INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

TinyLSTMs: Efficient Neural Speech Enhancement for Hearing Aids

Igor Fedorov[1*^], *Marko Stamenovic*[2*^], *Carl Jensen*[2*], *Li-Chia Yang*[2*], *Ari Mandell*[2], *Yiming Gan*[3], *Matthew Mattina*[1], *Paul N. Whatmough*[1]

[1]Arm ML Research Lab, Boston, MA
[2]Bose Corp., Boston, MA
[3]University of Rochester, Rochester, NY

igor.fedorov@arm.com, marko_stamenovic@bose.com

Abstract

Modern speech enhancement algorithms achieve remarkable noise suppression by means of large recurrent neural networks (RNNs). However, large RNNs limit practical deployment in hearing aid hardware (HW) form-factors, which are battery powered and run on resource-constrained microcontroller units (MCUs) with limited memory capacity and compute capability. In this work, we use model compression techniques to bridge this gap. We define the constraints imposed on the RNN by the HW and describe a method to satisfy them. Although model compression techniques are an active area of research, we are the first to demonstrate their efficacy for RNN speech enhancement, using pruning and integer quantization of weights/activations. We also demonstrate state update skipping, which reduces the computational load. Finally, we conduct a perceptual evaluation of the compressed models to verify audio quality on human raters. Results show a reduction in model size and operations of 11.9× and 2.9×, respectively, over the baseline for compressed models, without a statistical difference in listening preference and only exhibiting a loss of 0.55dB SDR. Our model achieves a computational latency of 2.39ms, well within the 10ms target and 351× better than previous work.
Index Terms: Noise Suppression, Speech Enhancement, Recurrent Neural Networks (RNNs), Pruning, Quantization

1. Introduction

The healthy ear is a complex, non-linear system, capable of operating over a large dynamic range. When the ear is damaged, the auditory system can be augmented with a hearing aid (HA), which performs some of the amplification and filtering functions that the ear is no longer able to do. Speech enhancement (SE) could ease listening difficulty in noisy environments, which ranks among the top concerns of HA users [1, 2, 3].

Recent SE approaches are often embodied by recurrent neural networks (RNNs) [4, 5]. The SE model must achieve low audio latency to ensure listener comfort. Audio latency is defined as the delay between noise arriving at the HA and clean sound being produced for the listener. The amount of latency that can be tolerated varies depending on the HA type and how the user's own voice is processed [6, 7, 8]. Using previous work [6, 7, 8] as a guideline, we target a maximum audio latency of 30ms. For the frame-based approach we employ, with 50% overlap between frames and a causal model, the compute latency constraint for processing each frame is 10ms.

The HA form factor imposes another set of constraints, especially in combination with the frame processing requirement. Due to their small form factor, HAs use a microcontroller unit (MCU) hardware (HW) platform. MCUs enable cheap, low power computation, at the cost of severe memory and computation constraints [9]. The MCU Flash memory restricts the maximum allowable model size (MS), whereas the on-chip SRAM memory upper bounds model working memory (WM), i.e. the memory used to store intermediate results. To achieve efficient computation, the SE model must be quantized to integer data types and we must minimize the number of operations (ops) required per second (ops/s), where an op represents a single addition or multiplication. In this work, we target the STM32F746VE MCU as a typical HW platform, which contains a 216MHz Arm Cortex-M7 [10] with 512KB Flash memory, and 320KB SRAM. We used Mbed OS [11] and CMSIS kernels [12, 13]. Table 1 summarizes the SE model constraints.

A few recent works have considered similar constraints. For example, Wilson et al. [5] use a black box optimizer to search for SE models in a family of causal and non-causal models that include compute-heavy convolutions on the model inputs. Model complexity is not explicitly constrained in the search, with the reported models in the $3.7 - 248$ MB range, violating the MS constraint. Moreover, some of the models include many layers of dilated convolutions on the front-end, which require roughly 4.4 MB of WM, violating the WM constraint.

