LETTER *Special Section on Recent Advances in Machine Learning for Spoken Language Processing*

Speeding up Deep Neural Networks in Speech Recognition with Piecewise Quantized Sigmoidal Activation Function

Anhao XING[†a)], ***Nonmember***, **Qingwei ZHAO**[†], ***Member***, ***and*** **Yonghong YAN**[†,††], ***Nonmember***

SUMMARY This paper proposes a new quantization framework on activation function of deep neural networks (DNN). We implement fixed-point DNN by quantizing the activations into powers-of-two integers. The costly multiplication operations in using DNN can be replaced with low-cost bit-shifts to massively save computations. Thus, applying DNN-based speech recognition on embedded systems becomes much easier. Experiments show that the proposed method leads to no performance degradation.

key words: *deep neural networks, speech recognition, activation function, fixed-point quantization*

1. Introduction

The application of deep neural networks (DNN) in speech recognition has brought a great improvement in performance. DNN-based acoustic model outperforms conventional Gaussian mixture model (GMM) in various tasks [1]–[3]. However, DNN has much more parameters and requires a larger amount of computation, which is therefore difficult to be implemented on devices with lower computational capacity. All neurons of DNN are processing on the weighted sum of its inputs. Multiplication is the dominating computation, which is not hardware-friendly. Compared with other arithmetic operations, floating-point multiplication is much more energy-consuming.

This issue has drawn extensive attention. The technique using low-rank matrix factorization to reduce the number of parameters is proposed in [4], [5]. The full-rank weight matrix is represented by the product of two low-rank matrices, thus reducing the number of neurons as well as multiplications. The method in [6], [7] tries to find out the most critical weights based on some salience criterion and prunes all other weights which are considered unimportant. After pruning, the weight matrix becomes sparse, and sparse matrix-vector multiplication can be leveraged. Both methods exploit the redundancy in DNN and try to limit the number of free parameters. However, that would impact the modeling capacity of DNN, so deterioration in performance is inevitable [4]–[7].

As the energy cost of fixed-point arithmetic is much smaller than floating-point, a fixed-point implementation of DNN is introduced in [8]. All floating-point parameters—including weights, biases and activations—are converted into fixed-point integers using linear quantization. The method can save memory cost and computation time. However, the large amount of multiplication operations still hinders DNN from being applied on embedded systems, because the number of multipliers is usually limited on digital IC hardware such as FPGA.

In this paper, we present a novel quantization method to remove all multiplications. Differing from linear quantization, the proposed method quantizes the activations of DNN in a piecewise manner. Specifically, all activations are quantized into powers-of-two integers. Thus all multiplications can be replaced by bit-shifts. This makes it convenient to implement DNN on embedded systems since the demand for multipliers is greatly reduced, if not eliminated. The proposed quantization technique is orthogonal to the aforementioned redundancy-removing methods and can be easily combined with them.

The remainder of this paper is organized as follows. The next section reviews DNN and its fixed-point implementation. The piecewise activation quantization method is presented in Sect. 3. Experimental results are shown in Sect. 4. Section 5 concludes the work.

2. Fixed-Point Implementation of Sigmoidal DNN

A general deep neural network is a feed-forward neural network with more than one hidden layer. In speech recognition, DNN is used to estimate the posteriors given the observations. The vector of speech features $\boldsymbol{f}$ is propagated through each layer as follows:

$$\boldsymbol{x}^{(0)} = \boldsymbol{f} \tag{1}$$

$$\boldsymbol{x}^{(l)} = \sigma^{(l)}(\boldsymbol{W}^{(l)} \cdot \boldsymbol{x}^{(l-1)} + \boldsymbol{b}^{(l)}) \quad l = 1, \cdots, L \tag{2}$$

in which $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are the weight matrix and biases of layer l respectively; $\sigma^{(l)}(\cdot)$ is the activation function. For hidden layers, the activation function is usually element-wise sigmoidal function defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{3}$$

It is worth noting that the output of sigmoidal function ranges between 0 and 1. The small dynamic range of sigmoid makes sigmoidal DNN suitable for fixed-point implementation. Activations are linearly quantized into 8-bit

Manuscript received February 4, 2016.
Manuscript revised April 29, 2016.
Manuscript publicized July 19, 2016.

[†]The authors are with the Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China.
[††]The author is with the Xinjiang Laboratory of Minority Speech and Language Information Processing, China.
a) E-mail: xinganhao@hccl.ioa.ac.cn

DOI: 10.1587/transinf.2016SLL0007

unsigned integers that falls in the $[0, 256]$ interval in fixed-point implementation of DNN. Besides, weights are normalized and converted into 8-bit signed integers [8]. However, weights are quantized using high-precision data types such as the 16-bit short integer to ensure the performance in practical application. In this work, the weights of DNN are all converted into 16-bit short integers using linear quantization.

