

Memristor-Based Biologically Plausible Memory Based on Discrete and Continuous Attractor Networks for Neuromorphic Systems

Yanghao Wang, Liutao Yu, Si Wu, Ru Huang,* and Yuchao Yang*

To approach an advanced neuromorphic system, a significant unsettled problem is how to realize biologically plausible memory structures that are dramatically different from classical computers. Herein, a physical system based on memristors is simulated to realize associative memory based on discrete attractor networks, which is essentially content-based storage, and the influence of device characteristics on network performance is systematically studied. An in situ unsupervised learning method is applied to make greater use of array structure and competitions between neurons, demonstrating significant performance improvement in memory capacity and noise tolerance compared with existing supervised approaches. By extending to continuous attractor neural networks (CANNs), working memory is realized based on memristors for the first time via simulation, and the write and read noises in memristor arrays are found to have different impacts on the ability of CANN in maintaining dynamic information. This work lays a foundation for the construction of future advanced neuromorphic computing systems.

exactly an inspiration to establish a complete set of new mathematical tools to analyze and estimate future neuromorphic computing systems.

A memristor, proposed in 1971^[4] and experimentally established in 2008,^[5] is a resistive device as an optimized future neuromorphic device to this kind of non-von Neumann computing. Memristor can change its resistance according to the interior state and the exterior stimulation, such as voltage pulse. Previous studies have shown that a crossbar structure based on memristors can accelerate various artificial neural networks (ANNs) by directly mapping vector-matrix multiplication (VMM), a most intensive computing component, to electrical parameters relying on Ohm's law and Kirchhoff's law.^[6,7] Under this principle, the VMM computing process directly happened in situ, thus avoiding

1. Introduction

Neuromorphic computing, a novel concept proposed in 1990s, suggests that future computer system can imitate the operation principles of human brains by processing analog signals in parallel directly on the physical level.^[1] It is a promising way to move beyond the deterministic von Neumann model of computation due to its three significant advantages—high parallelism, low power consumption, and in-memory computing^[2]—and can be included in a Shannon-inspired statistical computation model.^[3] Guided by information theory, it is possible to explore its design principles of circuits and architectures that approach the limits of energy efficiency, latency, and accuracy. This is

the memory wall (von Neumann Bottleneck) caused by fetching data from memory. Especially in supervised learning, it can reduce the computational complexity of feed forward process and back-propagation from NP to P.^[8] Therefore, current studies mostly focus on classification and regression tasks to make use of this new computing mechanism as a complement to complementary metal oxide semiconductor (CMOS) circuits. However, different physical mechanisms of memristors, such as conducting filament formation/dissolution and phase change, decide that there are device imperfections that require further optimizations.^[9,10] Therefore, many theoretical models have been set up to analyze the influence of device imperfect characteristics,^[11–13] such as number of weight states, nonlinearity, asymmetry, variation,

Y. Wang, Prof. R. Huang, Prof. Y. Yang
Key Laboratory of Microelectronic Devices and Circuits (MOE)
Department of Micro/Nanoelectronics
Peking University
Beijing 100871, China
E-mail: ruhuang@pku.edu.cn; yuchaoyang@pku.edu.cn

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202000001>.

© 2020 The Authors. Published by WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202000001

Dr. L. Yu, Prof. S. Wu
School of Electronics Engineering and Computer Science
Peking University
Beijing 100871, China

Prof. R. Huang, Prof. Y. Yang
Academy for Artificial Intelligence
Peking University
Beijing 100871, China

Prof. R. Huang, Prof. Y. Yang
Frontiers Science Center for Nano-optoelectronics
Peking University
Beijing 100871, China

and so on, mostly in supervised learning. It is therefore an important question whether the requirements of devices in unsupervised learning are different from supervised learning, when using other training methods rather than backpropagation.

Furthermore, ANNs based on memristors are dedicated accelerators targeted for specific computing applications. It is actually more challenging and important to build a universal neuromorphic computer. In this case, not only do we need to accelerate feedforward neural network to do classification, but also need to build brain-inspired memory, which may generate totally new computing architectures. There are a variety of memories in our brain, and previous studies have realized long-term memory, short-term memory, and mutual transformation of them on single device level.^[14] An interesting thing is, as for memory, classical digital computers use address-based storage, while our brain uses content-based storage. From the perspective of computational neuroscience, biological neural networks always have an effective energy function. Through recurrent connections, the activity of the network will transfer from one initial state to a locally lowest point of energy through time, which is called an attractor. Biological associative memory is assumed to be stored in this abstract state. There are many memory patterns stored in one neural network, which depends on the weights of synapses.

