nature COMMUNICATIONS

ARTICLE

Received 30 Aug 2014 | Accepted 15 May 2015 | Published 25 Jun 2015

DOI: 10.1038/ncomms8522

Associative memory realized by a reconfigurable memristive Hopfield neural network

S.G. Hu[1], Y. Liu[1], Z. Liu[2], T.P. Chen[3], J.J. Wang[1], Q. Yu[1], L.J. Deng[1], Y. Yin[4] & Sumio Hosaka[4]

Although synaptic behaviours of memristors have been widely demonstrated, implementation of an even simple artificial neural network is still a great challenge. In this work, we demonstrate the associative memory on the basis of a memristive Hopfield network. Different patterns can be stored into the memristive Hopfield network by tuning the resistance of the memristors, and the pre-stored patterns can be successfully retrieved directly or through some associative intermediate states, being analogous to the associative memory behaviour. Both single-associative memory and multi-associative memories can be realized with the memristive Hopfield network.

[1] State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu 610054, China. [2] School of Materials and Energy, Guangdong University of Technology, Guangzhou 510006, China. [3] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore. [4] Graduate School of Engineering, Gunma University, 1-5-1Tenjin, Kiryu, Gunma 376-8515, Japan. Correspondence and requests for materials should be addressed to Y.L. (email: yliu1975@uestc.edu.cn) or T.P.C. (email:echentp@ntu.edu.sg).

The idea of building a cognitive system that can adapt like the biological brain has existed for a long time[1]. However, building an artificial brain with the conventional digital computer based on von-Neumann paradigm[2] is of great difficulty. Digital computers and biological brains process information in fundamentally different ways. Digital computers process information in sequence and are inflexible, precise and deterministic[3,4]; in contrast, biological brains process data in parallel and are flexible, not precise, error-prone and good at learning new matters[1,4]. Obviously, to efficiently realize the functionalities of biological brains, new computing architectures are required. In the past few decades, artificial neural networks (ANNs) have received much attention, as they have a natural capability for storing information and making it available for use[5]. In 1980s, Hopfield proposed a dynamic ANN called Hopfield network[6–9]. The Hopfield network has been proved useful in content-addressable memories[10], and combinatorial optimization problems, such as the travelling salesman problem and the location allocation problem[8,11]. Previous Hopfield network was realized by constructing complementary metal–oxide–semiconductor circuits as the synapses at a cost of large chip area and power consumption[10]. In 1971, Chua predicted the fourth basic circuit element, namely, memristor[12,13], which was later demonstrated in the laboratory by Williams *et al.* in 2008 (ref. 14). Subsequently, many studies demonstrated that a memristor can be used as an electronic synapse with its conductance representing the synaptic weight[15–25]. Although synaptic operation of memristors has been widely demonstrated, implementation of even a simple ANN is challenging. Encouragingly, some significant advances have been reported recently. For example, ANN consisting of neurons and synapses has been constructed to realize the Pavlov's dog model[26–28]; Alibart *et al.* reported the realization of linear pattern classification using a memristive network[29]; Park *et al.* realized neuromorphic speech systems using resistive random-access memory -based synapse[30]; and Eryilmaz *et al.* reported brain-like associative learning using phase-change synaptic device array[31]. Burr *et al.* demonstrated a neural network with 165K synapses implemented with phase-change devices[32].

In this work, we have successfully constructed a Hopfield network using HfO_2 memristors and peripheral devices to realize the associative memory that is capable of retrieving a piece of data upon presentation of partial information from that piece of data. The network can be reconfigured to realize various positive and negative synaptic weights. Both single-associative memory and multi-associative memories can be realized with the memristive Hopfield network (MHN). Associative memories via or not via intermediate states can be used to emulate humans' 'weak' or 'strong' memories, respectively. In addition, the proposed MHN shows good robustness to device variation and variations in threshold voltage of the neurons. This study provides a possible method for hardware implementation of artificial neuromorphic networks to emulate memorization.

Results

Memristor characterization. The memristor used in this work has a metal/oxide/metal structure, as shown in Fig. 1a. After a forming process with high voltage, resistance of the memristor can be increased or decreased by voltage bias depending on the voltage polarity. Figure 1b shows the I–V characteristics of five repeated cycles of voltage sweeping for a typical memristor. Each cycle of sweeping takes ~10 s and the period of each applied voltage is 40 ms. For the positive voltage polarity, the current increases very little with voltage at low voltages, but it rises rapidly when the voltage is increased to ~1 V; for the negative voltage polarity, the current increases with the voltage magnitude, but it decreases gradually when the voltage reaches ~ −1 V.

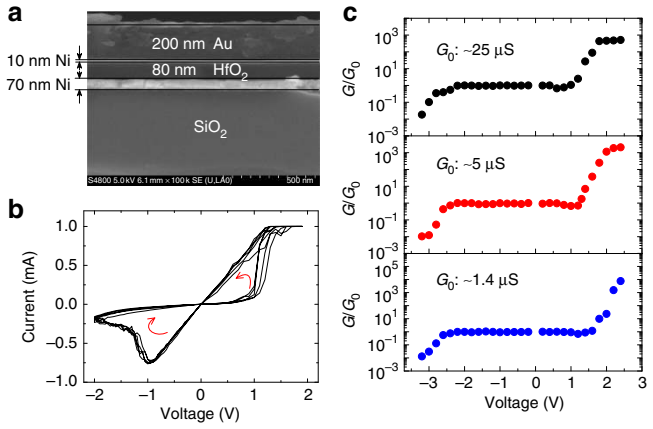

Figure 1 | The HfO_2-based memristor. (**a**) Scanning electron microscopic image of the cross-section of the memristor; (**b**) I–V characteristics of a typical memristor measured for five repeated cycles of voltage sweeping with the compliance current of 1 mA; and (**c**) conductance (G) of the memristors with different initial conductance (G_0) as a function of the pulse voltage, with the pulse width fixed at 5 ms.

