SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2023, Vol. 66 182403:1–182403:10
https://doi.org/10.1007/s11432-022-3601-8

How could imperfect device properties influence the performances of spiking neural networks?

Jingyang CHEN†, Zhihao WANG†, Tong WANG, Heming HUANG, Zheyuan SHAO, Zhe WANG & Xin GUO*

State Key Laboratory of Material Processing and Die & Mould Technology, School of Materials Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074*, China*

Received 1 June 2022/Revised 25 August 2022/Accepted 20 October 2022/Published online 4 July 2023

Abstract Spiking neural networks (SNNs) provide an efficient way to apply artificial intelligence systems on edge devices. Memristive devices with tunable conductance states can be used to emulate the functions of biological neurons and synapses and to build SNNs. In this work, fully-connected SNNs based on various memristive devices are constructed, and a hardware-compatible spike-timing-dependent plasticity (STDP) learning rule is applied to train the SNNs. Strategies are designed to suppress the overfitting problem and improve the performance of the SNNs in the case of a small training set. However, the properties of memristive devices are never perfect. The effects of imperfect device properties, e.g., asymmetric weight update, insufficient number of conductance states, and low on/off ratio, on the performance of the SNNs are elaborated.

Keywords oxides, memristive devices, device properties, spiking neural networks, neuromorphic computing

Citation Chen J Y, Wang Z H, Wang T, et al. How could imperfect device properties influence the performances of spiking neural networks?. Sci China Inf Sci, 2023, 66(8): 182403, https://doi.org/10.1007/s11432-022-3601-8

1 Introduction

In recent years, computer vision has been successfully applied in many areas, such as image recognition [1–3], motion tracking [4,5], action recognition [6,7], and autonomous driving [8,9]. The success of computer vision relies on the development of artificial neural networks (ANNs) [10, 11], which can learn high-level features from a large amount of input data [12]. However, the further development of ANNs is largely inhibited by the contemporary computing architecture, i.e., the von Neumann architecture [13–15], in which data have to shuttle between memories and computing units, which results in large power consumption and latency. Therefore, ANNs cannot be applied in some energy-insufficient scenarios, such as edge devices [16].

Owing to special architectures, a human vision system outperforms a computer vision system in terms of processing speed and energy consumption when processing some complex tasks [17]. As shown in Figure 1(a), a human vision system includes a variety of biological cells. When light enters the eyeball through the pupil, photoreceptor cells on the retina are stimulated, causing neuron spikes to be transmitted along optic nerves to the visual center of the cerebral cortex, which eventually creates a vision in the brain [18]. Therefore, a human vision system can be simplified as a combination of optical sensors and a biological neural network consisting of basic computing units, i.e., neurons and synapses [19]. With inherently integrated capabilities of sensing, memory, and computing, a human vision system acquires and processes environmental information in real time and, therefore, can handle many difficult tasks with low energy consumption and high efficiency [20].

Spiking neural networks (SNNs) working like biological neural networks are the next generation of neural networks [21,22]. Similar to biological neural networks, the information in SNNs is encoded by the

* Corresponding author (email: xguo@hust.edu.cn)
† Chen J Y and Wang Z H have the same contribution to this work.