If the attractors are discrete, an initial state will fall into the nearest attractor. This is the model of associative memory, which has been performed on memristor crossbar earlier.^[15] However, these previous studies used offline learning called Hebbian rule, which means the weights of synapses had been calculated in software and simply mapped to the memristor crossbar. The performance of Hebbian rule is poor and unable to support large-scale attractor networks. Other algorithms were also applied to improve the performance,^[16] but the learning process is still offline.

By bringing in translationally invariant bell-shaped connectivity pattern, the network attractors can form a plane marvelously. This is called a continuous attractor neural network (CANN), which has received broad attention from both theoreticians and experimentalists.^[17] From its advent, CANN has been successfully applied to theoretically describe the representation and processing of continuous features in neural systems, such as orientation,^[18] head direction,^[19] spatial location,^[20] and so on. Recently, experimental discoveries about *Drosophila* central brain supported the existence of CANN in real biological neural systems.^[21] When it comes to memory, it is believed that the brain can memorize the current state temporarily during dynamic assignments and use this information to do computation afterward. This is called working memory, which can be realized by a CANN naturally. In contrast, associative memory represents as long-term learning, while working memory is on behalf of dynamically processing temporary information for computing. Due to the biological plausibility and powerful computational capabilities of CANN, an early study has tried to implement a CANN using electrical circuits based on CMOS,^[22] and recently CANN was also implemented for the tracking function of an unmanned bicycle.^[23]

In this study, an effective in situ online learning method named Oja rule is applied in associative memory based on unsupervised learning by introducing competition and cooperation

between neurons, showing that this method can get at least 10 times performance improvement, which will greatly reduce chip area and enhance robustness of the hardware. The influence of nonideal device characteristics including weight precision, nonlinearity, asymmetry, device-to-device variation, and cycle-to-cycle variation in unsupervised learning is systematically studied, and the results reveal that the weight precision has a more significant impact compared with nonlinearity. Furthermore, for the first time CANN is applied on neuromorphic devices to perform working memory based on offline training, and the impact of device noise on the ability of CANN in maintaining dynamic information is studied. This work will pave the way for producing brain-like memory for future neuromorphic computing systems.

2. Results and Discussion

2.1. Discrete Attractor Network

2.1.1. Network Model

Discrete attractor neural networks, also known as Hopfield Neural Networks, are fully connected networks, where each neuron has connections with the other neurons but does not have self-connections (Figure 1c). If the weight matrix is a symmetric matrix and the diagonal element is 0, there must be attractors in the network. By following Equation (1) and iterating over time, the network will finally converge to a certain pattern according to predefined weights of connections.

$$X_i^{t+1} = \text{sgn} \left(\sum_j W_{ij} \times X_j^t + b_i \right) \quad (1)$$

The energy function can be defined as $E = -\frac{1}{2} \sum_{i \neq j} W_{ij} X_i X_j$, which can represent the state of the network, and the energy function is always reduced or unchanged during network operation. When all states are tiled into a 2D plane, the energy function can form a surface (Figure 1b), where all the minima in the hollows act as attractors. The network state will converge to one of the attractors during evolution, and therefore the neurons of the network will have a determined mode eventually.

Therefore, by setting its weights, the network has a capacity to store data and can restore images from corrupted images. Here, we use 50 32×32 images selected from CIFAR 100^[24] to test the capacity of the network (Figure 1a). These images were transformed into binary to clearly explain the problem, when black means 1 and white means -1 . The size of the network is 1024×1024 .

As the most intensive part of computation is VMM, we can map this network into memristor crossbar and the external circuits only need to judge the output values and send the binary outputs back to inputs, as shown in Figure 1d. If the number of neurons is n , the required size of synaptic weights is $n \times n$. To represent negative weight and simplify the operation of external circuitry, we adopted a strategy that represents one synapse weight using a differential pair of memristors. It is noteworthy that the equivalent weights of diagonal line can always be set as zero because neurons do not have self-connections. If these

