Design of a DBN Hardware Accelerator for Handwritten Digit Recognitions

Ching-Che Chung, *Senior Member, IEEE*, Yi-Zeng Lee, and Huai-Xiang Zhang

ABSTRACT

With the development of artificial intelligence, researches on speech recognition and deep learning become increasingly popular. In this paper, the deep belief network (DBN) hardware accelerator for handwritten digit recognition was implemented in TSMC 90nm CMOS process. Addition, the MNIST database is used as a functional verification for the proposed hardware architecture, and the accuracy of the proposed design is 97.3%. The gate count of the proposed design is 1,160k, and power consumption is 353mW at 73.6MHz. The energy efficiency of the proposed design is 1.7337GOPs/W.

INTRODUCTION

In the past few years, the restricted Boltzmann machine (RBM), is applied to many applications, such as image recognition and sound analysis. When the RBMs are stacked, we can form a deep neural network named as deep belief neural network (DBN). For DBN implementation, in [1], the stochastic number generator generates a set of evenly distributed stochastic streams. It uses the stochastic number generator to process the input data and weight values and then performs calculations for each neural unit. To improve the speed performance of the DBN, [2] uses FPGAs serial connections. Through the sharing of resources in each FPGA, computational allocation and parallel operations increase the overall performance. However, the speedup of the proposed architecture is limited by the performance of the FPGA board itself. In [3], a hardware-efficient sigmoid function with adjustable precision is proposed.

THE PROPOSED DBN ARCHITECTURE

In this paper, the proposed DBN architecture uses the MNIST database to verify the proposed design. The MNIST database contains 60,000 training digits and 10,000 test digits. The DBN is trained by the MATLAB toolbox. The range of weight values and the fractional bit requirements simulated by MATLAB are observed to determine the bits requirement in hardware design to retain the accuracy.

The DBN neural network architecture is 784×256×256×256 ×10, totally has four layers. In the proposed architecture, the sigmoid function is used as an activation function, and the sigmoid function is implemented with a segmented look-up table, as shown in Fig. 1. By dividing the input value of the sigmoid function into different intervals, the index value of each interval contains a 4-bit integer and a different number of fractional bits according to the interval. When the input value falls in 0 to 1 or 0 to -1, the index value consists of a 4-bit integer and a 6-bit fractional number. When the input value falls in 1 to 2 or the input value falls in -1 to -2, the index value consists of a 4-bit integer and a 5-bit fractional number, and so on. After the look-up table, the input value will be converted into a 16-bit fixed-point number with a 4-bit integer part and 12-bit fractional part.

The block diagram of the proposed DBN architecture is shown in Fig. 2. The first layer (L1) has 32 weight ROMs, the second layer (L2) and the third layer (L3) has eight weight ROMs, and the fourth layer (L4) has one weight ROM. The ROM compiler has the smallest memory size limitation, and thus, all bias values are stored in the same ROM. Addition, the first layer has 32 multiplication-and-addition (MAC) units, the second layer (L2) and the third layer (L3) has 8 MAC units, and the fourth layer (L4) has one MAC unit.

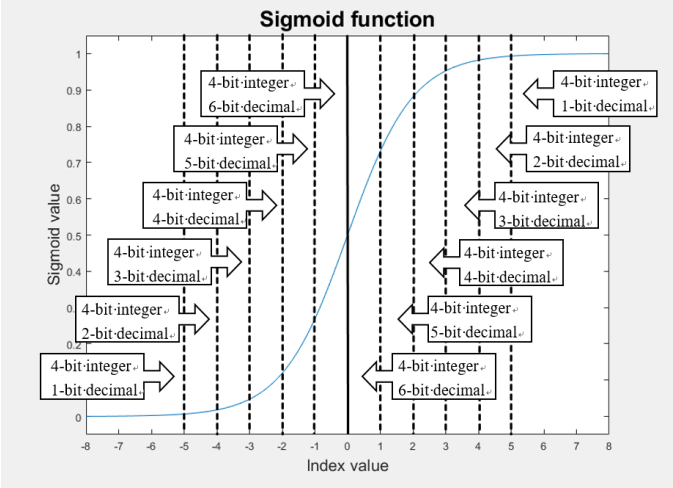

Fig. 1. The proposed segmented sigmoid function look-up table.

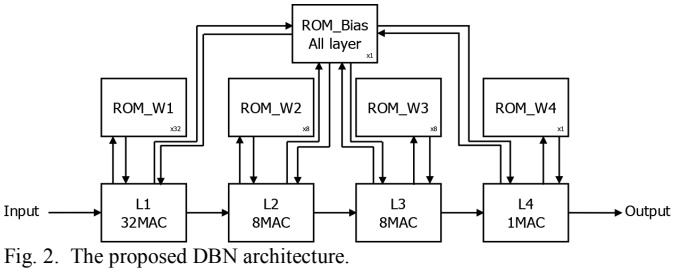

Fig. 2. The proposed DBN architecture.

CIRCUIT IMPLEMENTATION

The timing diagram of the first layer (L1) is shown in Fig. 3. When the In_valid signal rises, the inputs of the first layer are stored into Input_REG sequentially. Subsequently, MAC units calculate the neurons of different parts through the switching

This work was supported in part by the Ministry of Science and Technology of Taiwan, under Grant MOST-107-2221-E-194-031.