Other works seek to prune [14] and quantize [15] RNNs, but do not apply their techniques to SE. Although parameters are quantized in [15], activations are not and the resulting computation is not suitable for integer arithmetic. Moreover, it is unclear from [14, 15] if pruning and quantization can be jointly applied to RNNs. In Wu et al. [16], a non-recurrent convolutional SE model is pruned and quantized. Their use of non-uniform quantization, however, requires non-standard HW support [17] to avoid incurring non-trivial performance overhead from decoding each weight after loading it from memory. With a large receptive field, a convolutional model may also need large buffers operating at the audio sample rate. This greatly inflates WM and dramatically tightens the constraints on computational time. Hsu et al. [18] investigated separately quantizing the floating point mantissa and exponent values of recurrent and convolutional SE models [18], but these quantized weights would still need to run in floating point HW and incur overheads for decompression. Although, Kim et al. [19] binarize the weights and activations of a particular RNN topology for SE, the practical benefits of binary networks are unclear since current HW and software deployment tools do not support them. Moreover, the large 64ms frame size and restricted experimental setup, where the training and test data have the same uniform signal-to-noise (SNR), make it difficult to evaluate how the results in [19] would generalize to a low-latency system deployed in an environment with non-stationary SNR.

*equal contribution
^corresponding authors

http://dx.doi.org/10.21437/Interspeech.2020-1864

Constraint	Specification	Rationale
Compute Complexity	≤ 1.55 MOps/inf	Compute budget to achieve $\leq$10ms latency on target MCU
Model Size	≤ 0.5 MB	0.5 MB device Flash memory
Working Memory	≤ 320 KB	320 KB device SRAM
Data Type	Integer	Lowest energy on target MCU
Perceptual Quality	Minimal degradation vs. uncompressed model	Compression should not affect user preference

Table 1: *Model constraints. MOps/inf denotes 10^6 operations per frame inference. Target MCU is STM32F746VE.*

In this work, we present a methodology to generate optimized RNN SE models which meet the requirements in Table 1. Firstly, we demonstrate pruning of SE LSTMs to reduce MS, WM and ops without incurring SE performance degradation. Extending [14], we directly learn the pruning thresholds within the optimization, obviating costly hyperparameter search, resulting in $255\times$ less GPU hours (GPUH) compared to previous work [5]. Secondly, we show for the first time that standard weight and activation quantization techniques extend well to SE RNNs. Moreover, we show that pruning and quantization can be applied to SE RNNs jointly, which is also unique to our work. Finally, we propose a scheme for skipping RNN state updates, reducing the average number of operations.

Our optimized SE models are evaluated using traditional objective metrics, as well as a subjective perceptual evaluation of the audio output. The audio source files we used for this are available online[1]. Our perceptual study is a significant improvement over [4, 5, 16, 18, 20, 21], because (compressed) SE models often exhibit acoustic artifacts not reflected in objective metrics such as SNR. Finally, we profile our models on an MCU to validate that they meet HW constraints, as shown in Table 1.

2. Background

Let lowercase and uppercase symbols denote vectors and matrices, respectively, and let $X = \begin{bmatrix} x^1 & \cdots & x^N \end{bmatrix} \in \mathbb{R}^{M \times N}$.

2.1. Speech Enhancement

Let $x \in \mathbb{R}^N$ denote N samples of a single channel time-domain speech signal. In SE, x is corrupted by noise v and the goal is to extract x from $y = x + v$. In this work, denoising is applied in the time-frequency domain, whereby y is transformed into $Y \in \mathbb{C}^{B_f \times B_t}$ using a short time Fourier transform where B_t is the number of frames that N is decomposed into and B_f is the number of frequency bins. In this work, the denoiser is a mask $M \in \mathbb{R}_+^{B_f \times B_t}$ applied to the spectrogram, such that the approximation of the target is given by $\hat{X} = M \odot |Y| \exp(\measuredangle Y)$, where $\odot$ denotes the Hadamard product and $\measuredangle Y$ is the phase of the noisy input [22]. The mask is a function of Y and learnable parameters θ, i.e. $M = f_\theta(Y)$. Specifically, $f_\theta(\cdot)$ is a neural network whose parameters are learned by minimizing a phase-sensitive spectral approximation loss [20]:

