A 41.3/26.7 pJ per Neuron Weight RBM Processor Supporting On-Chip Learning/Inference for IoT Applications

Chang-Hung Tsai, Wan-Ju Yu, Wing Hung Wong, and Chen-Yi Lee

***Abstract*— An energy-efficient restricted Boltzmann machine (RBM) processor (RBM-P) supporting on-chip learning and inference is proposed for machine learning and Internet of Things (IoT) applications in this paper. To train a neural network (NN) model, the RBM structure is applied to supervised and unsupervised learning, and a multi-layer NN can be constructed and initialized by stacking multiple RBMs. Featuring NN model reduction for external memory bandwidth saving, low power neuron binarizer (LPNB) with dynamic clock gating and area-efficient NN-like activation function calculators for power reduction, user-defined connection map (UDCM) for both computation time and bandwidth saving, and early stopping (ES) mechanism for learning process, the proposed system integrates 32 RBM cores with maximal 4k neurons per layer and 128 candidates per sample for machine learning applications. Implemented in 65nm CMOS technology, the proposed RBM-P chip costs 2.2 M gates and 128 kB SRAM with 8.8 mm^2 area. Operated at 1.2 V and 210 MHz, this chip achieves 7.53G neuron weights (NWs) and 11.63G NWs per second with 41.3 and 26.7 pJ per NW for learning and inference, respectively.**

***Index Terms*— Low-power design, machine learning, memory bandwidth reduction, non-linear functions, restricted Boltzmann machine (RBM).**

I. Introduction

RECENTLY, massive datasets have been generated by sensors, social networks, and mobile devices. How to extract meaningful information from the raw dataset to support in-time model learning and real-time decision making has become an emerging research topic. Consequently, machine learning techniques [1]–[4] have been widely and successfully applied to the multimedia and signal-processing systems, and the deep neural network (DNN) approach is considered one of the state-of-the-art solutions for applications [5]. With a DNN structure, multi-level features can be detected layer-by-layer to extract meaningful information from input data. And each neuron is turned ON or OFF based on the neural network (NN) model and non-linear activation function.

Since the NN becomes deeper and deeper to provide more powerful and accurate performance for machine learning applications, the traditional learning algorithm with forwarding and error backpropagation is inefficient to train an NN with multiple layers. And the other drawback of the traditional approach is that the algorithm belongs to supervised learning which means the label information for each training data is required to calculate the error value in the model learning process.

In this paper, a restricted Boltzmann machine-processor (RBM-P) chip is proposed to support diverse network structures, in-time NN model training, and real-time decision making in the application phase, and the key contributions of the proposed system are as follows.

1) Since the data access from external memory costs long latency and dominates the system performance, the external memory bandwidth is reduced by the proposed computation-efficient model reduction method with negligible accuracy loss.
2) To implement non-linear activation functions, the look-up table (LUT) approach is commonly used in conventional hardware designs, but a dedicated memory is required to store values for each non-linear activation function calculator (AFC). To reduce the power consumption and hardware cost, a low-power neuron binarizer (LPNB) with dynamic clock gating and area-efficient NN-like AFC is designed for neuron computation, and both sigmoid and hyperbolic tangent non-linear functions are supported.
3) To flexibly and efficiently support diverse NN structures, a user-defined connection map (UDCM) module is implemented to save both computation time and external memory bandwidth.
4) In the conventional learning engine, the number of model training iterations are configured by user during the learning process. In our system, an early stopping (ES) mechanism is proposed to continuously monitor the NN model to automatically terminate the learning process to save computation time.

The rest of this paper is organized as follows. In Section II, the RBM learning algorithm and related works are described. The system architecture of the RBM-P chip and proposed low-power techniques is presented in Section III. The

Manuscript received February 12, 2017; revised May 11, 2017; accepted June 6, 2017. Date of publication July 3, 2017; date of current version September 21, 2017. This work was approved by Guest Editor Jaeha Kim. This work was supported in part by the MOST of Taiwan under Grant MOST 103-2221-E-009-198 and in part by the NSF of U.S. under Grant DMS 1330132 and Grant DMS 1407557. *(Corresponding author: Chang-Hung Tsai.)*

C.-H. Tsai, W.-J. Yu, and C.-Y. Lee are with the Institute of Electronics, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: allen@si2lab.org; cylee@si2lab.org).

W. H. Wong is with the Department of Statistics, Stanford University, Stanford, CA 94305 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2017.2715171

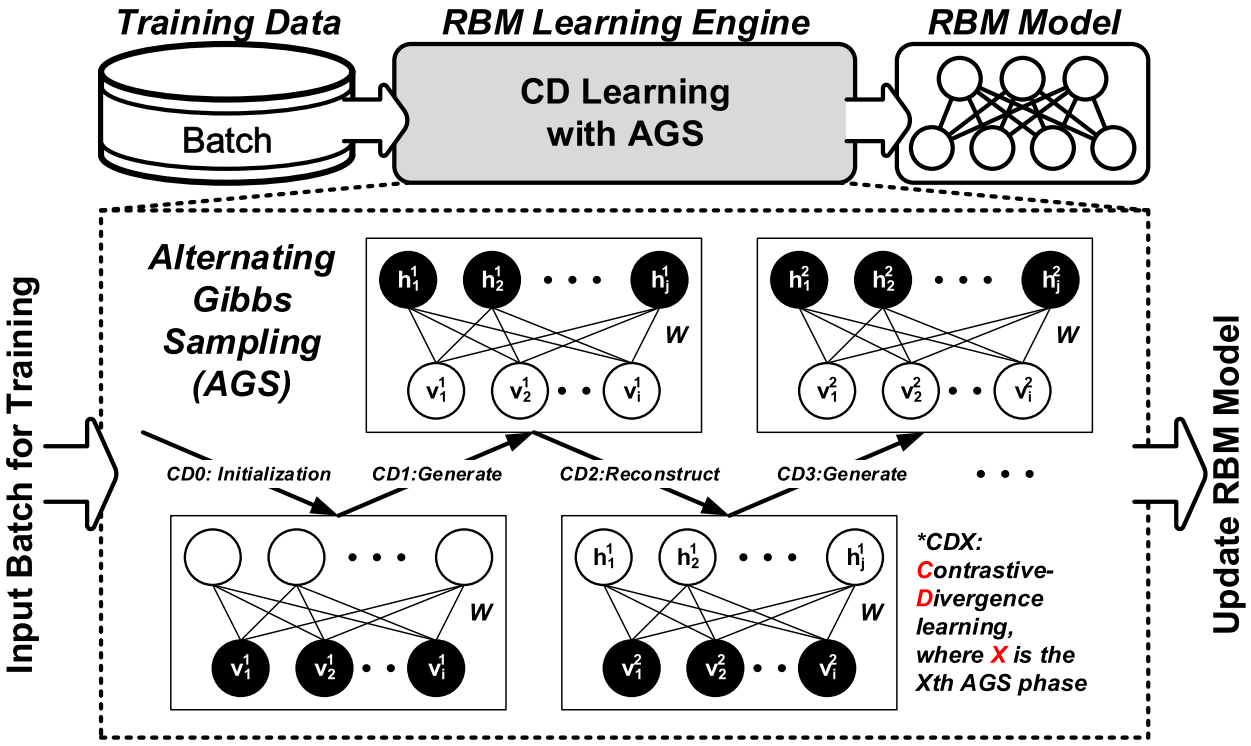


Fig. 1. AGS with CD algorithm for the RBM model learning.

