ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.15, No.2, 1–7

# A novel memristor-based restricted Boltzmann machine for contrastive divergence

**Yan Chen**[1], **Zhiqiang You**[2a)], **Yingjie Zhang**[2], **Jishun Kuang**[2], **and Jing Zhang**[1]

[1] *College of Electrical and Information Engineering, Hunan University*

[2] *College of Computer Science and Electronic Engineering, Hunan University*

a) *you@hnu.edu.cn*

**Abstract:** In this letter, we present a novel memristor-based restricted Boltzmann machine (RBM) system for training the brain-scale neural network applications. The proposed system delicately integrates the storage component of neuron outputs and the component of multiply-accumulate (MAC) in memory, allowed operating both of them in the same stage cycle and less memory access for the contrastive divergence (CD) training. Experimental results show that the proposed system delivers significantly 2770x speedup and less than 1% accuracy loss against the x86-CPU platform on RBM applications. On average, it achieves 2.3x faster performance and 2.1x better energy efficiency over recent state-of-the-art RBM training systems.

**Keywords:** contrastive divergence, memristor, restricted Boltzmann machine

**Classification:** Integrated circuits

## References

[1] G. E. Hinton and R. R. Salakhutdinov: "Reducing the dimensionality of data with neural networks," Science **313** (2006) 504 (DOI: 10.1126/science.1127647).

[2] A. M. Sheri, *et al.*: "Contrastive divergence for memristor-based restricted Boltzmann machine," Eng. Appl. Artif. Intell. **37** (2015) 336 (DOI: 10.1016/j.engappai.2014.09.013).

[3] H. Akinaga and H. Shima: "ReRAM technology; challenges and prospects," IEICE Electron. Express **9** (2012) 795 (DOI: 10.1587/elex.9.795).

[4] M. N. Bojnordi and E. Ipek: "Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning," HPCA (2016) 1 (DOI: 10.1109/HPCA.2016.7446049).

[5] A. Shafiee, *et al.*: "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," ISCA (2016) 14 (DOI: 10.1145/3007787.3001139).

[6] P. Chi, *et al.*: "PRIME: A novel processing-in-memory architecture for neural network computation in reram-based main memory," ISCA (2016) 27 (DOI: 10.1145/3007787.3001140).



DOI: 10.1587/elex.15.20171062
Received October 19, 2017
Accepted December 19, 2017
Publicized January 11, 2018
Copyedited January 25, 2018

[7] M. Suri, *et al.*: "Neuromorphic hybrid RRAM-CMOS RBM architecture," NVMTS (2015) 1 (DOI: 10.1109/NVMTS.2015.7457484).
[8] G. M. Ribeiro and M. D. Pickett: U.S. Patent 8780610 (2014).
[9] L. Gao, *et al.*: "Programming protocol optimization for analog weight tuning in resistive memories," IEEE Electron Device Lett. **36** (2015) 1157 (DOI: 10.1109/LED.2015.2481819).
[10] X. Dong, *et al.*: in *Emerging Memory Technologies*, ed. Y. Xie (Springer, New York, 2014) 15 (DOI: 10.1007/978-1-4419-9551-3_2).
[11] C. Yakopcic, *et al.*: "A memristor device model," IEEE Electron Device Lett. **32** (2011) 1436 (DOI: 10.1109/LED.2011.2163292).
[12] S. Han, *et al.*: "EIE: Efficient inference engine on compressed deep neural network," ISCA (2016) 243 (DOI: 10.1145/3007787.3001163).

## 1 Introduction

Deep neural networks (DNNs), achieved many significant breakthroughs in deep learning, are widely used in recognition, mining, and synthesis. Restricted Boltzmann machines (RBMs) [1], pre-trained the neural networks (NNs) in an effective and feasible layer-by-layer manner, are massively applied to tune the NN parameters into an optimized region. One of the most famous training approaches for RBMs is the stochastic approximation gradient called contrastive divergence (CD) [2]. With the advent of superior DNNs, where the parameter features even up to brain-scale and the continues enormous memory access overhead for big-data workloads, the design of efficient RBM accelerating systems, which enable processing huge amount of computation in memory for the brain scale of training and inference, has drawn great attention [3].

