Perspective

Towards spike-based machine intelligence with neuromorphic computing

https://doi.org/10.1038/s41586-019-1677-2

Received: 23 July 2018

Accepted: 9 July 2019

Published online: 27 November 2019

Kaushik Roy[1]*, Akhilesh Jaiswal[1] & Priyadarshini Panda[1]

Guided by brain-like 'spiking' computational frameworks, neuromorphic computing—brain-inspired computing for machine intelligence—promises to realize artificial intelligence while reducing the energy requirements of computing platforms. This interdisciplinary field began with the implementation of silicon circuits for biological neural routines, but has evolved to encompass the hardware implementation of algorithms with spike-based encoding and event-driven representations. Here we provide an overview of the developments in neuromorphic computing for both algorithms and hardware and highlight the fundamentals of learning and hardware frameworks. We discuss the main challenges and the future prospects of neuromorphic computing, with emphasis on algorithm–hardware codesign.

Throughout history, the promise of creating technology with brain-like ability has been a source of innovation. Previously, scientists have contended that information transfer in the brain occurs through different channels and frequencies, as in a radio. Today, scientists argue that the brain is like a computer. With the development of neural networks, computers today have demonstrated extraordinary abilities in several cognition tasks—for example, the ability of AlphaGo to defeat human players at the strategic board game Go[1]. Although this performance is truly impressive, a key question still remains: what is the computing cost involved in such activities?

The human brain performs impressive feats (for example, simultaneous recognition, reasoning, control and movement), with a power budget[2] of nearly 20 W. By contrast, a standard computer performing only recognition among 1,000 different kinds of objects[3] expends about 250 W. Although the brain remains vastly unexplored, its remarkable capability may be attributed to three foundational observations from neuroscience: vast connectivity, structural and functional organizational hierarchy, and time-dependent neuronal and synaptic functionality[4,5] (Fig. 1a). Neurons are the computational primitive elements of the brain that exchange or transfer information through discrete action potentials or 'spikes', and synapses are the storage elements underlying memory and learning. The human brain has a network of billions of neurons, interconnected through trillions of synapses. Spike-based temporal processing allows sparse and efficient information transfer in the brain. Studies have also revealed that the visual system of primates is organized as a hierarchical cascade of interconnected areas[2] that gradually transforms the representation of an object into a robust format, facilitating perceptive abilities.

Inspired by the brain's hierarchical structure and neuro-synaptic framework, state-of-the-art artificial intelligence is, by and large, implemented using neural networks. In fact, modern deep-learning networks (DLNs) are essentially artefacts of hierarchy built by composing several layers or transformations that represent different latent features in the input[6] (Fig. 1b). Such neural networks are fuelled by hardware computing systems that fundamentally rely on basic silicon transistors. Digital logic in massive computing platforms comprises billions of transistors integrated on a single silicon die. Reminiscent of the hierarchical organization of the brain, various silicon-based computational aspects are arranged in a hierarchical fashion to allow efficient data exchange (see Fig. 1c).

Despite this superficial resemblance, there exists a sharp contrast between the computing principles of the brain and silicon-based computers. A few key differences include: (1) the segregation of computations (the processing unit) and storage (the memory unit) in computers contrasts with the co-located computing (neurons) and storage (synapses) mechanisms found in the brain; (2) the massive three-dimensional connectivity in the brain is currently beyond the reach of silicon technology, which is limited by two-dimensional connections and finite number of interconnecting metal layers and routeing protocols; and (3) transistors are largely used as switches to construct deterministic Boolean (digital) circuits, in contrast to the spike-based event-driven computations in the brain that are inherently stochastic[7]. Nevertheless, silicon computing platforms (for example, graphics processing unit (GPU) cloud servers) have been one of the enabling factors in the current deep-learning revolution. However, a major bottleneck prohibiting the realization of 'ubiquitous intelligence' (spanning cloud-based servers to edge devices) is the large energy and throughput requirement. For example, running a deep network on an embedded smart-glass processor supported by a typical 2.1 W h battery would drain the battery completely within just 25 minutes (ref. [8]).

Guided by the brain, hardware systems that implement neuronal and synaptic computations through spike-driven communication may enable energy-efficient machine intelligence. Neuromorphic computing efforts (see Fig. 2) originated in the 1980s to mimic biological neuron and synapse functionality with transistors, quickly evolving to encompass the event-driven nature of computations (an artefact of discrete 'spikes'). Eventually, in the early 2000s, such research efforts facilitated the emergence of large-scale neuromorphic chips. Today, the advantages and limitations of spike-driven computations (specifically, learning with 'spikes') are being actively explored by algorithm

[1]Purdue University, West Lafayette, IN, USA. *e-mail: kaushik@purdue.edu

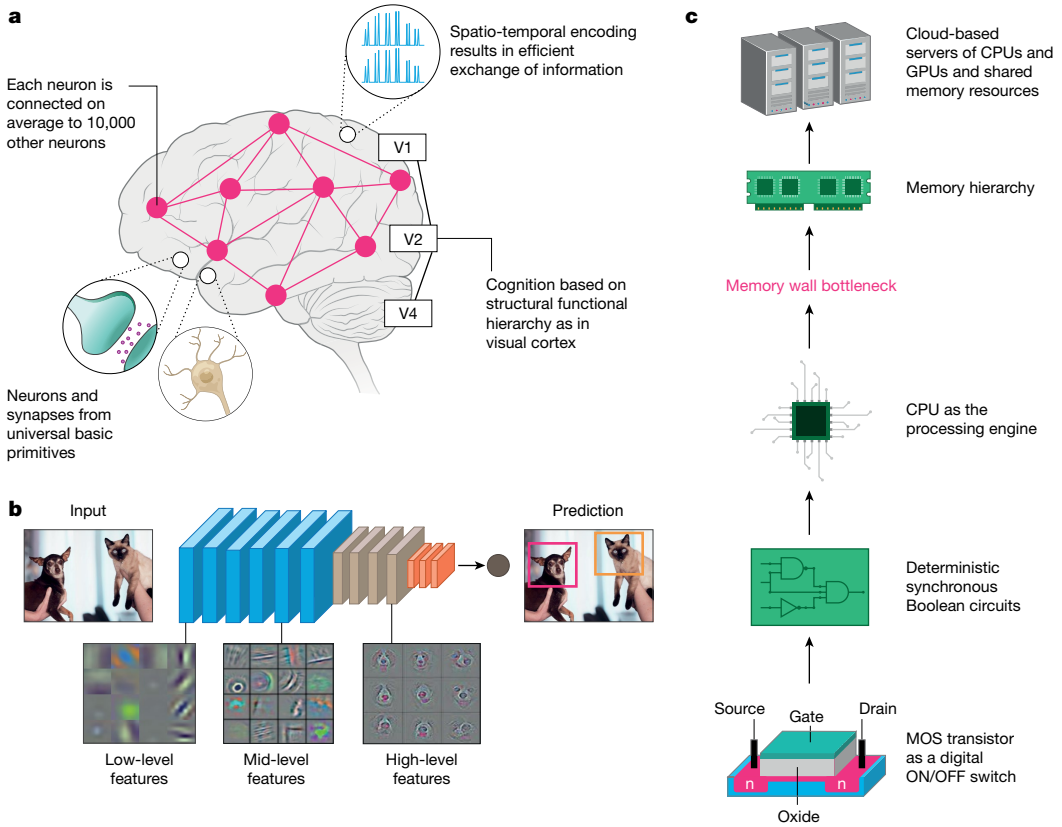

Fig. 1 | Key attributes of biological and silicon-based computing frameworks. a, A schematic of the organizational principles of the brain. The intertwined network of neurons and synapses with temporal spike processing enables rapid and efficient flow of information between different areas. **b**, A deep convolutional neural network performing objection detection on an image. These networks are multi-layered and use synaptic storage and neuronal nonlinearity that learn broad representations about the data. After training using backpropagation[12], the features learned at each layer show interesting patterns. The first layer learns general features such as edges and colour blobs. As we go deeper into the network, the learned features become more specific, representing object parts (such as the eyes or nose of the dog) to full objects (such as the face of the dog). Such generic-to-specific transition is representative of the hierarchical arrangement of the visual cortex. **c**, A state-of-the-art silicon computing ecosystem. Broadly, the computing hierarchy is divided into processing units and memory storage. The physical separation of the processing unit and the memory hierarchy results in the well known 'memory wall bottleneck'[94]. Today's deep neural networks are trained on powerful cloud servers, yielding incredible accuracy although incurring huge energy consumption.

designers to drive scalable, energy-efficient 'spiking neural networks' (SNNs). In this context, we can describe the field of neuromorphic computing as a synergistic effort that is equally weighted across both hardware and algorithmic domains to enable spike-based artificial intelligence. We first address the 'intelligence' (or algorithmic) aspects, including different learning mechanisms (unsupervised and supervised spike-based or gradient-descent schemes), while highlighting the need to exploit spatio-temporal event representations. A majority of this discussion focuses on applications for vision and related tasks, such as image recognition and detection. We then investigate the 'computation' (or hardware) aspects including analog computing, digital neuromorphic systems, beyond both von Neumann (the state-of-the-art architecture for digital computing systems) and silicon (representing the basic field-effect-transistor device that fuels today's computing platforms) technology. Finally, we discuss the prospects of algorithm–hardware codesign wherein algorithmic resilience can be used to counter hardware vulnerability, thereby achieving the optimal trade-off between energy efficiency and accuracy.

Algorithmic outlook

Spiking neural networks

The seminal paper from Maass[9] categorizes neural networks into three generations based on their underlying neuronal functionality. The first generation, referred to as McCulloch–Pitt perceptrons, performs a thresholding operation resulting in a digital (1, 0) output[10]. The second generation—based on, for example, a sigmoid unit or a rectified linear unit[11] (ReLU)—adds continuous nonlinearity to the neuronal unit, which enables it to evaluate a continuous set of output values. This nonlinearity upgrade between the first- and second-generation networks had a key role in enabling the scaling of neural networks for complex applications and deeper implementations. Current DLNs, which have multiple hidden layers between input and output, are all based on such second-generation neurons. In fact, owing to their continuous neuronal functionality, these models support gradient-descent-based backpropagation learning[12]—the standard algorithm for training DLNs today. The third generation of networks use spiking neurons primarily of the 'integrate-and-fire' type[13] that exchange information via spikes (Fig. 3).

