LIMITING NUMERICAL PRECISION OF NEURAL NETWORKS TO ACHIEVE REAL-TIME VOICE ACTIVITY DETECTION

Jong Hwan Ko[*], *Josh Fromm*[†], *Matthai Philipose*[‡], *Ivan Tashev*[‡], *and Shuayb Zarar*[‡]

[*] School of Electrical and Computer Engineering, Georgia Institute of Technology, GA 30332, USA
[†] Department of Electrical Engineering, University of Washington, WA 98195, USA
[‡] Microsoft Research, Redmond, WA 98052, USA
[*] jonghwan.ko@gatech.edu, [†]jwfromm@uw.edu, [‡]{matthaip, ivantash, shuayb}@microsoft.com

ABSTRACT

Fast and robust voice-activity detection is critical to efficiently process speech. While deep-learning based methods to detect voice have shown competitive accuracies, the best models in the literature incur over a 100 ms latency on commodity processors. Such delays are unacceptable for real-time speech processing. In this paper, we study the impact of lowering the representation precision of the neural-network weights and neurons on both the accuracy and delay of voice-activity detection. Based on a design-space exploration, we not only determine the optimal scaling strategy but also adjust the network structure to accommodate the new quantization levels. Through experiments conducted with real user data, we demonstrate that optimized deep neural networks with lower bit precisions outperform the state-of-the-art WebRTC voice-activity detector with 87x lower delay and 6.8% lower error rate.

Index Terms— Voice-activity detection, VAD, Precision scaling, Neural networks

1. INTRODUCTION

Voice activity detection (VAD) is a process of identifying the presence of human speech in an audio sample that contains a mixture of speech and noise. Thanks to its ability of filtering out non-speech segments, VAD has become a critical front-end component of many speech-processing systems such as automatic speech recognition and speaker identification [1-3].

Conventional VAD algorithms are generally based on statistical signal processing that make strong assumptions on the distributions of speech and background noise. One of the commonly used conventional approaches is ITU-T Recommendation G.729-Annex B [4]. This method was improved by Sohn *et al*. with an addition of speech presence probability [5]. A hangover scheme with a simple hidden Markov model (HMM) was added in [6], and further optimized for better performance as described in [7]. Recently, another VAD algorithm based on the Gaussian mixture model was developed in line with the WebRTC project, including an open-source implementation that targets real-time performance [8]. This algorithm has found wide adoption and has recently become one of the gold-standards for delay-sensitive scenarios like web-based interaction. Despite these algorithmic advances, performance of conventional algorithms has not yet reached levels that are routinely expected by modern applications (< 5% error rate). Their performance limitation is typically attributed to two factors: (1) difficulty of finding an analytical form of speech-presence probability [9] and (2) not having enough parameters that capture global signal distributions [3]. Therefore, these conventional approaches can be either approximate or computationally expensive [9].

Emerging deep-neural networks (DNNs) implicitly model data distributions with high-dimensionality. Besides, they allow us to fuse multiple features and separate speech from fast-varying non-stationary noises [9][10]. Thus, DNNs provide a new opportunity to improve the performance of voice-activity detection [11]. Indeed, recent work has demonstrated its benefits via simple fully-connected networks, recurrent networks, and deep-belief networks [9], [12-14]. However, in most prior work, the improvements were obtained in cases where the training and test sets had the same types of noise. Thus, the performance of existing neural-network models has suffered significantly when applied to unseen test scenarios [3]. Another limitation of

	WebRTC [8]	**DNN**		
		Baseline (1729-512-512-512-257)		**Optimized** (256-32-257)
		W32/N32 [9]	**W1/N2 [This work]**	
kOPs/frame	-	3073	144 **(21x ↓)**	1.5 **(2048x ↓)**
Memory (MB)	-	6.0	0.19 **(32x ↓)**	0.13 **(46x ↓)**
Processing delay /sample (ms)	17	138 **(8.2x ↑)**	4.7 **(3.6x ↓)**	0.2 **(87x ↓)**
VAD error rate (%)	20.88	8.20 **(12.68% ↓)**	11.34 **(9.54% ↓)**	14.10 **(6.8% ↓)**

Table I. Comparison of the computation/memory demand and performance of conventional WebRTC and DNN-based VADs. DNN models include baseline/optimized structures and two different precisions (*Wi/Nj* indicates *i* bits for weights and *j* bits for neurons). The reference for the kOPs/frame and memory comparison is W32/N32 DNN, and the reference for the processing delay and VAD error rate comparison is the WebRTC.

