Unsupervised learning in probabilistic neural networks with multi-state metal-oxide memristive synapses

Alexander Serb *[1], Johannes Bill[3,4], Ali Khiat[1], Radu Berdan[2], Robert Legenstein[3], and Themis Prodromakis[1]

[1]Electronics and Computer Science dept., University of Southampton, Southampton, SO17 1BJ, United Kingdom
[2]Dept. of Electrical and Electronic Engineering, Imperial College, London, SW7 2AZ, United Kingdom
[3]Institute for Theoretical Computer Science, Graz University of Technology, Graz, 8010 Graz, Austria
[4]Kirchhoff Institute for Physics, University of Heidelberg, Heidelberg, 69120 Heidelberg, Germany

In an increasingly data-rich world the need for developing computing systems that can not only process, but ideally also interpret big data is becoming continuously more pressing. Brain-inspired concepts have shown great promise towards addressing this need. Here, we demonstrate unsupervised learning in a probabilistic neural network that utilises metal-oxide memristive devices as multi-state synapses. Our approach can be exploited for processing unlabelled data and can adapt to time-varying clusters that underlie incoming data by supporting the capability of reversible unsupervised learning. The potential of this work is showcased through the demonstration of successful learning in the presence of corrupted input data and probabilistic neurons, thus paving the way towards robust big-data processors.

Plastic synaptic connections are a key computational element of both the brain and brain-inspired neuromorphic systems. Outnumbering neurons by approximately 1000 to 1 in the human brain [1], synapses have to perform their main function, namely interconnecting neural cells via an often modifiable coupling strength (a weight), within extremely tight volume and power budgets. The desire to build and operate large neural networks with vast amounts of synapses has rendered the task of creating similarly efficient and yet practically implementable artificial synapses a high priority.

---

[*] Correspondence and requests for materials should be addressed to A.S. (email: `A.Serb@soton.ac.uk`)

A major route towards that goal has been the development of hardware synapse analogues, which has traditionally relied on commercially available Complementary Metal-Oxide Semiconductor (CMOS) technologies [2, 3, 4, 3, 5]. However, the visionary ideas of the early days of the field of memristor research [6, 7] have led to a different approach: the exploitation of the intrinsic electrical properties of a large and diverse group of emerging nanoelectronic devices exhibiting the phenomenon of resistive switching, nowadays also referred to as memristive devices [8, 9, 10]. The scalability [11], thresholded input voltage time-integration [12], multi-level storage [13], simple two-terminal structure, potential for low power operation [14] and back-end-of-line integration [15] features demonstrated thus far in various memristive device technologies attracted study in the field of memristive synapses.

So far, the potential of memristors to act as ersatz synapses has been studied through simulation [16, 17, 18, 19, 20, 21] and the demonstration of in-silico learning rule implementation, most notably -but not exclusively- that of Spike Timing-Dependent Plasticity (STDP) [22] generated by appropriate electrical memristor biasing schemes [23, 24, 25, 26, 27]. Other advances include the emulation of basic heterosynaptic plasticity in multi-terminal memristive devices [28] as well as the demonstration of STDP by exploiting the internal dynamics of memristors, albeit in volatile devices (that is devices that do not retain their memory state for long periods of time for example one day) [29, 30] and efforts towards the integration of memristors with neuromorphic circuits [31]. More recently the first examples of practical, small-scale artificial neural network operating with memristive synapses have been demonstrated, all using deterministic, supervised learning techniques. These include reference [32], where learning was implemented using a variant of the perceptron learning rule (the Manhattan update rule) and reference [33] where phase-change memory (PCM) rather than metal-oxide technology-based memristors are used to demonstrate learning in a Hopfield network using Hebbian learning. Finally, the first large-scale neural network using PCM technology was demonstrated by IBM [34] where a modified back-propagation rule was used in a three-layer artificial neural network.

In this work we exploit the gradual, multi-level switching characteristics of metal-oxide-based memristors for demonstrating unsupervised learning in a probabilistic neural network. Our work consolidates the current state-of-art in single-component synapse emulators (for example [24, 25, 26, 30, 21]) and advances the field of operating memristors as hardware synapse emulators in practical neural networks (for example [27, 32]). Particularly we demonstrate in a neural network using memristor synapses: First, pattern classification in a probabilistic neural network. Second, unsupervised learning achieved through the implementation of a winner-take-all (WTA) network. Third, reversible learning, an often neglected but essential aspect of truly flexible and useful learning systems. Fourth, the exploitation of the intrinsic properties of our memristors to successfully allow the neural network to encode conditional probabilities without any special input signal waveform engineering.

RESULTS

Weight-dependent STDP in TiO_2-based memristors

STDP is one of the most widely studied plasticity rules for spiking neural networks. In its pure form it relies on the premise that the relative timing between pre- and post-synaptic spike events is the major determinant of both the direction (potentiation/depression) and the magnitude of synaptic weight changes. Recently the hardware-friendly, pulse-based biasing scheme shown in Fig. 1(a) has been proposed as a possible method for implementing STDP in memristor-based synapses [19, 20, 35]. The memristor's resistive state (conductance) is interpreted as the equivalent of a synaptic efficacy (weight). To implement plasticity events, the scheme exploits the inherent capability of some memristive devices to act as thresholded voltage time-integrators, that is to change their resistive state as a function of input voltage, so long as its magnitude exceeds a certain threshold (the switching threshold). When the pre-synaptic neuron spikes, a prolonged low-voltage pulse is applied across the memristor. This pulse is by itself unable to induce any resistive switching (Fig. 1(a)). Spiking of the post-synaptic neuron, on the other hand, leads to the application of a brief, biphasic, bipolar pulse (Fig. 1(b)) that causes the memristor to undergo Long-Term Depression (LTD). Concurrent pre- and post-synaptic terminal spiking causes the memristor to sense the superposition of the pre- and post-synaptic spike waveforms and thereby undergo Long-Term Potentiation (LTP) (Fig. 1(a)).

We fabricated TiO_2-based devices (see Methods) and studied their behaviour during exposure to trains of STDP events. Each device under test (DUT) was exposed to four blocks of events, each consisting of 2400 individual events: LTD-inducing post-only events, LTP-inducing combined pre- and post-events, LTD events again and finally plasticity-neutral pre-events only. Fig. 1(b) shows typical measured results from our prototype DUTs for all mentioned electrical biasing schemes. First, we observe that the STDP rules are followed throughout the entire test, including the plasticity-neutrality of pre-only events (confirmed by experiments where pre-only events were applied at the high conductance boundary of the DUT's operating range - supplementary figure S4). Next, we observe the marked dependence of changes in resistive state on the running resistive state (DUT conductance g) for both LTP and LTD (Fig. 1(c)). Such dependence of conductance changes on the actual memristive state has commonly been observed in memristors including both metal-oxide [23] and phase-change [36] implementations. In supervised learning rules, such as the perceptron rule, this property is undesirable as updates independent of memristive state are required [32]. Here we particularly leverage this property to enable for the first time unsupervised learning in a practical network, in a manner similar to the work presented previously in ref. [37] that is based on simulations of phase-change memory models.

The experimental results in Fig. 1(b) suggest that the STDP rule being implemented can be described for each plasticity event by:

$$\Delta g = \mathrm{POST} \cdot \left(f^{+}(g) \cdot \mathrm{PRE} - f^{-}(g) \right) \qquad (1)$$

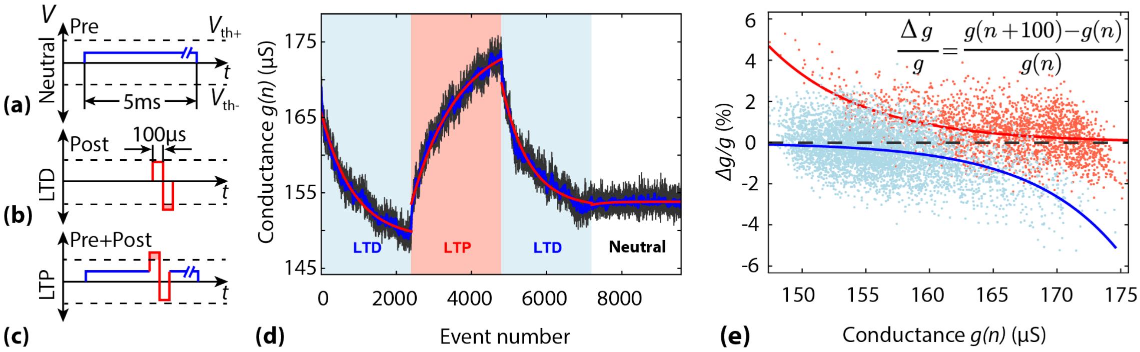


Figure 1: **Weight-dependent STDP in memristors.** (a, b, c): Memristor electrical biasing scheme used to test STDP. $V_{\text{th+}}, V_{\text{th-}}$: Memristor switching thresholds. Data for individual device thresholds in Supplementary Table T1. Voltage levels used to induce LTP and LTD in Supplementary Table T4. Red shading: Supra-threshold portions of the input affecting the memristor resistive state. (d): Typical experimental results from TiO_2 device. Black trace: raw data, blue trace: 10-point moving average, red trace: exponential fitting. Red shading: LTP. Blue shading: LTD. No shading: neutral region, no plasticity triggered. (e): Experimental data and exponential fittings describing STDP magnitude (relative change in device conductance g) as a function of initial memristor conductance. Red line: LTP fitting. Blue line: LTD fitting. Black dashed line: zero conductance change level. Same data as in (d).

where PRE and POST are binary values indicating whether a pre- or post-spike has occurred in the given event respectively, whilst $f^{+}(g)$ and $f^{-}(g)$ are functions that capture the influence of DUT conductance on LTP and LTD strength.

Normalising to obtain relative changes in g and rearranging we get:

$$\frac{\Delta g}{g} = \text{POST} \cdot \left[\text{PRE} \cdot f^{\text{LTP}}(g) - (1 - \text{PRE}) \cdot f^{\text{LTD}}(g)\right] \tag{2}$$

where $f^{\text{LTD}}(g) = \frac{f^{-}(g)}{g}$ and $f^{\text{LTP}}(g) = \frac{f^{+}(g) - f^{-}(g)}{g}$; both fitted by exponentials in Fig. 1(c).

Plotting $\Delta g/g$ *versus* g for both LTP and LTD reveals that our solid-state synapse features inherently self-stabilising plasticity (Fig. 1(c)): at higher conductance levels, further increases in conductance (LTP) become progressively smaller. Similarly, at the bottom end of the conductance scale LTD induction becomes increasingly ineffective. The gradual and monotonic dependence of weight changes on the running value of weight is an essential feature for memory models of unsupervised learning. If a stochastic data stream that triggers LTP and LTD with probabilities p and $(1 - p)$ respectively is fed into the DUT, we can expect its conductance to converge towards a unique equilibrium point. In other words, the memristive synapse should be able to encode and store in its resistive state the conditional probability $p(\text{PRE}|\text{POST} = 1)$ that a given postsynaptic spike is preceded by a presynaptic spike at the synapse within a short time interval. For

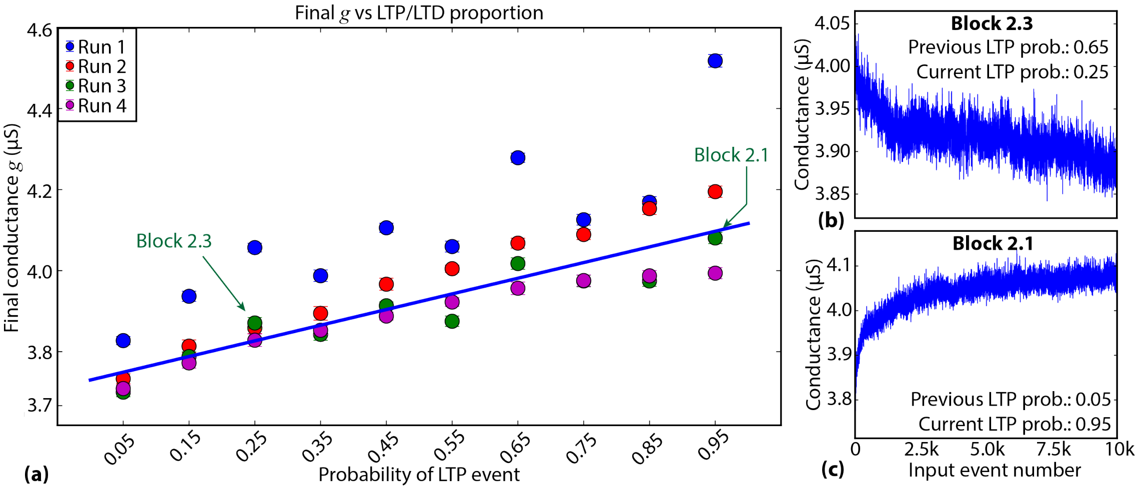


Figure 2: **TiO_2-based memristors encode conditional probabilities.** (a): Final memristor conductance after application of 10^4 input event blocks featuring different LTP/LTD compositions. Blue line corresponds to linear fit for runs 2-4. Error bars: standard deviation, number of samples (individual resistive state readings) per data point $n = 25$. Typical traces showing resistive state migration during two typical blocks: (b) one where the device is overall depressed (3rd block in run 2: Block 2.3) and (c) where it is potentiated (1st block in run 2: Block 2.1).

instance, consider a memrisitve synapse that is exposed to STDP events that consist of a mixture of 90% LTP events and 10% LTD events. We can expect the DUT conductance to eventually stabilise close to the upper boundary of the DUT's resistive state operating range.