Besides the voltage sweeping, voltage pulses can also be used to change the conductance of the memristor. Figure 1c shows the conductance change of the memristors with different initial conductance (G_0) for the pulse duration of 5 ms. Generally, a positive or negative pulse voltage leads to an increase or decrease in the conductance, respectively. The conductance shows little change for small pulse voltage, but the change is large for a large pulse voltage (for example, +2 V for positive pulse voltage, −2 V for negative pulse voltage). The result indicates that the resistance of the memristor can be adjusted to the desired values with an appropriate voltage–pulse programming scheme (see Supplementary Fig. 1). The conductive filament (CF) model can be used to explain the resistance change in the HfO_2-based resistive memristors[33,34]. Considering the bipolar nature of the memristor, the formation or rupture of some CF consisting of oxygen vacancies are responsible for the resistance change[35]. In the initialization process (that is, the forming process), CF are formed in the HfO_2 thin film to connect the two electrodes, resulting in a low-resistance state. Subsequently, a negative (positive) voltage can lead to the gradual rupture (recovery) of CF, resulting in an increase (decrease) in the resistance. The memristive switching from high-resistance state to low-resistance state could be attributed to CF formation at the grain boundaries containing a high concentration of oxygen vacancies[33]. On the other hand, electric field could play an important role in switching from low-resistance state to high-resistance state[34]. It was suggested that electrical pulses lead to a progressive narrowing of the CF, and finally a gap is formed and the memristor switches to the high-resistance state[33,34].

MHN implementation. Basic findings from the biological neuron operation have enabled researchers to model the operations of artificial neurons[36]. A Hopfield network consists of a set of interconnected artificial neurons and synapses. In this work, a Hopfield network is constructed with nine synapses realized with six memristors and three neurons. As shown in Fig. 2a, the artificial neuron has three inputs and each input, N_i ($i = 1$, 2

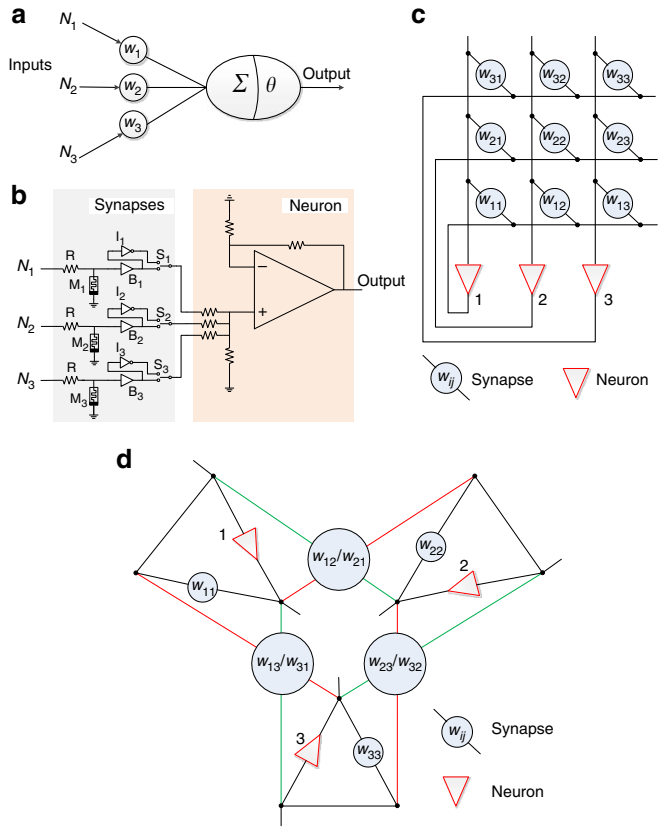

Figure 2 | The MHN. (**a**) Mathematical abstraction of the neuron model; (**b**) circuit schematic of the designed 3-bit neuron; (**c**) architecture of the 3-bit MHN consisting of nine memristors; and (**d**) architecture of the MHN with symmetrical configuration consisting of six memristors.

and 3), is connected to a synapse with synaptic weight of w_i. The output of the three-input binary artificial neuron is expressed as

$$y = \mathrm{sign}\left(\sum_{i=1}^{3} \omega_i N_i - \theta\right) \quad (1)$$

where θ is the neuron's threshold; and the sign function is defined as:

$$\mathrm{sign}(N) = \begin{cases} 1 & \text{if } N \geq 0 \\ 0 & \text{if } N < 0 \end{cases} \quad (2)$$

An artificial neuron was constructed, as shown in Fig. 2b. An operational amplifier is used to sum the inputs. The switches, S_1, S_2 and S_3, are controlled by external signals to obtain positive or negative synaptic weights. The synaptic weights corresponding to input N_1, N_2, and N_3 are $w_1 = \pm\frac{M_1}{M_1+R}$, $w_2 = \pm\frac{M_2}{M_2+R}$ and $w_3 = \pm\frac{M_3}{M_3+R}$, respectively ($M_1$, M_2 and M_3 are the resistance of the memristors, respectively, and the resistance of R is fixed at 3 MΩ). In the circuit shown in Fig. 2b, transmission gates B_1, B_2 and B_3 are used to transfer signals without modifying the polarity of the signals; inverters I_1, I_2 and I_3 are used to achieve negative synapse weights.

Figure 2c shows the architecture of a 3-bit MHN realized with nine synapses. The synaptic weight from neuron i to neuron j is given by $w_{i,j}$, which can be conveniently adjusted by tuning the resistance of the corresponding memristor M_{ij}. M_{ij} and w_{ij} are represented by the resistance matrix $\mathbf{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}$ and the synaptic weight matrix $\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{pmatrix}$, respectively. As the Hopfield network is symmetric, that is, $M_{12} = M_{21}$, $M_{23} = M_{32}$ and $M_{13} = M_{31}$, the network can be realized with only six memristors as shown in Fig. 2d (see Supplementary Fig. 2 for the complete circuit schematic of the MHN). And all the discussions below are based on this optimized architecture. The threshold of the artificial neurons (neurons 1, 2 and 3) are represented by the threshold vector $\mathbf{T} = (\theta_1\ \theta_2\ \theta_3)$; and the states of the three neurons are represented by the state vector $\mathbf{X} = (x_1\ x_2\ x_3)$, where x_1, x_2 and x_3 are the states of neurons 1, 2 and 3, respectively. In an updating cycle, new states of the neurons are updated according to the function:

$$\mathbf{X}(t+1) = \mathrm{sign}(\mathbf{X}(t) \cdot \mathbf{W} - \mathbf{T}) \quad (3)$$

where t represents the number of updating cycles and $t = 0$ represents no update taking place and the corresponding state vector is the initial vector $\mathbf{X}(0)$. In one updating cycle, new states of the neurons are asynchronously updated from x_1, x_2 to x_3 in three stages, which are defined as stages a, b and c, respectively.