Recently, memristor-based RBM accelerators [2, 4, 5, 6, 7], guaranteed the non-volatile storage of NN parameters and enabled operating the multiply and accumulator (MAC) computation in memory, are potential to beyond the scale limitation of conventional CMOS technology for brain-scale NN applications. However, the introduced huge CMOS latches in [4] for the storage of neuron outputs will result in intolerable area overhead when scaling up the system for brain level accelerations. [5, 6] allow only the inference of DNN applications and are bounded by the disability of training. Meanwhile, [2, 7] accomplish CD procedure with long stage cycles due to the conditional provision in CD, which incurs a huge amount of memory access overhead. Moreover, [2] implements a CD-like accelerating system by altering the original CD to simplify its hardware implementation. This, in return, dramatically limits their ability in RBMs applications. In brief, all the works above cannot efficiently train and inference the brain-scale RBMs.

In this brief, we build an efficient memristor-based RBM system for training and inference the brain-scale NNs. The system ingeniously assembles the storage component of neuron outputs and the component of MAC computations through a memristor-based latch (m-latch), allowed processing them in the same stage cycle and avoided frequently reloading NN parameters.


DOI: 10.1587/elex.15.20171062
Received October 19, 2017
Accepted December 19, 2017
Publicized January 11, 2018
Copyedited January 25, 2018

## 2 Preliminaries

The RBM model consisted of two layer NNs with each "hidden" ($h$) neuron connected to all "visible" ($v$) neurons, is defined via an energy function as

$$E(v,h) = -\sum_i c_i v_i - \sum_j d_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{1}$$

where $c_i$ and $d_j$ are bias weights. $w_{ij}$ is a synapse connected between $v_i$ and $h_j$ neurons. Notably, the conditional probabilities of $v_i$ and $h_j$ are given by

$$P(h_j = 1|v,\theta) = \sigma\left(d_j + \sum_i v_i w_{ij}\right); \quad P(v_i = 1|h,\theta) = \sigma\left(c_i + \sum_j h_j w_{ij}\right) \tag{2}$$

where $\sigma(\cdot)$ is an activation function such as *sigmoid*, and $\theta = \{c_i, d_j, w_{ij}\}$.

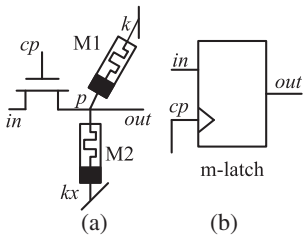
CD algorithm provides a feasible way to train the RBM model by estimating the gradient of $w_{ij}$, $c_i$, and $d_j$ in Eq. (1) as follows

$$\begin{cases} \Delta w_{ij} = \varepsilon(\langle v_i h_j\rangle_{in} - \langle v_i h_j\rangle_{rec}); \\ \Delta c_i = \varepsilon(\langle v_i\rangle_{in} - \langle v_i\rangle_{rec}); \quad \Delta d_j = \varepsilon(\langle h_j\rangle_{in} - \langle h_j\rangle_{rec}) \end{cases} \tag{3}$$

where $\varepsilon$ denotes a learning rate. $\langle\cdot\rangle_{in}$ and $\langle\cdot\rangle_{rec}$ are the expectations of the input neurons and reconstruction neurons respectively.

Nevertheless, CD has to generate $\langle v_i\rangle_{in}$ and $\langle h_j\rangle_{in}$ of input neurons, and $\langle v_i\rangle_{rec}$ and $\langle h_j\rangle_{rec}$ of reconstruction neurons in sequence. This derives from the conditional provision of $v_i$ and $h_j$ on Eq. (2). That is, a long computation path and huge memory access are taken in the CD procedure. When we could process the dependent and sequential computation in memory simultaneously and avoid frequently memory access, training and inference the brain-scale RBMs will be completed in a more practical and efficient way.

Fig. 1 illustrates the perspective view of a memristor-based latch (named as m-latch) [8]. m-latch, consisted of two memristors M1 and M2 with either the low or high resistance state (LRS or HRS), functions as a latch for storing the neuron output. Specifically, the potential at the "*out*" wire of m-latch denotes not only the logic states ("1" or "0") of the read operation, and also the results of the write operation ("0" potential at "*out*" wire corresponds to M1:HRS and M2:LRS, and "1" relates to M1:LRS and M2:HRS). In the proposed system, m-latch not only stores the neuron outputs, the potential at "*out*" wire of the m-latch also be reused to trigger both the storage of neurons and MAC computation components.




| | input | | | | memristor | | "out"wire | output |
|---|---|---|---|---|---|---|---|---|
| | *cp* | *in* | *k* | *kx* | M1 | M2 | | |
| read | 0 | 0 | 1 | 0 | HRS | LRS | 0 | 0 |
| | | $2V_{SET}$ | 1 | 0 | LRS | HRS | 1 | 1 |
| write | 1 | 0 | $V_{RST}$ | $V_{SET}$ | HRS | LRS | 0 | - |
| | | $2V_{SET}$ | $V_{RST}$ | $V_{SET}$ | LRS | HRS | 1 | - |