Memristor synapses can encode conditional probabilities

We experimentally tested the theoretical prediction that conditional probabilities can be encoded and stored in the resistive state of a memristor. We performed four measurement runs on the same test device. Each run consisted of 10 blocks of plasticity events (10^4 events per block, that is 10^5 events per run, blue dots in Fig. 2). Individual plasticity events were randomly chosen to be LTP events with probability p_{LTP} and LTD events with probability $1 - p_{\mathrm{LTP}}$, where the probability of an LTP event was fixed within each block. In runs 2 and 4, p_{LTP} was 95%, 85%, ..., 5% for blocks 1 to 10 respectively, that is, the probability of LTP events was decreased after each event block. In runs 1 and 3, the same LTP probabilities were tested, but in random order (see Supplementary Table T3). At the end of each block the final resistive state of the memristor was measured (average of 25 read-outs after the end of each block).

The results of the experiment are shown in Fig. 2. After a burn-in phase, during which the memristor gradually reaches its normal operating range observed during the first run (10^5 events) we obtained consistent convergence points for the remaining three runs ($3 \cdot 10^5$ events) and a clear mapping between LTP/LTD composition and convergence conductance emerges: converged conductance data from runs 2-4 (that is excluding burn-in) is first pooled (convergence points at each LTP/LTD composition are averaged) and then fitted to a linear function (equation and fitting parameters in supporting material on page 38) of converged conductance *versus* LTP/LTD composition by least squares regression. The root-mean-square error (RMSE) of this fitting is approximately $5.25 \cdot 10^{-2}$µS. Moreover, we notice that the runs where the order of the LTP/LTD composition points was scrambled (1 and 3) show less well-behaved convergence points. Attempting to extrapolate memristor behaviour by exponential fitting, as presented in Supplementary Figure S14 indicates that even 10^4 events seem insufficient to achieve convergence given the choice of biasing parameters. We believe that this could be potentially addressed as more realistic memristor models appear. Thus we can conclude that TiO_2 memristor-based synapses appear to be able to practically support the encoding of conditional probabilities $p(\mathrm{PRE}|\mathrm{POST} = 1)$ in their resistive states.

Probabilistic neural networks with memristor synapses

The ability of individual memristors to encode conditional probabilities can be leveraged for the implementation of self-adapting spiking neural networks. In particular, WTA networks [38] have repeatedly been proposed for hardware implementations [39, 40, 41, 42], motivated in part by the fact that WTA structures play an important role in cortical information processing [43]. Recent rigorous analyses revealed that WTA networks consisting of stochastic spiking neurons subject to weight-dependent STDP are capable of performing probabilistic inference that essentially carries out clustering of input patterns. While a number of different types of WTA networks have been considered [44, 45, 46, 47, 35], optimal parameter adaptation is in any case accomplished by weight-dependent STDP rules of the form $\Delta w \propto \mathrm{POST} \cdot (\mathrm{PRE} - f(w))$, that is, by rules similar to the memristor-implemented plasticity rule from eq. (1).

To test whether memristor-based synapses can perform adequately as components of WTA networks, we implemented a WTA network that consisted of two stochastic spiking neurons with four inputs each. All four input synapses to one WTA neuron were implemented by TiO_2-based devices, while the synapses to the other neuron were implemented in software (see Fig. 3(a)). This hybrid network allowed us to directly compare software-simulated synaptic connections with memristive synapses in the same setup and with exactly the same inputs. It also allowed us to directly manipulate the software synapses and study the influence on memristive plasticity.

The 2-neuron probabilistic WTA network was implemented on an in-house developed instrumentation board for memristor device characterisation [48]. The two artificial neurons, WTA lateral inhibition and synapses feeding one of the neurons were all implemented in software on the board's microcontroller unit. During each experiment run 1200 4-bit patterns were presented to the network at the inputs $\mathbf{y} = (y_0, y_1, y_2, y_3)$. Determining the values of $\mathbf{y}$ begins by randomly and equiprobably drawing a pattern to be presented from a set of prototype test patterns (in our case 0110 and 1001). Next, each bit in the selected pattern is flipped with a probability of 10 % so that the network is presented with noisy instantiations of the prototype patterns. The resulting generated input vector is then multiplied by the weight vectors of both neurons and translated into membrane potential values, one for each neuron, as per eq. (3):

$$U_i(\mathbf{y}, t) = \theta_i(t) + \mathbf{w}_i(t) \cdot \mathbf{y}(t) \tag{3}$$

where $U_i(\mathbf{y}, t)$ denotes the membrane potential for neuron i during event t, $\theta_i(t)$ an adaptive excitability term that homoeostatically regulates neuron activity and $\mathbf{w}_i$ the weight vector from inputs $\mathbf{y}$ to neuron i. The symbol $\cdot$ represents the dot product operator. Importantly, whilst U_i represents the membrane potential of neuron i for the purposes of driving its firing behaviour, it does not directly translate to a physical voltage value to be applied to all synapse terminals (pre or post) it is connected to. Neuron firing events are instead translated into appropriate pre- or post-type voltage waveforms that are used to bias the affected memristor synapses. The homoeostatic term $\theta_i(t)$ has been used before for memristor learning [19] and has been theoretically justified in [45] for unsupervised learning in probabilistic WTA networks. By reducing the propensity to fire for neurons that show high average response, homoeostasis ensures that both neurons participate in the WTA competition over the long run (details in methods section).

The probability $p_i(\mathbf{y}, t)$ with which neuron i wins the WTA competition and therefore spikes at event t is given by:

$$p_i(\mathbf{y}, t) = \frac{e^{U_i(\mathbf{y},t)}}{\sum\limits_j e^{U_j(\mathbf{y},t)}} \tag{4}$$

Using computed p_i values for each pattern at each time step we can define a specialisation metric S that directly quantifies how attuned each neuron is to the two prototype input patterns:

$$S_i(t) = p_i(1001, t) - p_i(0110, t) \tag{5}$$

where $S_i(t)$ is the specialisation of neuron i at time t and takes values between 1 (perfectly specialised on 1001) and -1 (perfect specialisation on 0110).

By definition, at every event exactly one of the neurons wins and fires, thus triggering plasticity at its synapses. In the case of software synapses, weights are updated through a simple STDP rule that aims to approximately mirror memristor plasticity. The variability in resulting STDP-driven weight changes Δw and measurement noise observed in memristor synapses have both been included in the software synapse plasticity mechanism (see Methods). In the case of the hardware synapses the STDP conditions that determine whether LTP or LTD is required are the same as for their software counterparts, but the LTP and LTD events are translated into pulse voltage stimulation and therefore the magnitude of weight change is inherently set by each memristor. For the purposes of this experiment and since the non-invasiveness of the pre-only event has already been confirmed (Fig. 1), the pulsing scheme for LTP and LTD is reduced to only the above-threshold portions of the original waveforms, that is both LTP and LTD are represented by simple square-waves of appropriate amplitude. In order to map device resistive states onto weights all memristive synapses were first subjected to the protocol described in Fig. 1. Estimated maximum and minimum operational conductance values (extracted from the constant term of exponential fittings to traces in Fig. 1(b) - also see Supporting Figure S1) were mapped linearly to a weight range of $[-2.2, +2.2]$. The conductance-weight mappings are summarised in Supplementary Table T4.

Results from a WTA network experiment (run no. 1) are shown in Fig. 3. Both hardware and software synaptic weights w_{ij} were initialized close to 0 (see methods section) and subsequently the network was allowed to react to the incoming patterns freely. According to theoretical WTA models, unsupervised synaptic adaptations through STDP should lead to a clustering of inputs such that each neuron is preferentially activated by one of the prototype patterns and noisy variations of it. Fig. 3 demonstrates this behaviour in our setup with memristive synapses. The specialisation evolution in Fig. 3(b) shows how after a brief initial phase of uncertainty where the neurons are approximately equally attuned to both patterns and none can claim dominance over either pattern (approx. first 20-30 samples), the hardware synapse neuron develops a clear preference for pattern 0110 (specialisation S approaches -1). Similarly we can use the weights of software and hardware synapses at each trial to plot computed membrane potentials for each neuron in response to each pattern. This is shown in Fig. 3(c) where we observe how at the beginning of the run neither neuron has any intrinsic preference for any pattern (that is independent of the neuron-neuron interaction through the WTA); this only starts developing afterwards. The robustness of these experiments was confirmed by repeating the experiment three times in total. Results from all three runs are summarised in Supplementary Figure S8 and related text.

Examining the evolution of weight values throughout the run (Fig. 3 (d)) we observe that the hardware synapse weights experience noisy and slow drift from their initial values. In order to quantify this the evolution of each weight over trials was fitted to an exponential function and the standard deviation of the residual was then computed. This yielded estimates of both the noise levels and the overall weight change for each synapse over the trial (for full results see supporting material page 39 and Supplementary Figure S11). The software synapses concurrently experience similarly imperfect drift towards their final state. For comparison, see Supplemetary Figure S13 and S12 in the case where software synapses are noise-free. These results are confirmed by Fig. 3 (e,f) where we see a substantially clearer classification of pattern 0110 and related patterns different from 0110 in only one position (0110_δ) on the one hand (purple shading) and 1001 with 1001_δ (patterns different from 1001 in only one position) on the other hand (green shading) towards the end of the experiment *versus* the beginning. Specifically, at the beginning of the run patterns 1001 and 1001_δ cause the neuron that ultimately assigns itself to them (software-synapse) to fire only approximately 56% of the time whilst similarly the hardware-synapse neuron responds to its corresponding patterns (0110 and 0110_δ) approximately 77% of the time. In contrast, at the end of the run classification accuracy increases to 100% for both neurons. Thus the WTA network successfully segregates the prototype patterns despite the presence of noise. This result was achieved in a fully unsupervised manner.

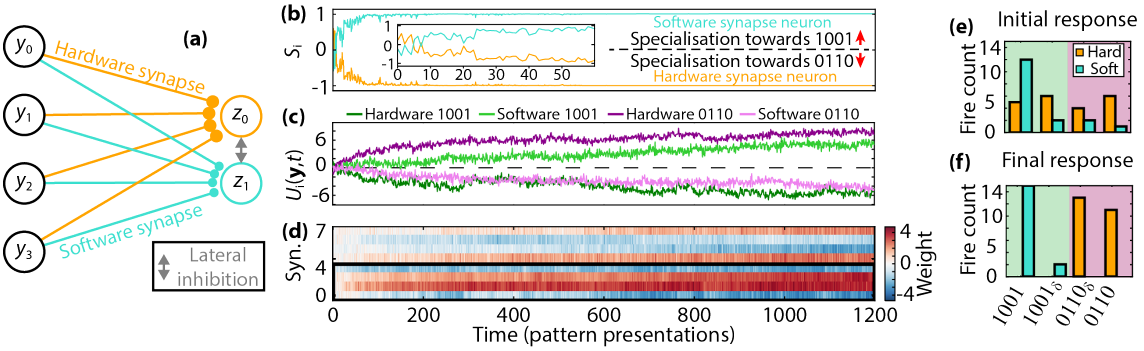


Figure 3: **Learning in a WTA network with a mixture of software and memristor synapses.** (a): Diagram of the 2-neuron, WTA network used in this work. (b): Evolution of neuron specialisations S_i to patterns 0110 and 1001 as weights change over successive events, illustrating the interplay between the two neurons. Inset: close-up of first 60 trials. (c): Computed membrane potentials of each neuron to both prototype patterns according to their weights at every trial illustrating the intrinsic pattern preferences of each neuron, that is independent of their interaction in the WTA network. (d): Evolution of hardware (synapses 0-3, enclosed in thick, black frame) and software (synapses 4-7) weights. (e,f): Responses of the WTA network to the initial (e) and final (f) 41 input samples. The fire count of both the hardware-synapse neuron (orange) and the software-synapse neuron (turquoise) is shown for patterns 0110, 1001 and patterns that differ from these prototypes in one position ($0110_\delta, 1001_\delta$). The different pattern groups are perfectly segregated by the end of the run.

Finally, in order to demonstrate that the WTA network is capable of not only learning a pattern, but also if demanded forgetting and relearning it, a further set of experiments was conducted. This consisted of two further, consecutive WTA learning runs (runs no. 2 and 3) immediately following the main run from Fig. 3 (by the end of which we recall the memristor synapses had specialised their neuron to pattern 0110). At the beginning of each of these additional runs the software synapses were initialised such that the network specialisation acquired during the immediately preceding learning run was reversed (hardware synapses were left unchanged). Under these circumstances the memristor-based synapses are expected to respond by flipping their intrinsic preference to the opposite pattern. Results are shown in Fig. 4.

In the case of the first additional run, the software synapses were initialised in such way as to instantly reverse the preferred pattern-to-neuron mapping outcome of the previous learning session and start the learning run with the software-synapse, rather than the memristor-synapse neuron more responsive to pattern 0110. Such initialisation should induce the memristor synapses to attempt specialising on pattern 1001 instead. The top half of Fig. 4 shows that this is indeed the case: at the end of the run the hardware-synapse neuron has lost its intrinsic preference to pattern 0110 and began switching to 1001 as evidenced by the membrane potential plot (Fig. 4(b)), which allowed the software neuron to consolidate its dominance of 0110 (Fig. 4(a)). Simultaneously, the software synapse weights remain relatively static around their extreme values, as initialised. The second additional run similarly initialises the software synapses appropriately in order to guide the memristor synapses to re-specialise on pattern 0110. This successfully occurs as evidenced by Fig. 4(f-j). In both cases, the fire count histograms (Fig. 4(d,e,i,j)) show how the initial classification preferences of each neuron become entrenched during each run as a result of the combined changes in both software and hardware synapse weights with hardware synapses mainly driving the process (Fig. 4(b,g) and Supplementary Table T5).