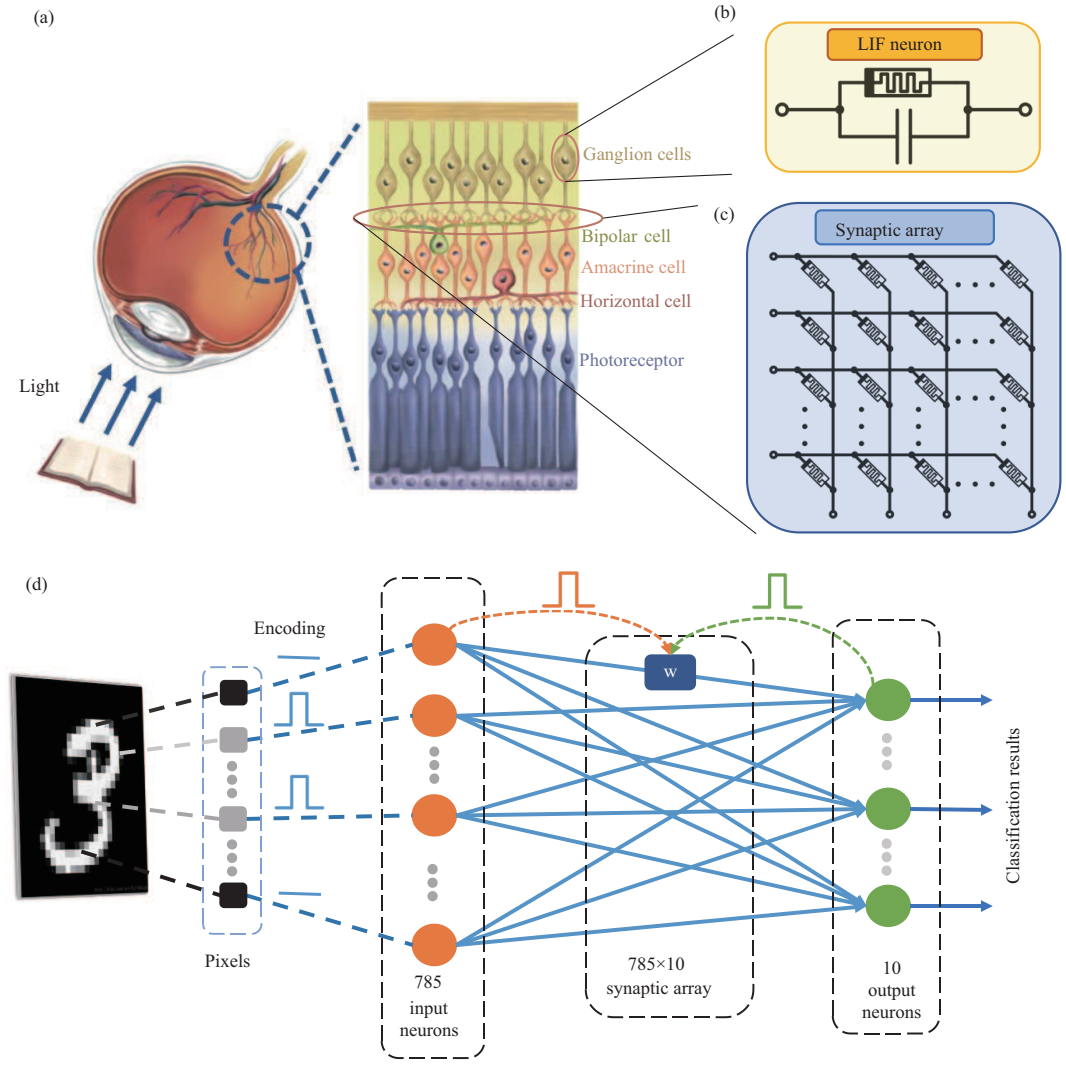

Figure 1 (Color online) (a) Human vision system; (b) LIF neuron based on memristive devices; (c) synaptic array based on memristive devices; (d) SNN based on memristive devices.

timing or rating of spikes [23], and SNNs work only when events occur, so they are more energy-efficient than ANNs [24–26]. Some SNNs have been realized with complementary metal-oxide-semiconductor (CMOS) circuits, such as Neurogrid [27], SpiNNaker [28], TrueNorth [29], Loihi [30], and Tianjic [31]. However, the realization of synapses and neurons in these SNNs typically requires several CMOS transistors [32,33]. Therefore, the development of artificial synapses and neurons with a simplified method is urgent for the development of SNNs.

Memristive devices, with a simple structure, high programming speed, efficient energy consumption, great scalability, and versatile performances (analog/digital and volatile/non-volatile switching), have been successful in emulating the functions of biological neurons and synapses and building SNNs [34–39]. Different artificial neurons have been realized by memristive devices. For example, the so-called memristive integrate-and-fire (IF) model can be implemented by commercially available low-cost components, which demonstrates that memristive circuits do not rely on prohibitively expensive fabrication processes [40]. The Quasi-Hodgkin-Huxley (HH) neuron with leaky integrate-and-fire (LIF) functions physically realized with memristive devices has multiple functions similar to their biological counterparts [41]. A simplified memristive Hodgkin-Huxley neuron model is implemented by analyzing the dynamic properties of the Hodgkin-Huxley neuron model, which provides a good simulation tool for analyzing the HH neuron circuit [42]. Among various neuron models, the LIF neuron is most widely used for the algorithms and hardware implementations of SNNs due to its simple structure. As shown in Figure 1(b), a volatile memristive device in parallel with a capacitor can realize the LIF functions.

Compared with an IF neuron, a LIF neuron with the LIF function can achieve a high spike sparsity and increase the robustness of SNNs [43].

Non-volatile memristive devices, as shown in Figure 1(c), can emulate the functions of biological synapses and realize in-memory computing and parallel computing to accelerate ANNs [44]. However, ANNs based on precise computations normally impose high requirements on non-volatile devices, such as symmetric weight updating, sufficient conductance states, and a high on/off ratio [45]. However, SNN computing based on sparse spikes has the potential to tolerate non-ideal properties of memristive devices [46].

In this work, a fully connected SNN based on ideal memristive devices was first constructed and trained by a hardware-compatible spike-timing-dependent plasticity (STDP) learning rule [13,47,48]. Two strategies were designed to improve the performance of the SNN in the case of a small training set. The SNN trained by 200 training samples realized a classification accuracy of 74.4% with a testing set of 10000 samples, higher than the classification accuracy acquired by an equivalent ANN. To demonstrate the effects of imperfect device properties, three different memristive devices, i.e., Pd/W/WO_3/Pd, Pt/Ta/TaO_x/Pt, and Pt/WO_3/$LiSiO_x$/Pt, were used to construct SNNs, and the effects of the non-ideal properties of the memristive devices on the performances of the SNNs were analyzed.

2 Results and discussion

2.1 SNNs based on memristive devices

A fully connected SNN with two layers was constructed, as shown in Figure 1(d). The input layer and output layer contain 785 and 10 LIF neurons, respectively; the synaptic array connecting the two layers contains 785×10 non-volatile memristive devices. The LIF neurons can transmit analog signals into spike frequencies without using analog-to-digital conversion, so this SNN works in the analog domain. More details about the LIF neuron based on the memristive device of Ag/Ta_2O_5:Ag/Pt are provided in Appendix A. The synaptic arrays were realized by non-volatile memristive devices with continuously tunable conductance states. The synaptic array can perform vector-matrix multiplication and store synaptic weights. Therefore, the synaptic array enables a high area- and energy-efficient in-memory computation of weighted inputs, thus avoiding moving data from memories to computing units, as practiced in the von Neumann architecture.

Before inputting the samples from the MNIST dataset, the sample images were first binarized and then expanded into a vector with 784 elements. A bias was applied to the vector as an individual input. Accordingly, 785 input neurons were used to construct the input layer of the SNN, and then the input vector was encoded by the neurons into spikes. When the pixel was equal to 1, a positive spike (2.5 V, 1 ms) was sent to the corresponding input neuron; otherwise, no spike was sent. The input neurons responded to the encoded signals and generated spikes when their membrane potential exceeded the corresponding threshold. Then the spikes generated by the input neurons were transmitted to the output neuron by the synaptic array according to the synaptic weights. The output neurons were also LIF neurons, similar to the input neurons. In addition to the temporal integration of input signals from synapses, the output neurons also spatially integrated signals from different synapses, which was different from the input neurons. The firing frequencies of the output neurons indicate the classification results.

