Memristive devices based hardware for unlabeled data processing

Zhuojian Xiao1, Bonan Yan2, Teng Zhang1, Ru Huang1,2,* and Yuchao Yang1,3,4,*

1 Key Laboratory of Microelectronic Devices and Circuits (MOE), School of Integrated Circuits, Peking University, Beijing 100871, People's Republic of China
2 Center for Brain Inspired Chips, Institute for Artificial Intelligence, Peking University, Beijing 100871, People's Republic of China
3 Center for Brain Inspired Intelligence, Chinese Institute for Brain Research (CIBR), Beijing, Beijing 102206, People's Republic of China
4 Beijing Academy of Artificial Intelligence, Beijing 100084, People's Republic of China
* Authors to whom any correspondence should be addressed.
E-mail: ruhuang@pku.edu.cn and yuchaoyang@pku.edu.cn

Keywords: memristive devices, neuromorphic, unlabeled data processing

Abstract
Unlabeled data processing is of great significance for artificial intelligence (AI), since well-structured labeled data are scarce in a majority of practical applications due to the high cost of human annotation of labeling data. Therefore, automatic analysis of unlabeled datasets is important, and relevant algorithms for processing unlabeled data, such as $k$-means clustering, restricted Boltzmann machine and locally competitive algorithms etc, play a critical role in the development of AI techniques. Memristive devices offer potential for power and time efficient implementation of unlabeled data processing due to their unique properties in neuromorphic and in-memory computing. This review provides an overview of the design principles and applications of memristive devices for various unlabeled data processing and cognitive AI tasks.

1. Introduction

The enormous success of artificial intelligence (AI) in recent years is inextricably linked to the explosive growth of data and rapidly evolving computing hardware for better performance [1–4]. The main task of AI research is to find efficient ways to extract information and knowledge from the huge and continuously increasing amount of available data. When the data are labeled (i.e. one has a set of input variables and corresponding output variables), the algorithms make predictions by adjusting the model parameters relying on the priori labels from the training dataset. Such supervised learning approach has achieved amazing performance in accuracy, sometimes even outperforming human [5]. However, the scarcity of labeled data and the high cost of manually labeling data arouse unlabeled data processing problems in the most common applications.

High-dimensional unlabeled data processing is a more general technology toward realistic vast amounts of data [6–8]. It exists in various fields, such as medicine [9, 10], bioinformatics [11–13], biology [14–16], computer science [17, 18] and multiple engineering fields [19, 20]. Aimed for different applications, many models for various unlabeled data processing have been developed, including autoencoders and sparse-coding for feature extraction [21, 22], $k$-means for classification [23, 24], Hopfield networks, and restricted Boltzmann machines for optimization problems [25, 26]. These models offer many fascinating abilities to AI systems, such as getting insights from large volumes of raw data, discovering inherent unknown patterns and features structure, and adapting to any change in the external world in a cognitive way. Some models even hold the potential to outperform supervised models in accuracy, generalization ability, and robustness [27, 28].

Accessed from the perspective of computing, unlabeled data processing models contain dominating complex computing-intensive operations such as vector matrix multiplications (VMMs) and locally competitive algorithms (LCAs) [29]. Traditional scalar von Neumann architecture computing systems based on transistor electronics are not optimized for these computing tasks [30]. Existing unlabeled data processing models implemented on the conventional hardware face a high design overhead in speed, power, and area [31, 32].
Therefore, it is a vital task to develop new nanoscale devices beyond CMOS and novel hardware architectures to achieve high-efficiency unlabeled data processing.

To achieve this, memristive devices are under intense investigation, including resistive random access memory (RRAM) [33–35], phase-change memory (PCM) [36–38], ferroelectric devices [39, 40], ion-gating [41–43], and charge-trap transistors [44–46], as well as spintronic devices [47, 48]. They have a unique characteristic that their internal resistance (or conductance) state is determined by the history of applied voltages and currents. These nanoscale devices feature good scalability, stacking-ability, and other intriguing properties that exceed conventional integrated circuit technology [49–54]. Besides, there exists a high similarity between the variable synaptic strengths of biological synapses and tunable internal resistance states of memristive devices. Memristive device crossbar arrays are thereby power-efficient and fast in the calculation of VMMs and LCAs. Memristive devices offer the fascinating possibility of in-memory computing and neuromorphic computing, thus bypassing the restrictions of computing speed as well as the power consumption of the von Neumann architecture because of the distanced memory and processing units [55–58].

Therefore, as shown in figure 1(a), realizing unlabeled data processing models with the memristive device is a promising way to low-power computing systems. In this paper, we provide a high-level overview of the design principles and the methods for building energy-efficient memristive device-based computing hardware. These memristive AI hardware process various input data such as digital, analog, and spiking efficiently, realizing cognitive AI tasks, such as recognition and classification tasks in computer vision [59, 60], healthcare [61, 62], and neuromorphic computing [63], as shown in figures (b) and (c). The memristive hardware is described according to different unlabeled data processing models. These unlabeled data processing models include:

(a) Classic machine learning algorithms: principal component analysis, sparse coding, and $k$-means clustering.
(b) Dynamic neural networks: Hopfield network and restricted Boltzmann machine.
(c) Generative models: autoencoders and generative adversarial network (GAN).
(d) Spiking neural networks: the third generation of artificial neural network models.

2. Principal component analysis

Principal component analysis (PCA) is an important traditional machine learning algorithm for unlabeled data processing. By projecting the high-dimensional unlabeled data into lower dimensions principal components
Figure 2. (a) PCA geometrically projects 2D points onto 1D lines. (b) $\sigma^2$ of the projections for the same set of points can vary with different lines. Reproduced from [64], with permission from Springer Nature. (c) Schematic of PCA neural network implemented the memristor crossbar hardware. The conductance of the memristor devices in the column correspond to PCs after training. (d) Simulated PCA results using the memristive computing system. Reproduced from [66], CC BY 4.0. (e) SEM image of the Ta$_2$O$_5$-based memristor array. Inset: structure of the memristor cell. (f) Optical image of PCA computing system. Reprinted with permission from [70]. (g) cross-section of the WO$_x$ memristor/CMOS integrated chip and the WO$_x$ memristor. (h) Circuit diagram of the peripheral circuits for the crossbar. (i) Optical image of integrated memristor/CMOS chip. Insert: the testing system for the memristor/CMOS integrated chip. Reproduced from [71], with permission from Springer Nature.

(PCs), PCA finds the best representations of the unlabeled data using a finite number of PCs. As shown in figure 2(a), by minimizing the distance between the input unlabeled data and their projection, the first PC is chosen. Meanwhile, the variance ($\sigma^2$) of the projected points is also maximized, as shown in figure 2(b). The second and subsequent PCs are extracted in the same way, with another requirement that they must be orthogonal to all previous PCs. PCA reduces data dimensionality while retaining models and trends so that it is widely applied in machine learning applications, such as facial recognition, image denoising, analyzing genetic data, and disease predictions [64].

However, the traditional way for PCs is to resolve the eigenvectors of the covariance matrix of the high-dimensional input data, which is precise but compute-intensive. Nowadays, a more hardware-friendly method with low computation cost is developed. This new approach finds the approximate PCs through unsupervised, online learning of PCA neural network, which can be implemented very efficiently by the memristive crossbar. To be specific, Sanger’s rule is often utilized in PCA neural networks, as shown in equation (1) [65]:

$$\Delta g_{ij} = \eta y_j \left( x_i - \sum_{k=1}^{j} g_{ik} y_k \right)$$  \hspace{1cm} (1)$$

where $i$ is the row number, $j, k$ is the column number of crossbars, $x$ and $y$ are the input and output, respectively, $g_{ij}$ is the weight, and $\eta$ is the learning rate. The input data are encoded into voltage pulses with different widths, and the weights of columns after the training process correspond to PCs one by one.

The first memristive PCA neural network is proposed based on Ta$_2$O$_5$ memristor crossbar array, as shown in figure 2(c). The crossbar array structure can efficiently perform the matrix operation and the tantalum-oxide memristor can implement on-chip in situ learning with simple programming pulses [66]. Therefore, unsupervised, on-chip learning of PCA neural networks is trained under Sanger’s learning rule to obtain the PCs.
shown in figure 2(d), the experiments show that even using memristors with non-idealities, the memristive PCA network can linearly separate different classes from breast cancer data with an excellent success rate of 97.6%.