$$L(\theta) = \left\| |X|^{0.3} - |\hat{X}|^{0.3} \right\|_F^2 + 0.113 \left\| X^{0.3} - \hat{X}^{0.3} \right\|_F^2, \quad (1)$$

where frames are power-law compressed with an exponent of 0.3 to reduce the dominance of large values.

2.2. Baseline Model Architecture

Due to the latency requirement, we restrict our attention to causal models [4], whereby $m^t = f_\theta\left(\begin{bmatrix} y^1 & \cdots & y^t \end{bmatrix}\right)$. We use an architecture consisting of a series of recurrent layers followed by fully connected (FC) layers. The recurrent layers serve to model interactions across time. For a recurrent layer, we use long-short term memory (LSTM) cells, which are stateful and have update rules $i^t = \sigma(W_{xi}x^t + h^{t-1}W_{hi} + b_i)$, $r^t = \sigma(W_{xr}x^t + h^{t-1}W_{hr} + b_r)$, $o^t = \sigma(W_{xo}x^t + h^{t-1}W_{ho} + b_o)$, $u^t = \tanh(W_{xu}x^t + h^{t-1}W_{hu} + b_u)$,

$$c^t = r^t \odot c^{t-1} + i^t \odot u^t, \quad h^t = o^t \odot \tanh\left(c^t\right), \quad (2)$$

where σ is the sigmoid function [23]. The baseline architecture consists of 2 unidirectional LSTM layers with 256 units each and 2 FC layers with 128 units each, with batch normalization between the last LSTM and first FC layers. ReLU activation is applied after the first FC layer and sigmoid after the second. In all cases, the spectral input to the network is mapped onto a 128-dimensional mel space [24] and power-law compressed with an exponent of 0.3. The network output, which shares the dimensionality of the input, is inverted using the corresponding transposed mel matrix to produce spectral mask M.

3. Optimizing LSTMs for HA Hardware

This section introduces optimizations for SE models, such as those in Section 2.2, to satisfy the constraints given in Table 1. We begin by delineating the dependence of MS and computational cost on the properties of the model. Then, in Sections 3.2-3.3 we describe our proposed approach.

The MS is the total number of parameters in all layers, multiplied by the datatype of each matrix. The number of operations required per inference also depends on the number of parameters, since (almost) all of the operations performed in our model are matrix-vector multiplications, which require 2 ops (multiply and add) per parameter. Although the operation count is independent of model quantization, the throughput that can be achieved on real HW is much higher with lower precision integer data types. Therefore, to reduce overall latency, we employ two optimizations: 1) pruning to reduce operations, and 2) weight/activation quantization, which reduces MS and enables deployment with low-precision integer arithmetic [25].

3.1. Structural Pruning

Pruning is a well established network optimization [26, 27]. We specifically use structured pruning because it leads to direct benefits in both model size and throughput [28]. This is in contrast to random pruning, which is harder to exploit on real HW, unless the sparsity is very high. We begin by grouping weights in θ into a set Γ, where $w_g \in \Gamma$ denotes the set of weights in a particular group, stacked into a vector. The organization of the groups defines the kind of structures we can learn. For FC layers, we group weights by the neuron they are connected to in the previous layer. For LSTM layers, we group weights by the element of h^t to which they are connected [14]. We assign a binary mask $r_g = \mathbb{1}\left(\|w_g\|_2 - \tau_k \geq 0\right)$ to each group of weights in layer k, where $\mathbb{1}[\cdot]$ denotes the indicator function and $\tau_k \geq 0$ is a learnable threshold. Let $P = \{r_g, 1 \leq g \leq |\Gamma|\}$ be the set