978-1-7281-3279-2/19/$31.00 ©2019 IEEE

of STATE. Each part has 32 neurons calculated. The first layer (L1) calculates 256 neurons through eight parts operations. Unit_REG adds bias value when the STATE is 8. Then, Unit_REG sends the results to the Layer1 sigmoid_table. Finally, the data are sent to the next layer.

The second layer (L2) and the third layer (L3) have the same operation flow. In the fourth layer (L4), the fourth layer calculates ten neurons through ten parts operations with one MAC unit. Unit_REG adds bias value when the STATE is 10. Then, a maximum search unit (MAX_REG) selects the largest output neuron from the ten neurons and output the handwriting digits classification result.

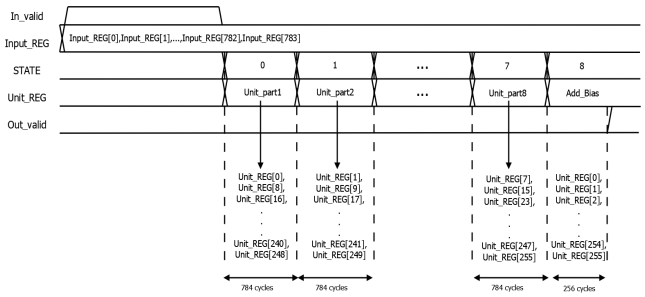

Fig. 3. The timing diagram of the first layer (L1).

TABLE I
PERFORMANCE COMPARISON

	This work	[4]	[5]
Technology	90nm	40nm	65nm
Design target	Fully-connected layer	Fully-connected layer	Convolutional layer
Power(mW)	353	125*	278
Clock rate(MHz)	73.6	250	200
Area(gate count)	1160k	N/A	1176k
Area(mm^2)	N/A	3.145	16
Architecture	16-bit	16-bit	16-bit
Accuracy	97.3%	99.6%	N/A

*estimated.

EXRPERIMENTAL RESULTS

The DBN model trained in MATLAB are floating-point numbers, and then the weight and bias values are formatted as 27-bit, 16-bit, 8-bit and 4-bit fixed-point numbers. The classification accuracy for different bit configuration is compared with MATLAB simulation results using floating-point numbers. The proposed DBN architecture can still maintain the same classification accuracy when using 16-bit weight and bias data. Under the 8-bit weight and bias data condition, only 1.3% accuracy loss between the floating-point calculations can be found. Addition, in 4-bit weight and bias data condition, the proposed DBN architecture loses the ability to perform classification.

After the proposed DBN is implemented in TSMC 90nm CMOS process. The power consumption at 73.6MHz for the best case, typical case, and the worst case conditions are 418mW, 353mW, and 311mW, respectively. The gate count of the proposed design circuit is 1,160k. The maximum clock rate of the proposed DBN circuit is 73.6MHz which is restricted by the critical path of the MAC units. The precision of the proposed architecture is 16-bit fixed-point. The proposed DBN architecture uses 334k MAC operations for computing one classification result. The energy efficiency of the DBN circuit for the best case, typical case, and the worst case conditions are 1.46411GOPs/W, 1.7337GOPs/W, and 1.9678GOPs/W, respectively. Table I illustrates the comparison between the proposed DBN circuit and the existing systems [4-5]. The energy efficiency of the proposed DBN circuit can be further improved by using the advanced CMOS process, such as 40nm CMOS process.

CONCLUSION

In this paper, a hardware accelerator for the DBN neural network is implemented and the impact of the recognition result due to the bit representation of the weight and bias data is discussed. In the DBN model extraction, we use the MATLAB toolbox to simulate the operation of the DBN network. First, the MATLAB toolbox is used to find out the suitable neuron number in each layer. Next, the DBN hardware circuit is implement in TSMC 90nm CMOS process. In the simulation results, different precision on the extracted DBN models are tested, and then, the minimum precision required for weight and bias can be obtained. The power consumption of the proposed DBN circuit is 353mW, and the maximum clock rate of the DBN circuit is 73.6MHz. The proposed DBN architecture uses 334k MAC operations for computing one classification result. The energy efficiency of the proposed DBN circuit is 1.7337GOPs/W.

REFERENCES

[1] K. Sanni, et al., "FPGA implementation of a deep belief network architecture for character recognition using stochastic computation," *in Proc. Information Sciences and Systems*, Mar. 2015.

[2] D. L. Ly and P. Chow, "High-performance reconfigurable hardware architecture for restricted Boltzmann machines, " *IEEE Trans. Neural Networks*, vol. 21, no. 11, pp. 1780-1792, Nov. 2010.

[3] C.-H. Tsai, et al, "A hardware-efficient sigmoid function with adjustable precision for a neural network system, " *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 62, no. 11, Nov. 2015.

[4] B. Reagen, et al., "Minerva: enabling low-power, highly-accurate deep neural network accelerators," *in Proc. ACM/IEEE 43rd Annual International Symposium on Computer Architecture*, Jun. 2016, pp. 267-278.

[5] 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.