INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Lossless 4-bit Quantization of Architecture Compressed Conformer ASR Systems on the 300-hr Switchboard Corpus

Zhaoqing Li[1], *Tianzi Wang*[1], *Jiajun Deng*[1], *Junhao Xu*[1], *Shoukang Hu*[2], *and Xunying Liu*[1]

[1]The Chinese University of Hong Kong; [2]Nanyang Technological University

[1]{zqli,twang,jjdeng,jhxu,xyliu}@se.cuhk.edu.hk; [2]shoukang.hu@ntu.edu.sg

Abstract

State-of-the-art end-to-end automatic speech recognition (ASR) systems are becoming increasingly complex and expensive for practical applications. This paper develops a high-performance and low-footprint 4-bit quantized Conformer ASR system. A key feature of the system design is to account for the fine-grained, varying performance sensitivity at different Conformer components to quantization errors. Neural architectural compression and mixed precision quantization approaches were used to auto-configure the optimal substructures and quantization bit-widths within each Conformer sub-module. Experiments conducted on the 300-hr Switchboard data suggest that the obtained auto-configured systems consistently outperform the uniform precision quantized baseline Conformer of comparable bit-widths in terms of word error rate (WER). An overall "lossless" compression ratio of 16.2 times was obtained over the 32-bit full-precision baseline while incurring no statistically significant WER increase.

Index Terms: speech recognition, model quantization, neural architecture search

1. Introduction

State-of-the-art end-to-end (E2E) automatic speech recognition (ASR) systems [1, 2, 3, 4, 5, 6, 7] are becoming increasingly complex [8, 9, 10] and expensive for practical applications. An ultimate goal for resource-intensive deep-learning-based AI applications, including E2E ASR systems, is to derive "lossless" model compression approaches that allow high-performance and low-footprint systems to be constructed[1]. To this end, a powerful solution is to apply low-bit neural network quantization techniques, which have attracted intensive research interest in recent years and have successfully been applied to a range of AI applications, including computer vision [12, 13, 14, 15, 16], language modelling [17, 18] and speech recognition systems [19, 20, 21, 22]. Quantization methods replace deep neural network model parameters of full-precision floating points with quantized low-precision integer values. This allows model sizes and inference cost to be aggressively reduced for efficient implementations in FPGA systems.

Existing quantization studies for E2E ASR systems are predominantly uniform-precision-based. An identical setting of quantization bit width, for example, 4-bit, is applied to all the weight parameters [12, 22, 23]. However, prior research on time delay neural network (TDNN) acoustic models [24] and Transformer or LSTM-RNN language models [18] have shown that the ASR system performance sensitivity to quantization errors varies among different internal components, and can be better accounted for using non-uniform, mixed precision approaches.

This paper aims to develop an ultra-compact 4-bit quantized Conformer-based E2E ASR system with high performance and low footprint on the 300-hr Switchboard corpus. In contrast to conventional uniform-precision-based quantization approaches, the overall system design of this paper accounts for the fine-grained, varying performance sensitivity of different model components to architecture compression and parameter quantization errors. The fundamental objective is to achieve the best model size versus performance trade-off operating points by locally optimizing each Conformer model component layer's architecture size and quantization precision bit width. The overall system development comprises two major stages. In the first stage, the model structural redundancy is minimized using automated neural architectural compression techniques. Penalized differentiable architecture search [25, 26] based neural architecture search (NAS) is applied to auto-configure the hidden layer dimensionality within each Conformer Encoder block. Inspired by the semi-orthogonal low-rank weight matrix factorization technique adopted in TDNN-F [27], the feed-forward sublayers' weight parameter matrices are decomposed into two low-rank matrices. Their respective bottleneck dimensionalities are then auto-configured using NAS. The architecturally compressed full-precision system serves as a streamlined starting point for the following mixed-precision quantization stage. The optimal local bit-width configurations are learned by minimizing the KL divergence measured sublayer-level performance sensitivity to quantization. Quantization-aware training (QAT) [28, 29] is also used to reduce the performance degradation in the quantized parameter estimation stage after the local precision settings are determined.