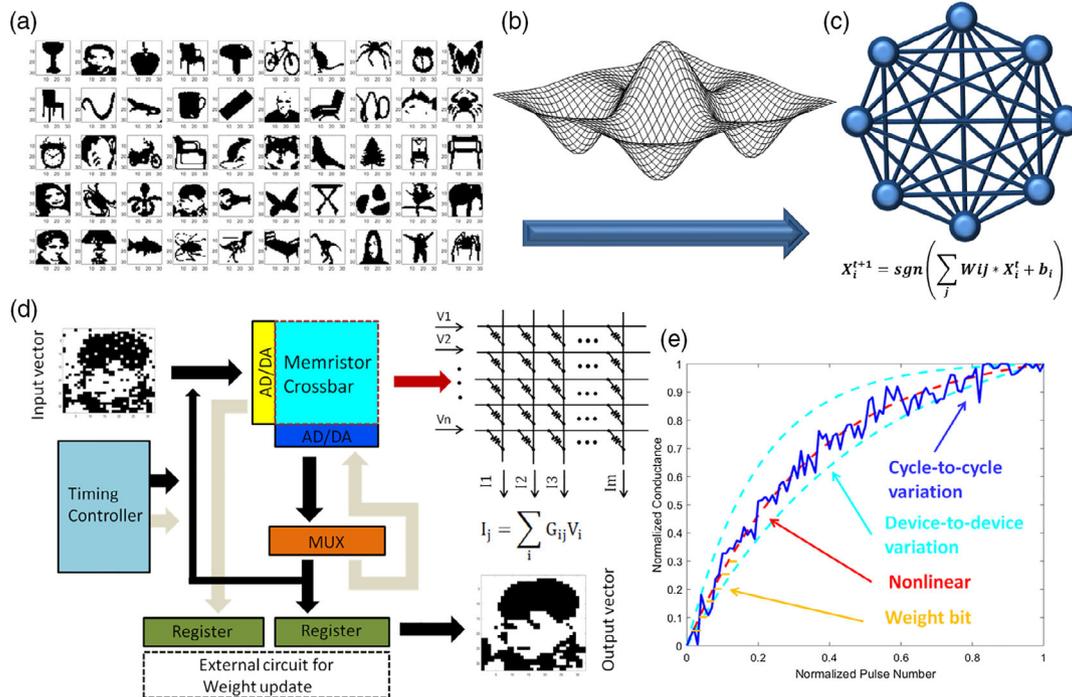


Figure 1. a) Dataset selected from CIFAR100. b) Diagram of discrete attractors, while information stored at relatively low points. c) Schematic diagram of discrete attractor network. Each dot represents a neuron, all linked to each other. d) Implementation of associative memory on memristor array. e) Different kinds of nonideal properties from memristive synaptic devices.

weights are set as nonzero, it can be assumed as a self feedback. If they are set properly, the network would change into a chaotic neural network that can solve combinatorial optimization problems,^[25] namely, to find the global minimum instead of nearest minimum. Recently, this has also been realized utilizing the random telegraph noise in memristive devices.^[26] In this associative memory task to find nearest minimum, the input image is corrupted with noise and the output in simulation can perfectly get the prestored images. The methods of adding noise x is randomly choosing one pixel inverted for x times rather than randomly choosing x pixels inverted. Otherwise, when x equals the total number of pixels, the corrupt image just becomes inverse and it is not proper.

2.1.2. Oja Rule

To set the weights, we apply an advanced rule named Oja rule

$$Y = X' \times W \quad (2)$$

$$\Delta W' = \alpha(X - W' \times Y') \times Y \quad (3)$$

where W is the weight matrix and α is the learning rate.

Different from the previous work using software calculated weights, this rule can train the network online in the crossbar and will therefore get significant performance improvement. The main principle of this rule is to use the competition between neurons, which is also used in locally competitive algorithm (LCA),^[27] explained as Winner Takes All (WTA). This rule needs to use the weight matrix and transposition of weight matrix, as

backpropagation algorithm in supervised learning also needs transposition of weight matrix. By exchange I/O port, we can use the transposition of weight matrix in an in situ way. The computing process is also shown in Figure 1d.

Oja rule is a training method, grasping the characteristics of principal components. It needs high weight precision, and will get more refined and functional weight structure in return. As shown in Figure 2a, when the number of stored patterns $N > 5$, the weight map of Hebbian rule has almost no change, always following a fixed frame. In contrast, the weight map by Oja rule has higher distribution evenness, thus owning more delicate weight distribution structures. As reflected in results (Figure 2b, where lines of different colors represent different noise levels), when the number of stored patterns is larger than 5, the recall accuracy of Hebbian rule drops abruptly, while the recall accuracy of Oja rule still remains high even if $N = 50$. Evidently, the neuromorphic system can still recall the full pattern with high accuracy using Oja rule, even when 50 patterns are stored and only half of the original information is given. Therefore, the memory capacity of Oja rule is roughly 10 times better than Hebbian rule, and the neuromorphic memory system trained by Oja rule also has better robustness, as manifested by the much higher noise tolerance, demonstrating overall significant performance improvement.

2.1.3. Device Characteristics

The theoretical memristor device model in Neurosim^[11] is adopted here for the evaluation of device characteristics.