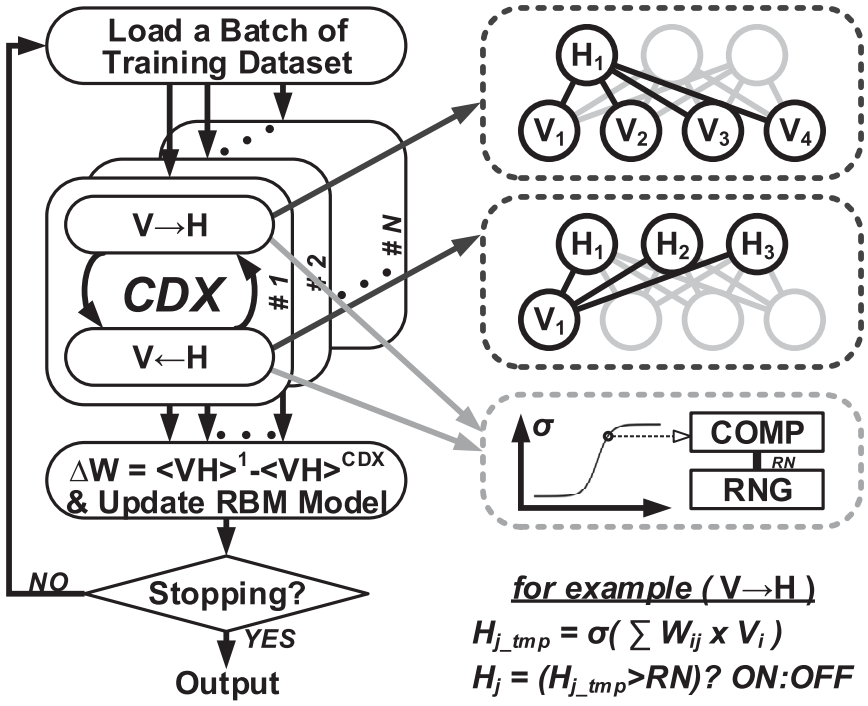


Fig. 2. Flow chart of the CD learning with AGS for RBM-based NN model learning.

implementation results, measurement results, and system platform are shown in Section IV. Finally, the conclusion is given in Section V.

II. Background

In this section, we first briefly review state-of-the-art NN processors related to the proposed system for machine learning applications. Then, the details of the applied NN model learning algorithm are described.

A. Related Works

Currently, the NN algorithms are considered one of the state-of-the-art solutions in multimedia and signal-processing systems. Therefore, several NN processors [6]–[9] have been

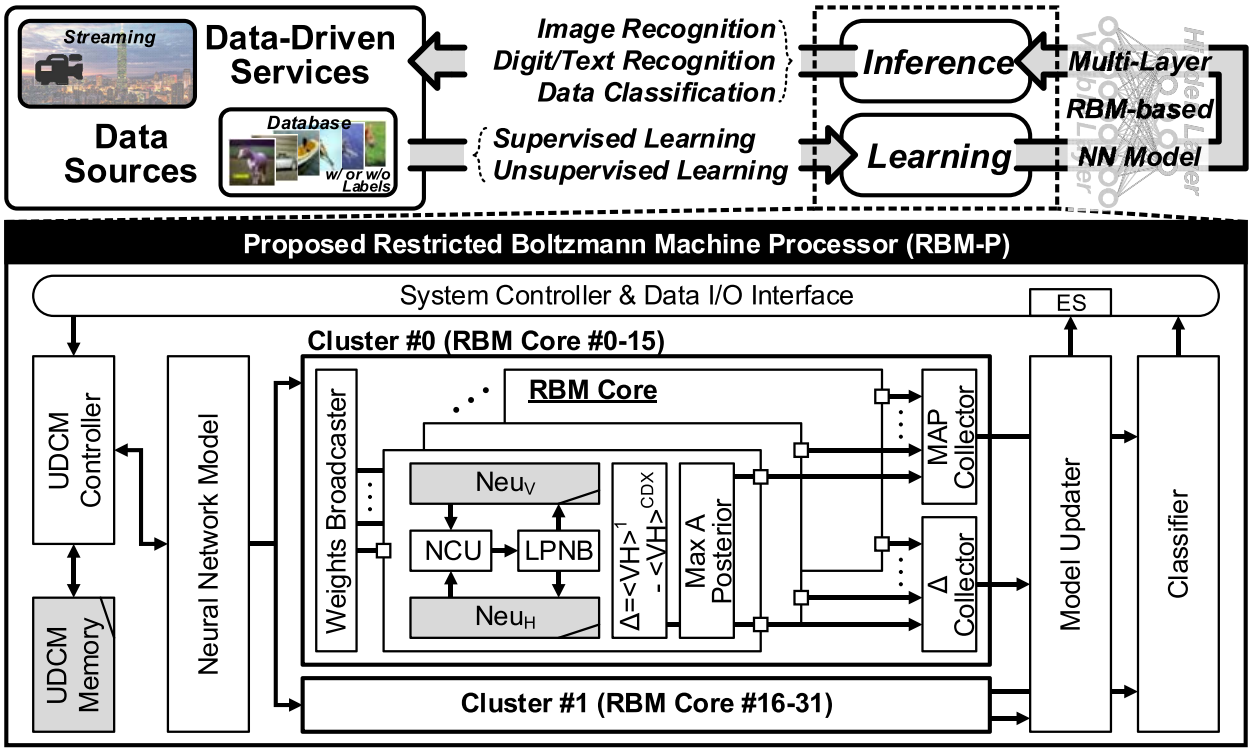


Fig. 3. System architecture of the proposed RBM-P chip.

proposed for object classification and recognition in multimedia applications.

Kim *et al*. [6], [8] proposed an event-driven NN processor to support on-chip learning and inference. However, the neuron signals are transposed to spike data then fired into the sparse NN for integrations, and 10 objects that can be recognized for object classification. Park *et al*. [7] proposed a scalable deep learning and inference processor for the big data applications. The RBM method is applied to the model learning, and dual network structures, convolutional and fully connected networks, are supported in the inference phase. Nevertheless, the NN topology of convolutional filter is fixed, and only 256 neurons are processed in each learning and inference core.

B. Restricted Boltzmann Machine

In the proposed system, the RBM is applied to the NN model learning [10], [11], and a multi-layer NN can be constructed by stacking multiple RBMs. During the multi-layer NN model learning, each layer is modeled to an RBM which consists of visible neurons (v_i) and hidden neurons (h_j) in visible layer and hidden layer, respectively. And the contrastive divergence (CD) learning algorithm with alternating Gibbs sampling (AGS) [12] is exploited to efficiently train an RBM, as shown in Fig. 1.

In the learning process, the hidden and visible neurons are generated and reconstructed from the visible and hidden layer, respectively. First, the neuron data of visible layer is assigned from the training dataset. Then, the weighted summation operation is performed to add up the products of visible neurons multiplied by the corresponding weights. In the third step, the weighted summation result is fed into a non-linear activation function (σ), and then compared with a random number (RN) to decide the neuron is turned ON or OFF. After the iterative computation, the update data are calculated in the first and last AGS phase. Finally, the RBM model is updated in the CD learning, and the model learning process is terminated until the value of iteration equals system parameters configured by user or the RBM model is convergent, as shown in Fig. 2.

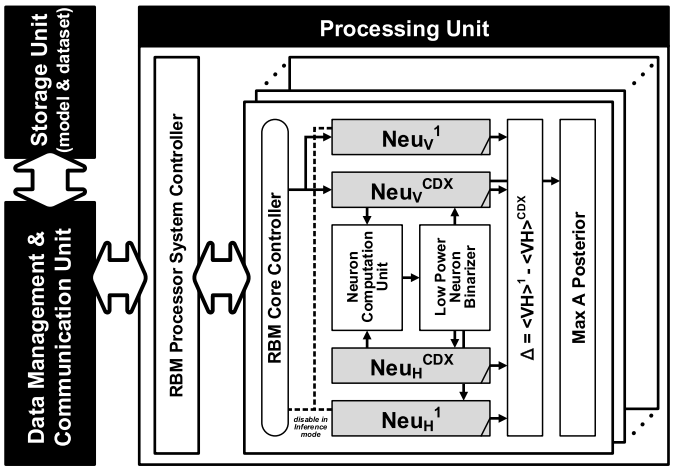


Fig. 4. Architecture of the proposed RBM core.