The main contributions of the paper are summarized below:

1) To the best of our knowledge, this is the first work to apply neural architecture compression, low-rank weight factorization, and mixed precision quantization techniques to obtain ultra-compact Conformer ASR systems. Both these individual techniques and their combination have not been previously studied for Conformer ASR systems. Specifically, such combination was previously studied only with hybrid TDNN ASR systems [24]. In terms of individual techniques, the KL divergence based automatically learned mixed precision quantization approach was only studied for hybrid TDNN systems [24], temporal convolutional network based speech separation [30] and neural language models based on LSTMs and standard (non-Conformer) Transformers [18]. Existing works on low bit precision quantization of Conformer (or other) ASR models largely focused on uniformly or manually configured quantization precision settings [31, 32, 22, 33, 34, 35]. Other prior works on quantization for computer vision tasks [12, 13, 14, 15, 16] lie

[1]In all the experiments of this paper, "Lossless" compression is achieved when no statistically significant (MAPSSWE [11], $\alpha = 0.05$) WER increase is observed after model compression.

10.21437/Interspeech.2023-1665

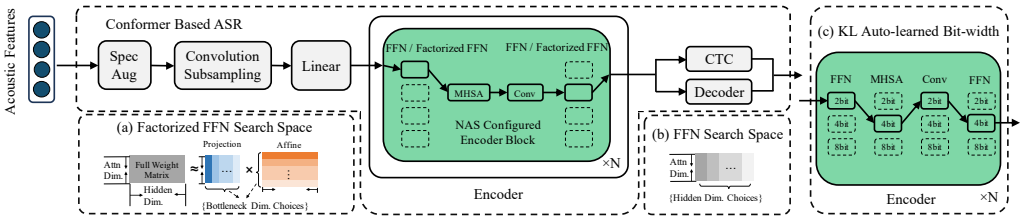

Figure 1: *A paradigm of Conformer based end-to-end ASR system with neural architecture compression applied to the feed-forward modules (FFN) (in the green colored box) to locally select the dimensionality of either (a) the bottleneck subspace for their low-rank factorized weight matrices (left bottom); or (b) that of their hidden layer outputs (right bottom); or both. (c) auto-configuration of local bit-widths for each Encoder sub-module using the KL metric of Section 5.*

in different application domains. Similarly, the automatic neural architecture compression method of this paper was previously only studied for hybrid TDNN ASR systems [24]. In this paper, we aim to achieve the largest compression ratio while incurring no statistically significant WER increase. Such objective substantially differs from that of conventional NAS based auto-machine-learning tasks, which aim to find the best hyper-parameters while taking no account of the trade-off between performance and model size, whether the latter increases or otherwise [36, 37].

2) In contrast to the use of Librispeech data in related prior works [22, 35, 31, 34], where both the individual systems' WERs and their differences are very small (e.g., varying between 2.0% and 3.1% in [22]; or between 2.78% and 4.03% reported in [35] on test-clean data for Conformer systems), we choose to use the benchmark Switchboard 300-hour dataset where a larger disparity in WERs among different test sets can better analyze the efficacy of different architecture compression and quantization methods. This serves to provide valuable insights for practical low-footprint Conformer ASR system development targeting diverse ASR accuracy operating points. To the best of our knowledge, this paper is the first work to develop a "lossless" 4-bit compressed Conformer E2E ASR system without statistically significant WER increase on the benchmark 300-hr Switchboard data.

2. Conformer ASR System Architecture

The convolution-augmented Transformer (Conformer) [9] is an E2E ASR system combining convolution neural networks and transformer modules [38] to capture both local and global dependencies in audio sequences. As a key component, a Conformer Encoder is built by multiple blocks stacked together, with each block composed of the following modules in sequence: a feed-forward module (FFN), a self-attention module (MHSA), a convolution module (Conv), and a second FFN module (macaron-like) in the end. Among all these modules, FFN modules account for a major part of an Encoder in terms of the number of weight parameters. Particularly, it consists of a linear layer followed by a Swish activation [39] and a dropout, and then a second linear layer with a dropout. The first linear layer expands the model dimensionality (e.g., from 256 to 2048), and the second linear layer restores the dimensionality to before. In addition, a post-layer normalization and residual connections are applied to all Encoder blocks. An example Conformer ASR system is shown in Fig. 1. To train a Conformer model, a multitask criterion interpolation [40] between the CTC and attention of the Decoder (i.e., a Transformer) error cost is applied. This is given by

$$\mathcal{L}_{conformer} = (1-\lambda)\mathcal{L}_{att} + \lambda\mathcal{L}_{ctc}, \tag{1}$$

where λ is a constant and empirically set as 0.2 in this paper.

