Interspeech 2024
1-5 September 2024, Kos, Greece

USM RNN-T model weights binarization

Oleg Rybakov, Dmitriy Serdyuk, Chengjian Zheng

Google LLC, USA

{rybakov,dserdyuk,cjzheng}@google.com

Abstract

Large-scale universal speech models (USM) are already used in production. However, as the model size grows, the serving cost grows too. Serving cost of large models is dominated by model size that is why model size reduction is an important research topic. In this work we are focused on model size reduction using weights only quantization. We present the weights binarization of USM Recurrent Neural Network Transducer (RNN-T) and show that its model size can be reduced by 15.9x times at cost of word error rate (WER) increase by only 1.9% in comparison to the float32 model. It makes it attractive for practical applications.

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

1. Introduction

In the last several years the size of automatic speech recognition (ASR) models increased by more than 10x: from hundreds of millions [1, 2] to several billions parameters [3, 4, 5]. Serving cost goes up with model size increase, that is why ASR models compression is a hot research topic. Sparse network pruning [6, 7, 8, 9, 10] as well as quantization [11, 12, 13, 14, 15] are successfully applied on end-to-end ASR. In [13, 14] authors show that it is possible to quantize 160M parameters model (trained on 900 hours of Librispeech data [16]) with 4 bits and 2 bits with no accuracy loss. But at the same time they show that a 160M parameters model quantized on a production data set (with millions of hours of speech) has 1.4% (absolute) WER degradation for 2 bits quantization [14]. Accuracy reduction can be even higher, e.g. in [17, 15] for 2 bits weights only quantization WER is increased by several times. To address this issue, authors in [15] explored the combination of quantization with sparsity. Based on these observation we can conclude that 4 bits weights quantization can be quality neutral on most of ASR models, 2 bits quantization works on some set of models (it depends on both model and data size and can require more experiments) and as expected 1 bit quantization (aka binarization) is the most challenging (e.g. authors of [17] reported 3x accuracy degradation), so it motivates us to investigate quantization aware weights binarization.

Models binarization is popular topic, e.g. it is explored for computer vision [18, 19, 20, 21], language [22, 23, 24] and speech recognition [25, 26, 17] applications. Standard weights binarization [18, 19, 25] is based on sign function to binarize weights to {-1, 1}. It uses Straight-Through Estimator to overcome non-differentiability of the sign function. There can be stochastic or deterministic sign function [18]. It can be combined with batch normalization [18] and *hard tanh* [25] to deal with the gradient. The output of the binarized weight can be scaled by *absmean* [19, 23]. As shown in [19] *absmean* is an optimal scaler for binarization problem. Group Quantization [23] or sub-channel quantization [14] can be applied to parallelize weights quantization (and it also can reduce quantization error). In this work we propose a combination of sub-channel quantization (also called block-wise [27]) with binarization based on *absmean* [19, 23, 22]. As in [22, 23] we use *absmean* binarization but without weights centralization. It simplifies model quantization and does not impact model accuracy in comparison to quantization with weights centralization.

Our main contributions are outlined as below:

- We simplify USM-CTC model training by switching to USM-RNNT model and reducing the number of training procedures from 4 to 2 with no accuracy loss (in comparison to baseline USM-CTC model). We show that with a reduced number of training procedures USM-RNNT is more friendly for quantization. With 2bits weights quantization USM-RNNT has only 1.4% WER increase in comparison to 3x WER degradation in the USM-CTC model.
- We develop a new approach of USM-RNNT model binarization. It combines sub-channel quantization with *absmean* binarization. It is open sourced at [28].
- We benchmark proposed binarization approach on USM RNN-T 1 Billion model trained on millions of hours of speech data and show that model size can be reduced by 15.9x times at cost of 1.9% (absolute) WER increase in comparison to the float32 baseline model.

The rest of the paper is organized as follows. In Section 2, we describe the baseline model, the RNN-T model we use for quantization, and the datasets. In Section 3, we present our quantization approaches. We report and discuss our experimental results in Section 4. Finally, we review related work in Section 5 and conclude in Section 6.

2. Model and Data

2.1. CTC-based Universal Speech Model

We base our work on Google Universal Speech Model (USM [5]). This is a single large model which performs ASR on a multitude of languages. This model uses a conformer encoder [2] and a CTC loss [29]. The USM uses a complex training procedure, therefore below we outline only the most important components.