(c)

**Fig. 1.** M-latch and its operations. (a) Schematic. (b) Symbol. (c) Read and write operations.


DOI: 10.1587/elex.15.20171062
Received October 19, 2017
Accepted December 19, 2017
Publicized January 11, 2018
Copyedited January 25, 2018

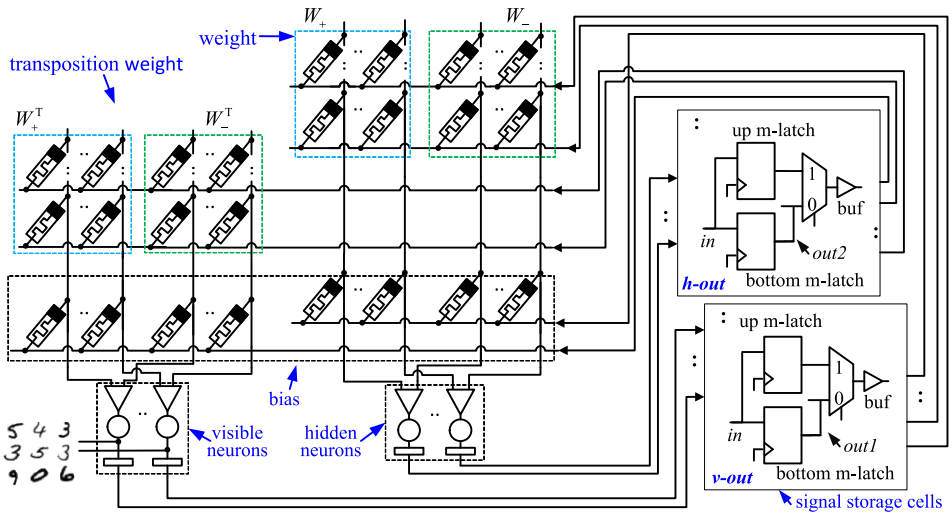



**Fig. 2.** Profile of the memristor-based RBM system.

## 3 Memristor-based RBM pre-training system

Fig. 2 depicts the architecture of the proposed memristor-based RBM system. It consists of two neuron components (visible and hidden), two signal storage cell components (*v-out* and *h-out*), a bias component, and the twos weight synapse ($W_+$ and $W_-$, referring to positive and negative weights respectively) and transposition weight synapse ($W_+^T$ and $W_-^T$) components. The transposition synapse is a copy of the weight synapse. Both of them perform the MAC computation with the visible and hidden neurons as inputs respectively. The signal storage cell consisted of the bottom and up m-latches, stores outputs of neurons and reconstructed neurons respectively.

Fig. 3 illustrates how the proposed system at Fig. 2 generate the sequenced $\langle v \rangle$ and $\langle h \rangle$ of Eq. (3) in the same stage cycle. The potential at "*out 1*" of m-latch plays a key role to integrate both the storage of neuron outputs and the followed MAC computing. First, visible neuron component outputs the $\langle v \rangle_{in}$ results to the bottom m-latches of the *v-out* component. Simultaneously, the followed MAC computations on $W_+$ and $W_-$ can be triggered when storing neuron outputs at *v-out*. This is because that the potential of "*out 1*" always equals to them of "*a*". That is, the cascaded hidden neuron component can directly output the results $\langle h \rangle_{in}$ to the bottom m-latches of the *h-out* component. Therefore, $\langle v \rangle_{in}$ and $\langle h \rangle_{in}$ can directly be generated in the same cycle and avoid reloading $\langle v \rangle_{in}$ during computation. Similarly, $\langle v \rangle_{rec}$ and $\langle h \rangle_{rec}$ in Eq. (3) are respectively generated and stored back to the up m-latches of both *v-out* and *h-out* at the same cycle.

Based on the above procedure, the proposed system is able to compute the CD procedure with fewer step cycles and avoid part of the read neuron operations. The CD procedure of the proposed RBM architecture is summarized in Algorithm 1. The computation of $\langle v \rangle$ and $\langle h \rangle$ are completed in the same stage cycle, shown in both stage 2 and 3. That is, the proposed RBM system can perform the CD procedure in a more efficient way.


