min_val*) for every row (using function *scale_and_min* from Figure 5) then subtracts it from input tensor and divides by scale. In Figure 1, "quantized tensor" is the output of function *round_clip*, which rounds and clips the tensor between zero and three (according to 2-bit quantization range). The de-quantized tensor is shown at the bottom of Figure 1. It is computed using the *dequantize* function (shown in Figure 5). As we can see in Figure 1, most de-quantized values (highlighted by bold) are different from the original values in the input tensor. The reason for such quantization error is outliers in the input tensor. We use Figure 1 for illustration purposes. In Figure 2, we show quantization error of asymmetrical per channel quantization (blue curve in Figure 2) applied on an input tensor with size [32, N], where 32 is the number of channels and N can be 32, 64, 128, 256 or 512 (x-axis). The input tensor is filled with standard normal noise (with zero mean and unit variance), and then it is quantized and de-quantized. The quantization error (y-axis) is defined as the mean (over all entries) absolute difference between input tensor and corresponding de-quantized tensor.

2.3. Asymmetric quantization with scale backpropagation: *I2WasymSc*

In [18], the authors used full Straight-Through Estimator(STE) for quantization aware training of 4-bit ASR. We call it full STE because the gradient did not propagate through round and scale computation. To improve model quality we enable gradient over scale [30] and set *stop_gradient* equal false in the function *scale_and_min* in Figure 5. This approach allows to reduce outliers as described in [30]. 2-bit asymmetric quantization with scale backpropagation will be labeled as *I2WasymSc*. All subsequent QAT methods will use backpropagation over scale as well.

2.4. Asymmetric quantization with scale backpropagation, sub-channel and adaptive clipping: *I2WasymScSubchClip*

One of the methods of dealing with the outliers in the input tensor (model weights) is based on sub-channel quantization [31], which is also similar to group based quantization [32, 33]. The key idea is to split a channel into several sub-channels and then quantize them independently. It can introduce additional overhead because with more sub-channels, we will need to keep more quantization metadata: scales and minimum values. An example of such an approach is shown in Figure 3. The input tensor with two channels is reshaped, so that there are 4 sub-channels. Afterwards, the same quantization operations, described on Figure1, are applied. As we can see, the de-quantized

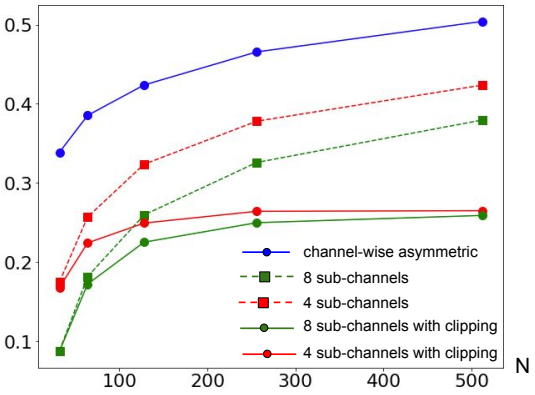

Figure 2: *The per-entry mean absolute quantization error plotted against the size of the input dimension.*

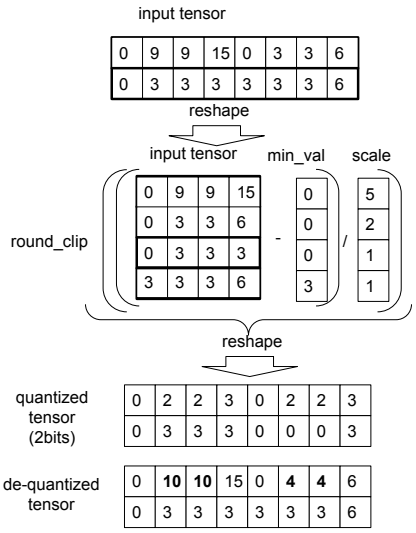

Figure 3: *Sub-channel quantization.*

tensor has only four different values (highlighted by bold on Figure 3) when compared against the input tensor. In Figure 2 we show the quantization error of this approach when splitting into 4 sub-channels (red dashed line) and 8 sub-channels (green dashed line). As expected, the quantization error with 4 sub-channels is lower than that of channel-wise asymmetric quantization (described in section 2.2). Furthermore, the quantization error becomes even lower after increasing the number of sub-channels to 8.

