EE 222 W 18 Lecture 14, Mar 1, 2018

2nd Midterm Exam on March 6

How to Prepare ?

Focus on concepts, designs with moderate calculations

- SRAM analysis, noise margins
- power comparison of different architectures (parallel, pipeline, ...)
- Adiabatic circuits & energy recycling
- Multiple-V_{DD} designs & voltage level shifting
- Memristors
- Neuromorphic computing

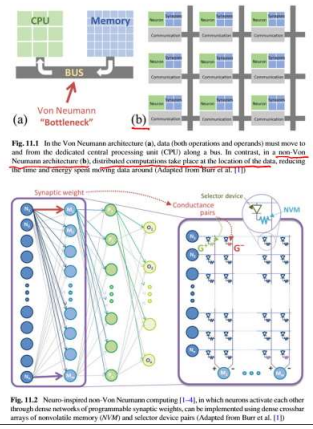


Fig. 11.1 In the Von Neumann architecture (a), data (both operations and operands) must move to and from the dedicated central processing unit (CPU) along a bus. In contrast, in a non-Von Neumann architecture (b), distributed computations take place at the location of the data, reducing the time and energy spent moving data around (Adapted from Burr et al. [1])

Fig. 11.2 Neuro-inspired non-Von Neumann computing [1–4], in which neurons activate each other through dense networks of programmable synaptic weights, can be implemented using dense crossbar arrays of nonvolatile memory (*NVM*) and selector device pairs (Adapted from Burr et al. [1])

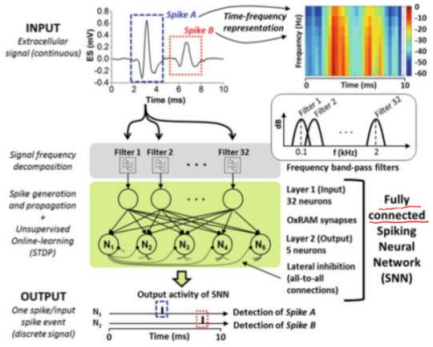


Fig. 13.10 Functional schematic of FCNN. The extracellular signal is fed through 32 frequency band-pass filters which are connected to the FCNN. Synapses are based on HfO_2-based OxRAM devices. Output neurons become selective to different input spikes shapes (Adapted from Werner et al. [23])

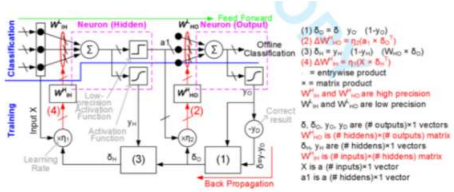
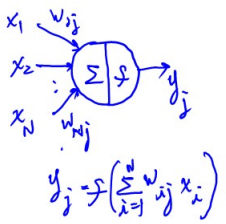
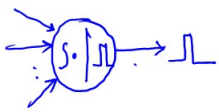


Figure 19 The algorithm flow of the feedforward inference (FF) and backward propagation (BP) for weight update in a feedforward neural network. In the FF inference, the low precision (i.e. 1-bit binary) weights and 1-bit step function neuron could be used for the computation. In the backward propagation (BP), still the low precision weights could be used for calculating the error for weight update, but the weight update should be accumulated on a higher precision, e.g. 6-bit weights (for MNIST dataset). Adapted from [127].

Artificial Neural Network (ANN)
vs. Spiking Neural Network (SNN)



$$y_j = f\left(\sum_{i=1}^{N} w_{ij} x_i\right)$$



Synaptic weights change as a function of relative timing of pre- and postsynaptic spikes

$\Delta t = t_{post} - t_{pre}$

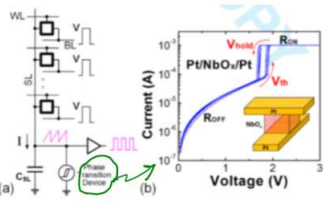


Figure 15 (a) Using a phase transition device at the end of the column to perform the thresholding function, serving as an oscillatory neuron node. Adapted from [112]. (b) The threshold switching I-V characteristics of the NbO_2 device based on meal-insulator-transition mechanism. Adapted from [113].