[1] https://github.com/BoseCorp/efficient-neural-speech-enhancement

	SISDR (dB)	BSS SDR (dB)	Params (M)	MS (MB)	WM (KB)	MOps/inf.	Latency (ms/inf.)	Energy (mJ/inf.)	GPUH
Baseline (FP32)	11.99	12.77	0.97	3.70	26.0	1.94	12.52*	6.76*	14
Pruned (FP32)	11.99	12.78	0.52	1.97	18.8	1.04	6.71*	3.62*	72
Pruned (INT8) 1	11.80	12.69	0.61	0.58	5.1	1.22	7.87	4.25	61
Pruned (INT8) 2	11.47	12.22	0.33	0.31	3.7	0.66	4.26	2.30	144
Pruned Skip RNN (INT8)	11.42	12.07	0.46	0.43	4.67	0.37†	2.39†	1.29†	275
Erdogan et al. [20]	-	13.36	0.96	3.65	25.9	1.92	12.39	6.69	-
Wilson et al. [5]	-	14.60	65	247.96	4472.6	130	839*	453	18360

Table 2: *Model performance on CHiME2 development set and STM32F746VE, which was measured to run at 155MOps/s while drawing 0.54W. The symbol * denotes best-case estimates because the underlying models are in floating point and measurements were taken for integer arithmetic. The symbol † denotes average performance, reflecting the stochasticity of the skip RNN model. Blue (red) denotes measurements that pass (violate) one of the constraints in Table 1. Measurements do not include cost of STFT or Mel transforms.*

of pruning masks and $\theta \odot P$ be shorthand for the set of model weights with each weight multiplied by its corresponding binary mask. We then modify the learning objective in (1) to penalize the energy of non-zero weights:

$$\min_{\theta,\{\tau_k,1\leq k\leq K\}} L(\theta \odot P) + \lambda \sum_{g=1}^{|\Gamma|} r_g \left\|w_g\right\|_2 \tag{3}$$

where λ is a hyperparameter controlling the degree of pruning and K is the number of layers. To differentiate the indicator function, we approximate it by the sigmoid function in the backward pass [29]. The pruning approach described above is unique amongst the pruning literature and is particularly suited for our particular task. We adopt the structural grouping of LSTM weights from [14], but we improve upon the manually selected pruning thresholds in [14] by learning them directly. The result is that we do not need to perform a hyperparameter search for $\{\tau_k, 1 \leq k \leq K\}$, which can be prohibitively costly since SE RNNs take roughly 14 GPUH to train and the hyperparameter space grows exponentially with K.

3.2. Quantization

Let $w \in \mathbb{R}$ denote a real-valued (floating point) value and $Q_{\alpha,\beta}(w)$ its quantized value, where the quantization is performed uniformly over the range (α, β) with $2^{\text{bits}} - 1$ levels, i.e. $Q_{\alpha,\beta}(w) = \zeta \text{round}\left((\text{clip}\left(w, \alpha, \beta\right) - \alpha)/\zeta\right) + \alpha$, where $\alpha < \beta$ and $\zeta = (2^{\text{bits}-1})(\beta - \alpha)$. For brevity, we omit α and β subscripts moving forward. We adopt a standard approach of making the model resilient to quantized tensors by performing training-aware quantization [25]. This exposes the model outputs to quantization noise, while still allowing the model to backpropagate on real-valued weights. Concretely, (3) becomes

$$\min_{\substack{\theta,\Omega \\ \{\tau_k,1\leq k\leq K\}}} L_Q\left(Q\left(\theta \odot P\right)\right) + \lambda \sum_{g=1}^{|\Gamma|} r_g \left\|Q\left(w_g\right)\right\|_2 \tag{4}$$

where Ω is the set of quantization parameters for all weights and activations, $Q\left(\theta \odot P\right)$ represents the fact that the masked network weights are quantized, and L_Q denotes that activations are quantized. During backpropagation, the round$(\cdot)$ operation is ignored [25]. We quantize the weights, activations and model input to 8 bits; the mask itself, is quantized to 16 bits.