2.2 STDP learning rule

Different from the complex STDP learning rule in biology, a simplified and hardware-compatible STDP learning rule is proposed here to train the non-volatile memristive devices in the SNN, as shown in Figure 2(a). Before training the SNN, input samples were encoded into spikes through the coding method mentioned above. At the same time, the labels of the input samples were encoded as teacher signals. The label of a sample is a signal array of 10×1, the line corresponding to the sample number is 1, and all the other lines are 0. For example, the label of the first class is [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]. Then, the teacher signal was acquired by encoding the labels into spikes. When the label was equal to 1, a positive spike (5 V, 1 ms) was sent to the corresponding output neuron; otherwise, no spike was sent. The firing activities (ΔW_{11}, ΔW_{10}, and ΔW_{01}) of the input and output neurons were recorded to calculate the updated strides of the synapses. To be specific, ΔW_{11} is responsible for recording the activities when the input neuron and output neuron fire simultaneously, and ΔW_{10} and ΔW_{01} are responsible for recording

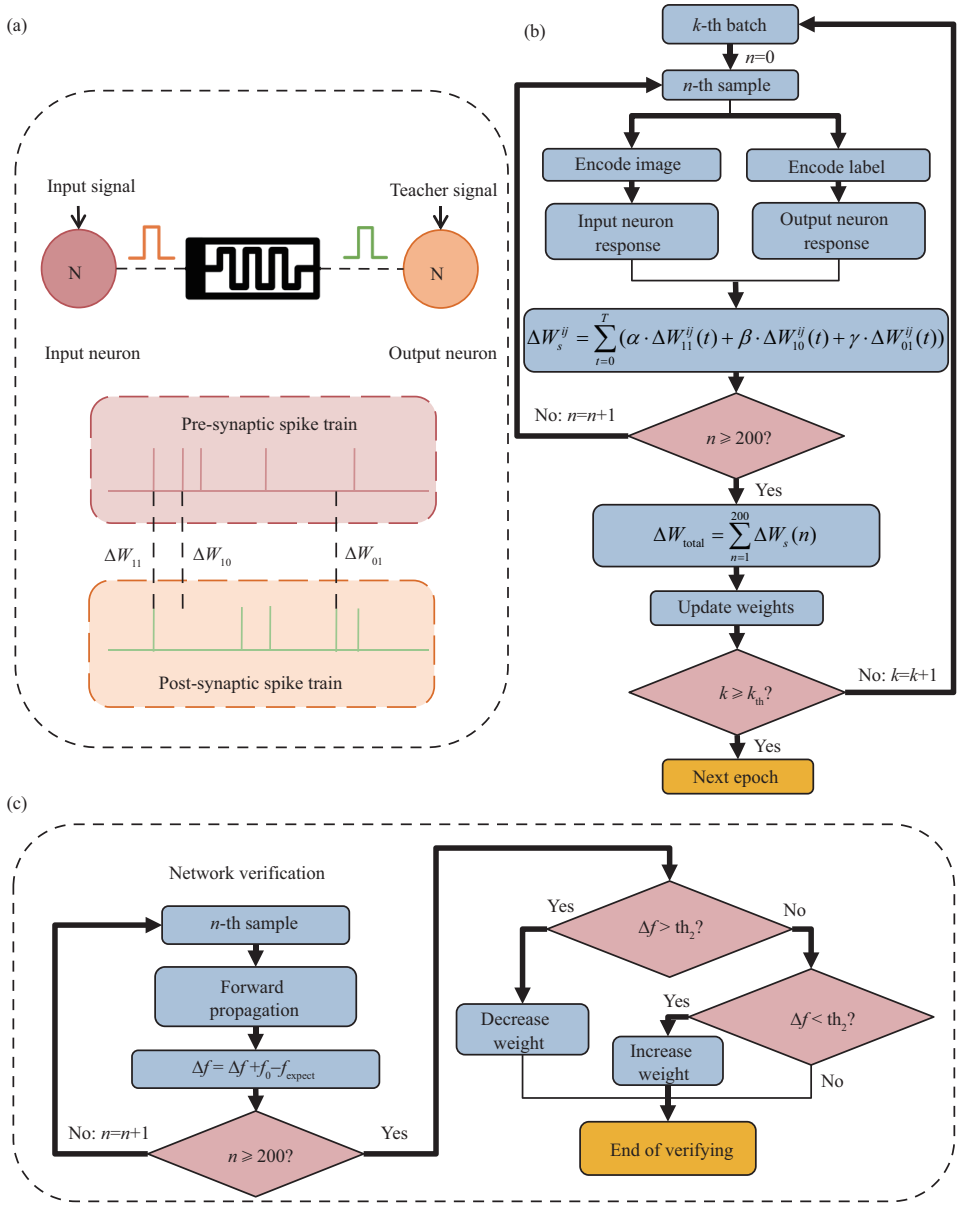

Figure 2 (Color online) (a) Simplified STDP learning rule. Updating the synaptic weight according to the timing of the pre- and post-synaptic spikes generated by the input neuron and output neuron. (b) Training flowchart of the SNN. (c) Working flowchart for the network verification.

the activities when only the input neuron or output neuron fires. In addition, in the training process, the output spikes of the neurons do not directly act on the synapses, and the recording of the pre- and post-synaptic spikes is achieved through a simple peripheral circuit. The STDP rule is applied to two spikes to update the corresponding synaptic weights. The hardware implementation of the simplified STDP learning rule can be realized by several basic components, and a solution is provided in Appendix B.