The USM-CTC model consists of a conformer encoder which requires the 128-dimensional filter-bank features as input. A trainable language embedding is injected as a first token to the encoder. This allows the model to work with about 300 languages. The encoder features are fed into the CTC decoder.

The model is trained in a 4-step procedure:

10.21437/Interspeech.2024-47

1. First, a 600M parameter CTC model is trained on a small supervised set. This model is used as a teacher for the Noisy Student Training [30] procedure. Both supervised and unsupervised sets are relabeled with this model to produce pseudo labels.
2. Then, the encoder is pre-trained with BEST-RQ [31].
3. Then, the full model is pre-trained with the supervised relabeled subset obtained in Step 1.
4. Finally, the model is fine-tuned with task-specific data.

2.2. RNN-T-based Universal Speech Model

We expand the USM-CTC model and simplify its training.

We use the RNN-T loss [32, 33] which is known to have higher performance. In particular, we use a linear joint network and a 3-layer LSTM decoder. A downside of using the RNN-T instead of CTC is that the training may be less stable. We address this issue by using the Adafactor [34] optimizer. Furthermore, Adafactor allowed us to reduce the memory footprint of the model during training.

Using a stronger loss allowed us to exclude the BEST-RQ pre-training and the fine-tuning steps. Therefore, our approach has two steps:

1. Similarly to the USM-CTC, we use a 600M CTC model trained on a supervised set to relabel supervised and unsupervised speech datasets.
2. Then we simply train the RNN-T USM on the mix of all the supervised and semi-supervised data obtained in Step 1.

In this paper we use a USM RNN-T model with 893 million parameters, we call it the USM RNN-T 1B model. It has a good ratio of performance to number of parameters. The USM RNN-T model is composed of a mel spectrogram followed by a feature extractor followed by 16 conformer layers with a decoder. Mel spectrogram returns frames with 128 features (where every frame is generated per every 40ms with 20ms step). The feature extractor is composed of a convolution layer with kernel size (3, 3) channel size (128, 32) and stride (2, 2), followed by a normalization layer with ReLU activation function. These layers are repeated two times so that the input signal is subsampled by 4x in time dimension. After that output features are projected to 1536 dimensions and it is augmented with positional embedding. Output of the feature extractor is processed by sequence of conformer layers. Conformer layer parameters are: the model hidden size is equal to the input size (1536); the kernel size of convolution is 5; local self attention uses left and right context equal 129 and 128 accordingly; number of heads in the attention is 12. Conformer layer is open sourced at [28]. Output of the conformer layers is processed by a standard RNN-T decoder, which has a joint network and predictor. Where the predictor has 3 LSTM layers with 640 hidden units. The joint network also has 640 units. Decoder vocabulary size is 16,384.

In Table 1 we verify that our model is comparable to the previously published results. We conclude RNN-T simplified training produces results comparable to the baseline USM-CTC model.

2.3. Data

We used two datasets for training and one dataset for testing:

1. **Semi-supervised:** this is a large dataset extracted from YouTube videos. We extracted segments of lengths 30 seconds. We ensured that the extracted segments contain speech with high probability. This dataset contains videos of varying quality and a mix of formal and informal speech. The set contains 75 languages and its total length is about 4.2 million hours. We produce pseudo-labels for this set described in the procedure described in Section 2.1
2. **Supervised:** is a smaller dataset also extracted from YouTube videos. Similarly to the semi-supervised set, we extracted the segments and ensured that they contain speech. In contrast to the semi-supervised set, we used a heuristic approach to include only high quality videos. This dataset was labeled by the human transcribers. Then, we used it to train a teacher model (see Section 2.1). Finally, we produced the pseudo-labels labels for this set. We call the result a **supervised-relabeled** set. This set contains 56 languages and its total length is about 680,000 hours.
3. **YouTube test set:** the test set contains a combination of various topics for each given language. The human transcribers labeled this dataset.