3.3. Skip RNN Cells

Finally, we evaluated the skip RNN approach [30], which can be considered a form of dynamic temporal pruning. A binary neuron $g^t \in \{0, 1\}$ is introduced that acts as a state update gate on the candidate LSTM states $\tilde{s}$, representing both c^t and h^t from (2):

$$g^t = \text{round}\left(\tilde{g}^t\right), \quad s^t = g^t \tilde{s}^t + (1 - g^t) s^{t-1} \tag{5}$$

SDR [dB]	Input SNR [dB]						
	-6	-3	0	3	6	9	Avg.
Baseline (FP32)	10.01	11.54	13.08	14.23	15.85	17.46	13.70
Pruned (FP32)	10.07	11.59	13.10	14.31	15.89	17.50	13.74
Pruned (INT8) 1	9.82	11.37	12.91	14.20	15.74	17.44	13.58
Pruned (INT8) 2	9.33	10.91	12.46	13.79	15.46	17.16	13.18
Pruned Skip RNN (INT8)	9.13	10.70	12.27	13.54	15.20	16.93	12.96
Erdogan et al. [20]	-	-	-	-	-	-	14.14
Wilson et al. [5]	12.17	13.44	14.70	15.83	17.30	18.78	15.37

Table 3: *Model performance evaluated on CHiME2 test set.*

where $\tilde{g}^t$ is the update probability, updated using

$$\Delta\tilde{g}^t = \sigma\left(W_b c_*^{t-1} + b_b\right) \tag{6}$$

$$\tilde{g}^{t+1} = g^t \Delta\tilde{g}^t + (1 - g^t)(\tilde{g}^t + min(\Delta\tilde{g}^t, 1 - \tilde{g}^t)) \tag{7}$$

where c_*^{t-1} is the state of the final LSTM layer. Whenever a state update is skipped, the state update probability $\tilde{g}^t$ is incremented by $\Delta\tilde{g}^t$ until $\tilde{g}^t$ is high enough that an update occurs, in which case $\tilde{g}^{t+1}$ becomes $\Delta\tilde{g}^t$. Because $\tilde{g}^t$ is fixed when the LSTMs are not updating, (6) only needs to be calculated when the LSTMs are updated.

In practice, this method of skipping updates performs well on training and evaluation metrics, but produced audio artifacts as the mask itself is not updated when the LSTM skips. To remedy this, two exponential moving averages (EMAs) were introduced to smooth the model in time. First, a context vector, $c_x^t = 0.9c_x^{t-1} + 0.1\left(W_c x^t + b_c\right)$, is calculated as an EMA projected from the input spectrogram frame, x_t, and concatenated to the LSTM output. Second, an EMA is applied to the masks, m^t, to calculate a smoothed mask $\tilde{m}_t = 0.15\tilde{m}^{t-1} + 0.85m^t$.

4. Experimental Results

For all experiments, we optimize the objective using stochastic gradient descent in Tensorflow. We use a 32ms frame, 16ms hop size, and 16KHz sample rate for our baseline, pruning, and quantization experiments. For skip RNN experiments, we use frame and hop sizes of 25ms and 6.25ms, respectively. All methods are trained and evaluated with the CHiME2 WSJ0 dataset [31], which contains 7138 training, 2560 development, and 1980 test utterances, respectively. All three subsets include utterances with SNRs in the range -6 to 9dB. Noise data consists of highly non-stationary interference sources recorded in living room settings including vacuum cleaners, television and children. While the dataset is provided in binaural stereo, we pre-process by summing over the channel dimension to obtain a monaural input and target, in contrast to [5], which uses the full binaural input to predict a binaural mask. For final objective evaluations, we use the signal-to-distortion ratio (SDR) [32]. During training, however, we use the simpler scale-invariant signal-to-distortion ratio (SISDR) since it is computationally less expensive and correlates well to SDR [33].