DOI: 10.1587/elex.15.20171062
Received October 19, 2017
Accepted December 19, 2017
Publicized January 11, 2018
Copyedited January 25, 2018

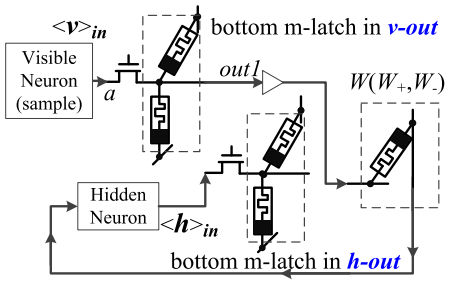



**Fig. 3.** Illustration of computing $\langle v\rangle_{in}$ and $\langle h\rangle_{in}$ in the same cycle. The equivalent potential of "*out 1*" from the bottom m-latch and "*a*" allows storing the $\langle v\rangle_{in}$ and $\langle h\rangle_{in}$ within one cycle.

**Algorithm 1** Schedule of the memristor-based CD.

1: stage 1: Initialization.
2: stage 2: Parallel computing $\langle v\rangle_{in}$ and $\langle h\rangle_{in}$.
3: stage 3: Parallel computing $\langle v\rangle_{rec}$ and $\langle v\rangle_{rec}$.
4: stage 4: Parameters updating.
5: $W = W + \varepsilon(\langle v\rangle_{in}^{T}\langle h\rangle_{in} - \langle v\rangle_{rec}^{T}\langle h\rangle_{rec})$; $c = c + \varepsilon(\langle v\rangle_{in} - \langle v\rangle_{rec})$;
6: $W^{T} = W^{T} + \varepsilon(\langle h\rangle_{in}^{T}\langle v\rangle_{in} - \langle h\rangle_{rec}^{T}\langle v\rangle_{rec})$; $d = d + \varepsilon(\langle h\rangle_{in} - \langle h\rangle_{rec})$;

## 4 Experiments and results

### 4.1 Experiments setup

**Baselines for comparison**. **CPU**: the x86-CPU baseline is the Intel Xeon E5-2620 v3 with 15 M cache and 2.40 GHz. **Prior RBM training systems**: The compared state-of-the-art memristor-based RBM systems include the hybrid RRAM-CMOS RBM (Hyb) [7], the two memristors mode (Two-m) [2], and the memory-centric accelerator (Mem-cen) [4].

**Performance**: The execution time of the CPU platform for the standard CD procedure [1] is measured with an NN of the 784-1000 size. Meanwhile, the performance of the proposed system is estimated based on the tuning time of the physical memristor device [9] under the same NN size. **Energy and area**: Since the storage of the neuron output and synapse dominating in the memristor-based system, we focus on the energy and area of them for all RBM systems, which are evaluated under a 784-500 NN size, 8-bit scale data. Specifically, the memristor components are measured with the circuit-level simulator Nvsim [10] based on a TiO2 memristor model [11] in 45-nm technology, while the CMOS components are synthesized by the Design Compiler in 45-nm technology. Detailed parameters of the memristors in the proposed RBM system are shown in Table I.

**Accuracy**: In the proposed system, we use different applications for the accuracy comparisons with the standard CD procedure platform [1]. The auto-encoder and classification applications are conducted for training, cross-validation (cro.-val.), and testing under two well-known workloads, including the MNIST dataset (60,000 training and 10,000 testing images) and the ImageNet dataset (randomly selected 1,600 training and 400 testing images). Details of the training characteristics are shown in Table II.


DOI: 10.1587/elex.15.20171062
Received October 19, 2017
Accepted December 19, 2017
Publicized January 11, 2018
Copyedited January 25, 2018

**Table I.** Memristor parameters for the latch and synapse

| Params | $V_p$ | $V_n$ | $A_p$ | $A_n$ | $x_p$ | $x_n$ | $\alpha_p$ | $\alpha_n$ | $a_1$ | $a_2$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Latch | 0.42 | 0.22 | 5E2 | 3E3 | 0.7 | 0.8 | 4 | 24 | 2.3E-3 | 3.8E-3 | 1 |
| Synapse | 1.1 | 1.1 | 3E3 | 3E3 | 0.7 | 0.8 | 4 | 24 | 2.3E-3 | 2.3E-3 | 0.7 |

**Table II.** Training parameters on auto-encoder and classification applications

| | auto-encoder | | classification |
|---|---|---|---|
| Dataset | MNIST | ImageNet | MNIST |
| K-fold | 10 | 5 | 10 |
| NN size | 784-1000-500 -250-30 | 16384-700-1000 -500-1000-500 | 784-1000-500 -250-10 |
| Training epoch | 10 CD | 50 CD | 10 CD + 50 BP |
| CDs' learning rate | 0.1 | 0.001 | 0.1 |