III. System Architecture and Low-Power Design

Fig. 3 shows the system architecture of the proposed RBM-P chip for NN applications. Considering the hardware cost and performance, 32 RBM cores are integrated in two clusters for massively parallel computing in both learning and inference phases with the structure of maximal 4k neurons per

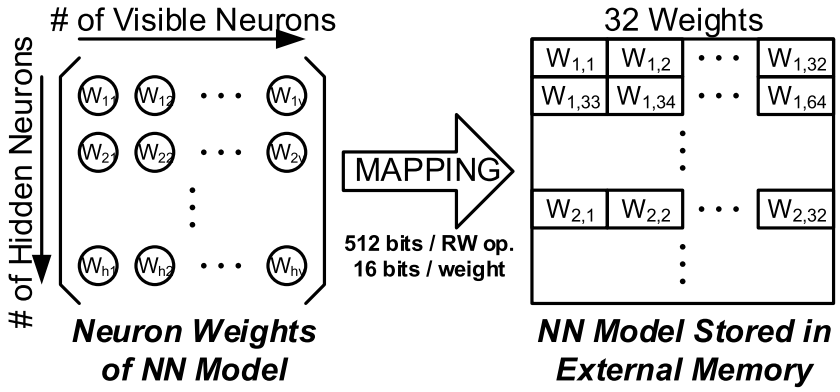


Fig. 5. Data layout of the NN model stored in the external memory.

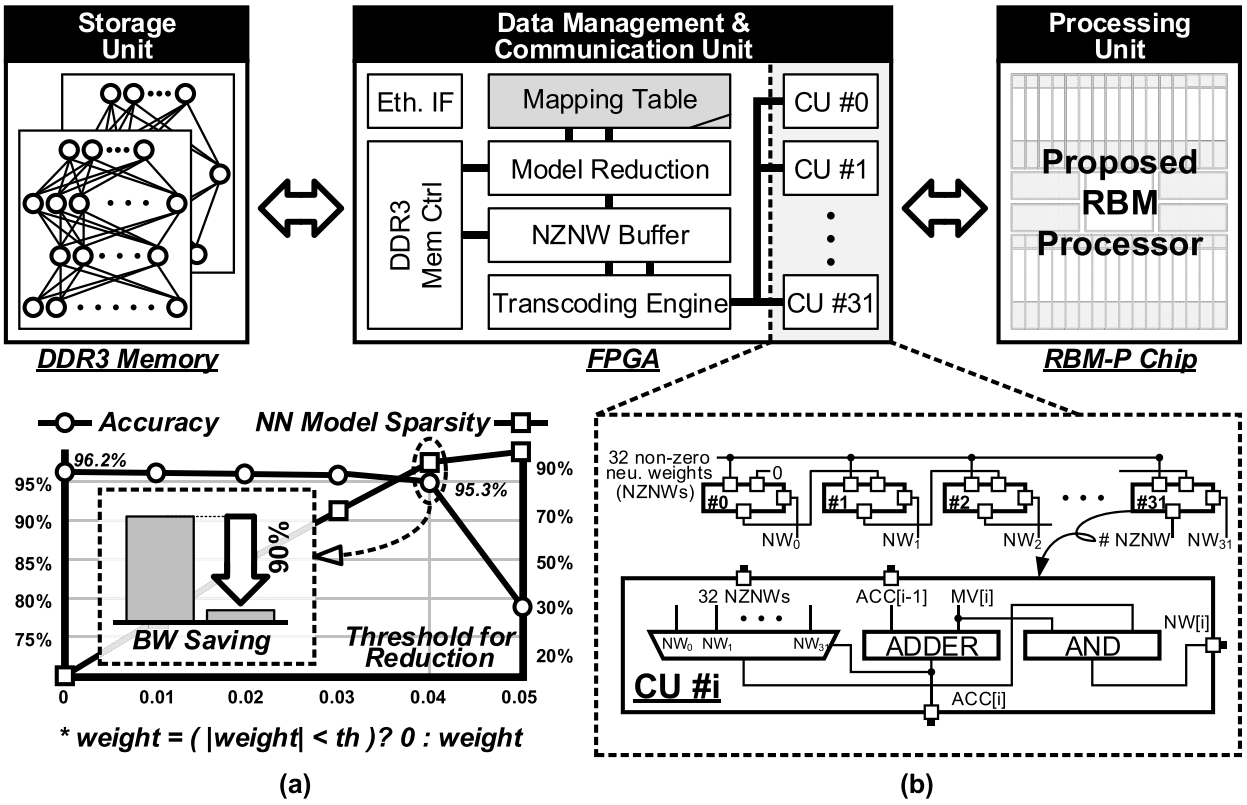
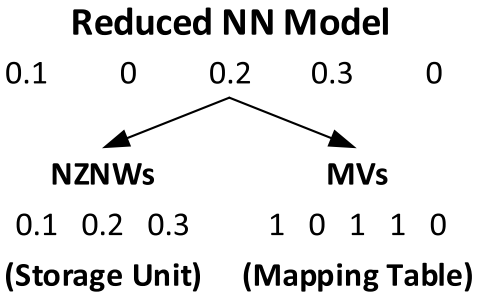


Fig. 6. Architecture of the proposed model reduction method. (a) Experimental results of model reduction. (b) Hardware architecture of CU.

layer for Internet of Things (IoT) applications in our system. Operated in the learning mode, the proposed RBM-P chip loads the network structure setting and system parameters to configure each RBM core. Then, a batch of the training dataset is loaded to perform the CD learning algorithm and update the RBM-based NN model iteratively. Operated in the inference phase, testing data are fed into RBM cores, and the maximum a posterior (MAP) inference results will be outputted after computation. Moreover, an UDCM is proposed for diverse networks to meet different system configuration requirements, and the proposed ES engine is embedded in the system I/O interface to continuously monitor the NN model during learning process.

To perform neuron computation, an RBM core is designed for supporting both learning and inference in our system. Fig. 4 shows the architecture of the proposed RBM core, and the key building blocks include visible (Neu_V) and hidden (Neu_H) neuron memories which support maximal 4k neurons per layer for computation, neuron computation unit for hidden neuron generation (visible layer to hidden layer) and visible neuron reconstruction (hidden layer to visible layer), LPNB for neuron state decision, delta calculator to perform computation of $\langle \text{VH} \rangle^1 - \langle \text{VH} \rangle^{\text{CDX}}$ for model learning, and MAP calculator for data inference. In the proposed system, an auxiliary platform is applied to implement the storage unit and data management & communication unit. For neuron computation, the NN model and training/testing dataset are stored in the storage unit, and the processing unit leverages data management & communication unit to access data from storage unit and user program for target application.



Decode NN Model

i	0	1	2	3	4
MV[i]	1	0	1	1	0
ACC[i]	1	1	2	3	3
mux out	0.1	0.1	0.2	0.3	0.3
NW[i]	0.1	0	0.2	0.3	0

Fig. 7. Example of NN model reduction and decoding.

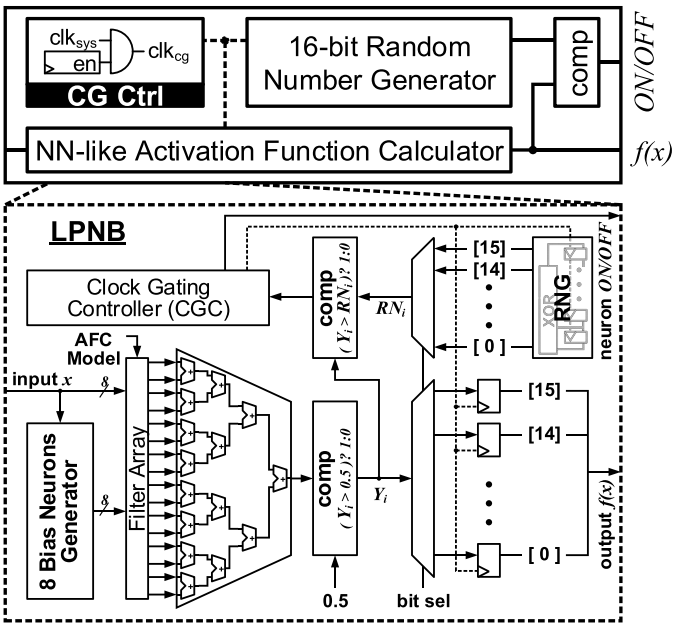


Fig. 8. Architecture of the proposed LPNB.

