

Reservoir Computing Using Diffusive Memristors


Rivu Midya, Zhongrui Wang, Shiva Asapu, Xumeng Zhang, Mingyi Rao, Wenhao Song, Ye Zhuo, Navnidhi Upadhyay, Qiangfei Xia,* and J. Joshua Yang*

Reservoir computing (RC) is a framework that can extract features from a temporal input into a higher-dimension feature space. The reservoir is followed by a readout layer that can analyze the extracted features to accomplish tasks such as inference and classification. RC systems inherently exhibit an advantage, since the training is only performed at the readout layer, and therefore they are able to compute complicated temporal data with a low training cost. Herein, a physical reservoir computing system using diffusive memristor-based reservoir and drift memristor-based readout layer is experimentally implemented. The rich nonlinear dynamic behavior exhibited by a diffusive memristor due to Ag migration and the robust in situ training of drift memristor arrays makes the combined system ideal for temporal pattern classification. It is then demonstrated experimentally that the RC system can successfully identify handwritten digits from the Modified National Institute of Standards and Technology (MNIST) dataset, achieving an accuracy of 83%.

Artificial neural networks (ANNs) are biological neural networks like computational models. They constitute the most important information-processing technology in the fields of artificial intelligence (AI) and machine learning. There has been dramatic progress in the field of AI recently, which is expected to become increasingly ubiquitous in our day-to-day lives in the near future.^[1]

ANNs are comprised of a network of neural nodes, which are interconnected by weighted synapses. Architectures of ANNs are classified into feedforward networks^[2] and recurrent networks.^[3] These two networks excel at different types of computational tasks. In feedforward networks, individual input data is processed independently, even though it could be provided sequentially to the network. Hence, they are suitable for static or non-temporal data processing. Recurrent neural networks (RNNs) are able to embed temporal dependence of the inputs

R. Midya, Dr. Z. Wang, S. Asapu, X. Zhang, M. Rao, W. Song, Y. Zhuo, N. Upadhyay, Prof. Q. Xia, Prof. J. J. Yang
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA 01003, USA
E-mail: qxia@umass.edu; jjyang@umass.edu

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.201900084>.

© 2019 The Authors. Published by WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.201900084

into their dynamical behavior and hence are capable of representing dynamical systems driven by sequential inputs due to their feedback connections. Hence, RNNs are suitable for dynamic (temporal) data processing.

Initially, reservoir computing (RC) was an RNN-based framework and hence is suitable for temporal/sequential information processing.^[4] RNN models of echo state networks (ESNs)^[5] and liquid state machines (LSMs)^[6] were proposed independently. These aforementioned models led to the development of the unified computational framework of RC.^[7] The backpropagation decorrelation (BPDC)^[8] learning is also viewed as a predecessor of RC.

The input data is transformed into spatiotemporal patterns in a high-dimensional space by an RNN in the reservoir (Figure 1a).

Subsequently, the spatiotemporal patterns generated are analyzed for a matching pattern in the readout. The input weights and the weights of the recurrent connections in the reservoir are fixed. The only weights that need to be trained are the weights in the readout layer. This can be done using a simple algorithm-like linear regression. This offers an inherent advantage, since such simple and fast training reduces the computational cost of learning compared with standard RNNs.^[9] RC models have been used for various computational problems such as temporal pattern classification, prediction, and generation. However, to maximize the effectiveness of a certain RC system, it is necessary to appropriately represent sample data and optimize the design of the RNN-based reservoir.^[7]

Specifically, the role of the reservoir is to nonlinearly transform sequential inputs into a high-dimensional space such that the features of the inputs can be read out efficiently using a simple learning algorithm. Hence, instead of traditional RNNs, other nonlinear dynamical systems can also play the role of reservoirs. Physical RC systems using reservoirs are in vogue at this time with multiple research communities reporting such physical RC systems.^[10,11] One of the motivations for this interest in physical RC systems is to realize fast information processing systems with low learning cost. Traditional hardware implementations of reservoirs that require training the readout layer often demanded power-hungry neuromorphic hardware.^[12,13] On the contrary, physical implementation of reservoirs can be achieved using a variety of physical phenomena in the real world, because a mechanism for adaptive changes for training is not necessary.

Memristors are two terminal devices having the capability to change their resistance when subjected to a voltage bias.^[14,15]