However, the as-prepared TaOₓ-based memristor needs an initial forming process, where a high voltage of ∼5 V (the typical value of the TaOₓ-based memristor set voltage is ∼1 V) is applied to get the ionic distributions, which are vital for subsequent resistive switching behaviors. When the high forming voltage is applied to form a device, the already formed devices that share the same row can be damaged. To solve this problem, a forming-free TaOₓ-based memristor is developed by using Ta₂O₅ layer as the switching layer and Ta metal as the reactive top electrode, as shown in figure 2(e). A rapid thermal annealing process after device fabrication to create oxygen vacancies in the Ta₂O₅ switching layer is responsible for forming-free behavior. Similar results have also been reported in other forming-free metal-oxide memristors [67–69]. The device is utilized for the successful implementations of a practical memristor crossbar-based PCA neural network computing system. As shown in figure 2(f), the computing system includes a CMOS chip as a microcontroller and a test board in which the wire-bonded memristor crossbar array is connected to the peripheral circuitry [70]. The memristive PCA neural network hardware reached 97.1% accuracy in the classification task on standard breast cancer data even using devices with large variations, closing to results of software solving (97.6%).

As for the previously mentioned implementations of PCA neural network, an external printed-circuit board is needed to provide the interface and control circuitry for memristor arrays. Therefore, integrating all these functions on chip will be a big step forward to the practical use of memristive computing systems by enabling the whole system to be scaled. WOₓ memristor is a good choice to fabricate integrated memristor/CMOS system because it is compatible with the 180 nm logic process and can be fabricated on top of the CMOS circuits using back-end-of-line (BEOL) process as shown in figure 2(g). Therefore, in a single memristor/CMOS chip, WOₓ memristor crossbar array can be integrated with peripheral circuits consisting of analog-to-digital converters/digital-to-analog converters (ADCs/DACs), digital buses, and a programmable processor, as shown in figures 2(h) and (i). With the integrated WOₓ memristor/CMOS system, PCA neural network and the following classification layer are implemented [71]. Compared with software PCA programs, this memristive hardware system can achieve similar classification accuracy with a high power efficiency (1.37 TOPS/W), which potentially endows PCA on large amount of data.

3. Sparse coding

An interesting fact is that only a small subset of elements from a dictionary is needed to well approximate a natural image. The problem of selecting a proper subset of dictionary elements and their coefficients to reconstruct a signal is called sparse approximation. Sparse coding originally refers to the intriguing hypothesis that biological neural systems exhibit similar sparse representations with their population codes, encoding stimulus information only in a few active neurons. For example, the significant sensitivity in the receptive field of primary visual cortex cells can be explained by sparse coding. Within the receptive field, a single neuron is coded to the stimuli, such as edges, stripes, line segments, as well as other image features in a certain direction. The cost function of sparse coding can be mathematically expressed as:

\[
E = \min_{\mathbf{a}} \left( |\mathbf{x} - \mathbf{D} \mathbf{a} |_2^2 + \lambda |\mathbf{a}|_0 \right)
\]

where \(x\) is the input signal, \(a\) is a vector of sparse coefficients with only a few non-zero elements, \(D\) is a dictionary of features, \(|\cdot|_2\) and \(|\cdot|_0\) are the \(L^2\) and \(L^0\)-norm, respectively, and \(\lambda\) represents the sparsity parameter.

Sparse coding described by equation (2) is a complex nonconvex optimization problem. Fortunately, a convenient unsupervised machine learning algorithm, the LCA, is developed to solve this sparse coding problem. LCA consists of several nonlinear ordinary differential equations which depict the dynamics of the membrane potential and the firing rate of interacting neurons. These neurons continually compete with neurons nearby using lateral inhibition to compute coefficients of input data using a complete dictionary. The LCA significantly eases the hardware implementation of sparse coding in that the used computational primitives can be realized by analog circuit elements. Thus, using memristive devices to implement LCAs also can be high speed and energy efficient.

A detailed description of how memristor arrays can be used to implement the LCA is proposed by Lu et al [72]. In this implementation, an analog WOₓ memristor crossbar array structure is employed in which VMM and matrix transpose can be easily performed. Input \(x\) is an \(m\)-element column vector which is coded into the width of voltage pulses. These pulses are applied to the rows of the array simultaneously. The amplitude of the voltage is carefully chosen to be high enough to guarantee an adequate read-out margin, and not so high to switch the resistance states. Memristors in a given column serve as an \(m\)-element vector representing a feature.
$N$ feature vectors form an $m \times n$ matrix, the feature dictionary $D$. A leaky-integrate and fire (LIF) neuron is connected to each column of $D$ as output. The activity coefficients are represented by an $n$-element row vector $a$, in which the $i$th element of $a$ is the sparse coefficients of the corresponding feature. Equation (3) shows the dynamics of such a process:

$$\frac{du}{dx} = \frac{1}{\tau} (-u + (x - \hat{x})^T D + a) \quad (3)$$

where $u_i$ represents the membrane potential of the $i$th neuron, $\tau$ and $\lambda$ represents time constant and threshold, respectively. After sending the original input $x$ into the network, a reconstruction of $x$ can be obtained as $\hat{x} = Da^T$. As shown in figure 3(a), the difference between the reconstructed signal and the input is the residual. Lateral neuron inhibition is realized via an iterative strategy by feeding the residual to the network as the new inputs [65]. In this ping-pong way, the LCA network settles to a steady state and the vector $a$ representing neuron activity no longer changes, thus the cost function described by equation (2) is minimized. Thus, a sparse representation with minimum reconstruction error and high sparsity is found in a memristive computing system.

Compared with CMOS implementation of sparse coding, the stimulated memristor crossbar sparse coding computing architecture proves to be more efficient by offering a saving of $154 \times$ and $\sim 2\times$ for area and energy,
Figure 4. (a) Ta$_2$O$_x$ memristor crossbar array used for k-means algorithm. (b) Schematic of the method to map the k-means algorithm onto Ta$_2$O$_x$ memristor hardware. (c) Experimental obtained results of the memristor-based computing system. Reprinted with permission from [75]. Copyright © 2018, American Chemical Society. (d) Structure and SEM image of the graphene memristor device. (e) Schematic of the architecture for operating VMM. (f) Output current ($I_{\text{out}}$) obtained experimentally and the error estimated in theory. Reproduced from [76]. CC BY 4.0.

respectively. The distortion of the reconstructed signal is comparable with the result obtained by software at high sparsity, even implemented by memristor with large variations [72]. Following this idea, the prototype system of sparse coding algorithms is implemented by a 32 × 32 WO$_x$ memristors crossbar array (figure 3(b)) [73]. According to the nature of the input images, various feature dictionaries can be learned on-chip and stored in this crossbar. As shown in figures 3(c) and (d), data-intensive tasks such as natural image processing and real-time video analysis showcase the great prospect for piratical unsupervised AI tasks [71]. Based on the threshold adaptive memristor model, a sparse coding algorithm named memristive neural network-based soft-threshold adaptive sparse coding (MMN-SLCA) is proposed. The tuneable thresholds further improve the flexibility of intelligent sparse coding. The MMN-SLCA computing system reduces the dimensional and complexity of the original input unlabeled signals and provides enhanced feature extraction, pattern recognition, and super-resolution reconstruction functions [74].

4. K-means clustering

Clustering analysis is a basic unsupervised learning method that refers to the classification of a set of signals such as data points, feature vectors, or observations into meaningful groups with no predefined labels. Among different clustering analysis methods, K-means clustering is the prominent one and is applied in numerous fields of science and engineering such as gene expression analysis and document clustering [70]. The basic principle of k-means is quite simple. Before classification, the k centroids are initialized randomly; first, each input data point is classified to its nearest cluster by computing the distances between the input unlabeled data point and the k centroids; second, update the coordinates of k centroids with the new clusters; third, the algorithm repeats two steps until convergence is reached.

The k-means clustering algorithm can be implemented efficiently in memristive devices crossbars by representing weight vectors as conductance. As shown in figure 4(a), crossbar arrays of Ta$_2$O$_x$ memristor with a low forming voltage and ideal analog programmable conductance is used for the implementation. Figure 4(b) depicts the implementation of the k-means algorithm with the Ta$_2$O$_x$ memristor crossbar array. Before classification, the coordinates of the centroids are interpreted as the conductance matrix of the memristors crossbar. The coordinates of input data points are encoded with pulse width modulation and the resulting voltage pulses are applied to the rows of the crossbar. By adding a row to introduce the W$^2$ term and comparing the accumulated charges in the columns, the closest centroids of the corresponding input data point can be found. The experimental system reaches high classification accuracy (93.3%), approximately the same as the result directly obtained from software (95.3%) for the standard IRIS data set (figure 4(c)) [75].

Besides Ta$_2$O$_x$ memristor, memristive implementation of k-means is also demonstrated by graphene field effect transistor (GFET) devices with programmable conductance, as shown in figure 4(d). VMM operations are executed by the memory architecture shown in figure 4(e). For a given back-gate voltage ($V_{BG}$), the product of the matrix represented by GFET device’s conductance and the vector encoded by input voltage amplitude is
given spontaneously by the output current. The results of output currents in experiment and theory are very close, as shown in figure 4(f). Because GFETs allow reliable programming to specific conductance states, which improves the computing accuracy of \( k \)-means clustering [76].