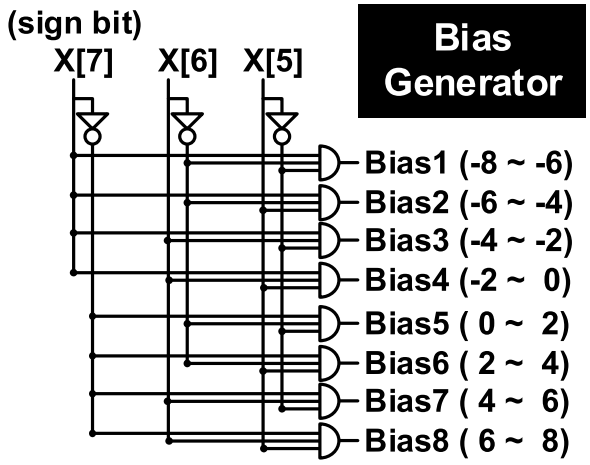


Fig. 9. Architecture of the bias neurons generator.

In addition, the $\text{Neu}_{\text{V}}^{1}$ and $\text{Neu}_{\text{H}}^{1}$ memory blocks are disabled during the inference mode for further power saving, and each building block will be described in the following.

A. Neural Network Model Reduction

For neuron computation, a huge storage space is required to store the weight matrix which is the model of RBM-based NN, as shown in (1). For example, an NN with 4k/4k visible/hidden neurons and 16-bit data representation per neuron connection weight requires 32 MB data storage space to store this model

$$\text{Storage Space of Model} = N_V * N_H * N_{\text{Bit}} \tag{1}$$

where

N_V number of neurons in visible layer;
N_H number of neurons in hidden layer;
N_{Bit} number of bits for neuron weight representation (16-bit per neuron weight in our system).

Due to the hardware resource limitation, the proposed system uses an external memory as storage unit to store NN models and training/testing dataset. Moreover, considering the performance of data accesses from the external memory, the burst operation mode is enabled to access 512-bit data in each read/write operation. And 16-bit data format is applied to each neuron connection weight in our system, 32 neuron connection weights are accessed from/to the external memory in each read/write operation, as shown in Fig. 5. And an external memory controller is implemented on field-programmable gate array (FPGA) to access data from DDR3 memory. To further reduce the data accessing time from external storage unit, a model reduction method is proposed to efficiently load NN model, as shown in Fig. 6.

By adopting this solution, the model parameters are converted to a sparse data in the model reduction module. Compared to the threshold set by user, the neuron weights are truncated to 0 when the absolute value is smaller than the threshold, and only non-zero neuron weights (NZNWs) are stored in external memory and read from storage unit to computation unit to reduce memory bandwidth. To decode the reduced NN model for neuron computation, 32 coding units (CUs) load NZNWs from local buffer and reconstruct NWs according to the mapping table which records the structure of the sparse NN model. In the experiment, an RBM-based NN model is trained and consisted of 794 neurons (28×28 pixel image for input and 10 labels for classification) and 1k neurons in the visible layer and hidden layer, respectively. According to the experimental results with the Modified National Institute of Standards and Technology (MNIST) handwritten digit database, 90% neuron connection weights are truncated to 0 when the threshold is set to 0.04. Therefore, 90% external memory bandwidth can be saved with <1% classification accuracy drop in this case. Moreover, 32 NWs can be

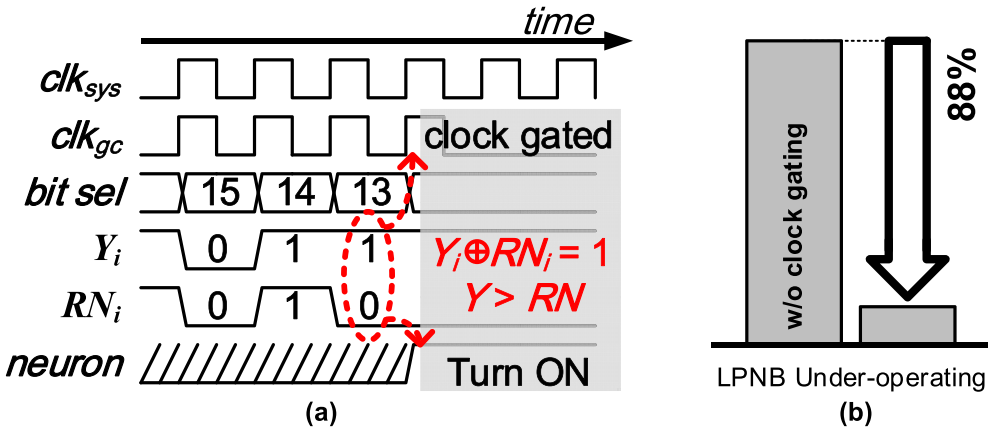


Fig. 10. (a) Timing diagram. (b) Experimental result of dynamic clock gating mechanism in the proposed LPNB.

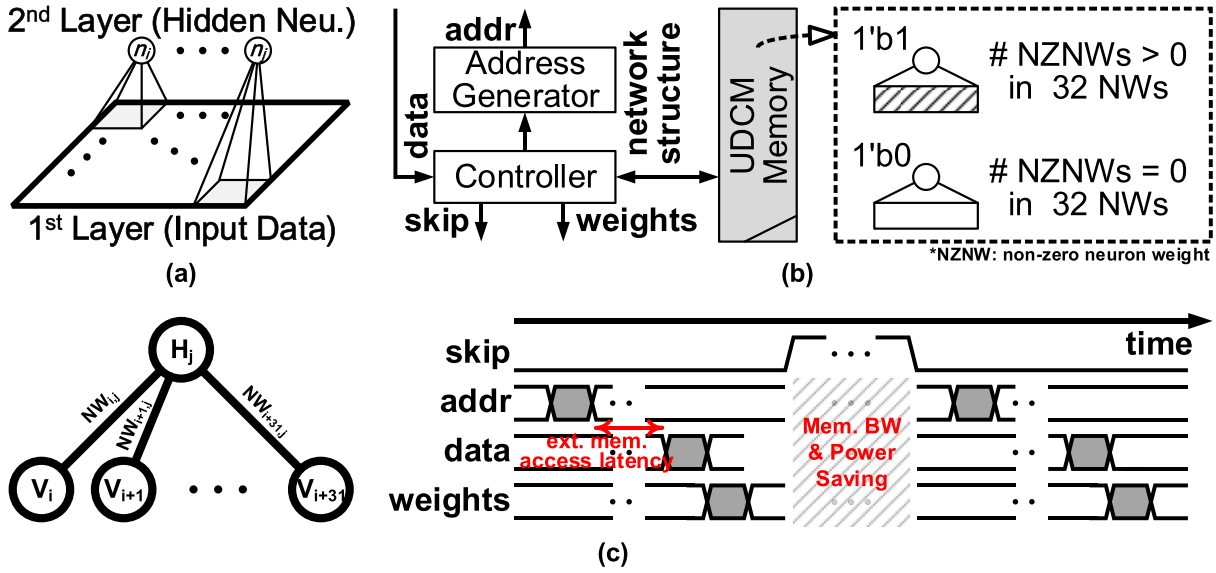


Fig. 11. (a) Region-based illustration. (b) Block diagram. (c) Timing diagram of the UDCM.

decoded in 1 cycle without complicated decoding processing, and an example of the proposed NN model reduction and decoding is shown in Fig. 7.

B. Low-Power Neuron Binarizer

Fig. 8 shows the proposed LPNB to support both sigmoid and hyperbolic tangent (tanh) non-linear functions and clock gating mechanism to disable clock signal dynamically. To implement non-linear activation functions, look-up table (LUT) [7], [9] and approximation method with polynomial function [13] are commonly used in hardware design. However, the LUT method requires large memory blocks to store values sampled from the target function. And multipliers cause a long computation latency in the approximation method.