Another approach to reduce the number of the outliers in the input tensor is based on clipping. For example, in PACT [15], the authors propose to learn clipping parameters to improve the quality of activation quantization. In [16, 34], the clipping value is estimated based on percentile. In [34], the authors design OCTAV [34] algorithms for online estimation of clipping values. In this work, we propose to use greedy search for clipping parameter estimation. Hence, we combine sub-channel quantization with clipping as demonstrated on Algorithm 1, and label it as *I2WasymScSubchClip*. The input tensor (weight matrix) is reshaped so that the channel dimension is divided into several sub-channels. For example, in Figure 4, two input channels are divided into four sub-channels. Then we run *greedy_search* over clipping values in the range of $[0.5, 1.0]$ with 0.05 step,

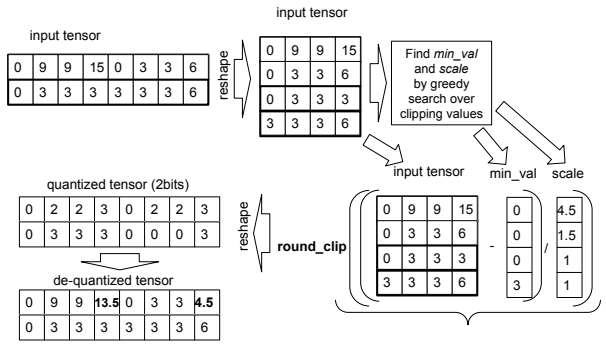

Figure 4: *Sub-channel quantization with clipping per sub-channel.*

and compute mean absolute error between the input weights and the quantized/de-quantized weights (for quantization and de-quantization we call functions presented on Figure 5). We apply different clipping values per sub-channel and select the quantized tensor with clipping values that correspond to the minimal quantization error. An example of quantized tensor (generated by Algorithm 1) is shown in Figure 4. As we can see, the combination of sub-channel quantization with greedy search over clipping parameter allows us to further reduce quantization error. In this example, the de-quantized tensor has only two numbers that are different from the input tensor. In Figure 2, we show the quantization error of these approaches: 4 sub-channels with clipping (solid red line) and 8 sub-channels with clipping (solid green line). We observe that sub-channel quantization with clipping has the lowest quantization error, and we hypothesize that this approach will be useful for low bit quantization aware training.

Algorithm 1 Sub channel quantization with clipping

procedure QUANTIZESUBCHCLIP(w) ▷ Input weights
  $input\_shape \leftarrow w.shape$ ▷ Weights shape
  $w\_sub \leftarrow reshape(w)$ ▷ Split into sub channels
  $w\_q, scale, min\_val \leftarrow greedy\_search(w\_sub)$
  $w\_deq \leftarrow dequantize(w\_q, scale, min\_val)$
  $w\_deq \leftarrow reshape(w\_deq, input\_shape)$
  return w_deq ▷ De-quantized weights
end procedure

3. Experimental setups

3.1. Datasets

Similar to [18], we use LibriSpeech [35] to conduct QAT experiments. The LibriSpeech training set contains 960 hours of speech, where 460 hours of them are "clean" speech and the other 500 hours are "noisy" speech. We use "dev-clean" data to select the best model and then report its accuracy on "test-clean" and "test-other" data sets.

In terms of the experiments with large-scale datasets, we train the models with an in-house training set consisting of ∼1000 million United States English audio-text pairs from multiple domains, such as YouTube, search, and dictation. A small portion of the dataset is anonymized and hand-transcribed, while the rest is pseudo-transcribed with a 600M-parameter teacher system [36]. Word error rates are reported on 5.5K anonymized and hand-transcribed utterances representative of voice search traffic.

```python
@tf.custom_gradient
def round(x):
  # Use STE for gradient.
  return tf.math.floor(x + 0.5), lambda dy: dy

def round_clip(x, prec):
  x = round(x)
  x = tf.clip_by_value(x, 0, 2.**prec - 1)
  return x

def scale_and_min(x, prec, axis, clipping =
    1.0, stop_gradient_scale=False):
  min_val = tf.math.reduce_min(x, axis=axis,
    keepdims=True)
  max_val = tf.math.reduce_max(x, axis=axis,
    keepdims=True)

  min_val = tf.multiply(min_val, clipping)
  max_val = tf.multiply(max_val, clipping)
  scale = tf.divide(max_val-min_val,2.**prec-1)

  if stop_gradient_scale: # STE over scale
    scale = tf.stop_gradient(scale)
  return scale, min_val

def quantize(x, prec, axis, clipping=1.0,
    stop_gradient_scale=False):
  scale, min_val = scale_and_min(x, prec, axis,
      clipping, stop_gradient_scale)
  x = x - min_val
  x = tf.math.divide_no_nan(x, scale)
  qx = round_clip(x, prec)
  return qx, scale, min_val

def dequantize(qx, scale, min_val):
  deqx = qx * scale
  deqx = deqx + min_val
  return deqx
```

