A DBN Hardware Accelerator for Auditory Scene Classification

Ching-Che Chung, *Senior Member*, *IEEE*, Huai-Xiang Zhang, Ming-You Hung, and Hong-Jin Jian
Department of CSIE, National Chung Cheng University
No. 168, University Rd., Min-Hsiung, Chia-Yi, Taiwan
Email: wildwolf@cs.ccu.edu.tw

***Abstract*-- As the population ages and enters the aging society, related medical problems are gradually being taken seriously, and the deterioration of hearing due to aging is also a common problem. However, hearing aids are easily affected by different auditory scene, and different compensation methods are needed in the different auditory scene. In this paper, the deep belief network (DBN) is used to identify the scene of the auditory scenes. After the deep neural network operation, the current auditory scene is judged. In this paper, the memory requirement for the DBN is reduced in post-training quantization using the K-mean clustering algorithm, and 93.743% memory space of the DBN is reduced with only 1.65% accuracy loss.**

I. INTRODUCTION

In recent years, with the development of speech recognition technology and the coming of the aging society, hearing aids (HA) become an essential research topic. However, the effect of traditional HA's auditory compensation is susceptible to background noise. A better prescription for auditory compensation is provided by using a deep neural network (DNN) to identify the sound field. To be made into a wearable device, the power consumption and chip area are limited. Also, the compensation of the auditory is time-limited. Therefore, the latency of the circuit is also limited. Because of these limitations, GPUs and CPUs are not suitable for this application. In this paper, the deep belief network (DBN) hardware accelerator is implemented for recognizing the auditory scene. The proposed DBN hardware accelerator can be used for hearing aids to increase the compensation effect of the hearing aids with the different auditory scene.

A DBN is built by stacking multiple restricted Boltzmann machines (RBMs) [1]. An RBM consists of hidden units and visible units. The hidden layer of the RBM will be the visible layer of the next layer RBM. Generally, more hidden layers, the accuracy of the deep neural network will be better. The DBN uses the sigmoid activation function, $f(x)=1/(1+e^{-x})$. The sigmoid activation function is non-linear, and the output is restricted between 0 and 1. Generally, a large number of weights are trained in DBN. If all weights are stored in floating-point numbers, many memory spaces are needed and will increase the chip area. Also, the power consumption in memory read and write operations is increased accordingly. Therefore, quantizing and reducing bits of weights is essential.

In [2], energy-efficient data flow methods for the DNN are proposed to reduce power consumption and speed up the operation of the circuit. A quantitative method with the pre-trained DNN is presented in [3]. Also, the other methods of compressing, quantifying, and reducing weights of the DNN are proposed in [4].

II. PROPOSED DBN ARCHITECTURE

The DBN model used in this paper was trained using "hearing in noise test (TMHINT) sentences" of the Chinese version. The sound data are extracted to 39-dimension features by Mel-frequency cepstral coefficients (MFCCs) and then input to the DBN for classification. The test dataset contains 12 different auditory scenario data. After the auditory scenes is determined, the auditory compensation is applied.

Table I. Accuracy and compression rate.

Architecture	Memory (bits)	Compression rate	Accuracy
Original 32bit	813,184	-	100%
Fixed-Point: 8bit	203,296	25%	100%
4 groups K-means clustering algorithm by each layers			
W 8bit, B 2bit	50,952	6.265%	98.88%
W 4bit, B 2bit	50,888	6.257%	97.91%
W 3bit, B 2bit	**50,884**	**6.257%**	**98.35%**
4 groups K-means clustering algorithm by all layers			
W 8bit, B 2bit	50,856	6.253%	98.34%
W 4bit, B 2bit	50,840	6.251%	95.46%

The design flow of the proposed DBN hardware accelerator is explained as follows. First, different post-training quantization methods are applied to reduce the memory size of the DBN model and then the classification accuracy is verified by MATLAB. When the best quantization method for the DBN is determined, the model parameters can be extracted. Subsequently, the DBN hardware architecture is developed to meet the latency requirement. Finally, the function of the DBN circuit and the classification accuracy are verified in gate-level simulation and FPGA.

The accuracy and compression rate for the DBN model with different quantization algorithm is shown in Table I. The proposed DBN model consists of 25,100 weights and 312 biases. K-means clustering algorithm is applied for DBN post-training quantization, and weights are grouped as fewer kinds of values. Then, the lookup table can be used to further reduce the memory spaces of the DBN model. For example, when the K-means clustering algorithm divides the weight values in each layer into 4 groups, because each layer has only four weight values, so 2 bits are enough to store one weight value. Subsequently, the look-up table can be used to convert 2 bits

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST-108-2221-E-194-051- and was financially/partially supported by the Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

978-1-7281-7399-3/20/$31.00 ©2020 IEEE

into the actual 3 bits fixed-point number in neuron computation. Using this method can effectively reduce the memory usage by 93.743%, and the compression ratio is 6.257%. However, the accuracy of the DBN drops slightly to 98.35%. The latency requirement is the critical issue for designing the proposed DBN hardware accelerator. The architecture with parallel hardware units is used to accelerate the operation of the DBN. Then, 10 MAC units is used in the proposed DBN at the clock rate 250 MHz to meet the latency requirement and keep relatively low power consumption.