In our previous work [14], a hardware-efficient sigmoid function calculator was proposed with an NN-like architecture. By adopting the NN-like architecture, the hardware cost can be reduced compared to the LUT and approximation method. To further reduce the computational complexity and hardware cost, an improved NN-like AFC is proposed to support both sigmoid and tanh non-linear functions for neuron computation in the proposed LPNB, and the AFC model is pre-trained with traditional feedforward and error-backpropagation approach to support both sigmoid and tanh functions. Considering the accuracy of calculated sigmoid and tanh non-linear functions, 3 MSB bits of the input x are applied to generate 8 bias neurons in the first stage, as shown in Fig. 9. Then, the filter array outputs the products of 16 bits, 8 bits from input x, and 8 bits from bias neurons generator, multiplied by the AFC model in the second stage. Since the data format of input 16 bits in the filter array is binary, the multipliers can be replaced by 16 AND operators to further save the computational complexity and hardware cost. In the third stage, the four-layer adder tree sums up all filtered results, and the bit value is determined by the comparator and assigned to the correlative register of output $f(x)$.

Furthermore, a dynamic clock gating mechanism is designed to further save power consumption in learning process, as shown in Fig. 10. Since the output neuron is turned

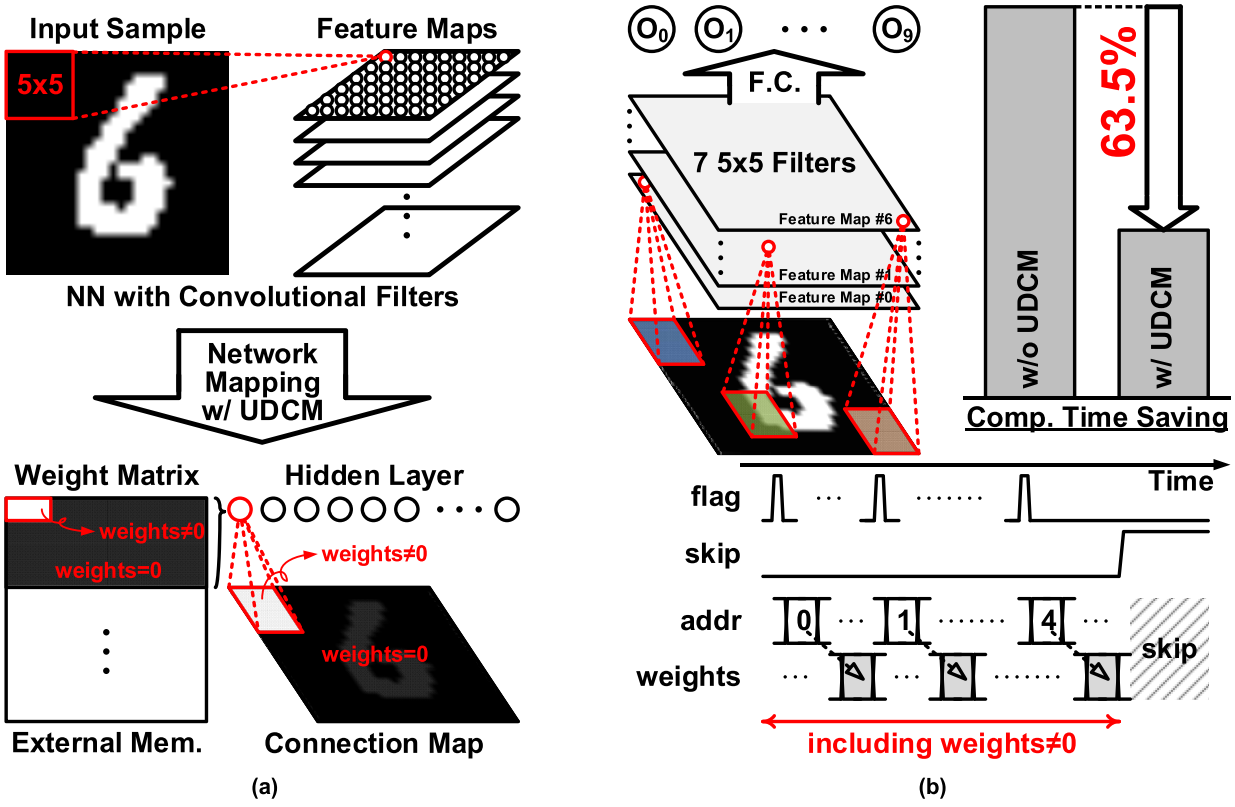


Fig. 12. Example of inference with a convolutional NN. (a) Network translation by the proposed UDCM. (b) Experimental result of performance improvement.

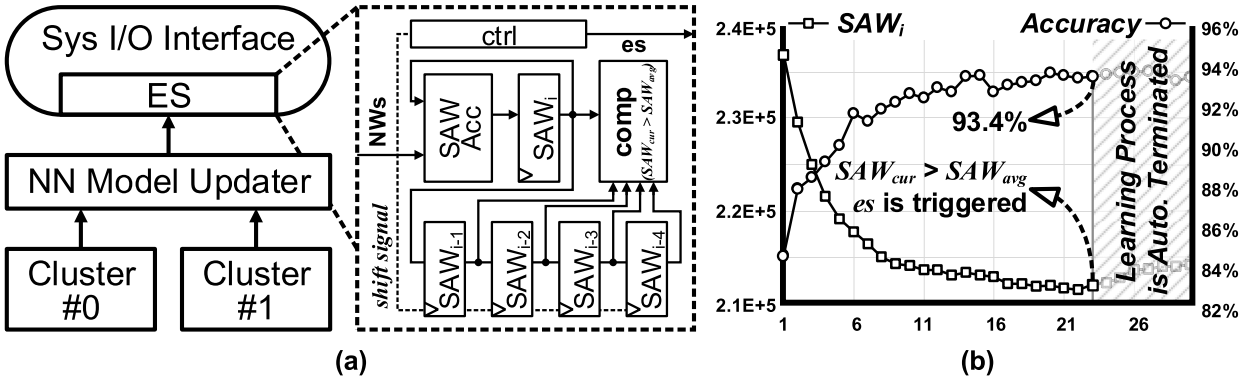


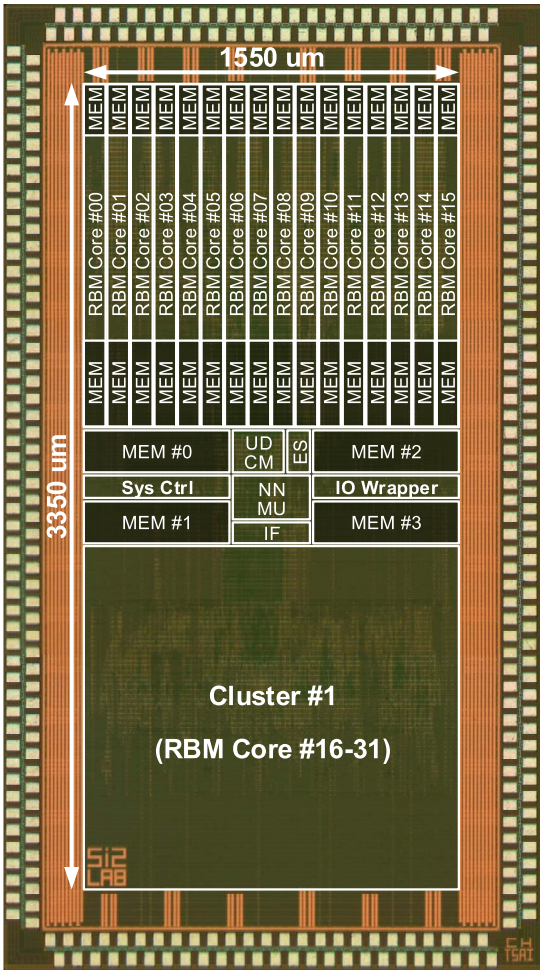
Fig. 13. (a) Architecture and (b) experimental result of the proposed ES mechanism.