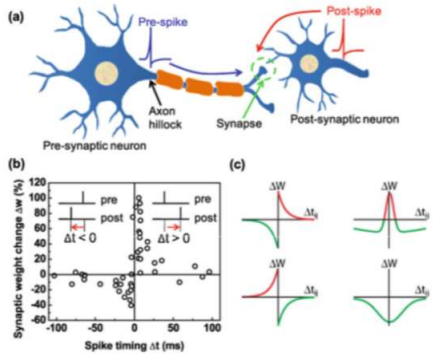


Fig. 2.9 Synaptic plasticity. (**a**) Relative timings of neuronal spikes from the presynaptic neuron and the postsynaptic neuron determine the weight change in synapse. (**b**) Synaptic weight change is plotted as a function of relative timing of pre- and post-spikes (Reprinted with permission from Bi and Poo [82]) (**c**) Diverse forms of STDP (Reprinted with permission from Shouval et al. [76])

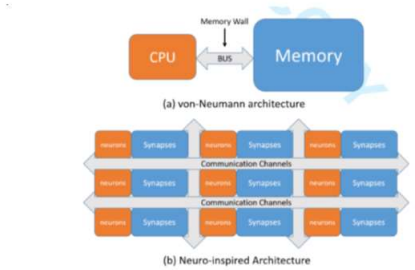


Figure 1 A revolutional shift of the computing paradigm from the computation-centric (von-Neumann architecture) to the data-centric (neuro-inspired architecture).

Table 1.1 Categories of different design options for hardware implementation of neuro-inspired computing. Representative porotypes are shown

	Off-the-shelf technologies	CMOS ASIC	Emerging resistive synaptic devices
Digital representation	GPUs [9] FPGAs [10]	TPU [13] CNN accelerators [11, 12]	Analog synapses: UCSB's 12 × 12 crossbar array [18]
			Binary synapses: ASU/Tsinghua's 16 Mb RRAM macro [19]
Spike representation	SpiNNaker [14]	Analog neuron: HICANN [15]	IBM's 256 × 256 PCM array with STDP neuron circuits [20]
		Digital neuron: TrueNorth [16]	

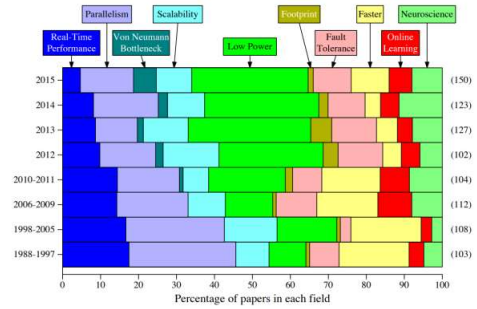


Fig. 3. Ten different motivations for developing neuromorphic systems, and over time, the percentage of the papers in the literature that have indicated that motivation as one of the primary reasons they have pursued the development of neuromorphic systems.

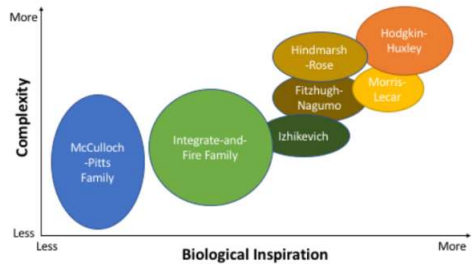


Fig. 5. A qualitative comparison of neuron models in terms of biological inspiration and complexity of the neuron model.

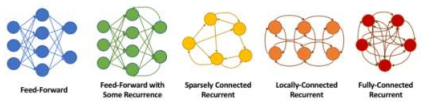


Fig. 6. Different network topologies that might be desired for neuromorphic systems. Determining the level of connectivity that is required for a neuromorphic implementation and then finding the appropriate hardware that can accommodate that level of connectivity is often a non-trivial exercise.