Single-associative memory. Associative memory is a function of brain that is capable of recalling a piece of data on the information relevant to that piece of data. In this work, patterns are stored into the MHN by tuning the resistance matrix $\mathbf{M}$ to obtain the desired weight matrix $\mathbf{W}$ (refs 11,36). An optimized scheme based on the outer-product (Hebbian) rule was employed to determine the weight matrix[6,37]. The relationship between the initial state and the final state is determined by the weight matrix, threshold and the refreshing sequence. The target memory that needs to be associatively recalled was set at '110'. To store the pattern binary '110' into the MHN, the resistance matrix was set as

$$\mathbf{M} = \begin{pmatrix} 0.1 & 220 & 41 \\ 220 & 0.1 & 211 \\ 41 & 211 & 0.1 \end{pmatrix} \mathrm{k\Omega} \quad (4)$$

By selecting a proper switch state for each synapse, the weight matrix was set as

$$\mathbf{W} = \frac{1}{60}\begin{pmatrix} 0 & 4.10 & 0.81 \\ 4.10 & 0 & -3.94 \\ 0.81 & -3.94 & 0 \end{pmatrix} \quad (5)$$

To achieve the targeted matrixes in equations (4) and (5), the resistances of the relevant memristors are tuned by applying step-like voltage pulses with an appropriate scheme of voltage magnitudes and pulse numbers to the memristors, which is called the training process. An offline training scheme for setting the predetermined resistances on the memristors is implemented with a C Language program embedded in the semiconductor characterization system (Keithley 4200). The training process for M_{13}/M_{31}, M_{12}/M_{21} and M_{23}/M_{32} is illustrated in Supplementary Fig. 3. On the other hand, the targeted resistances of M_{11}, M_{22} and M_{33} can be achieved directly from the low-resistance states that have a resistance of around 0.1 kΩ; thus no training is needed for these elements.

Once the targeted resistances are achieved, they remain unchanged during the network operation. The description of the circuit operation process is presented in Supplementary Note 1.

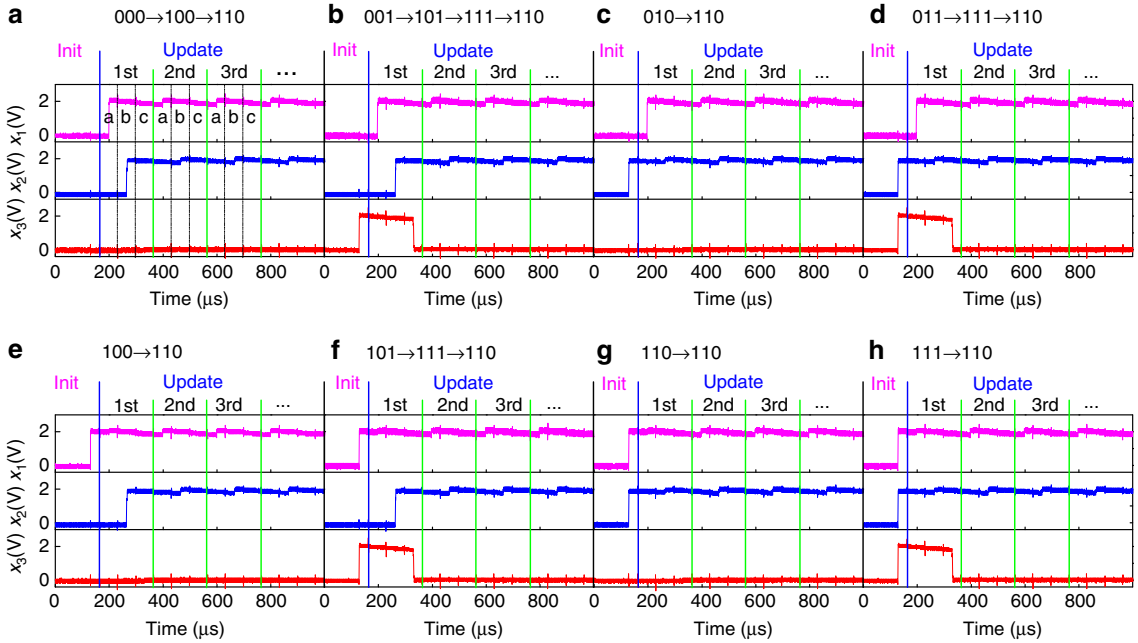

Figure 3 | Waveforms of the states (x_1, x_2 and x_3) for different initial states. The MHN had '110' was pre-stored in it. Initial states of (**a**) '000'; (**b**) '001'; (**c**) '010'; (**d**) '011'; (**e**) '100'; (**f**) '101'; (**g**) '110' and (**h**) '111'.

The threshold vector of the three neurons was set as

$$\mathbf{T} = \frac{1}{60}(-2 \quad -2 \quad -2) \tag{6}$$

If the associative memory works, the MHN can converge to '110' automatically from any state in the range from '000' to '111'. Figure 3 shows the waveforms of state vector $\mathbf{X}(t)$ in the process to retrieve the pre-stored '110'. Clocks at 5 kHz were used to control the MHN. The network requires seven control signals in total (see Supplementary Fig. 4). Figure 3 shows the waveforms of the states (x_1, x_2 and x_3) for different initial states. In each refreshing cycle, three memristors are selected together in one column in the matrix as shown in Fig. 2c and Supplementary Fig. 2. The MHN starting from any initial state vector can successfully retrieve the pre-stored '110'. In an updating cycle, $\mathbf{X}(t)$ was updated in three stages and only 1 bit was updated in one stage. As an example, the updating cycles of the MHN starting from $\mathbf{X}(0) = (0\ 0\ 0)$, as shown in Fig. 3a, are described subsequently. In the first updating cycle, the element x_1 was first updated according to equation (3), and $\mathbf{X}(1)^{a} = (1\ 0\ 0)$ (stage a); x_2 was then updated in stage b according to equation (3) also and $\mathbf{X}(1)^{b} = (1\ 1\ 0)$. Now the MHN 'recalled' the pre-store pattern '110'. In stage c of the first updating cycle and in the following updating cycles, no real updating occurred and the MHN stabilized at '110'. 'Recalling' the '110' by experiencing some intermediate states emulates a weak memorization, that is, sometimes we really think hard to recall a thing via some associative intermediate states: from one thing to another associative one, … and ultimately to the final memory.