2.1. Semi-orthogonal Low-rank Factorized Conformer

As the linear layers in the FFN module primarily account for the large model size of the Conformer. A simple way to compress the FFN modules is to reduce the hidden feature expansion dimensionality directly. Another option is inspired by the semi-orthogonal low-rank weight matrix factorization technique presented in [26, 27] for TDNNs. In [27], a large weight matrix is factorized to two smaller factors (i.e., projection and affine matrix), with one of the two constrained to be semi-orthogonal. For example, $\mathbf{W} = \mathbf{AB}$, where $\mathbf{W} \in \mathbb{R}^{m\times n}$, $\mathbf{A} \in \mathbb{R}^{m\times k}$, $\mathbf{B} \in \mathbb{R}^{k\times n}$, with $\mathbf{B}$ constrained to be semi-orthogonal, and k defines the bottleneck dimensionality. An illustration of Conformer FFN sublayer factorization is depicted in Fig. 1(a).

3. Neural Architecture Search

It is practically infeasible to manually search for an optimal configuration of the Conformer Encoder hidden layer or bottleneck dimensionality for large systems. In traditional studies, the commonly adopted uniform configuration fails to consider the varying performance sensitivity of different component configurations. To automatically learn a compressed and streamlined architecture configuration for the Conformer system, differentiable neural architecture search (DARTS) [25] is used to optimize the dimensionality of 1) their hidden layer outputs (Fig. 1(b)); or 2) that of the bottleneck projection subspace for their low-rank factorized weight matrices (Fig. 1(a)); or 3) both 1) and 2). In DARTS, each candidate architecture in any layer is assigned a trainable architecture parameter α,

$$\mathbf{h}^l = \sum_{i=1}^{N^l} \lambda_i^l \phi_i^l = \sum_{i=1}^{N^l} \frac{\exp(\alpha_i^l)}{\sum_{j=1}^{N^l}\exp(\alpha_j^l)} \phi_i^l(\mathbf{h}^{l-1}; \boldsymbol{\Theta}_i^l), \tag{2}$$

where ϕ_i^l denotes the i-th candidate function of the l-th layer, α_i^l and $\boldsymbol{\Theta}_i^l$ are corresponding architecture parameter and function weights, respectively. λ_i^l denotes architecture weight obtained by applying a Softmax function to the architecture parameter vector α^l. The output of the l-th layer $\mathbf{h}^l$ can be modeled as a weighted sum of all the candidates in the l-th layer. Then the architecture parameter and function weights can be trained in an alternative manner until convergence. Finally, the optimal architecture candidate at each layer is obtained according to λ.

3.1. Gumbel-softmax DARTS

In traditional DARTS methods, the differences among architecture weights obtained by a flattened Softmax function are relatively small. This is prone to yield confusion over different candidates and thus leads to search errors. To minimize the confusion among different architectures, a Gumbel-Softmax distribution [41, 42] is adopted to sharpen the architecture weights distribution to an approximately one-hot vector as,

$$\lambda_i^l = \frac{\exp(\log(\alpha_i^l + G_i^l)/T)}{\sum_{j=1}^{N^l}\exp(\log(\alpha_j^l + G_j^l)/T)}, \tag{3}$$

Table 1: *Performance of architecture compression of Conformer systems. "FW" and "FW-m" denote the hidden layer dimensionality of the first and the second feed forward layers in each FFN module, respectively. Similarly, "BN" and "BN-m" are corresponding bottleneck dimensionality for low-rank factorized FFNs. The hyper-parameter search scope for "FW" and "FW-m" hidden layer dimensions (Sys. 10-12) is from 1024 to 4096 with a step size of 128, denoted as indices [0, 24]. The hyper-parameter scope for "BN" and "BN-m" in Sys.13 is from 80 to 240 with a step size of 20 denoted as indices [0, 8], with "FW" and "FW-m" fixed as 2048. $\{[i,j]\}:\{x\}$ denotes hyper-parameters from i-th layer to j-th layer inclusive set as "x". ";" is used as a delimiter between hidden and factorized bottleneck layers' dimensionality settings. "S" and "L" denote low-rank weight matrices with and without semi-orthogonal constraints, respectively. † denote no significant (MAPSSWE [11], α=0.05) WER difference observed over the baseline system (sys.1).*