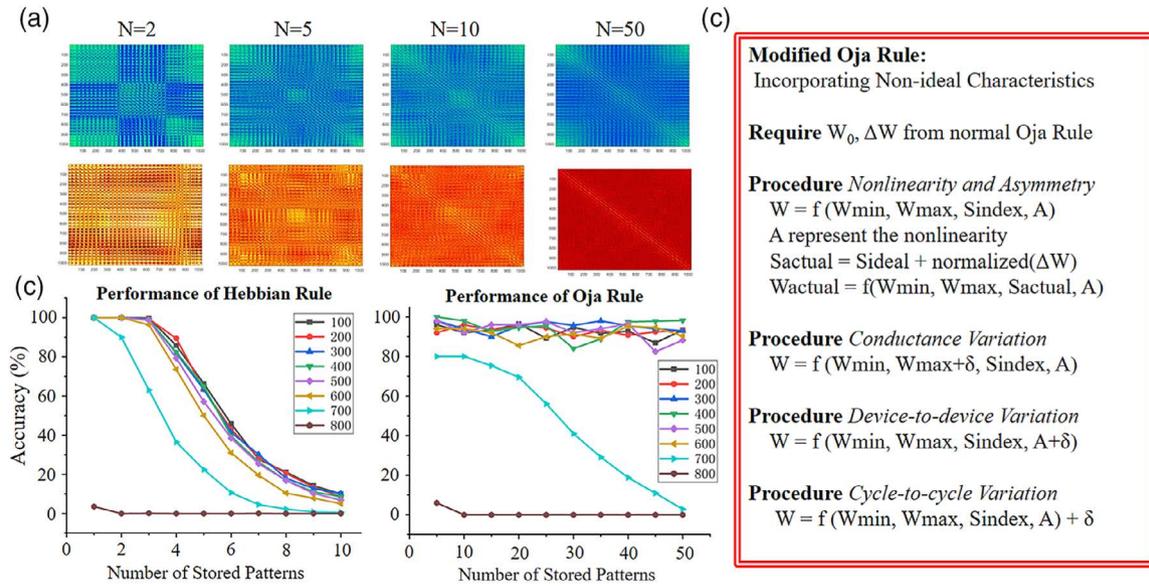


Figure 2. a) Weight array under different storage capacity requirements. The above is Hebbian rule. The bottom is Oja rule. b) The comparison between Oja rule and Hebbian rule. Lines of different colors represent different levels of noise. c) The process of considering nonideal effects of devices.

$$G_{LTP} = B \left(1 - e^{-\left(\frac{s}{A}\right)} \right) + G_{min}$$

$$G_{LTD} = -B \left(1 - e^{-\left(\frac{S - S_{max}}{A}\right)} \right) + G_{max} \quad (4)$$

$$B = \left(G_{max} - G_{min} \right) / \left(1 + e^{-\frac{S_{max}}{A}} \right)$$

Here, we mainly discuss weight precision, nonlinearity, conductance variation, device-to-device variation, and cycle-to-cycle variation. The detail process is shown in Figure 2c.

2.2. Continuous Attractor Neural Network

2.2.1. Network Model

CANN is a candidate for canonical models of neural information representation and processing. Intuitively, a CANN is endowed with a bunch of continuous attractors, which are neutrally stable (Figure 3a). Traditionally, the dynamics of a CANN describing head direction system can be described as^[28]

$$\begin{cases} \tau \frac{\partial U(x,t)}{\partial t} = -U(x,t) + \rho \int_{x'} J(x, x') r(x', t) dx' + I^{ext}(x, t) \\ r(x, t) = \frac{[U(x,t)]_+^2}{1 + k\rho \int_{x'} [U(x', t)]_+^2 dx'} \end{cases} \quad (5)$$

where $U(x,t)$ represents the total synaptic inputs of the neuron x ($x \in [-\pi, \pi]$) at time t , $r(x,t)$ is the instantaneous firing rate, τ is the time constant, ρ is the neuron density, and k is the strength of global inhibition. $J(x, x') = \frac{1}{2\pi a^2} \times \exp[-(x - x')^2 / (2a^2)]$ is the connectivity strength between neuron x and neuron x' , which is a translationally invariant function, ensuring the neutral stability of those continuous attractors. $I^{ext}(x, t) = \frac{1}{2\pi b^2} \times \exp[-(x - x_0)^2 / (2b^2)]$ is the Gaussian-shaped external input, where x_0 indicates the center of the stimulus, i.e., head direction of the animal. Note that in this model inhibitory neurons are not

explicitly defined, while global inhibition is achieved through divisive normalization, which may be implemented by shunting inhibition. Under such settings, network response $r(x,t)$ is also a bell-shaped function, and the representation of the head direction in the neural system is calculated through $z(t) = \sum_{x=-\pi}^{\pi} \frac{r(x,t)}{\sum_{x'} r(x', t)} x$. Due to its neutral stability, the internal representation $z(t)$ can smoothly track the rotation of head direction x_0 , while the internal representation still holds even in a dark environment (Figure 3b).