For different initial state vectors, the MHN may experience different intermediate state vectors before 'recalling' the pre-stored pattern, as shown in Fig. 3a,b,d,f. For some initial state vectors, intermediate state vectors are not necessary. For example, as shown in Fig. 3h, the MHN started from $\mathbf{X}(0) = (1\ 1\ 1)$ and it directly stabilized at $\mathbf{X}(1) = (1\ 1\ 0)$, and no intermediate states were experienced. Direct memorization emulates a simple associative memory, that is, we can retrieve some strong memories without experiencing associative states. In Fig. 4, the retrievals of the pre-stored '110' from different initial states vectors are schematically summarized using a cube with its each

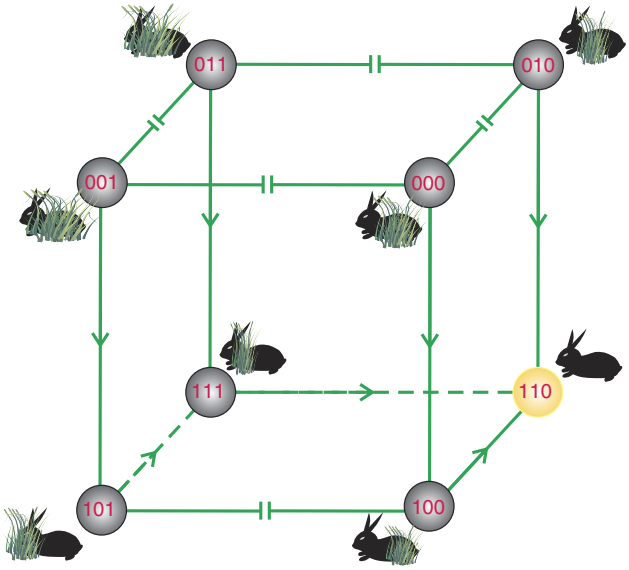

Figure 4 | Schematic illustration of single-associative memory for pre-stored binary code '110' and schematic illustration of associative memory by presentation of different parts of the image of a rabbit.

corner representing a state of the MHN[11]. For different initial state vectors, the MHN may experience different intermediate state vectors before 'recalling' the pre-stored pattern. In addition to convergence to '110', other final states can also be realized by modifying the weight matrix and threshold vector.

The single-associative memory for pre-stored binary code '110' can also be illustrated with the presentation of different parts of the image of a rabbit, as shown in Fig. 4. The image of a full rabbit is equivalent to the final state '110', while the images of the rabbit partially covered up by grass are equivalent to the initial states, such as '000', '100' and so on. State '000', '100' or other initial states are associated with the final state '110' through the synaptic weight matrix in equation (5), threshold vector in equation (6)

and refreshing sequence, and these initial states represent information relevant to the final state '110'. The final state can be retrieved from the initial states; this means that the image of full rabbit can be recalled by associative memorization.

Multi-associative memories. In human brain, one can recall a piece of data on the information relevant to that piece of data by experiencing some associative states; if the given data are different, one can recall another piece of data via experiencing some other intermediate states. In the MHN of this work, more than one pattern can be stored at the same time by reconfiguring the resistances of the memristors. To verify the multi-associative memories, '000' and '101' were pre-stored into the MHN, and the resistance matrix was set as

$$\mathbf{M} = \begin{pmatrix} 0.1 & 56 & 472 \\ 56 & 0.1 & 248 \\ 472 & 248 & 0.1 \end{pmatrix} \mathrm{k}\Omega \quad (7)$$

The offline training process for achieving the predetermined resistances of M_{13}/M_{31}, M_{12}/M_{21} and M_{23}/M_{32} shown in equation (7) is presented in Supplementary Fig. 5. By selecting proper switch states for each synapse, the weight matrix was set as

$$\mathbf{W} = \frac{1}{60}\begin{pmatrix} 0 & 1.1 & 8.16 \\ 1.1 & 0 & 4.58 \\ 8.16 & 4.58 & 0 \end{pmatrix} \quad (8)$$

The threshold vector of the three neurons was set as

$$\mathbf{T} = \frac{1}{60}(6 \quad 6 \quad 6) \quad (9)$$

Figure 5 shows the signal waveforms of $\mathbf{X}(t) = (x_1\ x_2\ x_3)$. As shown in Fig. 5a–d, the MHN could retrieve the pattern '000' when the initial state vectors were $\mathbf{X}(0) = (0\ 0\ 0)$, $\mathbf{X}(0) = (1\ 0\ 0)$, $\mathbf{X}(0) = (0\ 1\ 0)$ or $\mathbf{X}(0) = (1\ 1\ 0)$, respectively. In Fig. 5e–h, the MHN successfully 'recalled' the pre-stored '101' with the initial states $\mathbf{X}(0) = (0\ 0\ 1)$, $\mathbf{X}(0) = (1\ 0\ 1)$, $\mathbf{X}(0) = (0\ 1\ 1)$ and $\mathbf{X}(0) = (1\ 1\ 1)$. Similar to the single-associative memory, the MHN exhibited either strong or weak associative memories. For some initial state vectors, the MHN can directly 'recall' '000' or '101', as they have good associability. Starting from some other initial state vectors, the MHN has to experience associative intermediate state(s) before the success of retrieval, as shown in Fig. 5d,g, due to weak associability. In Fig. 6, we schematically summarize the retrieval of pre-stored '000' and '101' from different initial states vectors in a cube with each corner