5. Hopfield network

Hopfield network is an asynchronous, recursive, and dynamic artificial neural network (ANN) proposed by Hopfield in the 1980s and is considered as a kind of constrained optimization network that minimizes certain preprogrammed energy functions. With a simple structure of several interconnected neurons, the nonlinear Hopfield networks have proved extremely useful in unlabeled signal processing such as pattern recognition, data compression, combinatorial optimization problem, as well as location allocation problems [77, 78].

The dynamics and energy function are crucial to understanding how the Hopfield networks work. We take a binary discrete-time asynchronous model as an example because this model operates based on a simple mechanism and is often adopted in practical applications. For binary discrete-time asynchronous model consisting of \( N \) fully connected binary neurons, only one randomly-selected neuron update in one step and the dynamics governing the system evolution is described by equation (4):

\[
U_j (t + 1) = f \left( \sum_{i=1}^{N} \omega_{ij} (t) U_i (t) + T^b_j \right)
\]

where \( U_j (t) \) represents the state of the neuron \( j \) at iteration \( t \), \( f(.) \) is the threshold function, \( \omega_{ij} (t) \) and \( T^b_j \) is the strength of synaptic weight and bias respectively.

Alternatively, the energy function also can describe the evolution of the state in Hopfield network, as given by equation (5).

\[
E(t) = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij} (t) U_i (t) U_j (t) - \sum_{i=1}^{N} T^b_j U_j (t).
\]

The energy function spontaneously decreases during the runtime. It is proved mathematically that the evolution of the nonlinear system given by equation (4) is equivalent to the energy function minimization process described by equation (5). Therefore, the network gradually converges to the attractors represented by state variables of the network including the neuron states and synaptic strength [79]. By mapping the problems into the state variables and energy function of the Hopfield network, the dynamic allows the network to solve optimization problems and realize recurrent associative memory.

The update rule of equation (4) ensures that energy function is decreasing as the network evolves in time. But the network is easy to trap into local minima (not the final goal) near the initial states. Steady states are the key mechanism for convergence but trapping in local minima is undesired for the solution of optimization problems. To avoid the local minima problem, simulated annealing is developed to get out of local minima by harnessing thermally controlled probabilistic jumps [80].

Hopfield network is realized by constructing CMOS synapses with large chip areas and high power consumption. In contrast, memristive devices provide a more efficient way to realize Hopfield networks [81, 82]. This is because the dominating computation task in Hopfield networks, VMM, can be efficiently performed by the high-density memristor crossbar [63]. Furthermore, by harnessing the inherently stochastic and nonlinearity during switching, memristive devices can significantly reduce the implementation overhead of simulated annealing.

As shown in figure 5(a), Hopfield neural network solving max-cut problems is implemented by programming binary weight matrices on 1-transistor-1-memristor (1T1R) crossbar arrays of TaOx memristors integrated with the control circuits at 180 nm technology node. 1T1R crossbar arrays enable highly efficient computations of VMM of input vectors (figure 5(b)). Intrinsic analog noise of the system includes the conductance fluctuations in the nanoscale devices and crossbar array non-ideal properties, such as IR drop. They are exerted to demonstrate the contribution to improving both solution quality and efficiency by enabling the system to escape local minima (figure 5(c)). Compared to quantum, optical, and fully digital methods, this memristor-based stochastic annealing system provides throughput over 4 orders of magnitude higher per power consumption, i.e. power efficiency [83]. Chaos from nonlinearity in TaOx memristors is also introduced to simulate annealing to develop effective memristive optimizer hardware for combinatorial/continuous function optimizations [84].

Besides harnessing the intrinsic noise and the chaotic annealing, a weight annealing strategy is proposed to overcome the local minima trap problem in the memristive Hopfield neural networks. By initializing all
synaptic weights to zero and then gradually recovering the weights by the annealing strategy, convergence to global minima can be achieved more quickly because the network keeps near its ground state during the whole process. The weight annealing is implemented by a $20 \times 20$ crossbar array of TiO$_x$ memristor and mixed-signal circuits (figure 5(d)). The pre-synaptic drivers are used to realize the scaling of the synaptic weights, which is the key to the weight annealing, as shown in figure 5(e). The numerical simulations prove that the weight annealing strategy leads to better results for several classic combinatorial problems compared to chaotic or stochastic annealing (figure 5(f)) [85].

In addition to solving combinatorial optimization problems, associative memory is another cognitive function of the Hopfield neural network by allowing the system to reconstruct the whole data upon a part of the presentation for that piece of data. As mentioned before, the trapping in local minima (attractors) is favorable in associative memory as the attractors represent the stored information. Single/multi-associative memory is experimentally prototyped in HfO$_x$ memristors [87]. The inputs of the computing system encoded into voltage pulses are also used as the outputs to train the analog memory states of memristors. Thus, different features are stored in the network, described by another set of values of the parameters for the recall, also known as the ‘synaptic weights’. A power-efficient associative memory implementation is proved by a Y-flash memristive crosswise array. The cell of the array consists of a floating gate two-terminal n-channel metal–oxide semiconductor transistor (figure 5(g)). As shown in figure 5(h), a 3 bit content-addressable memories (CAMs) Hopfield network is implemented with 12 Y-flash arrays as memristive synapses integrated with three decision-making neurons. The resulting hardware performs single/multi-associative memory (figure 5(i)) and minimal power dissipation is achieved in a winner-take-all network configuration [86]. Thanks to the power-efficient Y-flash memristive device, the memristive network achieves a low power consumption of only 3.6 $\mu$W on a testing epoch.

In the memristive associative memory system discussed earlier, the attractors of the Hopfield network are discrete. However, network attractors can be continuous by introducing a translationally invariant bell-shaped connectivity model. Continuous attractor is considered as the mechanism of working memory, namely the ability of the brain to store and maintain pertinent information and apply this information to do higher-level computations afterward. Through the simulation of continuous attractor in the memristor-based Hopfield network, working memory is achieved based on memristive device in [88], demonstrating the great potential
of the memristive device for the development of highly efficient computing hardware for cognitive unlabeled data processing AI tasks.

6. Restricted Boltzmann machine

As an unsupervised learning model, restricted Boltzmann machine (RBM) is regarded as a generalization version of the Hopfield networks. RBM is a simple neural network consisting of a hidden layer and a visible layer. Each node of the hidden layer is connected to every node of the visible layer and connections within a layer are forbidden. Just like the Hopfield network, the evolution of RBM states is determined by an energy function, as described in equation (6).

$$E(v,h) = - \sum_i c_i v_i - \sum_j d_j h_j - \sum_{ij} v_i h_j \omega_{ij}$$

where \(c_i\) and \(d_j\) are the bias weights, \(\omega_{ij}\) represents a synaptic weight connected between \(v_i\) and \(h_j\) nodes.

Different from the original Hopfield network, the RBM evolves in a stochastic mechanism. The distribution over states of RBM is given by a Boltzmann–Gibbs distribution:

$$P(v,h) = \frac{\exp(-E(v,h))}{Z}$$

$$Z = \sum_{\{v,h\}} \exp(-E(v,h))$$

However, it is difficult to exactly calculate the value of normalizing factor \(Z\) due to the terms of exponentially scaling. Fortunately, it is easy to see that the hidden nodes are conditionally independent given the values of the visible nodes by using Bayes rule (and vice versa). Therefore, a conditional distribution of RBM states can be expressed as:

$$P(h_j|v,h) = \sigma \left( \sum_i v_i \omega_{ij} + d_i \right)$$

$$P(v_i|v,h) = \sigma \left( \sum_j h_j \omega_{ij} + c_i \right)$$

where \(\sigma(.)\) represents an activation function. RBMs are universal approximators of discrete distributions. After learning, any unknown data distribution can be represented and the minima of the energy function corresponding to a sample of the unknown data set. Thus, RBMs have found a variety of applications in statistics, information theory [89], machine learning problems [90], and even physics [91].

Hardware implementation of RBM by CMOS is intolerable when scaled up due to the storage required all-to-all communication among the RBM computing process [95, 96]. To train the brain-scale neural network application, an efficient RBM system is in need. Memristive crossbar provides a good solution by integrating the computing-intensive storage operations of neuron outputs and VMMs within the same stage cycle [97, 98]. As shown in figure 6(a), RBM is implemented using a \(20 \times 20\) Pt/Al2O3/TiO2−x/Pt memristor crossbar array. By utilizing intrinsic and extrinsic current fluctuations of memristive device, efficient stochastic dot production computation given by equations (7) and (8) is performed. This is the most intensive computation during inference and training of RBM, as shown in figure 6(b). By introducing simulated annealing, the RBM is a fast and practical method to solving combinatorial optimization problems. As an example, shown in figure 6(c), the memristive RBM computing system quickly converges to thermal equilibrium and finds the solution only in 500 epochs [92].