2.2.2. Working Memory

In this study, we explore the ability of a CANN to hold persistent activities triggered by a brief external stimulus.^[29,30] Figure 3c shows the task of working memory based on continuous attractor neural networks in ideal situation, where the ordinate represents the normalized strength, and the abscissa represents the number of neurons (50 neurons from $-\pi$ to π). First of all, when no noise exists in the system, Figure 3c shows that a CANN can perfectly hold persistent activities, i.e., the internal representation is anchored to the center of the external stimulus and no deviation is observed as time elapses (see Movie 1, Supporting Information). Specifically, when external stimulus is applied to the neurons, the population of neurons will respond to it and obtain the shape and central position of the stimulus; after the external stimulus is removed, the neuron population can still remember the previous shape and central position of stimulus (Figure 3c). However, in electronic neuromorphic systems, there are three major problems in offline learning, namely, weight precision, write noise, and read noise. Regarding the weight precision, offline learning usually has much lower precision requirements than online learning. In addition, it is the network structure that ensures ability of CANN in working memory rather than the weight distribution. It is found that CANN

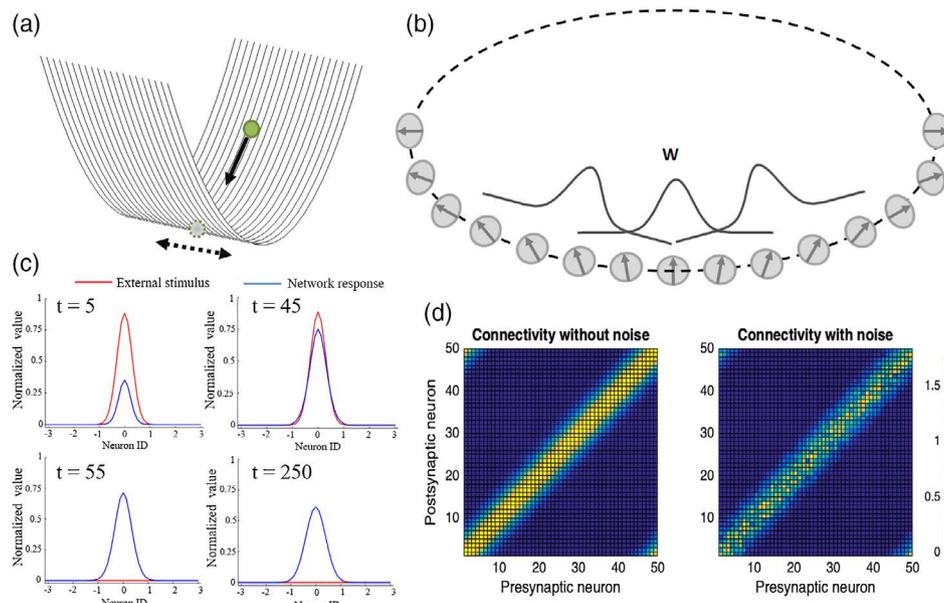


Figure 3. a) Descent toward continuous attractors, as well as neutral equilibrium between these attractors. b) Schematic diagram of continuous attractor network, which has self-sustained activities based on the interaction of positive connection and group inhibition. c) Ideal working memory. The x -axis means neuron ID (x in Equation (5)) and the y -axis represents normalized value of external stimulus I^{ext} (red curve) and network response r (blue curve). The external stimulus lasts from $t = 0$ to $t = 50$, the center of which represents a direction. The external stimulus induces network responses, the center of which represents the encoding of that direction by the neural network ($t = 5$, $t = 45$). After the external stimulus is withdrawn, the network still holds the dynamic memory ($t = 55$, $t = 250$). d) Conductance matrix mapped on the memristor array, without noise (left) and with noise (right).

can even work under any weight matrix of arbitrary precision, including 1-bit. When programming connection strengths of the network onto memristor crossbars, the weight matrix may have some nonideal factors even after multiple verifications, embodied in the difference between conductance values and target values (write noise, Figure 3d). In addition, the parasitic effects and thermal fluctuations during reading also affect the final result (read noise). A CANN based on memristor model is therefore set up to analyze the tolerance of neuromorphic computing to the aforementioned imperfections. The dynamic location of the network center after the external stimulus is withdrawn can therefore reflect the reliability of neuromorphic computing in this dynamic working memory.