DNNs is their computational complexity and memory demand, which increase significantly depending on the depth and breadth of the networks. For instance, on an Intel CPU, even a simple 3-layer DNN incurs a processing delay of 138 ms per frame [see Table I]. This is due to the 3073 kOPs of computation and 6 MB of memory required to evaluate every frame of audio data. Such overheads are unacceptable in real-time applications. In this paper, we aim to address both of these issues by optimizing the neural network architectures.

To lower the computation and memory demands of DNNs, a number of optimization methods have been proposed [15][16]. One of the recently proposed methods is a precision-scaling technique that represents the weights and/or neurons of the network with reduced number of bits [17]. While recent studies have effectively applied binarized (1-bit) networks in image classification tasks [18][19], to the best of our knowledge, no work has been done to analyze the effect of various bit-width pairs of weights and neurons on the processing delay and the detection accuracy of VAD.

In this paper, we investigate the design of efficient DNNs for VAD by scaling the precision of data representation within the network. To minimize bit-quantization error, we use a bit-allocation scheme based on the global distribution of the values. We determine the optimal pair of weight/neuron bits by exploring the impact of bit widths on both the processing performance and delay. We further reduce the processing delay by optimizing the network structure. We compare the detection accuracy of the proposed DNN model with conventional approaches using the test set with unseen noise scenarios. Our results show that the DNN with 1-bit weights and 2-bit neurons reduces the processing delay by 30x with 3.12% increase in accuracy, compared to the baseline 32-bit DNN. By shrinking the network, it outperforms the state-of-the-art WebRTC VAD with 87x lower delay and 6.8% lower error rate.

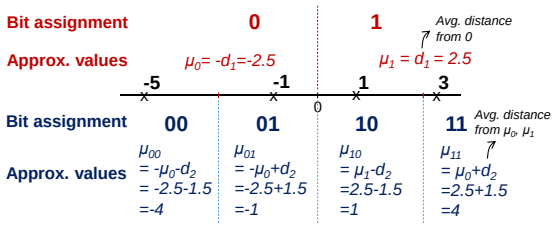

Fig. 1. An example bit assignment using the proposed method. Four different values (-5, -1, 1, 3) are represented by 2-bit precision with the approximate values of (-4, -1, 1, 4).

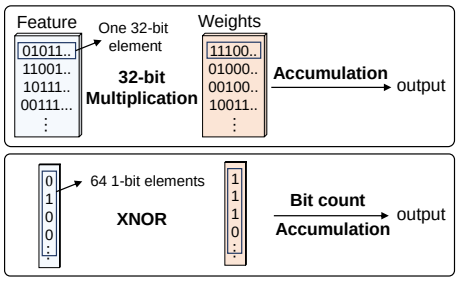

Fig. 2. Illustration of output feature computation with 32-bit (top) and 1-bit (bottom) weights and neurons.

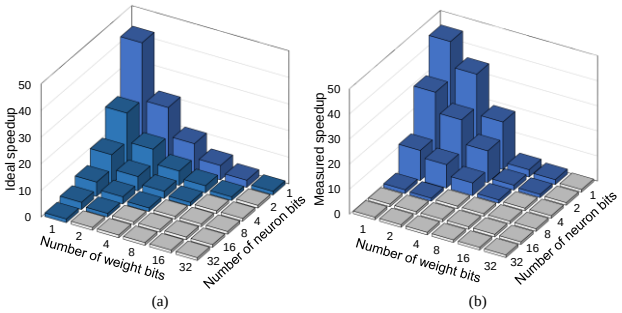

Fig. 3. Speedup due to reduced bit precision of neurons and weights. (a) Ideal and (b) measured speedup. Blue bars indicate speedup>1 and gray bars indicate no speedup.