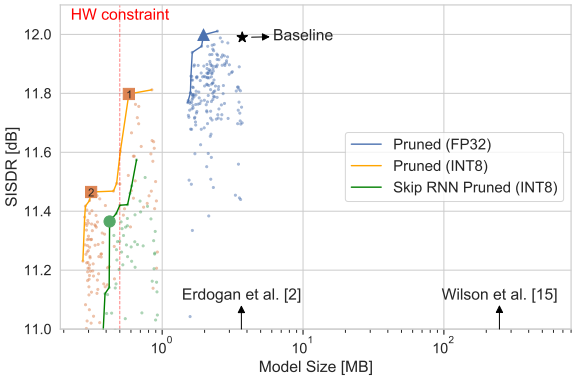

Figure 1: *MS vs. SISDR. Each point represents a model checkpoint and lines represent a Pareto front.*

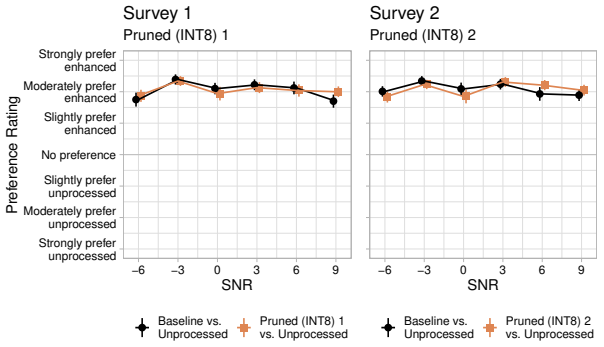

Figure 2: *Preference of perceptual study participants for enhanced audio vs. unprocessed audio for both uncompressed and pruned & quantized models across input SNR's. Left is Pruned (INT8) 1 and right is Pruned (INT8) 2.*

4.1. Baseline Model

We begin by confirming that our baseline SE model is competitive with the state of the art. Our baseline achieves 12.77dB SDR on the CHiME2 development set (Table 2) and 13.70dB SDR on the test set (Table 3), which is comparable to [34, 20].

4.2. Structured Pruning and Quantization

Next, we examine the effect of structural pruning and quantization on the baseline model. In all cases, we set $\lambda = 10^{-9}$. The trade-off between model size and performance is illustrated in Fig. 1, where each point represents a snapshot during the optimization process. We plot the pareto frontier of SISDR values with respect to the MS. Our experiments show that structural pruning can achieve a 47% pruned model with identical performance to the baseline. Moreover, with both pruning and quantization in effect, a 37% pruned model achieves around 0.2dB reduction in SISDR (Pruned (INT8) 1), and a 66% pruned model shows around 0.5dB decay (Pruned (INT8) 2). Table 3 shows SDR evaluations of our models on the CHiME2 test set.

Our optimized models achieve latency suitable for a smaller frame processing time (hop size) in the audio pipeline. However, a small hop size leads to increased inference frequency and energy consumption. So, to address this challenge, we apply the skip RNN architecture on top of our compressed model. Results for "Pruned Skip RNN (INT8)" show 12.07dB SDR on the CHiME2 development set (Table 2) and 12.96dB SDR on the test set (Table 3). Although the skip RNN requires more inferences per second, the skip rate of 63% results in reduced average energy comsumtion per inference compared to Pruned (INT8) 2.