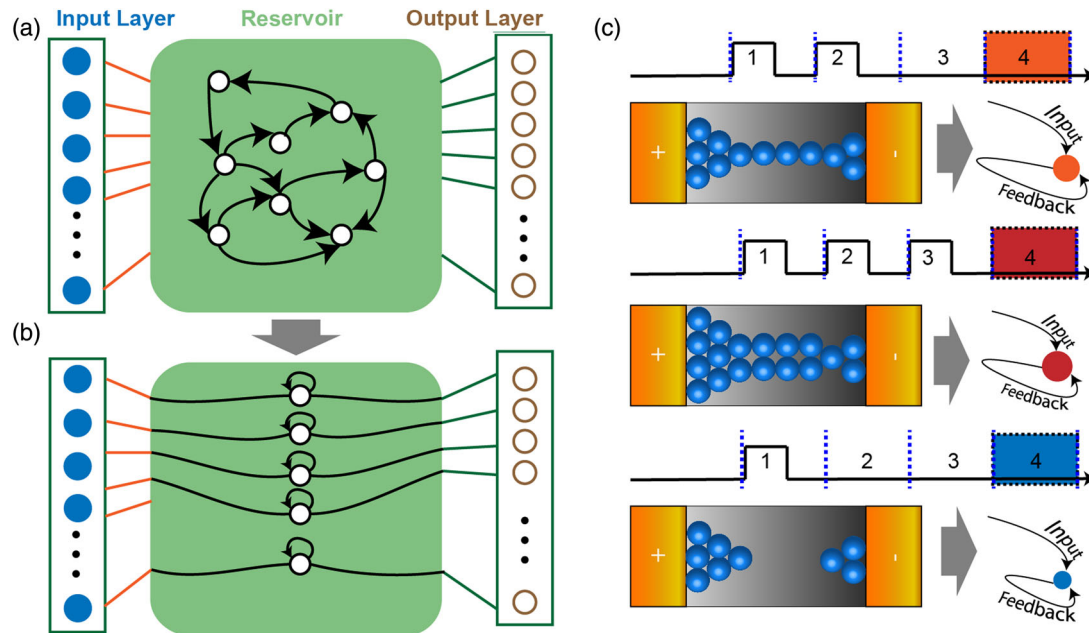


Figure 1. Reservoir computing system based on diffusive memristor: a) Schematic of an RC system, showing the reservoir with internal dynamics and a readout function. The weight matrix connecting the reservoir state and the output needs to be trained. b) Equivalent schematic of a simplified system where the reservoir is populated with nodes with recurrent connections having a magnitude less than 1. c) The conductivity of the diffusive memristor is influenced by the periodic voltage stimulation that is provided on its top electrode (+) while grounding the bottom electrode (-). In the top panel, in two consecutive time slots, a voltage stimulus is provided. This results in a continuous Ag filament and high conductivity when the device state is analyzed in the fourth time slot. In the middle panel, in three consecutive time slots, a voltage stimulus is applied resulting in a much thicker filament and even higher conductivity when the device state is analyzed in the fourth time slot. In case of the bottom panel, a voltage stimulus is applied only in the first time slot. When the device state is evaluated in the fourth time slot, the filament has already broken down resulting in very low conductivity. This is because the device has enough time to relax back to its initial high resistance state, owing to its volatile nature. The diffusive memristor is analogous to a node with a recurrent connection having a weight less than 1. Hence, it is always decaying its state in every time frame unless a sufficiently large input is provided to counteract the effect of the feedback.

Memristors can be broadly classified into volatile and non-volatile memristors based on whether they are able to maintain their resistive state upon the removal of the electrical bias.^[16] A memristor's resistance is governed by the ion configuration inside the dielectric and/or dielectric/electrode interfaces.^[15,17] The internal redistribution of oxygen ions or metal cations inside the device leads to change in the resistivity and the overall device resistance. The compact device structure and the ability to both store and process information at the same physical locations make memristors and memristor crossbar arrays attractive candidates for neuromorphic computing applications. Recently, several synaptic functions of a biological synapse have been emulated by memristor-based synapses. These functions include, but are not limited to, paired pulse facilitation/paired pulse depression (PPF/PPD),^[18] long-term potentiation/long-term depression (LTP/LTD),^[18-20] and spike time dependent plasticity (STDP).^[18,20] Recently, there have also been several reports of novel artificial neurons based on metal-insulator-transition (MIT, e.g., Nb₂O₅,^[21]VO₂^[22]), ferromagnetic,^[23] phase change materials,^[24] as well as diffusive memristors.^[25] Accordingly, memristor-based neural networks have been built following the tenets of ANN^[26-29] and a few that can be classified as SNN.^[30-33]

Very interestingly, memristors have also been used for building a physical RC system.^[34] It was used to demonstrate Modified

National Institute of Standards and Technology (MNIST) handwritten digit classification with an accuracy of 88% on a reduced MNIST dataset of 14 000 training samples and 2000 test samples. In this interesting demonstration, the readout was still implemented in software. In addition, although the memristors used in this study had short-term memory effects, they had a long decay time, which resulted in a longer pattern-to-pattern period and thus an extra time cost. A volatile and yet faster device might be better suited for an RC system. There are several memristors that are inherently volatile and nonlinear^[35] and are hence suitable naturally to serve the role of a reservoir.

In this article, we discuss a physical implementation of a RC system where a diffusive memristor-based reservoir was equipped with drift memristor-based 1T1R (1 Transistor 1 Memristor) readout layer. The input to the reservoir was provided in the form of bitstreams. These bitstreams were provided in the form of engineered waveforms, which took advantage of the rich short-term dynamics of the device. The readout system was in situ trained to classify the temporal-version of MNIST handwritten digits. An accuracy of 83% for the complete MNIST dataset was achieved using hardware drift memristor-based 1T1R readout layer.