2. PRECISION SCALING OF NEURAL NETWORKS

One of the most commonly used precision-scaling method is the rounding scheme with round-to-nearest or stochastic rounding mode [20]. However, rounding can result in large quantization error as it does not consider global distribution of the values. In this work, we use a precision scaling method based on residual error mean binarization [21], in which each bit assignment is associated with a corresponding approximate value that is determined by the distribution of the original values. Fig. 1 illustrates an example of 2-bit assignment of 4 values. First representation bit is assigned based on the sign – positive values are assigned bit 1 and negative values are assigned bit 0. Then the approximate value for each bit assignment is computed by adding/subtracting the average distance from the reference value (0 in the first bit assignment). For next bit assignment, each approximate value becomes the reference of each section of the bit. This process allocates the same number of values in each bit assignment bin to minimize the quantization error.

We estimate the ideal inference speedup due to the reduced bit precision by counting the number of operations in each bit-precision case [see Fig. 2]. In the regular 32-bit network, we need two operations (32-bit multiplication and accumulation) per one pair of input feature and weight elements to compute the output feature. When the network has 1-bit neurons and weights, multiplication can be replaced with XNOR and bit count operations, which can be performed in sets of 64 operations per CPU cycle. In this case, we need three operations per 64 elements, which translates to a 42.7x speedup. When the network has 2 or more bit neurons and weights, we need to perform the operation for all the combinations of the bits. Therefore, the ideal speedup is computed as

$$Speedup = \max\left(1, \frac{128}{3 \times Wi \times Nj}\right)$$

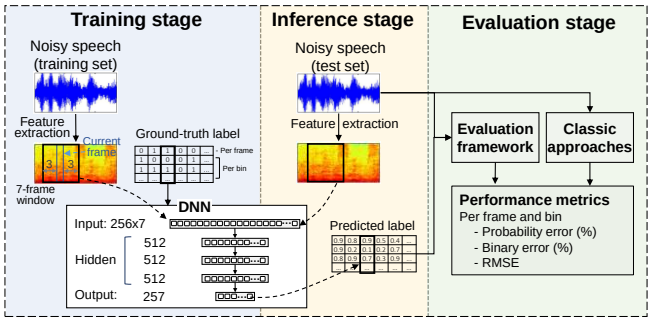

Fig. 4. Experimental framework that we use in this paper.

where *Wi* and *Nj* denote *i*-bit and *j*-bit representations used for the weights and the neurons, respectively. Fig. 3(a) shows that the ideal speedup decreases as we reduce weight/neuron bit width. When the product of the two bit-precision values is larger than 42.7, there is no advantage from bit truncation since XNOR and bit-count operations will take more computation than regular multiplication. We have implemented our precision scaling methodology within the CNTK framework [22], and measured the actual inference speedup that was attained on an Intel processor [see Fig. 3(b)]. The measured speedup is similar to or even higher than the ideal values because of the benefits of loading the low-precision weights, as the bottleneck of the CNTK matrix multiplication is memory access. The figure also indicates that reducing weight bits leads to higher speedup than reducing neuron bits since the weights can be pre-quantized, making their memory loads very efficient.

3. EXPERIMENTAL FRAMEWORK

3.1. Dataset

We created 750/150/150 files of training/validation/test datasets by convolving clean speech with room impulse responses and adding pre-recorded noise at different signal-to-noise ratios (SNRs) ranging between 0-30 dB and distances from the microphone ranging between 1-3m. Each clean speech file included 10 sample utterances that were collected from voice queries to the Microsoft Windows Cortana Voice Assistant. Further, our noise files contained 25 types of recordings in the real world from a single-channel microphone array. Using noise files with different noise scenarios, we also created 150 files of the test set with unseen noise.