DISCUSSION

In this work we demonstrated that metal-oxide-based synapses with inherent, gradual, self-limiting switching properties are capable of learning and re-learning of input patterns in an unsupervised manner within a probabilistic WTA network. Key to the learning process is the memristors' capability of encoding conditional probabilities of the expected input signal within their resistive states. As a notable consequence of the probabilistic learning scheme, ubiquitous (and unavoidable) noisy changes in the resistive states are continuously counterbalanced by the ongoing alignment of present weights with future presented inputs.

This study was performed on TiO_2-based devices which has historically been one of the significant metal-oxide systems employed in memristive devices [49]. In previous work, we have identified that these devices support multi-level switching [50], the emulation of short and long term plasticity [30, 21], and bidirectionally gradual switching [51], which we can reliably detect using our tailor-made instruments [48] even at low OFF/ON resistive

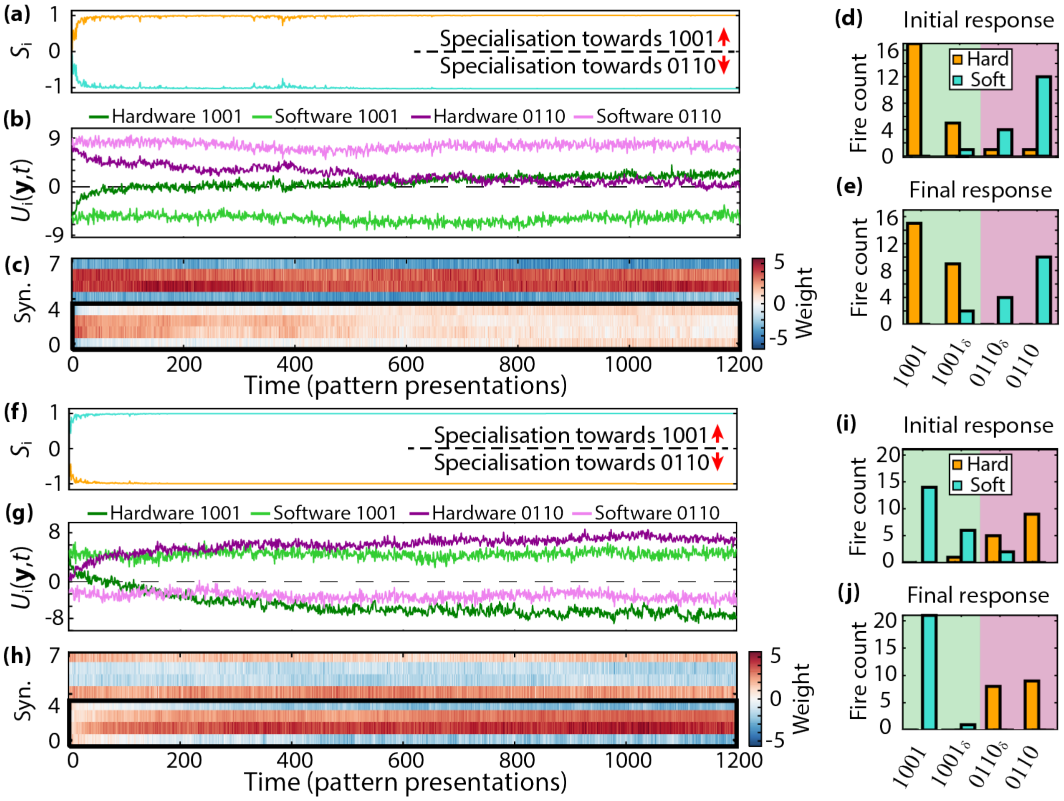


Figure 4: **Reversible learning is supported in WTA networks employing TiO_2 memristor-based synapses.** (a-e): First run attempting to unteach the pattern recognition abilities gained in Fig. 3. (a): Evolution of neuron specialisations S_i to patterns 0110 and 1001 as weights change over successive events, illustrating the interplay between the two neurons. (b): Computed membrane potentials of each neuron to both prototype patterns according to their weights at every trial illustrating the intrinsic pattern preferences of each neuron, that is independent of their interaction in the WTA network. (c): Evolution of hardware (synapses 0-3, enclosed in thick, black frame) and software (synapses 4-7) weights. (d,e): Responses of the WTA network to the initial (d) and final (e) 41 input samples. The fire count of both the hardware-synapse neuron (orange) and the software-synapse neuron (turquoise) is shown for patterns 0110, 1001 and patterns that differ from these prototypes in one position ($0110_\delta, 1001_\delta$). (f-j): Corresponding data as in (a-e) for second run attempting to reteach the memristor synapses to prefer pattern 0110. The abrupt changes between final and initial responses over consecutive experiments mainly arise from the different initialisations of the software synapses in each case.

state ratios. Endurance and retention data on our devices is shown in the Supplementary Figures S5 and S6. Here, we build on our previous results for demonstrating a memristor-based, system-level application. The presented concept may extend to other memristor technologies based on different metal-oxides such as HfO_2 and Ta_2O_5 that have shown great promise towards memory applications.

For the purposes of this work, our prototypes were operated under low voltage conditions, that is close to their threshold voltages (see Supplementary Tables T1, T4). Importantly, the devices' threshold voltages are not rigidly fixed, but rather depend on stimulus waveform shape as well as the initial memory state of the devices. For example, the threshold voltage dependence on square-wave pulse duration is shown in the Supplementary Figure S3. As a result, the voltage amplitude of the pre-waveform, as shown in Fig. 1(a), is important as it determines the voltage contrast between: First, the super-threshold peak in the pre+post waveform and the sub-threshold peak in the post-only waveform and second the pre-only waveform and the post-only peak. Larger contrasts mean that spurious drift effects induced by threshold voltage variability can be mitigated more effectively. This reduces the risks arising from unwanted plasticity caused by repeated pre- pulsing without any post-response as well as unwanted, concurrent DUT resistive state disturbance by both the peak and the trough of the post- waveform.

Considering future implementations of practical memristor-based systems we note the following: First, the downscaling of the memristor component itself as a memory storage element is already comparing favourably to mainstream technologies (for example SRAM), as memristors in the $10 \times 10 = 100\,\text{nm}^2$ range have already been demonstrated [11]. SRAM scaling is projected to become difficult at below $50000\,\text{nm}^2$ even under favourable process variability conditions [52] (but note 1T-SRAM technology [53]). Even though the performance of memristor devices is also known to be impacted by downscaling, through for example increased access wire resistance, the advantage over SRAM is expected to dominate. Furthermore, we note that memristors can pack more than 1 bit/storage element in a non-volatile manner whilst SRAM is purely digital and volatile. Second, at the array level, the packing density of memristors can be in -principle increased by the development of high density 3D crossbar arrays [54], where back-end integrable selector elements [55] could mitigate the well-known sneak path problem [56]. Third, at the peripheral circuit level the trade-off between memristor functionality and circuit complexity needs to be studied more in-depth. Standard square pulse generators (for write) and sense amplifiers (for read), also employed in conventional memory systems, might suffice if memristors are to be treated as binary data storage elements. More complicated circuits capable of generating multiple voltage levels (write) and reading absolute resistance values will be, however, needed for multi-state operation; a compromise between higher bit resolution operation required silicon real estate for peripheral circuits. Finally, the challenges of interfacing with analogue hardware-based artificial neurons have to be considered. Optimised operation will be achieved if: a) the artificial neurons output spikes of the forms exhibited in Fig. 1(a) and b) all the voltages involved are within the headroom required by the artificial neuron circuitry. If the former condition is not met, then each neuron will need to be equipped with a suitable output waveform-shaping circuit at the moderate cost of 1/neuron. This can be expected to be a relatively minor inconvenience if the waveforms involved are simply variable duration square waves; easily obtainable via digital clock signals. If the latter condition is not met, then additional supply rails will have to be introduced on-chip and the output waveform-shaping circuits will require level shifters of voltage difference-related levels of complexity; yet the cost will remain at the 1/neuron level. Notably, in this work biasing conditions were individually tailored for each memristive synapse, a result of device-to-device variability which is expected to become increasingly challenging with downscaling. Improvements in control over fabrication and electroforming conditions are needed to counterbalance that effect and deliver memristors that operate under sufficiently uniform biasing conditions in order to use a single, non-programmable waveform shaping circuit for all devices in practical systems.

The WTA architecture used in this study can be seen as a simplified version of cortical layers 2/3 where parvalbumin-positive interneurons provide feedback inhibition to pyramidal cells (see for example [57, 58, 44, 45, 46] for similar models). Recent experimental data on the connectivity dynamics in cortical circuits suggest that synaptic modifications in the cortex are stochastic (for example [59, 60, 61]). This is of particular relevance to our study as our results demonstrate WTA architectures are particularly robust against the noisy synaptic plasticity exhibited by our memristive prototypes, also noted through simulations in [19]. In addition, the theoretical framework introduced in [62, 63] indicates that stochastic plasticity may even have advantageous computational properties, in that it performs Bayesian inference on optimal circuit parameters; suggesting that the inherent stochastic properties of memristors could even be beneficial to learning.

In our experiments, the prototype patterns 1001 and 0110 were presented as noisy versions where each component was independently inverted with a probability of 10%. Hence, the presented patterns for prototype 1001 included patterns 0001, 1000, 1101, 1011. These patterns were denoted by 1001_δ (analogous noisy versions of 0110 were denoted by 0110_δ). In particular the noisy versions 1101 and 1011 show significant overlap with the other prototype 0110 since they include one of their two non-zero bits. Our results (see for example Fig. 3(e),(f)) show that the system is very robust to such pattern overlap since those neurons that specialized on the prototype also responded to the corresponding δ patterns after learning. For the current set-up, we did not use pattern overlap in the prototype patterns because of their very low dimensionality. The theory for WTA networks and experience from computer simulations (see for example [44]) show that such overlap poses no difficulties for the circuit for high-dimensional inputs. Hence we do not expect any additional hardware cost to account for pattern overlap due to the inherent robustness of WTA circuits to such pattern sets.

In the WTA experiments, the Hebbian-type synaptic plasticity rule was complemented with a homoeostatic plasticity rule which regulates the intrinsic excitability of the neurons. Notably, homoeostatic intrinsic plasticity only adds a bias to the neuronal membrane potential and, thus, does not affect a neuron's relative firing preference to different input patterns. It also influences the emerging synaptic weight configuration only indirectly by ensuring that all WTA neurons maintain a long-term average firing rate and thereby modulates the succession of LTP / LTD plasticity signals which the memristor synapses observe. While homoeostatic intrinsic plasticity has been proven mathematically to harden robustness of unsupervised learning in stochastic WTA circuits, its implementation in neuromorphic designs is possible for example via a local accumulator circuit per neuron. Notably, homoeostatic contributions to the overall membrane potential during learning (Fig. 3) were significantly smaller than synaptic contributions as depicted in Supporting Figure S7.

In conclusion, in this work we have demonstrated for the first time that individual, solid-state memristors can emulate complex, weight-dependent plasticity, including unsupervised classification, forgetting and relearning, within an experimental WTA network setting. This paves the way towards real-time on-node processing of big, unstructured data; an enabling technology for addressing the challenges arising from the volume of data generated by the internet-of-things revolution.

METHODS

Device fabrication and preparation: For all experiments, TiO_2-based micrometer-scale devices are used employing a metal-insulator-metal (MIM) structure. The process flow started by thermally oxidising a 6-inch Silicon wafer in order to create a layer that serves as an insulator medium. Then, three major steps were realised to obtain the bottom electrode (BE), active layer and top electrode (TE) consecutively. Each step consisted of optical lithography, material deposition and liftoff process. 10 nm Platinum layers were deposited for TE and BE by electron beam evaporation, whilst 25 nm TiO_2 was deposited by reactive magnetron sputtering. These fabrication steps resulted in a MIM stack of $\mathrm{Pt}(10\,\mathrm{nm})/\mathrm{TiO_2}(25\,\mathrm{nm})/\mathrm{Pt}(10\,\mathrm{nm})$; devices used with slight variations for many other purposes in our group (see [27, 51]). Prior to use, all devices were electroformed using positive polarity (top electrode at higher potential than bottom electrode) pulsed voltage ramps. A series resistor was used as a current-limiting mechanism in all cases. Typical electroforming voltages were in the range of $7 - 8\,\mathrm{V}$.

WTA network set-up: In the WTA network, neurons fire with probability p_i as determined by the abstract membrane potentials $U_i(\mathbf{y}, t)$ according to eq. 3 and eq. 4. The network response in turn triggers plasticity of the hardware and software synapses, as well as of the excitabilities θ_i. For the WTA network set-up, we hence have to define three quantities: the plasticity rule of software synapses, a function that maps the memristor conductance values g to abstract weights w in eq. 3 (conductance to weight map function), and the plasticity rule of the excitability θ_i. The plasticity rule of hardware synapses is inherently controlled by the memristors.