In this work, the short-term dynamics of a volatile memristor are harnessed to build the reservoir of our RC system. Diffusive memristors are a class of volatile memristors whose switching is

governed by fast diffusive species (e.g., Ag).^[18,35,36] We use a SiO_x (doped with Ag)-based diffusive memristor to build our reservoir. The temporal dynamics of this device has been investigated through the demonstration of short-term plasticity (STP).^[18] In the STP demonstration, it had been shown that when several pulses are applied at short intervals, the resistance will gradually drop, whereas if there is a long time interval between consecutive pulses, the device resistance increases during this time. An interesting corollary of the former observation is that when the device is being programmed, the state of the device does not only depend on the programming pulse itself but also on whether other programming pulses have been applied in the past. Thus, pulses that were applied closer to the present time will have a stronger effect compared to those that were applied in the far past. There is a threshold to this effect as if the pulses were applied sufficiently long time before the arrival of the next pulse pattern, then the device will have enough time to return to its initial high resistance state. It is hence reasonable to compare a diffusive memristor with a neuron having a recurrent connection with a weight less than 1 (Figure 1b). Such a recurrent node would continuously decay its state if not provided with an input. More specifically, the diffusive memristor will have a thicker filament or a higher conductivity if it is continuously subjected to pulses, whereas if sufficiently long time is spent before the application of a new pulse, the filament breaks down (Figure 1c).

If we refer to Figure 1c and analyze the state of the diffusive memristor on the application of a pulse train, we find that when we apply two pulses (top panel) in the first and second time frames consecutively and check the state of the memristor at the fourth time frame, the memristor still has an intact filament and the conductivity is relatively high. When we apply three pulses (middle panel) in the first, second, and third time frames, respectively, the memristor has a thicker filament, indicating a higher conductivity than the former case. On the contrary, when we apply only one pulse (bottom panel) during the first time frame and consider the state of the memristor in the fourth time frame, we find that the filament has broken down and conductivity of the device is very low. These scenarios are akin (considering that all input pulses are identical) to what will happen in case of a recurrent node with a recurrent weight having a magnitude less than 1. In the first case, by the fourth time frame, the state of the node would decrease twice. In the second case, by the fourth time frame, the state of the node would decrease once. In the third case, by the fourth time frame, the state of the node would decrease thrice.

Physically, a single diffusive memristor populates the reservoir of our RC system. Thus, the state of the reservoir is decided by the resistance state of the device. Once the bit-coded pulse streams are applied to the reservoir input, the state of the reservoir is dependent on the input patterns and can be used to analyze the input. When a pulse is applied, the conductance of the device will increase. If many pulses are applied within a short interval, a large increase in conductance can be achieved; whereas if the inter-pulse distance is sufficiently large, the device relaxes back to its initial high-resistance state.

We devised our experiments based on these aforementioned observations. We have specifically tested the response of our reservoir to 16 4-bit patterns. The 4-bit patterns were encoded into a pulse stream where the high bits are represented by a high pulse and low bits are represented by 0 voltage (Figure 2a,b). The state

of the diffusive memristor is read after the application of the encoded pulse stream through a read pulse (0.1 V).

A set of three experiments were devised to investigate the effect of waveform design on the filament evolution of the diffusive memristor. In the first set of experiments, we had applied pulses of 200 μ s with 100 μ s inter-pulse distance (Figure S1b, Supporting Information). In the second set of experiments, we had applied pulses of 100 μ s with 100 μ s inter-pulse distance (Figure 2a). The first set of experiments led to a random distribution of the read currents (Figure S1a, Supporting Information), whereas the second set of experiments led to more uniform distribution of read currents (Figure 2c). This can be explained by the fact that the complete rupture of the filament in the diffusive memristor is substantially random, whereas the thinning of the filament is more predictable. This happened because in the first set of experiments the state of the memristor was read hundreds of microseconds after the bit pattern was applied, whereas in the second case the state of the memristor was read 5 μ s after the bit pattern was applied. Hence, the first set of experiments allowed the device to relax more (filament got ruptured) than the second set of experiments and that led to more nonuniform read currents.

Although the read current distribution of the second set of experiments was relatively more uniform compared with the first set, there were several outliers in the data that needed to be removed to make this data suitable for further application. A third set of experiments were devised where we used an initial excitatory pulse (0.8 V, 700 μ s). This long relatively low voltage pulse excited the device to a partially conductive state after which the aforementioned coded pulse patterns were applied (Figure 2b). This set of experiments yielded a much more uniform set of read currents as shown (Figure 2d). The device conductance was read at two points. One was in the 30 μ s read voltage window that was applied right after the bit pattern terminated, and another point was in a 200 μ s read voltage window that was applied hundreds of microseconds after the bit pattern terminated. The conductivity of the device was less uniform in the read window that was applied later compared to the former. This can be explained again using the rationale that relaxation of the device is stochastic in nature and the relaxation dynamics of the device dominate the current distribution relatively to a higher extent in the latter read window. The filament is not completely ruptured but just thinned.