The most important distinction between the second- and third-generation networks is in the nature of information processing. The former generation uses real-valued computation (say, the amplitude of the signal), whereas SNNs use the timing of the signals (or the spikes) to process information. Spikes are essentially binary events, either 0 or 1. As can be seen in Fig. 3a, a neuronal unit in an SNN is only active when it receives or emits spikes—it is therefore event-driven, which can contribute to energy efficiency over a given period of time. SNN units that do not experience any events remain idle. This is in contrast to DLNs, in which all units are active irrespective of the real-valued input or output

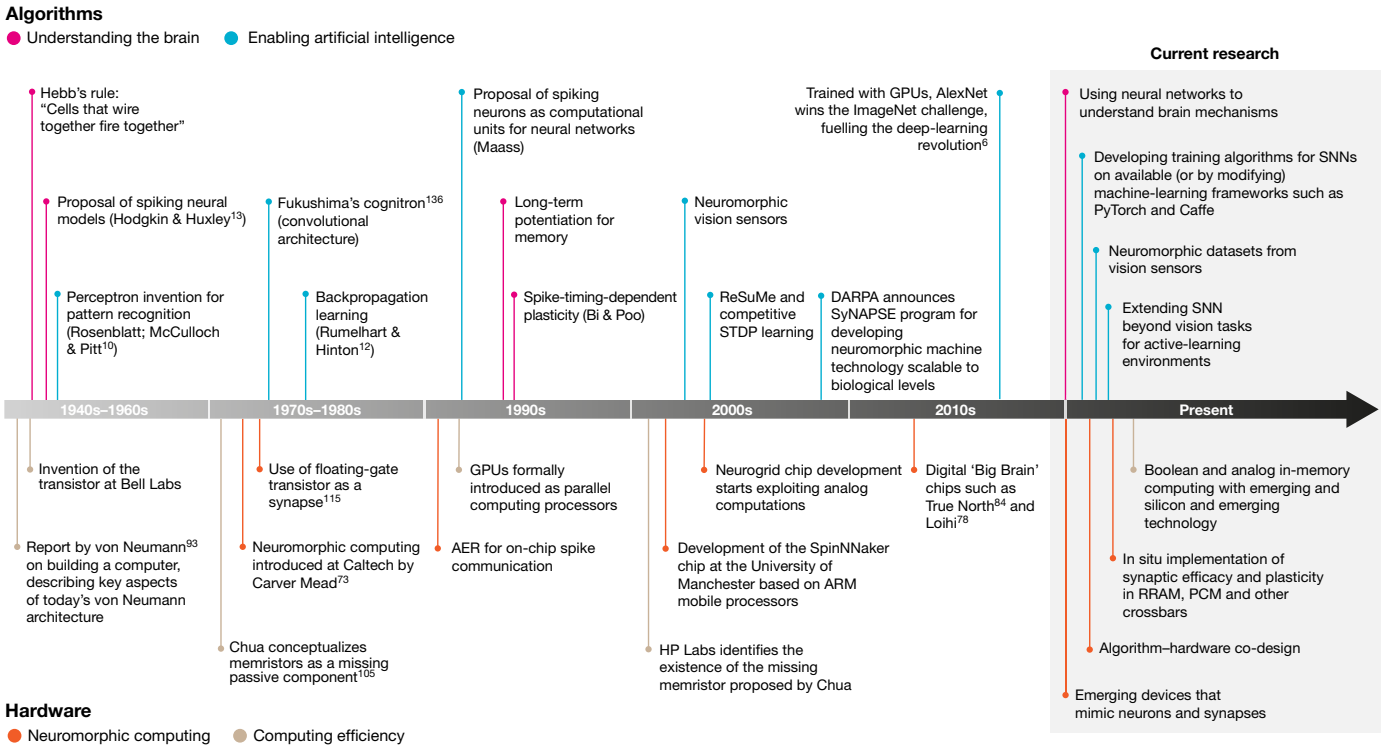

Fig. 2 | Timeline of major discoveries and advances in intelligent computing, from the 1940s to the present[6,10,14,73,78,84,93,105,115,136,150,151]. For hardware, we have indicated discoveries from two perspectives—those motivated towards neuromorphic computing or that have enabled brain-like computations and 'intelligence' with hardware innovations; and those motivated towards computing efficiency, or that have enabled faster and more energy-efficient Boolean computations. From an algorithmic perspective, we have indicated the discoveries as motivated towards understanding the brain, that is, driven by neuroscience and biological sciences; and motivated towards enabling artificial intelligence, that is, driven by engineering and applied sciences. Note that this is not a complete or comprehensive list of all discoveries. 'Current research' does not necessarily imply that such efforts have not been explored in the past; instead, we have emphasized key aspects of ongoing and promising research in the field.

values. Furthermore, the fact that the inputs in an SNN are either 1 or 0 reduces the mathematical dot-product operation, $\sum_i V_i \times w_i$ (detailed in Fig. 3a), to a less computationally intensive summation operation.

Different spiking neuron models, such as leaky integrate-and-fire (LIF) (Fig. 3b) and Hodgkin–Huxley[13], have been proposed to describe the generation of spikes at different levels of bio-fidelity. Similarly, for synaptic plasticity, schemes such as Hebbian[14] and non-Hebbian have been proposed[15]. Synaptic plasticity—the modulation of synaptic weights, which translates to learning in SNNs—relies on the relative timing of pre- and post-synaptic spikes (Fig. 3c). A suitable spiking neuron model with proper synaptic plasticity while exploiting event-based, data-driven updates (with event-based sensors[16,17]) is a major goal among neuromorphic engineers, to enable computationally efficient intelligence applications such as recognition and inference, among others.

Exploiting event-based data with SNNs

We believe that the ultimate advantage of SNNs comes from their ability to fully exploit spatio-temporal event-based information. Today, we have reasonably mature neuromorphic sensors[16,18] that can record dynamic changes in activity from an environment in real-time. Such dynamic sensory data can be combined with the temporal processing capability of SNNs to enable extremely low-power computing. In fact, using time as an additional input dimension, SNNs record valuable information in a sparse manner, compared with the frame-driven approaches that are traditionally used by DLNs (see Fig. 3). This can lead to efficient implementation of SNN frameworks, computing optical visual flow[19,20] or stereo vision to achieve depth perception[21,22], that, in combination with spike-based-learning rules, can yield efficient training. Researchers in the robotics community have already demonstrated the benefit of using event-based sensors for tracking and gesture recognition, among other applications[19,21,22]. However, most of these applications use a DLN to perform recognition.

A major restriction in the use of SNNs with such sensors is the lack of appropriate training algorithms that can efficiently utilize the timing information of the spiking neurons. Practically, in terms of accuracy, SNNs are still behind their second-generation deep-learning counterparts in most learning tasks. It is evident that spiking neurons have a discontinuous functionality, and emit discrete spikes that are non-differentiable (see Fig. 3); hence they cannot use the gradient-descent backpropagation techniques that are fundamental to conventional neural network training.

Another restriction on SNNs is spike-based data availability. Although the ideal situation requires the input to SNNs to be spike trains with timing information, the performance of SNN training algorithms is evaluated on existing static-image datasets, for example CIFAR[23] or ImageNet[24], for recognition. Such static-frame-based data are then converted to spike trains using appropriate encoding techniques, such as rate coding or rank-order coding[25] (see Fig. 3d). Although encoding techniques enable us to evaluate the performance of SNNs on traditional benchmark datasets, we need to move beyond static-image classification tasks. The ultimate competence of SNNs should arise from their capability to process and perceive continuous input streams from the ever-changing real world, just as our brains do effortlessly. At present, we have neither good benchmark datasets nor the metrics to evaluate such real-world performance of SNNs. More research into gathering appropriate benchmark datasets, such as dynamic vision sensor data[26] or driving and navigation instances[27,28], is vital.

(Here we refer to the second-generation continuous neural networks as DLNs to differentiate them from spike-based computing. We note

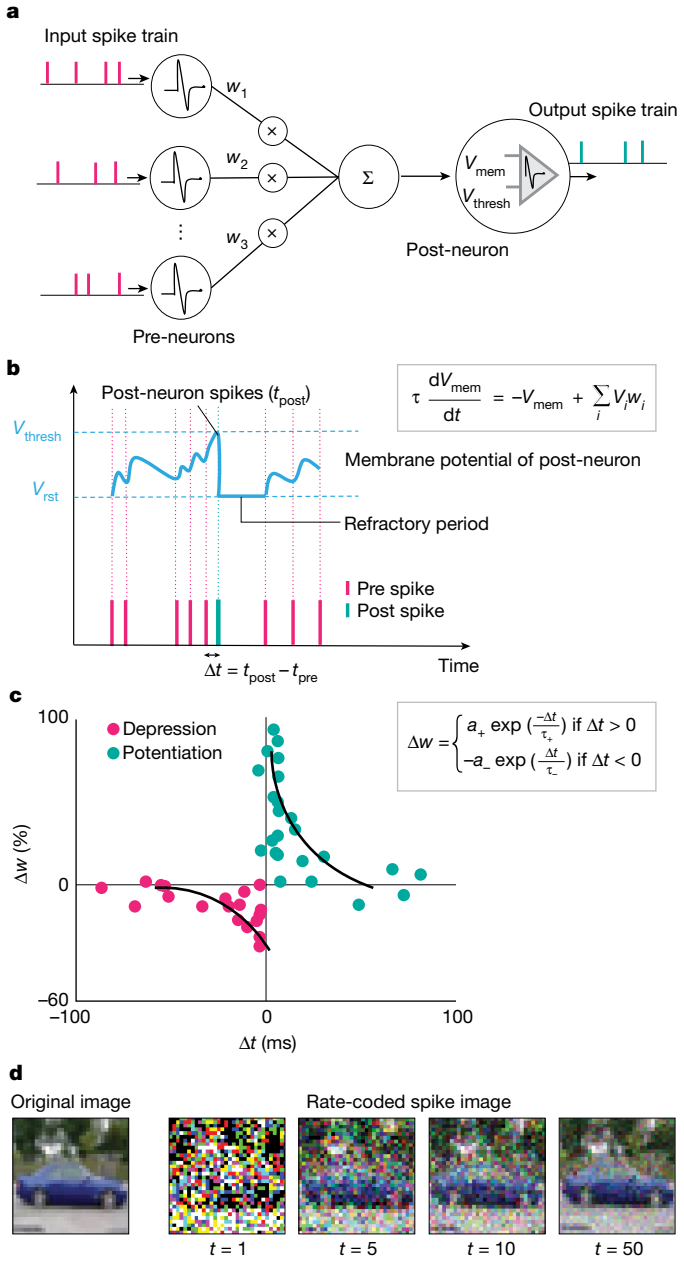

Fig. 3 | SNN computational models. a, A neural network, comprising a post-neuron driven by input pre-neurons. The pre-neuronal spikes, V_i are modulated by synaptic weights, w_i to produce a resultant current, $\sum_i V_i \times w_i$ (equivalent to a dot-product operation) at a given time. The resulting current affects the membrane potential of the post-neuron. **b**, The dynamics of LIF spiking neurons is shown. The membrane potential, V_{mem} integrates incoming spikes and leaks with time constant, τ in the absence of spikes. The post-neuron generates an outgoing spike whenever V_{mem} crosses a threshold, V_{thresh}. A refractory period ensues after spike generation, during which V_{mem} of the post-neuron is not affected. **c**, The spike-timing-dependent plasticity (STDP) formulation based on experimental data is shown, where a_+, a_-, τ_+ and τ_- are learning rates and time-constants governing the weight change, Δw. The synaptic weights w_i are updated on the basis of the time difference ($\Delta t = t_{post} - t_{pre}$) between the pre-neuron and post-neuron spikes. **d**, An input image (static-frame data) is converted to a spike map over various time steps using rate coding. Each pixel generates a Poisson spike train with a firing rate proportional to the pixel intensity. When the spike maps are summed over several time steps (the spike map labelled $t = 5$ is a summation of maps from $t = 1$ to $t = 5$), they start to resemble the input. Hence, spike-based encoding preserves the integrity of the input image and also binarizes the data in the temporal domain. It is evident that LIF behaviour and random-input spike-generation bring stochasticity to the internal dynamics of an SNN. Note that rank-order coding can also be used to generate spike data[25].

that SNNs can also be implemented on a deep architecture with convolutional hierarchy while performing spiking neuronal functions.)