### 4.2 Experimental results

**Performance and Energy Efficiency**. **a) Performance** The proposed system delivers a 2770x speedup against the CPU platform (x86-CPU needs 4.355 s, the proposed system takes $1.572 * 10^{-3}$ s). Compared to prior memristor-based RBM systems, we achieve a speedup of 2.3x on average, delivering 2x, 3.3x, and 1.7x higher performance than Hyb, Two-m, and Mem-cen respectively, shown in Table III. **b) Energy Efficiency** Table III also shows the energy efficiency of the recent state-of-the-art RBM systems. The proposed system delivers on average 2.1x energy efficiency with 2.7x, 1.1x, and 2.4x better than Hyb, Two-m, and Mem-cen respectively. The benefits of performance and energy efficiency of the proposed system derive from the reduced cycles and memory accesses on the CD procedure.

**Accuracy**. Table IV shows the accuracy results of the auto-encoder and classification applications compared to the standard CD. The auto-encoder is evaluated in term of the average squared reconstruction error (SRE), which is also used in [1]. The classification is measured in accuracy, which is a correct fraction of all the predictions.

Compared to the standard CD, on average, the proposed system achieves a similar SRE in auto-encoder applications, showing the offsets of 3.57 and 52.59 in MNIST and ImageNet dataset respectively. These differences are very small. Only 0.4% and 0.3% pixels are different for the input images in MNIST and ImageNet, respectively. Compared to the standard CD in classification applications, the proposed system has less than 1% accuracy loss.

**Area Overhead**. Table V shows the area comparisons among the recent memristor-based RBM systems. Compared to the Mem-cen and Hyb systems, the proposed design takes 46.5% and 22.5% lower area consumption respectively. This is because that the Mem-cen system uses the CMOS latches for neuron outputs, while the Hyb system uses four times more memristors for synapses than the proposed system. Compared to the Two-m system, the proposed design consumes double the memristors for the storage of weights and neurons. Eventually


DOI: 10.1587/elex.15.20171062
Received October 19, 2017
Accepted December 19, 2017
Publicized January 11, 2018
Copyedited January 25, 2018

**Table III.** Performance and energy of recent RBM training systems

| Platforms | Hyb | Two-m | Mem-cen | Prop. |
|---|---|---|---|---|
| Cycles of each CD | 6 | 10 | 5 | 3 |
| Performance speedup | 2x | 3.3x | 1.7x | - |
| Energy (pJ) | 932.418 | 379.525 | 823.021 | 341.689 |
| Energy efficiency | 2.7x | 1.1x | 2.4x | - |

**Table IV.** Accuracy results on the auto-encoder and classification

| Application | Dataset | Standard CD [1] | | | Prop. | | |
|---|---|---|---|---|---|---|---|
| | | train val. | cro.- | test | train val. | cro.- | test |
| auto-enc. (SRE) | MNIST | 12.80 | 13.02 | 12.59 | 16.46 | 16.57 | 16.16 |
| | Imagenet | 674.81 | 708.19 | 684.66 | 622.56 | 663.97 | 632.07 |
| classif. (accuracy) | MNIST | 99.99% | 98.73% | 98.60% | 99.96% | 98.38% | 97.90% |

**Table V.** Areas of the signal storage cells and synapses

| Platforms | Hyb | Two-m | Mem-cen | Prop. |
|---|---|---|---|---|
| Area ($mm^2$) | 0.2138 | 0.1003 | 0.3100 | 0.1657 |

it takes more than 39.5% area consumption. However, the area consumption is very small and only takes 0.41% of the recent famous ASIC-based NN accelerator [12] (40.8 $mm^2$) on the same technology.

## 5 Conclusion

We propose a novel RBM training system to boost up the CD procedure for big-data applications. The system enables storing the neuron outputs and performing the MAC operations in the CD procedure within the same cycle and less memory access. Experiments show that we achieve a speedup of 2770x against CPU, and deliver 1.7x~3.3x faster performance, 1.1x~2.7x better energy efficiency than recent state-of-the-art RBM training systems.

## Acknowledgments

This work is partially supported by the National Natural Science Foundation of China under Grant No. 61573299, 61472123, and the Hunan Provincial Natural Science Foundation under Grant No. 20174003.


DOI: 10.1587/elex.15.20171062
Received October 19, 2017
Accepted December 19, 2017
Publicized January 11, 2018
Copyedited January 25, 2018