Besides optimization problems, the memristive device with intrinsic fluctuations and stochastic can be used for pattern classifications tasks by constructing a fuzzy RBM network. The weight states in fuzzy RBM network are fuzzified following Gaussian distributions (figure 6(d)), which can be implemented naturally by stochastic Pt/TaOx/Ta memristors. The device stochasticity of the memristor originated from the cycle-to-cycle as well as device-to-device variations of the device internal resistance states (figure 6(e)). Compared with other RBM networks, the fuzzy RBM network achieves a lower error rate in handwritten digit recognition test on the MNIST database, demonstrating its enhanced tolerance ability to device stochasticity (figure 6(f)) [93].

By using RBMs as a building block to extract features, training deeper models is much easier [89], such as deep belief networks. Discriminative RBM uses both the input pictures and the associated label as
visible neurons and learned jointly. This architecture is believed to be more powerful in features extraction than the simple RBM [99]. By using the conductance of two memristive devices to represent the synaptic weights (figures 6(g) and (h)), a discriminative RBM and other RBM-based networks are simulated and compared. The result reveals that the discriminative RBM overperforms other RBM-based structures using devices with typical parameters (figure 6(i)), suggesting the great potential of memristive RBM architectures to achieve on-chip unlabeled data processing [94].

7. **Autoencoder**

Autoencoders is another generative model developed to produce representations. By using the input data as the teacher, autoencoders can transform inputs into outputs with minimal distortion. Therefore, they can extract useful information needed for solving unlabeled data processing problems, such as video recognition or anomalous data detection. Autoencoder is also a biologically-plausible mechanism. By adopting an autoencoder model, the behaviors of the sensory areas in the human visual cortex were well understood. Furthermore, autoencoders play a critical role in the deep learning architecture where bottom-up stacked autoencoders layers are frequently used followed by a supervised learning top layer [21].

Similar to other ANN, autoencoders can be implemented by memristive devices-based systems with significantly lower power consumption than the alternative conventional approaches. A memristive autoencoder computing system is designed for real-time intrusion detection and anomaly detection. The benchmark results show that the system has an overall detection accuracy of 92.91% and the malicious packet detection accuracy is 98.89% [100]. Autoencoder hardware is implemented by Cu:ZnO:Nb:STO memristor which can denoise sample imagines of the MNIST dataset [101]. By modulating the conductance of the memristive device, the STDP learning rule is obtained. Employing this learning rule, a spike-based autoencoder learns the features of the original image and is tested with images adding different kinds of noise. Since autoencoders have the ability
to precisely encode the input features, the output of tested images is denoised. The obtained results show that the recovered images achieve high accuracy even with a considerable amount of noise, proving that memristive devices play an important role in autoencoder-based unlabeled image processing and facial recognition systems.

8. Generative adversarial network

GAN is a type of unsupervised learning model that can extract feature representations and generate new patterns from unlabeled data. Generally, GAN is a complex architecture consisting of two independent neural networks with a competitive learning approach. In detail, as shown in figure 7, a generator (G) learns from the real data distribution and generates new data from the uniform random input noise to deceive a discriminator (D). The discriminator acts as a detective to discriminate fake samples from G and real ones from the training data. The G and D are trained together and influence each other. The backbone neural networks to construct G and D are still the convolutional neural networks (CNNs) [102].

The training and testing processes of GAN consist of large numbers of forward and backward passes, which are more sophisticated than deep neural networks. Thus, GAN is computationally complex and requires significant computational resources. Therefore, the implementations of GAN with conventional hardware are slow and energy inefficient, even in GPU and distributed computational networks. Therefore, realizing an efficient accelerator for GAN is necessary but challenging, especially for time-constrained object tracking applications and energy- and area-constrained IoT edge devices. Thanks to their superior properties, various memristors, and their composite circuits are excellent candidates for implementing efficient GAN. For example, a novel circuit based on a spintronic memristor crossbar is proposed for the acceleration of CNN operations (figure 8(a)) [103]. Based on the memristor-based CNN unit, a GAN architecture is developed to solve the single image super-resolution problem (figure 8(b)). As aforementioned, WO
memristors are compatible with the 180 nm logic process, and integrated memristor/CMOS chip can be fabricated using BEOL process by fabricating the memristor crossbars on top of the CMOS circuits [69, 104]. Therefore, hardware named analog memristive deep convolutional GAN (AM-DCGAN) is designed to accelerate the intensive neural computations of GAN [105]. By mapping weights to the conductance of WO memristors, convolution, deconvolution, and mean-pooling filters are realized by memristive crossbars (figure 8(c)). System- and circuit-level simulation results prove that the minimum average power consumption per neural computation is only 47 nW and cell-to-cell variability of memristor affects the performance of the generator system adversely and image quality degrades after the process variation standard deviation over 0.08 (figure 8(d)). Therefore, the AM-DCGAN system efficiently achieves high-quality image generation on the MNIST database.

Compression techniques of CNNs in software, such as quantization and pruning, can be utilized to accelerate memristive GAN architecture since the G and D units are still conventional CNNs. Through binarizing the weights for several CNN layers within both the G and D in the traditional deep convolutional GAN (DCGAN), an approximate GAN (ApGAN) algorithm and its hardware implementation for accelerating GANs are proposed [102]. As shown in figure 8(e), the architecture of ApGAN accelerator consists of image and kernel sub-arrays, external processing unit, and the computational sub-arrays based on memristor. The sub-array is composed of row decoder, column decoder, as well as sense circuitry with 512 rows and 256 columns per memory array. Ag–Si memristor device parameters are used for SPICE models at the circuit-level simulation of the memristive sub-array. Then the SPICE models of memristors and CMOS transistors are combined for system-level evaluations. As shown in figure 8(f), the images generated by ApGAN architectures
9. Spiking neural network

Biological synapses and neurons use discrete spikes to transmit and process information [106]. Inspired by biological neurons, spiking neural networks (SNNs) are developed as the third generation of artificial neuron models. Compared with second-generation DNNs in which neurons communicate using single, static, continuous-valued activations, SNNs bring superior accuracy because of the added temporal dimension and...
intrinsic sensitivity to the unstructured temporal data. This mechanism is prevalent in the biological neural system [107].

The lack of effective training methods is a challenge for SNNs because the spikes are discrete both in the time and space domain. Spiking neural transfer function is non-differentiable preventing the use of widely-used backpropagation. For now, only a few biologically-plausible local learning rules are developed to train SNNs. These unsupervised learning rules describe the two types of synaptic plasticity: long-term plasticity and short-term plasticity [108–110]. Long-term plasticity includes long-term potentiation (LTP) and long-term depression (LTD), spike timing dependent plasticity (STDP), and spike rate dependent plasticity (SRDP), while short-term plasticity contains short-term potentiation (STP), paired-pulsed facilitation (PPF) and depression (PPD). They are originally discovered in neuroscience and thought to be the underlying mechanisms for complex learning and memory functions in the human brain [111].

9.1. Long-term plasticity
STDP is the most commonly adopted learning mechanism in unsupervised training in SNN. The interpretation of biological STDP is intuitive: when the pre-neuron fires shortly before the post-neuron, the synaptic weight is strengthened (called ‘LTP’). When the pre-neuron fires shortly after the post-neuron, the synaptic weight is thereby weakened (called ‘LTD’) [59, 108]. Although standard CMOS elements have been used to implement STDP, memristive devices are of high interest due to the analogy of the basic STDP mechanism found in memristors [112–118].

It is important to point out that the hardware implementation strategies for STDP should be very different, depending on the conductance switching dynamics of memristive devices. As shown in figure 9(a), a ferroelectric tunnel junction (FTJ) is fabricated by using BiFeO3 (BFO) as a ferroelectric tunnel barrier, which is sandwiched between a top electrode of Pt/Co and a bottom electrode of (Ca, Ce) MnO3 [119]. Applying voltage pulses to the FTJs changes the area of ferroelectric domains, resulting in junction conductance switching. As shown in figure 9(b), when both pre- and post-neuron spikes are applied to the FTJ closely in time, the voltage in the device temporarily exceeds the threshold. The conductance of FTJ will increase or decrease depending on the sign of time delay. An SNN network is simulated by a crossbar of 9 × 5 FTJs and successful unsupervised pattern recognition is demonstrated, as shown in figure 9(c).