Learning in SNNs

Conversion-based approaches

The idea of a conversion-based approach is to obtain an SNN that yields the same input–output mapping for a given task as that of a DLN. Essentially, a trained DLN is converted to an SNN using weight rescaling and normalization methods to match the characteristics of a nonlinear continuous output neuron to that of the leak time constants, refractory period, membrane threshold and other functionalities of a spiking neuron[29–34]. Such approaches have thus far been able to yield the most competitive accuracy on large-scale spiking networks in image classification (including on the ImageNet dataset). In conversion-based approaches, the advantage is that the burden of training in the temporal domain is removed. A DLN is trained on frame-based data using available frameworks such as Tensorflow[35] that offer training-related flexibility. Conversion requires parsing the trained DLN on event-based data (obtained by rate coding of the static-image dataset) and then applying simple transformations. However, such methods have inherent limitations. The output value of a nonlinear neuron—using, for example, a hyperbolic tangent (tanh) or a normalized exponential (softmax) function—can take both positive and negative values, whereas the rate of a spiking neuron can only be positive. Thus, negative values will always be discarded, leading to a decline in accuracy of the converted SNNs. Another problem with conversion is obtaining the optimal firing rate at each layer without any drastic performance loss. Recent works[29–31] have proposed practical solutions to determine optimal firing rates, and additional constraints (such as noise or leaky ReLUs) are introduced during training of the DLN to better match the spiking neuron's firing rate[36]. Today, conversion approaches yield state-of-the-art accuracy for image-recognition tasks that parallel the classification performance of DLNs. It is noteworthy that the inference time for SNNs that are converted from DLNs turns out be very large (of the order of a few thousand time steps), leading to increased latency as well as degraded energy efficiency.

Spike-based approaches

In a spike-based approach, SNNs are trained using timing information and therefore offer the obvious advantages of sparsity and efficiency in overall spiking dynamics. Researchers have adopted two main directions[37]: unsupervised (training without labelled data), and supervised (training with labelled data). Early works in supervised learning were ReSuMe[38] and the tempotron[39], which demonstrate simple spike-based learning in a single-layer SNN using a variant of spike-timing-dependent plasticity (STDP) to perform classification. Since then, research efforts have been directed towards integrating global backpropagation-like spike-based error gradient descent to enable supervised learning in multi-layer SNNs. Most works that rely on backpropagation estimate a differentiable approximate function for the spiking neuronal functionality so that gradient descent can be performed (Fig. 4a). SpikeProp[40] and related variants[41,42] have derived a backpropagation rule for SNNs by fixing a target spike train at the output layer. Recent works[43–46] perform stochastic gradient descent on real-valued membrane potentials with the goal that the correct output neuron will fire more spikes randomly (instead of having a precise target spike train). These methods have achieved state-of-the-art results for deep convolutional SNNs for small-scale image recognition tasks such as digit classification on the MNIST (Modified National Institute of Standards and Technology) handwritten digits database[47]. However, supervised learning—although more computationally efficient—has not been able to outperform conversion-based approaches in terms of accuracy for large-scale tasks.

On the other hand, inspired from neuroscience and with hardware-efficiency as the prime goal, unsupervised SNN training using local

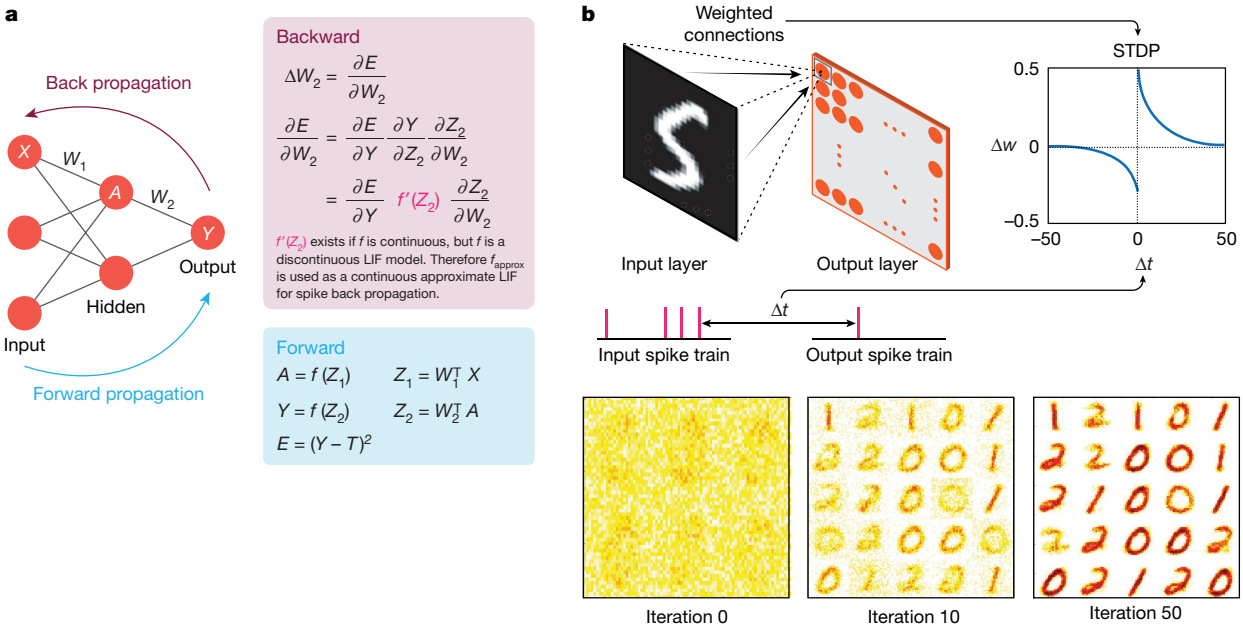

Fig. 4 | Global and local-learning principles in spiking networks. **a**, Supervised global learning with known target labels, T for a classification task. Given a feedforward network, the network forward-propagates the input values, X through hidden layer units, A to yield an output, Y. The neuronal activation values, A at the hidden layer are calculated using the weighted summation of inputs—denoted $\mathbf{Z}_1 = \mathbf{W}_1^T\mathbf{X}$ in matrix notation, combined with a nonlinear transformation, $f(\mathbf{Z}_1)$. The outputs are calculated in a similar fashion. The derivative of the error, $\mathbf{E}$ with respect to the weights ($\mathbf{W}_1$, $\mathbf{W}_2$) is then used to calculate the subsequent weight updates. Iteratively conducting the forward and backward propagation results in learning. The calculation of error derivatives requires f', which necessitates that f is continuous. Consequently, the rules of spike-based backpropagation approximate the LIF function with differentiable alternatives. The details of time-based information processing are not shown here. **b**, Local STDP unsupervised learning for digit classification. Given a two-layer topology with an input layer fully connected to all neurons in the output layer, the synaptic connections are learned through STDP. The weights are modulated on the basis of the difference in the spike timing of the input- and output-layer neurons. The weight value is increased (or decreased) if the input neuron fires before (or after) the output. With iterative training over multiple time steps, the weights—which were randomly initialized at the beginning—learn to encode a generic representation of a class of input patterns as shown (in this case, '0', '1' and '2'). Here, target labels are not required in order to perform recognition.

STDP-based learning rules[48] is also of great interest. With local learning (as we will see later in the hardware discussion), there are interesting opportunities to bring memory (synaptic storage) and computation (neuronal output) closer together. This architecture turns out to be more brain-like, as well as suitable for energy-efficient on-chip implementations. Diehl et al.[49] were one of the first to demonstrate completely unsupervised learning on an SNN, yielding comparable accuracy to deep learning on the MNIST database (Fig. 4b).

However, extending the local-learning approach to multiple layers for complex tasks is a challenge. As we go deeper into a network, the spiking probability (or the firing rate) of the neurons decreases, which we term 'vanishing forward-spike propagation'. To avoid this, most works[46,48,50–53] train a multi-layer SNN (including convolutional SNNs) with local spike-based learning in a layer-wise fashion followed by global backpropagation learning, to perform classification. Such combined local–global approaches, although promising, are still behind conversion approaches in terms of classification accuracy. Further, recent works[54,55] have shown proof-of-concept that the random projection of error signals through feedback connections in deep SNNs does enable improved learning. Such feedback-based learning methods need to be investigated further to estimate their efficacy on large-scale tasks.

Implications for learning in the binary regime

We can obtain extremely low-power and efficient computing with only binary (1/0) bit values rather than 16- or 32-bit floating point values that require additional memory. In fact, at the algorithmic level, learning in a probabilistic manner— wherein neurons spike randomly and weights have low-precision transitions—is being investigated to obtain networks with few parameters and computation operations[56–58]. Binary and ternary DLNs—in which the neuronal output and weights can take only the low-precision values −1, 0, and +1—have been proposed, which yield good performance on large-scale classification tasks[59,60]. SNNs already have a computational advantage as a result of binary spike-based processing. Furthermore, the stochasticity in neuronal dynamics of LIF neurons can improve the robustness of a network to external noise (for example, noisy input or noisy weight parameters from hardware)[61]. Then, it remains to be seen whether we can use this SNN temporal-processing architecture with appropriate learning methods, and compress weight training to a binary regime with minimal accuracy loss.