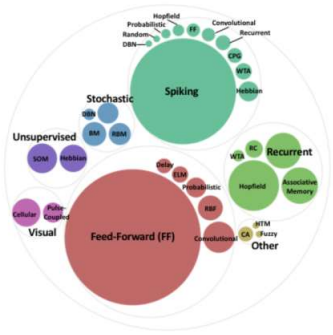


Fig. 7. A breakdown of network models in neuromorphic implementations, grouped by overall type and sized to reflect the number of associates papers.

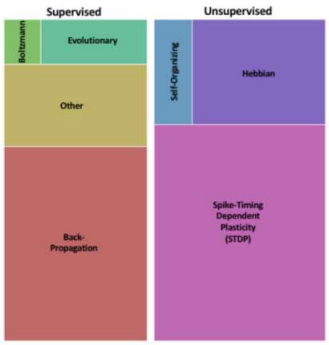


Fig. 9. An overview of on-chip training/learning algorithms. The size of the box corresponds to the number of papers in that category.

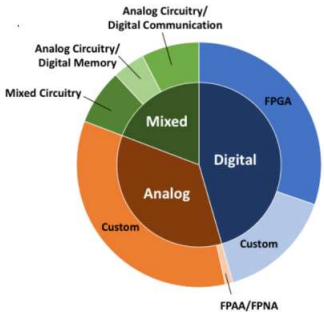


Fig. 10. An overview of hardware implementations in neuromorphic computing. These implementations are relatively basic hardware implementations and do not contain the more unusual device components discussed in Section V-B.

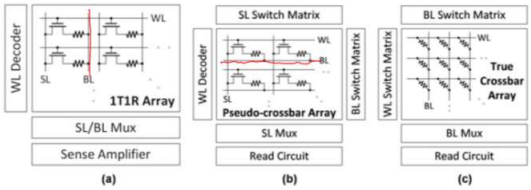


Fig. 9.1 Representative resistive NVM arrays with peripheral circuits are shown. (**a**) 1T1R array with row-by-row operation. (**b**) Pseudo-crossbar array implemented by rotating BL in 1T1R memory array. (**c**) True crossbar array without selector transistors for array-level parallelism

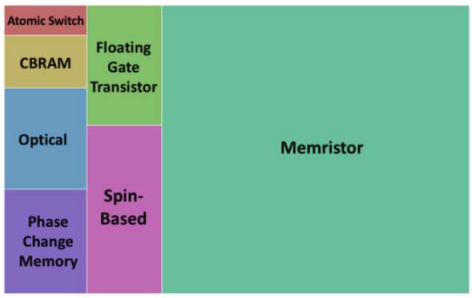


Fig. 11. Device-level components and their relative popularity in neuromorphic systems. The size of the boxes corresponds to the number of works referenced that have included those components.

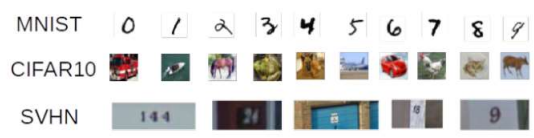
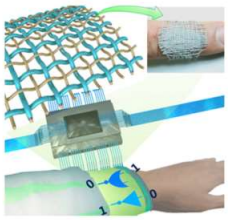


Fig. 14. Examples from different image data sets (MNIST [2618], CIFAR10 [2619], and SVHN [2620]) to which neuromorphic systems have been applied for classification purposes.

Fig. 15. Major neuromorphic computing research challenges in different fields.