2.3. Influence of Device Characteristics

The influence of device imperfections on the implementations of associative memory and working memory was systematically studied, as shown in Figure 4 and 5. As shown in Figure 4a, when the weight precision is lower than 6-bit, Oja rule cannot work properly. However, once the weight precision is higher than 6-bit, the attractor network has better performance than the network based on Hebbian rule, as shown in Figure 2b. Figure 4b further shows that the training based on Oja rule has greater endurance to nonlinearity (>2) than backpropagation because previous results by Chen et al. have shown a dramatic performance drop when the nonlinearity exceeds 1.5 using backpropagation.^[11] Figure 4c shows the real weight matrix distribution of memristor crossbar (with weight bit of 10, nonlinearity

of 1.5, $N=20$), where one can see that nonideal effects of the devices increase the deviation of adjacent weights in the memristive crossbar.

A universal model of working memory based on memristive neuromorphic systems has also been set up, and Figure 5 shows the simulation results. Through simulating weight matrix with different weight precisions, it is verified that CANN can be easily implemented by neuromorphic devices. As shown in Figure 5a, even simplest binary devices can realize CANN, just by mapping the conductance to +1 and 0. The maximum tolerance of write noise in this assignment is 5%, as shown in Figure 5b, which can be realized by closed-loop verifications during programming or adding transistors to regulate the currents. The maximum tolerance of read noise in this assignment is 10% (Figure 5c). It is worth mentioning that the write noise and read noise have different impacts on network activities. The existence of write noise will reshape the response activity. Figure 5d shows that the network activity with write noise of 0.1 becomes sharper. When the write noise is stronger (e.g., 0.3), the single peak of neuronal activities will even split into multiple peaks (see Movie 2, Supporting Information). In contrast, the existence of read noise (with an amplitude of 0.1) will make the location of peak drift with the overall shape almost kept unchanged (Figure 5e). When the read noise is stronger (e.g., 0.3), the peak drifts faster within a larger range (see Movie 3, Supporting Information). It may explain why people could be distracted because the working memory is originally temporary memory for the next computing tasks and in this situation its peak is shifted all the time.

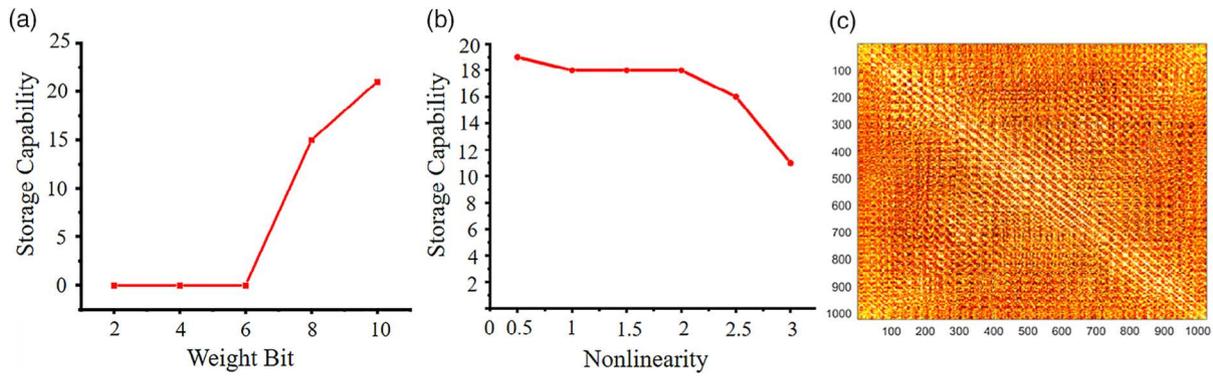


Figure 4. a) The impact of weight bit on storage capability. b) The impact of nonlinearity to storage capability. c) Weight matrix based on the device model in associative memory. (bit = 10, nonlinearity = 1.5, stored pattern $N = 20$).