We then used the read current data that we had gathered from our initial set of experiments for the purpose of MNIST handwritten digit classification. The recorded read currents from the aforementioned experiments can be thought of as responses of the reservoir to a combination of black and white pixels. The white and black pixels are represented by a high write pulse (1.25 V, 50 μ s) and no pulse (0 V amplitude), respectively. Each MNIST handwritten digit image has 28 \times 28 pixels. Each image is cropped to 22 \times 20 pixels and then binarized in a software program (Figure 3a). The columns are then divided into five sets of four columns each and then joined one above the other to form a 110(22 \times 5) \times 4 matrix. All these 4-bit rows in this matrix are a subset of the 4-bit patterns that have been applied to the diffusive memristor-based RC system. The corresponding currents for each row are extracted in a random manner from a set of 100 measured read current values (for a certain bit pattern). These current values are then applied to the input of 220 (110 \times 2 memristor per

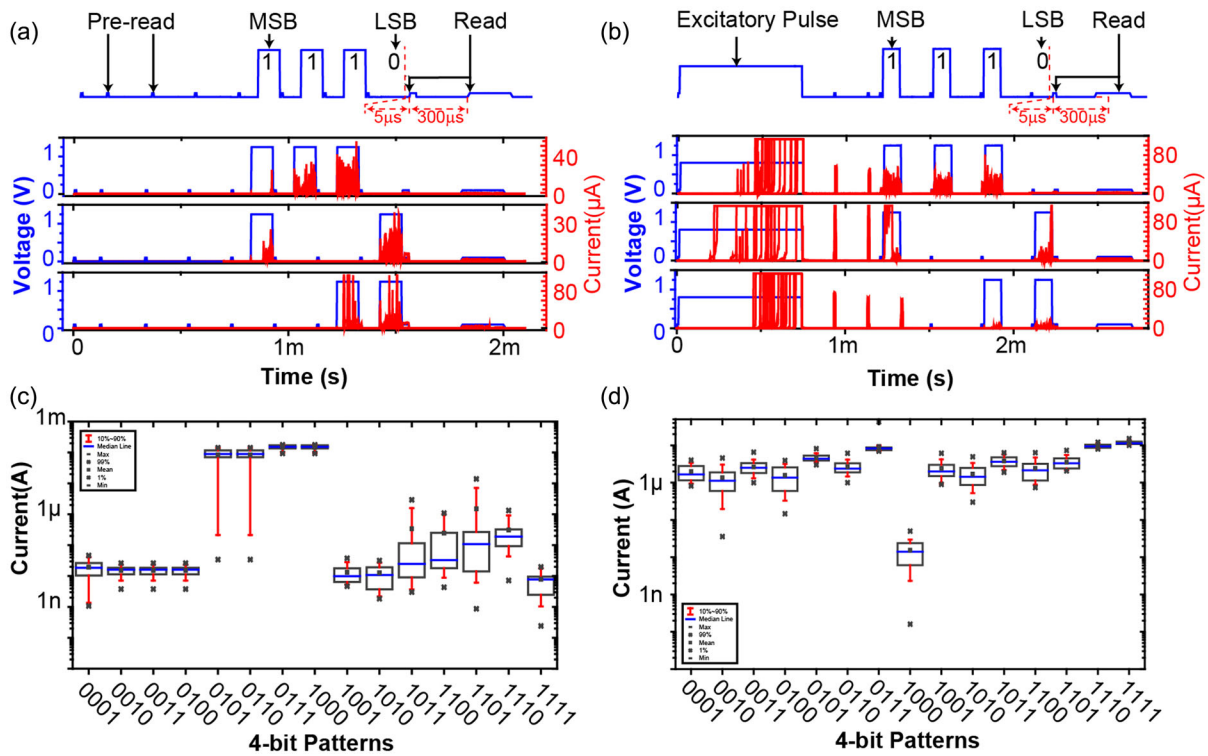


Figure 2. Waveform Design for the Reservoir: First set data can be found in Figure S1, Supporting Information. a) Schematic illustration of second set waveforms. 100 μ s, 1.25 V pulses signify 1, whereas 0 V pulse signifies 0. The 4-bit patterns are encoded in the form of four pulses. The first pulse in the pattern signifies the most significant bit (MSB), whereas the last pulse in the pattern signifies the least significant bit (LSB). As an example, in the topmost pulse pattern, the MSB is high or “1,” whereas the LSB is low or “0.” Pre-Read pulses are applied to ensure that the device is at its initial state before the application of a new pulse. The horizontal dashed red line indicates the end of the bit pattern. After the application of the pulse pattern with a 4-bit pattern encoded, the state of the memristor is read at two points. The first one is a fast read that is done around 5 μ s after the pulse pattern ends, and the second one is a slow read that is done around 300 μ s after the pulse pattern ends. The top panel shows the pattern. The bottom three panels show the current responses of the reservoir to the applied bit patterns. The applied bit patterns are (from top to bottom) 1110, 1001, and 0011. b) Schematic illustration of third set waveforms. 100 μ s, 1.25 V pulses signify 1, whereas 0 V pulse signifies 0. In this set of experiments, a 700 μ s, 0.8 V excitatory pulse is applied at the beginning of the pattern to ensure that the device is in an already low resistance state. The 4-bit patterns are encoded in the form of four pulses. The first pulse in the pattern signifies the MSB or the most significant bit, whereas the last pulse in the pattern signifies the LSB or the least significant bit. As an example, in the topmost pulse pattern, the MSB is high or “1,” whereas the LSB is low or “0.” Pre-Read pulses are applied to ensure that the device is at its initial state before the application of a new pulse. The horizontal dashed red line indicates the end of the bit pattern. After the application of the pulse pattern with a 4-bit pattern encoded, the state of the memristor is read at two points. The first one is a fast read that is done right after the pulse pattern ends, and the second one is a slow read that is done around 300 μ s after the pulse pattern ends. The top panel shows the pattern. The bottom three panels show the current responses of the reservoir to the applied bit patterns. The applied bit patterns are (from top to bottom) 1110, 1001, and 0011. c) The distribution of current responses of a diffusive memristor corresponding to all possible 4-bit inputs including those illustrated in panel (a). One hundred measurements were experimentally conducted for each 4-bit input at the fast read. d) The distribution of current responses of a diffusive memristor corresponding to all possible 4-bit inputs for this set of data including those shown in panel (b). One hundred measurements were experimentally conducted for each 4-bit input at the fast read.