Sys.	Search Method	Search Object	η	Searched Architecture		Hub5'00 swbd	Hub5'00 callhm	Hub5'00 Avg.	#Param. (Million) All / Encoder
1	Baseline	-	-	FW$\{[1:12]\}$ & FW-m$\{[1:12]\}:\{4096\}$		7.2	15.0	11.1	69.81M / 56.90M
2	Manual	-	-	FW$\{[1:12]\}$ & FW-m$\{[1:12]\}:\{1024\}$		7.6	15.5	11.6	31.98M / 19.07M
3				FW$\{[1:12]\}$ & FW-m$\{[1:12]\}:\{2048\}$		7.3	15.3	11.3	44.59M / 31.68M
4				FW$\{[1:12]\}$ & FW-m$\{[1:12]\}:\{5120\}$		7.1	15.0	11.1	82.41M /69.50M
5				FW$\{[1:12]\}$ & FW-m$\{[1:12]\}:\{8192\}$		7.3	14.9	11.1	120.24M / 107.33M
6		-	-	BN & BN-m$\{[1:12]\}:\{160\}$ FW & FW-m$\{[1:12]\}:\{5120\}$ (LLLL)		7.7	16.3	12.1	60.79M / 47.88M
7				BN & BN-m$\{[1:12]\}:\{160\}$ FW & FW-m$\{[1:12]\}:\{5120\}$ (SLLL)		7.2	15.4	11.3	60.79M / 47.88M
8				BN & BN-m$\{[1:12]\}:\{160\}$ FW & FW-m$\{[1:12]\}:\{5120\}$ (SLLS)		**7.2**	**15.1**	**11.2**	60.79M / 47.88M
9				BN & BN-m$\{[1:12]\}:\{160\}$ FW & FW-m$\{[1:12]\}:\{5120\}$ (LSSL)		9.1	18.5	13.8	60.79M / 47.88M
10	PipeGumbel	FW	0	FW	10 14 14 19 2 23 10 15 8 13 17 1	7.2†	14.9†	11.1†	55.10M / 42.19M
				FW-m	12 17 13 10 21 24 20 21 17 18 16 17				
11			0.05	FW	0 0 0 1 0 0 0 1 2 2 13 1	**7.4**†	**15.1**†	**11.3**†	**34.54M / 21.63M**
				FW-m	0 0 1 2 2 1 0 0 5 0 0 8				
12	PipeSoftmax		0.01	FW	0 0 0 0 0 0 0 0 0 0 0 22	7.4†	15.7	11.6	34.81M / 21.90M
				FW-m	0 0 0 0 0 0 0 0 0 0 0 21				
13	PipeGumbel	BN	0.03	BN	5 6 5 7 2 0 2 2 4 1 3 1	7.3†	15.3†	11.3†	35.01M / 22.10M
				BN-m	0 1 1 5 1 0 3 4 6 7 6 1				
14	PipeGumbel	FW& BN	0.03	FW-BN	16;6 8;2 0;8 4;2 0;4 4;4 0;2 0;2 0;4 0;2 0;2 0;2	7.3†	15.3†	11.3†	32.30M / 19.39M
				FW-BN-m	4;6 12;8 8;6 0;2 0;6 0;4 0;2 4;8 0;4 0;4 0;2 0;6				

where $G_i^l = -\log(-\log(U_i^l))$ is the Gumbel variables and U_i^l is a uniform random variable. Eq. (3) approaches a categorical distribution as the temperature T decreases to zero.

3.2. Pipelined and Penalized DARTS

To avoid Gumbel-Softmax DARTS systems prematurely selecting sub-optimal architectures at an early stage, a pipeline style DARTS [43] is used to decouple the architecture parameter training from the model weights training process. Specifically, in pipelined DARTS, a supernet containing model parameters of all the candidates are trained to convergence at the first stage. The architecture parameters are then updated in the second stage on separate held-out data to avoid overfitting, with the other candidate internal parameters kept fixed.

During architecture compression, a penalty loss incorporating the complexity of each candidate choice is jointly optimized with the original Conformer loss function:

$$\mathcal{L} = \mathcal{L}_{conformer} + \eta \sum\nolimits_{i,l} \alpha_i^l C_i^l, \tag{4}$$

where C_i^l is the complexity penalty term expressed as the number of parameters of the i-th candidate in the l-th layer. η is a trade-off hyper-parameter to be tuned.