ON or OFF based on the comparison result with an RN, the LPNB performs bit-level comparison simultaneously from MSB to LSB of the calculated non-linear function. When the calculated bit Y_i and the related bit of RN RN_i are exclusive, the clock signal is gated to achieve low-power design. According to the experimental result, 88% clock signal can be gated during model learning to further save power consumption.

C. User-Defined Connection Map

Since the handcraft feature extractors are usually designed to extract specific patterns in local regions, these filters are only connected to the correlative area, and the neuron weights are 0 in non-connected regions, as shown in Fig. 11(a). To efficiently and flexibly support non-fully connected and diverse NN structures, an UDCM module is implemented in the proposed system, as shown in Fig. 11(b). Since 32 neuron weights are loaded in each read operation, the 1-bit flag signal equals 0 when 32 continuous neuron weights are all 0, and neuron computation can be skipped to further save both external memory bandwidth and power consumption. If the flag signal equals 1, the address generator calculates the memory address to load 512-bits data from external memory, then 32 neuron weights are sent to RBM cores to perform neuron computation, as shown in Fig. 11(c).

Fig. 12 shows an example for supporting convolutional NN application by the proposed RBM processor, 7 5×5 convolutional filters detect specific features from the input 28×28 pixel image, and a three-layer convolutional NN is translated to an RBM which consists of 794 visible neurons and 4032 hidden neurons for neuron computation. According



RBM-P Chip Summary	
Technology	UMC 65nm 1P10M CMOS
Chip Size	4.0*mm* x 2.2*mm*
Gate Count	2.2M Gates
On-chip Memory	128kB SRAM
Working Frequency	210 MHz @ 1.2V
Machine Learning Spec.	
Applied Algorithm	Neural Network
Model Training	Restricted Boltzmann Machine
Number of Cores	32 RBM Cores in 2 Clusters
Data Dimension	4096 Neurons / Layer
Classification	128 Candidates / sample
On-chip Learning Performance @CD2	
Throughput	7.53G NWs / sec (GNWPS)
Energy Efficiency	41.31 pJ / NW
On-chip Inference Performance	
Throughput	11.63G NWs / sec (GNWPS)
Energy Efficiency	26.74 pJ / NW

Fig. 14. Die photograph and chip specification.

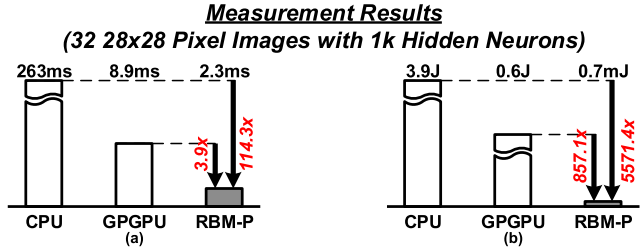


Fig. 15. Measurement results and comparison.

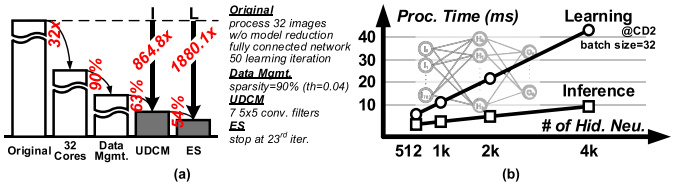


Fig. 16. (a) Performance improvement. (b) Execution time in learning and inference.

to the experimental result, 63.5% computation time can be saved by adopting the proposed UDCM module compared to using traditional fully connected NN structure for neuron computation.

D. Early Stopping

In the CD with AGS model learning algorithm, the main training loop is terminated until the number of iterations equals system parameters configured by user or the model are convergent. Conventionally, a cross validation dataset is required to periodically verify the trained NN model to detect the model is convergent or not. However, it is difficult to check the model quality in the unsupervised learning due to no label information for validation.

In the proposed system, a sum of absolute weights (SAWs) index is designed to sum up all absolute value of neuron weights when the NN model is updated and written back to external memory during the learning process, as shown in (2). Generally speaking, the SAW value is increasing when overfitting occurs during NN model learning phase. Therefore, an ES mechanism is proposed to automatically terminate learning process to save computation time.

Fig. 13(b) shows the experimental result of the MNIST handwritten digits database with 30 training iterations, and the applied RBM structure is 28×28 pixel image for input, 1k hidden neurons, and 10 label neurons for classification.

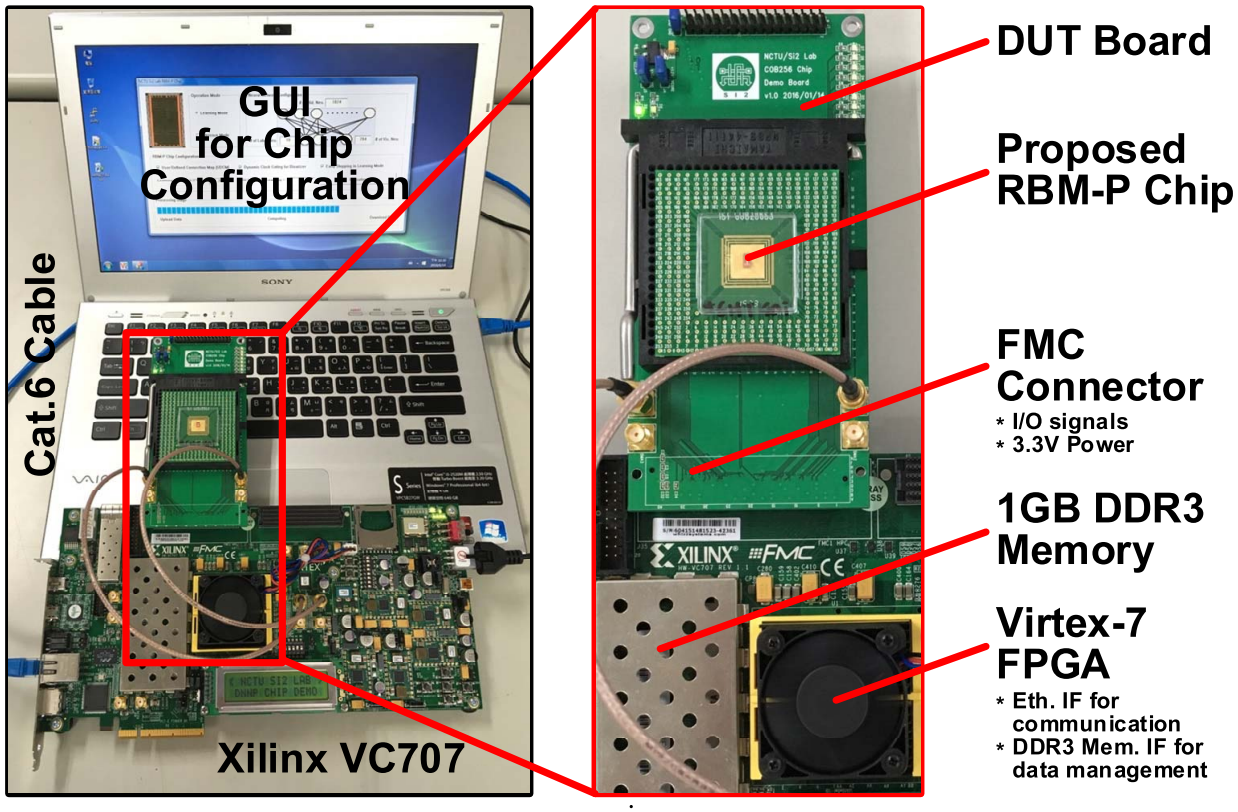


Fig. 17. FPGA-based platform for system evaluation.