Twenty samples in each class were used to train the SNN according to the STDP learning rule. A training batch includes all the 200 samples. After training with all the samples in the batch, a testing set was sent to the network to acquire classification accuracy. Figure 2(b) shows the training process of the SNN by means of the STDP learning rule. Before training, the synaptic weights were randomly set. Then, the samples in the batch were individually inputted into the SNN. When inputting a sample, the image pixels and labels were first encoded into spike trains, which were sent to the input neuron and output neuron, respectively. While the neurons were processing the spike trains, timers were used to record the firing activities (ΔW_{11}, ΔW_{10}, and ΔW_{01}). Then, the firing activities were used to calculate

the update stride ΔW_s^{ij} according to the following equation:

$$\Delta W_s^{ij} = \sum_{t=0}^{T} (\alpha \Delta W_{11}^{ij}(t) + \beta \Delta W_{10}^{ij}(t) + \gamma \Delta W_{01}^{ij}(t)), \tag{1}$$

where ΔW_s^{ij} represents the update stride of the synapse connected to the i-th input neuron and the j-th output neuron, ΔW_{10}^{ij} is the number of the firing activities where the i-th input neuron fires while the j-th output neuron does not fire at some moment, and ΔW_{01}^{ij} is the number of the firing activities where the i-th input neuron does not fire while the j-th output neuron fires at some moment. α, β, γ are the parameters related to the synapse properties when training the SNN with ideal devices, i.e., $\alpha = 20$, $\beta = -2$, and $\gamma = -1$.

After inputting all the samples in the batch, the synapses were updated according to $\Delta W_{\text{total}}^{ij}$:

$$\Delta W_{\text{total}}^{ij} = \sum_{n=1}^{200} \Delta W_s^{ij}(n), \tag{2}$$

where $\Delta W_s^{ij}(n)$ represents the update stride when inputting the n-th sample in the batch and $\Delta W_{\text{total}}^{ij}$ is the accumulated update stride after inputting the whole batch. If $\Delta W_{\text{total}}^{ij} > 0$, a series of positive spikes are applied to the memristive device to increase the synaptic weight; if $\Delta W_{\text{total}}^{ij} < 0$, a series of negative spikes are applied to the memristive device to decrease the synaptic weight. Here, the number of spikes applied on the device was set to 2, which determines the learning rate of the SNN.

Two strategies were designed to improve the generalization ability of the SNN. The first strategy is similar to the dropout method used in ANNs, and by using this strategy, one can reduce the overfitting problem of the SNN. Specifically, in one training epoch, input signals were randomly discarded with a 50% chance before being fed into the SNN. Therefore, the SNN was trained by different input signals with different features, including the most common features that contribute to the right classification and rare features that only exist in a few samples and can reduce the classification accuracy. After training, the synaptic weights of the SNN can be seen as the average of different features, so this method dilutes the rare features and inhibits the overfitting problem. Eventually, the SNN performs better on the testing dataset.

The second strategy is the network verification strategy. After training all the samples in a batch, the network verification strategy was used to verify whether the SNN learned appropriate features. The flowchart for the network verification strategy is shown in Figure 2(c). The samples in the batch were sent to the SNN and forwardly propagated to all the layers. After the forward propagation, the output neurons generated spike trains with different frequencies. Then, the error of the whole batch was used to indicate the training efforts of the SNN. The error is calculated by

$$\text{error}_f^j = \sum_{n=1}^{200} f_{\text{output}}^j(n) - f_{\text{expect}}^j(n), \tag{3}$$

where error_f^j is the error of the j-th output neuron when inputting the whole batch samples and $f_{\text{output}}^j(n)$ and $f_{\text{expect}}^j(n)$ are the output frequency and the expected frequency of the j-th output neuron to the n-th sample, respectively. The expected frequencies are acquired by encoding the label of the n-th sample and inputting the encoded signals into the output neurons. For example, the label of the handwritten digit "1" is [1, 0, 0, 0, 0, 0, 0, 0, 0, 0], which is encoded as a spike train [1 V, 0 V, 0 V, 0 V, 0 V, 0 V, 0 V, 0 V, 0 V, 1 ms], and the expected frequencies of this class are [218 kHz, 0 Hz, 0 Hz, 0 Hz, 0 Hz, 0 Hz, 0 Hz, 0 Hz, 0 Hz, 0 Hz]. If error_f^j is higher than the first threshold (84 kHz), negative spikes are applied to the memristive devices connecting with the j-th output neuron; if error_f^j is lower than the second threshold (−84 kHz), positive spikes are applied to the memristive devices connecting with the j-th output neuron. The network verification strategy prevents the SNN from learning too many features and thus improves the generalization ability of the SNN.

2.3 Classification results of the SNN based on ideal devices

Figure 3(a) shows the classification accuracy of the SNN. Non-volatile memristive devices with ideal properties were first adopted as the synapses in the SNN. The conductance range is from 50 to 200 μS, and the number of the conductance states is 300, as shown in the inset of Figure 3(a).