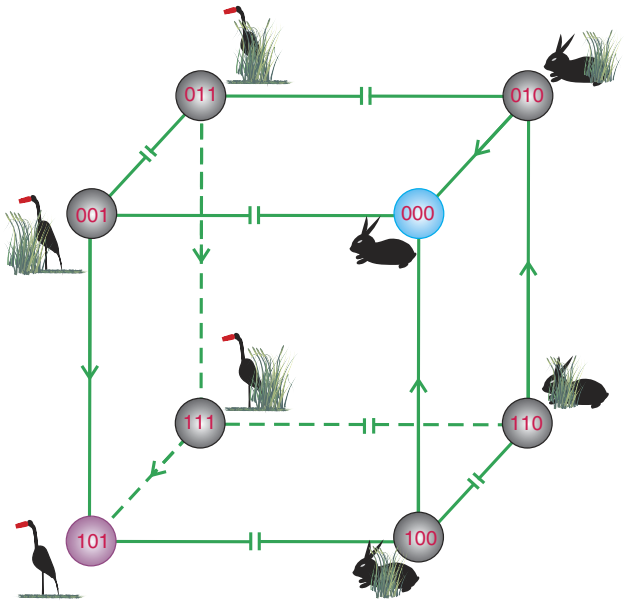

Figure 6 | Schematic illustration of multi-associative memories for pre-stored binary codes '101' (representing the image of a full crane) and '000' (representing the image of a full rabbit).

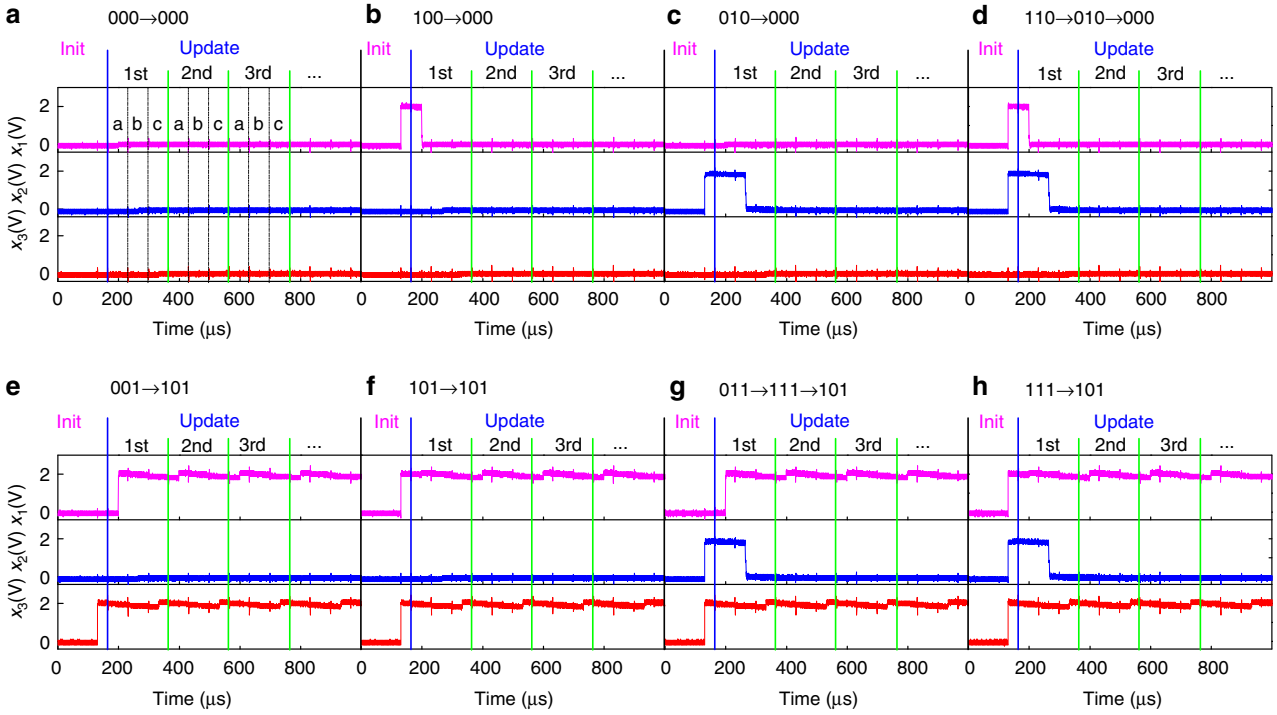

Figure 5 | Waveforms of x_1, x_2 and x_3 for different initial states. The MHN had '000' and '101' pre-stored in it. Initial states of (**a**) '000'; (**b**) '100'; (**c**) '010'; (**d**) '110'; (**e**) '001'; (**f**) '101'; (**g**) '011' and (**h**) '111'.

representing a state of the MHN. For different initial state vectors, the MHN may experience some intermediate state vectors (or does not experience any intermediate state) and finally stabilizes at '000' or '101', realizing the multi-associative memories. In addition to convergence to '000' and '101', some other final states can also be realized by modifying the weight matrix and threshold vector. The multi-associative memories for pre-stored binary codes '000' and '101' can be also illustrated with the presentation of different parts of the images of a rabbit and a crane, respectively, as shown in Fig. 6. The image of a full rabbit is equivalent to the final state '000', while the images of the rabbit partially covered up by grass (or other images relevant to a full rabbit) are equivalent to the initial states '010', '100' and '110'. Similarly, '101' represents the image of a full crane, while the images of the crane partially covered up by grass are equivalent to initial states '001', '011' and '111'. With the information associated to the full rabbit (or the crane), the MHN can successfully recall the full image of the rabbit (or the crane).

Power consumption. Figure 7a,b shows the effect of threshold variation on the network for single- and multi-associative memories, respectively. The error rate is defined as

$$\text{Error rate} = \frac{\text{Number of error states}}{\text{Number of total states}} \times 100\% \qquad (10)$$

An error state means the convergence is done to a wrong final state caused by the variation of threshold voltage (or resistance).The threshold voltage change, ΔV_{TH}, is defined as

$$\Delta V_{TH} = \frac{V_{TH1} - V_{TH0}}{|V_{TH0}|} \times 100\% \qquad (11)$$

where V_{TH0} is the initial threshold voltage ($V_{TH0} = 2\theta$) and V_{TH1} is the threshold voltage after adjustment. As can be observed in Fig. 7a, the ΔV_{TH} tolerance for single-associative memory can be around 100% in the adjustments of both negative and positive directions. For multi-associative memories, the ΔV_{TH} tolerance in the negative direction can be around 25%, while it can be $\sim$15% in the positive direction as shown in Fig. 7b. The threshold variation tolerance of the single-associative memory is better than that of the multi-associative memory.