4. Neural Network Quantization

Neural network quantization can further compress a neural network model by replacing the full precision weight parameters stored in 32-bit floating-point numbers with low-precision integers. An n-bit symmetric linear quantization can be denoted as:

$$\hat{\mathbf{\Theta}} = \Pi_{\mathcal{Q}(\beta,n)}(\mathbf{\Theta}), \tag{5}$$

where $\mathbf{\Theta}$ and $\hat{\mathbf{\Theta}}$ are full-precision and quantized model weights, respectively. $\Pi(\cdot)$ is a projection function that projects each element of $\mathbf{\Theta}$ to its closest value in a quantization table $\mathcal{Q}(\beta, n)$,

$$\mathcal{Q}(\beta, n) = \beta \times \{0, \pm 1, \pm 2, \ldots, \pm(2^{n-1} - 1)\}, \tag{6}$$

where β is a scaling factor to control the dynamic range of the weights and is stored separately as a full-precision floating-point number for the whole $\mathbf{\Theta}$. The most common choice is to set $\beta = \max(|\mathbf{\Theta}|)$, which is also adopted in this paper.

5. Mixed Precision Quantization

Different from uniform-precision quantization that fails to account for varying performance sensitivity of different layers (or components), the main idea of mixed-precision quantization is to keep more sensitive parts at higher precision while squeezing the bit widths more from less sensitive parts without increasing overall model size. The following Kullback–Leibler (KL) divergence metric serves to measure the output distribution difference between the original and quantized models [16, 24],

$$\Omega_i(n) = \frac{1}{T} \sum_{t=1}^{T} D_{KL}(f(x_t; \mathbf{\Theta}) || f(x_t; \hat{\mathbf{\Theta}}_i(n))), \tag{7}$$

where $\Omega_i(n)$ denotes the sensitivity of the i-th layer to n-bit quantization. $\hat{\mathbf{\Theta}}_i(n)$ denotes model parameters in the i-th layer is quantized to n-bit. $f(\cdot) = \text{Softmax}(\phi(\cdot))$ is to normalize the Encoder output into a distribution. x_t is the input vector at time step t, among a total of T frames. The mixed precision settings for a L-layer Conformer system with a target model size S_{target} can be obtained by solving the following:

$$\underset{\{n_i\}_{i=1}^{L}}{\text{argmin}} \ \Omega = \sum_{i=1}^{L} \Omega_i(n_i) \quad s.t. \sum_{i=1}^{L} N_i * n_i \leq S_{target}, \tag{8}$$

where N_i and n_i are the numbers of parameters and quantization bit width of the i-th layer. The optimization problem can be solved by a dynamic programming method to first obtain a Pareto frontier and then select the configuration with the lowest sensitivity that satisfies the model size constraint [16].

6. Experiments

The experiments are conducted on the 300-hr benchmark Switchboard corpus [44] with a single NVIDIA A40 GPU. The developed systems are evaluated on NIST Hub5'00, RT02, and RT03 evaluation sets containing 80, 120, and 144 speakers, and 3.8, 6.4, and 6.2 hours of speech, respectively. The baseline Conformer system is configured using the ESPnet [45] recipe[2]. As the Encoder module accounts for 81.5% of the baseline Conformer model parameters, architectural compression is performed on the Encoder only. In the quantization stage, both the Encoder and Decoder are low-bit quantized.

Experiments on neural architecture compression using complexity-aware NAS are shown in Tab. 1, where several trends can be found: **1)** Before compressing, a series of manually re-configured Conformer systems (Sys. 1-9) are explored

[2]ESPnet: egs/swbd/asr1/run.sh

Table 2: *Performance of the baseline full-precision, uniform-precision, and mixed-precision quantized Conformer systems with local precision settings automatically learned using the KL metric of Section 5 on NIST Hub5'00, RT02, and RT03. † denotes no significant (MAPSSWE [11], α=0.05) WER increase over the 32-bit baseline (ID 0). Speech processing real time factors (RTFs) in the last column.*