Other underexplored directions

Beyond vision tasks

So far, we have laid out approaches that have provided competitive results in, mostly, classification tasks. What about tasks beyond perception and inference on static images? SNNs offer an interesting opportunity for processing sequential data. However, there have been very few works[34] that have demonstrated the efficacy of an SNN in natural-language-processing tasks. What about reasoning and decision making with SNNs? Deep-learning researchers are heavily invested in reinforcement-learning algorithms that cause a model to learn by interacting with the environment in real time. Reinforcement learning with SNNs is very much underexplored[62,63]. The current research efforts into SNNs—particularly in the area of training algorithms—shows that the grand challenge in SNNs is to match the

performance of deep learning. Although deep-learning serves as a good baseline for comparison, we believe that SNNs can create a niche for sensory-data processing, including in robotics and autonomous control.

Lifelong learning and learning with fewer data

Deep-learning models suffer from catastrophic forgetting when they undergo continual learning. For instance, when a network trained on task A is later exposed to task B, it forgets all about task A and remembers only task B. Establishing lifelong learning in a dynamically changing environment as humans do has garnered considerable attention from the research community. This is also a nascent direction in deep-learning research, but we need to think whether the additional temporal dimension of data processing in SNNs may help us to achieve continual learning[64]. A similar direction worth exploring is one-shot learning. Learning with fewer data is the ultimate challenge and this is arguably one area where SNNs can achieve better results than deep learning. Unsupervised learning in SNNs can be combined with minimal supervision using only a fraction of the labelled training data to perform data-efficient training[46,50,65].

Forging links with neuroscience

We can take inspiration from neuroscience and apply those abstractions to learning rules in order to come up with efficient strategies. For instance, Masquelier et al.[65] employed STDP with temporal coding to mimic the visual-cortex pathway and found that such learning causes neurons to become feature selective—that is, different neurons learning different features—to different visual aspects of an image, resulting in a convolutional hierarchy of features. Similarly, incorporating dendritic learning[66] and structural plasticity[67] to improve spike-based learning by adding dendritic connections as an additional hyperparameter (a user-defined design parameter), offers interesting possibilities. A complementary body of work in the SNN domain is that of liquid state machines (LSMs)[68]. LSMs use unstructured, randomly connected recurrent networks paired with a simple linear readout. Such frameworks with spiking dynamics have shown a surprising degree of success for a variety of sequential recognition tasks[69–71], but implementing them for complex and large-scale tasks remains an open problem.

Hardware outlook

From the above description of information processing and spike-based communication, a few characteristics of hardware systems that aim to form the underlying computational framework for SNNs can easily be hypothesized. Among these are the sparse-event-driven nature of the underlying hardware as a direct manifestation of the spike-based information exchange; the requirement for tightly intertwined computing and memory fabrics inspired by the ubiquitous presence of neurons and synapses throughout the biological brain; and the need to implement complex dynamical functions—for example, neuronal and synaptic dynamics using minimal circuit primitives.

The emergence of neuromorphic computing

In the 1980s, almost four decades after the invention of the transistor, Carver Mead envisioned "smarter" and "more-efficient" silicon computer fabrics based on certain aspects of biological neural systems[72,73]. Although he suggested that his initial attempts to build such neuromorphic systems were "simple and stupid"[74], his work represented a new paradigm in hardware computing. Instead of focusing on Boolean computing based on basic AND and OR gates, Mead focused on analog distributed-computing circuits that mimicked neurons and synapses[74]. He exploited the inherent device physics of the metal-oxide–silicon (MOS) transistor in the subthreshold regime—where current–voltage characteristics are exponential—to mimic exponential neuronal dynamics[72]. Such device–circuit codesign is currently one of the most intriguing areas in neuromorphic computing, driven by novel emerging materials and associated devices.

The advent of parallel-processing GPUs

As opposed to CPUs (central processing units) that consist of one (or a few) complex computing core(s) integrated with on-chip memories, GPUs[75] consist of many simple computing cores that function in parallel, leading to high-throughput processing. GPUs were traditionally hardware accelerators for speeding up graphics applications. Of the many non-graphics applications that benefited from high-throughput computations of GPUs, deep learning is the most remarkable[6]. In fact, GPU servers are the go-to hardware platforms not only for running DLNs, but also for exploring inference and training for SNNs[76,77]. While GPUs do provide an obvious advantage via their increased programming flexibility, they do not explicitly leverage the event-driven nature of spiking computations. In this regard, event-driven 'Big Brain' neuromorphic chips can yield the most energy-efficient solutions[78,79].

The 'Big Brain' chips

'Big Brain' chips[80] are distinguished by integrating millions of neurons and synapses that render spike-based computations[78,81–86] (see Fig. 5a). Neurogrid[82] and TrueNorth[84] are two model chips based on mixed-signal analog and digital circuits, respectively. TrueNorth uses digital circuits because analog circuits tend to accumulate errors easily, and are much more susceptible to process-induced variations in chip fabrication. Neurogrid was designed to assist computational neuroscience in emulating brain activity, with complex neuronal mechanisms such as opening and closing of various ion channels and the characteristic behaviour of biological synapses[82,87]. By contrast, TrueNorth originated as a neuromorphic chip geared towards solving commercially important tasks such as recognition and classification using SNNs, and is based on simplified neural and synaptic primitives.

Taking the example of TrueNorth, two features that span different implementations of neuromorphic chips[78,88] can be highlighted as follows.

Asynchronous address event representation. First, asynchronous address event representation (AER; Fig. 5b); this differs from conventional chip design, in which all computations are performed in parts with reference to a global clock. Because SNNs are sparse and computation is only required when a spike (or an event) is generated, asynchronous event-driven computation is much more suitable. In fact, enabling event-driven computations based on spikes has historically been one of the most attractive aspects of spike-based computations[89,90].

Network-on-chip. Second, networks-on-chip (NOCs) are used for spike communication. NOCs are networks of on-chip routers that receive and transmit packets of digital information through a time-multiplexed shared bus. The use of NOCs for large-scale chips is imperative, because connectivity in a typical silicon fabrication process is largely two-dimensional, with limited flexibility in the third dimension. We note that, despite the use of NOCs, on-chip connectivity still cannot rival the three-dimensional connectivity found in the brain. TrueNorth—and subsequent large-scale digital neuromorphic chips like Loihi[78]—have demonstrated energy efficiency for SNN-based applications, taking us a step closer towards bio-fidelic implementations. However, limited connectivity, constrained bus bandwidth for NOCs and the all-digital approach remain key areas that require further investigation.

Beyond-von-Neumann computing

The sustained dimensional scaling of transistors—referred to as Moore's law[91]—has driven the ever-increasing computing power of CPUs and GPUs as well as the 'Big Brain' chips. In recent years, this increase has

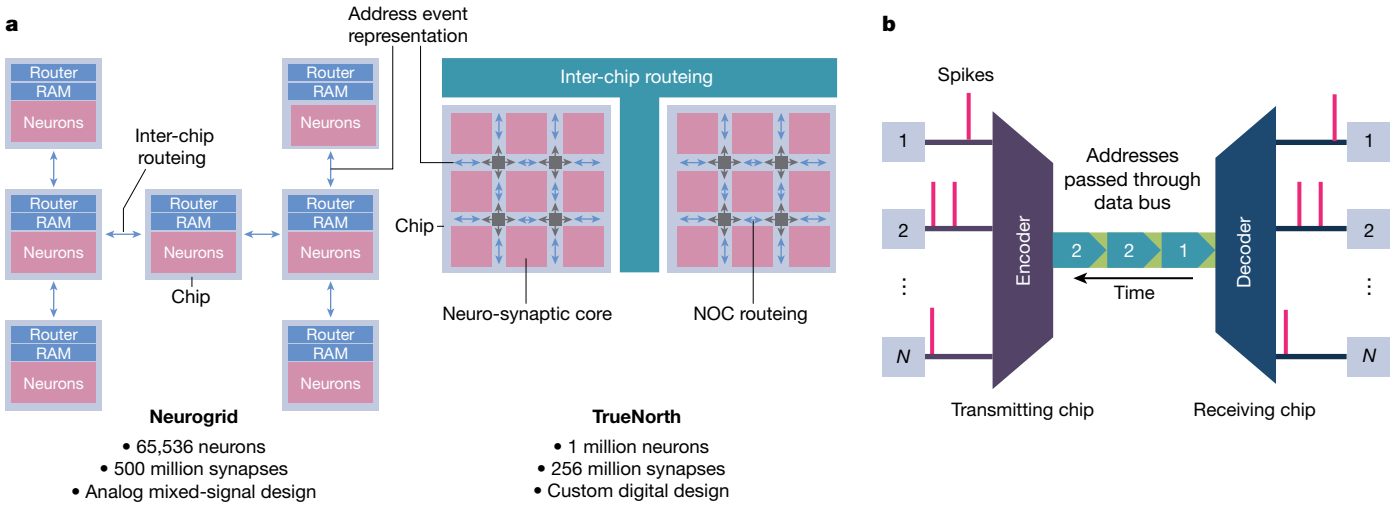

Fig. 5 | Some representative 'Big Brain' chips and AER methods. a, Among many works[78,81–84] aimed at building large-scale neuromorphic chips, we highlight two representative systems—Neurogrid and TrueNorth. Neurogrid hosts more than 65,000 neurons and 500 million synapses, and TrueNorth has 1 million neurons and 256 million synapses. Neurogrid and TrueNorth use tree and mesh routeing topology, respectively. Neurogrid uses an analog mixed-signal design and TrueNorth relies on digital primitives. In general, digital neuromorphic systems such as TrueNorth represent the membrane potential of a neuron as an *n*-bit binary word. Neuronal dynamics such as LIF behaviour are implemented by appropriately incrementing or decrementing the *n*-bit word. By contrast, analog systems represent the membrane potential as a charge stored on a capacitor. Current sources feeding into and sinking through the capacitor node mimic the desired neuronal dynamics. Despite circuit differences, in general both analog and digital systems use event-driven AER for spike communication. Event-driven communication is one of the key enablers that allows integration of such large-scale systems, while simultaneously achieving low power dissipation. **b**, The basic AER communication system. Whenever an event (a spike) is generated on the transmitter side, the corresponding address is sent over the data bus to the receiver. The receiver decodes the incoming addresses and reconstructs the sequence of the spikes on the receiver side. Thus, each spike is explicitly encoded by its location (its address) and implicitly encoded by the time that its address is sent to the data bus.

slowed down as silicon-based transistors approach their physical limit[92]. To keep pace with soaring demand for computing power, researchers have recently begun exploring a two-pronged approach to enable both 'beyond von Neumann' and 'beyond silicon' computing models. A key shortcoming of the von Neumann model[93] is the clear demarcation of a processing unit physically separated from a storage unit, connected through a bus for data transfer (see Fig. 1c). The frequent movement of data between the faster processing unit and the slower memory unit through this bandwidth-constrained bus leads to the well-known 'memory wall bottleneck' that limits computing throughput and energy efficiency[94].