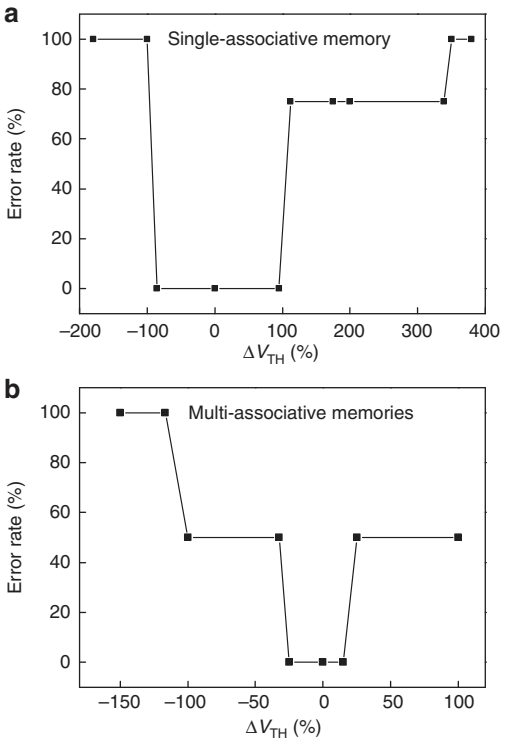

Figure 7 | Threshold voltage variation tolerance. Effect of threshold voltage variation on single-associative memory (**a**) and multi-associative memories (**b**).

Figure 8 shows the effect of resistance variation in the matrix on single-associative memory. The resistance variation ΔR is defined as

$$\Delta R = \frac{R_1 - R_0}{|R_0|} \times 100\% \qquad (12)$$

where R_0 is the initial resistance and R_1 is the resistance after adjustment. As one memristor is used to represent two elements in the symmetric positions in equation (4), elements M_{12} and M_{21} are adjusted together as shown in Fig. 8a. As can be observed in Fig. 8a, in the negative direction (that is, $\Delta R < 0$), the error rate is zero for $\Delta R \geq -40\%$; however, the error rate jumps up to 50% when $\Delta R \leq -52\%$. In the positive direction (that is, $\Delta R > 0$), the MHN did not exhibit any error for a resistance adjustment; even when the resistance is increased by 100%, the error rate is still zero. For M_{23} and M_{32}, the MHN did not exhibit any errors for ΔR in the range of -42 to 44%, as shown in Fig. 8b. As shown in Fig. 8c, M_{13} and M_{31} show up to 290% adjustment tolerance in the positive direction; in the negative direction, the MHN still did not exhibit any error when the resistance is adjusted for -88%. The influence of the resistance variations of M_{11}, M_{22} and M_{33} are also examined. M_{11}, M_{22} and M_{33} are in the low-resistance states with the resistances ranging from several tens of Ohm to below 300 Ω (normally around 100 Ω), which is much smaller than R (= 3 MΩ) in Fig. 2b. Thus the weights of M_{11}, M_{22} and M_{33} are very small, which practically meet the requirement of equation (5). It means that variations in the resistances of memristors M_{11}, M_{22} and M_{33} in the low-resistance states will not cause any error in the network.

Figure 9 shows the effect of resistance variation in the matrix on multi-associative memories. In equation (8), symmetric elements M_{12} and M_{21} are adjusted together as shown in Fig. 9a. As can be observed in Fig. 9a, M_{12} and M_{21} can be adjusted for up to around 166% in the positive direction. In the negative direction, the network still does not exhibit any error when the resistance is adjusted for -86%. As shown in Fig. 9b, M_{23} and M_{32} can be adjusted for up to $\sim$45.6% in the positive direction; in the negative direction, when the resistance is adjusted for -75%, the network still does not exhibit any error. M_{13} and M_{31} show -34.7% adjustment tolerance in the negative direction as shown in Fig. 9c. In the positive direction, the resistance of M_{13} and M_{31} is adjusted for 50%, the network still does not exhibit any error. It is also observed that variations in the resistances of M_{11}, M_{22} and M_{33} in the low-resistance states does not lead to any error in the network.

Effect of resistance and threshold variations on power consumption. The power consumption of the network is around 80 mW. The core memristor array consumes only 100–300 nW; while >99% of the power is consumed by the commercial operational amplifiers (Texas Instruments LM324NP). The power consumption can be improved if the operational amplifiers are optimally designed by integration with the memristors on a single chip. Figure 10a shows the effect of the threshold variation on the power consumption of the core memristor array for single- and multi-associative memories. As can be observed in the figure, the threshold variation does not affect the power consumption for both single- and multi-associative memories. Figure 10b,c shows

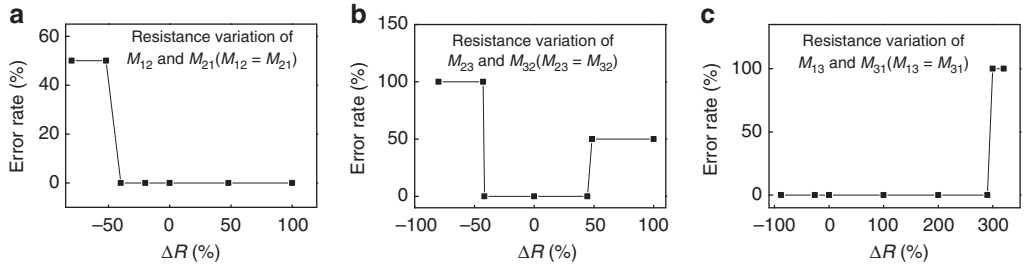

Figure 8 | Resistance variation tolerance of single-associative memory. Effect of resistance variation of the following elements in equation (4) on single-associative memory: (**a**) M_{12} and M_{21}; (**b**) M_{23} and M_{32}; and (**c**) M_{13} and M_{31}.