3. Quantizing Activations into Powers-of-Two Integers

As mentioned in Sect. 1, fixed-point implementation doesn't reduce the amount of free parameters, or the number of multiplications. Application of fixed-point DNN on embedded systems still bears huge computational load. To address the problem, we propose to remove all the multiplications by piecewise quantizing the activations. Instead of performing linear quantization on the activations, the activations are converted into powers-of-two integers. This means the outputs of quantized sigmoidal function can only take values from the set $\{0, 1, 2^1, \cdots, 2^{N-2}\}$, where N is the number of quantization stages. The process is equivalent to the following two procedures:

1) linearly quantizing the activations to the $[0, 2^{N-2}]$ interval in the same way as [8];
2) binding each quantized activation to its nearest powers-of-two integer.

The quantized sigmoid for the case $N = 8$ is illustrated in Fig. 1. It can be viewed from Fig. 1 that the sigmoidal function is quantized into 8 stages. Under the proposed quantization framework, computing in the DNN feed-forward mode can be simplified by substituting bit-shifts for multiplications.

The activations are quantized in the decoding phase. As the dynamic range of input features is large and uncertain, quantization is only performed on intermediate nodes after sigmoidal function. In our work, sigmoidal activations are obtained without any approximation because computing sigmoid function costs far less time compared with calculating the product of weight matrix and activation vector.

During the entire work, the weights are linearly quantized into short integers as described in Sect. 2. We don't perform piecewise quantization on weights as [9], because we find DNN is more susceptible to the quantization error of weights than to that of activations, which is shown in Fig. 2. Moreover, the convergence rate of retraining a network with powers-of-two weights can be very slow.

Figure 1 shows that the input-output mapping of sigmoidal function is changed by the proposed quantization method. This will lead to a reduction of DNN's classification accuracy in speech recognition tasks. To remedy the performance gap, we fine-tune the weights to better adapt to the piecewise quantized activations. The fine-tuning process is conducted on the converged floating-point DNN model in a straightforward way. It is almost the same as the standard backward propagation except that the floating-point activations are converted in the following manner:

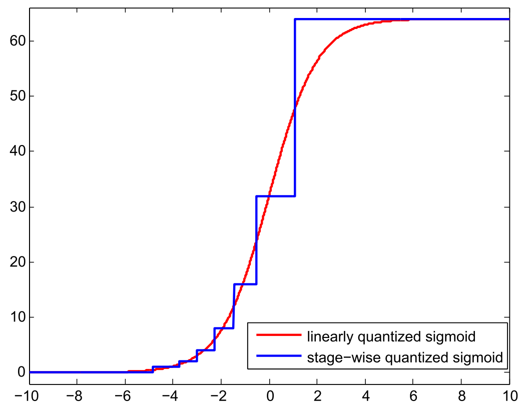

Fig. 1 Curve of quantized sigmoid for $N = 8$ (in comparison with linear quantization).

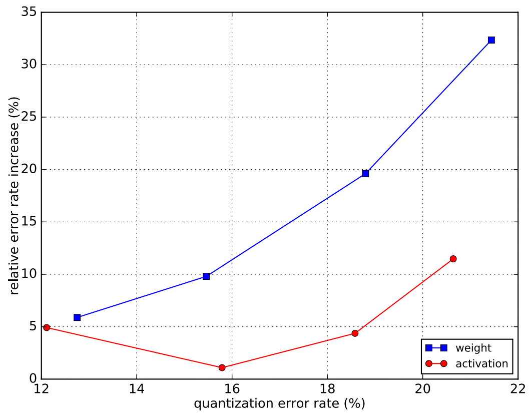

Fig. 2 The relative WER increase in speech recognition with regard to the quantization error. In each case, only weight or activation is quantized into powers of two while the other quantized linearly. Different quantization error is obtained by modifying the value of N.

$$\tilde{y} = \begin{cases} 0 & y \in (0, 2^{-N+1}) \\ 2^{-N+2} & y \in [2^{-N+1}, \frac{3}{4} \times 2^{-N+3}) \\ 2^{-n+1} & y \in [\frac{3}{4} \times 2^{-n+1}, \frac{3}{4} \times 2^{-n+2}) \\ & \qquad n = N-2, \cdots, 2 \\ 1 & y \in [\frac{3}{4}, 1) \end{cases} \tag{4}$$

where N is the number of stages into which the activations are quantized and larger than 2. In experiments, since the performance of piecewise quantized model is slightly worse than the floating-point model, only a single sweep over the training data has to be carried out.

4. Experiments

To evaluate the performance of the proposed method, experiments were carried out on a Chinese large vocabulary continuous speech recognition (LVCSR) task. The DNN model we used was trained with about 3000 hours of speech data, including varied styles such as broadcast, conversation, and telephone calls. 2 hours of speech data was used for testing.