One of the most promising approaches in mitigating the effect of the memory wall bottleneck is to enable 'near-memory' and 'in-memory' computing[95,96]. Near-memory computing enables co-location of memory and computing by embedding a dedicated processing engine in close proximity to the memory unit. In fact, the 'distributed computing architecture' of various 'Big Brain chips' (refer to Fig. 5) with closely placed neurons and synaptic arrays are representative of near-memory processing. By contrast, in-memory computing embeds certain aspects of computational operations within the memory array by enabling computation in the memory bit-cells or the peripheral circuits (see Fig. 6 for an example).

Non-volatile technologies

Non-volatile technologies[97–103] are usually compared to biological synapses. In fact, they exhibit two of the most important characteristics of biological synapses: synaptic efficacy and synaptic plasticity. Synaptic plasticity is the ability to modulate the weights of the synapses based on a particular learning rule. Synaptic efficacy refers to the phenomenon of generating an output based on incoming spikes. In its simplest form, this means that incoming spikes are multiplied by the stored weights of synapses, which is usually represented as programmable, analog, non-volatile resistance. The multiplied signals are summed from all the pre-neurons (neurons in a particular layer that receive input spikes) and applied as the input signal to the post-neuron (neurons in a particular layer that generate output spikes) (see Fig. 3). Figure 6 illustrates how in situ synaptic efficacy and synaptic plasticity can be accomplished using emerging non-volatile memristive technologies, arranged in a crossbar fashion[103,104]. Additionally, such crossbars can be connected in an event-driven manner using NOCs to build dense, large-scale neuromorphic processors featuring in situ in-memory computations.

Various works based on memristive technologies[105,106] such as resistive random-access memory (RRAM)[107], phase-change memory (PCM)[108] and spin-transfer torque magnetic random-access memory (STT-MRAM)[109] have been explored for both in situ dot-product computations and synaptic learning based on STDP rules. RRAMs (oxide-based and conductive-bridge-based[107]) are electric-field-driven devices that rely on filament formation to achieve analog programmable resistance. RRAMs are prone to device-to-device and cycle-to-cycle variations[110,111], which is currently the major technical roadblock. PCMs comprise a chalcogenide material sandwiched between two electrodes that can switch its internal state between amorphous (high resistance) and crystalline (low resistance). PCM devices have comparable programming voltages and write speed to RRAMs although they suffer from high write-current and resistance drift over time[108]. Spintronic devices consist of two magnets separated by a spacer; they exhibit two resistive states depending on whether the magnetization of the two layers is in the parallel or anti-parallel direction. Spin devices exhibit almost unlimited endurance, lower write energy and faster reversal compared to RRAMs and PCMs[109]. However, the ratio of the two extreme resistive states (ON and OFF) is much smaller in spin devices than in PCMs and RRAMs.

Another class of non-volatile devices that allows tunable non-volatile resistance is a floating-gate transistor; such devices are being actively explored for synaptic storage[112–114]. In fact, floating-gate devices were the first to be proposed as non-volatile synaptic storage[115,116]. Because of their

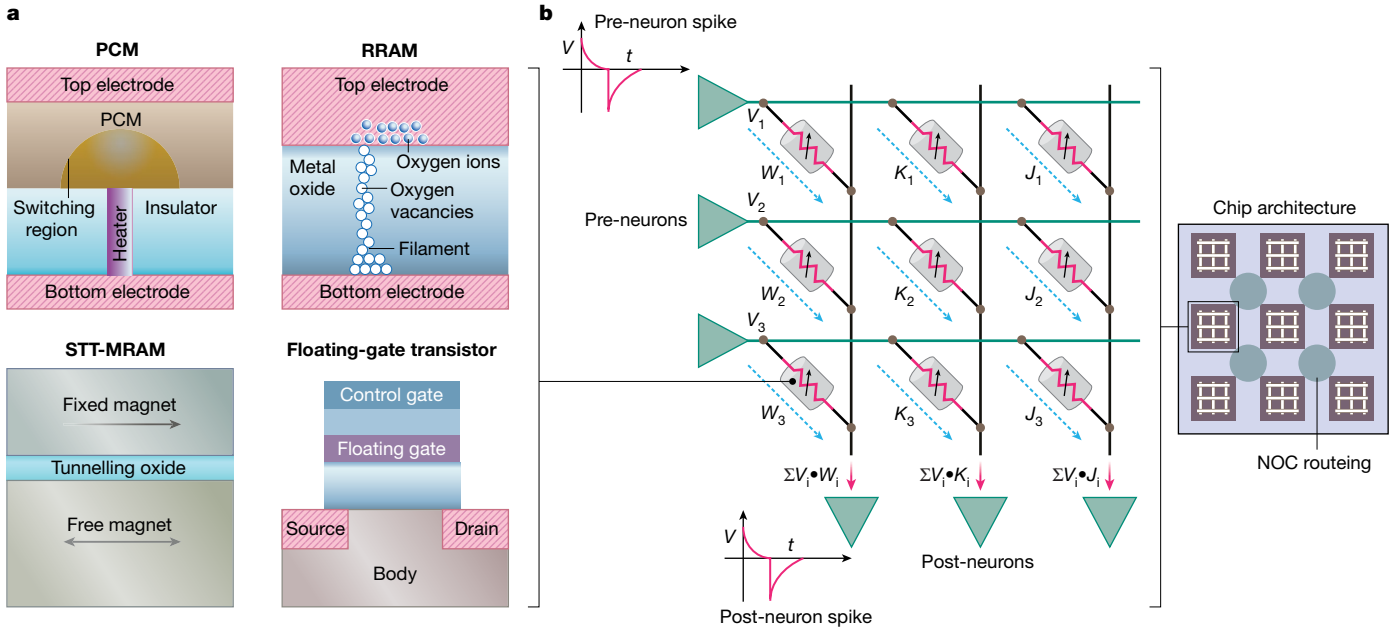

Fig. 6 | The use of non-volatile memory devices as synaptic storage. **a**, Schematics of various non-volatile technologies: PCM, RRAM, STT-MRAM and floating-gate transistor. Such non-volatile devices have been used as synaptic storage and for in situ neuro–synaptic computations[56,112–114,135,139–144], and as in-memory accelerators for a wide range of generic non-neuromorphic applications[128,145–149]. **b**, The implementation of synaptic efficacy and plasticity using memristive technologies. We show an array of memristors connected in a crossbar fashion. An incoming spike on the horizontal lines (green) results in a current that is proportional to the conductance of the memristive element in accordance with Ohm's law. Currents through multiple spiking pre-neurons are summed along vertical lines (black), a consequence of Kirchoff's current law. This results in the in-memory dot-product operation that represents synaptic efficacy. Synaptic plasticity is generally implemented in situ by appropriately applying a voltage pulse whenever the pre- and post-neurons spike on the horizontal and vertical lines, respectively, in accordance with a specific learning rule (as in STDP). The resistance values of the constituent memristors are programmed on the basis of the resulting voltage difference across the respective horizontal and vertical lines. The shape and timing of the voltage pulses to be applied for programming are chosen depending on the specific device technology. Note that floating-gate transistors, because they are three-terminal devices, require additional horizontal and/or vertical lines to enable crossbar functionality[115]. The figure also shows memristive arrays connected in a tiled fashion with NOCs that enable high-throughput in situ computations[97].

compatibility with MOS fabrication process, they are more mature than other emerging device technologies. However, the major limitation with floating-gate devices is their reduced endurance and high programming voltage in comparison to all other non-volatile technologies.

Although in situ computing and synaptic learning present attractive prospects for large-scale beyond-von-Neumann distributed computing, many challenges are yet to be overcome. Given device-to-device, cycle-to-cycle and process-induced variations, the approximate nature of computation is prone to errors that degrade overall computing efficiency as well as the accuracy of end applications. Further, the robustness of crossbar operation is affected by the presence of current sneak paths, line resistances, the source resistance of driving circuits and sensing resistance[117,118]. Non-idealities of the selector device (either a transistor or a two-terminal nonlinear device), the requirement to have analog–digital converters and limited bit precision also add to the overall complexity of designing robust computing using non-traditional synaptic devices. Additionally, writing into non-volatile devices is usually energy intensive. Furthermore, the inherent stochastic nature of such devices can result in unreliable write operations that necessitate expensive and iterative write–verify schemes[119].

Silicon (in-memory) computing

Apart from non-volatile technologies, various proposals for in-memory computing using standard silicon memories including static and dynamic random-access memories are under extensive investigation. Most of these works are focused on embedding Boolean bit-wise vector computations inside the memory arrays[120–122]. Additionally, mixed-signal analog in-memory computing operations and binary convolutions have recently been demonstrated[123,124]. In fact, in-memory computing in various forms is currently being explored for almost all the major memory technologies, including static[125] and dynamic silicon memories[126], RRAMs[127], PCMs[128] and STT-MRAMs[129]. Although most of these works have focused on generic computing applications like encryption and DLNs, they can easily find application in SNNs.

Algorithm–hardware codesign

Mixed-signal analog computing

Analog computing is highly susceptible to process-induced variations and noise, and is largely limited both in terms of area and energy consumption by the complexity and precision of analog and digital converters. Employing on-chip learning with tightly coupled analog computing frameworks will enable such systems to intrinsically adapt to process-induced variations, thereby mitigating their effect on accuracy. Localized learning with an emphasis on on-chip and on-device learning solutions has been investigated in the past[130,131] and also in more recent bio-plausible algorithmic works[54]. In essence, whether in the form of localized learning or in the use of paradigms like dendritic learning, we are of the opinion that a class of better error-resilient localized-learning algorithms—even at the cost of additional learning parameters—will be key in moving forward with analog neuromorphic computing. Additionally, the resilience of on-chip learning can be used to develop low-cost approximate analog–digital converters, without reducing the accuracy of a targeted application.