3.2. Experimental Framework

As Fig. 4 shows, the experiments are performed through training, inference, and evaluation stages. We utilized noisy speech spectrogram windows of 16 ms and 50% overlap with a Hann weighting, along with the corresponding ground-truth labels for DNN training and inference. For the baseline DNN, we utilized the model presented in [9]. The input feature to the DNN was prepared by flattening symmetric 7-frame windows of the spectrogram. The network had three hidden layers with 512 neurons each, and an output layer of 257 neurons; one for the speech probability for the entire frame and the other 256 for frequency bins. At the end of each layer, we applied the tanh non-linearity function.

Model		Classic	WebRTC	DNN	
				W32/N32	W1/N1
Regular testset	RMSE	0.411	0.408	**0.268**	0.389
	Probability (%)	24.24	-	**5.96**	21.63
	Binary (%)	24.90	20.46	**5.55**	14.95
Testset w/ unseen noise	RMSE	0.343	0.389	**0.228**	0.312
	Probability (%)	17.89	-	**15.32**	24.51
	Binary (%)	18.08	20.88	**8.20**	17.76

Table II. Comparison of voice detection error rates with different approaches and test sets. Probability error rates of WebRTC are omitted since it only provides the binary-detection result.

The network was trained to minimize the squared error between the ground-truth and predicted labels. Each training involved 100 epochs with a batch size of 400. We trained the network with the reduced bit precision from scratch, instead of re-training the network after bit quantization. During inference, we supplied the noisy spectrogram from the test dataset to the trained network to generate the predicted labels.

The predicted labels were compared with the ground-truth labels to compute performance metrics including probability/binary detection error and mean-square error. We define detection error as the average difference between the ground-truth labels and probability/binary decision labels for each frame or frequency bin. Further, we determined the binary decision by comparing the probability with the fixed threshold 0.5. For performance comparison with conventional approaches, we also obtained the performance metrics of the classic VAD in [7] and WebRTC VAD.

4. EXPERIMENTAL RESULTS

Table II compares the per-frame detection accuracy for the regular test set and the test set with unseen noise. With the regular test set, the baseline 32-bit DNN provides much higher detection accuracy than conventional approaches. It is important to note that even the DNN with 1-bit weights and neurons achieved lower detection error than the conventional methods.

To illustrate the performance advantage, we show the binary detection output from each method for a sample file that has similar error rates to the average error rates [Fig. 5]. The DNN approach shows very similar detection output as the ground truth, even with 1-bit weights and neurons. However, the classic methods are prone to false positives, leading to a higher detection error than the DNN models.

Table II indicates that the detection performance of the conventional methods is not significantly affected by the dependency of noise types in the training and test set. However, the DNN gives higher error rates with the unseen test set since the network is dealing with the noise types different from the ones used for training. Nevertheless, the binary detection error of the 1-bit DNN is lower than the classic approaches even with the unseen test set. As we target

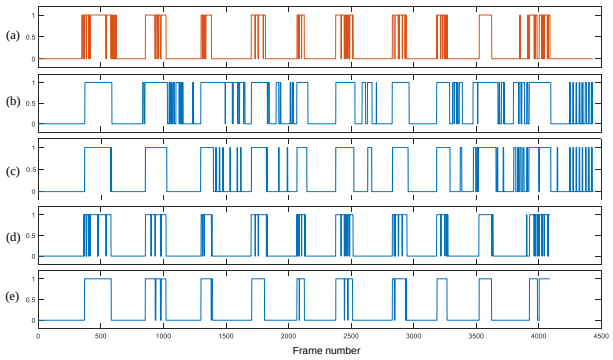

Fig. 5. Illustration of voice detection output from different VAD approaches for a sample noisy speech file. (a) Ground-truth label, (b) classic VAD, (c) WebRTC VAD, (d) DNN with 32-bit weights/neurons, and (e) DNN with 1-bit weights/neurons.

for the practical solution that makes a detection on each frame under the various noise types, we focus on the frame-level binary detection error on the unseen test set for the rest of the analysis.