Unlike FTJs where continuous conductance adjusting is achievable due to the polarization switching of ferroelectric domains under applied voltage pulse, the asymmetric phase-change dynamics in PCM makes the implementation of STDP a challenge [120, 121]. Usually, crystallization and amorphization dynamics of Ge2Sb2Te5 (GST) are utilized to realize the synaptic potentiation and depression, respectively. However, a gradual update of the state of the cell cannot be achieved in the amorphization process like crystallization and high energy consumption is required for creating the amorphization (figure 9(d)) [122]. To overcome these problems, a simplified power-efficient asymmetric STDP learning rule is proposed by differential configuration comprised of two cells, where a synaptic weight decrease operation is achieved by an increase of a negatively contributing second cell (figure 9(e)). Unsupervised learning as well as detection of multiple temporal correlations in parallel input streams are demonstrated in the phase-change memristors implemented SNN architecture (figure 9(f)) [123].

In the case of conductive bridging random access memory (CBRAM) devices, the change of resistance state depends on the formation and break of the conductive filament. This process is determined by the polarity and amplitude of the applied voltage. Thus, to implement STDP, a 1T1R array structure is adopted to provide access to fine control of individual CBRAM devices. With filaments consisting of semiconductor or semimetal subquatum, CBRAM cells with low programming energy (about 0.2 pl) and excellent filament stability are fabricated in a standard 130 nm CMOS process (figure 9(g)). As shown in figure 9(h), linear increasing and decreasing device conductance are provided by 1T1R CBRAM cells allowing fine synaptic weight updates (LTP/LTD) for SNN training, and high recognition accuracy is achieved in MNIST digit classification (figure 9(i)) [32].

Besides the standard STDP mechanism, other local training rules such as stochastic simplified STDP [124–126] and SRDP [127, 128] can be implemented by memristor naturally. For example, WO3−x devices (figure 9(j)) is second-order memristor which has delicate dynamics originating from the temporary effects such as the electric double-layer capacitance [129]. If a second spike reaches the device before the completely disappear of the first excitatory postsynaptic current (EPSC), the diffusion of oxygen ions will be suppressed. Then the oxygen ions will accumulate on the Pt/WO3−x interface and result in greater resistance change. As shown in figure 9(k), higher rate spikes lead to a much larger EPSC amplitude. This behavior of WO3−x device is similar to SRDP in the biological synapse. Furthermore, the Bienenstock–Cooper–Munro (BCM) learning rule, the mechanism for complex spatiotemporal pattern recognition in the human visual cortex such as rate-based orientation selectivity, can be implemented by second-order WO3−x devices [130]. The BCM
9.2. Short-term plasticity

The other type of synaptic plasticity, short-term plasticity, is also essential for SNN by offering much more dynamics. For example, as a common form of short-term plasticity, PPF was reported to be crucial for complex decoding and processing of temporal information functions in the human brain such as memory and forgiveness activity [142, 143]. Thus, to develop more powerful memristive SNN hardware for unlabeled date processing, a lot of efforts have been devoted to implementing short-term plasticity in memristive devices.

Artificial synapse with short-term plasticity, low-programming voltage (~0.2 V), and high-precision (1024 states) is realized by a three-terminal memristor with Li$_x$WO$_3$ channel and self-gate (figure 10(a)) [144]. As shown in figure 10(b), a typical PPF behavior is observed where the conductance change in response to a pair of voltage pulses is determined by the time delay of these two pulses. Based on the three-terminal Li$_x$WO$_3$ memristor, an SNNs model was built and reached 128× improvement in class separability when compared with SNN with NO-STP synapse in the classification performance benchmark test (figure 10(c)). The above Li$_x$WO$_3$ memristor demonstrates only one kind of short-term plasticity. However, multiple forms of short-term plasticity can be realized in a single device. As shown in figure 10(d), SiO$_x$N$_y$:Ag memristor...
compose of a SiO$_x$N$_y$ dielectric layer embedded with Ag nanoclusters and inert electrodes [145]. By in situ high-resolution TEM observation and simulation, the Ag dynamics of the SiO$_x$N$_y$:Ag memristor is revealed. The revealed Ag dynamics highly resemble the Ca$^{2+}$ behavior in synapses of the biological system, which allows the realization of multiple short-term plasticity phenomena such as PPF, PPD, and PPD following PPF, as shown in figures 10(e) and (f). Recently, memristive devices with short-term plasticity have proved to be very efficient in the realization of spatio-temporal computing, such as recognition of spatio-temporal patterns with a reduced training cost [146] and real-time motion detection [147].

Long-term plasticity and short-term plasticity are closely related. It was mentioned before that both long-term plasticity and short-term plasticity are indispensable in the physiological process of cognitive function [148, 149]. Similarly, by device-level hardware combination of long-term plasticity and short-term plasticity, the memristive computing system can gain some new features which cannot reach by only one synaptic plasticity. As shown in figure 10(g), four-terminal PCM cell consists of Ge$_{15}$Sb$_{85}$ on Al$_2$O$_3$ and SiO$_2$ dielectrics [150]. The voltage pulses applied on the source–drain terminals can program the non-volatile conductance states of the device by Joule heating while the pulses applied on the gate–drain terminals induces the transient modulation in the conductance states. As shown in figure 10(h), long-term plasticity and short-term plasticity can be represented by analog non-volatile conductance states and volatile conductance states, respectively. Therefore, device-level hardware integration of tunable long-term and short-term plasticity in a single artificial synapse is realized. Moreover, using this synapse with rich dynamics, sequential learning is demonstrated and synapse without short-term plasticity is unable to finish the sequential learning task successfully.

10. Summary of the memristive computing hardware for unlabeled data processing

Various kinds of memristive devices can be used for the realization of unlabeled data processing system including classic machine learning algorithms, dynamic neural networks, generative models, and SNN. Table 1 summarizes and compares the performance of these memristive unlabeled data processing systems. Although memristive computing systems have shown great success in simple unlabeled data processing, this promising
| Unlabeled data processing models | Memristive device | Precision (analog/binary) | Crossbar size | Peripheral circuit (Y/N) | Applications (accuracy) |
|---------------------------------|-------------------|--------------------------|---------------|--------------------------|-------------------------|
| PCA neural networks             | TaO<sub>x</sub> memristor [70] | Analog | 9 x 2 | Y | Analyzing the standard breast cancer screening database (97.1%) |
|                                 | WO<sub>x</sub> memristor [71]  | Analog | 54 x 108 | Y | Analyzing nine-dimensional input data (94.6%) |
|                                 | WO<sub>x</sub> memristor [73]  | Analog | 32 x 32 | Y | Sparse coding of simple inputs (94%) |
| Sparse coding                   | TaO<sub>x</sub> memristor [71] | Analog | 54 x 108 | Y | Reconstructed images (near 100%) |
|                                 | TaO<sub>x</sub> memristor [75] | Analog | 16 x 3 | Y | Classification of the standard IRIS data set (93.3%) |
| K-means clustering              | Graphene field effect transistor [76] | Analog | Single device | N | Not provided (not provided) |
|                                 | TaO<sub>x</sub> memristor [83] | Analog | 60 x 60 | Y | NP-hard max-cut problems (not provided) |
|                                 | TiO<sub>x</sub> memristor [85] | Analog | 20 x 20 | Y | Graph partitioning problem and maximum-weight independent set problem (not provided) |
| Hopfield network                | Y-flash cell [86] | Analog | 12 x 8 | Y | Associative memory |
|                                 | Al<sub>2</sub>O<sub>3</sub>/TiO<sub>2</sub>-x memristor [92] | Analog | 20 x 20 | N | Boltzmann machine (not provided) |
|                                 | TaO<sub>x</sub> memristor [93] | Analog | Single device | N | Discriminative RBM (not provided) |
| RBM network                     | Memristor model [94] | Analog | Single device | N | MNIST handwritten digit classifications (>85%) |
| Autoencoder                     | Pt/Cu/ZnO/Nb:STO memristor [101] | Analog | Single device | N | Denoising digits taken from the MNIST dataset. (Not provided) |
|                                 | Spintronic memristor models [103] | Analog | Not provided | Y | Single image super-resolution (not provided) |
|                                 | WO<sub>x</sub> memristors models [105] | Binary | 1.7 million | Y | Generation of more MNIST handwritten digits (not provided) |
| Generative adversarial network  | Ag–Si memristor models [102] | Binary | 512 x 256 | Y | Generation of more images from MNIST, fashion-MNIST, CIFAR-10, STL-10, and celeb-A (not provided) |
|                                 | BFO FTJ [119] | Analog | 5 x 9 | N | Simple pattern recognition (100%) |
|                                 | PCM [123] | Analog | 2 x 2 M cells | Y | Learning temporal correlations spatio-temporal patterns (not provided) |
| Spiking neural network          | CBRAM [32] | Analog | 512 kbit | Y | Unsupervised learning for MNIST (82%) |
|                                 | Pt/WO<sub>x</sub>/W memristor [131] | Analog | 9 x 9 | N | Rate-based orientation selectivity (100%) |
In a nutshell, the integration of memristive devices and computing systems for unlabeled data can revolutionize technology is still in its infancy. Only a few devices such as WO$_x$ memristor and PCM have implemented full-custom integrated circuits and the crossbar size is not big enough. Many of these systems are simulation results or simple prototypes. Therefore, the realization of practical memristive unlabeled data processing computing systems faces many obstacles, especially for data-intensive tasks.