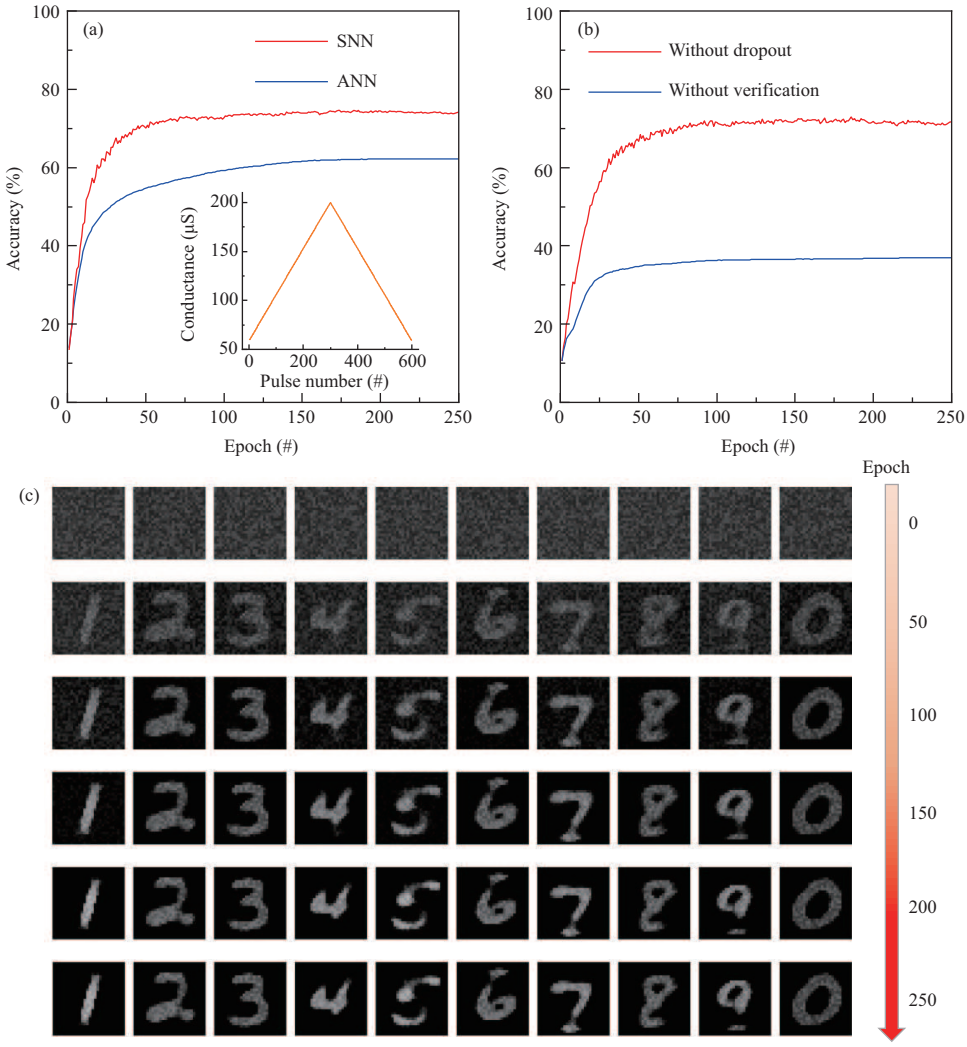

Figure 3 (Color online) (a) Classification results of the SNN (74.4%) and ANN (62.4%) trained by 200 samples randomly selected from the MNIST dataset. The inset shows the electrical property of the ideal non-volatile memristive device used in the SNN. (b) Classification results of the SNN trained without using the dropout strategy (68.91%) and the network verification strategy (36.95%). (c) Conductance updating of the memristive devices in the synaptic array when training the SNN.

A total of 200 samples with 20 samples in each class were randomly selected from the MNIST dataset to train the SNN by the STDP learning rule mentioned above. After each training epoch, 10000 testing samples were fed into the SNN to acquire the testing accuracy to indicate the training efforts. As shown in Figure 3(a), the SNN acquires a testing accuracy of 74.4% after 250 training epochs. However, an ANN built by TensorFlow 2.2 with similar structure and training conditions only acquired a testing accuracy of 62.4%. The details of the ANN are shown in Appendix C. Therefore, the SNN trained by the STDP learning rule has a better generalization ability than the back-propagation (BP)-based ANN in the case of a limited training dataset (200 samples).

Figure 3(b) demonstrates the ability of the dropout and network verification strategies. When training the SNN without the dropout strategy, the SNN only acquired a testing accuracy of 69.92%; when training the SNN without the network verification strategy, the testing accuracy of the SNN was lower than 40%. These results demonstrate that the proposed optimization strategies significantly improve the SNN performance.

Figure 3(c) shows the conductance updating of the memristive devices in the synaptic array when training the SNN. After 250 training epochs, the visualized synaptic weights gradually showed the images of the inputted digits. For comparison, the conductance updating of the memristive devices in the synaptic array when training the SNN without the network verification strategy is given in Appendix D. Afterward, during the testing period, the output neuron connected to the synapses, whose visualized weights show

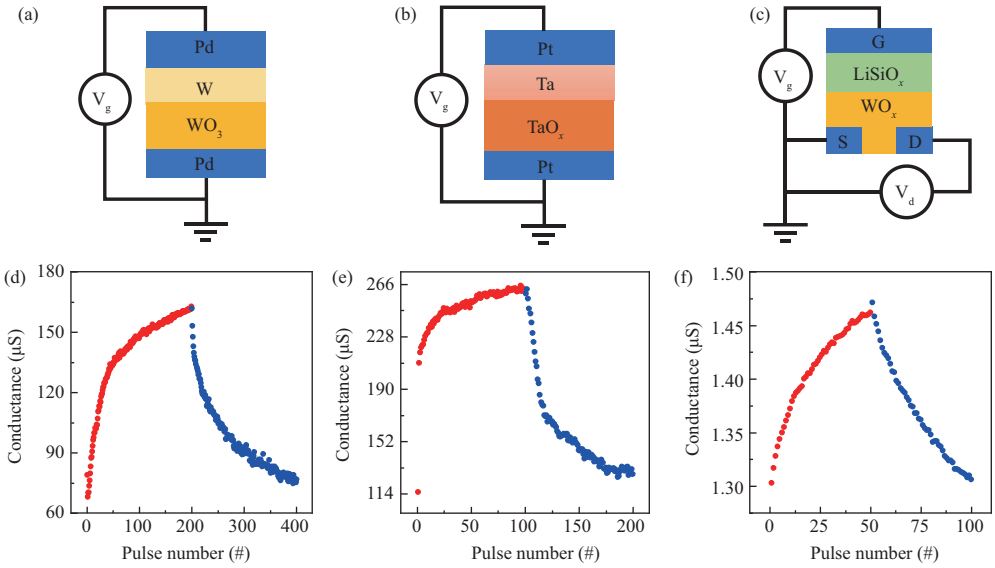

Figure 4 (Color online) Structures of the memristive devices of (a) Pd/W/WO_3/Pd, (b) Pt/Ta/TaO_x/Pt, and (c) Pt/$LiSiO_x$/WO_x/Pt. Electrical properties of the memristive devices of (d) Pd/W/WO_3/Pd (asymmetric ratio: 0.65; on/off ratio: 2.39; the number of conductance states: 200), (e) Pt/Ta/TaO_x/Pt (asymmetric ratio: 0.80; on/off ratio: 2.30; the number of conductance states: 120), and (f) Pt/$LiSiO_x$/WO_x/Pt (asymmetric ratio: 0.33; on/off ratio: 1.15; the number of conductance states: 61).

the image of a digit, quickly fires when inputting the corresponding digit into the SNN. For example, when the digit "1" is input, the input neurons first process the input signals and generate spike trains, then the spike trains are multiplied by the conductance states of the synaptic devices, the synapses connecting to the first output neuron produce a larger current, and then the first output neuron fires severely, which indicates the classification result.