Finally, Table 2 details each of the models described. Although the models in [20, 5] achieve slightly better SISDR/SDR performance, their MS, WM, and MOps/inf severely violate HA HW constraints. In contrast, Pruned (INT8) model 2 and Pruned Skip RNN (INT8) can be deployed on a real HA MCU and deliver significant SE capabilities. In contrast to [20, 5], our models achieve compute latency in the 2.39-6.71ms range, satisfying the 10ms requirement. Moreover, the proposed models consume significantly less energy per inference than [20, 5], leading to better HA battery life.

4.3. Perceptual Evaluation

Human perception of audio quality is highly subjective and does not always correlate with objective metrics. Therefore, to understand real-world performance, we conducted perceptual studies to get subjective feedback on the quality of the optimized model compared to our baseline. We conducted surveys for both Pruned (INT8) models (Table 2), each consisting of disjoint sets of 50 participants. Two samples were randomly chosen from each of the 6 SNR levels of the CHiME2 evaluation set, for a total of 12 sample utterances. Each participant was randomly presented with paired comparisons of the original and processed utterance for both the baseline and pruned & quantized models, resulting in 24 paired comparisons per participant. Given the prompt *Considering both clarity and quality of speech, which recording do you prefer?*, the participants rated comparative preference on a 7-point Likert scale [35] ranging from "Strongly prefer unprocessed' to "Strongly prefer enhanced", with "No preference" as a midpoint.

The results in Fig. 2, show that participants expressed a "moderate preference" on average for the enhanced audio. We note this compares favorably with industry-standard approaches to improving HA speech-in-noise performance, where participants in a similar study expressed a "slight preference" for directionally processed audio compared to unprocessed [36]. We compare preference for the uncompressed (baseline) and the compressed (pruned and quantized) models to the original unprocessed utterances using a Wilcoxon signed rank test [37] and find no statistical difference between ratings across SNRs (Survey 1: $Z = 0.09$, $p = 0.92$; Survey 2: $Z = 0.19$, $p = 0.85$), indicating that participants prefer the enhanced audio, independently of whether it was produced by the baseline or optimized model.

5. Conclusions

Neural speech enhancement is a key technology for future HA products. However, the latency and power consumption constraints for tiny battery-powered HW is extremely hard to meet due to the large NNs required to achieve satisfactory audio performance. In this work, we applied structural pruning and integer quantization of inputs, weights, and activations to reduce model size by 11.9×, compared to the baseline. We also applied skip RNN techniques to further reduce the ops per inference by 1.78× compared to our smallest compressed model. Our optimized models demonstrate negligible degradation in objective (SISDR) metrics and no statistical difference in subjective human perceptual evaluation. While our baseline model has a computational latency of 12.52ms on our target HW platform, the optimized implementation achieves 4.26ms, which is more than sufficient to meet the 10ms compute latency target.

6. References

[1] S. Kochkin, "MarkeTrak V: 'Why my hearing aids are in the drawer' the consumers' perspective," *The Hearing Journal*, vol. 53, no. 2, pp. 34–36, 2000.

[2] H. B. Abrams and J. Kihm, "An introduction to marketrak ix: A new baseline for the hearing aid market," *Hearing Review*, vol. 22, no. 6, p. 16, 2015.

[3] (2020) Hearing aids, the ultimate guide: Types, features, prices, reviews, and more. [Online]. Available: https://www.hearingtracker.com/hearing-aids

[4] D. Takeuchi, K. Yatabe, Y. Koizumi, Y. Oikawa, and N. Harada, "Real-time speech enhancement using equilibriated rnn," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 851–855.

[5] K. Wilson, M. Chinen, J. Thorpe, B. Patton, J. Hershey, R. A. Saurous, J. Skoglund, and R. F. Lyon, "Exploring tradeoffs in models for low-latency speech enhancement," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 366–370.

[6] M. A. Stone and B. C. J. Moore, "Tolerable hearing aid delays. i. estimation of limits imposed by the auditory path alone using simulated hearing losses," *Ear and Hearing*, vol. 20, no. 3, pp. 182–192, 1999.