100% real fabric functional circuit with a missing 4th electronic component of memristor beyond resistor, capacitor and inductor

A functional circuit with a memristor, which was fabricated on an interwoven fabric, was demonstrated. The memristor is a missing fourth electronic component beyond well-known resistor, capacitor, and inductor. This research paves the way for wearable and smart fibertronics, finally constructing a fabric computer

Article | Spring 2018

Fibertronics is an emerging technology in which electronic components are embedded on textile fibers. Sometimes called advanced electronic-textile (e-textiles), fibertronics has been explored as an ideal platform for developing wearable electronics. Conventional fabric-based electronics suffer from various technical problems that need to be solved, including their requirements for low power consumption, high integration density, water-stable material for outdoor use, and complicated fabrication processes, such as masking, etching, and evaporation steps. For these reasons, fibertronics has recently attracted a great deal of attention as true e-textile nanotechnology.

read more >>

Functional Circuitry on Commercial Fabric via Textile-Compatible Nanoscale Film Coating Process for Fibertronics

Hagyoul Bae†, Byung Chul Jang†§, Hongkeun Park‡, Soo-Ho Jung∥, Hye Moon Lee∥, Jun-Young Park†, Seung-Bae Jeon†, Gyeongho Son†, Il-Woong Tcho†, Kyoungsik Yu†, Sung Gap Im‡, Sung-Yool Choi*†§, and Yang-Kyu Choi*†

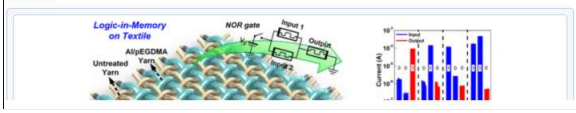
†School of Electrical Engineering, ‡School of Chemical and Biomolecular Engineering, and §Graphene/2D Materials Research Center, Korea Advanced Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, South Korea

∥ Powder and Ceramics Division, Korea Institute of Materials Science (KIMS), 797 Chanwondaero, Changwon, 51508, South Korea

Nano Lett., **2017**, *17* (10), pp 6443–6452
DOI: 10.1021/acs.nanolett.7b03435
Publication Date (Web): September 11, 2017

*E-mail: ykchoi@ee.kaist.ac.kr., *E-mail: sungyool.choi@kaist.ac.kr.

Abstract



volatile power-hungry electronic components, and modest battery storage. Here, we report a novel poly(ethylene glycol dimethacrylate) (pEGDMA)-textile memristive nonvolatile logic-in-memory circuit, enabling normally off computing, that can overcome those challenges. To form the metal electrode and resistive switching layer, strands of cotton yarn were coated with aluminum (Al) using a solution dip coating method, and the pEGDMA was conformally applied using an initiated chemical vapor deposition process. The intersection of two Al/pEGDMA coated yarns becomes a unit memristor in the lattice structure. The pEGDMA-Textile Memristor (ETM), a form of crossbar array, was interwoven using a grid of Al/pEGDMA coated yarns and untreated yarns. The former were employed in the active memristor and the latter suppressed cell-to-cell disturbance. We experimentally demonstrated for the first time that the basic Boolean functions, including a half adder as well as NOT, NOR, OR, AND, and NAND logic gates, are successfully implemented with the ETM crossbar array on a fabric substrate. This research may represent a breakthrough development for practical wearable and smart fibertronics.

The research team also demonstrated that the basic Boolean functions, including NOT, NOR, OR, AND, and NAND logic gates, were reliably implemented for data processing and storage within the ETM based crossbar array. Furthermore, they experimentally demonstrated a "half adder", which is a kind of logic functional block. It was just comprised of only 5 ETMs. The results of the study show the feasibility of the ETM circuit for energy-efficient wearable fibertronics.

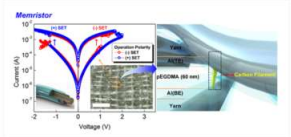


Figure 2. Current-voltage characteristics of the fabricated ETM array on the cotton substrate with a pEGDMA film thickness of 60 nm. The inset shows the optical microscope image of the fabricated ETM array and the schematic image of the Al/pEGDMA-coated yarn. The memristor device operates under low operating voltage, ranging both from -1 to +1.5 V and from -1.5 to +1 V in negative-bias SET and positive-bias SET operation, respectively. The right figure shows the formation of a conductive carbon filament bridging between top and bottom electrodes via the pEGDMA.