Table 1: *Comparison of non-quantized models on the YouTube test set (WER, %). Note that the USM-CTC results are not strictly comparable to ours.*

Method	en_us	multi-lang
Published Baselines		
USM-CTC	13.7	26.7
Our Results		
USM-RNNT 1B	8.8	25.3
USM-RNNT 2B	8.3	24.9

3. Quantization approaches

We apply quantization-aware training on USM RNNT 1B model by quantizing weights of linear and projection attention layers in the Conformer encoder, during training of this model. Note that we keep convolution layers and 3 decoder RNN layers in float format because their size is negligible in comparison to the size of the encoder.

3.1. 2bits quantization

Several methods of 2 bits weights quantization are presented in [14] (they are based on *absmax* quantization). Here we explore two methods from [14]:

1. Asymmetric per channel quantization. It uses scale backpropagation [35]. In [14] it is labeled as *I2WasymSc*, here we tagged it as *Exp1*
2. Asymmetric per channel quantization with scale backpropagation, clipping and sub channel split. We use subchannel quantization with block size equal 64. In [14] it is labeled as *I2WasymScSubchClip*. Here we labeled it as *Exp2*.

Please refer to [14] for more details about the above approaches.

3.2. 1bit quantization

We explore several methods of model binarization. The first one is based on *absmax* asymmetric quantization, labeled as *I2WasymSc* in [14]. In this work we set the number of bits equal to 1 and apply static clipping, we label it as *Exp3*.

Another method of binarization is based on *absmean* [23, 22]. It is shown on Figure 1: input weights x can be centralized by subtracting per channel mean value (in line 10 on Figure 1

```
1  def absmean_binarize(
2      x,
3      contract_dims,
4      centralize=False
5  ):
6    if centralize:
7      mean = jnp.mean(x,
8                      axis=contract_dims,
9                      keepdims=True)
10     x = x - mean
11   x = jnp.where(x == 0.0, eps, t)
12   scale = jnp.mean(jnp.abs(x),
13                    axis=contract_dims,
14                    keepdims=True)
15   x = ste(x, jnp.sign)
16   return x, scale
17
18 def ste(x, fn):
19     return x - jax.lax.stop_gradient(x) +
20            jax.lax.stop_gradient(fn(x))
```

Figure 1: *Absmean binarization function.*

we disable it). Then weights x are binarized (in line 15) and the dequantization scale is estimated as per channel *absmean* (in line 14), that is why it is called *absmean* binarization. Note that we make all zeros equal to small epsilon number (in line 11), so that *sign* function returns only 1 and -1 (as shown on Figure 2 output of *absmean_binarize* has only 1 and -1 values). To deal with the gradient of a *sign* function we use straight-through estimator based on function *ste*, shown in line 18 in Figure 1. Note that we do not apply weights centralization as in [22, 23], because there was no accuracy difference and it simplifies model quantization. We labeled this approach as *Exp4*.

We combine *Exp4* binarization technique with sub channel quantization [14, 36], which has a predefined block size equal 64. We labeled this approach as *Exp5*. It is explained in Figure 2, where input weights have shape [2x6]. We increase the number of channels in the input weights by splitting weights into sub-channels with block size equal 3 (it is selected for illustration purpose). After that weight shape becomes [4x3]. These weights are quantized using *absmean* binarization approach [23, 22] as shown on Figure 1. We use fake quantization aware training, so binarized weights are dequantized by multiplying them with scale, as shown on Figure 2. Then the dequantized weights are reshaped back to the original weight shape: [2x6]. The last step is *einsum*, which is computed over input activation with dequantized weights. Note that during the inference dequantization step (multiplication by *scale*) is done on *einsum* output, also input activation will be reshaped to match the shape of binarized weights (its shape [4x3] as shown on Figure 2).

4. Experiments

4.1. Baseline USM RNN-T 1B model

USM RNNT 1 Billion model described in Section 2.2 is trained on 256 TPU v4 [37] for 1.2 days. It takes 200,000 training iterations to converge. The model is trained on about 5 millions of hours of speech as described in section 2.3. Its mean WER is equal to 25.3% (computed over 61 languages) and model size in bits are presented in Table 2. This experiment is labeled as *Exp0* in Table 2.