For software synapses w_{ij} we fundamentally use a plasticity rule of the form:

$$\Delta w_{ij} = \eta \cdot \mathrm{POST} \cdot (\mathrm{PRE} - f(w_{ij})) \tag{6}$$

where the learning rate $\eta = 0.03$. The weight-dependent function $f(w_{ij})$ will be determined such that it approximately mirrors the plasticity of memristor synapses. The structure of $f(w_{ij})$ can be estimated from the measured memristor plasticity in Fig. 1(c). Using eq. 1 and eq. 2 we find:

$$\Delta g \propto \mathrm{POST}\cdot\left(\mathrm{PRE}-\frac{f^-(g)}{f^+(g)}\right)=\mathrm{POST}\cdot\left(\mathrm{PRE}-\frac{\frac{f^-(g)}{g}}{\frac{(f^+(g)-f^-(g))}{g}+\frac{f^-(g)}{g}}\right)= \tag{7}$$

$$=\mathrm{POST}\cdot\left(\mathrm{PRE}-\frac{f^{\mathrm{LTD}}(g)}{f^{\mathrm{LTP}}(g)+f^{\mathrm{LTD}}(g)}\right) \tag{8}$$

Supporting Figure S10 shows the fraction on the right hand side of eq. (8) based on the fitted functions $f^{\mathrm{LTP}}(g)$ and $f^{\mathrm{LTD}}(g)$ from Fig. 1(c). As can be seen in the figure, the measured plasticity curves of the memristor suggest a sigmoidal shape for the function $f(w_{ij})$ in eq. (6). This observation can be substantiated analytically: By inserting the exponential fits $f^{\mathrm{LTP}}(g)=\exp(-\frac{1}{2}\alpha_{\mathrm{P}}\cdot(g-\beta_{\mathrm{P}}))$ and $f^{\mathrm{LTD}}(g)=\exp(\frac{1}{2}\alpha_{\mathrm{D}}\cdot(g-\beta_{\mathrm{D}}))$ into eq. (8), a few lines of algebra yield:

$$\Delta g \propto \mathrm{POST}\cdot\left(\mathrm{PRE}-\sigma\left(\frac{\alpha_{\mathrm{P}}+\alpha_{\mathrm{D}}}{2}\cdot\left[g-\frac{\alpha_{\mathrm{P}}\beta_{\mathrm{P}}+\alpha_{\mathrm{D}}\beta_{\mathrm{D}}}{\alpha_{\mathrm{P}}+\alpha_{\mathrm{D}}}\right]\right)\right) \tag{9}$$

where we define $\sigma(x)=(1+\exp(-x))^{-1}$. Such sigmoidal shape was qualitatively observed for all memristor synapses (as seen in Supplementary Figure S1), which served as a reference for the shape of software plasticity. Based on the comparison of eq. (6) and eq. (9), we map memristor conductances g to abstract weights w via a linear function

$$w=\alpha\cdot(g-\beta) \tag{10}$$

and set $f(w_{ij})=\sigma(w_{ij})$ in eq. (6), thereby tackling the software synapse plasticity rule and the conductance to weight function.

Adding realistic imperfections in software synapse function: On top of this ideal, theoretical framework we have added two mechanisms of software synaptic weight corruption in order to better match the memristors' own noisy and variable behaviour. Under this more realistic framework we make a distinction between the true, underlying weight w_{ij} and the as measured weight including measurement noise v_{ij}. The first weight corruption mechanism reflects the memristors' cycle-to-cycle variation, which in our case manifests itself as variable conductance jumps given identical stimulus and initial conductance conditions. This is modelled by adding a switching variability term w_{var} drawn from a Gaussian distribution with $\sigma_{\mathrm{sw}}=0.04$ (units of abstract weight) limited to $\pm 5\,\sigma$. The weight update equation thus becomes:

$$\Delta w_{ij}=\eta\cdot\mathrm{POST}\cdot[(\mathrm{PRE}-f(w_{ij}))+w_{var}] \tag{11}$$

where σ_{sw} was chosen to qualitatively force the software synapses to show slightly worse cycle-to-cycle variation than what was being observed in the hardware. This is evidenced in the Supplementary Figure S11, where the evolution of individual synaptic weights during an ANN learning trial is plotted.

The second weight corruption mechanism introduces a degree of measurement noise in the software synapses, that is allows the system to use a slightly distorted weight value without causing any change in the underlying value of w_{ij}. As such, at every time step, the weight values used to compute neuron membrane potentials and by extension contribute to deciding which neuron fires to each presented input are calculated by the following formula:

$$v_{ij} = w_{ij} + w_{\mathrm{mn}} \tag{12}$$

where w_{mn} is an added measurement noise term drawn from a Gaussian distribution with $\sigma_{\mathrm{meas}} = 0.4$ (abstract weight), limited to $\pm 5\,\sigma$. σ_{meas} was determined by estimating/quantifying the measurement noise in our devices and adjusting the software so as to behave slightly more stochastically than the memristors (see Supplementary Table T6).

Homoeostatic plasticity: Furthermore, in order to facilitate robust learning we employ a homoeostatic plasticity mechanism for the excitabilities θ_i. At the beginning of each learning experiment (initial learning only, this does not apply to reversibility learning experiments where continuity of θ_i is maintained), the θ_i are initialized at 0. Then, prior to each time step t the excitability is updated according to:

$$\theta_i(t) = \begin{cases} \theta_i(t-1) - \eta_\theta/2 & \text{if neuron } i \text{ wins event } t-1 \\ \theta_i(t-1) + \eta_\theta/2 & \text{otherwise} \end{cases} \tag{13}$$

with learning rate $\eta_\theta = 0.03$. The homoeostatic plasticity rule (13) makes sure that both neurons will participate in the competition and fire, on average, equally often: If a neuron fires on average during one half of the time steps, the value of its θ_i will remain approximately stable. Otherwise, its θ_i will slowly increase (if the neuron fires rarely) or decrease (if the neuron fires frequently). The rule (13) defines the plasticity rule of the excitability. Notably this homoeostasis rule is very similar to the one used in [19] where although specific details are not given, the spiking frequency of all neurons is periodically assessed and an equivalent to the θ_i term is adjusted accordingly. In this work this procedure takes place at every trial which may allow finer and more responsive homoeostatic control. The behaviour of this plasticity rule is described in [45].

Memristor parameter extraction: For the WTA experiment, the parameters in eq. (10) must be individually determined for each memristor. To this end, the conductance operating range of each device was extracted in the set-up of Fig. 1(b) prior to the WTA experiment. The parameters α and β were then implicitly defined by directly mapping two conductance points g_{LOW} and g_{HIGH} to abstract weight values -2.2 and $+2.2$ respectively. The values for g_{LOW} and g_{HIGH} for each device are shown in Supplementary Table T4. The numerical values for all initial and final weights during the WTA experiments are provided in Supplementary Table T5 for both software and memristive synapses.

Network initialisation procedures: The experimental run corresponding to Fig. 3 (and similar, confirmation runs included in the supporting material) required all weights to be initialised as close to 0 as possible. For the hardware synapses this was done through the memristor handling instrument (Supplementary Figure S15) by manually applying a suitable number of square wave pulses on each device. We did not seek to automate this process at this stage. For the software synapses, the initial underlying weights w_{ij} were set to 0, but then corrupted by measurement noise before use as described above.

A summary of the key network operating parameters is given in Supplementary Table 1.

Table 1: Key ANN operating parameters.

Symbol	Value	Units	Parameter
η	0.03	-	Synaptic weight learning rate
η_θ	0.03	-	Homoeostatic plasticity learning rate
σ_{sw}	0.04	Abstr. weight	Switching noise (soft. syn.)
σ_{meas}	0.4	Abstr. weight	Measurement noise (soft. syn.)

Instrumentation: All experiments were carried out using an upgraded version of the in-house instrumentation described in [48]. More details provided in supporting material on page 32.

Bibliography

[1] Drachman, D. A. *Neurology* **64**(12), 2004–2005 (2005).

[2] Hasler, P., Diorio, C., Minch, B., Mead, C., et al. In *Circuits and Systems, 1995. ISCAS'95., 1995 IEEE International Symposium on*, volume 3, 1660–1663. IEEE, (1995).

[3] Schemmel, J., Grübl, A., Meier, K., and Mueller, E. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, 1–6. IEEE, (2006).

[4] Indiveri, G., Chicca, E., and Douglas, R. *Neural Networks, IEEE Transactions on* **17**(1), 211–221 (2006).

[5] Indiveri, G., Linares-Barranco, B., Hamilton, T. J., Van Schaik, A., Etienne-Cummings, R., Delbruck, T., Liu, S.-C., Dudek, P., Häfliger, P., Renaud, S., et al. *Frontiers in neuroscience* **5** (2011).

[6] Snider, G. S. *Nanotechnology* **18**(36), 365202 (2007).

[7] Snider, G. S. In *Nanoscale Architectures, 2008. NANOARCH 2008. IEEE International Symposium on*, 85–92. IEEE, (2008).

[8] Chua, L. O. *Circuit Theory, IEEE Transactions on* **18**(5), 507–519 (1971).

[9] Chua, L. O. and Kang, S. M. *Proceedings of the IEEE* **64**(2), 209–223 (1976).

[10] Waser, R., Dittmann, R., Staikov, G., and Szot, K. *Advanced Materials* (21), 2632–2663 (2009).

[11] Govoreanu, B., Kar, G., Chen, Y., Paraschiv, V., Kubicek, S., Fantini, A., Radu, I., Goux, L., Clima, S., Degraeve, R., et al. In *Electron Devices Meeting (IEDM), 2011 IEEE International*, 31–6. IEEE, (2011).

[12] Gupta, I., Serb, A., Khiat, A., Zeitler, R., Vassanelli, S., and Prodromakis, T. *arXiv preprint arXiv:1507.06832* (2015).

[13] Waser, R. and Aono, M. *Nature materials* **6**(11), 833–840 (2007).

[14] Schroeder, H., Zhirnov, V. V., Cavin, R. K., and Waser, R. *Journal of Applied Physics* **107**(5), 054517 (2010).

[15] Ma, X., Strukov, D. B., Lee, J. H., and Likharev, K. K. In *Nanotechnology, 2005. 5th IEEE Conference on*, 175–178. IEEE, (2005).

[16] Sheri, A. M., Hwang, H., Jeon, M., and Lee, B.-g. *Industrial Electronics, IEEE Transactions on* **61**(6), 2933–2941 (2014).

[17] Yu, S., Gao, B., Fang, Z., Yu, H., Kang, J., and Wong, H.-S. P. *Frontiers in neuroscience* **7** (2013).

[18] Li, Q., Serb, A., Prodromakis, T., and Xu, H. *PloS one* **10**(3), e0120506 (2015).

[19] Querlioz, D., Bichler, O., and Gamrat, C. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 1775–1781. IEEE, (2011).

[20] Serrano-Gotarredona, T., Masquelier, T., Prodromakis, T., Indiveri, G., and Linares-Barranco, B. *Frontiers in neuroscience* **7** (2013).

[21] Wei, S. L., Vasilaki, E., Khiat, A., Salaoru, I., Berdan, R., and Prodromakis, T. *arXiv* **abs/1509.01998** (2015).

[22] Bi, G.-Q. and Poo, M.-M. *The Journal of neuroscience* **18**(24), 10464–10472 (1998).

[23] Jo, S. H., Chang, T., Ebong, I., Bhadviya, B. B., Mazumder, P., and Lu, W. *Nano letters* **10**(4), 1297–1301 (2010).

[24] Mayr, C., Stärke, P., Partzsch, J., Cederstroem, L., Schüffny, R., Shuai, Y., Du, N., and Schmidt, H. In *Advances in Neural Information Processing Systems*, 1700–1708, (2012).

[25] Du, N., Kiani, M., Mayr, C. G., You, T., Bürger, D., Skorupa, I., Schmidt, O. G., and Schmidt, H. *Frontiers in neuroscience* **9** (2015).

[26] He, W., Huang, K., Ning, N., Ramanathan, K., Li, G., Jiang, Y., Sze, J., Shi, L., Zhao, R., and Pei, J. *Scientific reports* **4** (2014).

[27] Mostafa, H., Khiat, A., Serb, A., Mayr, C. G., Indiveri, G., and Prodromakis, T. *Frontiers in neuroscience* **9** (2015).

[28] Yang, Y., Chen, B., and Lu, W. D. *Advanced Materials* **27**(47), 7720–7727 (2015).

[29] Du, C., Ma, W., Chang, T., Sheridan, P., and Lu, W. D. *Advanced Functional Materials* **25**(27), 4290–4299 (2015).

[30] Berdan, R., Vasilaki, E., Khiat, A., Indiveri, G., Serb, A., and Prodromakis, T. *Scientific reports* **6** (2016).

[31] Indiveri, G., Legenstein, R., Deligeorgis, G., Prodromakis, T., et al. *Nanotechnology* **24**(38), 384010 (2013).

[32] Prezioso, M., Merrikh-Bayat, F., Hoskins, B., Adam, G., Likharev, K. K., and Strukov, D. B. *Nature* **521**(7550), 61–64 (2015).

[33] Eryilmaz, S. B., Kuzum, D., Jeyasingh, R., Kim, S., BrightSky, M., Lam, C., and Wong, H.-S. P. *arXiv preprint arXiv:1406.4951* (2014).

[34] Burr, G. W., Shelby, R. M., Sidler, S., Di Nolfo, C., Jang, J., Boybat, I., Shenoy, R. S., Narayanan, P., Virwani, K., Giacometti, E. U., et al. *Electron Devices, IEEE Transactions on* **62**(11), 3498–3507 (2015).

[35] Bill, J. and Legenstein, R. *Frontiers in neuroscience* **8** (2014).

[36] Suri, M., Bichler, O., Querlioz, D., Traoré, B., Cueto, O., Perniola, L., Sousa, V., Vuillaume, D., Gamrat, C., and DeSalvo, B. *Journal of Applied Physics* **112**(5), 054904 (2012).

[37] Querlioz, D., Bichler, O., Vincent, A. F., and Gamrat, C. *Proceedings of the IEEE* **103**(8), 1398–1416 (2015).

[38] Rumelhart, D. E. and Zipser, D. *Cognitive science* **9**(1), 75–112 (1985).

[39] Starzyk, J. and Fang, X. *Electronics Letters* **29**(10), 908–910 (1993).

[40] Choi, J. and Sheu, B. J. *Solid-State Circuits, IEEE Journal of* **28**(5), 576–584 (1993).

[41] Urahama, K. and Nagao, T. *Neural Networks, IEEE Transactions on* **6**(3), 776–778 (1995).

[42] Indiveri, G. *Analog Integrated Circuits and Signal Processing* **28**(3), 279–291 (2001).

[43] Douglas, R. J. and Martin, K. A. *Annu. Rev. Neurosci.* **27**, 419–451 (2004).