Figure 5: *Quantization functions*

3.2. Details in Conformer model architecture

We use the same state-of-the-art Conformer Transducer [5] backbones as in [18] for experiments on LibriSpeech and large-scale datasets. For LibriSpeech experiments, the model has a single encoder with different number of layers for *Small* (16 layers, 10M parameters) and *Large* (17 layers, 118M) models. The decoder is a standard RNN-Transducer decoder with 1-layer LSTM. The backbone of the experiments with large-scale data is based on [37], consisting of a 7-layer causal conformer encoder (23-frame left context) and a 6-layer non-causal encoder (additional 30-frame right context). Each RNN-T decoder is comprised of an embedding prediction network and a fully-connected joint network.

4. Results

4.1. Experiments on LibriSpeech

We experiment with the Conformer *Large* and *Small* models to examine the behaviors of 2-bit QAT with different model sizes. We hypothesize that the larger the model size, the easier to quantize its weights with no accuracy loss. The Conformer ASR model size is dominated by the Conformer blocks of the encoder, which account for 95% and 82% of the *Large* and *Small* models' disk utilization, respectively. For small models, decoder quantization disproportionately impacts model quality,

Table 1: *Results of our proposed int2 QAT on Conformer Large(L) and Small(S) models with the baseline approaches on LibriSpeech test-clean and test-other subsets. Please see Section 4.1 for the meanings of the method abbreviations.*

Conformer (L)			
Method	**test-clean**	**test-other**	**Model size (MB)**
[18] Float	**2.0**	**4.4**	474.5
I2Wsym	3.6	8.1	53.8
I2Wasym	2.2	5.0	54.0
I2WasymSc	2.2	4.6	54.0
I2WasymScSubchClip	**2.0**	**4.5**	55.3
[18] I4W	**2.0**	**4.4**	81.9
[38] I6W8A	4.0	8.5	92.8
[38] I8W	3.1	7.1	123.7
[18] I8W	**2.0**	**4.5**	138.1
Conformer (S)			
Float	**2.5**	**6.1**	41.5
I2Wsym	8.7	19.6	10.1
I2Wasym	4.3	10.3	10.2
I2WasymSc	3.1	7.3	10.2
I2WasymScSubchClip	3.1	7.0	10.5
[18] I4W	2.7	6.3	12.2
[18] I8W	**2.5**	**6.0**	16.4
Other Architectures			
[39] I8W	8.7	22.3	60
[39] I6W	8.9	22.8	45
[40] I8WA	6.9	—	8

while for large models the theoretical benefit of using decoder quantization becomes increasingly negligible. Hence, we decided to emulate [18]'s approach and implement encoder-only quantization. As in [18], we additionally opted to exclude quantization of the depthwise convolutional layers, as their contribution to model size is negligible. All other weights in the encoder's Conformer blocks are quantized with 2 bits, and their disk utilization is calculated assuming weight packing of 0.25 bytes/param.

We evaluate several 2-bit QAT configurations to see the impact of symmetric, asymmetric quantization, backpropagation on scale, and sub-channel quantization with clipping:

- *Float*: float32 weight, float32 activation (baseline)
- *I2Wsym*: int2 weight with symmetrical quantization (we applied the approach in [18]).
- *I2Wasym*: int2 weight with per-channel asymmetrical quantization, described in section 2.2, with full straight through estimator (stop_gradient_scale is set to True) as in [18].
- *I2WasymSc*: int2 weight with per-channel asymmetrical quantization, described in section 2.3 and partial straight through estimator: with stop_gradient_scale is set to False, so that scale is part of the backpropagation.
- *I2WasymScSubchClip*: int2 weight with per-channel asymmetrical quantization combined with sub-channel and clipping greedy search presented as Algorithm 1 and described in section 2.4. As *I2WasymSc*, it uses partial straight through estimator: with stop_gradient_scale is set to False, so that scale is part of the backpropagation. We use 4 sub-channels for Conformer (L) and 8 for Conformer (S). Clipping greedy search was done in a range from 0.8 to 1.0 with 0.02 step.