Table 1 Experimental results of piecewise quantized DNN on speech recognition tasks. '-' and '+' denote that the result is obtained before and after fine-tuning respectively.

method		WER (%)	
floating-point		18.3	
linear quantization		19.0	
piecewise quantization	#stages	-	+
	N=8	19.2	19.0
	N=7	18.4	18.3
	N=6	19.3	19.2

The input feature was 13-dimention perceptual linear predictive (PLP) feature [10] with up to third-order derivatives. The feature was augmented with previous and next 5 frames to form a 572-dimension vector, which acted as the input to DNN. The DNN model has five hidden layers with 2048 nodes in each layer. The output layer consists of 19508 nodes, corresponding to tied context-dependent phonetic states.

The floating-point DNN was trained to convergence with the training data. We first applied fixed-point implementation on the DNN using linear quantization. Then the activations of the fixed-point model was re-quantized into power-of-two values. Different numbers of stages were tried and fine-tuning was performed. All the above models were used to perform speech recognition on the testing data and word error rate (WER) was calculated. For all the models tested, the decoding configuration stayed identical. Table 1 presents the results.

It can be noted the performance gap between piecewise and linear quantized model is fairly small. In the case of $N = 7$, the piecewise quantized model even performs as well as the original unquantized floating-point model in terms of WER. N here can be treated as a hyper-parameter that needs to be cross-validated. Meanwhile, we can observe that the WER reduction brought by fine-tuning is rather limited (0.5% to 1% relative). We would like to ascribe this to DNN's insensibility to the quantization error of activations. The fixed-point model with piecewise quantized activations performs well enough, so there is not much room for improvement left to fine-tuning.

We also evaluated the speed performance of two quantization frameworks. The platform on which we carried out experiments is Cyclone IV E FPGA (EP4CE22E22C8). This device has 132 embedded 9-bit multiplier elements which can be used for calculating multiplications, and 22,320 logic elements reserved for bit-shifting or other operations. As is expected, the number of multipliers is rather small, which is quite common among FPGA-like embedded platforms. In this experiment, only the speed of DNN's feed-forward process was tested. The result is shown in Table 2. The speed performance is measured in the number of speech frames processed by FPGA per second (FPS). The clock frequency of the used FPGA is 50MHz.

As is verified by the experiment, since there are abundant resources for bit-shifting operations on FPGA, piecewise quantized DNN can be easily designed to compute in a more efficient parallel fashion and thus obtains a better speed performance on FPGA than linearly quantized DNN.

Table 2 Comparison of speed performance of two quantization frameworks on FPGA.

method	FPS
linear	109
piecewise	316

5. Conclusion

In this paper, we present a more hardware-friendly quantization framework for deep neural networks. Under this new framework, activations are converted into fixed-point values using piecewise quantization instead of conventional linear quantization. By quantizing the output of activation function into powers-of-two integers, the multiplication operations that dominate in DNN can be replaced by bit-shifts. Thus, the computational resource demand can be alleviated. The piecewise quantization is not performed on weights because DNN is more sensitive to the quantization precision of weights. In experiments, the proposed quantization method is found to be well compatible with speech recognition scenario: the model with piecewise quantized activations performs as well as the model with linearly quantized activations. Also, the accuracy gap between piecewise quantized and the original floating-point models is negligible; this is achieved before a fine-tuning process is conducted. Nevertheless, the performance can be further improved by fine-tuning. Moreover, the speed superiority of the proposed quantization method is verified on an FPGA platform. This work makes it possible to deploy DNN-based speech recognizer on embedded systems with restricted computational resources.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (Nos. 11461141004, 61271426, 11504406, 11590770, 11590771, 11590772, 11590773, 11590774), the Strategic Priority Research Program of the Chinese Academy of Sciences (Nos. XDA06030100, XDA06030500, XDA06040603), National 863 Program (No. 2015AA016306), National 973 Program (No. 2013CB329302) and the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (No. 201230118-3).

References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," Signal Processing Magazine, IEEE, vol.29, no.6, pp.82–97, 2012.
[2] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," IEEE Trans. Audio, Speech, Language Process., vol.20, no.1,

pp.30–42, 2012.

[3] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," Interspeech, pp.437–440, 2011.

[4] T.N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.6655–6659, IEEE, 2013.

[5] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," INTERSPEECH, pp.2365–2369, 2013.

[6] D. Yu, F. Seide, G. Li, and L. Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.4409–4412, IEEE, 2012.

[7] C. Liu, Z. Zhang, and D. Wang, "Pruning deep neural networks by optimal brain damage," Proceedings of Interspeech, 2014.

[8] V. Vanhoucke, A. Senior, and M.Z. Mao, "Improving the speed of neural networks on cpus," Proc. Deep Learning and Unsupervised Feature Learning NIPS Workshop, 2011.

[9] C.Z. Tang and H.K. Kwan, "Multilayer feedforward neural networks with single powers-of-two weights," IEEE Trans. Signal Process., vol.41, no.8, pp.2724–2727, 1993.

[10] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," the Journal of the Acoustical Society of America, vol.87, no.4, pp.1738–1752, 1990.