[44] Nessler, B., Pfeiffer, M., and Maass, W. In *Advances in neural information processing systems*, 1357–1365, (2009).

[45] Habenschuss, S., Bill, J., and Nessler, B. In *Advances in Neural Information Processing Systems*, 773–781, (2012).

[46] Kappel, D., Nessler, B., and Maass, W. *PLoS Comput. Biol* **10**, e1003511 (2014).

[47] Habenschuss, S., Puhr, H., and Maass, W. *Neural computation* **25**(6), 1371–1407 (2013).

[48] Berdan, R., Serb, A., Khiat, A., Regoutz, A., Papavassiliou, C., and Prodromakis, T. *Electron Devices, IEEE Transactions on* **62**(7), 2190–2196 (2015).

[49] Strukov, D. B., Snider, G. S., Stewart, D. R., and Williams, R. S. *nature* **453**(7191), 80–83 (2008).

[50] Berdan, R., Prodromakis, T., and Toumazou, C. *Electronics letters* **48**(18), 1105–1107 (2012).

[51] Serb, A., Khiat, A., and Prodromakis, T. *Electron Devices, IEEE Transactions on* **62**(11), 3685–3691 Nov (2015).

[52] Makosiej, A., Thomas, O., Amara, A., and Vladimirescu, A. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, 1460–1463. IEEE, (2013).

[53] Leung, W., Hsu, F.-C., and Jones, M.-E. In *ASIC/SOC Conference, 2000. Proceedings. 13th Annual IEEE International*, 32–36. IEEE, (2000).

[54] Kügeler, C., Meier, M., Rosezin, R., Gilles, S., and Waser, R. *Solid-State Electronics* **53**(12), 1287–1292 (2009).

[55] Burr, G. W., Virwani, K., Shenoy, R. S., Fraczak, G., Rettner, C. T., Padilla, A., King, R., Nguyen, K., Bowers, A., Jurich, M., et al. In *VLSI Technology (VLSIT), 2013 Symposium on*, T66–T67. IEEE, (2013).

[56] Linn, E., Rosezin, R., Kügeler, C., and Waser, R. *Nature materials* **9**(5), 403–406 (2010).

[57] Bill, J., Buesing, L., Habenschuss, S., Nessler, B., Maass, W., and Legenstein, R. *PloS one* **10**(8), e0134356 (2015).

[58] Masquelier, T., Guyonneau, R., and Thorpe, S. J. *PloS one* **3**(1), e1377 (2008).

[59] Trachtenberg, J. T., Chen, B. E., Knott, G. W., Feng, G., Sanes, J. R., Welker, E., and Svoboda, K. *Nature* **420**(6917), 788–794 (2002).

[60] Zuo, Y., Lin, A., Chang, P., and Gan, W.-B. *Neuron* **46**(2), 181–189 (2005).

[61] Holtmaat, A. J., Trachtenberg, J. T., Wilbrecht, L., Shepherd, G. M., Zhang, X., Knott, G. W., and Svoboda, K. *Neuron* **45**(2), 279–291 (2005).

[62] Kappel, D., Habenschuss, S., Legenstein, R., and Maass, W. *PLoS Comput Biol* **11**(11), e1004485 (2015).

[63] Kappel, D., Habenschuss, S., Legenstein, R., and Maass, W. In *Advances in Neural Information Processing Systems*, 370–378, (2015).

Acknowledgements

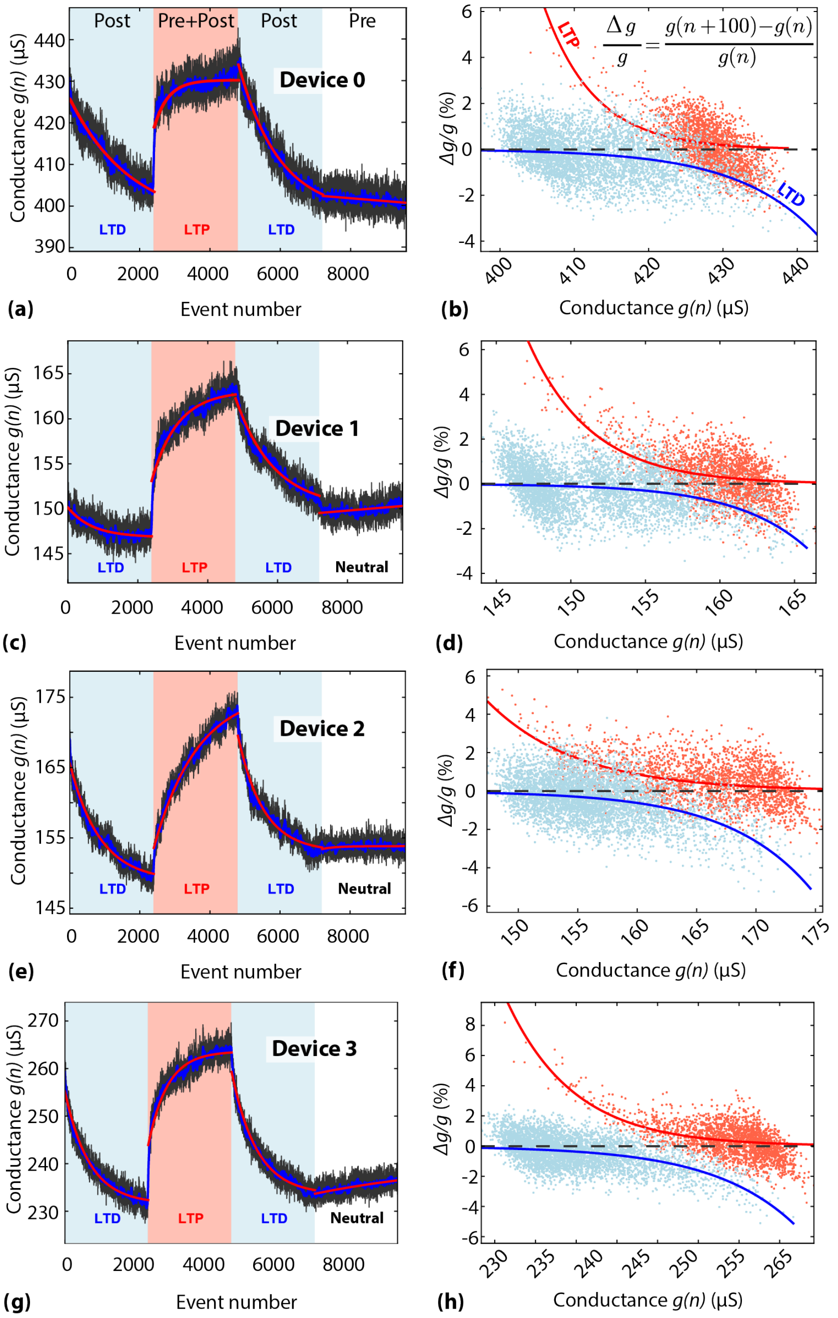
Author contributions. The first two authors contributed equally to this work by setting up and performing the experiments. AK fabricated the devices. RB assisted with the experiments and instrument preparation. TP, RL, JB, AS conceived ideas and have written the manuscript. TP and RL are joint last authors.

Data availability. All data supporting this study and its findings are available within the article, its Supplementary Information and associated files. Any source data deemed relevant is available from the corresponding author upon request.

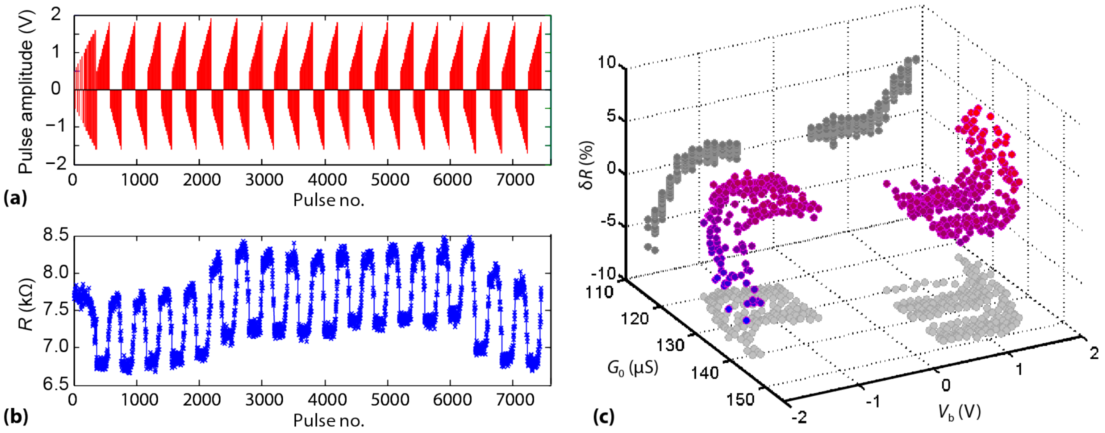
Conflict of interest statement. The authors declare no conflict of interest of any kind.

SUPPORTING MATERIAL

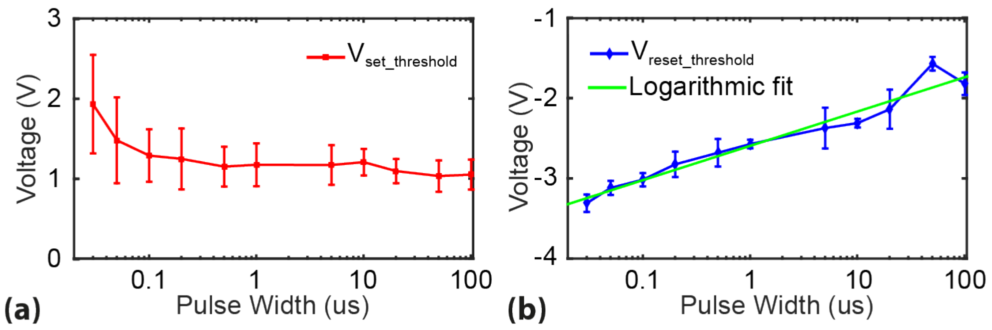
Supplementary Figures



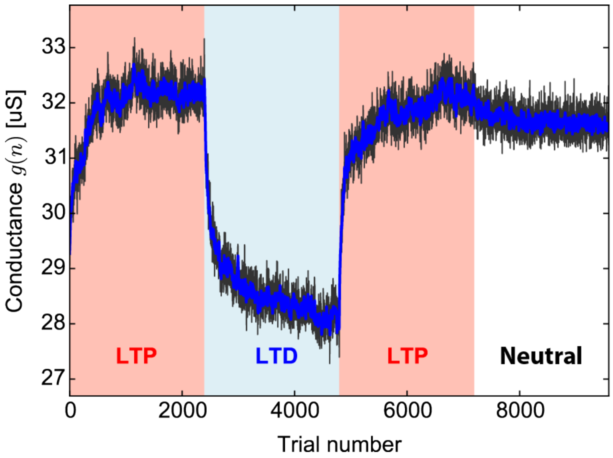
Supplementary Figure S1: **Typical memristor behaviours under experimental protocol used for Supplementary Figure 1.** Panels (a,c,e,g) and (b,d,f,h) as in Fig. 1(b,c) respectively for the four devices used as artificial synapses in this work.



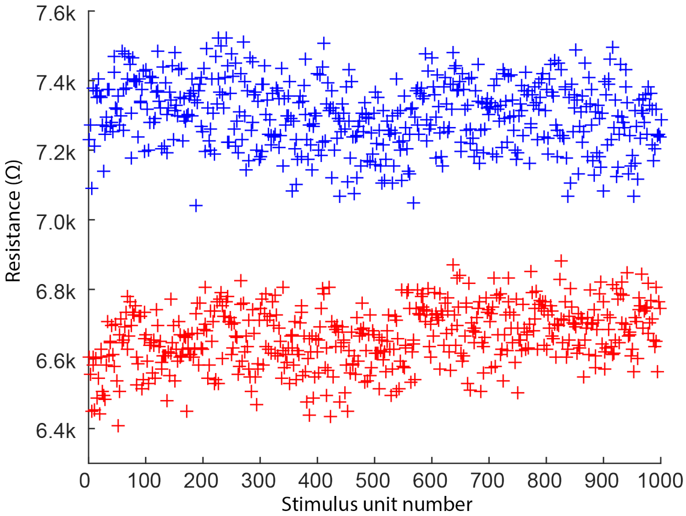
Supplementary Figure S2: **TiO_2 devices support gradual switching.** (a) Biasing parameter optimiser testing protocol pulsing sequence as applied to subject device. (b) Resistive state time evolution in reaction to the stimulation from (a). (c) Change in resistive state as a function of applied pulse voltage (pulse duration 100 μs) and starting resistive state. Note that overall resistive state change is restricted to approximately ±15% around a baseline value (in this case approximately 130 μS). Dark gray projection shows resistive state change *versus* pulse voltage independent of starting resistive state.



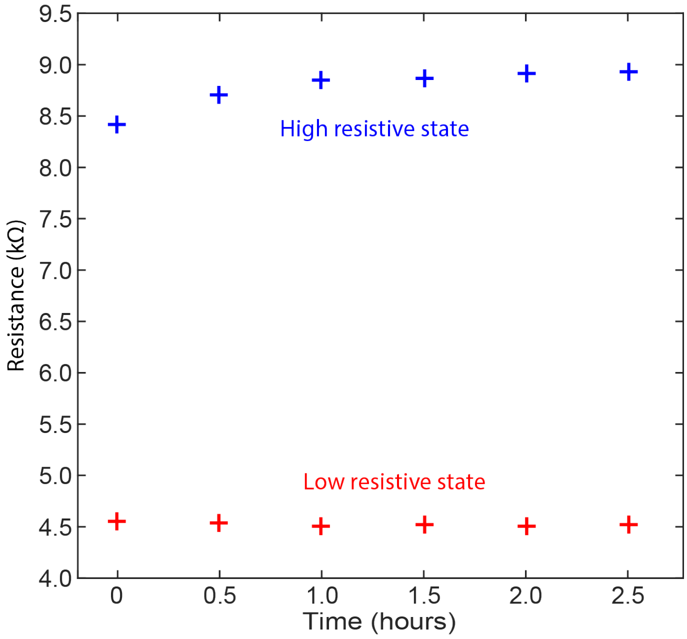
Supplementary Figure S3: **Voltage-time dilemma in TiO_2-based memristors.** Estimated threshold voltages for LTP-like SET resistive state transitions (a) and for LTD-like RESET transitions (b) are shown as a function of applied pulse durations. In the case of RESET transitions we see a good exponential fit, whilst in the case of SET transitions the relation grows faster-than-exponentially. Standard error bars shown, number of samples (individual resistive state readings) per data point $n = 17$. Data presented previously by our group in reference [1].