Fig. 6(a) shows detection error of the DNN model with different weight/neuron bit precision pairs. As expected, the detection error increases as lower bit precision is used. One important observation from this result is that the accuracy is more sensitive to neuron bit reduction than weight bit reduction. Thus, to choose the optimal pair of weight/neuron bit precision we need to consider both detection accuracy and processing delay. Therefore, we introduce the new metric computed by multiplying speedup and VAD error, with both of them normalized to lie in the range [0,1]. As shown in Fig. 6(b), the optimal bit-precision pair is determined as 1-bit weights and 2-bit neurons (W1/N2).

We measured the average processing delay per file of the different approaches based on their Python implementation and an Intel processor. As our implementation of the classic VAD was based on MATLAB, we focused on the WebRTC VAD to compare the processing delays. The baseline 32-bit DNN required 138 ms per file, which was much higher delay than the WebRTC VAD (17 ms). As we scaled the precision to W1/N2, which we chose as the optimal precision pair in the last section, the processing delay reduced by 30x (4.7 ms), which was 3.6x lower than the WebRTC VAD.

We reduced the processing delay further by optimizing the network structure such as the number of layers, number of neurons in each layer, and the input window size. As shown in Fig. 7, the network size reduction leads to a decrease in processing delay as well as VAD accuracy. One interesting conclusion that we can make at this point is that wide and shallow DNNs provide better accuracy than narrow and deep DNNs at the same delay (e.g. three 128-neuron vs. one 512-neuron). By further reducing the network into one 32-neuron layer and single-frame window, we observe that the W1/N2 DNN outperforms the WebRTC VAD with 87X lower delay and 6.8% lower error rate.

Lower precision of the weights not only reduces the computational demand, but also reduces the size of the

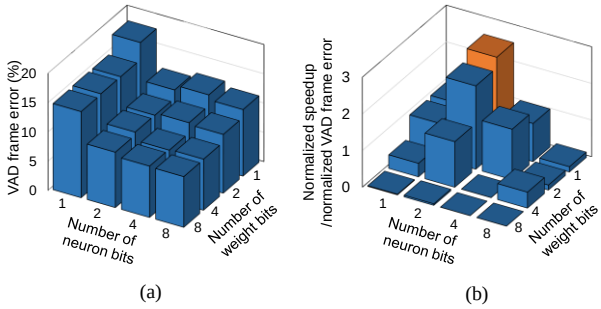

Fig. 6. VAD performance of DNN with different pairs of weight/neuron bit precision. (a) Frame-level binary detection error and (b) normalized speedup/normalized VAD frame error. A red bar indicates the optimal pair of bit precision (W1/N2).

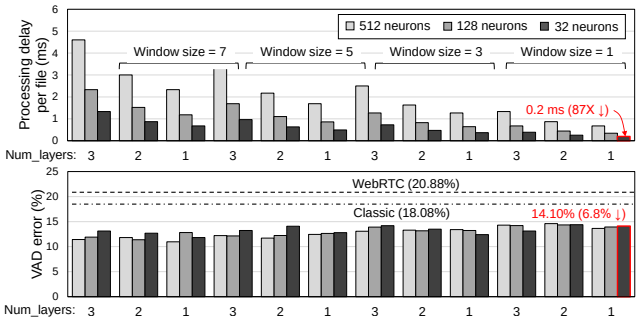

Fig. 7. Optimization of the DNN model. Processing delay per file (top) and frame-level binary detection error (bottom). A red bar indicates the smallest model in the experiments, which shows 87X lower delay and 6.8% lower VAD error than WebRTC model.

weights, which potentially decreases the effective memory access latency and energy. As the weights of the baseline 32-bit DNN (6MB) cannot typically be fit into an on-chip cache of usual mobile devices, we recommend that they be stored in an off-chip memory such as DRAM, where the system throughput and energy is dominated by the weight access. Since the entire set of weights for the W1/N2 DNN (128 KB) can be stored in the on-chip cache, a significant reduction in energy and latency is achieved per our expectation.