Sys.	ID	Quant. Prec.	Config. Method	Bit	Hub5'00		rt02			rt03		Model / Enc. Size(MB)	Ratio All / Enc.	RTFs $\times 10^{-2}$
					swbd	callhm	swbd1	swbd2	swbd3	fsh	swbd			
Sys.1 (Tab. 1)	0	Baseline		32	7.2	15.0	8.6	12.7	15.3	10.4	16.5	266.3 / 217.1	1/1	0.900
Sys.11 (Tab. 1)	1	Quant. Start. Point		32	7.4†	15.1†	9.0†	12.7†	15.4†	10.5†	16.6†	131.8 / 82.5	2.0/2.6	0.472
Quantized Sys.11 (Tab. 1) (Encoder Only)	2	Uniform	Manual Define	2	8.1	16.3	9.5	13.9	17.2	11.1	17.8	54.4 / 5.2	4.9/41.6	0.384
	3			4	7.7	15.5	9.1	13.1†	16.0	10.7	17.2	59.6 / 10.3	4.5/21.1	0.398
	4			8	7.4†	15.1†	8.7†	13.0†	15.7†	10.4†	16.8†	69.9 / 20.6	3.8/10.5	0.402
	5	Mixed		2.6	7.9	16.1	9.2	13.3	16.5	11.0	17.1	55.9 / 6.7	4.8/32.4	0.421
	6		KL	2.5	7.9	15.6	9.2	13.1†	15.9	10.8	17.4	55.7 / 6.5	4.8/33.4	0.412
	7			3	7.9	15.7	9.1	13.3	16.1	10.8	17.4	56.9 / 7.6	4.7/28.5	0.471
	8			4	**7.4**†	**15.4**†	**9.0**†	**12.9**†	**15.6**†	**10.4**†	**16.8**†	59.5 / **10.2**	4.5/**21.2**	0.438
Quant. Sys.11 (Tab.1) (Encoder+Decoder)	9	ID 3 with 4Bit Dec.		4	7.7	15.6	9.2	13.0†	16.0	10.5†	17.0	16.5 / 10.3	16.1/21.2	0.396
	10	ID 8 with 4Bit Dec.		4	**7.5**†	**15.2**†	**9.0**†	**12.9**†	**15.4**†	**10.5**†	**16.2**†	**16.4 / 10.2**	**16.2/21.2**	0.418

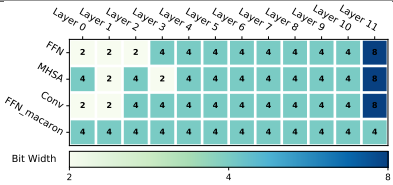

Figure 2: *Layer level bit-widths of an average 4-bit mixed precision quantized system (Tab. 2, ID 8) learned using KL metric.*

first to establish the best hand-crafted architectural hyper-parameters to serve as the baseline starting point. The hidden expansion dimensionality of the FFN feed forward layer is uniformly raised from 1024 to 8192, where the best performance (WER) occurs at 4096 hidden dimensions (Sys. 1). No statistically significant improvement is observed if it is increased to 5120 or 8192 (Sys. 4, 5). Based on these results, Sys. 1 (Tab. 1) is selected as the uncompressed baseline system. **2)** The architecture compressed systems obtained using Gumble Softmax based architecture weights consistently outperform systems obtained using standard, non-Gumble Softmax (Sys. 10 and 11 vs. Sys. 12). **3)** The complexity penalty factor η has an effective control on the model size of the NAS-searched systems (Sys. 10 of 55.10M parameters with $\eta = 0$ vs. Sys. 11 of 34.54M with $\eta = 0.05$.). It allows an optimal system of the best trade-off between model size and performance to be selected (Sys. 11), achieving an overall 2.0 times size reduction without statistically significant WER increase over the baseline (Sys. 1).

4) The effect of further using factorized low-rank weight matrices is then examined on the manually configured system with 5120 hidden nodes and the bottleneck dimensions fixed as 160 (Sys. 6-9). The resulting system performance suggests that the best location to introduce the semi-orthogonal constraint is at the left (projection) and right (affine) low-rank weight matrices of the first and second FNN layers, respectively (Sys. 8 (SLLS), "S" for low-rank matrices with semi-orthogonal constraint, "L" for those without the constraint). **5)** Considering only locally compressing the low-rank subspace dimensionality of the baseline (Sys. 1) at each factored FFN layer, using pipelined Gumble-DARTS with $\eta = 0.03$ (Sys. 13), a compressed model size similar to that of only optimizing the FNN feed forward layer dimensionality (Sys. 11) was obtained. **6)** Furthermore, performing architecture compression on both the hidden layer and factored FFN bottleneck dimensionality[3] produces a "lossless" compression ratio of 2.2 times (2.9 times for Encoder) over the manually crafted baseline (Sys. 1) while incurring no statistically significant WER increase (Sys. 14)

[3]Due to memory constraint in joint double hyper-parameter search, a smaller search scope is used for each hyper-parameter: from 1024 to 4096 with a step size of 512 for "FW" and "FW-m", and {120,160,200,240} for "BN" and "BN-m".