Firstly, the main engineering obstacle is non-ideal effects including device defects, device-to-device variation, cycle-to-cycle variation, and parasitics [151, 152]. Several serious consequences arise when non-ideal memristive devices are used, such as the large latency, drop in resolution and accuracy, as well as non-convergence [153]. Compensating these non-ideal effects is crucial to the implementation of unlabeled data processing, especially for realizing neural networks-based models with memristor crossbars. Thus, many strategies have been proposed. For example, device defects could be compensated by using defect map in the training process [151]. Fine programming methods such as ‘write-verify’ can be used to suppress the variations [154]. Parasitics causes errors in currents/voltages and RC delay. Therefore, correction circuits are usually added to compensate parasitics [155]. Recently, a ‘variation- and defect-aware training’ strategy is proposed to compensate variation and defects. This strategy reduces the drop in inference accuracy significantly even under severe variations [156]. Nevertheless, some non-ideal effects in a proper range, such as intrinsic resistance drift of PCM [157], fluctuations and stochastic of RRAM [93] and intrinsic hardware noise of memristor [83] can be utilized to improve the accuracy of the computing system.

Secondly, IR drop is another critical challenge in memristor crossbar toward high-efficient unlabeled data processing systems. The IR drop problem, caused by the limited electrical conductivity of metal wire connections, constrains the scalability of these systems and hinders the degree of parallelism in computing [158, 159]. The IR drop problem could be suppressed in some way such as through compensation technique [160] and designing new CMOS-memristor hybrid cells [161]. Nevertheless, it still needs special attention when designing memristive unlabeled data processing systems, especially for data-intensive tasks.

Thirdly, in the memristor crossbar, the high voltage applied in the initial forming process for the memristors can damage the devices in the same array without selectors. To fully exploit the benefits of memristive devices over traditional CMOS devices for developing advanced computing systems, electroforming-free memristive devices are in urgent need. The key parameters for forming processes are the voltage for electroforming, the electroforming time, and the presence and quality of switching behavior after electroforming. These parameters strongly depend on the insulating material as well as the deposit method [140]. Electroforming-free has been achieved in memristive tunneling junctions [162] and TaO$_x$-based memristor [67], but still needs to be developed in many other memristive devices.

Last but not least, not all the as-prepared memristive devices can be utilized for all the proposed models and systems. To overcome this obstacle, a lot of effort is dedicated both at the device level and at the circuit level. For example, a lot of memristive devices do not have enough internal states to ensure gradual continuous conductance modulation under subsequent voltage signals. Therefore, these devices are unfit for realizing complex synaptic plasticity, let alone SNN [163]. This problem can be solved by introducing a buffer layer such as AlO$_x$ to reduce current overshoot [164] or an electro-thermal modulation layer to control the electric field and temperature of the switching layer [165]. Memristor has nonlinear $I$–$V$ characteristics and which is an obstacle when performing VMMs in the current/voltage domain. This restriction can be bypassed by developing computing systems that support charge domain operation with pulse width modulation and custom ADCs [71, 166].

11. Conclusion and prospects

In a nutshell, the integration of memristive devices and computing systems for unlabeled data can revolutionize classification, optimization, and neuromorphic computing research thanks to the great improvement in energy efficiency and computation speed. To promote the development of this exciting research topic, researchers in different fields, such as AI, computer science, signal processing, circuits and microelectronics, and materials, should work together to make significant improvements and overcome the challenges occurring in cross-layers of device, circuits, microarchitectures, systems, and software. Here are a few points that we have to make about the latest trends.

As mentioned before, memristive devices have large fluctuations in parameters due to non-ideal effects. These defects dramatically reduce yields which are crucial for their commercial application [167]. Many strategies have been proposed to promote the compatibility of the memristive device with the CMOS process [168–170]. Continued material and process optimizations are still required to address the remaining variation and yield issues before mass production for commercial applications.

While SNNs are the better choices to deal with spatiotemporal unlabeled data, it still falls behind the traditional ANNs in the benchmark test on some tasks and the highly effective training of SNNs is still an
open question. These algorithmic issues must be resolved for developing future efficient SNN unlabeled data processing systems.

Hardware/software co-design is a key strategy for building efficient computing systems by combining the advances in memristive devices with algorithmic methods. For example, high efficiency and accuracy memristive computing system for pattern recognition is experimentally demonstrated with the integration of low-energy consumption subquantum CBRAM devices and network pruning algorithm [32]. An adaptive quantization method in software is utilized to compensate for the accuracy loss due to limited conductance levels of low-precision PCM in hardware [171].

In summary, we envision that there will be innovations in hardware, software as well as their co-design for the field of low-power, efficient computing platforms. The near-term development of computing platforms for the unlabeled data processing shall be built with mixed-signal architectures via the co-design of memristive devices and conventional CMOS technology. In the long run, breakthroughs are expected from advanced computational memristive computing systems with new hardware and software, such as efficient nano memristive devices of new operating principles and materials as well as unlabeled data processing models with more complex human-like cognitive abilities.

Acknowledgments

This work is supported by the National Key R & D Program of China (2017YFA0207600), National Natural Science Foundation of China (61925401, 92064004, 61927901, 92164302), Project 2019BD002 and 2020BD010 supported by PKU-Baidu Fund, and the 111 Project (B18001). YY acknowledges the support from the Fok Ying-Tong Education Foundation and the Tencent Foundation through the XPLORER PRIZE.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

ORCID iDs

Yuchao Yang https://orcid.org/0000-0003-4674-4059

References

[1] Schmidhuber J 2015 Deep learning in neural networks: an overview Neural Netw. 61 85–117
[2] Patel U et al 2021 Artificial intelligence as an emerging technology in the current care of neurological disorders J. Neurol. 268 1623–42
[3] Duan Y, Edwards J and Dwivedi Y 2019 Artificial intelligence for decision making in the era of big data—evolution, challenges and research agenda Int. J. Inf. Manage. 48 63–71
[4] Esmin A, Coelho R and Matwin S 2015 A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data Artif. Intell. Rev. 44 23–45
[5] Slonim N et al 2021 An autonomous debating system Nature 591 379–84
[6] Mallat S 2016 Understanding deep convolutional networks Phil. Trans. R. Soc. A 374 20150203
[7] Tanaka G, Yamane T, Heroux J, Nakane R, Kanazawa N, Takeda S, Numata H, Nakano D and Hirose A 2019 Recent advances in physical reservoir computing: a review Neural Netw. 115 100–23
[8] Dong X, Yu Z, Cao W, Shi Y and Ma Q 2020 A survey on ensemble learning Front. Comput. Sci. 14 241–58
[9] Alpert A et al 2019 A clinically meaningful metric of immune age derived from high-dimensional longitudinal monitoring Nat. Med. 25 487–95
[10] Vallieres M et al 2017 Radiomics strategies for risk assessment of tumour failure in head-and-neck cancer Sci. Rep. 7 10117
[11] Yang X, Wang Y, Byrne R, Schneider G and Yang S 2019 Concepts of artificial intelligence for computer-assisted drug discovery Chem. Rev. 119 10520–94
[12] Jamshidi M et al 2020 Artificial intelligence and COVID-19: deep learning approaches for diagnosis and treatment IEEE Access 8 109581–95
[13] Benjamens S, Dhunnoo P and Mesko B 2020 The state of artificial intelligence-based FDA-approved medical devices and algorithms: an online database NPJ Digit. Med. 3 118
[14] Santosh K C 2020 AI-driven tools for coronavirus outbreak: need of active learning and cross-population train/test models on multitudinal/multimodal data J. Med. Syst. 44 93
[15] Hosseini M S, Muhammad G and Guizani N 2020 Explainable AI and mass surveillance system-based healthcare framework to combat COVID-19 like pandemics IEEE Netw. 34 126–32
[16] Shin H, Orton M, Collins D, Doran S and Leach M 2013 Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data IEEE Trans. Pattern Anal. Mach. Intell. 35 1930–43
[17] Fardanesh M and Erosy O 1998 Classification accuracy improvement of neural network classifiers by using unlabeled data IEEE Trans. Pattern Anal. Mach. Intell. 36 1020–5
[18] Han W, Feng R, Wang L and Cheng Y 2018 A semi-supervised generative framework with deep learning features for high-resolution remote sensing image scene classification ISPRS–J. Photogramm. Remote Sens. 145 23–43
[19] Guo L, Lei Y, Xing S, Yan T and Li N 2019 Deep convolutional transfer learning network: a new method for intelligent fault diagnosis of machines with unlabeled data IEEE Trans. Ind. Electron. 66 7316–25
[20] Khodayar M, Kaynak O and Khodayar M 2017 Rough deep neural architecture for short-term wind speed forecasting IEEE Trans. Ind. Inf. 13 2770–9
[21] Tao C, Pan H, Li Y and Zou Z 2015 Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification IEEE Geosci. Remote Sens. Lett. 12 2438–42
[22] Zhang S, Yao H, Sun X and Lu X 2013 Sparser coding based visual tracking: review and experimental comparison Pattern Recognit. 46 1772–88
[23] Bu Z, Li H, Zhang C, Cao J, Li A and Shi Y 2020 Graph k-means based on leader identification, dynamic game, and opinion dynamics IEEE Trans. Knowl. Data Eng. 32 1348–61
[24] Abualigah L and Khader A 2017 Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering J. Supercomput. 73 4773–95
[25] Wang H, Yu Y, Wen G, Zhang S and Yu J 2015 Global stability analysis of fractional-order Hopfield neural networks with time delay Neurocomputing 154 15–26
[26] Tavares C, Santos T, Lemes N, Dos-Santos J, Ferreira J and Braga J 2021 Solving ill-posed problems faster using fractional-order Hopfield neural network J. Comput. Appl. Math. 381 112984
[27] Langkort M, Karlsson L and Louiti A 2014 A review of unsupervised feature learning and deep learning for time-series modeling Pattern Recognit. Lett. 42 11–24
[28] Solorio-Fernandez S, Carrasco-Ochoa J and Martinez-Trinidad J 2020 A review of unsupervised feature selection methods Artif. Intell. Rev. 53 907–48
[29] Saxena A, Prasad M, Gupta A, Bharill N, Patel O, Tiwari A, Er M, Ding W and Lin C 2017 A review of clustering techniques and developments Neurocomputing 267 641–84
[30] Roy K, Jaiswal A and Panda P 2019 Towards spike-based machine intelligence with neuromorphic computing Nature 575 607–17
[31] Silver D et al 2016 Mastering the game of Go with deep neural networks and tree search Nature 529 484–9
[32] Shi Y, Nguyen L, Oh S, Liu X, Koushan F, Jameson J and Kuzum D 2018 Neuroinspired unsupervised learning and pruning with subquantum CBRAM arrays Nat. Commun. 9 5312
[33] Yang Y, Zhang X, Qin L, Zeng Q and Huang R 2017 Probing nanoscale oxygen ion motion in memristive systems Nat. Commun. 8 15173
[34] Strukov D, Snider G, Stewart D and Williams R 2008 The missing memristor found Nature 453 80–3
[35] Lin P et al 2020 Three-dimensional memristor circuits as complex neural networks Nat. Electron. 3 225–32
[36] Wong H, Raoux S, Kim S, Liang J, Reifenberg J, Rajendran B, Asheghi M and Goodson K 2010 Phase change memory J. Appl. Phys. 107 084507
[37] Wang Y, Zhang X, Qin L, Zeng Q, Zhang W, Yang J and Qian H 2020 Fully hardware-implemented memristor convolutional neural network Nature 577 641–6
Neuromorph. Comput. Eng. (2022) 022003