differential pair) $\times 10$ Fully Connected Neural Network (Figure 3b), where each differential pair represents a single signed weight of the 1T1R network. The fully connected layer serves the role of the readout where the classification is performed. During readout, the output neurons of the fully connected 1T1R crossbar applied softmax activations to the dot product of the 220 inputs and the weights associated with each output neuron. The readout layer is trained in a supervised fashion based on error backpropagation that uses RMS prop method to minimize a cross-entropy loss (see Experimental Section) that updates the conductance of 1T1R every mini-batch. This process is repeated with two epochs of passing all the 60 000 handwritten digits from the MNIST

training data set, and it is then tested with 10 000 digits from the MNIST test set.

The 1T1R readout layer quickly learns the classification with temporal feature maps in the experimental testing. The experimental accuracy/loss curve tightly follows the simulation that uses software reservoir neurons and software readout, or software readout alone (Figure 4a,b), indicating robust in situ learning of the 1T1R array using a one-shot blind weight update.^[37–39] After the training, the network could correctly classify $\approx 83\%$ of the MNIST test set. The misclassifications were mainly with identifying digit “5” or “8” as “3,” which are also hard to be distinguished by human beings at the 20×16 resolution (Figure 4c).

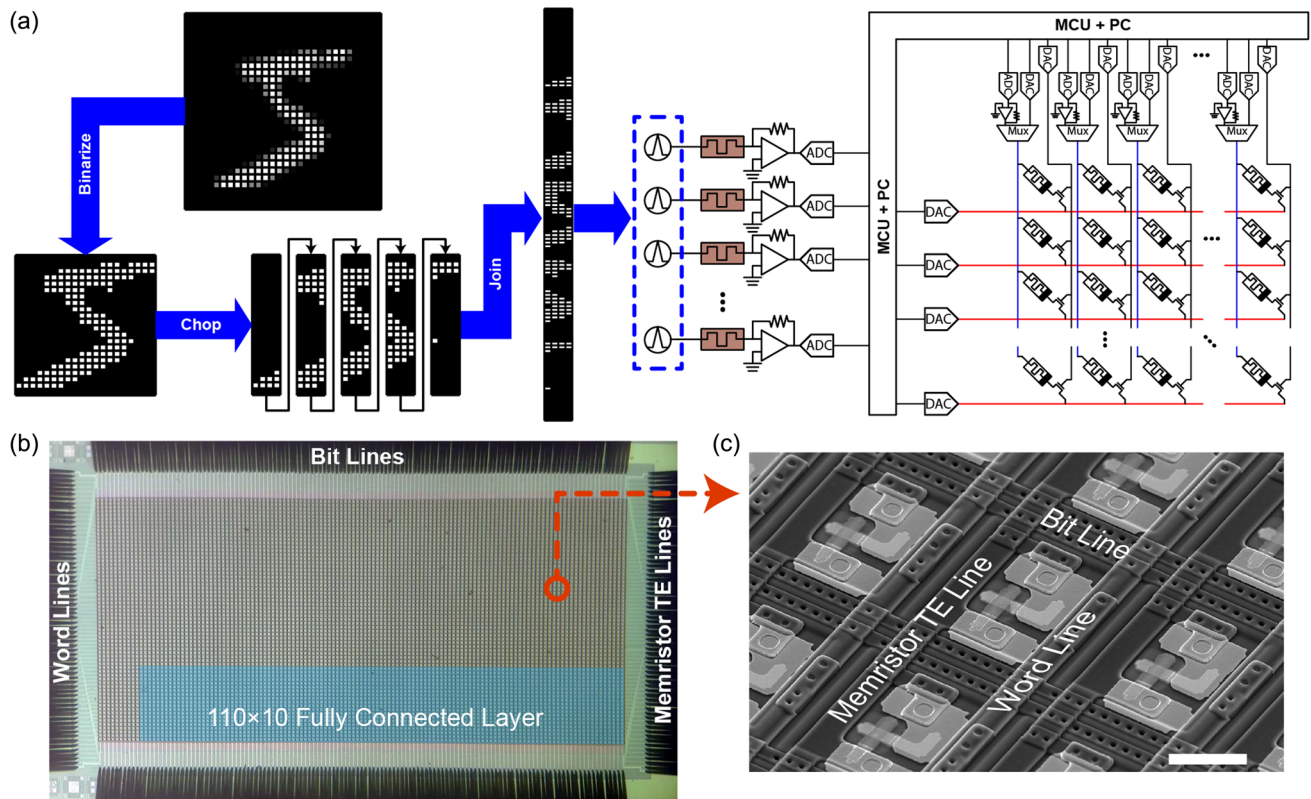


Figure 3. Schematic of the diffusive memristor-based dynamic reservoir for classifying MNIST-based temporal sequences. a) The original 28×28 image is binarized and cropped to size 22×20 , before being divided into five columns and sequentially joined with each other. The resultant 110×4 input is then converted to 110 4-bit temporal voltage patterns as inputs to diffusive memristors. The current responses of the diffusive memristors are used to in situ train the 1T1R crossbar array. b) Optical micrograph of the 128×64 1-transistor 1-memristor (1T1R) crossbar with the probe card landed. The color blocks illustrate the sub-array used for implementing the readout layer, consisting of 110×20 devices using differential pairs. c) Scanning electron micrograph of a single 1T1R cell in (b). 1T1R cells of the same row share the common bottom bit line, while those of the same column share the common word line and memristor top electrode (TE) line. Scale bar: $20 \mu\text{m}$.