Memristive dot products

As a specific example of analog computing, memristive dot products are a promising approach towards enabling in situ neuromorphic

computing. Unfortunately, the resulting currents in memristive arrays representing the dot products have both spatial and data dependence, making crossbar circuit analysis a non-trivial, complex problem. Few works have studied the effect of crossbar non-idealities[117,132,133] and explored training approaches to mitigate the effect of dot-product inaccuracies[118,134]. Note that most of these works are focused on DLNs as opposed to SNNs. However, it is reasonable to assume that the basic device and circuit insights developed in these works are relevant for SNN implementations as well. Existing works require detailed device–circuit simulation runs that must be tightly coupled with training algorithms to diminish the accuracy loss. We believe an abstracted version of crossbar array models based on state-of-the-art devices, along with efforts to establish theoretical bounds in dot-product inaccuracies, are of immediate interest. This will enable an algorithm designer to explore new training algorithms while accounting for the hardware inconsistencies without time-consuming and iterative device–circuit–algorithm simulations.

Stochasticity

Stochastic SNNs are of substantial interest owing to the availability of emerging devices that are inherently stochastic[135,136]. Most of the recent works on the implementation of stochastic binary SNNs have focused on small-scale tasks such as MNIST digit recognition[56]. The common theme across such works is using stochastic STDP-like local learning rules to generate weight updates. We think that the temporal dimension in STDP learning provides additional bandwidth for weight updates to head in the right direction (towards achieving overall accuracy), even when constrained to the binary regime. The combination of such binary local-learning schemes with gradient-descent-based learning rules for large-scale tasks, while leveraging the stochasticity in hardware, provides interesting opportunities for energy-efficient neuromorphic systems.

Hybrid design approaches

We believe that hardware solutions based on hybrid approaches—that is, combining the advantages of various techniques on a single platform—is another important area that requires intensive investigation. Such approaches can be found in recent literature[137], where low-precision memristors are used in combination with a high-precision digital processor. There are many possible variants of such hybrid approaches, including significance-driven segregation of computational data, mixed-precision computations[137], reconfiguring conventional silicon memories as on-demand in-memory approximate accelerators[125], locally synchronous and globally asynchronous designs[138], locally analog and globally digital systems; wherein both emerging and silicon-based technologies can be used in unison to achieve improved accuracy and energy efficiency. Furthermore, such hybrid hardware can be used in tandem with hybrid spike-based learning approaches, such as locally unsupervised learning followed by globally supervised backpropagation[53]. We believe that such combined local–global learning schemes can be leveraged to reduce hardware complexity, while also minimizing performance degradation for end applications.

Conclusion

Today, enabling 'intelligence' in almost all of the technology around us has become a central theme of research spanning various disciplines. In that regard, this Perspective sets out the case for neuromorphic computing as an energy-efficient way to enable machine intelligence through synergistic advancements in both hardware (computing) and algorithms (intelligence). We began by discussing the algorithmic implications of using a spiking neural paradigm, which uses event-driven computations, in contrast to real-valued computing in conventional deep-learning paradigms. We have described the advantages and limitations for realizing learning rules (such as spike-based gradient-descent learning, unsupervised STDP and related conversion approaches from deep learning to the spiking domain) for standard classification tasks. Future algorithmic research should exploit the sparse and temporal dynamics of spike-based information processing, together with complementary neuromorphic datasets that can result in real-time recognition; and hardware development should focus on event-driven computations, co-location of memory and computational units, and mimicking dynamical neuro-synapse functionality. Of special interest are emerging non-volatile technologies enabling in situ mixed-signal analog computing. We have also discussed prospects for cross-layer optimization that enables algorithm–hardware co-design—for example, exploiting algorithmic resilience (as in local learning) and hardware feasibility (as in ease of implementing stochastic primitives). Finally,the promise of spike-based energy-efficient and intelligent systems built with traditional and emerging devices is in sync with the current interest in enabling ubiquitous intelligence. Now is the time for the interchange of ideas, with multidisciplinary efforts spanning devices, circuits, architecture and algorithms to synthesize a truly energy-efficient and intelligent machine.

1. Silver, D. et al. Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).
2. Cox, D. D. & Dean, T. Neural networks and neuroscience-inspired computer vision. *Curr. Biol.* **24**, R921–R929 (2014).
3. Milakov, M. *Deep Learning With GPUs*. https://www.nvidia.co.uk/docs/IO/147844/Deep-Learning-With-GPUs-MaximMilakov-NVIDIA.pdf (Nvidia, 2014).
4. Bullmore, E. & Sporns, O. The economy of brain network organization. *Nat. Rev. Neurosci.* **13**, 336–349 (2012).
5. Felleman, D. J. & Van Essen, D. C. Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* **1**, 1–47 (1991).
6. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* Vol. 28 (eds Pereira, F. et al.) 1097–1105 (Neural Information Processing Systems Foundation, 2012).
   This work—using deep convolutional networks—was the first to win the ImageNet challenge, fuelling the subsequent deep-learning revolution.
7. Deco, G., Rolls, E. T. & Romo, R. Stochastic dynamics as a principle of brain function. *Prog. Neurobiol.* **88**, 1–16 (2009).
8. Venkataramani, S., Roy, K. & Raghunathan, A. Efficient embedded learning for IoT devices. In *21st Asia and South Pacific Design Automation Conf.* 308–311 (IEEE, 2016).
9. Maass, W. Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* **10**, 1659–1671 (1997).
   This paper was one of the first works to provide a rigorous mathematical analysis of the computational power of spiking neurons, categorizing them as the third generation of neural networks (after perceptron and sigmoidal neurons).
10. McCulloch, W. S. & Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**, 115–133 (1943).
11. Nair, V. & Hinton, G. E. Rectified linear units improve restricted Boltzmann machines. In *Proc. 27th Int. Conf. on Machine Learning* (eds Fürnkranz, J. & Joachims, T.) 807–814 (IMLS, 2010).
12. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**, 533–536 (1986).
    This seminal work proposed gradient-descent-based backpropagation as a learning method for neural networks.
13. Izhikevich, E. M. Simple model of spiking neurons. *IEEE Trans. Neural Netw.* **14**, 1569–1572 (2003).
14. Hebb, D. O. *The Organization of Behavior: A Neuropsychological Theory* (Wiley, 1949).
15. Abbott, L. F. & Nelson, S. B. Synaptic plasticity: taming the beast. *Nat. Neurosci.* **3**, 1178–1183 (2000).
16. Liu, S.-C. & Delbruck, T. Neuromorphic sensory systems. *Curr. Opin. Neurobiol.* **20**, 288–295 (2010).
17. Lichtsteiner, P., Posch, C. & Delbruck, T. A. 128×128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits* **43**, 566–576 (2008).
18. Vanarse, A., Osseiran, A. & Rassau, A. A review of current neuromorphic approaches for vision, auditory, and olfactory sensors. *Front. Neurosci.* **10**, 115 (2016).
19. Benosman, R., Ieng, S.-H., Clercq, C., Bartolozzi, C. & Srinivasan, M. Asynchronous frameless event-based optical flow. *Neural Netw.* **27**, 32–37 (2012).
20. Wongsuphasawat, K. & Gotz, D. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Trans. Vis. Comput. Graph.* **18**, 2659–2668 (2012).
21. Rogister, P., Benosman, R., Ieng, S.-H., Lichtsteiner, P. & Delbruck, T. Asynchronous event-based binocular stereo matching. *IEEE Trans. Neural Netw. Learn. Syst.* **23**, 347–353 (2012).
22. Osswald, M., Ieng, S.-H., Benosman, R. & Indiveri, G. A spiking neural network model of 3D perception for event-based neuromorphic stereo vision systems. *Sci. Rep.* **7**, 40703 (2017).
23. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. Preprint at http://arxiv.org/abs/1207.0580 (2012).