[99] Sheri A, Rafique A, Pedrycz W and Joom M 2015 Contrastive divergence for memristor-based restricted Boltzmann machine Eng. Appl. Artif. Intell. 37 336–42

[100] Alam M, Fernando B, Isoudi Y, Yakopcic C, Hasan R, Taha T and Subramanyam G 2019 Memristor based autoencoder for unsupervised real-time network intrusion and anomaly detection Proc. Int. Conf. Neuro. Syst. (https://doi.org/10.1145/3354265.3354267)

[101] Subh-E. Boppi, P, Rao B, Banerjee S and Kundu S 2019 Realizing spike-timing dependent plasticity learning rule in Pt/CuZnO/Nb:STO memristors for implementing single spike based denoising autoencoder J. Microelec. Microeom. 29 9

[102] Roohi A, Sheikhfaal S, Angizi S, Fan D and DeMara R 2020 ApGAN: approximate GAN for robust low energy learning from imprecise components IEEE Trans. Comput. 69 349–60

[103] Dong Z, Fang Y, Huang L, Li J and Qi D 2019 A compact memristor-based GAN architecture with a case study on single image super-resolution 31st Chin. Control Decis. Conf. (https://doi.org/10.1109/ccdc.2019.8832460)

[104] Hu M et al 2018 Memristor-based analog computation and neural network classification with a dot product engine Adv. Mater. 30 1705914

[105] Krestinskaya O, Choueby B and James A 2020 Memristor GAN in analog Sci. Rep. 10 5838

[106] Hayashi-Takagi A, Yagishita S, Nakamura M, Shirai F, Wu Y, Loshbaugh A, Kuhlman B, Hahn K and Kasai H 2015 Labelling and optical erasure of synaptic memory traces in the motor cortex Nature 525 333–8

[107] Mead C 1990 Neuromorphic electronic systems Proc. IEEE 78 1629–36

[108] Ohno T, Hasegawa T, Tsuruoka T, Terabe K, Gimzewski J and Aono M 2011 Short-term plasticity and long-term potentiation mimicked in single inorganic synapses Nat. Mater. 10 591–5

[109] Eriksson J, Vogel E, Lansner A, Bergström F and Nyberg L 2015 Neurocognitive architecture of working memory Neuron 88 33–46

[110] Wang Z, Xu H, Li X, Yu H, Liu Y and Zha X 2012 Synaptic learning and memory functions achieved using oxygen ion migration/diffusion in an amorphous InGaZnO memristor Adv. Funct. Mater. 22 2759–65

[111] Adnan M, Sayyapanaraju S, Brown S D, Shawkat M, Schuman C and Rose G 2021 Design of a robust memristive spiking neuromorphic system with unsupervised learning in hardware J. Emerg. Technol. Comput. Syst. 17 1–26

[112] Querlioz D, Zhao W, Dolfus P, Klein J, Bichler O and Gamrat C 2012 Bioinspired networks with nanoscale memristive devices that combine the unsupervised and supervised learning approaches IEEE/ACM Int. Symp. Nanoscale Archit. (https://doi.org/10.1109/2765491.2785528)

[113] Prezioso M, Merrilh-Bayat F, Hoskins B, Likharev K and Strukov D 2016 Self-adaptive spike-time-dependent plasticity of metal-oxide memristors Sci. Rep. 6 21331

[114] Serb A, Ill J, Khati A, Berdan R, Legenstein R and Prodromakis T 2016 Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses Nat. Commun. 7 12611

[115] Sarim M, Schultz T, Jha R and Kumar M 2016 Ultra-low energy neuromorphic device based navigation approach for biomimetic robots IEEE Natl. Aerosp. Electron. Conf. (https://doi.org/10.1109/naecon.2016.7856805)

[116] Payvand M, Rofeh J, Sodhi A and Theogarajan L 2014 A CMOS-memristive self-learning neural network for pattern classification applications IEEE/ACM Int. Symp. Nanoscale Archit. (https://doi.org/10.1109/nanosarch.2014.6880486)

[117] Such O, Klimo M, Kemp N and Skvarek O 2018 Passive memristor synaptic circuits with multiple timing dependent plasticity mechanisms AEU-Int. J. Electron. Commun. 96 252–9

[118] Lammie C, Hamilton T and Azghadi M 2018 Unsupervised character recognition with a simplified FPGA neuromorphic system IEEE Int. Symp. Circuits Syst. (https://doi.org/10.1109/iscas.2018.8351532)

[119] Boynt S et al 2017 Learning through ferroelectric domain dynamics in solid-state synapses Nat. Commun. 8 14736

[120] Brivio S, Ly D, Vianello E and Spiga S 2021 Non-linear memristive synaptic dynamics for efficient unsupervised learning in spiking neural networks Front. Neurosci. 15 580909

[121] Munoz-Martín I, Bianchi S, Pedretti G, Melnick O, Ambrogio S and Ielmini D 2019 Unsupervised learning to overcome catastrophic forgetting in neural networks IEEE J. Emerg. Sel. Top. Circuits Syst. 2 201822

[122] Boynat I et al 2018 Neuromorphic computing with multi-memristive synapses Nat. Commun. 9 2514

[123] Fantazi A, Wozniak S, Tuma T and Eleftheriou E 2016 All-memristive neuromorphic computing with level-tuned neurons Nanotechnology 27 355205

[124] Demin V, Nekhaev D, Surazhevsy I, Nikiruy K, Emelyanov A, Nikolaev S, Rylkov V and Kovalchuk M 2021 Necessary conditions for STDP-based pattern recognition learning in a memristive spiking neural network Neurol. Netw. 134 64–75

[125] Bill J and Legenstein R 2014 A compound memristive synapse model for statistical learning through STDP in spiking neural networks Front. Neurosci. 8 421