The most compact and best performing model (Sys. 11) in Tab. 1 is selected to serve as a streamlined starting point in the following low-bit quantization stage. During quantization, quantization-aware training (QAT) with the straight-through estimator (STE) [28] is adopted to reduce the performance degradation to quantization errors. Results reported in Tab. 2 (ID 2-10) show several trends[4]. **1)** There is clear precision redundancy in un-quantized Conformer systems. For example, the uniformly configured 8-bit quantization of the Encoder incurs no significant WER increase over that of the 32-bit full-precision baseline (ID 4 vs. ID 0). **2)** Systems with KL-configured mixed precision consistently outperform uniformly and manually configured ones of comparable averaged bit-widths: the 4-bit systems (ID 8 vs. ID 3) and the 2.5-bit systems (ID 6 vs. ID 5). The detailed local sub-layer bit-widths of 4-bit KL-configured mixed-precision (system ID 8) are shown in Fig. 2. **3)** Compared with the full precision, manually configured baseline (Sys. 1, Tab. 1, also as ID 0 in Tab. 2), after also applying 4-bit quantization of the Decoder, an ultra compact system (ID 10) with an overall "lossless" compression ratio of 16.2 times (21.2 times for Encoder alone) was produced using both architecture compression and KL based mixed precision quantization while incurring no statistically significant WER increase. **4)** The model architecture compression ratio is generally found to be linearly correlated with the changes of RTF (ID 0 vs. 1, Table 2 last column). However, the mixed-precision quantization methods are currently not fully supported by GPU CUDA libraries (e.g., 4-bit precision is still practically represented as 8-bit). Hence, the model quantization ratios have not been fully translated into the changes of RTF (ID 1 vs. 3, 4, Tab. 2).

7. Conclusion

This paper presents an aggressively compressed Conformer ASR system on the 300-hr Switchboard data using a novel combination of neural architectural compression and mixed-precision quantization approaches. They are designed to account for the varying performance sensitivity of different model components to structural compression and precision quantization. An overall "lossless" model compression ratio of 16.2 times (21.2 times for Encoder alone) is finally obtained over the full-precision manually configured baseline while incurring no statistically significant WER increase. Future work will focus on improving model compression and inference efficiency.

8. Acknowledgements

This research is supported by Hong Kong RGC GRF grant No. 14200220, 14200021, Innovation Technology Fund grant No. ITS/218/21.

[4]In Tab. 2, model storage size (measured in MB) is reported instead of #parameters (in Tab. 1) as quantization only reduces storage cost.

9. References

[1] W. Chan, N. Jaitly, Q. Le *et al.*, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[2] A. Graves, S. Fernández, F. Gomez *et al.*, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[3] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *NIPS*, 2017.

[4] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in *ICASSP*, 2017.

[5] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP*, 2018.

[6] J. Li, R. Zhao, H. Hu *et al.*, "Improving RNN transducer modeling for end-to-end speech recognition," in *ASRU*, 2019.

[7] Q. Zhang, H. Lu, H. Sak *et al.*, "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *ICASSP*, 2020.

[8] Z. Tüske, G. Saon, K. Audhkhasi *et al.*, "Single headed attention based sequence-to-sequence model for state-of-the-art results on Switchboard," in *INTERSPEECH*, 2020.

[9] A. Gulati, J. Qin, C.-C. Chiu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *INTERSPEECH*, 2020.

[10] C.-C. Chiu, T. N. Sainath, Y. Wu *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *ICASSP*, 2018.

[11] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *ICASSP*, 1989.

[12] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *NIPS*, 2015.

[13] I. Hubara, M. Courbariaux, D. Soudry *et al.*, "Quantized neural networks: Training neural networks with low precision weights and activations," *JMLR*, vol. 18, no. 1, pp. 6869–6898, 2017.

[14] M. Rastegari, V. Ordonez, J. Redmon *et al.*, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *ECCV*, 2016.