2.4 Influence of imperfect device properties on the SNN performance

Real memristive devices are never perfect; imperfect or non-ideal device properties, such as the asymmetric update (AR) [49,50], small on/off ratio [49], and small number of conductance states (NCS) [49], affect the performances of SNNs.

The asymmetric ratio is used to describe the symmetry of conductance updating in the training period. The definition of the asymmetric ratio is provided in Appendix E. When the asymmetric ratio is equal to 0, the device is identical to an ideal device shown in the inset of Figure 3(a). The on/off ratio reflects the conductance range of a memristive device, which determines the precision of mapping the weights in the algorithm to the device conductance [49]. The number of conductance states reflects the number of analog states of a memristive device [49], which determines the weight tuning precision. The definition of the number of conductance states is provided in Appendix E.

To demonstrate the effects of imperfect properties, three non-volatile memristive devices, i.e., Pd/W/WO_3/Pd (AR: 0.65; on/off ratio: 2.39; NCS: 200), Pt/Ta/TaO_x/Pt (AR: 0.80; on/off ratio: 2.30; NCS: 120), and Pt/WO_3/$LiSiO_x$/Pt (AR: 0.33; on/off ratio: 1.15; NCS: 61), were used to test the SNN performance. The structures and electrical properties of the three devices are shown in Figure 4. Compared with an ideal memristive device, the three real devices showed property defects. For example, the Pt/Ta/TaO_x/Pt device showed a high asymmetric ratio (0.80), and the Pt/WO_3/$LiSiO_x$/Pt device exhibited a limited number of conductance states (61) and a small on/off ratio (1.15). Among the three devices, the Pd/W/WO_3/Pd device has a relatively balanced property.

Figure 5(a) shows the classification results of the SNNs based on the three memristive devices. Among them, the highest accuracy of 71.74% was acquired by the SNN based on the Pd/W/WO_3/Pd device, which is pretty close to the classification accuracy acquired by the SNN based on the ideal device. However, the SNNs based on the Pt/Ta/TaO_x/Pt and Pt/WO_3/$LiSiO_x$/Pt devices acquired classification accuracies of 69.30% and 62.14%, respectively. Furthermore, the classification accuracy of the SNNs based on the three memristive devices when training with 3000 samples is provided in Appendix F. The classification accuracy was improved when the number of samples was increased to 3000.

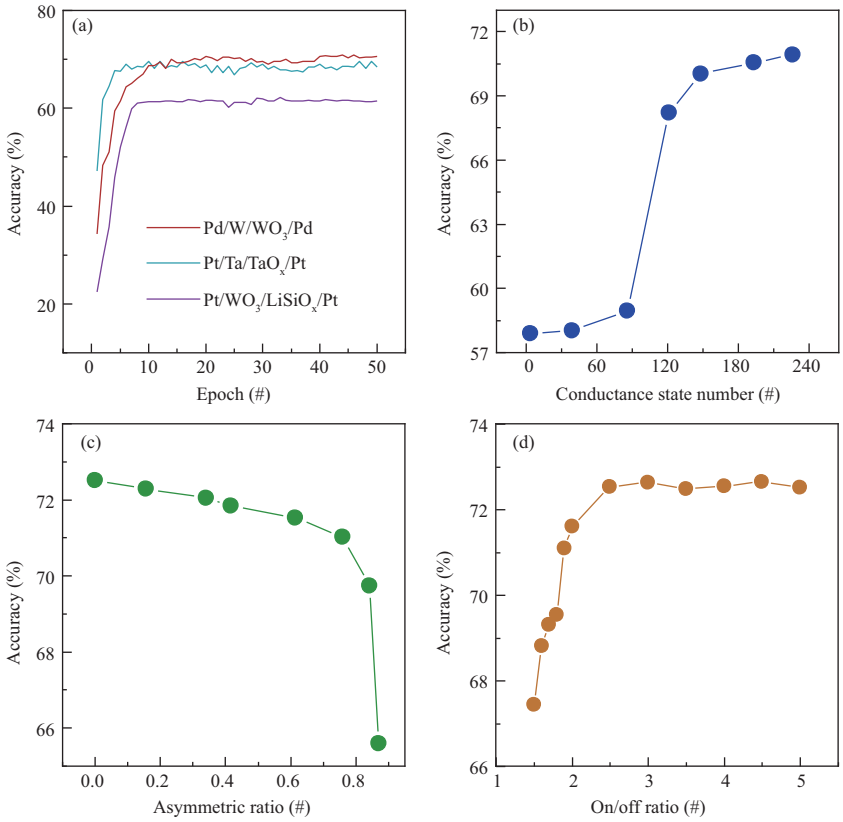

Figure 5 (Color online) (a) Classification accuracy of the SNNs based on the three memristive devices. The SNN based on the device Pd/W/WO_3/Pd acquires the highest classification accuracy (71.74%). Classification accuracy of the SNN based on memristive devices with different (b) numbers of conductance states, (c) asymmetric ratios, and (d) on/off ratios.