The accuracy of the trained model is nearly saturated after 20 iterations, and the SAW value is increasing after the 21st iteration. Therefore, the ES detector and criterion are proposed to continuously monitor the SAW value during model learning, as shown in (3) and (4). When the current SAW value is larger than the average of previous 4 SAWs, the ES signal is triggered to automatically terminate the NN model learning process. According to the experimental result, the ES signal is triggered at the 23rd iteration with 93.4% classification accuracy, and the learning process can be terminated to save both learning time and power consumption. Furthermore, the NN model training with the fixed learning iteration set by user is permitted when the ES mechanism is disabled. And Fig. 13(a) shows the architecture of the proposed ES mechanism which is embedded in the system I/O interface to calculate SAW and perform ES checking

$$\text{SAW} = \sum |\text{neuron connection weight}| \tag{2}$$

$$\text{SAW}_{\text{avg}} = \frac{1}{4}\sum_{i=1}^{4} \text{SAW}_{\text{cur}-i} \tag{3}$$

$$\text{ES Signal} = \begin{cases} 1, & \text{SAW}_{\text{cur}} > \text{SAW}_{\text{avg}} \\ 0, & \text{others.} \end{cases} \tag{4}$$

IV. Implementation Results

A. Implementation Results

Fig. 14 shows the chip photograph and specification summary. The proposed RBM-P chip is fabricated in UMC 65nm CMOS technology and costs 2.2 M gates and 128 kB internal SRAM with 8.8 mm^2 area. This chip integrates 32 RBM cores in two clusters to support both learning and inference operations with the NN structure of maximal 4k neurons per layer and 128 candidates per sample for machine learning applications. Operated at 1.2 V and 210 MHz, the proposed system achieves 7.53G and 11.63G NWs per second performance with 41.31 and 26.74 pJ per NW energy efficiency for learning and inference, respectively. Applying the proposed RBM-P chip, ~2500 32×32 pixel images are classified per second with 4k hidden neurons and 32 candidates for intelligent IoT devices.

B. Measurement Results

Fig. 15(a) and (b) shows the measurement and comparison results based on our proposal. Compared to the single- and multi-thread software implementation on CPU and GPGPU to process 32 28×28 pixel images for handwritten digit classification with MNIST database, the proposed RBM-P chip achieves 114.3× and 3.9× faster processing time, respectively. And our chip consumes 0.7 mJ to classify 32 28×28 images with 1k hidden neurons, which reaches over 5000× and 800× of energy reduction compared to the CPU and GPGPU, respectively.

In our system, the performance can be roughly evaluated by (5). 32 RBM cores are implemented inside the proposed RBM-P chip, and the system performance can be linearly improved by integrating more proposed RBM cores to increase M_{cores} to perform learning and inference operations in parallel.

TABLE I
COMPARISON WITH RELATED RBM PROCESSORS

	[9]	[15]	[7]	Ours
Implementation	FPGA	FPGA	65nm	65nm
Model Storage	Block Mem	SDRAM	SDRAM	SDRAM
Gate Count	N/A	N/A	3.75 M	2.20 M
Chip Size	N/A	N/A	2.5x4.0 mm^2	2.2x4.0 mm^2
Power Consumption	N/A	N/A	213.1 mW	310.0 mW
Area Efficiency	N/A	N/A	41.13 GOPS/mm^2	51.16 GOPS/mm^2
Power Efficiency	N/A	N/A	1.93 TOPS/W	1.45 TOPS/W
RBM Size / Core	128/layer	256/layer	1024/layer	4096/layer
Architecture of AFC	LUT	LUT	LUT	NN-like
Auto. Stopping for Learning	N/A	N/A	N/A	Y
Supported Network Structure	FC	FC	FC Conv*	FC User-defined

FC: Fully Connected Network Structure Conv*: Fixed Convolution Network Structure

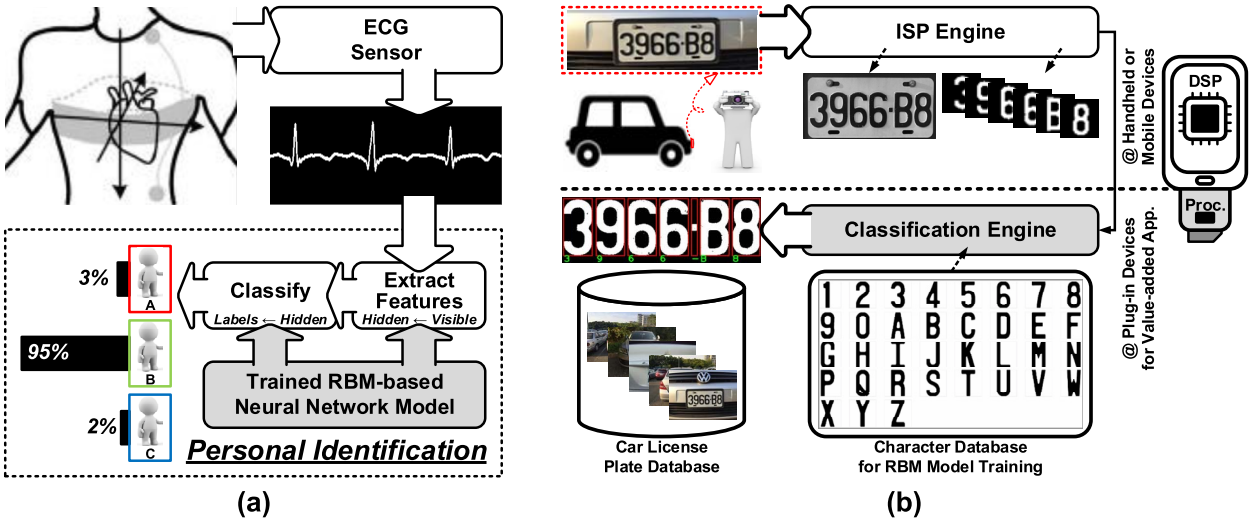


Fig. 18. (a) ECG-based personal identification. (b) Car License plate recognition.

To efficiently load NN model from external memory, the data accessing time T_{LM} can be reduced by the proposed data management with NN model reduction for external memory bandwidth saving. To efficiently and flexibly support diverse NN structures, the proposed UDCM module is designed to reduce N for skipping the unnecessary neuron computation to save both T_{LM} and T_{NC}. In addition, the proposed ES mechanism is able to terminate the learning procedure automatically to save both computation time and power consumption, and the proposed LPNB with dynamic clock gating and area-efficient NN-like AFC is designed to reduce both power consumption and hardware cost. Overall, the proposed RBM-P chip, respectively, achieves 864.8× and 1880.1× processing time reduction in inference and learning, as shown in Fig. 16(a). And the detailed results of execution time with 512–4k hidden neurons are shown in Fig. 16(b). Moreover, a comparison between the proposed RBM-P chip and related RBM processors is shown in Table I.

$$\text{Processing Time} \propto \frac{S_{\mathrm{Total}}}{M_{\mathrm{cores}}} * [N * (T_{\mathrm{LM}} + T_{\mathrm{NC}})] \quad (5)$$

where

S_{Total}	number of samples;
M_{cores}	number of proposed RBM-Cs;
N	number of neurons to perform computation;
T_{LM}	processing time to load neural network model;
T_{NC}	processing time to perform neuron computation.

C. System Platform

Fig. 17 shows the proposed FPGA-based prototype for system evaluation. A dedicated device-under-test board is designed to leverage the peripheral resources including onboard 1-GB DDR3 external memory for data storage and 1-Gbps Ethernet interface for communication on the Xilinx VC707 FPGA development board. To control our system, a graphical user interface is designed on the host NB/PC, and user can upload/download data and configure the proposed RBM-P chip through cat.6 Ethernet cable. Moreover, two applications are shown in Fig. 18. For healthcare applications [16], the electrocardiogram (ECG) signal is applied

to personal biomedical information, and an RBM-based NN model with 4k hidden neurons is trained by the proposed RBM-P chip for personal identification to protect private data. In the application phase, the ECG signal is sampled in 16-bit resolution, and each measured ECG signal consists of maximal 248 sample points for personal identification. For this example, the input ECG signal belongs to person B with 95% probability, as shown in Fig. 18(a). And Fig. 18(b) shows another application, the proposed processor is embedded in a plug-in device for the value-added services. In the application phase, each character is segmented to 32×32 pixel image, and an RBM-based NN model with 1k hidden neurons is applied to car license plate recognition. Then, the recognized car license can be further analyzed with a database for surveillance applications.