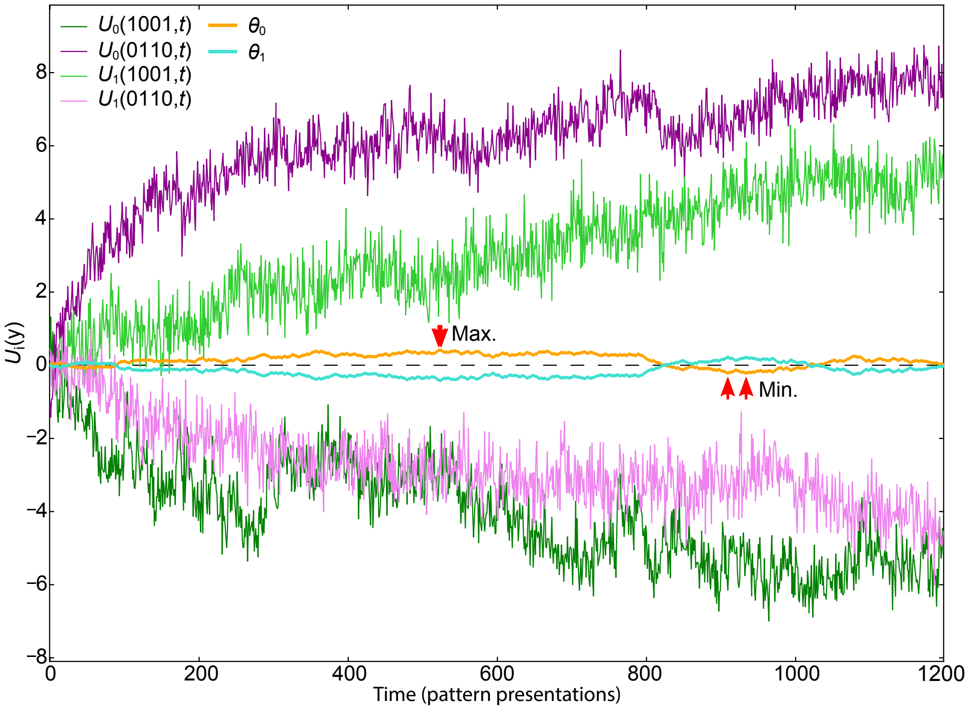
Supplementary Figure S4: **The read operation used in this work was minimally invasive.** In the white, neutral region of the plot each pulse represents 5 ms at 300 mV. The read-out operation lasts about 20 ms at 200 mV by comparison. The device remains firmly at high conductance during this final stage of pulsing. Thus, in combination with Fig 1(b) we confirm that the read-out operation used in this work most likely does not affect DUT resistive state regardless of the conductance value at the time of read-out.



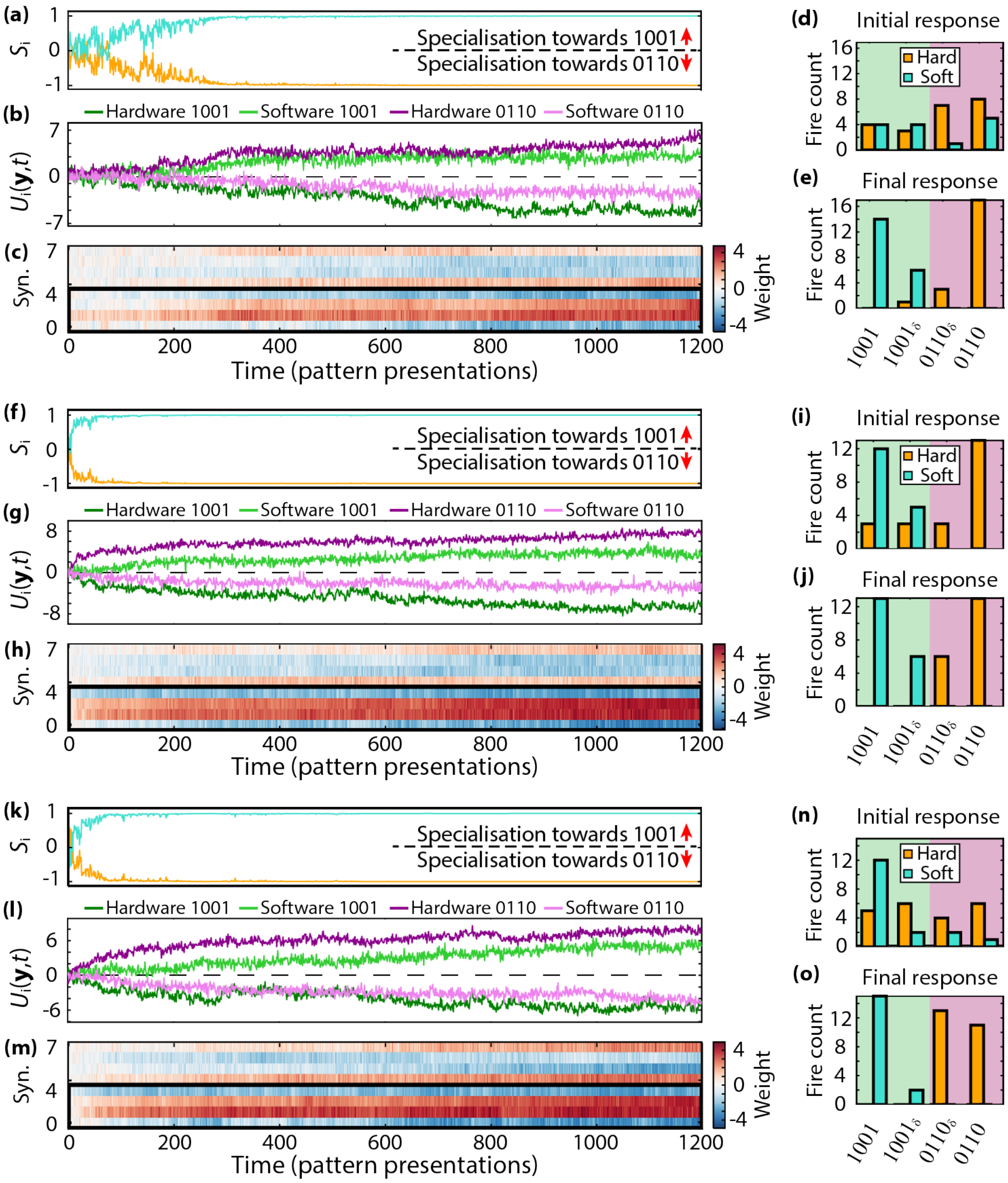
Supplementary Figure S5: **Endurance data from TiO_2-based device family used for this work.** 500 full cycles of stimulate-assess are shown in the figure.



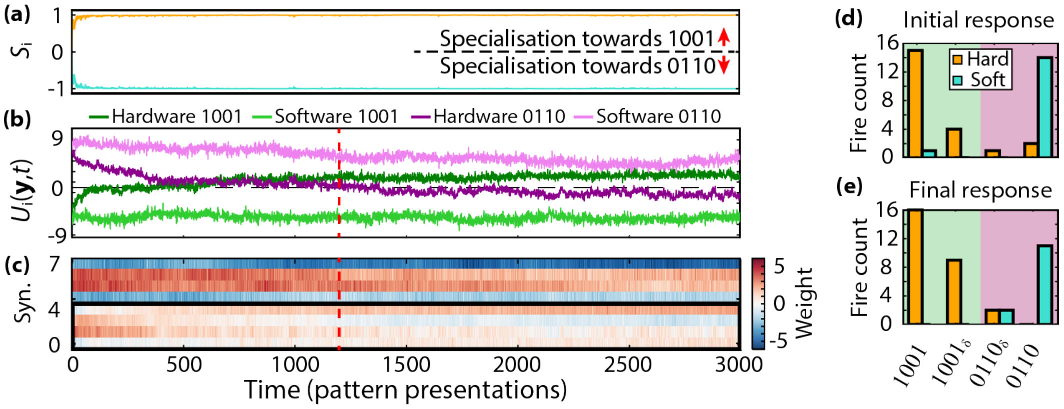
Supplementary Figure S6: **Retention data from TiO_2-based device family used for this work.** Both high and low resistive states were individually checked for drift.



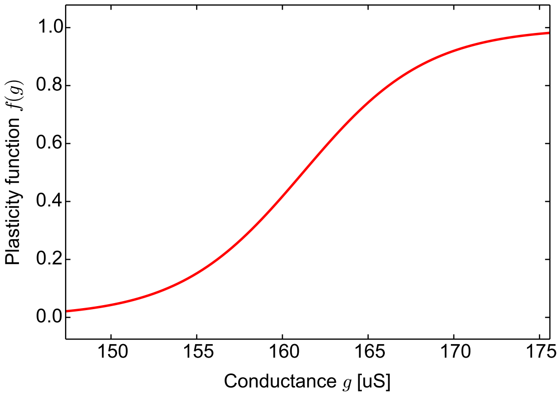
Supplementary Figure S7: **Membrane potentials and homoeostasis during learning.** Computed full membrane potentials U_i(pattern, time) for both hardware ($i = 0$) and software ($i = 1$) neurons to patterns 0110 (purple) and 1001 (green) including the influence of the homoeostatic term θ_i from eq. (3); same as Fig. 3(c). Additionally, the contribution of the homoeostatic plasticity to the membrane potential is also plotted alone in orange/cyan. Red arrows indicate the trials when the homoeostatic correction term reaches its maximum (+0.419) and minimum (-0.225) values.



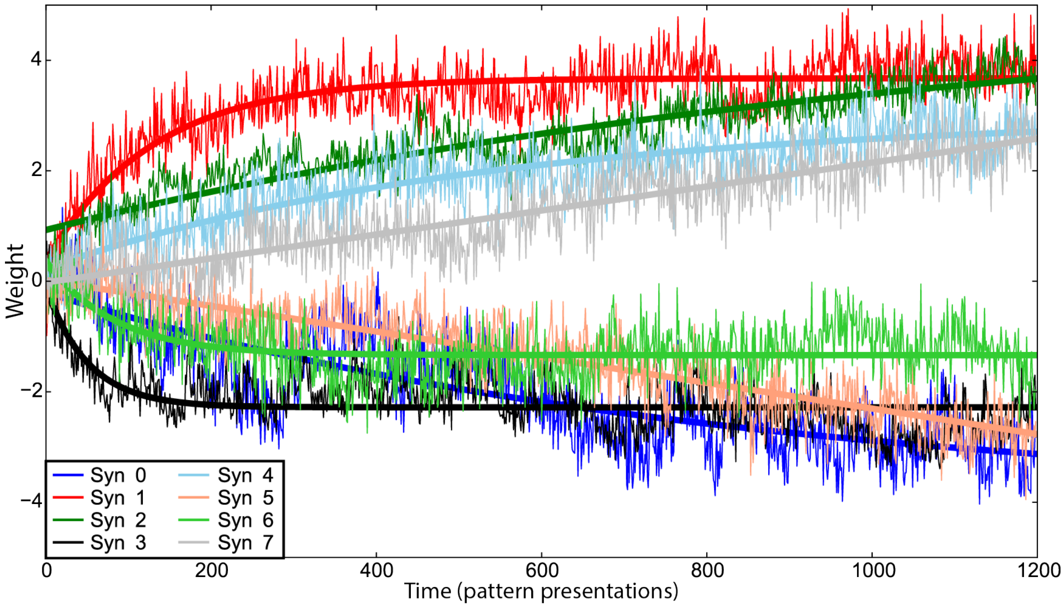
Supplementary Figure S8: **Three consecutive runs of the ANN unsupervised learning experiment.** ((a-e), (f-j), (k-o)): as in Fig. 4. Before the beginning of each run all synapses are initialised so that no neuron has any preference for either prototype pattern. At the end of each run the prototype patterns have been segregated successfully. Legend similar to Fig. 4. The last run (k-o) corresponds to figure 3.