[7] ——, "Tolerable hearing aid delays. ii. estimation of limits imposed during speech production, ear and hearing," *Ear and Hearing*, vol. 23, no. 4, pp. 325–338, 2002.

[8] ——, "Tolerable hearing aid delays. iii. effects on speech production and perception of across-frequency variation in delay," *Ear and Hearing*, vol. 24, no. 2, pp. 175–183, 2003.

[9] I. Fedorov, R. P. Adams, M. Mattina, and P. N. Whatmough, "SpArSe: Sparse architecture search for CNNs on resource-constrained microcontrollers," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 4978–4990.

[10] Arm Cortex-M7 Embedded Processor. [Online]. Available: https://developer.arm.com/ip-products/processors/cortex-m/cortex-m7

[11] Arm Mbed. [Online]. Available: https://os.mbed.com/

[12] Arm CMSIS Library. [Online]. Available: https://github.com/ARM-software/CMSIS

[13] L. Lai, N. Suda, and V. Chandra, "CMSIS-NN: efficient neural network kernels for arm cortex-m cpus," *CoRR*, vol. abs/1801.06601, 2018.

[14] W. Wen, Y. He, S. Rajbhandari, M. Zhang, W. Wang, F. Liu, B. Hu, Y. Chen, and H. Li, "Learning intrinsic sparse structures within long short-term memory," in *International Conference on Learning Representations*, 2018.

[15] L. Hou, J. Zhu, J. Kwok, F. Gao, T. Qin, and T.-y. Liu, "Normalization helps training of quantized lstm," in *Advances in Neural Information Processing Systems*, 2019, pp. 7344–7354.

[16] J. Wu, C. Yu, S. Fu, C. Liu, S. Chien, and Y. Tsao, "Increasing compactness of deep learning based speech enhancement models with parameter pruning and quantization techniques," *IEEE Signal Processing Letters*, vol. 26, no. 12, pp. 1887–1891, 2019.

[17] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: efficient inference engine on compressed deep neural network," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243–254, 2016.

[18] Y.-T. Hsu, Y.-C. Lin, S.-W. Fu, Y. Tsao, and T.-W. Kuo, "A study on speech enhancement using exponent-only floating point quantized neural network (eofp-qnn)," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 566–573.

[19] S. Kim, M. Maity, and M. Kim, "Incremental binarization on recurrent neural networks for single-channel source separation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 376–380.

[20] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 708–712.

[21] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 91–99.

[22] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *Journal of the Acoustical Society of America*, vol. 8, pp. 185 – 190, 1937.

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

[26] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, 1990, pp. 598–605.

[27] M. C. Mozer and P. Smolensky, "Skeletonization: A technique for trimming the fat from a network via relevance assessment," in *Advances in neural information processing systems*, 1989, pp. 107–115.

[28] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in neural information processing systems*, 2016, pp. 2074–2082.

[29] D. Stamoulis, R. Ding, D. Wang, D. Lymberopoulos, N. B. Priyantha, J. Liu, and D. Marculescu, "Single-path mobile automl: Efficient convnet design and nas hyperparameter optimization," *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–1, 2020.

[30] V. Campos, B. Jou, X. Gir'o i Nieto, J. Torres, and S. Chang, "Skip RNN: learning to skip state updates in recurrent neural networks," in *International Conference on Learning Representations*, 2018.

[31] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second chimespeech separation and recognition challenge: Datasets, tasks and baselines," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 126–130.

[32] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on audio, speech, and language processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[33] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr–half-baked or well done?" in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 626–630.

[34] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2014, pp. 577–581.

[35] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, vol. 140, pp. 1–55, 1932.

[36] J. M. Vaisberg, A. Sabin, and S. Banerjee, "Speech-in-noise benefits using Bose directional technology," in *American Academy of Audiology Conference*, 2020.

[37] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80–83, 1945.