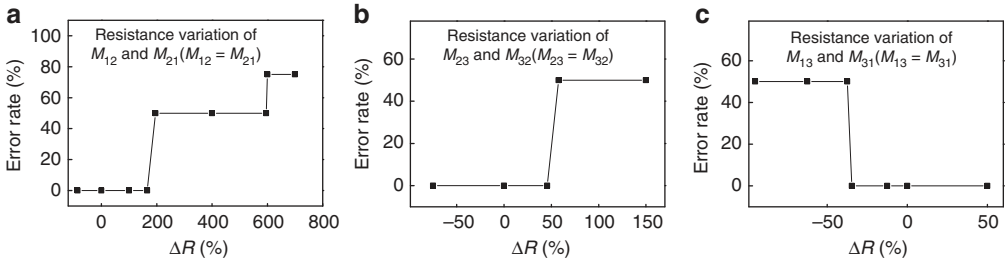

Figure 9 | Resistance variation tolerance of multi-associative memory. Effect of resistance variation of the following elements in equation (7) on multi-associative memories: (**a**) M_{12} and M_{21}; (**b**) M_{23} and M_{32}; and (**c**) M_{13} and M_{31}.

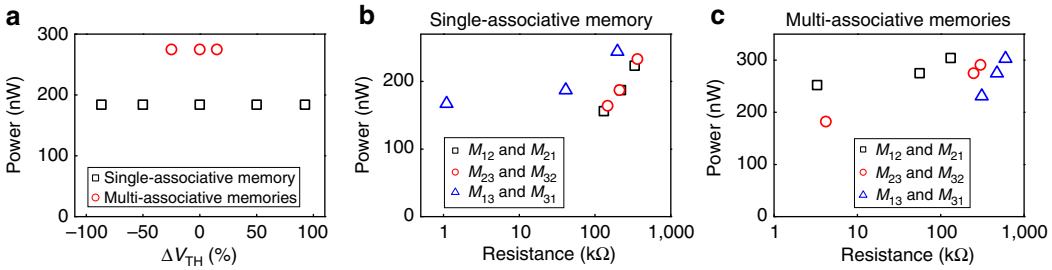

Figure 10 | Power consumption. Effect of threshold voltage variation on power consumption of the core memristor array for single-associative memory and multi-associative memories (**a**). Effect of resistance variation on power consumption of the core memristor array for single-associative memory (**b**) and multi-associative memories (**c**).

the effect of resistance variation on the power consumption of the core memristor array for single- and multi-associative memories, respectively. With the increase of resistance of the memristors, the power consumption also increases. This is due to the fact that the current in the memristor array does not change much, and thus the power of the core memristor array is determined by the resistance of the array (the power is approximately proportional to the resistance). On the other hand, as the resistances of M_{11}, M_{22} and M_{33} are much smaller than that of other elements in the matrix, the influence of resistance variation of the three elements on power consumption is insignificant.

Simulation of a MHN consisting of 6561 synapses. A larger-scale MHN consisting of 6,561 synapses (that is, an 81 × 81 matrix) has been designed with Cadence based on a standard 0.18-μm complementary metal–oxide–semiconductor process. The simulation results indicate that the network has a good tolerance towards the variation of the weight elements, as well as the variation of the threshold. The details are described in Supplementary Note 2 and Supplementary Figs 7–12.

Discussion

In conclusion, a 3-bit MHN has been constructed. The synaptic weights of the MHN are programmable and can be conveniently programmed to positive or negative by adjusting the conductance of the memristors. Single- and multi-associative memories have been realized with the MHN. The study paves the way for the hardware implementation of artificial neuromorphic networks to emulate memorization via associative states.

Methods

Device fabrication and characterization. The memristor used in the MHN is based on a metal–insulator–metal structure with a thin HfO_2 layer as the insulator. The metal–insulator–metal structure was fabricated onto a SiO_2 film, which had been thermally grown on a p-type silicon wafer. An ~70-nm Ni layer was deposited on the SiO_2 film using electron beam evaporation to form the bottom electrode. An HfO_2 thin film of ~80 nm thickness was deposited onto the Ni layer by Radio Frequency (13.6 MHz) magnetron sputtering of an HfO_2 target (>99.99% in purity) with the Ar flow rate of 75 sccm at the Radio Frequency power of 200 W. A 200-nm Au/10-nm Ni layer was finally deposited onto the HfO_2 film by electron beam evaporation to form the top electrode with the diameter ranging from 10 to 100 μm. The final thin film structure of the device was formed as Au/Ni/HfO_2/Ni/SiO_2 as shown in Fig. 1a. Dies with different electrode areas were cut from the wafer and packaged in standard 28-pin dual in-line package for constructing the MHN. Scanning electron microscopic image of the cross-section

of the memristor was carried out with JSM-7500 F scanning electron microscope (JEOL). Electrical characteristics of the memristor were measured with a Keithley 4200 semiconductor characterization system at room temperature.

MHN fabrication and measurement. The MHN was fabricated on a printed circuit board (PCB) and was connected to the HfO_2 memristors with wires (Supplementary Fig. 6). The MHN consists of six memristors, four transmission gate chips (Texas Instruments CD4066), seven operational amplifiers (Texas Instruments LM324N),and one comparator chip (Texas Instruments LM339) (see Supplementary Fig. 6). The complete circuit schematic of the MHN is shown in Supplementary Fig. 2. In the measurement of the MHN, a field programming gate array (model no. ALTERA EP2C8Q208C8) was used to generate the clock signals, and the waveforms of the clock signals and outputs were recorded with a RIGOL oscilloscope (model no. DS4024). The waveforms of control signals are presented in Supplementary Fig. 4.