The SNN based on the Pt/Ta/TaO_x/Pt device exhibited the highest speed of convergence. The SNN acquired a classification accuracy of 67.75% in only four training epochs. By contrast, the SNN based on the Pt/WO_3/$LiSiO_x$/Pt device exhibited the lowest learning rate, and the SNN needed nine training epochs to acquire a classification accuracy of 61.14%. The learning rate of the SNN is partially determined by the asymmetric weight update. For example, when training the SNN based on the Pt/Ta/TaO_x/Pt device that has the highest asymmetric ratio, a positive spike causes a dramatic synaptic weight change, which leads to quick learning. However, such an asymmetric weight update also reduces the SNN precision. The performance of the SNN also relies on other device properties, such as the number of conductance states and on/off ratio. For example, the Pt/WO_3/$LiSiO_x$/Pt device has the fewest number of conductance states and lowest on/off ratio, and the corresponding SNN has the lowest classification accuracy.

To further explore the influence of imperfect properties of memristive devices on the SNN performance, models of the Pd/W/WO_3/Pd device with different properties were constructed. The details for modeling the Pd/W/WO_3/Pd device are exhibited in Appendix G.

To analyze the influence of the number of conductance states on the SNN performance, models with different numbers of conductance states but the same asymmetric ratio were built. Figure 5(b) shows the classification results of the SNNs based on different device models. When the number of conductance states increased from 148 to 226, the classification accuracy only increased from 71.04% to 71.93%. Therefore, when the number of conductance states is higher than 148, increasing the conductance state number has little effect on improving the classification accuracy of the SNN. However, when the number of conductance states is less than 148, the classification accuracy of the SNN dramatically drops. To be specific, when the numbers of conductance states were 121 and 86, the classification accuracy of the SNN dropped to 68.2% and 58.96%, respectively.

To analyze the influence of the asymmetric ratio on the SNN performance, device models with different asymmetric ratios (0.87, 0.84, 0.76, 0.61, 0.42, 0.34, 0.16, and 0) were constructed. Figure 5(c) shows the classification results of the SNNs based on different device models. Before the asymmetric ratio increased to 0.76, the classification accuracy of the SNNs decreased only slightly. However, when the asymmetric

ratio exceeded 0.84, the classification accuracy significantly dropped.

To analyze the influence of the number of conductance states on the SNN performance, device models with different on/off ratios were constructed by fixing the lowest conductance (60 μS) and modifying the highest conductance. Figure 5(d) shows the classification results of the SNNs based on the different devices. When the on/off ratio is lower than 2.5, the classification accuracy is low. However, when the on/off ratio is higher than 2.5, it has little effect in improving the classification accuracy of the SNNs.

The classification results of the SNNs based on different device models clearly demonstrate the effects of the imperfect properties of memristive devices. Among the three non-ideal properties of memristive devices, the number of conductance states affects the accuracy of the SNN most significantly and, therefore, should be given high priority among all the device properties.

3 Conclusion

In this work, we implemented fully connected SNNs based on memristive devices and designed a simplified supervised STDP learning rule for network training. Moreover, the dropout strategy and network verification strategy were applied to improve the generalization ability of SNNs. A classification accuracy of 74.4% was achieved by the SNN based on the ideal memristive device. Then, three real devices, i.e., Pd/W/WO_3/Pd, Pt/Ta/TaO_x/Pt, and Pt/WO_3/$LiSiO_x$/Pt, were used to construct SNNs, which acquired classification accuracies of 71.74%, 69.30%, and 62.14%, respectively. Imperfect properties of memristive devices, such as the small number of conductance states, high asymmetric ratio, and small on/off ratio, were analyzed by building different device models. Among the three non-ideal properties of memristive devices, the number of conductance states affects the accuracy of SNNs most significantly. By connecting such SNNs with vision sensors, artificial vision systems can be realized and applied in edge devices to process sensory signals in real time.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2018YFE0203802) and National Natural Science Foundation of China (Grant No. 61971202).

Supporting information Appendixes A–G. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Sun X, Qian H N, Xiong Y L, et al. Deep learning-enabled mobile application for efficient and robust herb image recognition. Sci Rep, 2022, 12: 6579
2 Ouyang W, Zeng X, Wang X, et al. DeepID-Net: object detection with deformable part based convolutional neural networks. IEEE Trans Pattern Anal Mach Intell, 2016, 39: 1320–1334
3 Debnath T, Reza M M, Rahman A, et al. Four-layer ConvNet to facial emotion recognition with minimal epochs and the significance of data diversity. Sci Rep, 2022, 12: 6991
4 Doulamis N. Adaptable deep learning structures for object labeling/tracking under dynamic visual environments. Multimed Tools Appl, 2018, 77: 9651–9689
5 Pereira T D, Tabris N, Matsliah A, et al. SLEAP: a deep learning system for multi-animal pose tracking. Nat Methods, 2022, 19: 486–495
6 Lin L, Wang K, Zuo W, et al. A deep structured model with radius-margin bound for 3D human activity recognition. Int J Comput Vis, 2016, 118: 256–273
7 Chen C, Zhao X, Wang J, et al. Dynamic graph convolutional network for assembly behavior recognition based on attention mechanism and multi-scale feature fusion. Sci Rep, 2022, 12: 7394
8 Al-Hasan S, Vachtsevanos G. Intelligent route planning for fast autonomous vehicles operating in a large natural terrain. Robot Auton Syst, 2002, 40: 1–24
9 Grigorescu S, Trasnea B, Cocias T, et al. A survey of deep learning techniques for autonomous driving. J Field Robot, 2020, 37: 362–386
10 Zupan J. Introduction to artificial neural network (ANN) methods: what they are and how to use them. Acta Chim Slov, 1994, 41: 327–327
11 Ma S, Wu T, Chen X, et al. An artificial neural network chip based on two-dimensional semiconductor. Sci Bull, 2022, 67: 270–277
12 Deng L, Wu Y, Hu X, et al. Rethinking the performance comparison between SNNS and ANNS. Neural Networks, 2020, 121: 294–307
13 Wang S, Zhang D W, Zhou P. Two-dimensional materials for synaptic electronics and neuromorphic systems. Sci Bull, 2019, 64: 1056–1066
14 Zou X Q, Xu S, Chen X M, et al. Breaking the von Neumann bottleneck: architecture-level processing-in-memory technology. Sci China Inf Sci, 2021, 64: 160404
15 Ma W, Zidan M A, Lu W D. Neuromorphic computing with memristive devices. Sci China Inf Sci, 2018, 61: 060422
16 Shi W, Dustdar S. The promise of edge computing. Computer, 2016, 49: 78–81
17 Liao F, Zhou Z, Kim B J, et al. Bioinspired in-sensor visual adaptation for accurate perception. Nat Electron, 2022, 5: 84–91
18 Sagi D, Julesz B. "Where" and "What" in vision. Science, 1985, 228: 1217–1219
19 Grossi E, Buscema M. Introduction to artificial neural networks. Eur J Gastroenterol Hepatol, 2007, 19: 1046–1054