We train Conformer(L) and Conformer(S) models with all above QAT configurations on 64 TPUs. Conformer(L) model converges with 150k training steps (it takes 2.3 days). Conformer(S) model converges with 400k steps (it takes 4.8 days). Note that QAT with *I2WasymScSubchClip* is 40% slower than other models due to the greedy search algorithm, but other presented approaches have no impact on training speed). In Table 1, we report the word error rate(WER) of Conformer(L) and Conformer(S) models trained with quantization techniques: *I2Wsym*, *I2Wasym*, *I2WasymSc* and *I2WasymScSubchClip*. QAT with asymmetric quantization (*I2Wasym*) allows us to use all four quantization buckets and reduce WER by 3% absolute (in comparison to *I2Wsym*) on Conformer(L) "test-other" data. QAT with asymmetric quantization and enabled backpropagation over scale (*I2WasymSc*) further reduces WER by 0.4% absolute on Conformer(L) "test-other" data. Combining *I2WasymSc* with sub-channel and clipping greedy search *I2WasymScSubchClip* gives the same WER with the state of the art int8 weights quantization model [18], which is only 0.1% worse than the float baseline on Conformer(L) "test-other" data. On 2-bit ASR model quantization we observe that the larger the model, the easier it is to quantize its weights with minimal accuracy loss.

Table 2: *Results of applying int2 QAT to production ASR model on large-scale data.*

Exp	Model	WER	Model size (MB)
B0	float32 model	6.0	480
B1	int4	6.3	65
E0	int2	12.6	37.5
E1	int2 I2WAsymScSubchClip	7.6	37.5
E2	int2 I2WAsymScSubchClip + MSQE [41]	7.4	37.5

4.2. Exploring the limit of 2-bit quantization on large data

As shown in Table 2, when training the model with large-scale datasets, 4-bit quantization [18] (*B1*: 6.3) can mostly retain the float model (*B0*: 6.0) performance with minimal regression, which corresponds to the observations in [18]. However, quantizing the model to 2-bit becomes increasingly challenging, as the model is usually under-fitting. If we simply apply ASR quantization from [18] for 2-bit QAT to the model (*E0*), there is a significant WER regression compared to the float model (12.6 vs. 6.0). Alternatively, if we train the model with our best setup obtained from Section 4.1 (*E1*; i.e., asymmetric quantization, backpropagation on scale, and sub-channel quantization with clipping), the WER has been from 12.6 to 7.6 over the naive 2-bit QAT version *E0*. In addition, we add an extra quantization regularization MSQE [41] term *E2* as MSE between weights and de-quantized weights. It can further mitigate the gap between 2-bit and float model by 0.2%. In summary, our best performance 2-bit model *E2* has a 1.4% WER gap compared to the float model but with over 90% of model size saving, compared to the float model. In terms of the 4-bit model *B1*, the 2-bit model *E2* has a 1.1% WER gap but achieves over 40% of model size reduction.

5. Conclusion

We proposed a novel approach of 2-bit QAT based on dynamic asymmetrical sub-channel quantization with adaptive per channel clipping. We reduced model size down to 55MB with minimal or no accuracy loss and established state of the art ASR model in terms of model size and WER. We also showed that the larger the model ($>$ 100M parameters), the easier it is to quantize its weights with 2bit with no accuracy loss (it is important for large speech models). When training the model with large-scale datasets, we illustrated the inevitable WER regression from the 2-bit model, and we showed that our proposed techniques can significantly mitigate the gap between 2-bit model and the float counterpart.

6. References

[1] J. Li *et al.*, "On the comparison of popular end-to-end models for large scale speech recognition," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association*. ISCA, 2020, pp. 1–5.

[2] C. Chiu *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2018.

[3] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017*. IEEE, 2017, pp. 4835–4839.

[4] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019*. IEEE, 2019, pp. 114–121.

[5] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association*.

[6] N. Ström, "Sparse connection and pruning in large dynamic artificial neural networks," in *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997*. ISCA, 1997.