References

1. Mead, C. Neuromorphic electronic systems. *Proc. IEEE* **78,** 1629–1636 (1990).
2. Von-Neumann, J. The principles of large-scale computing machines. *IEEE Ann. Hist. Comput.* **10,** 243–256 (1988).
3. Backus, J. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *Commun. ACM* **21,** 613–641 (1978).
4. Furber, S. & Temple, S. Neural systems engineering. *J. R. Soc. Interface* **4,** 193–206 (2007).
5. Ahmad, Z., Mat Noor, R. A. & Zhang, J. Multiple neural networks modeling techniques in process control: a review. *Asia-Pac. J. Chem. Eng.* **4,** 403–419 (2009).
6. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci.* **79,** 2554–2558 (1982).
7. Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl Acad. Sci.* **81,** 3088–3092 (1984).
8. Hopfield, J. J. & Tank, D. W. "Neural" computation of decisions in optimization problems. *Biol. Cybern.* **52,** 141–152 (1985).
9. Hopfield, J. J. & Tank, D. W. Computing with neural circuits: a model. *Science* **233,** 625–633 (1986).
10. Verleysen, M., Sirletti, B., Vandemeulebroecke, A. & Jespers, P. G. A high-storage capacity content-addressable memory and its learning algorithm. *IEEE Trans. Circuits Syst.* **36,** 762–766 (1989).
11. Du, K.-L. & Swamy, M. (eds) in *Neural Networks and Statistical Learning* Ch. 6, 159–186 (Springer, 2014).
12. Chua, L. Memristor-the missing circuit element. *IEEE Trans. Circuit Theory* **18,** 507–519 (1971).
13. Chua, L. O. & Kang, S.-M. Memristive devices and systems. *Proc. IEEE* **64,** 209–223 (1976).
14. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *Nature* **453,** 80–83 (2008).
15. Jo, S. H. *et al.* Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* **10,** 1297–1301 (2010).
16. Li, Y. *et al.* Ultrafast synaptic events in a chalcogenide memristor. *Sci. Rep.* **3,** 1619 (2013).
17. Wang, Z. Q. *et al.* Synaptic learning and memory functions achieved using oxygen ion migration/diffusion in an amorphous InGaZnO memristor. *Adv. Funct. Mater.* **22,** 2759–2765 (2012).
18. Kuzum, D., Jeyasingh, R. G. D., Lee, B. & Wong, H. S. P. Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Lett.* **12,** 2179–2186 (2012).
19. Kuzum, D., Yu, S. & Wong, H. S. P. Synaptic electronics: materials, devices and applications. *Nanotechnology* **24,** 382001 (2013).
20. Hu, S. G. *et al.* Emulating the paired-pulse facilitation of a biological synapse with a NiO_x-based memristor. *Appl. Phys. Lett.* **102,** 183510 (2013).
21. Krzysteczko, P., Münchenberger, J., Schäfers, M., Reiss, G. & Thomas, A. The memristive magnetic tunnel junction as a nanoscopic synapse-neuron system. *Adv. Mater.* **24,** 762–766 (2012).
22. Chang, T. *et al.* Synaptic behaviors and modeling of a metal oxide memristive device. *Appl. Phys. A* **102,** 857–863 (2011).
23. Seo, K. *et al.* Analog memory and spike-timing-dependent plasticity characteristics of a nanoscale titanium oxide bilayer resistive switching device. *Nanotechnology* **22,** 254023 (2011).
24. Yu, S. M., Wu, Y., Jeyasingh, R., Kuzum, D. G. & Wong, H. S. P. An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation. *IEEE Trans. Electron Devices* **58,** 2729–2737 (2011).
25. Choi, S. J. *et al.* Synaptic behaviors of a single metal–oxide–metal resistive device. *Appl. Phys. A* **102,** 1019–1025 (2011).
26. Ziegler, M. *et al.* An electronic version of Pavlov's dog. *Adv. Funct. Mater.* **22,** 2744–2749 (2012).
27. Hu, S. G. *et al.* Design of an electronic synapse with spike time dependent plasticity based on resistive memory device. *J. Appl. Phys.* **113,** 114502–114504 (2013).
28. Pershin, Y. V. & Di Ventra, M. Experimental demonstration of associative memory with memristive neural networks. *Neural Networks* **23,** 881–886 (2010).
29. Alibart, F., Zamanidoost, E. & Strukov, D. B. Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training. *Nat. Commun.* **4,** 2072 (2013).
30. Park, S. *et al.*in *Electron Devices Meeting (IEDM), 2013 IEEE Int.* 25.6.1–25.6.4 (Washington, DC, USA, 2013).
31. Eryilmaz, S. B. *et al.* Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array. *Front. Neurosci.* **8,** 205 (2014).
32. Burr, G. *et al.*in *Electron Devices Meeting (IEDM), 2014 IEEE Int.* 29.5.1–29.5.4 (San Francisco, CA, USA, 2014).
33. Calka, P. *et al.* Chemical and structural properties of conducting nanofilaments in TiN/HfO_2-based resistive switching structures. *Nanotechnology* **24,** 085706 (2013).
34. Zhao, L. *et al.* Multi-level control of conductive nano-filament evolution in HfO_2 ReRAM by pulse-train operations. *Nanoscale* **6,** 5698–5702 (2014).
35. Yang, J. J. S., Strukov, D. B. & Stewart, D. R. Memristive devices for computing. *Nat. Nanotechnol.* **8,** 13–24 (2013).
36. Basheer, I. A. & Hajmeer, M. Artificial neural networks: fundamentals, computing, design, and application. *J. Microbiol. Methods* **43,** 3–31 (2000).
37. Rojas, R. *Neural Networks: A Systematic Introduction* (Springer, 1996).

Acknowledgements

Y.L. acknowledges the support by NSFC under project no. 61274086 and the Young Scholar Fund of Sichuan under project no. 2011JQ0002. T.P.C. acknowledges the support by MOE Tier 1 Grant (grant no. RG 43/12) and NTU-A*STAR Silicon Technologies Centre of Excellence (grant no. M4070176.040).

Author contributions

The experiment was conceived by S.G.H., Y.L., T.P.C., Q.Y. and L.J.D.; it was carried out by S.G.H., Z.L., J.J.W., L.J.D., Y.Y. and S.H.; data analysis was conducted by S.G.H., Q.Y., J.J.W. and Y.L.; and S.G.H., Y.L., T.P.C. and Q.Y. prepared the manuscript with contributions from all authors.

Additional information

Supplementary Information accompanies this paper at http://www.nature.com/naturecommunications

Competing financial interests: The authors declare no competing financial interests.

Reprints and permission information is available online at http://npg.nature.com/reprintsandpermissions/

How to cite this article: Hu, S. G. *et al.* Associative memory realized by a reconfigurable memristive Hopfield neural network. *Nat. Commun.* 6:7522 doi: 10.1038/ncomms8522 (2015).