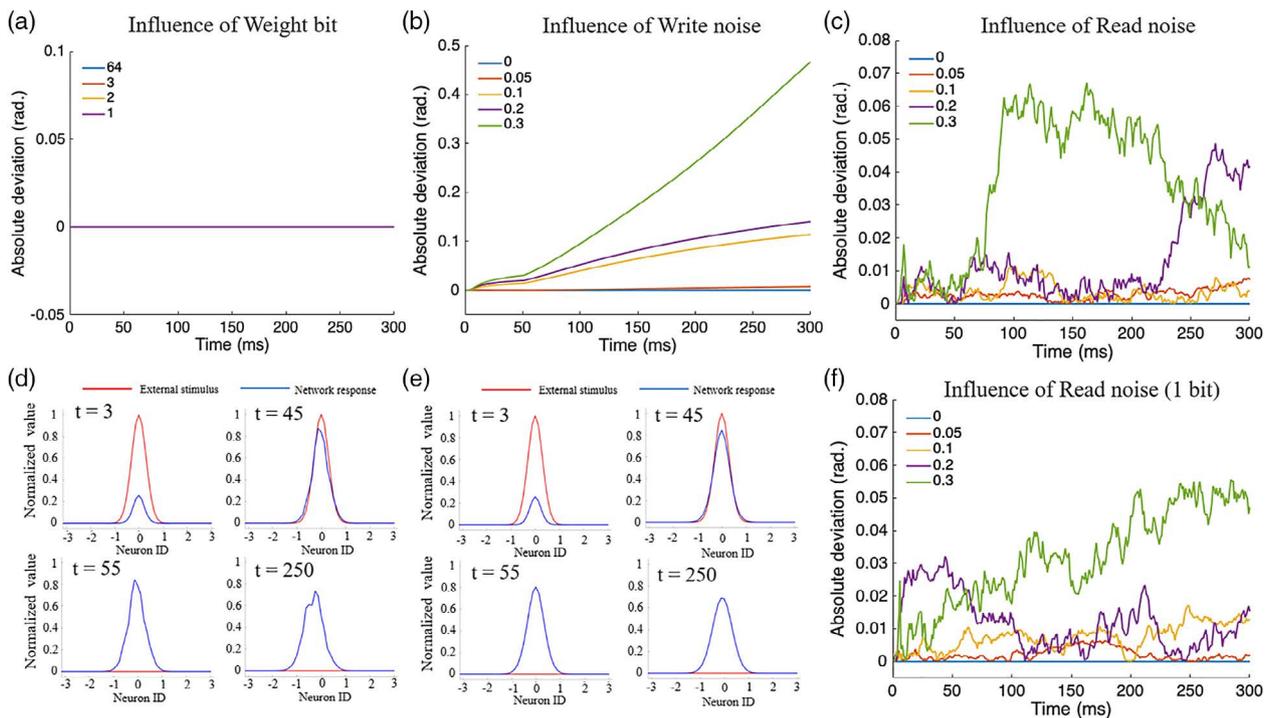


Figure 5. a) The influence of weight precision on network activity. b) The influence of off-line write noise on network activity based on weight matrix of full precision. c) The influence of read noise on network activity based on weight matrix of full precision. d) The write noise reshapes the pattern of network activity. e) The read noise sways the pattern of network activity. f) The influence of read noise on network activity based on single-bit weight matrix.

3. Conclusion

Biologically inspired and plausible neural networks have always been the ultimate goal of neuromorphic computing. Previous studies have shown that supervised learning can be capable of classification and regression. In this work, a new unsupervised learning algorithm is adopted and implemented on associative memory. From the perspective of actual function and efficiency, associative memory is essentially a kind of content-based storage. There are other methods based on memristors in realizing content-based storage, but these methods still need a searching algorithm with computational complexity N , while the iteration number of attractor network is only $\log N$. As for working

memory, it has been considered an effective way to approaching the edge of chaos's decision because it can dynamically accept input from outside and can maintain the information itself. Here, associative memory and working memory are realized based on discrete and continuous attractor networks, respectively, and the influence of device characteristics on network performance is systematically studied. The in situ online training demonstrates significant performance improvement in memory capacity and noise tolerance. By cascading CANN with other computing units, such as reservoir computing, the total system can have better dynamic information processing capability because CANN can constrain external input into a specific form and be able to temporarily remember external stimulus in the

network level. Previous studies have shown the application of memristors in reservoir computing,^[31] and suggested that combination of discrete attractor neural network with reservoir computing can have better performance, showcased by the generation of handwritten numbers.^[32] Using attractors to constrain the complex dynamics in reservoir will help the computing system converge faster and avoid sustained oscillations. The realization of brain-inspired content-based storage may facilitate the development of real neuromorphic systems.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This work was supported by the National Outstanding Youth Science Fund Project of National Natural Science Foundation of China (grant no. 61925401), National Key R&D Program of China (grant no. 2017YFA0207600), the National Natural Science Foundation of China (grant nos. 61674006, 61927901, and 61421005), and the 111 Project (B18001). Y.Y. acknowledges the support from Beijing Academy of Artificial Intelligence (BAAI) and the Tencent Foundation through the XPLOER PRIZE.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

associative memory, attractor neural networks, memristors, neuromorphic computing, working memory

Received: January 2, 2020

Revised: January 23, 2020

Published online: February 20, 2020

-
- [1] C. Mead, *Proc. IEEE* **1990**, *78*, 1629.
 [2] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, J. S. Plank, **2017**, arXiv:1705.06963.
 [3] N. R. Shanbhag, N. Verma, Y. Kim, A. D. Patil, L. R. Varshney, *Proc. IEEE* **2018**, *107*, 90.
 [4] L. O. Chua, *IEEE Trans. Circuit Theory* **1971**, *18*, 507.
 [5] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, *Nature* **2008**, *453*, 80.
 [6] J. J. Yang, D. B. Strukov, D. R. Stewart, *Nat. Nanotechnol.* **2013**, *8*, 13.