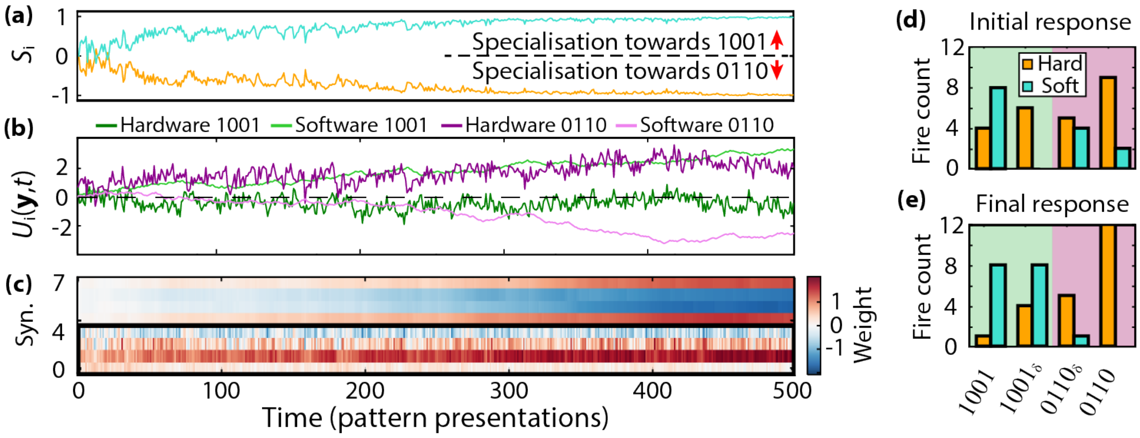
Supplementary Figure S9: **3000 trial learning reversal experiment immediately following the last trial from Fig. 4.** Legend similar to Fig. 4. The 1200 trial point is marked by the red, vertical, dashed lines in (b,c). The hardware synapses successfully switch their preference to pattern 1001 by the end of the run as evidenced by the computed membrane potentials shown in (b). At the 1200 trial mark this hasn't yet occurred very clearly.



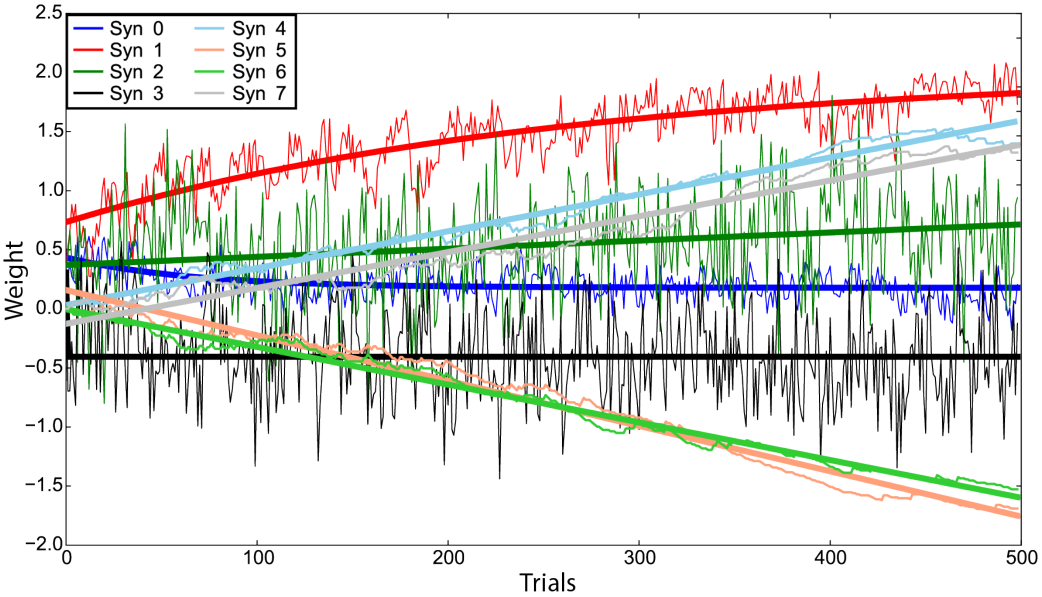
Supplementary Figure S10: **Estimate of software plasticity from the fitted exponentials f^{LTP} and f^{LTD}.** A sigmoidal shape emerges.



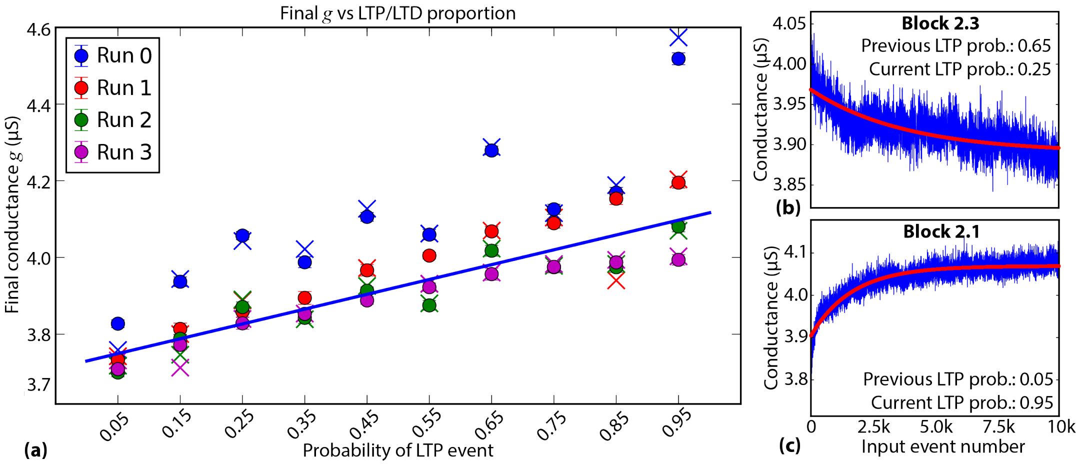
Supplementary Figure S11: **Synapse behaviour during learning.** Evolution of hardware (synapses 0-3) and software (syn. 4-7) weights over the WTA network run from Fig. 3 (thin traces) and corresponding exponential fits (thick traces).



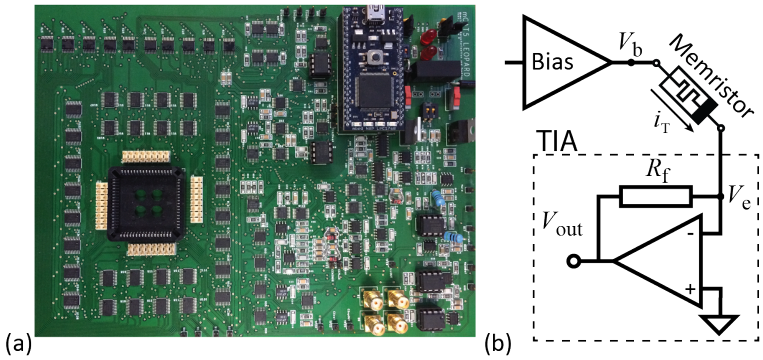
Supplementary Figure S12: **Example of WTA learning in a test run where the software synapses are not afflicted by noise.** Legend similar to Fig. 4.



Supplementary Figure S13: **Synapse behaviour during learning.** Evolution of hardware (synapses 0-3) and software (syn. 4-7) weights over the WTA network run from Supplementary Figure S12 (thin traces) and corresponding exponential fits (thick traces).



Supplementary Figure S14: **Memristors encode conditional probabilities.** Same as Fig. 2 but extrapolated resistive state convergence points are now also shown as cross marks in panel (a). Points with the same colour at each LTP probability point arise from the same data block. Error bars: standard deviation, number of samples (individual resistive state readings) per data point $n = 25$. In (b,c) exponential fits are added to both data blocks.



Supplementary Figure S15: **Memristor characterisation and handling instrumentation.** (a) Photograph of and (b) read-out scheme used by the instrumentation used to carry our all experiments in this work.

Supplementary tables

Supplementary Table T1: Voltage threshold levels extracted under 100 µs pulsed stimulation for the devices used in this work.

Device ID	$V_{\text{th, LTD}}$ (V)	$V_{\text{th,LTP}}$ (V)
0	-1.07	+0.98
1	-1.08	+0.91
2	+1.31	-1.19
3	-1.40	+1.19

Supplementary Table T2: Drift in memristor resistive state as a result of the application of only pre-type events. Case Read-check corresponds to Supplementary Figure S4; all other cases directly from Supplementary Figure S1. The resistive state range is directly computed from Supplementary Table T4 as the difference between the conductance levels corresponding to the weight values $[+2.2\,V, -2.2\,V]$ (final - initial). The read-check case has no defined operating range.

DUT ID	Resistive state drift (µS)	% of resistive state range (abs. val.)
0	-1.65	5.15
1	+0.79	6.05
2	+0.40	1.76
3	+2.83	9.42
Read-check	+0.31	-

Supplementary Table T3: Order of test blocks in each conditional probability encoding experiment run - see Fig. 2.

Run no.	LTP probability (%)									
	0.05	**0.15**	**0.25**	**0.35**	**0.45**	**0.55**	**0.65**	**0.75**	**0.85**	**0.95**
0	7th	5th	3rd	6th	4th	8th	2nd	10th	9th	1st
1	10th	9th	8th	7th	6th	5th	4th	3rd	2nd	1st
2	7th	5th	3rd	6th	4th	8th	2d	10th	9th	1st
3	10th	9th	8th	7th	6th	5th	4th	3rd	2nd	1st

Supplementary Table T4: Biasing parameters and conductance-to-weight mappings used for all WTA network runs.

Device ID	**LTD ampl. (V)**	**LTP ampl. (V)**	**Conduct. at weight = -2.2 (μS)**	**Conduct. at weight = +2.2 (μS)**
0	-0.90	+0.90	398	430
1	-0.90	+0.90	150	163
2	+1.20	-1.10	153	176
3	-1.30	+1.10	234	264

Supplementary Table T5: Initial and final weights for software and hardware synapses for WTA network runs 1 (Fig. 3) and 2 and 3 (Fig. 4).

			Synapse ID			
Run no.	**Synapse type**	**Initial/final**	1	2	3	4
1	Memristive	Initial	0.046	-0.423	-0.578	-0.414
		Final	-2.823	4.088	3.188	-2.269
	Software	Initial	0.064	0.228	-0.069	0.381
		Final	2.471	-2.542	-1.969	2.626
2	Memristive	Initial	-3.400	3.862	3.528	-2.362
		Final	1.909	-0.101	0.006	1.705
	Software	Initial	-2.734	4.465	3.286	-2.922
		Final	-3.322	3.614	2.982	-3.964
3	Memristive	Initial	1.614	0.263	0.348	1.549
		Final	-2.600	4.267	3.106	-2.675
	Software	Initial	2.179	-1.092	-0.697	2.859
		Final	2.926	-2.100	-2.167	1.886

Supplementary Table T6: Noise quantification for weight evolution during WTA network run from Fig. 3. σ_{residual}: Standard deviation of residuals of weight *versus* exponential fit from Supplementary Figure S11. Δw_{total}: Overall change in weight over the duration of the learning run as extracted from the polynomial fitting (final - initial). σ_{meas}: Uncertainty directly attributable to measurement error. $\sigma_{\text{unexplained}}$: Remaining uncertainty. All values in units of abstract weight.

Synapse ID	Δw_{total}	σ_{residual}	$\sigma_{\text{measerror}}$	$\sigma_{\text{unexplained}}$
0	-2.9205	0.6062	0.3673	0.4823
1	3.3125	0.4167	0.3539	0.2200
2	2.7363	0.3470	0.2002	0.2834
3	-2.1748	0.4364	0.2534	0.3553
4	2.4748	0.4236	-	-
5	-2.7820	0.4152	-	-
6	-1.7032	0.4580	-	-
7	2.5998	0.4329	-	-

Supplementary note 1

Device characterisation and behaviour. The capability of TiO_2-based memristors to encode conditional probabilities largely relies on their ability to support gradual switching. Supplementary Figure S2 shows the biasing parameter optimiser test routine [2] being applied on a single device. During this routine the device under test (DUT) is subjected to a series of pulse trains in alternating polarities. Each pulse train consists of a succession of progressively higher voltage pulses; all at fixed duration (Supplementary Figure S2(a)). The effect of each voltage amplitude used on DUT resistive state is assessed by measuring resistive state between pulses. The test shows how the choice of bias voltage determines the speed of switching (Supporting Figure S2(c)). We find that in our devices appropriate choice of pulsing voltage can lead to gradual switching corresponding to very small δR in response to input stimulation.

Applying successive barrages of identical, pulsed stimuli (LTP only or LTD only) as described in Fig. 1 confirms the capability of gradual switching and uncovers the dependence of the magnitude of switching on the value of the running conductance. Supplementary Figure S1 shows results from the experimental procedure carried out in Fig. 1 on all devices used for this work. We note that all devices are well-behaved, with LTP and LTD easily fitting to the exponential model used in Fig. 1.

Moreover, Supplementary Figure S1(b) shows a typical case of cycle-to-cycle variation in memristive devices. Final resistive state at the end of the second LTD event block is slightly different compared to the first LTD block. Whilst this may be at least partially explained by incomplete convergence to an equilibrium point, our experience with the TiO_2-based devices suggests that cycle-to-cycle variation is likely to play a role in this phenomenon.

Another important aspect of device behaviour is the voltage-time dilemma, that is the trade-off between pulse duration and voltage. We tested our samples with the biasing parameter optimiser routine at different pulse widths and recorded the pulse voltages at which the resistive state of the DUT had changed by 2% *versus* its state at the start of the test. The obtained values provide rough, but comparably obtained estimates of the DUT voltage threshold. Supplementary Figure S3 shows extracted threshold voltages from a typical device in the same family as used in this work *versus* pulse duration whilst Supplementary Table T1 summarises the 100 µs pulse thresholds extracted for the devices used in this work. The exponential relation between pulse duration and pulse voltage is encouraging towards the notion that switching can be achieved at significantly lower power cost if shorter, but stronger pulses are used as stimulation.

The thresholded nature of switching in our devices as shown in Supplementary Figure S2 provided good read-disturb immunity to our devices. Fig. 1(b) shows that the DUT read-out operation did not lead to appreciable changes in DUT resistive state when the DUT is at its minimum operational conductance. We ran experiments to confirm that this is still the case when the DUT is at its maximum operational conductance. Results are shown in Supplementary Figure S4, confirming the immunity of our devices to read-disturb at both extremes of their operating resistive state range. In addition, we quantified these results by fitting conductance evolution data from the neutral regions (pre-type stimulation only) of Supplementary Figures S1(a,c,e,g) and S4 to exponentials via least-squares optimisation and then computing the fitted change in conductance at the start *versus* the end of the region. The results, summarised in Supplementary Table T2, indicate that the effect is small (less than 10% of DUT resistive state range as defined in Supplementary Table T4).