V. Conclusion

In this paper, an RBM-P supporting both on-chip learning and inference is designed for machine learning-based IoT applications. To save external memory bandwidth, an NN model reduction approach is proposed to efficiently load NN models from external memory for neuron computation. A LPNB with dynamic clock gating and area-efficient NN-like AFC is proposed to save power consumption. And the proposed UDCM is designed to efficiently and flexibly support diverse network structures to save both computation time and external memory bandwidth. In addition, an ES mechanism is proposed to save computation time in learning process. Implemented in 65nm CMOS technology, the proposed RBM-P chip costs 2.2 M gates and 128 kB SRAM with 8.8 mm^2 area to integrate 32 RBM cores with maximal 4k neurons per layer and 128 candidates per sample. Operated at 1.2 V and 210 MHz, this chip, respectively, achieves 7.53G and 11.63G NWs per second with 41.3 and 26.7 pJ per NW for learning and inference, making it very suitable for intelligent IoT applications.

Acknowledgment

The authors would like to thank Prof. C.-C. Chung for chip design environment setting and all members of NCTU/Si2 Lab for fruitful discussions.

References

[1] H. Shikano, K. Ito, K. Fujita, and T. Shibata, "A real-time learning processor based on K-means algorithm with automatic seeds generation," in *Proc. Int. Symp. Syst.-Chip*, Nov. 2007, pp. 1–4.

[2] H. Hussain, K. Benkrid, C. Hong, and H. Seker, "An adaptive FPGA implementation of multi-core K-nearest neighbour ensemble classifier using dynamic partial reconfiguration," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug. 2012, pp. 627–630.

[3] M. A. B. Altaf and J. Yoo, "A 1.52 uJ/classification patient-specific seizure classification processor using linear SVM," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2013, pp. 849–852.

[4] C.-H. Tsai, H.-H. Lee, W.-J. Yu, and C.-Y. Lee, "A 2 GOPS quad-mean shift processor with early termination for machine learning applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2014, pp. 157–160.

[5] V. Gokhale, J. Jin, A. Dundar, B. Martini, and E. Culurciello, "A 240 G-ops/s mobile coprocessor for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 696–701.

[6] J. K. Kim, P. Knag, T. Chen, and Z. Zhang, "A 640 M pixel/s 3.65 mW sparse event-driven neuromorphic object recognition processor with on-chip learning," in *Proc. Symp. VLSI Circuits*, Jun. 2015, pp. 50–51.

[7] S. Park, K. Bong, D. Shin, J. Lee, S. Choi, and H.-J. Yoo, "A 1.93 TOPS/W scalable deep learning/inference processor with tetra-parallel MIMD architecture for big-data applications," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2015, pp. 80–81.

[8] J. K. Kim, P. Knag, T. Chen, and Z. Zhang, "A 6.67 mW sparse coding ASIC enabling on-chip learning and inference," in *Proc. Symp. VLSI Circuits*, Jun. 2014, pp. 1–2.

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

[10] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 5. Apr. 2009, pp. 448–455.

[11] A. Fischer and C. Igel, "Training restricted Boltzmann machines: An introduction," *Pattern Recognit.*, vol. 47, no. 1, pp. 25–39, 2014.

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[13] M. Al-Nsour and H. S. Abdel-Aty-Zohdy, "Implementation of programmable digital sigmoid function circuit for neuro-computing," in *Proc. IEEE Midwest Symp. Circuits Syst.*, Aug. 1998, pp. 571–574.

[14] C.-H. Tsai, Y.-T. Chih, W. H. Wong, and C.-Y. Lee, "A hardware-efficient sigmoid function with adjustable precision for a neural network system," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 11, pp. 1073–1077, Nov. 2015.

[15] S. K. Kim, L. C. McAfee, P. L. McMahon, and K. Olukotun, "A highly scalable restricted Boltzmann machine FPGA implementation," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug./Sep. 2009, pp. 367–372.

[16] S.-Y. Hsu, Y. Ho, P.-Y. Chang, C. Su, and C.-Y. Lee, "A 48.6-to-105.2 μW machine learning assisted cardiac sensor SoC for mobile healthcare applications," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 801–811, Apr. 2014.

Chang-Hung Tsai received the B.S. degree in computer science and information engineering from National Dong Hwa University, Hualien, Taiwan, in 2008, and the M.S. degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan, in 2010. He received the Ph.D. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2016.

He is currently with MediaTek Inc., Hsinchu, Taiwan. His research interests include algorithms, architectures, and low-power hardware designs for source coding and machine learning systems.

Wan-Ju Yu received the B.S. and M.S. degrees in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2014 and 2016, respectively.

She is currently with MediaTek Inc., Hsinchu. Her research interests include algorithms and low-power hardware designs for machine learning systems.

Wing Hung Wong received the B.A. degree in statistics from the University of California, Berkeley, CA, USA, in 1976, and the Ph.D. degree in statistics from the University of Wisconsin–Madison, Madison, WI, USA, in 1980.

He held teaching positions at the University of Chicago, Chicago, IL, USA, The Chinese University of Hong Kong, Hong Kong, The University of California, Los Angeles, CA, USA, and Harvard University, Cambridge, MA, USA, before joining Stanford University, Stanford, CA, USA, in 2004. He was a Chair with the Stanford Department of Statistics from 2009 to 2012. He is currently with the faculty of Stanford University, where he is a Professor of Statistics, Professor of Biomedical Data Science, and holder of the Stephen R. Pierce Family Goldman Sachs Professorship in Science and Human Health. His current research interests include mathematical statistics, where he clarified the large sample properties of sieve maximum likelihood estimates in general spaces; Bayesian statistics, where he introduced sampling-based algorithms into Bayesian computational inference; and computational biology, where he developed basic models and methods for the analysis of microarrays gene expression data and RNA sequencing data. Technologies from his group have led to the formation of companies in the domains of genomic data management, machine learning and predictive medicine.

Dr. Wong is a member of the National Academy of Sciences, the Academia Sinica, and the Academy of Sciences of Hong Kong. He was a recipient of the COPSS Presidents' Award in Statistics.

Chen-Yi Lee received the B.S. degree in electrical engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 1982, and the M.S. and Ph.D. degrees in electrical engineering from Katholieke University Leuven, Leuven, Belgium, in 1986 and 1990, respectively.

From 1986 to 1990, he was with IMEC/VSDM, Leuven, Belgium, focusing on the area of architecture synthesis for DSP. In 1991, he joined the Department of Electronics Engineering and from 2003 to 2006, he was the Chairman. During 2000–2003, he was the Director of the National CHIP Implementation Center, Hsinchu, Taiwan. During 2003–2005, he was the Coordinator of Microelectronics Program of Engineering Division, National Science Council (NSC), Taipei, Taiwan. He was the Dean of the Office of Research and Development, NCTU, from 2007 to 2010. He is currently the Co-Program Director of the National Program of Intelligent Electronics and a Professor of the Department of Electronics Engineering, NCTU. His current research interests include very large scale integration algorithms, architectures for high-throughput DSP applications, micro-sensing, low-power system-on-chip, and big data analysis.

Dr. Lee served as a Program Committee Member of the IEEE ISSCC from 2004 to 2006, DATE TPC Member from 2006 to 2007, IEEE ASSCC TPC Member from 2006 to 2014, IEEE VLSI Symposium JFE Program Committee Member from 2010 to 2014, IEEE TCAS-II Associate Editor from 2010 to 2011, and the Past-Chair of the Taipei Chapter of the IEEE Circuits and Systems Society. He received the Award of Outstanding on Technology Licensing in 2007 and 2008 from NSC, 2009 from the Ministry of Economic Affairs, and the Outstanding Research Award from NSC in 2009.