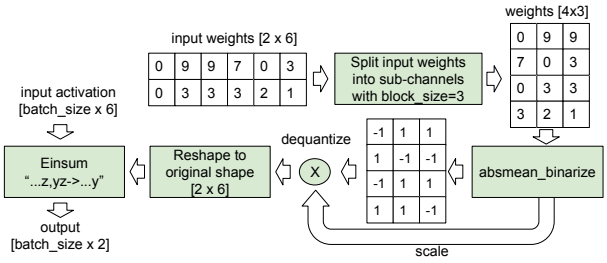

Figure 2: *Combination of sub-channel split with absmean binarization during training.*

4.2. USM RNN-T 1B model with 2bits weights quantization

We run quantization-aware training of *Exp1* and *Exp2* (described in section 3.1) on 256 TPU v5 for 270,000 training iterations (it takes 1.7 days).

Results of USM RNN-T 1B with 2bits quantization approaches *Exp1* and *Exp2* are presented in Table 2. *Exp1* has mean WER 26.7% and it is only 1.4 points worse than its float baseline. In contrast, USM CTC [15] has 3x WER degradation with 2bits weights quantization. We hypothesize that quantization difficulties of the USM CTC model can be due to multi-stage training procedures, described in section 2.1 [15], and differences in data sets. The model size of *Exp1* (in bits and estimated model size reduction) is presented in Table 2. With 2bits weights quantization *Exp1*, we reduce model size by 11.4x times. As expected *Exp2* with sub-channel quantization (block size 64) has a larger quantized model size (due to additional meta data) in comparison to *Exp1*.

4.3. USM RNN-T 1B model with binarized weights

Training binarized models *Exp3*, *Exp4* and *Exp5* (described in section 3.2) takes 3.4 days (446,000 training iterations) on 256 TPU v5 [38]. Method *Exp3* diverged and has 100% WER. But methods *Exp4* and *Exp5* converged and have WER 27.5% and 27.2% accordingly. We hypothesize that method *Exp3* diverged because it is based on *absmax* quantization. Its quantization scale coefficient is defined by max value of abs weights and normalization by max value can be less stable in comparison to normalization by mean abs value. Both *Exp4* and *Exp5* are based on *absmean* quantization (scale coefficient is defined by mean value of abs weights). We observe that a model binarized with *Exp5* can reduce model size by 15.9x times at the cost of increasing mean WER by 1.9% in comparison to float32 baseline.

The WER comparison of float baseline model *Exp0* vs binarized model *Exp5* over 61 languages is shown on Figure 3. We can see that some languages have WER $>$50%, we explain it by data quality of such languages.

Limitations of the binarization approach presented in this paper: there still is an accuracy gap between binarized and float baseline models; model binarization can take at least 2x times longer because it needs more time to converge.

5. Related work

In this section we review the prior literature on ASR model quantization, compare and contrast it to our approach.

Recent studies are focused on the Universal Speech Model (USM, [3, 39, 40, 41, 42, 5]) which can recognize mul-

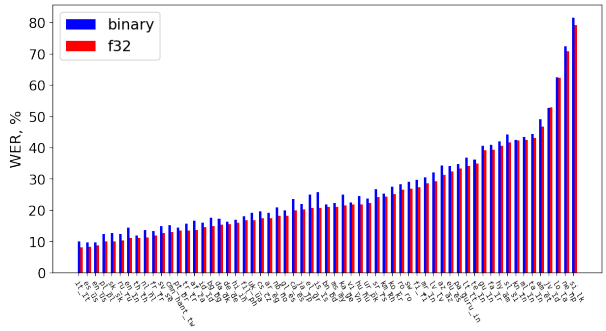

Figure 3: *The WER of f32 (Exp0) and binarized (Exp5) models on 61 languages.*

Table 2: *Results on baseline model and proposed quantization approaches. Mean WER over 61 languages, model size[bits] and estimated model size reduction, defined as: (float model size) / (quantized model size). For each quantized model, we estimate the quantized size with the quantization metadata (scale only for binary quantization, scale and zero point for 2-bit quantization) stored as either f32 or int8. We estimate the 95% confidence interval for WER* ± 0.25.