Finally, basic endurance and retention data is shown in Supplementary Figures S5, S6. The endurance run was conducted by repeatedly applying stimulus units (trains of 10 identical pulses lasting 100 μs at +1 V or −1 V amplitude) of alternating polarities, each followed by resistive state assessments (1 assessment = average of 5 reads). Results indicate reliable and repeatable switching of our TiO_2 devices for 500 cycles (that is 1000 stimulus units) with a small but clear (approximately 3% of LRS resistive state level) window between HRS and LRS. The retention run was carried out by driving a test device at its operational resistive state ceiling, measuring resistive state for 2.5 hours in 30 minute intervals and then driving the device to its operating resistive state floor and taking another set of half-hourly resistive state measurements. We notice that the low resistive state is very stable (max. - min. value: approximately 44 Ω) whilst the high resistive state experiences a slight upward drift (max. - min.: approximately 505 Ω corresponding to approximately 13% of the resistive state operating range of approximately 4 kΩ).

Supplementary note 2

Experimental protocols. In the experiment testing for the capability of memristors to encode conditional probabilities, four test runs were carried out. Two of them used test blocks visiting the LTP probability points in scrambled order for the purposes of confirming that results obtained from the other two runs were not a consequence of visiting the various LTP probability points in a monotonically decreasing order. The precise sequence in which LTP probability points were visited are shown in Supplementary Table T3.

In all WTA network experiments both the biasing parameters used to implement plasticity and the mappings between memristor conductance and synaptic weight were kept constant. The numbers used are summarised in Supplementary Table T4. The initial and final software and hardware weights for each WTA network run are summarised in Supplementary Table T5.

The effects of homoeostasis can be observed by examining the computed membrane potential response of the hardware-synapse neuron for the two prototype patterns and noting how significant the effect of the homoeostatic term is. This is shown in Supplementary Figure S7 for the ANN run corresponding to Fig. 3, where the homoeostatic term fluctuated between +0.419 and −0.225 units of abstr. weight. However, the homoeostatic term can take much larger values, reaching a magnitude maximum at −1.333 abstract weight units during the learning reversibility check ANN run, which indicates a potentially powerful effect on overall membrane potential.

Supplementary note 3

Repeatability of learning. In order to demonstrate that the memristive synapses can repeatably perform learning as shown in Fig. 3, the experiment was performed three consecutive times. In each experiment run all devices were initialised through the memristor control instrument to values corresponding to an abstract weight of 0 (within the limits of the measurement noise). The software was then initialised to 0 weights (on top of which measurement noise was added during operation). The three experimental runs are shown in Supplementary Figure S8 where we observe that the last run is the one from Fig. 3. In all cases the data clearly shows that both neurons start from a situation where they both display no specialisation on either pattern and simultaneously their membrane potentials show no inherent preference to either pattern. At the end of each run, the prototype patterns have been successfully segregated.

Supplementary note 4

Learning reversibility timescale check. The learning experiments shown in Fig. 4 did not fully elucidate whether the system is truly capable of developing a new, stable weight configuration during reversal learning since the memristor synapses still exhibited notable changes by the end of the 1200 trial WTA run. For that reason, immediately after the conclusion of the experimental runs from Fig. 4 an additional reversibility run lasting 3000 trials was carried out. Results are shown in Supplementary Figure S9 where we notice that after 1200 trials the system has not yet fully settled at a stable weight configuration. After 3000 trials, however, the reversal is very clear as indicated by the computed membrane potentials of the hardware neuron. Thus the system is truly capable of not just learning, but if necessary also complete relearning.

Supplementary note 5

Functional form of plasticity. The estimated functional form for software plasticity is shown in Supplementary Figure S10. This relies on the two exponential fits for f^{LTP} and f^{LTD} from Fig. 1(c).

Supplementary note 6

Fitting converged conductance *versus* LTP/LTD composition. The linear fitting used for Fig. 2(a) followed the formula:

$$S(p) = a \cdot p + b \tag{14}$$

where $S(p)$ is final, converged conductance as a function of LTP/LTD composition p, $a = 3.87 \cdot 10^{-7}$ and $b = 3.73 \cdot 10^{-6}$).

Supplementary note 7

Quantifying the weight evolution noise during WTA network runs. In order to quantify the noise present in the evolution of the memristive synapse weights throughout the WTA network trial shown in Fig. 3 we first fitted the weight data to first order exponentials of the form:

$$w(k) = a \cdot e^{-\frac{k}{b}} + c \tag{15}$$

where $w(k)$ the memristor synapse weight at input event k and a, b, c fitting parameters. Results are shown in Supplementary Figure S11. The residuals were then extracted and their standard deviations computed. These results are summarised along with overall weight change throughout the WTA run Δw_{total} as estimated by the fittings in Supplementary Table T6.

It is important to note that the standard deviations of the residual levels computed will include contributions from at least three main components: First, The stochastic nature of the input signal. Second, in the case of the hardware synapses, random measurement error. Third, extra error introduced by the mismatch between the choice of fitting function and the underlying synaptic weight evolution dynamics. The random measurement error can be quantified by examining the standard deviation in the resistive state of the hardware synapses as computed from the neutral region seen in the left half of Supplementary Figure S1 (residual *versus* exponential fitting to mitigate spontaneous drift effects). If we then combine the standard deviation in the resistive state with the mapping between resistive state and weight from Supplementary Table T4 we can compute the contribution of the measurement error to overall noise levels in units of abstract weight. These values are shown for the hardware synapses in Supplementary Table T6. Notably, software and hardware synapses show similar levels of overall noise. Note: throughout this analysis we have assumed that the distribution of residuals is normal. Whilst this may not be necessarily true, the overall values of standard deviation are still indicative of noise levels in the system.

Supplementary note 8

Comparison case: what if software synapses are immune to noise? For the purposes of comparison we have also carried out a WTA learning experiment where the software synapses were implemented without added noise. Results are seen in Supplementary Figures S12 and S13. The difference is very clear especially with regard to the progress of learning between the neuron using software and the neuron using hardware synapses (Supplementary Figure S12(b)), but also when examining the evolution of synaptic weight.

Supplementary note 9

Quantifying quality of convergence. The quality of convergence achieved during the experimental runs shown in Fig. 2(a) is very hard to assess reliably given the difficulty in extrapolating how memristors might behave after the end of each 10^4-point data block. However, as a simple check the memristor resistive state evolution during each data block -conductance $g(k)$- was fitted to an exponential as per eq. 15. The constant offset term c then denotes the expected resistive state saturation level for each data block. Extrapolated convergence values are plotted in Supplementary Figure S14 along with two data block runs and their corresponding exponential fits. We note that the exponential fits in both cases tend to qualitatively underestimate the degree with which the resistive state continues to drop/increase towards the end of each data block. Further study is required in order to understand precisely why this occurs and determine a more suitable fitting model. Moreover, we notice that on most occasions (38/40), despite the possible unsuitability of the exponential model as a fitting function, the extrapolated resistive state convergence points are within 400 nS of their counterparts as extracted from the experimental data. In the remaining two cases the conductance *versus* input event number plots does not exhibit a sufficiently strong saturating trajectory and causes the extrapolation to fail. We therefore conclude that: i) Incomplete convergence cannot be ruled out as a reason behind the qualitatively worse convergence observed for run no. 2, ii) preliminary checks attempting to fit data to exponentials do not lend support to this hypothesis but do not disprove it either and iii) exponential fits may be poor predictors of future memristor behaviour.

Supplementary note 10

Measurement instrumentation. All experiments in this work were carried out using our in-house developed instrument shown in Supplementary Figure S15(a) that derives from the work in [3]. The instrument uses a trans-impedance amplifier-based (TIA) read-out procedure which is schematically described in Supplementary Figure S15. DUT resistance is always assessed at the read-out voltage of 0.2 V.

Supplementary note 11

Materials level interpretation. In this section we attempt to link the observations made throughout this paper to a materials-level interpretation on a working hypothesis basis, which is, however, not the focus of the current publication.

Pristine memristive Pt/TiO2/Pt devices used in this study being their lives at highly insulating states due to the stoichiometry of the oxide layer. The process of electroforming then serves to create a conductive path within the oxide, commonly called conductive filament (CF). During electroforming an external electric field is applied between the two electrodes oxygen vacancies and/or metal (titanium in our case) interstitials migrate towards the anode and accumulate until bridging the electrodes, consequently, reducing the pristine resistive state towards a low resistive state (LRS). It is now well known that the CF consists of oxygen vacancies in devices operating through valence change memory (VCM) mechanisms such as ours. Subsequent application of voltage in the opposite polarity (in systems exhibiting switching of the bipolar type) resets the device towards high resistive states (HRS) by thinning, or breaking the CF. In this step, the oxygen ions fill some of oxygen vacancies cites disrupting the filament continuity, thus increasing the resistance the high resistive state (HRS) [4, 5, 6, 7]. It is worth mentioning, on one hand, that the pristine state is never recovered because of the influence of all the filament branches that where created during electroforming step, thus forcing the device to toggle between some LRS and HRS resistance values far below the initial, pristine level. The stochastic nature of the CFs explains the variability in the threshold voltages; the voltage levels at which the device begin to experience switching towards lower (higher) resistive states. Notably, the precise magnitude of the voltage stimulus pulses affects the values of HRS and LRS between which the device can toggle: higher applied voltages enhance the HRS/LRS contrast, but at the expense of endurance and switching graduality (higher voltages - most of the resistive state change tends to occur upon the first pulse [8]). When applying long trains of constant voltage pulses the vacancies/ions susceptible to drift under the accumulated energy gradually migrate, resulting in a progressive shift in resistance until reaching a plateau (convergence), where no more vacancies/ions can drift unless the pulse amplitude or/and width are increased.

It is important to specify that especially when operating at near-threshold levels many pulses are needed to migrate all the vacancies/ions sensitive to the applied voltage. This is the basic explanation of the results depicted in Figure 2. The more LTP (LTD) events are applied to the device the higher (lower) the conductance becomes. At a probability of 0.05% of LTP events for example, the number of positive pulses overcomes the negative ones resulting in drifting more vacancies thus building the CF. The nature of the experiment in runs 2 and 4, which consists of applying LTP and LTD events to the device and slowly and regularly increasing (decreasing) the number of LTP (LTD) events at each event block, causes the final (and ideally converged) conductance to increase smoothly. However, larger variability in converged conductances was observed for run 1 and run 3, where the probabilities of LTP (LTD) events was randomly applied. These more abrupt changes in pulsing regime render the overall vacancy/ion drift more aggressive throughout each run and thus are the possible cause of the increased end result variability.

It is worth mentioning that the filamentary nature of the switching of our devices makes the ON state very stable, possibly because at that state the filament bridge is completely formed; determining whether this is indeed the case requires further study. However, the CF is disrupted and interrupted in the OFF state, and at the end of each pulse train the OFF resistive state drifts slightly, particularly immediately following stimulation interruption. This is observed in Supplementary Figure S6 where the test device drifts from 8.4 kΩ to 8.9 kΩ within the first 30 minutes after stimulus interruption. The drift continued with smaller changes, from 8.9 kΩ to 9 kΩ for the following 2 hours, as can be seen. We attribute this to the active component of the resistive switching devices, named nano-battery effect [9]. Indeed, Valov et al. have studied this phenomenon and demonstrated that an inherent electromotive force (emf) exists within the device that causes the resistance value to change even when no external voltage is applied. This emf or diffusion is generated by the inhomogeneous charge distribution and charge motion resulting from the electroforming or set/reset processes. This happens at HRS where vacancy/ion drift occurs slowly, however, when a CF is completely formed, in the LRS, this phenomenon does not occur. The nanobattery effect is partially masked in this proof of principle study, as learning occurs under a constant barrage of input data, which allows the vacancies/ions to drift and achieve repeatedly relatively stable conductance values. However, carefully studying the influence of this phenomenon in further exploiting this work should be considered. Interesting open questions for further research would be whether the presence of this emf materially affects the balance between potentiation and depression during network operation and to precisely what extent drift in resistive state after stimulation interruption is tolerable (even though results from Supplementary Figures S1 and S4 suggest the overall effect is relatively small).

Supplementary References

[1] Xing, J., Serb, A., Khiat, A., Berdan, R., Xu, H., and Prodromakis, T. (2015).

[2] Serb, A., Khiat, A., and Prodromakis, T. *Electron Devices, IEEE Transactions on* **62**(11), 3685–3691 Nov (2015).

[3] Berdan, R., Serb, A., Khiat, A., Regoutz, A., Papavassiliou, C., and Prodromakis, T. *Electron Devices, IEEE Transactions on* **62**(7), 2190–2196 (2015).

[4] Waser, R. and Aono, M. *Nature materials* **6**(11), 833–840 (2007).

[5] Kim, K. M., Jeong, D. S., and Hwang, C. S. *Nanotechnology* **22**(25), 254002 (2011).

[6] Yang, J. J., Pickett, M. D., Li, X., Ohlberg, D. A., Stewart, D. R., and Williams, R. S. *Nature nanotechnology* **3**(7), 429–433 (2008).

[7] Shima, H., Zhong, N., and Akinaga, H. *Applied Physics Letters* **94**(8), 2905 (2009).

[8] Gupta, I., Serb, A., Berdan, R., Khiat, A., Regoutz, A., and Prodromakis, T. *Circuits and Systems II: Express Briefs, IEEE Transactions on* **62**(7), 676–680 July (2015).

[9] Valov, I., Linn, E., Tappertzhofen, S., Schmelzer, S., Van den Hurk, J., Lentz, F., and Waser, R. *Nature communications* **4**, 1771 (2013).