5. CONCLUSIONS

In this paper, we presented a methodology to efficiently scale the precision of neural networks for a voice-activity detection task. Through a careful design-space exploration, we demonstrated that a DNN model with optimal bit-precision values reduces the processing delay by 30x with only a slight increase in the error rate. By further optimizing the network structure, it outperforms a state-of-the-art VAD from the literature with 87x lower delay and 6.8% lower error rate. The results show the promising potential of precision scaling for optimization of DNNs for a classification task. As part of future work, we intend to further explore the effect of scaling the neural-network bit precision for other classification tasks such as source separation and microphone beam forming as well as estimation tasks such as acoustic echo cancellation.

6. REFERENCES

[1] J. Ramírez, J. C. Segura, J. M. Górriz, and L. García, "Improved Voice Activity Detection Using Contextual Multiple Hypothesis Testing for Robust Speech Recognition," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 15, no. 8, 2007.

[2] M. W. Mak and H. B. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Comput. Speech Lang.*, vol. 28, no. 1, pp. 295–313, 2014.

[3] X. Zhang and D. Wang, "Boosting Contextual Information for Deep Neural Network Based Voice Activity Detection," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 2, pp. 252–264, 2016.

[4] "Recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," 1997.

[5] J. Sohn and W. Sung, "A voice activity detector employing soft decision based noise spectrum adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998, pp. 365–368.

[6] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, 1999.

[7] I. Tashev, A. Lovitt, and A. Acero, "Unified Framework for Single Channel Speech Enhancement," in *Proceedings of the 2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2009, pp. 883–888.

[8] "WebRTC," 2017. [Online]. Available: https://webrtc.org/.

[9] I. Tashev and S. Mirsamadi, "DNN-based Causal Voice Activity Detector," in *Information Theory and Applications Workshop*, 2016.

[10] X.-L. Zhang and J. Wu, "Denoising Deep Neural Networks Based Voice Activity Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[11] M. W. Hoffman, Z. Li, and D. Khataniar, "GSC-Based Spatial Voice Activity Detection for Enhanced Speech Coding in the Presence of Competing Speech," *IEEE Trans. Speech Audio Process.*, vol. 4419, no. 1, pp. 1–9, 2001.

[12] F. Eyben, F. Weninger, and S. Squartini, "Real-Life Voice Activity Detection with LSTM Recurrent Neural Networks And An Application To Hollywood Movies," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 483–487.

[13] T. Hughes and K. Mierle, "Recurrent Neural Networks for Voice Activity Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7378–7382.

[14] X. Zhang and J. Wu, "Deep Belief Networks Based Voice Activity Detection," *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 21, no. 4, pp. 697–710, 2013.

[15] P. Wang and J. Cheng, "Accelerating Convolutional Neural Networks for Mobile Applications," in *ACM Multimedia Conference*, 2016, pp. 541–545.

[16] L. Song, Y. Wang, Y. Han, X. Zhao, B. Liu, and X. Li, "C-brain: a deep learning accelerator that tames the diversity of CNNs through adaptive data-level parallelization," in *Design Automation Conference*, 2016, p. 123:1-6.

[17] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations," *J. Mach. Learn. Res.*, vol. 1, pp. 1–48, 2000.

[18] I. Hubara, D. Soudry, and R. El-Yaniv, "Binarized Neural Networks," in *Advances in Neural Information Processing Systems*, 2016.

[19] M. Courbariaux and Y. Bengio, "BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," *arXiv:1602.02830*, p. 9, 2016.

[20] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep Learning with Limited Numerical Precision," in *Int.Conf. Machine Learning*, 2015.

[21] W. Tang, G. Hua, and L. Wang, "How to Train a Compact Binary Neural Network with High Accuracy ?," in *AAAI Conference on Artificial Intelligence*, 2016, pp. 2625–2631.

[22] D. Yu *et al.*, "An introduction to computational networks and the computational network toolkit," *Tech. Rep., Microsoft MSR-TR-2014-112*, 2014.