Experiment	WER[%]	size[# bits billion] (size reduction)	
Exp0 (f32)	25.3%	28.6 (N/A)	
		f32 meta	*int8 meta*
Exp1(2bit)	26.7%	2.5 (11.4x)	2.5 (11.4x)
Exp2(2bit)	27.5%	3.6 (7.9x)	2.8 (10.2x)
Exp3(1bit)	N/A	1.6 (17.9x)	1.6 (17.9x)
Exp4(1bit)	27.5%	1.6 (17.9x)	1.6 (17.9x)
Exp5(1bit)	27.2%	2.2 (13x)	1.8 (15.9x)

tiple languages. We base our work on [5]. These models show state of the art results with large model size (>1B parameters). As a result it increases serving cost. A standard approach of speech model serving cost reduction is model compression based on weights quantization [25, 26, 17, 11, 12, 13, 14, 15]. In [15] authors quantize weights of USM-CTC model with 2bits but accuracy drops by 3x. So they addressed it by combining quantization with sparsity. In our paper we replace USM CTC by USM RNN-T and simplify training step procedures as a result we can quantize USM RNN-T model (*Exp1* in Table 2) with 2bits at a cost of 1.4% WER increase (instead of 3x increase in USM CTC). We explore 2bits quantization approach based on [14]. In addition to 2bits quantization we explore USM RNN-T binarization, which has not been investigated in previous studies. Standard weights binarization is based on *absmean* [19, 23, 22]. As in [22, 23] we use *absmean* binarization but without weights centralization. It simplifies model quantization and does not impact model accuracy (on USM model) in comparison to quantization with weights centralization. We combine *absmean* binarization with sub-channel quantization [14] and show that it is possible to binarize USM RNN-T 1B model and reduce its model size by 15.9x times at a cost of 1.9% WER increase in comparison to baseline float32 model.

6. Conclusion

We simplified USM-CTC 1B model model training by replacing it with USM RNNT 1B model and reducing the number of training step procedures. As a result, the quantized USM RNNT 1B model with 2 bits weights quantization has reduced model size by 11.4x times at cost of only 1.4% (absolute) WER increase vs 3x WER increase of 2bits weights quantization in USM-CTC 1B. We presented a new weights binarization for the USM RNNT 1B model and showed that model size can be reduced by 13x or 15.9x (if double quantization is applied) at a cost of 1.9% WER increase in comparison to baseline float32 model. It makes it attractive for real applications.

7. Acknowledgements

We would like to thank Jian Li, Eugene Weinstein and Pedro Moreno for their support and valuable discussions.

8. References

[1] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An End-to-End Convolutional Neural Acoustic Model," in *Proc. Interspeech*, 2019.

[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.

[3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," ser. ICML, 2023.

[4] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, A. Baevski, Y. Adi, X. Zhang, W.-N. Hsu, A. Conneau, and M. Auli, "Scaling speech technology to 1,000+ languages," 2023.

[5] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang, Z. Meng, K. Hu, A. Rosenberg, R. Prabhavalkar, D. S. Park, P. Haghani, J. Riesa, G. Perng, H. Soltau, T. Strohman, B. Ramabhadran, T. Sainath, P. Moreno, C.-C. Chiu, J. Schalkwyk, F. Beaufays, and Y. Wu, "Google USM: Scaling automatic speech recognition beyond 100 languages," 2023.

[6] 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 *Proc. Interspeech*, 2017.

[7] Y. Shangguan, J. Li, L. Qiao, R. Alvarez, and I. McGraw, "Optimizing speech recognition for the edge," *CoRR*, vol. abs/1909.12408, 2019.

[8] C.-I. J. Lai, Y. Zhang, A. H. Liu, S. Chang, Y.-L. Liao, Y.-S. Chuang, K. Qian, S. Khurana, D. Cox, and J. Glass, "PARP: Prune, adjust and re-prune for self-supervised speech recognition," in *Advances in Neural Information Processing Systems*, 2021.

[9] L. Emili, T. Fraga-Silva, E. Pusateri, M. Nußbaum-Thom, and Y. Oualil, "Neural language model pruning for automatic speech recognition," 2023.

[10] H. Jiang, L. L. Zhang, Y. Li, Y. Wu, S. Cao, T. Cao, Y. Yang, J. Li, M. Yang, and L. Qiu, "Accurate and structured pruning for efficient automatic speech recognition," in *Proc. INTERSPEECH*, 2023.

[11] A. Fasoli, C.-Y. Chen, M. Serrano, X. Sun, N. Wang, S. Venkataramani, G. Saon, X. Cui, B. Kingsbury, W. Zhang, Z. Tüske, and K. Gopalakrishnan, "4-bit quantization of LSTM-based speech recognition models," 2023.