20 Sengupta B, Stemmler M B. Power consumption during neuronal computation. Proc IEEE, 2014, 102: 738–750
21 Maass W. Networks of spiking neurons: the third generation of neural network models. Neural Networks, 1997, 10: 1659–1671
22 Taherkhani A, Belatreche A, Li Y, et al. A review of learning in biologically plausible spiking neural networks. Neural Networks, 2020, 122: 253–272
23 Xu Y, Zeng X, Han L, et al. A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks. Neural Networks, 2013, 43: 99–113
24 Zhang X, Lu J, Wang Z, et al. Hybrid memristor-CMOS neurons for in-situ learning in fully hardware memristive spiking neural networks. Sci Bull, 2021, 66: 1624–1633
25 Midya R, Wang Z, Asapu S, et al. Artificial neural network (ANN) to spiking neural network (SNN) converters based on diffusive memristors. Adv Electron Mater, 2019, 5: 1900060
26 Schultz S K. Principles of Neural Science. New York: McGraw-hill, 2001. 662–662
27 Benjamin B V, Gao P, McQuinn E, et al. Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. Proc IEEE, 2014, 102: 699–716
28 Furber S B, Galluppi F, Temple S, et al. The SpiNNaker project. Proc IEEE, 2014, 102: 652–665
29 Pei J, Deng L, Song S, et al. Towards artificial general intelligence with hybrid Tianjic chip architecture. Nature, 2019, 572: 106–111
30 Merolla P A, Arthur J V, Alvarez-Icaza R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. Science, 2014, 345: 668–673
31 Imam N, Cleland T A. Rapid online learning and robust recall in a neuromorphic olfactory circuit. Nat Mach Intell, 2020, 2: 181–191
32 Pan C, Wang C Y, Liang S J, et al. Reconfigurable logic and neuromorphic circuits based on electrically tunable two-dimensional homojunctions. Nat Electron, 2020, 3: 383–390
33 Rachmuth G, Poon C S. Transistor analogs of emergent iono-neuronal dynamics. HFSP J, 2008, 2: 156–166
34 Sun H, Liu Q, Li C, et al. Direct observation of conversion between threshold switching and memory switching induced by conductive filament morphology. Adv Funct Mater, 2014, 24: 5679–5686
35 Zhang Z, Wang Z, Shi T, et al. Memory materials and devices: from concept to application. InfoMat, 2020, 2: 261–290
36 Wang T, Huang H M, Wang X X, et al. An artificial olfactory inference system based on memristive devices. InfoMat, 2021, 3: 804–813
37 Pickett M D, Medeiros-Ribeiro G, Williams R S. A scalable neuristor built with Mott memristors. Nat Mater, 2013, 12: 114–117
38 Chen B, Wang X, Gao B, et al. Highly compact (4F2) and well behaved nano-pillar transistor controlled resistive switching cell for neuromorphic system application. Sci Rep, 2014, 4: 6863
39 Ravichandran V, Li C, Banagozar A, et al. Artificial neural networks based on memristive devices. Sci China Inf Sci, 2018, 61: 060423
40 Kang S M, Choi D, Eshraghian J K, et al. How to build a memristive integrate-and-fire model for spiking neuronal signal generation. IEEE Trans Circuits Syst I, 2021, 68: 4837–4850
41 Huang H M, Yang R, Tan Z H, et al. Quasi-Hodgkin-Huxley neurons with leaky integrate-and-fire functions physically realized with memristive devices. Adv Mater, 2019, 31: 1803849
42 Hu X, Liu C. Dynamic property analysis and circuit implementation of simplified memristive Hodgkin-Huxley neuron model. Nonlinear Dyn, 2019, 97: 1721–1733
43 Abbott L F. Lapicque's introduction of the integrate-and-fire model neuron (1907). Brain Res Bull, 1999, 50: 303–304
44 Wang W, Pedretti G, Milo V, et al. Learning of spatiotemporal patterns in a spiking neural network with resistive switching synapses. Sci Adv, 2018, 4: eaat4752
45 Wen J, Huang H M, Wang Z, et al. Neuromorphic systems based on ionic memristors: from materials, devices to chips. Chin Sci Bull, 2022, 67: 1054–1071
46 Mehonic A, Sebastian A, Rajendran B, et al. Memristors—from in-memory computing, deep learning acceleration, and spiking neural networks to the future of neuromorphic and bio-inspired computing. Adv Intell Syst, 2020, 2: 2000085
47 Zhou Y, Wang Y, Zhuge F, et al. A reconfigurable two-WSe_2-transistor synaptic cell for reinforcement learning. Adv Mater, 2022, 34: 2107754
48 Moon J, Wu Y T, Zhu X J, et al. Neural connectivity inference with spike-timing dependent plasticity network. Sci China Inf Sci, 2021, 64: 160405
49 Zhang W, Gao B, Tang J, et al. Neuro-inspired computing chips. Nat Electron, 2020, 3: 371–382
50 Yang C S, Shang D S, Liu N, et al. All-solid-state synaptic transistor with ultralow conductance for neuromorphic computing. Adv Funct Mater, 2018, 28: 1804170