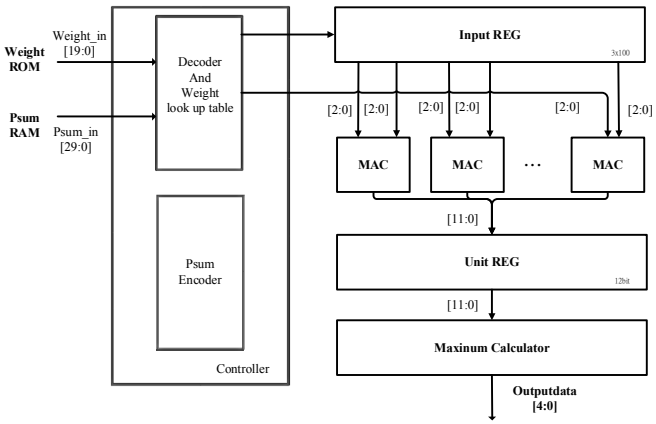

Fig. 1. The last layer architecture of the proposed DBN.

Input data are stored in Input REG before performing the neuron calculation of the first layer, as shown in Fig. 1. When the input is ready, the controller will read 20 bits data from the Weight_ROM. The 20 bits data will be cut into 10 pieces of 2-bit data by the decoder. Then, each 2-bit data will be converted to the actual value of the weight by the weight look-up table. Also, the weight data format is a 6-bit signed number in the first layer which consists of a 1-bit integer and 5-bit decimal bits, and is a 3-bit signed number which consists of 3-bit decimal bits in the following layer. Then, the weight values will be sent to 10 MAC units to perform neuron calculation. When the calculation is completed, the partial sum result will be accumulated in Unit REG. When the calculation for the same neuron is completed, the sigmoid function is calculated by the lookup table. Subsequently, the Psum Encoder in the controller combines 10 neurons' output into 30-bit data and sends them to Psum_RAM for storage.

In other layers, the controller reads the 30-bit data of the previous layer from Psum_RAM at each clock cycle, and each input is expressed 3-bit decimal bits format. The weight value is also expressed in 3-bit decimal bits format, and these values are sent to 10 MAC units for neuron computation. The neuron calculation process is similar in the second, third, and the fourth layers. Also, the MAC units in the second, third, and fourth layers are shared. The last layer does not calculate the sigmoid function but the maximum value is selected as output.

III. Experimental Results

The proposed DBN is implemented in TSMC 40nm CMOS process. The gate count of the proposed DBN is 12,134 with 50,856 bits weight ROM, and the highest clock rate is 250 MHz in gate-level simulation. The GOPS of the proposed DBN using 10 MAC units is 3.957, and the classification accuracy is 98.35%. Table II shows the comparison with the other studies. By reducing the memory usage by 93.747%, it can greatly reduce the size of the memory required to store the network model and the number of times the memory is read and written. At the same time, in this paper, except for the first layer, only 3 bits of data are used for calculation, so the power consumption of the circuit operation can be effectively reduced, and the execution speed of the DBN circuit is accelerated. As a result, in this paper, the power consumption and area cost of the DBN circuit have been improved, and it is very suitable for low-power applications applied to hearing aids.

Table II. Performance Comparisons

	This work	[5]	[6]	[7]
Technology	40nm	65nm	40nm	40nm
Design target	Fully-connected layer	Convolutional layer	Fully-connected layer & FFT	Fully-connected layer
Power(mW)	3.137	278	0.288*	18.5
Clock rate(MHz)	250	100-250	1.9-19.3	250
Efficiency(GOPS)	3.957	2.6688	0.107	N/A
Architecture precision	2-bit fixed-point	16-bit fixed-point	16-bit fixed-point	16-bit fixed-point
Accuracy	98.35%	N/A	N/A	99.6%

*DRAM power not included.

IV. Conclusion

In this paper, the post-training quantization method using the K-means clustering algorithm is presented to reduce the memory usage of the DBN model and maintain over 98% of classification accuracy. Besides, the proposed DBN circuit which implemented in TSMC 40nm CMOS process can runs at 250 MHz. The GOPS of the proposed DBN is 3.957. The proposed DBN hardware accelerator can be used for hearing aids to increase the compensation effect of the hearing aids with the different auditory scene.

Reference

[1] A. Fischer, et al., "An introduction to restricted Boltzmann Machines," *in Proc. CIARP*, Sep. 2012, pp. 14-36.

[2] Y.-H. Chen, et al., "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127-138, Jan. 2017.

[3] H. Song, et al., "Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding, arXiv:1510.00149v5 [cs.VC]," *arXiv.org*, Feb. 2016.

[4] J.-H. Ko, et al., "Adaptive weight compression for memory-efficient neural networks," *in Proc. DATE*, Mar. 2017, pp. 199-204.

[5] A. Biswas, et al., "Conv-RAM: An energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications," *in Proc. ISSCC*, Feb. 2018, pp. 488-489.

[6] S. Bang, et al., "A 288μW programmable deep-learning processor with 270KB on-chip weight storage using non-uniform memory hierarchy for mobile intelligence," *in Proc. ISSCC*, Feb. 2017, pp. 250-251.

[7] B. Reagen, et al., "Minerva: Enabling low-power, highly-accurate deep neural network accelerators," *in Proc. ISCA*, Jun. 2016, pp. 267-278.