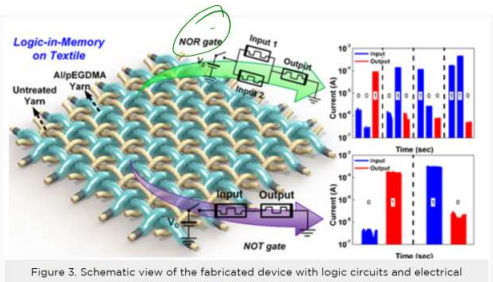


Figure 3. Schematic view of the fabricated device with logic circuits and electrical measured data of the NOR and NOT gate on fabric.

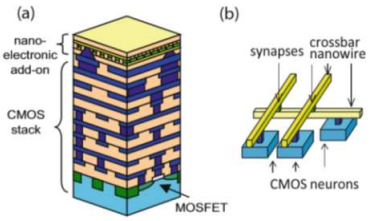


Fig. 6.1 CMOL circuits. (**a**) A cartoon of a hybrid CMOS/memristor integrated circuit. (**b**) The example of three CMOS cells (neurons) interconnected via corresponding crossbar nanowires (dendrites and axons) and cross-point memristive devices (synapses), which are located above CMOS layer

Fig. 1.2 An analogy between a biologic synapse and the resistive synaptic device

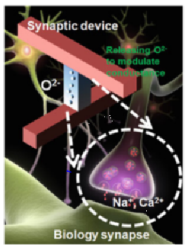
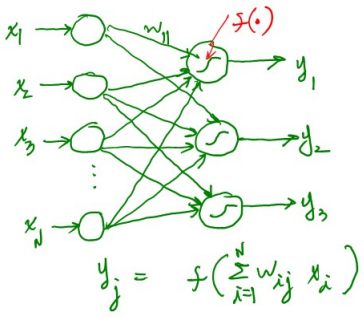


Table 1.2 Summary of the desirable performance metrics for synaptic devices

Performance metrics	Desired targets
Device dimension	<10 nm
Multilevel states' number	>100[a]
Energy consumption	<10 fJ/programming pulse
Dynamic range	>100[a]
Retention	>10 years[a]
Endurance	$>10^9$ updates[a]

Note: [a]These numbers are application dependent



Linearity in Weight Update The linearity in weight update refers to the linearity of the curve between the device conductance and the number of identical programming pulses. Ideally, this should be a linear relationship for the direct mapping of the weights in the algorithms to the conductance in the devices. However, the resistive synaptic devices generally have the nonlinearity in weight update (see Fig. 1.3). The trajectory of the long-term-potentiation (LTP) process that increases the conductance differs from that of the long-term-depression (LTD) process that decreases the conductance. The weight tends to saturate at the end of LTP or LTD processes. This nonlinearity is undesired because the change of the weight (ΔW) depends on the current weight (W), or in other words, the weight update has a history dependence. Recent results have shown that this nonlinearity has caused the learning accuracy loss in the neural networks [41, 42].

Programming Energy Consumption The estimated energy consumption per synaptic event is around 1 ~ 10 fJ in biological synapses. Most RRAM/CBRAM devices show a programming energy around 100 fJ ~ 10 pJ, while most PCM devices may have even higher programming energy 10 ~ 100 pJ. The fundamental challenge is that it is much more difficult (thus paying more energy) to move the ions/defects in solid-state devices than moving calcium ions in the liquid environment in biological synapses. A back-of-envelope calculation is given as follows. In biological synapses, the spike voltage is ~10 mV, the ionic current ~1 nA, and the spike period ~1 ms; therefore, the energy is about 10 fJ. In resistive synaptic devices, the typical programming voltage is ~1 V, and the programming current is typically >μA; although the programming speed can be accelerated less than the real time to be <μs, still the energy is on the order of pJ. Further device engineering is thus needed to reduce the energy consumption.