The effect of the training is also reflected by the broadened quasi-normal distribution of the weights of the readout layer (Figure 4d).

A significant advantage of using the RC system is the reduction of network size and training cost. A conventional neural network for this task will have 440×2 ($22 \times 20 \times 2$) inputs corresponding to the 440 pixels (differential pairs) and minimum 10 outputs. Even without any hidden layers, that is, with the 440 inputs directly connected to the 10 outputs forming a 440×10 network, 4440 weights need to be trained. This number will grow very quickly if one or more hidden layers are used. In the RC system, the spatial information is encoded in the temporal domain so a smaller network (e.g., a 220×10 readout function with only 2200 weights) needs to be trained, while the reservoir consisting of only one memristor does not need training.

In conclusion, we have developed an RC system by using a diffusive memristor reservoir element. The rich dynamics of the diffusive memristor empowers the reservoir to faithfully extract critical features from the inputs. The extracted features are inherently encoded in the form of currents that are then provided to a drift memristor-based 1T1R layer for classification. We use this reservoir system for a MNIST handwritten digit classification task. The entire training dataset of 60 000 images is used to train the network, followed by which 10 000 images are used

for testing the network. An accuracy of 83% has been achieved using our RC system.

Experimental Section

Fabrication: The diffusive memristor devices were fabricated on a p-type (100) Si wafer with 100 nm thermal oxide. The bottom electrodes were patterned by photolithography, followed by evaporation and liftoff of $\approx 20/2$ nm Pt/Ti. The ≈ 16 nm thick doped dielectric layer was deposited at room temperature by reactively co-sputtering SiO_2 and Ag in Ar. The ≈ 25 nm Pd top electrodes were subsequently patterned by photolithography, followed by evaporation and liftoff processes. Electrical contact pads of the bottom electrodes were first patterned by photolithography and then subjected to reactive ion etching with mixed CHF_3 and O_2 gases. The synapses were built by integrating drift memristors with foundry-made transistor arrays using back-end-of-the-line (BEOL) processes. Each Pd/Ta₂O₅/Ta memristor^[40] was connected to a series n-type enhancement-mode transistor. Figure 3c shows the detailed structure of a single 1T1R cell and associated connections. When all the transistors are turned on, the 1T1R array works as a fully connected memristor crossbar.

Measurement Set-Up: An in-house measurement system was built to electrically read and write the 1T1R chip.^[41] The system featured $128 + 64 + 64$ -way concurrent analog voltage inputs (up to ± 10 V) with a minimum pulse width of ≈ 100 ns and parallel current sensing capability. Each row wire or gate wire of the 1T1R memristor array could be