[12] A. Bie, B. Venkitesh, J. Monteiro, M. A. Haidar, and M. Rezagholizadeh, "A simplified fully quantized transformer for end-to-end speech recognition," 2020.

[13] 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 *Proc. Interspeech*, 2022.

[14] O. Rybakov, P. Meadowlark, S. Ding, D. Qiu, J. Li, D. Rim, and Y. He, "2-bit conformer quantization for automatic speech recognition," in *Proc. INTERSPEECH*, 2023.

[15] S. Ding, D. Qiu, D. Rim, Y. He, O. Rybakov, B. Li, R. Prabhavalkar, W. Wang, T. N. Sainath, Z. Han, J. Li, A. Yazdanbakhsh, and S. Agrawal, "USM-Lite: Quantization and sparsity aware fine-tuning for speech recognition with universal speech models," 2024.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

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

[18] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems*, 2016.

[19] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: Imagenet classification using binary convolutional neural networks," 2016.

[20] Y. He, Z. Lou, L. Zhang, J. Liu, W. Wu, H. Zhou, and B. Zhuang, "BiViT: Extremely compressed binary vision transformers," in *ICCV*, October 2023.

[21] A. Bulat and G. Tzimiropoulos, "XNOR-Net++: Improved binary neural networks," in *BMVC*, 2019.

[22] Z. Liu, B. Oguz, A. Pappu, L. Xiao, S. Yih, M. Li, R. Krishnamoorthi, and Y. Mehdad, "BiT: Robustly binarized multi-distilled transformer," in *Advances in Neural Information Processing Systems*, 2022.

[23] H. Wang, S. Ma, L. Dong, S. Huang, H. Wang, L. Ma, F. Yang, R. Wang, Y. Wu, and F. Wei, "BitNet: Scaling 1-bit transformers for large language models," 2023.

[24] S. Ma, H. Wang, L. Ma, L. Wang, W. Wang, S. Huang, L. Dong, R. Wang, J. Xue, and F. Wei, "The era of 1-bit LLMs: All large language models are in 1.58 bits," 2024.

[25] X. Xiang, Y. Qian, and K. Yu, "Binary deep neural networks for speech recognition," in *Proc. Interspeech*, 2017.

[26] Y. Qian and X. Xiang, "Binary neural networks for speech recognition," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, pp. 701 – 715, 2019.

[27] 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.

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

[29] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[30] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *CVPR*, 2020.

[31] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *ICML*, 2022.

[32] A. Graves, "Sequence transduction with recurrent neural networks," in *Representation Learning Workshop ICML*, Nov. 2012.

[33] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *ICASSP*, 2020.

[34] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *ICML*, 2018.

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

[36] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "QLoRA: Efficient finetuning of quantized llms," in *Advances in Neural Information Processing Systems*, 2023.

[37] N. P. Jouppi, G. Kurian, S. Li, P. Ma, R. Nagarajan, L. Nai, N. Patil, S. Subramanian, A. Swing, B. Towles, C. Young, X. Zhou, Z. Zhou, and D. Patterson, "TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings," 2023.

[38] [Online]. Available: https://cloud.google.com/tpu/docs/v5e-inference

[39] W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le, and M. Norouzi, "SpeechStew: Simply mix all available speech recognition data to train one large neural network," 2021.

[40] Y. Zhang, D. S. Park, W. Han, J. Qin, A. Gulati, J. Shor, A. Jansen, Y. Xu, Y. Huang, S. Wang, Z. Zhou, B. Li, M. Ma, W. Chan, J. Yu, Y. Wang, L. Cao, K. C. Sim, B. Ramabhadran, T. N. Sainath, F. Beaufays, Z. Chen, Q. V. Le, C.-C. Chiu, R. Pang, and Y. Wu, "BigSSL: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[41] B. Li, R. Pang, T. N. Sainath, A. Gulati, Y. Zhang, J. Qin, P. Haghani, W. R. Huang, M. Ma, and J. Bai, "Scaling end-to-end models for large-scale multilingual ASR," in *ASRU*, 2021.

[42] X. Li, F. Metze, D. R. Mortensen, A. W. Black, and S. Watanabe, "ASR2K: Speech recognition for around 2000 languages without audio," in *Proc. Interspeech*, 2022.