Retention and Endurance During the online training, the weights are frequently updated, and the data retention requirement can be relaxed. When the training is complete, the resistive synaptic should behave as a long-term memory with a data retention in the order of 10 years at elevated temperature similarly as the

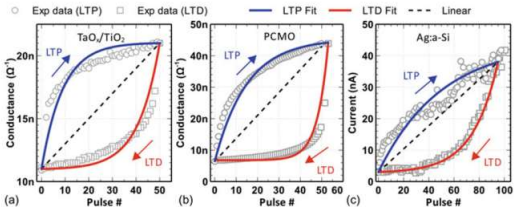


Fig. 1.3 The measured nonlinearity in the weight update reported from the literature: (**a**) TaO_x/TiO_2 device [39], (**b**) PCMO device [36], and (**c**) Ag:a-Si device [33]

requirement of NVM. The number of endurance is much application dependent, relying on how many weight updates are required in the training processes. For a relatively simple task (i.e., the MNIST handwritten digit recognition [43]), 60,000 training images with 50 training epochs (to repeated) give a maximum weight update possibility to be 3×10^6 updates. Actually not every synapse is updated in the training; thus, an endurance $\sim 10^4$ is sufficient for training MNIST dataset [19]. However, considering more challenging tasks (i.e., ImageNet challenge [44]), much more endurance may be required.

Uniformity and Variability Poor uniformity or significant variability in emerging NVMs is a major barrier for digital memory applications. In contrast, the neural networks promise robustness against device variations. The device variations could partially be tolerated by two mechanisms: the massive (thus maybe redundant) connections between neuron nodes by synaptic arrays and the iterative weight update process during the training. The degree of variations that can be tolerated at the system level strongly depends on the network architecture and the accuracy required by the target application. The device-algorithm co-simulations have shown the reasonable robustness against device variations in different neural networks [42, 45].

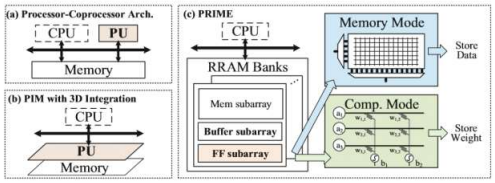
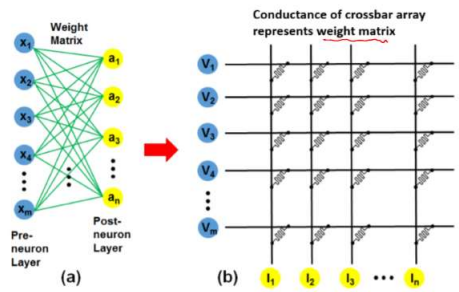
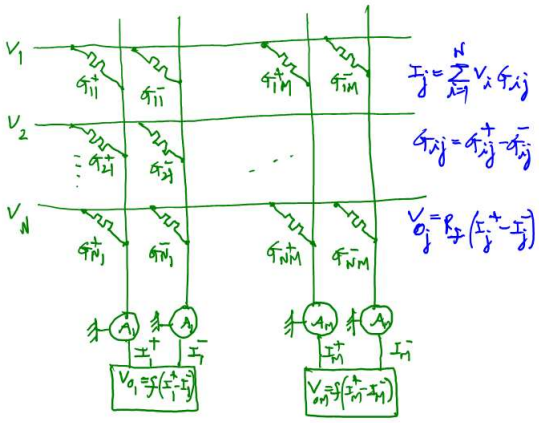


Fig. 10.1 (a) Traditional processor-coprocessor architecture with shared memory; (b) PIM architecture using 3D integration technologies; (c) PRIME design





weight update

$$\Delta w_{ij} = \alpha \sum_{\text{training data}} \left(y_j^* - y_j\right) f'\left(\sum_{i=1}^{N} w_{ij} x_i\right) x_i$$

y_j^* = target, α = learning rate

$f'(\cdot)$ = derivative of the activation function

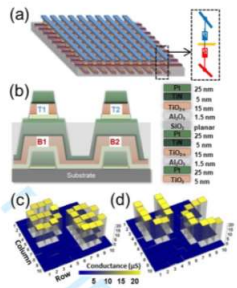


Figure 11 3D integration of memristor crossbar: (a) Circuit, (b) cross-section, and (c, d) experimental results of two vertically integrated TiO_x planar memristor crossbars. Adapted from Ref. [103].