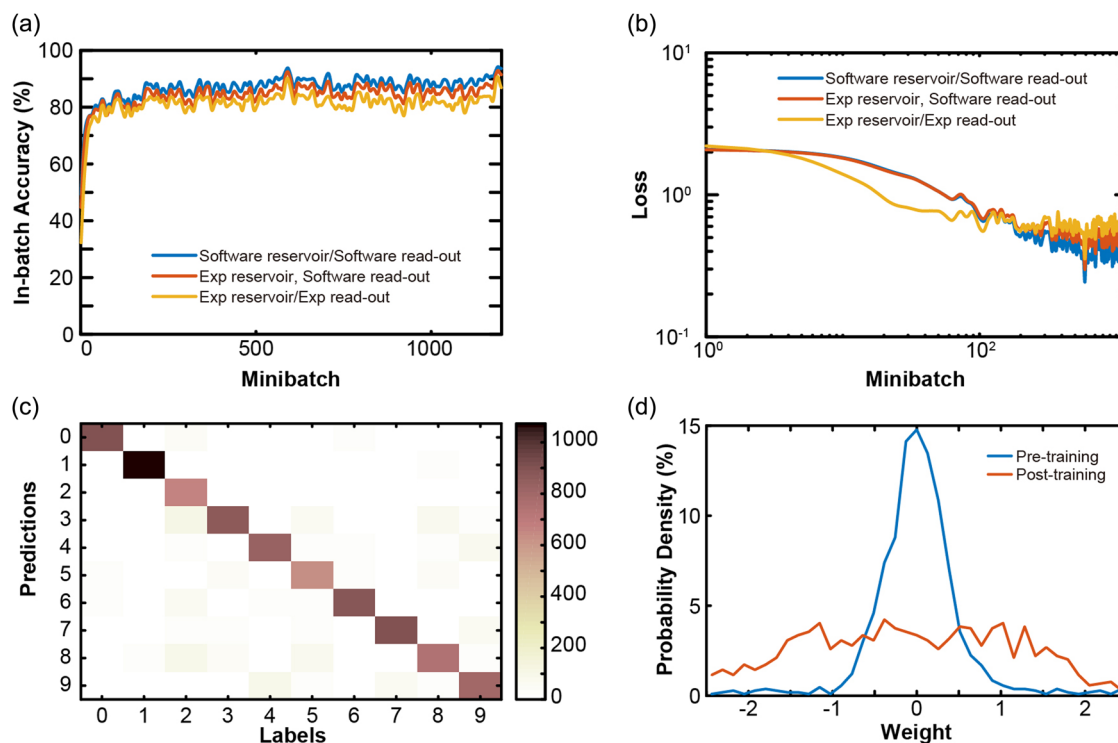


Figure 4. In situ training of diffusive memristor reservoir with 1T1R fully connected array for MNIST classification. a) The in-batch accuracy along the 2-epoch in situ training. The experimental curve tightly follows the simulation. b) The loss along the training. c) The inference result on classifying the MNIST test set. d) The distribution of the weights of the 1T1R fully connected layer prior to and after the training.

independently configured for voltage biasing, ground, or high-impedance. For voltage biasing, each wire was driven by a channel of a digital-to-analog converters (DAC) (LTC2668, Analog Devices). The DAC communicates with the micro-controller unit (MCU) via digital I/O ports. A fast trigger pulse controlled by the MCU allowed a minimum ≈ 100 ns DAC pulse generation. Each column wire of the 1T1R memristor array could be independently configured for voltage biasing, ground, high-impedance, or current sensing. The voltage biasing was implemented in the same manner with the row or gate wires. For current sensing, each column wire connected to one of the four transimpedance amplifiers (TIAs) (LTC6268, Analog Devices) with different gains. The voltage outputs of the TIAs were read by the analog-to-digital converters (ADCs) (MAX11046, Maxim Integrated) and fetched by the MCU via its digital I/O before being sent back to the computer.

For comparison, the same number (i.e., 110) of software non-leaky integrator neurons was used. For the i -th software neuron with a 4-bit input stream x_i^t (where $1 \leq t \leq 4$), the neuron output was the sum $\sum_{1 \leq t \leq 4} x_i^t$. Both experimental and software neuron outputs were normalized to voltages that the maximum voltage of each kind was 0.1 V, before being fed to the 1T1R array.

The voltage pulses applied to the diffusive memristors and the currents across them were generated and measured with a Keysight B1530 waveform generation function measurement unit (WGFMU), which supports concurrent voltage sourcing and current sensing. The Keysight B1530 was controlled by a customized software that passes the data of measured currents to a customized MATLAB program that communicates with an MCU via serial ports. The digital IOs of the MCU connected to the printed circuit boards (PCB) comprising of DACs, ADCs, and TIAs, which drive the 1T1R array via a probe card.

Basic Electrical Array Operations: The basic electrical operations of the 1T1R memristor array included potentiation programming, weight readout, and vector matrix multiplication. All operations were performed by the measurement system with the aid of the on-chip transistors. For potentiation programming, the memristor array was programmed row by row.^[42]

To program a selected row, all row wires was floated, except the selected one, which was grounded. Each gate wire was assigned a different voltage based on the targeted conductance of the associated memristor. All column wires (TEs of memristors) were biased with the same SET voltages. Therefore, this scheme programmed all memristors of the same row simultaneously.

Supporting Information

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

Acknowledgements

R.M. and Z.W. contributed equally to this work. The work was also partially supported by Air Force Office of Scientific Research (AFOSR) for the support through the MURI program under contract number FA9550-19-1-0213. This work was supported in part by the US Air Force Research Laboratory (AFRL) (Grant No. FA8750-18-2-0122). Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL or AFOSR.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

diffusive memristors, drift memristors, modified national institute of standards and technology, readout layers, reservoir computing

Received: July 19, 2019
Revised: August 25, 2019
Published online: September 25, 2019