[15] Z. Dong, Z. Yao, A. Gholami *et al.*, "Hawq: Hessian aware quantization of neural networks with mixed-precision," in *ICCV*, 2019.

[16] Y. Cai, Z. Yao, Z. Dong *et al.*, "Zeroq: A novel zero shot quantization framework," in *CVPR*, 2020.

[17] X. Liu, D. Cao, and K. Yu, "Binarized lstm language model," in *NAACL-HLT*, 2018.

[18] J. Xu, J. Yu, S. Hu *et al.*, "Mixed precision low-bit quantization of neural network language models for speech recognition," *IEEE/ACM TASLP*, vol. 29, pp. 3679–3693, 2021.

[19] Y. Wang, J. Li, and Y. Gong, "Small-footprint high-performance deep neural network-based speech recognition using split-VQ," in *ICASSP*, 2015.

[20] Y. He, T. N. Sainath, R. Prabhavalkar *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*, 2019.

[21] Y.-m. Qian and X. Xiang, "Binary neural networks for speech recognition," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 5, pp. 701–715, 2019.

[22] S. Ding, P. Meadowlar, Y. He *et al.*, "4-bit conformer with native quantization aware training for speech recognition," in *INTERSPEECH*, 2022.

[23] A. Fasoli, C.-Y. Chen, M. Serrano *et al.*, "4-bit quantization of LSTM-based speech recognition models," in *INTERSPEECH*, 2021.

[24] J. Xu, S. Hu, X. Liu *et al.*, "Towards green ASR: Lossless 4-bit quantization of a hybrid TDNN system on the 300-hr Switchboard corpus," in *INTERSPEECH*, 2022.

[25] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *ICLR*, 2018.

[26] S. Hu, X. Xie, M. Cui *et al.*, "Neural architecture search for LF-MMI trained time delay neural networks," *IEEE/ACM TASLP*, vol. 30, pp. 1093–1107, 2022.

[27] D. Povey, G. Cheng, Y. Wang *et al.*, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *INTERSPEECH*, 2018.

[28] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[29] M. Nagel, M. Fournarakis, R. A. Amjad *et al.*, "A white paper on neural network quantization," *arXiv preprint arXiv:2106.08295*, 2021.

[30] J. Xu, J. Yu, X. Liu *et al.*, "Mixed precision DNN quantization for overlapped speech separation and recognition," in *ICASSP*, 2022.

[31] H. D. Nguyen, A. Alexandridis, and A. Mouchtaris, "Quantization aware training with absolute-cosine regularization for automatic speech recognition," in *INTERSPEECH*, 2020.

[32] A. Fasoli, C.-Y. Chen, M. Serrano *et al.*, "Accelerating inference and language model fusion of recurrent neural network transducers via end-to-end 4-bit quantization," in *INTERSPEECH*, 2022.

[33] M. Someki, Y. Higuchi, T. Hayashi *et al.*, "Espnet-onnx: Bridging a gap between research and production," in *APSIPA ASC*, 2022.

[34] K. Zhen, M. Radfar, H. Nguyen *et al.*, "Sub-8-bit quantization for on-device speech recognition: A regularization-free approach," in *SLT*, 2023.

[35] S. Kim, A. Gholami, Z. Yao *et al.*, "Integer-only zero-shot quantization for efficient speech recognition," in *ICASSP*, 2022.

[36] X. Shi, P. Zhou, W. Chen *et al.*, "Efficient gradient-based neural architecture search for end-to-end ASR," in *ICMI*, 2021.

[37] S. Lu, Y. Wang, P. Yao *et al.*, "Conformer space neural architecture search for multi-task audio separation," in *INTERSPEECH*, 2022.

[38] Z. Dai, Z. Yang, Y. Yang *et al.*, "Transformer-XL: Attentive language models beyond a fixed-length context," in *ACL*, 2019.

[39] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[40] S. Watanabe, T. Hori, S. Kim *et al.*, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[41] C. Maddison, A. Mnih, and Y. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," in *ICLR*, 2017.

[42] S. Xie, H. Zheng, C. Liu *et al.*, "Snas: Stochastic neural architecture search," in *ICLR*, 2018.

[43] Z. Guo, X. Zhang, H. Mu *et al.*, "Single path one-shot neural architecture search with uniform sampling," in *ECCV*, 2020.

[44] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *ICASSP*, 1992.

[45] S. Watanabe, T. Hori, S. Karita *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.