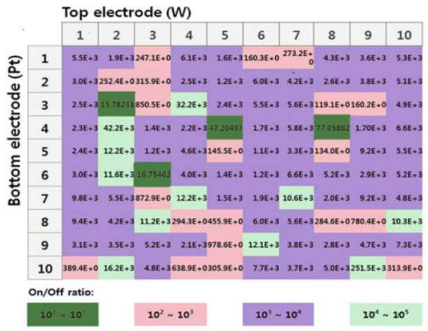


Fig. 3.6 Similar on/off ratio of 100 bits in the 1 kb PCMO-synapse array

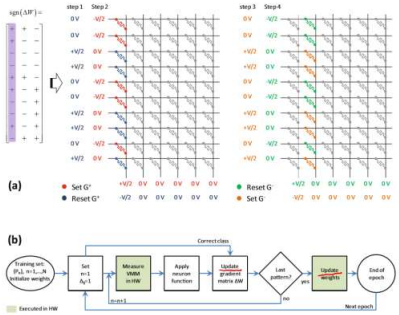
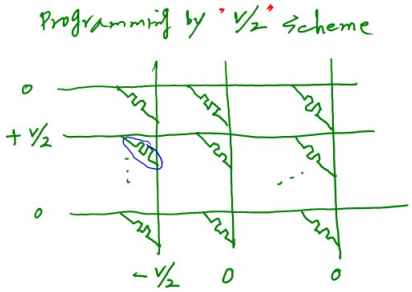


Fig. 6.9 (**a**) The first four steps of crossbar update. The sign of the gradient matrix (on the *left*), which is obtained after one epoch of training, specifies the direction of the state update for each device in crossbar circuit, i.e., whether to incrementally set or reset the device. The update is performed using the V/2 scheme with appropriate chosen voltages (on the *right*). The voltage shown in *red(green)* and *blue(orange)* are for the first and second steps, respectively. (**b**) Flow chart of the training algorithm. *Gray boxes* show the steps implemented in hardware, while all remaining steps were emulated in software



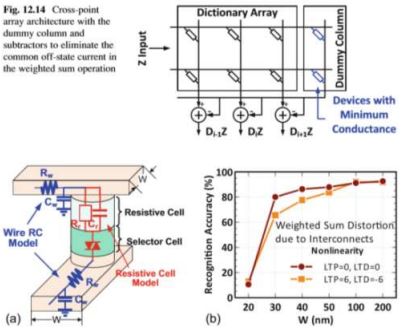


Fig. 12.14 Cross-point array architecture with the dummy column and subtractors to eliminate the common off-state current in the weighted sum operation

Fig. 12.15 (**a**) Sub-circuit module of a synaptic device cell (*W* wire width). The cell consists of a resistive synaptic device and a selector. The resistive cell has capacitor (C_j) in parallel with the cell resistor (R_j). There are also wire resistors (R_W) and capacitors (C_W) for top and bottom interconnect. Sub-circuit is duplicated for the entire array to perform SPICE simulation. (**b**) Learning accuracy with different wire widths. Smaller wire width will degrade the learning accuracy due to the IR drop along the interconnects

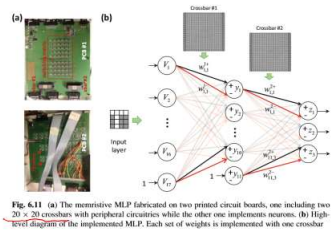
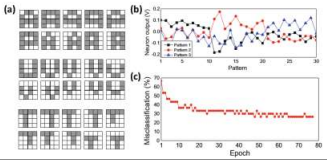