- [1] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*, Auerbach Publications, Boca Raton **2016**.
- [2] J. Schmidhuber, *Neural Networks* **2015**, 61, 85.
- [3] D. P. Mandic, J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, John Wiley & Sons, Inc., New York **2001**.
- [4] H. Jaeger, H. Haas, *Science* **2004**, 304, 78.
- [5] H. Jaeger, in *German National Research Center for Information Technology GMD Technical Report*, Vol. 148, Bonn, Germany **2001**, p. 13.
- [6] W. Maass, T. Natschläger, H. Markram, *Neural Comput.* **2002**, 14, 2531.
- [7] M. Lukoševičius, H. Jaeger, *Comput. Sci. Rev.* **2009**, 3, 127.
- [8] J. J. Steil, in *2004 IEEE Int. Joint Conf. on Neural Networks (IEEE Cat. No.04CH37541)*, Vol. 842, IEEE, Budapest **2004**, pp. 843–848.
- [9] H. Jaeger, *Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the “Echo State Network” Approach*, Vol. 5, GMD-Forschungszentrum Informationstechnik Bonn, Bonn **2002**.
- [10] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, S. Massar, *Sci. Rep.* **2012**, 2, 287.
- [11] J. Torrejon, M. Riou, F. A. Araujo, S. Tsunegi, G. Khalsa, D. Querlioz, P. Bortolotti, V. Cros, K. Yakushiji, A. Fukushima, H. Kubota, *Nature* **2017**, 547, 428.
- [12] J. Misra, I. Saha, *Neurocomputing* **2010**, 74, 239.
- [13] J. Hasler, H. B. Marr, *Front. Neurosci.* **2013**, 7, 118.
- [14] D. B. Strukov, G. S. Snider, D. R. Stewart, R. S. Williams, *Nature* **2008**, 453, 80.
- [15] R. Waser, M. Aono, in *Nanoscience and Technology: A Collection of Reviews from Nature Journals*, World Scientific, Singapore **2010**, pp. 158–165.
- [16] J. J. Yang, D. B. Strukov, D. R. Stewart, *Nat. Nanotechnol.* **2013**, 8, 13.
- [17] Y. Yang, P. Gao, S. Gaba, T. Chang, X. Pan, W. Lu, *Nat. Commun.* **2012**, 3, 732.
- [18] Z. Wang, S. Joshi, S. E. Savel'ev, H. Jiang, R. Midya, P. Lin, M. Hu, N. Ge, J. P. Strachan, Z. Li, Q. Wu, *Nat. Mater.* **2017**, 16, 101.
- [19] T. Ohno, T. Hasegawa, T. Tsuruoka, K. Terabe, J. K. Gimzewski, M. Aono, *Nat. Mater.* **2011**, 10, 591.
- [20] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, W. Lu, *Nano Lett.* **2010**, 10, 1297.
- [21] M. D. Pickett, G. Medeiros-Ribeiro, R. S. Williams, *Nat. Mater.* **2013**, 12, 114.
- [22] J. Lin, Annadi, S. Sonde, C. Chen, L. Stan, K. V. L. V. Achari, S. Ramanathan, S. Guha, in *2016 IEEE Int. Electron Devices Meeting (IEDM)*, IEEE, San Francisco **2016**, pp. 34.35.31–34.35.34.
- [23] A. Jaiswal, S. Roy, G. Srinivasan, K. Roy, *IEEE Trans. Electron Devices* **2017**, 64, 1818.
- [24] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, E. Eleftheriou, *Nat. Nanotechnol.* **2016**, 11, 693.
- [25] Z. Wang, S. Joshi, S. Savel'ev, W. Song, R. Midya, Y. Li, M. Rao, P. Yan, S. Asapu, Y. Zhuo, H. Jiang, *Nat. Electron.* **2018**, 1, 137.
- [26] M. Prezioso, F. Merrikkh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, D. B. Strukov, *Nature* **2015**, 521, 61.
- [27] P. M. Sheridan, F. Cai, C. Du, W. Ma, Z. Zhang, W. D. Lu, *Nat. Nanotechnol.* **2017**, 12, 784.
- [28] F. Alibart, E. Zamanidoost, D. B. Strukov, *Nat. Commun.* **2013**, 4, 2072.
- [29] P. Yao, H. Wu, B. Gao, S. B. Eryilmaz, X. Huang, W. Zhang, Q. Zhang, N. Deng, L. Shi, H. S. Wong, H. Qian, *Nat. Commun.* **2017**, 8, 15199.
- [30] M. Suri, D. Querlioz, O. Bichler, G. Palma, E. Vianello, D. Vuillaume, C. Gamrat, B. DeSalvo, *IEEE Trans. Electron Devices* **2013**, 60, 2402.
- [31] A. Serb, J. Bill, A. Khat, R. Berdan, R. Legenstein, T. Prodromakis, *Nat. Commun.* **2016**, 7, 12611.
- [32] E. Covi, S. Brivio, A. Serb, T. Prodromakis, M. Fanciulli, S. Spiga, *Front. Neurosci.* **2016**, 10, 482.
- [33] W. Maass, *Neural Networks* **1997**, 10, 1659.
- [34] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, W. D. Lu, *Nat. Commun.* **2017**, 8, 2204.
- [35] R. Midya, Z. Wang, J. Zhang, S. E. Savel'ev, C. Li, M. Rao, M. H. Jiang, S. Joshi, H. Jiang, P. Lin, K. Norris, *Adv. Mater.* **2017**, 29, 1604457.
- [36] W. Wang, M. Wang, E. Ambrosi, A. Bricalli, M. Laudato, Z. Sun, X. Chen, D. Ielmini, *Nat. Commun.* **2019**, 10, 81.
- [37] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, *Nat. Commun.* **2018**, 9, 2385.
- [38] C. Li, Z. Wang, M. Rao, D. Belkin, W. Song, H. Jiang, P. Yan, Y. Li, P. Lin, M. Hu, N. Ge, *Nat. Mach. Int.* **2019**, 1, 49.
- [39] Z. Wang, C. Li, W. Song, M. Rao, D. Belkin, Y. Li, P. Yan, H. Jiang, P. Lin, M. Hu, J. P. Strachan, *Nat. Electron.* **2019**, 2, 115.
- [40] J. J. Yang, M. X. Zhang, J. P. Strachan, F. Miao, M. D. Pickett, R. D. Kelley, G. Medeiros-Ribeiro, R. S. Williams, *Appl. Phys. Lett.* **2010**, 97, 232102.
- [41] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, *Nat. Electron.* **2018**, 1, 52.
- [42] Z. Wang, C. Li, W. Song, M. Rao, D. Belkin, Y. Li, P. Yan, H. Jiang, P. Lin, M. Hu, J. P. Strachan, *Nat. Electron.* **2019**, 2, 115.