- [7] M. Prezioso, F. Merrikkh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, D. B. Strukov, *Nature* **2015**, *521*, 61.
 [8] M. Le Gallo, A. Sebastian, R. Mathis, M. Manica, H. Giefers, T. Tuma, C. Bekas, A. Curioni, E. Eleftheriou, *Nat. Electron.* **2018**, *1*, 246.
 [9] Y. Yang, W. Lu, *Nanoscale* **2013**, *5*, 10076.
 [10] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola, L. L. Sanches, *Adv. Phys. X* **2017**, *2*, 89.
 [11] P. Y. Chen, X. Peng, S. Yu, IEEE International Electron Devices Meeting (IEDM), IEEE, San Francisco, USA **2017**, p. 6.1.
 [12] S. R. Nandakumar, M. Le Gallo, I. Boybat, B. Rajendran, A. Sebastian, E. Eleftheriou, in IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, Florence, Italy **2018**, pp. 1–5.
 [13] X. Sun, S. Yu, *IEEE Trans. Emerging Sel. Topics Circuits Syst.* **2019**, *9*, 570.
 [14] J. Zhu, Y. Yang, R. Jia, Z. Liang, W. Zhu, Z. U. Rehman, L. Bao, X. Zhang, Y. Cai, L. Song, R. Huang, *Adv. Mater.* **2018**, *30*, 1800195.
 [15] S. G. Hu, Y. Liu, Z. Liu, T. P. Chen, J. J. Wang, Q. Yu, L. J. Deng, Y. Yin, S. Hosaka, *Nat. Commun.* **2015**, *6*, 7522.
 [16] Y. Zhou, H. Wu, B. Gao, W. Wu, Y. Xi, P. Yao, S. Zhang, Q. Zhang, H. Qian, *Adv. Funct. Mater.* **2019**, *29*, 1900155.
 [17] S. Amari, *Biol. Cybern.* **1977**, *27*, 77.
 [18] R. Ben-Yishai, R. L. Bar-Or, H. Sompolinsky, *Proc. Natl. Acad. Sci.* **1995**, *92*, 3844.
 [19] K. Zhang, *J. Neurosci.* **1996**, *16*, 2112.
 [20] A. Samsonovich, B. L. McNaughton, *J. Neurosci.* **1997**, *17*, 5900.
 [21] S. S. Kim, H. Rouault, S. Druckmann, V. Jayaraman, *Science* **2017**, *356*, 849.
 [22] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, H. S. Seung, *Nature* **2000**, *405*, 947.
 [23] J. Pei, L. Deng, S. Song, M. Zhao, Y. Zhang, S. Wu, G. Wang, Z. Zou, Z. Wu, W. He, F. Chen, N. Deng, S. Wu, Y. Wang, Y. Wu, Z. Yang, C. Ma, G. Li, W. Han, H. Li, H. Wu, R. Zhao, Y. Xie, L. Shi, *Nature* **2019**, *572*, 106.
 [24] A. Krizhevsky, V. Nair, G. Hinton, Cifar-10 and cifar-100 datasets. <https://www.cs.toronto.edu/kriz/cifar.html> (accessed: October 2009).
 [25] M. R. Mahmoodi, M. Prezioso, D. B. Strukov, *Nat. Commun.* **2019**, *10*, 5113.
 [26] F. Cai, S. Kumar, T. Van Vaerenbergh, R. Liu, C. Li, S. Yu, Q. Xia, J. J. Yang, R. Beausoleil, W. Lu, J. P. Strachan, **2019**, arXiv:1903.11194.
 [27] P. M. Sheridan, C. Du, W. D. Lu, *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *27*, 2327.
 [28] S. Wu, K. Hamaguchi, S. Amari, *Neural Comput.* **2008**, *20*, 994.
 [29] A. Compte, N. Brunel, P. S. Goldman-Rakic, *Cereb. Cortex* **2000**, *10*, 910.
 [30] A. Renart, P. Song, X. J. Wang, *Neuron* **2003**, *38*, 473.
 [31] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, W. D. Lu, *Nat. Commun.* **2017**, *8*, 2204.
 [32] R. Pogodin, D. Corneil, A. Seeholzer, J. Heng, W. Gerstner, **2019**, arXiv:1910.10559.