[7] R. Takeda, K. Nakadai, and K. Komatani, "Node pruning based on entropy of weights and node activity for small-footprint acoustic model based on deep neural networks," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association*.

[8] C. Li *et al.*, "Compression of acoustic model via knowledge distillation and pruning," in *24th International Conference on Pattern Recognition, ICPR 2018*.

[9] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *4th International Conference on Learning Representations, ICLR 2016*.

[10] R. Alvarez, R. Prabhavalkar, and A. Bakhtin, "On the efficient representation and execution of deep acoustic models," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association*, 2016.

[11] I. Hubara *et al.*, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Mach. Learn. Res.*, 2017.

[12] Y. He *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[13] T. N. Sainath *et al.*, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[14] [Online]. Available: https://www.tensorflow.org/lite/performance/post_training_quantization

[15] J. Choi *et al.*, "Pact: Parameterized clipping activation for quantized neural networks," 2018.

[16] H. Wu *et al.*, "Integer quantization for deep learning inference: Principles and empirical evaluation," 2020.

[17] A. Abdolrashidi *et al.*, "Pareto-optimal quantized resnet is mostly 4-bit," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021*.

[18] S. Ding *et al.*, "4-bit conformer with native quantization aware training for speech recognition," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association*. ISCA, 2022.

[19] N. Wang *et al.*, "Training deep neural networks with 8-bit floating point numbers," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*.

[20] Y. Mishchenko *et al.*, "Low-bit quantization and quantization-aware training for small-footprint keyword spotting," in *18th IEEE International Conference On Machine Learning And Applications, ICMLA 2019*, 2019.

[21] A. Fasoli *et al.*, "4-bit quantization of lstm-based speech recognition models," in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*. ISCA, 2021, pp. 2586–2590. [Online]. Available: https://doi.org/10.21437/Interspeech.2021-1962

[22] O. Rybakov *et al.*, "Streaming parrotron for on-device speech-to-speech conversion," 2022.

[23] K. Zhen *et al.*, "Sub-8-bit quantization for on-device speech recognition: A regularization-free approach," in *IEEE Spoken Language Technology Workshop, SLT 2022*.

[24] M. Courbariaux *et al.*, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Advances in Neural Information Processing Systems 28: Annual Conference, 2015*.

[25] O. Shayer, D. Levi, and E. Fetaya, "Learning discrete weights using the local reparameterization trick," in *6th International Conference on Learning Representations, ICLR 2018*.

[26] M. Rastegari *et al.*, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Computer Vision - ECCV 2016 - 14th European Conference*, ser. Lecture Notes in Computer Science, 2016.

[27] C.-F. Yeh, W.-N. Hsu, P. Tomasello, and A. Mohamed, "Efficient speech representation learning with low-bit quantization," 2023.

[28] "praxis: https://github.com/google/praxis."

[29] M. Nagel *et al.*, "A white paper on neural network quantization," 2021.

[30] D. Qiu, D. Rim, S. Ding, O. Rybakov, and Y. He, "Rand: Robustness aware norm decay for quantized seq2seq models," 2023.

[31] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," 2021.

[32] H. Yu *et al.*, "Low-bit quantization needs good distribution," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops*.

[33] Z. Yuan *et al.*, "Ptq-sl: Exploring the sub-layerwise post-training quantization," 2021.

[34] C. Sakr *et al.*, "Optimal clipping and magnitude-aware differentiation for improved quantization-aware training," in *International Conference on Machine Learning, ICML 2022*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022.

[35] V. Panayotov *et al.*, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[36] D. Hwang *et al.*, "Pseudo label is better than human label," in *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association*. ISCA, 2022.

[37] S. Ding *et al.*, "A unified cascaded encoder ASR model for dynamic model sizes," in *Interspeech 2022, 23rd Annual Conference of the International Speech*. ISCA, 2022, pp. 1706–1710.

[38] S. Kim *et al.*, "Integer-only zero-shot quantization for efficient speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022*.

[39] 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*.

[40] A. Prasad, P. Motlicek, and S. Madikeri, "Quantization of acoustic model parameters in automatic speech recognition framework," *arXiv preprint arXiv:2006.09054*, 2020.

[41] Y. Choi, M. El-Khamy, and J. Lee, "Learning sparse low-precision neural networks with learnable regularization," *IEEE Access*, vol. 8, pp. 96 963–96 974, 2020.