24. Deng, J. et al. ImageNet: a large-scale hierarchical image database. In *IEEE Conf. on Computer Vision and Pattern Recognition* 248–255 (IEEE, 2009).
25. Rullen, R. V. & Thorpe, S. J. Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. *Neural Comput.* **13**, 1255–1283 (2001).
26. Hu, Y., Liu, H., Pfeiffer, M. & Delbruck, T. DVS benchmark datasets for object tracking, action recognition, and object recognition. *Front. Neurosci.* **10**, 405 (2016).
27. Geiger, A., Lenz, P., Stiller, C. & Urtasun, R. Vision meets robotics: the KITTI dataset. *Int. J. Robot. Res.* **32**, 1231–1237 (2013).
28. Barranco, F., Fermuller, C., Aloimonos, Y. & Delbruck, T. A dataset for visual navigation with neuromorphic methods. *Front. Neurosci.* **10**, 49 (2016).
29. Sengupta, A., Ye, Y., Wang, R., Liu, C. & Roy, K. Going deeper in spiking neural networks: VGG and residual architectures. *Front. Neurosci.* **13**, 95 (2019).
This paper was the first to demonstrate the competitive performance of a conversion-based spiking neural network on ImageNet data for deep neural architectures.
30. Cao, Y., Chen, Y. & Khosla, D. Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* **113**, 54–66 (2015).
31. Diehl, P. U. et al. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *Int. Joint Conf. on Neural Networks* 2933–2341 (IEEE, 2015).
32. Pérez-Carrasco, J. A. et al. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing—application to feedforward ConvNets. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 2706–2719 (2013).
33. Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M. & Liu, S.-C. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* **11**, 682 (2017).
34. Diehl, P. U., Zarrella, G., Cassidy, A. S., Pedroni, B. U. & Neftci, E. Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware. In *Int. Conf. on Rebooting Computing* 20 (IEEE, 2016).
35. Abadi, M. et al. Tensorflow: a system for large-scale machine learning. In *12th USENIX Symp. Operating Systems Design and Implementation* 265–283 (2016).
36. Hunsberger, E. & Eliasmith, C. Spiking deep networks with LIF neurons. Preprint at http://arxiv.org/abs/1510.08829 (2015).
37. Pfeiffer, M. & Pfeil, T. Deep learning with spiking neurons: opportunities and challenges. *Front. Neurosci.* **12**, 774 (2018).
38. Ponulak, F. & Kasiński, A. Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. *Neural Comput.* **22**, 467–510 (2010).
39. Gütig, R. & Sompolinsky, H. The tempotron: a neuron that learns spike-timing-based decisions. *Nat. Neurosci.* **9**, 420–428 (2006).
40. Bohte, S. M., Kok, J. N. & La Poutré, H. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* **48**, 17–37 (2002).
41. Ghosh-Dastidar, S. & Adeli, H. A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural Netw.* **22**, 1419–1431 (2009).
42. Anwani, N. & Rajendran, B. NormAD: normalized approximate descent-based supervised learning rule for spiking neurons. In *Int. Joint Conf. on Neural Networks* 2361–2368 (IEEE, 2015).
43. Lee, J. H., Delbruck, T. & Pfeiffer, M. Training deep spiking neural networks using backpropagation. *Front. Neurosci.* **10**, 508 (2016).
44. Orchard, G. et al. HFirst: a temporal approach to object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 2028–2040 (2015).
45. Mostafa, H. Supervised learning based on temporal coding in spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **29**, 3227–3235 (2018).
46. Panda, P. & Roy, K. Unsupervised regenerative learning of hierarchical features in spiking deep networks for object recognition. In *Int. Joint Conf. on Neural Networks* 299–306 (IEEE, 2016).
47. LeCun, Y., Cortes, C. & Burges, C. J. C. *The MNIST Database of Handwritten Digits* http://yann.lecun.com/exdb/mnist/ (1998).
48. Masquelier, T., Guyonneau, R. & Thorpe, S. J. Competitive STDP-based spike pattern learning. *Neural Comput.* **21**, 1259–1276 (2009).
49. Diehl, P. U. & Cook, M. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.* **9**, 99 (2015).
This is a good introduction to implementing spiking neural networks with unsupervised STDP-based learning for real-world tasks such as digit recognition.
50. Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J. & Masquelier, T. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Netw.* **99**, 56–67 (2018).
51. Neftci, E., Das, S., Pedroni, B., Kreutz-Delgado, K. & Cauwenberghs, G. Event-driven contrastive divergence for spiking neuromorphic systems. *Front. Neurosci.* **7**, 272 (2014).
52. Stromatias, E., Soto, M., Serrano-Gotarredona, T. & Linares-Barranco, B. An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data. *Front. Neurosci.* **11**, 350 (2017).
53. Lee, C., Panda, P., Srinivasan, G. & Roy, K. Training deep spiking convolutional neural networks with STDP-based unsupervised pre-training followed by supervised fine-tuning. *Front. Neurosci.* **12**, 435 (2018).
54. Mostafa, H., Ramesh, V. & Cauwenberghs, G. Deep supervised learning using local errors. *Front. Neurosci.* **12**, 608 (2018).
55. Neftci, E. O., Augustine, C., Paul, S. & Detorakis, G. Event-driven random back-propagation: enabling neuromorphic deep learning machines. *Front. Neurosci.* **11**, 324 (2017).
56. Srinivasan, G., Sengupta, A. & Roy, K. Magnetic tunnel junction based long-term short-term stochastic synapse for a spiking neural network with on-chip STDP learning. *Sci. Rep.* **6**, 29545 (2016).
57. Tavanaei, A., Masquelier, T. & Maida, A. S. Acquisition of visual features through probabilistic spike-timing-dependent plasticity. In *Int. Joint Conf. on Neural Networks* 307–314 (IEEE, 2016).
58. Bagheri, A., Simeone, O. & Rajendran, B. Training probabilistic spiking neural networks with first-to-spike decoding. In *Int. Conf. on Acoustics, Speech and Signal Processing* 2986–2990 (IEEE, 2018).
59. Rastegari, M., Ordonez, V., Redmon, J. & Farhadi, A. XNOR-Net: ImageNet classification using binary convolutional neural networks. In *Eur. Conf. on Computer Vision* 525–542 (Springer, 2016).
60. Courbariaux, M., Bengio, Y. & David, J.-P. BinaryConnect: training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems* Vol. 28 (eds Cortes, C. et al) 3123–3131 (Neural Information Processing Systems Foundation, 2015).
61. Stromatias, E. et al. Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms. *Front. Neurosci.* **9**, 222 (2015).
62. Florian, R. V. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Comput.* **19**, 1468–1502 (2007).
63. Vasilaki, E., Frémaux, N., Urbanczik, R., Senn, W. & Gerstner, W. Spike-based reinforcement learning in continuous state and action space: when policy gradient methods fail. *PLOS Comput. Biol.* **5**, e1000586 (2009).
64. Zuo, F. et al. Habituation-based synaptic plasticity and organismic learning in a quantum perovskite. *Nat. Commun.* **8**, 240 (2017).
65. Masquelier, T. & Thorpe, S. J. Unsupervised learning of visual features through spike-timing-dependent plasticity. *PLOS Comput. Biol.* **3**, e31 (2007).
66. Rao, R. P. & Sejnowski, T. J. Spike-timing-dependent Hebbian plasticity as temporal difference learning. *Neural Comput.* **13**, 2221–2237 (2001).
67. Roy, S. & Basu, A. An online unsupervised structural plasticity algorithm for spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **28**, 900–910 (2017).
68. Maass, W. Liquid state machines: motivation, theory, and applications. In *Computability in Context: Computation and Logic in the Real World* (eds Cooper, S. B. & Sorbi, A.) 275–296 (Imperial College Press, 2011).
69. Schrauwen, B., D'Haene, M., Verstraeten, D. & Van Campenhout, J. Compact hardware liquid state machines on FPGA for real-time speech recognition. *Neural Netw.* **21**, 511–523 (2008).
70. Verstraeten, D., Schrauwen, B., Stroobandt, D. & Van Campenhout, J. Isolated word recognition with the liquid state machine: a case study. *Inf. Process. Lett.* **95**, 521–528 (2005).
71. Panda, P. & Roy, K. Learning to generate sequences with combination of Hebbian and non-Hebbian plasticity in recurrent spiking neural networks. *Front. Neurosci.* **11**, 693 (2017).
72. Maher, M. A. C., Deweerth, S. P., Mahowald, M. A. & Mead, C. A. Implementing neural architectures using analog VLSI circuits. *IEEE Trans. Circ. Syst.* **36**, 643–652 (1989).
73. Mead, C. Neuromorphic electronic systems. *Proc. IEEE* **78**, 1629–1636 (1990).
This seminal work established neuromorphic electronic systems as a new paradigm in hardware computing and highlights Mead's vision of going beyond the precise and well defined nature of digital computing towards brain-like aspects.
74. Mead, C. A. Neural hardware for vision. *Eng. Sci.* **50**, 2–7 (1987).
75. *NVIDIA Launches the World's First Graphics Processing Unit GeForce 256.* https://www.nvidia.com/object/IO_20020111_5424.html (Nvidia, 1999).
76. Nageswaran, J. M., Dutt, N., Krichmar, J. L., Nicolau, A. & Veidenbaum, A. V. A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors. *Neural Netw.* **22**, 791–800 (2009).
77. Fidjeland, A. K. & Shanahan, M. P. Accelerated simulation of spiking neural networks using GPUs. In *Int. Joint. Conf. on Neural Networks* 3041–3048 (IEEE, 2010).
78. Davies, M. et al. Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* **38**, 82–99 (2018).
79. Blouw, P., Choo, X., Hunsberger, E. & Eliasmith, C. Benchmarking keyword spotting efficiency on neuromorphic hardware. In *Proc. 7th Annu. Neuro-inspired Computational Elements Workshop* 1 (ACM, 2018).
80. Hsu, J. How IBM got brainlike efficiency from the TrueNorth chip. *IEEE Spectrum* https://spectrum.ieee.org/computing/hardware/how-ibm-got-brainlike-efficiency-from-the-truenorth-chip (29 September 2014).
81. Khan, M. M. et al. SpiNNaker: mapping neural networks onto a massively parallel chip multiprocessor. In *Int. Joint Conf. on Neural Networks* 2849–2856 (IEEE, 2008).
This was one of the first works to implement a large-scale spiking neural network on hardware using event-driven computations and commercial processors.
82. Benjamin, B. V. et al. Neurogrid: a mixed-analog–digital multichip system for large-scale neural simulations. *Proc. IEEE* **102**, 699–716 (2014).
83. Schemmel, J. et al. A wafer-scale neuromorphic hardware system for large-scale neural modeling. In *Int. Symp. Circuits and Systems* 1947–1950 (IEEE, 2010).
84. Merolla, P. A. et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).
This work describes TrueNorth, the first digital custom-designed, large-scale neuromorphic processor, an outcome of the DARPA SyNAPSE programme; it was geared towards solving commercial applications through a digital neuromorphic implementation.
85. Furber, S. Large-scale neuromorphic computing systems. *J. Neural Eng.* **13**, 051001 (2016).
86. Qiao, N. et al. A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Front. Neurosci.* **9**, 141 (2015).
87. Indiveri, G. et al. Neuromorphic silicon neuron circuits. *Front. Neurosci.* **5**, 73 (2011).
88. Seo, J.-s. et al. A 45 nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons. In *Custom Integrated Circuits Conf.* 311–334 (IEEE, 2011).
89. Boahen, K. A. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Trans. Circuits Syst. II* **47**, 416–434 (2000).
This paper describes the fundamentals of address event representation and its application to neuromorphic systems.
90. Serrano-Gotarredona, R. et al. AER building blocks for multi-layer multi-chip neuromorphic vision systems. In *Advances in Neural Information Processing Systems* Vol. 18 (eds Weiss, Y., Schölkopf, B. & Platt, J. C.) 1217–1224 (Neural Information Processing Systems Foundation, 2006).
91. Moore, G. E. Cramming more components onto integrated circuits. *Proc. IEEE* **86**, 82–85 (1998).