[126] Nishiy N, Yomura K, Murakume T and Mizushima K 2021 Stochastic binary synapses having sigmoidal cumulative distribution functions for unsupervised learning with spike timing-dependent plasticity Sci. Rep. 11 181282

[127] Milo V, Pedretti G, Carboni R, Calderoni A, Ramaswamy N, Ambrogio S and Jelmini D 2018 A 4-transistors/l-resistor hybrid synapse based on resistive switching memory (RRAM) capable of spike-rate-dependent plasticity (SRDP) IEEE Trans. VLSI Syst. 26 2806–15

[128] Rachmuth G, Shouval H, Bear M and Poon C 2011 A biophysically-based neuromorphic model of spike rate- and timing-dependent plasticity Proc. Natl Acad. Sci. USA 108 1266–74

[129] Yang C, Shang D, Liu N, Fuller E, Agrawal S, Talin A, Li Y, Shen B and Sun Y 2018 All-solid-state synaptic transistor with ultralow conductance for neuromorphic computing Adv. Funct. Mater. 28 1804170

[130] Gjorgieva J, Clopath C, Audet J and Pfister J 2011 A triplet spike-timing-dependent plasticity model generalizes the Bienenstock–Cooper–Munro rule to higher-order spatiotemporal correlations Proc. Natl Acad. Sci. USA 108 19383–8

[131] Wang Z, Zeng T, Ren Y, Lin Y, Xu H, Zhao X, Liu Y and Jelmini D 2020 Toward a generalized Bienenstock–Cooper–Munro rule for spatiotemporal learning via triplet-STDP in memristive devices Nat. Commun. 11 1510

[132] Majumdar S, Tan H, Qin Q and van Dijken S 2019 Energy-efficient organic ferroelectric tunnel junction memristors for neuromorphic computing Adv. Electron. Mater. 5 1800795

[133] Pedretti G, Milo V, Ambrogio S, Carboni R, Bianchi S, Calderoni A, Ramaswamy N, Spinelli A and Jelmini D 2017 Memristive neural network for on-line learning and tracking with brain-inspired spike timing-dependent plasticity Sci. Rep. 7 52388

[134] Pedretti G, Milo V, Ambrogio S, Carboni R, Bianchi S, Calderoni A, Ramaswamy N, Spinelli A and Jelmini D 2018 Stochastic learning in neuromorphic hardware via spike timing dependent plasticity with RRAM synapses IEEE J. Emerg. Sel. Top. Circuits Syst. 8 77–85
[135] Ambrogio S, Balatti S, Milo V, Carboni R, Wang Z Q, Calderoni A, Ramaswamy N and Ielmini D 2016 Neuromorphic learning and recognition with one-transistor-one-resistor synapses and bistable metal oxide RRAM IEEE Trans. Electron Devices 63 1508–15
[136] Sidler S, Pantazi A, Wozniak S, Leblebici Y and Eleftheriou F 2017 Unsupervised learning using phase-change synapses and complementary patterns 26th Int. Conf. Artif. Neural Netw. (https://doi.org/10.1007/978-3-319-48600-4_33)
[137] Ly D, Grossi A, Fenouillet-Beranger C, Nowak E, Querlioz D and Vianello E 2018 Role of synaptic variability in resistive memory-based spiking neural networks with unsupervised learning J. Phys. D: Appl. Phys. 51 444002
[138] Wang Z et al 2018 Fully memristive neural networks for pattern classification with unsupervised learning Nat. Electron. 1 137–45
[139] Andreeva N, Ryndin E and Gerasimova M 2020 Memristive logic design of multifunctional spiking neural network with unsupervised learning BioNanoScience 10 824–33
[140] Wang T et al 2020 Electroforming in metal-oxide memristive synapses ACS Appl. Mater. Interfaces 12 11806–14
[141] John R et al 2018 Ionotronic halide perovskite drift–diffusive synapses for low-power neuromorphic computation Adv. Mater. 30 1805454
[142] Catterall W, Leal K and Nahum M 2014 Brain plasticity-based therapeutics Front. Hum. Neurosci. 8 358
[143] Sarwat S, Kersting B, Moraitis T, Jonnalagadda V and Sebastian A 2018 Phase-change memristive synapses for mixed-plasticity neural computations Nat. Nanotechnol. 17 507–13
[144] Nguyen T, An J and Oh S 2021 Training, programming, and correction of memristor-crossbar neural networks with non-ideal effects such as defects, variation, and parasitic resistance 14th Int. Conf. ASIC (ASICON) pp 1–4
[145] Pham K and Min K 2019 Non-ideal effects of memristor-CMOS hybrid circuits for realizing multiple-layer neural networks IEEE Int. Symp. Circuits. Syst. pp 1–5
[146] Li J et al 2021 In situ learning in hardware compatible multi-layer memristive spiking neural network IEEE Trans. Cogn. Dev. Syst. (https://doi.org/10.1109/tcods.2021.3049487)
[147] Homouz D, Mohammad B, Elgabra H and Farahat I 2011 Memristor: modeling read and write operations Proc. Int. Conf. Mechatronics pp 1–5
[148] Prokopenko N, Gaiduk A, Budyakov P and Butyrlavgin N 2014 The synthesis of the correction circuit of the high speed signals of the physical quantities and current–voltage converters with the parasitic capacitance Proc. IEEE East-West Des. Test Symp. (EWDTS) pp 1–4
[149] Gao D, Zhang G, Yin X, Li B, Schlichtmann U and Zhoo C 2021 Reliable memristor-based neuromorphic design using variation- and defect-aware training IEEE ACM Int. Conf. Comput. Des. Dig. Tech. Pap. (ICCAD) pp 1–9
[150] Lim D, Wu S, Zhao R, Lee J, Jeong H and Shi L 2021 Spontaneous sparse learning for PCM-based memristor neural networks Nat. Commun. 12 319
[151] Sun Z, Pedretti G, Ambrosi E, Bricalli A, Wang W and Ielmini D 2019 Solving matrix equations in one step with cross-point resistive arrays Proc. Natl Acad. Sci. USA 116 4123–8
[152] Chen P Y et al 2015 Technology-design co-optimization of resistive cross-point array for accelerating learning algorithms on chip 2015 Des. Autom. Test Europe Conf. Edlib. (DATE) pp 854–9
[153] Lepri N, Baldo M, Mannocci P, Glukhov A, Milo V and Ielmini D 2022 Modeling and compensation of IR drop in crosspoint accelerators of neural networks IEEE Trans. Electron Devices 69 1575–81
[154] Wang W, Pedretti G, Milo V, Carboni R, Calderoni A, Ramaswamy N, Spinelli A and Ielmini D 2018 Learning of spatiotemporal patterns in a spiking neural network with resistive switching synapses Sci. Adv. 4 eaat4
[155] Hansen M, Zahari F, Kohlstedt H and Ziegler M 2018 Unsupervised Hebbian learning experimentally realized with analogue memristive crossbar arrays Sci. Rep. 8 8914
[156] Li C et al 2018 Large memristor crossbars for analog computing IEEE Int. Symp. Circuits. Syst. (ISCAS) pp 1–4
[157] Kim S, Kim H, Hwang S, Kim M, Chang Y and Park B 2017 Analog synaptic behavior of a silicon nitride memristor ACS Appl. Mater. Interfaces 9 40420–7
[158] Wu W et al 2018 A methodology to improve linearity of analog RRAM for neuromorphic computing Dig. Tech. Pap. Symp. VLSI Technol. pp 103–4
[159] Genov R and Cauwenberghs G 2000 Charge-mode parallel architecture for matrix-vector multiplication Proc. 43rd IEEE Midwest. Symp. Circuits. Syst. Midwest. pp 906–9
[160] Sung C, Hwang H and Yoo I 2018 Perspective: a review on memristive hardware for neuromorphic computation J. Appl. Phys. 124 151903
[161] Hazra J, Liehr M, Beckmann K, Rafiq S and Cady N 2020 Impact of atomic layer deposition co-reactant pulse time on 65 nm CMOS integrated hafnium dioxide-based nanoscale RRAM devices IEEE Int. Integr. Reliab. Workshop (IIRW) pp 1–4
[162] Hazra J, Liehr M, Beckmann K, Rafiq S and Cady N 2019 Improving the memory window/resistance variability trade-off for 65 nm CMOS integrated HfO2-based nanoscale RRAM devices IEEE Int. Integr. Reliab. Workshop (IIRW) pp 1–4
[163] Kim W et al 2019 Confined PCM-based analog synaptic devices offering low resistance-drift and 1000 programmable states for deep learning Dig. Tech. Pap. Symp. VLSI Technol. pp T66–7
[164] Shi Y, Huang Z, Oh S, Kaslan N, Song J and Kazum D 2019 Adaptive quantization as a device-algorithm co-design approach to improve the performance of in-memory unsupervised learning with SNNS IEEE Trans. Electron Devices 66 1722–8