Fig. 6.11 (**a**) The memristive MLP fabricated on two printed circuit boards, one including two 20 × 20 crossbars with peripheral circuitries while the other one implements neurons. (**b**) High-level diagram of the implemented MLP. Each set of weights is implemented with one crossbar



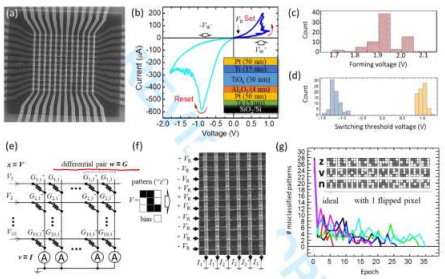


Figure 10 Perceptron classifier demonstration: (a) integrated 12×12 crossbar with an Al_2O_3/TiO_{2-x} memristor at each cross-point; (b) a typical *I-V* curve of a formed memristor; histograms of forming voltages (c) and effective switching thresholds voltages (d) for set and reset transitions; (e) perceptron implementation using a 10×6 fragment of the memristive crossbar; (f) example of the classification operation for a specific input pattern; and (g) the convergence of network outputs, in the process of training, to the perfect (zero-error) set, for 6 different initial states. The classification was considered successful when the output signal corresponding to the correct class of the applied pattern was larger than all other outputs. The insets in panels (b) and (g) show device's cross-section and the used input pattern set, correspondingly. On panel (d), the positive / negative switching threshold voltages were defined as the smallest amplitudes of 500-μs voltage pulses that caused resistance change by more than 2 kΩ in memristors pre-set to their high / low resistive states. Adapted from [37, T01].

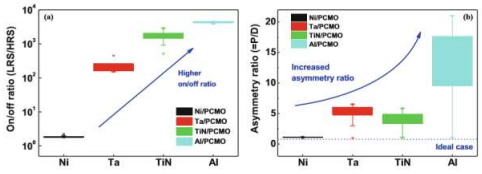


Fig. 3.13 Each sample exhibited different (**a**) on/off ratio and (**b**) asymmetry ratio

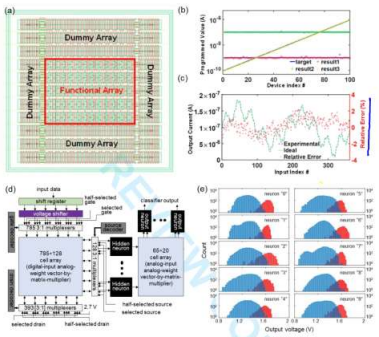
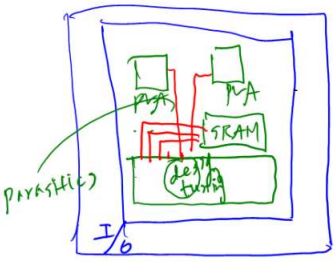


Figure 12 NOR flash memory circuits redesigned for neuro-inspired computing: (a) Layout of a 55-nm vector-matrix multiplication circuit with a 10×(10×2) cell array and auxiliary pass-gates and (b, c) its experimental test results, for (b) cell tuning (measured vs. target weights) and (c) 4-input vector-by-vector multiplication. The four inputs are quasi-DC currents sampled from sine functions with different frequencies. 2-layer MLP based on 180-nm industrial-grade floating-gate devices: (a) high-level architecture (with the weight tuning circuitry for the 2nd array not shown for clarity), and (b) histograms of output voltages for all 10,000 MNIST test patterns. The classification of one pattern takes time below 1 μs time and energy below 20 nJ. Adapted from [107, 108].



what blocks?
Datapath
control logic
PLA
ROM
SRAM
what size?
Interconnect-centric design ← wiring first

B. T. Murphy
defect density d_0
chip area A

$$Y = Y_0 e^{-d_0 A} > p^n$$

Murphy's yield model