92. Waldrop, M. M. The chips are down for Moore's law. *Nature* **530**, 144 (2016).
93. von Neumann, J. First draft of a report on the EDVAC. *IEEE Ann. Hist. Comput.* **15**, 27–75 (1993).
94. Mahapatra, N. R. & Venkatrao, B. The processor–memory bottleneck: problems and solutions. *Crossroads* **5**, 2 (1999).
95. Gokhale, M., Holmes, B. & Iobst, K. Processing in memory: the Terasys massively parallel PIM array. *Computer* **28**, 23–31 (1995).
96. Elliott, D., Stumm, M., Snelgrove, W. M., Cojocaru, C. & McKenzie, R. Computational RAM: implementing processors in memory. *IEEE Des. Test Comput.* **16**, 32–41 (1999).
97. Ankit, A., Sengupta, A., Panda, P. & Roy, K. RESPARC: a reconfigurable and energy-efficient architecture with memristive crossbars for deep spiking neural networks. In *Proc. 54th ACM/EDAC/IEEE Annual Design Automation Conf.* 63.2 (IEEE, 2017).
98. Bez, R. & Pirovano, A. Non-volatile memory technologies: emerging concepts and new materials. *Mater. Sci. Semicond. Process.* **7**, 349–355 (2004).
99. Xue, C. J. et al. Emerging non-volatile memories: opportunities and challenges. In *Proc. 9th Int. Conf. on Hardware/Software Codesign and System Synthesis* 325–334 (IEEE, 2011).
100. Wong, H.-S. P. & Salahuddin, S. Memory leads the way to better computing. *Nat. Nanotechnol.* **10**, 191 (2015); correction **10**, 660 (2015).
101. Chi, P. et al. Prime: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In *Proc. 43rd Int. Symp. Computer Architecture* 27–39 (IEEE, 2016).
102. Shafiee, A. et al. ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In *Proc. 43rd Int. Symp. Computer Architecture* 14–26 (IEEE, 2016).
103. Burr, G. W. et al. Neuromorphic computing using non-volatile memory. *Adv. Phys. X* **2**, 89–124 (2017).
104. Snider, G. S. Spike-timing-dependent learning in memristive nanodevices. In *Proc. Int. Symp. on Nanoscale Architectures* 85–92 (IEEE, 2008).
105. Chua, L. Memristor—the missing circuit element. *IEEE Trans. Circuit Theory* **18**, 507–519 (1971).
This was the first work to conceptualize memristors as fundamental passive circuit elements; they are currently being investigated as high-density storage devices through various emerging technologies for conventional general-purpose and neuromorphic computing architectures.
106. Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *Nature* **453**, 80–83 (2008).
107. Waser, R., Dittmann, R., Staikov, G. & Szot, K. Redox-based resistive switching memories—nanoionic mechanisms, prospects, and challenges. *Adv. Mater.* **21**, 2632–2663 (2009).
108. Burr, G. W. et al. Recent progress in phase-change memory technology. *IEEE J. Em. Sel. Top. Circuits Syst.* **6**, 146–162 (2016).
109. Hosomi, M. et al. A novel nonvolatile memory with spin torque transfer magnetization switching: spin-RAM. In *Int. Electron Devices Meeting* 459–462 (IEEE, 2005).
110. Ambrogio, S. et al. Statistical fluctuations in HfO_x resistive-switching memory. Part I—set/reset variability. *IEEE Trans. Electron Dev.* **61**, 2912–2919 (2014).
111. Fantini, A. et al. Intrinsic switching variability in HfO_2 RRAM. In *5th Int. Memory Workshop* 30–33 (IEEE, 2013).
112. Merrikh-Bayat, F. et al. High-performance mixed-signal neurocomputing with nanoscale floating-gate memory cell arrays. *IEEE Trans. Neural Netw. Learn. Syst.* **29**, 4782–4790 (2017).
113. Ramakrishnan, S., Hasler, P. E. & Gordon, C. Floating-gate synapses with spike-time-dependent plasticity. *IEEE Trans. Biomed. Circuits Syst.* **5**, 244–252 (2011).
114. Hasler, J. & Marr, H. B. Finding a roadmap to achieve large neuromorphic hardware systems. *Front. Neurosci.* **7**, 118 (2013).
115. Hasler, P. E., Diorio, C., Minch, B. A. & Mead, C. Single transistor learning synapses. In *Advances in Neural Information Processing Systems* Vol. 7 (eds Tesauro, G., Touretzky, D. S. & Leen, T. K.) 817–824 (Neural Information Processing Systems Foundation, 1995).
This was one of the first works to use a non-volatile memory device—specifically, a floating-gate transistor—as a synaptic element.
116. Holler, M., Tam, S., Castro, H. & Benson, R. An electrically trainable artificial neural network (ETANN) with 10240 'floating gate' synapses. In *Int. Joint Conf. on Neural Networks* Vol. 2, 191–196 (1989).
117. Chen, P.-Y. et al. Technology–design co-optimization of resistive cross-point array for accelerating learning algorithms on chip. In *Proc. Eur. Conf. on Design, Automation & Testing* 854–859 (IEEE, 2015).
118. Chakraborty, I., Roy, D. & Roy, K. Technology aware training in memristive neuromorphic systems for nonideal synaptic crossbars. *IEEE Trans. Em. Top. Comput. Intell.* **2**, 335–344 (2018).
119. Alibart, F., Gao, L., Hoskins, B. D. & Strukov, D. B. High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm. *Nanotechnology* **23**, 075201 (2012).
120. Dong, Q. et al. A 4 + 2T SRAM for searching and in-memory computing with 0.3-V V_{DDmin}. *IEEE J. Solid-State Circuits* **53**, 1006–1015 (2018).
121. Agrawal, A., Jaiswal, A., Lee, C. & Roy, K. X-SRAM: enabling in-memory Boolean computations in CMOS static random-access memories. *IEEE Trans. Circuits Syst. I* **65**, 4219–4232 (2018).
122. Eckert, C. et al. Neural cache: bit-serial in-cache acceleration of deep neural networks. In *Proc. 45th Ann. Int. Symp. Computer Architecture* 383–396 (IEEE, 2018).
123. Gonugondla, S. K., Kang, M. & Shanbhag, N. R. A variation-tolerant in-memory machine-learning classifier via on-chip training. *IEEE J. Solid-State Circuits* **53**, 3163–3173 (2018).
124. Biswas, A. & Chandrakasan, A. P. Conv-RAM: an energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications. In *Int. Solid-State Circuits Conf.* 488–490 (IEEE, 2018).
125. Kang, M., Keel, M.-S., Shanbhag, N. R., Eilert, S. & Curewitz, K. An energy-efficient VLSI architecture for pattern recognition via deep embedding of computation in SRAM. In *Int. Conf. on Acoustics, Speech and Signal Processing* 8326–8330 (IEEE, 2014).
126. Seshadri, V. et al. RowClone: fast and energy-efficient in-DRAM bulk data copy and initialization. In *Proc. 46th Ann. IEEE/ACM Int. Symp. Microarchitecture* 185–197 (ACM, 2013).
127. Prezioso, M. et al. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* **521**, 61–64 (2015).
128. Sebastian, A. et al. Temporal correlation detection using computational phase-change memory. *Nat. Commun.* **8**, 1115 (2017).
129. Jain, S., Ranjan, A., Roy, K. & Raghunathan, A. Computing in memory with spin-transfer torque magnetic RAM. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **26**, 470–483 (2018).
130. Jabri, M. & Flower, B. Weight perturbation: an optimal architecture and learning technique for analog VLSI feedforward and recurrent multilayer networks. *IEEE Trans. Neural Netw.* **3**, 154–157 (1992).
131. Diorio, C., Hasler, P., Minch, B. A. & Mead, C. A. A floating-gate MOS learning array with locally computed weight updates. *IEEE Trans. Electron Dev.* **44**, 2281–2289 (1997).
132. Bayat, F. M., Prezioso, M., Chakrabarti, B., Kataeva, I. & Strukov, D. Memristor-based perceptron classifier: increasing complexity and coping with imperfect hardware. In *Proc. 36th Int. Conf. on Computer-Aided Design* 549–554 (IEEE, 2017).
133. Guo, X. et al. Fast, energy-efficient, robust, and reproducible mixed-signal neuromorphic classifier based on embedded NOR flash memory technology. In *Int. Electron Devices Meeting* 6.5 (IEEE, 2017).
134. Liu, C., Hu, M., Strachan, J. P. & Li, H. Rescuing memristor-based neuromorphic design with high defects. In *Proc. 54th ACM/EDAC/IEEE Design Automation Conf.* 76.6 (IEEE, 2017).
135. Tuma, T., Pantazi, A., Le Gallo, M., Sebastian, A. & Eleftheriou, E. Stochastic phase-change neurons. *Nat. Nanotechnol.* **11**, 693–699 (2016).
136. Fukushima, A. et al. Spin dice: a scalable truly random number generator based on spintronics. *Appl. Phys. Express* **7**, 083001 (2014).
137. Le Gallo, M. et al. Mixed-precision in-memory computing. *Nature Electron.* **1**, 246 (2018).
138. Krstic, M., Grass, E., Gürkaynak, F. K. & Vivet, P. Globally asynchronous, locally synchronous circuits: overview and outlook. *IEEE Des. Test Comput.* **24**, 430–441 (2007).
139. Choi, H. et al. An electrically modifiable synapse array of resistive switching memory. *Nanotechnology* **20**, 345201 (2009).
140. Serrano-Gotarredona, T., Masquelier, T., Prodromakis, T., Indiveri, G. & Linares-Barranco, B. STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Front. Neurosci.* **7**, 2 (2013).
141. Kuzum, D., Jeyasingh, R. G., Lee, B. & Wong, H.-S. P. Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Lett.* **12**, 2179–2186 (2012).
142. Krzysteczko, P., Münchenberger, J., Schäfers, M., Reiss, G. & Thomas, A. The memristive magnetic tunnel junction as a nanoscopic synapse–neuron system. *Adv. Mater.* **24**, 762–766 (2012).
143. Vincent, A. F. et al. Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems. *IEEE Trans. Biomed. Circuits Syst.* **9**, 166–174 (2015).
144. Sengupta, A. & Roy, K. Encoding neural and synaptic functionalities in electron spin: a pathway to efficient neuromorphic computing. *Appl. Phys. Rev.* **4**, 041105 (2017).
145. Borghetti, J. et al. 'Memristive' switches enable 'stateful' logic operations via material implication. *Nature* **464**, 873–876 (2010).
146. Hu, M. et al. Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication. In *Proc. 53rd ACM/EDAC/IEEE Annual Design Automation Conf.* 21.1 (IEEE, 2016).
147. Sheridan, P. M. et al. Sparse coding with memristor networks. *Nat. Nanotechnol.* **12**, 784–789 (2017).
148. Wright, C. D., Liu, Y., Kohary, K. I., Aziz, M. M. & Hicken, R. J. Arithmetic and biologically-inspired computing using phase-change materials. *Adv. Mater.* **23**, 3408–3413 (2011).
149. Le Gallo, M., Sebastian, A., Cherubini, G., Giefers, H. & Eleftheriou, E. Compressed sensing recovery using computational memory. In *Int. Electron Devices Meeting* 28.3.1 (IEEE, 2017).
150. Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **65** 386 (1958).
151. Bi, G. Q. & Poo, M. M. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* **18**, 10464–10472 (1998).

Acknowledgements We thank A. Sengupta (Pennsylvania State University), A. Raychowdhury (Georgia Institute of Technology) and S. Gupta (Purdue University) for their input. The work was supported in part by the Center for Brain-inspired Computing Enabling Autonomous Intelligence (C-BRIC), a DARPA-sponsored JUMP center, the Semiconductor Research Corporation, the National Science Foundation, Intel Corporation, the DoD Vannevar Bush Fellowship, the ONR-MURI programme, and the US Army Research Laboratory and the UK Ministry of Defence under agreement number W911NF-16-3-0001.

Author contributions All authors contributed equally in devising the structure of the paper, designing the figures and writing the manuscript.

Competing interests The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.R.